

IDECE Newsletter

IDECE Newsletter | 통권: 제163호 발행일 | 2010년 12월 30일 발행인 | 경중민 편집인 | 김이섭 제 작 | 무을디자인
기 획 | 전형기 전 화 | 042) 350-8535~6 팩 스 | 042) 350-8540 http | //idec.or.kr
E-mail | jhg0929@idec.kaist.ac.kr 발행처 | 한국과학기술원 반도체설계교육센터(IDECE)

Vol.163

2011
January

IDECE NEWS | 02 2011 신년사 | 04 TSV 기반 3D stacked IC 테스트 설계기술 | 06 IT기기에의 무선 에너지 공급 기술동향 | 10
FreeRTOS 실시간 커널과 Core-A 프로세서 (3) | 14

반도체설계교육센터 사업은 지식경제부, 반도체산업협회, 반도체회사(삼성전자, 하이닉스반도체, 매그나칩반도체, 동부하이텍, 앰코테크놀로지코리아, KEC, 세미텍, TowerJazz)의 지원으로 수행되고 있습니다.

2011 신년사

IDECE의 지난 15년간 행보는 '느려도 소결음'이라는 말처럼 큰 변화 없는 '꾸준한 나아감'이었다. 정부예산 규모가 너무 작아서 우리 반도체 산업의 규모나 요구에 턱없이 못 미치지만 그래도 오랜 기간 사업을 해오면서 정부, 대학, 산업체의 지원과 후원을 받고 있는 것은 감사한 일이다. 2011년에 우리가 할 중요한 일과 갈아가야 할 길에 대해 경중민 소장의 신년사를 통해 살펴보고자 한다. (관련기사 P04~05참조)

TSV 기반 3D stacked IC 테스트 설계기술

지속적인 미세 공정기술의 개발과 더불어 2D IC에서 interconnect 연결선이 회로지연과 전력소모의 주된 요소가 되고 있지만, 3D IC로 말미암아 회로지연과 전력소모를 획기적으로 줄일 수 있게 되었다. 3D IC는 무어의 법칙에 따라 지속적으로 집적도가 증가할 수 있는 획기적인 기술이지만 칩의 생산성과 테스트 기술의 향상 없이 실용화하는 데는 한계가 있다. 본 고에서는 TSV 기반 3D stacked IC 테스트 설계기술의 중요성에 대해 알아보고자 한다. (관련기사 P06~08 참조)

IT기기에의 무선 에너지 공급 기술동향

무선 전력 전송의 개념은 약 100년 전 테슬라로부터 시작되어 현재 다양한 방식으로 산업체에서 실제 적용되고 있다. 물론, 최근의 눈부신 통신 기술의 발전에 비해서 그 변화속도는 더딘 편이었다. 가정 및 사무용 기기에서의 적용은 에너지 전송 거리상의 제약, 낮은 효율, 그리고 인체에 미치는 영향에 의해 널리 보급되고 있지 않은 편이다. 본 고에서는 IT기기에서의 무선 에너지 공급 기술동향에 대해 살펴보고자 한다. (관련기사 P10~P13 참조)

FreeRTOS 실시간 커널과 Core-A 프로세서 (3)

FreeRTOS는 원천 코드가 공개되어 있고, 무료로 사용할 수 있는 실시간 커널이며 ARM, AVR, MSP430, PIC, 8051 등 8-비트부터 32-비트까지 다양한 프로세서에 이식되어 사용되고 있다. 특히 코드의 양이 비교적 적고 대부분 C 언어로 작성되어 있으며 프로세서 의존적인 부분이 최소화되어 있어 매우 제한된 부분만 어셈블리로 구현하면 쉽게 새로운 프로세서에 이식 할 수 있다. 본 고에서는 총 4회에 걸쳐 "FreeRTOS 실시간 커널과 Core-A 프로세서"란 주제를 다루는 세 번째 시간으로 FreeRTOS를 활용하여 응용 프로그램을 개발할 때 사용하는 중요 API에 대해 정리하고자 한다. (관련기사 P14~P18 참조)

IDEC January | 2011 news

MPW (Multi-Project Wafer)																	
신청 현황 I						MPW 칩제작 현황 I											
구분	공정	제작가능면적 (mm2 x 칩수)	신청 칩수	채택 칩수	설계면적 (mm2 x 칩수)	DB 마감	Die-out	비고	구분	공정	제작가능면적 (mm2 x 칩수)	제작 칩수	제작면적 (mm2 x 칩수)	Die-out 예정일	현재 상태	비고	
제98회 (11-01)	TJ SiGe 0.18	5x5mm ² x 1	5	4	2.5x2.5mm ² x 4	2011. 2.11	2011. 5.24		제94회 (10-05)	M/H 0.35	5x4mm ² x 20	17	5x4mm ² x 17	2010. 11.10	제작 완료	PKG :12,14	
	M/H 0.35	5x4mm ² x 20	19	19	5x4mm ² x 19	2011. 2.11	2011. 6.7			M/H 0.18	4.5x4mm ² x 20	17	4.5x4mm ² x 17	2010. 2.11			
	M/H 0.18	4.5x4mm ² x 20	20	20	4.5x4mm ² x 20	2011. 2.11	2011. 6.7			삼성 0.13	4x4mm ² x 48	42	4x4mm ² x 42	2010. 12.5	PKG 제작중	Die배포 :12,15 PKG: 12,30예정	
제99회 (11-02)	TJ RF 0.18	5x5mm ² x 2	5	5	5x5mm ² x 1 2.5x2.5mm ² x 4	2011. 2.21	2011. 5.11		TJ SiGe 0.18	5x5mm ² x 1	3	5x2.5mm ² x 1 2.5x2.5mm ² x 2	2010. 12.6	칩제 작중	Die배포 :12,30예정		
	동부 BCD 0.35	5x5mm ² x 3	6	6	5x2.5mm ² x 6	2011. 2.22	2011. 6.1		TJ CIS 0.18	5x5mm ² x 1	1	5x5mm ² x 1	2010. 12.30	칩제 작중			
100회 (11-03)	동부 BCD 0.35	5x5mm ² x 3	7	6	5x2.5mm ² x 6	2011. 4.7	2011. 7.12		제95회 (10-06)	동부 BCD 0.35	5x5mm ² x 2	4	5x2.5mm ² x 4	2010. 12.30	칩제 작중	Die배포 :12,10 PKG :12,23	
	삼성 0.13	4x4mm ² x 48	40	40	4x4mm ² x 40	2011. 4.8	2011. 8.10	추가 모집중	제96회 (10-07)	삼성 90n	4x4mm ² x 20	15	4x4mm ² x 15	2011. 2.20	칩제 작중		
101회 (11-04)	TJ RF 0.18	5x5mm ² x 2	8	8	2.5x2.5mm ² x 8	2011. 5.2	2011. 7.19		M/H 0.18	4.5x4mm ² x 40	32	4.5x4mm ² x 32	2011. 5.10	칩제 작중			
	M/H 0.18	4.5x4mm ² x 20	15	15	4.5x4mm ² x 15	2011. 5.13	2011. 9.5	추가 모집중	제97회 (10-08)	동부 BCD 0.35	5x5mm ² x 2	4	5x2.5mm ² x 4	2011. 2.1	칩제 작중		
102회 (11-05)	동부 BCD 0.35	5x5mm ² x 3	-	-	-	2011. 5.25	2011. 8.31		동부 0.13	5x5mm ² x 6	18	5x2.5mm ² x 6 2.5x2.5mm ² x 12	2011. 2.25	칩제 작중			
103회 (11-06)	TJ CIS 0.18	5x5mm ² x 1	3	2	2.5x2.5mm ² x 2	2011. 6.21	2011. 9.21		제4회 동부IP 0.35	5x5mm ² x 16	14	5x5mm ² x 14	2011. 1.20	칩제 작중	Die제작 :11,15예정		
	동부 BCD 0.35	5x5mm ² x 3	2	2	우선모집결과 5x5mm ² x 1 2.5x2.5mm ² x 1	2011. 7.7	2011. 10.12		* 103 ~ 104회는 우선모집 결과임(~12.5) - 전체모집 가능수의 50% 선정 * M/H = 매그나칩/하이닉스, TJ = TowerJazz * 우선/정규모집에서 마감될 경우 후기모집은 진행되지 않음. * Die-out 일정은 Diechip 제작완료 기준임. Package는 Die-out후 3주 이상 소요됨. * 위의 내용은 12/24 기준임. * 2011년 MPW 참여 일정 및 방법은 홈페이지(www.idec.or.kr) 참조								
104회 (11-07)	TJ BCD 0.18	5x5mm ² x 2	0	0	-	2011. 8.29	2011. 11.15		* 문의 : 이의숙(042-350-4428, yslee@idec.kaist.ac.kr)								
	M/H 0.35	5x4mm ² x 20	3	3	5x4mm ² x 3	2011. 8.29	2011. 12.20										
104회 (11-07)	M/H 0.18	4.5x4mm ² x 20	1	1	4.5x4mm ² x 1	2011. 8.29	2011. 12.20										
	삼성 65n	4.5x4mm ² x 20	11	11	4x4mm ² x 11	2011. 8.19	2011. 1.15										

반도체설계교육센터(IDEC) 장학금 안내

- 대상 WG 소속 재학생 또는 입학예정자로서 WG 책임교수의 추천을 받은 자
- 책임교수 추천기준 (1) 가정형편이 곤란하여 학비조달이 어려운 자 (2) 학업성적이 우수한 자 (3) 기타 장학금 지급이 필요하다고 인정된 자
- 장학금액 130만원(1인/학기) * 원천징수 후 현금지급
- 선발인원 0명
- 신청서류 IDEC 장학금 신청서, WG 책임교수 추천서, 성적증명서, 재학증명서 (입학예정자의 경우 이전학교의 성적증명서, 입학증명서)
- 서류접수기간 2011년 1월 3일 ~ 1월 31일까지
- 선발공지 2월

* 상세내용은 반도체설계교육센터(IDEC)홈페이지를 참고해주세요.
담당 : 최신희(042-350-4045, shchoi@idec.kaist.ac.kr)

2011년 1월 교육 프로그램 안내

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

본 센터 강좌 일정 I

강의 일자	강의 제목	분 류
1월 10-14일	Cadence - VSE & VADE Training	Tool
1월 17일	[DAC] CMOS 혼성모드 시스템 설계 및 실습	설계
1월 18-19일	[ADC] CMOS 혼성모드 시스템 설계 및 실습	설계
1월 20-21일	[PLL] CMOS 혼성모드 시스템 설계 및 실습	설계
1월 24-25일	고성능 ADC 설계를 위한 이론 및 최신 설계기법	설계
1월 27-28일	고성능 PLL 주파수 합성기 설계	설계

[사전지식, 선수과목]
 · 석사과정 대학원생 초급 반도체 집적회로 설계 경험
 · 설계/실습 툴 사용 경험이 없는 학부 학생은 수강 불가

- 강좌일 : 1월 20일-21일
- 강좌 제목 : [PLL] CMOS 혼성모드 시스템 설계 및 실습
- 강사 : 유종근 교수(인천대)

[강좌개요]
 PLL의 기본 개념, 응용분야, Design issue 등 설계에 필요한 기본적인 내용에 대해 알아보고, PLL을 구성하는 기본 블록들(VCO, PFD, Charge Pump, Loop Filter, Divider 등)의 역할과 특성, 그리고 실제 샘플 회로들을 대상으로 구성 블록들의 구체적인 설계 과정을 다룬다. 그리고 각 블록들이 전체 PLL 성능에 미치는 영향, 안정성 문제 등에 대해 논한다. 또한, 학습한 구성블록들과 전체 PLL 회로를 모의실험을 해봄으로써, 구성블록들의 동작원리 및 특성 등을 직접 경험해 보도록 한다.

[수강대상]
 · 아날로그회로 설계를 연구하는 대학원생 및 기업체 엔지니어

[강의수준]
 · 아날로그회로 설계를 연구하는 대학원 석사과정 정도의 수준으로 함.
 학부 수준의 전자회로 과목에서 기본적인 회로를 해석하고 시뮬레이션 해본 것으로 가정함.

