

논문 2008-08-08

임베디드 시스템의 가상 ARM 머신의 개발

(Virtual ARM Machine for Embedded System Development)

이 소 진, 안 영 호, 한 현 희, 황 영 시, 정 기 석*

(So-Jin Lee, Young-Ho An, Alex H Han, Young-Si Hwang, Ki-Seok Chung)

Abstract : To reduce time-to-market, more and more embedded system developers and system-on-chip designers rely on microprocessor-based design methodology. ARM processor has been a major player in this industry over the last 10 years. However, there are many restrictions on developing embedded software using ARM processor in the early design stage. For those who are not familiar with embedded software development environment or who cannot afford to have an expensive embedded hardware equipment, testing their software on a real ARM hardware platform is a challenging job. To overcome such a problem, we have designed VMA (Virtual ARM Machine), which offers easier testing and debugging environment to ARM based embedded system developers. Major benefits that can be achieved by utilizing a virtual ARM platform are (1) reducing development cost, (2) lowering the entrance barrier for embedded system novices and (3) making it easier to test and debug embedded software designs. Unlike many other purely software-oriented ARM simulators which are independent of real hardware platforms, VMA is specifically targeted on SYS-Lab 5000 ARM hardware platform, (designed by Liberton, Inc.) which means that VMA imitates behaviors of embedded software as if the software is running on the target embedded hardware as closely as possible. This paper will describe how VMA is designed and how VMA can be used to reduce design time and cost.

Keywords : ARM, virtual platform, embedded system, SoC

1. 서 론

임베디드 시스템이란 마이크로프로세서나 마이크로 컨트롤러를 내장하고 있는 하드웨어와 소프트웨어가 조합된 기술로 미리 정해진 특정 기능을 수행하기 위해 제작된 장치를 말하며, 필요에 따라서 일부 기계에 포함될 수 있다. 전기, 전자, 컴퓨터 기술이 발달하면서 이들 기술을 이용한 다양한 기기들이 우리의 생활 주변에 들어오게 되었다. 임베디드 시스템은 일상생활에서 사용되고 있는 전자

가전제품뿐만 아니라 핸드폰, PDA, 자동차, 항공편 제 시스템, 우주선 제어장치, 군사용 제어장치 등에 매우 폭 넓은 분야에 사용되어 우리 생활에 도움을 주고 있다. 전 세계적으로 생산되는 마이크로프로세서의 80% 이상이 임베디드 시스템에서 사용될 정도로 임베디드 시스템의 응용분야는 광범위하고, 이에 대한 연구도 매우 활발히 진행되고 있다[6].

임베디드 시스템에서 중요한 하드웨어 파트가 마이크로프로세서이다. 수많은 연구를 통해 만들어진 마이크로프로세서는 다양한 용도로 사용되었고 PC의 대중화, 인터넷 정보화 등에 매우 중요한 역할을 하고 있다. 현재 수집 중의 마이크로프로세서들이 개발되어 사용되고 있다. 각 프로세서는 고유한 특성을 가지며 그 특성에 따라 사용되는 용도도 다르다. 그 중 ARM 계열은 32비트의 프로세서로서, 고성능과 저전력 소비, 저가격 등의 장점을 가진 RISC 프로세서이다. RISC 프로세서이기 때문에 디코딩 과정이 상대적으로 간단하고, 이로 인하여 높은 명령어 처리량을 가지며, 실시간 인터럽트 응

* 교신저자(Corresponding Author)

논문접수 : 2008. 3. 5., 채택확정 : 2008. 4. 10.

이소진 : 한양대학교 정보통신대학

안영호, 한현희, 정기석 : 한양대학교 정보통신대학

황영시, 이현명 : 한양대학교 정보통신대학

※ 본 논문은 2008년도 서울시 산학연 협력사업의 지원과 2007년도 정부(과학기술부)의 재원으로 한국과학재단의 지원을 받아 수행된 연구임(Na. R01-2007-000-20891-0).

답에 있어서도 좋은 성능을 지니고 있다. 또한 메모리와의 인터페이스도 높은 비용을 요구하지 않도록 설계 되어 있다[7]. 이러한 이유로 핸드폰이나 PDA 등 많은 임베디드 시스템이 ARM 프로세서를 사용하고 있다.

