반도체설계교육센터
IC DESIGN EDUCATION CENTER

IDEC Newsletter

IDEC Newsletter | 통권 : 제188호 발행일 | 2013년 1월 31일 발행인 | 박세철 편집인 | 김이섭 제 작 | 푸울디자인
기 획 | 전형기 전 화 | 042) 350-8535 팩 스 | 042) 350-8540 홈페이지 | http://idec.or.kr
E-mail | jhg0929@idec.or.kr 발행처 | 반도체설계교육센터(IDEC)

Vol.188

2013 February

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

모바일 애플리케이션 프로세서 설계 기술

스마트 기기들의 두뇌 역할을 수행하면서 애플리케이션을 실행해주는 핵심 부품이 바로 애플리케이션 프로세서 (Application Processor, AP)이다. 이 AP는 여러 가지 구성 요소들로 이루어지는데 특히 AP에 내장된 CPU와 GPU 같은 핵심 모듈에 대한 설계가 AP의 경쟁력을 결정짓는 중요한 요인이 된다. AP에서는 점점 다양하고 복잡해지는 애플리케이션들을 실행하기 위해 고성능화가 매우 중요한 설계변수이면서도 이동 단말기에서 사용되는 특성 상 저전력 설계가 필수적인 기술이 된다. 본 고에서는 AP 설계를 위한 고성능 저전력 CPU 및 GPU 설계기법에 대해 살펴보고자 한다. (관련기사 P04~08참조)

반도체 메모리(DRAM) Refresh 및 연구동향

현대의 메인 메모리는 DRAM 셀로 구성된다. DRAM 셀은 커패시터에 데이터를 충전하여 저장한다. 이때 DRAM셀에 저장된 데이터는 셀 자체의 누설 전하에 의해 시간이 흐름에 따라 소멸된다. 이런 현상을 방지하기 위해 DRAM에 저장된 데이터는 주기적으로 읽고 다시 쓰는데 이러한 일련의 작업을 리프레시(refresh)라 한다. 이와 같이 DRAM의 집적도는 4Gbit~16Gbit 에 달하고 향후 더욱 증가될 것이다. 본 고에서 DRAM 리프레시 방법에 대해 간략히 알아보고 시스템에서 리프레시의 영향을 최소화 하기 위한 연구 동향에 대해 알아본다. (관련기사 P10~15 참조)

시스템 및 메모리 반도체 칩에 적합한 sub-20-nm 반도체 소자 구조

대용량 고성능 서버급 컴퓨터부터 개인용 컴퓨터(PC) 및 스마트폰(Smart-phone) 등에 이르기까지, 일상생활에 널리 쓰이는 디지털 기기들의 동작을 책임지고 있는 시스템 및 메모리 반도체 집적회로 칩은, 지난 50년간의 지속적인 반도체 소자/공정 기술발전에 힘입어, 최근에는 단위 칩당 약 10^9개 이상의 반도체 소자들로 구성되어 있는 수준에 이르렀다. 이러한 시스템 및 메모리 반도체 칩 내에서 가장 핵심적인 역할을 수행하는 전자 소자는 "트랜지스터"라 할 수 있다. 여러 종류의 트랜지스터들 가운데, 전계(electric field)를 이용한 MOSFET가 오늘날 가장 널리 사용되고 있는 대표적인 트랜지스터이다. 본 고에서는 차세대 20-nm이하급 반도체 공정 기술을 사용하여 제작될 시스템 및 메모리 반도체 칩에서 활용될 수 있는 다양한 종류의 차세대 실리콘 MOSFET 소자들의 구조, 디자인, 특성 등을 비교분석한다. (관련기사 P16~19 참조)

IDEC news February | 2013

MPW (Multi-Project Wafer)

MPW 설계 현황 |

구분	공정	제작가능 면적 (㎟x 칩수)	채택 팀수	설계면적 (㎟ x 칩수)	DB마감	Die-out	비고
117회 (12-10)	삼성 65nm	20개서버 (4x4㎟)	20 (서버)	4x4㎟x21	2012. 12.17	2013. 5.3.	DB 전달 : 1.24
118회 (13-01)	M/H 0.18	4.5x4㎟x20	23	4.5x4㎟x17 4.5x2㎟x6	2013. 2.18	2013. 7.22	
	동부 0.35BCD	5x2.5㎟x6	6	5x2.5㎟x6	2013. 2.27	2013. 6.12	
119회 (13-02)	TJ0.18 SiGe	2.5x2.5㎟x4	4	2.5x2.5㎟x4	2013. 3.12	2013. 7.1	
	동부 0.11	5x2.5㎟x24	28	5x2.5㎟x20 2.5x2.5㎟x8	2013. 3.20	2013. 7.31	
	삼성 65nm	4x4㎟x48	28	4x4㎟x28	2013. 3.15	2013. 8.15	
120회 (13-03)	M/H 0.18	4.5x4㎟x20	21	4.5x4㎟x19 4.5x2㎟x2	2013. 2.18	2013. 7.22	
	동부 0.35BCD	5x2.5㎟x6	10	5x2.5㎟x2 2.5x2.5㎟x8	2013. 5.1	2013. 5.15	
	동부 0.18BCD	5x2.5㎟x4	4	5x2.5㎟x4	2013. 5.15	2013. 5.29	
	TJ0.18 CIS	2.5x2.5㎟x4	4	2.5x2.5㎟x4	2013. 5.6	2013. 5.13	
	TJ0.18 RF	2.5x2.5㎟x4	4	2.5x2.5㎟x4	2013. 5.20	2013. 5.27	
	TJ0.18 BCD	2.5x2.5㎟x4	4	2.5x2.5㎟x4	2013. 5.20	2013. 5.27	

* 현재 모집 공정(~1. 30마감) : 121회 (13-04), 122회 (13-05) – 정규모집, 125회(13-08) : 우선모집

MPW 칩 제작 현황 |

구분	공정	제작 칩수	제작면적 (㎟x칩수)	Die-out 예정일	현재상태	비고
	동부 0.18BCD	4	5x2.5㎟x4	2012. 9.26	제작중	
114회 (12-7)	M/H 0.18	20	4.5x4㎟x20	2012. 12.31	제작중	-Die chip:1.7 -PKG:1.24
	삼성 0.13	32	4x4㎟x32	2013. 1.25	제작중	
115회 (12-8)	동부 0.18BCD	2	5x5㎟x2	2013. 1.4	제작중	
	동부 0.35BCD	8	5x2.5㎟x4 2.5x2.5㎟x4	2013. 1.16	제작중	-Die chip:1.8
	동부 0.11	30	5x2.5㎟x25 2.5x2.5㎟x5	2013. 2.6	제작중	
116회 (12-9)	TJ0.18 CIS	4	2.5x2.5㎟x4	2013. 2.22	제작중	
	TJ0.18 BCD	1	5x5㎟x1	2013. 2.29	제작중	
	TJ0.18 RF	4	2.5x2.5㎟x4	2013. 2.29	제작중	
117회 (12-10)	M/H 0.18	19	4.5x4㎟x19	2013. 3.4	제작중	
	M/H 0.35	18	5x4㎟x18	2013. 3.4	제작중	

* M/H = 매그나칩/SK하이닉스, TJ : TowerJazz
* 기준 : 2013. 2. 28

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

20th 한국반도체학술대회 Chip Design Contest 개최

1. 일정 및 장소
   - 일 시 : 2012년 2월 5일(화), 10:00~16:00 (*참고 : KCS 일정 _ 2.4~6)
   - 장 소 : 횡성 웰리힐리파크 루비 1-2호(5층) ((구)성우리조트, 강원도 횡성)
2. 최종 참여팀

   80팀 (Demo : 11 Panel : 69)
3. 시상 내역

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

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

2013년 2월 교육프로그램 안내

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

센터별 강좌 일정 |

센터명	강의일자	강의제목	분류
KAIST IDEC	2월 13일-15일	CMOS 아날로그 회로 설계 및 실습	설계
	2월 18일-19일	Verilog HDL을 활용한 IP 설계	설계
	2월 20일-22일	Mixed Analog Layout	설계
	2월 25일-26일	High speed broadband transceiver IC design technique	설계
	2월 27일-28일	Intuitive analysis of analog and RF circuits based on industrial practice in Silicon Valley	설계
경북대	2월 5일-7일	안드로이드 플랫폼 설계 방법 및 하드웨어 제어 응용설계	
부산대	2월 13일-15일	Verilog HDL을 이용한 SoC 설계	설계

▷KAIST IDEC 개설 강좌 안내

- **강좌일 : 2월 13일-15일**
- **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] [강의수준]
· 석사과정 신입생 · 초급

[강의형태] [사전지식, 선수과목]
· 이론+실습 · 회로이론, 전자회로 1, 2

- **강좌일 : 2월 18일-19일**
- **강좌 제목 : Verilog HDL을 활용한 IP 설계**
- **강사 : 김지훈 교수(충남대)**

[강좌개요]
Verilog HDL의 기초 및 효율적인 IP설계를 위한 방법론
– Verilog HDL 기초 – IP 설계시 고려사항 – Coding Guideline for Synthesis

[수강대상] [강의수준]
· 학부생 및 석사과정 · 중초급

[강의형태] [사전지식, 선수과목]
· 이론+실습 · 논리회로 및 컴퓨터구조

- **강좌일 : 2월 20일-22일**
- **강좌 제목 : Mixed Analog Layout**
- **강사 : 박의근 이사(파인스)**

[강좌개요]
경험을 바탕으로 한 Layout 진행시의 중요점을 설명하고, 실습을 통하여 체득하게 함으로써, 실무에 적용할 수 있도록 하는데 목표를 갖는다. 또한 특성과 원가 개념 모두에 대한 고취가 이루어지도록 한다.

[수강대상] [강의수준]
· Analog Design 관련 석박사 과정 · 중초급

[강의형태] [사전지식, 선수과목]
· 이론+실습 · Mixed Analog Circuit 이해
· Layout Editor 및 Verification Tool 사용 필수

- **강좌일 : 2월 25일-26일**
- **강좌 제목 : High speed broadband transceiver IC design technique**
- **강사 : 배현민 교수(KAIST)**

[강좌개요]
광대역 송수신기와 관련된 통신이론과 구현방법 그리고 각 구성요소들(PLL, equalizer, MUX/DeMUX, VGA)에 관한 설계 기법을 배운다.

[수강대상] [강의수준]
· 석박사과정 대학원생, 회사원 · 중급

[강의형태] [사전지식, 선수과목]
· 이론 · 전자회로, 디지털 통신

- **강좌일 : 2월 27일-28일**
- **강좌 제목 : Intuitive analysis of analog and RF circuits based on industrial practice in Silicon Valley**
- **강사 : 박진호 박사(Marvell Semiconductor)**

[강좌개요]
반도체 산업은 점차적으로 더 빠른 속도와 고주파수, 그리고 고성능의 집적회로들이 주류를 이루고 있고, 이 같은 경향은 앞으로도 더욱 가속화될 전망이다. 고주파수회로 설계자들, 특히 RF 디자이너들은 크게 두 가지 부류를 이루어왔다. 첫째는 microwave분야에서 접근하는 디자이너, 둘째는 analog회로설계분야에서 접근하는 부류들이다. 하지만, IC의 고집적화가 가속화되면서, analog회로설계자들의 RF 지식이 더욱 요구되는 것이 현실이다. 이번 강의는 국내의 analog /RF 회로설계 분야에 종사하거나 관심을 갖는 이들에게 analog의 기초와 RF 설계의 기본을 완전히 다른 각도에서 다시한번 다지는 내용들로 이루어질 예정이다. 저전력Wireless와 high-speed 통신용 IC에 관한 실리콘벨리 최신 반도체 디자인 트렌드를 예제로 삼 직관적 분석력 향상을 위한 analog/mixed-signal design fundamental 지식 습득을 목표로 한다.

[수강대상]
· Analog / RF Design Engineering 백그라운드를 가진 전문인, 직장인, 대학원생

[강의수준] [강의형태]
· 초중급 · 이론

[사전지식, 선수과목]
· Fundamental understanding about CMOS devices, analog circuitries, RF theory.

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

▷경북대 IDEC 개설 강좌 안내

- **강좌일 : 2월 5일-7일**
- **강좌 제목 : 안드로이드 플랫폼 설계 방법 및 하드웨어 제어 응용**
- **강사 : 홍성현(휴인스)**

[강좌개요]
임베디드 시스템에 대한 전반적인 이해 및 임베디드 리눅스기반의 디바이스 드라이버 개발 과 안드로이드 기반의 디바이스 제어

[수강대상] [강의수준] [강의형태]
학부 및 대학원생 · 초급 · 강의 및 실습 연동

* 문의 : 경북대 IDEC 주현아, 변보련 (053-950-6857~8, ipc-mobile@ee.knu.ac.kr)

▷부산대 IDEC 개설 강좌 안내

- **강좌일 : 2월 13일-15일**
- **강좌 제목 : Verilog HDL을 이용한 SoC설계**
- **강사 : 강봉순 교수(동아대)**

[강좌개요]
디지털 하드웨어 설계의 국제 표준언어인 Verilog HDL의 기초이론과 응용에 대하여 학습한다. 습득한 이론을 바탕으로 기능 블록을 설계하고, 최종적으로 디지털 전자시계의 전체 기능을 Verilog HDL을 이용하여 설계한 후에 Training Kit에 Download 하여 기능을 확인한다.

