Analytical Evaluation of Energy and Throughput for Multilevel Caches

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

***Abstract*--With the increase of processor-memory performance gap, it has become important to gauge the performance of cache architectures so as to evaluate their impact on energy requirement and throughput of the system. Multilevel caches are found to be increasingly prevalent in the high-end processors. Additionally, the recent drive towards multicore systems has necessitated the use of multilevel cache hierarchies for shared memory architectures. This paper presents simplified and accurate mathematical models to estimate the energy consumption and the impact on throughput for multilevel caches for single core systems.**

Keywords-cache mathematical models; energy; throughput; multilevel-cache

I. INTRODUCTION

Cache memories were primarily introduced to bridge the gap between processor and memory performance. With the processors operating at remarkable speeds and DRAM speeds operating at fraction of that, multilevel cache hierarchies provided a viable solution to keep up the trend of the Dave House's revision of Moore's law predicting that computer performance will double every 18 months [1]. The mathematical models presented in this paper analyze the energy consumption and throughput for multilevel data cache using PowerPC750, and UltraSPARC-II processors. The throughput model was validated using SIMICS full system simulator and energy model using per cycle energy consumption statistics presented in the processor datasheet. The models are multilevel extension of the single level models presented by the authors in [2-4]. The aim of this research is to propose simplified mathematical models for multilevel cache so in the future multiprocessor configurations could be evaluated using them, in a resource constrained environment such as the system itself.

The rest of this paper is divided into five sections. In the following section related work is discussed. The energy and throughput models for multilevel cache are introduced in section 3. In the fourth section the models are validated using a two level cache hierarchy, and the final section presents the conclusion.

II. RELATED WORK

This section presents the related research findings in the area of cache performance estimation and its usage for various applications.

Lim et al. [5] have proposed a set of equations to estimate accurately worst case time analysis (WCTA) for RISC processors. Their models detail the pipelining, instruction cache and data cache effects on real-timeliness of the system. Another reference of timing models is the work by Peuto et al. [6], in which the authors proposed an instruction timing model accounting cache effects. Taha et al. [7] presented an instruction throughput model of Superscalar processors. Their model include parameters such as superscalar width, depth of pipeline, instruction fetch mechanism (in-order/out-of-order), branch predictor, central issue window width, number of functional units their latencies and throughputs, re-order buffer width and cache size and latency etc. Their model resulted in errors up to 5.5% when compared to the SimpleScalar simulator [8]. Wada et al. [9] proposed detailed circuit level analytical access time model for on-chip cache memories. The model takes inputs such as number of tag/data array per word/bit line etc. On comparing with SPICE results the model gives 20% error for an 8 ns access time cache memory.

Simunic et al. [10] proposed analytical models for energy estimation in embedded systems. The per cycle energy model presented in their work comprises components such as energy consumption of processor, memory, interconnects and pins, DC-to-DC converters and level two (L2) cache. The model validation was performed using integrated simulator of ARM SDK [11]. This was found to be within 5% of the hardware measurements for the same operating frequency. The models presented in their work holistically analyze the embedded system power and do not estimate energy consumption for individual components of a processor. Kamble et al. in [12] also presented detailed cache energy model. The analytical models for conventional caches were found to be accurate to within 2% error. However, the technique over–predicts the power dissipations of low–power caches by as much as 30%.

Revisiting the models in [13] the authors found them to be accurate within 20% for low power caches upon comparing with their designed simulator *CA*che *P*ower *E*stimator (CAPE) for the simulated execution of several SPEC95 benchmarks. Li et al. [14] proposed a full system energy model comprising of cache, main memory, and software energy components. Their work also details a framework to assess and optimize energy dissipation of embedded systems. Tiwari et al. [15] proposed an instruction level energy model estimating energy consumed in individual pipeline stages. An identical methodology was applied in [16] by the authors to study the effects of cache enabling and disabling.

