

# 시스템 수준의 전력 예측을 위한 회귀분석에 기반하는 분석 방법.

김희석<sup>0</sup> 임채석 하순희  
서울대학교 전기 컴퓨터 공학부  
{netti<sup>0</sup>, csim, sha}@iris.snu.ac.kr

## Mixed Approach for Fast System-Level Power Analysis based on Regression Analysis

Hui-Seok Kim<sup>0</sup> Chae-Seok Im Soon-hoi Ha  
The School of Electrical Engineering and Computer Science, Seoul National University

### 요 약

이 논문에서는 시스템 수준의 전력 소모를 분석하는 방법론을 설명한다. 응용의 시스템 수준 전력 모델을 구하기 위해서, 시스템을 이루는 각 부분들을 선형적으로 모델링하고, 이를 모두 더한다. 선형적으로 모델링된 식의 파라미터들을 구하기 위해서, 회귀분석에 기반한 분석을 한다. 이를 위해서 다양한 벤치마크들을 준비하고, 응용에 대해서 측정을 한 것과 수정된 시뮬레이터에서 필요한 정보를 얻어야 한다. 이렇게 분석한 전력 모델의 예측치는 5% 내의 정확도를 가짐을 확인하였다.

### 1. 서 론

최근 이동용 기기를 위한 시스템의 설계 복잡도가 증가함에 따라, 시스템을 분석하는 다양한 도구들이 점점 더 필요하게 되었다. 특히 시스템의 에너지 소모와 관련한 분석과 예측 기법은 더욱 중요하게 되었다. 그러나 차별화되고 효과적인 분석 도구가 없는 것이 현실이다.

기존의 많은 연구들에서 마이크로프로세서, 메모리, 버스 같은 시스템 구성요소의 전력은 실행 사이클 또는 접근 횟수 같은 파라미터에 대해 선형적으로 모델링될 수 있다는 것을 보여준다. 이것은 시스템 수준의 전력 예측 모델 또한 선형적인 모델로써 설명할 수 있음을 뜻한다.

시스템 모델을 선형적으로 파악했을 때, 각 파라미터에 적용되는 계수들을 어떻게 구하는가 하는 것이 필수적이다. 일반적으로 이러한 계수들을 구하기 위해서 i) 데이터 시트를 참고하거나 ii) 개별적으로 측정을 하거나 iii) 회귀분석을 한다.

이 논문에서 우리는 위의 세 가지 방법들을 모두 통합하는 방법론을 제시한다. 각 방법들은 각각의 고유한 장 단점들이 있는데, 각 방법들이 제시하는 장점들만을 취합하는 것이 목적이다. 각 방법들을 독립적으로 사용하는 것 보다, 이렇게 통합적으로 접근한 방법이 더 효과적임을 결과에서 보여준다.

이 논문의 구성은 다음과 같다. 다음 장에서 관련 연구들, 특히 선형 수식의 계수들을 결정하기 위한 것을 중심으로 소개한다. 3장에서는 시스템의 에너지 모델을 설명하고, 4장에서 이를 풀어내기 위한 방법론을 제시한다. 5장에서는 실험의 결과와 검증을 보여주고 6장에서 결론을 내었다.

### 2. 관련된 연구들

마이크로프로세서 기반의 컴퓨팅 시스템과 각 하부 부분들은

그 모델링의 수준에 상관없이 전력 소모를 선형적으로 표현할 수 있음을 기존의 많은 연구들에서 보여준다. 이렇게 시스템을 모델링한 수식의 계수들은, 실행 시간이나 접근 회수, 그리고 상태의 전이 횟수 등 각 부분들의 가장 지배적인 전력 소모에 대한 전력화 값이다. 따라서 이러한 계수들을 결정하는 것이 문제의 핵심이며, 이를 위해서 시뮬레이션을 하거나 측정을 한다.

이러한 계수를 결정하는 데에는 크게 세 가지 방법이 있다. 데이터 시트를 참고, 개별적인 측정, 회귀분석이 그것들이다.

