Implementing the LEON3 Statistics Unit in 28nm FD-SOI: Power Estimation by Activity Proxy

Martin Cochet, Guillaume Bonnechere, Jean-Marc Daveau[*],
Fady Abouzeid and Philippe Roche

STMicroelectronics, 850 Rue Jean Monnet, Crolles, France

Power estimation of a complex circuit such as a processor is an important but complicated task. The power information is important either for accurate power budgeting at design time or for energy optimization algorithms at run time. However, full simulations are slow and direct analog power monitors are complicated to integrate on-chip. Another approach consists in *power estimation by activity as a proxy*: it does not rely on the direct measurement of the physical voltage and current quantities; it rather proceeds in estimating the circuit digital activity, and based on known specific ASIC topology and process characteristics, estimate the power consumption. This monitoring is flexible and directly implementable at RTL level.

This research note first describes the principle of power estimation by activity proxy and gives a literature overview. Then it details its implementation in a LEON3 processor, taking advantage of its existing L3Stat statistics unit. Quantitative accuracy of the model is presented, showing between 1.5% and 2.1% average error across different testbenches.

1 Power modeling and previous work

This method of power monitoring has been reported in the literature since 2001 [1]. The typical method proposed is to weigh linearly with coefficients c_i the result of different activity counters A_i over a time duration of N cycles $T = N.T_{clk}$ to estimate the dynamic energy per cycle $E_{dyn/op}$, and the total energy per cycle[1] E_{op}, accounting for leakage P_{leak}:

$$E_{dyn/op} = \sum_i c_i.A_i \tag{1}$$

$$E_{op} = E_{dyn/op} + P_{leak}.T_{clk} \tag{2}$$

[*] jean-marc.daveau@st.com

[1] This energy is proportional to the dissipated power, at a fixed frequency, but is distinct from the energy per processor instruction, as some instructions execute in a different number of cycles.

In Eq.1 and 2, each activity A_i corresponds to the ratio of the number of events i counted by the monitor, with the number of clock cycles N and the leakage P_{leak} is a model parameter. The energy per cycle and the total consumed energy over the measurement time can then also easily be derived. The main challenge is to find the most pertinent signals to monitor. The signals need to be relevant and in a large enough number to cover power estimation for varied activity profiles, but should not result in too much implementation overhead, or complicated calibration step to determine the coefficients c_i. Note that the coefficients will depend on the processor architecture and implementation, but should be generic enough to cover a wide range of application code running on the processor. There are two main purposes of this estimation: power simulations and in-situ power monitoring.

The first approach is used to offer pre-silicon design estimate of the power consumption of complex circuits on specific testbenches. In theory this can be done by running RTL or gate level simulations, extracting all of the nets activity and using a CAD tool such as Synopsys Prime Power (PT-PX) to compute power based on the standard cells library model. This approach is straightforward but highly impractical for large ASICs and/or very long testbenches. This is why a FPGA emulation method has been proposed to offer significant speedups in power estimates on new architectures. The FPGA emulator provides a very quick measurement of the activity value at the different nets A_i from which the power can be reconstructed using Eq. 1. [2–4] reported 9-10% estimated accuracies with a 35x-100x speedup via FPGA emulation compared with gate-level simulations.

The approach we will be most interested in and discuss further is the use of activity counters for on-chip run-time power estimation. This method has been discussed in details in [5] and is now implemented in several processors [6, 7] reporting errors in the order of 6-10%. The instrumentation and measured performance presented in [6], is illustrated in Fig. 1. Some higher level models are based on non-linear aggregation of the measured parameters as proposed in [8].

As the power estimation method can be significantly architecture, library and technology dependent, a study of these existing methods is important. Another practical reason for this study is that most of the literature doesn't report on the exact counters used in their methodology.

2 LEON3 processor case study

A study is proposed to instrument the LEON3 [9] SPARC V8 [10] processor, commonly used for low power [11] and radiation reliability [12] projects within STMicroelectronics. This processor is a good candidate because its release v3.3 includes a statistics unit. Its open source nature also simplifies

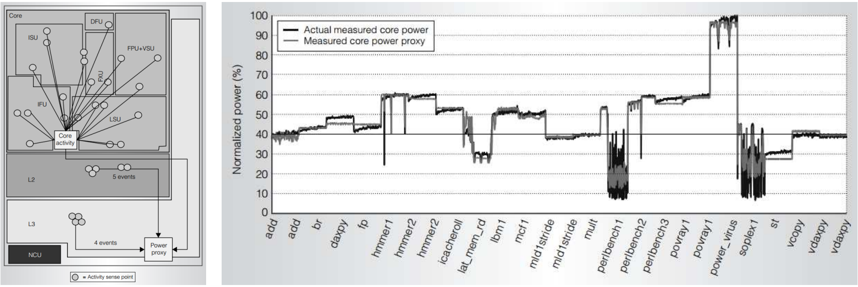


Figure 1: Power proxy strategy (left) and power estimate (right) of IBM POWER7 processor, adapted from [6]