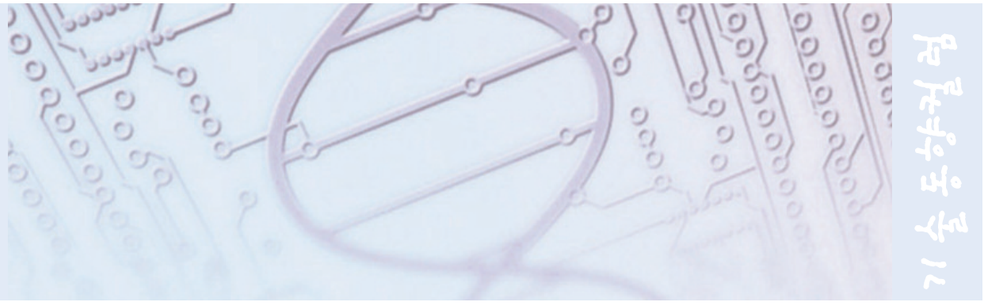
[수강대상] [강의수준] [강의형태]
– 2학년 수료자 · 중급 · 이론+실습
– 디지털논리회로 설계 초보자

* 문의 : 지화준 (051-510-2828, idec@pusan.ac.kr)

모바일 애플리케이션 프로세서 설계 기술

충남대학교 컴퓨터공학과

남병규 교수
연구분야 : 모바일 GPU, 임베디드 CPU, 애플리케이션 프로세서 (AP), SoC 설계
E-mail : bgnam@cnu.ac.kr
http://mssl.cnu.ac.kr



서론

최근의 "스마트폰 혁명"을 필두로 하여 앱스토어 (App Store) 기반의 스마트폰 및 태블릿과 같은 스마트 기기들이 일상생활 속에 깊숙히 자리잡아가고 있다. 이러한 기술의 발전은 생활 속에서 다양한 애플리케이션 (Application)의 활용을 가능토록 하여 사용자들의 생활방식을 크게 바꾸어 놓고 있는데, 이러한 스마트 기기들의 두뇌 역할을 수행하면서 애플리케이션을 실행해주는 핵심 부품이 바로 애플리케이션 프로세서 (Application Processor, AP)이다. 이 AP는 여러 가지 구성요소들로 이루어지는데 특히 AP에 내장된 CPU와 GPU 같은 핵심 모듈에 대한 설계가 AP의 경쟁력을 결정짓는 중요한 요인이 된다. AP에서는 점점 다양하고 복잡해지는 애플리케이션들을 실행하기 위해 고성능화가 매우 중요한 설계변수이면서도 이동 단말기에서 사용되는 특성 상 저전력 설계가 필수적인 기술이 된다. 최근 삼성과 Qualcomm 및 Nvidia 등의 애플리케이션 프로세서 예에서도 볼 수 있듯이 이미 1GHz 이상의 고성능 프로세서 코어들이 탑재되고 있으며 전력소모 또한 sub-W 대로 최적화되어 있는 것을 볼 수 있다. 본 컬럼에서는 이러한 AP 설계를 위한 고성능 저전력 CPU 및 GPU 설계기법에 대해 살펴보고자 한다 [1],[2].

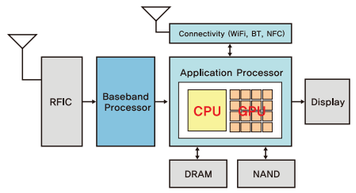


그림1. 모바일 AP 구조

본론

모바일 AP 설계

최근 GPU가 주요한 컴퓨팅 자원으로 등장하면서 이를 활용한 이종 컴퓨팅 (heterogeneous computing)이 컴퓨터의 성능을 향상시키는 방안으로 주목 받고 있다. 기존의 컴퓨터가 CPU 중심으로 버스시스템과 메모리, 스토리지 등의 구성요소를 통해 연산을 수행하던 반면, 그래픽 처리를 담당하기 위해 도입된 GPU가 그래픽스 이외의 다양한 연산을 수행할 수 있게 되면서 컴퓨터 시스템에는 CPU 외에 또 하나의 컴퓨팅 자원으로 GPU가 등장하게 되었다. 이러한 GPU의 등장으로 인해 멀티미디어 응용처럼 기존의 CPU가 처리하는데 부담을 가졌던 병렬성이 높은 연산들을 GPU가 담당할 수 있게 되면서 컴퓨팅 성능에 획기적인 발전을 가져오고 있다. 이러한 GPU의 등장은 모바일 AP에서도 부각되면서 그림1에서 보인 바와 같이, 각종 스마트폰 및 태블릿용 AP에도 이미 GPU가 내장되어 사용됨으로써 이종 컴퓨팅을 위한 발판을 마련하고 있는데, 본 고에서는 모바일 AP 관점에서 효과적인 이종 컴퓨팅을 구현하기 위한 CPU 및 GPU의 설계 기법을 살펴보고자 한다.

$$S = \frac{1}{(1-\alpha)+\frac{\alpha}{N}} \qquad (1)$$

식 1에서 보인 암달의 법칙 (Amdahl's law)에 따르면 어떤 프로그램에서 병렬화 할 수 있는 부분의 비율을 α라고 했을 때 프로그램을 수행하는 시스템의 성능은 그 프로그램을 병렬화하여 처리하는 부분의 성능 (α/N)과 직렬로 처리하는 부분의 성능 $(1-\alpha)$의 합으로 결정된다는 것을 알 수 있다. 즉, 이런 경우 병렬처리 성능과 직렬처리 성능이 모두 향상되었을 때 가장 높은 성능향상을 기대할 수 있으며, 어느 한쪽에서만 성능향상이 이루어졌을 경우에는 다른 쪽의 성능이 병목현상을 일으켜 기대했던 만큼의 성능향상을 이루기가 어렵게 된다. 따라서 AP에서 직렬처리와 병렬처리를 각각 담당하는 CPU와 GPU의 성능을 함께 개선해야 원하는 성능향상을 이루게 되는데 서로 다른 특성을 가지는 이들의 성능향상을 위해서는 각각 서로 다른 설계전략을 가져갈 필요가 있다. 주로 직렬연산을 담당하는 CPU는 작업을 시작하여 끝나는데 걸리는 시간 즉, 레이턴시(latency)가 중요한 성능잣대가 되는 반면, 주로 병렬처리를 담당하는 GPU는 많은 양의 병렬작업이 존재하므로 하나의 작업을 처리하는 레이턴시 보다는 전체 병렬작업을 처리하는데 걸리는 시간이 중요하게 되어 스루풋(throughput) 즉, 단위 시간당 얼마나 많은 결과를 낼 수 있는지가 중요한 성능잣대가 된다.

한편, 레이턴시를 결정하는 요인들을 살펴보면 사이클 수(cycle count)와 사이클 시간(cycle time)이 중요한 결정요소임을 알 수 있는데, 작업이 시작하여 끝나는데 몇 사이클이 걸리는 지와 하나의 사이클 시간이 얼마나 되는지가 전체 작업을 처리하는데 걸린 시간(즉, 사이클 수×사이클 시간)을 결정하게 된다. 이러한 사이클 수는 주로 프로세서의 아키텍처에 의해서 결정되며 사이클 시간은 프로세서를 구현하는 회로에 의해서 결정되므로, CPU 설계는 사이클 수와 사이클 시간을 최소화하는 방향으로 아키텍처와 회로설계가 각각 병행되어야 함을 알 수 있다. 한편, GPU는 스루풋 즉, 단위 시간당 처리량을 높이는 방향으로 설계가 이루어져야 하므로 아키텍처와 회로설계 모두 코어개수를 최대한 많이 가져갈 수 있는 방향으로 진행되어야 함을 알 수 있다.
또한, 모바일 AP는 사용 환경의 특성상 성능뿐만이 아니라 전력소모 측면에서도 최적화가 필수적이어서 저전력 설계를 함께 고려해야 한다. 스마트폰과 같은 휴대기기들은 대기상태에 머무는 시간이 상대적으로 긴 만큼 누설전류에 대한 고려가 매우 중요하다. 특히 공정이 미세화됨에 따라서 누설전류가 크게 증가하고 있는데 최근에는 동작 상태에서 소모하는 전력에서도 누설전류가 차지하는 비중이 절반에 이를 정도로 크게 증가하고 있다. 따라서, 모바일 AP를 위한 저전력 설계는 트랜지스터의 누설전류를 줄이기 위한 다양한 기법들이 적용된 저전력 공정 (LP process)을 기반으로 진행되어야 하는데, 이러한 저전력 공정은 트랜지스터의 누설전류뿐만 아니라 포화전류에도 함께 영향을 끼치게 되므로 이에 대한 보상차원에서도 앞서 소개한 성능향상 기법이 함께 적용되어야만 원하는 AP의 성능 및 전력소모 기준을 만족시킬 수 있게 된다.

모바일 CPU 설계

앞서 소개한대로 모바일 CPU 설계는 레이턴시를 최적화하는 방향으로 진행되는 것이 중요하므로 이 장에서는 아키텍처 및 회로설계 측면에서 모바일 CPU에 대한 최적화 방안들을 살펴보겠다.

CPU 아키텍처

아키텍처 측면에서 성능을 최적화하기 위한 방안으로 기존의 고성능 CPU에서 사용하던 예측 수행 (specular execution), 비순차 수행 (out-of-order execution), 수퍼스칼라 파이프라인 (superscalar pipeline) 등의 고성능 파이프라인 구조(그림2)가 이미 모바일 CPU에 채택되어 사용되고 있다[3].

먼저, 예측 수행은 분기 명령의 경우가 대표적인데, 명령어 페치 (instruction fetch) 단계에서 분기명령을 페치한 경우 분기를 할 것인지에 대한 여부가 결정되어야 다음 명령어를 가져올 수가 있다는 데서 출발한다. 분기여부에 대한 판단은 분기 명령어가 프로세서 파이프라인의 후반부에 있는 실행 단계 (execution stage)까지 진행되어야 가능한데 이 동안 프로세서 파이프라인은 다음 명령을 가져와서 실행하지 못하고 멈추어 있게 되는 문제점을 안고 있다. 따라서 이 시간 동안 파이프라인을 멈추어두는 대신에 분기여부를 예측함으로써 파이프라인의 성능을 높이는 방안으로 대두된 것이 예측수행 기법이다. 분기예측을 위해서는 각 분기명령에 대한 과거의 분기이력과 분기목적지를 저장해 둘 메모리가 필요하며, 분기가 잘 못 되었을 경우 이를 복구할 수 있는 장치가 마련되어 있어야 한다. 한편, 비순차 수행은 상호 의존성이 없는 명령어들이 명령어 순서에 관계없이 수행될 수 있도록 만들어 주는 것으로서 앞서 진행되던 명령어가 캐쉬 미스 (cache miss) 등의 이유로 인해 파이프라인에서 진행되지 못하고 있을 경우 이와 의존성이 없는 다음 명령어들은 이를 기다릴 필요 없이 먼저 수행될 수 있도록 하여 파이프라인의 성능을 높이는 방법이다. 이와 더불어 고성능 파이프라인의 대표적인 구조 중 하나인 수퍼스칼라 구조는 상호간 의존성이 없는 명령어들이 파이프라인 상에서 동시에 수행될 수 있도록 하여 파이프라인의 성능을 더욱 높이고자 하는 것인데, 여기서 소개한 각각의 기법은 개념적으로는 서로 독립적이지만 실제 설계에서는 서로 맞물려서 사용되어 CPU 파이프라인의 성능향상을 극대화하고 있다.

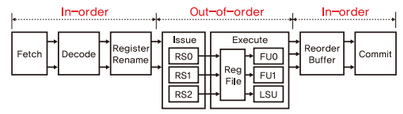


그림2. 고성능 CPU 파이프라인 구조

고속 회로 설계

CPU의 파이프라인 스테이지를 구성하는 요소들을 살펴보면 크게 세 가지로 분류할 수 있는데, ALU와 같은 기능블럭을 구성하는 조합논리회로 (combinational logic), 파이프라인 레지스터를 구성하는 플립플롭, 그리고 캐쉬메모리를 비롯한 각종 버퍼 메모리를 구성하는데 사용되는 SRAM 등으로 분류된다. 이러한 블록들을 고속화하기 위해서는 주로 다이나믹 로직 (dynamic logic)에 기반한 회로들을 활용할 수 있는데, 다이나믹 로직은 NMOS 네트워만을 사용함으로써 스위칭에 필요한 문턱전압을 낮추고 고속동작을 가능하게 한다. 이 장에서는 고속 조합논리회로를 설계하는데 주로 사용되는 도미노 로직 (domino logic), 파이프라인 레지스터를 고속화하기 위한 세미-다이나믹 플립플롭 (semi-dynamic flip-flop), 고속 SRAM 설계를 위한 계층적 비트라인 (hierarchical bitline) 기법들을 소개한다. ALU와 같은 조합논리회로를 설계하기 위해서는 도미노 로직 (domino logic)을 유용하게 활용할 수 있는데, 도미노 로직은 잘 알려진 바와 같이 충전 페이즈 (precharge phase)와 계산 페이즈 (evaluation phase)로 구간을 나누어 동작한다. 이러한 도미노 로직은 ALU 등을 구현하는데 필요한 반전논리(inverting logic)를 구현할 수 없다는 문제를 가지고 있다. 이에 대한 해결책으로 듀얼-레일 (dual-rail) 도미노가 제안됨으로써 참과 거짓 논리에 대한 회로를 모두 구현하여 도미노 로직이 가졌던 문제점을 해결할 수 있다. 한편, 이를 N-레일 (N-rail)로 일반화하여 확장하면 흥미로운 특징들을 보이게 되는데 표 1에 나타난 바와 같이 듀얼-레일 인코딩에 비하여 N-레일 인코딩이 스위칭 확률을 낮추게 되어 더욱 저전력으로 동작하는 회로를 구현할

수 있음을 확인할 수 있다[4]. 또한, 도미노 로직은 시간 빌림 (time borrowing) 기법을 사용할 수 없다는 점이 단점으로 지적된다. 시간 빌림 기법은 래치(latch) 기반의 파이프라인에서 스테이지간에 발생하는 여분의 슬랙(slack)을 이를 필요로 하는 쪽으로 할당해 줌으로써 타이밍을 최적화하는 방법인데, 래치 기반으로 동작하는 도미노 로직에서도 이러한 시간 빌림 기법을 이용하면 더 높은 성능향상을 기대할 수 있으므로 이를 구현하기 위해 겹침 클럭 (overlapped clocking) 기법이 제안되었다[5]. 이는 그림 3에서 보인 바와 같이 서로 다른 페이즈 (phase)의 클록을 일정 부분 겹치도록 만듦으로써 클록이 겹치는 구간에서는 시간 빌림이 가능하도록 하였다. 또한, 첫 번째 페이즈 로직이 충전 상태로 들어가기 전에 이의 연산결과가 두 번째 페이즈 로직에 전달될 수 있도록 하여 페이즈 경계에 삽입되던 래치를 제거해주는 효과도 가져온다.

