논문 2006-43SD-10-9

효율적인 통합시뮬레이션에 의한 스피커 연결 시스템의 SoC 설계

(SoC Design of Speaker Connection System by Efficient Cosimulation)

송 문 빈*, 송 태 훈*, 오 재 곤**, 정 연 모*

(Moonvin Song, The-Hoon Song, Chaegon Oh, and Yunmo Chung)

요 약

본 논문에서는 SoC(System On a Chip)의 효율적인 설계와 빠른 검증을 위해서 Active-HDL과 Matlab의 Simulink를 연동하여 HDL, SystemC 및 알고리즘 레벨의 추상화를 동시에 통합하여 시뮬레이션 할 수 있는 방법론을 제시하고, 이를 이용한 다채널 스피커의 직렬연결 기법을 설계 및 구현하였다. 구현은 ARM 프로세서와 Xilinx Virtex4 FPGA를 기반으로 하고 AMBA 버스를 사용하여 연동하는 SoC Master 보드 상에서 이루어졌다. 이러한 방법은 하드웨어 부분의 RTL 코드를 IP화하여 소프트웨어 부분과 동시에 검증 할 수 있는 장점을 가지고 있으며 직렬 연결 스피커 시스템과 같이 많은 신호처리를 하는 부분에서 쉽고 빠르게 설계를 진행할 수 있음을 보였다.

Abstract

This paper proposes a cosimulation methodology that results in an efficient SoC design as well as fast verification by integrating HDL, SystemC, and algorithm-level abstraction using the design tools Active-HDL and Matlab's Simulink. To demonstrate the proposed design methodology, we implemented the design technique on a serial connection multi-channel speaker system. We have demonstrated the proposed cosimulation method utilizing an ARM processor based SoC Master board with the AMBA bus interface and a Xilinx Vertex4 FPGA. The proposed method has the advantage of simultaneous simulation verification of both software and hardware parts in high levels of abstraction mixed with some performance critical parts in more concrete RTL codes. This allows relatively fast and easy design of a speaker connection system which typically requires significant amount of data processing for verification.

Keywords: Cosimulation, SoC, Serial connection, Speaker, Acoustic signal processing

I. 서 론

현재 SoC를 효율적으로 설계하기위한 다양한 방법론이 제시되고 있다. 그러나 이러한 설계 방법론의 대부분은 SoC 설계를 지원하는 회사의 CAD 프로그램의 기능을 사용하는 것으로 국한되어 있는 것이 현실이다. 따라서 하나의 CAD 프로그램을 선택하면 설계 방법 또한 고정이 되는 문제가 있다[6].

SoC 설계는 다양한 방법으로 접근을 할 수 있어야 한다. 예를 들어, 시스템의 성능을 극대화시키기 위한 설계 방법과, 시스템의 자원을 최소화하기 위한 접근 방법, 그리고 시스템의 설계 시간을 최대한 단축하기 위한 설계 방법 등이다. 이러한 다양한 접근 방법에도 불구하고 하나의 SoC 시스템을 설계해서 최종적으로 구현을 마치는 순간까지 중요한 것은 바로 설계의 일관성을 유지 하는 것이다[1]. 바람직한 설계 방법은 알고리즘 검증 단계에서의 자원을 그대로 HW 설계 및 SW 설계에서 사용하고 최종적으로 검증에 까지 사용하는 것이다. 이러한 기본적인 요구 사항을 만족하는 설계 방법으로서 본 논문에서는 [그림 1]과 같이 RTL

* 정회원, 경희대학교 전자공학과
(Dept. of Electronic Eng., Kyung Hee University)
** 정회원, 한국산업기술대학교 전자공학과
(Dept. of Electronic Eng., Korea Polytechnic University)
* 본 논문은 2005년 경희대학교 지원에 의한 결과임
접수일자: 2006년5월23일, 수정완료일: 2006년9월15일

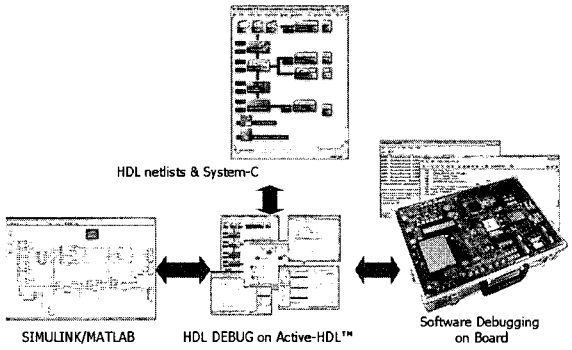


그림 1. 제안하는 설계 방법론
Fig. 1. Proposed design methodology.

