

© 2011 Samsung Electronics Co., Ltd.
 * Samsung internal test result, compared to Samsung 60 nano class DDR2 memory chip.
 Actual performance difference may vary depending on the test environment.

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



IDEDEC Newsletter

IDEDEC Newsletter | 통권 제184호 | 발행일 | 2012년 9월 30일 | 발행인 | 박인철 | 편집인 | 김민준 | 제작 | 푸을디자인
 기획 | 전희기 | 전화 | 0421-351-8535 | 팩스 | 042) 350-8540 | 홈페이지 | http://idec.or.kr
 E-mail | hg0929@idec.or.kr | 발행처 | 반도체설계교육센터(IDEDEC)

Vol.184

2012
 October

전력 IT를 위한 고효율 파워반도체 소자 기술동향 | 04

SystemC simple_bus의 이해 | 08 생물모방 지능의 뉴로모픽 VLSI와 자동차의 첨단 안전기술 | 14

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



www.towerjazz.com

TowerJazz Technology

TowerJazz Overview

TowerJazz manufactures integrated circuits for more than 150 customers worldwide. Our vision is to be the world leader in specialty foundry solutions as measured by our customers, employees and investors.

TowerJazz achieved the highest percentage of growth in 2010 vs. 2009. By further accelerating our growth rate in 2011, we are challenging to become the #1 specialty foundry.

Digital CMOS process technology, TowerJazz offers specialty CMOS processes with geometries ranging from 1.0- to 0.13-micron. We are divided into five business units:

- RF/High Performance Analog (including SiGe BiCMOS and RF CMOS offerings),
- Power Management (including our patented Y-Flash, the leading solution for NVM),
- CIS (CMOS Image Sensors),
- Mixed-Signal/CMOS and TOPS (Technology Optimization Process Services) to expand capacity or provide second sourcing.
- We also offer foundry MEMS enablement solutions combined with high volume 150mm and 200mm CMOS wafer manufacturing.

2010 Major IC Foundries

2010 Rank	2009 Rank	Company	Foundry Type	Location	2009 Sales (\$M)	2008 Sales (\$M)	09/08 Sales (%)	2010 Sales (\$M)	10/09 Sales (%)
1	1	TSMC	Pure-Play	Taiwan	10,556	8,989	-15%	13,307	48%
2	2	UMC	Pure-Play	Taiwan	3,070	2,815	-8%	3,965	41%
3	4	GlobalFoundries	Pure-Play	U.S.	0	1,101	N/A	3,510	219%
4	5	SMIC	Pure-Play	China	1,253	1,070	-21%	1,555	45%
5	3	TowerJazz	Pure-Play	Europe	282	309	18%	510	70%
6	7	Vanguard	Pure-Play	Taiwan	511	382	-25%	508	33%
7	6	Dongbu	Pure-Play	South Korea	490	395	-19%	495	25%
8	8	IBM	IDM	U.S.	400	335	-16%	430	28%
9	12	MagnaChip	IDM	South Korea	346	262	-24%	420	60%
10	10	Samsung	IDM	South Korea	340	290	-18%	400	38%
11	11	SSMC	Pure-Play	Singapore	340	280	-18%	330	18%
12	15	X-Fab	Pure-Play	Europe	368	212	-42%	320	51%
13	14	Hua Hong NEC	Pure-Play	China	280	240	-14%	295	23%
14	13	TI	IDM	U.S.	315	250	-21%	285	14%
15	16	Grace	Pure-Play	China	230	180	-22%	250	44%
---	3	Chartered*	Pure-Play	U.S.	1,743	1,540	-12%	0	N/A

Source: IC Insights, company reports

*Purchased by GlobalFoundries in 4Q09

The Global Specialty Foundry Leader

전력 IT를 위한 고효율 파워반도체 소자 기술동향

최근 에너지 위기와 환경규제 강화 및 친환경, 녹색성장 등의 이슈가 대두하여 에너지 절감과 환경보호분야에 IT 기술을 접목하는 패러다임이 부각되는 상황에서 고효율/친환경의 경쟁력 있는 제품 개발 및 그린오션을 주도할 수 있는 고효율 파워반도체 기술개발은 필수적이다. 이러한 파워 반도체는 명확한 정의는 없지만 1W 이상의 전력을 제어할 수 있는 능력을 가진 반도체를 일컫는다. 본 고에서는 전력 IT를 위한 고효율 파워반도체 소자에 대해 소개하기로 한다. (관련기사 P04~07 참조)

SystemC simple_bus의 이해

전적인 의미에서의 '시스템 설계'란 프로세서, 메모리, 디지털 및 아날로그 부품 등을 모아 원하는 기능을 수행할 수 있도록 보드 상에 결합하는 것이다. 이처럼 소프트웨어의 중요성이 크게 부각되지 않은 채 ASIC 등을 RTL에서 설계하고 결합하는 경우 HDL은 적절한 선택이라 할 수 있다[1]. 그러나 1990년대 후반부터 하드웨어와 소프트웨어가 결합하여 기능을 발휘하는 SoC가 주류를 이루면서, HDL로 시스템을 탐색 및 기술하고 검증하는데 어려움이 따랐다. 본 고에서는 SystemC simple_bus에 대해 알아보도록 한다. (관련기사 P08~12 참조)

생물모방 지능의 뉴로모픽 VLSI와 자동차의 첨단 안전기술

최근 사람 지능을 모방하고자 활발한 연구가 진행되고 있으며, 이를 통하여 다양한 환경하에서도 안정적인 인식 능력을 구현하고 활용하려 하고 있다. 신경망 인식기에 사용되는 머리와 상체 템플레이트 대신 차량의 특성 템플레이트를 이용하면, 뉴로모픽 지능을 이용하여 전방의 차량 물체를 인식하여 자동차의 안전 운전 지원에 활용이 가능함을 보여준다. 이와 같은 다양한 물체의 구별이나 상태인식의 뉴로모픽 기술은 차량용 안전 기술을 구현하는 전장장치 부품으로, 완성차 산업에서 요구하는 가격대와 성능의 안정성을 만족하게 하기 위하여 전용 뉴로모픽 VLSI 화가 되어야 한다. 본 고에서는 생물모방 지능의 뉴로모픽 VLSI와 자동차의 첨단 안전기술에 대해 알아보도록 한다. (관련기사 P14~18 참조)

IDEC October | 2012 news

MPW (Multi-Project Wafer)														
MPW 신청 현황						MPW 칩 제작 현황								
구분	공정	제작가능 면적 (mm² x 칩수)	채택 칩수	설계면적 (mm² x 칩수)	DB마감	Die-out	비고	구분	공정	제작 칩수	제작면적 (mm² x 칩수)	Die-out 예정일	현재상태	비고
114회 (12-7)	삼성 0.13	4x4mmx48	32	4x4mmx32	2012. 8.31	2013. 1.4	DB 검토중	109회 (12-2)	삼성 0.13	40	4x4mm x 40	2012. 8.3	제작완료	-Die:8,17 -PKG:9,7
115회 (12-8)	동부 0.18BCD	5x5mmx2	5	5x2.5mmx3 2.5x2.5mmx2	2012. 9.26	2013. 1.4	설계중	110회 (12-3)	동부 0.11	29	5x2.5mm x 22 2.5x2.5mm x 7	2012. 8.1	PKG 제작중	-Die:8,30 -PKG:10,20예정
	TJ0.18 CIS	2.5x2.5mmx4	4	2.5x2.5mmx4	2012. 10.15	2013. 2.22	설계중		M/H 0.18	20	4.5x4mmx20	2012. 9.3	제작중	-Die:10,20 예정
	TJ0.18 BCD	5x5mmx2	1	5x5mmx1	2012. 10.22	2013. 2.29	설계중		M/H 0.35	20	5x4mmx20	2012. 9.3	PKG 제작중	-Die:9,17 -PKG:10,5예정
116회 (12-9)	TJ0.18 RF	2.5x2.5mmx4	4	2.5x2.5mmx4	2012. 10.22	2013. 2.29	설계중	111회 (12-4)	동부 0.35BCD	7	5x2.5mmx4 2.5x2.5mmx3	2012. 8.30	PKG 제작중	-Die:8,30 -PKG:10,10예정
	동부 0.35BCD	5x2.5mmx6	8	5x2.5mmx4 2.5x2.5mmx4	2012. 10.10	2013. 1.16	설계중		TJ0.18 CIS	2	5x2.5mmx2	2012. 9.14	제작중	-Die:10,5 예정
	동부 0.11	5x2.5mmx30	33	5x2.5mmx27 2.5x2.5mmx6	2012. 10.2	2013. 2.6			TJ0.18 BCD	2	5x5mmx2	2012. 9.21	제작중	-Die:8,30
117회 (12-10)	M/H 0.18	4.5x4mmx20	20	4.5x4mmx20	2012. 11.12	2013. 3.4		112회 (12-5)	삼성 65nm	23	5x5mmx23	2012. 11.9	제작중	
	M/H 0.35	5x4mmx20	20	5x4mmx20	2012. 11.12	2013. 3.4	설계중		동부 0.18BCD	4	5x2.5mmx4	2012. 9.26	제작중	
	삼성 65nm	20개서버 (4x4mm)	19 (서버)	4x4mmx19	2012. 11.26	2013. 5.3		113회 (12-6)	동부 0.35BCD	9	5x2.5mmx4 2.5x2.5mmx4	2012. 10.10	제작중	
								114회 (12-7)	동부 0.18BCD	4	5x2.5mmx 4	2012. 9.26	제작중	
									M/H 0.18	20	4.5x4mm x 20	2012. 12.3	제작중	

* 2012년 MPW 모집이 완료됨. 2013년 MPW 지원 공정 및 일정은 2012. 11월 이후 공지 예정
* 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 9.28 기준임.

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

Chip Design Contest (CDC)																						
<p>● 2012 ISOC Chip Design Contest 개최</p> <p>가. 일정 및 장소 : 2012. 11. 5(월), 제주 라마다 호텔 나. 논문 접수 결과 : 144편(데모 : 16, 패널 : 128) 다. 주요 일정</p> <table border="1"> <tr> <th>선정안내</th> <th>최종논문접수</th> <th>발표(패널/데모)</th> </tr> <tr> <td>10. 4</td> <td>10. 8</td> <td>11. 5(제주, 라마다호텔)</td> </tr> </table> <p>라. 시상내역</p> <table border="1"> <tr> <th>시상명</th> <th>내역</th> </tr> <tr> <td rowspan="3">ISOC Design Award</td> <td>최우수상 (1팀)</td> <td>상장 및 상금 100만원</td> </tr> <tr> <td>우수상 (2팀)</td> <td>상장 및 상금 50만원</td> </tr> <tr> <td>SSCS 서울캠퍼스상(1팀)</td> <td>상장 및 상금 50만원</td> </tr> <tr> <td>ISOC CDC Poster Award</td> <td>우수상(0팀)</td> <td>상장 및 상금 20만원</td> </tr> <tr> <td></td> <td>* 참여팀수에 따라 수상팀 수는 달라질 수 있습니다.</td> <td></td> </tr> </table>	선정안내	최종논문접수	발표(패널/데모)	10. 4	10. 8	11. 5(제주, 라마다호텔)	시상명	내역	ISOC Design Award	최우수상 (1팀)	상장 및 상금 100만원	우수상 (2팀)	상장 및 상금 50만원	SSCS 서울캠퍼스상(1팀)	상장 및 상금 50만원	ISOC CDC Poster Award	우수상(0팀)	상장 및 상금 20만원		* 참여팀수에 따라 수상팀 수는 달라질 수 있습니다.		<p>● 20th KCS Chip Design Contest 개최</p> <p>가. 일정 및 장소 · 일 사: 2012년 2월 5일(화) (*참고: KCS 일정_2.4~6) · 장 소: 황성 성우리조트</p> <p>나. 논문 마감 : 10. 19(금) 다. MPW 설계팀 참여 대상 - 2011 MPW 설계팀 중 미참여팀 + 2012년(108~109회) 설계팀</p> <p>바. 시상 내역 : ISOC CDC와 동일</p> <p>* 문의 : 이의숙 (042-350-4428 yslee@idec.or.kr)</p>
선정안내	최종논문접수	발표(패널/데모)																				
10. 4	10. 8	11. 5(제주, 라마다호텔)																				
시상명	내역																					
ISOC Design Award	최우수상 (1팀)	상장 및 상금 100만원																				
	우수상 (2팀)	상장 및 상금 50만원																				
	SSCS 서울캠퍼스상(1팀)	상장 및 상금 50만원																				
ISOC CDC Poster Award	우수상(0팀)	상장 및 상금 20만원																				
	* 참여팀수에 따라 수상팀 수는 달라질 수 있습니다.																					