Data Value	Dual-rail Dynamic 4 wires 2 wires switch				N-rail Dynamic 4 wires 1 wire switches			
Null	A1	$\overline{A1}$	A0	$\overline{A0}$	A3	A2	A1	A0
0	0	1	0	1	0	0	0	1
1	1	0	0	1	0	0	1	0
2	0	1	1	0	0	1	0	0
3	1	0	1	0	1	0	0	0

표1. N-레일 인코딩

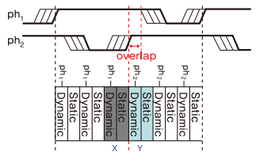


그림3. 겹침 클럭 (Overlapped Clocking)

또한, 파이프라인 설계에서 빠질 수 없는 부분이 파이프라인 레지스터 구현을 위한 플립플롭의 설계인데, 고속 파이프라인의 경우 플립플롭의 지연시간이 사이클 시간의 30%에 육박하므로 이를 줄이기 위한 고속 플립플롭의 설계가 반드시 필요하다. 펄스 기반의 플립플롭은 단순한 구조로 인하여 고속동작이 용이한데, 그림 4(a)에서 보인 바와 같이 하나의 래치와 펄스 발생기만으로 이루어진다. 특히, 그림 4(b)에서 보인 세미-다이나믹 (semi-dynamic) 구조를 사용하면 다이나믹 회로를 이용하여 래칭노드를 구현하므로 스위칭 속도를 더욱 빠르게 할 수 있다. 또한, 이는 플립플롭 내부에 필요한 로직을 내장할 수 있는 장점도 가지고 있어서 고속화에 유리한 구조를 가진다.

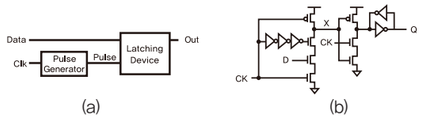


그림4. 펄스 기반의 플립플롭

마지막으로 SRAM과 같은 메모리 구조에 대해서는 계층적 비트라인 (hierarchical bitline) 기법을 도입함으로써 고속화를 이룰 수 있다. 캐쉬 메모리는 고속 CPU 파이프라인을 구현 시 병목을 일으키는 주요 지점 중의 하나인데, 이런 캐쉬 메모리를 구현하는데 사용되는 SRAM은 최근의 공정이 미세화되면서 고속화하는 것이 더욱 어려워지고 있다. 고속 SRAM 구현이 어려워지는 근본적인 원인은 비트라인의 스케일링이 용이하지 않고 공정 변이가 심화되면서 센스앰프 (sense amplifier)의 센싱 마진 (sensing margin)이 점차 증가하고 있는데 따른 것이다. 따라서 하나의 비트라인에 많은 비트셀 (bitcell)들을 연결하는 기존의 구조에서 벗어나서 비트라인을 계층적으로 분리하여 지역 비트라인 (local bitline)과 전역 비트라인 (global bitline)으로 나누고, 지역 비트라인에 연결되는 비트셀의 수를 16 ~ 32개로 줄여줌으로써 대신호 센싱을 가능하게 하여 센스앰프를 없앨 수 있고 따라서 고속화가 가능하게 된다. 그림 5는 계층 비트라인의 구조를 보여준다.

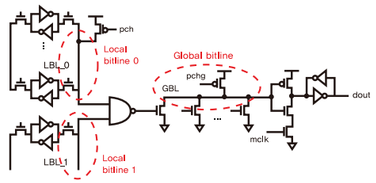


그림5. 계층적 비트라인 구조

모바일 GPU 설계

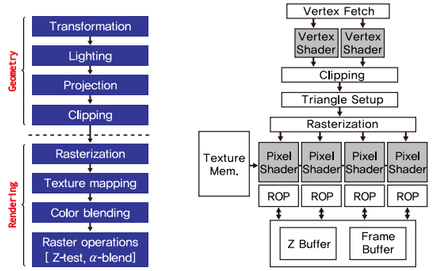
이 장에서는 모바일 GPU의 스루풋을 최적화하는 방향에 대해 살펴보고자 한다. 우선 GPU 파이프라인의 근간을 이루는 3차원 그래픽스 파이프라인에 대해 간략히 소개하고 이를 구현하는 GPU 아키텍처에 대해서 소개한다. 그리고 GPU의 스루풋을 최적화하기 위한 매니코어 아키텍처와 소면적 셀 라이브러리를 소개한다.

그래픽스 파이프라인

3차원 그래픽스를 처리하기 위한 표준 파이프라인 구조를 그림 6(a)에 보였다[6]. 3차원 그래픽스에서는 삼각형을 이용하여 물체를 모델링하고 이를 렌더링함으로써 사실감 있는 그림을 그려낸다. 그래픽스 파이프라인은 이러한 삼각형들을 입력으로 받아들여 크게 기하연산 과정과 렌더링 연산 과정을 거치도록 되어있다. 먼저 기하연산 과정에서는 화면상에서 삼각형이나 시점의 이동을 나타내기 위해 삼각형의 꼭지점들에 대한 변형(transform) 과정을 거치게 되며 이를 통해 삼각형 꼭지점들의 이동된 좌표 값을 구할 수 있다.

이렇게 이동된 꼭지점들은 광원과의 상관관계를 계산하여 색깔 값을 구하게 되는데 이를 조명(lighting) 연산이라고 한다. 이는 물체가 광원을 향하고 있는 정도와 관찰자 방향으로 빛을 반사하는 정도를 계산함으로써 물체의 밝기를 결정하고, 물체가 가진 물질상수와 광원의 반응을 계산하여 최종 색깔 값을 계산하게 된다. 이렇게 삼각형 세 꼭지점들의 위치와 색깔을 결정하고 나면 삼각형을 2차원 화면으로 투영하는 과정 (projection)과 화면 밖으로 벗어나는 삼각형의 부분을 잘라내는 과정 (clipping)을 거쳐 렌더링 단계로 넘어가게 된다. 렌더링 단계에서는 먼저 삼각형의 내부를 채우는 픽셀(pixel)들을 생성하는 과정을 거치는데, 이를 래스터화(rasterization)라고 한다. 래스터화 과정에서는 앞서 기하연산 단계에서 구한 삼각형의 세 꼭지점이 가지는 좌표, 색깔 등의 인자들을 보간(interpolation)하여 삼각형 내부의 픽셀들에 대한 인자들을 구하게 된다. 이렇게 생성된 픽셀들은 더욱 사실감 높은 렌더링을 위해 텍스처 매핑(texture mapping) 과정을 거치게 되는데, 이는 실사 무늬 패턴을 물체에 입혀줌으로써 복잡한 무늬를 가지는 물체를 세밀한 모델링 없이도 사실감 있게 표현 가능하게 하는 효과적인 렌더링 기법이다.

이렇게 텍스처가 입혀지는 픽셀에 대해서는 원래의 픽셀 색깔 값과 텍스처의 색깔 값을 혼합해줌으로써 (color blending) 기하연산 과정에서 계산한 조명효과를 반영해줄 수가 있다. 픽셀이 최종적으로 화면에 그려지기 전에는 다른 물체에 의해 가려지는지 여부를 깊이 검사 (depth test, z test)를 통하여 판별하게 되며, 이 과정을 통해 실제 화면에 그려질 픽셀들만 최종적으로 프레임 버퍼(frame buffer)에 저장되어 화면에 나타나게 된다.



(a) 그래픽스 파이프라인 (b) GPU 구조

그림6. 그래픽스 파이프라인 및 GPU 구조

GPU 설계

지금까지 소개한 그래픽스 파이프라인은 고정된 기능만을 수행하는 파이프라인으로서 다양하게 변화하고 발전하는 최신의 그래픽스 알고리즘을 반영할 수 없다는 단점을 가지고 있다. 이를 극복하기 위해 프로그램 가능 파이프라인(programmable pipeline)의 개념이 도입되었는데 이는 그래픽스 파이프라인의 일정부분을 프로그램할 수 있도록 만듦으로써 다양한 그래픽스 알고리즘을 구현할 수 있도록 한 것이다. 프로그램 가능 파이프라인은 버텍스 셰이더(vetex shader)와 픽셀 셰이더(pixel shader)라는 수치연산에 최적화된 벡터 프로세서를 도입하고 있다.

버텍스 셰이더는 기하연산 단계의 변형(transformation) 및 조명(lighting) 연산을 프로그램할 수 있도록 만든 것이며, 픽셀 셰이더는 렌더링 단계의 텍스처 매핑과 블렌딩이 가능 하도록 만든 것으로서 이들을 적절히 프로그래밍하면 다양한 그래픽스 효과를 만들어낼 수 있는 장점이 있다. 그림 6(b)에서는 이러한 프로그램 가능 파이프라인을 반영한 GPU 구조를 나타내고 있다[6]. 이는 앞서 설명한 그래픽스 파이프라인의 스테이지들을 구현하고 있으며, 메모리 시스템으로 텍스처 이미지를 저장하기 위한 텍스처 메모리와 깊이 검사를 수행하는데 필요한 깊이 버퍼 (depth buffer) 그리고 최종 화면에 그려질 이미지를 저장하는 프레임 버퍼 (frame buffer)를 포함하고 있다.

GPU에 사용되는 버텍스 셰이더와 픽셀 셰이더는 모두 벡터형태로 주어지는 버텍스와 픽셀의 다양한 속성들을 처리하기 위하여 벡터 프로세서 구조를 취하고 있으며 4개의 부동 소수점 곱셉기를 SIMD 형태로 가지고 있다. 또한 그래픽스 연산에 필요한 다양한 초월함수들 (나눗셈, 제곱근, 삼각함수, 멱함수 등)을 구현하기 위한 초월함수 연산기 (special function unit, SFU)를 포함한다. 그리고 이러한 연산기들에게 데이터를 공급하고 연산결과를 저장하기 위한 다양한 버퍼/레지스터들이 존재한다. 특히, 픽셀 셰이더에서는 텍스처 매핑을 수행하기 위해서 텍스처 유닛이 추가되어 있는데 이는 최근 버텍스 연산에도 텍스처 매핑이 도입됨에 따라 버텍스 셰이더에도 공통적으로 필요한 블록이 되고있다.

이와 같이 버텍스 셰이더와 픽셀 셰이더가 거의 유사한 구조를 가지게 됨에 따라 두 가지 셰이더를 통합하려는 움직임이 발생하였는데, 이를 통합 셰이더 (unified shader)라고 부른다. 통합 셰이더를 이용하게 되면 처리할 화면의 작업량이 버텍스 또는 픽셀 어느 한쪽으로 몰리더라도 자연스레 로드 밸런싱(load balancing)이 이루어져 병목현상을 일으키지 않고 처리할 수 있는 장점이 있다. 또한 이러한 통합 셰이더를 이용하여 그래픽스 이외의 프로그램을 수행하도록 하는 움직임도 나타나고 있는데, GPGPU (general-purpose GPU) 기술이라고 불리는 이 기술은 멀티미디어 응용과 같이 데이터처리가 많은 그래픽스 이외의 응용분야에 적용되어 효과를 보이고 있다. 최근에는 모바일 GPU에서도 이러한 GPGPU 기술이 도입되고 있어 향후 모바일 증강현실 등의 분야에서 활용이 클 것으로 보인다.

이러한 GPGPU의 아키텍처는 앞서 논의한 것처럼 스루풋을 높이기 위한 방향으로 설계되어야 하므로 스루풋을 높이기 위해서는 기본적으로 코어 개수를 늘려주는 것이 가장 직접적인 방법이 된다. 따라서 GPU에 수십~수백 개의 코어를 집적하는 매니코어 아키텍처가 활발하게 연구되고 있으며 이는 SIMT (single instruction multiple threads)라는 프로세싱 모델로 나타나고 있다. 이는 기존의 SIMD (single instruction multiple data) 모델을 바탕으로 하여 각 코어가 서로 독립적인 분기(branch)를 수행할 수 있도록 한 것인데, SIMD와 MIMD의 중간형태를 취함으로써 각 코어가 독립적인 쓰레드 (thread)를 수행하도록 하면서도 각 코어의 컨트롤러 복잡도가 증가하는 것을 최소화하여 매니코어 구조에 적합하도록 하였다. 이와 더불어 GPU의 회로설계 측면에서도 고성능 보다는 집적도가 높은 표준 셀 라이브러리 (high-density standard cell library)를 사용함으로써 코어의 집적 수를 높이도록 하고 있다.

