

MPW (Multi-Project Wafer) 2014년 MPW 진행 현황

공정	회차구분 (공정_년도순서)	모집팀수 ((mmxmm)x 칩수)/회별	정규모집 신청마감	참여팀수 ((mmxmm)x 칩수)	DB 마감 (Tape-out)	Die-out	비고
삼성 65nm	S65-1401	(4x4)x48	2013.12.09	(4x4)x20	2014.02.17	2014.08.18	제작중
	S65-1402		2014.02.03	(4x4)x33	2014.08.25	2015.02.27	설계중
	S65-1403		2014.06.02	(4x4)x10	2014.12.15	2015.06.12	정규모집예정
매그나칩/ SK하이닉스 0.18μm	MS18-1401	(3.8x3.8)x20	2013.12.09	(3.8x3.8)x20	2014.02.24	2014.07.28	제작중
	MS18-1402		2014.01.06	(3.8x3.8)x25	2014.05.19	2014.10.20	설계중
	MS18-1403		2014.02.03	(3.8x3.8)x19	2014.08.11	2015.01.12	설계중
	MS18-1404		2014.05.05	(3.8x3.8)x10	2014.11.10	2015.04.13	정규모집중
매그나칩/ SK하이닉스0.35μm	MS35-1401	(5x4)x20	2014.01.06	(5x4)x20	2014.06.16	2014.10.06	설계중
	MS35-1402		2014.06.02	(5x4)x2	2014.12.01	2015.03.23	정규모집예정
동부0.11μm	D11-1401	(5x2.5)x24	2013.12.09	(5x2.5)x24	2014.04.02	2014.08.06	제작중
	D11-1402		2014.04.07	(5x2.5)x17 (2.35x2.35)x14	2014.10.01	2015.02.04	설계중
동부0.18μm BCD	D18-1401	(5x2.5)x4	2013.12.09	(5x2.5)x4	2014.03.27	2014.07.02	제작중
	D18-1402		2013.12.09	(5x2.5)x4	2014.04.30	2014.08.06	설계중
	D18-1403		2014.02.03	(5x2.5)x3 (2.35x2.35)x2	2014.08.13	2014.11.19	설계중
	D18-1404		2014.05.05	(5x2.5)x1	2014.11.05	2015.02.11	정규모집중
동부0.35μm BCD	D35-1401	(5x2.5)x6	2013.12.09	(5x2.5)x6	2014.02.19	2014.05.28	설계중
	D35-1402		2014.01.06	(5x2.5)x6	2014.05.28	2014.09.03	설계중
	D35-1403		2014.03.03	(5x2.5)x2 (2.35x2.35)x8	2014.09.10	2014.12.17	설계중
	D35-1404		2014.05.05	(5x2.5)x2	2014.11.19	2015.02.25	정규모집중
TowerJazz 0.18μm BCD	TJB18-1401	(5x2.5)x6	2014.01.06	(5x5)x2	2014.05.12	2014.09.08	설계중
	TJB18-1402		2014.04.07	-	2014.10.20	2015.02.16	추가모집예정
TowerJazz 0.18μm CIS	TJC18-1401	(2.5x2.5)x4	2014.01.06	(2.5x2.5)x4	2014.05.05	2014.09.01	설계중
	TJC18-1402		2014.04.07	(2.5x2.5)x4	2014.10.13	2015.02.09	설계중
TowerJazz 0.18μm CA18HA	TJR18-1401	(2.5x2.5)x4	2014.01.06	(2.5x2.5)x4	2014.05.12	2014.09.08	설계중
	TJR18-1402		2014.04.07	(2.5x2.5)x2	2014.10.20	2015.02.16	추가모집예정
TowerJazz 0.18μm SiGe	TJS18-1401	(2.5x2.5)x4	2013.12.09	(2.5x2.5)x4	2014.03.11	2014.07.08	제작중

* 일정은 사정에 따라 다소 변경될 수 있음.
 * 우선/정규 모집은 마감일 2주전부터 신청 가능함
 * 회차 표기 방법 변경 : 공정코드_년도 모집순서 (예시) 삼성65nm 2014년1회차:S65-1401)
 * Package 제작은 Die out 이후 1개월 소요됨
 * 동부와 TowerJazz 공정은 sub chip(5mmx2.5mm 또는 2.35mmx2.35mm)으로 분리하여 모집
 * 선정 결과는 모집 마감 후 15일 이내 개별 통보됨
 * 2014년 우선모집은 완료됨에 따라 일정에서 제외함
 * 기준일 : 2014. 4. 30.

* 담당 : 이의숙 (042-350-4428, yslee@dec.or.kr)

2014년 5월 교육프로그램 안내

수강을 원하는 분은 IDEC홈페이지(www.idec.or.kr)를 방문하여 신청하시기 바랍니다.

KAIST 개설 강좌 안내

센터명	강의일자	강의제목	분류
본센터	5월 9일	하루만에 배우는 고주파 임피던스 정합	설계강좌
	5월 28일-30일	[FPGA 초급] ISE를 이용한 FPGA 설계 교육	

• 강좌일 : 5월 9일
 • 강좌제목 : 하루만에 배우는 고주파 임피던스 정합
 • 강사 : 이창석 교수 (한밭대학교)

강좌개요 임피던스 정합을 이해하기 위해 전송선의 특성 및 특성임피던스의 의미를 이해하고, 반사계수를 정의하여 전압 전류가 결정되는 과정을 이해하며, 임피던스 변환의 과정을 이해한다.
수강대상 Smith Chart를 보기는 했으나 원치 잘 모르겠다고 느끼는 분
강의수준 초급
강의형태 이론
사전지식 선수과목 옴의 법칙, 직병렬 저항 계산, 복소수의 사칙연산

• 강좌일 : 5월 28일-30일
 • 강좌제목 : [FPGA 초급] ISE를 이용한 FPGA 설계 교육
 • 강사 : 김민석 팀장 (리버트론)

강좌개요 • 1일차 : FPGA의 구성물을 이해함으로써 차후 보드 개발이나 디자인 개발을 할 때에 FPGA 구성물을 활용하여 쉽게 디자인 하고 또한 다른 외부 장치와 인터페이스 할 때에 서로의 속성에 맞게끔 연결, 연구원의 디자인 개발을 돕는다.

• 2일차 : FPGA를 개발하기 위해서 사용하는 S/W인 ISE 전체 개발 흐름을 이해하고 이것은 FPGA를 구성하는 흐름으로써 어떠한 문제점이 생겼을 때에 어느 부분에서 해결을 해야 할지 알게 되며, 간단 디자인을 통해 기본 디자인 습득을 하는 과정이다.

• 3일차 : 기본 응용 수업으로써 HDL을 이용하여 7-Segment를 구현하는 실습이다. 기본 스텝모터 구동에서부터 외부 입력이 들어 오는 것에 따라 모터의 속도를 제어함으로써 FPGA를 이용한 컨트롤 디자인을 실습하는 과정이다.

수강대상 해당 분야 전공 대학원생, 학부생
강의수준 초급
강의형태 이론+실습

*문의 : KAIST IDEC 구매처 (042-350-8536, kjh9@idec.or.kr)

경북대 모바일-AP 플랫폼 센터 신/간/소/개



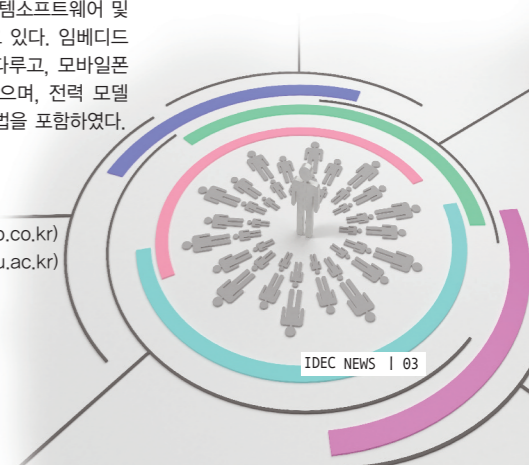
모바일 디바이스를 위한 소프트웨어 개발 방법 - 이론 및 구현

- 김철홍, 김중면, 정성우, 황인태 저 -

이 책은 모바일 AP로 지원되는 다양한 종류의 임베디드시스템을 구성시키는 소프트웨어 개발과 관련된 내용을 다룬다. 정보기전부터 차량, 로봇, 산업, 의료기기 등의 다양한 제품에 내장되어 다양한 컴퓨팅 파워를 구현하게 하는 요소로 모바일 AP의 활용도를 높이기 위해 필수적으로 요구되는 핵심기술인 임베디드 소프트웨어를 개발하는 과정에서 참고할 수 있도록 내용을 구성하였다. 이 책은 이론적인 내용과 병행하여 시장 점유율이 높은 모바일 디바이스에서의 시스템소프트웨어 및 응용소프트웨어 개발과 관련된 다양한 실습 내용을 함께 포함하고 있다. 임베디드 시스템에서 동작하는 응용 프로그램 개발 예제들을 실습용으로 다루고, 모바일 응용 소프트웨어 구현을 위해 필요한 이론적인 내용들을 소개하였으며, 전력 모델 개발을 예제로 모바일 디바이스에서의 시스템 소프트웨어 활용 방법을 포함하였다.

가격 : 19,000원

출판사 : 홍릉과학출판사 (Tel: 02-999-2274, hongpub@hongpub.co.kr)
 문 의 : 변보련 행정원(Tel: 053-950-6858, ipc-mobile@ee.knu.ac.kr)



제6회 동부 IP 설계공모전 시상식

김영일 한양대 박사과정, 전압조절칩 성능 향상으로 대상 수상

동부하이텍(대표: 최창식 사장)과 KAIST 반도체설계교육센터(박인철 소장, IDEC)은 24일 서울 르네상스호텔에서 '대학생 시스템반도체 설계 공모전'에 참가한 논문의 최종 심사를 진행했으며, 우수 논문을 발표한 5개 팀을 선정하여 시상했다.



6회째를 맞은 이날 공모전 심사에서는 지난해 6월부터 약 10개월간 설계생 산패키징-테스트 등의 과정을 거치면서 최종 칩의 성능을 검증한 결과를 담은 논문 28편이 발표되었다. 이날 심사에서는 전국 16개 대학 28개 팀이 최종 연구논문을 발표하였으며, 채수익 서울대 전기정보공학부 교수 등 반도체 전문가 15명으로 구성된 심사위원단이 설계의 우수성·상용화 가능성·창의성 등을 기준으로 최종 5개 팀을 선정했다.

수상	소속	수상자	지도교수	논문명
대상	한양대	김영일	이상선	Fast-Transient Capacitor-less LDO Regulator with an Assistant NMOS Bootstrap Technique
금상	한양대	김종석, 윤진오	최병덕	Low-ripple, High Efficiency Dynamic Pulse-Frequency Modulation (PFM) Buck Converter for Mobile Applications
은상	서경대	신용진, 장우진, 이준협	이승훈	A 521J/Conversion-Step 12b 10MS/s SAR ADC Using Minimum Unit Capacitors for Various TSC Applications
동상	서경대	윤병훈, 김민기, 조인수	임신일	적은 면적의 외부 커패시터가 없는 LDO 설계
동상	중앙대	이정윤, 류혁	백동현	LDO 병렬구조를 이용한 높은 출력 전력 해상도를 갖는 완전 집적된 스위치드 커패시터 DC-DC 변환기

VOD 서비스 안내

반도체 관련 강좌를 IDEC 오프라인 교육 뿐 만 아니라, IDEC 홈페이지를 통해서 VOD 서비스(유료)로도 강좌를 들으실 수 있습니다. 서비스 이용을 원하는 분은 IDEC 홈페이지(www.idec.or.kr)를 방문하여 신청하시기 바랍니다.

동영상 강좌 목록

No.	강의제목	강사명 (소속)	강의시간	이용료 (일반)	이용료 (학생)
1	CMOS 아날로그 회로 설계 및 실습	이강윤 교수 (성균관대)	10시간	30,000원	20,000원
2	고성능 ADC 설계를 위한 이론 및 설계기법	류승탁 교수 (KAIST)	7시간	21,000원	14,000원
3	최신 비디오 코덱 기술 이해 및 응용	김용환 책임 (KETI)	19.5시간	58,500원	39,000원
4	[DAC]CMOS 혼성모드 시스템 설계 및 실습	윤광섭 교수 (인하대)	2.3시간	6,900원	4,600원
5	SoC architecture 설계	유승주 교수 (포항공대)	7.5시간	22,500원	15,000원

6	RF 시스템 개요	유형준 교수 (KAIST)	2.6시간	7,800원	5,200원
7	[IDEC 연구원 교육]IDEC MPW 설계를 위한 교육	선혜승연구원의외 2명(IDEC)	24시간	72,000원	48,000원
8	무선전력전송용 송수신 회로 설계	이강윤 교수 (성균관대)	5시간	15,000원	10,000원
9	고성능 PLL 주파수 합성기 설계	조성환 교수 (KAIST)	10.6시간	31,800원	21,200원
10	[IDEC 연구원 강의]Full-Custom 설계 Flow 교육	조인신 연구원 (IDEC)	5시간	15,000원	10,000원

*문의 : KAIST IDEC 구제회 (042-350-8536, kjh9@idec.or.kr)



NDA가 체결된 Design Data 유출 금지 안내

IDEC의 MPW 참가를 통해 전달받은 Design Data 일체는 NDA를 통해서 법적인 구속력을 가지며, 관리 소홀로 인한 외부로의 공개 또는 유출 시 개인뿐만 아니라 개인이 속해 있는 WG에 자격 박탈과 같은 강력한 규제가 가해질 수 있습니다. 협약에 의해, 형사상 책임을 물을 수 있음을 알려 드립니다. MPW 참여자 분들은 Design Data 및 관련 자료의 관리를 철저히 하시어 불이익을 당하는 일이 없도록 거듭 당부 드립니다. NDA 체결 후 수령한 Design Kit 일체는 IDEC에 칩 수령 후 2개월 이내에 반드시 삭제하고, NDA 폐기확인서를 제출하여 제3자에 의한 공개 및 유출이 일어나지 않도록 주의 바랍니다.



11th International SoC Design Conference
Nov. 3-6, 2014, Jeju Island, S.Korea

CALL FOR PAPERS

<http://www.isocc.org>
Contact : secretary@isocc.org

International Organizing Committee

- ▶ General Chair
Jun Rim Choi (Kyungpook National Univ., Korea)
- ▶ General Co-Chairs
Jin-Ku Kang (Inha Univ., Korea)
Makoto Ikeda (Univ. of Tokyo, Japan)
Yeo Kiat Seng (Nanyang Tech. Univ., Singapore)
Shyh-Jye (Jerry) Jou (National Chiao Tung Univ., Taiwan)
Jun Jin Kong (Samsung Electronics, Korea)
- ▶ Conference Secretary
Kyung Ki Kim (Daegu Univ., Korea)

Technical Program Committee

- ▶ Technical Program Chair
Jinwook Burm (Sogang Univ., Korea)
- ▶ Technical Program Co-Chairs
Ken Choi (Illinois Institute of Tech., USA)
Tony Tae Hyung Kim (Nanyang Tech. Univ., Singapore)
An-Yeu (Andy) Wu (National Taiwan Univ., Taiwan)

A Unique Venue

- ▶ Seongsan Ilchulbong Tuff Cone



Seongsan Ilchulbong Tuff Cone was created by hydrovolcanic eruptions 100,000 years ago. The peak is a prime spot from which to view the sunrise. It was designated as a natural monument on July 19, 2000; a UNESCO World Natural Heritage site on July 2, 2007; a Global Geopark on October 1, 2010; and one of the New7Wonders of Nature on November 12th, 2011.