2012년 10월 교육프로그램 안내

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

강좌 일정 |

센터명	강의일자	강의제목	분류
한양대 IDEC	10월 5일	영상인식을 위한 특징 검출	설계
	10월 9일	스마트 융합과 미래	설계

■ 강좌일 : 10월 5일
■ 강좌 제목 : 영상인식을 위한 특징 검출
■ 강사 : 원치선 (동국대학교)

[강좌개요]
영상 내에 존재하는 특징점들을 검출하고 각 특징점의 고유 특징을 서술하는 서술자를 생성하는 것은 객체 추적과 모바일 클라이언트-서버의 영상 인식을 위한 중요한 전처리 과정으로 본 교육을 통해 최근까지 제안된 특징점 검출 및 서술 방법을 소개한다.

[수강대상]
· SoC 설계자
[강의수준]
· 중급
[강의형태]
· 이론

■ 강좌일 : 10월 9일
■ 강좌 제목 : 스마트 융합과 미래
■ 강사 : 김경호 (한양대학교)

[강좌개요]
애플이 촉발한 스마트폰에 의한 스마트혁명이 무엇인지 짚어보고 2012년 CE Show의 메인이슈를 파악한다. 급변하는 IT/가전 시장 및 기술환경에서 미래의 단말을 예측해보고, 보다 높은 사양과 앞선기능의 미래 제품을 구현하기 위해 필요한 SOC분야에서의 기술적인 난관들을 이해한다. 마지막으로 가까운 장래에 실현될 기술과 우리주변에서 사라지는 것들, 새롭게 부상하는 것들을 살펴봄으로써 수강생들이 미래를 그려보고 준비하는데 도움을 주고자 한다.

[수강대상]
· 전기전자관련 학과 학부생, 대학원생
[강의수준]
· 중급
[강의형태]
· 이론

[사전지식, 선수과목]
· 디지털 논리회로, SOC설계

* 문의 : 한양대 IDEC 안지혜 (031-400-4079, ipc@idec.hanyang.ac.kr)

1st Call for Papers

제20회 The 20th Korea Conference on Semiconductors 한국반도체학술대회

일시: 2013년 2월 4일 (월) - 6일(수)
장소: 성우리조트
논문접수마감: 2012년 10월 19일(금)
논문채택통보: 2012년 12월 05일(수)

■ 주관 동부하이텍
한국반도체산업협회
한국반도체연구조합

■ 주최 한국물리학회 반도체분과회
한국재료학회
대한전기학회 전기재료연구회
대한전기학회 MEMS 연구회
대한전자공학회 반도체재료 및 부품연구회
대한전자공학회 SoC 설계연구회
반도체설계교육센터 (IDEC)

■ 후원 IEEE Electron Device Society Korea Chapter
IEEE SSC Seoul Chapter
IEEE SSC Seoul Chapter

제20회 한국반도체학술대회
Dongbu HiTek

Analog Semiconductor Leaders' Forum

아날로그반도체 리더스 포럼

"Exploring new opportunities for analog technology in the smart convergence era"

· 일시: 2012년 10월 10일(수) 10:00~16:55
· 장소: 경기도 일산 킨텍스 전시장 2층 204호실

■ 행사안내

- 좌석이 한정되어 있으니 사전 등록을 서둘러 주시기 바랍니다.
- 행사 등록 및 진행 관련 사항은 아래 연락처로 문의하여 주시기 바랍니다.
연락처 032-880-4053 / youngja.lyu@dongbu.com
- 오시는 길: 지하철 3호선 대화역 1번출구 또는 2번 출구

Agenda

시 간	내 용	강연자
10:00~10:10	환영사	최창식 동부하이텍 사장
10:10~10:20	축 사	지식경제부
10:20~11:20	기조 연설 Remote Data Acquisition System & Technology Requirements	Timothy V. Kalthoff (텍사스인스트루먼트 최고기술책임자, Fellow)
11:20~12:20	시장 동향 Analog in the Twilight: The Role of Analog in an Expanding Digital Universe	Stephan Ohr (기브너 반도체담당 수석애널리스트)
12:20~13:30	중 식	
13:30~14:20	Addressing Analog, Mixed-Signal and Advanced Node Custom Design Scalability, Convergence and Throughput	Tom Beckley (레이덴스 부사장)
14:20~15:10	기술 동향 AMOLED 고속전류구동 및 다중출력 PMIC	조규형 KAIST 전기전자공학과 교수
15:10~16:00	Powering the Future	Albert Lu (ZTE 반도체 개발총괄)
16:00~16:50	아날로그반도체 제조공정 기술동향	박홍수 동부하이텍 부사장
16:50~16:55	맺음말	최창식 동부하이텍 사장

전력 IT를 위한 고효율 파워반도체 소자 기술 동향



고려대학교 전기전자전파공학부

성만영 교수
 연구분야 : 반도체 CAD
 E-mail : semicad@korea.ac.kr
 http://semicad2.korea.ac.kr



부러워할수록

서론

최근 에너지 위기와 환경규제 강화 및 친환경, 녹색성장 등의 이슈가 대두하여 에너지 절감과 환경보호분야에 IT 기술을 접목하는 패러다임이 부각되는 상황에서 고효율/친환경의 경쟁력 있는 제품 개발 및 그린오션을 주도할 수 있는 고효율 파워반도체 기술개발은 필수적이다. 이러한 파워 반도체는 명확한 정의는 없지만 1W 이상의 전력을 제어할 수 있는 능력을 가진 반도체를 일컫는다.

본론

파워 디바이스는 수 W에서 GW급까지의 전력변환이나 제어를 하는 반도체로, 정류 다이오드(Rectifying diode), 바이폴라 트랜지스터(Bipolar Transistor), 사이리스터(Thyristor), GTO(Gate Turn Off thyristor), 다이액(Diac), 트라이액(Triac), 파워 MOSFET(Metal Oxide Semiconductor Field Effect Transistor, 금속 산화막 반도체 전계 효과 트랜지스터), IGBT(Insulated Gate Bipolar Transistor, 절연 게이트 바이폴라 트랜지스터), IPM(Intelligent Power Module, 지능형 파워모듈) 등이 널리 알려졌다.

파워 반도체는 반도체 중에서도 매우 오랜 역사를 가지고 있으며, 그 탄생은 약 60년 전으로 거슬러 올라간다. 이를테면 바이폴라 트랜지스터 발명은 1948년, 사이리스터는 1957년으로 당초에는 저내압, 소용량에서 출발하였고, 파워 MOSFET가 1970년대에, 그리고 IGBT와 같은 MOS/바이폴라 소자가 1980년대에 소개되면서 전력전자산업은 집적회로가 가능하게 되었고, 그 후 눈부신게 진보를 거듭하여 고내압, 대전류, 고속 고주파, 고기능화로 발전되었다. 또한, MOSFET와 MOS/바이폴라 파워 소자들은 입력 게이트 전류가 낮아 아주 작은 게이트 구동회로를 사용할 수 있기 때문에 매우 집적도가 높다. 그림 1은 파워 반도체 소자들의 시대별 출현과정을 보여준다.

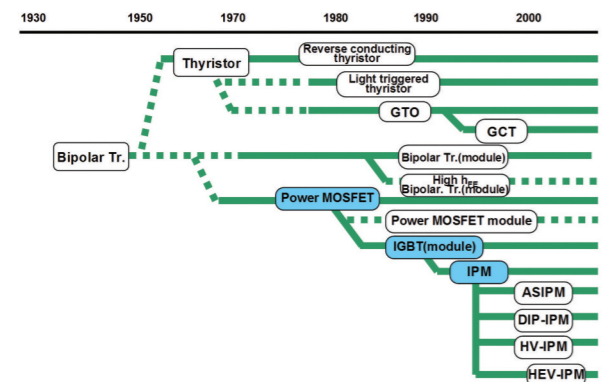


그림 1. 파워 반도체의 발전

파워 반도체의 응용사례를 살펴보면, 우선 가정에서는 에어컨, 냉장고, 세탁기, 청소기, 조광장치 등의 인버터 장치, IH(Induction Heating: 유도가열) 조리기에는 고주파 인버터용 IGBT가 사용된다. TV 수상기의 전원부에는 MOSFET, 소자회로에 사이리스터, 카메라 플래시에는 IGBT 또는 IPM이 널리 사용되고 있다. 자동차 엔진의 점화장치에는 바이폴라 트랜지스터나 IGBT, 제어용 IC를 하나의 패키지에 수용한 파워 IC가 사용되고 있다.

그 외에 바이폴라 트랜지스터, 파워 MOSFET 등을 박막 기판상에 조립한 HVIC(High Voltage Integrated Circuit, 고전압 집적회로) 및 원칩화된 스마트 파워 IC 등, 파워 반도체가 차량 제어계통과 헤드램프, 파워 윈도우, 파워시트, 미러 등 곳곳에 사용되고 있다.

도로에는 교통신호 램프의 점등 제어를 비롯하여 도로 표시판의 램프 제어에 IGBT, 사이리스터 등이 사용되고 있다. 항공과 철도 관련 전동차와 전철의 제어에는 고내압 IGBT(HV IGBT) 모듈, GTO, 구동과 보호 및 자기진단회로를 내장한 HV-IPM 등의 MOS계 파워 반도체가 주류이다[3]. 사무실과 공장에도 도처에 파워 반도체가 사용되고 있다.

예를 들면, 엘리베이터 제어에는 IGBT 모듈이나 IPM의 사용이 불가피하다. 초고층 빌딩에서 분속 750~900m의 엘리베이터 제어에는 1,200V, 600A급의 IGBT 모듈이 사용되며, 빌딩의 옥상에 제어실을 갖추지 않는 기계실 엘리베이터에는 제어반을 슬림으로 설계할 수 있는 IPM이 사용되고 있다.

형광등의 인버터에는 MOSFET, 안전 보안용의 화재경보기, 누전차단기, 비상등, 무정전 전원장치(UPS, CVCF 인버터), 전력을 절약하기 위한 목적의 파워세이버 회로, 심지어 프린터 CRT 모니터, FAX, 전화기에도 IGBT, IPM 등의 파워 반도체가 사용되고 있다. 공장에서는 프로그래머블 로직 컨트롤러(PLC), 로봇관련 제어장치, 컴퓨터 제어 공장기계(CNC), 크레인(Crane), 호이스트(Hoist), 컨베이어 벨트, 자동반송기계 등의 모터 제어용 인버터에는 IPM과 IGBT는 필수적이다.

이외에 제련소나 제지 공장의 압연기에는 수천 kW 이상의 인버터가 필요하며, 여기에는 대용량 GTO, IGCT, 고내압 IGBT 모듈이 사용되고 있다. 유도가열과 고주파 가열장치, 다이분당, 와이어분당 장치 플라스틱 성형기, 풍력발전 및 태양광 발전의 컨버터에는 1,200V~1,700V, 800A 급의 IPM이나 IGBT가 사용되며, 마이크로 가스터빈

이나 연료전지 등에도 IPM 또는 IGBT 모듈 등의 파워 디바이스가 사용되고 있다. X선장치, MRI, CT와 같은 의료기기의 전원부에는 고속 IGBT 모듈 등이 사용되고 있다. 그림 2는 동작 주파수와 정격 용량별 사용 가능한 파워 디바이스의 종류를 나타낸다.

