Hybrid System Level Power Modeling and Estimation of Embedded Processors

Sungchul Lee, Naeem Maroof, Jinman Kang, and Hyunchul Shin

***Abstract*— As the power dissipation becomes an important design constraint, especially in embedded systems, early and accurate power estimation is compulsory. The early power estimation dictates the design to meet the required specifications. In this paper, we describe efficient power modeling technique for embedded processors at higher level. We also present power models of two different processors using our methodology. Virtual Prototyping (VP) environment is used for benchmarking and power estimation using derived power models. Our methodology combines Functional Level Power Analysis (FLPA) with processor parameters, derived from processor counter information. Overall methodology applies Voltage and Frequency Scaling (VFS) along with FLPA and processor counters for the processor power modeling. We use a simulator to obtain such counters like total processor cycles and cache access cycles, which are highly dependent on algorithm. We have used an ARM™ embedded board for experimental power measurements. From real measured power data at different voltages, frequencies and cache access ratios we derive power models using regression for two embedded ARM™ processors. We used Carbon™ SoC Designer for VP of the system, and run different benchmark and integrated our power models to estimate processor power. Evaluation using four benchmark programs over different voltages and frequencies shows less than 9% and 4% errors for two processors. Our modeling techniques, as well as power models can be used for multicore processors.**

***Index Terms*—Power Estimation, Power Modeling, System Level, Virtual Prototype, Embedded Processors**

I. Introduction

WITH newer technological nodes, we have achieved more and more density, but with the breakdown of Dennard's scaling, we have already hit the power wall. Power has become first class design constraint and it dictates the performance [1]. With ever increasing performance demands of mobile computing platforms, many core processors have already been chosen as architecture for application processors. Multicores provided an alternative to boost the performance, but studies suggest that due to Dark Silicon, multicore performance will saturate because of power consumption [2]. Multicores are currently used as mainstream computing platform in embedded designs, especially in mobile applications. RISC based cores are the most widely used cores in embedded devices. We find rich profiles of computation powers of these platforms in current designs, ranging from a single core, to dual, to quad and even octa-processing cores in embedded domain. With some homogenous and some heterogeneous architectures, designers have their own justifications and constraints. Battery operated devices, e.g. mobile phones, require efficient utilization of energy source (battery). Manufacturers have innovated the ways to control (or reduce) the power consumption and boost the performance by using Power Gating (PG), Clock Gating (CG), Voltage and Frequency Scaling (DVS). But they still require power dissipation profiles for efficient power management. Power measurement devices are expensive and cannot find their way in mobile phones. They are hard to tackle for application developers and end-users as well. All of the modern computing SoCs have their own Power Management Units (PMUs) and certain Dynamic Power Management (DPM) policies at hardware level, software level or firmware level.

Most importantly, we need power profiles of processors for early system level design and optimization. Although true profiles are obtained from physical design, but in this case system cannot be optimized or modified to meet power or performance specifications. Thus we need power estimates of processors at higher level, so that different system designs could be explored and optimized for certain application area. Increasing complexity of the designs and emergence of multicore platforms requires designs to be analyzed at higher levels than RTL. Limited power budget and related power dissipation constraints require accurate early power estimation. Therefore, early power estimation of a system is necessary and critical. This have led to active research in early and accurate power estimation.

This paper addresses the problem of higher level power estimation of processor. In this article, we propose hybrid system level power modeling methodology for embedded processors, and use it to model power estimates of two different embedded processors from ARM™ [3]. We evaluate our estimated power results against real hardware measurements. Using four benchmark programs and extensively experimenting over different voltages and frequencies, our models show less than 9% and 4% errors for the two target processors used in the experiment.

Our power modeling methodology is based on FLPA [4], but is quite unique. We combined FLPA along with processor

Manuscript received February 5, 2014.
This work was supported in part by Samsung Electronics, Korea. Hardware and Software tools were provided by IDEC, Hanyang University.
Authors are with the Department of Electronics and Communication Engineering of Hanyang University, ERICA Campus, Ansan 426-791, South Korea. All correspondence should be directed at shin@hanyang.ac.kr
Tel: +82-31-400-4673

counters and have applied VFS. We used Carbon™ SoC Designer [5] to obtain processor counter information. We have used cache access and processor cycle counters to define cache access ratio. This cache access ratio is dependent on application. We used real power measurements to obtain true power consumption results for different voltages, frequencies, and cache access ratios. Using regression, we obtained parameters for processor power equation. Our method is fairly simple yet accurate.

For benchmarking purpose, we constructed a complete system in VP environment with Carbon™ SoC Designer and our power models, and verified our results using power measurements from the hardware board. We also present multicore power model by extending our single core power models, but we use estimation equation instead of measuring the power dissipation for multicore operations, since direct measurements are not easily feasible.