본 논문에서는 이러한 ARM 프로세서를 내장하고 있는 임베디드 시스템을 개발하는데 이점을 제공하는 가상 ARM 머신을 설계 및 구현하였다. 가상 ARM 머신은 임베디드 시스템의 소프트웨어 개발을 하드웨어와 병렬적으로 진행 가능하게 하여 효율성을 증대시킬 뿐 아니라, 임베디드 시스템의 소프트웨어 개발에 필요한 일련의 과정을 축소시킴으로써 초보자도 쉽고 빠르게 개발을 할 수 있도록 한다[11].

ARM 프로세서를 내장하고 있는 리버트론의 SYS-Lab5000 보드를 가상 ARM 머신의 타겟 임베디드 시스템으로 설정하여, SYS-Lab5000 User's Manual[1]을 따라 가상 ARM 머신을 설계 및 구현하였다. ARM Simulation을 위해 SimIT-ARM[2]을 이용하였으며, 실행 결과를 보여주기 위한 GUI는 QT library[5]를 이용하였다.

본 논문의 구성은 다음과 같다. 2장에서는 기존의 가상 머신에 관해 살펴보고, 3장에서는 임베디드 시스템을 개발하는 과정과 가상 ARM 머신을 이용한 개발 과정의 차이를 설명하였다. 4장에서는 가상 ARM 머신 설계 및 구현 방법을 설명 하였고, 5장에서는 결론과 향후 계획을 기술 하였다.

II. 기존의 가상 머신

가상 프로토타입이라는 것은 가상 머신을 포괄하는 의미로, 컴퓨터에서 가상으로 만들어진 프로토타입으로써 현실에 있는 제품이나 혹은 만들어질 제품과 동일한 물리적, 기능적 특성을 가지면서 시뮬레이션을 가능하게 하는 것을 말한다[8]. 본 논문에서는 가상 프로토타입으로 ARM 프로세서를 내장하고 있는 임베디드 시스템을 구현하였고, 이를 가상 ARM 머신이라고 명명하였다.

ARM 명령어 집합을 시뮬레이션 해주는 기존의 시뮬레이터들은 ARM 명령어만을 시뮬레이션해주기 때문에 주변 장치에 관한 제어를 할 수 없다[2-4]. 이러한 시뮬레이터만을 사용하는 것이 아닌, 시뮬레이터를 기반으로 가상 ARM 머신을 구현 해주어야 주변 장치의 제어가 가능하다. 시뮬레이터를 이용하여 가상 ARM 머신을 구현한 기존의 연구[11]는 제어결과를 Text문이나 Graph로 출력

하여 사용자가 실행결과를 직관적으로 확인하기에 용이하지 않다. 또한, 동시에 주변 장치의 제어가 가능한 임베디드 시스템의 특성과 달리 기존의 연구는 주변 장치의 제어가 하나만 가능하였다.

본 논문에서 제안한 가상 ARM 머신은 주변 장치를 임베디드 시스템의 특성과 똑같이 동시에 제어할 수 있고, 이 결과를 임베디드 시스템에서 동작하는 것과 똑같이 보여줌으로써 직관적으로 결과를 확인할 수 있게 하였다. 가상 ARM 머신이 제어할 수 있는 주변 장치는 타겟 보드의 주변 장치와 같으며 LED, LCD, 7-Segment, Dot-matrix, Step Motor, keypad 등이 이에 속한다.

III. 가상 ARM 머신

1. 임베디드 시스템의 개발

PC와 같은 범용 시스템을 위한 소프트웨어 개발과 임베디드 시스템을 위한 소프트웨어 개발은 많은 차이가 있다. 그렇기 때문에 PC 시스템의 개발 작업만을 수행해 온 대부분의 소프트웨어 개발자들은 임베디드 시스템의 개발에 어려움을 겪기도 한다. 기존의 소프트웨어 개발과정은 소스코딩을 하고 컴파일하여 실행하는 형태가 모두 PC 시스템이라는 한 곳에서 이루어졌었다. 그러나 임베디드 시스템의 개발이란 PC 시스템에서 소스코딩을 하고 컴파일을 하여 빌드된 실행프로그램을 임베디드 시스템에 전송하여 실행하게 된다.