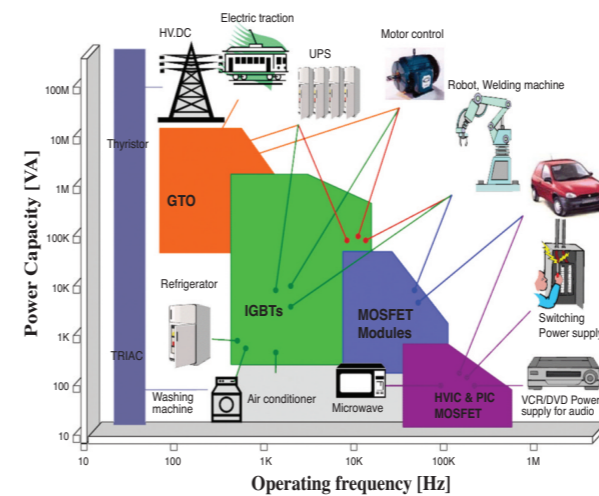


그림 2. 동작 주파수와 정격 용량별 파워 소자의 분류

지금까지의 파워 소자라고 하면 사이리스터나 트라이액, 파워 트랜지스터 등이 일반적이었다. 현재의 파워 소자는 파워 MOSFET나 IGBT, IPM으로 변화되어가고 있다. 더구나 사용하기 쉬워진 IPM과 트렌치(Trench) IPM이 새로운 제품 설계의 주류로 자리 잡아가고 있다[2].

IGBT는 파워 MOSFET의 입력 특성과 바이폴라 트랜지스터의 출력 특성을 결합한 파워 스위칭 소자이다. 이 소자는 MOSFET의 전압 입력으로 구동되지만, 그 출력 전류는 BJT(Bipolar Junction Transistor, 바이폴라 접합 트랜지스터)의 특성을 따른다.

그렇기 때문에 이 소자는 절연 게이트 바이폴라 트랜지스터(IGBT)라고 불린다. 이 소자의 장점중 하나는 파워 바이폴라 트랜지스터의 낮은 온-상태 저항과 파워 MOSFET의 높은 입력 임피던스를 갖는다는 것이다. 파워 MOSFET의 높은 입력 임피던스로 게이트 제어 턴-온(Turn-on)과 턴-오프(Turn-off)가 가능하다[1].

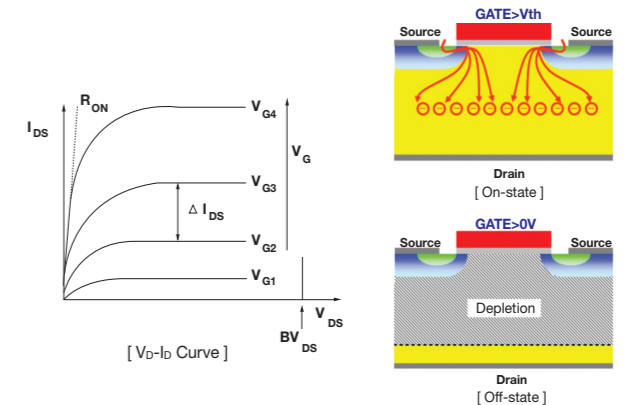


그림 3. 파워 MOSFET의 On-Off 동작 및 Vd-ID 특성

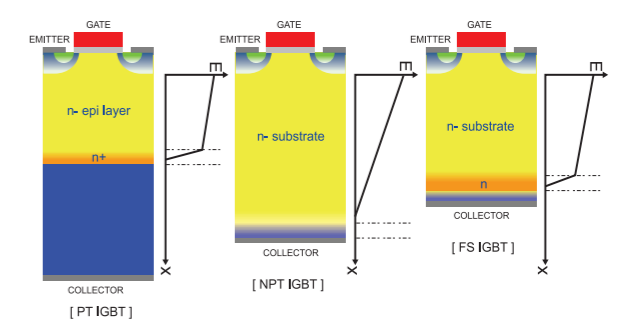


그림 4. 파워 IGBT의 종류 및 전계 특성

파워 반도체의 발전은 이상적인 스위치를 찾아온 역사라고 할 수 있다. 현재까지 여러 분야의(산업, 교통, 발전, 송배전, 우주 등) 전력전자시스템의 요구가 증가해왔기 때문에 이상적인 스위치에 관한 연구는 지속되어져 왔고 앞으로도 그럴 것이다. 서로 다른 기술들과 몇몇 분야의 연구 성과들(예를 들어, 실리콘 기술, 금속, 세라믹, 전기공학, 기계공학 등)이 적용되어 반도체의 특성을 향상하게 시켰다.

연구개발의 목표는 온-상태 손실과 스위칭 손실을 줄이고, 반도체의 안전 동작 영역을 증가시키고(전류와 전압 모두) 가능한 높은 스위칭 주파수에서 동작하고, 간단하고 효율적인 드라이버 회로를 갖도록 하는 것이다. 파워 소자들은 파괴되기 쉬우므로 과전류 또는 고온 등에 대비한 보호 기능들이 매우 중요하다. 몇 가지 집적회로의 기능을 갖

는 개별소자를 하나의 칩 위에 집적시킴으로써 파워시스템 구조는 굉장히 간단해지고 시스템의 신뢰성도 향상된다. 그 예로 대전력 파워 소자와 저전력 컨트롤 회로를 하나의 칩 위에 집적시켜 "스마트 파워 IC"라는 새로운 기술이 나타나게 되었으며, 이 스마트 파워 IC 기술 덕분에 소자의 개수와 시스템 가격, 크기, 무게를 상당히 줄일 수가 있게 되었다[3]. 스마트 파워 IC 기술은 소신호 논리 회로와 파워 소자를 같은 칩 위에 집적시키는 것이다.

스마트 파워 IC 기술의 주요한 3가지 기능은 전력 컨트롤, 센싱/보호 그리고 인터페이스이다. 스마트 파워 IC 기술을 이용하여 마이크로프로세서가 전력부하를 컨트롤할 수 있도록 구동회로나 레벨 쉬프팅 기능을 하나의 칩에 넣을 수 있다. 마이크로프로세서와 다른 회로들보다 큰 고전압을 요구하는 용량성 부하를 갖는 디스플레이 구동은 이러한 칩이 아주 유용하다. 그림 5는 스마트 파워 IC 기술의 전체적인 구성과 3가지 주요한 기능을 구성하는데 필요한 요소들을 보여준다 [5].

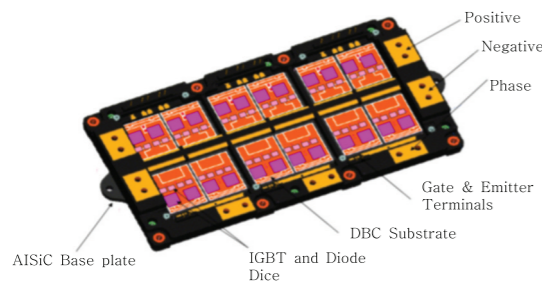
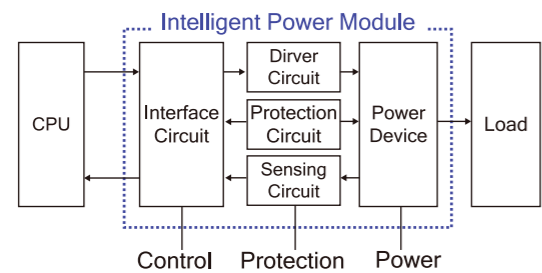


그림 5. IPM 모듈의 구성

전력 컨트롤 기능은 외부 센서를 활용하여 주요 시그널의 오차를 탐지하고 이 오차를 보상하도록 파워 소자를 제어한다. 예를 들어 전압 제어는 소자의 출력전압을 탐지하여 소자의 입력 전류를 제어한다. 전력 컨트롤은 여러 파워 소자와 구동회로에 유용하다. IGBT와 파워 MOSFET와 같은 새로운 MOS 구동 소자들은 낮은 입력 게이트 전류와 간단한 구동회로 때문에 가장 선호되고 있다. 전력 컨트롤은 30V 정도의 고전압이 구동 가능하도록 설계되어야 한다.

파워 소자는 소자 자체 혹은 주변 회로들에 해를 끼칠 수 있는 파괴 모드로 가기 쉬우므로 센싱 기능과 보호기능을 갖는 스마트 파워 IC 기술을 추가하여 전체시스템의 오동작이나 소자자체가 파괴될 가능성을 줄여준다. 센싱과 보호기능은 파워 소자를 보호하기 위해 외부 센서로 피드백을 이용하여 과전압, 과전류 또는 과온도를 탐지할 수 있

다. 고속 바이폴라 트랜지스터와 고성능 아날로그 회로는 내장된 센싱 기능을 이용해야 하는데 그 이유는 급격히 파괴전류 레벨로 증가하기 때문에 아주 빠른 응답속도가 필요하기 때문이다. 인터페이스 기능으로 파워 소자와 마이크로프로세서 또는 다른 복잡한 디지털 회로와 연결이 가능하다. 파워 소자는 인터페이스 회로를 통해 마이크로프로세서로부터 제어신호를 받을 수 있을 뿐만 아니라 과도한 전압 및 전류 레벨에 대한 정보를 마이크로프로세서로 보낼 수 있다. 저전력 디지털 회로와 통신을 할 수 있도록 암호/복호 기능을 수행하기 위해서도 인터페이스 기능이 필요하다. 그림 6은 IPM 모듈 개발을 위한 각각의 세부 기술을 나타내었다.

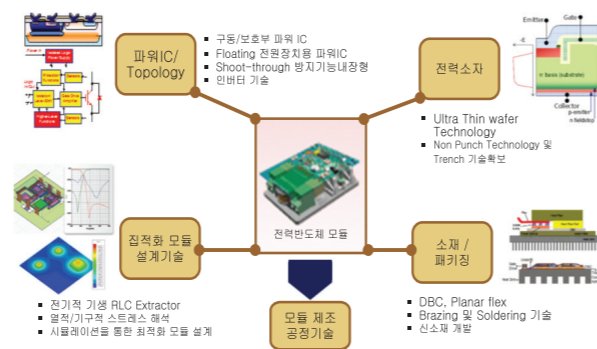


그림 6. IPM 모듈 개발을 위한 세부 기술 사항

현재의 스마트 파워 IC 기술은 스마트 개별 소자와 HVIC의 두 가지 주요 분야로 나뉠 수 있다. 스마트 개별소자는 개별 파워 소자에 제어회로와 센싱회로가 하나의 칩 위에 집적된 소자이다. 수요자의 파워 소자에 대하여 높은 신뢰성을 요구하기 때문에 이러한 시장이 생겨나게 되었다. 파워 소자에 같이 집적된 센싱회로, 제어회로, 보호회로 때문에 견고하고(robust) 강건한(rugged) 시스템 디자인을 할 수가 있으며, 여러 가지 격리기술로 바이폴라 트랜지스터, 저항, 소신호 MOSFET와 파워 소자를 하나의 칩 위에 구현하는 것이 가능하다는 장점이 있다.

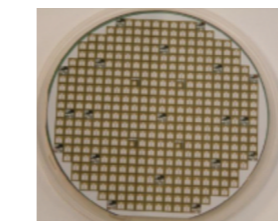
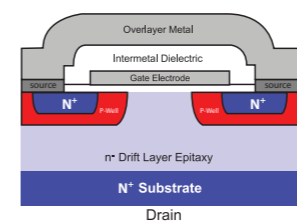
이러한 스마트 개별소자는 자동차용 전자산업에 아주 많이 사용되는 반면 HVIC는 통신분야, 디스플레이 구동, 모터 제어 일부에 한해 개발되고 있다[5]. HVIC는 고전압 소자와 저전압 컨트롤 회로 간의 연결을 위해 여러 가지 격리기술을 사용한다. 이러한 격리기술은 자기 격리(Self-isolation), 접합격리(Junction isolation), 그리고 절연격리(Dielectric isolation)[6]의 3가지 종류로 크게 나눌 수가 있다. 스마트 개별소자와는 달리 HVIC는 모든 컨택이 칩의 상단 표면에 위치해 있어 집적이 가능하다. 수직형 소자 즉 전류의 흐름이 수직인 파워 소자들은 파워 소자의 고전압 능력과 저전압 소자의 전류 이득간의 상관관계 설계 때문에 HVIC에는 거의 사용되지 않는다.

RESURF(REDuced SURface Fields, 표면전계감소) 원리의 연구로 수평형 고전압 소자가 HVIC에 사용 가능하게 되었다. 수평형 고전압 소자는 아주 얇은 에피택셜(Epitaxial)층에서 제작되고 저전압 컨트롤 소자의 성능에 영향을 끼치지 않는다. 수평형 고전압 소자의 주요 장

점은 파워 소자의 게이트 소스와 드레인 영역 간의 거리인 드리프트(Drift) 영역의 길이를 늘이거나 줄임으로써 항복전압의 최적화가 가능하다는 것이다. 포토리소그래피(Photo-lithography) 공정으로 항복전압을 제어할 수 있게 됨으로써 동일한 칩 위에 다른 정격을 가진 최적화된 고전압 소자들을 제작할 수 있게 되었다.