- ▶ Yacht Tour



General Purpose Of The Conference

International SoC Design Conference (ISOCC) aims at providing the world's premier SoC design forum for leading researchers from academia and industries. Prospective authors are invited to submit papers of their original works emphasizing contributions beyond the present state of the art. ISOCC 2014 is technically co-sponsored by IEEE CAS Society and accepted papers will be published on IEEE Xplore. We also welcome proposals for special sessions.

Conference Theme

The theme for ISOCC 2014 is "SoC for Smart Connectivity". Solutions for providing smart and secure connectivity will need to evolve new approaches to securing the shared resources. ISOCC 2014 is looking for novel SoC solutions to create truly smart connectivity.

Paper Submission

A complete 2-page manuscript must be submitted electronically in PDF format (in Standard IEEE double-column format posted on the conference website). Only electronic submission will be accepted. For more information, please refer to the conference website : <http://www.isocc.org>

Key Dates

- Deadline for submission of special session proposal : June 30, 2014
- Acceptance notice of special session proposal: July 10, 2014
- Deadline for submission of regular session full paper : July 15, 2014
- Deadline for submission of chip design contest: July 31, 2014
- Deadline for submission of special session full paper: July 31, 2014
- Notification of acceptance (all submitted papers): Sep. 01, 2014
- Deadline for author and early-bird registration: Sep. 15, 2014
- Deadline for submission of accepted papers : Sep.15,2014
- Deadline for chip design contest registration: Sep. 30, 2014

Conference Venue

RAMADA PLAZA JEJU HOTEL



Topics of Interest

- Analog and Mixed-Signal Circuits
 - Analog and Mixed-Signal Techniques
 - Data Converters
 - High-Speed Signal Interfaces
 - Wireline and Wireless ICs (RF ICs)
- Digital VLSI Circuits and Embedded Systems
 - Memory Circuits and Embedded Memory
 - Digital Circuits and VLSI Architectures
 - Multimedia (A/V) Algorithm and SoCs
 - Communication SoCs
 - Processors / Multi-Core Architectures & Software
 - Embedded Systems and Software
- SoC Design Methodology
 - HW-SW Co-design
 - SoC Testing
 - Design Verification
 - Signal Integrity / Interconnect Modeling and Simulation
- Low Power & Power Management ICs
 - Power Electronics / Energy Harvesting Circuits
 - Energy-Aware Systems
 - Low Power Design Techniques
- Application Specific SoCs & Emerging Technologies
 - Display Drivers
 - Image Sensors
 - Sensors and MEMS Circuits
 - Biomedical SoCs
 - Automotive SoCs
 - Nanoelectronic Devices and Circuits
 - 3-D SoCs & System-in-Package

From Jeju International Airport
by Taxi: 10 minutes' ride

- Website: <https://www.ramadajeju.co.kr>



IDEC 논단

Processing In Memory로의 진화

필자가 중사하는 메모리 특히 DRAM 분야는 대한민국이 20년 이상 세계 1위를 유지하고 있다 보니, System LSI나 ASIC 등의 산업에 비해 나라 경제에 이바지하는 바는 컸으나 참신성과 미래 지향성이 떨어져 상대적으로 대학의 Lab 그리고 연구 분야에 중사하는 학생들에게는 외면을 받아왔다. 그리고 최근의 반도체 공정 Scaling의 어려움과 이에 따른 DRAM 산업의 한계는 역설적으로 위기가 기회를 주는 가능성을 보여주고 있다. 이러한 가능성은 전체적인 Computer Architecture에서 하나의 분야로 발전해 갈 수 있으며, 한편은 많은 관련 연구의 Theme로도 방향성을 잡아갈 기대하면서 짧은 소견을 얘기하고자 한다.

프로세서의 진화와 메모리의 진화는 서로 다른 목적으로 진행되어왔다. 프로세서는 고가 공정을 써서라도 여러 가지 계산을 속도가 빠르면서 병렬적으로 처리하는 방향으로 진화하였고 메모리는(이하 DRAM과 같이 사용) 저가공정으로 대 용량을 가지면서 프로세서가 요구하는 데이터를 빠른 입출력 인터

페이스를 이용하여 전달해 줄 수 있는 방향으로 진화되었다. 이 진화를 시간상으로 뒤돌아보면 1990년대에는 시스템이 요구하는 용량을 늘리기 위해 DRAM 셀 종류의 경합을 통한 진화가 시작되었고 2000년대는 시스템의 DRAM Bandwidth의 증가 요구에 따라 DRAM 인터페이스의 입출력 속도를 늘리기 위한 방향으로 전개되었다. 2010년대는 시스템의 저전력 DRAM의 요구를 충족시키기 위해 공급전압을 낮추는 기술을 집중으로 진화하였다. 이렇듯 주 진화 방향이 시스템 요구를 충족하는 수동적인 방향으로 DRAM이 진화한 것이다. 이 때문에 프로세서와 메모리와의 성능은 차이가 커지게 되었고 이를 Von Neumann Bottleneck 또는 Memory Bandwidth Wall이라고 한다. 이러한 문제를 해결하기 위한 일반적인 방법은 프로세서와 가까운 곳에 큰 용량의 임시저장 장소(Cache)를 계층적으로 배치하는 것이다. Cache가 차지하는 면적이 프로세서 칩의 50%가 넘는 경우도 발생하고 있으나 사실 그 성능차를 극복했다고 말하기에는 부족한 현실이다. 이 방법은 기존의 프로세서 중심의 컴퓨팅에서 크게 벗어나지 않

은 접근방법이었다. 이와 다르게 메모리 중심의 컴퓨팅(Memory Centric Computing) 관점에서 이 성능차를 극복하려는 움직임이 1990년대 초부터 나타나기 시작했다. 이 컴퓨팅 방법은 용량이 큰 메모리가 중심이 되고 각종 연산기능을 할 수 있는 프로세서들을 메모리에 근접시키거나, 분할된 메모리 블록과 작은 단위의 프로세서를 한 단위로(Processing Element) 구성하여 2차원으로 배열시키는 등의 구조를 취하게 된다. 이 구조에 따라 메모리 중심 메모리 근접 컴퓨팅(Near Data Computing) 또는 Processing In Memory(PIM)등으로 불리게 되었다. 이 변화를 메모리 입장에서 보면 새로운 시스템 환경을 창출해 나가는 보다 창조적인 메모리로의 진화가 시작되었다고 볼 수 있다.

PIM은 여러 방향으로 진화될 수 있다. 예를 들어 특정 연산을 빠르게 수행할 수 있도록 돕는 가속기나 현재 DRAM에 PIM 기능만 추가시킨 스마트 메모리가 될 수 있을 것이다. 또한, 프로세서 없이 독립적으로 동작할 수 있는 혁신적인 형태의 PIM도 기대할 수 있다. 1990년 초에 업계에서는 최초로 실리콘으로 IBM에서 EXCUBE라는 DRAM 공정기반의 멀티프로세서를 처음 개발하였고 학계에서는 Berkeley에서 실리콘으로 DRAM 공정기반의 IRAM(Intelligent DRAM)을 개발하였다. 이 대표적인 두 경우 모두 다 프로세싱 로직을 집적도가 큰 DRAM 칩으로 이동시켜서 탑재된 프로세서와 DRAM 간에 대량의 데이터가 한 칩 내에서 전송할 수 있게 하였으며 이를 이용할 수 있는 ISA(Instruction Set Architecture)와 마이크로 아키텍처가 같이 개발되었다. 그러나 이 두 경우 모두 다 주요 DRAM 제조업체와는 협업이 없는 상태에서 이루어졌고 DRAM공정으로 로직의 성능을 구현하는 것에 있어 한계점에 부딪히게 되었다. 또한, DRAM의 용량이 Scalable 하지 않기 때문에 실제 제품으로 성공하기에는 부족한 점이 있었다. 이와 더불어 시스템의 환경을 많이 바꾸어야 하는 부담으로 범용로의 진입은 불가능하였다. 이후에도 PIM이 지속해서 연구 개발되고 있고, 최근 PIM이 재조명을 받고 있는데 그 이유는 두 가지 관점에서 설명할 수 있다. 첫째는 응용적인 측면이다. 향후 수십억 개의 디바이스가 서로 연결되는 IoT(Internet of Things)등을 통해 생성되는 비정형 또는 정형의 Big data를 처리하기 위해서는 대량의 병렬처리가 가능한 지능적 시스템(Intelligent System)이 요구되고 있다. 이 요건을 충족시킬 수 있는 프로세서가 PIM인 것이다. 또 다른 한가지 측면은 TSV 등을 통한 3D 메모리를 구현할 수 있는 기술이 확보되고 있다는 것이다. 이 3D 구조를 이용할 경우 프로세서가 탑

재된 실리콘을 위한 로직 공정과 DRAM을 위한 DRAM 공정이 별도로 최적화될 수 있기 때문에 기존 PIM 구현 시 한 쪽에 최적화된(DRAM 또는 프로세서) 공정을 사용함으로써 오는 성능저하를 회피할 수 있다. 그러나 아직 전반적인 시스템 환경(Ecosystem)은 보수적인 성향이 강하여 PIM을 받아들이기에는 준비가 되어 있지 않다. 따라서 Killer Application을 발굴하거나 전략적인 접근방법이 수반되어야 PIM의 제품화가 비로서 이루어 질 수 있을 것으로 예상된다. 예를 들어 서버시장은 다른 시장에 비해 더 보수적이고 시스템 환경(Ecosystem) 변화가 더 어렵다. 따라서 시스템 환경 변화의 수반을 요구하는 PIM이 서버시장에 진입하는 것은 쉬운 일이 아닐 것이다.

정리하면 PIM은 프로세서 중심의 컴퓨팅에서 벗어나서 기존 컴퓨터 시스템의 한계를 극복함과 동시에 Big Data 시대를 여는 열쇠 역할을 할 수 있다. 하지만 PIM의 첫 번째 상업적인 성공은 쉬운 일이 아니며 시스템 환경 변화가 적은 응용에서 이루어질 것이다. 이런 PIM은 프로세서 업체보다는 DRAM 제조업체에서 만들어 낼 가능성이 높다. DRAM 내의 구조를 PIM에 친화적으로 구현하면서 현 DRAM에 호환성 있게 만들 수 있는 잠재력을 충분히 가지고 있기 때문이다. 문제는 대량으로 표준화된 DRAM 생산체제가 시장의 규모가 불확실하고 응용에 따라 변화가 필요한 주문형의 메모리 비즈니스로 DRAM 업체가 진화할 수 있는냐는 것이다. 결론적으로 예전 DRAM의 진화는 프로세서가 요구하는 조건을 만족시키는 수동적인 진화였다면 향후 DRAM의 진화는 DRAM이 지능을 가지면서 새로운 환경을 창출해 나가는 창조적인 진화일 것이며 그 선두에 PIM이 있을 것으로 생각된다.



최 주 선 전무
삼성전자 메모리사업부 DRAM개발실장



Security SoC의 해킹 동향 및 대응 방법



컴퓨팅 기술의 발달로 정보 기기는 소형화되고 컴퓨팅 능력은 증가되었으며 네트워크 기술의 발달로 통신 기기의 연결성은 극대화되어 서로에 대한 접근성이 증대되었다. 한때 단순히 통화를 위해 사용했던 전화기는 현재 손안의 컴퓨터로서 스마트폰, 태블릿 등의 휴대용 기기로 진화하였으며 이를 이용하여 언제 어디서든 음원, 서적, 영화 등 다양한 콘텐츠를 이용하거나 모바일 뱅킹, 온라인 쇼핑, 게임 등의 서비스를 이용할 수 있게 되었다. 휴대용 기기 뿐만 아니라 정보 통신 기기는 자동차, 가전제품 등 다양한 형태의 기기와 융합하여 진화하고 있으며 서로 유기적으로 연결되어 온라인으로 서로의 정보에 접근하고 제어하는 것이 가능해지고 있다.

다양한 기기에 정보 통신 기기가 적용되어 인터넷으로 연결됨에 따라 사용자는 언제 어디서나 집안 상황을 확인하고 집 안의 기기를 제어하거나 원격으로 자동차의 시동을 걸어 미리 차안의 온도를 조절하는 등의 더욱 편리한 서비스를 이용할 수 있을 것으로 기대된다. 그러나 이와 함께 보안 문제가 크게 증가할 것으로 예상되고 있다. 실제로 스마트폰에 대한 피싱, 파밍, 스미싱 등의 공격은 사회적으로 이미 큰 문제가 되고 있으며 스마트 TV나 냉장고를 해킹해 스팸을 발송한 사례도 존재한다. 이러한 문제를 해결하기 위해 보안 기술의 필요성이 증가하고 있으나 보안 기술은 오버헤드로써 기존 기기의 성능을 저하시킬 수 있는 요인으로 작용할 수 있다. 또한 기존의 소프트웨어 기반의 보안 기술은 개방형으로 인해 끊임없이 취약점을 보완해야하는 단점이 존재한다. 이에 기존 시스템과 별도로 보안 기능을 독자적으로 수행하는 Security SoC(System On Chip) 기술이 대안이 될 수 있다.

Security SoC과 물리적 공격

Security SoC는 HSM(Hardware Security Module), 스마트카드와 같이 외부로 인터페이스만 존재하는 보안 전용 칩으로 내부 동작이 외부의 영향을 받지 않으므로 기존의 시스템으로부터 받는 영향이 최소화되며 기존 시스템과 별도로 동작하므로 기존 시스템의 성능을 저하시키지 않는다. 그러나 Security SoC도 모든 공격

에 안전한 것은 아니다. 일반적인 소프트웨어적인 해킹 방법과는 달리 물리적인 방법을 이용하는 방법은 그 만큼 고도의 기술과 고가의 장비, 오랜 분석 시간을 요하나 상대적으로 안전하다고 알려진 Security SoC에도 유효한 공격을 할 수 있을 만큼 매우 강력하다. 이러한 강도 높은 공격에는 부채널 공격, 메모리 공격, 역공학(reverse engineering)을 통한 버스 프루빙(bus probing) 공격 등이 있으며 공격 방법과 함께 대응 방법에 대한 연구가 활발히 수행되고 있다.

부채널 공격과 그 대응 방안

부채널 공격(SCA, Side Channel Attack)은 칩이 동작할 때 변화하는 전력 소모, 열, 연산 소모 시간, 전자기파(electro-magnetic wave) 등의 부가적인 정보, 즉 부채널 정보를 이용하는 공격방법이다. <그림 1>은 부채널 공격 모델을 나타낸다.

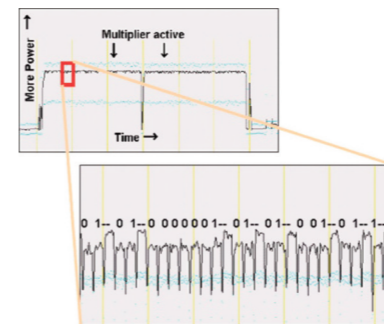


<그림1> 부채널 공격 모델

부채널 정보를 수집 및 분석하여 비밀 데이터 및 키 등의 주요 보안 정보를 추출해낼 수 있으며 공격 시 칩의 외부에서 접근 가능한 인터페이스만을 이용하기 때문에 상대적으로 적은 시간과 비용만을 필요로 한다. 따라서 security SoC의 가장 현실적이고 위협적인 공격 방식으로 인식되고 있다.