(Register Transfer Level) 시뮬레이션을 지원하는 Aldec사의 Active-HDL과 알고리즘 단계의 검증을 지원하는 Matlab의 Simulink를 연동하여 통합으로 하드웨어와 소프트웨어를 시뮬레이션 하는 설계 방법을 제시하고 이를 이용해 직렬 연결 스피커 시스템을 설계하였다[2,3].

하드웨어 구현은 ARM 프로세서와 Xilinx Virtex4 FPGA를 기반으로 하고 AMBA 버스를 연동하여 사용하는 SoC Master 보드 상에서 이루어졌다. 소프트웨어 부분은 Velos RTOS를 사용하여 구현하였다.

Ⅱ. 제안하는 통합시뮬레이션 방법

Aldec의 Active-HDL은 VHDL, Verilog-HDL, EDIF, SystemC, System-Verilog, C/C++과 같은 다양한 설계 언어를 같이 통합 설계하고 시뮬레이션을 지원하는 프로그램이다. 또한 버전 7.0 이상의 Matlab 또는 Matlab의 Simulink와 연동하여 통합 시뮬레이션을 수행할 수 있다. 그리고 설계를 마친 결과를 대상 시스템에 직접 구성할 수 있다[2].

Matlab은 알고리즘을 검증하는데 유용한 도구로서 그 중에서도 Simulink는 IP(Intellectual Property) 형태의 수많은 컴포넌트를 가지고 블록 다이어그램 형태의 설계를 할 수 있으며 HDL이나 SystemC 보다 높을 추상화를 지원한다[3].

통합시뮬레이션 방법은 알고리즘 레벨을 검증하기 위해 설계한 Simulink의 특정 블록을 Active-HDL에서 RTL과 SystemC로 구현한 블록으로 교체 후 시뮬레이션하는 개념이다. 교체 후 Simulink를 실행 하면 교체된 블록의 연산은 Active-HDL이 담당하여 그 결과를 알려준다. 즉, 하드웨어 부분과 소프트웨어 부분을 동시에 검증할 수 있다. 각 소프트웨어의 구체적인 적용 범위에 따른 설계 순서는 [그림 2]와 같다.

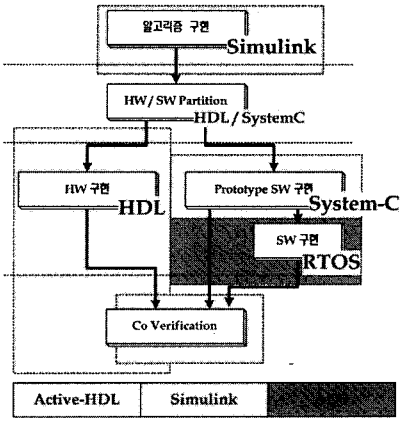


그림 2. 각종 SW의 구체적인 적용 범위에 따른 설계 순서
Fig. 2. Design flow showing where various programs fit in this cosimulation environment.

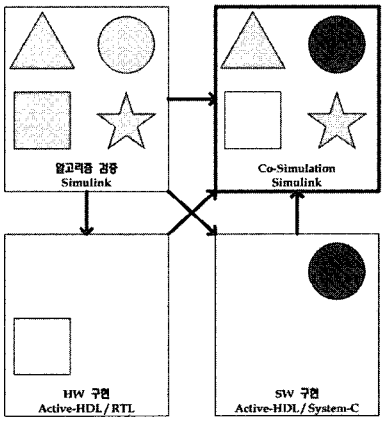


그림 3. 설계 적용 방법
Fig. 3. Design implementation method.

[그림 2]에서 알고리즘의 구현 및 검증은 Simulink상에서 수행한다. 그 후, 하드웨어 부분과 소프트웨어 부분으로 기능을 분할하여 하드웨어 부분은 HDL로 구현하고, 소프트웨어 부분은 SystemC로 구현한다. HDL과 SystemC로 구현된 내용은 Active-HDL과 Simulink를 사용하여 통합으로 시뮬레이션이 가능하다. 시뮬레이션을 수행한 뒤에는 SystemC로 구현된 SW 부분은 C 언어로 구현되어 있으므로 쉽게 RTOS의 응용 프로그램으로 변경이 가능하고 ARM의 컴파일러인 ADS를 사용한다.