저전력 CMOS 기술

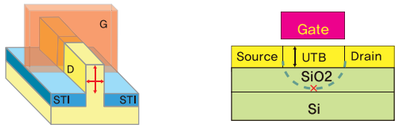
최근 공정기술이 급격히 미세화 됨에 따라 누설전류가 크게 증가하고 있어 이를 줄이기 위한 다양한 노력들이 전개되고 있다. 본 장에서는 누설전류를 줄이기 위해 공정개발 방향들을 소개하고 설계상의 대응 기법들을 살펴본다.

저전력 CMOS 공정

MOS 트랜지스터에서 발생하는 누설전류는 크게 3가지 요소로 이루어진다[7]. 첫째는 문턱전압 이하에서 발생하는 전류 (sub-threshold

current)로서 게이트 전극의 영향이 미치지 못하는 부분에서 발생하는 전류이다. 또 하나는 게이트 누설 전류로서 이는 공정 미세화에 따라 얇은 게이트 절연막을 터널링하여 흐르는 전류를 말한다. 마지막으로 소스(source) 및 드레인(drain)과 바디(body)의 정합(junction) 부분에서 발생하는 누설전류가 존재한다. 이들 각각에 대해 공정에서 적절한 대응 방법을 세워 관리하고 있는데 문턱전압 이하 전류에 대해서는 MOS 채널의 도핑 프로파일 (doping profile)을 조절함으로써 관리하고 있으며 게이트 누설전류는 고유전체-금속게이트 (high-k metal-gate, HkMg) 공정을 도입하여 대응하고 있다. 정합 부분의 누설전류는 아직 크게 이슈가 되고 있지는 않지만 소스 및 드레인 경계의 도핑 프로파일을 완만히 함으로써 대응할 수 있다. 하지만 이와 같이 채널의 도핑농도를 조절하는 기법은 도핑되는 불순물의 양에 의존하는 것인데 공정이 미세화 됨에 따라 불순물 량의 편차에 따른 공정변이(process variation)가 커지고 있어, 불순물 도핑에 의존하기 보다는 새로운 구조를 통해 게이트의 채널에 대한 영향력을 강화하려는 노력들이 나타나고 있다.

대표적인 것이 요즘 상용화되고 있는 FinFET (그림 7(a))과 UTB-SOI (그림 7(b))인데, FinFET은 트랜지스터의 구조를 삼차원으로 만들고 게이트가 채널을 삼면에서 감싸도록 하여 채널에 대한 게이트의 영향력을 극대화하였고, 이를 통해 누설전류를 감소시킬 뿐만 아니라 온 전류 (on corrent)도 증가시키는 효과를 가져왔다. 하지만 이러한 삼차원 구조는 공정개발비용이 크게 증가하는 단점이 있는 반면, UTB-SOI (ultra thin body SOI)는 전통적인 평면 트랜지스터 구조를 유지하면서도 트랜지스터의 바디(body)부분을 매우 얇게 만들어 게이트의 영향력 하에 있도록 하였고, 또한 바디의 아래쪽을 절연하여 게이트의 영향을 벗어나는 누설전류가 발생하지 못하도록 하고 있다.



(a) FinFET (b) UTB-SOI

그림7. FinFET 및 UTB-SOI

저전력 설계기법

공정기술뿐만 아니라 설계기술에서도 누설전류를 억제하기 위한 기법들이 적용되고 있는데 파워 게이팅 (power gating) 기법과 적응적 바디전압 인가(adaptive body biasing) 기법이 대표적이다. 누설전류를 억제하기 위해 설계 단계에서 적용할 수 있는 가장 효과적인 방법은 사용하지 않는 모듈의 전원을 차단해 주는 것이다. 전원을 차단하기 위해서는 PMOS를 사용한 헤더 스위치 (header switch)를 사용하거나 NMOS를 이용한 푸터 스위치 (footer switch)를 사용할 수 있는데, 헤더는 PMOS의 특성상 누설전류가 작은 장점을 가지는 대신에 NMOS에 비하여 면적이 커지는 단점이 있다.

반면, 푸터는 이와는 반대되는 특성을 가지므로 면적비용과 전압강하 (IR-dron) 제약 등 주어진 설계조건을 따져서 결정해야 한다. 한편, 그림 8에 보인 적응적 바디전압 인가기법은 트랜지스터의 바디전압을 조절해줌으로써 문턱전압을 조절하는 기법인데, 바디 전압을 낮추어 주면 문턱전압이 높아지는 성질을 이용하여 트랜지스터의 누설전류를 줄일 수 있게 된다. 반대로 바디 전압을 올려줌으로써 트랜지스터의 성능을 개선할 수도 있으므로 바디전압을 적절히 인가하여 누설전류를 줄이거나 성능을 개선하는데 사용할 수 있다. 하지만 일반적인 CMOS 공정에서는 NMOS의 바디가 분리되지 않으므로 이 방법을 광범위하게 적용하는 데에는 한계가 따르는데, 이를 극복하기 위한 트리플 웰 (triple well) 공정은 공정비용이 증가하는 단점을 가진다. 아울러 공정이 미세화되면서 바디팩터가 (body factor)가 점차 낮아지고 있어 바디 전압의 변화에 대한 문턱전압의 반응이 무뎌지고 있는 것도 미세공정에서 이 방법의 적용을 제한하는 요인이 되고 있다.

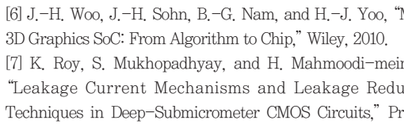


그림8. 적응적 바디전압 인가기법

결론

지금까지 모바일 AP에 대한 설계전략으로서 AP의 핵심 모듈인 CPU와 GPU에 대한 설계기법들을 살펴보았다. 정리하면, CPU는 레이턴시에 최적화되는 것이 바람직하며 이를 위한 고성능 아키텍처 및 고속 회로설계를 적용하는 것이 필요한 반면, GPU에 대해서는 스루풋을 최적화하는 것이 중요하여 이를 위한 매니코어 아키텍처 및 소면적 회로를 적용하는 것이 바람직하다. 특히, 이들은 모바일 기기의 특성상 저전력 구현이 중요한데 무엇보다도 누설전류를 줄이는 것이 매우 중요하여 LP 공정을 바탕으로 한 저전력 설계가 필수적임을 설명하였다. 한편, 이러한 LP 공정이 가져오는 성능감소를 보상하기 위해서도 CPU와 GPU의 고성능, 고출력 설계방법들이 다시 중요해짐을 알 수 있다.

Reference

[1] Byeong-Gyu Nam, "Mobile GHz Processor Design Techniques,"Tutorial at IEEE ISSCC, 2012.

[2] Byeong-Gyu Nam, "High-Performance Mobile CPU and GPU Design,"Tutorial at IEEE A-SSCC, 2011.

[3] ARM, Exploring the Design of the Cortex-A15 Processor.

[4] S. Horne, D. Glowka, S. McMahon, P. Nixon, M. Seningen, and G. Vijayan, "Fast$_{14}$ Technology: Design Technology for the Automation of Multi-Gigahertz Digital Logic,"IEEE ICICDT, 2004.

[5] D. Harris, "Skew-Tolerant Circuit Design," Morgan Kaufmann, 2000.

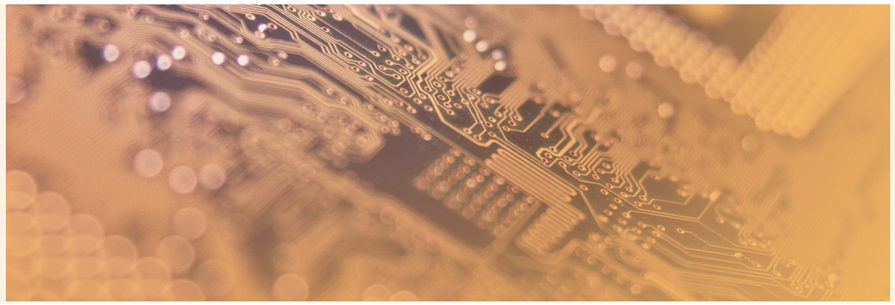
[6] J.-H. Woo, J.-H. Sohn, B.-G. Nam, and H.-J. Yoo, "Mobile 3D Graphics SoC: From Algorithm to Chip," Wiley, 2010.

[7] K. Roy, S. Mukhopadhyay, and H. Mahmoodi-meimand, "Leakage Current Mechanisms and Leakage Reduction Techniques in Deep-Submicrometer CMOS Circuits," Proc. of the IEEE, vol.91, no.2, Feb., 2003.

반도체 메모리(DRAM) Refresh 및 연구동향

용인대학교 컴퓨터과학과

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



서론

현대의 메인 메모리는 DRAM 셀로 구성된다. DRAM 셀은 커패시터에 데이터를 충전하여 저장한다. 이때 DRAM셀에 저장된 데이터는 셀 자체의 누설 전하에 의해 시간이 흐름에 따라 소멸된다. 이런 현상을 방지하기 위해 DRAM에 저장된 데이터는 주기적으로 읽고 다시 쓰는데 이러한 일련의 작업을 리프레시(refresh)라 한다.

DRAM 리프레시 동작은 대기모드(standby mode)에서도 에너지를 소모하고, 일반 메모리 동작을 지연시켜 시스템 성능을 저하시키는 문제를 가지고 있다. 대용량의 컴퓨터 시스템에서 DRAM의 저비용성과 대용량성 때문에 그 비중이 더욱 커지고, DRAM 용량이 증가할수록 이러한 문제는 시스템 성능에 더 큰 악영향을 준다.

프로세서 설계자는 패키지 상에서 메모리 지연과 대역폭문제를 완화시키기 위해 3D die-stack DRAM(또는 3D DRAM) 통합 쪽으로 방향 이동을 함으로서[1, 2], 리프레시 동작의 오버헤드는 증가한다. 3D DRAM은 마지막 레벨 SRAM-based 캐시와 시스템메모리 사이의 캐시로서 사용될 수 있다. 따라서 리프레시 동작은 상대적으로 큰 오버헤드가 될 것이다. 3D DRAM이 프로세서위에 직접 접착될 경우, 프로세서로부터 소멸되어야할 열이 DRAM으로 전도되어 DRAM의 동작 환경이 더 높은 온도에 노출된다.

Annavaram et al.는 64MB 3D DRAM 의 동작 온도가 90.27℃가 되는 것을 보여주었다[3]. ITSY 컴퓨터의 상세 전력분석에서 최소 전력모드에서도 리프레시 전력은 전체 DRAM 전력의 약 1/3가량을 소모하는 것으로 나타났다[4]. 게다가 메모리 셀에서의 누설은 동작 온도를 증가함으로 기하급수적으로 증가한다. Micron의 데이터 sheet에 의하면, 동작온도가 85℃를 초과하면 리프레시 속도가 두 배가 되어야 한다[5]. 따라서 3D SDRAM에서는 두배의 리프레시가 요구되어, 상대적으로 에너지 오버헤드를 증가 시킨다.

이전에는 하드웨어와 소프트웨어 양 측면에서 DRAM의 리프레시로 인한 문제를 공략하였다. 몇몇 하드웨어 만의 접근에서 DRAM 셀을 다른 리프레시 속도를 가지도록 DRAM 장치를 수정하였지만[6, 7, 8, 9], DRAM die에서 area overhead를 5~20% 증가를 초래하였고[7, 8], 생산 단가에 민감한 DRAM 시장에서 이를 적용하기도 쉽지 않다.

또 다른 방식의 접근은 메모리 제어기를 수정하여 불필요한 리프레시를 제거하거나 리프레시 속도를 감소시키거나, ECC[10, 11, 12]를 사용하여 retention error에 견딜 수 있도록 하였지만, 이것들도 대역폭 오버헤드 문제와 큰 저장 공간으로부터 어려움을 겪고 있다.

본 고에서 DRAM 리프레시 방법에 대해 간략히 알아보고 앞에서 살펴본 시스템에서의 리프레시의 영향을 최소화 하기위한 연구 동향에 대해 알아본다.

2. DRAM Refresh Techniques

2.1. Standard and Extended Refresh

DRAM 셀이 가지는 누설 전류의 영향으로 DRAM의 리프레시 주기는 앞에서도 언급하였지만 온도에 따라서 달라진다. 일반 서브에서 DRAM 동작 온도는 0℃~85℃범위이지만, 요즘에는 JEDEC (Joint Electron Device Engineering Council)에서도 확장된 온도 영역 85℃~95℃까지 포함하고 이 동작 영역이 서브 동작에서 일반화 되었다[13, 14]. 확장온도 영역에서는 DRAM 셀 데이터 유지시간이 표준 온도 환경보다 1/2로 줄어든다.

표 1에 DRAM 집적도에 따른 리프레시 간격(tREFI : REFresh Interval)과 리프레시 지연시간을 나타내었다. 표준 동작 영역에서, 각 DRAM 셀은 매 64ms마다 리프레시 되어져야 하는데, 메모리 제어기가 리프레시 동작을 인가함으로써 DRAM 내에 있는 리프레시 제어로직은 64ms내에서 리프레시가 완료되도록 모든 주소를 통해 순차적 단계를 반복한다.