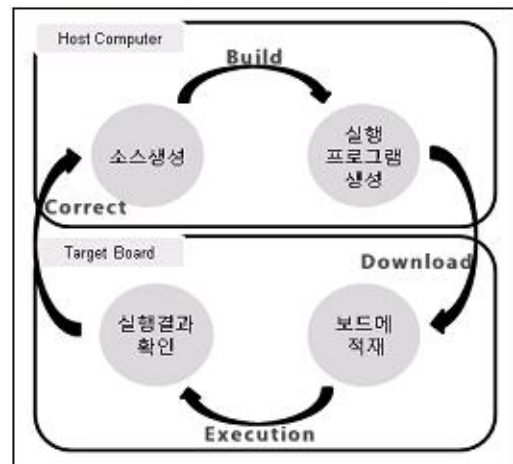


그림 1. 임베디드 시스템의 개발

Fig 1. Development of Embedded System

기존의 소프트웨어 개발자가 가장 먼저 어려움

을 겪고 있는 것은 바로 이러한 PC 시스템과 같은 개발 host와 실제 실행이 이루어지는 임베디드 타겟 시스템이 다르다는 것이다.[10] 그림 1은 개발 환경과 실행 환경의 차이와 빌드 된 실행프로그램을 다운로드하는 과정을 보여준다.

또한, PC 시스템의 소프트웨어 개발은 별도의 임베디드 시스템 장비 없이 PC상에서 모든 것이 이루어지기 때문에 임베디드 시스템 장비가 없이도 개발이 진행된다. 그러나 임베디드 시스템의 소프트웨어 개발은 최종 검증용 위해서는 실제 임베디드 시스템 하드웨어에 실행시키는 것이 필요하다.

임베디드 시스템의 개발을 처음 접하고, 고가의 보드를 개인적으로 구입할 수 없는 개발자들은 개인적으로 보드에서 테스트를 하는데 어려움을 겪게 된다. 개발자들이 임베디드 시스템의 개발을 하기 위해서는 임베디드 시스템에 관한 전반적인 지식과 내용을 배워야하고 하드웨어를 다루거나 다운로드 하는 법에 관해 익숙해져야만 한다. 그리고 임베디드 시스템에서 테스트를 해보기 위해선 그 타겟 보드를 구입하거나 구입되어진 장소에서만 테스트를 해볼 수 있다. 이러한 개발자들에게 개인적으로 집에서 쉽게 개발하고 테스트를 해볼 수 있도록 만들어진 것이 이 논문에서 제시한 VMA(Virtual ARM Machine 가상 ARM 머신)이다.

2. VMA(가상 ARM 머신)의 개발

PC 시스템에서 소스코딩을 하고 컴파일하여 빌드 된 실행프로그램을 가상 ARM 머신에서 실행시킴으로써 보드에서 돌아가는 결과를 가상 ARM 머신을 통해 확인 할 수 있다.

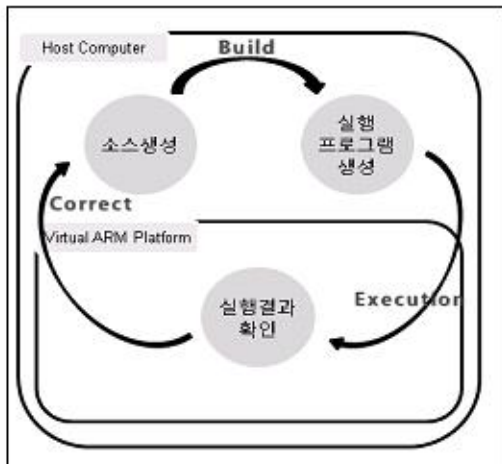


그림 2. 가상 ARM 머신 개발
Fig 2. Development of Virtual ARM Machine

즉, 임베디드 시스템의 하드웨어 없이 PC 시스템만으로 임베디드 시스템의 소프트웨어를 개발할 수 있게 된 것이다. 또한, 별도의 다운로드과정도 필요치 않기 때문에 임베디드 시스템의 개발의 어려운 점을 가상 ARM 머신의 이용을 통해 상당 부분 해결할 수 있다. 그림 2는 별도의 다운로드 없이 PC 시스템만으로 돌아가는 가상 ARM 머신의 실행과정을 보여준다.

IV. 구현 및 개발

1. VMA의 구성

VMA는 크게 가상 ARM 시뮬레이터, Graphic User Interface, 입력 이벤트를 처리하는 부분으로 구성되어 있다. 그림 3은 전체적인 가상 ARM 머신의 수행과정을 보여주고 있다.