하지만 실리콘이 물리적 한계에 다다름에 따라 실리콘 카바이드(SiC)와 같은 새로운 물질이 이상적인 스위치를 실현하기 위해 고려되고 있다[4].

Schematic DMOSFET Cross-section



100mm 4H-SiC Wafer
390 Devices per wafer
4.08 mm x 4.08 mm

그림 7. 4H-SiC Power MOSFET

결론

지금까지 간략하게 고효율 파워 반도체 소자 기술에 대하여 살펴보았다. 그러나 지면 관계상 더욱 깊은 내용을 다루지 못함을 아쉽게 생각한다.

전력 IT를 위한 고효율 파워 반도체 소자 기술은 에너지 과소비 시대에 전력 효율 향상을 위해 반드시 필요한 핵심 기술이고, 전기를 사용하는 모든 제품과 우리의 생활에 많은 영향을 주고 있다. 게다가 국내외 시장에 미치는 파급 효과도 매우 크다고 할 수 있으므로, 파워 반도체 소자의 기술 동향을 예의주시하면서 국산화 개발에 많은 노력이 필요하다.

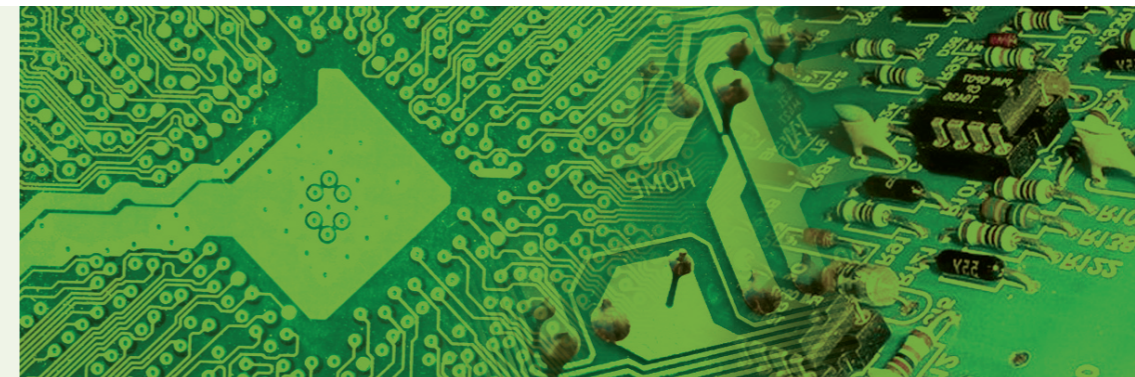
Reference

- [1] B.Jayant Baliga, "Power semiconductor devices", PWS, 199
- [2] 강이규, 성만영, "인텔리전트 파워 IC의 기술동향", 전기전자재료학회지, vol 15, 3, 2002
- [3] 최연익, "전력 반도체 소자의 어제와 내일", 전기학회지, vol 46, 2, 1997
- [4] V. Barkhordarian, "Power MOSFET Basic", International Rectifier
- [5] "Fairchild IGBT Basics", Fairchild Semiconductor Korea.

SystemC simple_bus의 이해



충북대 전자정보대학 반도체공학과
 송기용 교수
 연구분야 : SoC 설계 · 검증, SystemC 모델링
 E-mail : gysong@cbnu.ac.kr
 http://bnadi.cbnu.ac.kr/~gysong



본문 키워드

서론

고전적인 의미에서의 '시스템 설계'란 프로세서, 메모리, 디지털 및 아날로그 부품 등을 모아 원하는 기능을 수행할 수 있도록 보드 상에 결합하는 것이다. 이처럼 소프트웨어의 중요성이 크게 부각되지 않은 채 ASIC 등을 RTL에서 설계하고 결합하는 경우 HDL은 적절한 선택이라 할 수 있다[1]. 그러나 1990년대 후반부터 하드웨어와 소프트웨어가 결합하여 기능을 발휘하는 SoC(System-on-a-Chip)가 주류를 이루면서, HDL로 시스템을 탐색 및 기술하고 검증하는데 어려움이 따랐고, 이를 극복하고자 추상화 수준의 방법론을 하드웨어 관점에서 구현하는 언어의 필요성이 제기되었으며, 그 대표적인 것이 C++를 기반으로 하는 SystemC 이다.

본론

SystemC는 Synopsys, Inc. 와 Univ. of California, Irvine이 공동으로 진행한 scenic 프로젝트와 Frontier Design, IMEC 연구소 등의 연구 결과를 이용하여 OSCI(Open SystemC Initiative)의 Language Working Group에서 개발하였다. 2000년 중반 이후 OSCI는 시스템수준 설계 개념을 채택한 SystemC 2.0을 개발하면서 core language와 개별적인 라이브러리를 분리하고, 동적 감응, 사용자 정의 인터페이스, 채널, 포트 등을 포함하였다[1][2].

SystemC를 사용한 설계의 주요 목적 중 하나는 RTL보다 높은 추상화 수준에서 소프트웨어와 하드웨어 또는 이 둘을 결합한 시스템 수준의 모델링을 가능하도록 하는 것이다.[1] 또한 SystemC는 일반적인 C++ 범용 컴파일러에 SystemC 클래스 라이브러리를 추가함으로써 별도의 디버깅 환경 구축 없이 기존의 C/C++ 프로그램 개발 환경만으로 시뮬레이션할 수 있도록 했다. 이를 통해 다양한 알고리즘 및 기능들을 검증 또는 최적화할 수 있어 하드웨어/소프트웨어 통합 개발에 많은 장점을 제공한다.

본 고에서는 SystemC 2.0.1에서 예제로 제공하는 simple_bus의 구조와 동작을 살펴봄으로써 SystemC의 이해를 돕고자 한다.

simple_bus의 소개

simple_bus의 전체 소스 코드와 문서들은 SystemC 사이트[3]에서 다운로드 받아 실행시켜볼 수 있다. simple_bus는 사이클 상세에 기반을 두는 플랫폼 전송 수준의 버스 모델로 전체 소스 코드의 길이가 1,000줄 정도이며, TLM 수준의 설계 방법에 대한 예제로 적당하다. simple_bus에서 'simple'은 파이프라인, 분할전송, 마스터 대기상태 등의 요소들을 가지고 있지 않다는 의미이고 버스에서의 전송을 핀 상세 수준이 아닌, 함수 호출을 통한 데이터 이동으로 구현하고 있다. 버스에 연결되어 시스템을 구성하는 각 모듈은 버스에서 제공하는 인터페이스를 사용하여 통신 채널에 접근하고, 전송은 채널에

구현된 읽기/쓰기 함수에 의해 이루어진다. simple_bus는 다음과 같은 블록들로 구성되어 있다.

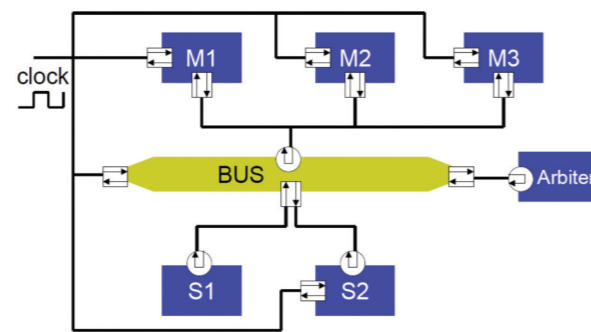
◆Master: CPU 또는 DSP와 같은 버스 마스터로 3가지 유형 - direct, non_blocking, blocking -의 인터페이스를 가진다.

◆Slave: 마스터에 의한 전송 요청에 응답하는 다양한 디바이스들이 슬레이브가 될 수 있으며, simple_bus의 구성에는 2가지 유형 - fast_memory, slow_memory - 이 사용된다. fast_memory는 단일 사이클에서 읽기/쓰기 동작을 수행하고, slow_memory는 다수의 사이클을 필요로 한다.

◆Arbiter: 버스에 의해 호출되며, 다수의 버스 사용 요청으로부터 각 마스터의 우선 순위와 lock 상태에 따라 중재한다.

◆Bus: 마스터, 슬레이브, 중재기가 연결되어 전송이 실제로 수행되며 계층 채널로 구현되어 있다.

그림 1은 simple_bus의 구성도이다.



simple_bus의 동작

simple_bus의 모듈별 동작

본 절에서는 simple_bus를 구성하는 각 모듈의 동작을 살펴본다. simple_bus의 충분한 이해를 위해서는 전체 소스 코드의 검토가 필요하나, 지면의 제약으로 각 모듈의 부분 코드를 가지고 구성이나 해당 동작을 설명한다.

마스터 모듈

direct, non_blocking, blocking의 3가지 동작 유형을 가지며, 모든 마스터는 템플릿으로 선언된 sc_port(class IF, unsigned

n_channels= 1)를 이용하여 포트를 선언한다. sc_port(class IF, int N)는 sc_interface로부터 파생되는 클래스로 실제로 전송되는 데이터의 타입과 포트에 연결될 수 있는 인터페이스의 개수를 지정한다. sc_port는 포트에 연결된 인터페이스의 주소를 반환하는 연산자 operator -> () 를 이용해서 인터페이스 함수를 호출하여 채널에 접근한다[1].

◆direct 유형: 버스 프로토콜을 따르지 않고, 슬레이브에 즉시 접근하는 방식으로 디버깅 용도로 사용되며, 슬레이브로 사용되는 메모리의 지정 주소에 직접 접근한다. 시뮬레이션 수행 시 슬레이브의 내용 확인, 버스의 동작 감시에 이용하는 것으로 실제 구현 하드웨어에서는 사용되지 않으며, sc_port(simple_bus_direct_if)를 이용하여 포트를 선언한다.

```
#ifndef _simple_bus_master_direct_h
#define _simple_bus_master_direct_h

#include <systemc.h>
#include "simple_bus_direct_if.h"

SC_MODULE(simple_bus_master_direct)
{ // ports
  sc_in_clk clock;
  sc_port<simple_bus_direct_if> bus_port;

  SC_HAS_PROCESS(simple_bus_master_direct);
  ... ..
}; // end class simple_bus_master_direct
#endif
```

인터페이스 헤더파일 simple_bus_direct_if.h는 순수가상함수의 형태로 direct_read() / direct_write()를 선언한다.

```
class simple_bus_direct_if : public virtual sc_interface
{
public:
  // direct BUS/Slave interface
  virtual bool direct_read(int *data, unsigned int address) = 0;
  virtual bool direct_write(int *data, unsigned int address) = 0;
}; // end class simple_bus_direct_if

#endif
```

◆non_blocking 유형: sc_port(simple_bus_non_blocking_if)로 포트를 선언하고, 우선순위와 버스 및 lock 상태에 의해 즉시수행 여부가 결정되며 read() / write()로 1개의 데이터를 전송한다.

```
SC_MODULE(simple_bus_master_non_blocking)
{
  sc_in_clk clock;
  sc_port<simple_bus_non_blocking_if> bus_port;
```

```
SC_HAS_PROCESS(simple_bus_master_non_blocking);

simple_bus_master_non_blocking(sc_module_name _name
, unsigned int unique_priority
, unsigned int start_address
, bool lock, ... ..)

: sc_module(_name) , ... ..
{
  SC_THREAD(main_action);
  sensitive << clock.pos();
}

void main_action();
... ..;
};
#endif
```

SC_THREAD 프로세스가 clock.pos()에 반응하여 main_action()이 수행된다. main_action()은 bus_port를 통해 read() / write()를 호출하여 동작을 수행하며, 버스의 상태를 확인해 동작을 일시 정지하거나, 다음 동작을 위한 주소와 데이터값을 경신한다.

```
#include "simple_bus_master_non_blocking.h"

void simple_bus_master_non_blocking::main_action()
{
  int mydata;
  int cnt = 0;
  unsigned int addr = m_start_address;

  wait(); // ... for the next rising clock edge
  while (true)
  {
    bus_port->read(m_unique_priority, &mydata, addr, m_lock);
    while ( (bus_port->get_status(m_unique_priority) != SIMPLE_BUS_OK) &&
           (bus_port->get_status(m_unique_priority) != SIMPLE_BUS_ERROR))
    {
      ... ..
      wait();
      mydata += cnt;
      cnt++;
    }
    ... ..
  }
}
```