메모리 컨트롤러가 리프레시를 인가해야하는 속도는 DRAM에 있는 행의 수에 64ms를 나누어 산출 되어 초기에 결정된다. 이 값이 tREFI이며 256M DDR2에서 7.8μs로 정의되었다.

DRAM의 집적도가 증가하면 행에 연결된 메모리 셀 수가 증가하여 리프레시 하는데 부하가 증가하여 리프레시 지연이 발생하는데 이러한 시간 지연을 tRFC(ReFresh Cycle time)라고 한다. 표 1에 메모리 집적도에 따른 리프레시 주기를 온도 영역별로 나타내었다.

DRAM Type.	tRFC	tREF@85℃	tREF@95℃
512Mb	90ns	7.8μs	3.9μs
1Gb	110ns	7.8μs	3.9μs
2Gb	160ns	7.8μs	3.9μs
4Gb	300ns	7.8μs	3.9μs
8Gb	350ns	7.8μs	3.9μs

표 1. 리프레시 집적도 증가에 따른 파라미터

2.2. Refresh Cycle Time

표 1에 tRFC(JEDEC DDR3) 값을 나타내었는데, 향후 DRAM집적도 증가에 따른 예측치를 tRFC 추세선으로 나타내었다. 512Mb와 4Gb의 사이를 잇는 추세선으로 부터 8Gb에 대한 tRFC는 550ns일 것으로 예측되며, 실제 JEDEC DDR3 spec.에는 350ns로 명시되어있다. DDR4에서는 tRFC가 확장되어야 할 필요성에 대해 JEDEC에서 논의 되고있다. 그림 1.에 tRFC 추이도를 나타내었다[15].

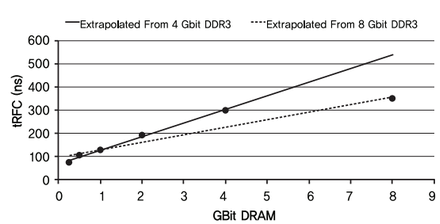


그림 1. DDR3 tRFC에 따른 추이 예측

DRAM에서 burst refresh와 distribute refresh 두가지 리프레시 모드가 있다.

Burst Refresh : 모든 행(row)이 하나씩 순차적으로 엑세스(access) 될 때 까지 일련의 리프레시 주기를 수행하여 burst 방식으로 달성된다. 리프레시 동안 다른 명령어는 허용되지 않는다. 따라서 리프레시 동안 DRAM 모듈은 일반 DRAM 동작을 수행할 수 없어서 일시적으로 성능을 감소시키는 원인이 된다. 게다가 DRAM의 최대 전력(peak power) 소모를 증가시켜 바람직하지 않다.

Distributed Refresh : 균일 간격으로 분산하여 리프레시 주기를 갖는 것을 distributed refresh라 하며, 매 7.8μs 주기로 실행된다. 이 방식은 메모리 제어기가 리프레시 간격에 걸쳐 균일하게 서로 다른 행에 대해 리프레시 주기를 분산시킨다. 따라서 적절한 시점에 각 DRAM 을 리프레시 함으로서 리프레시가 수행되지 않는 행은 외부 접속이 가능하고 일반 메모리 동작의 지연을 최소화할 수 있어서 더 유리하다. 부가적으로 DRAM 리프레시 주기는 별개의 두가지 방식으로 구현될 수 있다[16]. 리프레시 주기는 distribute모드나 burst모드로 실행될 수 있다는 것을 위에서 설명하였다. 그림 2에 위의 두 가지 리프레시에 대해 나타내었다.

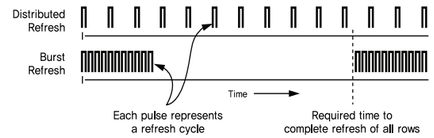


그림 2. Burst 와 Distribute Refresh

2.3. Refresh Cycles

앞에서 언급한 distribute 리프레시나 burst 리프레시 모두에 사용될 수 있는 여러 주기가 있는데, RAS-only 리프레시, CAS-Before-RAS 리프레시 및 Hidden 리프레시가 그것이다.

RAS-Only Refresh : RAS-Only 리프레시를 실행하기 위해 행의 주소를 주소라인에 싣고 RAS(Row Address Strobe) 신호를 low로 인가한다. RAS가 low로 떨어지면, CAS (Column Address Strobe) 신호가 high로 있는 동안 해당 주소의 행은 리프레시 되어진다.(그림 3.) 그것은 리프레시 할 주소를 제공하고 모든 행이 적절한 시간에 리프레시 되도록 하는 것이 DRAM 컨트롤러의 기능이다. 리프레시 동작 동안 행 순서는 문제가 되지 않고 셀에 저장된 데이터가 파괴되기 전에 리프레시되는 것이 중요하다.

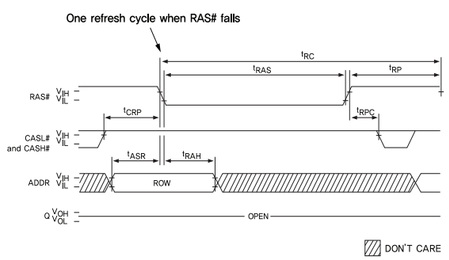


그림 3. RAS-only 리프레시 타이밍도

CAS Before RAS Refresh : 이것은 CBR 리프레시로 알려져 있는데, 사용이 용이하고 저전력에 유리하기 때문에 흔히 사용되는 방식이다. CBR 리프레시 주기는 RAS신호가 high에서 low로 바뀌기

전에 CAS신호를 low로 설정함으로서 실행된다. 하나의 리프레시 주기는 각각의 RAS 신호가 low로 떨어질 때 실행된다. WE(Write Enable)신호는 RAS신호가 떨어질 때 이 주기 동안 high를 유지해야한다. CBR 리프레시가 실행될 때 마다 행에 대한 카운터의 값이 증가한다. 다시 CBR 리프레시가 실행될 때, 카운터가 지시하는 다음 행은 카운터의 증가에 이어 리프레시 된다.

이 행의 수에 상응하는 허용되는 최대값에 도달하면 카운터가 자동으로 처음으로 돌아간다. 한번 초기화후 설정된 카운터는 재설정할 수 없다. 그림 4에 CBR 리프레시와 그림 5에 CBR 리프레시 주기를 나타내었다. CAS신호가 low상태를 유지할 동안 RAS신호만 토글(toggle)한다. RAS 신호가 매번 떨어질 때마다 리프레시를 수행한다. CBR 리프레시 정책은 주소가 버스에 실리지 않기 때문에 낮은 전력 소모로 더 유리하다.

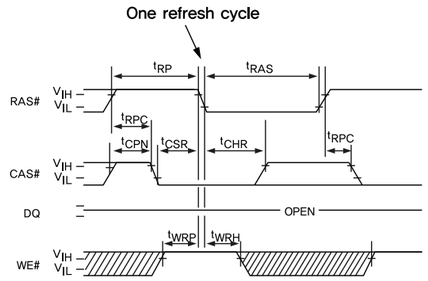


그림 4. CBR 리프레시 타이밍도

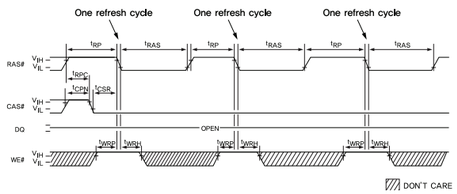


그림 5. 3개의 CBR 리프레시 사이클 타이밍도

CBR 리프레시는 외부 주소를 사용하지 않고 내부 카운터를 사용하기 때문에 주소 버퍼는 power-down된다. 이는 버스라인 상에서 주소를 싣기 위한 부가적인 동작이 없으므로, 전력에 민감한 어플리케이션의 경우 유리하다.

Hidden Refresh : 사용자가 읽기나 쓰기를 한후, CAS 신호를 low, RAS 신호를 tRP동안 high에서 low로 하면 Hidden 리프레시에 들어간다. 따라서 RAS가 low로 가기 전에 CAS가 low가 되면 부분적으로 CBR 리프레시를 실행한다. hidden 리프레시는 읽어낸 데이터가 DQ line상에 있을 동안 리프레시가 수행되는 것이다.

read와 hidden 리프레시 수행 시간은 동일하게 tRC가 걸린다. 그림 6에 read이후 hidden 리프레시동작에 대한 타이밍도를 나타내었다.

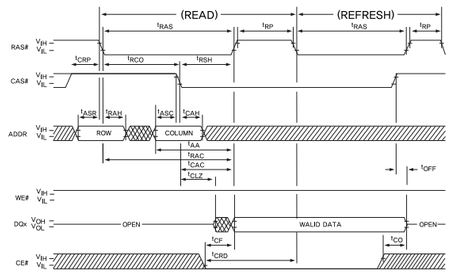


그림 6. read 이후 hidden 리프레시 타이밍도

3. DRAM Refresh 연구동향

3.1. Smart Refresh

Smart 리프레시 방법은 M. Ghosh와 Hsien-Hsin S.Lee가 제안하였다[17]. 일반 메모리 동작에서 데이터를 일거나 쓰기동작이 완료된 후 해당 행이 열리고 쓰기작업이 완료되어도 다른 행이나 뱅크가 접속되기 전까지 데이터가 센스앰프(sense amp.)상에 머문다. 이 센스앰프에 있는 데이터는 다시 원래의 셀에 쓰여지고 새로운 행은 프리차지(precharge)된다. 이렇게 엑세스(access)된 행은 다음 리프레시 간격이 될 때까지 리프레시가 필요 없다. 즉, 잠재적으로 최근에 엑세스 한 행의 리프레시를 지연 시킬 수 있다.

본 기술에서는 이에 착안하여 리프레시를 지연할 수 있는 방법을 제시하였다. 따라서 과도한 리프레시와 에너지 소모를 제거 할 수 있다. 시뮬레이션 결과로 모든 리프레시 동작을 86%와 2GB DRAM에서 평균 59.3%까지 감소시킬 수 있다. 이것은 리프레시 동작에서 52.6%의 에너지를 절감할 수 있다. 6MB 3D DRAM에서 64ms로 리프레시 할 경우 에너지 절감이 최대 21%와 평균 9.37%까지 이다. 32ms 리프레시에서는 최대 12%, 평균 6.8% 이다.

요구되는 리프레시 주기를 최소화하기 위해 메모리 제어기에서 각 행에 대해 time-out 카운터를 유지하여, 엑세스된 행을 기본값(리프레시 간격)을 가지게 하기 전에 다음에 따라오는 주기적 리프레시 동작은 수행되지 않도록 하였다. 각 time-out 카운터는 2-bit 혹은 3-bit 이진 카운터로 구성되며, DRAM의 리프레시 간격 내에서 최대값으로부터 0으로 균일하게 count down한다. 카운터 값이 0에 도달하면, 특정 행이 리프레시 되어야 한다는 것을 나타낸다. 메모리 내에서 해당 뱅크나 행이 엑세스되고 행이 열리면 카운터는 최대값으로 재 설정된다. 메모리 컨트롤러는 해당 카운터의 값이 0이 아니면 행을 리프레시 하지 않는다.

3.1.1 Staggered Countdown

Smart Refresh 메모리 제어기는 DRAM의 각 행에 대한 2bit time-out 카운터를 가진다고 가정한다. 그림 7에 카운터 배열을 나타내었고, 예에서 64ms 리프레시 주기를 가정하였다. 카운터 값은 DRAM의 각 행에 대한 메모리 제어기로 부터 업데이트 되어지는 것을 그림에서 보여준다. 2bit 카운터는 3에서 0까지 64ms내에 모든 행이 올바른 데이터를 유지하도록 적시에 리프레시를 보장하는 down-count로 설계되었다.

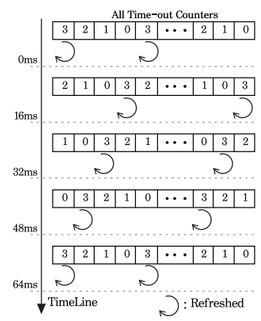


그림 7. Down-counting Time-out 카운터

그림에서 time-out 카운터의 초기화가 엇갈리게 되도록 하였다. 이 경우에는 모든 카운터의 1/4이 16ms에서 0으로 감소되는데, 다른 1/4은 32ms에서 0으로, 등과 같이 감소된다. 다수의 메모리 행이 하나씩 차례로 리프레시 되어져야 하는 burst 리프레시와 유사한 상황을 가진다. 그러나 카운터의 1/4이 0으로 초기화 되었지만, 초기에 비록 모든 행이 리프레시 되었기 때문에 이 staggering은 초기에 전력 오버헤드를 초래한다. 따라서 처음 64ms내에서 다시 한번 리프레시 된다. 일반 read 와 write 프로세서동안 행이 액세서될 때, 해당 카운터는 최대값으로 재설정된다. 잠재적인 같은 값을 가지는 많은 수의 카운터가 동시에 감소 될 것이기 때문에 동시에 0으로 count down할 것이므로, 이런 조건에서는 burst 리프레시로 이어질 수 있다. 이러한 문제는 단지 카운터를 감소하는 것을 초기화에 따라 지그재그(staggered)로 하여 해결할 수 있다.

그림 8에 본 방식의 해법을 나타내었다. 본 scheme에서는 카운터가 균일하게 논리 세그먼트 N(본 논문에서는 N=4)으로 해시 된다. N값의 선정은 리프레시 queue의 크기에 기초한다. 기존의 방식과의 주요 차이는 모든 카운터가 메모리 컨트롤러에 의해 동시에 엑세스 되지 않는다는 것이다. 제안한 staggered scheme은 리프레시 하거나 메모리 컨트롤러에 의한 카운터 감소는 주어진 시간에 4개의 색인된 카운터만 사용할 수 있다.

