Energy Modeling for Mobile Devices using Performance Counters

Sriram Sankaran, Ramalingam Sridhar
University at Buffalo, The State University of New York, Buffalo, NY Email: rsridhar@buffalo.edu

***Abstract*— The increasing complexity of mobile applications coupled with growing user demands lead to rapid battery drain in mobile devices. However, battery technology cannot keep up with these trends thus making power management one of the foremost concerns. While system-level approaches to power management exist, the energy impact of applications on individual system components needs to be better understood for energy efficient system design. In this work, we develop energy models for mobile devices using performance counters and estimate the power consumption of system components for numerous embedded applications. Our models provide enhancements in I/O towards estimating I/O energy and cache to incorporate energy consumed during cache refill and write-back in the energy estimation process. We further compare our power estimates with existing models and demonstrate the uniqueness of our model.**

I. Introduction

Power dissipation is one of the foremost concerns in battery powered embedded systems such as mobile devices. Advancements in computing and communication technologies coupled with user mobility have propelled the usage of mobile applications which significantly impact the power consumption of system components. This necessitates the need for power models to understand the impact of applications on the overall system. Depending on the power estimates generated by a power-model, an energy optimization plan can be developed for facilitating efficient energy usage in mobile devices.

Power modeling has gained prominence due to increasing functionality in mobile devices. While Dynamic Power refers to the application-dependent switching activity and can be tracked by analyzing architectural-level power and performance traces of individual applications, leakage power is consumed whenever components are powered on and is independent of application usage. We focus our analysis on dynamic power and leave leakage power for future work.

In this work, we present energy models for mobile devices using Performance Counters and estimate the power consumed by individual system components for embedded applications. In contrast to existing works, we account for the energy consumed by I/O and improve upon the existing cache models primarily focused on access counts to incorporate power consumed during cache refill and write-back in the power estimation process. Further, we present a detailed comparison between power estimation values reported by our model and previous models and demonstrate the uniqueness of our model.

II. Related Work

The problem of power modeling has been addressed at all levels of the system ranging from hardware, system [12], applications, network protocols [10] as well as individual components such as displays [7] [8] and disks [19]. Approaches have been categorized into Measurement and Modeling. Measurement based Methods [3] [9] utilize an external measurement device while modeling relies on cycle-level simulation tools [4] [6] to model the power consumption of individual components. Tiwari et al [1] modeled the instruction-level power consumption of embedded processors. McCullough et al. [2] utilized performance counters to model the power consumption of full systems. Rivoire et al. [11] provided a comparative evaluation of different power models for servers.

III. Power Modeling in Mobile devices

Power Modeling is necessary to understand the various factors that impact power consumption in mobile devices. While power can be modeled at different levels of the system, architectural-level models capture a high-level view of the system and can be used to make better power-performance trade-offs by considering application behavior. In the recent past, power modeling has become a significant challenge due to the integration of numerous processing cores and Graphics Processing Units (GPUs) on a mobile device.

Current approaches for power modeling can be classified into utilization-based power modeling [4] and performance counters [18] based power estimation.

Utilization-based Modeling: These approaches model the utilization of individual system components and estimate the corresponding power consumption. In this approach, the main challenge lies in mapping the utilization to the corresponding power consumption using statistical methods if per-component estimates are unknown. The main drawback of this approach is that it does not consider component-specific events such as cache misses.

Performance Counters: Mobile processors are augmented with special-purpose registers called Performance Counters for collecting component-specific statistics. These statistics can be used to build power models for individual components. The procedure lies in collecting events that are representative of the power consumed in mobile devices and understanding the relationship between them. Statistics for each of these events can be obtained either through cycle-level simulation tools such as Wattch [6] or measurement-based approaches for power estimation.

