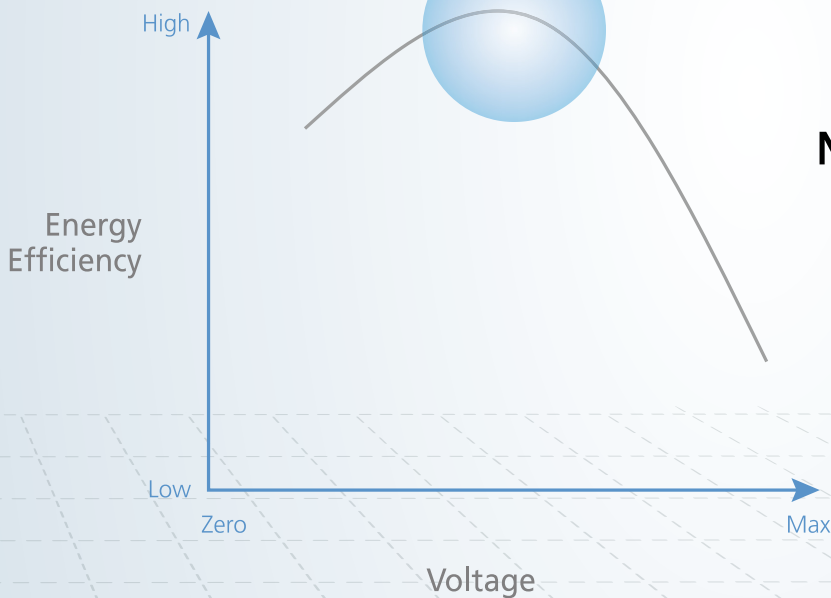


IDEC Newsletter



Near Threshold Voltage(NTV)

Super Threshold Voltage(STV)



NTV 회로 환경에서 발생 가능한
테스트 이슈 및 방법론



MPW 관련 문의

이의숙 책임 (yslee@idec.or.kr, 042-350-4428)

2018년 MPW 공정 및 진행 일정

• 지원 공정 세부 내역

회사	공정 [μm]	공정내역	설계면적 (팀별)	칩수 /1회	모집 횟수	Package 사용가능 pin수(Design)	Package type
삼성	65nm RFCMOS	CMOSRF 1-poly 8-metal	4mm x4mm	40	1	208pin	LQFP/ BGA 208pin
매그나칩/ SK하이닉스	180nm CMOS	CMOS 1-poly 6-metal (6 metal을 Thick metal(TKM)로만 사용 가능) (Optional layer (DNW, HRI, BJT,MIM) 추가)	3.8mm x3.8mm	25	5	200pin	MQFP/ BGA 208pin
	350nm CMOS	CMOS 2-poly 4-metal (Optional layer (DNW, HRI, BJT, CPOLY) 추가)	5mm x4mm	20	2	144pin	

진행 일정 및 공정 내역

• 추가 모집 : MS180-1804회/1803회 매그나칩/SK하이닉스 180nm, MS350-1801회 매그나칩/SK하이닉스 350nm

공정	회차구분 (공정_년도순서)	모집팀수 ((mmxmm)x칩수)	정규모집 신청마감	참여팀수 ((mmxmm)x칩수)	DB마감 (Tape-out)	Die-out	비고
MS 180nm	MS180-1801	(3.8x3.8) x25	2018.01.12	(3.8x3.8)x22 (3.8x1.9)x6	2018.03.19	2018.08.20	칩제작중
	MS180-1802		2018.01.12	(3.8x3.8)x23 (3.8x1.9)x4	2018.05.21	2018.10.22	설계중
	MS180-1803		2018.02.09	(3.8x3.8) x20 (3.8x1.9)x1	2018.07.23	2018.12.24	추가모집
	MS180-1804		2018.04.13	(3.8x3.8) x19 (3.8x1.9)x1	2018.09.17	2019.02.18	추가모집
	MS180-1805		2018.06.08	-	2018.12.03	2019.05.06	모집예정
MS 350nm	MS350-1801	(5x4) x20	2018.02.09	(5x4)x16 (5x2)x2	2018.06.11	2018.10.08	추가모집
	MS350-1802		2018.07.06	-	2019.01.14	2019.05.13	모집예정
삼성 65nm	S65-1801	(4x4) x40	2018.01.12	(4x4)x35	2018.05.07	2018.11.12	설계중
	S65-1802		2018.03.09	(4x4)x39	2018.09.10	2019.03.18	설계중
	S65-1803		2018.07.06	-	2019.01.07	2019.07.19	모집예정