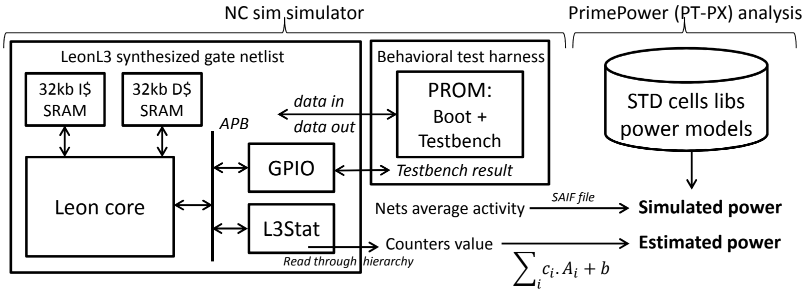


Figure 2: LEON3 implementation and simulation methodology for power estimation by proxy

the simulation process and makes reproducibility and future silicon implementations more straightforward.

For a simple implementation as a proof of concept, the LEON3 was configured in the following way: integer unit only, 4kB instruction and data cache in SRAM, 8 windows inferred register file. The statistics unit (L3Stat) is defined as a peripheral in the LEON3 library. Both the L3Stat and the GPIO are connected to the processor Advanced Peripheral Bus (APB). The processor was mapped on the 28 nm Low Voltage Threshold (LVT) Fully Depleted Silicon On Insulator (FD-SOI) standard cell library. The different tests were performed on the TT 0.9 V 25 °C no body-biasing Process, Voltage and Temperature (PVT) corner. In this context of preliminary study, the simulations were performed on the post-synthesis, pre-P&R gate level netlist. The results will be slightly impacted by the missing clock-tree power, but still accurate enough to validate the methodology, while saving the time dedicated to floorplaning and P&R steps. A block diagram of the proposed implementation and simulation methodology is presented in Fig. 2.

We considered an example of low power application, with an operating frequency of 25 MHz[2]. A sampling time of $N = 1000$ cycles was chosen,

[2] This result can directly be scaled to other frequencies, as the dynamic energy $E_{dyn/op}$

Table 1: Simulated leakage power across the different testbenches

Testbench	Idle	Dhrystone	Basestation	Codebook	Kalman	Atkinsieve
P_{leak} [µW]	68.44	68.63	68.59	68.37	68.50	68.59

which corresponds to a conversion time of 40 µs, which is compatible with typical power management strategies. The value of the counters is read directly by the behavioral testbench through the hierarchy, so that there is no overhead in the measurement process, which could affect the power simulations.

A set of five typical testbenches was used to evaluate the power management strategy: *Dhrystone* (Synthetic reference testbench [13]), *Basestation* (network simulation), *Codebook* (data encoding), *Kalman* (Kalman filtering [14] computation) and *Atkinsieve* (prime number search based on [15]), as well as a *Idle* testbench, consisting only of a wait operation, executing no operation (NOP) instructions.

3 Simulation Results

The first parameter to estimate in Eq. 2 is the leakage power P_{leak}. Simulation results in table 1 show that the leakage power is application independent ($< 0.3\%$ variation), so it can be considered as a constant[3] in the model.

Then, to estimate the dynamic energy $E_{dyn/op}$ from Eq. 1 based on the L3Stat counters readings, the activity terms A_i must be computed, the value of the coefficients c_i must be chosen and the most relevant subset of counters identified.

Table 2 presents the full list of counters available form the L3Stat, after a pruning of those resulting in a 0 activity for all of the considered testbenches. Based on those absolute counters readings, the activity is estimated as $A_i = D_i/D_1$, where the D_i is the absolute integer reading of the counter i and D_1 gives the number of core clock cycles in the measurement window ("Execution time").

Then, the most relevant counters and their value c_i must be chosen. This process intends to find the smallest subset of counters giving significant power estimation, and which additional ones to choose whenever a more refined estimation is required. For this process, a linear regression is performed on a simulated dataset including the 6 testbenches previously described[4]. The linear regression was first run on all 14 counters, resulting in

is independent of the clock frequency, and the leakage energy per operation is equal to $P_{leak}.T_{clk}$, i.e. is inversely proportional to the frequency

[3] This analysis, and the rest of the derivations proposed in this section only consider a fixed PVT condition, and would need to be further parametrized to account for PVT changes.