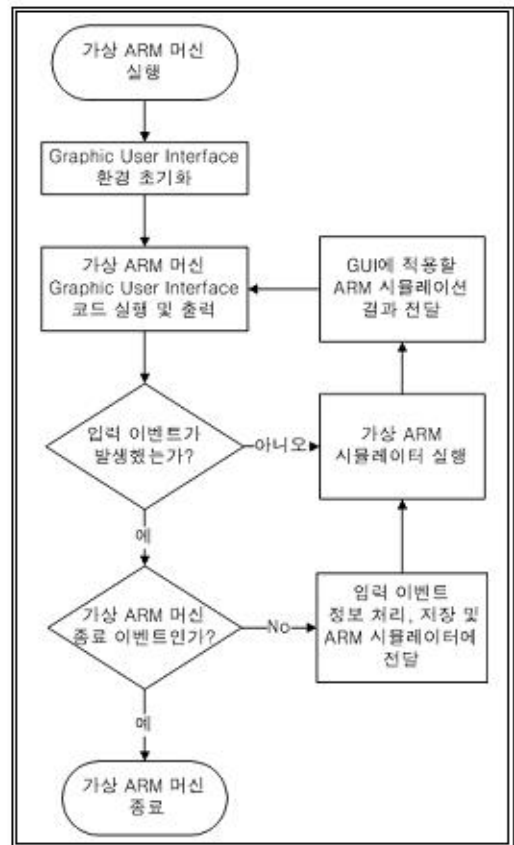


그림 3. 가상 ARM 머신의 수행과정
Fig 3. Execution procedure of Virtual ARM Machine

가상 ARM 머신을 실행하면 Graphic User Interface 환경을 초기화하고 ARM 시뮬레이션 결과가 넘어오면 GUI 코드를 실행 및 출력하게 된다. Graphic User Interface는 ARM 시뮬레이터의 결과를 장비에서와 아주 유사하게 시각적으로 구현하며, 때로는 keypad나 switch 등의 외부적인 입력을 처리하여 입력 이벤트를 발생시키게 한다. 외부 입력 이벤트가 들어오면, 입력 이벤트에 따라 정보를 처리하고 저장하여 ARM 시뮬레이터에 전달해 주게 된다. ARM 시뮬레이터에서는 주변장치 제어 정보를 추출하여 ARM 시뮬레이션을 하게 되고, 그 결과를 다시 Graphic User Interface에 전해 주어 사용자가 확인할 수 있도록 한다.

2. ARM 시뮬레이터의 구성

이 절에서는 그림 3의 수행과정에서 ARM 시뮬레이터의 실행이 어떻게 진행되는지 설명한다. ARM 시뮬레이터의 실행은 가상 ARM 머신에서 전달해주는 이벤트를 ARM 시뮬레이터에서 실제 ARM과 동일하게 처리하게 된다. ARM 시뮬레이터가 시작하면 ARM 명령어들을 디코딩하여 명령어들을 실행한다. 명령어 디코딩 결과, 명령어가 주변장치 제어주소를 포함할 경우 기억장치 대응 입출력(Memory-Mapped I/O) 방식을 이용하여 각 주변장치들의 제어 데이터와 비교하여 해당 주변장치를 찾아 제어 주소와 데이터를 저장하게 되며, 가상 ARM 머신의 GUI 제어 코드에 주변장치 제어 정보를 전달해준다. 이러한 ARM 시뮬레이터의 주변장치 제어 정보 추출 흐름도는 그림 4에 순서대로 확인해 볼 수 있다.

기억장치 대응 입출력 방식은 ARM 아키텍처에서 채택하는 I/O 방식으로, ARM 입출력 명령이 없이 기억장치의 특정 주소를 액세스 하는 것으로 입출력이 되도록 하는 방식이다.

본 논문에서 가상 ARM 머신의 타겟 임베디드 시스템으로 잡은 ARM 프로세서를 내장하고 있는 리버트론의 SYS-Lab5000의 경우, ARM 시뮬레이터에서의 주변장치 제어 데이터 주소를 그림 5와 같이 설정함으로 Memory Mapped I/O를 구성하였다.