우선 데이터 시트를 참고하는 방법은 전력화 계수값들을 데이터 시트를 보고 결정하는 것이다. [3, 4, 5, 6]의 연구들이 이 방법을 적용해서 시스템 수준의 전력 예측을 하였다. 또한 [7]의 연구에서는 버스와 캐쉬의 전력 소모를 bit line의 상태 전이 횟수에 대한 선형식으로 모델링하고, 이를 데이터 시트를 참고해서 분석하였다. 그러나 이 방법은 일반적으로 다양한 응용에 대해서 유연하지 못하다는 단점이 있다.

두 번째 방법인 개별적인 측정은, 특정한 계수들을 간단한 벤치마크 프로그램을 사용해서 얻는 방법이다. [8]의 연구는 마이크로프로세서의 전력을 모델링하기 위해서 우선 개별적인 instruction과 inter-instruction의 전력을 측정하고, 이를 instruction순서에 맞추어서 최종 전력을 구한다. [9]의 연구에서는 SDRAM의 전력을 구하기 위해서 energy state machine을 설정하고, 각 state에서 전력을 측정하고 state의 transition의 에너지를 측정해서, energy state machine을 완성하였다. 이 방법은 실제 응용에 매우 정확한 계수 값을 얻을 수 있지만 모든 경우에 적용하기는 힘들다.

마지막 방법은 계수들을 통계적 방법인 회귀분석을 통해서 얻는다. [10]의 연구에서 RT-level, [11]의 연구에서 instruction-level의 분석을 하였다. 이 방법은 가장 강력하지만 동시에 자유도가 너무 크기 때문에 많은 데이터가 필요하다.

### 3. 시스템 수준의 전력 모델링

이 장에서는, 시스템 수준의 전력 모델링을 어떻게 하는가를 보여준다. 일반적인 마이크로프로세서 기반 시스템의 몇 부분들을 모델링하고, 우리가 대상으로 하는 응용에 대해서 어떻게 수식화 했는지 설명한다. 그림1은 대상으로 하는 개발보드의 대략적인 구조를 보여준다. 이미 언급하였듯이, 시스템 수준의 전력은 시스템을 이루는 각 개별 부분들을 선형적으로 모델링한 것들의 합으로 표현할 수 있다. 따라서 전체적인 전력 소모 수식은 다음과 같다.

$$E_{System} = \sum_i E_{Component}(i) + E_{Static} \quad (1)$$

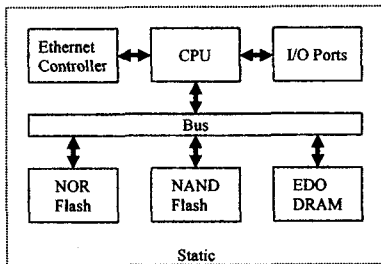


그림 1. EP7209 개발 보드의 블록 구성도

#### 1.1 마이크로프로세서

마이크로프로세서는 크게 active와 idle의 두 가지 전력 상태를 가진다. active 상태는 코어가 내부의 캐쉬를 사용해서 instruction을 수행하는 상태이고, idle 상태는 코어가 cache miss로 인해 기다리고 있는 상태이다. 이러한 전력 모델은 프로세서의 전력을 대부분 차지하는 것이 캐쉬이며, 캐쉬가 active하는 상태가 곧 프로세서가 active한 상태라는 사실에 기반한다[8]. 따라서 프로세서의 전력 소모는 다음과 같이 표현할 수 있다.

$$E_{Cpu} = P_{CpuActive} \times T_{CpuActive} + P_{CpuIdle} \times T_{CpuIdle} \quad (2)$$

우리가 대상으로 하고 있는 프로세서는 ARM720T기반의 프로세서로, 내부에 8Kbyte unified cache를 가지고 있다. 따라서 수식2를 사용해서 모델링할 수 있다.