[강의형태]
 · 이론 + 실습

[사전지식, 선수과목]
 · 학부 수준의 전자회로 및 집적회로 설계
 · Cadence 사용 경험 필요

- 강좌일 : 1월 10일-14일
- 강좌 제목 : Cadence - VSE & VADE Training
- 강사 : 이상철 과장(Cadence Korea)

[강좌개요]
 본 과정에서는 Virtuoso® Schematic Editor & Analog Design Environment System하 에서 Front-to-Back Design을 다루게 된다. 디자인의 Top-level Block description부터 시작하여, Run simulation 그리고 Virtuoso Analog Design Environment & Schematic Editor 전체를 Utilize할수 있는 방법을 배우게 된다.

[강의수준]
 · 중급

[강의형태]
 · 이론 + 실습

[사전지식, 선수과목]
 · SPICE 유경험자, Analog Design경험자

- 강좌일 : 1월 17일
- 강좌 제목 : [DAC] CMOS 혼성모드 시스템 설계 및 실습
- 강사 : 윤광섭 교수(인하대)

[강좌개요]
 데이터 변환기의 개요, 응용분야 및 데이터 변환기 성능 변수에 대해서 알아본다. DAC 구조에 따른 성능을 살펴보고, 나이퀴스트 DAC에서 전하구동 및 전류구동 2진 가중치형 DAC, 온도계 코드 형 DAC, 혼합형 DAC의 설계 방법론을 설계 예제를 통해서 알아보도록 한다. 또한 최근 고해상도 DAC 설계 동향을 살펴본다.

[수강대상]
 · 아날로그회로설계를 연구하는 학부/대학원생/기업체 엔지니어

[강의수준]
 · 전자회로, 디지털 논리회로 사전지식을 갖춘 공대생

[강의형태]
 · 이론 + 실습

[사전지식, 선수과목]
 · 전자회로, 디지털 논리회로, HSpice 또는 Spice 사용자

- 강좌일 : 1월 18일-19일
- 강좌 제목 : [ADC] CMOS 혼성모드 시스템 설계 및 실습
- 강사 : 이승훈 교수(서강대)

[강좌개요]
 아날로그 SoC 개요, ADC의 기본적인 원리, 구조 및 최근 연구개발 동향을 논의하며, 비교기 등 기본 회로 블록에서 시작하여 6b Gbps ADC 핵심 구성회로를 HSPICE 및 CADENCE 등 툴을 이용하여 설계 등 실습

[수강대상]
 · 아날로그 회로를 연구하는 석박사 대학원생 및 기업체 연구원

[강의수준]
 · 아날로그 회로를 연구하는 대학원 석사과정 이상 수준
 · HSPICE 및 CADENCE 레이아웃 툴을 다룰 수 있는 수준

[강의형태]
 · 이론 + 실습

- 강좌일 : 1월 24일-25일
- 강좌 제목 : 고성능 ADC 설계를 위한 이론 및 최신 설계기법
- 강사 : 류승탁 교수(KAIST)

[강좌개요]
 본 강좌는 Nyquist A/D converter를 중심으로 설계에 관해 기본 원리 부터 시작하여 최신 설계 동향까지 다루게 된다. 먼저 ADC의 동작원리와 성능척도에 대해 소개하고, 여러 형태의 ADC에서 성능저하를 일으키는 요인에 대해 고찰하여 실제 설계에서 고민해야 할 점들을 이야기한다. 또한, 최근 활발히 연구되고 있는 고속 flash, 저전력 pipeline, 고속 SAR ADC 등을 중심으로 고속 저전력 설계기법들을 살펴본다.

[수강대상]
 · 아날로그 IC를 전공하는 대학원생, 직장인

[강의수준]
 · 중급

[강의형태]
 · 이론 + 실습

[사전지식, 선수과목]
 · 학부수준의 전자회로 지식은 필수로 하고, 대학원 수준의 아날로그회로설계 지식이 있으면 수강이 용이함.

- 강좌일 : 1월 27일-28일
- 강좌 제목 : 고성능 PLL 주파수 합성기 설계
- 강사 : 조성환 교수(KAIST)

[강좌개요]
 PLL의 기초와 원리를 이해하고 이를 집적회로로 응용한 고성능 주파수 합성기, 클럭 발진기 등에 대하여 알아본다.

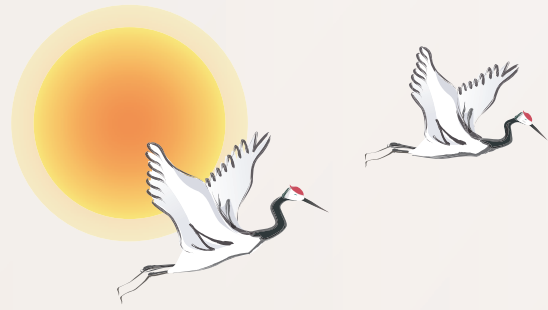
[수강대상]
 · 산업체 인력 및 대학원생

[강의수준]
 · 중급 (대학원 수준, 전자과 학부를 졸업한 인력 대상)

[강의형태]
 · 이론

[사전지식, 선수과목]
 · 회로이론, 전자회로, 신호 및 시스템, 제어시스템, 랜덤 프로세스

2011 新年賀辭



대만과 비교해 본 우리 반도체 산업

2011년에 우리가 할 가장 중요한 일은 무엇인가. 우리가 걸어가야 할 미래로 가는 길은 어떤 것일까. 격변의 지난 한 해를 되돌아보며 길게 뻗어난 미래로 가는 길을 걸어갈 차비를 해보자.

IDEC 의 지난 15 년간 행보는 '느려도 소결음' 이라는 말처럼 큰 변화 없는 '꾸준한 나아감' 이었다. 정부예산 규모가 너무 작아서 우리 반도체 산업의 규모나 요구에는 턱없이 못 미치지만 그래도 나름대로 오랜 기간 사업을 해오면서 정부, 대학, 산업체의 지원과 후원을 받고 있는 것은 감사한 일이다.

그러나 잘나가면서 나라를 일으켜 세우는데 혁혁한 역할을 한 대만의 반도체 인력양성 정책과 그 업적에 비교하면 우리의 반도체설계인력 양성사업은 매우 빈약하다. IDEC 의 counterpart 될 수 있는 대만의 CIC (Chip Implementation center) 를 통한 반도체설계 인력양성 사업과 비교하면 IDEC 사업은 예산규모 면에서 대만의 약 20~30 분의 일, 연구원 수는 20 분의 일, 연간 칩 설계수는 6 분의 일, 연간 배출하는 반도체 설계 분야의 석사 졸업생수는 5 분의 일 수준이다.

지난 10 년간 대만 fabless 산업이 눈부신 성장을 해 온 반면 우리 fabless 산업은 거의 제자리 걸음을 해온 것은 어디에 까닭이 있을까? fabless 산업 뿐만 아니라 TSMC 라는 거대 foundry 기업으로 시작된 대만의 IT 회회리는 이제 우리의 아성이던 assembly, packaging 까지 장악해버렸고 시스템 산업, software 까지 그 기세를 확장해가고 있다. 이 거대한 차이가 어디에서 비롯된 것일까?

그것은 설계인력양성에 대한 비전의 차이에서 온 것이다. 우리가 정부주도사업도 제품개발과 기업체 직접 지원에만 집중하며 항상 서둘러 왔어도 열매는 별로 없다. 우리가 연 10억원도 안되는 정부투자로 인력양성사업을 15년 동안 끌어온 반면 대만은 실리콘소프트라는 프로젝트를 통해 1조원도 넘는 돈을 인력양성과 인프라 구축에 투자했다. 우리도 더 이상 늦추어선 안 된다. 과감하게 설계인력을 키워내는데 나랏돈을 투자해야 한다. 지금부터라도 상황을 직시하고 전략을 세워 집행해가지 않으면 우리 반도체 산업은 물론 시스템 산업의 미래도 어두울 수밖에 없다. software 도 혼자 번성할 수 없다.

우리 산업이 살려면 It 산업의 여러 player 가 공존 번영할 수 있는 생태계를 만들어야 한다. 생태계의 기본틀 구축에는 반드시 정부가 나서야 한다. 이 기본틀은 바로 platform 이라 하는 것이다. 화전민과 달리 정착 농부는 농사를 짓기 전에 관개시설을 먼저 만들 듯이 우리 역시 플랫폼을 만들어야 한다.

우선 mobile, DTV, 자동차, ubiquitous, health 와 같은 유망분야에 대하여 engineering platform을 만들어야 한다. platform 이 공간이라면 이 공간은 반도체 설계, embedded software, 시스템 설계의 세 축으로 이루어져야 한다. 여기에는 산업체, 연구소, 학교와 정부가 모두 선수(選手)로 들어와 같이 가꾸고 같이 먹으며 어울려야 한다. 대학의 교과목, 학생들의 연구 토픽과 환경이 모두 이 platform 상에서 이루어져야 한다. 정부주도 사업도 좀 더 장기적인 안목에서 반드시 제품의 핵심기술 개발과 아울러 이를 따라가며 지원하고 이끌고 갈 핵심 설계인력 양성에 대한 투자가 따라 붙어야 한다. platform 은 장터와 같다. 이것을 만드는데 정부의 역할이 매우 크다. 일단 생태계가 굴러가기 시작하면 정부는 할 일이 더 없다. 성공한 것이다. 그것이 성공하는 멋진 정부의 모습이다. 대만은 여기에서 앞서 갔기 때문에 지금 획기적인 성과를 거두고 있다.

우리 정부도 반드시 장기적인 안목에 의거하여 사람을 키우고 platform을 만드는데 더 많은 돈을 써야 한다. software 산업에 대한 투자도 반드시 이 플랫폼 상에서 시스템, 반도체 칩의 개발과 같이 진행되어야 한다. 이것이 그래도 반도체산업과 시스템산업의 골격을 모두 가지고 있는 우리가 경쟁국가들, 즉, 칩 산업만 있거나 시스템, 혹은 software 산업만 있는 다른 나라에 비하여 상대적 산업경쟁력을 가질 수 있는 길이다. 이것을 우리의 2011 비전 삼아 모두 같이 열심히 뛰는 새해가 되길 간절히 바랍니다.

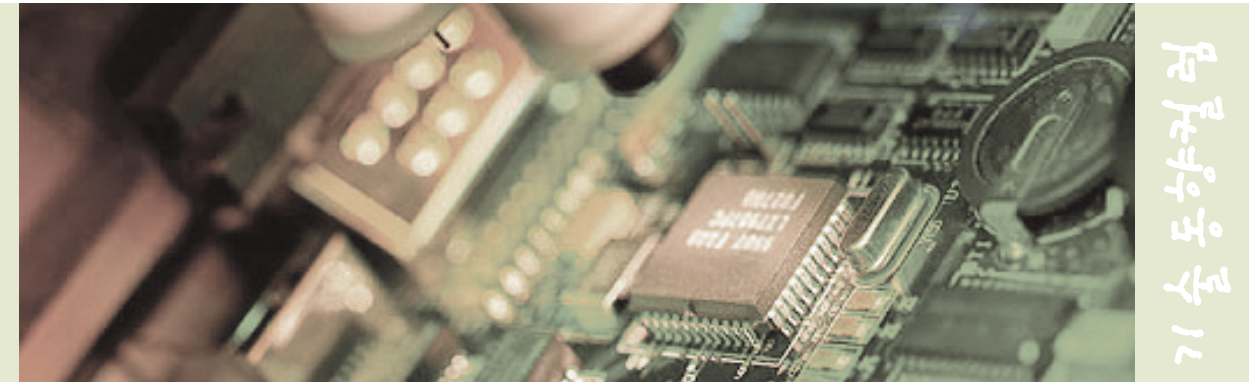
이렇게 하여 정말 탄탄 산업 발전대로 만들어 부강한 나라 열심히 만들어 가는 땀방울 속에서 우리 모두 같이 **복 많이 받는 새해가 되길 바랍니다.**

KAIST IDEC
소장 경종민



TSV 기반 3D stacked IC 테스트 설계기술

한양대학교 전자컴퓨터공학부
 박성주 교수
 연구분야 : SoC 테스트 (스캔, 경계스캔, AMBA, 연결선, BIST) 및 설계
 E-mail : parksj@msslab.hanyang.ac.kr
 http://msslab.hanyang.ac.kr



서론

지속적인 미세 공정기술의 개발과 더불어 2D IC에서 interconnect 연결선이 회로지연과 전력소모의 주된 요소가 되고 있지만, 3D IC로 말미암아 회로지연과 전력소모를 획기적으로 줄일 수 있게 되었다. Wire bond, microbump, 및 through-silicon-via (TSV) 등의 3D IC 집적 기술이 개발되어 다양한 분야에 응용되고 있다. TSV 기반 3D IC 칩은 CMOS Image Sensor, 16층 NAND flash (삼성), DRAM 메모리 stack (NEC 등), 스마트폰 용 메모리와 로직 통합 칩 (Intel), 게임기, 지능형 바이오센서 (NXP, ST) 등에 광범위하게 사용되고 있다. 그림 1과 같이 3D interconnect 기술은 메모리, 로직 및 CIS가 독립적인 모듈에서 TSV에 의한 통합 3D IC로 변하고 있다.

