

SYSTEM-ON-CHIP POWER CONSUMPTION REFINEMENT AND ANALYSIS

David Y. Feinstein, Mitchell A. Thornton and Fatih Kocan
Department of Computer Science and Engineering
Southern Methodist University
Dallas, TX, USA
{dfeinste,mitch,kocan}@engr.smu.edu

Abstract

Accurate power consumption estimation of a System-on-Chip (SoC) using modeling techniques is difficult due to the diverse mixture of processes with radically different current consumption. It is very important that these estimations will be fine tuned to the specific SoC with accurate current measurement during the design and prototyping phase. We introduce an accurate method to measure power consumption using a single measurement point and a dynamic logging algorithm. We present a demonstration tool for continuous logging of the instantaneous power consumption with an identification of the running process within the SoC. Our approach can also be used to steer the dynamic power management (DPM) of a SoC.

1. INTRODUCTION

The emergence of battery-operated System-on-Chip (SoC) in recent years increased the efforts to reduce the power consumption. Early works identified the need of minimizing the power consumption at the initial hardware/software co-design process [4,10]. More current works showed the crucial need for efficient power consumption simulation and estimation tools [2,7,8]. The increased complexity of modern SoC applications limits the capabilities of such simulation tools to predict the exact measured power consumption. In particular, the memory, the processing, and the analog components of the typical SoC have radically different power consumption profiles.

The ability of a SoC system to perform within its power budget is often achieved using dynamic power management (DPM) methods. DPM must employ a reliable real-time means for measuring the actual power consumption [1,5].

This paper discloses a new method to dynamically obtain accurate power consumption measurements of the individual sub-circuits and their associated processes using a *single measurement point* within the SoC architecture. The data obtained can be used to fine tune the SoC power simulation tools. When incorporated within the design methodology of SoC, our approach can pinpoint the hardware/software sub-components that require power consumption refining in order to meet the system's power budget. Our approach can be further integrated in the SoC design to augment its DPM method.

2. RELATED WORK

The general "power wall" problem that limits the Moore's law exponential growth of semiconductor density also affects all types of contemporary SoC systems [1,5]. Recent SoC power consumption reduction efforts employ various innovative approaches. Lackey et al. surveyed the Voltage Islands methods that reduce the active and static SoC power consumption [6]. They emphasized the need for accurate current measurement that does not require costly simulation-based switching vectors. Hilman described the Virtual-Silicon's VIP mobilized power management approach which is also based on power islands [5].

State-based power analysis for SoC was offered by Bergamaschi and Jiang [2]. They noted that the generic power consumption used for the power estimation can be very inaccurate if applied blindly without proper "tuning" to the target application. Lee et al. developed the **Power ViP** framework to provide cycle-accurate power estimation for SoC at the transaction level [7]. They took the component-based approach in order to achieve a fast and easy power model. Their work demonstrated the need for accurate actual power consumption verification based on experimental measurement.

Chandrakasan et al. [3] offered the following analysis of the main power consumption contributors in CMOS devices:

$$P_{total} = C_L V_{swing} V_{dd} f_{clk} + I_{sc} V_{dd} + I_{leak} V_{dd} \quad (1)$$

The first term is the dynamic power consumption. C_L is the circuit capacitance; V_{swing} is the voltage swing which is equal or less to the supply voltage V_{dd} . f_{clk} is the circuit clock frequency, or the switching rate of the circuits. Their paper covered traditional power consumption reduction techniques based on voltage scaling, frequency reduction and leakage current minimization.

Benini et al. surveyed several dynamic power management techniques based on system level considerations [1]. They also indicated the need for an accurate real-time current measurement to dynamically steer the DPM system.

The average current consumption can be easily measured by an ammeter connected between the power supply and the circuit. This method cannot capture the instantaneous power consumption of a SoC where different tasks are performed in a complex order, often for a fraction of a second. A common technique to determine the current consumption of a specific task using a similar arrangement runs the task in an endless loop [12].

