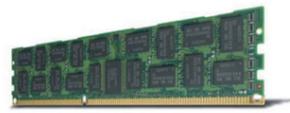


Less energy. More speed.



The new 30 nano class Green DDR3

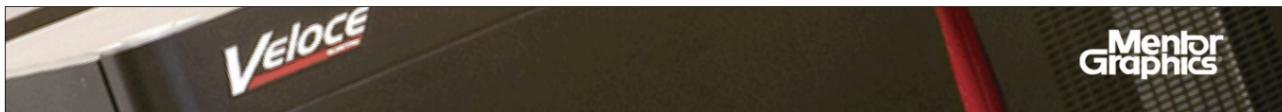
Samsung's 30 nano class 4G bit DDR3 server memory chip is the most advanced, best-performing chip we've ever created. It saves 86% more energy, processes two times faster and is far more reliable than its predecessor.* In fact, its energy usage is so small, operating and maintenance costs of your server farm are significantly reduced. Welcome the eco-innovation that doesn't compromise performance – just one more reason the leader in green memory technology is Samsung.

www.samsung.com/greenmemory



Planet First

© 2011 Samsung Electronics Co. Ltd.
* Samsung internal test result, compared to Samsung 60 nano class DDR3 memory chip. Actual performance difference may vary depending on the test environment.



Veloce 2



New custom emulation chip



Veloce2 Quattro

From the Custom Emulation Chip to Veloce 2 Enterprise Emulation Server with capacity up to 2B gates

Highest ROI of all Hardware Emulation Platforms Speeding to First Pass Silicon Success

- Veloce2 is four times more productive and competitive
- Single platform for Acceleration & Emulation
- Total Verification of HW, SW, and peripheral interfaces
- Codelink multiplies SW debug seats
- Boosts productivity, shortens system integration time
- Increases efficiency for post-silicon debug
- Founded on standards - SystemVerilog, DPI, SCE-MI2.0, SystemC, TLM, OVM/UVM, ABV
- Veloce2 is the Emulator of choice for Multimedia, Wireless, Embedded Systems, and Networking applications.
- Power, footprint deliver low cost of ownership

한국멘토그래픽스 | 서울특별시 강남구 삼성동 무역센터 무역회관 2104호 Tel. 02-551-3434 www.mentorkr.com



IDEC Newsletter

IDEC Newsletter | 통권: 제187호 | 발행일: 2013년 12월 31일 | 발행처: 반도체 설계교육센터 | 편집인: 김이섭 | 제작: 푸물디자인
기획: 전항기 | 전화: 042) 350-8535 | 팩스: 042) 350-8640 | 홈페이지: http://idec.or.kr
E-mail: jing092@idec.or.kr | 발행처: 반도체 설계교육센터(IDEC)

Vol.187

2013 January

2013 신년사 | 04 OpenCL (Open Compute Language): 이중 병렬 프로그래밍 프레임워크의 표준 | 06 반도체 메모리(DRAM) core 구조 및 특성 | 10 AUTOSAR의 "전화" 측면: ECU 리소스 템플릿 사용 사례 | 14 MPW 후기담: 프라이머리 디자인과 함께한 MPW 117회 Layout 후기 | 18

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

2013 신년사

IDEC은 우리나라 반도체 설계 환경 인프라를 구축하고 반도체 설계인력을 양성하기 위하여 1995년에 설립된 이래로 지난 17년 동안 꾸준히 발전해 왔다. 이제 IDEC은 70개의 대학 워킹그룹, 7개의 지역센터와 3개의 플랫폼 센터를 운영하면서 매년 300여 개의 반도체 칩 제작 기회와 다양한 EDA 설계 도구를 대학교에 지원하는 반도체 설계 교육의 허브가 되었다. 2013년을 맞이하여 IDEC이 우리나라의 시스템 반도체, 더 나아가 스마트 시스템 분야의 전문 인력을 양성하는 중심이 되기 위한 포부를 취임한지 1주년이 되는 박인철 소장의 신년사를 통해 살펴보고자 한다. (관련기사 P04~05 참조)

OpenCL (Open Compute Language): 이중 병렬 프로그래밍 프레임워크의 표준

최근 시스템의 키워드는 단순히 멀티코어, 매니코어를 넘어서서, 이중 컴퓨팅이라고 할 수 있을 것이다. 이 때 서로 다른 프로세서에서 생산적이고 효율적으로 프로그래밍을 할 수 있도록 Khronos 그룹에서 제정한 업계 표준인 OpenCL (Open Compute Language)가 있다. 본 고에서는 이중 병렬 프로그래밍 프레임워크의 표준인 OpenCL에 대해서 살펴보고자 한다. (관련기사 P06~09 참조)

반도체 메모리(DRAM) core 구조 및 특성

DRAM(Dynamic Random Access Memory)은 1970년대에 발명된 이후로 컴퓨터 및 모바일 제품에 없어서는 안 될 중요한 요소가 되었다. JEDEC에 표준화된 DDR3 DRAM 제품은 이미 상용화 되었으며, 수년 내에 DDR4가 상용화될 것이다. 이와 같이 DRAM의 집적도는 4Gbit~16Gbit 에 달하고 향후 더욱 증가될 것이다. 본고에서는 DRAM 코어를 구성하는 코어구조의 설계방식에 대한 noise의 영향 및 공정기술과 연관된 chip size의 영향에 대해 알아보고자 한다. (관련기사 P10~13 참조)

AUTOSAR의 "전화" 측면: ECU 리소스 템플릿 사용 사례

최근 몇 년간 AUTOSAR는 자동차에 사용되는 소프트웨어 아키텍처를 기술하기 위한 표준으로 확실히 자리를 잡게 되었다. 사용자들은 이 표준의 본래 관점을 확장하고 "가상 기능 버스" 개념을 개발함으로써, AUTOSAR 방법론을 채택하여 가상 전기 아키텍처의 일관된 개발을 위한 전체론적 E/E 접근 방식을 구현할 수 있다. 따라서 전기적으로 관련된 모델을 이용하여 논리적 설계는 물론 궁극적으로 와이어링 하네스까지 생성할 수 있다. 본 고에서는 (주)한국멘토를 통해 AUTOSAR의 "전화" 측면: ECU 리소스 템플릿 사용 사례에 대해 알아보고자 한다. (관련기사 P14~16 참조)

MPW 후기담 : 프라이머리 디자인과 함께한 MPW 117회 Layout 후기

칩 제작 실무인력양성을 위해 IDEC은 국내외 Foundry업체와 긴밀한 관계를 통해 최신공정의 MPW 칩 제작 기회를 지원하고 있다. 이에 MPW 칩 제작에 참여한 학생들의 후기담을 통해 참여 예정인 후배들에게 경험과 정보를 제공하고자 한다. 본 고에서 MPW 117회차에 참여한 부경대학교 남정훈 석사과정 학생을 통해 경험담을 듣고자 한다. (관련기사 P18~19 참조)

IDEC January | 2013 news

2013년 1월-2월 교육프로그램 안내

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

센터별 강좌 일정 |

센터명	강의일자	강의제목	분류
KAIST IDEC	1월 21일-22일	Linux Administration과정 및 Linux보안과정	설계
	1월 23일-25일	CMOS 아날로그 회로 설계 및 실습	설계
	1월 28일-30일	Mentor-Calibre Using nmDRC and nmLVS	Tool
	1월 31일-2월 1일	Mentor-Calibre xRC	Tool
충북대 IDEC	1월 2일-8일	Full custom IC 설계 입문	Tool

[수강대상]
· Calibre DRC, LVS user
[강의수준] [강의형태]
· 초급 · 이론+실습

[사전지식, 선수과목]
· Basics 내용이 포함 되어있기 때문에 처음 Tool을 사용하시는 분도 가능 합니다.

>KAIST IDEC 개설 강좌 안내

- 강좌일 : 1월 21일-22일
- 강좌 제목 : Linux Administration과정 및 Linux보안과정
- 강사 : 송승우 대표 (크리시스)

[강좌개요]
Linux install & Backup & Management에 관련 전반적인 이해를 돕고 보안 설정에 대해 이해를 높인다.

[수강대상]
· 리눅스관리자
[강의수준] [강의형태]
· 초급 · 이론+실습

[사전지식, 선수과목]
· 리눅스 기본 cmd와 OS설치경험이 있다면 쉽게 접할 수 있음

- 강좌일 : 1월 23일-25일
- 강좌 제목 : CMOS 아날로그 회로 설계 및 실습
- 강사 : 이강윤 교수 (성균관대)

[강좌개요]
아날로그 증폭기 및 Op-Amp 의 이론을 배우고 실습을 통해서 특성을 파악한다.

[수강대상]
· 석사과정 신입생
[강의수준] [강의형태]
· 초급 · 이론+실습

[사전지식, 선수과목]
· 회로이론, 전자회로 1, 2

- 강좌일 : 1월 28일-30일
- 강좌 제목 : Mentor-Calibre Using nmDRC and nmLVS
- 강사 : 박지훈 사원 (Mentor Korea)

[강좌개요]
Calibre DRC/LVS의 개념과 Tool을 효율적으로 이용하는 방법, 그리고 간단한 LVS Debugging 방법

- 강좌일 : 1월 31일-2월 1일
- 강좌 제목 : Mentor-Calibre xRC
- 강사 : 정재만 부장(Mentor Korea)

[강좌개요]
이 교육은 Parasitic 저항 및 커패시터를 추출하는 Calibre xRC의 사용법과 Rule File Generation에 대하여 교육 합니다. 다양한 Design Style에 맞는 Extraction 방법(Transistor Level Extraction, Gate-Level Extraction, Hierarchical Extraction Flow에 대하여 실습 위주로 교육 합니다.

[수강대상]
· Physical Layout 담당자, 설계 담당자, CAD 담당자를 포함한 모든 Calibre DRC/LVS를 적용하는 User
[강의수준] [강의형태]
· 중초급 · 이론+실습

[사전지식, 선수과목]
· Calibre 기본적인 수행 및 debugging 능력, Calibre Using

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

>충북대 IDEC 개설 강좌 안내

- 강좌일 : 1월 2일-8일 (토, 일 제외)
- 강좌 제목 : Full custom IC 설계 입문
- 강사 : 양병도 교수 (충북대)

[강좌개요]
1. Full custom IC 설계에 필요한 tool 사용법을 배운다.
2. Unix 명령어를 배우고, cadence를 이용하여 schematic과 layout을 만들고 simulation 하는 방법을 배운다.
3. 디지털 로직, Bandgap 기준전압 발생기, 증폭기 회로 예제를 이용하여 회로 설계 및 레이아웃을 실습한다.

[수강대상]
· 관공학과 전공 대학원생
[강의수준] [강의형태]
· 초중급 · 이론+실습

* 문의 : 충북대 지역센터 라해미 (043-261-3572, idec_haemi@naver.com)

MPW (Multi-Project Wafer)

2012년 MPW 설계 현황 |

구분	공정	제작가능 면적 (mm ² x칩수)	채택 칩수 (서버)	설계면적 (mm ² x 칩수)	DB마감	Die-out	비고
117회 (12-10)	삼성 65nm	20개서버 (4x4mm)	20	4x4mmx21	2012.12.3	2013.5.3	DB 마감 : 12.21
118회 (12-10)	M/H 0.18	4.5x4mmx20	24	4.5x4mmx16 4.5x2mmx8	2013.2.18	2013.7.22	설계설명회 : 12.27
	동부 0.35BCD	5x2.5mmx6	4	5x2.5mmx6	2013.2.27	2013.6.12	설계설명회 : 12.27
119회 (13-02)	TJ0.18 SiGe	2.5x2.5mmx4	4	2.5x2.5mmx4	2013.3.12	2013.7.1	
	동부 0.11	5x2.5mmx24	28	5x2.5mmx20 2.5x2.5mmx8	2013.3.20	2013.7.31	설계설명회 : 12.27
	삼성 65nm	4x4mmx48	26	4x4mmx26	2013.3.15	2013.8.15	설계설명회 : 12.28

* 120회 (2013-03) MPW 모집 마감 : 2012.12.30

2012년 MPW 칩제작 현황 |

구분	공정	제작 칩수	제작면적 (mm ² x칩수)	Die-out 예정일	현재상태	비고
112회 (12-5)	삼성 65nm	23	5x5mmx23	2012.11.9	Package 제작중	
	동부 0.18BCD	4	5x2.5mmx4	2012.9.26-11.14	Package 제작중	-Die chip : 12.20제작완료
113회 (12-6)	동부 0.35BCD	8	5x2.5mmx4 2.5x2.5mmx4	2012.10.10	제작완료	-Die:10.8 -PKG:10.23
	동부 0.18BCD	4	5x2.5mmx4	2012.9.26	제작중	
114회 (12-7)	M/H 0.18	20	4.5x4mmx20	2012.12.31	제작중	
	삼성 0.13	32	4x4mmx32	2013.1.25	제작중	
115회 (12-8)	동부 0.18BCD	2	5x5mmx2	2013.1.4	제작중	
	동부 0.35BCD	8	5x2.5mmx4 2.5x2.5mmx4	2013.1.16	제작중	
116회 (12-9)	동부 0.11	30	5x2.5mmx25 2.5x2.5mmx5	2013.2.6	제작중	
	TJ0.18 CIS	4	2.5x2.5mmx4	2013.2.22	제작중	
117회 (12-10)	TJ0.18 BCD	1	5x5mmx1	2013.2.29	제작중	
	TJ0.18 RF	4	2.5x2.5mmx4	2013.2.29	제작중	
117회 (12-10)	M/H 0.18	19	4.5x4mmx19	2013.3.4		
	M/H 0.35	18	5x4mmx18	2013.3.4		

* M/H = 매그나칩/하이닉스, TJ = TowerJazz
* 우선/정규모집에서 마감될 경우 후기모집은 진행되지 않음.
* Die-out 일정은 Diechip 제작완료 기준임. Package는 Die-out후 4주 이상 소요됨.
* MPW 참여 일정 및 방법은 홈페이지(www.idec.or.kr) 참조
* 위의 내용은 12.31 기준임.