양품 die도 무용지물이 되므로 wafer 수준에서 다양한 테스트 기술을 적용해야 한다. 본 기고문에서는 TSV 기반 3D stacked IC에서 연구되고 있는 테스트 설계기술을 소개하고자 한다.

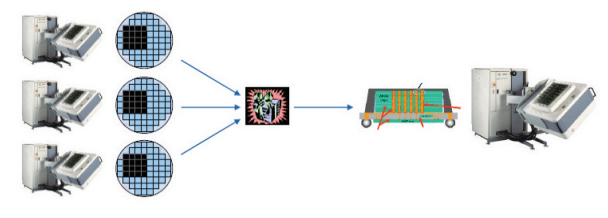


그림 2. TSV 방식 3D stacked IC 테스트 과정
 pre-bond die test → post-bond stack test → package test

■ TSV 기반 3D stacked IC 테스트 설계기술
 Wafer 수준에서 structural, DC, functional, spec-based의 완벽한 테스트를 통하여 Known Good Die(KGD) 테스트를 할 수 있다면 package 테스트 비용을 대폭 절감할 수 있다.

하지만, die간에 연결이 제한적일 경우에만 사용하며 at-speed 및 RF 테스트는 wafer 수준에서 적용하기 어렵다. 또한, stack quality 200ppm을 유지하기 위해서 wafer 수준에서 20ppm이 요구되는데 일반 DFT 기술을 적용해서는 현실적으로 어려운 점이 많다.

새로운 공정기술의 발달로 TSV 방식으로 연결하는 3D stacked IC 기술이 널리 사용되고 있다. Pre-bond die 및 post-bond stack 테스트에 적합한 DFT 기술이 개발되고 있다. 여러 die, 내장된 core, die 간 interconnect, 및 외부 I/O를 개별적이고 또한 종합적으로 테스트할 수 있는 3D IC 테스트 기술이 개발되고 있다[1-4].

그림 3은 3D IC를 구성하는 core, die, TSV stacked die가 계층적으로 연결된 테스트 구조를 보여준다. CPU, DSP, Memory, ASIC, Flash core 각각은 1500 wrapper로 씌워져 있고 BIST 등 내부 테스트장치는 1687 interface로 연결되어 있다.

Core로 구성된 각 die는 1149.1 경계스캔을 통해 접근하며 stacked die는 그림 3의 가운데 선과 같이 star형 1149.7 테스트 제어부에 연결되어 있다[4]. Package된 3D IC는 TMS와 TCK 핀만을 통하여 외부 테스트장치와 데이터를 교류한다.

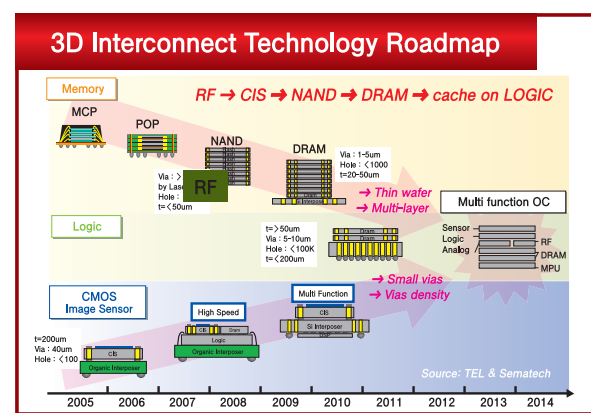


그림 1. 3D IC technology road map

3D IC는 무어의 법칙에 따라 지속적으로 집적도가 증가할 수 있는 획기적인 기술이지만 칩의 생산성과 테스트 기술의 향상 없이 실용화하는 데는 한계가 있다. 특히 테스트와 관련된 wafer-probing 기술, test access 기술, known-good-die(KGD), pretty-good-die(PGD), TSV 고장모델링 및 prebond design-for-testability (DFT), postbond test integration 기술 등의 선행연구가 수행되어야 한다[1-4].

3D IC는 그림 2와 같이 좌측의 pre-bond die 테스트를 거친 후 가운데 post-bond stack 테스트를 거쳐 최종적으로 package 테스트를 거친다. 불량 die 한 개로 말미암아 package 이후 다른

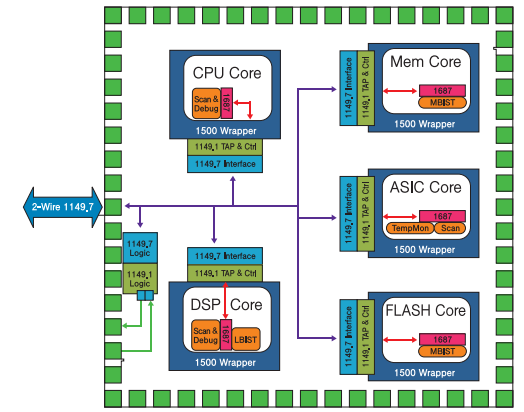


그림 3. 3D IC 계층적 Test Access Mechanism
 IEEE 1500, pi687, IEEE 1149.1, IEEE 1149.7

그림 4와 같이 package 핀이 연결된 bottom die, 위와 아래가 다른 die와 연결된 중간 die, 및 한쪽만 다른 die와 연결된 top die를 구분하여 intra-die 및 inter-die 테스트 비용을 최적화할 수 있는 테스트 구조가 개발되고 있다. Die-level wrapper는 적용된 다른 die에 대한 외부 인터페이스와 내부 die간에는 기능 핀의 연결 및 die 내에서 DFT 핀을 연결한다.

일반적으로 사용하는 IEEE 1500 wrapper에 top die, bottom die, 및 내부 die 간에 제한적인 핀 연결을 위하여 wrapper boundary register(WBR)이 사용된다. 정상모드와 테스트모드에서 Functional / Serial / Parallel 동작을 지원한다. Pre-bond 및 post-bond에서 전용 테스트 패드 사용을 고려하며 각 테스트 모드에 따른 테스트데이터 체인 구성과 테스트결과 출력이 용이하도록 설계한다.

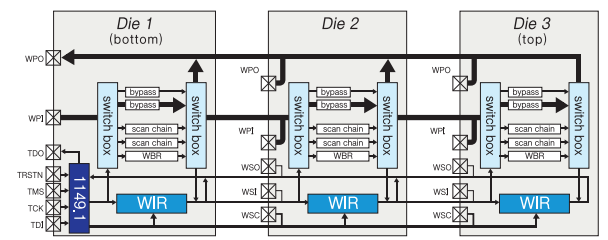


그림 4. 1500 wrapper 및 1149.1 기반의 die-to-die 테스트접근구조

그림 5는 layer 별로 연결된 코어가 서로 다른 layer와 테스트 체인을 연결하여 사용하는 경우를 보여준다. 위 layer는 5개의 코어가 3개의 TAM으로, 아래 layer는 6개의 코어가 3개의 TAM으로, 그리고 layer 간은 6개의 TSV를 통해 전체 3개의 TAM이 구성되어 있다. Inter-die와 intra-die TAM에 공통으로 사용되는 셀은 multiplexor를 이용하여 pre-bond와 post-bond에서 테스트통로를 적절히 설정할 수 있어야 한다.

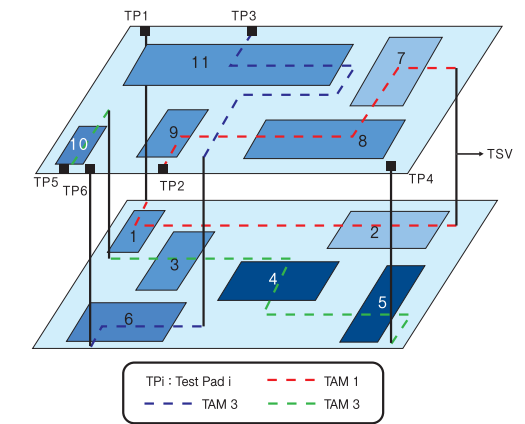


그림 5. 3D SoC의 Test Architecture

IEEE 1149.1을 이용하여 TSV를 테스트하는 여러 방법이 제안되고 있지만, 직렬 스캔 체인구조를 TSV 기반 3D IC 테스트에 적용하는데 여러 문제가 있다[6-8].

병렬구조로 1149.1을 연결해도 고장이 난 특정 layer를 진단할 수 없다는 단점이 있다. 1149.7은 기존 1149.1을 액세스할 수 있는 System Test Logic과 4 와이어 또는 2 와이어의 칩-레벨 TAP.7을 연결해주는 칩-레벨 TAP.7 컨트롤러를 제공한다.

그림 6은 die 4개를 TSV로 연결한 3D IC에서 TCK와 TMS 두 개의 테스트 핀을 사용하여 IEEE 1149.7을 구현하는 예제를 보여준다. IEEE 1149.7 표준을 적용하면 TSV의 interconnect는 물론 각 layer를 독립적으로 테스트할 수 있다. 메모리 셀이 포함된 다층 layer로 구성된 경우에는 BIST 및 BISR를 병행해서 고장 layer를 대체하고 soft error를 correct하는 회로도 널리 사용된다.

기술동향컬럼



다층의 메모리 블록과 제어블록이 TSV로 연결된 칩에서 IEEE 1149.1 표준을 사용하여 고장블록을 찾아내고 수리 및 재구성함으로써 활용도를 극대화할 수 있다. 또한, 타겟 테스트 layer만 파워를 공급함으로써 테스트 전력을 최소화할 수 있다.

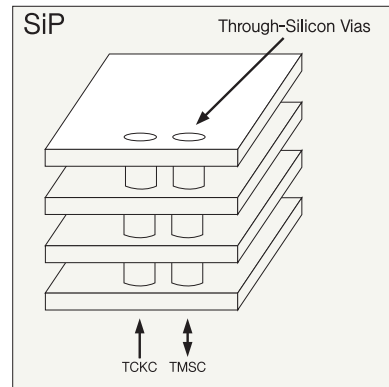


그림 6. 4-die SiP의 스타 형태 구조

맺음말

3D IC는 무어의 법칙에 따라 지속적으로 집적도가 증가할 수 있는 획기적인 기술이지만 생산성 제고를 위해서는 테스트 기술 개발이 선행되어야 한다. 2D IC에서 사용되고 있는 스캔 및 경계스캔 등 다양한 테스트 설계기술을 3D에 확장하는 기술에 대하여 살펴보았다. TSV에서 발생하는 다양한 hard 및 soft error를 체계적으로 점검하고 수리할 수 있는 테스트 설계기술의 개발이 필요한 시점이다.