Rest of the paper is organized as follows. Section II describes the related works. In section III, we describe our methodology. Section IV presents power models, Section V describes evaluation, and Section VI finally concludes with future directions.

II. Related Works

Energy consumption estimation of a processor can be done at different levels, from Transistor, to Gate, to RTL, to Architecture, to System level, and can be further classified into different categories. Although lower level (physical design) power estimation gives accurate results, but it takes a lot of time and hence is not preferred for larger designs. Much work has already been done at RTL level, and certain commercial tools are also available which gives relatively accurate results. But due to complexity of current SoCs and time-to-market constraints, early power estimation (at higher level) has gained a lot of importance. At this level, we can get reasonably accurate results, as no technology information is available, and we make several assumptions and limit our scope due to huge design space. Our work is on high level power estimation of processors. Even at high level, power can be estimated using (1) Instruction Level (2) Component Level (3) Function Level (4) or Processor Event based modeling.

Many works has been done at system level. In [6], authors developed power models for different components of a system including CPU, and have used processor event counters. They accurately modeled the CPU power but their methodology is only viable for certain processors, as all processors do not have many counters to be monitored for events. Also, their domain is desktop / server, and hence their methodology is not feasible for embedded systems. In [7], power estimation method is described for processors and other components of mobile devices, but they have only considered a single core with frequency changes and utilization. M. Kim *et al.* considered multicore processors but again they only considered utilization and frequency [8]. S. Kumar *et al.* developed power estimation methodology for RISC based platforms and developed a power model for a single embedded processor [9]. Our methodology is similar to theirs with considerable improvements and is explained in section III. Reference [10] proposed component based power models for multicore processors but they have used fixed capacitance model for the different components of processors.

III. Methodology

We present Hybrid System Level Power modeling of embedded processors. Our methodology is a combination of FLPA and processor counter information, applied in conjunction with VFS to model processor power. Many previous works have only considered frequency of the processor to estimate power, but we have utilized voltage of the processor as well. Our models are highly dependent on voltage from the intuition that dynamic power of CMOS circuits is directly proportional to square of the voltage and leakage power is proportional to cubic power of the voltage. Carbon™ SoC Designer is used to obtain processor counter information. We used ARM™ embedded board for real measurements and used those readings for regression. Fig. 1 shows algorithm for our power modeling methodology, where *'PP'* is the list of processor parameters which are deemed most related with power consumption. These parameters are computed from the processor counters.

Algorithm: Hybrid System Level Power Modeling Methodology

- *Define PP, the list of processor parameters to be used*
  $PP=\{ pp_1, pp_2, \ldots pp_L \}$; *(L=# of parameters)*
- *Design macros to get different value of each pp in PP*
- *Obtain different values of pp using VP*
  $pp_1=\{p_{10}, p_{11}, \ldots p_{1K}\} \ \ldots \ pp_L=\{ p_{L0}, p_{L1}, \ldots p_{LJ} \}$

1. *Set:* $l:=1$, $P_c:= pp_l$, $S:=length(P_c)$
2. *Set:* $i:=1$, $cpv:=P_c[i]$ *(start from 1st value)*
3. *Set* $V_j=V_{min}$ *(lowest applicable voltage)*
4. *Set* $F_k=F_{min}$ *(lowest applicable frequency)*
5. *Run macro which generated 'cpv' on hardware board*
6. *Measure the power, P*
7. *Store results (cpv,* V_j, F_k, *P)*
8. *Set* $F_k=F_{k+1}$ *and repeat 5-8 till* $F_k<F_{max+1}$
9. *Set* $V_j=V_{j+1}$ *and repeat 4-9 till* $V_j<V_{j+1}$
10. *Set:* $i:=i+1$, $cpv:=P_c[i]$ *and repeat 3-10 till* $i<S+1$
11. *Set:* $l:=l+1$, $P_c:= pp_l$ *and repeat 2-11 till* $l<L+1$

- *Do the regression analysis using variable* $C_1 \ldots C_L$, *V, F*
- *Obtain power model*

Fig. 1: Hybrid System Level Power Modeling

First step is to design macros which, when run, generates different values of processor parameters. This populates a vector for each processor parameter. For example, we generated macros to get different values of cache access ratios. This parameter, cache access ratio, is related with total processor cycles and cache access cycles, which are processor counters obtained through Carbon™ SoC Designer. Next, we start with first parameter and choose its first value. V_j and F_k represents voltage and frequency of a core and we set these to minimum allowable voltage and frequency respectively. We run the macro on hardware and measure the power. We repeat this procedure for all possible values of frequency and voltage and measure the power consumption. Next we choose new value of the parameter and repeat above steps (as shown in algorithm). Next we choose another parameter and do experimentation again. Our methodology is generalized for any number of processor parameters chosen for processor power modeling.