* 문의 : 이의숙 (042-350-4428, yslee@idec.or.kr)

IDEC 인사동정

신규 채용 |



문 관 식

- 담당업무 : 시스템 관리, DB, 보안업무
- E-mail : mks@idec.or.kr

20th 한국반도체학술대회 Chip Design Contest 개최

1. 일정 및 장소
 - 일 시: 2013년 2월 5일(화) (*참고: KCS 일정 _ 2.4~6)
 - 장 소: 황성 웰리힐리파크 (구, 성우리조트)
2. 최종 참여팀
76팀 (Demo : 10 Panel : 66)
3. 시상 내역

Award명	대상	선정팀수	상금
Best Design Award	논문 제출 전체	1팀	100만원
Best Demo Award	데모팀 중 우수팀 시상	특별상1팀 (SSCS 서울철편타상) 우수상 1팀	각 50만원
Best Poster Award	패널참여팀 중 우수팀	2팀	20만원

* 문의 : 이의숙 (042-350-4428, yslee@idec.or.kr)

IDEC 17주년, 시스템반도체설계 고급 인재양성을 위해 맞춤형 지원 약속

IDEC은 설립 17주년을 맞은 12월 14일, 그간의 성과와 앞으로의 과제에 대해 전하며, 대한민국 시스템반도체산업을 위해 석·박사급 고급인재를 양성하는 기관으로 칩 제작 지원과 맞춤형 교육을 통한 효율성 향상을 다짐했다.

박인철 소장(IDEC)은 "IDEC은 지난 17년간 적은 예산에도 효율적인 조정을 통해 시스템반도체산업의 인재를 양성하는 뿌리기관으로 대한민국 반도체산업 발전을 위해 노력했다."라며 "빠르게 변화하는 IT 세상에 맞춰서 MPW 칩 제작, EDA Tool 지원, 맞춤형 교육 등 기업과 예비설계인들이 공감할 수 있는 지원을 위해 스스로가 발전하는 마음가짐이 필요하다."라고 IDEC 전 직원의 노력을 당부했다.

IDEC은 지난 1년간의 성과로 352개의 칩 제작, 국내·외를 포함한 저널논문 143

편과 학회논문 257편 등록, 그리고 이를 통해 332명의 설계인력을 배출하여 취업을 한 것을 꼽았다. 또한, MPW는 100회차 지원을 지나 현재 120회차 모집을 진행 중에 있다.

Message

from president

신년사



안녕하십니까?
계사년 새해가 밝았습니다.

작년에 취임사를 쓴 지가 어제 같은데 벌써 1년이 흘러서 2013년 신년사를 쓰고 있다니 감회가 새롭습니다. 지난 1년간 많은 일들이 있었습니다. 큰 일을 맡아 본 경험이 별로 없어서 서툴기도 하였고 힘들고 복잡한 일도 많았지만 많은 분들이 알게 모르게 마니또 역할을 해 주셔서 큰 탈 없이 해결해 갈 수 있었습니다. 그 동안 함께 걱정해 주시고 도움을 주신 분들께 마음 깊이 감사 드립니다.

IDEC은 우리나라 반도체 설계 환경 인프라를 구축하고 반도체 설계인력을 양성하기 위하여 1995년에 설립된 이래로 지난 17년 동안 꾸준히 발전해 왔습니다. 이제 IDEC은 70개의 대학 워킹그룹, 7개의 지역센터와 3개의 플랫폼 센터를 운영하면서 매년 300여 개의 반도체 칩 제작 기회와 다양한 EDA 설계 도구를 대학교에 지원하는 반도체 설계 교육의 허브가 되었습니다. IDEC의 성장과 더불어 우리나라 반도체 산업도 비약적인 성장을 지속해 왔으며, 최근 들어서는 시스템 반도체 분야도 급성장을 거듭하여 메모리 반도체와 더불어 우리나라 발전을 견인하는 핵심 동력이 되었습니다. 이러한 성장은 우리나라 반도체 관련 기업, 패키지 기업, EDA 벤더들, 그리고 정부 관련 기관의 지속적이고 적극적인 협조를 통하여 이루어진 것이며, IDEC도 반도체 설계 인력 양성을 통하여 이러한 발전에 이바지하였다는 자부심을 가집니다.

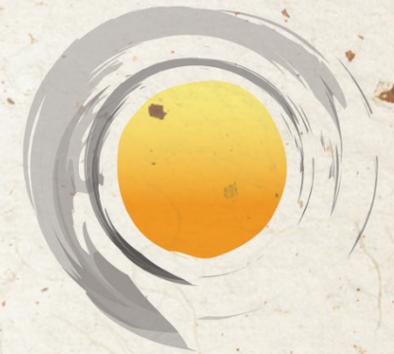
반도체 산업은 단위 부품 단계를 벗어나서 시스템 산업으로 발전하고 있습니다. 이제는 반도체 부품과 관련 소프트웨어와 구분하여 생각할 수 없으며, 반도체 칩 하나가 응용 시스템인 시대가 되었습니다. 시스템 소프트웨어와 유기적으로 연동되도록 설계 초기부터 응용 시스템을 고려해야 하며 이에 따라 반도체 설계 경향, 관련 기술, 응용 분야 등 많은 부분이 변화하고 있습니다. IDEC은 이러한 변화를 수용하여 보편적인 교육 환경은 물론 전문적인 시스템 교육 체계를 구축해 갈 것입니다. 시스템 반도체도 중요하지만 더 중요한 것은 시스템 자체이며 시스템 설계에 대한 교육 환경 구축이 필요한 시점이 되었기 때문입니다.

대학교의 반도체 설계 인프라가 구축되는데 많은 시간과 노력이 필요했습니다. 그러나 힘들게 구축된 인프라를 계속 유지하고 발전시키기 위해서는 더 많은 열정이 필요하고 여러 분야의 협력이 절실합니다. 더구나 시스템 교육을 위해서는 반도체 설계 뿐만 아니라, 시스템 소프트웨어와 응용시스템 전문 지식에 대한 교육도 포함하여야 하므로 한층 넓어지고

전문화된 교육 체계가 필요합니다. 이러한 교육 및 인프라 체계 구축을 위하여 함께 참여해 주시고 도움을 주시기 바랍니다. IDEC은 우리나라의 시스템 반도체, 더 나아가 스마트 시스템 분야의 전문 인력을 양성하는 중심이 되도록 최선을 다하겠습니다.

끝으로 웃음과 기쁨이 넘치는 새해 맞으시고 항상 건강 하시기를 기원합니다.

IDEC 소장 박인철 배상



OpenCL (Open Compute Language): 이종 병렬 프로그래밍 프레임워크의 표준



서울시립대 전자전기컴퓨터공학부
 이명민 교수
 연구분야 : Parallel System Design Methodology, System Simulation, GPU Programming
 E-mail : ymyi@uos.ac.kr
 http://parlab.uos.ac.kr

서론

최근 시스템의 키워드는 단순히 멀티코어, 매니코어를 넘어서서, 이종 컴퓨팅이라고 할 수 있을 것이다. 이 때 서로 다른 프로세서에서 생산적이고 효율적으로 프로그래밍을 할 수 있도록 Khronos 그룹에서 제정한 업계 표준인 OpenCL (Open Compute Language)[1]에 대해서 살펴보고자 한다.

본론

■ OpenCL이란 무엇인가?

저전력 시스템을 구현하기 위해, 프로세서의 동작주파수를 낮추는 대신 코어의 수를 늘리는 멀티코어/매니코어 패러다임이 시작된 지도 수 년이 지났고, 스마트폰에 들어가는 AP들도 이미 쿼드코어로 설계되고 있다. 그리고, 이러한 추세는 CPU에만 한정된 것이 아니고, 오히려 GPU (Graphics Processing Unit)에서 두드러진다.

원래 그래픽 계산과 디스플레이 처리를 담당하기 위해 설계되었던 GPU는 여러 개의 독립적인 ALU가 집적된 가속기였는데, 갈수록 집적된 ALU 혹은 GPU의 코어의 개수가 증가하여, 서버 시스템에 사용되는 GPU의 경우에는 하나의 GPU에 집적된 코어의 수가 이미 512에서 1532까지 이른대[2].

한편, 2007년부터 GPU가 범용의 프로세서처럼 활용가능해지면서[3] 많은 시스템에서 CPU와 GPU가 함께 사용되고 있다. 이러한 추세는, 서버 시스템과 임베디드 시스템에서 모두, 저전력이 성능과 더불어 중요한 요구조건이 되면서 보편화되고 있다. 저전력과 성능을 동시에 고려하면, 즉 Watt당 성능 기준으로 시스템을 고려하면, 범용의 동종 프로세서로 구성된 시스템보다 이종의 특화된 프로세서들로 구성된 시스템이 더욱 보편화될 것이다. 비단 GPU뿐만 아니라, DSP나 기타 가속기를 CPU와 함께 사용하는 시스템이 더욱 전력효율적일 것이다.

이 때, CPU와 GPU처럼 서로 다른 이종의 프로세서들을 가진 시스템에서 병렬 프로그래밍을 어떻게 해야 생산적이고 효율적일 수 있을 까라는 문제가 발생한다. 가장 먼저 GPU를 범용의 프로세서로 활용 가능하게 한 NVIDIA의 CUDA 프레임워크는 자사의 GPU만을 지원하기 때문에, 다른 제조사의 GPU를 사용하는 시스템에서는 사용할 수 없다.

OpenCL은 CPU와 GPU, 그리고 기타 프로세서들로 구성된 시스템

을 프로그래밍하기 위해 제정된 산업계의 표준 프레임워크이다. 전력 효율을 최대화하기 위해 이종 시스템들이 부상하면서, 이종 시스템에서의 병렬 프로그래밍에 대한 필요가 커졌는데, 이를 직접 다루고 있는 업계 최초의 표준이다. Apple의 주도로 2008년도 겨울에 처음 제정되어 2009년 가을에 초기 제품들이 나왔다.

많은 제조사들이 자사의 프로세서가 이 표준을 따르도록 지원하고 있다. Intel의 x86 CPU들과, IBM의 Cell/B.E, AMD의 CPU와 GPU, ARM의 GPU (Mali-T604), NVIDIA의 GPU, Imagination의 GPU가 OpenCL을 지원하고 있다. ARM의 GPU인 Mali는 삼성의 Exynos 칩에 ARM CPU (Cortex-A9)와 함께 집적되는 GPU인데, 이미 출시된 Mali 기종 중 가장 최신인 400시리즈는 OpenCL을 지원하지 않지만, 향후 출시될 T604는 2012년 8월에 OpenCL 적합성 여부를 판별하기 위해, OpenCL 표준을 관장하는 Khronos 그룹에 제출된 상태이다[4].

또한, FPGA 제조사인 Altera는 최근에 자사의 SDK에서 OpenCL을 지원하여, OpenCL 커널을 FPGA 로직으로 변환하는 기능을 제공하였다[5].