Reference

- [1] S. Lee and K. Chakrabarty, "Test Challenges for 3D Integrated Circuits", IEEE Design & Test of Computers, pp. 26 - 35, 2009
- [2] Li Jiang; Qiang Xu; Chakrabarty, K.; Mak, T.M.; "Layout-driven test-architecture design and optimization for 3D SoCs under pre-bond test-pin-count constraint", 2009, ICCAD 2009, Page(s): 191 - 196
- [3] X. Wu, Y. Chen, K. Chakrabarty, and Y. Xie, "Test-Access Mechanism Optimization for Core-Based Three-Dimensional SOCs", pp. 212 - 218, International Conference on Computer Design, 2008.
- [4] S. Bahukudumbi and K. Chakrabarty, "Wafer-Level Modular Testing of Core-Based SoCs", IEEE Transaction VLSI systems, pp. 1144 - 1154, 2007
- [5] "IEEE Standard for Reduced-Pin and Enhanced-Functionality Test Access Port and Boundary-Scan Architecture", IEEE Std 1149.7-2009, vol., no., pp.c1-985, Feb. 10 2010
- [6] X. Wu, P. Falkenstern, and Y. Xie, "Scan Chain Design for Three-Dimensional Integrated Circuits (3D ICs)", IEEE International Conference on Computer Design, pp. 208-214, 2007.
- [7] Marinissen, E.J.; Verbree, J.; Konijnenburg, M.; "A structured and scalable test access architecture for TSV-based 3D stacked ICs," VLSI Test Symposium (VTS), 2010 28th, vol., no., pp.269-274, 19-22 April 2010
- [8] Ang-Chih Hsieh; TingTing Hwang; Ming-Tung Chang; Min-Hsiu Tsai; Chih-Mou Tseng; Li, H.-C.; "TSV redundancy: Architecture and design issues in 3D IC," Design, Automation & Test in Europe Conference & Exhibition (DATE), 2010, vol., no., pp.166-171, 8-12 March 2010

2011년 IDEC 교육프로그램 안내						
No.	월	개설날짜	강좌종류	강좌명	강사	
					이름	소속
1	1월	10-14	Tool	Cadence-VSE & VADE Training	이상철 과장	Cadence
2		17	설계	[DAC] CMOS 혼성모드 시스템 설계 및 실습	윤광섭 교수	인하대
3		18-19	설계	[ADC] CMOS 혼성모드 시스템 설계 및 실습	이승훈 교수	서강대
4		20-21	설계	[PLL] CMOS 혼성모드 시스템 설계 및 실습	유중근 교수	인전대
5		24-25	설계	고성능 ADC 설계를 위한 이론 및 최신 설계기법	류승탁 교수	KAIST
6		27-28	설계	고성능 PLL 주파수 합성기 설계	조성환 교수	KAIST
7	2월	8-11	설계	RF IC설계 교육	유형준 교수	KAIST
					김천수 박사	ETRI
8		17-18	Tool	Cadence-Virtuoso Schematic Editor Training	이길재 차장	Cadence
9		23-24	Tool	Cadence-Virtuoso Layout Editor Training	오윤석 부장	Cadence
10	3월	16	설계	High speed broadband transceiver IC design technique	배현민 교수	KAIST
11	4월	4-5	Tool	Cadence-Floorplanning, Physical Synthesis, Place and Route (Flat) v9.1	황재현 부장	Cadence
12	5월	3-4	Tool	Mentor-Calibre xRC	김봉준 차장	Mentor
13		11-13	Tool	Mentor-Calibre DRC/LVS	김봉준 차장	Mentor
14		27	설계	DC-DC Converter IC Design	미정	
15	7월	4-5	설계	고성능 ADC 설계를 위한 최신 설계기법	류승탁 교수	KAIST
16		8	설계	CMOS 혼성모드 시스템 설계 및 실습 - Analog 및 Digital I/O 및 ESD 설계	심진섭 수석 정종석 대표	매그나칩 제퍼로직
17		18-22	설계	MPW Chip 제작을 이용한 ASIC 설계 교육	이찬호 교수 유은광 연구원	송실대 IDEC
18	8월	8-9	설계	SoC architecture 설계	유승주 교수	포항공대
19		10-12	Tool	Cadence-AMS Designer	이길재 차장	Cadence
20		30-31	Tool	Cadence-NC Simulator	미정	Cadence
21	9월	6-8	Tool	Synopsys Design Compiler를 이용한 synthesis 교육	미정	Synopsys
22		9	Tool	Synopsys - Formality Jump Start	미정	Synopsys
23		20-22	Tool	Synopsys Prime Time tool을 이용한 Static timing Analysis 교육	미정	Synopsys
24		27-29	Tool	The next-generation physical design system IC Compiler 교육	미정	Synopsys
25		11월	18	설계	DC-DC Converter IC Design	미정
26	12월	21-23	설계	Platform을 이용한 SoC 설계 강좌	이상철 선임	에이디칩스

* 일정은 사정에 따라 변경될 수 있습니다.

* 문의 : 이승자 (042-350-8536, sjlee@idec.kaist.ac.kr)

Chip Design Contest (CDC)

● 제18회 한국반도체학술대회 CDC, 동아리설계경진대회 개최

1. 일정 및 장소
가. 일 시: 2011년 2월 17일(목), 09:30 ~ 16:00
나. 장 소: 제주 해비치호텔

2. 선정 결과

구분	논문제출		타학회 논문 기제출합 데모/패널만 참여 (논문미제출)	합계	설계동아리 경진대회
	ASIC	FPGA			
제출 편수	47	4	14	65	-
채택 편수	43 (D:14, P:29)	4 (D:4)	14 (D:2, P:12)	61 (Demo : 20 Panel : 41)	~12,31

* D : Demo, P :Panel

3. 동아리설계경진대회 참여 신청

- 참가신청 마감일 연기 : 10. 12. 10(금) -> 12. 31(금)

4. 시상 내역

시상명	내역		
	일반 부문	최우수상(1팀) 우수상(2팀)	상장 및 상금 100만원 각 상장 및 상금 50만원
Best Design Award	특별상 부문	SSCS 서울캠퍼스상(1팀)	상장 및 상금 100만원
FPGA Award	최우수상(1팀) 우수상(1팀)	상장 및 상금 50만원 상장 및 상금 30만원	
동아리설계 경진 대회	1등(1팀) 2등(1팀) 3등(2팀)	상장 및 상금 50만원 상장 및 상금 30만원 상장 및 상금 10만원	

* 데모 및 동아리팀 수가 확정되면 수상팀 수는 조정될 수 있음.

* CDC 참여와 관련한 자세한 사항은 홈페이지(www.idec.or.kr)를 참고해 주시기 바랍니다.
* 문의 : 이의숙(042-350-4428, yslee@idec.kaist.ac.kr)

채용 안내

● KAIST 반도체설계교육센터(IDEC)를 이끌어 갈 젊음과 열정을 가진 인재를 찾습니다.

1. 모집 분야

- 모집인원 : 연구직 (신입·경력), 0명
- 자격조건 : 대졸 이상, 반도체설계 관련 전공자, 영어 능통자 우대.
- 주요업무 : - 국·내외 Fab을 통한 칩 설계, 제작 지원
- Digital, Analog 칩 TEST기술 지원
- EDA tool 교육 및 기술 지원

2. 제출 서류

- 당사 입사지원서(홈페이지에서 다운로드)
- 최종학교 성적증명서 1부
- 졸업(예정) 증명서 1부

3. 접수 기간 및 접수처

- 접수기간 : 2011년 1월 31일(금)
- 접 수 처 : (305-701) 대전광역시 유성구 구성동 373-1 한국과학기술원
반도체설계교육센터 104호
- 접수방법 : 우편접수, 이메일 접수
- 문 의 처 : 042-350-4045, shchoi@idec.kaist.ac.kr
- 홈페이지 : www.idec.or.kr

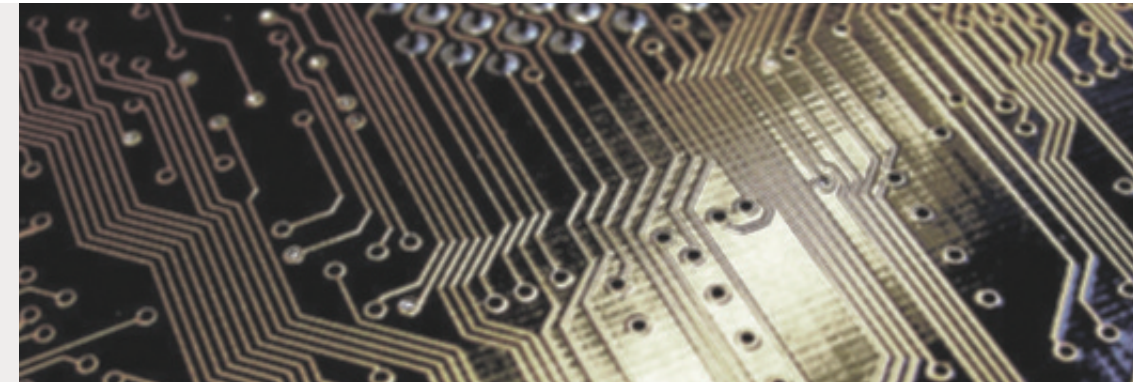
4. 기타

- 기타 자세한 사항은 홈페이지(www.idec.or.kr) 참조
- 접수된 서류는 인비 처리하며 반환하지 않음.

IT기기에의 무선 에너지 공급 기술 동향



경희대학교 국제캠퍼스 전자정보대학
 이종욱 교수
 연구분야 : 밀리미터파 CMOS회로설계, RFID 태그 칩 설계, 무선전력 전송 및 충전회로 설계
 E-mail : jwlee@khu.ac.kr
 http://web.khu.ac.kr/~jwlee



이종욱 교수

서론

최근의 급속한 IT 기술의 발전에 힘입어 개인 간의 통신은 단순한 문자 및 음성 전달에서 고속 데이터 및 화상 전송까지 언제 어디서나 가능하게 되었다. 주변에서 흔히 볼 수 있는 컴퓨터의 키보드, 마우스, 프린터 등도 무선통신 기능을 지원하게 되었고, 가전 기기에도 이러한 기능이 추가되고 있다. 하지만 이런 기기를 작동하는데 필요한 에너지는 유선으로 공급하거나 전지를 충전하여 사용한다. 만약, 그림 1에 나타난 바와 같이 이런 에너지 또한 무선으로 공급할 수 있게 된다면 현재의 우리 가정의 정보통신 기기 들은 새로운 방식으로 운용되고, 또 다른 기술적/문화적 발전을 하게 될 것이다[1].

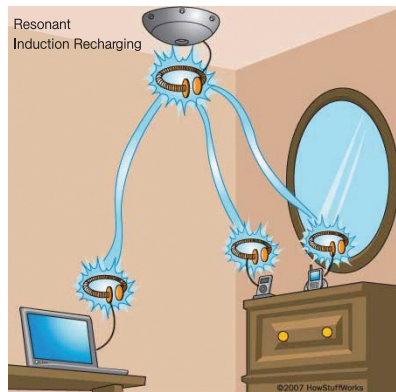


그림 1. 다수의 정보통신 기기로의 무선 에너지 공급 개념

무선 전력 전송의 개념은 약 100년 전 테슬라로부터 시작되어 현재 다양한 방식으로 산업체에서 실제 적용되고 있다. 물론, 최근의 눈부신 통신 기술의 발전에 비해서 그 변환속도는 더딘 편이었다. 가정 및 사무용 기기에서의 적용은 에너지 전송 거리상의 제약, 낮은 효율, 그리고 인체에 미치는 영향에 의해 널리 보급되고 있지 않은 편이다.

현재, 가장 실용화되어 있는 무선전력 전송 방법은 전자기 유도 커플링 (electromagnetic inductive coupling)을 이용한 방법으로 두 개의 코일 간 상호 인덕턴스 커플링을 이용하여 근접한 거리에서 이용 가능하다. 예를 들어, 면도기나 전동 칫솔과 같은 가전기기의 충전기가 이에 포함된다. 또한, 버스 및 지하철 카드에도 사용된다. 이들 카드에는 전자회로들이 내장되어 있는데, 카드가 자체 배터리가 없어도 동작

하는 것은 리더(reader)에서 전자기 유도 커플링을 통해 에너지를 공급 받아 사용하기 때문이다. 하지만 비접촉식 무선 전력 전송의 범위는 수 mm에서 수 cm과 같이 매우 근접한 거리에서만 사용이 가능하며 이는 모바일 환경에서 실질적인 무선 에너지 전송으로는 거리가 너무 제한되어 있다. 이는 기존의 무선 에너지 전송 방식에 통신에서 이용되는 방사형 방식이 그대로 사용되기 때문이다.

공진을 이용한 무선 에너지 전송 방법

최근, MIT 물리학과 Marin Soljacic 교수 팀은 기존의 무선 전력 전송과는 다른 방법으로 에너지를 전송하는 기술을 발표하였다[2]. 이는 근접 장을 이용하면서 송/수신부의 공진 주파수를 일치시켜 높은 효율로 에너지를 전달하는 방식이다. 이 방식은 공진 구조의 특성상 기존의 전자기유도 방식 보다 장거리에서 사용이 가능하고 더 높은 효율로 에너지를 전달 할 수 있다.

기존의 방사형(radiative type) 전력 전송 방식은 원거리 정보 전송에는 효율적인 방식이지만, 전력 전송으로 사용할 경우 효율이 매우 낮다. 이에 반해 비방사형(non-radiative type) 공진 커플링 현상에 바탕을 둔 무선 전력 전송 방식은 수 미터의 거리에서 고효율의 높은 전력 전송이 가능하다. 이는 전자기 유도 커플링 무선 전력 전송 방식의 좁은 동작 범위, 그리고 마이크로파 대역에서 사용되는 방사형 무선 전력 전송의 낮은 전달 전력의 단점을 해결할 수 있는 새로운 기술이다.