위와 같은 방법을 설계에 적용하면 [그림 3]과 같이 알고리즘 레벨의 설계를 그대로 설계 검증 단계까지 쉽게 가져갈 수 있다.

[그림 3]에서 알고리즘을 검증하는 단계는 Simulink를 사용하여 IP 기반의 컴포넌트들을 가지고 쉽고 빠르게 구현 할 수 있다. 이렇게 구현한 알고리즘 블록들을

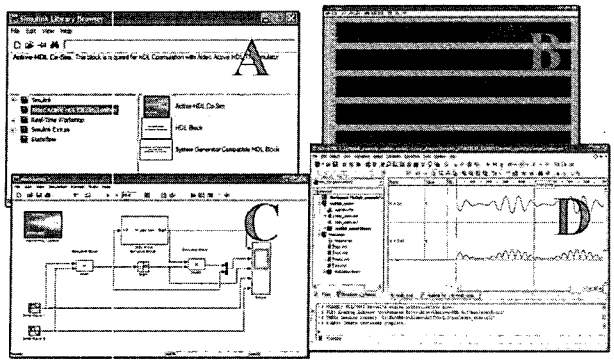


그림 4. Active-HDL과 Simulink 통합시뮬레이션 내용
Fig. 4. Example of Active-HDL and Simulink cosimulation.

하드웨어와 소프트웨어로 분할한 뒤에 하드웨어 부분은 Active-HDL에서 RTL로 설계를 진행한다. 그리고 소프트웨어 부분은 Active-HDL에서 SystemC를 사용하여 구현한다. 이어서 Active-HDL에서 설계한 블록으로 Simulink의 해당 블록을 교체하고 시뮬레이션하면 일관된 설계 환경을 가지고 통합으로 시뮬레이션을 수행하여 검증한다. Simulink의 컴포넌트를 Active-HDL에서 설계한 블록으로 교체하여 통합 시뮬레이션을 수행한 예는 [그림 4]와 같다[2].

그림에서 A는 Simulink의 컴포넌트로 Active-HDL에서 설계한 블록이 추가된 것을 확인할 수 있다. C는 기존의 알고리즘 중 하나의 컴포넌트를 Active-HDL에서 설계한 특정 블록으로 교체한 것을 나타내고 있다. Simulink를 수행하면 B와 같이 Simulink 상의 수행 결과와 D의 Active-HDL에서 실행한 블록의 시뮬레이션 결과를 확인 할 수 있다.

III. 설계 방법론 적용

다채널 스피커를 갖는 음향 기기에서 디지털 기술을 이용하여 스피커들을 직렬로 연결하는 기술이 최근에 제안되어 있다[8]. 본 논문에서는 제안하는 설계 방법론을 직렬 연결 스피커 시스템에 적용하여 [그림 5]와 같은 시리얼 전송 신호 생성 블록을 갖는 시스템을 구현하였다.

각 채널별 아날로그 신호들은 음향신호 전용 ADC (Analog-to-Digital Converter)를 사용하여 I^2S (Inter-IC Sound) 버스 신호로 변환하여 Channel Separation 블록으로 보낸다. 이 경우 ADC에서 생성된 신호는 각 채널별로 분리한다[4,5].

Channel Separation 블록에서는 연속적으로 들어오

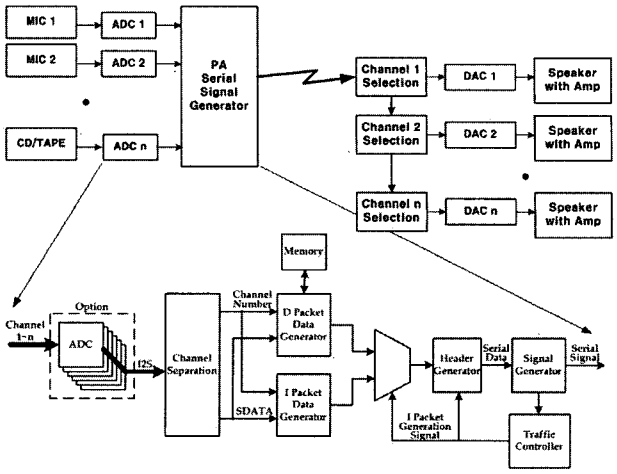


그림 5. 시리얼 전송 신호 생성 블록 다이어그램
Fig. 5. Block diagram of the serial signal generator.