1. 부채널 공격 종류

부채널 공격은 사용되는 부채널 정보에 따라 시차 공격, 전력 분석 공격, 전자기파 분석 공격, 오류 주입 공격 등으로 분류할 수 있다. 시차 공격은 입력 데이터에 따라 달라지는 연산의 시간이나 연산 회수를 관찰하여 분석하는 방법이다. 예를 들어 대표적인 공개키 알고리즘인 RSA의 경우 키의 각 bit 값에 따라 내부에서 수행되는 연산의 종류가 달라지므로 이를 관찰하여 키 값을 추출하는 것이 가능하다 [1]. 전력 분석 공격은 공격 대상 암호 장치의 소비 전력과 연산 중 처리되는 데이터간의 상관도를 이용하는 것으로 부채널 공격 방법들 중에서 가장 효과적이고 위협적인 공격 방법으로 알려져 있다. 전력 분석 공격은 분석 방식의 복잡도에 따라 단순 전력 분석(Simple Power Analysis : SPA)과 차분 전력 분석(Differential Power Analysis : DPA)으로 나누어진다. SPA는 특정 명령어가 수행되는 한 시점에서 데이터에 따라 달라지는 소비전력을 분석하여 비밀정보를 유추하는 방법이다.



<그림2> RSA 연산 파형

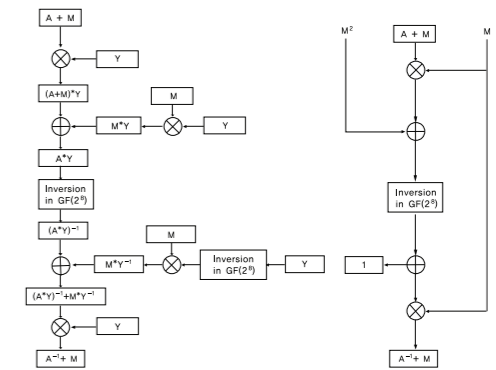
<그림 2>는 RSA의 파형을 분석한 것으로 연산 시간이나 파형 모양을 관찰하여 키 값을 분석하는 과정을 나타낸다. 그러나 이러한 SPA는 공격자가 공격하고자 하는 시점의 구현방법을 정확히 알고 있어야하는 단점이 있다. 반면 DPA는 비밀 정보 비트와 소비전력의 통계적인 상관관계를 이용하여 비밀 정보를 유추할 수 있는 방법으로 노이즈에 강인하고 아주 적은 자원을 사용하기 때문에 부채널 공격법 중 가장 강력한 공격법 중 하나로 알려져 있다.

전자기파 분석 공격은 공격 대상 암호 장치로부터 방사되는 전자기파 신호를 분석하는 방법으로 전력 분석 공격과 달리 원거리에서 정보의 습득이 가능하며 다중 채널로 구성되고 있어서 전력 분석 공격 대응 장치에서도 전자기파 정보의 분석이 가능하다는 장점이 있다. 전력 분석 공격과 비슷하게 단순 전자기파 분석 (Simple Electro-Magnetics Analysis : SEMA)과 차분 전자기파 분석 (Differential ElectroMagnetics Analysis : DEMA)으로 분류된다. 오류 주입 공격은 암호 연산을 위한 칩이나 하드웨어에 예상치 못한 결함을 유발시켜 발생된 잘못된 출력 값을 분석함으로써 내부의 비밀 정보를 알아내는 공격 방법이다. 칩의 패키징을 제거한 후 오류를 유발시키는 방법과 외부에서의 전기적 스파크를 일으켜 오류를 유발시키는 방법 등이 있다.

2. 부채널 공격 대응 방법

부채널 공격의 대응 방법으로 가장 많이 사용되는 것은 마스킹(masking) 방법과 하이딩(hiding) 방법이다. 마스킹 방법은 중간 과정에서의 연산 값을 랜덤하게 만드는 방식으로 원래 중간 연산 값과의 관계를 약하게 만들어 주는 것을 의미한다. 하이딩 방법은 연산 중간에 전력 소모량의 데이터 연관성을 제거하기 위해 소모량을 통일하거나 랜덤하게 만드는 방식을 의미한다.

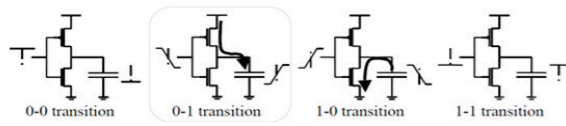
마스킹은 랜덤 수를 적용하는 연산 종류에 따라 부울 마스킹(boolean masking)과 산술 마스킹(arithmetic masking) [2]으로 구분된다. 부울 마스킹은 XOR이나 AND와 같은 선형 연산에 랜덤 수를 적용하며 산술 마스킹은 AES의 S-box와 같은 비선형 연산에 랜덤 수를 적용한다. 그러나 이렇게 추가된 랜덤 성분은 DPA 시 약해지기 때문에 고차원의 DPA [3]에 취약하다는 단점이 있다. 따라서 하이딩 기법으로 대체하거나 이와 병행해서 사용해야 한다.



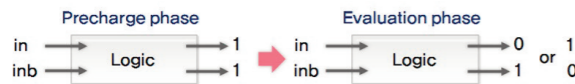
<그림3> 무작위수 M을 이용한 마스킹 기법

하이딩 방법의 주된 방식은 실행되는 동작과 데이터에 따른 전력소모량이 독립적이 되도록 회로를 구성하는 것이다. 일반적으로 DRP 회로 방식(Dual-rail precharge logic style)을 기반으로 하여 만들어진다.





〈그림4〉 일반적인 CMOS



〈그림5〉 DRP 회로

일반 CMOS(그림 4))는 입력 값에 따라 출력 값의 경로가 다르나 DRP(그림 5)의 경우 입력 값과 관계없는 출력 값을 가지며 모든 경우에 대하여 일정한 load capacitance를 갖는다. SABL (Sense Amplifier Based Logic) [4]과 WDDL (Wave Dynamic Differential Logic) [5]가 대표적인 DRP 회로로 SABL은 트랜지스터 수준에서 구현되며 WDDL은 일반 셀 라이브러리를 이용하여 만들어진다. 이러한 회로를 구성할 때에는 연결의 대칭이 중요하므로 대칭되는 노드 간 capacitance의 차이가 적도록 대칭적으로 설계하는 것이 중요하다.

메모리 공격과 그 대응 방안

메모리 공격은 메모리 내의 내용을 추출하거나 복제, 변경하는 방법이다. 공격 대상은 메모리의 위치에 따라 Off-SoC 메모리와 On-SoC 메모리로 나뉠 수 있으며 메모리의 종류에 따라 휘발성 메모리와 비휘발성 메모리로 나뉠 수 있다.

1. 공격 방법

Off-SoC 메모리의 경우 쉽게 메모리에 연결된 버스를 관찰하거나 메모리에 특수 처리를 하여 메모리 내의 정보를 획득할 수 있으며 공격자의 악의적인 코드가 담긴 메모리로 교체하는 등의 공격도 가능하다. On-SoC 메모리는 off-SoC 공격에 비해 상대적으로 공격이 매우 어려우나 위에서 설명할 역공학을 통해 칩 내부의 메모리에 접근하여 메모리의 내용을 추출하는 것이 가능하다. 그 밖에 메모리에 필요한 공급 전력을 차단하여 시스템의 불안정한 틈을 노릴 수도 있다.

메모리 공격의 대상은 비휘발성 메모리로 한정되지 않으며 휘발성 메모리도 공격이 가능하다. 휘발성 메모리의 경우 공급 전원이 차단될 경우 저장된 데이터가 사라지기 때문에 공격하기에 어려움이 있으나 완전히 데이터가 사라지기까지 약간의 시간을 소모하며 온도를 급격히 낮출 경우 그 시간은 수분에서 수 십분 이상으로 늘어난다. 따라서 전원이 차단된 비휘발성 메모리로부터도 저장되어 있던 데이터를 획득하는 것이 가능하다. 소거 기능을 사용하더라도 SRAM의 경우 반전에 의해 데이터가 소거되므로 소거 후 데이터로부터 원래 데이터를 추측할 수 있다.

2. 대응 방법

메모리 공격으로부터 중요 데이터를 보호하기 위해서는 메모리에 중요 데이터가 평문 상태로 저장되는 것을 방지해야한다. 그러나 데이터를 암호화하기 위해서는 반드시 키가 필요하므로 적어도 하나 이상의 키는 근원키로써 평문 형태로 저장되어야 한다. 따라서 이러한 문제를 해결하기 위한 방법이 필요하다. PUF(Physical Unclonable Function)는 메모리 공격의 대응방안으로 근원키를 구현하는데 사용될 수 있다. PUF는 말 그대로 복제 불가능한 함수로 똑같은 레이아웃을 이용하여 제작되나 칩마다 예측 불가능한 랜덤한 값을 생성하는 회로이다. PUF를 사용함으로써 각 칩은 고유한 근원키를 가지게 되며 이 키는 메모리에 저장되지 않는다. 추가적으로 키가 필요한 경우 근원키를 이용하여 암호화 후 메모리에 저장하게 된다. 〈표 1〉은 다양한 방식의 PUF를 나타낸다.

연도	구현 방법	제조사	layout
2000 [6]	CMOS의 랜덤 drain voltage	SiidTech, Portland State University	
2005 [7]	delay path의 속도 차	MIT	
2007 [8]	SRAM의 불안정한 초기 상태	Philips	
2009 [9]	랜덤 capacitance 값	semiconductors	

〈표1〉 다양한 방식의 PUF

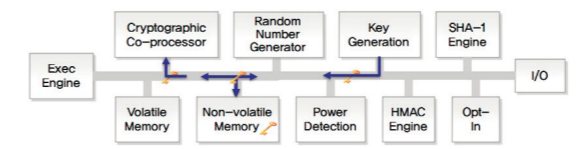
PUF는 키로 사용되는 것 외에도 값싼 ID로도 사용될 수 있는 등 다양하게 활용하는 것이 가능하나 랜덤성과 안정성, 높은 수율 등의 필요조건이 있어 쉽게 구현하기 어려운 기술이다. 현재 delay path의 속도 차를 이용한 arbiter PUF [7]와 SRAM의 초기 불안정한 상태를 이용한 SRAM PUF [8]를 각각 VERAYO와 INTRINSIC ID에서 상용화 중에 있으나 시불변성과 안정성을 확보하기 위한 ECC(Error Correction Code)를 필요로 하는 등의 문제가 있어 아직 출시된 제품은 전무한 상태이다. 그러나 PUF가 가지는 보안 특성으로 인해 기존의 단점을 개선할 수 있는 PUF에 대한 연구가 계속해서 진행되고 있으며 메모리 공격의 대응 방법뿐만 아니라 다양한 보안 프로토콜에서 사용될 수 있을 것으로 기대되고 있다.

역공학을 통한 버스 프루빙 공격과 그 대응 방안

역공학은 칩의 패키지를 제거하고 칩의 각 층을 하나씩 제거하며 레이아웃을 찍은 후 컴퓨터로 획득한 레이아웃을 분석하는 방법이다. 이를 통해 칩 내부의 레이아웃 정보를 모두 획득할 수 있으며 기술에 따라 칩이 동작하는 상태에서 외부 패키지만 제거한 후 원하는 곳의 신호를 관찰하는 것이 가능하다.

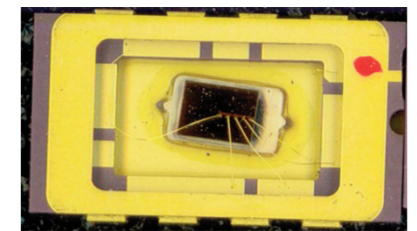
1. 공격 방법

Security SoC는 하나의 칩으로 되어있어 외부로부터 공격에 안전한 편이나 역공학 기술을 사용할 경우 칩 내부까지 관찰되어 중요 정보가 노출될 수 있다. SoC 칩의 내부가 관찰 가능할 경우 취약점을 살펴보기 위해 내부 구성을 살펴보면 〈그림 6〉과 같다.

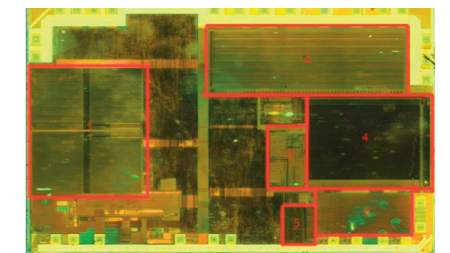


〈그림6〉 기본적인 내부 구성도

〈그림 6〉은 Secure SoC의 기본적인 내부 구성을 나타낸다. 기본적으로 칩의 동작을 주관하는 core인 execution engine이 있으며 데이터 저장을 위한 메모리와 외부와 인터페이스하기 위한 I/O, 마지막으로 보안 기능을 지원하기 위한 다양한 보안 모듈들로 구성되어 있다. 대부분의 보안 모듈들은 데이터의 암호화 및 복호화, 디지털 서명 생성 및 증명, HMAC 생성을 통해 중요 데이터에 대한 무결성 및 기밀성을 보장하고 사용자 인증, 부인 방지 등의 기능을 지원한다. 그러나 이러한 보안 기능들을 수행하기 위해서는 키를 사용하는데 이 키는 키 생성 모듈에서 생성되어 각 보안 모듈에서 사용되고 필요에 따라 메모리에 저장된다. 만약 bus를 관찰하는 것이 가능하다면 이러한 키 값이 노출될 수 있다. 실제로 스마트카드 칩의 내부 bus가 역공학 공격에 의해 관찰된 사례 [10]가 존재한다. 〈그림 7〉은 역공학을 하기 위해 칩의 외부 패키지를 제거한 모습이며 〈그림 8〉은 역공학에 의해 칩의 레이아웃이 분석된 모습이다.



〈그림7〉 외부 패키지가 제거된 칩

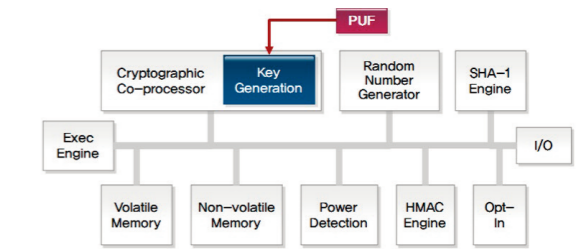


〈그림8〉 내부 레이아웃

역공학에 의해 내부의 레이아웃이 획득되면 〈그림 8〉과 같이 메모리나 코어 등의 IP는 쉽게 구별 가능하므로 이들 IP에 연결된 bus도 쉽게 찾아낼 수 있다. Infineon 66PE chip에 대한 해킹 사례에서는 이렇게 내부 bus를 관찰하여 지나는 데이터를 수집하고 분석하여 코드를 얻어내고 이를 이용하여 공격을 시도하였다. 이는 실제로 역공학을 통해 SoC 내부의 bus를 관찰하는 것이 가능하며 이는 bus를 통해 이동하는 데이터를 추출할 수 있다는 것을 의미한다. 비슷한 방법으로 bus를 통해 내부 코드뿐만 아니라 이동 중인 키 값도 추출될 수 있으며 특히 키가 노출될 경우에는 쉽게 내부 비밀정보를 복호화하거나 서명을 위조할 수 있다는 취약점이 존재한다.

2. 대응방안

SoC 내부의 bus를 오가는 데이터 중 특히 키 값을 보호하기 위하여 앞서 메모리 공격의 대응 방안에서 사용한 PUF를 사용할 수 있다. 〈그림 9〉는 PUF가 적용 시 변경되는 내부 구성을 나타낸다.



〈그림9〉 PUF가 적용된 내부 구성도

변경된 구성에서는 키 생성 모듈이 암호 모듈 내부에 존재하고 키 생성 모듈은 PUF를 내부에 포함하고 있다. 키 생성 모듈은 PUF를 기반으로 키를 생성하므로 각 SoC 칩마다 PUF 기반의 고유 키 값을 가지며 필요에 따라 키를 동적으로 생성하고 키를 메모리에 저장하지 않는다. 또한 키가 필요한 암호 모듈은 내부의 키 생성 모듈로부터 바로 키를 전달받으므로 키가 bus를 이동할 필요성이 사라진다. 따라서 결과적으로 키가 bus를 이동하지 않으므로 버스 프루빙에 의해 키가 노출될 위험성이 사라지게 된다.

