

비동기 방식의 직렬통신 시스템에서 헤드 검출 기능을 가진 회전기용 리시버의 FPGA 구현

강봉순* · 이창훈** · 김인규** · 하주영** · 김주현***

A FPGA Implementation of a Rotary Machine Receiver with Detecting a Header on the Asynchronous Serial Communication System

Bongsoon Kang* · Chang-hoon Lee** · In-kyu Kim** · Ju-young Ha** · Ju-hyun Kim***

요 약

본 논문에서는 회전기의 위치에 대한 정보를 가지고 있는 Encoder와 정해진 경로(Serial Signal)를 통해 Data를 전송하거나 받고, DSP로 Data를 정해진 경로(Parallel Signal)를 통해 Data를 전송하거나 받는 기능을 가진 Receiver를 FPGA Design한 것이다. 동일 동작 Clock을 사용하지 않는 다른 System의 Serial Data 통신을 하는 경우, 새로운 헤드 검출 방법을 제시하여 Serial Data의 유효한 각 Bit의 정보를 얻기 위해서는 헤더 내의 Sync. Code를 해석하여 유효 Data의 길이를 찾을 수 있다. 또한 Receiver의 동작 주파수를 'clk_select' Port를 사용하여 내부 동작 주파수를 20MHz 또는 60MHz를 선택할 수 있다.

ABSTRACT

This paper presents the design and implementation of a receiver operating between a rotary machine encoder and DSP. The receiver connects with the encoder using 1 bit serial data and DSP using 16 bits bus line.

The receiver and encoder use the different operating frequency each other. We suggest a new apparatus and method of synchronized code for header detection in 1bit serial communication. The system operating frequency can be changed into 20MHz or 60MHz by using the external port such as 'clk_select'.

키워드

Rotary Machine Encoder, Sync. code

1. 서 론

최근 컴퓨터의 발달로 모든 기계가 마이크로프로세서를 내장하여 고속, 고정밀도의 디지털화로 변화하고 있다. 그 중 회전기에 장착되어 있는 Encoder는 속도, 토크(torque), 위치 제어를 위한 회전 정보를 검출하는 장치로써 계측기기, 통신기기, 건설장비 등의 산업기기 시장에 널리 공급되어

있다[1-3]. 특히 광학식 Encoder는 산업용의 Robot, 모터 등 기기의 가동부 위치나 속도를 정확히 검출하고 그 정보를 구동부에 Feedback 하기 위하여 사용되고 있다. 광학식 Encoder의 구조는 발광 소자에서 나오는 빛은 유리로 된 원판에 등간격으로 분할이 되어 있는 회전격자와 고정격자를 통과하여 수광 소자에서 검출되는 Pulse로 현재 회전의 위치 정보를 알 수 있다[3-6].

* 교신저자: 동아대학교 전기전자컴퓨터공학부 조교수 (bongsoon@daunet.donga.ac.kr) ** 동아대학교 전자공학과 석사과정
접수일자 : 2004. 9. 22

*** 동아대학교 전자공학과 박사과정

본 논문에서는 회전기 내부의 Encoder에서의 출력인 비동기 1 bit Serial Data을 외부의 DSP에게 16 bits Parallel Data의 형태로 전달하는 기능을 가진 Receiver를 FPGA Design한 것이다.

이를 위한 본 논문의 구성은 다음과 같다. 2장에 본 논문에서 제시한 System에서 처리하는 외부 신호의 구성과, 3장에 각 외부 신호를 처리하기 위한 내부 하드웨어 구조, 그리고 4장에 본 System의 합성 및 실험 결과를 나타내었고, 마지막 5장에서는 결론으로 맺는다.

II. 외부 신호의 구성

2.1 Data Format Between Receiver and Encoder

그림 1에서는 Receiver와 Encoder사이의 주고받는 통신 사이의 Data Format을 각 가능 Field 별로 구분한 것이다.

Transmitting Request는 DSP에서 받은 명령어(Request ID)를 CF(Control Field) 형태로 변환하여 Encoder에게 Serial 형태로 전송을 한다. Receiving Request는 Encoder의 출력으로 Receiver에서 전송 받은 Request ID에 해당하는 유효 Data를 각 Field 별로 구분하여 Serial 형태로 전송을 한다.

그림 1에 해당하는 정보를 각 Field 별로 구분하여 설명하면 다음과 같다. 각 Filed는 "start bit = 0₂"과 "delimiter bit = 1₂", 그리고 유효 Data(8 bits)을 포함한 총 10 bits로 구성되어 있다.