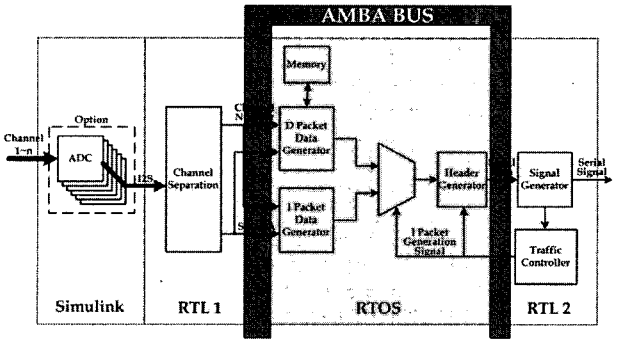


그림 6. 하드웨어 / 소프트웨어 기능 분할
Fig. 6. Hardware / Software functionality partition.

는 여러 채널들의 I^2S 버스 신호에서 채널별로 분리해 Packet Generator 블록으로 보낸다. 수신 측에서는 패킷의 헤더에서 Channel number를 검출하여 해당 채널일 경우에만 원래의 I^2S 버스 신호를 복원해 낸다. 이렇게 복원된 각 채널의 I^2S 버스 신호는 DAC (Digital-to-Analog Converter) 변환기와 앰프를 사용하여 스피커들을 구동 시켜 실제 음향을 만들어 낸다.

시리얼 전송 신호 생성 블록 다이어그램을 하드웨어와 소프트웨어 부분으로 분할하면 [그림 6]과 같다.

검증하는 단계에서 ADC에서 출력되는 I^2S 버스 신호는 Simulink에서 생성시키고, 시리얼로 입력되는 음향 신호의 ADC 결과를 RTL1에서 24 비트의 데이터로 묶어서 더(dummy) 비트를 포함한 32 비트를 ARM의 내부 버스 시스템인 AMBA 버스를 통해 소프트웨어 부분인 RTOS로 보낸다. RTOS는 패킷 생성 조건을 판단하여 압축을 수행한 후 AMBA 버스를 사용하여 하드웨어 부분인 RTL2로 보낸다. RTL2는 패킷을 시리얼로 변환하여 전송한다.

여기에서 하드웨어 부분은 Simulink, RTL1, 그리고

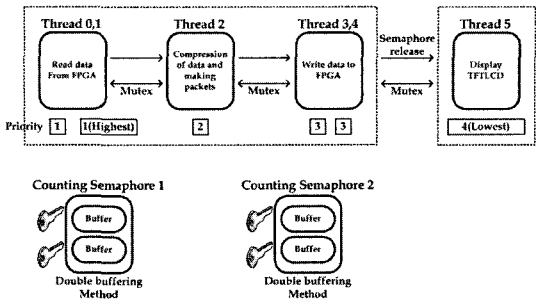


그림 7. SW 부분의 RTOS 구성
Fig. 7. RTOS formation of software part.

RTL2 이고 소프트웨어 부분은 RTOS 영역이다. 하드웨어와 소프트웨어를 분할하는 것은 전문적인 분석기를 사용하여야 하나 SoC 설계 시간을 최대로 단축하는데 목적으로 하드웨어 부분은 외부 인터페이스를 담당하는 부분과 시간에 민감한 기능을 수행하는 부분을 할당 하였다. 그리고 소프트웨어 부분은 많은 양의 연산을 수행하는 부분을 할당 하였다. 그리고 그 기능을 원활히 수행하기 위하여 Velos RTOS를 사용하였다. RTOS의 구성은 [그림 7]과 같이 총 5개의 thread를 가지고 있으며, queue와 같은 double buffering 기법을 사용하였다. 또한 4 개의 semaphore를 사용하여 mutex로 하였다.

AMBA 버스 매퍼는 HDL로 구현하여 Active-HDL을 사용하여 IP로 변환한 후 통합시뮬레이션을 수행하기 위하여 Simulink에서 하나의 컴포넌트로 사용한다[6,7,8]. 구현한 전체 시스템을 Active-HDL과 Simulink로 통합 시뮬레이션을 수행하여 전체 시스템을 검증하였다.

Ⅳ. 구현 결과 및 성능 평가

시스템을 구현하기 위하여 SoC Master 3에 사용자 보드를 제작하여 구성하였고 연결 방법은 [그림 8]과 같다.

