Cycle Accurate Energy and Throughput Estimation for Data Cache

Muhammad Yasir Qadri*, Klaus D. McDonald-Maier†
School of Computer Science and Electronic Engineering
University of Essex, CO4 3SQ, UK
*Email: yasirqadri@acm.org *, kdm@essex.ac.uk †*

Abstract

Resource optimization in energy constrained real-time adaptive embedded systems highly depends on accurate energy and throughput estimates of processor peripherals. Such applications require lightweight, accurate mathematical models to profile energy and timing requirements on the go. This paper presents enhanced mathematical models for data cache energy and throughput estimation. The energy and throughput models were found to be within 95% accuracy of per instruction energy model of a processor, and a full system simulator's timing model respectively. Furthermore, the possible application of these models in various scenarios is discussed in this paper.

1. Introduction

Cache structures are in widespread use in embedded processors and can have a significant impact on processor timing and energy consumption. Hence, it becomes increasingly important to be able to evaluate the impact of cache. For battery powered real-time adaptive systems allowing runtime reconfiguration of cache memory based on numerical analysis, such models need to be accurate and require minimum computational power. Basic models for energy and throughput analysis were previously presented in [1, 2], which have been further enhanced in this paper. The remainder of this paper is divided into five sections. In the following section related work is discussed. The energy and throughput models are introduced in section 3. In the fourth and fifth sections the models are validated and example applications are discussed, and the final section forms the conclusion.

2. Related Work

In this section related research in the areas of cache energy estimation, timing models and tools such as full system simulators and virtual platforms are discussed.

Simunic et al. [3] presented cycle accurate mathematical models for energy analysis in embedded systems. The per cycle energy model presented in their work comprises processor, interconnects, memory, DC-to-DC converters and level two (L2) cache energy components. The model was validated using SmartBadge [4] prototype based on ARM-1100 processor and it was found to be within 5% of the hardware measurements for the same operating frequency. The models holistically analyze the processor power and do not furnish an estimate on individual components such as level one (L1) cache, on-chip memory, etc inside a processor. Kamble et al. [5, 6] presented detailed bit-level mathematical models for SRAM cache energy consumption analysis and propose some architectural techniques to reduce power dissipation. Li et al. [7] proposed a full system detailed energy model comprising cache, main memory, and software energy components. Their work also includes description of an Avalanche framework for estimating and optimizing energy dissipation of embedded systems. In [8] the authors introduce a hardware/software fine-grained (instruction/operation-level) partitioning approach to address low-power optimizations used in the Avalanche system. Tiwari et al. [9] presented an instruction level energy model comprising energy consumption in individual pipeline stages. The model was evaluated by executing benchmarks for inter-instruction effects by measuring the current flow in the processor. The same methodology was applied in [10] with inclusion of the effects of cache enabling and disabling. Balasubramonian et al. [11] propose a cache and Translation Lookaside Buffer (TLB) design that allows dynamic configurability trading off size and speed on a per application basis. Their work also includes a novel configuration management algorithm based on application hit/miss rate information to reconfigure cache to improve performance while taking energy consumption into consideration.

Wada et al. [12] presented detailed circuit level analytical access time model for on-chip cache memories. The model takes inputs such as (N_{dwl}, N_{dbl}, N_{twl}, and N_{tbl}) number of tag/data array per word/bit line etc. On comparing with SPICE results the model gives 20% error for an 8ns access time cache memory. Taha et al. [13] presented an instruction throughput model of Superscalar processors. The main parameters of the model are

superscalar width of the processor, pipeline depth, instruction fetch mechanism, branch predictor, central issue window width, number of functional units their latencies and throughputs, re-order buffer width and cache size and latency etc. The model results in errors up to 5.5% as compared to the SimpleScalar out-of-order simulator.