Watanbe et al. investigated the use of pipeline task scheduling for power reduction while satisfying both the throughput and the latency constraints [11]. Actual power measurements to support such task scheduling are hard to achieve. The running tasks are often preempted by other tasks so that the measurements must gain insight into the real-time power consumption profile of the application.

This paper is motivated by such needs for accurate power consumption measurements that can improve the SoC simulation tools and improve the power consumption refinement within the design process.

3. OUR APPROACH

We propose a SoC design that allows an accurate real-time measurement and analysis of the power requirements of the various processes *using a single measurement point*. We integrate a low cost current measurement sensor into the SoC architecture and develop a dynamic protocol that determines the *individual power consumption of each process*. This allows the circuit designer to iteratively refine the power consumption of the design by identifying the processes that demand the most power and trace their behavior in the real-time target application. A SoC utilizing our approach provides accurate power consumption measurements to tune up power simulation tools like those developed by Bergamaschi and Jiang [2].

The sensor can often be readily implemented with very minimal cost using the existing resources of the SoC. Although the current measurement circuitry may be removed from the SoC after the design phase, it can be left in the design to help steer the DPM system if implemented in the SoC [1].

It is important to insure that the integrated current measurement process poses a minimal and *fixed* execution time overhead on the SoC. Similarly, the sensor itself and the SoC overhead in performing the current measurement should have a small and *fixed* current consumption overhead. Meeting these two constraints make it possible to “calibrate out” the effect of the measurement process from the overall measurements. We should note that typical existing SoC A/D channel resources include a built-in data averaging capability that minimizes the overhead.

Since the SoC runs numerous tasks that may preempt each other, it is crucial to identify the tasks that are associated with each power measurement. We offer the protocol shown in Fig. 1. The Current Log Routine periodically reads the average current consumption and logs this reading together with the time ticker. This process is repeated indefinitely based on a timer interrogation or other polling techniques.

Each process (or task) of the SoC is assigned a unique ID number. Whenever a process is called, its ID number and the time ticker are stored as an entry signature. The process ID number and time ticker are logged again at the end of the process to create an exit signature. The entry and exit signatures *do not include current reading* in order to keep the current measurement overhead low. The resulting power consumption data log is analyzed off-line to determine the power consumption of each process. This is performed by analyzing the process entry and exit signatures, revealing the interaction and preemption among the processes.

It is often true that the SoC has too many fast processes that are difficult to track within the time resolution of our protocol. The designer needs to define which process should be ignored based on the given time resolution. While such processes still

may interfere with the measurements, their combined effect can be “calibrated out” by observing the log of a given process along an extended path of repeated executions.

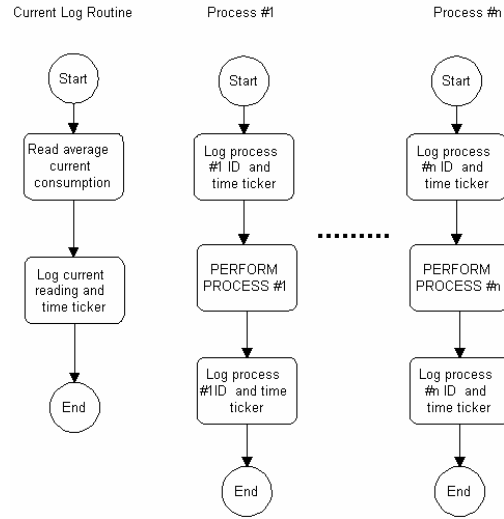


Figure 1: Data Logging of Multiple Processes

4. THE SoC DEMONSTRATION TOOL

We present a demonstration tool of an emulated SoC in which the voltage across a *single* small series resistor R_s at the power input is measured by a high-side current sense amplifier (Maxim Semiconductor MAX4172 [9]). We have emulated a typical SoC design using a common mixed signal microcontroller (C8051F321) combined with the external functional circuitry shown in Fig. 2. The design allows the user to control the access rate of the DRAM memory devices to demonstrate how the reduction in memory access rate reduces the current consumption. The entire circuitry can be readily integrated within a SoC. As mentioned, we use an A/D converter channel that features extensive averaging capabilities. Therefore, it is sufficient for the system to read the current average 10-50 times per second in order to create a detailed real-time profile of the current consumption.