[4] Note that the coefficient c_1 actually corresponds to a constant b, as the activity A_1

Table 2: List of the 14 activity counters selected

1	Execution time	8	Instruction cache hold
2	Regular type 2 instr.	9	STORE instructions
3	LOAD instructions	10	Data cache (read) miss
4	Integer instructions	11	Data write buffer hold
5	Branch prediction miss	12	Data cache hold
6	LOAD and STORE inst.	13	CALL instructions
7	Instruction cache miss	14	Integer branches

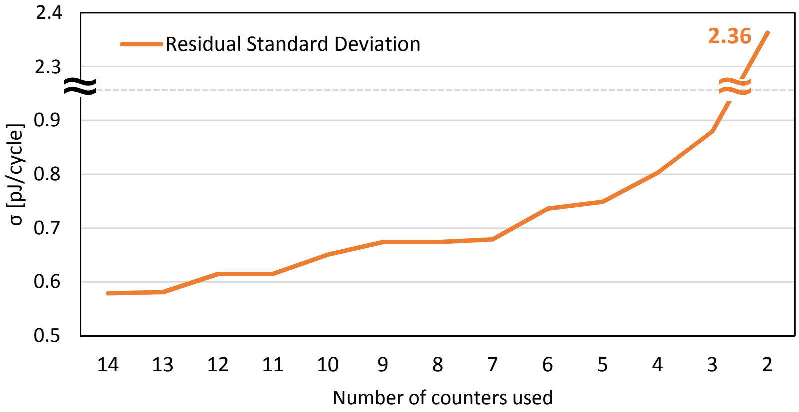
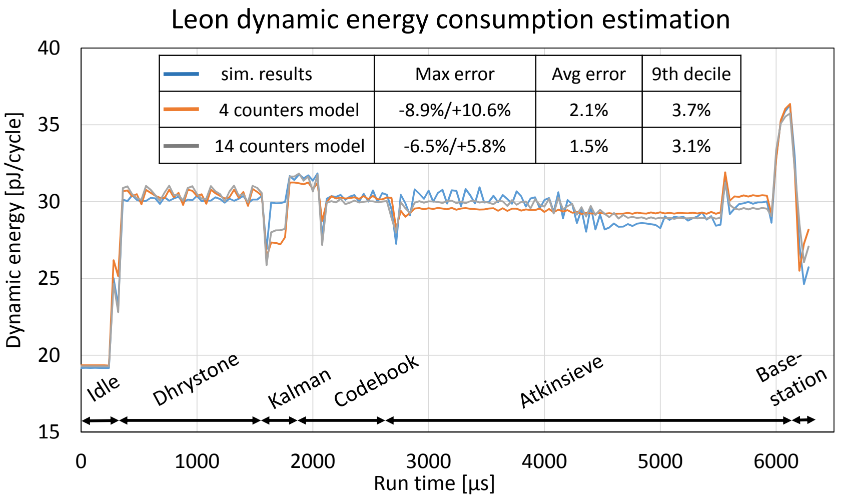


Figure 3: Proxy model error standard deviation change with the number of activity counters used.

an error of standard deviation $\sigma_{14} = 0.579pJ/cycle$. Then, the counter with the smallest contribution $|c_i. < A_i > |$ is removed from the list and a new regression is performed, resulting in a larger σ_{13} value. The process is iterated and the resulting σ_i values are reported in Fig. 3. Down to 4 total counters, the results are not too degraded, 3 counters results in a 52% increase in error vs. 14 coutners. Last, two counters give an unsatisfying estimate with a 2.36 pJ/cycle error, close to the error obtained with a single counter c_1 (i.e. a constant power approximation, which leads to 2.66 pJ/cycle error standard deviation). Table 2 counters index had already been ordered based on this process: the subset of i counters resulting in σ_i is the set of the counters numbered 1 to i of the table.

Last, to illustrate the power estimation results, the dynamic energy per operation of the LEON3 processor is plotted over time across different testbenches in Fig. 4, comparing the power estimate directly from simulations,

is always equal to 1 (one clock cycle event every clock cycle). This constant can be interpreted as the baseline energy used every cycle independently of the instruction that is run, e.g. to fetch and decode the instruction, to increment the address...



sim. results	Max error	Avg error	9th decile
4 counters model	-8.9%/+10.6%	2.1%	3.7%
14 counters model	-6.5%/+5.8%	1.5%	3.1%

Figure 4: Simulated LEON3 power consumption compared with 4 counters and 14 counters based power proxy.

and from 4 counters and 14 counters proxys. The inset table quantifies the energy estimate error in a more natural way than through residual standard deviation only. The 14 counters model offers the best accuracy, but the 4 counters based model still guarantees an accuracy within ±3.7% for 90% of the samples, even with energy variations of more than a factor 2 across testbenches. This result is actually better than the ±10% reported in [6]. However the IBM Power7 processor from [6] is a much more complex processor and a larger coverage of test cases (26 vs 6).

