임베디드 마이크로프로세서에서 산술 및 논리 명령어에 대한 전력 예측 모델

신동하* · 강경희**

A Power Estimation Model for Arithmetic and Logic Instructions of Embedded Microprocessors

Dongha Shin* · Kyunghee Kang**

본 연구는 2006년 상명대학교 소프트웨어 · 미디어 연구소에서 연구비를 지원받았음

요 약

임베디드 마이크로프로세서가 소프트웨어를 수행하면서 소비하는 전력을 예측하기 위해서는 마이크로프로세서의 각 명령어가 수행하면서 소비하는 전류를 측정하여 활용한다. 본 논문에서는 임베디드 마이크로프로세서 adc16s310의 산술 및 논리 명령어에 대한 소비 전류를 측정 및 분석하고, 이를 바탕으로 적은 수의 측정 소비 전류 값을 사용하여 비교적 정확하게 모든 명령어 수행의 소비 전류 값을 예측할 수 있는 전력 예측 모델을 제안한다. 본 예측 모델은 마이크로프로세서 adc16s310의 산술 및 논리 명령어에 대하여 총 측정 공간 중 약 5.84%의 공간에 대한 측정 전류 값만을 사용하여 평균 오차 0.34%에서 소비 전류 값을 예측할 수 있다.

ABSTRACT

In order to estimate the power consumed by an embedded microprocessor during an execution of software, we measure and utilize the current consumed by the processor during the execution of each instruction. In this paper, we measure and analyse the current consumed by the microprocessor adc16s310 during the execution of arithmetic and logic instructions, and propose a power estimation model which estimates the current for all instruction executions precisely by using a small numbers of current measurements. The proposed model can estimate the current with an average 0.34% error by using only 5.84% of total current measurements for arithmetic and logic instructions of the processor.

키워드

microprocessor, embedded system, instruction, power consumption, power estimation

Ⅰ. 서 론

임베디드 시스템 연구에 있어서 저 전력(low power) 설계는 중요한 연구 분야 중의 하나이다. 저 전력 설계와 관련된 연구 분야는 회로(circuit) 단계, 논리(logic) 단계, 구조(architecture) 단계 및 소프트웨어(software) 단계로 나

* 상명대학교 소프트웨어학부 부교수

** 상명대학교 대학원 컴퓨터과학과 석사과정

접수일자 : 2006. 5. 30

누어질 수 있다[1]. 본 연구는 소프트웨어 단계에서의 저전력 연구의 하나로서 임베디드 마이크로프로세서가 소프트웨어를 수행하면서 소비하는 전력을 예측하기 위하여 마이크로프로세서의 각 명령어가 수행하면서 소비하는 전류를 측정하고 예측하는 방법에 대한 연구이다[2][3]. 이러한 연구 결과와 컴파일러 기술을 적용하면 마이크로프로세서가 소프트웨어를 수행하면서 소비하는 전력을 줄일 수 있는 방법을 고안할 수 있다[4][5]. 일반적으로 임베디드 마이크로프로세서가 많은 명령어를 가지고 있고, 하나의 명령어도 오퍼랜드 값들의 조합에 따라 많은 다른 수행이 존재하기 때문에 명령어에 대한 모든 가능한 수행에 대하여 소비 전류를 측정하여 사용한다는 것은 비현실적이다. 예를 들어 본 연구에서 사용하는 16비트 임베디드 마이크로프로세서인 adc16s310[6][7]의 경우에 오퍼랜드로 사용되는 레지스터에 저장된 값은 고려하지 않더라도 42,866개의 서로 다른 명령어 수행이 존재하기 때문에 이들 모든 경우에 대하여 소비 전류 값을 측정하기 위해서는 매우 오랜 시간이 필요하다. 본 연구에서는 적은 회수의 명령어 수행에 대하여 측정한 전류 값을 사용하여 명령어의 모든 수행에 대한 소비 전류 값을 비교적 정확하게 예측할 수 있는 전력 예측 모델을 제안한다. 본 논문에서 제안한 전력 모델을 임베디드 마이크로프로세서인 adc16s310의 산술 및 논리 명령어에 적용해 본 결과 측정 공간을 크게 줄이면서도 오차 0.34%에서 정확하게 소비 전력을 예측할 수 있음을 전류 측정 실험을 통하여 보였다.