A simulator for analyzing cache energy dissipation called *CA*che *P*ower *E*stimator (CAPE) was developed by Kamble et al. based on the mathematical models they proposed [12, 13]. Based on these models another tool named INCAPE was developed by Korkmaz et al. [17]. This simulator is capable of estimating energy dissipation for complete memory hierarchy including multilevel cache system. Another widely referred, open source tool is CACTI (cache access and cycle time model) [18] by HP Laboratories Inc. It provides thorough, near accurate memory access time and energy estimates. However it is not a trace driven simulator, so energy consumption resulting in number of hits or misses is not accounted for a particular application. Magnusson et al. [19] presented SIMICS, a full system simulation tool. The simulator is targeted to provide accurate timing profile, but presently does not support energy profiling of the target platforms. Austin et al. [8] presented SimpleScalar another full system virtualization platform. The tool uses an execution-driven simulation technique that reproduces a device's internal operation. Exploiting SimpleScalar interface, Brooks et al. [20] developed a tool called Wattch for architectural level power analysis. It maintains accuracy within 10% as compared to results of circuit level power analysis tools. A further addition to this was done by Flores et al. [21] 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 [22] presented by Hughes et al.

III. THE CACHE ENERGY AND THROUGHPUT MODELS

This section presents the energy and throughput models for a two-level cache hierarchy.

A. Energy Model

If E_{ic}, E_{dc} , and E_{l2c} is the energy consumed by instruction, data and level 2 (L2) cache operations, E_{misc} is the Energy consumed by the instructions which do not require data memory access, and E_{leak} the leakage energy of the processor, then the total energy consumption of the code E_{total} in Joules [J] can be defined as,

$$E_{total} = E_{ic} + E_{dc} + E_{l2c} + E_{misc.} + E_{leak} \tag{1}$$

Where,

L1 Instruction Cache

$$E_{ic} = E_{ic-read} + E_{ic-mp} \tag{2}$$

$$E_{ic-read} = E_{ic-rcycle} \cdot \eta_{ic-read} \tag{3}$$

$$E_{ic-mp} = E_{cycle} . P_{ic-rmiss} . \eta_{ic-rmiss} \tag{4}$$

L1 Data Cache

$$E_{dc} = E_{dc-read} + E_{dc-write} + E_{dc-mp} \tag{5}$$

$$E_{dc-read} = E_{dc-rcycle} \cdot \eta_{dc-read} \tag{6}$$

$$E_{dc-write} = E_{dc-wcycle} \cdot \eta_{dc-write} \tag{7}$$

$$E_{dc-mp} = E_{cycle} . (P_{dc-rmiss} . \eta_{dc-rmiss} + P_{dc-wmiss} . \eta_{dc-wmiss}) \tag{8}$$

L2 Cache

$$E_{l2c} = E_{l2c-read} + E_{l2c-write} + E_{dc-mp} + E_{l2c \to ram} + E_{l2c \to rom} \tag{9}$$

$$E_{l2c-read} = E_{l2c-rcycle} \cdot (\eta_{l2c-if} + \eta_{l2c-dread}) \tag{10}$$

$$E_{l2c-write} = E_{l2c-wcycle} \cdot \eta_{l2c-dwrite} \tag{11}$$

$$E_{l2c-mp} = E_{cycle} . \{P_{l2c-rmiss} . (\eta_{l2c-if} + \eta_{l2c-dread}) + P_{l2c-wmiss} . \eta_{l2c-dwrite}\} \tag{12}$$

In the above equations E_{x-read}, $E_{x-write}$, and E_{x-mp} denote the read, write and miss penalty energy of the corresponding cache *x* (i.e. instruction, data or L2 cache). The read and write cycle energy per cache access is denoted by $E_{x-rcycle}$ and $E_{x-wcycle}$. The number of data read and write transactions of the cache (including all hits and miss) is denoted by η_{x-read} and $\eta_{x-write}$. Furthermore η_{l2c-if}, $\eta_{l2c-dread}$, $\eta_{l2c-dwrite}$ denote the L2 cache's instruction fetch, data read and data write transactions respectively. The processor's per cycle energy consumption is denoted by E_{cycle}; $P_{x-rmiss}$, $P_{x-wmiss}$, $\eta_{x-rmiss}$ and $\eta_{x-wmiss}$ denote the read/write miss penalty (in terms of number of cycles) and their corresponding miss rates. The energy consumed in L2 cache to data and code memory is denoted by $E_{l2c\to ram}$ and $E_{l2c\to rom}$ that could also be calculated by multiplying the number of memory accesses with their read and write cycles energy.