Using this methodology, we developed power models for two ARM™ processors, with very different capabilities. Cortex™-A15 is computational intensive but consumes more energy while Cortex™-A7 is for less intensive jobs and is energy efficient. For our purpose of modeling, we defined one processor parameter, which is 'cache access ratio'. We obtained this parameter from processor counters for 'total processor cycles' and 'cache access cycles'. Value of this parameter can vary from 0 to 1, depending on running application. Our intuition was simple yet accurate. We specifically used cache access ratio as the only parameter effecting power dissipation of processor. Power measurement data is obtained for all voltage and frequency pairs for both processors, as supported by the hardware board. The regression analysis with the measured data gives the coefficients of these parameters for power equation. Our power estimation methodology using VP is shown in Fig. 2, where we also show the modeling step to describe the overall procedure.

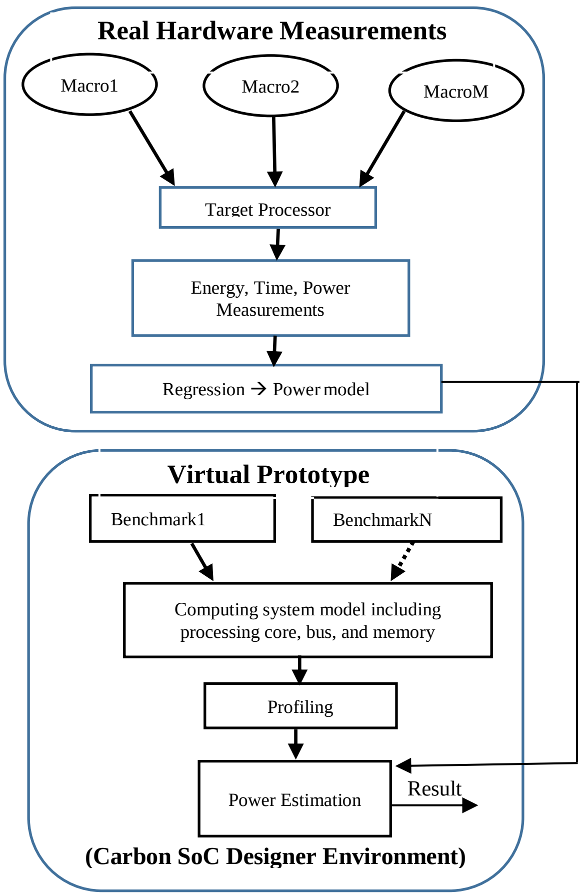


Fig. 2: Power Estimation: Overall flow of power estimation including hardware measurements for modeling and VP for simulation and estimation

Fig. 2 shows the development of power model using hardware measurements and virtual prototyping for system level simulation and estimation of power using power model and profile data. For the target processors, we changed macros to model cache access effect because accesses to the cache can lead to different amount of power consumption. We used these power model and constructed virtual prototype to demonstrate high level power estimation of processors. We have used ARM board for power measurements. Our target board is equipped with energy sensors. We reset the energy sensor using System Configuration Registers before the task is activated, and we read the value stored in energy register via same configuration registers. This gives us the total amount of energy that certain task has consumed. Energy consumed during time 'T' is related to average power consumption over time 'T' through (1), where 'T' is the execution time for a certain task.

$$P_{avg} = \frac{E_{total}}{T} \qquad (1)$$

Hence, the average power consumed during the computation of task is evaluated. We run same task for different voltage and frequency pairs. So, we obtained power for different tasks for each voltage and frequency pair. This is done because the two very important factors affecting the power are the frequency of operation and the voltage at which the processor is operating. Many works have not considered DVFS, they only model their equation on the basis of frequency, while our power models are highly dependent on the voltage of the processor plane. We then construct virtual prototype using Carbon SoC Designer. We run different tasks in that environment to get the profiling data. This data along with power model is used to provide the estimation of power consumption of the task at higher level.

IV. POWER MODELS

Fig. 3 shows the measured power of macros for modeling purpose for Cortex-A15 processor. Similar graphs are obtained for Cortex-A7 processor for modeling purpose. It shows linear relation of frequency and power, while non-linear relation of voltage and power. We also measured power for different cache access rates as described earlier.