◆blocking 유형: burst_read() / burst_write() 함수를 이용하여 일괄전송을 수행하며, simple_bus_blocking_if에 선언되는 burst_read() / burst_write() 함수의 전달인자로 데이터 전송 시작 주소와 전송할 데이터의 크기를 주고, 일괄전송을 완료한 후 인터페이스 함수를 되돌린다. 다음 코드는 simple_bus의 버스에 구현된 burst_read() 코드의 일부로 하단 부분의 wait(...)에서 일괄전송의 완료를 통지하고 버스상태를 반환한다.

```

simple_bus_status simple_bus::burst_read(unsigned int unique_priority
                                     , int *data
                                     , unsigned int start_address
                                     , unsigned int length
                                     , bool lock)
{
    ...
    simple_bus_request *request = get_request(unique_priority);
    request->do_write          = false; // we are reading
    request->address          = start_address;
    ...
    if (lock)
        request->lock = (request->lock == SIMPLE_BUS_LOCK_SET) ?
            SIMPLE_BUS_LOCK_GRANTED : SIMPLE_BUS_LOCK_SET;
    request->status = SIMPLE_BUS_REQUEST;
    -----read() , burst_read()가 동일한 코드 -----
    wait(request->transfer_done);
    wait(clock->posedge_event());
    return request->status;
}
    
```

■ 버스 모듈

버스는 SC_METHOD로 등록되는 프로세스와 인터페이스 함수의 정의, 중재기 포트, 다양한 슬레이브 연결을 위한 다중포트[4]를 가지는 계층채널로 구성된다. 버스는 마스터로 호출되는 인터페이스를 구현함과 동시에, 버스에 의해서 호출되고 다른 모듈에 구현된 슬레이브/중재기 인터페이스를 선언하고 있다.

```

#include "simple_bus_master_non_blocking.h"
...
#include "simple_bus_slave_if.h"
#include "simple_bus_arbiter_if.h"

class simple_bus
: public simple_bus_direct_if, public simple_bus_non_blocking_if
, public simple_bus_blocking_if, public sc_module
{
public:
    sc_in_clk clock;
    sc_port<simple_bus_arbiter_if> arbiter_port;
    sc_port<simple_bus_slave_if, 0> slave_port;
    ...
    SC_HAS_PROCESS(simple_bus);
    ...
    {
        SC_METHOD(main_action);
        dont_initialize();
        sensitive << clock.neg();
    }
}
    
```

버스는 SC_HAS_PROCESS 매크로를 통해 구성자를 기술한다. 내부 프로세스로 SC_METHOD가 사용되는데, SC_METHOD는 다른 프로세스와는 다르게 독자적인 스레드를 갖지 않고, 감응 리스트에 등록된 사건이 발생하면 C/C++에서의 함수와 같이 해당 프로세스가 수행된다. 따라서 이 프로세스는 내부에 wait()와 같이 프로세스 수행을 일시 정지시키는 함수를 사용하지 못한다[2].

simple_bus의 내부 스레드는 main_action() 함수의 수행으로, 대기 중인 수행요청에 대하여 중재기 처리를 통해 다음에 수행될 요청을 지정하는 get_next_request()와 지정된 요청을 수행하는 handle_request(), lock 상태를 조정하는 clear_locks()로 구성된다. 중재기에 의해 수행 대상으로 지정되면, 요청상태를 일시적으로

SIMPLE_BUS_WAIT로 변경하고 슬레이브를 선택한다.

■ 슬레이브 모듈

슬레이브 인터페이스는 버스와 슬레이브 사이의 통신을 기술하고 있으며, 버스에 선언되어 있고 버스는 다중포트 기능을 이용하여 이 인터페이스를 구현하는 다수의 슬레이브와 연결을 할 수 이 있다. 다음 코드는 slow_memory의 구현을 보인다.

```

#include <systemc.h>
#include "simple_bus_types.h"
#include "simple_bus_slave_if.h"

class simple_bus_slow_mem
: public simple_bus_slave_if , public sc_module
{
public:
    ...
    SC_HAS_PROCESS(simple_bus_slow_mem);
    simple_bus_slow_mem(sc_module_name name_, ... )
        : sc_module(name_) , ...
    {
        // process declaration
        SC_METHOD(wait_loop);
        dont_initialize();
        sensitive << clock.pos();
    }
}; // end class simple_bus_slow_mem
    
```

버스는 포트를 통해 slow_memory / fast_memory에 구현된 인터페이스 함수를 호출한다.

■ 중재기 모듈

데이터의 전송을 위해 동일한 버스에 두 개 이상의 마스터가 연결되어 있다면 각 마스터의 버스 요청을 적절하게 통제하고, 연결하기 위한 부분이 필요하다. simple_bus는 중재기를 클래스로 구현하고 있다. 중재기는 전용 인터페이스인 simple_bus_arbiter_if에 의해 버스에 연결되고, 다음과 같이 큐에 저장된 하나 이상의 요청으로 호출된다.

```

simple_bus_request *arbitrate(const simple_bus_request_vec &Q);
    
```

arbitrate()는 마스터의 요청에 대하여 각 요청의 우선순위, stauts 및 lock 상태를 조사한 후 수행될 요청을 지정하여 반환한다.

■ 서브 코드

simple_bus를 구현하기 위해서는 버스의 상태, lock 관련 정보, 디버깅을 위한 정보 등 다양한 내용이 정의 및 선언되어야 한다. 이러한 것들은 버스를 구현하면서 변수로 선언해 놓아도 되지만, 다양한 위치에서 사용되는 것을 감안할 때, 독립적인 코드로 분리하는 것이 효과적이다.

simple_bus는 이러한 코드들을 simple_bus_request.h , simple_bus_types.h/cpp, simple_bus_tools.cpp 등으로 구분해 놓고 있으며, 이 코드들을 사용하고자 하면 단지 #include "file_name.h" 형태로 추가하면 된다.

simple_bus의 전체 동작

C/C++가 main()에서 동작을 시작하는 것처럼 SystemC는

sc_main()에서 모든 동작을 시작한다. simple_bus도 simple_bus_main.cpp 파일의 sc_main()을 통해 모든 동작을 시작하며, sc_main()이 수행되면 simple_bus_test 모듈에서 simple_bus를 구성하는 각 모듈의 생성과 초기화, 생성된 모듈 간의 포트 연결이 진행되고, 연결된 모듈들은 각각 자신의 동작 수행을 시작한다.

각 마스터 모듈은 클럭의 상승에지에서 내부 프로세스인 SC_THREAD가 동작하여 버스 사용을 요청한다. 버스는 클럭의 하강에지에서 동작을 수행하며, 중재기의 호출을 통하여 일련의 사용 요청 중에서 다음에 실행될 마스터의 요청을 선택한다. 이후 선택된 마스터의 요청에 따라 슬레이브를 검색하고, 함수를 호출하여 해당 기능을 수행한다.

마스터의 선택을 위해 사용되는 중재기는 미리 정해진 규칙에 따라 매 클럭마다 가장 우선순위가 높은 요청을 선택한다. 그러나 이미 사용하고 있는 마스터가 burst_read() / burst_write()와 같이 연속적인 데이터 처리를 요청하고 있으면 종료되지 않았고, lock 상태가 SIMPLE_BUS_LOCK_SET인 경우, 더 높은 우선순위의 요청을 거부하고 직전 지정 마스터를 연속적으로 선택한다.

최종적으로 호출된 슬레이브의 read() / write() 함수는 즉시 실행되며, slow_memory는 대기상태의 삽입으로 지연 수행된다. 슬레이브가 전송 과정을 완료하면 버스의 상태가 경신되고 마스터로부터의 다음 요청이 진행된다. 마스터는 클럭의 상승에지에서, 버스와 중재기는 클럭의 하강에지에서 활성화되도록 설계하는 방식을 '2상 동기화 구조'라고 부르며 이는 TLM 수준 설계에 매우 유용한 방법으로 SystemC가 제공하는 기초채널과 하드웨어를 모델링하기 위한 시뮬레이션 커널의 request_update / update 방식과도 잘 부합한다[1].

simple_bus를 시뮬레이션하기 위해 simple_bus_test.h라는 테스트 벤치를 작성한다. simple_bus_test.h는 SC_CTOR 매크로를 이용하여 각 모듈을 생성 및 초기치를 할당하고, 이후 사례화 된 각 모듈을 연결한다. 이때 각 마스터의 우선순위와 슬레이브의 시작 주소, 전송 요청 사이의 대기시간 등이 지정된다.

다음 코드는 simple_bus_test.h의 모듈 생성과 초기치 할당을 수행하는 부분이다.

```

SC_CTOR(simple_bus_test)
: C1("C1")
{
    // create instances
    master_b = new simple_bus_master_blocking("master_b", 4, 0x4c, false, 300);
    master_nb = new simple_bus_master_non_blocking("master_nb", 3, 0x38, false, 20);
    master_d = new simple_bus_master_direct("master_d", 0x78, 100);
    mem_fast = new simple_bus_fast_mem("mem_fast", 0x00, 0x7f);
    mem_slow = new simple_bus_slow_mem("mem_slow", 0x80, 0xff, 1);
    bus = new simple_bus("bus");
    arbiter = new simple_bus_arbiter("arbiter");
    ...
}
    
```

non_blocking과 blocking 마스터의 우선순위는 각각 3, 4로 주어지고, fast_memory는 0x00 ~ 0x7f, slow_memory는 0x80 ~ 0xff의 주소 범위를 가진다. 다음은 시뮬레이션 결과의 일부를 보여 주고 있다.

```

SystemC 2.0.1 --- Apr 28 2012 16:20:52
Copyright (c) 1996-2002 by all Contributors
ALL RIGHTS RESERVED
0 top.master_d : mem[78:87] = (0, 0, 0, 0)
0 top.bus : burst_read(4) @ 4c
0 top.bus : read(3) @ 38
0.5 top.bus : request (4) [SIMPLE_BUS_REQUEST]
0.5 top.bus : request (3) [SIMPLE_BUS_REQUEST]
0.5 top.arbiter :
R[4](-SIMPLE_BUS_REQUEST@4c)
R[3](-SIMPLE_BUS_REQUEST@38) -> R[3] (rule 3)
0.5 top.bus Handle Slave(3)
--> status=(SIMPLE_BUS_OK)
1 top.bus : write(3) @ 38
1.5 top.bus : request (4) [SIMPLE_BUS_REQUEST]
1.5 top.bus : request (3) [SIMPLE_BUS_REQUEST]
1.5 top.arbiter :
R[4](-SIMPLE_BUS_REQUEST@4c)
R[3](-SIMPLE_BUS_REQUEST@38) -> R[3] (rule 3)
1.5 top.bus Handle Slave(3)
--> status=(SIMPLE_BUS_OK)
2.5 top.bus : request (4) [SIMPLE_BUS_REQUEST]
2.5 top.arbiter :
R[4](-SIMPLE_BUS_REQUEST@4c) -> R[4] (rule 3)
2.5 top.bus Handle Slave(4)
--> status=(SIMPLE_BUS_OK)
3.5 top.bus : request (4) [SIMPLE_BUS_WAIT]
3.5 top.arbiter :
R[4](-SIMPLE_BUS_WAIT@50) -> R[4] (rule 3)
3.5 top.bus Handle Slave(4)
--> status=(SIMPLE_BUS_OK)
...
...
14.5 top.bus : request (4) [SIMPLE_BUS_WAIT]
14.5 top.arbiter :
R[4](-SIMPLE_BUS_WAIT@7c) -> R[4] (rule 3)
14.5 top.bus Handle Slave(4)
--> status=(SIMPLE_BUS_OK)
15.5 top.bus : request (4) [SIMPLE_BUS_WAIT]
15.5 top.arbiter :
R[4](-SIMPLE_BUS_WAIT@80) -> R[4] (rule 3)
15.5 top.bus Handle Slave(4)
--> status=(SIMPLE_BUS_WAIT)
16.5 SLV [128]
16.5 top.bus Handle Slave(4)
--> status=(SIMPLE_BUS_OK)
...
19.5 top.bus : request (4) [SIMPLE_BUS_WAIT]
19.5 top.arbiter :
R[4](-SIMPLE_BUS_WAIT@88) -> R[4] (rule 3)
19.5 top.bus Handle Slave(4)
--> status=(SIMPLE_BUS_WAIT)
20.5 SLV [136]
20.5 top.bus Handle Slave(4)
--> status=(SIMPLE_BUS_OK)
    
```

```

22 top.bus : read(3) @ 3c
22.5 top.bus : request (3) [SIMPLE_BUS_REQUEST]
22.5 top.arbiter :
    R[3](-SIMPLE_BUS_REQUEST@3c) -> R[3] (rule 3)
22.5 top.bus Handle Slave(3)
--> status=(SIMPLE_BUS_OK)
23 top.bus : write(3) @ 3c
23.5 top.bus : request (3) [SIMPLE_BUS_REQUEST]
23.5 top.arbiter :
    R[3](-SIMPLE_BUS_REQUEST@3c) -> R[3] (rule 3)
23.5 top.bus Handle Slave(3)
--> status=(SIMPLE_BUS_OK)
37 top.bus : burst_write(4) @ 4c
37.5 top.bus : request (4) [SIMPLE_BUS_REQUEST]
37.5 top.arbiter :
    R[4](-SIMPLE_BUS_REQUEST@4c) -> R[4] (rule 3)
37.5 top.bus Handle Slave(4)
--> status=(SIMPLE_BUS_OK)
...
...
Press any key to continue
    
```

에 대해서는 [1]을 살펴보기 바란다.