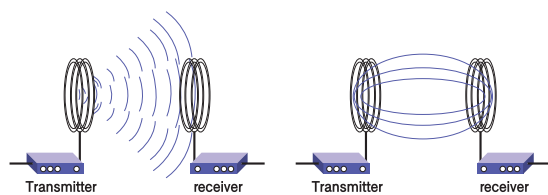


그림 2. 좌. (a) 기존의 방사형 에너지 전송방식
 우. (b) 공진을 이용한 에너지 전송방식

그림 2에 두 가지 무선 전력 전송 방식을 비교하였다. 기존의 방사형 송신 방식은 모든 방향으로 균일하게 전력이 방사되기 때문에 수신부에서 받는 전력이 거리에 따라 급격하게 감소하는 반면, 공진 커플링 방식은 같은 주파수로 공진하는 송신부와 수신부가 밀접하게 결합되어 에너지를 주고받아 높은 무선 전력 전송 효율을 나타낸다.

공진 커플링 방식이 높은 전력 전송 효율을 얻을 수 있는 이유는 송신부와 수신부가 근거리 장 내에서 비방사 영역 (reactive region)을 공유하며 안테나의 저항성 손실이나 방사 손실 속도보다 더 빠르게 커플링에 의한 전력 교환이 일어나기 때문이다. 근거리 장을 이용하는 공진 커플링 무선 전력 전송 시스템에서는 수신부와 송신부가 강한 자계 커플링에 의해 에너지 흐름이 기술되어 전자파를 이용하는 원거리 장 방사형 전력 전송 시스템과 구분된다.

또한, 기존의 HF 대역 13.56MHz RFID 시스템에서 사용하는 전자기 유도 커플링과도 전력 전송 원리를 달리한다. 기존의 방사형 방식은 주로 원거리 장 영역에서 사용되고 방사를 효율적으로 하기 위해 방사 저항 값을 크게 해서 quality factor(Q)값이 낮다. 이로 인해 낮은 전력 전송 효율을 나타내는데 비해, 비방사 공진 커플링에서는 저장된 에너지와 물리적 안테나의 크기가 같을 경우 기존의 방사형 자계 커플링 방식 보다 Q²에 비례하는 더 많은 전력을 부하로 전달할 수 있는 점이 다르다. 또한, 같은 공진 주파수를 가진 기기에만 대부분의 전력이 전달되기 때문에 주변의 다른 기계나 신체에는 영향이 크지 않다.

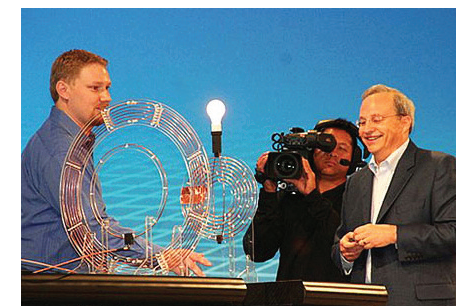


그림 3. 2008년도 Intel에서 데모한 무선전력전송 기술

이와 같은 물리적 현상에 바탕을 둔 무선 에너지 전송 방식은 높은 에너지 전송 효율을(1m 거리에서 약 40% 효율) 가지면서도 수 m에 이르는 거리까지 60W의 전력 전송이 가능함이 2007년도 Science지에 실험적으로 보고된 이후, 이 기술을 MIT에서는 2008년 10대 기술로 선정하고 지속적인 연구가 진행되고 있다. 산업체에서는 Intel사에서 2008년 Intel Developer Forum[3]에서는 동일한 기술을 "Wireless Resonant Energy Link(WREL)"라는 이름으로 개발하고 있다고 발표하였다(그림 3).

이러한 최근 추세에 따라 국내외에서 이를 이용한 모바일 기기 전력 충전 시스템 개발이 활발하다. 현재, 무선 전력 전송으로 상품화를 추진하고 있는 업체로는 필립스의 자회사인 파워캐스트(Powercast LLC)[4], 전자기유도를 이용하여 핸드폰이나 PDA, MP3 플레이어, 노트북컴퓨터까지 충전할 수 있는 새로운 충전기를 선보인 영국의 플래시파워(Splash-Power Ltd.)[5]과 풀톤 이노베이션(Fulton Innovation, LLC.)[6] 등의 회사들이 있다. 이러한, 새로운 무선 전력 전송 기술은 기술개발의 초기단계이기는 하나 그 파급효과는 매우 광범위 할 것으로 기대되고 있다.

공진을 이용한 무선 에너지 실험 결과

기존의 전자기 유도 방식의 무선전력전송 구조에서는 소오스측 저항 (R_s)과 부하 측 저항(R_L)에 낮은 값이 사용되면 전체 코일의 Q값 (load-Q factor)가 저하되어 전체 시스템의 전력전송 효율이 감소된다. 이러한 문제점을 해결하기 위해 그림 5와 같이 4개의 코일로 구성된 공진형 무선 전력 전송 시스템을 사용하고 있다. 여기서 K₁₂, K₂₃, K₃₄ 는 각각 코일-1과 코일-2, 코일-2와 코일-3, 그리고 코일-3과 코일-4의 결합계수 (coupling coefficient)를 나타낸다.

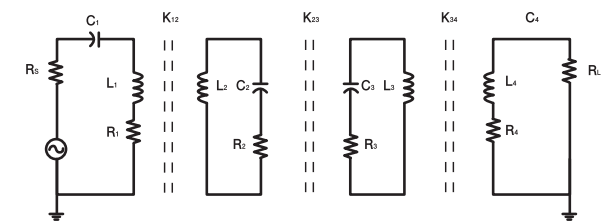


그림 4. 무선 전력 전송 시스템의 등가 회로 모델

이 구조는 소오스측과 부하 측을 기존의 시스템과 달리 4개의 코일을 이용하여 부하에 의한 loading 효과를 감소시킴으로써, 소오스측과 부하 측에 50Ω 정도의 저항이 부착되어도 공진 코일의 Q값 저하가 크지 않아 효율 저하가 상대적으로 적게 된다. 이 구조에서 사용된 코일은 각각 코일-1의 인덕턴스 값은 10.8μH, 코일-2의 인덕턴스 값은 97.297μH, 코일-3의 인덕턴스 값은 97.297μH, 코일-4의 인덕턴스 값은 10.8μH이다. 코일-1의 캐패시턴스 값은 23.45pF, 코일-2의 커패시턴스 값은 2.6pF, 코일-3의 캐패시턴스 값은 2.6pF, 코일-4의 캐패시턴스 값은 23.45pF이다.

이 구조에 대한 에너지 전송효율을 살펴보기 위해 EM 시뮬레이터를 이용하여 S-parameter값을 구하였다. 소오스와 부하 저항 값을 알고 있을 때 전력 전송 효율은 전압 전송 계수(transmission coefficient) $|S_{21}|$ 로부터 추출할 수 있다. 그림 8에 이와 같은 EM 시뮬레이션을 통해 얻은 효율을 코일 간의 거리에 대한 함수로 나타내고 있다.

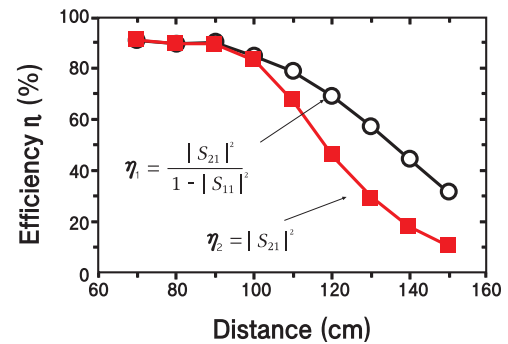


그림 5. 무선 전력 전송 시스템의 에너지 전송 효율 시뮬레이션 결과

그림 6은 제작된 무선 에너지 전송 시스템의 구조와 측정 환경을 보여주고 있다. 자기장에 영향을 미치지 않고 코일의 형태를 고정하기 위해 아크릴로 제작한 구조를 이용하였다. 코일의 반지름은 29cm이며 코일-1과 코일-4는 1번 감았으며 코일-2와 코일-3은 10번을 감았다. 코일이 가지는 간격에 의한 커패시턴스 값은 각각 80pF과 47.6pF이다.

LCR미터(HP 4263B)를 이용하여 측정된 각 코일의 인덕턴스 값은 코일-1은 15μH, 코일-2은 113μH, 코일-3은 110μH, 코일-4는 15μH의 인덕턴스 값을 가지며 각 코일의 코일 간 간격은 4mm로 고정하였다.

측정은 코일-2와 코일-3의 거리를 이동해 가며 측정을 하였다. 측정은 네트워크 분석기(HP8714ES)를 이용하여 $|S_{21}|$ 의 값을 측정하여 소신호 인가시의 효율과 공진 주파수를 확인하였다.

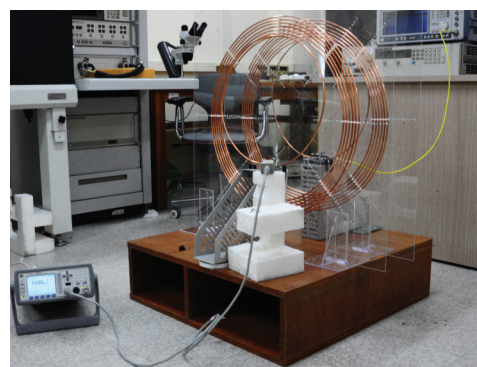


그림 6. 무선 전력 전송 시스템의 전력 전송 효율 실험 환경

실제 대신호 전력 전송 효율 측정하기 위해 신호 발생기(SMIQ06B)의 출력 주파수를 설정하고, 이 신호를 전력 증폭기를 이용하여 25dBm까지 증폭시켜 코일-1에 인가하였다. 수신 부는 코일-4에 해당하며 감쇠기를 거쳐 파워 센서(HP8481B)와 파워 미터(437B)를 이용하여 수신되는 전력을 측정하였다. 거리에 따른 측정 결과 간격 15cm에서 92.5%의 최대 전력 전송 효율, 60cm에서 효율 78%, 그리고 1m에서 32%의 효율이 측정되었다.

최근 Qualcomm을 비롯한 국내외의 회사들은 핸드폰의 무선충전 기술에 많은 연구를 진행하고 있다. 본 연구실에서는 기존에 개발된 HF 대역 RFID 태그 IC에 이러한 공진형 무선에너지 전송 기법을 적용할 수 있는지 연구하고 있다.[7~10].

그림 7에는 EEPROM과 쇼트키 다이오드 공정이 제공되는 국내 foundry의 0.18um CMOS 공정을 이용하여 설계된 13.56MHz HF 대역 RFID 태그 IC를 보여주고 있다. 칩의 크기는 1.1x1 mm²로서 상용으로 나와 있는 칩 중에 비교적 소형에 속한다.

이 태그 IC는ISO 14443-type B 프로토콜을 만족할 수 있도록 설계되었다. ISO 14443-type B 프로토콜은 10%의 변조지수를 사용하기 때문에 전력변동은 적은 편이지만 복조기의 설계가 까다로운 편이다.

본 연구실에서는 노이즈에 둔감한 opamp와 바이어스 버퍼링 기법을 도입한 새로운 복조기를 이용하여 이 문제를 해결하였다. 리더와의 안전한 데이터 교환을 위한 암호화를 위해 cyclic key generation을 이용한 128bit AES 엔진을 내장하였으며, 이러한 AES 동작을 지원하기 위해 비교적 대용량인 4Kb EEPROM을 사용하였다[11].

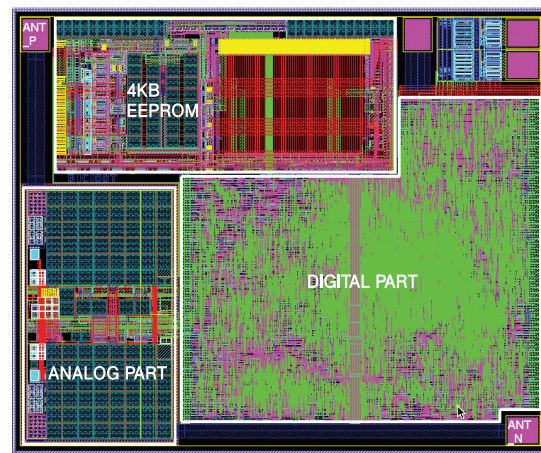


그림 7. 설계된 13.56MHz HF 대역RFID 태그 IC의 레이아웃

그림 8에는 제작된 13.56 MHz HF 대역 RFID 태그 IC 측정 환경을 보여주고 있다. 패키지 된 IC 칩에 13.56MHz 주파수에 공진하는

태그 안테나를 사용하여, 0리더와 무선 환경에서 데이터와 전력을 주고 받는지 확인하였다.