OpenCL을 사용하면, 단 하나의 프로그램을 작성해도 핸드폰으로부터 노트북, 대규모 슈퍼컴퓨터의 노드에 이르기까지 다양한 종류의 시스템에서 모두 동작시킬 수가 있다. 상당히 넓은 스펙트럼을 대상으로 표준을 정하고 있기 때문에, 이에 대해 많은 비판이 있기도 하지만, 전력효율성이 서버 시스템이나 임베디드 시스템에서 모두 중요한 기준으로 부각되고 있는 현실을 고려하면 그 이유를 알 수 있다.

이 때 OpenCL은 추상화를 통해서 하드웨어를 감추지 않고, 오히려 하드웨어를 노출시킴으로써 이식성을 제공한다. 이 말은, OpenCL 프로그래머가 플랫폼과 문맥(context), 그리고 수행할 작업들을 서로 다른 프로세서에서 어떻게 스케줄 할 것인지도 명시적으로 정의해야 한다는 말이다.

다음 절에서는 OpenCL에서 정의하는 플랫폼과 실행모델, 메모리 모델을 살펴보도록 한다.

■ OpenCL 플랫폼 모델

다음 그림은 OpenCL 프로그래밍 프레임워크에서 가정하는 플랫폼 모델이다.

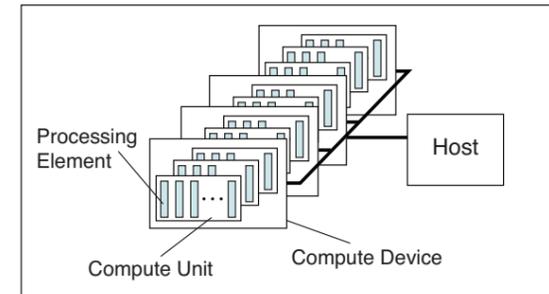


그림 1. OpenCL 플랫폼 모델

OpenCL 플랫폼 모델은 OpenCL이 사용되는 모든 이종 플랫폼을 상위 수준에서 어떻게 표현할지 정의한다. OpenCL 플랫폼은 항상 단 하나의 호스트를 가진다. 호스트는 입출력이나 프로그램 사용자 등 OpenCL 프로그램 외부의 환경과 상호작용한다. 호스트는 한 개 이상의 OpenCL 디바이스들과 연결되어 있다. 디바이스는 GPU와 같이 계산을 위한 프로그램(즉 커널)이 실행되는 곳이다. 따라서 OpenCL 디바이스는 종종 계산 디바이스라고 불린다.

디바이스는 앞서 살펴본 것처럼, CPU, GPU, DSP, 또는 기타 하드웨어로서, 서로 다른 여러 종류가 존재한다. OpenCL 디바이스는 계산 유닛 (Compute Unit)들로 이루어지고, 계산 유닛은 한 개 이상의 PE (Processing Element)들로 이루어진다. 디바이스에서의 실제적인 계산은 PE 안에서 이루어진다.

■ OpenCL 실행 모델

OpenCL 응용은 호스트 프로그램과 한 개 이상의 커널로 구성된다. 호스트 프로그램은 호스트에서 실행되는 프로그램으로서, OpenCL은 호스트 프로그램에 대한 특별한 규정은 없고, 단지 호스트 프로그램이 OpenCL 안에서 정의된 객체와 어떻게 상호 연동해야 하는지를 정의하고 있다. 커널은 OpenCL 디바이스에서 실행되는 계산 프로그램으로서, 실제적인 작업을 수행한다. 커널은 OpenCL 디바이스의 모든 PE들이 병렬적으로 수행하는 프로그램이다. 일반적으로 커널은 입력 메모리 객체를 출력 메모리 객체로 변환하는 (상대적으로 계산 시간이 오래 걸리는) 함수로 이해될 수 있다.

호스트와 OpenCL 디바이스는 서로 다른 종류의 프로세서일 수 있

기 때문에, 커널은 호스트 프로그램과 다른 명령어 집합(ISA)으로 구성된 프로그램으로서, 호스트 프로그램에서 동적으로 빌드하고, 빌드가 된 커널 실행파일은 OpenCL 디바이스로 다운로드한다.

이 때, OpenCL 실행 모델은 커널이 어떻게 실행되는지를 정의한다. 호스트 프로그램이 OpenCL 디바이스에서 커널을 실행하라는 명령을 내리면, OpenCL 런타임은 정수로 이루어진 인덱스 공간을 만들고, 커널의 각 인스턴스(한 PE에 할당된 부분)는 이 인덱스 공간의 각 점에 해당되어 실행된다. 실행되는 커널의 각 인스턴스를 작업-아이템이라고 하고, 인덱스 공간에서의 좌표로 구분되는데, 이 좌표가 작업-아이템의 전역 ID이다. 따라서, 데이터 병렬성을 지닌 함수를 커널로 구현할 때, 각 데이터에 대한 인덱스를 작업-아이템의 전역 ID로 매핑하여 수행시킨다. 각 작업-아이템은 하나의 커널에서 정의된 동일한 일련의 명령어들을 수행한다. 이 명령어들은 동일하지만, 코드 내에 분기문이 있을 수도 있고 전역 ID를 통해서 데이터가 선택되기도 하므로 각 작업-아이템의 실행은 서로 다르게 이루어질 수 있다.

커널은 작업-아이템의 수로만 정의되어 병렬 수행되지 않고, 대부분의 경우, 여러 개의 작업-아이템들을 하나의 단위로 묶어서 계층적인 병렬 수행이 이루어진다. 이와 같은 단위를 작업-그룹이라고 한다. 각 작업-그룹들은 모두 동일한 수의 작업-아이템을 가진다.

주어진 한 작업-그룹의 작업-아이템들은 한 계산 유닛에 들어있는 여러 개의 PE들 위에서 병렬적으로 실행된다. 이 정의가 OpenCL의 병렬성을 이해하는데 핵심적인 부분이다. OpenCL은 오로지 작업-그룹 안의 작업-아이템들이 병렬적으로 실행되어야 한다는 것만을 규정하고 있다. 따라서, 각 제조사에서 자사의 프로세서(디바이스)를 위해 지원하는 OpenCL 구현에 따라, 여러 커널들은 순차적으로 실행될 수 있고, 심지어 한 커널 안의 작업-그룹들도 순차적으로 실행될 수 있다. 실제로 대부분의 경우 작업-그룹들이 병렬적으로 실행되지만, OpenCL 표준에서는 이를 강제하고 있지 않다는 것을 기억할 필요가 있다.

■ OpenCL 문맥

CUDA와 달리, OpenCL은 특정 GPU뿐만 아니라, 여러 종류의 OpenCL 디바이스들이 시스템에 함께 존재할 수 있다는 가정을 하고, 이를 모두 활용하기 위해 제정된 프레임워크이다. 이 때, OpenCL 응용마다 해당 응용에게 필요한 OpenCL 디바이스들만 선

택하여 커널을 수행할 수 있는데, 이와 같이 커널과 이를 수행할 시스템의 디바이스 집합을 정의한 환경을 문맥이라고 한다.

문맥은 호스트에서 정의한다. 예를 들어, 어떤 이종 플랫폼에 두 개의 멀티코어 CPU들과 하나의 GPU를 가지고 있다고 가정하자. 호스트 프로그램은 둘 중의 하나의 CPU에서 실행되면서, 시스템에 대한 질의를 수행하여 이 자원들을 발견하고 어떤 디바이스를 OpenCL 응용을 위해 사용할지 결정한다. 주어진 응용과 실행될 커널에 따라, 호스트는 GPU를 선택할 수도 있고, 나머지 CPU를 선택할 수도 있다. 혹은 현재 호스트가 실행 중인 동일 CPU에 있는 다른 코어를 선택할 수도 있고, 이들 조합 중 하나를 선택할 수도 있다.

문맥은, OpenCL 응용이 사용할 디바이스에 대해서도 정의하지만, 한 개 이상의 프로그램 객체를 정의한다. 프로그램 객체는 커널을 위한 코드를 가지고 있어서, 앞서 설명한 것처럼, 동적으로 커널을 빌드하는데 사용된다. 또한, 메모리 객체도 정의한다. 이종 플랫폼에서는 대부분 여러 개의 주소 공간이 존재하기 때문에, 호스트와 디바이스가 동일 메모리 주소공간에서 입력과 결과를 손쉽게 공유하는 것이 불가능하다. 또한 디바이스들은 메모리 구조가 매우 다를 수 있다. 이 상황을 해결하기 위해서, OpenCL은 메모리 객체라는 개념을 도입했다. 메모리 객체는 호스트에서 명시적으로 정의되고 호스트와 OpenCL 디바이스 사이를 명시적으로 이동한다.

■ OpenCL 명령-큐

호스트와 OpenCL 디바이스 사이의 상호연동은 호스트가 명령-큐에 명령을 내림으로써 이루어진다. 이 명령은 OpenCL 디바이스에 의해 실행될 때까지 명령-큐에서 기다리게 된다. 명령-큐는 호스트에 의해 생성되고, 문맥이 정의된 후 OpenCL 디바이스마다 하나씩 연결된다. 호스트는 명령을 명령-큐에 담고, 명령은 연관된 디바이스에서 실행되기 위해 스케줄된다. OpenCL은 세 가지 종류의 명령을 지원한다.

- 커널 실행 명령은 OpenCL 디바이스의 PE에서 커널을 실행하게 만든다.
- 메모리 명령은 호스트와 다른 메모리 객체 사이에서의 데이터 전송을 발생시키고, 메모리 객체들 사이에서 데이터를 이동시키며, 호스트 주소 공간에 메모리 객체를 매핑시키거나 매핑을 해제시킨다.
- 동기 명령은 명령들이 실행되는 순서에 제한을 가한다.

호스트 프로그램에서 문맥과 명령-큐를 정의하고, 메모리 객체와 프로그램 객체를 정의하고 나면, 커널에서 인자로 사용할 메모리 객체들에 대해 메모리 명령을 명령-큐에 삽입한다. 그리고, 커널 실행 명령을 명령-큐에 삽입한다. 이 때, 여러 개의 커널들이 큐에 들어가면, 서로 상호 연동해야 하는 경우도 있다.

예를 들어, 어떤 커널들이 메모리 객체들을 생성하는데, 이를 다음의 커널들이 입력으로 받아들이야 하는 경우 경우, 동기 명령을 사용함으로써 이전의 커널들이, 후속 커널들이 시작되기 이전에 반드시 실행을 완료하도록 강제할 수 있다.

단일 큐 안에 명령들은 다음의 두 모드에서 서로 상대적으로 실행된다.

- **순차 실행:** 명령들은 명령-큐에 존재하는 순서대로 개시되고 역시 순차적으로 완료한다. 다시 말해, 큐에서 앞선 명령은 다음 명령이 시작하기 전에 완료가 된다. 따라서 하나의 큐에 있는 명령들은 모두 순차적으로 실행된다.
- **비순차 실행:** 명령들은 순차적으로 개시되지만, 실행이 끝나기를 기다리지 않고 다음의 명령들이 실행된다. 순서에 대한 제약을 주려면, 명시적인 동기 메커니즘을 통해서 프로그래머가 강제해야 한다.

모든 OpenCL 플랫폼들은 순차 실행 모드를 지원하지만, 비순차 실행 모드는 추가적인 옵션이다. 언제 비순차 실행이 필요할까? 부하조절에 대한 개념을 도시했던 그림 1.5를 다시 살펴보면 하자. 응용 프로그램은 다른 모든 커널의 실행이 완료되어야 종료된다. 따라서, 실행시간을 최소화하는 효율적인 프로그램을 작성하려면, 모든 계산 유닛들이 빠르게 동작하면서 모두 엇비슷한 시간에 실행이 끝나야 한다. 순차 실행에서 부하조절을 잘 하기 위해서는, 집어넣는 순서를 잘 고민해서 명령을 큐에 넣어야 한다.

하지만, 서로 다른 실행시간을 가지는 명령들이 다수 있을 때 모든 계산 유닛들이 모두 빠르게 동작하면서 거의 비슷한 시간에 실행이 완료되도록 부하를 잘 분배하는 것은 어려운 일이다. 비순차 큐라면 이 문제를 여러분 대신 해결해 줄 수 있다. 명령은 어떤 순서로든 실행될 수 있어서, 어떤 계산 유닛이 일찍 실행을 마치면, 그 즉시 새로운 명령을 명령-큐에서 가져와서 새로운 커널을 바로 시작할 수 있다. 이것을 자동 부하조절이라고 하고, 명령-큐를 사용하는 병렬 알고리즘을 설계할 때 널리 쓰이는 기법이다 (T. G. Mattson 외 저자들이 쓴 Patterns for Parallel Programming에서 주인-일꾼 (Master-Worker) 패턴을 살펴보기 바란다).

■ OpenCL 메모리 모델

OpenCL 메모리 모델은 5가지 서로 다른 메모리 영역을 정의한다.