해시함수의 결과로 N개의 카운터만 동시에 활성화된다. 이 scheme의 목표는 카운터의 크기로부터 리프레시 간격(64ms)을 나누어 정의한 소위 카운터 엑세스 주기내에 정확히 한번 각 카운터를 색인하는 것 이다. 그림 8에서 카운터 엑세스 주기는 16ms이다. 색인은 다음 카운터로 진행하기 위해 각 세그먼트내(segement)에서 time-out카운터의 수로, 한 클럭 주기에 카운터 엑세스 주기를 나눈다. 리프레시 주기는 16ms이고, 각 세그먼트에 16개의 메모리 행이 있다면, 카운터 인덱스는 매 1ms 마다 한번씩 진행한다. 카운터 색인값이 0이되면, 해당 행에 리프레시가 요구된 후로 카운터가 최대값으로 재설정함으로, 다음번에 새로이 색인되어 진다.

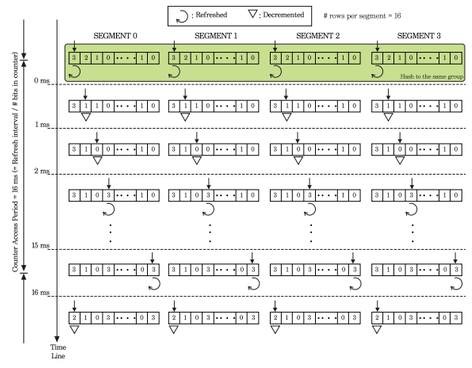
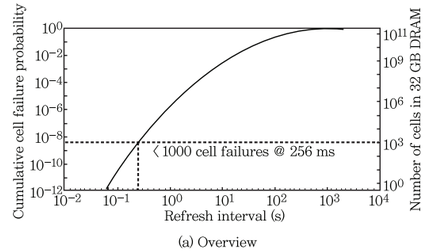


그림 8. 로직 세그먼트로 나누어진 카운터다운 카운터 및 카운터다운 방식 예

3.2. RAIDR : Retention-Aware Intelligent DRAM Refresh

RAIDR 방식은 Jamie Liu 등에 의해 제안되었다[18]. 표준 DRAM에서 64ms의 리프레시 최소 보수적 간격을 요구하는 소수의 취약한 DRAM 셀들을 관찰해보면, 32GB DRAM에서 1000개 이하의 셀이 최소 리프레시 간격의 4배인 리프레시 간격 256ms를 요구한다(예, 그림 9). 따라서 대부분의 DRAM 셀에 대해서는 낮은 리프레시율을 적용하고 일부 취약한 셀들에 대해 선택적으로 높은 리프레시율을 적용하므로써 리프레시 오버헤드를 크게 줄일 수 있다. 이렇게 하기위해서 RAIDR을 제안하였다.



(a) Overview

그림 9. 60nm process에서 DRAM 셀 지연시간

높은 리프레시율이 요구되는 row를 더 자주 리프레시해주기 위해 각 bin의 row를 서로 다른 비율로 리프레시한다. RAIDR은 DRAM을 수정하지 않기위해 메모리 제어기에 retention time bin을 저장하는데 Bloom 필터를 사용한다[19]. 단지 두 개의 retention time bin으로 1.25KB 메모리 제어기 오버헤드를 가지는 32GB DRAM 시스템에서 성능을 8.6% 개선할 수 있고, 이러한 시스템에서

DRAM 시스템 전력을 16.1% 감소시킬 수 있다. 그림 10에 RAIDR의 개념도를 나타내었다.

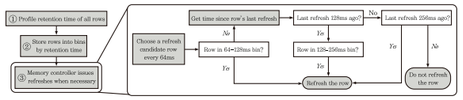


그림 10. RAIDR 동작 개념도

행의 모든 셀에 대해 최소유지 시간으로서 행 유지시간(row's retention time)을 정의한다. bin의 세트는 메모리 컨트롤러에 추가되어 각 세트는 유지시간의 범위와 연관되도록 하였다. 각 bin은 그 bin의 범위에 속하는 유지시간을 가진 모든 행을 포함한다. 주어진 bin으로부터 가장 짧은 유지시간을 커버하는 것이 bin의 리프레시 간격이다. bin으로 부터 커버되지 않는 가장 짧은 유지 시간은 새로운 new default refresh interval로 설정한다. 그림 10에서 2bin의 예를 나타내고 있다.

하나의 bin은 64 와 128ms 사이의 유지시간으로 모든 행을 포함하는 bin 리프레시 간격은 64ms이다. 나머지 bin은 128과 256ms 사이의 유지시간으로 bin 리프레시 간격은 128ms이다. new default refresh interval 256ms로 설정된다.

각 행의 지연시간을 결정하는 정보수집 단계로 그림 11의 ①에 해당한다. 각 행에 대해, 행의 지연시간이 new default interval보다 작다면, 메모리 제어기는 적절한 bin ②에 삽입한다. 시스템 동작 ③ 동안 메모리 제어기는 각 행이 리프레시 후보마다 64ms로 선정될 수 있도록 한다. 하나의 행이 리프레시 후보로 선정되면 메모리 제어기는 행의 지연시간을 결정하기 위해 각 bin을 점검한다.

3.2.1. Retention Time Profiling

행지연시간의 측정하기 위해, 행의 각 셀의 지연시간 측정이 필요하다. 이러한 측정을 수행할 간단한 방법은 static 패턴("all 1s" or "all 0s")을 기록한 후 리프레시를 해제하고 첫 번째 비트 변화를 관찰하는 것이다.

시스템의 행 보존 시간을 수집하기 전에, 메모리 제어기는 기존 auto-refresh를 사용하여 리프레시를 한다. 시스템의 행 보존 시간이 측정 된 후, 결과는 운영 시스템에 의해 파일에 저장할 수 있다. DRAM 셀의 유지시간은 life time동안 크게 변화되지 않기 때문에, 향후의 부트 업(Boot up) 동안, 결과는 더 이상 프로파일링을 요구하지 않고 메모리 제어기에서 복원 할 수 있다.

3.2.2. Storing Retention Time Bins : Bloom Filters

메모리 제어기는 각 bin에서 행의 세트를 저장해야하는데, 각 빈에 있는 행의 정확한 숫자는 시스템에서 DRAM의 용량뿐만 아니라 DRAM 칩 사이의 유지 시간 변화에 따라 달라질 수 있다. 각 bin의 행을 저장하는 테이블 용량이 불충분 한 경우, 이 구현은 정확성을 제공할 수 없다. 이러한 어려움을 극복하기 위해, 본 논문에서는 유지 시간 bin을 구현하는 Bloom 필터를 사용하였다.

Bloom 필터는 비트 어레이 길이를 m, 해시 함수 k로 구성된다. 그림 11은 m=16, k=3인 예이다. 초기에 모든 비트는 0으로 초기화 되어있다. 인자를 삽입하기 위해 해시값 k로부터 해시되고 모든 비트는 해당 위치에 모두 1값으로 설정된다(그림에서 ①). 해당비트 위치에 있는 모든 비트가 1인 경우, 그 요소는 세트 ②에서 *present*로 선언된다.

해당 비트 중 하나라도 0인 경우, 그 요소는 세트 ③에서 *not present*로 선언한다. 많은 서로 다른 요소들이 동일 비트로 매핑될 것이므로, 다른 요소 ④를 삽입하여 *false positive*를 끌어낸다.

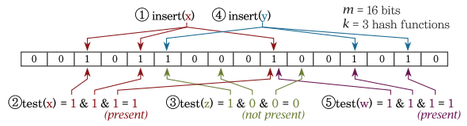


그림 11. RAIDR 상세 구현

본 메커니즘은 행이 필요 이상으로 빈번히 리프레시 될 수 있지만, 필요량보다 작게 되지는 않는다는 맥락에서 데이터의 무결성이 보장된다. Bloom 필터 매개변수 m과 k는 예상 용량과 원하는 *false positive* 가능성에 기초하여 최적화 할 수 있다.
앞에서 두 가지 연구 방향에 대해 살펴보았다. 그이 외에도 여러 가지 연구 방향들이 있지만 지면상 언급을 생략하였다.

결론

이상에서 DRAM 리프레시와 연구 동향에 대해 살펴보았다. 컴퓨터 시스템에서 DRAM이 차지하는 비중이 증가 할수록 리프레시 동작에 의한 시스템 성능문제와 전력 소모 측면에서 더욱 중요한 문제로 대두 될 것이며 산업체에서 이에 대한 개선책을 찾기 위해 많은 연구가 필요할 것으로 생각된다.

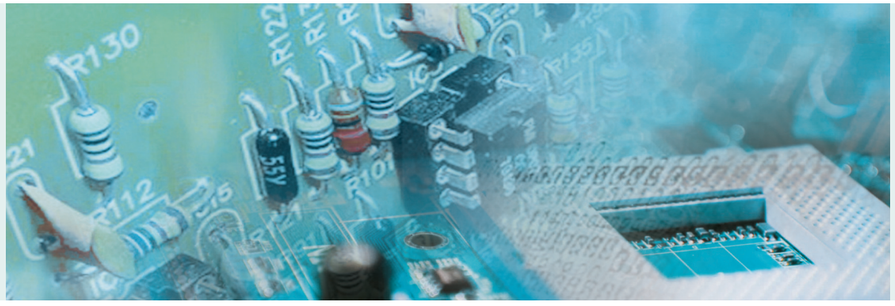
Reference

1. Samgsung Develops 3D Memory Package that Greatly Improves Performance Using Less Space. http://www.samsung.com/PressCenter/PressRelease/PressRelease.asp?seq=20060413_0000246668.
2. L. A. Polka, H. Kalyanam, G. Hu, and S. Krishnamoorthy. Package Technology to Address the Memory Bandwidth Challenge for Tera-Scale Computing.
3. B. Black, M. Annavaram, N. Brekelbaum, J. DeVale, L. Jiang, G. H. Loh, D. McCaule, P. Morrow, D. W. Nelson, D. Pantuso, P. Reed, J. Rupley, S. Shankar, J. Shen, and C. Webb. Die stacking (3d) microarchitecture. In Proceedings of the 39th International Symposium on Microarchitecture, pages 469.479, 2006.
4. Intel Technology Journal, 11(03), 2007.M.Viredaz and D. Wallach. Power Evaluation of a Handheld Computer: A Case Study.
5. Technical report, Compaq WRL, 2001. Micron. DDR2 SDRAM SODIMM 1GB 2GB Data Sheet. http://download.micron.com/pdf/datasheets/modules/ddr2/HTF16C64 128 256x64HG.pdf.
6. J. Kim and M. C. Papaefthymiou, "Dynamic memory design for low data-retention power," in PATMOS-10, 2000.
7. J. Kim and M. C. Papaefthymiou, "Block-based multiperiod dynamic memory design for low data-retention power," IEEE Transactions on VLSI Systems, 2003.
8. T. Ohsawa, K. Kai, and K. Murakami, "Optimizing the DRAM refresh count for merged DRAM/logic LSIs," in ISLPED, 1998.
9. K. Yanagisawa, "Semiconductor memory," U.S. patent number 4736344, 1988.
10. P. G. Emma, W. R. Reohr, and M. Meterelliyoz, "Rethinking refresh: Increasing availability and reducing power in DRAM for cache applications," IEEE Micro, 2008.
11. Y. Katayama et al., ""Fault-tolerant refresh power reduction of DRAMs for quasi-nonvolatile data retention," "in DFT-14, 1999.
12. C. Wilkerson et al., ""Reducing cache power with low-cost, multi-bit error-correcting codes," "in ISCA-37, 2010.
13. B. L. Jacob, ""DRAM Refresh is Becoming Expensive in Both Power and Time," "in The Memory System: You Can''t Avoid It, You Can''t Ignore It, You Can''t Fake It, ser. Synthesis Lectures on Computer Architecture. Morgan & Claypool Publishers, 2009.
14. Influent Corp., ""Reducing server power consumption by 20% with pulsed air cooling," "Jun. 2009, http://www.influentmotion.com/ServerWhite Paper.pdf.
15. Micron. Various Methods of DRAM Refresh. http://download.micron.com/pdf/technotes/DT30.pdf
16. Jeffrey Stuechelixz, et al., "Elastic Refresh: Techniques to Mitigate Refresh Penalties in High Density" MICRO '43 Proceedings of the 2010 43rd Annual IEEE/ACM International Symposium on Microarchitecture, Pages 375-384. 2010.
17. Mrinmoy Ghosh, Hsein-Hsin S. Lee "Smart Refresh: An Enhanced Memory Controller Design for Reducing Energy in Conventional and 3D DieStacked DRAMs" Proceedings of the 40th Annual IEEE/ACM International Symposium on Micro-architecture, page 134-145, 2007.
18. Jamie Liu, Ben Jaiyen, Richard Veras, Onur Mutlu "RAIDR: Retention-Aware Intelligent DRAM Refresh" International Symposium on Computer Architecture, vol. 40, no. 3, page 1-12, Sep. 2012.
19. J. L. Carter and M. N. Wegman, ""Universal classes of hash functions," "in STOC-9, 1977.