결론

앞으로 컴퓨팅 능력과 네트워킹 능력을 갖춘 정보 통신 기기는 더욱 다양화되고 그 수는 폭발적으로 증가할 것으로 예상된다. 이러한 성장을 안정적으로 이어나가기 위해서는 안전한 보안 기술의 뒷

받침되어야 하며 높은 보안성을 제공하는 Secure SoC가 주요 보안 방법이 될 것으로 기대된다. 그러나 Secure SoC도 위협할 수 있는 부채널 공격, 메모리 공격, 역공학에 의한 버스 프루빙 공격 등의 강도 높은 물리적 공격들이 존재한다. 앞서 설명한 대응 방안을 통해 이러한 취약점들이 해결된다면 Secure SoC의 보안성은 더욱 증대될 수 있을 것이며 그에 따라 정보 통신 기기의 안전성도 크게 향상될 것으로 기대된다.



김 동 규 교수
소속 : 한양대학교 융합전자공학부
연구분야 : 보안 SoC 설계, 스마트카드 보안, 전자결제시스템 보안, 제어시스템 기능안전 및 보안, 자동차용 보안시스템
E-mail : DOKIM@hanyang.ac.kr
Homepage : http://esslab.hanyang.ac.kr

Reference

- [1] P. Kocher, J. Jaffe and B. Jun, "Differential Power Analysis," CRYPTO '99, LNCS 1666, pp. 388-397, Springer-Verlag, 1999.
- [2] Kamal, A.A, Youssef, A.M, "An-Area-optimized Implementation for AES with Hybrid Countermeasure against Power Analysis," ISSCS 2009, pp. 1-4, 2009.
- [3] M. Joye, P. Paillier, and B. Schoenmaker, "On Second-Order Differential Power Analysis," CHES 2005, LNCS 3659, pp.293-308, 2005.
- [4] A. Hevia, M. Kiwi, "Strength of two data encryption standard implementations under timing attacks," ACM Trans, on Information and System Security, Vol. 2, pp.416-437, 1999.
- [5] D. D. Hwang, K. Tiri, A. Hodjat, B.-C. Lai, S. Yang, P. Schumont, I. Verbauwhede, "AES-Based Security Coprocessor IC in 0.18- μ m CMOS With Resistance to Differential Power Analysis Side-Channel Attacks," IEEE JOURNAL OF SOLID-STATE CIRCUITS, VOL. 41, NO. 4, pp. 781-791, 2006.
- [6] K. Lofstrom, W.R. Daasch, and D. Taylor, "IC identification circuit using device mismatch," Solid-State Circuits Conference, 2000. Digest of Technical Papers, ISSCC, 2000 IEEE International, 2000.
- [7] D. Lim, J.W. Lee, B. Gassend, G.E. Suh, M. Van Dijk, and S. Devadas, "Extracting secret keys from integrated circuits," Very Large Scale Integration (VLSI) Systems, IEEE Transactions on, VOL. 13, 2005.
- [8] J. Guajardo, S.S. Kumar, G.-J. Schrijen, and P. Tuyls, "FPGA intrinsic PUFs and their use for IP protection," Cryptographic Hardware and Embedded Systems-CHES 2007, VOL. 4727, 2007.
- [9] D. Roy, J. H. Klootwijk, N. Verhaegh, H. Roosen, and R. Wolters, "Comb capacitor structures for on-chip physical uncloneable function," Semiconductor Manufacturing, IEEE Transactions on, VOL. 22, 2009.
- [10] C. Tarnovsky, "Deconstructing a 'Secure' Processor", Black Hat, Washington DC., www.blackhat.com

2014 IDEC MPW 설계공모전

Multi Project Wafer 설계공모전

국내 대학(원)생의 SoC 설계 아이디어를 국내외 Foundry를 통해 구현해 볼 수 있는 기회를 드립니다

참여대상
IDEC Working Group(WG)대학의 학부생 및 대학원생

2014년 MPW 공정 지원 내역

- * 10개 공정 26회 진행
- * 아래 지원 내역은 공정사의 사정에 따라 변경될 수 있음.

회사	공정(μm)	공정내역	size	모집 횟수	모집수(1회)	Package
삼성	65nm	CMOSRF 1-poly 8-metal(RF)	4mmx4mm	3	48chips	208pin QFP
매그나칩/ SK하이닉스	0.35μm	CMOS 2-poly 4-metal (Optional layer [DNW/HR/BJT,CPOL] 추가)	5mmx4mm	2	20chips	Design-144pin, Package 제작 - 208pin QFP
	0.18μm	CMOS 1-poly 6-metal (6metal을 Thick metal [TKM]로만 사용가능) (Optional layer [DNW/HR/BJT,MM] 추가)	4.5mmx4mm	4	20chips	Design-200pin, Package 제작 - 208pin QFP
동부하이텍	0.11μm	CMOS 1-poly 6-metal [Top : UTM]	5mmx5mm	2	12chips	208pin QFP
	0.18μm BCDMOS	CMOS 1-poly 4-metal TM	5mmx5mm	4	2chips	지원하지 않음.
	0.35μm BCDMOS	CMOS 2-poly 4-metal TM	5mmx5mm	4	3chips	지원하지 않음.
TowerJazz	0.18μm CIS	CMOS 1-poly 4-metal	5mmx5mm	2	1chips	
	0.18μm CA18HA	RFCMOS 1-poly 6-metal	5mmx5mm	2	1chips	지원하지 않음.
	0.18μm BCDMOS	CMOS 1-poly 3-metal(MT)	5mmx5mm	6	3chips	
	0.18μm SiGe	SiGeBiCMOS 1-poly 6-metal	5mmx5mm	1	1chips	

* TowerJazz 0.18μm CA18HA는 (주)RFCMOS 공정과 특성이 동일한 공정임.

2014년 MPW 진행 일정

- * 회차 표기 방법 변경 : "공정코드-년도모집순서" (예시)삼성65nm 2014년 1회차:S65-1401)
- * 아래 일정은 공정사의 사정에 따라 변경될 수 있음.

공정사	회차구분 (공정_년도순서)	우선모집 신청마감	정규모집 신청마감	후기 접수시작 (선착순마감)	제작 칩 수	DB 마감 (Tape-out)	DB 전달 (Fab-in/Die-out)	Die-out
삼성 65nm	S65-1401	-	2013.12.09	-	48	2014.02.17	2014.03.15	2014.08.15
	S65-1402	2013.12.09	2014.02.03	2014.05.05	48	2014.08.25	2014.09.29	2015.02.27
	S65-1403	2014.03.03	2014.06.02	2014.09.01	48	2014.12.15	2015.01.12	2015.06.12
매그나칩/ SK하이닉스 0.18μm	MS18-1401	-	2013.12.09	-	20	2014.02.24	2014.03.10	2014.07.28
	MS18-1402	-	2014.01.06	2014.02.03	20	2014.05.19	2014.06.02	2014.10.20
	MS18-1403	2013.12.09	2014.02.03	2014.04.07	20	2014.08.11	2014.08.25	2015.01.12
매그나칩/ SK하이닉스 0.35μm	MS35-1401	-	2014.01.06	2014.03.03	20	2014.06.16	2014.06.30	2014.10.06
	MS35-1402	2014.03.03	2014.06.02	2014.09.01	20	2014.12.01	2014.12.15	2015.03.23
	D11-1401	-	2013.12.09	-	12	2014.04.02	2014.04.23	2014.08.06
동부 0.11μm	D11-1402	2014.02.03	2014.04.07	2014.07.07	12	2014.10.01	2014.10.22	2015.02.04
	D18-1401	-	2013.12.09	-	2	2014.02.05	2014.02.19	2014.05.14
	D18-1402	-	2013.12.09	-	2	2014.04.30	2014.05.14	2014.08.06
동부 0.18μm BCD	D18-1403	2013.12.09	2014.02.03	2014.05.05	2	2014.08.13	2014.08.27	2014.11.19
	D18-1404	2014.02.03	2014.05.05	2014.08.04	2	2014.11.05	2014.11.19	2015.02.11
	D35-1401	-	2013.12.09	-	3	2014.02.19	2014.03.05	2014.05.28
동부 0.35μm BCD	D35-1402	-	2014.01.06	2014.03.03	3	2014.05.28	2014.06.11	2014.09.03
	D35-1403	2013.12.09	2014.03.03	2014.06.02	3	2014.09.10	2014.09.24	2014.12.17
	D35-1404	2014.03.03	2014.05.05	2014.08.04	3	2014.11.19	2014.12.03	2015.02.25
TowerJazz 0.18μm BCD	TJB18-1401	-	2014.01.06	2014.02.03	3	2014.05.12	2014.05.19	2014.09.08
	TJB18-1402	2014.02.03	2014.04.07	2014.07.07	3	2014.10.20	2014.10.27	2015.02.16
TowerJazz 0.18μm CIS	TJC18-1401	-	2014.01.06	2014.02.03	1	2014.05.05	2014.05.12	2014.09.01
	TJC18-1402	2014.02.03	2014.04.07	2014.07.07	1	2014.10.13	2014.10.20	2015.02.09
TowerJazz 0.18μm CA18HA	TJR18-1401	-	2014.01.06	2014.02.03	1	2014.05.12	2014.05.19	2014.09.08
	TJR18-1402	2014.02.03	2014.04.07	2014.07.07	1	2014.10.20	2014.10.27	2015.02.16
TowerJazz 0.18μm SiGe	TJS18-1401	-	2013.12.09	-	1	2014.03.11	2014.03.18	2014.07.08

* Package 제작은 Die out 이후 1개월 소요됨.

* 동부와 TowerJazz 공정은 sub chip(5mmx2.5mm 또는 2.35mmx2.35mm)으로 분리하여 모집함.

* 선정 결과는 모집마감후 15일내 개별 통보됨.

* 문의처 : ysllee@idec.or.kr [홈페이지 http://www.idec.or.kr]

안드로이드의 소개와 개발 시작하기



1. 안드로이드란 무엇인가?

'안드로이드(Android)' 라는 단어 자체는 모습과 행동이 인간을 닮은 인간형의 로봇을 의미합니다. 로봇을 의미하는 이 안드로이드라는 개념이 현 구글 부사장인 앤디 루빈에 의해 스마트폰의 OS(Operating System, 운영체제)가 됩니다. 로봇 광이던 루빈은 2003년 '안드로이드' 라는 회사를 설립하고 최초의 오픈 소스형 휴대폰 운영체제를 개발하지만, 월세를 낼 돈이 모자라 친구에게 돈을 빌려야 하는 형편이었다고 합니다. 그러던 와중에 당시 세계 최고 수준의 휴대폰 판매량을 기록하던 한국의 삼성을 찾아왔지만 거절당하고 결국 구글의 창업자 중 한 명인 래리 페이지를 만나면서 구글이 수천만 달러에 안드로이드를 인수한 이야기는 유명합니다. 루빈이 OS를 개발하자마자 휴대폰 제조사

를 찾아간 까닭은 이렇습니다. 2005년, 구글이 앤디 루빈의 안드로이드사를 인수하던 당시만 해도 스마트폰 OS 시장은 저 유명한 마이크로소프트(MS)가 주름잡고 있었습니다. 당시 마이크로소프트는 OS를 휴대폰 제조사에 제공하며 비용을 물렸는데, 이런 방식과 달리 루빈은 안드로이드 OS 코드를 아예 제조사에 공개한 뒤 광고로서 다른 기업들의 투자를 얻는 방식을 고안했습니다. 구글은 루빈의 설계대로 OS와 콘텐츠만 제공하고, 스마트폰은 휴대폰 제조사가 만들며 여기에 통신사업자가 결합하는 방식의 소위 '안드로이드 연합군' 을 탄생시켰고, 그 구상이 현실화되던 2007년 6월, 또 다른 강적 애플사가 '아이폰' 이란 이름의 스마트폰을 출시하자 구글은 대만의 HTC와 손잡고 최초의 안드로이드폰 G1을 내놓게 됩니다. 이 과정에서 구글이 HTC 이전에 먼저 손을 내민 휴대폰 제조사가 LG였다는 이야기가 전해지면서 왠지 아쉽다는 느낌으로 다가오는 안드로이드에 관련된 이야기였습니다.

1 안드로이드(버전 1.0~1.2)



첫 안드로이드는 이름이 없이 1.0이었 습니다. 두 번째는 프티 푸르(Petit Four)였고, 버전은 1.1/1.2였습니다.

2 컵케익(버전 1.5)



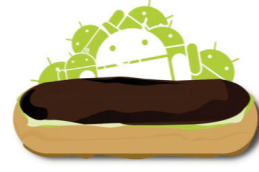
세 번째부터 많은 변화가 일어났으며, 한국어 지원이 시작되었습니다. 코드명은 컵케익(Cupcake)이며 안드로이드 버전 1.5입니다. 홈스크린과 음성인식 기능 지원이 시작되었습니다. 1.5부터가 정식버전입니다.

3 도넛(버전 1.6)



네 번째의 코드명은 도넛(Donut)이며, 버전 1.6입니다. 이때부터 WVGA 해상도가 지원되기 시작했으며 속도가 개선되었습니다. 멀티 터치 기능이 추가되었습니다.

4 이클레어(버전 2.0, 2.1)



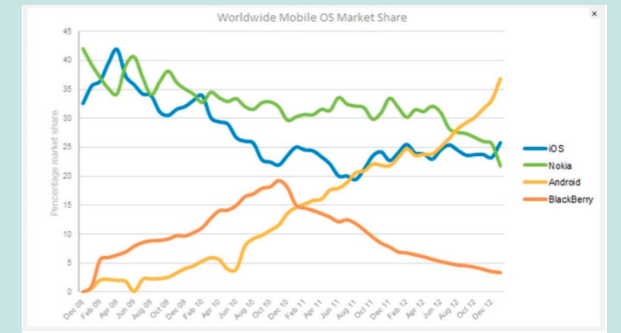
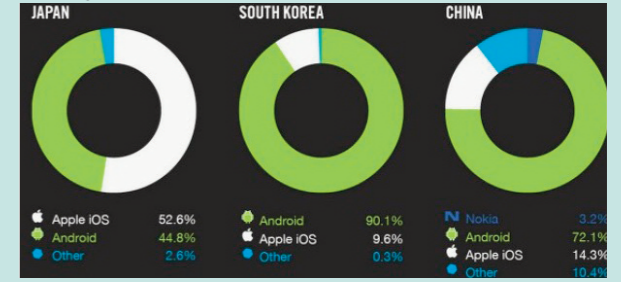
다섯 번째는 이클레어(Eclair, 초콜릿을 얹은 길쭉한 빵)입니다. 버전 2.0, 2.1입니다. WXGA(16:9) 이상의 해상도가 지원되고, UI 디자인이 변경되었습니다.