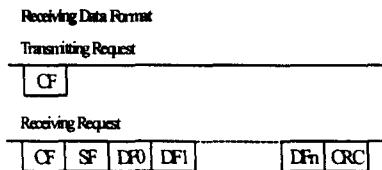


그림 1. 송신단과 수신단의 데이터 Format
FIG. 1 Transmitting and receiving data format

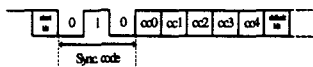


그림 2. CF의 내부 Data
FIG. 2 CF Field

그림 2는 CF의 내부 Bit의 구성이다. CF은 Encoder와 Receiver의 통신에서 헤더에 해당하는 첫 Field이다. "Sync. code"는 Encoder에서 전송되

는 1 bit Data의 유효 길이를 해석하기 위한 Code이다. cc0 ~ cc3은 4 bits(LSB First)로 구성되어 있는 명령어(ID Code)이다. Encoder로 전송 시에는 DSP에서 받은 Request ID를 Field형태로 변환하여 Serial Data로 전송한다. 또한 Encoder에서 전송 받은 CF내의 ID Code는 이후에 들어오는 DF(Data Field)의 정보 형태를 의미한다. cc4는 cc0 ~ cc3에 해당하는 Odd Parity이므로, ID code의 오류 검출을 할 수 있다.

SF(Status Field)는 Encoder의 특성 해당하는 Encoder Information, Encoder Alarm, Communication Alarm의 정보를 가지고 있다.

DF(Data Field)는 CF내의 ID Code에 해당하는 회전기의 위치 정보를 가지고 있다. 표 1은 각 ID Code에 해당하는 Data Field의 정보와 길이이다.

표 1. 각 ID Code에 해당하는 Data Field
Table. 1 Receiving data field of each ID code

ID Code	DF0	DF1	DF2	DF3
ID0	ABS 0	ABS 1	ABS 2	-
ID1	ABM 0	ABM 1	ABM 2	-
ID3	ABS 0	ABS 1	ABS 2	ENID
ID Code	DF4	DF5	DF6	DF7
ID0	-	-	-	-
ID1	-	-	-	-
ID3	ABM 0	ABM 1	ABM 2	ALMC

표 1에서 ABS(Absolute Single turn)는 회전기의 1회전에 해당하는 위치의 정보를 20 bits로 표현한다. ABM(Absolute Multi turn)는 회전기의 다회전에 해당하는 정보를 16 bits로 표현한다. ENID는 000100012로 고정되어 있는 Field이다. ALMC는 회전기의 전원에 관한 정보가 들어 있는 Field이다[7].

CRC Field는 "G(X) = X⁸ + 1"의 생성 다항식을 이용하여 Encoder에서 전송된 Data의 오류 검출을 할 수 있다.

2.2 Data Output Between Receiver and DSP

Encoder에서 받은 Serial Data를 Receiver내에서는 각 필요한 연산을 거친 후 DSP에게 전송하기 위한, 각 Address에 해당하는 Parallel Data를 Bus Line을 통해서 출력을 한다. 표 2는 ABS와 ABM의 정보를 모두 가지고 있는 Request ID(ID3)의 DSP Address에 해당하는 16 bits Parallel Data 형태이다.

표 2. ID 3에 해당하는 출력 Data
Table. 2 Output data bus : Request ID(ID3)

Address			Data Bus Line																	
A2	A1	A0	D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15		
0	0	0	Transmitting Request ID								0	0	0	0	0	0	0	0	0	0
0	0	1	Control Data								ABS									
0	1	0	ABS								ABS									
0	1	1	0	0	0	0	1	0	0	0	0	1	ABM							
1	0	0	ABM								0	0	0	0	0	0	0	0	0	0
1	0	1	ALMC								SF									
1	1	0	CRC								Communication Alarm									

Request ID가 0011₂일 때는 표 2와 같은 16 bits Data형식으로 DSP에게 전달하게 된다.

“Transmitting Request ID”의 내부에는 DSP로부터 받은 Request ID를 나타내는 영역이다. “Control Data”의 내부에는 “Communication Alarm”의 상태와 Encoder로 전송한 ID Code와 전송 받은 ID Code의 비교와 Encoder로부터 전송 받은 ID Code를 나타내는 영역이다. “Communication Alarm”의 내부에는 Encoder에서 전송 받은 각 Data에 대한 Error를 나타내는 영역이다.