- 일정은 사정에 따라 다소 변경될 수 있음.
- 회차표기 : 공정코드-년도 모집순서 (예시) 삼성 65nm 2018년 1회차 : S65-1801)
- 모집 기간 : 모집 마감일로부터 2주 전부터 접수
- 선정 결과 : 모집 마감 후 2주 후 결정
- NDA 접수, PDK 배포 : 선정 후 2주 이내 완료
- Package 제작은 Die out 이후 1개월 소요됨



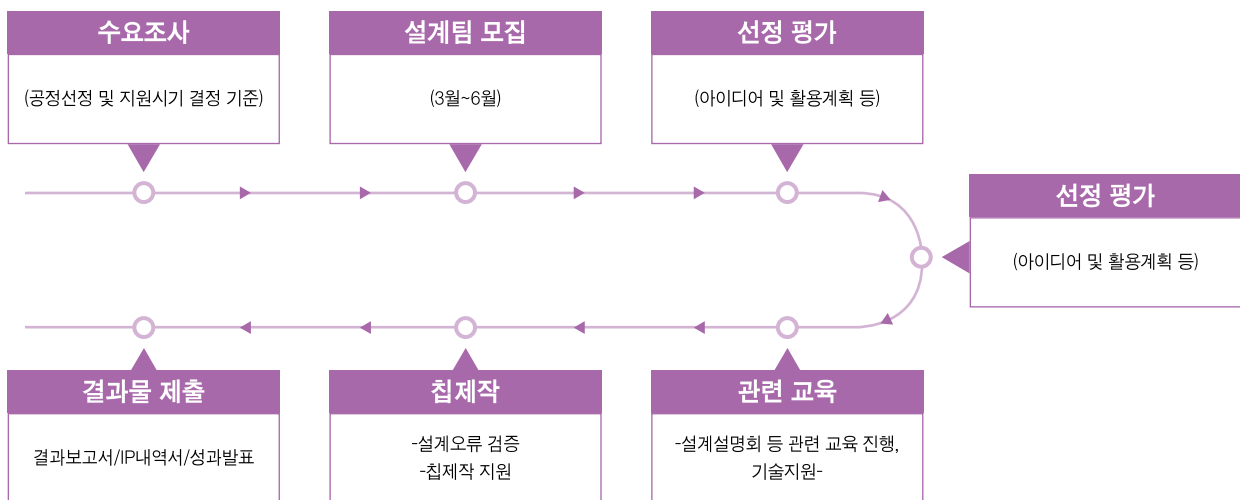
MPW 관련 문의

이의숙 책임 (yslee@idec.or.kr, 042-350-4428)

2018년 국내외 MPW 공정 추가 지원_1차 지원자 마감

IDEC에서는 칩제작 지원 확대를 통해 다양한 분야의 전문설계인력 양상으로 국내 기술력 향상을 도모하기 위해 기존 MPW 외 추가 공정 제작을 지원하게 되었습니다. 지난해 대학에서 추가로 필요한 공정 및 환경에 대해 2회 조사하여 칩제작 지원을 시행 기준을 마련하였습니다. 최종 설계 지원팀 선정은 설계 내용과 활용 계획 등을 기준으로 평가하여 진행될 예정입니다. 관련하여 지원이 필요한 참여 교수께서는 계획하시는 설계를 토대로 지원해 주시기 바랍니다.