Our SoC emulator board is interfaced to a custom graphic program running on a PC. The PC obtains a log of the instantaneous overall power consumption of the SoC in order to demonstrate the clear correlation between the active process and the measured power consumption. Our tool can demonstrate how current consumption is affected in real-time when the SoC performs different internal operations like DRAM accesses, SRAM accesses, or LED (fixed current consumption) activities. The user can setup the test flow for the desired mixture of activities to be performed at different durations.

The dynamic power consumption of P_{total} in equation (1) is demonstrated during the DRAM access when the user changes the SoC frequency. The LED loads provide an example of a process with fixed power consumption.

While this SoC emulator uses a single voltage power source, modern SoC designs call for multiple voltage sources as well as voltage scaling for power reduction. Our approach can still work in such systems with one measurement point at the main power entry to the SoC.

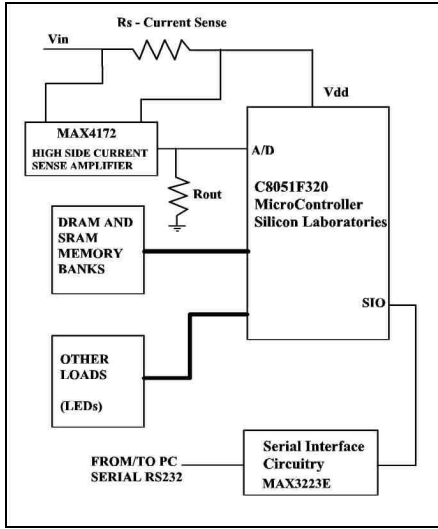


Figure 2: A SoC emulator utilizing our Power Measurement Technique

It is important to note that our approach is intended to be implemented within the framework of a power consumption refining paradigm without real-time PC interface. The SoC simply keeps a log of the continuous current measurements together with the corresponding ID markers for off-line analysis.

5. EXPERIMENTAL RESULTS

Since standard benchmarks for our approach are not yet established, we have captured the current consumption logs shown in Fig. 3 while running radically different test flow settings. In Fig. 3.a we switch various resistive and memory loads that run for relatively long periods. This setup produces almost step function changes in the overall instantaneous current consumption. In Fig. 3.b we have reduced the duration of the test periods to illustrate how our approach provides continuously report of the power consumption. Fig. 3.c illustrates the lower DRAM power consumption when using reduced board frequency.

Table 1 shows the typical current consumption obtained with our tool using different process mixtures. In the first line we have listed the base-line current which is obtained when turning off all the processes. The 35mA reading is therefore the current consumption overhead of the SoC effort in communicating with the PC, running the current measurement process, and performing other “standby” functions. This value is subtracted from all the subsequent readings of Table 1 to obtain the net process currents. Thus the second row shows the 22mA net current reading for the LED (fixed load) obtained by subtracting the base line current of 35mA from the 57mA overall reading. Similarly we obtain the net process current in rows 3 and 4 for the DRAM only (33mA) and SRAM only (16mA) processes. In line 5 we calculate the net current for the DRAM + SRAM + LED process to be 75mA. This reading slightly deviates (6%) from the expected sum of 71mA when the processes run separately (lines 2-4).