978-1-4799-0066-4/13/$31.00 ©2013 IEEE

TABLE I
ARM-CORTEX-A15 PERFORMANCE MONITORING EVENTS

Component	Performance Event	Description
CPU	CPU_CYCLE	CPU cycles
Memory	MEM_ACCESS_LD	Memory access read
	MEM_ACCESS_ST	Memory access write
Cache	L1I_CACHE	L1 instruction cache access
	L1D_CACHE	L1 data cache access
	L1I_CACHE_REFILL	L1 instruction cache refill
	L1D_CACHE_REFILL	L1 data cache refill
	L1D_CACHE_WB	L1 data cache write back
	L2D_CACHE	L2 data cache access
	L2D_CACHE_REFILL	L2 data cache refill
	L2D_CACHE_WB	L2 data cache write back

IV. ENERGY MODELING

We utilize Performance Counters to build an energy model for mobile devices. We consider ARM-Cortex-A15 processor due to its deployment in modern computing devices such as smartphones for which there exists numerous performance events [20] defined to track event-specific statistics. We select a subset of these events that are representative of energy consumption in mobile devices, group them by component and outline them in Table I. Below we present our model for each system component.

CPU Energy Model: Energy consumed by CPU, E_{cpu}, can be computed using the following equation

$$E_{cpu} = P_{cpu} * T_{cpu} \tag{1}$$

where, P_{cpu} is the average power consumed by CPU and T_{cpu} refers to CPU time taken for running the application. We computed CPU time by running benchmark applications on Simplescalar.

We calculate P_{cpu} using Wattch. In particular, we isolated the power consumption of CPU and cache and obtained the average power consumption of CPU using the following equation.

$$P_{cpu} = (P_{ren} + P_{bpre} + P_{win} + P_{lsq} + P_{reg} + P_{reb} + P_{clk} + P_{alu})/CycleTime \tag{2}$$

where $P_{ren} = Rename\ Unit\ Power$,
$P_{bpre} = Branch\ Predictor\ Power$,
$P_{win} = Instruction\ Window\ Power$,
$P_{lsq} = Load\ Store\ Queue\ Power$,
$P_{reg} = Register\ File\ Power$,
$P_{reb} = Result\ Bus\ Power$,
$P_{clk} = Clock\ Power$ and
$P_{alu} = ALU\ Power$

I/O Energy Model: While CPU time refers to the time taken by CPU for executing applications, it does not account for the time spent in I/O operations during which CPU remains in idle mode. This time is commonly referred as slack time. I/O energy consumption has recently gained significance due to the complexity involved in mobile applications [21]. Thus time spent in I/O operations along with the corresponding power consumption needs to be estimated. Energy consumed by I/O, $E_{i/o}$, can be computed using the following equation

$$E_{i/o} = (T_{elapsed} - T_{cpu}) * P_{i/o} \tag{3}$$

where, $T_{elapsed}$, T_{cpu} and $P_{i/o}$ refer to total time, CPU time and power consumed by I/O respectively. We obtained the statistics for each of them by running benchmark applications on Simplescalar. $E_{i/o}$ can be further calculated using the following equation

$$E_{i/o} = (T_{elapsed} - T_{cpu}) * (P_{Addressi/o} + P_{Datai/o}) \tag{4}$$

where, $P_{Addressi/o}$ and $P_{Datai/o}$ refer to power consumed by Address I/O and Data I/O respectively.

Memory Energy Model: Energy consumed by Memory depends on the number of read and write memory accesses along with the energy required to perform each of those accesses. We adapt the energy model from [17] for memory energy estimation. Energy consumed by Memory, E_{mem} can be calculated using the following equation

$$E_{mem} = (numreads) * E_{read} + (numwrites) * E_{write} \tag{5}$$

where numreads, numwrites, E_{read} and E_{write} refer to number of read accesses, number of write accesses, per-access read and write-energy respectively.

Cache Energy Model: Energy consumed by cache, E_{cache} can be computed using the following equation

$$E_{cache} = E_{il1cache} + E_{dl1cache} + E_{dl2cache} \tag{6}$$

where $E_{il1cache}$, $E_{dl1cache}$ and $E_{dl2cache}$ refer to energy consumed by Instruction Cache, Level 1 Data Cache and Level 2 Data Cache respectively.