2. 안드로이드의 점유율과 변천사

안드로이드 OS가 탑재된 스마트폰을 흔히 '안드로이드폰' 이라고 하고, 그 폰에서 사용되는 프로그램을 '앱(APP 혹은 APK, 응용 프로그램을 의미하는 Application의 약자)' 이라고 합니다. 안드로이드 앱들은 '안드로이드 마켓' 이라는 곳에서 무료 혹은 유료로 구매하여 설치할 수 있습니다. 이런 앱을 프로그램 명하면서 \$25를 지불하고 구글 개발자로 등록하면 누구나 프로그램을 만들어서 안드로이드 마켓에 등록할 수 있습니다. 전 세계의 스마트폰은 애플의 iOS폰과 구글의 Android OS를 사용한 Android폰, 이렇게 양진영으로 나뉘어 있습니다. 모바일 앱을 개발할 때는 먼저 두 가지 OS를 사용하는 폰을 모두 지원할 것인가, 아니면 어느 한쪽만을 목표로 개발할 것인가를 선택해야 합니다. 참고로, 2013년 2분기 가트너의 자료에 따르면 전세계 스마트 시장에서 Android의 점유율은 64.2%에 이릅니다. 그리고 국내의 Android 점유율은 90.1%에 육박합니다. 전 세계 사용자를 상대로 하는 앱이라면 iOS, Android에서 모두 플레이 가능한 모바일 웹 또는 하이브리드 앱으로 개발하거나 양쪽 OS 버전을 모두 개발해야 합니다. 하지만 국내 전용 앱이라면 9.6%인 iOS 사용자를 위해 멀티 플랫폼용 앱을 개발하는 것은 상당히 부담스러울 것입니다. 2013년 2월에 iCrossing에서 발표한 OS 점유율에 따르면 Android OS의 사용자 수는 급격하게 증가하고 있습니다. 이런 수치로 살펴볼 때 우리가 안드로이드로 앱을 개발해야 하는 이유가 조금은 설명이 되었을 것 같습니다. 이제 이 안드로이드 OS의 버전을 살펴보겠습니다. 구글은 안드로이드 OS를 업데이트할 때마다 디저트 음식의 이름을 알파벳 순서로 붙여왔습니다.

스마트폰 OS	2Q13 판매량	2Q13 시장점유율 (%)	2Q12 판매량	2Q12 시장점유율 (%)
안드로이드	177,898.2	79.0	98,664.0	64.2
iOS	31,899.7	14.2	28,935.0	18.8
마이크로소프트	7,407.6	3.3	4,039.1	2.6
블랙베리	6,180.0	2.7	7,991.2	5.2
바다	838.2	0.4	4,208.8	2.7
심비안	630.8	0.3	9,071.5	5.9
기타	471.7	0.2	863.3	0.6
합계	225,326.2	100.0	153,772.9	100.0

Trend of technology | 10



5 프로요(버전 2.2)



여섯 번째는 프로요(Froyo, Frozen yogurt, 얼린 요구르트)입니다. 안드로이드 버전 2.2이며 용량이 큰 앱을 외장메모리(SD-Card)에 옮기는 기능이 추가되었습니다. 어도비 플래시 10.1을 지원하기 시작했습니다.

6 진저브레드(버전 2.3)



일곱 번째는 진저브레드(Gingerbread, 생강빵)라는 이름의 버전 2.3입니다. SIP Voip(인터넷통화) 지원, webM/VP8 비디오 플레이, 다양한 오디오 효과를 지원하게 되었습니다.

7 허니콤(버전 3.0)



태블릿 전용인 허니콤(Honeycomb)입니다.

8 아이스크림 샌드위치(버전 4.0)



여덟 번째가 아이스크림 샌드위치(Ice Cream Sandwich, ICS)라는 이름의 버전 4.0입니다. 다양한 디바이스에 대응하여 폰용, 태블릿용(허니콤)을 따로 지원하던 것과 달리 4.0에서는 폰과 태블릿에서 공통적으로 사용 가능합니다. 얼굴인식 기능(Face unlock), 안드로이드빔(휴대폰을 서로 맞대어 정보교환)이 가능합니다.

9 젤리빈(버전 4.1~4.3)



아홉 번째는 젤리빈(Jelly Bean)라는 이름의 버전 4.1~4.3입니다. OpenGL Es 3.0으로 그래픽이 향상되었고, 한 기기에 다 계정 연결이 가능해졌습니다. 보안이 강화되고 포토 스피어 기능(파노라마를 3D로 가능) 및 무선 프로젝터 연결이 가능합니다.

10 킷Kat(버전 4.4)



열 번째는 네슬레의 초콜릿 바 이름을 딴 킷Kat(KitKat)이라는 이름의 버전 4.4입니다. 원래는 5.0 키라임파이로 예정되었는데 이름이 낯설어서 킷Kat으로 변경되었습니다.

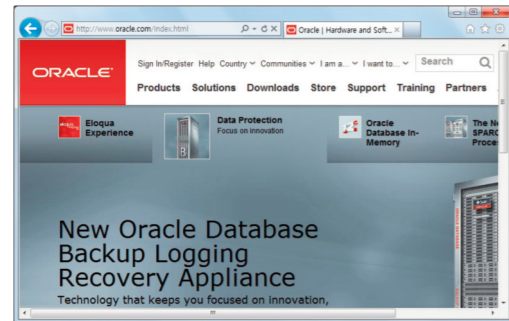
3. 안드로이드 개발 환경 설정하기

안드로이드 개발 환경인 개발툴 이클립스와 안드로이드 개발용 SDK인 ADK(Android Development Kit)를 셋팅해 보겠습니다. 그리고 기본 샘플 프로젝트를 만들어 보겠습니다.

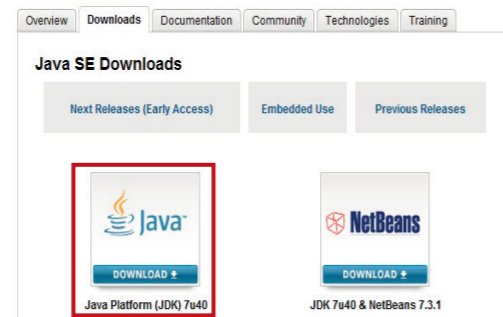
3.1. 자바 SDK 설치

안드로이드는 자바로 만들어진 운영체제다 보니, 안드로이드 앱을 만들기 위해서는 자바 SDK인 JDK가 기본적으로 설치되어 있어야 합니다. 스크린샷을 저장한 당시(2013년 10월 7일) JDK는 버전 7u40까지 출시되어 있었습니다. JDK의 버전업이 자주 이루어지므로 꼭 해당 버전이 아니어도 됩니다. 최신 버전을 동일한 방법으로 받아 주세요.

1. 오라클사의 홈페이지에 접속합니다. (<http://oracle.com>)



3. 다음과 같은 화면이 나타나면, [JDK 배너(Java)]를 클릭하여 버전 선택화면으로 넘어갑니다.



2. 상단 메뉴 중 [Downloads]를 클릭하여 들어간 뒤, 다음 그림과 같이 Java 메뉴에서 [Java SE]를 클릭합니다.



4. 우선 사용권 계약에 "동의(Accept)" 한 뒤 자신의 OS에 맞는 버전으로 선택하여 다운로드 합니다.

Java SE Development Kit 7u40 Demos and Samples Downloads			
Java SE Development Kit 7u40 Demos and Samples Downloads are released under the Oracle BSD License			
Product / File Description	File Size		Download
Linux ARM v5A7 VFP Hard Float ABI	9.89 MB	jdk-7u40-linux-arm-vfp-hft-demos.tar.gz	
Linux ARM v5A7 VFP Soft Float ABI	9.73 MB	jdk-7u40-linux-arm-vfp-sft-demos.tar.gz	
Linux x86	16.6 MB	jdk-7u40-linux-i586-demos.rpm	
Linux x86	16.56 MB	jdk-7u40-linux-i586-demos.tar.gz	
Linux x64	16.55 MB	jdk-7u40-linux-x64-demos.rpm	
Linux x64	16.63 MB	jdk-7u40-linux-x64-demos.tar.gz	
Mac OS X	15.3 MB	jdk-7u40-macosx-x86_64-demos.tar.gz	
Solaris x86	22.89 MB	jdk-7u40-solaris-i586-demos.tar.Z	
Solaris x86	16 MB	jdk-7u40-solaris-i586-demos.tar.gz	
Solaris x64	1.23 MB	jdk-7u40-solaris-x64-demos.tar.Z	
Solaris x64	0.82 MB	jdk-7u40-solaris-x64-demos.tar.gz	
Solaris SPARC	22.89 MB	jdk-7u40-solaris-sparc-demos.tar.Z	
Solaris SPARC	16 MB	jdk-7u40-solaris-sparc-demos.tar.gz	
Solaris SPARC 64-bit	1.24 MB	jdk-7u40-solaris-sparcv9-demos.tar.Z	
Solaris SPARC 64-bit	0.89 MB	jdk-7u40-solaris-sparcv9-demos.tar.gz	
Windows x86	17.41 MB	jdk-7u40-windows-i586-demos.zip	
Windows x64	17.49 MB	jdk-7u40-windows-x64-demos.zip	

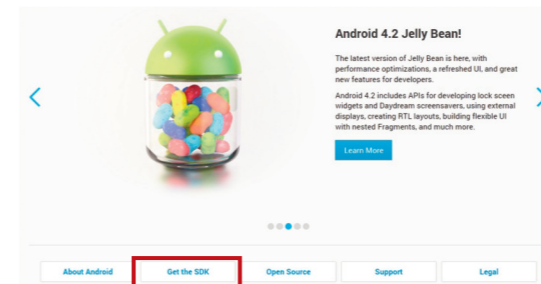
여러분이 사용하는 대부분의 PC는 윈도우(Windows)를 운영체제로 삼고 있을 것입니다. 위 그림에서 표시된 'Windows x86' 이 대부분의 32bit, 'Windows x64' 가 64bit 버전입니다. 본인이 사용 중인 OS에 맞는 버전으로 다운로드 합니다. 이때 다운 받은 파일을 바로 실행할 수도 있고, 적당한 위치에 파일을 받은 뒤 실행할 수도 있습니다. 설치를 하다보면 JDK 작동에 필요한 JRE(Java로 개발한 프로그램의 실행환경)의 설치 여부를 묻는데, 필수 설치항목이므로 디폴트 설치 폴더인 [C:\Program Files\Java\jre7]에 설치하면 됩니다. 이후 진행과정에서 JavaFX의 설치의 필요 없으므로 취소해도 됩니다.

3.2. 안드로이드 ADK 다운로드

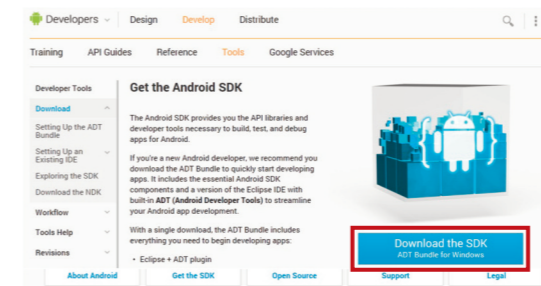
기본에는 이클립스를 받고 구글의 안드로이드 SDK인 ADK를 직접 다운 받아서 개발자가 연결을 해주어야 합니다. 이 과정이 복잡했는데, 최근에는 구글에서 안드로이드 ADK를 받으면 그 안에 안드로이드 개발용 ADK를 함께 포함시켜 배포하고 있어 매우 간단해졌습니다.

ADK를 다운받아서 이클립스를 실행 해보도록 하겠습니다.

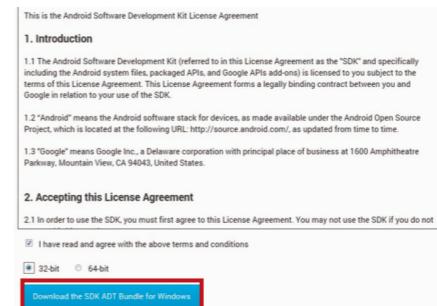
1. '안드로이드 개발자 사이트(<http://developer.android.com>)' 에 접속합니다.



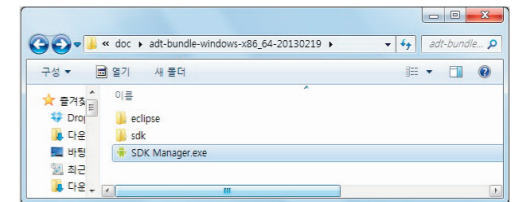
2. 하단 메뉴 중 'Get the SDK' 를 선택하면 SDK 패키지를 다운로드 할 수 있는 페이지가 열립니다.



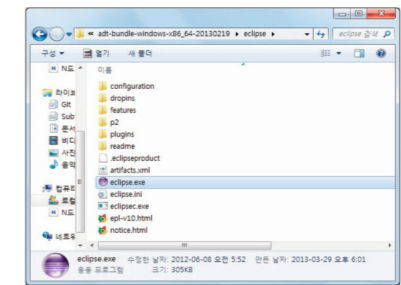
3. [Download the SDK] 버튼을 클릭하면, 구글의 정책에 대한 동의 항목이 나오고, 개발자의 운영체제를 선택하는 란이 있습니다. 정책에 동의하고 운영체제를 선택한 후에 [Download the SDK ADT Bundle for Windows]를 클릭합니다.



4. 압축된 파일을 다운로드한 후, PC의 적당한 장소에 압축을 풉니다. [eclipse] 폴더 안에 개발툴인 이클립스가 있습니다. [sdk] 폴더 안에는 최신버전의 안드로이드 SDK가 포함되어 있습니다.



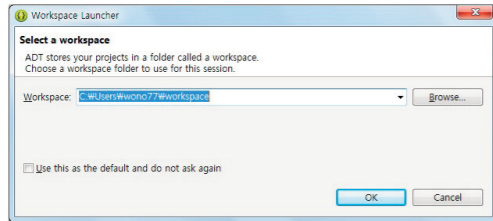
5. [eclipse] 폴더 안의 [eclipse.exe]를 클릭하여 이클립스를 실행합니다.



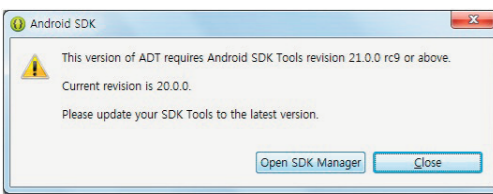
6. 그림과 같은 화면이 나타나면서 이클립스가 로딩을 시작합니다. 이 로딩 이미지를 보면 안드로이드 SDK와 함께 다운로드 받은 이클립스가 구글에서 별도로 Android에 최적화하였다는 것을 알 수 있습니다.



7 프로젝트 소스의 저장 위치를 지정합니다. 대화창 아래의 'Use this as the default and do not ask again' 을 체크한 상태로 [Browse]를 눌러 폴더를 지정한 뒤 [OK]를 누릅니다. 해당 폴더는 앞으로 개발할 프로젝트의 기본 저장 폴더가 됩니다.

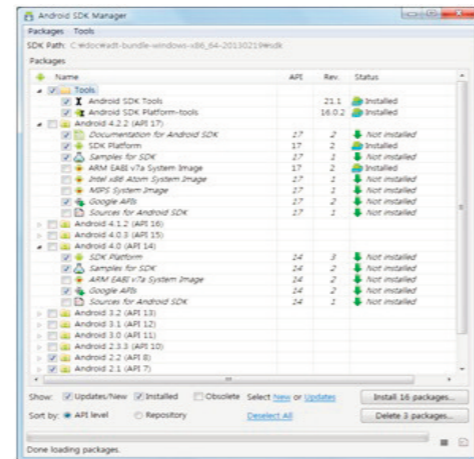


8 다음과 같이 신규 SDK를 설치하라는 메시지가 뜹니다. 이때는 [Open SDK Manager]를 클릭합니다.



9 그림과 같은 SDK Manager 창에서 개발에 사용할 안드로이드 버전과 하위 항목을 선택합니다. 모든 항목을 설치하면 시간이 오래 걸리므로, 안드로이드 4.2.2(젤리빈), 4.0(아이스크림 샌드위치), 2.3.3(진저브레드), 2.2(프로요), 2.1 정도를 체크해서 선택해 보았습니다. 처음 이클립스를 열면 당시 최신버전 ADK만 기본 설치되어 있습니다.

필요한 항목은 이클립스 메뉴에서 Android SDK Manager를 클릭해서 언제든지 추가 가능합니다.