Last, based on gate-netlist implementation and power simulations, the power and area overhead of the L3Stat unit are of only 0.3% and 2.9% respectively.

Conclusion

As demonstrated in this study, the use of power monitoring by proxy is a promising method to estimate in-situ power consumption, and our preliminary studies confirm its potential on two different CPU architectures. Next, an important simulation and silicon measurement work is still needed to increase testbenches coverage, study the impact of PVT variations on the estimation model and to integrate the full method as part of a SoC power management strategy.

References

[1] R. Joseph and M. Martonosi, “Run-time power estimation in high performance microprocessors,” in *Low Power Electronics and Design, International Symposium on, 2001.*, 2001, pp. 135–140.

[2] A. Bhattacharjee, G. Contreras, and M. Martonosi, “Full-system chip multiprocessor power evaluations using FPGA-based emulation,” in *Low Power Electronics and Design (ISLPED), 2008 ACM/IEEE International Symposium on*, Aug 2008, pp. 335–340.

[3] C. Berthet, P. Georgelin, J. Ntyame, and M. Raffin, “Peak power estimation using activity measured on emulator,” in *Electronics, Circuits and Systems (ICECS), 2012 19th IEEE International Conference on*, Dec 2012, pp. 440–443.

[4] S. Hesselbarth, T. Baumgart, and H. Blume, “Hardware-assisted power estimation for design-stage processors using FPGA emulation,” in *Power and Timing Modeling, Optimization and Simulation (PATMOS), 2014 24th International Workshop on*, Sept 2014, pp. 1–8.

[5] W. Huang, C. Lefurgy, W. Kuk, A. Buyuktosunoglu, M. Floyd, K. Rajamani, M. Allen-Ware, and B. Brock, “Accurate fine-grained processor power proxies,” in *2012 45th Annual IEEE/ACM International Symposium on Microarchitecture*, Dec 2012, pp. 224–234.

[6] M. Floyd, M. Allen-Ware, K. Rajamani, B. Brock, C. Lefurgy, A. J. Drake, L. Pesantez, T. Gloekler, J. A. Tierno, P. Bose, and A. Buyuktosunoglu, “Introducing the adaptive energy management features of the Power7 chip,” *IEEE Micro*, vol. 31, no. 2, pp. 60–75, March 2011.

[7] V. Krishnaswamy, J. Brooks, G. Konstadinidis, C. McAllister, H. Pham, S. Turullols, J. L. Shin, Y. YangGong, and H. Zhang, “Fine-grained adaptive power management of the SPARC M7 processor,” in *2015 IEEE International Solid-State Circuits Conference - (ISSCC) Digest of Technical Papers*, Feb 2015, pp. 1–3.

[8] M. Yasin, A. Shahrour, and I. A. M. Elfadel, “Unified, ultra compact, quadratic power proxies for multi-core processors,” in *2014 Design, Automation Test in Europe Conference Exhibition (DATE)*, March 2014, pp. 1–4.

[9] Leon3 processor. Aeroflex Gaisler. [Online]. Available: http://www.gaisler.com/index.php/products/processors/leon3

[10] Scalable processor architecture. SPARC International Inc. [Online]. Available: http://sparc.org/

[11] S. Clerc, M. Saligane, F. Abouzeid, M. Cochet, J.-M. Daveau, C. Bottoni, D. Bol, J. De-Vos, D. Zamora, B. Coeffic, D. Soussan, D. Croain, M. Naceur, P. Schamberger, P. Roche, and D. Sylvester, “A 0.33V/-40

C process/temperature closed-loop compensation SoC embedding all-digital clock multiplier and DC-DC converter exploiting FDSOI 28nm back-gate biasing," in *Solid- State Circuits Conference - (ISSCC), 2015 IEEE International*, Feb 2015, pp. 1–3.

[12] C. Bottoni, B. Coeffic, J. M. Daveau, L. Naviner, and P. Roche, "Partial triplication of a SPARC-V8 microprocessor using fault injection," in *Circuits Systems (LASCAS), 2015 IEEE 6th Latin American Symposium on*, Feb 2015, pp. 1–4.

[13] A. R. Weiss, "Dhrystone benchmark: History, analysis, "scores" and recommendations," EEMBC Certification Laboratories, LLC, Tech. Rep., 2002. [Online]. Available: http://www.johnloomis.org/NiosII/dhrystone/ECLDhrystoneWhitePaper.pdf

[14] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[15] A. O. L. Atkin and D. J. Bernstein, "Prime sieves using binary quadratic forms," *Mathematics of Computation*, vol. 73, no. 246, pp. 1023–1030, 2004.