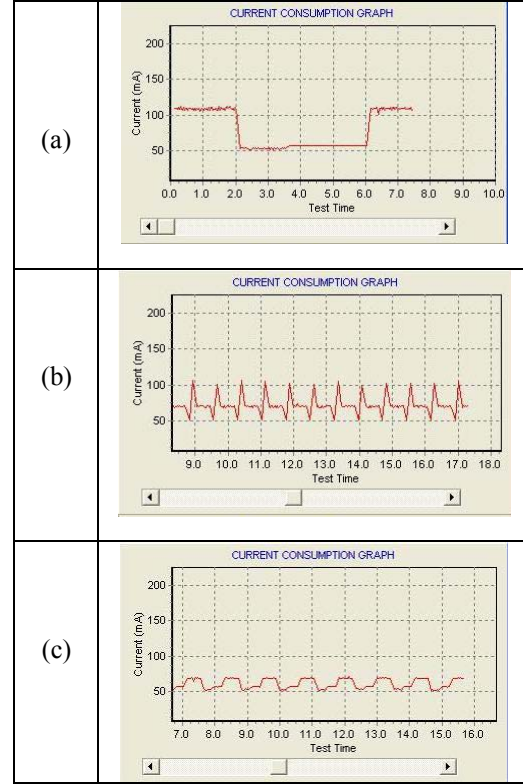


Figure 3: Running Different Test Flows

Table 1: Computing the Current Consumption of the Individual Processes

Process mixture setting	Overall current consumption	Net process current
SoC base-line	35 mA	--
LED only	57 mA	22 mA
DRAM only	68 mA	33 mA
SRAM only	51 mA	16 mA
DRAM + SRAM + LED	110 mA	75 mA

Table 2 summarizes the performance of our development system using the test flows of Fig. 3. In addition to the maximum and minimum currents taken from the scrolling power consumption graph, we analyzed the performance with various polling rates. The slowest polling rate is defined as the rate at which the SoC measurements start to average out the current consumption differences among the processes. The fastest polling rate (2mS) indicates the limit of our demonstration system, taking into account the SoC emulator’s computing power and the limitation on the serial channel interface.

Fig. 3 and Table 2 illustrate the viability of our approach to correlate power consumption data to the actual processes at a relatively high resolution. Even the low overhead of 30mS polling allows a detailed analysis for all the three different flows in Fig. 3. Since the PC was a fast 3GHz dual Xeon computer, the overhead at the PC side is negligible for the current tests.

Table 2: Performance results with different test flows.

Test Flow	Max Current	Min Current	Slowest Polling	Fastest Polling
Fig. 3a	110mA	53mA	250mS	2mS
Fig. 3b	100mA	52mA	60mS	2mS
Fig. 3c	70mA	50mA	150mS	2mS

Table 3 illustrates the power consumption data log in an SoC designed in accordance with Fig. 1 for off-line analysis. We show the data log during an arbitrary period from 100mS to 200mS and demonstrate how to resolve the power consumption of each process when multiple processes are active together or when preempting occurs. The ID number of the continuous current log routine is set to 0 and it is called every 10ms. We demonstrate two processes having ID numbers 1 and 2 (process ID#2 consumes more power than process ID#1). Process ID#1 starts at time 112ms, gradually increasing the current consumption to 75mA. Process ID#2 starts at time 136ms, while process ID#1 is still on. In this example both processes run together, quickly bringing their average power consumption to 129mA. At time 167ms process ID#1 terminates, leaving process ID#2 working along at its higher current consumption of 240mA. As process ID#2 ends at time 185ms, the consumption gradually (due to the effect of the filter capacitor) goes back to the 43mA base-line consumption. Each process may exhibit inherent, inter-process related, power fluctuations.

Table 3: Demonstration of Offloading Power Consumption Data

Process ID	Time Ticker	Current Consumption
...
0	100	35
0	110	40
1	112	--
0	120	73
0	130	75
2	136	--
0	140	121
0	150	129
0	160	113
1	167	--
0	170	218
0	180	240
2	185	--
0	190	180
0	200	43
...

Our technique does not severely interfere with the process mixture and order in a typical SoC application. Each process logs its entry and exit point in the log, to support multiple simultaneous processes and inter-process preemption. Table 3 further demonstrates that our power consumption data log requires minimal storage area. The entire log is downloaded to the SoC design framework for detailed power consumption analysis. The automated analysis allows the designer to identify which portion of the design's power budget is consumed by each process and determines which process needs to be further refined. The data is also used to fine tune the power simulation tools in the SoC design framework.