<참고>

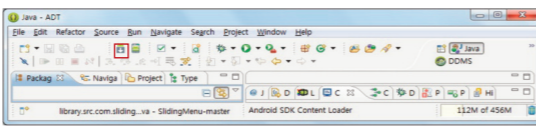
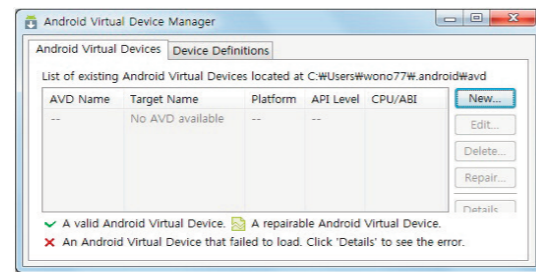


그림. SDK Manager 아이콘

Android SDK Manager : 안드로이드의 설치와 관리를 위한 버튼입니다. <참고>

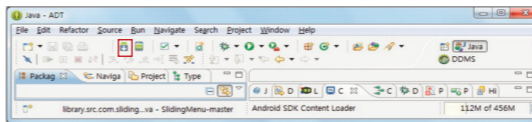
10 선택이 끝나면 그림의 [Install ()packages] 부분을 눌러줍니다. 그러면 마지막으로 선택된 프로그램 라이선스에 동의를 구하는 대화창이 나타납니다. 'Accept All' 을 선택한 후 [Install]을 클릭하면 설치가 진행됩니다.



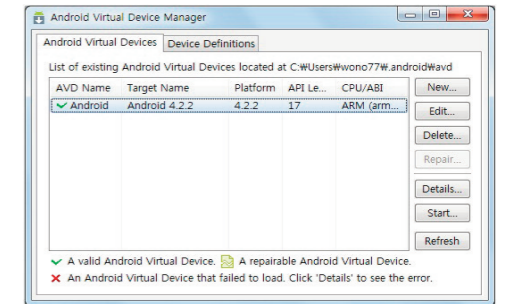
3.3. 에뮬레이터로 안드로이드 폰 화면 구현하기

안드로이드 에뮬레이터는 가상디바이스(AVD : Android Virtual Device)라고도 불리며, 컴퓨터에서 가상의 휴대폰 화면을 구현하는 프로그램입니다. 개발한 앱을 핸드폰에 옮기지 않고 AVD를 통해 앱을 테스트해 볼 수 있습니다. 하지만 가상으로 폰 화면을 구현하다보니 아무래도 실제 폰에 띄워서 개발하는 것보다는 속도가 느리고 폰에서는 되지만 에뮬레이터에서는 안 되는 등 여러가지 제약사항이 많습니다. 그래서 가능하면 실제 안드로이드폰을 PC에 연결해서 개발 화면을 폰에 띄우며 개발하는 방법을 추천 드립니다. 안드로이드 개발에서 안드로이드폰은 필수 준비물입니다. 아직 폰이 준비 안된 분들을 위해 이클립스에서 안드로이드 에뮬레이터를 띄우는 방법을 설명 드리겠습니다.

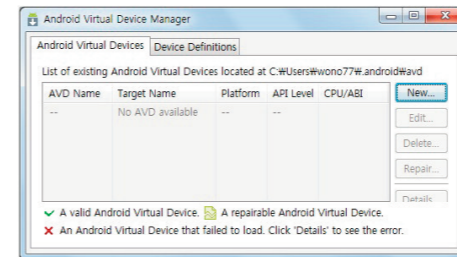
1 이클립스에서 [AVD Manager] 버튼을 클릭합니다.



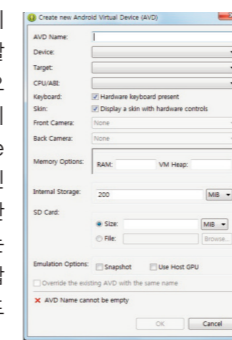
4 [Android Virtual Device Manager] 창으로 돌아오면 리스트에 방금 저장한 AVD 설정을 확인할 수 있습니다.



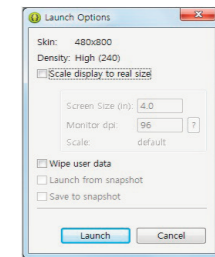
2 [Android Virtual Device] 화면이 열리면, [New] 버튼을 클릭합니다.



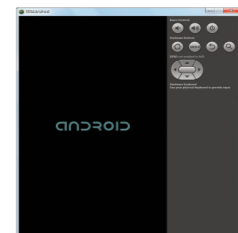
3 그림과 같은 창에서 에뮬레이터의 세부 내용을 설정할 수 있습니다. 'Name' 은 앞으로 사용할 가상 휴대폰의 이름이므로 적당히 지어 주세요. Device 는 핸드폰 기기를 선택하는 곳인데, 화면의 해상도를 따라 다양한 선택이 가능합니다. 이 해상도는 안드로이드의 제조사마다 다양합니다. 현재 보편적인 안드로이드 폰의 해상도는 480x800, 720x1280, 800x1280입니다. 앱을 출시 할때는 최소한 이 3가지 해상도에서 화면이 정상적으로 출시되도록 맞추어야 좋습니다. 480x800을 선택해 주세요. 'Target' 은 안드로이드 SDK의 버전을 의미합니다. 4.2.2 젤리빈을 선택합니다. 설정을 끝낸 후, [OK]를 눌러 저장합니다.



5 우측 하단의 [Start] 버튼을 클릭합니다. [Launch Options] 창에서는 [Launch] 버튼을 클릭합니다.



6 에뮬레이터가 정상적으로 가동되었습니다. 다음 화면이 보인다면 에뮬레이터가 정상적으로 시작된 것입니다.



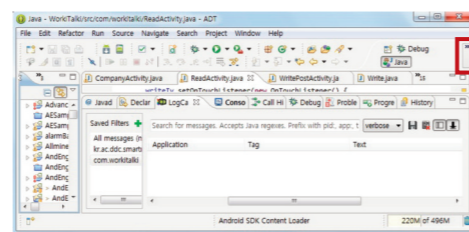
3.4. 이클립스와 휴대폰을 연결하기

안드로이드 폰 없이 안드로이드 앱 개발이 불가능한 것은 아니지만, 앞서 말씀드린대로 에뮬레이터 만으로 개발을 진행하면 로딩 속도가 느리고 여러가지 제약사항이 많으므로 반드시 안드로이드 폰을 연결하여 테스트할 것을 추천 드립니다.

- 1 이클립스 ADT를 연 후, usb선과 휴대폰을 연결한 후, usb를 pc에 연결합니다.
- 2 휴대폰의 드라이버가 PC에서 적용되지 않는다면, 각 휴대폰의 제조사 사이트에서 다운받아 설치 합니다.
- 3 휴대폰에서 [환경설정]으로 들어간 후, [개발자 옵션]에서 디버깅 메뉴의 [USB 디버깅], [모의 위치 허용]을 체크해서 활성화합니다.



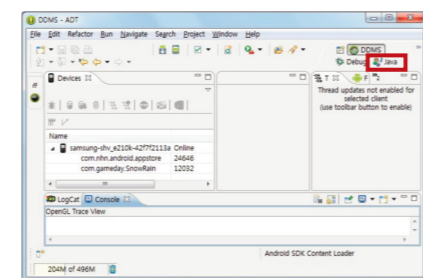
- 4 PC와 휴대폰이 연결되었으므로, 이클립스에서 정상적으로 인식되는지 확인합니다. 연결 확인은 이클립스의 DDMS(Dalvik Debug Monitor Service)에서 가능합니다. 이 메뉴는 이클립스 메뉴의 우측 끝에 존재합니다.



- 5 DDMS를 클릭합니다.



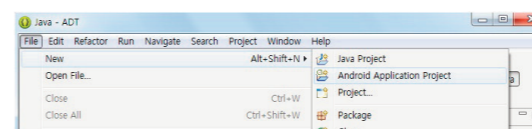
- 6 DDMS 화면에서 이클립스 좌측에 인식된 폰이 표시됩니다. DDMS를 한번 실행하고 난 후에는 이클립스에서 DDMS 메뉴가 활성화됩니다. 다시 개발이 가능한 창으로 돌아가기 위해서는 DDMS 아이콘 아래의 [Java] 버튼을 클릭하면 됩니다.



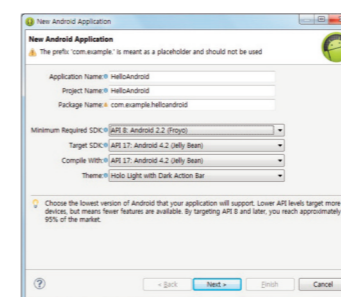
3.5. Hello Android 출력하기

안드로이드의 가장 기본적인 프로젝트를 생성하고 결과물을 에뮬레이터와 핸드폰에 띄워보도록 하겠습니다.

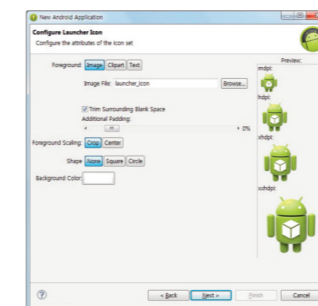
- 1 새로운 프로젝트를 생성하기 위해 이클립스 메뉴의 'File > New > Android Application Project'를 선택합니다. 새로운 프로젝트를 생성하면 안드로이드의 가장 기본적인 골격이 자동으로 생성됩니다.



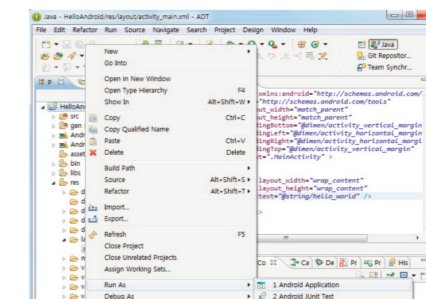
- 2 Application Name과 Project Name을 입력합니다.



- 3 이 후 나타나는 창에는 모두 'Next'를 눌러 설정을 완료합니다. 해상도에 따른 아이콘이러던가 프로젝트의 Activity 설정 등에 대한 것들인데, 프로젝트 생성 완료 후 수정이 언제든지 가능하므로 여기서는 자동 선택에 맡기도록 합니다.



- 4 이클립스의 좌측 [Workspace]란을 보면 방금 생성한 HelloAndroid 프로젝트를 볼 수 있습니다. 해당 프로젝트에서 마우스 오른쪽 버튼 클릭 후 [Run As > Android Application]을 선택합니다.



- 5 실행 결과, 휴대폰 화면과 에뮬레이터 화면에서 다음과 같은 "Hello world"가 출력된 결과를 확인할 수 있습니다.

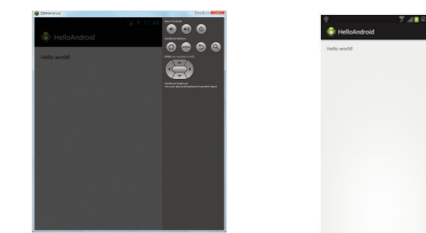


그림. 에뮬레이터에서 실행 핸드폰에서 실행

안드로이드 OS에 대해 살펴보고 간단한 에뮬레이터와 핸드폰에 출력해보았습니다. 모든 개발은 항상 개발 환경을 셋팅하고 Hello World를 출력하는 처음이 가장 어렵습니다. 여기까지만 하면 안드로이드 개발의 절반은 했다고 생각하시면 됩니다.



최원효 과장
㈜네이버 컨텐츠검색팀
E-mail : ytkim@dec.or.kr
http://www.idec.or.kr

디지털 시스템의 MPW 설계 방법에 대한 이해 (3)

Equivalence Check & Static Timing Analysis(ECO)

서론

최근 심각해지는 반도체 업계의 인력난과 어려움에 반하여 대학 및 관련 분야에서는 숙달된 고급 인력의 지속적인 양성이 힘들어 돌 사이에 장벽이 존재하고 있다. 이는 축적된 노하우를 바탕으로 지속적인 칩 제작을 하면서 고급인재를 양성하는 대학 및 연구실이 많지 않다는 것을 뜻하며, 칩을 만들고자 하는 대학은 HOW-TO 문서를 그때그때 검색하고 짧은 기간에 임기응변식의 칩 제작을 한다는 뜻이기도 하다. 실제 MPW 참

여자들의 경우 정기적으로 참여하는 팀과는 대조적으로 MPW 참여와 동시에 구글링부터 시작하는 사람들이 많다. 칩 설계 고급 인력 양성을 위해서는 시간과 노력이 많이 들어가야 하므로 직관적이고 정확한 이론 및 실습자료가 매우 필요하다. 지금까지 “SoC 시스템과 IDEC MPW Flow 소개”와 “FPGA 검증부터 Synopsys Design Compiler를 이용한 합성”에 대해 알아보았다. 이번 호에서는 “Equivalence Check와 Static Timing Analysis”에 대해 소개 하고자 한다.

본론

◆ Equivalence Check

Equivalence Check를 진행하는 이유는 logic synthesis와 layout 이후 function 변화가 없는지 점검하기 위함이다. Synthesis 과정에서 틀은 constraint 및 compile options 등에 따라 optimization을 위해 특정 모듈을 삭제하거나 timing을 위해 buffer 등을 삽입한다. 사용되지 않는 register 또는 unconnected net의 경우 compile 이후 삭제될 수 있으며 이러한 과정은 설계 면적을 줄이는 좋은 방법이 되기도 한다. 이를 통해 설계자는 register 나 net에 대해서 관련 모듈을 점검할 수 있으며 설계에 재 반영하여 진행할 수도 있다. 최근에는 IP의 재사용이 늘어나면서 외부 모듈을 활용하는 경우가 많아졌고 design이 커지면서 module을 여러 설계자가 나누어 설계하는 경우도 늘고 있다. 이러한 경우 design에 대해 충분히 고려하지 않고 synthesis를 진행할 경우 의도하지 않은 결과가 나타날 수 있으므로 주의해야 한다. 설계자는 synthesis 이후 function simulation을 진행하며 function에 문제가 없는지 점검한다. 하지만 function simulation은 test bench에 의존적이며 다양한 test bench를 통해 검증하고자 할 경우 runtime이 증가하게 되는 단점이 있다.

이러한 부분을 개선하고자 synthesis 이후 equivalence check 과정을 진행하며 대표적으로 사용되는 툴은 Synopsys 사의 Formality 툴 이다. Formality는 logic의 0과 1의 input에 따른 output 값만을 비교하며 딜레이는 고려하지 않는다. 예를 들어 synthesis 이후 단일 net 에 inverter 2개가 연속하여 삽입되었을 경우 Formality는 동일한 function으로 인식하고 error report를 하지 않는다.

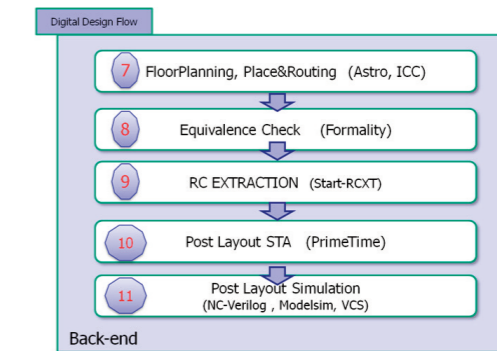


그림2. IDEC MPW FLOW (Digital Back-End)

Formality에 대한 자세한 기능과 사용 방법은 간단한 예제를 통해 확인해 보도록 하자.