· 시행절차



· 지원 대상 IDEC 참여교수

· 설계팀 모집 (1차 지원팀) 사전 신청모집 후 최종 설계계획서 제출 (~04.16, 월)

· 지원자 선정 조건

- 참가 대상 : IDEC 참여교수 중 사전 신청자 대상 우선 지원
- 제출 서류 : 설계계획서 (기존 설계회로설명서 제출 양식 이용)
- 대학의 연구와 교육을 위한 칩 설계 (산업체 과제 참여는 제한함.)
- 실적 제출 관련
 - 사사 문구에 "IDEC 지원"임을 표기
 - IP 내용 소개 (자료 제출)
 - 기존 MPW 참여 의무 이행 (결과보고서 제출, CDC 참여)
 - JICAS 게재 의무

· 지원 공정 4차산업의 필요 기술 개발 분야의 설계로 제작 희망공정 및 지정공정 (BCDMOS)

· 지원 규모 칩제작비의 일부 지원 (설계분야에 따라 차등 지원. 설계팀에 개별 안내)

· 1차 지원팀 선정을 위해 평가 중이며, 해당팀 결정은 4월말 결정하여 지원합니다.

· 2차 지원팀 모집 공고 : 7월초 예정

※ 당해년도 MPW 지원 공정 외 설계가 필요한 팀은 준비하시어 해당 기간에 참여해 주십시오.



CDC 관련 문의

김하늘 주임 (kimsky1230@idec.or.kr, 042-350-8535)

2018 IDEC Congress Chip Design Contest

개최 정보

- 개최 일정 : 2018년 7월 3일 (화)
- 개최 장소 : KAIST 학술문화관 (E9) 정근모컨퍼런스홀, 스카이라운지 (5층)

포스터 관련

2018 IDEC Congress부터 참여팀은 “논문”을 제출하지 않고 “포스터”를 제출합니다.

접수팀은 선착순 100명 마감됩니다.

CDC 마지막 참여 대상자이거나 미참여팀의 경우 반드시 늦지 않게 포스터 제출을 부탁드립니다.

- 포스터 제출 기간 : 4월 9일 (월) ~ 5월 3일 (목)
- 선정 결과 안내 : 5월 21일 (월)
- 포스터 양식 및 제출 : IDEC 홈페이지 참고
- 주의사항 : 동일한 설계로 여러 공정의 회차에 참여한 경우라도 같은 내용의 포스터를 중복하여 제출할 수 없습니다.

참여 대상

MPW 설계팀 외 참여 가능합니다.

비동작 칩의 경우도 반드시 참여해야 합니다.

• 필수 참여 공정

MS350-1602	S65-1602	MS180-1604	MS180-1605
------------	----------	------------	------------

• 선택 참여 가능 공정

S65-1603	MS180-1701	MS350-1701
S65-1701	MS180-1702	S65-1702
MS180-1703	MS180-1704	MS180-1705

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

강좌일정

센터명	강의일자	강의 제목	분류
본센터	5.16-18	Design Compiler 사용법 및 활용예	Tool강좌
	5.23-25	Xilinx ISE를 활용한 FPGA 설계 실습	설계강좌
한양대	5.11	CNN Processor 구조 설계	설계강좌
	5.18	미래 반도체 기술 전망 및 차량용 반도체 SoC	설계강좌



본센터

5/16-18

강좌제목 Design Compiler 사용법 및 활용예

강사 권영기 이사 (Synopsys)

강좌개요

This course covers the ASIC synthesis flow using Design Compiler Graphical -- from reading in an RTL design(Verilog, SystemVerilog and VHDL) to generating a final gate-level netlist. You will learn how to read in your design file(s), specify your libraries and physical data, constrain a complex design for timing and floorplan, apply synthesis techniques using Ultra, compile to achieve timing closure and an acceptable congestion, analyze the synthesis results for timing and congestion, and generate output data that works with downstream layout tools.