III. 하드웨어 구조

그림 3은 본 논문에서 제시하는 Receiver의 내부 하드웨어 Block Diagram이다.

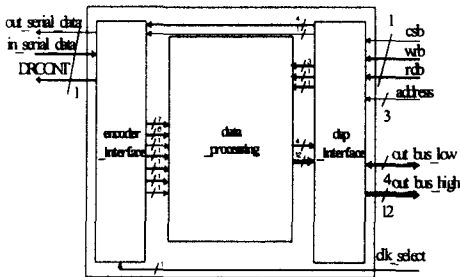


그림 3. Receiver의 내부 블록도
FIG. 3 Block diagram of the Receiver

“dsp_interface” Block은 DSP에서 Request ID (Bus Line 하위 4 bits)를 전송 받는 기능과 Bus Line으로 Data(16 bits)를 전송하는 기능을 한다. “data_processing” Block은 Encoder에서 받는 Data에 대한 몇 가지 연산을 거친 후 DSP에게 전송할 Data 형태로 정렬 시키는 기능을 한다. “clk_select” Port의 기능은 Receiver의 동작 주파

수를 20MHz 또는 60MHz을 사용하더라도 내부 동작을 동일하게 동작이 가능하게 구성되어 있어, 실제 동작 주파수를 사용자가 외부에서 선택할 수 있다.

Encoder와 Receiver System은 동일 동작 주파수를 사용하지 않고, 단지 Encoder는 2.5Mbps의 전송 속도로 Receiver와 Serial 통신을 한다. 그러므로 Serial Data의 유효한 각 Bit의 정보를 얻기 위해서는 Encoder에서 전송 받는 각 Field 중에서 처음으로 전송되는 CF내의 “Sync. code”를 해석하여 유효 1 bit Data의 길이를 찾을 수 있다. 종래 기술에서 “Sync. code”은 선별하기 위하여 Receiver에서 사용되는 동작 주파수를 사용하여 일정 시점에서 들어오는 Serial Data의 논리값을 비교한다. Encoder와 Receiver는 동작 주파수의 위상이 같을 때 Serial Data의 논리값을 판별할 수가 있지만, 동작 주파수의 위상이 달라질 때는 논리값을 판별할 수가 없어 전송되는 Serial Data에서 헤드에 포함되어 있는 “Sync. code”를 정확히 선별할 수 없는 경우가 발생함으로 전송되는 유효 Data를 인식하지 못하는 문제점이 발생한다.

그림 4는 종래기술을 사용하여 Receiver내에서의 사용되는 동작 주파수를 사용하여 일정 시점에서 들어오는 Serial Data의 논리값을 보여주는 것이다.

A와 B 시점인 Clock의 상승 에지(edge)에서 전송된 Serial Data의 논리 값들이 ‘1’로 같은 상태이므로 “Sync. code”를 선별하여 이후의 신호가 의미 있는 신호, 즉 유효 Data임을 알게 된다.

그림 5는 그림 4와 같은 종래기술을 사용하여 Receiver내에서의 사용되는 동작 주파수를 사용하여 일정 시점에서 들어오는 Serial Data의 논리값을 보여주는 것이다. 그림 5의 경우에는 논리값을 판별하는 A와 B 시점은 그림 4와 동일하나 전송된 1 bit Serial Data의 시점이 달라서 Receiver에서는 Data의 논리 값들이 ‘0’과 ‘1’로 서로 다른 상태이다. 따라서 Encoder에서는 Serial Data를 전송하고 있지만 Receiver는 “Sync. code”를 선별하지 못하여 이후의 전송되는 유효 Data를 인식하지 못하는 문제점이 발생한다.

이와 같은 문제점을 해결하기 위하여 본 논문에서 제시하는 Receiver의 내부 “encoder_interface” Block에서는 Encoder에서 Serial Data를 어느 시점에서 보내더라도 항상 “Sync. code”를 정확하게 선별하여 헤드 정보 이후의 유효 Data를 검출하는 방법을 사용하였다.