We compute $E_{dl1cache}$ using the following equation

$$E_{dl1cache} = E_{dl1access} + E_{dl1refill} + E_{dl1writeback} \tag{7}$$

where $E_{dl1access}$ = Cache Access Energy,
$E_{dl1refill}$ = Cache Refill Energy and
$E_{dl1writeback}$ = Cache write-back Energy
$E_{dl1access}$, $E_{dl1refill}$ and $E_{dl1writeback}$ can be computed using the following equation

$$E_{dl1cache} = (n_{access} * E_{dl1access}) + (n_{refill} * E_{dl1refill}) + (n_{writeback} * E_{dl1writeback}) \tag{8}$$

where n_{access}, n_{refill} and $n_{writeback}$ refer to number of cache accesses, cache refills and cache writebacks respectively.

Similarly, energy consumed by instruction cache and Level-2 data cache can be estimated.

V. ENERGY ESTIMATION

CPU Energy Estimation: To estimate the CPU energy consumption of embedded applications, we performed simulations using Simplescalar [14] and Wattch [6]. In particular, we estimated the CPU time using Simplescalar and the power consumed by CPU using Wattch. A clock cycle time of 1.25 ns was assumed for computing CPU time. These values were then provided as parameters to our model and the CPU energy consumption for different benchmark applications was

TABLE II
CPU ENERGY CONSUMPTION

Application	CPU Energy
ADPCM	1.639 J
Dijkstra	2.172 J
Patricia	10.559 J

TABLE III
I/O ENERGY CONSUMPTION

Application	I/O Energy
ADPCM	7.32 J
Dijkstra	16.39 J
Patricia	61.03 J

estimated. We used benchmark applications from the MiBench [16] benchmark suite for energy estimation. Table II displays the results for CPU energy estimation.

I/O Energy Estimation: To estimate the energy consumed by I/O, we performed simulations using Simplescalar to track CPU time and elapsed time. Since Wattch does not account for I/O power, we used Sim-PAnalyzer [20] to estimate I/O power consumption. Sim-PAnalyzer is a power analysis tool for the ARM model of Simplescalar.

With the CPU time and elapsed time obtained for each application along with the corresponding I/O power consumption, energy consumed by I/O was estimated. Table III displays the results for I/O energy estimation.

Memory Energy Estimation: To estimate the energy consumed by memory, we performed simulations using Simplescalar and CACTI. In particular, we wrote source code to track the number of read and write memory accesses on Simplescalar. We used CACTI to estimate the per-access read and write energy of DRAM memories. Our DDR3 memory configuration is described in Table IV

With the read and write memory access statistics obtained for each application along with per-access read/write energy, we estimated the energy consumption of DRAM memory. Table V displays the results for energy consumption of DRAM memories obtained using our model.

Cache Energy Estimation: To estimate the energy consumed by caches, we performed simulations using Simplescalar and CACTI 4.2 [15]. In particular, we obtained cache access statistics along with the number of writebacks and replacements using Simplescalar and the dynamic read and write energy of varying cache configurations using CACTI. Parameters for the CACTI tool are described in Table VI.

TABLE IV
DDR3 MEMORY CONFIGURATION

Parameter	Value
Capacity	1 GB
Number of banks	8
Associativity	1
Technology	90nm
Number of Read/Write Ports	1
Number of Read Ports	2
Number of Write Ports	2
Access Time	26.7655 ns
Read Energy	4.535 nJ
Write Energy	4.51144 nJ

TABLE V
MEMORY ENERGY CONSUMPTION

Application	Memory Energy
ADPCM	0.1245 J
Dijkstra	1.025 J
Patricia	1.721 J

TABLE VI
CACTI PARAMETERS

Parameter	Value
Instruction Cache	Cache Size: 16384 Line Size: 32 Associativity: 1
L1 Data Cache	Cache Size: 16384 Line Size: 32 Associativity: 4
L2 Data Cache	Cache Size: 262144 Line Size: 64 Associativity: 4
Technology	90nm
Number of Read/Write Ports	1
Number of Read Ports	2
Number of Write Ports	2

From the per-access statistics for dynamic read and write energy, total dynamic read energy was estimated as the sum of read/write ports and read ports times the per-access dynamic read energy. Similar procedure was followed to estimate the total dynamic write energy. The results for the total dynamic read and write energy for each of the caches are detailed in Table VII.