시스템 및 메모리 반도체 칩에 적합한 sub-20-nm 반도체 소자 구조

서울시립대학교 전자전기컴퓨터공학부

신창환 교수
연구분야 : Device and Circuit
E-mail : cshin@uos.ac.kr
https://sites.google.com/site/edlatuos/



서론

대용량 고성능 서버급 컴퓨터부터 개인용 컴퓨터 (PC) 및 스마트폰 (Smart-phone) 등에 이르기까지, 일상생활에 널리 쓰이는 디지털 기기들의 동작을 책임지고 있는 시스템 및 메모리 반도체 집적회로 칩은, 지난 50년간의 지속적인 반도체 소자/공정 기술발전에 힘입어, 최근에는 단위 칩당 약 10^9개 이상의 반도체 소자들로 구성되어 있는 수준에 이르렀다. 이러한 시스템 및 메모리 반도체 칩 내에서 가장 핵심적인 역할을 수행하는 전자 소자는 "트랜지스터"라 할 수 있다. 여러 종류의 트랜지스터를 가운데, 전계(electric field)를 이용한 Metal-Oxide-Semiconductor Field-Effect Transistor(MOSFET)가 오늘날 가장 널리 사용되고 있는 대표적인 트랜지스터이다. 물론, 몇몇 응용 분야들, 예를 들어, 전력용, 테라헤르츠(mm-wave)용의 경우에는, 해당 분야에 알맞게 특화된 트랜지스터인 IGBT, HBT 등이 각각 개발 및 사용 중이다. 본 특집기사에서는 차세대 20-nm이하급 반도체 공정 기술을 사용하여 제작될 시스템 및 메모리 반도체 칩에서 활용될 수 있는 다양한 종류의 차세대 실리콘 MOSFET 소자들의 구조, 디자인, 특성 등을 비교분석한다.

본론

현재 MOSFET이 가지고 있는 기술적 문제점

그림 1은 가장 널리 사용되고 있는 planar bulk MOSFET 구조를 보여준다: 소스(Source), 드레인(Drain), 그 둘을 이어주는 채널(Channel), 그리고 그 채널을 위에서 제어하는 게이트(Gate), 총 4개의 부분으로 구성된다. 게이트가 ON상태로 바뀌면, 전자(electron) 혹은 정공(hole)이 소스에서 드레인으로 움직일 수 있는 전도성 채널이 게이트 아래에 생성된다. 게이트가 다시 OFF상태로 돌아오면, 그 채널은 사라지게 되어 있다. 하지만, 반도체 집적회로의 집적도(단위면적당 트랜지스터의 개수)를 높이기 위해 소스와 드레인의 간격을 줄여감에 따라, 게이트의 전체 채널 영역에 대한 제어능력은 점점 약해져 오고 있다. 특히 게이트에서 가장 멀리 떨어진 채널의 일부영역을 통해 (다시 말해 게이트가 ON상태에서 OFF상태로 돌아왔으나 여전히 존재하는 conductive path을 통해), OFF상태에서도 미세 leakage 전류가 흐르게 된다. 그 결과, 그림 2에서 보듯, 20-nm급의 경우, 전자 소자가 동작하지 않는 경우 반도체 칩이 소모하는 전력량(static or sub-threshold power consumption)이, 전자 소자가 동작하는 동안 반도체 칩이 소모하는 전력량(active power consumption)과 거의 비슷한 수준에 이르게 되었다. 다시 말해, 단위 면적당 소모전력량(power density)이 반도체 칩의 집적도가 증가함에따라 꾸준히 상승하고 있는 셈이다. 미세 leakage 전류를 제어할 수 있는 방안으로 불필요한 silicon영역을 제거하거나 (extremely-thin-body MOSFET 소자 구조가 이에 해당함) 혹은 채널영역에 대한 게이트 제어능력을 배가시키기 위해 두 개 이상의 게이트를 가지는 전자 소자 구조 (multi-gate MOSFET이 이에 해당함)를 생각해볼 수 있다.

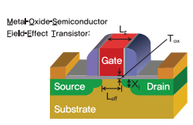


그림 1. Planar Bulk MOSFET의 3차원적인 모습

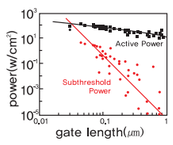


그림 2. 전력소모량 vs. 채널길이 [1]

꾸준히 증가되는 power density 문제 이외에, 반도체 소자의 크기가 30-nm 이하로 작아짐에 따라 급격히 증가하는 "고유한 문턱전압*의 변화(intrinsic threshold voltage variation) 문제"가 오늘날 널리 사용중인 planar bulk MOSFET이 극복해야할 가장 큰 기술적 난관이다. 그림 3에서 보듯이, sub-20-nm 반도체 소자의 경우 다양한 random variation sources(Line Edge Roughness(LER), Random Dopant Fluctuation(RDF), Poly/Metal-Grain Granularity(MGG))에 의해 문턱전압 변화량의 표준편차 값이 약 400mV(power supply voltage, V_{DD} = 1.0V 일 때)에 이르는 것을 볼 수 있다. 사실, MOSFET의 채널 길이를 줄여감에 따라, 채널에 대한 게이트의 제어정도(gate-to-channel capacitive coupling)를 높이기 위해, 다시 말해, short-channel-effects(V_T roll-off 및 Drain-Induced Barrier Lowering(DIBL))을 줄이기 위해, MOSFET 채널 영역 내의 불순물 농도를 꾸준히 높여왔다. 그 결과, 30-nm 이하의 물리적 채널 길이를 가지는 MOSFET의 경우 (보통, 채널영역의 불순물 농도 〉 $10^{18}cm^{-3}$ 필요), 그 소자가 가지는 불순물의 총 개수와 위치를 다른 소자들과 비교해보면, 그 총 개수와 위치가 소자별로 매우 달라(이를 Random Dopant Fluctuation(RDF)이라 함), 반도체 집적회로에 사용되는 전압값(power supply voltage, V_{DD})에 대한 집적회로 내 모든 소자들의 문턱전압 표준편차값이 상당한 수준 (5~40%)에 이르게 된다(그림3). 그 결과, planar bulk MOSFET구조를 이용한 10나노급 혹은 그 이하의 단일 반도체 소자는 제작 가능할 수는 있겠으나(retrograde channel doping profile활용, Suvolta Inc. CA, USA), 충분한 수율(〉 95%)이 보장된 집적회로 칩들을 양산하기는 매우 힘들어질 것이다. 그러므로, MOSFET의 크기를 줄여감에 따라 채널 농도를 꾸준히 증가시켜왔으나, RDF문제로 인해 더 이상 채널의 농도를 높일 수 없는 수준에 이르렀다고 판단되어, 최근에는 채널 농도를 높이지 않으면서 gate-to-channel coupling을 높일 수 있는 새로운 형태의 반도체 소자 구조들이 제시되고 있다. 몇 가지 대표적인 소자 구조에 대해 다음 섹션에서 조금 더 자세히 알아보도록 한다.

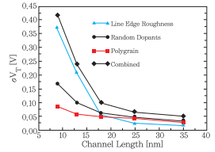


그림 3. 문턱전압(V_T)의 표준편차 값 vs. 채널길이 [2]

차세대 반도체 소자: FinFET vs. Fully-Depleted Silicon-On-Insulator (FDSOI) MOSFET

10억개 이상의 트랜지스터로 구성된, 현대 시스템/메모리 반도체 칩 설계철학은, "better, sooner, cheaper products"개발로 대변된다. 그러나, 20-nm이하의 feature sizes를 가진 반도체 소자는, 앞서 설명했던 short-channel-effects 및 varying dopant levels(즉, RDF)때문에 높은 수준의 leakage current(→higher power density) 및 intrinsic threshold voltage variation 문제들에 직면해 있다. 이로 인해, 기존의 반도체 소자 구조에 큰 변화 없이 더 작은 geometries 구현에 집중되어 있는 현재의 반도체 공정/소자 분야의 연구개발에, 새로운 반도체 소자의 구조를 고려한 연구개발이 필요한 시기가 왔다.

지금으로부터 약 20여년전인 1990년대, 미국의 Defense Advanced Research Projects Agency (DARPA)는 planar transistor를 대체할 새로운 반도체 소자 구조에 관한 연구를 지원했었다 [3]. Chenming Hu, Jeffrey Bokor, Tsu-Jae King Liu교수가 이끄는 UC Berkeley의 Device group은 thin-body MOSFET 구조를 활용함으로써 planar transistor의 불필요한 silicon영역을 제거함으로써 leakage current를 감소시킬 수 있으며, 향상된 gate-to-channel capacitive coupling을 통해 short-channel-effects를 제어할 수 있다고 제안하였다(그림 4). Thin-body MOSFET 구조를 가진 대표적인 두 가지 형태의 반도체 소자 (즉, FinFET, FDSOI MOSFET)에 대해 살펴보자.

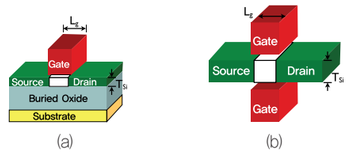


그림 4. Thin-body MOSFET structures: (a) ultra-thin-body planar structure, (b) multi-gate (double-gate) three-dimensional structure.

FinFET

기존의 planar MOSFET 공정에 사용되고 있는 standard photolithography 기술을 활용해야하는 조건 하에, 그림 4(b)의 multi-gate thin-body structure를 wafer표면 상에 구현하기 위해서는, 그 double-gate구조를 90도 회전시켜야 한다. 그러면, 그림5에 보여진 FinFET구조를 쉽게 도출해 낼 수 있다. 현재 Intel에서 22-nm 반도체 공정기술부터 새롭게 도입한 FinFET은 planar substrate위로 돌출된 3차원 구조를 가짐으로써, 기존의 planar MOSFET과 비교시, 같은 layout area에 대해 더 큰 채널 부피를 가지게 된다. 또한, 채널주위를 게이트가 감싸고 있는 구조 덕분에 채널에 대한 게이트 제어정도는 planar MOSFET보다 월등히 높다. 그 결과, FinFET이 OFF 상태일 때, body를 통해 새어나가는 전류량이 상대적으로 매우 적고, ON상태일 때는 multi-gate structure덕분에 body를 통해 흐르는 최대 전류량이 상대적으로 더 커지게 된다. 그로 인해 planar MOSFET보다 FinFET은 상대적으로 더 낮은 문턱전압(V_T)을 사용할 수 있게 되어서, FinFET을 사용한 반도체 칩의 switching speed는 더 빨라지고, power consumption은 더 줄일 수 있게 된다. 최근 연구 동향을 살펴보면, 기존 planar MOSFET대비 FinFET사용시, static leakage current는 약 90%까지, dynamic power consumption은 대략 절반 가까이로 낮추면서, 동시에 성능은 최대 약 37%까지 향상시킬 수 있다고 보고된다. 한편, planar MOSFET소자의 경우, short-channel-effect를 효과적으로 줄이기 위해 지속적으로 높여온 채널의 농도때문에 생긴 intrinsic threshold variation을 극복해야 한다고 앞서 설명하였다. FinFET은 기존의 planar MOSFET와 비교시, 약 1000배 이하까지 낮은 수준의 채널농도만을 필요로 한다. 그리고, 낮은 채널농도로 인해 유발되는 short-channel-effect는 소자구조적으로 매우 얇은 채널 두께 및 그 채널을 두 개이상의 게이트가 제어하게 됨으로써 쉽게 극복가능하다. 결국, short-channel-effect를 thin-body 및 double-gate구조를 활용하여 극복하고, 채널의 농도는 매우 낮추어서 intrinsic random variation에 강한 반도체 소자를 구현한 셈이다. 마지막으로, scalability관점에서 FinFET이 가지는 장점은, 채널의 두께(혹은 body두께)를 scaling할 수 있는 한, FinFET은 지속적으로 scaling이 가능한 반도체 소자 구조이다. 현재까지 sub-5-nm FinFET이 성공적으로 제작되어 보고되었다 [4].

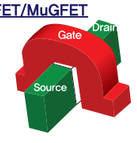
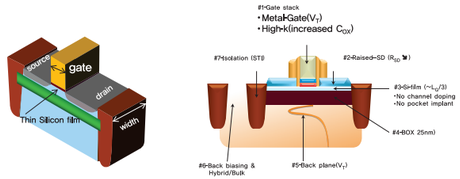


그림 5. FinFET 구조: 그림4(b)의 multi-gate FET(MuGFET)를 90도 회전시켜 wafer위에 구현함.

FinFET의 구조적 특성으로 인해 생기는 가장 큰 technical challenge는 3차원 형태를 띤 fin의 discrete size에 있다. 다시 말해, planar MOSFET의 size를 결정하는 가장 중요한 변수인 채널폭(width)이, FinFET의 경우에는 더 이상 연속적인 값이 아니라는 것이다. 이러한 discrete fin sizing은 회로설계자가 고려해야할 또 하나의 변수가 될 것이다. 또한, 이로 인해, device model의 복잡도(complexity)는 매우 높아지게 된다. 즉, planar MOSFET과 비교해서, 3차원 형태를 띤 FinFET은 더욱 많은 저항/캐패시터 성분을 가지고 있기 때문에, 디바이스 동작을 설명하기 위해 필요한 모델 파라미터가 더 많아지고, model calibration을 위해 더 많은 data manipulation이 필요할 것이다. 이러한 complexity는 RC extraction, layout, design rule checks (DRCs), layout versus schematic (LVS) 및 전반적인 back-end flow까지 영향을 주게 된다. 그러므로, FinFET개발은 단순히 Foundry회사의 반도체 소자 및 공정 개발에 머물러서는 안되며, EDA tool 업체까지 아우르는 전체 반도체 산업의 생태계가 FinFET에 맞게 형성되어야 한다.