결론

simple_bus 예제를 통하여 SystemC를 이용한 전송 수준 모델링에 대해 살펴보았다. 본 예제는 사이클 상세 수준에서 구현되었으나 이를 낮은 수준의 RTL 또는 핀 상세 수준으로 구체화하고 기존의 테스트벤치를 재사용하여 작동을 확인할 수 있다.

또한, 동일한 외형적 형태의 포트 또는 인터페이스를 가지는 여러 가지 마스터와 슬레이브를 추가하거나, 마스터의 우선 순위 및 중재기의 중재방식 변경, 변수들의 상태 변경 등을 통해 여러 가능성을 모델링 해 볼 수 있다. 이는 모델링 언어인 SystemC의 가장 큰 특징이라 할 수 있다. 다음 시뮬레이션 결과는 simple_bus_test의 초기화 설정 시 master_b와 master_nb의 lock 기능을 모두 false에서 true와 false로 변경했을 때의 결과이다. 이를 통해 중재기에서 마스터를 선택하는 규칙이 rule 3에서 rule 1로 전환됨을 확인한다.

시뮬레이션이 시작되는 0ns에서 각 모듈의 생성 및 초기화가 수행되며, direct_master를 명시적으로 보여주고 있다. 마스터의 요청과 버스의 처리를 살펴보면 0ns : burst_read()와 read() 요청 -> 0.5ns : 높은 우선 순위의 read()가 선택, 수행 -> 1ns : write() 요청 -> 1.5ns : 대기 중인 burst_read()와 write() 중 write() 수행 -> 2.5ns ~ 14.5ns : 매 클럭마다 버스가 fast_mem 에 접근하여 burst_read() 수행 -> 15.5ns ~ 20.5ns : 버스가 1개 클럭 지연을 하는 slow_mem 에 접근하여 burst_read() 연속 수행 후 종료 -> 22ns : 두 번째 read() 요청 -> 22.5ns : read() 수행 -> 23ns : 두 번째 write() 요청 -> 23.5ns : write() 수행 -> 37ns : burst_write() 요청의 진행을 확인할 수 있으며 소스코드의 wait(), wait(m_timeout)에 의한 지연을 조사해보면 정확한 시점에 마스터로부터 요청이 발생함을 확인할 수 있다. 0ns에서의 디폴트 클럭 값 '1'에서 상승예지를 인지하는 것을 확인할 수 있으며, 이는 SystemC의 시뮬레이터가 값의 변화를 0ns 이전에 감시하고 있다고 해석할 수 있다. 일괄전송 요청 마스터가 선택되는 2.5ns에서 SIMPLE_BUS_WAIT로의 상태 변화가 일어남을 확인한다.

생각해볼 문제

SystemC를 이용한 설계의 경우, 모듈은 포트를 통한 인터페이스 함수 호출로 채널에 접근하고, 채널은 인터페이스 함수를 구현하며 포트를 갖지 않는 것이 일반적이며, 이러한 방식에 기초하여 버스를 모델링해볼 수 있다[1].

simple_bus의 버스 모듈은 인터페이스를 구현함과 동시에 포트를 가지고 있으며, 이를 통해 인터페이스가 구현되어 있는 채널인 슬레이브와 중재기에 접근한다. 이 경우 simple_bus는 회로 구현을 더욱 충실히 반영한 모델링이 된다. SystemC에서 sc_channel은 typedef sc_module로 선언되어 있어 모듈과 채널의 언어적 구분은 없으며 개념적 구분을 가질 뿐이다. 그 외의 여러 생각해 볼 문제들

Reference

[1] 국일호 역, "SystemC를 이용한 시스템 설계", 에이콘, (2003)
 [2] 기안도, "SystemC 모델링 언어", IDEC 교재개발 시리즈 35, 대영사, (2005)
 [3] <http://www.accelera.org/downloads/standards/systemc>
 [4] David C. Black, Jack Donovan, Bill Bunton and Anna Keist, "SystemC: From the Ground Up", Springer, (2004)

2012 IDEC Platform Center Workshop

2012 IDEC Platform Center Workshop에 초대합니다!

- 일시 : 2012. 10. 25 (목)
- 장소 : 엘타워 8층 엘하우스홀 (서울 양재동)

더운 여름이 지나가고 아름답고 풍요로운 가을이 성큼 다가오고 있습니다. 반도체설계교육센터(IDECC)와 IDEC 플랫폼 센터도 한 해의 결실을 정리하여 보여드리기 위하여 "2012 IDEC Platform Center Workshop"을 마련하였습니다. IT와 융합 기술의 핵심 영역에서 반도체 산업의 시너지 효과를 극대화하고 정부정책 및 산업체 필요성을 반영하기 위하여 IDEC은 2011년에 IDEC 플랫폼 센터(IPC)를 시작하였습니다. 교육과 연구뿐만 아니라 산학연 교류를 위한 공동 플랫폼을 구축하여, 시스템반도체(SoC), 소프트웨어, 시스템 분야가 상호 연동할 수 있도록 한다는 구체적인 목표를 가지고 IPC는 설립되었으며, 현재 스마트 TV, 지능형 차량용 SoC, 스마트-AP 등 핵심 영역을 중심으로 3개의 IPC가 운영되고 있습니다. IPC의 성과를 정리해서 보여드리고 객관적인 평가와 IPC 사업이 나아갈 방향을 듣고자 이번 워크숍을 마련하였습니다. 바쁘시더라도 꼭 참석하셔서 자리를 빛내주시길 바랍니다.

2012년 10월 5일
반도체설계교육센터
소장 박인철

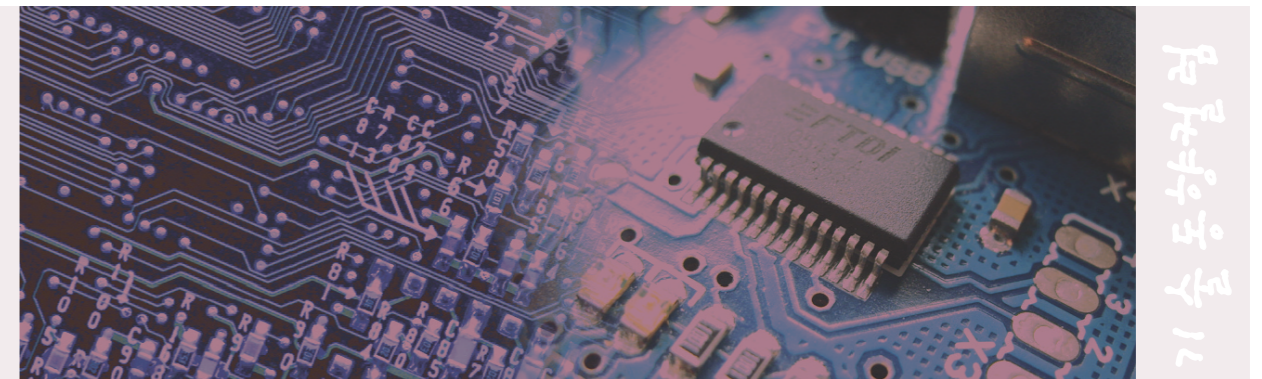
AGENDA

Time	Title
08:50 - 09:20	Registration
09:20 - 09:30	Workshop Opening
09:30 - 10:30	성과발표 (광운대 IPC)
10:30 - 11:00	Talk 1 : 차세대 스마트 TV 기술 및 생태계 특성 (홍진우 : ETRI/ 스마트TV사업 단장)
11:00 - 11:30	Talk 2 : 국내 스마트 TV 산업 경쟁력 제고 및 활성화 방안 (윤명현 : 지식경제부/ 홈네트워크정보기전 PD)
11:30 - 12:00	Talk 3 : 스마트 TV 서비스 동향 및 전망 (강배근 : LG전자/ TV연구소 상무)
12:00 - 13:30	Lunch
13:30 - 14:00	시상식
14:00 - 15:00	성과발표 (경북대 IPC)
15:00 - 15:30	Coffee break
15:30 - 16:30	성과발표 (한양대 IPC)
16:30 - 17:30	Talk 4 : 지능형 차량용 반도체 시장 전망 (김훈태 : HYUNDAI-AUTRON 이사)

생물모방 지능의 뉴로모픽 VLSI와 자동차의 첨단 안전기술 (Intelligent Active Safety Enhancement and neuromorphic VLSI vision)



KAIST 녹색교통대학원
한일송 초빙교수
연구분야 : Analog and mixed-signal CMOS VLSI Neuromorphic VLSI, Intelligent active safety for vehicular application Tunable RF analogue filter application, Electronics for Electric Vehicle
E-mail : i.s.han@kaist.ac.kr
http://gt.kaist.ac.kr



서론

■ 뉴로모픽 연구의 배경과 현황

최근 사람 지능을 모방하고자 활발한 연구가 진행되고 있으며, 이를 통하여 다양한 환경하에서도 안정적인 인식 능력을 구현하고 활용하려 하고 있다.

본론

대표적 연구로는 미국의 SyNAPSE(Systems of Neuromorphic Adaptive Plastic Scalable Electronics)와 EU의 FACETS(Fast Analog Computing with Emergent Transient States) 등이 있으며, 이들 모두 뇌신경계의 현상과 능력을 모방하고자 하는 뉴로모픽(neuromorphic) 기술 연구를 통하여 혁신적 기술 개발을 목표로 하고 있다(그림 1).

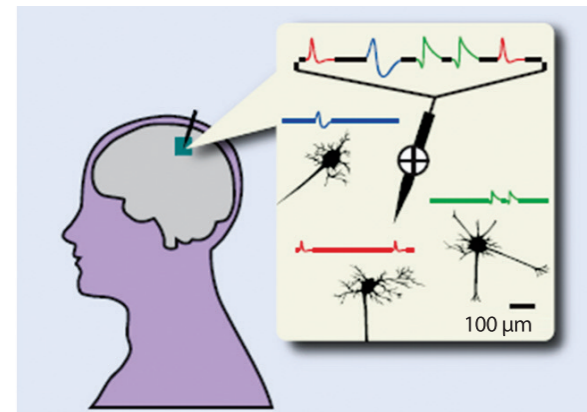


그림 1. 뇌 지능의 원천인 뇌 신경세포와 신호처리

뉴로모픽 연구는 바이오-뉴로 시스템(생체 신경계)의 전압-전류 다이내믹 특성을 아날로그-mixed (아날로그-디지털 회로 혼합형) VLSI로 모방하는 뉴로모픽 신경망을 구현하는 것으로, 궁극적으로 생물체처럼 다양한 환경에 적응할 수 있고 안정적 기능의 지능 모방 구현이나 신경계 기능 혹은 센서 능력의 대체 실현을 목표로 한다.