우측 상단의 음향 기기에서 출력되는 아날로그 음향신호는 상단 부분의 사용자 전송 보드에서 아날로그 음향신호를 받아 ADC를 통해 변환된 신호를 AMBA 버스를 통해 SoC Master 보드로 전달한다. SoC Master 보드에서는 각종 DSP 처리 과정을 거쳐 압축된 시리얼 음향신호를 AMBA 버스를 통해 다시 사용자 전송 보드로 전송한다. 이 신호는 시리얼로 연결된 여러 개의 수신 보드로 전달되며, 수신 보드에서는 해당 신호를 검출 하여 압축을 푼 후 DAC 과정을 거쳐 최종적으로 스피커로 전달한다. 구현한 전체 시스템은

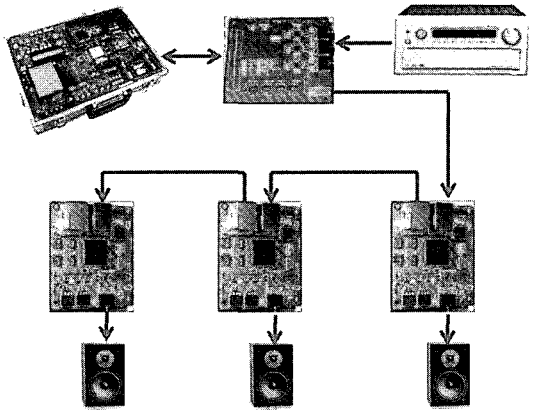


그림 8. 구현한 시스템 구성도
Fig. 8. Construction of the implemented system.

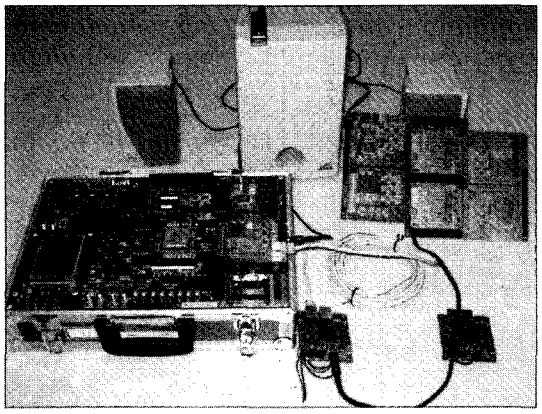


그림 9. 구현한 시스템 사진
Fig. 9. Photograph showing the implemented system.

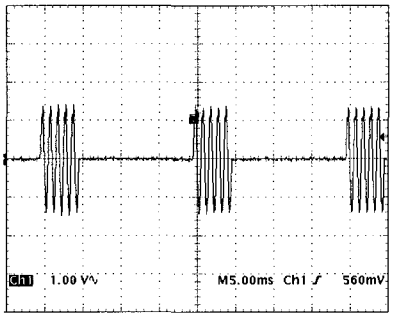


그림 10. 입력 테스트 신호
Fig. 10. Input test signal.

[그림 9]와 같다.

구현한 시스템의 검증을 위해서 [그림 10]과 같이 테스트 신호를 1kHz로 입력하여 출력 결과를 비교하면 [그림 11]과 같이 일정한 지연 시간을 갖고 출력하는 것을 확인할 수 있다.

직렬 연결 스피커 시스템은 실시간으로 입력되는 여러 채널의 음향 신호를 처리해야 한다. 따라서 검증을 위한 시뮬레이션을 수행하기 위해서는 매우 많은 양의 입력 벡터 데이터를 필요로 한다. 기존과 같이 수기로

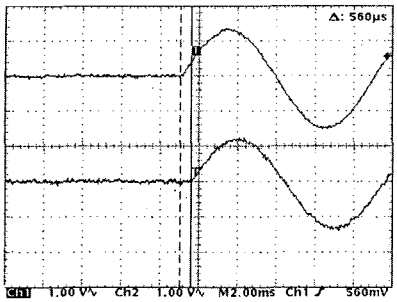


그림 11. 입력 및 출력 테스트 신호
Fig. 11. Input / output test signal.

표 1. 테스트 벡터 생성 시간 비교
Fig. 1. Comparison of test vector generation time.

	기존(수기)	제안한 방식
시 간	14 시간	15 분

테스트 벤치 파일을 생성할 경우 많은 시간과 노력을 필요로 한다. 그러나 제안하는 통합시뮬레이션 기법을 사용하면 Simulink로 구현한 알고리즘 수준의 모델에서 직접 테스트 벤치 파일을 생성 시킬 수 있다. [표 1]은 테스트 벡터 생성 시간을 비교한 것이다.