Virtual platforms and full system simulators provide an alternate to gauge the energy and timing performance of an embedded system. Magnusson et al. [14] presented SIMICS, a full system simulation tool. It simulates processors at the instruction-set level and supports to run unmodified OS such as VxWorks, Solaris, Linux, Tru64, and Windows XP virtually on the target platforms. The simulator is targeted to provide fairly accurate timing profile, but currently does not support energy profiling of the target platforms. Austin et al. [15] presented SimpleScalar another full system virtualization platform. The SimpleScalar models use an execution-driven simulation technique that reproduces a device's internal operation. An alternate to that is trace-driven simulation, which employs a stream of pre-recorded instructions to drive a hardware timing model. The execution-driven simulation approach provides access to all data transactions during program execution, which is primary requirement for dynamic power analysis. Utilizing SimpleScalar interface, Brooks et al. [16] designed a tool named Wattch for architectural level power analysis. It maintains accuracy within 10% as compared to results of circuit level power analysis tools. A further extension of this work was made by Flores et al. [17] by proposing Sim-PowerCMP for chip multiprocessors. The tool estimates both dynamic and leakage power for CMP architectures based on a Linux x86 model of RSIM [18]. Sim-PowerCMP features power models for dynamic power from Wattch [16], leakage power from HotLeakage [19], and the interconnection network from Orion [20] simulators.

The following section presents simple D-cache energy and throughput models which provide the results based on per application basis along with fair accuracy and less computation power requirement.

3. D-Cache energy and throughput models

The D-cache energy and throughput models are illustrated in Fig. 1. According to these models the energy or time required for an application to execute could be classified into two principal components i.e. the contribution of cache and of non-cache elements.

If E_{total} [J] is the total energy required to execute an application then the cache components will be E_{read} the energy consumed in cache read accesses [J], E_{write} the energy consumed in cache write accesses [J], $E_{c\rightarrow m}$ the energy consumed in cache to memory accesses [J], and E_{mp} the energy miss penalty [J]. The non-cache components are E_{leak} the leakage energy of the device [J], and E_{misc} the energy consumed in execution of other instructions which do not require data memory access [J].

$$\mathbf{E_{total} = E_{read} + E_{write} + E_{c\rightarrow m} + E_{mp} + E_{leak} + E_{misc}.} \quad (1)$$

The individual components can be further defined as

$$\mathbf{E_{read} = n_{read}.\, E_{dyn.read}.\left[1 + \frac{read_{mr}}{100}\right],} \quad (2)$$

$$\mathbf{E_{write} = n_{write}.\, E_{dyn.write}.\left[1 + \frac{write_{mr}}{100}\right],} \quad (3)$$

$$\mathbf{E_{c\rightarrow m} = E_{m}.\,(n_{read} + n_{write}).\left[1 + \frac{total_{mr}}{100}\right],} \quad (4)$$

$$\mathbf{E_{mp} = E_{idle}.\,(n_{read} + n_{write}).\left[P_{miss}.\frac{total_{mr}}{100}\right], \text{ and}} \quad (5)$$

$$\mathbf{E_{leak} = P_{leak}.\, t_{idle},} \quad (6)$$

where n_{read} is the total number of cache read accesses, n_{write} is the total number of write accesses, $E_{dyn.read}$ is the energy consumed per cache read access [J], $E_{dyn.write}$ is the energy consumed per write access [J], E_m is the energy consumed per data memory access [J], E_{idle} is the per cycle idle mode energy consumption of the processor [J], P_{leak} is the leakage power of the processor, t_{idle} is the time when processor was in idle state, $read_{mr}$, $write_{mr}$, and $total_{mr}$ are the read, write and total miss rate (in percentage) and P_{miss} is the miss penalty in number of stall cycles.

Similarly, if T_{total} is the total time taken by an application [Sec], then it could be expressed as the sum of t_{cache} the time taken by cache operations [Sec], t_{ins} the time taken in execution of cache access instructions [Sec], t_{mp} the time miss penalty [Sec] and t_{misc} the time while executing other instructions i.e. which do not require data memory access [Sec].

$$\mathbf{T_{total} = t_{cache} + t_{ins} + t_{mp} + t_{misc}.} \quad (7)$$

Furthermore,

$$\mathbf{t_{cache} = t_{c}.\,(n_{read} + n_{write}).\left[1 + \frac{total_{mr}}{100}\right],} \quad (8)$$

$$\mathbf{t_{ins} = (t_{cycle} - t_{c}).\,(n_{read} + n_{write}) \text{ and}} \quad (9)$$

$$\mathbf{t_{mp} = t_{cycle}.\,(n_{read} + n_{write}).\left[P_{miss}.\frac{total_{mr}}{100}\right],} \quad (10)$$

where t_c is the time taken per cache access [Sec] and t_{cycle} is the processor's cycle time [Sec].