특히, 야외 환경과 같이 변화가 많고 조명 여건을 제한받아 기존의 컴퓨터 비전 기술을 활용하기에 실용적으로 한계가 있는 분야에서 뉴로모픽 VLSI 기반의 비전 시스템으로 구현의 실용성과 성능의 안정성을 현실화 가능하다. 초기의 신경망 컴퓨터 연구에서 시각적 인

식능력이 주요 관심사였듯이, 자연환경 속에서도 사람과 동물이 지닌 안정적 시각 지능을 실현하려는 뉴로모픽 VLSI비전 시스템도 새로운 뉴로모픽 기술의 대표적 연구 방향의 한가지이다.

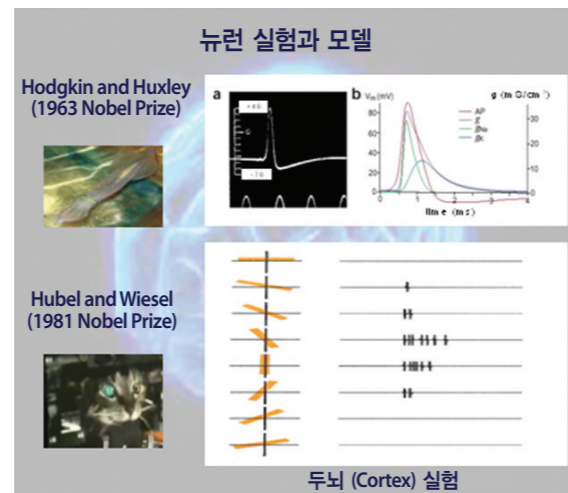


그림 2. 뉴런의 Hodgkin-Huxley formalism과 Hubel과Wiesel의 뇌 Visual Cortex 실험

뉴로모픽 연구 특히 뉴로모픽 비전 연구의 이론적 기반은 그림 2와 같이, Hodgkin-Huxley formalism의 뉴런 신경세포 다이내믹 모델과 Hubel 과 Wiesel의 동물 시각 피질(visual cortex) 신경 세포 실험 결과로서 다양한 뉴로모픽 플랫폼과 프레임워크, 시각 인지 기술들이 활발하게 연구되고 있다.

특히 Hubel과 Wiesel의 실험으로 포유류 동물의 시각 인식 기능이 시각 피질상의 뉴런이 시각 자극에 방향성 반응함을 기반으로 하는 것이 확인됨에 따라, 신경망이나 컴퓨터 비전 분야의 많은 연구와 개발의 기초가 되고 있다. 많은 뉴로모픽 VLSI와 응용 개발 연구가 Hubel과 Wiesel이 실험과 스파이크형 뉴런 신호 중심 연구로 진행되고 있으며, 과거의 신경망 컴퓨터나 지능형 기술 연구에 비교하면 생물의 뇌 신경계를 더욱 근원적으로 모방하고 있다.

VLSI 구현은 아날로그 IC 회로로 신경계 연결고리와 뉴런의 기능을 유사하게 모방하는 방안, 디지털 VLSI로 기본 단위 기능을 모방 구현하는 방안, 소프트웨어 프로그램으로 알고리즘적 원리를 모방하여 구

현하는 등 여러 가지로 연구되고 있다. 이처럼 뉴로모픽은 바이오 뉴로시스템을 구현하거나 활용하는 방안에 따라 여러 가지 형태를 취할 수 있으나, 최근 자동차 분야와 같이 다양한 자연환경 속에서 강한 성능이 주된 관심으로 떠오름에 따라 SyNAPSE 프로젝트의 추진 개념(그림 3)처럼 새로운 뉴로모픽 VLSI 혹은 플랫폼개발이 중심으로 연구되고 있다.

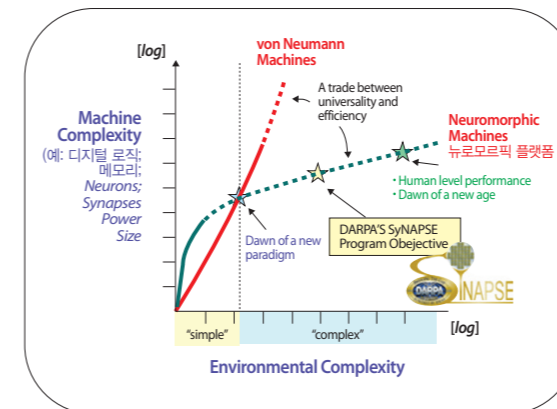


그림 3. 생물모방 뉴로모픽의 패러다임(DARPA)

디지털 VLSI 중심은 SpiNNaker(그림4)와 같이 칩당 18개의 Arm9을 집적화한 여러 개의 chip을 네트워크로 연결하여 구성한 병렬 Processing Element 방식이 연구되고 있고, FACETS 프로젝트의 Heidelberg 대학 연구처럼 아날로그-디지털 혼합형을 기반으로 웨이퍼 스케일로 개발한 아날로그 뉴로 VLSI 방식이 있다.

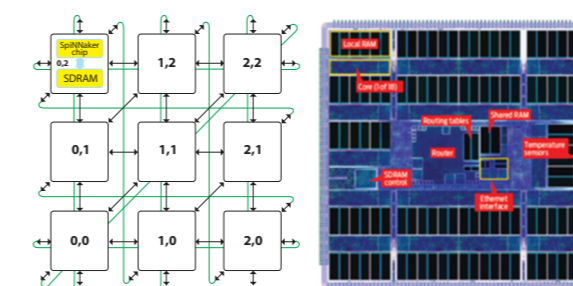


그림 4. SpiNNaker의 소자 VLSI: Arm9 병렬 연결 네트워크

그림 4의 대규모 병렬 연결된 Arm 9로 구성된 디지털 뉴로모픽 플랫폼 SpiNNaker를 이용하여, 그림 2의 Hubel과 Wiesel이 행한 시각피질 피질의 실험결과를 그림 5와 같이 재현하였다. 이처럼 뉴로모픽 플랫폼 혹은 VLSI가 사람이나 생물의 지적 인지 능력을 모방하고 구현하는 것이 최종 목표라면, 현재는 환경에 강인한 시각적 인지 같은 단위 기능을 개발하고 활용하려는 단계이다.

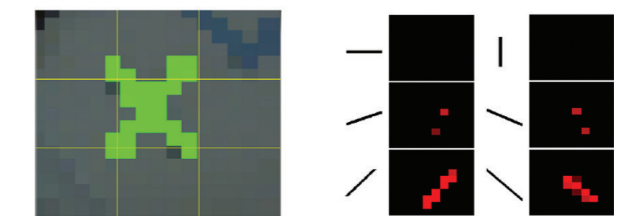
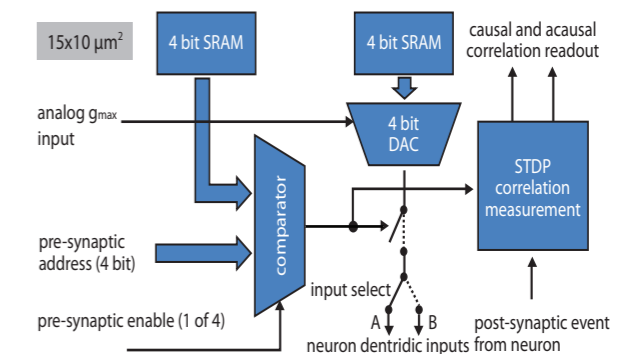


그림 5. SpiNNaker를 이용한 X 형태 입력 영상(좌측), Hubel과 Wiesel 실험 원리를 적용한 방향성 인식

그림 6은 아날로그 뉴로 VLSI회로를 이용하는 FACETS의 뉴로모픽 VLSI/플랫폼의 연구 개발 개념으로, 4bit의 정밀도를 기반으로 하며 SpiNNaker와 같은 Arm9과 네트워크 기반의 병렬 PE형의 뉴로모픽 플랫폼과 차별화되고 있다. DARPA의 SyNAPSE 중 멤리스터 기반의 뉴로모픽 시스템 연구와 아날로그 기반의 구현이라는 점에서 유사하다. 일반적으로 신경망 VLSI 연구나 실시간 최적화, 시각적 인식연구에서, 뉴로모픽 시스템에서 사람이나 생물의 신경세포 모방 구현은 4bit~8bit 정밀도 수준이면 가능한 것으로 알려져 왔다.



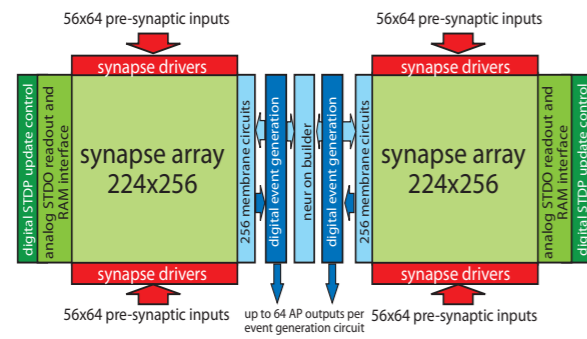


그림 6. 4 Bit 정밀도 기반의 뉴로모픽 VLSI 와 기본 구성 방식

■ 뉴로모픽 연구 사례(KAIST)와 자동차의 첨단 안전 기술
아직도 사람이나 생물의 지능은 미지의 세계로, 현재까지 뇌 연구를 통하여 많은 사실이 밝혀졌음에도, 학습이나 자연환경에서 적응하는 능력을 일상생활에 활용하도록 모방 구현하기에는 모자란 것이 현실이다.

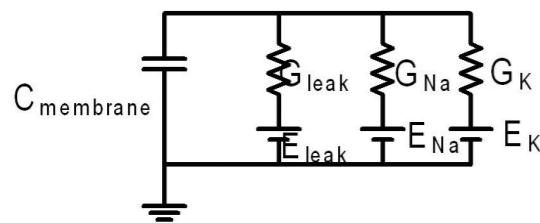


그림 7. 뉴런 모델: Hodgkin-Huxley formalism

70~80년대부터 시작된 인공지능 신경망 연구가 아직 실생활에 폭넓게 응용되지 못하는 한계를 해결하고자 하는 뉴로모픽 연구는 생물체 뇌와 신경계의 지능 현상을 기반으로 하고, 기본 이론인 신경세포의 정확한 다이내믹 모델은 Hodgkin-Huxley formalism(그림7)으로 아래와 같은 실험 이론적 기초를 제시한다.

$$G_{ion} = G_{ionmax} \cdot x$$

$$\frac{dx}{dt} = \alpha(b-x)$$

$$i_{ion} = G_{ion} (V_m - E_{ion})$$

신경 신호 상태(Action potential)에 따라 변화하는 세포막 도전성(conductance) 다이내믹으로 스파이크 신호 발생과 뉴런 세포 간의 연결고리 가중치에 따라 연산하여 지능의 기본적 기능으로 알려졌다. 위의 이론 식에서 V_{in} 은 그림 7의 세포막 전위(즉, $C_{membrane}$ 에 형성되는 전위)이며, G_{ion} 은 Na, K 등 여러 이온으로 말미암은 세포막의 다이내믹하게 변화하는 이온 채널을 의미하며 G_{leak} 은 누설 채널을 뜻한다.

가장 단순한 모델은 누설 채널만으로 LIF(Leaky-Integrator-Fire) 뉴런 수준으로 스파이크 신경 신호를 모델링하고, 심장 장애 치료와

같이 높은 복잡도의 뉴로모픽 뉴런 모델은 필요한 약물의 신경 세포 다이내믹 시뮬레이션을 위하여 20~30개의 이온 채널이 포함되지만, 일반적으로 그림 7과 같이 1~2 이온 채널의 모델이 널리 활용된다.

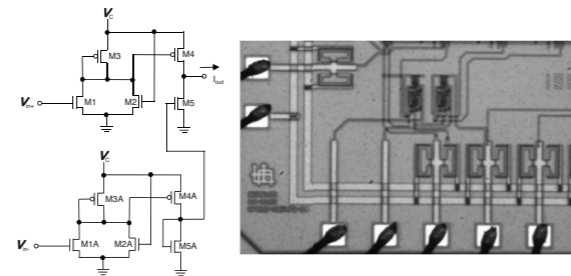


그림 8. 뉴로모픽 뉴런과 연결고리용 컨덕턴스(KAIST)

그림 8의 프로그램 가능한 정밀한 CMOS 컨덕턴스 회로는 Hodgkin-Huxley formalism의 신경 세포막 모델과 신경 연결고리 가중치 연산을 모방 구현 가능한 아날로그-mixed 회로이다.