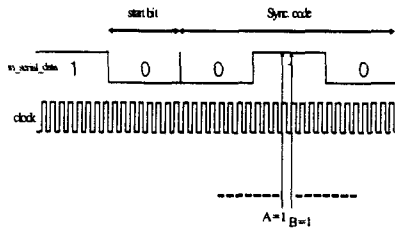


그림 4. 종래 기술을 사용한 예-1
FIG. 4 Post method of synchronized code (Example-1)

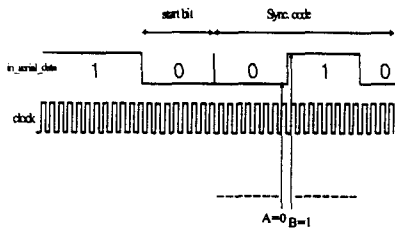


그림 5. 종래 기술을 사용한 예-2
FIG. 5 Post method of synchronized code (Example-2)

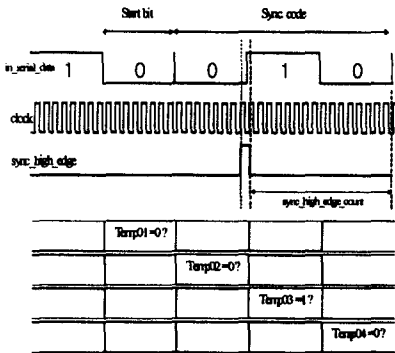


그림 6. 새로운 방법을 사용한 예
FIG. 6 New method of synchronized code (Example)

그림 6은 새로운 방법 "Sync. code"를 정확하게 선별하는 방법이다.

Serial Data의 값이 '0'→'1'로 상태가 변화하는 시점을 검출하여 이 시점을 "Sync. code"를 선별하는 시점으로 사용한다. 선별 시점을 기준으로 "start bit"인 Temp01시간에서 데이터의 값이 "0" 인지와 "Sync. code"인 Temp02, Temp03, Temp04 시간에서 데이터의 값들이 "010" 인지를 비교한다. 이상의 조건이 참일 경우, 전송된 Serial Data가 "0010" 이후에 전송되는 값들은 Encoder에서 전송하는 Data로 해석을 한다.

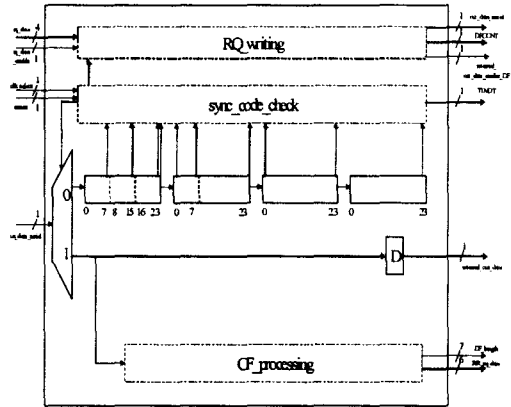


그림 7. "encoder_interface"의 내부 블럭도
FIG. 7 Data Path Block Diagram of "encoder_interface"

그림 7은 그림 6에서 제시하는 "Sync. code" 선별 방법을 이용하여 Encoder에서 전송되는 Serial Data를 선별하는 장치의 내부 구조를 나타낸 것이다.

본 논문에서 제시하는 Receiver의 동작 주파수를 20MHz 또는 60MHz를 사용할 수 있다. Encoder와 Receiver의 Serial Data 전송 속도는 2.5 Mbps이다. 본 논문에서는 어떠한 동작 주파수로 Receiver가 구동되더라도 "Sync. code"를 해석하기 위해서는 내부의 하드웨어 구조가 달라져야 한다. Serial Data의 값이 '0'→'1'로 상태가 변화하는 시점을 검출하는 방법은 동일하다. 그러나 Receiver의 동작 주파수가 60MHz일 때는 24 bits로 구성되어 있는 플립플롭, 4그룹의 상태를 비교하여 "Sync. code"를 선별하게 된다. Receiver의 동작 주파수가 20MHz일 때는 8 bits로 구성되어 있는 플립플롭, 4그룹의 상태를 비교하여 "Sync. code"를 선별하게 된다. Receiver에 사용되는 동작 주파수의 구분은 외부의 'clk_select' Port에 의해서 구분된다. 'clk_select'가 '0'일 때는 Receiver에 사용되는 동작 주파수가 20MHz를 기준으로 Encoder에서 전송되는 Serial Data의 "Sync. code"를 선별한다. 'clk_select'가 '1'일 때는 Receiver에 사용되는 동작 주파수가 60MHz를 기준으로 Encoder에서 전송되는 Serial Data의 "Sync. code"를 선별한다. 그림 8과 9는 Receiver의 동작 주파수가 20MHz와 60MHz일 때에 "Sync. code"를 선별하는 파형이다.