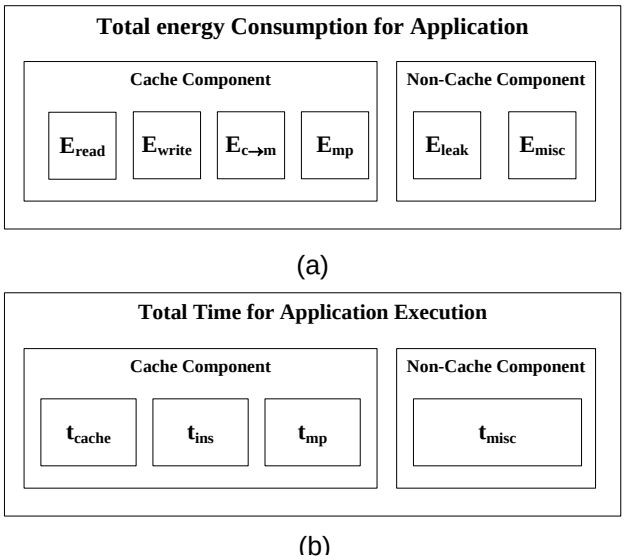

Figure 1(a) Total energy consumption for an application, **(b)** Total time taken in the execution of application

4. Model Validation

In order to validate the mathematical models presented in the previous section, the IBM/AMCC PPC440GP Evaluation Board (often referred to as Ebony reference board) model in SIMICS [14] was used. The MontaVista Linux 2.1 kernel was used as the target application to evaluate the performance. The generic cache model present in SIMICS was used for a range of associativity from 1 to 16-way. The cache size was set as 8 Kbytes, and block size as 256 bytes. The processor energy information was obtained from PowerPC440GP datasheet [21], and the cache timing and energy consumption data was obtained from CACTI 4.1 [22]. The processor model parameters are listed in Table 1.

The energy and timing model results are presented in Fig. 2. The graphs show that the average error of the energy model is around 4% and that of throughput model is 2.7%. The first set of results (marked *) was taken by setting miss penalty as 0 cycles in SIMICS, while the remainder of the results are based on 3 cycles miss penalty.

Table 1 Target Processor Parameters

Parameter	Value
Processor	PowerPC440GP
Execution mode	Turbo
Clock frequency	100 MHz
Cycles Per Instruction (CPI)	1
Technology	0.18um
Vdc (V)	1.8
Logic Supply (V)	3.3
DDR SDRAM (V)	2.5
IDD(A) active operating current	915mA
Energy per Cycle (J)	16.5nJ
Idle mode Energy (J)	4.12nJ

5. Example Applications

Most of the parameters used in the proposed mathematical models could be easily obtained using CACTI and some datasheet information. The number of read and write of an application could be determined by its instruction profile. The approximate hit or miss rate information could be gathered offline by using trace driven simulators like SIMICS, Dinero[23], Cheetah[24] etc. Assuming availablity of all the parameters, the models could be applied in various applications. Some of the possible applications are discussed below.

5.1. Real-time Cache Reconfiguration

Energy aware adaptive systems may be able to increase their efficiency by reconfiguration of the memory system on the go. Cache reconfiguration could be one of the applications in this scenario. The system may consider throughput effects by changing the cache size and associativity which directly affect the timing of the system. Also the energy profile obtained from the energy model could help an adaptive system to reconfigure cache keeping in view of available energy reserves. As the models are fairly accurate, their adoption for real-time systems is possible, while the parameters of the model could be stored in a lookup table. An example of reconfigurable cache systems could be found in a work by Balasubramonian et al. [11] as reviewed in section 2; other such examples are discussed in [25, 26].