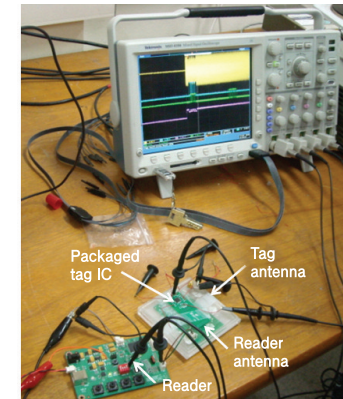


그림 8. 제작된 13.56MHz HF 대역 RFID 태그 IC 측정 환경

그림 9에 제작된 13.56MHz HF 대역 RFID 태그 IC 측정 결과를 나타내고 있다. 맨 위쪽으로부터 10% 변조된 리더로부터의 ATQB command, 리더로부터의 데이터, ATQB command에 대한 복조기 출력 및 이에 대한 응답, 그리고 태그 칩에서 정류된 DC 전압을 보여주고 있다.

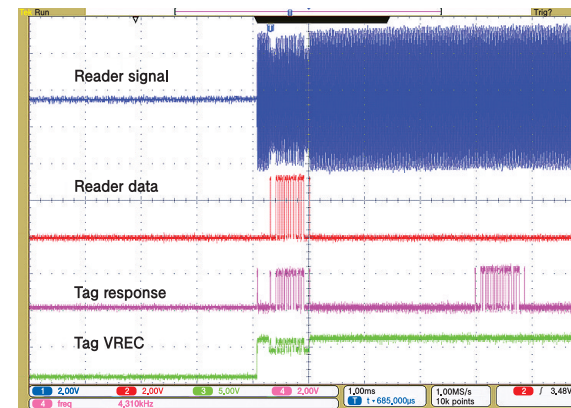


그림 9. 제작된 13.56MHz HF 대역 RFID 태그 IC 측정 결과

결론

공진을 이용한 무선 에너지 전송 기술의 높은 전력 전송 효율은 기존의 전자기 유도 커플링 기술로는 달성하기 어려운 결과이다. 이러한 새로운 무선 에너지 전송 기술은 RFID, 인체 내 바이오 기기, 가정용 충전기 등 배터리를 내장할 수 없거나 크기 상의 제약으로 소형의 배터리만을 내장할 수 있는 무선기기에 전력 공급을 위한 새로운 대안으로 떠오르고 있다.

현재, 국내외에서 발표된 공진형 에너지 무선전력 전송 기술은 수 MHz의 주파수를 사용하고 있어 기존의 HF 대역 RFID 시스템과 호환성을 유지할 수 있고, 이를 이용해 전력과 에너지를 동시에 주고 받을 수 있는 응용도 가능하다.

Reference

- [1] W. Stewart, "The Power to Set You Free," Science, Vol. 317, 2007, pp.55-56.
- [2] A. Kurs et al., "Wireless Power Transfer via Strongly Coupled Magnetic Resonances," Science, Vol. 317, 2007, pp.83-86.
- [3] Intel Developer Forum (IDF) 2008, www.intel.com/idf/
- [4] www.powercastco.com
- [5] www.splashpower.com
- [6] www.ecoupled.com
- [7] Jong-Wook Lee, Bomson Lee, and Hee-Bok Kang, "A High Sensitivity, CoSi2-Si Schottky Diode Voltage Multiplier for UHF-band Passive RFID Tag Chips," IEEE Microwave and Wireless Components Letters, vol. 50, No. 12, Dec. 2008, pp. 830-832.
- [8] Hee-Bok Kang, Jinyong Chung, and Jong-Wook Lee, "High Security FeRAM-Based EPC C1G2 UHF (860 MHz-960 MHz) Passive RFID Tag Chip," ETRI Journal, Vol. 30, no. 6, Dec. 2008, pp. 826-832.
- [9] Jong-Wook Lee and Bomson Lee, "A Long Range UHF-band Passive RFID Tag IC Based on High-Q Design Approach," IEEE Transactions on Industrial Electronics, vol. 56, No. 7, pp. 2308-2316, July 2009.
- [10] Jong-Wook Lee, Hyun Vo Duong, Quoc-Hung Huynh, and Sang-Hoon Hong, "A Fully Integrated HF-Band Passive RFID Tag IC Using 0.18 μm CMOS Technology for Low Cost Security Applications," IEEE Transactions on Industrial Electronics, to be published, 2011.
- [11] 김태훈, 이재형, 하관봉, 김영희, "저전압 DRAM용 VPP Generator 설계", 2008 반도체 학술대회.



집 트립 기 사

FreeRTOS 실시간 커널과 Core-A 프로세서(3) FreeRTOS API

Core-A 프로세서를 사용한 플랫폼에 FreeRTOS 실시간 커널을 이식하고, Core-A 개발 보드에서 실행해 본다. 첫 회에서는 FreeRTOS에 대해 살펴 보았고, 지난 회에서는 FreeRTOS를 Core-A 프로세서에 이식하는데 필요한 주요 코드들을 상세하게 살펴보았다.

이번 회에서는 FreeRTOS를 활용하여 응용 프로그램을 개발 할 때 사용하는 중요 API에 대해 정리한다.

다음 회에서는 FreeRTOS가 이식된 Core-A 개발 보드에서 몇 가지 멀티 태스킹 예제를 실행시켜 본다.

FreeRTOS API

FreeRTOS[1]의 응용프로그램인터페이스(API: application-programming interface)를 간략하게 살펴본다. FreeRTOS의 API에는 태스크 관리, 태스크 사이의 정보 전달, 태스크 사이의 동기화, 시간관리, 메모리 관리 등이 있다. 지면이 한정되어 중요한 것만 살펴본다. 보다 상세한 내용은 참고자료 [2]를 참조한다.

■ 선택적 구성

FreeRTOS는 응용 프로그램의 특성에 따라 많은 부분을 사용자가 선택적으로 값을 지정하거나 코드를 포함시키거나 배제할 수 있고, 'FreeRTOSConfig.h' 헤더 파일에서 결정한다. 한정된 지면 때문에 모든 것을 다 설명하지는 못하였다.

- configUSE_PREEMPTION: 1로 선언하면 선점형 스케줄링 기법을 사용한다.
- configCPU_CLOCK_HZ: CPU의 동작 속도를 Hz로 정의한다.
- configTICK_RATE_HZ: 틱 타이머 인터럽트가 생성되는 주파수를 Hz로 정의한다. 통상 1000 이상의 값으로 정의한다.
- configMAX_PRIORITIES: 태스크에 사용할 우선 순위의 최고 값을 정의한다.
- configMINIMAL_STACK_SIZE: 태스크에 할당할 스택의 최소 크기를 'portSTACK_TYPE' 단위로 정의하며, 아이들 태스크(idle Task)를 위해 사용한다.
- configTOTAL_HEAP_SIZE: 동적으로 관리할 메모리의 크기를 'portSTACK_TYPE' 단위로 정의한다. 이 영역을 이용하여 각 태스크에 스택을 할당한다.
- configMAX_TASK_NAME_LEN: 'xTaskCreate()'에서 지정하는 태스크 이름의 최대 글자 수를 정의한다.
- configUSE_MUTEXES: 뮤텡스 관련 API를 사용할 경우 1로 정의한다.
- configUSE_CO_ROUTINES: 코루틴을 사용할 때 1로 정의한다.
- configUSE_IDLE_HOOK: 1로 설정할 경우, 아이들 태스크가 실행 될 때 자동으로 호출되는 'vApplicationIdleHook()' 함수를 사용할 수 있게 된다.
- configUSE_TICK_HOOK: 1로 설정할 경우, 'vApplicationTickHook()' 함수를 사용할 수 있게 되며, 틱 타이머 인터럽트 처리 루틴에서 'vTaskIncrementTick()' 루틴을 호출할 때 자동으로 수행된다.
- INCLUDE_vTaskPrioritySet: 1로 선언하면 'vTaskPrioritySet()' 루틴을 사용할 수 있게 된다.
- INCLUDE_uxTaskPriorityGet: 1로 선언하면 'uxTaskPriorityGet()' 루틴을 사용할 수 있게 된다.

- INCLUDE_vTaskDelete: 1로 선언하면 'vTaskDelete()' 루틴을 사용할 수 있게 된다.
- INCLUDE_vTaskSuspend: 1로 선언하면 'vTaskSuspend()' 루틴을 사용할 수 있게 된다.
- INCLUDE_vTaskDelayUntil: 1로 선언하면 'vTaskDelayUntil()' 을 사용할 수 있게 된다.
- INCLUDE_vTaskDelay: 1로 선언하면 'vTaskDelay()' 를 사용할 수 있게 된다.

■ 시작

FreeRTOS는 'vTaskStartScheduler()' 로 멀티 태스킹을 시작하는데, 그 이전에 'vTaskCreate()' 로 사용자 태스크를 생성해야 하고 타이머 인터럽트가 생성되도록 하드웨어를 설정해 두어야 한다.

```
#include "FreeRTOS.h"
#include "task.h"
int main(void) {
    // set up hardware including enabling timer
    // create at least one task using 'xTaskCreate()'
    xTaskStartScheduler();
    return(0);
}
```

'vTaskStartScheduler()' 는 가장 낮은 우선순위의 '0' 을 갖는 아이들 태스크(idle Task)를 생성하고, 타이머의 인터럽트 개수를 관리하는 'xTickCount' 를 '0' 으로 초기화 한 후, 'xPortStartScheduler()' 를 호출하여 첫 태스크를 시작한다.

■ 커널 제어

다음 두 매크로는 프로세서의 인터럽트를 비활성화 하거나 활성화 시킨다.

```
taskDISABLE_INTERRUPTS();
tskENABLE_INTERRUPTS();
```

다음 두 매크로는 임계영역을 관리하기 위해 사용한다.

```
taskENTER_CRITICAL();
taskEXIT_CRITICAL();
```

다음 'taskYIELD()' 매크로는 태스크 수준에서 문맥 교환을 수행한다.

```
taskYIELD();
```

다음 두 루틴은 커널 스케줄러를 시작하거나 종료한다.

```
void vTaskStartScheduler(void);
void vTaskEndScheduler(void);
```

다음 루틴은 해당 루틴을 호출한 태스크를 제외한 나머지 태스크들을 모두 'Suspend' 상태로 만들거나 'Suspend' 상태를 'Ready' 상태로 만든다.

```
void vTaskSuspendALL(void);
void vTaskResumeALL(void);
```




■ 태스크 생성과 제거 그리고 관리

FreeRTOS 태스크를 생성하는 API인 'xTaskCreate()' 는 태스크 코드의 시작 번지, 태스크의 이름, 태스크의 스택 크기, 태스크에 전달할 인자, 태스크에 지정되는 우선순위, 태스크를 구별하는 핸들러를 인자로 받는다.

```
portBASE_TYPE xTaskCreate (pdTASK_CODE pvTaskCode, const char *pcName, unsignedshort usStackDepth, void *pvParameters, unsigned portBASE_TYPE uxPriority, xTaskHandle *pvCreatedTask);
```

FreeRTOS의 각 태스크에는 생성할 때 사용자가 해당 태스크에서 필요한 스택의 크기를 지정한다. 이때 크기는 'portSTACK_TYPE' 단위가 된다. 따라서 'portSTACK_TYPE' 이 4-byte word이면, 스택은 '4*usStackDepth' 바이트가 된다.

FreeRTOS의 각 태스크에는 생성할 때 사용자가 초기 우선 순위를 지정하고, 이 우선순위는 '0' 에서 'configMAX_PRIORITIES - 1' 사이의 수로 '0' 이 가장 낮은 우선 순위가 된다. 우선순위는 문맥 교환이 이루어 질 때 다음 실행 태스크를 결정하는데 사용되며, 우선 순위가 높은 태스크가 먼저 선택되고, 같은 우선 순위는 순차적으로 선택된다. 태스크의 우선 순위는 'uxTaskPriorityGet()' 과 'vTaskPrioritySet()' 으로 확인하거나 변경할 수 있다.

FreeRTOS의 각 태스크에는 생성할 때 핸들러가 생기며, 이 것을 이용하여 태스크의 여러 자원을 참조하거나 변경 또는 태스크를 제거할 수 있다.