- **호스트 메모리:** 이 메모리 영역은 호스트에게만 보인다. 호스트와 관련된 나머지 것들과 마찬가지로, OpenCL은 단지 호스트 메모리가 OpenCL 객체와 구문 사이에서 어떻게 동작하는지만 정의한다.
- **전역 메모리:** 이 메모리 영역은 모든 작업-그룹의 작업-아이템들이 읽고 쓸 수 있는 공간이다. 작업-아이템은 전역 메모리의 어떤 메모리 객체든지 읽고 쓸 수 있다. 디바이스에 따라 전역 메모리에 대한 읽기와 쓰기를 캐싱할 수도 있다.
- **상수 메모리:** 전역 메모리의 일부인 이 메모리 영역은 커널이 실행되는 동안 변하지 않고 상수로 유지된다. 상수 메모리에 존재할 메모리 객체들은 호스트에서 할당하고 초기화한다. 작업-아이템들은 이 객체들에 대해 읽기 권한만 가진다.
- **지역 메모리:** 이 메모리 공간은 작업-그룹에 지역적이다. 작업-그룹

안의 모든 작업-아이템들이 공유하는 변수들은 이 메모리 영역에 할당하면 된다. 지역 메모리는 OpenCL 디바이스에서 전용으로 구현되어 있을 수도 있고, 전역 메모리의 일부분을 지역 메모리로 매핑해 놓은 것일 수도 있다.

- **사유 메모리:** 이 메모리 영역은 하나의 작업-아이템이 개별적으로 소유하고 있는 메모리이다. 작업-아이템의 사유 메모리에서 정의된 변수들은 다른 작업-아이템들에게는 보이지 않는다.

메모리 영역들과 이 영역들이 어떻게 플랫폼과 실행모델과 연관되어 있는지 그림 2에 도시되어 있다. 작업-아이템들이 PE들에서 수행되면서 자기들만의 사유 메모리를 가진다. 작업-그룹은 계산 유닛에서 실행되면서 그룹 안의 작업-아이템들이 공유하는 지역 메모리를 가진다. OpenCL 디바이스 메모리는 전역 메모리를 지원하기 위해 호스트와 함께 동작한다.

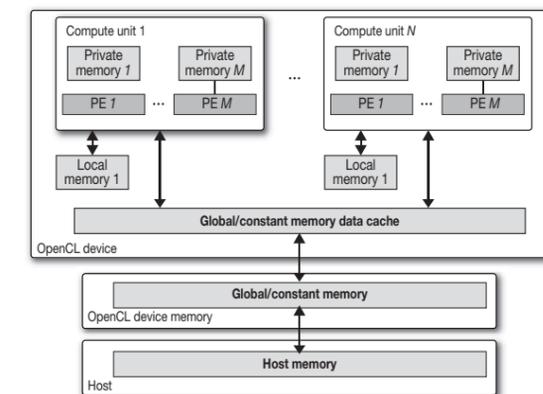


그림 2. OpenCL 메모리 모델 및 플랫폼 모델과의 연관도

결론

지금까지 OpenCL 이종 병렬 프로그래밍 표준에 대해서 왜 필요하고, 이를 이해하기 위해 플랫폼 모델, 실행 모델, 메모리 모델을 살펴보고, 구체적으로 문맥과 명령-큐를 설명하였다. 보다 자세한 설명은 OpenCL 표준명세[6]와 기타 문헌[7]을 참고하기 바란다.

전력효율을 최대화하기 위해 대두된 병렬 컴퓨팅, 이종 컴퓨팅이란 추세는 더욱 보편화되고 더욱 가속될 것이다. 삼성의 Exynos 5 시리즈에 집적될 ARM의 Mali 600 시리즈가 OpenCL을 지원하는 예에서 알 수 있듯이, 이러한 추세는 서버 시스템뿐만 아니라 임베디드 시스템에서도 주류가 되고 있다. 이러한 흐름에 능동적으로 대처하기 위해 OpenCL을 잘 이해하는 것이 중요하다고 생각한다.

Reference

- [1] <http://www.khronos.org/opencv/>
- [2] NVIDIA GeForce GTX680, http://www.geforce.com/Active/en_US/en_US/pdf/GeForce-GTX-680-Whitepaper-FINAL.pdf
- [3] J. Nickolls, I. Buck, M. Garland, K. Skadron, "Scalable parallel programming with CUDA", ACM Queue, vol. 6, no. 2, 2008
- [4] <http://www.khronos.org/conformance/adopters/conformant-products#opencv>
- [5] Altera, "Implementing FPGA Design with the OpenCL standard", Nov 2012
- [6] OpenCL 1.2 specification, <http://www.khronos.org/registry/cl/>
- [7] A. Munshi, B. Gaster, T. Mattson, J. Fung, D. Ginsburg, "OpenCL Programming Guide", 2011 (이영민 역저, "OpenCL 프로그래밍 가이드", Nov 2012)

반도체 메모리(DRAM) core 구조 및 특성



영인대학교 컴퓨터학과

이중호 교수
 연구분야 : 반도체 메모리 Testable Design, 설계 자동화 및 검증, 인공지능/영상인식 분야
 E-mail : joongho65@yonsei.ac.kr



서론

■ DRAM의 집적도와 코어(core) 구조

DRAM(Dynamic Random Access Memory)은 1970년대에 발명된 이후로 컴퓨터 및 모바일 제품에 없어서는 안 될 중요한 요소가 되었다. JEDEC (Joint Electron for Devices Engineering Council)에 표준화된 DDR3 DRAM제품은 이미 상용화 되었고[1], 수년 내에 DDR4가 상용화될 것이다. 이와 같이 DRAM의 집적도는 4Gbit~16Gbit 에 달하고 향후 더욱 증가될 것이다. DRAM의 집적도가 증가할수록 DRAM 코어(core)의 부담은 증가하여 코어의 고집적화에 필요한 기술이 요구되고 있다.

메모리 셀의 고집적화를 위해 DDR2 이후로 기존의 8F2 공정기술에서 6F2 공정기술로 전환하였고 산업체에서는 이 설계기술을 이미 상용화하였다[2,3,4]. 6F2 공정기술을 채용하기 위해서는 기존의 folded bit line구조에서 open bit line구조로 회로설계 변경이 필수적이며 이에 따른 설계의 안정적인 동작특성 확보가 필수적이다. 향후에도 메모리 장치의 고집적화는 지속적으로 요구될 것이며, 집적도를 향상시키기 위해 산업체에서는 4F2 공정 기술에 대한 연구도 진행 중이다.

본고에서는 DRAM 코어(core)를 구성하는 코어구조(core architecture)의 설계방식에 대한 noise의 영향 및 공정기술과 연관한 chip size의 영향에 대해 알아본다. 앞에서도 언급하였지만, DRAM의 코어를 구성하는 회로 방식으로 크게 두 가지로 구분된다. 하나는 전통적으로 안정적인 동작특성을 가지고 있는 folded bit line 구조이며, 다른 하나는 open bit line 구조인데 외부 noise에 대해 folded bit line 구조보다 취약한 특성을 가진다. 또한 folded bit line구조를 구성하기 위해 8F2 공정이 채택되며, open bit line구조를 구성하기 위해 6F2공정 이상이 요구된다. 이러한 방식과 그 장단점에 대해 하나씩 알아보자.

■ Folded vs. Open Bit Line 구조

1. Folded Bit Line 구조와 8F2공정기술

folded bit line 구조는 하나의 MAT(MAT : 하나의 bank를 여러 개의 MAT들로 구성)내에 비트라인 쌍이 존재하도록 구성된다. 즉 비트라인 쌍은 동일한 센스앰프(sense amplifier)에 연결된 두 개의 비트라인이며, 물리적으로 인접하게 위치한다. 메모리 셀에 저장된 데이터를 읽거나 쓰기위해 선택된 워드 라인에 연결된 트랜지스트의 게이트(gate)를 열어주면 선택된 비트라인을 통해 데이터를 읽거나 쓴다. 이때 선택된 비트라인에 데이터가 실리면 동일한 센스앰프에 연결된

반대편 비트라인은 기준전압($1/2 \cdot VDD$) 레벨을 유지하여 선택된 비트라인의 데이터를 증폭할 수 있도록 하여 준다. 그림 1.에 folded bit line구조에 대해 나타내었다. 그림에서 보듯이 하나의 워드라인(SWL)이 선택되면 비트라인 쌍 중에서 하나의 메모리 셀만 access 되도록 구성되어야하며, 이것은 선택된 셀의 비트라인을 제외한 반대편 비트라인이 기준전위 레벨을 유지할 수 있도록 하기 위함이다.

예를 들어 워드라인 SWL1이 선택되고 비트라인 BL0T가 선택되었을 경우 메모리 셀이 하나만 선택된다. 비트라인 BLOB에는 SWL1에 연결된 메모리 셀은 존재하지 않으며, 따라서 기준전위 레벨을 유지할 수 있다. 센스 앰프는 BLOT에서 읽혀진 전위레벨과 BLOB의 기준전위의 차이를 증폭하여 메모리 셀의 데이터를 읽어낼 수 있다. 그림 2.에 folded bit line 구조를 레이아웃 한 도면을 나타내었으며, 레이아웃에 대한 이해를 돕기 위해 그림 1.을 그림 3.으로 다시 나타내었다. 그림에서 하나의 비트라인 위를 지나는 워드라인 4개당 2개의 메모리 셀이 연결된다.

그림2의 동그라미는 메모리 셀 커패시터이고, 노란색 사각형은 active 영역을 나타낸다. 점선 사각형 내에 하나의 메모리 셀이 존재하며 반대쪽에도 동일한 메모리 셀이 존재한다. 두 개의 메모리 셀은 트랜지스트의 드레인(drain) 노드를 공유하여 비트라인에 연결된다. 이때 비트라인의 선폴을 "F"라고 하면 비트라인과 비트라인 사이 간격도 동일한 "F"를 유지한다. 따라서 비트라인에서 다음 비트라인까지의 총 거리(비트라인 pitch)는 "2F"이다. 또한 워드라인 선폴도 비트라인 선폴과 동일한 "F"를 유지하며 하나의 메모리 셀이 존재하기 위해 총 "4F" 만큼의 길이가 필요하다. 즉 워드라인 두개당 하나의 메모리 셀이 존재한다.

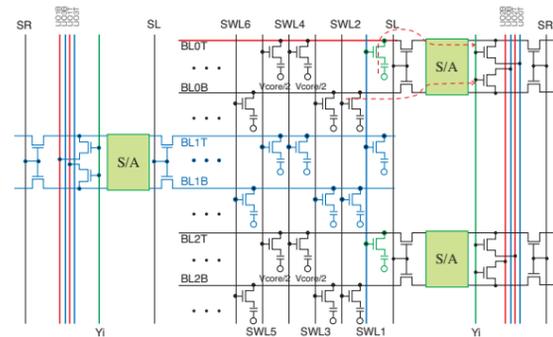


그림 1. folded bit line 구성 방식

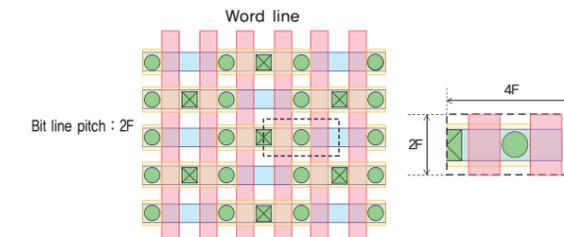


그림 2. 8F2 공정의 메모리셀 레이아웃도

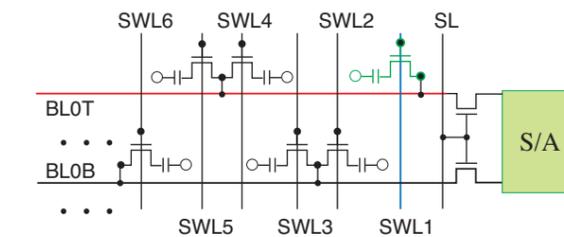


그림 3. 8F2 공정기술의 folded bit line 구성방식

따라서 하나의 메모리 셀이 차지하는 전체 면적은 $8F^2$ square ($2F \cdot 4F$)이며 편의상 $8F^2$ 라고 한다.