#### 1.2 메모리

마이크로프로세서가 cache miss를 발생하면 메모리로의 접근이 이루어진다. 메모리로의 접근은 크게 nonsequential, sequential 접근과 접근 형태로 read와 write가 있으므로, 4가지 고유한 접근 모드를 가진다. [9, 12] 연구에서 메모리의 전력 소모를 접근 횟수로 모델링할 수 있음을 보여준다.

$$E_{Memory} = E_{Nonsequential\_Read} \times N_{Nonsequential\_Read} + E_{Nonsequential\_Write} \times N_{Nonsequential\_Write} + E_{Sequential\_Read} \times N_{Sequential\_Read} + E_{Sequential\_Write} \times N_{Sequential\_Write} \quad (3)$$

우리가 대상으로 하고 있는 응용은 DRAM과 NOR Flash가 Read-only 메모리로 쓰인다. DRAM의 경우는 수식3으로 설명할 수 있고, NOR Flash는 수식3에서 write가 없으며, 또한 nonsequential/sequential 접근의 구분이 없다.

#### 1.3 버스

버스는 시스템간의 연결을 하는 부분이다. 버스는 데이터의 이동으로 인한 전기적 상태가 바뀌는 것에서 대부분의 전력을 소모한다. 일반적으로 버스가 float 버스이면, Hamming distance를 구하는 것으로 충분히 전력을 모델링할 수 있지만, pull-up 버스라면 버스에 실리는 0의 개수가 더 큰 부분을 차지할 수도 있다. Hamming distance를 구하는 식은 다음과 같다.

$$E_{Bus} = E_{HammingDistance} \times N_{BitToggles} \quad (4)$$

우리가 대상으로 하고 있는 응용은 data, address 버스 모두 float 버스이므로, 수식4로 모델링할 수 있다.

#### 1.4 시스템 전반의 static power

static power는 시스템 전반에 걸친 전력 소모를 뜻한다. 이것은 모델링하지 않아도 되거나, 사용하지 않는 부분들이 소모하는 전력의 합이다. 이 값은 응용에 따라 다르므로, 개별적인 측정으로 알아내는 것이 가장 좋은 방법이다.

### 4. 제안하는 방법론

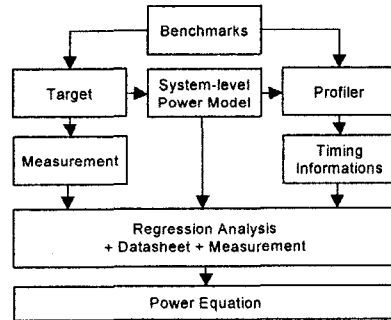


그림 2. 시스템 수준의 전력 모델을 구성하는 계수를 분석하기 위한 전체적인 흐름도

이 장에서는 회귀분석에 기반하고 있는 전력 분석 방법론을 제시한다. 그림2는 이 방법의 흐름을 보여준다. 분석을 할 응용 시스템이 결정되면, 위의 장에서 설명한 모델링 방법을 이용해서 전력 모델 수식을 얻어낸다. 이 수식의 전력과 값에 곱해질 각 구성 요소들의 파라미터 값들을 파악해야 하기 때문에 주어진 모델링을 바탕으로 필요한 파라미터들을 얻어내기 위해서 Profiler를 수정한다. 이렇게 수정된 Profiler와 응용 시스템을 사용해서, 다양한 벤치마크 프로그램을 이용한 전력 측정과 시뮬레이션으로 필요한 파라미터들을 얻어낸다. 파라미터 특성이 비슷한 벤치마크는 회귀분석에 도움이 되지 않는다. 즉, 각 벤치마크 프로그램들이 서로 linearly independent한 특성을 지니게 하는 것이 필요하다.