본 연구는 기존의 유사 연구인 [2], [3], [8] 및 [9]와 비교하여 다음과 같은 장점 및 특징이 있다. V. Tiwari 연구팀이 주도한 연구 [2] 및 [3]에서는 본 연구에서의 예측 모델과는 달리 각 명령어에 대하여 무작위로 선정된 하나의 수행에 대한 측정값을 사용하였다. 본 연구에서는 명령어와 더불어 명령어의 오퍼랜드 값도 고려한 예측 모델이여서 V. Tiwari 연구팀이 제안한 방법의 오차인 3%보다 더 낮은 오차를 가진다. S. Lee 연구팀이 수행한 연구 [8]은 실험적인 방법과 통계적인 방법을 동시에 사용한 에너지 예측 모델로서 통계적인 방법을 통하여 마이크로프로세서에 의존적이지 않는 모델을 제시하였다. 본 연구는 마이크로프로세서에 의존적인 측면이 있을 수 있지만 S. Lee 연구팀이 제안한 모델에 비하여 적용하기 간단하면서도 평균 오차 2.5%보다 낮은 오차를 가진다. 또한 대부분의 기존 연구는 Intel 486DX2, Fujitsu SPARClite 934 및 ARM7TDMI과 같은 외국산 마이크로프로세서를 사용하여 실험하였지만 본 연구는 국내에서 개발된 16비트 임베디드 마이크로프로세서인 adc16s310[6][7]을 사용하여 실험하였다는 점도 특징이다.

본 논문의 2장에서는 마이크로프로세서가 명령어를 수행하면서 소비하는 전류를 측정하는 방법을 기술하고, 본 연구에서 사용하는 마이크로프로세서 adc16s310의 산술 및 논리 명령어의 구성에 대하여 설명한 후, 이들 명령어에 대한 전류 측정 결과를 기술한다. 3장에서는 2장에서 측정한 명령어 단위 소비 전류의 특성에 관하여 기술한다. 분석 결과 명령어에서 사용되는 오퍼랜드인 레지스터 혹은 직접 데이터 값에 따라 소비 전류에 큰 차이가 있음을 알아내었다. 4장에서는 3장에서 분석한 소비 전류의 특성을 이용하여 소비 전류를 예측하는 모델을 정의한다. 제안된 모델은 전체 측정 공간의 약 5.84%에 대한 측정을 사용하여 전체 명령어 수행에 대한 소비 전류를 예측할 수 있다. 5장에서는 결론으로 본 연구의 의의 및 앞으로 더 연구할 분야 등에 대하여 설명한다.

Ⅱ. 소비 전류 측정

2.1. 소비 전류 측정 환경

마이크로프로세서가 어떤 하나의 명령어를 수행하면서 소비하는 전류를 측정하기 위해서는 하드웨어 환경과 소프트웨어 환경의 설치가 필요하다. 본 연구에서는 기존 연구 [2]에서 사용한 측정 환경을 따른다. 하드웨어 환경은 측정할 마이크로프로세서가 탑재된 보드, 측정 프로그램을 개발하고 교차 컴파일(cross compile)할 호스트 PC, 직류 전원 공급기, 전류 측정기로 구성된다. 하드웨어 환경이 설치되면 측정 시 수행할 프로그램을 작성하여 측정 보드에 다운로드하여 수행시킨다. 이 프로그램은 마이크로프로세서가 사용하지 않는 장치(예: USB, DMA, UART, PIO, TIMER 등)를 사용하지 않게(disable)하고, 사용자 레지스터를 초기화 한 후, 무한 루프 속에서 주어진 명령어를 반복 수행할 수 있도록 프로그램한다. 아래 그림은 명령어 "add %r0, 0x01, %r2"의 소비 전류 측정 시 수행할 프로그램의 모습이다.

```
int main(void)
{
  disable_devices();
  initialize_registers();
 loop:
  asm("add %r0, 0x01, %r2");
  asm("add %r0, 0x01, %r2");
    ...
  asm("add %r0, 0x01, %r2");
  goto loop;
}
```

그림 1. 소비 전류 측정 프로그램
Fig. 1. Current Measurement Program

2.2. 전류 측정 공간

마이크로프로세서 adc16s310는 총 9개의 산술 및 논리 명령어를 가지고 있고 이들 명령어들은 레지스터 오퍼랜드에 저장된 값은 고려하지 않더라도 총 5,257개의 서로 다른 수행을 가질 수 있다. 표 1은 산술 및 논리 명령어의 어셈블러 문법과 총 측정 공간을 나타낸 표이다. 본 연구에서는 정밀한 소비 전류 모델을 도출하기 위하여 이들 공간에 대하여 전류 측정을 수행하였다.