수강대상 ASIC digital designers who will be using Design Compiler to synthesize RTL designs to gates

강의수준 초중급 **강의형태** 이론+실습

사전지식·선수과목

1. An understanding of basic digital ASIC design concepts is assumed, including:
 - Combinational and sequential logic functionality
 - Setup and hold timing
2. The ability to work in a Unix/X-windo

5/23-25

강좌제목 Xilinx ISE를 활용한 FPGA 설계 실습

강사 김민석 책임 ((주)리버트론)

강좌개요

Xilinx ISE SW 이해를 기반으로 FPGA의 Architecture를 이해 실습을 하며, HDL 설계 실습을 통한 로직 설계 기반으로 디자인을 FPGA에 다운로드 하여 로직 검증 및 디버깅을 한다.

수강대상 ISE를 사용해서 Xilinx FPGA를 사용해야 하는 담당자 및 엔지니어

강의수준 초급 **강의형태** 이론+실습

사전지식·선수과목 HDL(VHDL, Verilog) 기본 지식

문의 | 본센터 IDEC 김영지 (042-350-8536, yjkim@idec.or.kr)



한양대

5/11

강좌제목 CNN Processor 구조 설계

강사 이광엽 교수 (서경대학교)

강좌개요

CNN 가속기 Processor 설계와 GPGPU 설계에 대해 알아본다.

수강대상 학생/일반인

강의수준 초중급 **강의형태** 이론강의

5/18

강좌제목 미래 반도체 기술 전망 및 차량용 반도체 SoC

강사 공정택 교수 (성균관대학교)

강좌개요

일반적인 SoC(System on Chip) 설계 방법론과 자동차용 SoC 설계 방법론을 정리하고, 미래 반도체 기술 전망 동향과 대응 방안을 논한다.

수강대상 학생/일반인

강의수준 초중급 **강의형태** 이론강의

문의 | 한양대 IDEC 박남선 (031-400-4079, ipc@hanyang.ac.kr)



최기영 교수 | 서울대학교 전기컴퓨터공학부

1. 서론

최초로 설계된 컴퓨터로는 보통 1830년경 Charles Babbage가 설계한 Analytical Engine이라는 mechanical computer를 든다. 그러나 그것은 설계만 되었지 완전히 구현된 것은 아니었다. 그로부터 100여년이 지난 제2차 세계대전 중에 비로소 실제 구현되어 동작하는 컴퓨터가 나타나기 시작했다. John von Neumann이 프로그램을 저장하여 동작시킬 수 있는 기본적인 컴퓨터 구조를 발표한 것도 그 때였으며¹, 그것이 아직도 폰노이만 구조(Harvard 구조와 대비시켜 Princeton 구조라고도 함)라는 이름으로 사용되고 있다. 그러나 그 구조(Harvard 구조를 포함하는 광의의 폰노이만 구조)는 기본적으로 메모리와 CPU 사이의 병목현상으로 더 이상 성능 향상이 어려워져서, 요즘에는 메모리 내에 processing element를 넣어 계산하거나 아예 메모리 자체를 계산 소자로 사용하는 등 다른 구조에 대한 연구가 활발히 이루어지고 있다^{2,3}. 이는 최근에 집중적으로 관심을 받고 있는 신경망회로와 관련해서 매우 중요하게 여겨지는 분야이기도 하다.

한편 메모리를 떠나 계산 자체만 보면, 대부분 기존의 이진 계산에 의존하고 있는데 여기에 사용되는 곱셈기나 덧셈기 등은 정확한 계산을 하는 반면에 복잡하고 오류에 취약하다는 단점을 갖고 있다. 흥미롭게도 이러한 문제를 확률적인 연산 회로를 이용하여 해결하고자 하는 아이디어를 1956년에 처음 발표한 사람도 von Neumann이었다⁴. 그리고 나서 1967년에는 이 개념을 발전시켜 stochastic computing(SC)으로 확장한 논문이 다수 발표되었다^{5,6,7}.