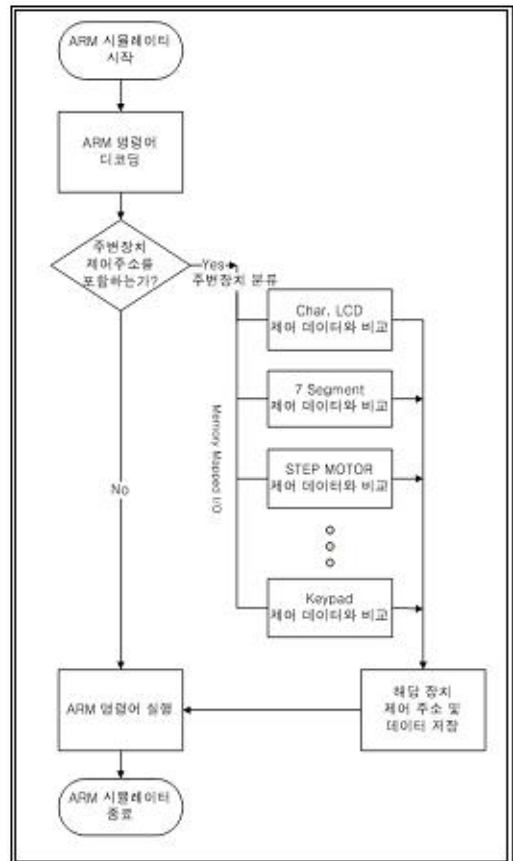


그림 4. ARM시뮬레이터의 주변 장치 제어 정보 추출 흐름도

Fig 4 Device control information extraction in ARM Simulator

주소>	레지스터 명칭 <설명>	R/W>	Size(Bit)>
0x0000_0004>	LCCR (LED Control Register)>	W>	24>
0x0000_0008>	SGCR (Segment Control Register)>	W>	14>
0x0000_000c>	DMCCR (Dot-Matrix LED Control Register)>	W>	4>
0x0000_0014>	TLICR (Test-LCD Instruction Control Register)>	W>	8>
0x0000_0018>	TLDCR (Test-LCD Data Control Register)>	W>	8>
0x0000_001c>	KMCCR (Key-Matrix Control Register)>	R>	4>
0x0000_0028>	SMCCR (Step Motor Control Register)>	r	r


```

#define LED          *(volatile unsigned int*)(0x00000004) //LCC>
#define SBD         *(volatile unsigned int*)(0x00000008) //SGCR>
#define DOT         *(volatile unsigned int*)(0x0000000c) //DMCCR>
#define LCD_INST    *(volatile unsigned int*)(0x00000014) //TLICR>
#define LCD_DATA    *(volatile unsigned int*)(0x00000018) //TLDCR>
#define KEYPAD      *(volatile unsigned int*)(0x0000001c) //KMCCR>
#define MTR         *(volatile unsigned int*)(0x00000028) //SMCCR>
    
```

그림 5. 기억장치 대응 입출력의 예
Fig 5. Example of Memory Mapped I/O

3. 가상 ARM 머신 실행

그림 6은 가상 ARM 머신에서 7-Segment와 step motor를 동시에 컨트롤하고 있는 모습이고, 그림 7은 타겟 보드인 리버트론사의 SYS-Lab5000 실제 모습이다. SYS-Lab5000을 모델링한 GUI환경에 실제로 보드가 실행되는 것과 같이 가상 ARM 머신이 실행되는 것을 확인할 수 있다.

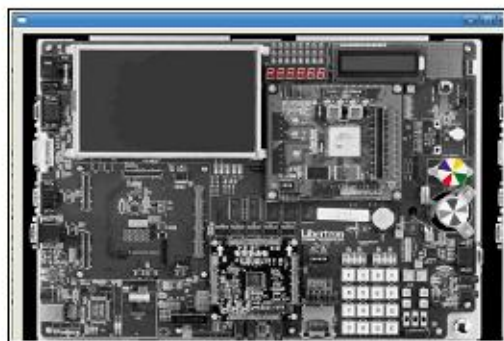


그림 6. 가상 ARM 머신의 실행
Fig 6. Execution of Virtual ARM Machine



그림 7. 실제 SYS-Lab 5000 하드웨어 보드
Fig 7. SYS-Lab 5000 Real H/W Board

V. 결 론

본 논문에서는 실제 임베디드 시스템 하드웨어가 없는 환경에서 개발 가능한 가상 ARM 머신을 구현해 보았다. 가상 ARM 머신은 다음과 같은 장점을 가지고 있다.