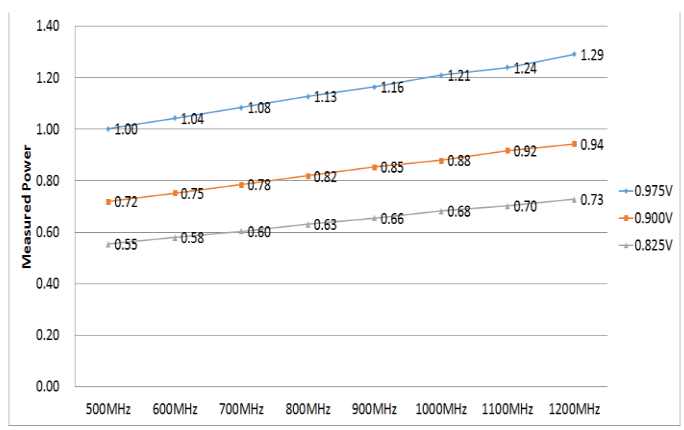


Fig. 3: Power consumption w.r.t voltage and frequency

This trend is modelled and presented in (2). This values of parametric-coefficients are obtained through regression analysis. This equation presents the power model for power dissipation of a processor, where α is the parameter for dynamic power of the operating core, modeling power as function of frequency multiplied with voltage squared. C is the cache access ratio (profile parameter), β is the parameter for dynamic cache power. γ is the static power coefficient and it shows high dependence on voltage. This equation is applicable to the active or 'ON' cores only. We didn't measure or model the leakage power of the processor, which is highly dependent on the underlying technology. Parameter ρ is correction coefficient. Table 1 shows these parameters for two different ARM processors.

$$P_n = \alpha f v^2 + \beta C v^2 + \gamma v^2 + \rho \qquad (2)$$

TABLE 1
PARAMETERS OF POWER MODEL

	$\alpha \times 10^{-3}$	$\beta \times 10^{-3}$	γ	ρ
Cortex-A15	39	4.7	1.58	-0.67
Cortex-A7	5.6	8.9	0.44	-0.2

Frequency f is in 100s of MHz and the resultant power is given in Watts. We define cache access ratio as the number of cycles cache is accessed to the total number of processor cycles taken for execution of a task. Different voltages and frequencies used for both processors are shown Table 2. Changing the voltage and frequency of operation is done via embedded system programming as we consider bare-metal benchmarking. We also utilized system level assembly programming to get the real measurement data from the board.

TABLE 2
VOLTAGE AND FREQUENCY TABLE FOR BOTH PROCESSORS

#	Frequency of Cortex-A15 *f(MHz)*	Voltage of Cortex-A15 *V(mV)*	Frequency of Cortex-A7 *f(MHz)*	Voltage of Cortex-A7 *V(mV)*
1	500	825	350	825
2	600	900	400	900
3	700	975	500	975
4	800		600	
5	900		700	
6	1000		800	
7	1100		900	
8	1200		1000	

We extend our model to be used for multicores. We consider MPSoC having N processors of type-1 and M processors of type-2. Power equation for this case is given in (3). This kind of equation is specifically useful for MPSoCs where we have different types of processors having different capabilities. This equation is applicable to a true heterogeneous multicore processor.

$$P = \sum_{i=1}^{N} P_{1i}\, x_{1i} + \sum_{k=1}^{M} P_{2k}\, x_{1k} \quad + P_{SM} \qquad (3)$$

In this equation, P is total estimated power of a multicore processor, N represents number of cores of type-1 processor and M represents number of cores of type-2 processor. P_{SM} is power estimate for the shared-memory, which is located outside of the cores but still on the chip and is shared by the cores. It may be a hierarchical memory where level-1 memory is shared between cores of the same type and level-2 memory is shared between cores of the other type. In (3), P_{1i} represents the power of i^{th} processor of type-1 and P_{2j} represents the power of j^{th} processor of type-2 processor. For the first and second type of processors, power is to be computed using (2) and parameters from Table 1. Here x_{1i} and x_{2k} represents the state of the processor. It can have value '0' or '1' indicating core is 'On' or 'Off' respectively. Thus (3) will estimate the combined power of a multicore processor where several cores are ON and others are OFF. Each of the cores can have its own voltage and frequency pair and can be executing different tasks. This equation doesn't model the inter-core communication, which effects the performance and the power consumption of the processor.

V. Evaluation

We ran different benchmark programs on target processors and measured the power consumption for each application program using an embedded ARM™ board and its built-in energy sensor. System registers were accessed to read and reset the register. We used Carbon™ SoC Designer for virtual prototyping (VP) of the systems, with which we constructed a complete system including processor cores (using IP integration). ARM™ DS-5 was used to compile and convert the benchmarks into the executables for ARM™ architecture. These executables were used for measurement on real board and in VP for estimation. By running these arm-executables in VP we get the profiling data. We integrated our power equation and this profiling data to output power estimation results for that benchmark on that specific processor. We repeated this for both of the target processors and for the different benchmarks listed below. We repeated our experiments several times to reduce experimental errors.