Ⅲ. 소비 전류 특성 분석

3.1. 명령어 종류와 소비 전류

2장에서 기술한 측정 환경에서 총 9개의 산술 및 논리 명령어에 대하여 소비 전류의 양을 측정 해본 결과 명령어 종류에 따라서 평균 전류 소비에 차이가 있음을 발견하였다. 산술 명령어의 경우 명령어 'add' 및 'adc' 보다 명령어 'sub' 및 'sbc'가 더 많은 전류를 소비하였고, 명령어 'cmp'는 최소의 전류를 소비하였다. 논리 명령어의 경우 명령어 'and' 및 명령어 'or'는 비슷한 전류 소비를 보였으며, 명령어 'xor'는 좀 더 높은 전류 소비를 보였고, 명령어 'cvb'는 최소의 전류 소비를 보였다. 명령어 종류별 평균 소비 전류의 최대 차이는 명령어 'cvb'와 명령어 'sbc'로서 1.57%의 차이를 보였다. 그림 2는 각 명령어의 평균 소비 전류를 그래프로 나타낸 그림이다.

표 1. 산술 및 논리 명령어의 전류 측정 공간
Table 1. Current Measurement Spaces for Arithmetic and Logic Instructions

어셈블러 문법	측정 회수
add %SrcRn, ImmData, %DesRn	392
add %Src1Rn, %Src2Rn, %DesRn	343
sub %SrcRn, ImmData, %DesRn	392
sub %Src1Rn, %Src2Rn, %DesRn	343
adc %SrcRn, ImmData, %DesRn	392
adc %Src1Rn, %Src2Rn, %DesRn	343
sbc %SrcRn, ImmData, %DesRn	392
sbc %Src1Rn, %Src2Rn, %DesRn	343
cmp %SrcRn, ImmData	56
cmp %Src1Rn, %Src2Rn	49
and %SrcRn, ImmData, %DesRn	392
and %Src1Rn, %Src2Rn, %DesRn	343
or %SrcRn, ImmData, %DesRn	392
or %Src1Rn, %Src2Rn, %DesRn	343
xor %SrcRn, ImmData, %DesRn	392
xor %Src1Rn, %Src2Rn, %DesRn	343
cvb %DesRn	7
합계	5,257

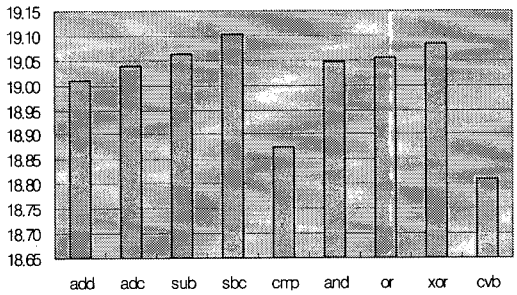

그림 2. 명령어 별 평균 소비 전류 (mA)
Fig. 2. Average Currents (mA)

그림 3은 각 명령어에서 최소 소비 전류 값과 최대 소비 전류 값을 그래프로 나타낸 그림이다. 같은 명령어라도 오퍼랜드에 따라 평균 5.86%, 최고 7.06%의 차이가 있음을 나타낸다.

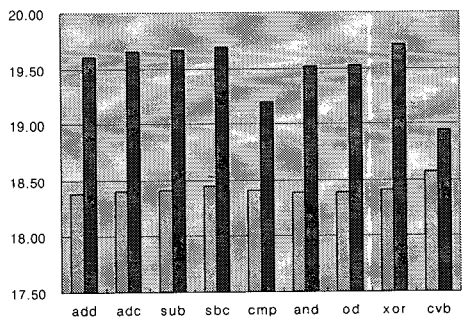

그림 3. 명령어 별 최소 및 최대 소비 전류 (mA)
Fig. 3. Min and Max Currents (mA)

3.2. 레지스터 번호와 소비 전류

같은 명령어를 수행하더라도 사용되는 소스(source) 및 목적(destination) 레지스터에 따라 소비 전류에 큰 차이가 나타남이 발견되었다. 이 현상은 이미 이전 연구에서도 언급된 바 있다[8]. 마이크로프로세서 adc16s310는 총 7개(%r0 ~ %r6)의 사용자 레지스터가 있는데 소스 레지스터 번호에 따른 소비 전류의 순서는 %r5, %r6, %r3, %r4, %r2, %r1, %r0이고, 평균 1.14%, 최대 2.09%의 차이를 보였다. 목적 레지스터 번호에 따른 소비 전류의 순서는 %r6, %r3, %r5, %r4, %r2, %r1, %r0 순서이며, 평균 1.26%, 최대 2.02%의 차이를 보였다. 그림 4 및 5는 각 명령어 별로 소스 및 목적 레지스터 사용에 따른 소비 전류를 나타낸 그림이다.