From the total dynamic read and write energy for each of the caches, energy consumption of the caches corresponding to each application was calculated and the results are described in Table VIII.

TABLE VII
TOTAL DYNAMIC READ AND WRITE ENERGY FOR CACHES

Cache	Total Dynamic Energy
IL1	Read Energy: 0.2466 nJ Write Energy: 0.0891 nJ
DL1	Read Energy: 0.5136 nJ Write Energy: 0.1104 nJ
DL2	Read Energy: 1.7457 nJ Write Energy: 0.375 nJ

TABLE VIII
CACHE ENERGY CONSUMPTION

Application	Cache Energy
ADPCM	IL1: 8.70 mJ
	DL1: 1.23 mJ
	DL2: 2.128 μJ
Dijkstra	IL1: 13.83 mJ
	DL1: 9.08 mJ
	DL2: 0.458 mJ
Patricia	IL1: 40.43 mJ
	DL1: 0.458 mJ
	DL2: 19.82 mJ

TABLE IX
COMPARISON OF AVERAGE CACHE POWER ESTIMATED USING WATTCH AND OUR MODEL

Component	Wattch	Our Model
Instruction Cache IL1	ADPCM: 2.2136 W	ADPCM: 0.8761 W
	Djikstra: 2.2136 W	Djikstra: 0.8694 W
	Patricia: 2.2136 W	Patricia: 0.5563 W
Data Cache DL1	ADPCM: 5.1830 W	ADPCM: 0.1308 W
	Djikstra: 5.1830 W	Djikstra: 0.6036 W
	Patricia: 5.1830 W	Patricia: 0.2690 W
Data Cache DL2	ADPCM: 4.2091 W	ADPCM: 0.1274 μ W
	Djikstra: 4.2091 W	Djikstra: 0.0171 W
	Patricia: 4.2091 W	Patricia: 0.1624 W

VI. MODEL COMPARISON

A. Cache Model Comparison

We estimate and compare the cache power consumed by Wattch and our model. Table IX shows the results for the comparison. We computed dynamic power consumed by our cache power model DPower$_{Cache}$ using the following equation.

$$DPower_{Cache} = DEnergy_{Cache}/cycletime \quad (9)$$

where DEnergy$_{Cache}$ refers to Dynamic Energy of the Cache.

We obtained the values for dynamic energy from Table VIII and cycle time from CACTI needed to calculate dynamic power. Further dynamic power was averaged to calculate the average power consumed for each simulation cycle.

Using our power model, we estimated the power consumed by Instruction Cache, Level 1 data Cache and Level 2 data cache for all benchmarks. We do not include the comparison for memory since we adapt the existing model for memory power estimation.

The uniqueness of our model lies in accounting for the energy consumed by cache accesses, refills and write-backs for realistic power estimation compared to existing models such as Wattch primarily focused on cache accesses.

VII. CONCLUSION

We presented energy models for mobile devices using Performance Counters for estimating the power consumed by system components for embedded applications. Our model provides significant enhancements in I/O and cache energy modeling, thus enabling design space exploration for energy optimization. We propose to explore architectural enhancements such as dynamic cache reconfiguration for embedded applications and user-efficient system solutions as part of future work.