SC는 여러 가지 단점이 있어서 그동안 그다지 많이 연구되고, 활용되어 오지는 않았다. 그러나 나름대로 장점도 많고 특정 분야나 특별한 조건 아래서는 기존의 이진 계산에 비해 월등히 우수하기 때문에 그 잠재적 효용성을 무시할 수 없다^{8,9}. 본 고에서는 SC의 기본 개념을 설명하고, SC 회로를 합성하는 방법과 신경망을 SC 회로로 구현하는 방법 등에 대해 필자가 연구한 내용을 간단히 설명하고자 한다.

2. Stochastic Computing의 개념

[그림 1]은 동물의 신경세포 연결 모형과 신경세포 사이에 전달되는 전기신호를 보여준다¹⁰. 일반적으로 계속해서 spike가 발생하면 강한 신호, 그렇지 않으면 약한 신호로 간주된다. 다시 말하면 spike가 발생할 확률을 신호의 세기로 볼 수 있다. [그림 1] (b)에서 spike가 발생할 수 있는 time slot의 수를 18개라고 보고, 그 중 9개의 slot에 spike가 발생했으니 신호의 세기를 9/18로 간주할 수 있다. [그림 1] (c)의 경우는 신호의 세기를 8/18로 간주할 수 있다. 이와 마찬가지로, SC에서는 주어진 신호에서 임의의 time slot에서의 값이 1이 될(spike 발생에 해당) 확률이 그 신호의 세기가 된다.

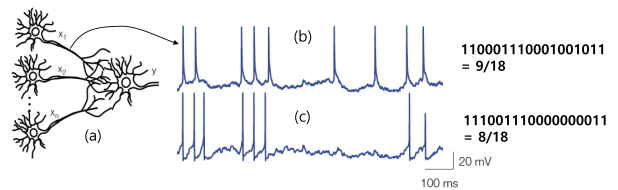


그림 1. (a) 신경세포 연결 모형 (b) 전기신호의 예 1 (c) 전기신호의 예 2 [10].

[그림 2]와 같이 0과 1 사이의 값으로 정규화 되어있는 두 입력 신호의 세기가 각각 6/8과 4/8라고 할 때 이 두 값을 곱하는 연산을 생각해보자. 이진 곱셈기에서는 [그림 2] (a)와 같이 두 수를 곱해서 3/8을 얻게 된다. 두 입력 신호의 세기를 1이 될 확률로 표현하기 위해 [그림 2] (b)와 같이 bit-stream을 사용할 수 있다. 예를 들면 6/8은 8bit 중에서 무작위로 선택한 6개 bit를 1로 만들어 표현할 수 있다. 여기에서 두 입력 A, B가 1이 될 확률은 각각 6/8과 4/8이고 두 입력이 서로 독립적이라고 하면 동시에 둘 다 1이 될 확률은 $6/8 \times 4/8 = 3/8$ 이 된다. 즉, 두 입력이 모두 1일 때 [그림 2] (b)와 같이 AND gate를 이용하여 출력을 1로 만들면

출력 신호의 값은 확률적으로 3/8이 되어 곱셈이 수행되는 셈이다. 즉 두 값을 곱하는 대신 두 값에 해당하는 bit-stream(stochastic number)을 만들면 AND gate로 값싸고 쉽게 곱셈을 할 수 있다. 이와 같이 연산의 대상이 되는 값들을 stochastic number로 표현하고 그 영역에서 연산을 수행하는 것을 stochastic computing이라고 한다.

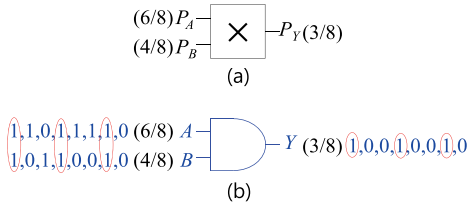


그림 2. (a) 기존의 이진 곱셈기와 (b) SC 곱셈기.