6. CONCLUSIONS

We suggest a simple enhancement for the SoC design that enables the designer to obtain a detailed insight into the power

consumption of each component and related process. Such detailed information can be used to fine tune and validate the power simulation tools within the design framework. The detailed power consumption data can be advantageously used in the power design refining process, identifying those hardware/software components of the SoC design that need special attention to meet the design power budget.

We have developed an efficient logging algorithm that achieves detailed power consumption measurements for each process using a *single* current sensor. The sensor measures the instantaneous current consumption at the power entry point to the SoC. It requires minimal resources that are abundantly available in the SoC with minimal cost implication.

We have demonstrated our approach with a development tool that emulates a SoC in conjunction with a PC interface. Using the internal averaging feature of the typical A/D channels of the SoC, the execution time overhead for the detailed power consumption measurements was minimal.

Our approach may be implemented in a DPM environment to provide real-time power consumption data to better steer the adaptive power control algorithms.

References

- [1] L. Benini, A. Bogliolo, and G. De Micheli, "A Survey of Design Techniques for System-Level Dynamic Power Management", *IEEE Trans. VLSI Systems*, vol. 8, no. 3 (June 2000): 299-316.
- [2] R. A. Bergamaschi, Y. W. Jiang, "State-based power analysis for systems-on-chip", In *DAC '03*, June 2003, pp. 638-641.
- [3] A.P. Chandrakasan, S. Sheng, and R.W. Brodersen, "Low Power CMOS Design", *IEEE J. Solid-State Circuits*, Vol. 27, No. 4, April 1992, pp. 473-483.
- [4] W. Fornaciari, P. Gubian, D. Sciuto, and C. Silvano, "Power Estimation of Embedded Systems: A Hardware/Software Codesign Approach", *IEEE Tran. on VLSI Systems*, vol. 6, no. (June 1998): pp. 266-275.
- [5] D. Hillman, "Using Mobilize Power Management IP for Dynamic & Static Power Reduction in SoC at 130 nm", In *DATE '05*, March 2005, 7 pages.
- [6] D. E. Lackey, P. S. Zuchowski, T. R. Bednar, D. W. Stout, S. W. Gould and J. M. Cohn, "Managing power and performance for System-on-Chip designs using Voltage Islands", In *ICCAD '02*, Nov. 2002, pp. 195-202.
- [7] I. Lee, H. Kim, P. Yang, S. Yoo, E.-Y. Chung, K.-M. Choi, J.-T. Kong and S.-K. Eo, "PowerViP: Soc power estimation framework at transaction level", In *ASP-DAC '06*, Jan. 2006, pp. 551-558.
- [8] Y. Li and J. Henkel, "A Framework for Estimating and Minimizing Energy Dissipation of Embedded HW/SW Systems", In *DAC'98*, 1998, pp. 188-194.
- [9] "MAX4172 Low-Cost, Precision, High-Side Current-Sense Amplifier", Maxim Integrated Products, Inc. 120 San Gabriel Drive, Sunnyvale, CA 94086 Tel. 408-737-7600. Available online: <http://www.maxim-ic.com/company/>
- [10] V. Tiwari, S. Malik, and A. Wolfe, "Power Analysis of Embedded Software: A First Step Towards Software Power Minimization", *IEEE Tran. VLSI Systems*, vol. 2, no. 4 (Dec. 1994) pp. 437-445.
- [11] R. Watanabe, M. Kondo, M. Imai, H. Nakamura and T. Nanya "Task scheduling under performance constraints for reducing the energy consumption of the GALs multi-processor SoC", In *DATE '07*, April 2007, pp. 797-802.
- [12] Wayne Wolf, *Computers as Components*, Morgan Kaufmann, 2001.