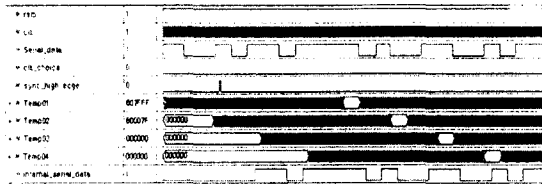


그림 8. 20MHz 시 "Sync. code" 선별 동작
FIG. 8 Detecting Sync. code : 20MHz

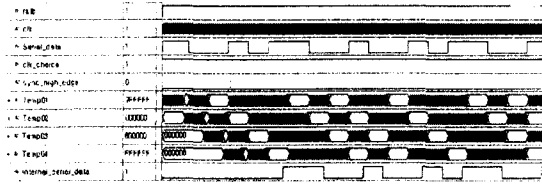


그림 9. 60MHz 시 "Sync. code" 선별 동작
FIG. 9 Detecting Sync. code : 60MHz

IV. 합성 및 실험 결과

Design된 Receiver의 동작을 검증하기 위하여 그림 10과 같이 Built-in Data인 "DSP"와 "Encoder", 그리고 Control Signal을 전송하는 "power_on_default"를 추가하여 전체 Simulation을 하였다.

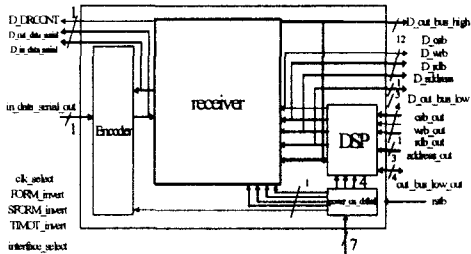


그림 10. 전체 Simulation을 위한 Block Diagram
FIG. 10 Block diagram for the simulation

설계된 Receiver는 Synopsys사의 Design Analyzer와 삼성사의 STD90(0.35 μ m, 3.3V) 라이브러리를 통하여 합성하여 비교 연구를 수행하였다. 구현된 Receiver와 Built-in Data인 DSP와 Encoder는 Altera사의 FPGA인 FLEX10K70RC2 40-4를 사용하여 구현하고, 검증을 수행하였다[8].

표 3. 각 모듈의 합성 결과
Table. 3 Synthesis and implementation result

Module Name	Altera FPGA FLEX10K70RC240-4		Synopsys Design Analyzer	
	Slice	Minimum Timing [ns]	Slice	Data Arrival Time [ns]
dsp_interface	36 (1%)	13.70	270	1.36
encoder_interface	309 (8%)	35.50	2,186	8.15
data_processing	378 (10%)	31.30	2,233	9.95
receiver	707 (18%)	34.80	4,683	6.60
dsp	116 (3%)	47.60	578	5.77
encoder	470 (12%)	41.80	4,127	9.19
power_on_default	16 (1%)	8.00	124	0.53
encoder_receiver_dsp	1,403(37%)	46.10	9,520	9.19

그림 11은 ID3일 때의 무한 반복적으로 전체 Simulation의 동작을 보여주는 것이다. "Sync. code processing"은 Encoder에서 전송되는 CF내의 "Sync. code"를 보여주는 것이다. "Parallel output"은 Encoder에서 받은 Serial Data를 Receiver내부에서 필요한 기능을 한 후 DSP로 각 Address에 해당하는 Data(16 bits)의 Bus Line을 보여주는 것이다. Encoder의 Data는 내부 Built-in Data를 사용하여 전체 Simulation을 실행한 것이다. 그림 12는 그림 11의 "Parallel Output"부분을 확대한 것이다. 출력되는 Data의 시점은 해당 Address가 들어온 후 4 clock이후에 출력되며 5 clock동안 유지된다.

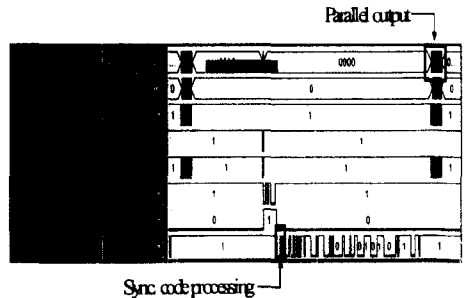


그림 11. Logic Analyzer에서 Display된 전체 Simulation Waveform
FIG. 11 Simulated Waveform on the Logic Analyzer