2. Open Bit Line 구조와 6F2공정기술

open bit line 구조는 비트라인 쌍의 각 비트라인이 서로 다른 인접한 MAT에 위치한다. 즉 비트라인 쌍은 동일한 센스앰프(sense amplifier)에 연결된 두 개의 비트라인이지만, 각 비트라인은 서로 다른 인접한 MAT에 서로 분리되어 위치하여 folded bit line과 달리 물리적으로 인접하게 위치하지 않는다. 메모리 셀에 저장된 데이터를 읽거나 쓰기위해 선택된 워드 라인에 연결된 트랜지스트의 게이트(gate)를 열어주면 선택된 비트라인을 통해 데이터를 읽거나 쓴다. 이때 선택된 비트라인에 데이터가 실리면 반대편 비트라인은 인접한 다른 MAT에 위치하여 기준전압($1/2 \cdot VDD$) 레벨을 유지한다.

따라서 선택된 비트라인의 데이터를 증폭할 수 있도록 하여 준다. 그림 4.에 open bit line구조에 대해 나타내었다. 그림 4.에서 워드라인 SWL1이 선택되고 비트라인 BL0T가 선택되었을 경우 메모리 셀의 데이터를 access하는 동안 반대편 비트라인 BLOB에는 선택된 메모리 셀은 존재하지 않으며, 따라서 기준전위 레벨을 유지할 수 있

다. 센스 앰프는 BLOT에서 읽혀진 전위레벨과 BLOB의 기준전위의 차이를 증폭하여 메모리 셀의 데이터를 읽어낼 수 있다. 그림 4.에서 보듯이 folded bit line 방식대비 메모리 셀이 모든 비트라인에 연결되어있다. 즉 그림 2.에서와 같이 메모리 셀이 비어있는 공간이 없이 공간을 메모리 셀로 모두 채워야해서 folded bit line방식대비 집적도측면에서 두 배 증가 시킬 수 있다.

그러나 실제로 비트라인 선폴이 너무 작아서 비트라인 피치 내에 메모리 셀을 모두 채워 넣을 수 없다. 따라서 비트라인 간격을 약간 증가시켜 메모리 셀 공간을 확보했다. 그림 5.에 folded bit line 구조를 레이아웃 한 도면을 나타내었으며, 그림5의 레이아웃에 대한 이해를 돕기 위해 그림 4를 그림 6으로 회로를 재배치하여 나타내었다. 그림5의 점선 사각형 내에 하나의 메모리 셀이 존재하며 대각선 방향 반대쪽에도 동일한 메모리 셀이 존재한다. 두 개의 메모리 셀은 트랜지스트의 드레인(drain) 노드를 공유하여 비트라인에 연결된다.

이때 folded bit line 방식의 경우 비트라인 "2F" 공간 안에 메모리 셀 커패시터가 위치하지만 open bit line방식은 "2F"공간 내에 메모리 셀 커패시터를 모두 채워 넣기에 공간이 너무 작다. 따라서 비트라인과 비트라인 간격을 기존 대비 "F"만큼 더 확보하여 메모리 셀 커패시터를 채워 넣었다. 결론적으로 비트라인 간격을 기존 "2F"대비 "3F"로 증가시켰으며 워드라인 선폴은 동일하게 유지하여 메모리 셀 영역을 확보하였다. 따라서 메모리 셀이 차지하는 전체 면적은 $6F^2$ square($3F \cdot 2F$)이며, 편의상 $6F^2$ 로 표기한다.

3. noise 특성

folded bit line 방식이 외부 noise에 대해 안정적인 동작 특성을 가진다. 그 이유는 비트라인 쌍이 서로 인접하게 위치하여 비트라인을 대칭이 되도록 설계할 수 있어서 비트라인 기생 커패시턴스 mismatch로 인한 영향을 최소화 할 수 있으며, 비트라인 쌍이 동일한 MAT내에 인접하게 위치하기 때문에 외부 noise에 대해 동일하게 영향을 받는다. 따라서 비트라인 쌍내 두 개의 비트라인은 동일하게 noise영향을 받게 된다.

따라서 센스앰프에서 데이터를 증폭 시 센스앰프에 미치는 offset noise량은 미미하여 안정적인 동작을 확보할 수 있다. 이는 파도치는 바다 위를 항해하는 배와 비교할 수 있다. 일정한 방향으로 일렁이는 파도의 마루와 골을 거스르지 않고 타면 돛단배는 파도로부터 안전하게 항해를 할 수 있다. open bit line의 경우 두 개의 비트라인이 서로 다른 MAT에 위치하여 서로 다른 외부 noise의 영향을 받는다.

따라서 두 개의 각 비트라인으로부터 타고 들어온 noise의 영향으로 센스앰프에서 데이터 증폭 시 offset noise량이 극대화 될 수 있어서 데이터 전위 레벨을 감쇄할 수 있다. 그림 7~8에 이에 대한 예를 나타내었다. 또 다른 측면은 비트라인 상호간의 coupling noise관점이다. folded bit line 방식에서 BL1T 비트라인이 선택되었고 메모리 셀의 데이터가 "high"값이 저장되어 있을 경우 반대쪽 비트라인(BL1B)은 기준전압(1/2*VDD) 레벨을 가지며,

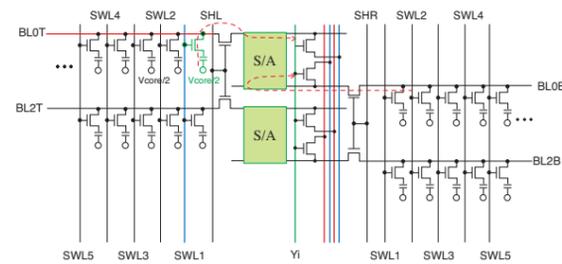


그림 4. open bit line 구성방식

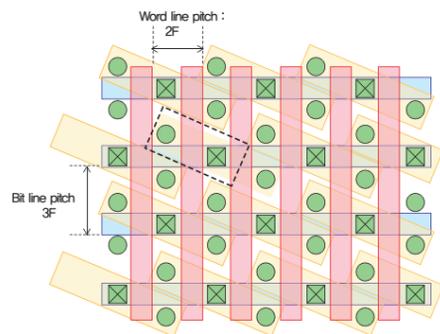


그림 5. 6F2 공정의 메모리셀 레이아웃도

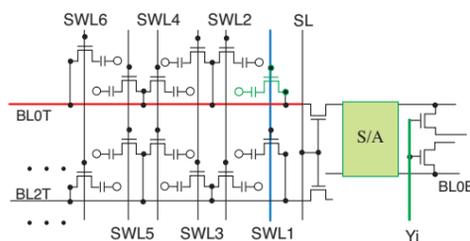


그림 6. 6F2 공정기술의 open bit line 구성방식

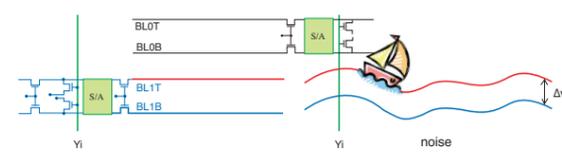


그림 7. folded bit line 방식의 외부 noise

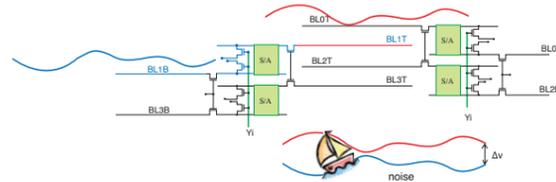


그림 8. open bit line 방식의 외부 noise

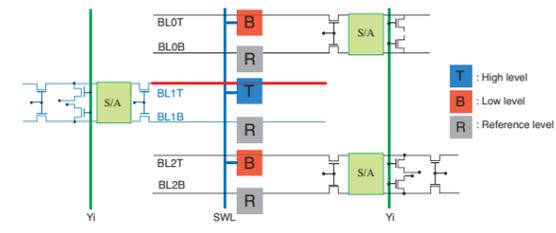


그림 9. folded bit line 방식의 coupling noise

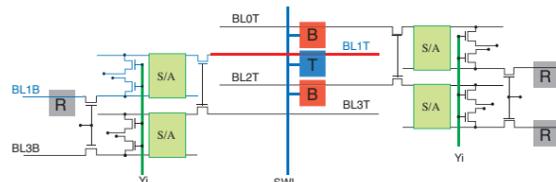


그림 10. open bit line 방식의 coupling noise

인접한 다른 비트라인 쌍에 연결된 인접한 비트라인(BLOB)도 기준전압 레벨을 가진다. 따라서 BL1T에 인접한 상하의 비트라인 들은 모두 기준전압레벨을 가져서 coupling noise의 영향에서 안정적이다 (그림 9 참조). 반면에 open bit line의 경우 선택된 BL1T 비트라인에 대해 상하로 인접한 비트라인이 각각 BL0T와 BL2T이며 이 비트라인들에 연결된 메모리 셀에 저장된 데이터가 "low" 일때가 최악의 영향을 미친다. 즉, BL1T의 "high" 데이터를 양쪽 비트라인에서 "low"로 끌어내리도록 coupling되어 noise의 영향이 folded bit line 방식보다 증가한다(그림 10).

앞에서 외부 noise와 coupling noise에 대한 영향을 살펴봤는데 open bit line이 noise immunity에 취약하다는 것을 확인할 수 있다. open bit line방식의 noise 취약점을 보완하기 위해 많은 연구가 진행되어 DDR2 제품 이후에 안정적인 동작을 확보 하였다.

4. Die size

Chip size를 증가시킬 수 있는 코어 공정기술은 단연 6F2 공정기술이다. 동일한 MAT size에 대해 6F2 방식이 8F2대비 두 배의 메모리 셀을 연결할 수 있다. 그러나 6F2는 noise에 취약한 특성을 고려하여 실제로는 두 배의 셀을 연결하지 않고 비트라인의 부하 (loading)를 줄이도록 하여 noise특성을 보상하도록 하였다. 예를 들어 8F2 방식에서는 비트라인에 연결되는 워드라인 수가 128(2^7), 256(2^8), 512(2^9) row등의 2의 x 제곱수로 연결되었다. 그러나

6F2 방식에서는 비트라인 위를 지나가는 워드라인 수를 약간 줄여 비트라인에 연결되는 메모리 셀 수를 줄였다.

이렇게 함으로써 비트라인의 부하를 줄일 수 있고 noise의 취약성을 보완할 수 있도록 하였다. 삼성의 DDR2 제품의 예에서 90nm Tech.의 제품은 비트라인 위를 지나가는 워드라인을 512개를 사용하였다. 그러나 80nm Tech.에서 6F2를 적용한 동일 제품에서는 비트라인 위를 지나가는 워드라인의 수를 320개로 줄여서 비트라인 부하를 37.5% 줄였다.

그러나 noise의 취약성을 보완하기 위해 워드라인 수를 줄임으로서 bank를 구성하는 단위 MAT 개수는 증가하게 되고 따라서 MAT와 MAT사이에 존재하는 센스앰프의 수도 증가한다. 표 1.에서 알 수 있듯이 증가한 센스앰프가 차지하는 면적이 기존 90nm Tech. 제품 대비 68% 증가였다. 그럼에도 불구하고 chip의 전체 면적은 90nm Tech.대비 47% 감소하였다. 이러한 효과는 Tech.가 감소한 영향 13%를 제외하고 표에는 나타나 있지 않지만 redundancy 셀을 줄인 효과도 있어서 이런 요인을 제외해도 6F2로 인해 최소 10% 정도 개선효과가 있다고 추정한다.

Items	Samsung 80-nm 512-Mbits DDR2 Rev.E	Samsung 90-nm 512-Mbits DDR2 Rev.E	Note
Process node	80nm	80nm	-
Cell size	0.038um ²	0.076um ²	-
Chip size	49mm ²	72mm ²	13% ↓
Cell structure	6F2	8F2	-
Array structure	Open bitline	Folded bitline	-
Wordlines per array block	320 wordlines	512 wordlines	-
Sense Amp. in chip height	104	62	68% ↓
Gross die(12" wafer)	1,266	862	47% ↓

표 1. DDR2 삼성 제품의 die size 비교

표 2.에 삼성제품과 SK하이닉스 제품을 비교하였다. 비교시 80nm대의 공정 technology 제품을 비교하였으며, DDR2 제품이다. 80nm대의 공정이지만 양사 Tech.에서 다소의 cell 크기 차이가 삼성 제품이 SK하이닉스 제품대비 24% 작다) 있고 설계 방법의 차이 등이 존재하여 1:1비교는 어려우나 대략적으로 6F2 제품이 약 10% 정도의 net die 개선효과가 있다고 추정 할 수 있다.