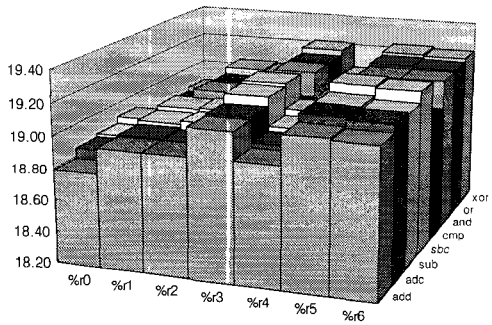

그림 4. 소스 레지스터 별 평균 소비 전류 (mA)
Fig. 4. Currents for Source Registers (mA)

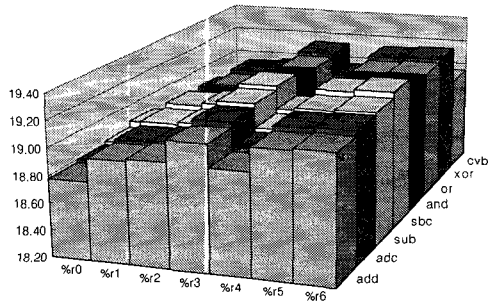

그림 5. 목적 레지스터 별 평균 소비 전류 (mA)
Fig. 5. Currents for Destination Registers (mA)

3.3. 직접(Immediate) 데이터의 값과 소비 전류

산술 및 논리 명령어 9개 중 8개는 오퍼랜드로 직접 데이터를 사용한다. 소비 전류 측정 결과 명령어의 오퍼랜드로 사용되는 직접 데이터의 값에 따라 소비 전류에 차이가 있음을 발견하였다. 이 현상도 이미 이전 연구에서도 언급된 바 있다[8]. 현재 직접 데이터로 0x03 및 0x07을 사용하는 경우 최고의 전류 소비를 보였다. 직접 데이터에 따른 소비 전류는 평균 1.10%, 최대 2.18% 차이를 보였다. 그림 6은 직접 데이터의 값에 따라 소비되는 전류의 값을 그래프로 나타낸 그림이다.

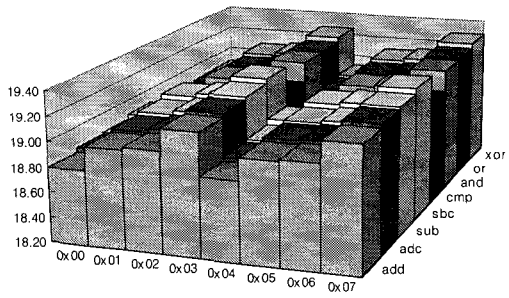

그림 6. 직접 데이터 별 소비 전류 (mA)
Fig. 6. Currents for Immediate Data (mA)

Ⅳ. 소비 전류 예측 모델

본 장에서는 3장에서 분석된 소비 전류의 특성을 바탕으로 주어진 명령어 수행에 대한 소비 전류를 예측하는 모델을 제시한다. 본 모델의 목표는 한 작은 회수의 측정 소비 전류를 사용하여 낮은 오차 범위에서 전체 명령어 수행에 대한 소비 전류를 예측하는 것이다. 우리는 3장에서 분석된 특성을 바탕으로 다음과 같은 사실을 알 수 있다.

- 명령어 종류 별 평균 소비 전류의 최대 차이는 1.57%로 비교적 낮다.
- 같은 명령어일 경우에 오퍼랜드에 따른 소비 전류의 차이는 평균 5.86%로 비교적 높다.
- 같은 명령어일 경우 소스 레지스터, 목적 레지스터 및 직접 데이터에 따른 소비 전류의 차이는 평균 1.14%, 1.25%, 및 1.10%이다.
- 명령어에 사용된 레지스터에 저장된 값은 소비 전류에 영향을 미치지 않는다.
- 같은 명령어일 경우 최소 소비 전류는 소스 레지스터 및 목적 레지스터가 각각 %r0일 때이고 직접 데이터의 값이 0x00일 때이다.