The benchmark programs along with their characteristics are shown in Table 3. We have used different programs to utilize different parts of cores. For example some programs exhaust 'Integer Unit', some initiate a lot of 'Memory Access', and some initiate a lot of swapping. We have not used programs to exploit the 'Floating point' unit.

TABLE 3
BENCHMARK PROGRAM DETAILS

Benchmark	Characteristics	Repetition
Matrix Multiplication	40 by 40 matrices & 100 by 100 matrices	100, 500 1000
Discrete Fourier Transform	40 & 400 numbers	100, 1000
Sorting	Insert, Shell, Quick Sort 500, 1000 numbers	100
Image Processing	64x64 image by 3x3 and 5x5 filter	1000

We measured power consumption for different kind of runs for same program. For example we did matrix multiplication of 40x40 matrices in one instance and 100x100 matrices in other. We also run these programs to run for different amount of time. For example, by putting a repetition loop of 100, 500 or 1000 in different cases. Fig. 6 to Fig. 13 presents the evaluation results. Each figure shows the estimated and measured power for different frequencies and voltage levels. Fig. 6 to Fig. 9 are power results of different benchmarks for Cortex-A15 and Fig. 10 to Fig. 13 are power results of different benchmarks for Cortex-A7. These graphs detail the estimated and measured power for different benchmark programs run over different frequencies and voltages. Execution of different benchmark programs also result in different execution times and they show different cache access ratios.

For the purpose of quantification, we define Percent-Error as the absolute of the ratio of error (which is difference between measured and estimated power data) to the measured data, converted to percentile (by multiplying with 100%). This error is given in (4).

$$\%Error = \left|\frac{P_{measured} - P_{estimated}}{P_{measured}}\right| x100\% \qquad (4)$$

We computed Percent-Error of our estimated power consumption using MATLAB™. We did a lot of experimentation using different characteristics (program inputs or variables, voltages and frequencies) and in the figures below, we plot percent-error for 72 different experiments for both processors. Fig. 4 shows the percent-error for Cortex-A15 and Fig. 5 shows the percent-error for Cortex-A7. For all the experiments, mean error is less than 4% and 2% and maximum error is less than 9% and 4% for Cortex-A15 and Cortex-A7 respectively.