Items	Samsung 80-nm 512-Mbits DDR2 Rev.E	SK Hynix 80-nm 512-Mbits DDR2	Note
Process node	80nm	80nm	-
Cell size	0.038um ²	0.05um ²	24% ↓
Chip size	49mm ²	56mm ²	13% ↓
Cell structure	6F2	8F2	-
Array structure	Open bitline	Folded bitline	-
Metal layers	3	3	-
Gross die(12" wafer)	1,266	1,100	15% ↓

표 2. 80nm Tech. DDR2 제품 die size 비교

5. 맺음말

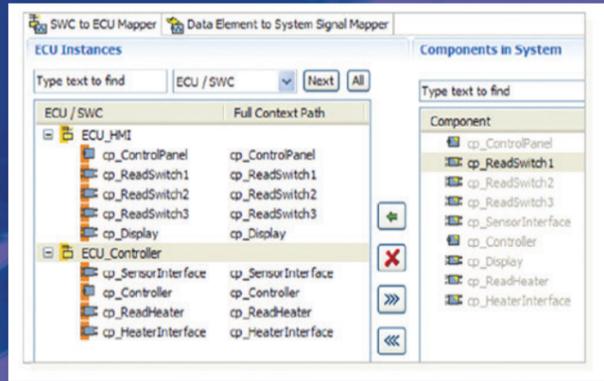
앞에서 folded bit line 구조와 open bit line 구조에 대해서 noise 관점에서 외부 noise 영향과 비트라인 간에 coupling noise의 영향에 대해 살펴보았으며 chip size 감소영향에 대해서도 살펴보았다.

산업체에서 대용량의 메모리 수요는 지속적으로 증가 할 것이며, 전통적으로 안정적인 동작특성을 가지는 folded bit line(8F2) 구조를 넘어서 noise의 취약성을 극복하여 open bit line(6F2) 구조의 적용이 산업체에서는 일반화 되고 있다. 4F2 처럼 향후에도 chip size를 줄일 수 있는 방안에 대한 연구는 지속적으로 진행될 것이며, 새로운 형태의 noise 문제들이 대두될 것이다. 이러한 문제 해결을 위해 지속적인 연구가 요구된다.

Reference

- [1] D. Graham-Smith, "IDF: DDR3 won't catch up with DDR2 during 2009," in PC Pro, Aug. 2008.
- [2] Busch, B. et al., "A 78nm 6F2 DRAM technology for multigigabit densities," VLSI Technology, Digest of Technical Papers, page 28-29, June 2004.
- [3] Benjamin C. Lee, Engin Ipek, Onur Mutlu, and Doug Burger, "Architecting phase change memory as a scalable dram alternative," ISCA '09 Proceedings of the 36th annual international symposium on Computer architecture, page 2-13, 2009.
- [4] Hoon Jeong, et al., "55 nm capacitor-less 1T1 DRAM cell transistor with non-overlap structure," Electron Devices Meeting, 2008. IEDM, page 1-4, Dec. 2008.

Volcano™ Technical Paper



Hans-Jürgen Mantsch
Mentor Graphics Corporation

www.mentor.com/volcano

SPECIAL Column

AUTOSAR의 “전화” 측면: ECU 리소스 템플릿 사용 사례

서론

최근 몇 년간 AUTOSAR는 자동차에 사용되는 소프트웨어 아키텍처를 기술하기 위한 표준으로 확실히 자리를 잡게 되었다. 사용자들은 이 표준의 본래 관점을 확장하고 “가상 기능 버스” 개념을 개발함으로써, AUTOSAR 방법론을 채택하여 가상 전기 아키텍처의 일관된 개발을 위한 전체론적 E/E 접근 방식을 구현할 수 있다. 따라서 전기적으로 관련된 모델을 이용하여 논리적 설계는 물론 궁극적으로 와이어링 하네스까지 생성할 수 있다.

본론

AUTOSAR(AUTomotive Open System ARchitecture)는 자동차 애플리케이션의 전자 제어 장치 (ECU: Electronic Control Unit)의 구성과 소프트웨어 개발에 중점을 두고 있다. AUTOSAR 방법론에서 정의하는 메타 모델에는 이 방법론의

최근 몇 년간 AUTOSAR는 자동차에 사용되는 소프트웨어 아키텍처를 기술하기 위한 표준으로 확실히 자리를 잡게 되었다. 사용자들은 이 표준의 본래 관점을 확장하고 “가상 기능 버스” 개념을 개발함으로써, AUTOSAR 방법론을 채택하여 가상 전기 아키텍처의 일관된 개발을 위한 전체론적 E/E 접근 방식을 구현할 수 있다. 따라서 전기적으로 관련된 모델을 이용하여 논리적 설계는 물론 궁극적으로 와이어링 하네스까지 생성할 수 있다.

특정 부분을 기술하는 일련의 템플릿이 포함되어 있다. 여기서 설명하는 접근 방식에서는 주로 “ECU 리소스 템플릿(ECU Resource Template)”을 사용한다. 이를 통해 아키텍처를 개발하고 평가하기 위한 반복적이면서 자동화된 프로세스 상에서 전기적으로 관련된 AUTOSAR 모델을 사용할 수 있다. 이와 동시에 적절한 인터페이스를 사용하면 다양한 개발 영역의 툴을 일관되면서 효과적인 방식으로 결합할 수 있다.

■ ECU 리소스 템플릿

ECU 리소스 템플릿에는 오실레이터, 타이머, 클럭, 스토리지 등 제어 장치를 구성하는 하드웨어 컴포넌트 목록에 대한 설명이 포함되어 있다. 또한 여기에는 제어 장치에서 주변 디바이스에 대한 연결 또는 신호 처리를 담당하는 아날로그-디지털 컨버터(ADC), 디지털-아날로그 컨버터(DAC), 디지털 IO 등의 모든 하드웨어 모듈도 포함된다. 이러한 “온보드” 컴포넌트 외에도 AUTOSAR 메타 모델

을 사용하여 신호 처리 라인을 통해 제어 장치의 주변 컴포넌트에 전기적으로 연결될 수 있는 모든 장치를 기술할 수도 있다. AUTOSAR 릴리스 3.x에서는 ECU 리소스 템플릿의 메타 모델이 대개 ECU 외부에 있는 하드웨어 유형(디스플레이, 액추에이터, 센서) 세 가지를 인식한다.

하드웨어 장치는 제어 장치 또는 하드웨어 컨테이너에 대해 정의할 수 있는 “AssemblyHWConnections”를 통해 해당하는 하드웨어 요소와 전기적으로 연결된다. ECU 리소스 템플릿에 기술된 내용은 적절한 툴의 지원을 통해 논리 회로도로 전환되고 와이어링 하네스 모델을 생성하기 위한 토대로 사용할 수 있다.

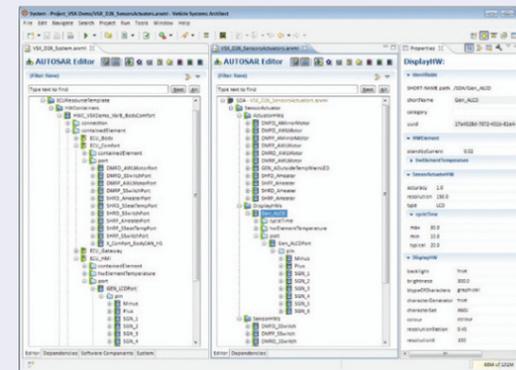


그림 1. ECU 리소스 템플릿의 하드웨어 설명

■ AUTOSAR 메타 모델

ECU 리소스 템플릿은 AUTOSAR 릴리스 2.x와 3.x에서는 거의 변경되지 않았다. 주된 이유는 제어 장치의 ECU 추출, 런타임 환경의 토대, 베이직 소프트웨어 구성을 생성하는 데에 필요한 모든 특정 아티팩트를 충분한 범위와 정확도로 기술해야 하기 때문일 수 있다. 릴리스 4.0에서는 특정 타입을 일반적인 접근 방식으로 대체하는 수준까지 이 템플릿의 완성도를 높였는데, 이러한 접근 방식을 활용하면 메타 모델에서 정의할 수 있는 유형, 범주 및 특성을 사용하여 모든 하드웨어 컴포넌트를 기술할 수 있다.

■ AUTOSAR 시스템 토폴로지

그림 2에 나와 있는 것처럼 메타 모델에 ECU로 기술되거나 하드웨어 컨테이너에 기술된 서로 다른 시스템 제어 장치가 AUTOSAR 시스템 템플릿(System Template)에 취합되어 시스템 토폴로지를 구성한다. 이는 전체 차량 모델을 포함할 수도 있고 개별(또는 대체) 서브시스템만 포함할 수도 있다. ECU 모델이 포함된 “AssemblyHWConnectors”는 ECU의 포트 및 핀을 해당하는 디스플레이, 센서 및 액추에이터 하드웨어 컴포넌트의 포트 및 핀과 연결한다. 그림 1에서는 멘토 그래픽스(Mentor Graphics)의 VSA(Vehicle Systems Architect) 툴을 이용하여 시스템 토폴로지를 구성한 예를 보여 준다. AUTOSAR 모델의 시스템 모델 아티팩트는 적절한 익스포트 인터페이스를 사용하여 AUTOSAR XML 파일로 추출될 수 있다.

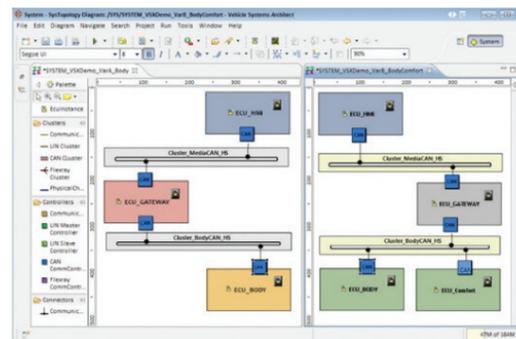


그림 2. VSA - 시스템 템플릿의 대체 토폴로지 시각화

■ 논리적 디자인 임포트

두 번째 단계에서는 AUTOSAR 모델에서 추출된 데이터가 논리 회로도 혹은 와이어링 하네스를 개발하는 데 적합한 툴에 임포트된다. 여기서 설명하는 사례에서는 이를 위해 멘토 그래픽스 캐피탈(Capital) 제품 라인의 캐피탈 아키텍처(Capital Architect)가 사용된다.

임포트는 특정 어댑터를 통해 수행된다. 이때 어댑터 내의 구성 파일을 사용하거나 수동으로 논리 회로도 임포트해야 할 시스템의 추출 부분을 설정할 수 있다. 임포트 도중 AUTOSAR ECU는 논리적 디바이스로 해석되고, “AssemblyHWConnections”는 신호로 해석된다. 이 외에도 해당하는 규칙 집합을 활용함으로써 이 접근 방식을 통해 AUTOSAR에서 추출한 모델 아티팩트 중 해당 특성에 따라 부품 라이브러리 내의 논리적 컴포넌트 유형과 연결할 수 있는 아티팩트를 결정할 수 있다. 또한 이 접근 방식을 사용하면 이러한 아티팩트를 해당하는 옵션 및 변형 모델에 나타낼지 여부(그럴 경우 방법까지)를 정할 수 있다.

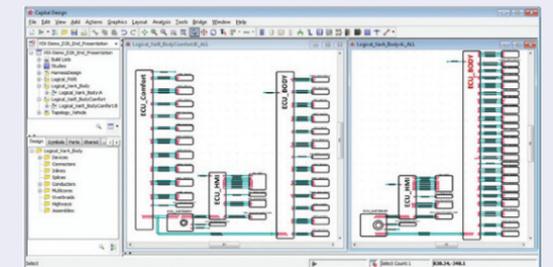


그림 3. 신호, 제어 장치 및 주변장치에서 생성된 캐피탈 아키텍처의 논리적 회로 다이어그램

이를 통해 개별 제어 장치 또는 전체 시스템의 핀 아웃이 포함된 서로 다른 논리 회로도가 생성된다(그림 3 참조). 이러한 회로도는 공유 컨넥터를 사용하여 다른 회로도의 논리적 신호와 연결될 수 있다. 예를 들어 전력 공급을 공유한 논리 회로도(VSA에서 개발한 AUTOSAR 모델의 ECU 주변장치 모델과 별도로)에 기술한 다음 추가적인 단계를 통해 이러한 주변장치 모델과 수동 또는 자동으로 연결할 수 있다. 이 방법을 활용하면 반복적인 병렬 프로세스가 마련되어, 서로 다른 개발 영역이 해당 개발 프로세스의 결과를 후속 프로세스 흐름의 특정 전송 지점에 제공할 수 있게 된다.