기존에는 테스트 벤치를 일일이 수기로 작성하였다. 직렬 연결 스피커 시스템과 같이 많은 양의 입력 데이터를 갖는 시스템과 같은 경우 테스트 벤치를 생성하기 위해 14시간의 작업을 필요로 했다. 그러나 제안하는 통합시뮬레이션 기법을 사용하면 단순히 Simulink와 Active-HDL을 연동하여 쉽고 빠르게 생성할 수 있다.

본 논문에서 제시한 기법의 또 다른 장점으로서는 SoC 시스템을 설계하여 최종적으로 구현을 마치는 순간까지 설계의 일관성을 유지할 수 있다는 것이다. 이 경우에 알고리즘 검증 단계에서의 자원을 그대로 HW 설계 및 SW 설계에서 사용하고 최종적으로 검증에 까지 사용할 수 있는 장점이 있다.

V. 결 론

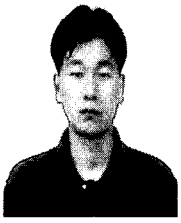
본 논문에서는 통합 시뮬레이션 방법을 적용한 SoC 설계기법을 사용하여 직렬 연결 스피커 시스템을 설계하였다. 기능을 하드웨어와 소프트웨어로 분할한 후 기존의 방법과 비교해서 효율적이고 빠르게 검증하고 보다 쉽게 구현하였다.

기존에 알고리즘을 검증하기 위해서 소프트웨어적으로 기능을 구현한 경우에 소프트웨어와 하드웨어 블록간의 입출력이 다르기 때문에 각 블록을 RTL로 직접 변경할 수 없다. 그러나 본 논문에서 제시한 방법은 각 블록 자체가 RTL의 입출력을 가지고 있다. 따라서 알고리즘 레벨의 기능 블록을 RTL로 대체할 수 있다.

참 고 문 헌

[1] Rajsuman, Rochit, *System On a Chip Design and Test,* Artech House, 2002.
[2] http://www.aldec.com
[3] http://www.mathwork.com
[4] Xie Bosun, "Signal Mixing for a 5.1-Channel Surround System - Analysis and Experiment," *AES Journal,* Vol 49, No. 4, pp. 263, 2001.
[5] C. Busbridgem Y. Huang, and P. A. Fryer, "Crossover Systems in Digital Loudspeakers," *AES Journal,* Vol. 50. No. 10, pp. 791, 2002.
[6] D. Jean-Pierre, B. Gery Jean, D. Sutter, *Synthesis of Arithmetic Circuits, Wiley,* 2006.
[7] R. Prakash, P. Peter, S. Leena, *System On a Chip Verification,* Kluwer Academic Publishers, 2002.
[8] Moonvin Song, Ohkyun Kwon, Yunmo Chung, "A Serial Connection Technique of Speakers for Multi-channel Audio systems," *IEEE Transactions on Consumer Electronics.* , Vol. 51, No. 2, May 2005.

저 자 소 개



송 문 빈(정회원)
1998년 한밭대학교 전자공학과 학사.
2002년 경희대학교 전자공학과 공학석사.
2006년 현재 경희대학교 전자공학과 박사과정.
<주관심분야 : SoC 설계, 임베디드 시스템, RTOS>

송 태 훈(정회원)
1985년 충북대학교 컴퓨터공학과 학사.
1990년 삼보컴퓨터 연구원.
1993년 한국산업기술대학원 공학석사.
2006년 현재 경희대학교 전자공학과 박사과정,
2006년 현재 (주)휴인스 대표이사.
<주관심분야 : ARM, 임베디드>

오 재 곤(정회원)
1989년 금오공과대학교 전자공학과 학사.
1989년 삼성전자 연구원
1995년 경희대학교 전자공학과 공학석사.
1999년 경희대학교 전자공학과 공학박사.
2006년 현재 한국산업기술대학교 전자공학과 교수.
<주관심분야 : SoC 및 임베디드 시스템, 차세대 디스플레이 시스템>

정 연 모(정회원)
1980년 경북대학교 학사.
1982년 KAIST 공학석사.
1987년 경제기획원 전산처리관.
1992년 미시간주립대학교 공학박사.
2006년 현재 경희대학교 전자정보대학 교수
<주관심분야 : SoC 설계, 임베디드 시스템, RTOS>