이다. 따라서 주어진 수식을 인수분해 할 때 (1-P)의 형태의 인수를 찾아서 그 수식을 나누고 그 때 얻어지는 몫과 나머지를 재귀적으로 계속 나누게 된다. 그렇게 인수 분해가 완료되면 [그림 4]의 논리 소자들 [그림 3]과 같이 서로 연결함으로써 주어진 수식을 계산하는 회로를 구현할 수 있다.

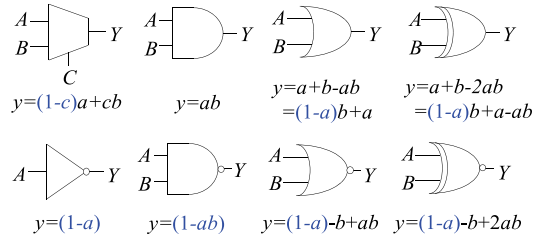


그림 4. 논리 소자와 해당 SC 연산.

3. Stochastic Computing 회로의 합성

앞에서 언급한 바와 같이 2-input AND gate는 SC에서 곱셈 연산을 한다. 또한 2-input OR gate는 두 입력 $a=P(A)$, $b=P(B)$ 에 대해서 $a+b-ab$ 의 연산을 하고, 2-way multiplexor(MUX)는 두 입력 a , b 와 select 입력 c 에 대해서 $(1-c)a+cb$ 와 같은 연산을 수행할 수 있다. 만일 $y=abd+abe+cd-abcd-abde=(1-ab)cd+ab(d+e-de)$ 와 같은 복잡한 연산을 수행하기 위해서는 [그림 3]과 같이 세 개의 gate와 한 개의 MUX만 있으면 된다.

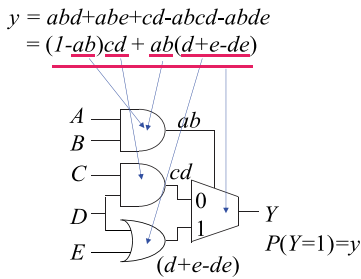


그림 3. 논리 소자를 이용한 SC 구현.

여기에서 당연히 제기되는 문제는 '주어진 임의의 수식에 대해서 이를 구현하는 SC 회로를 어떻게 합성할 것인가?'이다. [그림 3]에서 수식 y 가 주어지면 이를 적당히 인수분해해서 AND, OR, MUX로 대응시키면 되지만 최적의 해를 구하기도 어렵고, 경우에 따라서는 해가 없을 수도 있다. 이에 대해서는 이미 여러 개의 논문들이 발표되어 있는데^{[11][12]}, 여기에서는 필자의 연구실에서 개발된 방법을 소개하고자 한다^[13].

[그림 4]는 각 논리 소자들이 구현할 수 있는 수식을 보이고 있는데 AND gate를 제외하고는 모두(1-P) 형태의 인수를 갖고 있음을 알 수

실제로는 주어진 수식에 (1-P) 형태를 가진 여러가지 다른 인수가 존재할 수 있으므로 어떤 인수를 어떤 순서로 나누는가에 따라서 다양한 인수분해가 될 수 있고 따라서 다양한 회로 합성이 가능하다. 그러므로 그 중에서 가장 적은 비용을 가지면서 정확도가 가장 높은 회로를 선택하는 것이 중요하다. 또한 [그림 4]에서 보듯이 곱셈은 AND gate로 쉽게 할 수 있으나 덧셈을 잘 할 수 있는 논리 소자는 없다. 따라서 [그림 4]에 있는 수식들의 조합으로 표현되지 않으면서 덧셈을 필요로 하는 경우에는 MUX를 사용한다. 예를 들어 $a+b$ 를 계산하려면 [그림 4]에 있는 것과 같은 MUX를 사용하고 c 입력으로 0.5에 해당하는 값을 주면 MUX 출력은 $0.5(a+b)$ 가 되어 0.5배로 스케일 된 덧셈이 된다. 이러한 방법은 $c=0.5$ 를 추가로 생성해야 하는 문제도 있고, 스케일 되는 문제도 있어서 어쩔 수 없을 때에만 사용해야 한다.