위 사실을 살펴보면 주어진 명령어에 대한 소비 전류는 제로(zero) 오퍼랜드(레지스터의 경우 %r0, 직접 데이터의 경우 0x00)를 사용할 때 가장 낮고 다른 오퍼랜드를 사용할 경우 추가 전류의 소비가 있음을 알 수 있다. 이를 바탕으로 본 논문은 명령어 I(i,j,k)에 대한 소비 전류 예측 모델 CM을 다음과 같은 식으로 정의한다.

$$C_M(I(i,j,k)) = C(I(0,0,0)) + D(I(i,0,0)) + D(I(0,j,0)) + D(I(0,0,k))$$

여기서 $D(I(i,j,k)) = C(I(i,j,k)) - C(I(0,0,0))$

위 식에서 I(i,j,k)는 첫째 오퍼랜드 i, 둘째 오퍼랜드 j 및 셋째 오퍼랜드 k를 가지는 명령어 I를 의미하고 C는 측정한 전류 값을 의미한다. 예를 들어 명령어 "add %SrcRn, ImmData, %DesRn"에 대하여 소비 전류 예측 모델을 적용하려면 아래 표 2와 같이 총 20개의 측정 전류가 필요하다.

표 2. 모델을 위한 명령어 'add'의 측정 전류 (mA)
Table 2. Measured Currents for instruction 'add' for the Model (mA)

C(add(0,0,0))	18.38

오퍼랜드 i	1	2	3	4	5	6
C(add(i,0,0))	18.52	18.69	18.74	18.52	18.71	18.70

오퍼랜드 j	1	2	3	4	5	6	7
C(add(0,j,0))	18.59	18.64	18.87	18.48	18.70	18.70	18.89

오퍼랜드 k	1	2	3	4	5	6
C(add(0,0,k))	18.54	18.54	18.74	18.54	18.71	18.71

앞에서 정의한 모델을 활용하기 위해서는 아래 표 3와 같은 측정 공간에 대한 측정이 필요하다. 이 공간의 크기는 2장에서 기술한 전체 측정 공간의 약 5.84%이다.

마이크로프로세서 adc16s310에서 산술 및 논리 명령어에 대하여 직접 측정한 소비 전류 C와 본 논문에서 제안한 모델을 통하여 계산한 전류 CM이 있을 때 본 모델의 오차(%)를 (|C - CM|) / C * 100로 정의하고 명령어 별 평균 오차를 그래프로 나타내면 그림 7과 같다. (주: 명령 'cvb'는 오퍼랜드가 1개이기 때문에 모델의 오차가 0%이므로 오차 계산에서 제외하였음.) 모든 산술 및 논리 명령어를 고려하면 본 논문에서 정의한 모델의 평균 오차는 0.34%이다.

표 3. 모델을 적용하기 위한 측정 공간
Table 3. Current Measurement Spaces for the Model

어셈블러 문법	측정 회수
add %SrcRn, ImmData, %DesRn	20
add %Src1Rn, %Src2Rn, %DesRn	19
sub %SrcRn, ImmData, %DesRn	20
sub %Src1Rn, %Src2Rn, %DesRn	19
adc %SrcRn, ImmData, %DesRn	20
adc %Src1Rn, %Src2Rn, %DesRn	19
sbc %SrcRn, ImmData, %DesRn	20
sbc %Src1Rn, %Src2Rn, %DesRn	19
cmp %SrcRn, ImmData	14
cmp %Src1Rn, %Src2Rn	13
and %SrcRn, ImmData, %DesRn	20
and %Src1Rn, %Src2Rn, %DesRn	19
or %SrcRn, ImmData, %DesRn	20
or %Src1Rn, %Src2Rn, %DesRn	19
xor %SrcRn, ImmData, %DesRn	20
xor %Src1Rn, %Src2Rn, %DesRn	19
cvb %DesRn	7
합계	307

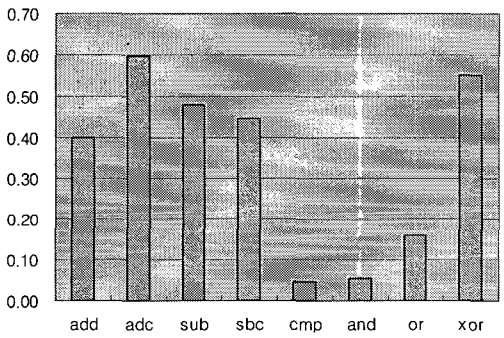

그림 7. 모델에 대한 명령어 별 평균 오차 (%)
Fig. 7. Average Errors using the Model (%)