'vTaskDelete()' 는 인자로 받은 핸들러에 해당하는 태스크를 스케줄러에서 제거하는 것이며, 인자가 'NULL' 이면 태스크 자신을 제거한다.

```
void vTaskDelete( xTaskHandle pxTaskToDelete );
```

'vTaskDelete()' 로 스케줄러에서 제거된 태스크가 실행 되는 동안 동적으로 할당 받은 메모리 등은 아이들 태스크가 정리한다. 따라서 전체 응용 프로그램이 실행되는 동안 아이들 태스크가 가깝게 수행될 수 있도록 해야 한다.

■ 태스크 사이의 정보 전달

FreeRTOS의 응용 프로그램은 여러 태스크들이 독립적인 스택으로 해당 루틴을 수행하며, 이들 태스크들이 상호 정보를 주고 받을 수 있도록 큐(queue)를 지원한다.

큐는 정해진 크기의 데이터를 정해진 수만큼 저장하며, 저장된 데이터는 FIFO(First-In First-Out, 시간적으로 우선된 것이 먼저 읽힘) 형식으로 전달된다. 큐에 저장할 데이터의 크기와 큐에 저장할 데이터의 수는 큐를 생성할 때 결정한다. 큐는 어느 태스크와도 무관하게 독립적이며, 여러 태스크가 큐에 데이터를 저장할 수 있고, 여러 태스크가 큐에서 데이터를 읽어 갈 수 있다. 큐에 저장하는 것은 꼬리(tail)에 하고, 큐에서 읽는 것은 머리(head)에서 한다.

큐에서 데이터가 없는 경우, 태스크가 'Blocked' 상태가 되고, 데이터가 추가되면 태스크가 'Running' 상태로 'switch-in' 된다. 만약 데이터를 가져가려는 태스크가 여럿인 경우, 우선 순위가 높은 태스크가 먼저 데이터를 가져가고, 같은 우선 순위에서는 오래 기다린 태스크가 데이터를 가져간다. 여기서 데이터를 가져간다는 것은 해당 태스크가 'Blocked'

상태에서 'Running' 상태가 된다는 의미이다. 큐에 데이터를 채우려는데 큐에 빈 공간이 없는 경우, 해당 태스크는 'Blocked' 상태가 되고, 빈 공간이 생기면 'Running' 상태로 'switch-in' 된다. 만약 여러 태스크가 큐에 쓸 공간이 없어 기다리는 경우, 공간이 생겼을 때 높은 우선 순위 태스크가 먼저 'switch-in' 되며, 동일 우선 순위에서는 오래 기다린 태스크가 'switch-in' 된다.

'xQueueCreate()' 는 큐를 생성하는 루틴이다.

```
xQueueHandle xQueueCreate ( unsigned portBASE_TYPE uxQueueLength, unsigned portBASE_TYPE uxItemSize );
```

'uxQueueLength' 인자는 큐에 저장할 수 있는 데이터의 개수를 지정하고, 'uxItemSize' 는 각 데이터의 크기를 바이트 단위로 지정한다. 'xQueueCreate()' 는 큐를 생성하고 핸들러를 복귀한다.

'xQueueSend()' 는 'xQueueSendToBack()' 와 동일한 것이며, 큐의 꼬리에 데이터를 저장한다. 필요에 따라 'xQueueSendToFront()' 를 사용하여 꼬리 대신 머리에 데이터를 저장할 수도 있다.

```
portBASE_TYPE xQueueSend ( xQueueHandle xQueue, const void *pvItemToQueue, portTickType xTicksToWait );
```

큐 핸들러 'xQueue' 로 특정 큐를 지정하고, 'pvItemToQueue' 는 큐에 저장한 데이터의 주소이며, 'xTicksToWait' 는 큐에 빈 공간이 없을 때 최대로 기다리는 시간을 타이머 인터럽트 수로 지정한다. 'xTicksToWait' 가 '0' 이면, 빈 공간이 없을 때 곧 바로 복귀하고, 'xTicksToWait' 가 'portMAX_DELAY' 이면 빈 공간이 생길 때까지 기다린다. 'xQueueSend()' 가 성공하면 'pdPASS' 를 복귀하고, 그렇지 않으면 'errQUEUE_FULL' 을 복귀한다.

'xQueueReceive()' 는 큐에서 데이터를 읽는 것으로 읽은 경우 해당 데이터는 큐에서 제거된다.

```
portBASE_TYPE xQueueReceive ( xQueueHandle xQueue, void *pvBuffer, portTickType xTicksToWait );
```

큐 핸들러 'xQueue' 로 특정 큐를 지정하고, 'pvBuffer' 는 큐에 읽은 데이터를 저장할 공간의 주소이며, 'xTicksToWait' 는 큐에 데이터가 없을 때 최대로 기다리는 시간을 타이머 인터럽트 수로 지정한다. 'xTicksToWait' 가 '0' 이면, 데이터가 없을 때 곧 바로 복귀하고, 'xTicksToWait' 가 'portMAX_DELAY' 이면 데이터가 생길 때까지 기다린다. 'xQueueReceive()' 가 성공하면 'pdPASS' 를 복귀하고, 그렇지 않으면 'errQUEUE_EMPTY' 을 복귀한다.

'xQueuePeek()' 는 'xQueueReceive()' 와 유사하게 큐에서 데이터를 읽지만, 읽은 데이터가 큐에 그대로 남는다.

```
portBASE_TYPE xQueuePeek ( xQueueHandle xQueue, void *pvBuffer, portTickType xTicksToWait );
```

'uxQueueMessagesWaiting()' 은 큐에 있는 데이터의 수를 복귀한다.

```
unsigned portBASE_TYPE uxQueueMessagesWaiting (const xQueueHandle xQueue);
```

■ 태스크 사이의 동기화

FreeRTOS에는 세마포와 뮤텁스를 지원하며, 전자는 실행 순서나 사건 전달 등 동기화에 주로 사용하고 후자는 공유 자원을 배타적으로 참조하는데 사용한다.

▶ Binary semaphore

'vSemaphoreCreateBinary()' 는 바이너리 세마포를 생성하는 매크로인데, 실제로는 크기가 '0' 인 데이터를 한 개 관리하는 큐를 생성한다. 이 것은 실제로 루틴이 아니므로, 인자는 포인터가 아닌 변수를 그대로 사용해야 한다.

```
void vSemaphoreCreateBinary ( xSemaphoreHandle xSemaphore );
```

처음 세마포가 생성되면 아래 설명할 'xSemaphoreGive()' 를 내부에서 호출하게 되므로, 세마포는 사용할 준비가 된다.

세마포를 확보하기 위해 'vSemaphoreTake()' 를 사용한다. 실제로는 'xQueueGenericReceive()' 를 사용하여 정의된 매크로이다.

```
portBASE_TYPE xSemaphoreTake ( xSemaphoreHandle xSemaphore, portTickType xBlockTime );
```

'xBlockTime' 은 세마포를 확보하지 못해 'Blocked' 상태로 머무를 최대 시간을 타이머 인터럽트 수로 지정한다. 'xBlockTime' 이 '0' 이면, 세마포에 관계 없이 곧 바로 복귀하고, 'xBlockTime' 이 'portMAX_DELAY' 이면 세마포가 확보 될 때까지 기다린다. 'xSemaphoreTake()' 가 성공하면 'pdPASS' 를 복귀하고, 그렇지 않으면 'pdFALSE' 을 복귀한다. 이 루틴은 인터럽트 서비스 루틴에서는 사용할 수 없다.

세마포를 넘기기 위해 'vSemaphoreGive()' 를 사용한다. 실제로는 'xQueueGenericSend()' 를 사용하여 정의된 매크로이다.

```
portBASE_TYPE xSemaphoreGive ( xSemaphoreHandle xSemaphore );
```

▶ Counting semaphore

'vSemaphoreCreateCounting()' 는 여러 세마포를 지원하는 것을 생성하며, 인자로 최대 값과 초기 값을 지정한다.

```
xSemaphoreHandle xSemaphoreCreateCounting ( unsigned portBASE_TYPE uxMaxCount, unsigned portBASE_TYPE uxInitialCount );
```

카운팅 세마포를 이용하는 것은 바이너리 세마포와 동일하게 'xSemaphoreGive()' 와 'xSemaphoreTake()' 를 사용한다.

▶ Mutex

FreeRTOS에서 뮤텁스는 바이너리 세마포의 특별한 것으로 여러 태스크 간에 상호 배타적으로 공유 자원을 참조하기 위해 사용한다. 'xSemaphoreCreateMutex()' 는 뮤텁스를 생성한다. 실제로는 'xQueueCreateMutex()' 를 호출하는 매크로이다.

```
xSemaphoreHandle xSemaphoreCreateMutex( void );
```

뮤텁스를 이용하는 것은 바이너리 세마포와 동일하게 'xSemaphoreGive()' 와 'xSemaphoreTake()' 를 사용한다.

■ 시간관리

FreeRTOS는 내부적으로 타이머 인터럽트가 발생할 때마다 1씩 증가하는 변수(xTickCount, 틱 카운터)를 이용하여 시간을 관리한다. 타이머 인터럽트는 'configTICK_RATE_HZ' 로 정의된 주파수로 발생하며, 1000 이상의 값을 주로 사용한다.

즉, 타이머 인터럽트로 실행되는 인터럽트 처리 루틴에서는 'vTaskIncrementTick()' 루틴을 호출하고, 이 루틴에서 틱 카운터를 증가시킨다.

틱 카운터 변수 'xTickCount' 는 'portTickType' 으로 그 데이터 타입이 결정되며, 이에 따라 최대 값이 결정된다. 예를 'configUSE_16_BIT_TICKS' 가 '0' 로 정의되고 'portTickType' 이 32-비트 양수 데이터 타입이면 0에서부터 0xFFFFFFFF까지 사이의 값을 갖게 된다.

다음 루틴은 틱 카운터의 값을 알려준다.

```
portTickType xTaskGetTickCount( void );
```

다음 루틴은 이 루틴을 호출한 태스크를 인자로 지정한 수만큼의 타이머 인터럽트가 생성될 때까지 'Blocking' 상태로 만든다. 따라서 'vTaskDelay()' 루틴이 언제 호출 되었는가에 따라 해당 태스크가 호출되는 주기가 결정된다.

```
void vTaskDelay( portTickType xTicksToDelay );
```

'vTaskDelay()' 는 언제 해당 태스크가 실행되었는가에 따라 다음 실행 시점이 결정된다. 즉, 반복적으로 실행되어야 할 태스크의 경우, 실행 주기를 정확하게 지킬 수 없다는 단점이 있다. 이러한 문제를 해결할 수 있도록 다음 루틴이 제공된다. 'pxPreviousWakeTime' 이 지칭하는 변수는 사용에 앞서 반드시 'xTaskGetTickCount()' 로 초기화해야 한다.

```
void vTaskDelayUntil( portTickType * const pxPreviousWakeTime, portTickType xTimeIncrement );
```

'vTaskDelay()' 와 'vTaskDelayUntil()' 에서 사용하는 시간 값은 타이머 인터럽트의 수이므로 절대 시간을 지정할 수 있도록, 'portTICK_RATE_MS' 매크로가 사용된다. 따라서 123 msec가 필요하다면 '123/portTICK_RATE_MS' 로 지정할 수 있다.

■ 메모리 관리

FreeRTOS는 메모리 관리를 위해 다음과 같은 몇 가지 루틴을 사용한다. 이들은 동적으로 메모리를 할당하는 것, 더 이상 필요치 않은 경우 시스템에 되돌려 주는 것, 초기화하는 것, 동적으로 할당이 가능한 메모리의 용량을 확인하는 것이다.

```
void *pvPortMalloc( size_t xWantedSize );
void vPortFree( void *pv );
void vPortInitialiseBlocks( void );
size_t xPortGetFreeHeapSize( void );
```

이들 메모리 관리 루틴은 세 가지 종류의 동적 메모리 관리 코드로 제공하며, 이 중 하나를 선택하여 컴파일한다.

- heap_1.c: 'pvPortMalloc()' 의 기능만 구현하고, 'vPortFree()' 의 기능은 제공하지 않음
- heap_2.c: 'pvPortMalloc()' 과 'vPortFree()' 의 기능을 제공하지만 가용한 큰 공간을 하나로 묶는 기능은 없음

IDEC | 특집기사

• heap_3.c: C 라이브러리인 'malloc()' 과 'free()' 를 사용하여 'pvPortMalloc()' 과 'pvPortFree()' 의 기능을 제공함

'heap_1.c' 와 'heap_2.c' 는 'configTOTAL_HEAP_SIZE' 로 정하는 크기의 메모리를 어레이로 만들어 사용한다.