REFERENCES

[1] V. Tiwari, S. Malik and A. Wolfe, *Power analysis of Embedded Software: A first step towards software power minimization*, In *IEEE Transactions on VLSI*, 1994

[2] J. C. McCullough, Y. Agarwal, J. Chandrasekhar, S. Kuppuswamy, A. C. Snoeren and R. Gupta, *Evaluating the effectiveness of model based power characterization*, In *USENIX ATC* 2011

[3] J. Flinn and M. Sathyanarayanan, *Powerscope: A tool for profiling the usage of mobile applications*, In *Proceedings of the Second IEEE Workshop on Mobile Computer Systems and Applications*, 1999

[4] V. Shnayder, M. Hempstead, B. Chen, G. W. Allen and M. Welsh, *Simulating the Power Consumption of Large Scale Sensor Network Applications*, In *ACM Sensys* 2009

[5] R. Mittal, A. Kansal and R. Chandra, *Empowering Developers to estimate App Energy Consumption*, In *Proceedings of ACM Mobicom* 2012

[6] D. Brooks, V. Tiwari and M. Martonosi, *Wattch: A Framework for Architectural-Level Power Analysis and Optimizations*, In *Proceedings of ISCA* 2000

[7] M. Dong and L. Zhong, *Self-constructive high-rate system energy modeling*, In *Mobisys* 2011

[8] M. Dong and L. Zhong, *Power modeling of graphical user interfaces on oled displays*, In *DAC* 2009

[9] L. Zhang, B. Tiwana, Z. Qian, Z. Wang, R. P. Dick, Z. M. Mao, L. Yang, *Accurate Online power estimation and automatic battery behavior based power model generation for smartphones*, In *CODES+ISSS* 2010

[10] N. Balasubramanian, A. Balasubramanian, and A. Venkataramani, *Energy consumption in mobile phones: Implications for Network Applications*, In *Proceedings of ACM IMC* 2009

[11] S. Rivoire, P. Ranganathan and C. Kozyrakis, *A comparison of full-system power models*, In *Proceedings of HotPower* 2009

[12] A. Roy, S. Rumble, R. Stutsman, P. Levis and D. Mazieres, *Energy Management in Mobile Devices using the CINDER operating system*, Technical Report, Stanford University 2010-02, Available at http://hci.stanford.edu/cstr/reports/2010-02.pdf

[13] ARM V7 Cortex A15 Events, http://oprofile.sourceforge.net/docs/armv7-ca15-events.php

[14] D. Burger and T. Austin, *The Simplescalar Tool Set, Version 2.0*, In University of Wisconsin-Madison Computer Sciences Department Technical Report 1342, June 1997

[15] CACTI. HP Laboratories Palo Alto, CACTI 4.2, http://www.hpl.hp.com/, 2008

[16] M. R. Guthaus, J. S. Ringenberg, D. Ernst, T. M. Austin, T. Mudge, R. B. Brown, *Mibench: A free commercially representative embedded benchmark suite*, In *IEEE 4th Annual Workshop on Workload Characterization*, Austin, TX, December 2001

[17] F. Rawson, *MEMPOWER: A Simple Memory Power Analysis Tool Set*, In Technical report, IBM Austin Research Laboratory,2004.

[18] C. Isci and M. Martonosi, *Runtime power monitoring in high-end processors: Methodology and empirical data*, In *Proceedings of the 36th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO'36)* 2003. IEEE Computer Society, 93-104.

[19] J. Zedlewski, S. Sobti, N. Garg, F. Zhen, A. Krishnamurthy and R. Wang, *Modeling Hard-Disk Power Consumption*, In *Proceedings of the 2nd Conference on File and Storage Technologies*, San Franscisco, California, Mar 2003.

[20] T. Mudge, T. Austin and D. Grunwald, *The Simplescalar-Arm Power Modeling Project*, http://web.eecs.umich.edu/ panalyzer/

[21] A. Pathak, Y. C. Hu and M. Zhang, *Where is the energy spent inside my app?: fine grained energy accounting on smartphones with Eprof*, In *Proceedings of the 7th ACM european conference on Computer Systems*, Bern, Switzerland 2012. ACM, 29-42.