The idle mode leakage energy of the processor $E_{leak(std)}$ can be calculated as

$$E_{leak} = P_{leak}. t_{idle}, \tag{13}$$

where t_{idle} [Sec] is the total time for which processor was idle.

B. Throughput Model

If t_{ic}, t_{dc}, and t_{l2c} is the time taken in instruction, data and level 2 (L2) cache operations, and t_{ins} the time taken in execution of cache access instructions [Sec], $t_{x-read}, t_{x-write}$ and t_{x-mp} the time taken in read, write and miss penalty for cache *x*; then T_{total} the total time taken by an application could be estimated as

$$T_{total} = t_{ic} + t_{dc} + t_{l2c} + t_{ins} \tag{14}$$

Furthermore,

L1 Instruction Cache

$$t_{ic} = t_{ic-read} + t_{ic-mp} \tag{15}$$

$$t_{ic-read} = t_{ic-rcycle}. \eta_{ic-read} \tag{16}$$

$$t_{ic-mp} = t_{cycle}. P_{ic-rmiss}. \eta_{ic-rmiss} \tag{17}$$

L1 Data Cache

$$t_{dc} = t_{dc-read} + t_{dc-write} + t_{dc-mp} \tag{18}$$

$$t_{dc-read} = t_{dc-rcycle} \cdot \eta_{dc-read} \tag{19}$$

$$t_{dc-write} = t_{dc-wcycle} \cdot \eta_{dc-write} \tag{20}$$

$$t_{dc-mp} = t_{cycle}. (P_{dc-rmiss}. \eta_{dc-rmiss} + P_{dc-wmiss}. \eta_{dc-wmiss}) \tag{21}$$

L2 Cache

$$t_{l2c} = t_{l2c-read} + t_{l2c-write} + t_{dc-mp} + t_{l2c\to ram} + t_{l2c\to rom} \tag{22}$$

$$t_{l2c-read} = t_{l2c-rcycle} \cdot (\eta_{l2c-if} + \eta_{l2c-dread}) \tag{23}$$

$$t_{l2c-write} = t_{l2c-wcycle} \cdot \eta_{l2c-dwrite} \tag{24}$$

$$t_{l2c-mp} = t_{cycle}. \{P_{l2c-rmiss}. (\eta_{l2c-if} + \eta_{l2c-dread}) + P_{l2c-wmiss}. \eta_{l2c-dwrite}\} \tag{25}$$

and,

$$t_{ins} = t_{cycle}. \eta_{cycle} - t_{ic-read} \tag{26}$$

where $t_{x-rcycle}$, $t_{x-wcycle}$ is the time taken per cache read and write cycle and t_{cycle} is the processor cycle time in seconds [sec].

TABLE I. PROCESSOR PARAMETERS

Parameter	Value	
Processor	PowerPC750	UltraSPARC II
Execution mode	In-order	In-order
Clock frequency [MHz]	373.5	168
Cycle Time [ns]	2.68	5.95
Fabrication Technology [µm]	0.2	0.35
V_{dc} [V]	2.2	3.3
Operating current I_{DD} [A]	2	9.4
Energy per Cycle [nJ]	11.8	185
Operating System	MontaVista Linux 2.1	Fedora Core 3

IV. MODEL VALIDATION

To validate the accuracy of the proposed models, Virtutech's SIMICS full system simulator was used to simulate two processor models i.e. IBM PowerPC750 [23] and Sun UltraSPARC-II . Although the SIMICS PowerPC750 and UltraSPARC-II [24] default configurations do not have a cache component, a two level cache hierarchy was introduced into the models to replicate the actual L1 and L2 cache of the said processor. The L1 and L2 cache sizes were selected as per datasheet of the processors. For L2 cache, a 256 KB (32K x 8) high speed CMOS SRAM [25] was used and data and code memory were selected as 4MB (512K x 8) CMOS SRAM [26], and 256KB (32K x 8) EEPROM [27]. The cache parameters are specified in Table 2. The cache access time and energy per access information was taken from CACTI 4.0.

To evaluate the cache performance various applications from MiBench [28] benchmarking tool suite have been selected (see Table 3 for the description). The energy and throughput model results are shown in Figures 1 and 2. The energy and throughput model for PowerPC platform resulted in an error up to 3%, whereas that of UltraSPARC-II was 6.5% and 9% respectively. The actual timing values are observed from SIMICS simulation, and the energy consumption information was taken from per cycle energy consumption energy information of PowerPC750 and UltraSPARC-II datasheets.