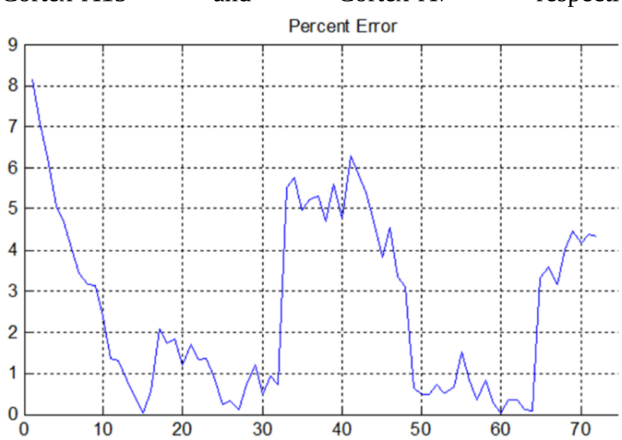


Fig. 4: Percent-Error for Cortex-A15 for each experimentation

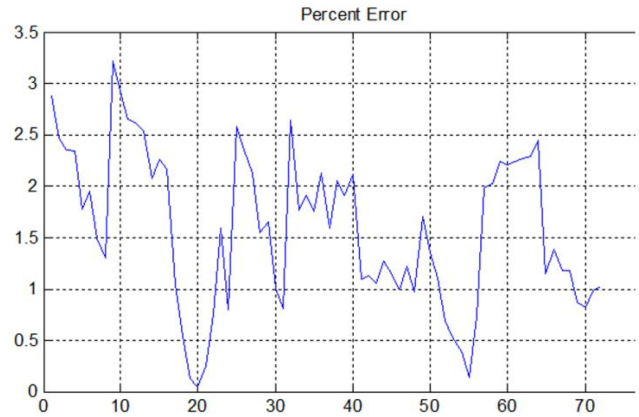


Fig. 5: Percent-Error for Cortex-A7 for each experimentation

Percent error is relatively higher for Cortex-A15 as compared with Cortex-A7 processor. This is because Cortex-A15 is much complex than Cortex-A7 core. As, for the purpose of modeling, we have used only one processor parameter for both of the processors. Single processor parameter is good enough to capture power dissipation of Cortex-A7 core but for Cortex-A15 it could not provide better estimate. There must be some more processor parameters which are highly correlated with power dissipation. Hence, to reduce the error, more processor parameters can be added which will result into more complex power estimation equation and also more extensive experimentation would be required. Fig. 6 shows the graphs for measured and estimated power of Cortex-A15 while running the matrix multiplication program. For the same processor, Fig. 7 to Fig. 9 show the measured and the estimated power graphs while running DFT, Sorting, and Image Processing (Filtering) benchmarks respectively. Fig. 10 shows the graphs for measured and estimated power of Cortex-A7 while running the matrix multiplication program. For the same processor, Fig. 11 to Fig. 14 show the measured and estimated power graphs while running DFT, Sorting, and Image Processing (Filtering) benchmarks respectively. From these graphs, we can see that measured and estimated power results are highly correlated for processor Cortex-A7 as compared to Cortex-A15. The reason, as explained earlier, is that these two are very different; Cortex-A15 is computationally intensive while Cortex-A7 is energy efficient and small; but we modeled both of the cores using same processor parameters and same form of power model equation. All of these results show that our models are very accurate at low voltage levels for both processors.

VI. Conclusion And Future Works

We presented our methodology to address the problem of power estimation at high level of design. Essentially our methodology is hybrid system level power estimation methodology in which we utilized FLPA (frequency in our model), Processor Counters (Cache access ratio), and applied VFS. We also presented power models for two modern embedded processors from ARM™ using our methodology. Results of power estimation using virtual prototyping environment are presented and we evaluated them against real board measurements. For selected benchmark program, percentile error among the measured and estimated power consumption results are less than 9% and 4% for Cortex-A15 and Cortex-A7 respectively. Our model is simple yet accurate and can be used at high level for processor power estimation for specific applications. Our methodology allows to use different processor parameters to be used for power modeling (in order to reduce the error). This methodology can be applied to any kind of embedded processors for which we know processor counter information. Also, we presented multicore power model for true heterogeneous multicore processors by extending our power models. Our future work is modeling accurate multicore power model and presenting new estimation methodology and comparing with existing methodologies in term of effort and accuracy. Real board measurements and experimentation with standard benchmarks are in our future plans. We are going to include other components of systems for power estimation.