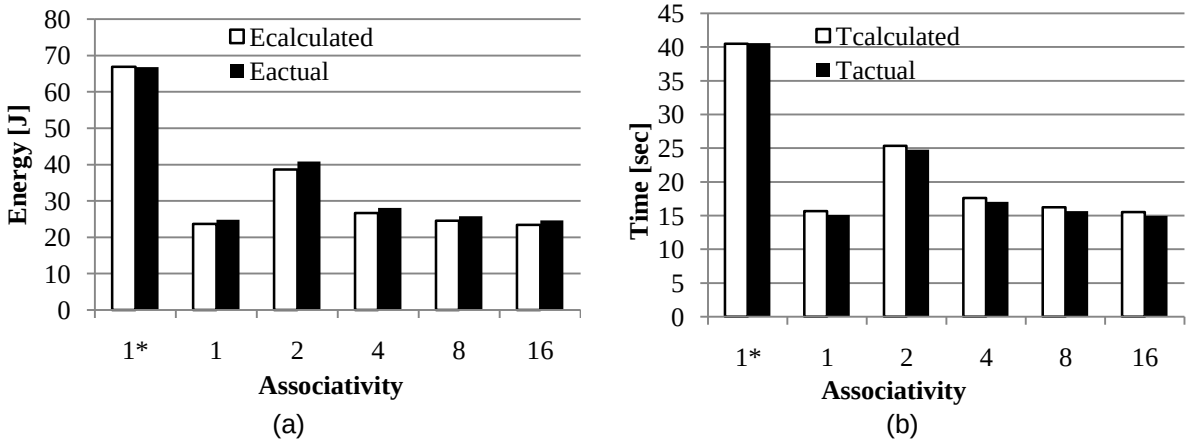

Figure 2. Validation of mathematical models **(a)** Energy model results vs. Actual energy consumption **(b)** Throughput model results vs. Actual time taken

5.2. Software Development

In the presence of a fixed (non-reconfigurable) hardware, the software remains the only part which could be optimized to perform as per timing and energy constraints. The proposed mathematical models could be used offline, during the application development processes in order to assess the performance before running it on the actual platform. Although there are full system simulators like SIMICS and SimpleScalar for such applications, but these generally have only a limited number of models available. The proposed models strive to provide an alternate to quickly analyze the software performance and could help in optimization process.

5.3. Hardware Selection

For a particular application software various hardware platforms could be analyzed using the proposed mathematical models. This provides an opportunity to the system designer to select the best performing hardware for the application in hand.

6. Conclusion

In this paper mathematical models analyzing data cache energy and throughput were presented. Upon comparing to the actual energy and timing information obtained from datasheet and SIMICS timing profile, it was found that the models resulted in less than 5% error. Such accuracy and lower computation complexity enable the models to be implemented in real-time adaptive systems supporting cache reconfiguration on the go. These models are to be further investigated for multicore architectures such as massively parallel processor arrays (MPPAs), symmetric chip multiprocessors (SCMPs), or asymmetric ship multiprocessors (ACMPs). Also an extension for multi-level cache systems to evaluate energy and throughput performance could be explored in future.

7. References

[1] M. Y. Qadri and K. D. M. Maier. "Towards Increased Power Efficiency in Low End Embedded Processors: Can Cache Help?" in *4th UK Embedded Forum*, Southampton, UK, 2008.

[2] M. Y. Qadri and K. D. M. Maier. "Data Cache-Energy and Throughput Models: A Design Exploration for Overhead Analysis," in *DASIP 2008*, Brussels, Belgium, 2008, pp. 153-159.

[3] T. Simunic, L. Benini, and G.D. Micheli, "Cycle-accurate simulation of energy consumption in embedded systems" in *36th Design Automation Conference, Proceedings* 1999, pp. 867-872.

[4] G. Q. Maguire, M.T. Smith, and H.W.P. Beadle, "SmartBadges: a wearable computer and communication system," in *6th International Workshop on Hardware/Software Codesign*, 1998.

[5] M. B. Kamble and K. Ghose, "Energy-Efficiency of VLSI Caches: A Comparative Study," in *Proceedings of the Tenth International Conference on VLSI Design: VLSI in Multimedia Applications*, 1997, pp. 261-267.

[6] M. B. Kamble and K. Ghose, "Analytical Energy Dissipation Models For Low Power Caches," in *Proceedings of the 1997 international symposium on Low power electronics and design*,Monterey, California, United States 1997, pp. 143-148

[7] Y. Li and J. Henkel, "A framework for estimation and minimizing energy dissipation of embedded HW/SW systems," in *Proceedings of the 35th annual conference on*

Design automation, San Francisco, California, United States 1998, pp. 188-193.