첫째, 실행 프로그램을 전송하는 과정이 없으므로 빠르고 편리하게 임베디드 시스템의 개발을 하고 그 결과를 확인할 수 있다.

둘째, ARM을 처음 접하고 개발하는 사람들이 쉽게 사용할 수 있다.

셋째, 타겟 보드를 구매하지 않아도 개발할 수 있으므로 적은 비용으로 임베디드 시스템의 개발을 할 수 있다.

위와 같은 장점이 있지만 현재 구현된 가상 ARM 머신은 제한된 임베디드 시스템의 디바이스만을 테스트 가능하고 타겟으로 잡은 제한된 임베디드 시스템 하드웨어만 검증이 가능하다. 향후 개발을 통하여 이러한 문제점들을 개선하려고 한다.

참고문헌

- [1] 리버트론, "Sys-Lab500 User Manual", http://www.libetrn.com/include/pdf/Sys-Lab500_0.pdf
- [2] "Simit-ARM USERS' GUIDE", <http://simit-arm.sourceforge.net/>
- [3] SimpleScalar LLC, "SimpleScalar tutorial", <http://www.simplescalar.com>
- [4] ARM Cop, "ARMLimulator, ADS1,2 / Developer Suite / Debug Target Guide",
- [5] QT <http://trolltech.com/products/qt/>
- [6] 이정배, 이두원, "임베디드 시스템 연구 동향" 정보처리학회지, 제9권, 제1호, 2002년 1월호, 쪽13-27
- [7] 심성준, 이재범, 강성태, 전주식, "ARM 7 프로세서를 위한 성능평가도구의 개발", 한국정보과학회, Vol.25, No.2
- [8] 원종혁, 한상용, "사용자 인터페이스 향상을 위한 3차원 VP시뮬레이터 설계", HCI2000 학술대회 발표 논문집, pp417-422, 2000.
- [9] W. Qin, S. Rajagopalan, S. Malik, "A Formal Concurrency Model Based Architecture Description Language for Synthesis of Software Development Tools", ACM 2004 Conference on Languages, Compilers, and Tools for Embedded Systems, June 2004, pp. 47-56.
- [10] 은성배, 한상숙, 진성기, "임베디드 시스템 프로그래밍 교육론 및 교육용 장비 개발 사례", 세계한민족과학기술자총합학술대회, 2002.
- [11] 최혁상, 조상영, 이정배, "ARMLimulator를 이용한 시스템 프로그래밍 관점의 가상 프로토타입 설계 및 구현", 한국컴퓨터총합학술대회 논문집, 2005

저 자 소 개

이 초 진



2008년 한양대학교 미디어통신공학과 학사 졸업.
현재, 한양대학교 전자컴퓨터통신공학과 석사과정
관심분야: 임베디드 하드웨어, SoC

Email: lsojinl@paran.com

안 영 호



2008년 한양대학교 컴퓨터학과 학사 졸업. 현재, 한양대학교 전자컴퓨터통신공학과 석사과정
관심분야: 임베디드 시스템, 데이터베이스

Email: ghduckks64@hotmail.com

한 현 희



2009년 한양대 컴퓨터학과 학사 졸업예정.

관심분야: 인공지능, 임베디드 시스템

Email:

alexhan@hanyang.ac.kr

황 영 시



2003년 대전대학교 컴퓨터공학과 학사 졸업.

2005년 (주)한국정보통신 기술연구소 연구원.

2008년 한양대학교 전자컴퓨터통신공학과 석사 졸업

현재, 한양대학교 전자컴퓨터통신공학과 박사과정.

관심분야: 임베디드 시스템, 저전력 시스템 설계, RTOS

Email : ysturtle@gmail.com

정 기 석



1989년 서울대학교 컴퓨터공학과 학사. 1997년 University of Illinois at Urbana-Champaign 전산학 박사. 1998년 University of Illinois at Urbana-Champaign, 강

의 전담 교수. 2000년 Synopsys, Inc. Sr. R&D Engineer. 2001년 Intel Corp. Staff Engineer. 2001년 홍익대학교 컴퓨터 공학과 조교수. 2004년 한양대학교 정보통신대학 조교수. 현재, 한양대 정보통신대학 미디어통신공학과 조교수.

관심분야: 임베디드 시스템, SoC, 저전력 시스템 설계, VLSI/CAD

Email: kchung@hanyang.ac.kr