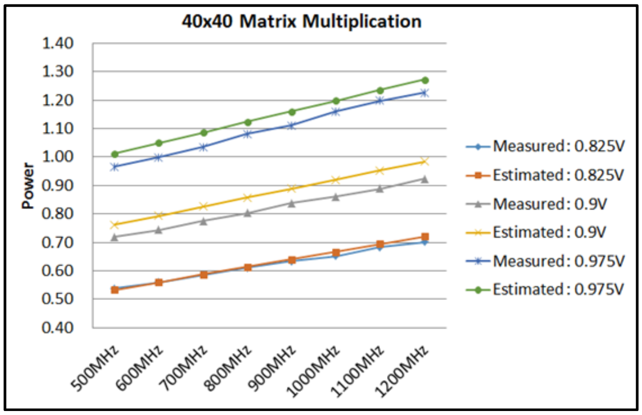


Fig. 6: Cortex-A15: Benchmark: Matrix Multiplication

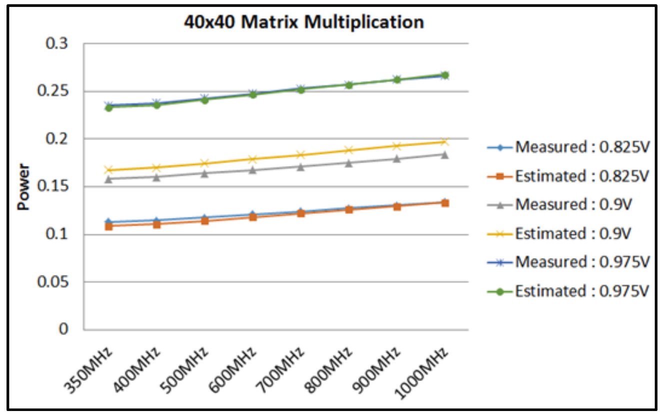


Fig. 10: Cortex-A7: Benchmark: Matrix Multiplication

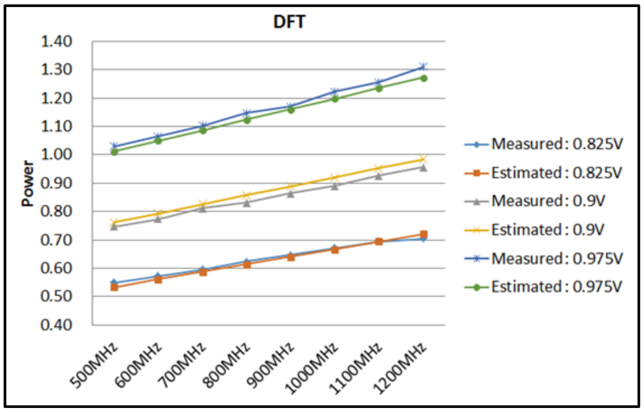


Fig. 7: Cortex-A15: Benchmark: DFT

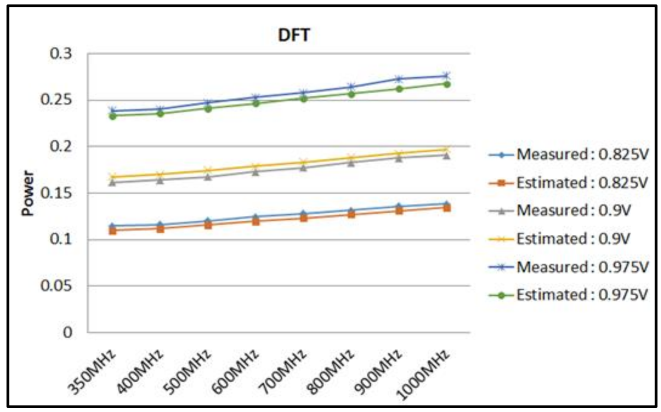


Fig. 11: Cortex-A7: Benchmark: DFT

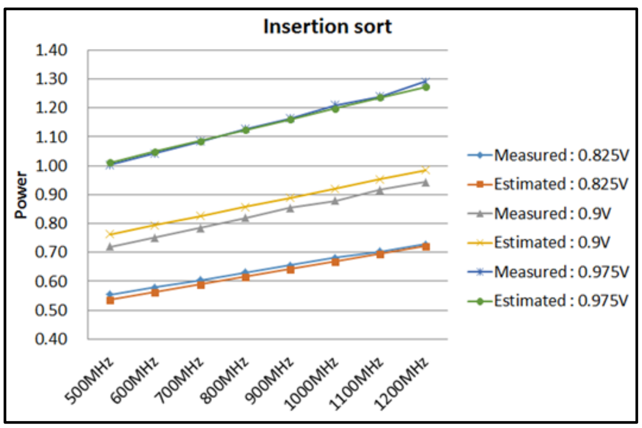


Fig. 8: Cortex-A15: Benchmark: Sorting

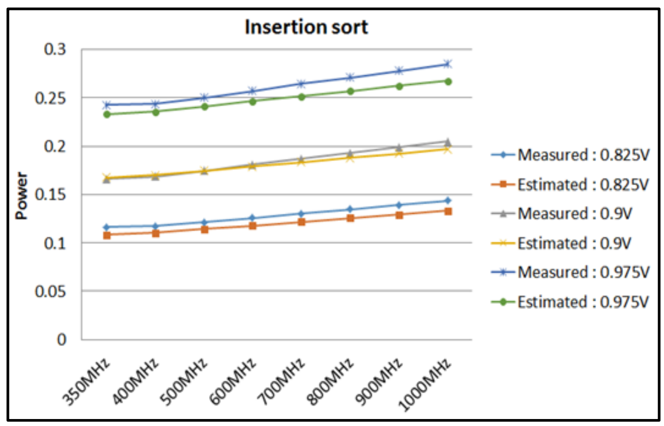


Fig. 12: Cortex-A7: Benchmark: Sorting

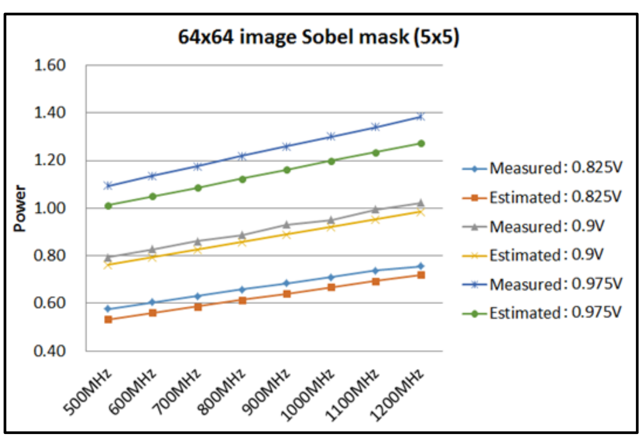


Fig. 9: Cortex-A15: Benchmark: Image Processing

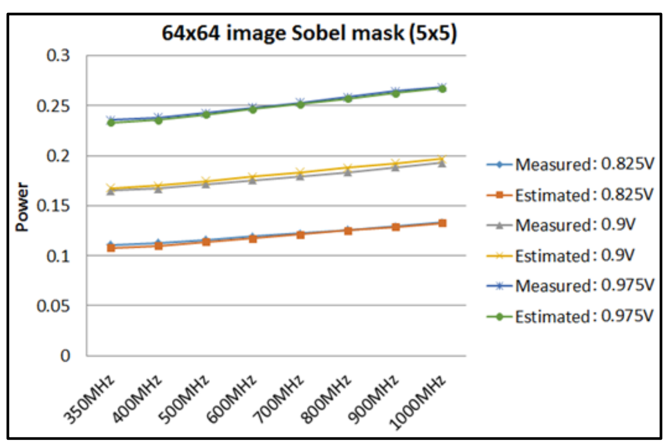


Fig. 13: Cortex-A7: Benchmark: Image Processing

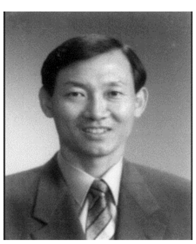
ACKNOWLEDGEMENTS

This work was partly supported by Samsung Electronics Co., Korea and the hardware / software tools were provided by IDEC Hanyang University, Korea.

Sungchul Lee received the BS and MS degree in Electrical Engineering from Hanyang University, Ansan, Korea, in 2001 and 2003. He is currently working toward the PhD degree at Digital Systems Laboratory of Electronics and Communication Engineering Department, Hanyang University, South Korea. His research interests include design repair, multi-core design and low power design techniques.

Naeem Maroof received the BS degree in Computer Engineering (CE) from CIIT Pakistan in 2006 and MS degree in Electronic Communications and CE from University of Nottingham UK in 2007. He is currently working toward the PhD degree at Digital Systems Laboratory of Electronics and Communication Engineering Department, Hanyang University, South Korea. His research interests include Digital Integrated Circuits, Low Power Design & Methodologies, and Multicore systems.

Jinman Kang received the BS degree in Electrical and Communication Engineering from Hanyang University, Ansan, Korea in 2013. He is currently MS student at Digital Systems Laboratory of Electronics and Communication Engineering Department, Hanyang University, South Korea. His research interests include Multicore systems, Low power design, embedded systems, and Hardware / Software Co-design.