아주 세밀한 모델링을 하거나 시스템의 구성요소가 많아져서 회귀분석에 참여한 변수들이 많아지면, 참여한 변수의 약 두 배 정도에 비해서 서로 linearly independent한 벤치마크 프로그램이 필요하게 된다. 변수들이 많아지게 된다면 이것은 상당히 풀기 어려운 문제이다. 우리가 가진 파라미터의 변수 공간은 그렇게 크지 않은데, 여기에서 독립적인 특성을 보이는 벤치마크 프로그램을 많이 생성한다는 것은 어려운 일이기 때문이다. 따라서 문제의 크기를 줄이기 위해서 데이터 시트를

참고하거나 독립적인 측정의 도움을 받는 방법을 사용했다. 데이터 시트를 참고하는 것은 쉽고 간단하지만 다양한 환경과 응용에 항상 사용할 수 있는 것은 아니다. 개별적인 측정은 정확도가 높지만 측정하고자 하는 계수만 독립적으로 측정가능해야 한다는 단점이 있다. 데이터 버스의 값이나 static power같은 계수들을 개별 측정을 통해서 효과적으로 알아낼 수 있다. 우리는 이와 같은 방법들을 병행해서 사용함으로써 회귀분석에 적용되는 변수의 숫자를 줄일 수 있었고, 필요한 벤치마크 프로그램 숫자를 줄임으로써 간단히 계수들을 구할 수 있었다.

5. 실험 결과

실험과 측정을 위해서 5개의 멀티미디어 프로그램과 4개의 일반 프로그램을 사용했다. 9개의 변수들과 관측치가 얻어지므로 4개의 변수에 대해 회귀분석 가능하지만 통합적인 방법으로 변수의 크기를 줄이면서도 효과적으로 결과를 얻을 수 있었다.

표 1. 데이터 시트에서 참고한 계수들

$P_{ProcActive}$	0.170
$P_{Rom\ Nonsequential\ Read}$	0.132
$P_{Rom\ Sequential\ Read}$	0.132

표 2. 개별 측정으로 얻은 관계식들

$P_{ProcIdle}$	$0.347 \times P_{ProcActive}$
$P_{RamNonsequentialWrite}$	$P_{RamNonsequentialRead}$
$P_{RamSequentialWrite}$	$P_{RamSequentialRead}$
$P_{DataBus}$	$2.06 \times P_{AddressBus}$

표 3. 회귀분석으로 얻은 계수들

$P_{RamNonSequentialRead/Write}$	$2.927 \times 10^{-8}$
$P_{RamSequentialRead/Write}$	$5.448 \times 10^{-9}$
$P_{DataBus}$	$1.792 \times 10^{-10}$
$P_{AddressBus}$	$3.692 \times 10^{-10}$
$P_{Static}$	0.496

표1, 표2, 표3은 각각 데이터 시트에서 참고한 값, 개별 측정으로 얻은 관계식, 회귀분석으로 얻은 계수들을 나타내고 있다. 표4는 이렇게 얻은 전력 계수들을 벤치마크에 적용했을 때의 수렴도를 나타낸다. 그림에서 볼 수 있듯이 5%의 오차로 전력 수식과 실제 측정치가 일치하는 것을 볼 수 있다. 이 결과는 선형적으로 모델링한 것이 타당하다는 것을 보여준다.

표 4. 회귀분석으로 구한 전력 식을 적용한 수렴도

Unit : mW	Estimated	Measured	Error (%)
adpcm enc	0.7617	0.7795	-1.4
adpcm dec	0.7516	0.7590	-0.1
epic enc	0.7271	0.7430	-0.7
jpeg enc	0.7997	0.8320	-2.5
mp3 enc	0.7476	0.7613	-1.8
matrix r/w	0.9308	0.9263	-0.1
prime test	0.8366	0.8191	4.0
memcpy	0.8265	0.8307	3.2
Hanoi tower	0.7906	0.8300	-4.7

표5는 회귀분석으로 얻은 계수들을 회귀분석에 참가하지 않은 벤치마크 프로그램에 적용해서 그 결과를 확인한 것이다. 표에서 볼 수 있듯이, 4% 내외의 정확도를 보여준다. 이 결과는 전력 모델과 이 수식에 방법론을 적용한 것이 시스템의 전력 소모를 잘 설명한다는 것을 뜻한다.