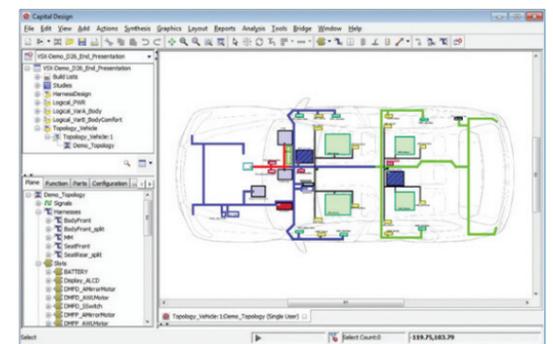


그림 4. 캐피탈 아키텍처의 물리적 차량 토폴로지: 제어 장치 및 하드웨어 장치용 번들 및 슬롯

■ 물리적 토폴로지에 대한 매핑

세 번째 단계에서는 논리 회로도가 물리적 토폴로지 내에서 자동으로 분배되거나 기술된다. 여기서 이 토폴로지는 전체 모델 또는 토폴로지의 특정 부분과 관련이 있을 수 있다. 토폴로지는 직접 생성할 수도 있고, 적절한 인터페이스가 있는 경우 “플래트닝(flattening)” 프로세스를 통해 CATIA 3D 모델 등에서 불러올 수 있다. 그림 4에서처럼 토폴로지 모델은 크게 제어 장치 및 보조 장치를 위한

슬롯과 슬롯을 서로 간에 연결하는 번들로 구성되어 있다.

“매핑 규칙”(배치 규칙)을 활용할 경우 제어 장치와 보조 장치를 컨테이너(슬롯)에 자동으로 분배하는 방법을 결정할 수 있다. 즉, 이렇게 하면 논리 회로도 와 AUTOSAR 모델에서 생성된 해당 컨테츠를 물리적 공간에 할당하는 방법이 정해진다. 이러한 자동 매커니즘은 각기 다른 시스템 및 기능 대안을 검토해야 하는 반복적인 접근 방식과 함께 사용할 경우 이점을 제공하는데, 이는 이를 통해 필요한 수작업 단계가 최소화되기 때문이다.

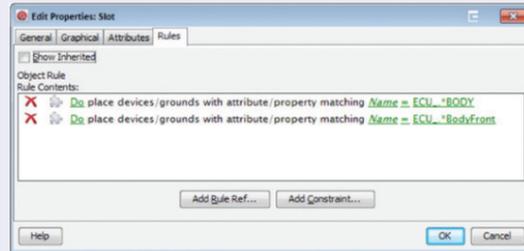


그림 5. 캐피탈 아키텍처에서의 하드웨어 슬롯에 대한 하드웨어 보조 장치의 규칙 기반 할당

마지막 단계로, 논리 회로도의 물리적 할당을 끝낸 후 “와이어링 합성(wiring synthesis)”을 수행한다. 이 단계에서는 AUTOSAR 모델의 논리적 신호가 전기 라인으로 전달되고, 케이블 하네스, 커넥터 및 접점의 차량 토폴로지에서 생성되며, 최적의 스플라이스(splice)가 생성된다. 이를 바탕으로 케이블 길이, 번들 직경, 무게 정보와 함께 초기 소재 및 생산 비용 같은 관련된 케이블 하네스 정보를 제공한다. 결과는 그림 6에서처럼 해당하는 차트를 통해 사용자에게 제공된다.

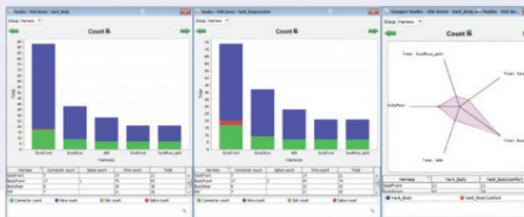


그림 6. 대체 시스템 토폴로지의 서로 다른 매핑에 대한 결과 및 비교

프로세스를 계속 진행함에 따라 사용자는 전력/접지 분석, 스네이크패스(sneakpath) 분석, FMEA, 과도 시뮬레이션 같은 기능을 활용할 수 있다. 그뿐 아니라 테이프, 클립, 그로멧(grommet) 같은 기계적인 컴포넌트를 추가하는 등의 노력으로 자동으로 생성된 케이블 하네스 다이어그램에 별도의 세부 사항을 추가할 수 있다(그림 7 참조).

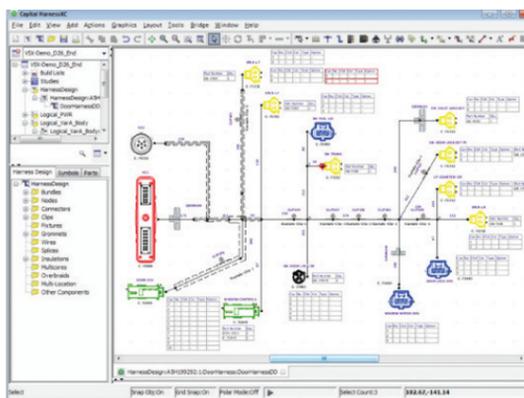


그림 7. 캐피탈 하네스(Capital Harness)의 도어 케이블 하네스 모델

■ 일관된 데이터 교환을 통한 협력

실제 소프트웨어를 개발하기 전에 표준화된 개방형 AUTOSAR 메타 모델을 사용하여 아키텍처를 개발할 수도 있다. ECU 리소스 템플릿은 이러한 작업에 특히 적합하며, 이는 전기적 아키텍처 프로세스에 실제 소프트웨어 지향적 AUTOSAR 개발 방법론을 적용할 수 있도록 하기 때문이다.

논리적 아키텍처 개발 또는 차량 아키텍처의 부분적인 측면 최적화를 위한 특수한 도구를 사용하는 경우와 달리, 여기서 설명하는 접근 방식은 확립된 개발 툴을 사용하는 것을 전제로 한다. 적절한 인터페이스의 도움을 받아 개발 툴에 제공되는 데이터를 바로 재사용할 수 있으며, 이를 통해 얻은 평가 결과에서는 새로운 정량화된 중요성을 제공한다. 설계 작업은 개발 툴에서 바로 수행할 수 있으므로 더 이상 개발 프로세스와 동떨어져 진행되지 않는다.

의미 있는 모델을 사용하여 사양의 교환을 보완함에 따라 자동차 제조업체 및 공급업체 내의 각기 다른 부서와 영역이 보다 긴밀하게 협력하여 오류를 최소화하는 일관된 개발 프로세스를 구축하는 것이다.

자세한 정보는 멘토 그래픽스에 전화로 문의하거나 www.mentor.com/volcano 에서 확인할 수 있다.

Copyright © 2010 Mentor Graphics Corporation. 이 문서는 Mentor Graphics Corporation에서 독자적으로 소유하는 정보를 포함하고 있으며, 원래 수신자가 내부 업무용으로만 전체 또는 일부를 복제할 수 있다. 단, 이 경우 모든 복사본에 본 공지 내용이 완전히 포함되어 있어야 한다. 수신자는 이 문서의 내용을 인정함으로써 본 정보가 무단으로 사용되지 않도록 하기 위해 합리적인 수준에서 모든 노력을 기울인다는 데 동의하게 된다.

(주)한국멘토
 주소 : 서울시 강남구 삼성동 무역센터
 트레이드타워 2104호
 Tel : 02-551-3434
 Web : www.mentorkr.com

IDEC MPW 설계공모전 2013

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



2013년 MPW 공정 지원 내역 및 진행 일정

회사	공정(μm)	공정내역	size	공모신청수
메그나칩/아이닉스	0.35μm	CMOS 1-poly 8-metal [Optional layer(DNW, HBL, BJT, CPOLY) 8-기]	4mmx4mm 5mmx4mm	3 2
	0.18μm	CMOS1-poly6-metal [6metal을 Thickmetal(TKM)로만 사용 가능] [Optional layer(DNW, HBL, BJT, PMM) 8-기]	4.5mmx4mm 4.5mmx2.5mm	4
동부하이텍	0.11μm	CMOS1-poly6-metal(Top-UTM)	5mmx2.5mm 2.5mmx2.5mm	2
	0.18μm BCD CMOS	CMOS 1-poly 4-metal TM		4
	0.35μm BCD CMOS	CMOS 2-poly 4-metal TM		4
Tower Jazz	0.18μm CIS	CMOS 1-poly 4-metal		2
	0.18μm	RFCMOS 1-poly 6-metal		2
	0.18μm	CMOS 1-poly 3-metal(MT)		2
	0.18μm BiCMOS	SiGe BiCMOS 1-poly 6-metal		1

2013년 공정 지원 변경 내역
 * 상성 공정 - ① 0.13μm 공정 지원 중단 ② 상성 45μm 지원 회사 증가 (2회~3회)
 * 동부 공정 - ① 0.11μm-3회 축소 ② 0.35μm 축소 (5회~4회) → 0.18μm 증가 (3회~4회)
 * 동부 BCD 공정 - PKG 지원 중단