Hyunchul Shin received the BS degree in Electronics Engineering from Seoul National University in 1978, the MS degree in Electrical Engineering from the Korea Advanced Institute of Science and Technology in 1980, and the PhD degree in Electrical Engineering and Computer Sciences from the University of California at Berkeley in 1987. From 1980 to 1983, he was with the Department of Electronics Engineering at the Kumoh National Institute of Technology, Korea. In 1983, he received a Fulbright scholarship. From 1987 to 1989, he was a member of the technical staff at AT&T Bell Laboratories, Murray Hill, New Jersey. Since 1989, he has been professor with the Department of Electronics and Communication Engineering of Hanyang University, South Korea. His research interests include VLSI Integrated Circuits, VLSI Computer Aided Design, Low power design methodologies, and Design and synthesis of integrated systems for multimedia & vision applications.

REFERENCES

[1] ITRS. Design, 2010 edition. http://public.itrs.net/
[2] H. Esmaeilzadehy, E. Blemz, R. St. Amant, K. Sankaralingam, D. Burger, "Dark silicon and the end of multicore scaling," International Symposium on Computer Architecture (ISCA), 2011 38th Annual, vol., no., pp.365,376, 4-8 June 2011
[3] ARM Ltd., www.arm.com
[4] J. Laurent, N. Julien, E. Senn, E. Martin, "Functional level power analysis: an efficient approach for modeling the power consumption of complex processors," Design, Automation and Test in Europe Conference and Exhibition, 2004. Proceedings , vol.1, no., pp.666,667 Vol.1, 16-20
[5] Carbon Design Systems, Inc. SoC Designer Plus http://www.carbondesignsystems.com/soc-designer-plus/
[6] W. Lloyd Bircher and L. K. John, "Complete System Power Estimation Using Processor Performance Events," Computers, IEEE Transactions on, vol.61, no.4, pp.563,577, April 2012 doi: 10.1109/TC.2011.47
[7] L. Zhang et al., "Accurate online power estimation and automatic battery behavior based power model generation for smartphones", Proceedings of CODES+ISSS 2010, Oct. 2010.
[8] M. Kim, J. Kong, S. W. Chung, (2012, May). "Enhancing Online Power Estimation Accuracy for Smartphones", IEEE Transactions on Consumer Electronics, Vol. 58, Issue 2
[9] S. Kumar Rethinagir, R. Ben Atitallah, Jean-Luc Dekeyser, S. Niar, E. Senn, "An efficient power estimation methodology for complex RISC processor-based platforms", Proceedings of the Great Lakes Symposium on VLSI, GLSVLSI '12, May 2012
[10] R. Basmadjian, H. de Meer, "Evaluating and Modeling Power Consumption of Multi-Core Processors", Third International Conference on Future Energy Systems: Where Energy, Computing and Communication Meet (e-Energy), 2012