그림 9. CMOS 컨덕턴스 뉴런을 이용한 Hubel과 Wiesel의 Visual Cortex simple cell의 뉴로모픽 뉴런 뉴로모픽

이와 같은 뉴로모픽 뉴런으로 그림 9와 같이 Hubel 과 Wiesel의 동물 시각 두뇌(visual cortex) simple cell 뉴런 실험 결과와 유사한 기능을 모방 구현할 수 있다. 그림 8과 같이 VLSI의 최소 구성단위인 MOS트랜지스터에서 선형화 트랜스 컨덕턴스를 만들어 뉴로모픽 뉴런을 구현할 수 있고 CMOS VLSI 회를 통한 저전력, 소형화 와 실시간 동작을 하는 안정적 지능을 실생활에 적용할 수 있다.

최근 자동차의 안전 운행 지원에 뉴로모픽 기술을 활용하려고 연구되고 있으며, 이는 자동차 내부나 외부 환경 모두 사람의 안전이 직결되어 환경에 강인한 특성이 있어야 자동차 사용 환경이 장소나 시간에 제약이 없어야 하는 점에서 사람과 동물의 시각적 지능을 모방하고 있다.

그림 8의 뉴로모픽 VLSI용 CMOS 컨덕턴스는 0.5V 다이내믹 영역에서 -60dB 이상의 선형성을 구현하므로, 그림 6의 아날로그-mixed 뉴로모픽 VLSI의 요구 성능을 만족한다. 그림 9의 방향성

반응을 모방하여 환경 변화에 강인성을 구현하여, 두상과 상체의 특성을 이용한 신경망 판별기로 사람을 인식하는 그림 10은 뉴로모픽 시각인지 적용 예로서, 그림 11의 차세대 자동차 사고 구난 시스템에 필요한 탑승자 상황, 내부 안전 장비 상태를 인식할 수 있고, 차량 운전시 교통사고 방지를 위한 보행자 인식, 전방 차량 혹은 장애물을 인식할 수 있다.

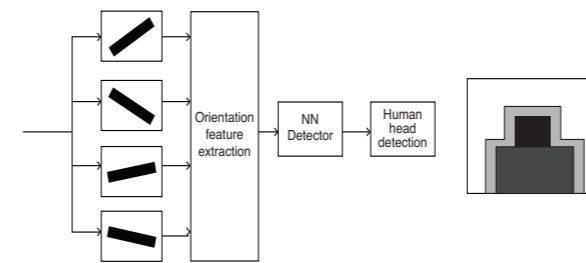


그림 10. 사람 인식을 위한 뇌(visual cortex) 모방의 뉴로모픽 시각인지 모델



그림 11. 사람 인식의 뉴로모픽 시각인지 모델을 이용한 차세대 교통사고 구난(eCall) 시스템

그림 12는 뉴로모픽 비전으로 탑승자 상태인식 과정을 보여주며, Hubel과 Wiesel의 방향성 반응 원리를 이용하여 조명 여건이나 배경의 급격한 변화에서도 강인한 안정적 인식 성능을 보여준다.

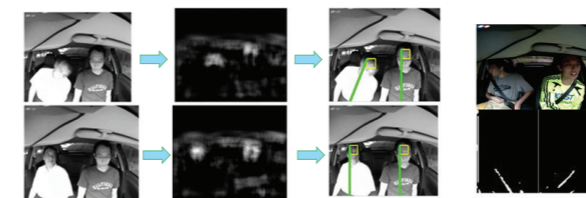


그림 12. 차세대 eCall용 자동차 첨단안전 기술 (차량 사고 대응형 탑승자 상태 감시 뉴로모픽 시스템: occupancy, 자세, 에어백, 안전벨트), 탑승자 머리 부분 인식 (흰색 saliency: 좌측으로부터 2번째), 착용된 안전 벨트만 추출(우측 끝)

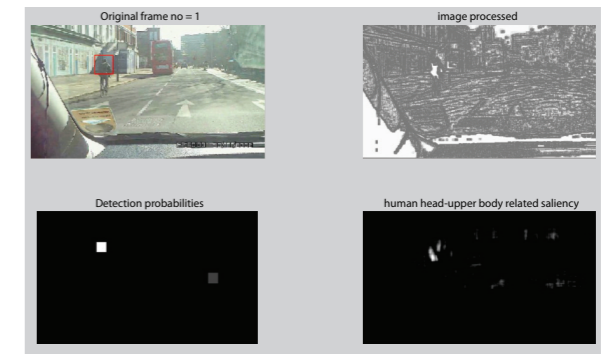


그림 13. 뉴로모픽 인식기의 근접 이륜차 탑승자 인식 (좌측 상단: 입력 영상과 인식된 자전거 탑승자 머리, 우측 상단: 뉴로모픽 방향성 처리된 입력 영상, 우측 하단: 머리 감지의 saliency, 좌측 하단: 인식된 자전거 탑승자 머리 위치)

그림 13의 방향성 반응에 기반을 둔 뉴로모픽 영상 처리 결과는 자전거를 타고 가는 사람의 두상과 상체 인식에도 활용 가능하며, 그림 10의 사람 인식기를 적용하여 운전 중 전방에 사람(자전거, 이륜차 탑승자 포함)이 있음을 인식한다. 이와 같은 인식은 방향성 뉴로모픽 처리를 근간으로 하며, 그림 14와 같이 야간 운전 시 복잡한 배경 속의 사람도 인식한다. 기존의 보행자 인식이 도로를 횡단하는 보행자 인식을 대상으로 발과 도로처럼 경계가 구별되는 것과 같은 상황에 대응하는 것에 비하여, 뉴로모픽 처리는 강인하게 동작한다.



그림 14. 뉴로모픽 비전 기반으로 사람 인식형

아날로그 CMOS 컨덕턴스를 이용한 뉴로모픽 VLSI를 적용하여, 고속, 소형화, 저전력화를 용이하게 구현할 수 있으며, 그림 4와 같은 디지털 뉴로모픽 VLSI를 적용하거나 임베디드 시스템을 프로그램하는 것으로도 뉴로모픽을 구현할 수 있다. 그림 14의 보행자 인식은 저소음 전기자동차의 보행자 안전을 위한 것으로, 차량 전방 주변에 보행자를 인식하여 인공적으로 자동차 소음을 발생시켜 보행자가 위험하지 않도록 하기 위한 것이다. 우측 상단의 적색 상자는 차량 진행 전방에 보행자가 있음을 나타낸다.

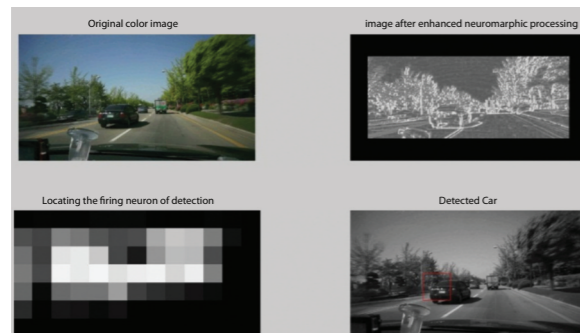


그림 15. 뉴로모픽 비전을 이용한 전방 차량 인식

■ 뉴로모픽 VLSI와 차량용 지능형 안전 기술

그림 15는 신경망 인식기에 사용되는 머리와 상체 템플레이트 대신 차량의 특성 템플레이트를 이용하면, 뉴로모픽 지능을 이용하여 전방의 차량 물체를 인식하여 자동차의 안전 운전 지원에 활용이 가능함을 보여준다.

결론

이와 같은 다양한 물체의 구별이나 상태인식의 뉴로모픽 기술은 차량용 안전 기술을 구현하는 전장장치 부품으로, 완성차 산업에서 요구하는 가격대와 성능의 안정성을 만족하게 하기 위하여 전용 뉴로모픽 VLSI 화가 되어야 한다. 최근 고급형 차량부터 시작되어 준중형 차에도 장착되기 시작한 영상 기반 능동형 안전장치는, 변화하는 다양한 주, 야간 주행 주변 환경에 적응하고 안정적 인식 성능을 구현하는 뉴로모픽 VLSI의 연구와 실용화를 요구한다.

Reference

- [1] D. H. Hubel and T. N. Wiesel, Receptive fields of single neurones in the cat's striate cortex, *J. Physiol.*, 148, 1959, pp 574-591.
- [2] M. Hausser, The Hodgkin-Huxley theory of action potential, *Nature neuroscience supplement*, vol. 3, 2000, pp. 1165
- [3] C. Patterson, F. Galluppi, A. Rast, and S. Furber, Visualising large-scale neural network models in real-time, *Proc. IEEE World Congress on Computational Intelligence*, June, 2012
- [4] W. Han, C. S. Kim and I. S. Han, Pedestrian detection by neuromorphic visual information processing, *Proc. FISITA 2012 (Beijing)*, Nov, 2012 (to be published)
- [5] I. S. Han, "Mixed-signal neuron-synapse implementation for large scale neural networks," *Int. Journal of Neurocomputing*, pp. 1860-1867, 2006.
- [6] W. Han and I. S. Han, "Bio-inspired visual signal processing based on neuromorphic Circuit," *Proc. WSEAS 2010*, pp.131-136, April, 2010



Call for Papers
ISOC 2012, Theme : SoC Design for Smart Living

2012 International SoC Design Conference
November 4-7, 2012 | Ramada Plaza Hotel, Jeju, Korea



International SoC Design Conference (ISOC) aims at providing the world's premier SoC design forum for leading researchers from academia and industries. Prospective authors are invited to submit papers of their original works emphasizing contributions beyond the present state of the art. ISOC 2012 is technically co-sponsored by IEEE CAS Society and accepted papers will be published on IEEE Xplore. We also welcome proposals on special sessions.

Paper Submission

Complete 2-page to 4-page manuscript (in Standard IEEE double-column format) is requested. Papers must be submitted electronically in PDF format. Only electronic submission will be accepted. For more information, please refer to the conference website: <http://www.isoc.org>.

Areas of Interest

- | | |
|--|---------------------------------------|
| Analog and Mixed-Signal Circuits | Communication SoCs |
| Display Driver and Imaging Devices | Embedded Memories |
| Embedded System Software | High Speed Signal Interfaces |
| Low Power Design Techniques | Microprocessor and DSP Architectures |
| Energy-Aware Systems | SoC Design Methodology |
| Multimedia (A/V) SoCs | SoCs for Automotive Technology |
| Wireline & Wireless ICs (RF ICs) | Sensor & MEMS |
| Signal Integrity/Interconnect Modeling | Power Electronics (Energy Harvesting) |
| SoC Testing and Verification | Bio & Medical Devices |

Special Sessions

Proposals are solicited for special sessions. Please submit proposals for special sessions to the special session chair.

Chip Design Contest

Design contest provides the academia with the opportunity to introduce their novel chip designs to the real world. The selected designs will be awarded. Papers should be submitted in electronic form via http://www1.idec.or.kr/conference/conference_isoc.asp.

Best Paper Awards

The authors of selected papers will be awarded for technical contributions and their papers will be invited for publication in the *Journal of Semiconductor Technology and Science (SCIE)* published by Institute of Electronic Engineers of Korea (IEEK). (Visit <http://www.jsts.org> for submission details).

Important Dates

- | | |
|--|---------------|
| • Deadline for submission of special session proposal; | Ju1. 16, 2012 |
| • Acceptance notice of special session proposal; | Jul. 21, 2012 |
| • Deadline for submission of regular session full paper; | Aug. 11, 2012 |
| • Deadline for submission of chip design contest; | Aug. 25, 2012 |
| • Deadline for submission of special session full paper; | Aug. 25, 2012 |
| • Notification of acceptance (all submitted papers); | Sep. 08, 2012 |
| • Deadline for final paper submission; | Sep. 22, 2012 |
| • Deadline for author and early-bird registration; | Sep. 22, 2012 |

At least one author of each accepted paper must register by September 22, 2012.