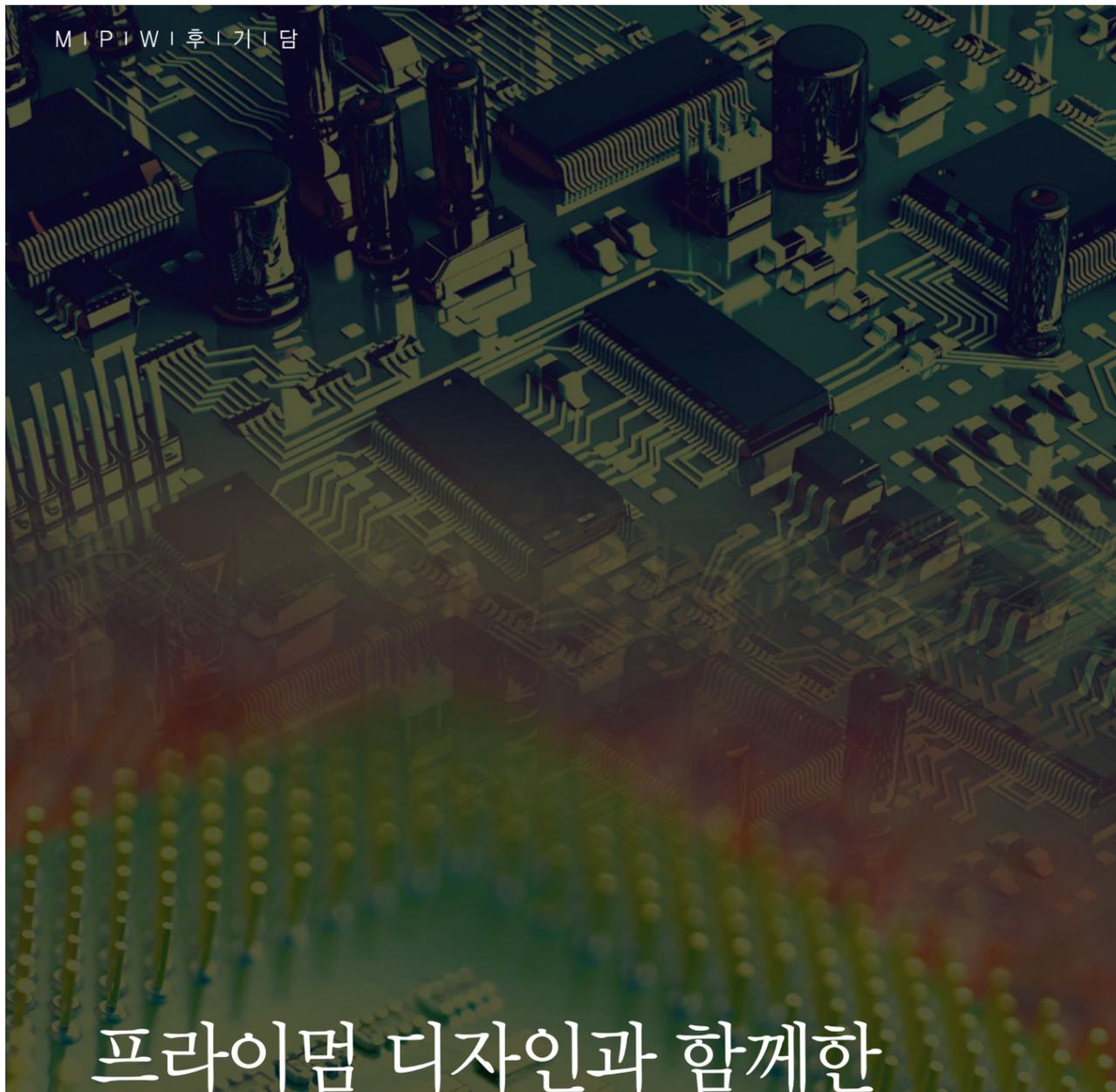
2013년 MPW 참여팀 현황

구분	공정사	공정	제작 횟수	우선모집		정규모집		추가	DB타입 (Tape-Out)	DB타입 (Fab-In)	Die-out	Package-out
				신청마감	신청발표	신청마감	신청발표					
110회 (13-01)	M/H	0.18μm	20	12.12.07	12.12.20	12.12.07	12.12.20		13.02.18	13.03.04	13.07.22	13.08.30
	T/J	0.18μm(SiGe)	1	12.12.07	12.12.20	12.12.07	12.12.20		13.02.27	13.03.13	13.06.12	13.07.12
119회 (13-02)	M/H	0.11μm	12	12.12.07	12.12.20	12.12.07	12.12.20		13.03.20	13.04.10	13.07.31	13.09.02
	상성	45nm(RF 지원)	48	12.12.07	12.12.20	12.12.07	12.12.20		13.03.15	13.04.05	13.06.15	13.09.16
120회 (13-03)	M/H	0.18μm	20	12.12.30	13.01.16	12.12.30	13.01.16		13.05.01	13.05.15	13.08.14	13.09.16
	T/J	0.18μm(CIS)	1	12.12.30	13.01.16	12.12.30	13.01.16		13.05.06	13.05.13	13.09.16	-
	동부	0.18μm	2	12.12.30	13.01.16	12.12.30	13.01.16		13.05.15	13.06.29	13.08.28	13.09.30
	T/J	0.18μm(RF)	2	12.12.30	13.01.16	12.12.30	13.01.16		13.05.20	13.06.27	13.09.16	-
121회 (13-04)	M/H	0.35μm	20	13.01.30	13.02.15	13.01.30	13.02.15		13.05.20	13.05.27	13.09.16	-
	동부	0.18μm	2	13.01.30	13.02.15	13.01.30	13.02.15	13.03.04-	13.06.17	13.07.04	13.10.04	13.11.08
	상성	45nm	48	13.01.30	13.02.15	13.01.30	13.02.15		13.07.05	13.07.10	13.12.06	14.01.06
122회 (13-05)	M/H	0.18μm	20	13.01.30	13.02.15	13.01.30	13.02.15	13.04.01-	13.07.29	13.08.12	13.12.24	14.02.07
123회 (13-06)	동부	0.18μm	2	13.02.28	13.03.15	13.02.28	13.03.15	13.05.02-	13.08.14	13.08.28	13.11.27	13.12.27
	동부	0.35μm	3	13.02.28	13.03.15	13.02.28	13.03.15		13.08.21	13.09.04	13.12.04	14.01.06
124회 (13-07)	동부	0.11μm	12	13.03.30	13.04.15	13.03.30	13.04.15	13.06.03-	13.09.11	13.10.02	14.01.22	14.02.24
	T/J	0.18μm(CIS)	1	13.04.30	13.05.17	13.04.30	13.05.17		13.10.14	13.10.21	14.02.17	-
125회 (13-08)	T/J	0.18μm(RF)	1	13.04.30	13.05.17	13.04.30	13.05.17	13.07.01-	13.10.21	13.10.28	14.02.17	-
	M/H	0.18μm(BCD)	2	13.04.30	13.05.17	13.04.30	13.05.17		13.10.21	13.10.28	14.02.25	14.04.25
126회 (13-09)	동부	0.35μm	3	13.04.30	13.05.17	13.04.30	13.05.17		13.10.23	13.11.06	14.02.05	14.03.05
	상성	45nm(RF 지원)	48	13.05.30	13.06.17	13.05.30	13.06.17	13.08.01-	13.11.08	13.11.29	14.04.11	14.05.12
127회 (13-10)	동부	0.18μm	2	13.02.28	13.03.15	13.02.28	13.03.15		13.11.13	13.11.27	14.02.26	14.03.26
	M/H	0.35μm	20	13.02.28	13.03.15	13.02.28	13.03.15		13.12.02	13.12.17	14.03.25	14.04.25

참여대상: IDEC Working Group(WG) 대학의 학부생 및 대학원생

- 표기 : 1) 년 월 일 2) M/H= 메그나칩/아이닉스 3) T/J= TowerJazz
- 모집 : 우선(5%), 정규(5%) 모집을 원칙으로 하며, 정규에 마감일 인원 공정에 대해서만 후기모집을 실시함.
- 동부 & TowerJazz 공정은 5mmx2.5mm 또는 2.5mmx2.5mm 사이즈로만 신청 가능함.
- 위의 일정과 기한에 따라 다소 변경될 수 있음.
- 설계설명회 우선모집은 개최되지 않으며, 후기모집도 경우에 따라 개최되지 않을 수 있음.
- 2011년 MPW부터는 칩제작이 개발 지원되는 경우 기간기간에 따른 일정 비율의 참가비를 환급하여 적립해 드림.

Multi Project Wafer Design Contest 2013



프라이머 디자인과 함께한 MPW 117회 Layout 후기

저희는 부경대학교 전자공학과에 재학 중이며 내년 2월에 졸업예정인 대학원 학생들입니다. 이번 MPW 117회에 참여하면서 느낀 점과 진행과정에 관해서 경험담을 써 보고자 합니다. 우리 연구실은 아날로그 회로설계를 연구하며, 그 중 PLL(Phase-Locked-Loop)을 집중적으로 다루며, 기존의 PLL에서 회로를 추가하거나 구조를 변화시켜 성능을 향상하는 것을 목표로 하고 있습니다. 항상 새로운 구조의 PLL을 설계하여 MPW를 통하여 칩으로 구현하였습니다.

저는 대학원에 진학한 이후 3회(98, 104, 117회)에 걸쳐 MPW에 참여하여

Layout을 진행했습니다. MPW 98회에 참여하면서 처음으로 Layout을 접했고, 처음 진행하는 Layout인 만큼 문제가 많이 발생했습니다. 별도의 교육이나 준비 없이 선배들로부터 배운 기초적인 지식만 가지고 막연하게 Rule(DRC, LVS)만 통과하는 식으로 Layout을 진행했고, 역시나 결과는 좋지 않았습니다. Layout 변수를 추출하여 post simulation도 실험실의 사정으로 하지 못했습니다.

두 번째로 참여하게 된 MPW 104회에서는 좀 더 준비해서 전원 공급이나 접지 연결 금속선의 두께를 고려하여 좀 더 신경을 써서 Layout을 진행했지만, Post-

simulation을 진행하지 않아 목표치보다 주파수가 매우 낮게 나왔습니다. 전압을 변화시켜 간신히 측정하였지만 결과는 simulation과는 너무 다르게 원하는 특성을 얻을 수가 없었습니다. Layout Tool 사용법에 대해 부족한 점이 많이 느껴, Tool 사용 교육을 신청해서 듣기도 하였지만, 정해진 내용에 맞춰서 진행되는 실습은 크게 도움이 되지는 않았습니다. Tool 사용법에 차중한 결과는 좋은 특성이 있는 칩을 layout 하려고 하는 저희의 의도와는 거리가 있었습니다.

이렇게 두 번의 실패를 하고 난 뒤 교수님이 '프라이머 디자인'이라는 회사에 제안하여 여름방학 기간 3개월 동안 회사에 나가 실무자와 함께 Layout을 진행하게 되었습니다. 이전에는 계획 없이 막연하게 Layout을 진행했다면, 회사에서는 진행하기에 앞서 각 IP와 블록 별 면적과 Pad 배분 및 위치 등을 고려하여 체계적으로 계획을 잡고 순서에 맞춰 업무를 진행하였습니다.

가장 문제점이라고 생각되었던 Tool 사용방법은 정말 속이 시원할 정도로 잘 해결되었습니다. 막히는 부분에 대해서 즉각적으로 실무자에게 도움을 요청할 수 있었고, 단순한 문제 해결뿐 아니라 다양한 팁까지 배울 수 있어 많은 도움이 되었습니다. 또한, Layout을 진행하면서 주기적으로 진행사항을 체크하고 문제점과 개선방안을 제시하여 계획에 차질이 없도록 하였고, 일반 교육에서는 들을 수 없었던 노하우나 경험 등을 얻을 수 있었습니다. 그리고 학교에서는 알 수 없었던 실제 현장의 분위기나 업무 진행 방식을 간접적으로나마 체험할 수 있었습니다. 같이 업무를 진행한 직원분께서 처음부터 끝까지 신경 써주시고 배려해 주신 덕분에 무사히 기간에 맞춰 DB 제출을 할 수 있었고, Chip은 2~3월에 나올 예정입니다.

우리 연구실뿐만 아니라 처음 Chip 제작을 시도하는 연구실에서는 비슷한 애로사항을 겪고 있을 것이라 예상됩니다. 이러한 부분에서 크게 와 닿지 않는 교육 참여보다는 직접 도움이 되는 디자인 하우스 업체와의 연계를 통해 Chip 제작을 준비하는 것이 도움 된다고 생각합니다. 또한, IDEC에서도 MPW를 진행할 때, Design house 업체들과의 연계를 통한 프로그램을 만들어 보는 것도 Chip 제작 성공률 향상에 큰 도움이 될 것이라 생각합니다.

MPW 104회에 제출했던 레이아웃과 '프라이머 디자인'에서 실무자와 함께한 레이아웃을 비교를 해본 것을 나타낸 것이 아래 그림1과 그림2입니다. 우선 그림1 같은 경우 각 블록 별로 체계가 잡혀 있지 않고 각 블록을 아무런 규칙 없이 배치하여 레이아웃을 한 모습이 보입니다. 한눈에 보기에도 각 블록 별 공간도 무작위적으로 많이 남아있고, IP의 가로, 세로 길이보다 남은 공간이 남아 있는 것을 알 수 있습니다. 반면 실무자와 함께 진행하며 레이아웃 한 그림 2는 앞선 것과는 다르게 체계가 잡혀있고, 가로, 세로 길이에 딱 차있는 사각형의 모습을 하는 것을 알 수 있습니다.

이렇게 변화가 있을 수 있는 이유는 레이아웃을 하기 전에 Flow-plan을 정하고, 블록별 크기 또한 일정하게 맞춰서 최적의 조건으로 가장 반듯한 모양이 나오도록 하기 때문입니다. 그뿐만 아니라 디지털 입/출력 파트와 아날로그 입/출력 파트들을 구분하여 디지털 블록에서 유발되는 잡음이 아날로그 블록의 동작 특성에 영향을 주지 못하도록 하였습니다. 이와 같은 모든 사항은 단지 연구실에서 석사생들만 모여서 할 때와는 다른 부분들이었습니다. 모르는 부분에 대해 여쭙볼 멘토가 있다는 것이 가장 컸고, 거기에 회사 내에서 함께 출/퇴근을 하며 진행하다 보니 회사생활 및 업무에 대한 것도 직/간접적으로 경험할 수 있었습니다.

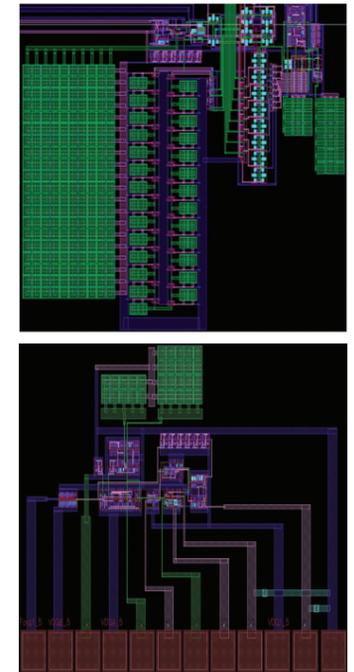


그림 1. 실무자 멘토링 전 Layout View

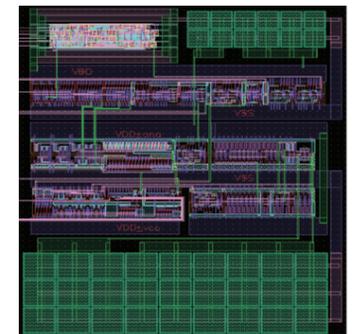
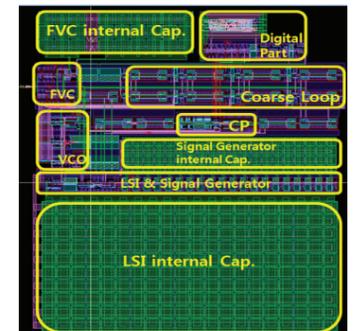


그림 2. 실무자 멘토링 후 Layout View



남정훈 학생
 소속 : 부경대학교 전자공학과 석사과정 (VLSI system lab)
 지도교수 : 최영식 교수
 연구분야 : Analog IC (PLL)
 E-mail : namjh3982@gmail.com
 http://ect.pknu.ac.kr/~vlsi