TABLE II. CACHE SIMULATOR DATA

CACTI Data			
L1 I-Cache			
PowerPC750		**UltraSPARC II**	
Cache Size [KBytes]	32	Cache Size [KBytes]	16
Line Size [Bytes]	32	Line Size [Bytes]	32
Number of Lines	1024	Number of Lines	512
R/W Ports	1	R/W Ports	1
Associativity	8	Associativity	2
Miss Penalty[cycles]	7	Miss Penalty[cycles]	10
Read ports	0	Read ports	0
Write ports	0	Write ports	0
Access Time [ns]	1.82	Access Time [nSec]	2.91
Cycle Time [ns]	1.04	Cycle Time [nSec]	1.58
Read Energy [nJ]	0.725	Read Energy [nJ]	0.49
L1 D-Cache			
PowerPC750		**UltraSPARC II**	
Cache Size [KBytes]	32	Cache Size [KBytes]	16
Line Size [Bytes]	32	Line Size [Bytes]	32
Number of Lines	1024	Number of Lines	512
R/W Ports	1	R/W Ports	1
Associativity	8	Associativity	2
Miss Penalty[cycles]	7	Miss Penalty[cycles]	10
Read ports	0	Read ports	0
Write ports	0	Write ports	0
Access Time [ns]	1.82	Access Time [nSec]	2.94
Cycle Time [ns]	1.04	Cycle Time [nSec]	1.64
Read Energy [nJ]	0.725	Read Energy [nJ]	0.48
Write Energy [nJ]	0.067	Write Energy [nJ]	0.186

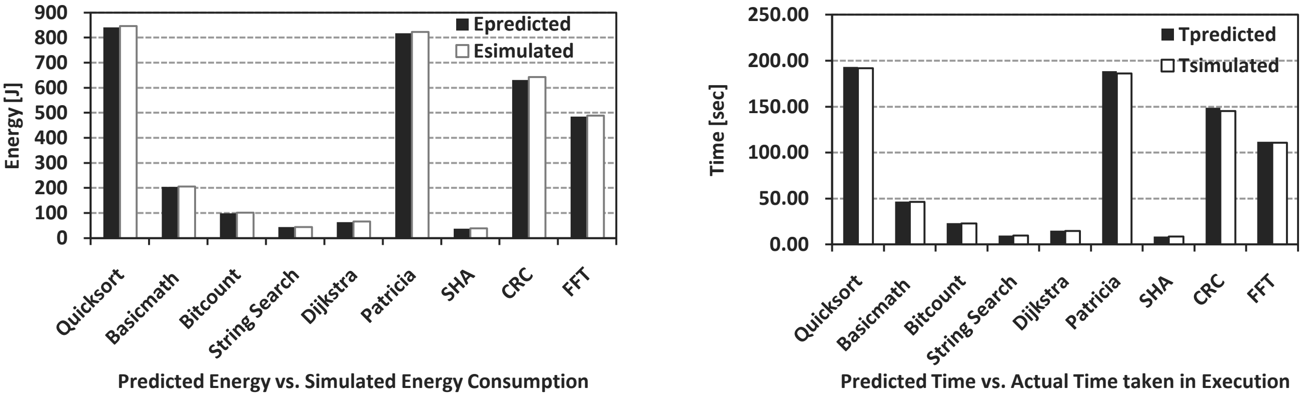
The higher error for UltraSPARC-II could be attributed for its being a server class processor with significant energy and throughput contributions from I/O bus operations etc.

V. CONCLUSION

In this paper multilevel cache energy and throughput models were introduced. The models require a significantly smaller number of parameters as compared to the existing methods discussed in related work. Additionally, the parameters can be easily obtained using the techniques adopted in the validation of the models. The models were validated with a two level cache model of PowerPC750 and Ultra SPARC-II processor, using standard benchmark applications and simulation tools. The results were found to be 3% accurate for PowerPC, whereas for UltraSPARC-II was 6.5% and 9% for energy and throughput models respectively when compared against the simulator data. In the future these models are to be extended for multicore architectures and may be applied in real-time adaptive memory systems, where an accurate estimate of throughput and energy consumption for cache is required.