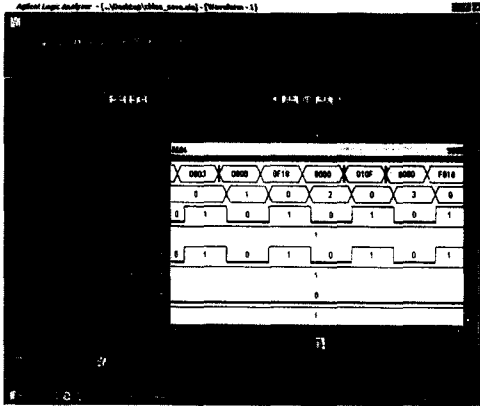


그림 12. Logic Analyzer에서 Display된 Parallel output
FIG. 12 Parallel output on the Logic Analyzer

V. 결론

본 논문에서 FPGA Design한 Receiver의 기능은 Serial-to-Parallel Conversion 시스템이다. Encoder와 Receiver가 1 bit Data만을 사용하여 데이터의 송수신을 수행하는 별개의 동작 주파수로 동작하는 경우에, Encoder에서 전송되는 1 bit Serial Data의 헤드에 포함되어 있는 "Sync. code"를 정확하게 해석하여 헤드정보 이후의 유효 Data를 검출하는 방법 및 장치를 사용하였다. 또한 Receiver의 동작 주파수의 종류를 사용자가 선택하더라도 전체 System의 동작은 동일하므로 고속으로 Serial-to-Parallel Conversion 기능을 할 수 있을 것이다.

감사의 글

본 논문에서 사용한 Synopsys사의 Design Analyzer와 Altera사의 Quartus, Active HDL software는 IDEC을 통해 지원 받았음.

참고문헌

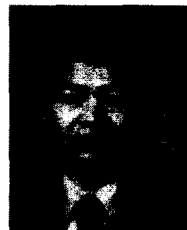
[1] K. Rajashekara, A. Kawamura and K. Matsuse, *Sensorless Control of AC Motor Drivers*, IEEE Press, 1996.
[2] R. Areny, "Sensor and signal conditioning," *Wiley interscience*, pp.433~444, 2000.
[3] J. Mayer, "High-resolution of rotary encoder analog quadrature signals," *IEEE Trans.*

ns. Instrum. Meas. Vol. 43, No. 3, pp.494~498, 1994.

[4] J. G. Webster, *The measurement, instrumentation, and sensors handbook*, CRC Press, 1999.
[5] R. P. Areny, *Sensors and signal conditioning*, Wiley Interscience, 2000.
[6] Y. Dote and S. Kinoshita, *Brushless Servomotors*, Clarendon Press OXFORD, 1990.
[7] S. X. Chen, T. S. Low and B. Bruhl, "The robust design approach for reducing Cogging Torque in Permanent Magnet Motor," *IEEE Trans. Magn.* Vol. 34, No. 4, pp.213~217, 1998.7.
[8] Altera Corporation, *FLEX10K EPF10K70 RC 240-4 (FPGA Device) Datasheet*, 1999.

저자 소개

강봉순(Bongsoon Kang)

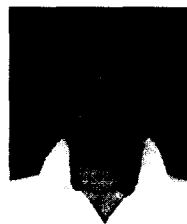


1985년 연세대학교 전자공학과 (공학사)
1987년 미국 University of Pennsylvania 전기공학과 (공학석사)
1990년 미국 Drexel University 전기 및 컴퓨터공학과 (공학박사)

1989년~1999년 삼성전자 반도체 수석연구원
1999년~현재 동아대학교 전기전자컴퓨터 공학부 조교수

※관심분야 : VLSI Design, ASIC Design

이창훈(Chang-hoon Lee)



2002년 동아대학교 전기전자컴퓨터 공학부 전자전공 (공학사)
2002년~현재 동아대학교 대학원 전자공학과 석사과정

※관심분야 : OFDM FFT, DAB FFT

김인규(In-kyu Kim)



1996년 동아대학교 전기전자컴퓨터 공학부 전자전공 (공학사)
2004년~현재 동아대학교 대학원 전자공학과 석사과정

※관심분야 : EnDat 2.1



하주영(Ju-young Ha)

2003년 동아대학교 전기전자컴퓨터
공학부 전자전공 (공학사)
2003년~현재 동아대학교 대학원
전자공학과 석사과정

※관심분야 : System IC Design



김주현(Ju-Hyun Kim)

2002년 동아대학교 전기전자컴퓨터
공학부 전자전공 졸업 (공학사)
2004년 동아대학교 대학원 전자공
학과 (공학석사)
2004년~현재 동아대학교 대학원
전자공학과 박사과정

※관심분야 : System IC Design