V. 결 론

본 논문은 임베디드 마이크로프로세서 adc16s310의 산술 및 논리 명령어에 대하여 소비 전류를 측정하고 그 결과를 분석하였다. 분석된 결과를 바탕으로 비교적 적은 회수의 측정 값을 사용하여 모든 명령어 수행에 대한 소비 전력을 예측하는 전력 예측 모델을 제시하였다. 제시한 모델은 5.84%의 측정 공간에 대한 전류 측정을 사용하여 오차 0.34%로 비교적 정확하게 소비 전류를 예측한다. 본 연구 결과는 기존 연구 [2], [3] 및 [8]의 결과에 비하여 오차가 작게 나타났으며 모델의 적용 방법도 매우 간단하

다. 본 연구팀은 현재 산술 및 논리 명령어를 포함한 모든 명령어에 대하여 적합한 소비 에너지 예측 모델을 구상 중이며, 명령어 간 소비 전류 예측 모델도 연구 중이다. 본 연구팀은 본 논문에서 제안한 모델을 바탕으로 소프트웨어가 수행하면서 소비하는 전력을 예측하는 파워 시뮬레이터를 개발 중이다.

참고문헌

[1] S. Devadas and S. Malik, A Survey of Optimization Techniques Targeting Low Power VLSI Circuits, Proceedings of 32nd ACM/IEEE Design Automation Conference, San Francisco, CA, 242-247, 1995.

[2] V. Tiwari, S. Malik, and A. Wolfe, Power Analysis of Embedded Software: A First Step Toward Software Power Minimization, IEEE Transactions on VLSI Systems, Vol. 2, No. 4, 437-445, Dec. 1994.

[3] V. Tiwari, S, Malik, A. Wolfe, and M. T. Lee, Instruction Level Power Analysis and Optimization of Software, Journal of VLSI Signal Processing Systems, Vol. 13, Issue 2-3, Kluwer Academic Publishers, Boston, 223-238, Aug./Sep. 1996.

[4] V. Tiwari, S. Malik, and A. Wolfe, Compilation Techniques for Low Energy: An Overview, Proceedings of the IEEE Symposium on Low Power Electronics, San Diego, CA, October, 1994.

[5] H. Mehta, R. M. Owens, M. J. Irwin, R. Chen, and D. Ghosh, Techniques for Low Energy Software, Proceedings of the 1997 International Symposium on Low Power Electronics and Design, 1997.

[6] Advanced Digital Chips Inc., SE16108 Core Manual, Extendable Instruction Set Computer, Version 1.0, Nov. 2000.

[7] Advanced Digital Chips Inc., adc16S310 USB 1.1 Core Embedded Microprocessor Databook, Jun. 2003.

[8] S. Lee, A. Ermedahl, S. L. Lee, and N. Chang, An Accurate Instruction-Level Energy Consumption Model for Embedded RISC Processors, Proceedings of the ACM SIGPLAN workshop on Languages, compilers and tools for embedded systems LCTES '01, 2001.

[9] N. Chang, K. Kim, and H. G. Lee, Cycle-Accurate Measurement and Characterization with a Case Study of the ARM7TDMI, IEEE Transaction on Very Large Scale Integration (VLSI) Systems, Vol. 10, No. 2, April 2002.

저자소개

신 동 하(Dongha Shin)

1980년 경북대학교 전자공학졸업 (학사)
1982년 서울대학교 대학원 전자계산기공학과 졸업 (석사)
1994년 Univ. of South Carolina 컴퓨터과학과 졸업 (박사)
1982년 ~ 1996년 한국전자통신연구원 책임연구원
1991년 ~ 1994년 Univ. of South Carolina, TA
1997년 ~ 현재 상명대학교 소프트웨어학부 부교수
※관심분야: 저 전력 임베디드 시스템, 실시간 임베디드 운영체제, 컴파일러, 프로그래밍 언어 등

강 경 희(Kyunghee Kang)

2005년 상명대학교 소프트웨어학부 졸업 (학사)
2005년 ~ 현재 상명대학교 대학원 컴퓨터과학과 재학 (석사과정)
※관심분야: 저 전력 임베디드 시스템, 실시간 임베디드 운영체제, 임베디드 응용 소프트웨어 등