■ Hook 루틴들

'configUSE_TICK_HOOK' 매크로를 '1' 로 선언하고, 다음 루틴을 정의하면 매 타이머 인터럽트가 발생할 때마다 해당 루틴이 호출 된다. 실제로는 'vTaskIncrementTick()' 에서 다음 루틴을 호출한다.

```
void vApplicationTickHook ( void );
```

'configUSE_IDLE_HOOK' 매크로를 '1' 로 선언하고, 다음 루틴을 정의하면 아이들 태스크가 실행될 때 해당 루틴이 실행된다.

```
void vApplicationIdleHook ( void );
```

'configUSE_MALLOC_FAILED_HOOK' 매크로를 '1' 로 선언하고, 다음 루틴을 정의하면 동적 메모리 할당에 문제가 발생하면 해당 루틴이 실행된다.

```
void vApplicationMallocFailedHook ( void );
```

태스크는 독자적인 스택을 사용하고, 태스크가 생성될 때 스택의 크기가 결정되므로 태스크가 수행되면서 스택 영역을 벗어나게 되면 전체 응용 프로그램에 영향을 미친다. 따라서 동적으로 변하는 스택의 상태를 점검함으로써 응용 프로그램의 안정성을 점검할 수 있다.

이를 위해 'INCLUDE_uxTaskGetStackHighWaterMark' 매크로를 '1' 로 선언하면, 다음 루틴을 사용할 수 있다. 이 루틴이 호출 될 때마다 'xTask' 핸들러가 지칭하는 태스크의 스택에 여유 공간이 얼마인지 알려 준다. 즉, 값이 '0' 에 가까울수록 스택에 공간이 적게 남아 있다는 뜻이다.

```
unsigned portBASE_TYPE uxTaskGetStackHighWaterMark ( xTaskHandle xTask );
```

태스크가 스택을 할당 받은 것 보다 많이 사용하는 경우, 자동으로 특정 루틴을 호출하도록 설정할 수 있다. 'configCHECK_FOR_STACK_OVERFLOW' 매크로를 1 또는 2로 선언하면, 스택 넘침(stack overflow) 현상이 발생할 때 다음 루틴이 자동으로 호출 된다. 다음 루틴의 내용은 사용자가 제공한다.

```
void vApplicationStackOverflowHook ( xTaskHandle *pxTask, signed char *pcTaskName );
```

'configCHECK_FOR_STACK_OVERFLOW' 매크로가 '1' 이면 해당 태스크의 문맥이 저장 될 때마다 스택 상태가 점검되고, 만약 스택 넘침 현상이 생기면 위 루틴이 실행된다. 이 방법의 단점은 태스크가 실행되는 과정에 스택을 많이 사용하여 발생하는 스택 넘침 현상은 알 수가 없게 된다.

'configCHECK_FOR_STACK_OVERFLOW' 매크로가 '2' 이면 처음 스택이 만들어 질 때 스택의 최하위 20 바이트에 미리 특정한 값을 기록하고, 이 영역의 값이 변하면 스택 넘침 현상이 발생한 것으로 간주하여 위 루틴이 실행된다.

3회를 마치며

FreeRTOS에는 태스크 관리, 큐 관리, 태스크 사이의 정보 전달과 동기화, 메모리 관리 등을 위한 API가 지원된다. 여기서는 다루지 않았지만 코루틴(co-routine)을 위한 API도 있다. 특히 후크(hook) 함수를 통해 일정한 사건 또는 조건에 사용자가 지정한 함수를 수행하게 지정할 수도 있고, 대부분의 API를 선택적으로 사용할 수 있어서 코드의 크기를 최적화할 수 있다는 장점이 있다.

다음 회에서는 FreeRTOS가 이식된 Core-A 개발 보드에서 응용 프로그램을 실행해 본다.

- 1 'void prvIdleTask()'
- 2 'FreeRTOSConfig.h' 에서 사용자가 정한다.
- 3 'portmacro.h' 에서 정의한다.
- 4 'portmacro.h' 에서 정의한다.
- 5 'portmacro.h' 에서 정의한다.
- 6 1Khz이며, 각 타이머 인터럽트는 1msec 마다 발생한다는 의미이다.
- 7 FreeRTOSConfig.h
- 8 Core-A의 스택은 높은 주소에서 낮은 주소로 증가한다.

저작권과 라이선싱

- Core-A는 무상으로 사용할 수 있지만, Core-A 개발자와 협약을 맺어야 한다. (www.Core-A.com)
- FreeRTOS는 공개코드 소프트웨어이며 무상으로 상용 제품에도 사용할 수 있다. (www.FreeRTOS.org)
- 본고에서 다루는 내용과 코드 중, FreeRTOS를 Core-A에 이식하기 위해 추가된 FreeRTOS와 분리된 코드는 교육용인 경우에 한하여 자유롭게 사용할 수 있지만, 정부과제나 기업과제 그리고 제품에 사용할 경우 개발자와 협의하여야 한다. (www.Dynalith.com, corea@Dynalith.com)

Reference

- [1] FreeRTOS Official Web-site, http://www.freertos.org.
- [2] Richard Barry, Using the FreeRTOS Real Time Kernel - A practical guide, Version 1.0.5, 2009.
- [3] Richard Barry, FreeRTOS Reference Manual - API Functions and Configuration Options.



(주)다이나리시스템
 기안도 박사
 연구분야 : 시스템집적반도체 설계와 검증
 E-mail : adki@dynalith
 http://dynalith.com

MPW(Multi-Project Wafer) Design Contest

2011 IDEC MPW 설계공모전

IDEC MPW 설계공모전을 통하여
 자신이 설계한 IC를 국내 최고의 Foundry 업체에서 제작할 수 있습니다.
 여러분이 주인공이 되어 생각을 현실로 구현해 보십시오.

2011년 MPW 공정 지원 내역

공정지원사	공정(μm)	공정내역	size	칩수	Package
삼성	0.13μm	CMOS 1-poly 6-metal	4mm x 4mm	96	208pin QFP
	65nm(*)	CMOS 1-poly 8-metal (RF차단, Open/ERL, Inductor, MIM)	미정	20	미정
메그나립/하이닉스	0.35μm	CMOS 2-poly 4-metal (Optional layer: QNW, HRI, BJT, QDOR 추가)	5mm x 4mm	40	Design 144pin Package 208pin QFP
	0.18μm	CMOS 1-poly 6-metal (6metal은 Thick metal 사용가능 / Optional layer: QNW, HRI, BJT, MIM 추가)	4.5mm x 4mm	80	Design 200pin Package 208pin QFP
동부하이텍	0.13μm	CMOS 1-poly 6-metal (RFCMOS, Top : UTM)	5mm x 5mm	23	208pin QFP
	0.11μm(*)	CMOS 1-poly 6-metal (RFCMOS, Top : UTM)	5mm x 5mm	13	208pin QFP
	0.35μm BCDMOS	CMOS 2-poly 4-metal	5mm x 5mm	15	144pin QFP
TowerJazz	0.18μm CIS	CMOS 1-poly 4-metal	5mm x 5mm	2	지원하지 않음
	0.18μm(*) BCDMOS	CMOS 1-poly 3-metal(MT)	5mm x 5mm	2	
	0.18μm RFCMOS	RF CMOS 1-poly 6-metal	5mm x 5mm	4	
	0.18μm SiGe	SiGe BiCOMOS 1-poly 6-metal	5mm x 5mm	1	
KEC	0.5μm	CMOS 1-poly 2-metal	3mm x 3mm	20	28pin ceramic
	4μm BJT	BJT 1-poly 2-metal	3mm x 3mm	20	28pin ceramic

2011년 MPW 진행 일정

구분	공정사	공정	제작칩수	우선모집		정규모집		후기	DB미감 (Tape-Out)	DB전달 (Fab-In)	Chip-out	Package ~out
				신청마감	선정발표	신청마감	선정발표					
98회 (11-1)	TowerJazz	0.18μm (SiGe)	1			10,10,31	10,11,15		11,02,11	11,02,22	11,05,24	11,06,24
	메그나립/하이닉스	0.35μm	20			10,10,31	10,11,15		11,02,11	11,02,25	11,06,07	11,07,07
	메그나립/하이닉스	0.18μm	20			10,10,31	10,11,15		11,02,11	11,02,25	11,06,07	11,07,07
	TowerJazz	0.18μm(RF)	2			10,11,20	10,12,06		11,02,21	11,03,07	11,05,11	-
99회 (11-2)	동부	0.35μm (BCDMOS)	3			10,11,20	10,12,06		11,02,22	11,03,09	11,06,01	11,07,01
	동부	0.13μm	23			10,11,20	10,12,06		11,03,21	11,04,06	11,07,06	11,08,06
	동부	0.35μm (BCDMOS)	3			10,12,05	10,12,20		11,04,07	11,04,20	11,07,12	11,08,12
100회 (11-3)	삼성	0.13μm	48			10,12,05	10,12,20		11,04,08	11,04,29	11,08,10	11,09,10
	TowerJazz	0.18μm(RF)	2			10,12,15	10,12,30	11,02,01	11,05,02	11,05,16	11,07,19	-
101회 (11-4)	메그나립/하이닉스	0.18μm	20			10,12,15	10,12,30	11,02,01	11,05,13	11,05,27	11,09,05	11,10,05
	동부	0.35μm (BCDMOS)	3			11,01,05	11,01,20	11,03,15	11,05,25	11,06,08	11,08,31	11,09,30
102회 (11-5)	TowerJazz	0.18μm(CIS)	1			11,02,06	11,02,20	11,04,06	11,06,21	11,07,05	11,09,21	-
	동부	0.35μm (BCDMOS)	3			11,02,06	11,02,20	11,04,06	11,07,07	11,07,20	11,10,12	11,11,12
104회 (11-7)	TowerJazz	0.18μm (BCDMOS)	2	10,12,05	10,12,20	11,03,02	11,03,15	11,05,06	11,08,29	11,09,14	11,11,15	-
	메그나립/하이닉스	0.35μm	20			11,03,02	11,03,15	11,05,06	11,08,29	11,09,14	11,12,20	12,01,20
	메그나립/하이닉스	0.18μm	20			11,03,02	11,03,15	11,05,06	11,08,29	11,09,14	11,12,20	12,01,20
	삼성	65nm	20			11,03,02	11,03,15	11,05,06	11,08,19	11,09,05	12,01,15	12,02,15
105회 (11-8)	TowerJazz	0.18μm(CIS)	1			11,04,15	11,04,29	11,07,15	11,10,10	11,10,24	12,01,11	-
	동부	0.11μm	13			11,04,15	11,04,29	11,07,15	11,10,10	11,10,26	12,01,10	12,02,10
	동부	0.35μm (BCDMOS)	3			11,04,15	11,04,29	11,07,15	11,10,13	11,10,26	12,01,18	12,02,18
	KEC	4μm	20	11,01,20	11,02,08	11,04,15	11,04,29	11,07,15	11,10,14	11,10,28	12,01,27	12,02,27
106회 (11-9)	KEC	0.5μm	20			11,04,15	11,04,29	11,07,15	11,10,14	11,10,28	12,01,27	12,02,27
	삼성	0.13μm	48			11,05,15	11,05,30	11,08,15	11,11,05	11,11,25	12,03,09	12,04,09
107회 (11-10)	메그나립/하이닉스	0.18μm	20			11,06,30	11,07,15	11,08,31	11,12,15	11,12,30	12,04,10	12,05,10

참여 대상 : IDEC Working Group(WG) 대학의 학부생 및 대학원생

- (*)는 2011 새롭게 지원되는 공정임.
 - 지원 공정 내역 및 일정은 사정에 따라 변경이 될 수 있음.
 - 우선모집 / 정규모집의 참가신청일은 50:50으로 한정하며, 후기모집은 정규모집 미달시에만 실시함.
 - 공정별 라이브러리 배포와 기술 지원은 IDEC에서 수행함.
 - 설계설명서는 정규모집을 기준으로 개최하며, 일정은 호환이후 공지함.
 - MPW 설계공모전은 지식경제부의 반도체 설계인력양성사업의 일환으로 참여기업의 지원금 없이 수행함.
 - 자세한 공정내역 및 신청방법은 IDEC 홈페이지 (http://idec.org) 가 참조.
 - 문의처 : 이의숙 (Tel. 012-350-4428, E-mail: yslee@idec.kaist.ac.kr)