Table III. Benchmark applications from MiBench

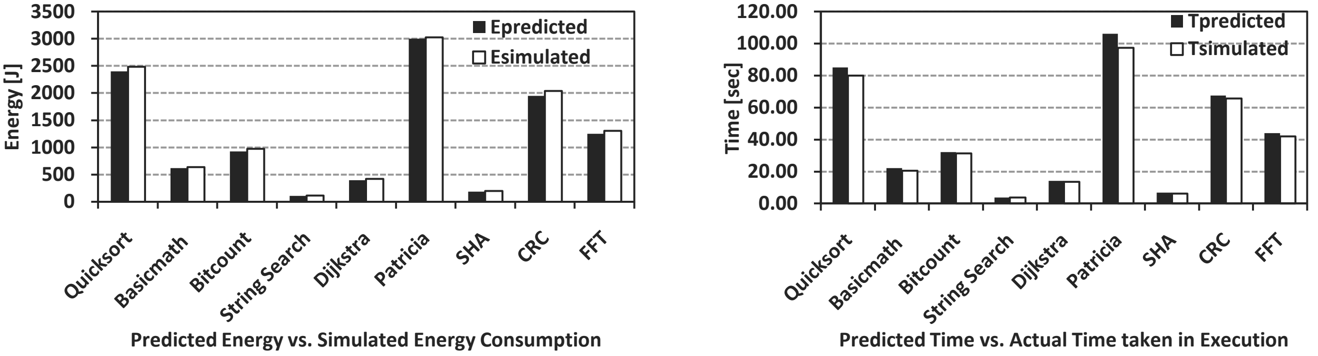
Benchmark	Description
Quicksort	Quick Sort algorithm with large dataset of 50000 three-tuples representing points of data.
Basicmath	Basic math operations such as cubic function solving, integer square root and angle conversions from degrees to radians. The input data is a fixed set of constants.
Bitcount	The bit count algorithm counts the number of bits in an array of integers with input dataset of 1200000 integers.
String Search	A case insensitive Pratt-Boyer-Moore string search implementation with a dataset of more than 2600 phrases.
Dijkstra	Constructs a large graph in an adjacency matrix representation and then calculates the shortest path between every pair of nodes for a data input of 10000 integers.
Patricia	Patricia tries are used to represent routing tables in network applications. The input data for this is a list of IP traffic from a highly active web server for a 2 hour period.
SHA	Secure hash algorithm that produces a 160-bit message digest for a given input. The input dataset is more than 3240000 ASCII characters.
CRC	A 32 bit Cyclic Redundancy Check implementation with large input dataset.
FFT	Fast Fourier Transform on an array of 8 waves each of 32768 points.



Predicted Energy vs. Simulated Energy Consumption

Predicted Time vs. Actual Time taken in Execution

Figure 1. Energy and Throughput for various benchmarks on PowerPC platform



Predicted Energy vs. Simulated Energy Consumption

Predicted Time vs. Actual Time taken in Execution

Figure 2. Energy and Throughput for various benchmarks on UltraSPARC-II platform

REFERENCES

[1] "Excerpts from A Conversation with Gordon Moore: Moore's Law," Video Transcript: Intel Corporation, 2005.

[2] 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.

[3] M. Y. Qadri and K. D. M. Maier, "Data Cache-Energy and Throughput Models: A Design Exploration for Overhead Analysis," in *Conference on Design and Architectures for Signal and Image Processing*, Brussels, Belgium, 2008, pp. 153-159.

[4] M. Y. Qadri and K. D. McDonald-Maier, "Data Cache-Energy and ThroughputModels: Design Exploration for Embedded

Processors," *EURASIP Journal on Embedded Systems,* vol. 2009, p. 7, 2009.

[5] S.-S. Lim, Y. H. Bae, G. T. Jang, B.-D. Rhee, S. L. Min, and E. P. Number, "An Accurate Worst Case Timing Analysis for RISC Processors," *IEEE Transactions on Software Engineering,* vol. 21, pp. 593 - 604, 1995 1995.