Fully-Depleted Silicon-On-Insulator (FDSOI) MOSFET

FinFET과 동일하게 thin-body(얇은 채널영역)구조를 가지되, 삼차원 구조가 아닌 기존의 planar MOSFET과 동일한 평면 구조를 가지는 반도체 소자가 Fully-Depleted Silicon-On-Insulator (FDSOI) MOSFET이다. 그림 6(a)에서 보듯이, FDSOI MOSFET은 그 채널이 게이트에 의해 완전히 공핍될 만큼 충분히 얇으며, SOI substrate를 반드시 필요로 하는 소자 구조를 가진다. FinFET과 마찬가지로, 채널의 농도가 planar MOSFET과 비교시 약 1000배 이하까지 낮은 수준이므로, RDF에 의한 intrinsic random variation문제가 거의 없다. 또한, 낮은 채널 농도로 인해 유발되는 short-channel-effect문제는 채널 영역이 게이트에 의해 완전히 제어될 정도로 얇기 때문에 (20-nm급 소자의 경우, thin-body가 6 ~ 8nm 정도인 경우) 쉽게 극복된다.



(a) (b)

그림6. FDSOI MOSFET: (a) 조망도, (b) 단면도.

그림 6(b)는 FDSOI MOSFET의 단면도를 보인다. 각 부분별 (#1 ~ #7)로 하나씩 살펴보자. 첫번째, FDSOI MOSFET의 gate-stack는 현재 널리 사용중인 high-k/metal-gate (HK/MG)기술을 사용함으로써, 유효 gate oxide capacitance값을 극대화시킬 수 있는 구조이다. 기존 Planar MOSFET을 위해 개발한 HK/MG기술을 그대로 적용할 수 있는 구조인 셈이다. 또한, work-function engineering을 통해 FDSOI MOSFET의 문턱전압을 바꿀 수도 있다. 두번째, FDSOI MOSFET은 buried oxide(BOX)위의 얇은 실리콘 박막을 사용하기 때문에, source/drain의 series 저항값(R_{SD})이 planar MOSFET보다 크다. 그래서, source/drain 영역의 박막 두께를 두껍게 만들어 줌으로써, R_{SD}을 최소화 시켜준다. 또한, 최근에는 vertically-raised source/drain구조가 아닌, faceted-shaped raised source/drain을 사용하여, R_{SD}도 줄이면서 gate-to-source/drain의 capacitance값도 최소화시켜 switching speed를 극대화 시켜준다. 세번째로, BOX위의 실리콘 박막의 두께는 디바이스의 채널 길이에 따라 결정되어야 한다. 보통, 채널 길이의 약 ¼수준의 body(channel)두께를 가져야, short-channel-effect가 효과적으로 극복된다. 그 보다 더 두꺼운 채널을 가지게 되는 경우, FDSOI MOSFET의 short-channel-effect가 커지는 단점이 생긴다.

이처럼 얇은 박막은 SOI substrate공급업체의 기술에 거의 전적으로 의존하게 되는데, 최근 Soitec은 단일 wafer의 박막두께 표준편차값이 1.6Å(=0.16nm)인 wafer를 대량 생산가능하다고 발표하였다. 넷째, BOX 두께에 관해서는, 현재 SOI substrate 공급업체에서 제공하는 제품군은 크게, 145/25/10-nm로 나누어진다. 더 얇은 BOX을 사용함 (예를 들어, 145nm vs. 10nm)으로써, BOX아래쪽의 back-plane의 불순물의 종류에 따라, FDSOI MOSFET이 가지는 문턱전압 값을 쉽게 다양화 시킬 수 있다. 즉, n-type FDSOI MOSFET의 back-plane을 n-type, no-doping, p-type으로 구현하게 되면, 그 MOSFET은 각각 low-V_T, regular-V_T, high-V_T값을 가지게 된다. 고로, FDSOI 기술을 사용하여 반도체 칩을 설계하는 회로설계자에게 multiple V_T devices를 제공하기 위해서는 ground-plane의 도핑을 적절히 조절해 주어야 한다. 마지막으로, 기존의 planar MOSFET를 반도체 회로에 그대로 사용해야 하는 경우, BOX 영역을 식각공정을 사용하여 제거한 후, FDSOI MOSFET과 함께 집적가능하다 (Hybrid technology).

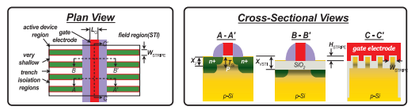
지속적으로 FDSOI MOSFET의 채널 길이를 scaling시키기 위해서는 body의 두께도 지속적으로 scaling시켜야 한다. 예를 들어, 물리적인 채널 길이가 15nm인 경우, body의 두께는 약 5nm이하 정도로 얇아야 short-channel-effect를 효과적으로 극복할 수 있다. 하지만, body의 두께가 5nm이하로 얇아지는 경우, quantum confinement 효과에 의해서, body두께 변화에 대한 V_T의 sensitivity가 매우 증가하게 되므로, 박막두께 표준편차값은 지금의 기술수준보다 훨씬 더 향상되어져야 할 필요가 있으며, 이는 추후 SOI substrate의 단가를 높이게 되는 요인으로 작용할 것이다. 하지만, FDSOI MOSFET을 위한 공정 단계수를 줄여줌으로써, planar MOSFET의 공정 단가와 비슷한 수준으로도 제공가능할 수 있을 것이다. 또 한편으로, 현재 널리 사용되고 있는 stress-engineering 기술을 FDSOI에 적용하기 위한 기술개발도 필요하다. 현재까지는, IBM에서 개발한 faceted-shaped source/drain을 사용하여 stress 효과를 극대화하면, 약 10%정도 성능향상을 기대할 수 있는 보고도 있다 [5].

Evolutionary pathway for planar MOSFET: Segmented-channel MOSFET structure (SegFET)

FDSOI MOSFET은 매우 얇으면서 균일한 두께의 도핑이 되지 않은 채널을 가짐으로써(20-nm급 소자의 경우 채널두께는 약 6 ~ 8 nm), variability 및 short-channel-effect을 줄일 수 있는 반도체 소자 구조이다. 하지만, 상대적으로 비싼 SOI substrate를 필요로 한다. (물론, planar bulk MOSFET과 비교하여, FDSOI MOSFET의 공정 단계를 줄임으로써, 최종 양산 단계의 FDSOI MOSFET의 단가를 현재의 planar bulk MOSFET과 비슷한 수준으로 맞출 수 있다는 보고가 있다.[6]) FinFET은 short-channel-effect를 줄이면서 layout area efficiency를 높이기 위해 물고기 지느러미 (Fin) 모양의 좁고 높은 채널(narrow and tall channel)을 활용하기 때문에, high-aspect-ratio의 구조를 필요로 한다. 이를 구현하기 위해 요구되는 복잡한 공정 기술을 사용하지 않으면서, 동시에 SOI substrate가 아닌 bulk substrate에 제작가능한 low-aspect-ratio Segmented-channel MOSFET(SegFET)을 이번 섹션에서 소개한다 [7, 8].

그림 7(a)에서 보듯이, SegFET의 채널 영역은 동일한 채널폭을 가진 평행한 여러 개의 stripes으로 구성된다 (초록색으로 표시된 active device region임). 그리고, 각 stripe은 very-shallow-trench-isolation(VSTI)에 의해 서로 전기적으로 절연된다. 이를 위해 corrugated substrate를 우선 제작해야한다 [9] (그림 8). VSTI의 깊이는 source/drain의 junction depth보다 깊게 설계되나, 트랜지스터들간의 전기적 절연을 위해 제작된 shallow-trench-isolation(STI)보다는 훨씬 얕은 수준으로 설계된다 (그림 7(b)의 C-C' 단면도 참고). Gate-stack을 제작하기 직전에 식각공정을 이용하여 VSTI isolation 영역을 recessing시킴으로써 (FinFET와 달리, 채널폭이 채널높이에 비해 더 큰 low-aspect-ratio를 유지시킴), SegFET내의 모든 stripes의 윗 부분(채널영역)이 게이트에 의해 감싸지도록 만들 수도 있다. 이러한 tri-gate structure을 이용한 SegFET는 planar SegFET에 비해, short-channel-effect를 더욱 줄이고, on-current는 더 크게 향상시킬 수 있다.

그림 9에서 보듯이, SegFET의 공정 순서는 기존의 planar MOSFET의 공정기술을 그대로 활용하면 되나, SegFET을 위한 wafer는 그림8에 보인 corrugated substrate (VSTI영역을 가진 bulk substrate)를 반드시 사용하여야 한다. 마지막으로, Planar bulk MOSFET, Bulk FinFET, SOI FinFET, Tri-Gate SOI MOSFET들과 비교하였을 때, SegFET이 가지는 장점을 표 1에 간단히 정리하였다.



(a) (b)

그림 7. SegFET 구조: (a) 평면도 및 (b) 단면도.

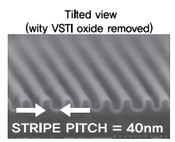


그림 8. Corrugated substrate의 SEM 이미지.

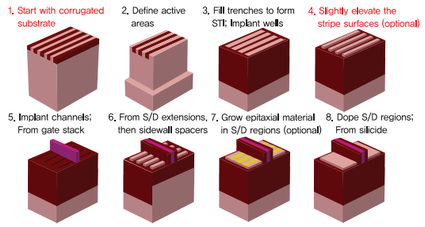


그림 9. SegFET의 간단한 공정 순서도.

vs. Planar Bulk MOSFET	▪ Improved electrostatic integrity & scalability ▪ Better immunity to variations due to - the narrow width effect and SCE - STI-induced mechanical stress - random intrinsic variation (LER/RDF/WFV)	*vs.* Bulk FinFET	▪ Valid to use Standard Compact Model ▪ Much-lower aspect-ratio of stripes, resulting in (i) ease of manufacture (ii) lower bulk punch-through
vs. Tri-Gate SOI MOSFET	▪ No need for SOI substrate (lower cost) ▪ Better layout area efficiency, due to low aspect-ratio (stripe height and width) ▪ Possibility of dynamic V_T adjustment ▪ Easier to adopt advanced channel materials ▪ Valid to use Standard Compact Model	*vs.* SOI-based FinFET	▪ Ease for manufacture, due to lower aspect ratio of stripes for SCE control ▪ Suitable for incorporating advanced channel materials (GiGe) for performance improvement ▪ Valid to use Standard Compact Model ▪ Easy to use body-biasing for dynamic V_T adjustment ▪ No need for SOI substrate (lower cost)

표 1. Planar bulk MOSFET, Tri-Gate SOI MOSFET, Bulk FinFET, SOI FinFET 대비, SegFET의 장점.

맺음말

본 기사에서는 기존의 planar bulk MOSFET이 가지는 기술적 한계에 대해 간단히 알아본 뒤, 차세대 반도체 소자 구조들, 특히 현재 반도체 종합회사 및 파운드리 회사에서 연구개발에 주력하고 있는 FinFET와 FDSOI MOSFET에 대해 살펴보았다. 마지막으로, 지속적인 planar bulk MOSFET의 scaling을 위한 하나의 대안으로 Segmented-channel MOSFET (SegFET)을 간단히 리뷰하였다. 2012년부터 미국의 인텔사에서는 본격적으로 FinFET을 이용한 제품을 양산 중이며, 프랑스의 ST Microelectronics사는 FDSOI MOSFET 공정 기술을 활용하여 에릭슨사와 함께 스마트폰에 사용되는 반도체 칩을 제작양산 중이다. 그리고, 아시아 시장에서는 한국의 삼성 Foundry, 대만의 TSMC 및 UMC Foundry업체는 sub-20-nm 기술을 위한 FinFET 소자 개발에 현재 주력하면서, FDSOI의 기술개발 동향도 예의주시하고 있다. 앞으로 향후 10년 이상 지속적으로 전개될, 두 라이벌 반도체 소자구조 (FinFET vs. FDSOI MOSFET)의 전쟁 결과가 어떻게 될지 귀추가 주목된다.

* 문턱전압 (Threshold voltage) : transistor를 ON시키는데에 필요한 최소의 게이트 전압값. 보다 정확한 문턱전압의 정의는 반도체 소자 및 소자물리와 연관된 저서들을 통해 확인가능함.

Reference

[1] T. C. Chen, "Where CMOS is going: trendy hype vs. real technology," in IEEE International Solid-State Circuits Conference (ISSCC), February 2006.
[2] A. Asenov, "Simulation of statistical variability in nano MOSFETs," in Symp. VLSI Technology, June 2007.
[3] T.-J. King Liu, 2012 Symposium on VLSI Technology Short Course, June 2012.
[4] H. Lee et al., "Sub-5nm All-Around Gate FinFET for Ultimate Scaling," in Symp. VLSI Technology, June 2006.
[5] K. Cheng et al., "ETSOI CMOS for system-on-chip applications featuring 22nm gate length, sub-100nm gate pitch, and 0.08μm^2 SRAM cell," in Symp. VLSITechnology, June 2011.
[6] Advanced substrate news (ASN) #20, NEWS and VIEWPOINTS -SOI in Action.
[7] C. Shin et al., "Tri-gate bulk CMOS technology for improved SRAM scalability," in IEEE European Solid State Device Research Conference (ESSDERC), September 2010.
[8] B. Ho et al., "First demonstration of quasi-planar segmented-channel MOSFET design for improved scalability," in Symp. VLSI Technology, June 2012.
[9] H. Nam and C. Shin, "The design optimization and variation study of segmented-channel MOSFET using HfO_2 or SiO_2 trench isolation," in Proc. of VLSI-TSA, April 2013.