표 5. 제안한 방법으로 얻은 계수들을 다른 벤치마크에 적용한 결과

Unit : mW	Estimated	Measured	Error (%)
mpeg dec	0.8645	0.8940	-3.2

h.263 dec	0.7841	0.7742	1.4
insert sort	0.7609	0.7920	-3.8
shell sort	0.7735	0.7689	0.8

6. 결론

이 논문에서는 효과적인 시스템 수준의 전력 분석 방법을 제시했다. 시스템 수준의 전력 분석을 위해서 각 부분들을 선형적으로 모델링하고 이것을 회귀분석으로 분석하는데, 데이터 시트를 참고하고 개별적인 측정으로 문제의 크기를 줄이는 방법을 적용하였다. 이 방법론은 더 나은 모델이 개발되거나 시스템의 설정이 바뀌어도 적용할 수 있다는 장점이 있다. 그러나 데이터 시트를 참고할 수 없거나, 개별적인 측정으로 많은 정보를 얻지 못하는 환경이거나, 변수들의 개수가 많아지면 이 방법은 근본적인 문제를 가진다. 문제 공간의 크기를 줄일 수 없다면 회귀분석은 파라미터 공간의 제약 때문에 상당히 어려울 수 있다.

좀 더 정확한 결과를 위해서 앞으로 할 일은, 데이터 시트를 참고하거나 개별 측정을 해서 적용하는 계수들이 얼마나 타당한 것인지 검증할 수 있는 방법을 고안하는 것이다.

7. 참고문헌

- [1] E. Macci, M. Pderam, and F. Somenzi, "High-Level Power Modeling, Estimation, and Optimization," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 17, no. 11, pp. 1061-1079, November 1998.
- [2] L. Benini and G. De Micheli, "System-Level Power Optimization: Techniques and Tools," *ACM Trans. on Design Automation of Electronic Systems*, vol. 5, no. 2, pp. 115-192, April 2000.
- [3] L. Benini, R. Hodgson, and P. Siegel, "System-Level Power Estimation and Optimization," *Proc. of ISLPED*, pp. 173-178, 1998.
- [4] T. Simunic, L. Benini, and G. De Micheli, "Cycle-Accurate Simulation of Energy Consumption in Embedded Systems," *Proc. of DAC*, pp. 867-872, 1999.
- [5] T. Simunic, L. Benini, and G. De Micheli, "Energy-Efficient Design of Battery-Powered Embedded Systems," *IEEE Trans. on VLSI Systems*, vol. 9, no. 1, pp. 15-28, February 2001.
- [6] Y. Li and J. Henkel, "A Framework for Estimating and Minimizing Energy Dissipation of Embedded HW/SW Systems," *Proc. of DAC*, pp. 188-193, 1998.
- [7] T. D. Givargis, F. vahid, and J. Henkel, "Fast Cache and Bus Power Estimation for Parameterized System-on-a-Chip Design," *Proc. of DATE*, pp. 333-338, 2000.
- [8] V. Tiwari, S. Malik, and A. Wolfe, "Power Analysis of Embedded Software: A Fast Step towards Software Power Minimization," *IEEE Trans. on VLSI Systems*, vol. 2, no. 4, pp. 437-445, December 1994.
- [9] Y. Joo, Y. Choi, H. Shim, H. Lee, and N. Chang, "Energy Exploration and Reduction of SDRAM Memory Systems," *Proc. of DAC*, pp. 892-897, 2002.
- [10] A. Bogliolo, L. Benini, and G. De Micheli, "Regression-Based RTL Power Modeling," *ACM Trans. on Design Automation of Electronic Systems*, vol. 5, no. 3, pp. 337-372, July 2000.
- [11] S. Lee, A. Ermedahl, and S. Min, "An Accurate Instruction-Level Energy Consumption Model for Embedded RISC Processors," *Proc. of LCTES*, pp. 1-10, 2001.
- [12] K. Itoh, K. Sasaki and Y. Nakagome, *Trends in Low-Power RAM Circuit Technologies*, Pro. Of the IEEE, VOL. 83, No. 4, 1995.
- [13] F. Cathoor, S. Wuytack, E. D. Greef, F. Balasa, L. Nachtergaele, and A. Vandecappelle, *Custom Memory Management Methodology - Exploration of Memory Organisation for Embedded Multimedia System Design*, Kluwer Academic Publishes, 1998.