[6] B. L. Peuto and L. J. Shustek, "An instruction timing model of CPU performance," in *International Symposium on Computer Architecture*, Barcelona, Spain, 1998, pp. 152-165.

[7] T. M. Taha and D. S. Wills, "An Instruction Throughput Model of Superscalar Processors," in *14th IEEE International Workshop on Rapid System Prototyping (RSP 2003)*, San Diego, CA, USA, 2003, pp. 156-163.

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

[9] T. Wada, S. 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.

[10] T. Simunic, L. Benini, and G. D. Micheli, "Cycle-accurate simulation of energy consumption in embedded systems," in *The 36th annual ACM/IEEE Design Automation Conference*, New Orleans, Louisiana, United States, 1999, pp. 867-872.

[11] "ARM Software Development Toolkit," 2.11 ed:: Advanced RISC Machines Limited (ARM), 1996.

[12] M. B. Kamble and K. Ghose, "Analytical energy dissipation models for low power caches," in *Low Power Electronics and Design, 1997. Proceedings., 1997*, Monterey, California, USA, 1997, pp. 143-148.

[13] M. B. Kamble and K. Ghose, "Modeling energy dissipation in low power caches," in *International Symposium on Low Power Electronics and Design*, 1998, pp. 143-148.

[14] Y. Li and Y. Jörghenkel, "A framework for estimation and minimizing energy dissipation of embedded HW/SW systems," in *35th annual Design Automation Conference*, San Francisco, California, United States 1998, pp. 188-193.

[15] V. Tiwari, S. Malik, and A. Wolfe, "Power Analysis of Embedded Software: A First Step towards Software Power Minimization," *IEEE Transactions on VLSI Systems,* vol. 2, pp. 437-445, 1994.

[16] V. Tiwari and M. T.-c. Lee, "Power analysis of a 32-bit embedded microcontroller," in *Asia and South Pacific Design Automation Conference*, Makuhari, Massa, Chiba, Japan, 1995.

[17] P. Korkmaz, K. Puttaswamy, and V. Mooney, "Energy Modelling of a Processor Core using Synopsys and of Memory Hierarchy using Kamble and Ghose Model," Center For Research on Embedded Systems and Technology, Georgia Institute of Technology, Atlanta, Georgia 2002.

[18] D. Tarjan, S. Thoziyoor, and N. P. Jouppi, "CACTI 4.0," HP Laboratories Palo Alto 2006.

[19] P. S. Magnusson, M. Christensson, J. Eskilson, D. Forsgren, G. Hallberg, J. Hogberg, F. Larsson, A. Moestedt, and B. Werner, "Simics: A full system simulation platform," *Computer,* vol. 35, pp. 50-58, 2002.

[20] D. Brooks, V. Tiwari, and M. Martonosi, "Wattch: A framework for architectural-level power analysis and optimizations," in *The 27th Annual International Symposium on Computer Architecture*, Vancouver, British Columbia, Canada 2000, pp. 83-94.

[21] A. Flores, J. L. Aragon, and M. E. Acacio, "Sim-PowerCMP: a detailed simulator for energy consumption analysis in future embedded CMP," in *Advanced Information Networking and Applications Workshops*, Niagara Falls, Ont., 2007, pp. 752-757.

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

[23] "PowerPC 740 and PowerPC 750 Microprocessor Datasheet Version 2.0," IBM Microelectronics Division, 2002.

[24] "UltraSPARC™-II Datasheet - Second Generation SPARC v9 64-Bit Microprocessor With VIS," Sun Microelectronics, 1997.

[25] "IS61C256AL Datasheet- 32K x 8 HIGH-SPEED CMOS STATIC RAM," Integrated Silicon Solution, Inc., 2006.

[26] "AS7C34096A Datasheet- 3.3V 512K × 8 CMOS SRAM," Alliance Semiconductor, 2004.

[27] "X28C256 Datasheet- 256K 5 Volt, Byte Alterable E2PROM," Xicor, Inc., 1996.

[28] M. R. Guthaus, J. S. Ringenberg, D. Ernst, T. M. Austin, T. Mudge, R. B. Brown, B. Ave, and A. Arbor, "MiBench: A free, commercially representative embedded benchmark suite," in *IEEE 4th Annual Workshop on Workload Characterization*, Austin, Texas, 2001, pp. 3-14.