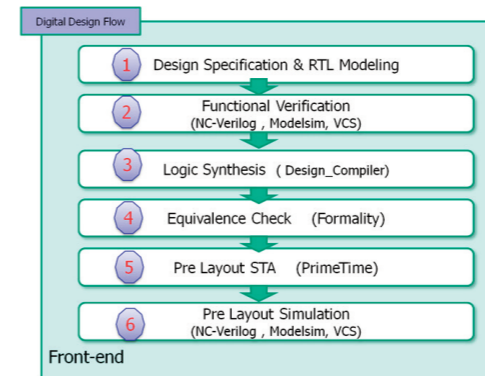


그림1. IDEC MPW FLOW (Digital Front-End)





◆ 환경설정

Formality는 다른 툴과 마찬가지로 command mode와 GUI mode에서 모두 실행이 가능하다. command mode에서 실행할 경우 fm_shell로 실행하며 GUI mode로 실행 시 formality 명령어로 실행할 수 있다. GUI 구성이 비교적 잘 구성되어 있으므로 처음 사용 시 GUI로 진행하는 것이 좋다. 단계별로 0번부터 6번까지 번호로 표기되어 있으며 사용자는 순서대로 진행하면 결과를 확인할 수 있다. 이상이 없으면 녹색으로 표기되며 다음 단계의 메뉴가 활성화된다.

환경설정을 위해 designware 등에 대한 경로 설정과 synthesis 과정에서 생성된 svf 파일이 필요하다.

- set_app_var hdlin_dwroot SYNOPSIS/design_compiler
- set_svf design_compiler.svf

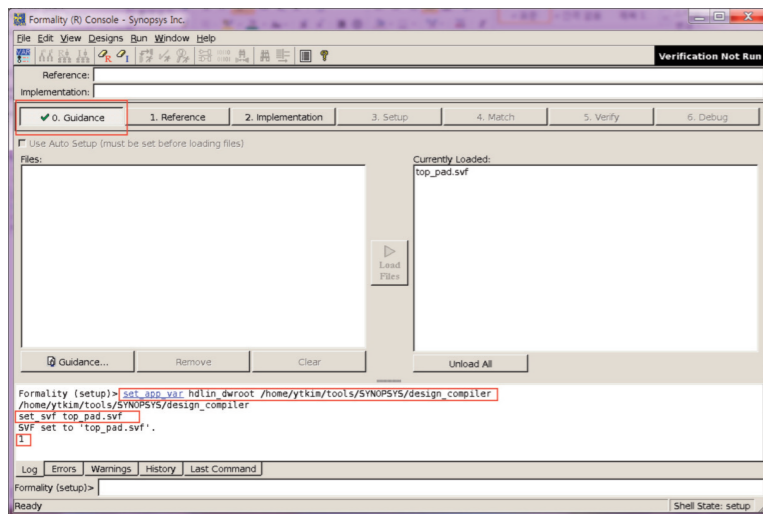


그림3. Formality GUI (환경설정)

◆ Reference

Synthesis 이후 점검 시 reference는 synthesis 이전 RTL 이 되며 layout 후 비교할 경우에는 synthesized netlist가 reference 가 된다. verilog를 load 한 이후에는 database Library 설정과 Top Design을 지정해 준다.

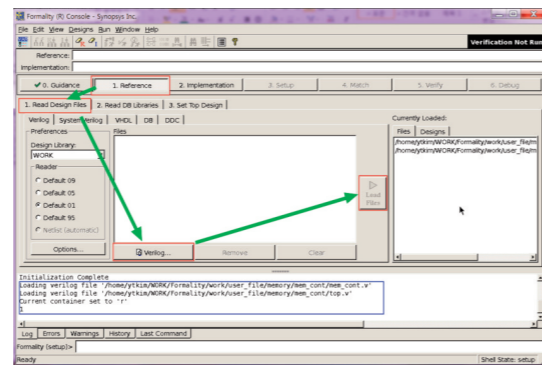


그림4. Reference Design Setup

- read_verilog container r library WORK top.v
- read_db -technology_library slow.db
- set_top r:WORK/top

◆ Implementation

Synthesis 된 netlist 파일을 load하고 Top Design 선언 후 다음 단계로 넘어간다. DB는 reference 과정에서 이미 load 하였으므로 read_db 과정은 생략해도 무방하다.

- read_verilog container i libname syned_top.v
- set_top i:WORK/top

◆ Setup

Setup 단계에서 설정해 주는 것은 scan en과 같은 constant 값이 있는 경우나 CTS 등의 과정이 후 net name이 변경된 경우 설정을 한다. scan을 사용하지 않았거나 단순히 synthesis 이후 진행을 할 경우 별도의 설정을 하지 않아도 된다.

- set_constant type port i:WORK/scan_en 0
- set_compare_rule r:/WORK/top -from {reg\[([0-9][0-9]*\)]} -to {reg_1}

◆ Match

위 과정까지 이상 없이 진행한 뒤 match를 진행하면 reference와 implementation design과 상이한 점을 report 한다.

- match

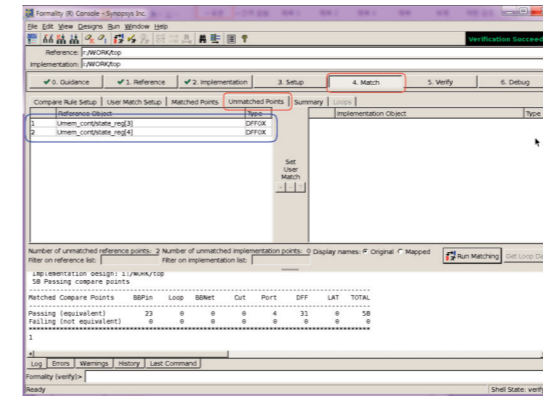


그림5. Results of Match

report 된 내용을 보면 register 두 개가 일치하지 않은 것을 보여주고 있으며 synthesis 이후 implementation design에서 state_reg[3] 과 state_reg[4] 의 두 개의 register가 삭제된 것을 확인할 수 있다. 설계자는 삭제된 register가 동작에 이상이 없는 부분인지 점검해야 하며 필요에 따라 RTL을 수정하거나 DesignCompiler 에서 constraint를 변경하여 다시 synthesis를 진행해야 한다.

report 된 곳의 source code를 살펴보면 state 상태를 저장하기 위한 register를 5개 생성하였으나 state는 6개만 사용하였으므로 3개의 register로만 처리가 가능한 상황이었음을 알 수 있다. DesignCompiler는 2개의 register를 삭제함으로써 면적에 대한 optimization을 진행한 것을 알 수 있다. 설계자는 reg [2:0] state: 로 RTL을 수정하여 다시 synthesis를 진행하거나 동작에 문제가 없는 구간이므로 SKIP 하여 다음 단계로 진행할 수 있다.

```
reg [4:0] state;
parameter IDLE = 4'h0;
parameter READ_DATA = 4'h1;
parameter RECEIVE_DATA = 4'h2;
parameter WRITE_RAM = 4'h3;
parameter ADDRESS_PLUS = 4'h4;
parameter READ_RAM = 4'h5;
parameter READ_RAM_SETADD = 4'h6;
```

위와 같이 Synthesis 과정에서 register 등이 삭제되었을 경우 Formality가 아닌 Design Compiler에서도 확인이 가능하다. synthesis 이후 report를 보면 삭제된 register 및 unconnected net에 대해 설계자가 확인할 수 있게 되어있다. Formality를 통해 Equivalence Check를 하는 이유는 아래 사례 같은 경우에 보다 유용하다.

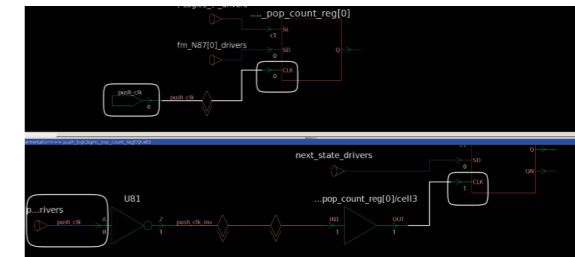


그림6. Unmatched ports

그림에서 상단부분이 reference design(RTL)이며 하단부분은 implementation design(Synthesized netlist)이다. push_clk의 값을 보면 reference design에서는 input 값인 0이 count_reg[0]에 그대로 인가되고 있으나 implementation design에서는 1로 값이 변경되어 인가되고 있음을 알 수 있다. 위 경우는 CTS 또는 ECO 과정 진행에서 timing violation 문제를 해결하기 위해 inverter가 삽입된 것이며, timing 문제는 해결되었으나 설계자가 의도하지 않은 동작을 하게 된다. function simulation을 통해 이상 여부가 확인될 수도 있으나 test bench에 의존적이므로 복잡한 logic일 경우 놓치고 지나칠 수 있으며, 문제가 되는 구간과 원인을 찾는 데 많은 시간이 필요하다. 그러므로 Equivalence Check를 점검하고 진행하는 것이 오히려 검증시간을 줄이는 좋은 방법이 된다.

최근 본 센터에서 지원하는 MPW Chip 제작 type을 보면 Mixed Design이 증가하고 있으며 이에 따라 다양한 DIGITAL IP가 활용되고 있다. Equivalence Check Tool은 대부분 공정에서 사용되기를 권장하고 있으며 사용법이 쉽고 간편하다는 장점이 있으므로 많은 설계자에게 도움이 될 것으로 생각된다.

◆ Static Timing Analysis (STA)

STA는 Function Simulation과 Equivalence Check와는 달리 timing에 대한 이상 여부만을 판별한다. 예를 들어 3+5=8을 연산하는 덧셈기에서 3+5=9 연산이 되더라도 timing 대해서만 분석할 뿐 logic의 true/false 여부에 대해서는 고려하지 않는다. 대표적인 툴은 Synopsys사의 PrimeTime이며 대부분 공정에서 sign off 툴로 사용되고 있으므로 Digital 설계자는 필수적으로 사용하게 되는 툴이다. PrimeTime은 크게 세 가지 Feature로 구분되며 기본 STA를 진행하는 PrimeTime, Crosstalk Analysis 기능이 포함된 PrimeTime-SI, Power Analysis가 가능한 PrimeTime-PX가 있다. 설치를 개별적으로 진행하는 것이 아니라 해당 기능을 사용할 수 있는 License를 보유한 상태에서 특정 명령어를 통해 기능을 활성화하는 형태로 동작한다. IDEC에서 제공하는 License는 SI와 PX까지 모두 사용할 수 있다.

STA에서 가장 중요한 개념은 setup과 hold time에 대한 것으로 clock을 기준으로 데이터가 원하는 시점에 유효한 값을 인식할 수 있는지를 판별한다. rising edge에서 data 값을 인식하기 위해서는 data는 clock rising edge보다 이전에 유효한 값을 일정 시간 동안 가지고 있어야 하며 (setup

time) 반대로 switching 후 상태 변화가 정확히 인식되도록 하기 위해서는 일정 시간 동안 값을 유지하고 있어야 한다(hold time). STA는 기본적으로 clock을 기준으로 delay를 점검하므로 로직에 clock이 없을 경우, virtual clock을 별도로 선언하여 timing 분석을 진행하거나 다른 방법으로 접근해야 한다. 이외에도 skew 및 latency에 대한 개념 등 timing에 대한 기본적인 사항을 알고 있어야 원활한 사용이 가능하다.

PrimeTime에 대한 사용방법을 모두 소개하는 것은 어려우므로 자세한 사용법은 IDEC에서 제공하는 교육 강좌를 수강하는 것을 추천한다. 본 호에서는 IDEC MPW 공정 진행 시 설계자들에게 꼭 필요한 개념과 많은 질문이 있었던 ECO 방법에 대해서 간략하게 소개하고자 한다. PrimeTime은 DesignCompiler 통해 synthesis가 완료된 시점과 Astro 및 ICC를 통해 layout을 진행한 이후 두 번에 걸쳐 진행하게 된다. Synthesis 이후 STA 진행 시 설계자는 setup time에 대해서만 violation이 없는지 점검한다. hold violation에 대해서 report 되더라도 무시하고 진행한다. hold Violation을 무시하는 이유는 DesignCompiler에서 clock에 대해 ideal하게 선언을 하고 실제 layout이 진행되지 않은 상태이기 때문이다.

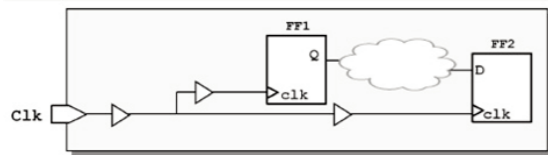
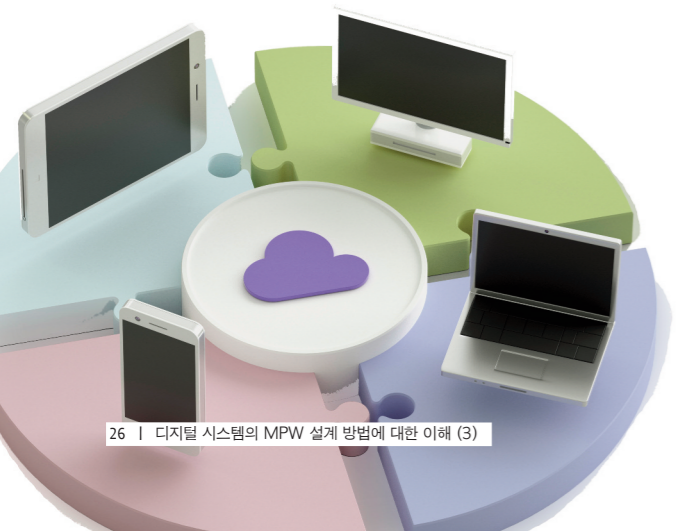


그림6. Clock 에 Buffer 가 삽입된 회로

예를 들어 위와 같이 두 개의 Flip Flop이 있으면 clock net의 길이가 다르므로 FF1에 clock이 인가되는 시점과 FF2에 clock이 인가되는 시점이 다르게 된다. Layout을 진행하기 이전에는 얼마큼의 시간이 필요인지 정확히 알 수 없으므로 그림 7에서와 같이 버퍼를 미리 삽입했다라도 정확한 값이 아니다. Synthesis 진행 시 clock을 ideal network로 선언하고 모든 clock이 동시에 active 되는 것으로 고려하며, Layout CTS 과정에서 기존에 적용된 clock에 대한 모든 buffer를 삭제하므로 Layout 이전에 hold violation을 점검하는 것은 의미가 없다. 그러므로 synthesis 이후에는 setup violation만을 점검하고 layout 이후 setup과 hold time violation이 모두 이상이 없는지 점검해야 한다. 물론 최종 Sign Off 과정에서는 setup, hold violation뿐만 아니라 max transition, max fan out 등도 점검되어야 한다.



Layout 이후 timing violation 발생 시 가장 좋은 해결 방법은 constraint를 명확히 확인한 뒤 합성부터 다시 진행하는 것이지만 이럴 경우 처음부터 검증을 다시 진행해야 하므로 많은 시간과 노력이 필요하다. 오류의 범위가 넓지 않고 일부만을 수정하여 해결이 가능한 상황일 경우 ECO를 진행하는 것이 효율적이다. ECO는 Engineering Change Order의 약자로 timing violation을 해결하기 위해 특정 구간에 buffer를 삽입하거나 cell의 size를 조정하는 방법 등으로 문제를 간편하게 해결하는 방법이다. 기본적으로 ICC나 Astro에서도 기본적인 timing analysis가 가능하고 buffer 등을 바로 삽입할 수 있으며 design에 바로 반영할 수 있다는 장점이 있으나 timing 분석의 결과에 대한 정확도가 떨어지므로 Timing Check 및 ECO는 PrimeTime 툴에서 진행하는 것이 일반적이다. 다만 PrimeTime은 netlist나 layout에 직접적인 영향을 주는 툴이 아닌 단순히 분석하는 툴이므로 PrimeTime을 통해 수정하고자 했던 사항은 Auto PnR 툴(Astro, ICC 등)이나 Synthesis 툴(DesignCompiler)을 통해 재 반영되어야 한다. ECO를 진행하기 전에 공정사에서 안내된 방법으로 Timing Analysis를 진행하였는지 점검해야 할 필요가 있다. 대부분의 IDEC MPW 공정은 Post STA 과정 시 StarRCXT(synopsys)를 통해 RC 값을 추출하고 Parasitic을 반영하도록 안내하고 있다. 간혹 Post STA 과정에서 parasitic을 반영하지 않거나 Layout 툴인 ICC나 Astro를 통해 RC 값을 추출하고 이를 Post STA 반영하는 경우가 있으나 이는 정확한 결과를 report 하지 않을 수 있으므로 주의해야 한다.