[8] J. Henkel and Y. Li, "Avalanche: an environment for design space exploration and optimization of low-power embedded systems," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 10, pp. 454-468, 2002.

[9] V. Tiwari, S. Malik, and A. Wolfe, "Power analysis of embedded software: a first step towards software power minimization," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 2, pp. 437-445, 1994.

[10] V. Tiwari and M.T.C. Lee, "Power analysis of a 32-bit embedded microcontroller," in *Proceedings of the 1995 conference on Asia Pacific design automation*, Makuhari, Massa, Chiba, Japan 1995

[11] R. Balasubramonian, D. Albonesi, A. Buyuktosunoglu, and S, Dwarkadas, "Memory hierarchy reconfiguration for energy and performance in general-purpose processor architectures," in *Proceedings of the 33rd annual ACM/IEEE international symposium on Microarchitecture*, Monterey, California, United States 2000, pp. 245-257.

[12] T. Wada, M. Suresh Rajan, and S.A. Przybylski, "An analytical access time model for on-chip cache memories," *IEEE Journal of Solid-State Circuits*, vol. 27, pp. 1147-1156, 1992.

[13] T. M. Taha and D. S. Wills, "An Instruction Throughput Model of Superscalar Processors," *IEEE Transactions on Computers*, vol. 57, pp. 389-403, 2008.

[14] P. S. Magnusson, et al., "Simics: A Full System Simulation Platform," *IEEE Computer*, vol. 32, pp. 50-58, 2002.

[15] T. Austin, E. Larson, and D. Ernst, "SimpleScalar: An Infrastructure for Computer System Modeling," *Computer*, vol. 35, pp. 59-67, 2002.

[16] D. Brooks, V. Tiwari, and M. Martonosi, "Wattch: a framework for architectural-level power analysis and optimizations," in *Proceedings of the 27th annual international symposium on Computer architecture*, Vancouver, British Columbia, Canada 2000, pp. 83-94.

[17] A. Flores, J. L. Arag´on, and M. E. Acacio, "Sim-PowerCMP: A Detailed Simulator for Energy Consumption Analysis in Future Embedded CMP Architectures," in *21st International Conference on Advanced Information Networking and Applications Workshops, 2007, AINAW '07*. Niagara Falls, Ont., 2007, pp. 752-757.

[18] C. J. Hughes, V. S. Pai, P. Ranganathan, and S. V. Adve, "RSIM: Simulating Shared-Memory Multiprocessors with ILP Processors," *IEEE Computer*, vol. 35, pp. 40-49, 2002.

[19] Y. Zhang, et al., D. Parikh, K. Sankaranarayanan, K. Skadron, and M. Stan, "HotLeakage: A Temperature-Aware Model of Subthreshold and Gate Leakage for Architects," University of Virginia, Department of Computer Science 2003.

[20] H. S. Wang, X. Zhu, L. S. Peh, and S. Malik, "Orion: a power-performance simulator for interconnection networks," in *Proceedings of the 35th annual ACM/IEEE international symposium on Microarchitecture*, Istanbul, Turkey 2002.

[21] AMCC, "PowerPC440GP Datasheet," [cited 2009].

[22] D. Tarjan, S. Thoziyoor, and N. P. Jouppi, "CACTI 4.0 Technical Report HPL-2006-86," HP Laboratories Palo Alto, June 2006.

[23] J. Edler, "Dinero IV Trace-Driven Uniprocessor Cache Simulator," [cited 2009]; Available from: http://pages.cs.wisc.edu/~markhill/DineroIV/.

[24] R. Sugumar and S. Abraham, "cheetah-Single-pass simulator for direct-mapped, set-associative and fully associative caches," in *Unix Manual Page*, 1993.

[25] A. S. Dhodapkar and J. E. Smith, "Managing multi-configuration hardware via dynamic working set analysis," *ACM SIGARCH Computer Architecture News*, vol. 30, pp. 233-244.

[26] P. Ranganathan, S. Adve, and N. P. Jouppi, "Reconfigurable Caches and their Application to Media Processing," in *Proceedings of the 27th International Symposium on Computer Architecture (ISCA-27)*, 2000, pp. 214-224.