구분	Pre STA	Pre STA	비고
Netlist	Design Compiler Netlist	Astro or ICC Netlist	.v
Parasitic	없음	StarRCXT	.spef
Target LIB	Worst Case	All Corner & Mode	.db
Analysis	Setup Time	Setup&Hold Time MTTV 등	

표 1 Pre STA 와 Post STA 비교

만약 violation이 발생하게 될 경우 설계자는 spec out으로 보고 합성부터 다시 진행할 것인지, ECO를 진행할 것인지, 또는 clock margin, uncertainty margin으로 보고 수정 없이 진행할 것인지를 판단해야 한다.

ECO를 진행하는 방법 중 기본적으로 사용하는 방법은 cell size를 변경하는 것이다. Cell은 size와 delay 사이에 trade off 관계에 있으므로 size가 증가하게 되면 delay는 줄어들게 된다. Delay 감소를 위해 cell의 size를 키우게 될 경우 전체적인 면적이 다소 증가할 수는 있으나 cell을 추가로 삽입하는 것보다는 면적에 대한 이득이 있으므로 우선 size를 변경하여 해결할 수 있는지 점검해 보는 것이 좋다. 특정 구간에 이전 cell 대신 사용이 가능한 cell을 확인하기 위해서는 PDK를 통해 제공되는 cell 문서를 통해 점검해 볼 수 있으며 아래와 같이 명령어를 통해 library에서 확인할 수 있다.

```
pt_shell>report_alternative_lib_cells Usoc/Uamba_ahb/Uahb_lite/U5
pt_shell> report_alternative_lib_cells Usoc/Uamba_ahb/Uahb_lite/U5
Information: Loading library data due to accessing lib cells not link
Information: Loading library data due to accessing lib cells not link
Information: Loading library data due to accessing lib cells not link
*****
Report : alternative_lib_cells
Usoc/Uamba_ahb/Uahb_lite/U5
-delay_type max
Design : pad_soc_uart_fifo/0 (unknown)
Version: F-2011.12-SP3-5
Date : Tue Sep 10 03:52:39 2013
*****
Alternative Library Cells Slack
-----
m18gm180s_wci/INVX8 7.64(f)
m18gm180s_wci/INVX4 7.64(f)
m18gm180s_wci/INVX3 7.60(f)
m18gm180s_wci/CLKINVX8 7.59(f)
m18gm180s_wci/RFRDX4 7.58(f)
m18gm180s_wci/CLKINVX4 7.56(f)
m18gm180s_wci/INVX16 7.54(f)
m18gm180s_wci/INVX20
```

사용이 가능한 cell의 종류를 확인한 뒤 size를 변경하기 위해서는 아래 명령어를 통해 진행할 수 있다. Size 변경 후에는 report timing 명령어를 통해 이상 없이 cell이 변경 되었는지를 점검하고 delay 값을 확인해 볼 수 있다.

```
pt_shell> size_cell Usoc/Uamba_ahb/Uahb_lite/U5 m18gm180s_wci/INVX8
```

Cell의 size 변경으로 timing violation이 해결되지 않을 경우 cell을 삽입해야 할 필요가 있다. 특정 구간에 buffer cell들을 삽입하기 위해서는 삽입하고자 하는 구간, 삽입 셀 이름을 확인한 뒤 insert buffer 명령어를 통해 삽입할 수 있다.

```
pt_shell> insert_buffer Usoc/Uamba_ahb/Uahb_lite/U5/Y m18gm180s_wci/BUF4
clock MAIN_CLOCK (rise edge) 7.50 7.50
clock network delay (propagated) 4.67 12.17
Usoc/Uamba_ahb/Uahb_lite/Uahb_s2m/u_hsel_reg_reg_3_/CK (DIFFHX0X1) 0.00 12.17 r
Usoc/Uamba_ahb/Uahb_lite/Uahb_s2m/u_hsel_reg_reg_3_/Q (DIFFHX0X1) 0.58 & 12.75 r
Usoc/Uamba_ahb/Uahb_lite/Uahb_s2m/U20/Y (NOR3X1) 0.30 & 13.05 f
Usoc/Uamba_ahb/Uahb_lite/Uahb_s2m/U34/Y (NAND4X1) 0.32 & 13.37 r
Usoc/Uamba_ahb/Uahb_lite/Uahb_s2m/U5/Y (CLKINVX3) 0.62 & 13.99 f
Usoc/Uamba_ahb/Uahb_lite/Uahb_s2m/U4/Y (NOR3X1) 0.24 & 14.23 r
Usoc/Uamba_ahb/Uahb_lite/Uahb_s2m/U3/Y (NAND4X1) 0.38 & 14.61 f
Usoc/Uamba_ahb/Uahb_lite/Uahb_s2m/U25/Y (NAND4X2) 0.52 & 15.13 r
Usoc/Uamba_ahb/Uahb_lite/Uahb_s2m/HREADY (ahb_s2m_s5) 0.00 & 15.13 r
Usoc/Uamba_ahb/Uahb_lite/U5/Y (INVX8) 0.07 H 15.20 f
Usoc/Uamba_ahb/Uahb_lite/II58/Y (BUF4) 0.17 H 15.37 f
Usoc/Uamba_ahb/Uahb_lite/U4/Y (INVX8) 0.16 & 15.53 r
Usoc/Uamba_ahb/Uahb_lite/M_HREADY (ahb_lite_s5) 0.00 & 15.53 r
Usoc/Uamba_ahb/M_HREADY (amba_ahb_m4s5) 0.00 & 15.53 r
Usoc/Ucore/D_HREADY (corea_ahb) <- 0.00 & 15.53 r
```

앞서 언급한 것처럼 변경된 내용을 layout 툴에서 적용하기 위해서는 수정된 내용을 별도의 파일로 export 한 뒤 layout 툴에서 다시 load 해야 한다. Layout 툴에서는 netlist 에 수정사항을 반영한 뒤 추가해야 하는 새로운 cell이 있을 경우 placement를 진행해야 한다. 이때 core filler가 있을 경우 삽입할 공간이 없으므로 filler는 제거한 뒤 진행한다. placement 후에는 해당 cell에 대해 route를 진행하여 실제 layout 상에서 net이 연결될 수 있도록 한다.

```
pt_shell> write_changes -format icctcl -output eco_test.eco
icc_shell> eco_netlist -by_tcl_file eco_test.eco
icc_shell> legalize_placement -eco -incremental
icc_shell> route_zrt_eco
```

위에서 사용된 format은 IC Compiler 용으로 저장한 것이며 기본적으로 write_change 명령어로 지원되는 format은 PrimeTime, DesignCompiler, IC Compiler, Text이다. Astro 의 경우 2009년도 이전 버전에서만 format을 지원한다.

◆ Multi-Scenario STA

Post STA에서 Worst Case와 Best Case를 각각 진행하는 과정에서 한쪽 Case에서 발생하는 violation을 해결하기 위해 ECO를 진행하게 될 경우 다른 case에서 violation이 추가로 발생할 수 있으며 mode와 corner가 많은 경우 STA에 많은 시간이 필요하게 된다.

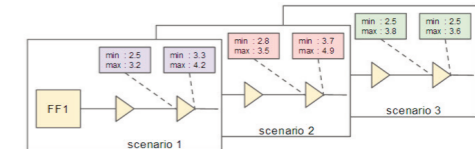


그림7. Concept of Multi-Scenario

이를 개선하기 위해서 최근 사용되고 있는 방법은 Multi-Scenario Mode를 사용하는 것으로 다양한 scenario를 구성하여 이에 대해 동시에 STA를 진행하는 방법이 많이 사용되고 있다. Multi-Scenario 진행을 위해서는 pt_shell -multi 명령어로 툴을 실행해야 하며 mode 수만큼 라이선스가 추가로 필요하다. 다양한 corner에 대한 timing violation을 동시에 확인할 수 있다는 장점과 ECO 역시 모든 corner 상황을 반영하여 진행할 수 있다는 장점이 있으므로 검증 시 활용해 보는 것도 좋을 것으로 생각된다.

이번 호에서는 Equivalence Check와 Static Timing Analysis & ECO 방법에 대해서 간략하게 알아보았다. STA는 중요한 검증 과정이며 알아두어야 할 것이 많으므로 반드시 한번은 IDEC 강좌를 수강할 것을 권장한다. 다음 호에서는 Digital Auto PnR 에 대해 주의사항과 기본 진행 과정에 대해 알아보도록 한다.



김 연 태 전임연구원
소속 : 반도체설계교육센터
E-mail : ytkim@idec.or.kr
http://www.idec.or.kr

본 내용은 2013년 9월에 IDEC 본센터에서 개설된 IDEC 연구원 교육 자료를 요약, 정리한 것으로서 자세한 자료는 IDEC 홈페이지에서 다운로드 및 VoD 시청이 가능합니다.
본 내용의 무단 배포 및 사용을 금합니다.



천지민 교수
금오공대
전자공학부

따뜻한 기술을 연구하고 싶은 연구자

스마트 센서 시스템은 디지털카메라, 스마트폰, 나아가 구글 글래스와 같은 웨어러블 디바이스에서 이미지 및 영상 획득을 위해 필수인 CMOS 이미지 센서와 IoT 세상을 구현하기 위한 필수적 기반 기술이다. 전기·전자공학을 전공하고 현재, 금오공과대학교 전자공학부 조교수로 재직 중인 천지민 교수를 만나 그간의 연구 인생에 대해 들어보았다.

“웨어러블 디바이스, IoT를 통해 부각되는 연구 분야”

천지민 교수는 CMOS 공정을 이용한 이미지 센서 및 스마트 센서 시스템 구현을 위한 analog/mixed-signal 집적회로를 연구한다. 디지털카메라, 스마트폰, 나아가 구글 글래스와 같은 웨어러블 디바이스에서 이미지 및 영상 획득을 위해 필수인 CMOS 이미지 센서와 사물인터넷(IoT: Internet of Things) 세상을 구현하기 위한 필수적 기반 기술을 연구하고 있다. 웨어러블 디바이스 및 사물인터넷이 대두하고 있는 요즘, 각종 스마트 센서 시스템 연구개발에 대한 요구는 더욱 커질 것으로 예상하며 특히 여러 기술과 산업 간 협력이 필요한 전형적인 융합 산업으로 다양한 분야의 많은 연구인력이 필요한 분야이다. 그는 연세대에서 학사·석사·박사를 모두 마친 후, 삼성전자 System LSI 이미지개발팀에서 근무했다. 그곳에서 국내 최초로 DSLR/Mirrorless 디지털카메라용 ADC 내장 APS-C 크기 CMOS 이미지 센서를 개발을 수행하고, 이후 SK telecom 융합기술원 반도체 Tech. Lab에서 반도체 분야 신기술/신사업 발굴 업무를 하다가, 지금의 금오공과대학교로 오게 되었다.

“꿈은 이루어진다.”

천지민 교수는 어렸을 때 꿈을 이룬 사람이다. “초등학교 때, 신문에 삼성전자 DRAM 개발 기사를 보고 감탄된 것 같은 전율을 느꼈습니다. 아무것도 모르는 어린 꼬마였던 제가 무작정 반도체 설계 엔지니어를 꿈꿨지요.” 천지민 교수의 석·박사 과정 중 진행한 핵심 연구 테마는 <CMOS 이미지 센서 시스템 설계>였다. 일반적인 디지털카메라를 위한 <CMOS 이미지 센서에 필요한 픽셀 설계 및 리드아웃 회로 설계>, <구조광(structured light)을 이용한 고조도 환경에 적용 가능한 3D 센서>, <생체 신호>, <연기 및 온도 측정 등을 위한 각종 스마트 센서 시스템>도 함께 진행했다. 어릴 적 꿈을 이룬 그였지만 전자회로 과목은 힘들었다고 회상했다. 그러면서도 “Spice 시뮬레이션을 통해 진행되는 회로 설계 프로젝트는 너무 재미가 있었어요. 설계 결과를 바로 눈으로 확인할 수 있는 매력이 지금 전공분야를 결정하게 되었습니다.” 그러면서 자신의 석·박사 과정을 지도해 주신 교수님의 연구자로서의 열정과 비전도 지금의 자신을 있게 해준 계기였다고 한다.



“처음의 중요성과 긍정의 힘”

처음 진행한 일이 손조롭게 진행이 되면 누구나 자신감도 생기고 계속해서 더 좋은 결과를 얻는 원동력이 된다. 아마도 그가 석사과정 입학 후 첫 칩 설계가 가장 기억에 남는 것도 같은 이유가 아닐까? “직접 설계하고 레이아웃을 진행한 첫 번째 칩이 동작에 성공했을 때는 최고였던 거 같습니다.” 연구 진행의 어려움이 없었느냐는 질문에서도 천지민 교수는 무한 긍정을 표현했다. “누구나 새로운 아이디어 창출의 어려움은 있고 저 또한 마찬가지입니다.” 세계 각지에서 비슷하거나 같은 연구를 하는 연구자들이 많기 때문에 생각해 낸 새로운 아이디어가 새롭지 않은 상황을 많이 맞이할 때가 많다는 것이다. “실망은 하지 않아요. 다만 세상에 나와 같은 생각을 하는 사람들이 많다는 사실이 놀랍고 신기합니다. 그땐 ‘난 어떤 아이디어를 먼저 내어서 다른 사람을 놀라게 할 수 있을까?’ 하고 한 번 더 생각을 합니다.”

“현재를 소중히 생각하고 즐기자”

천지민 교수는 연구자에게 조급함은 금물이라고 말한다. “무언가를 이루겠다는 조급함보다는 기본에 충실한 연구가 중요한 거 같습니다. 기본이야말로 연구 기반을 단단하게 만들고 이를 통해 더 창의적인 아이디어가 나올 수 있다고 생각합니다.” 그러면서 그는 현재의 시간을 소중하게 생각하고 즐기기를 강조했다. “나 자신에게 ‘나중에’, ‘다음에’ 라는 말은 없다고 생각하면서 하루하루를 소중히 보냈으면 좋겠습니다.” 천지민 교수는 삼성전자의 AP 분야에서의 활약으로 성장하는 것으로 보이는 비메모리 분야는 여전히 핵심 시스템 설계, 응용 S/W 개발 등에서 절대적 열세를 보이고 있다고 말한다. 그러면서 비메모리 분야는 설계자인 핵심인력의 경쟁력에 의해서 산업의 경쟁력이 좌우되기에 국가 및 관련 기업에서 핵심인력 양성을 위한 대책을 다시 점검할 필요가 있다고 지적했다. 어떤 분야의 도전보다는 본인이 갖춘 연구능력을 활용하여 사회적인 안전과 장애인 및 고령자 등 사회적 약자의 생활에 도움이 되는 따뜻한 기술을 연구하고 싶다는 그의 포부에서 앞으로의 연구 활동에 대한 기대와 희망이 느껴진다.

문의 금오공과대학교 전자공학부 마이크로시스템Tech.연구실 전화 054-478-7436
E-mail jimin.cheon@kumoh.ac.kr Homepage <http://sites.google.com/site/kmstlab/>



Dreams come true!