SC는 덧셈이 잘 안 되기 때문에 벡터의 내적을 구하는 것과 같이 덧셈이 많이 필요한 계산에 적용하면 그 장점을 살리기 어렵게 된다. 반면에 다음과 같이 정육면체의 가운데 값을 구하는 tri-linear interpolation 연산(volume rendering에 사용)에는 SC가 매우 효과적으로 적용된다.

$$q = xyzv_1 + xyzv_2 + xyzv_4 + xyzv_7 + xyv_0 + xyv_3 + xv_0 + xv_5 + xv_1 + yzv_0 + yzv_6 + yv_2 + zv_4 + v_0 - xyzv_0 - xyzv_3 - xyzv_5 - xyzv_6 - xyv_1 - xyv_2 - xv_1 - xv_5 - xv_0 - yzv_2 - yzv_4 - yv_0 - zv_0$$

실제로 위 수식은 많은 곱셈과 덧셈을 필요로 하는 복잡한 연산임에도 불구하고 [그림 5]와 같이 7개의 MUX로 간단하게 구현된다.

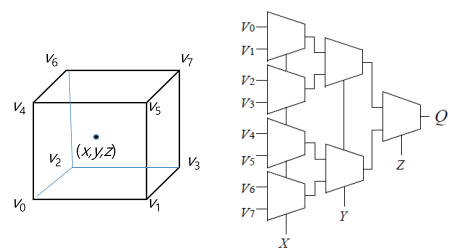


그림 5. Tri-linear interpolation의 SC 구현.

SC는 덧셈을 잘 구현하지 못한다는 것 외에도 bit-stream으로 계산해야 하기 때문에 이진 계산과 비슷한 정밀도를 가지려면 지수함수적으로 bit 수가 증가하게 된다는 단점이 있다. 따라서 높은 정밀도를 요구하는 응용에는 적절하지 않다. 필자의 경험에 의하면 대강 10bit(1024 길이의 bit-stream)을 상회하는 정밀도를 필요로 하는 경우에는 기존의 이진 계산을 사용하는 것이 유리할 것으로 판단된다.

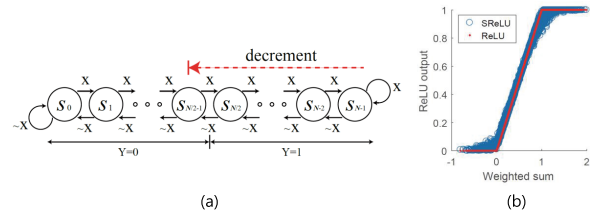


그림 7. ReLU의 SC 구현 (a) FSM (b) 전달함수 비교.

4. Stochastic Computing을 이용한 신경망 구현

인공 신경망은 일반적으로 매트릭스, 벡터 등의 곱이 많이 사용되기 때문에 위에서 언급한 것처럼 벡터의 내적을 효율적으로 수행하지 못하는 SC로서는 좋은 적용 대상이 아니다. 그러나 인공 신경망은 정확한 계산을 필요로 하지 않기 때문에 정밀도를 낮추어 계산하면 SC로 구현해 볼 만도 하다. 요즘 많이 사용되고 있는 심층 신경망은 여러 가지 모델이 사용되고 있으나 여기에서는 CNN(convolutional neural network)을 구현하는 방법을 소개하고자 한다¹⁴⁾.

CNN은 보통 convolutional layer나 fully connected layer에서 사용되는 MAC(multiply and accumulate) 연산, pooling layer에서 maximum을 찾는 MAX 연산, ReLU(rectified linear unit)와 같은 non-linear 연산 등으로 구성된다. 그러므로 이들을 SC로 잘 구현할 수 있으면 좋은 신경망을 구현할 수 있게 된다. MAC 연산은 기본적으로 AND gate를 이용한 곱셈과 곱셈 결과를 더하는 연산을 하게 되는데 후자의 경우 [그림 6] (a)와 같이 병렬 카운터를 사용하는 것도 좋은 방법이다.

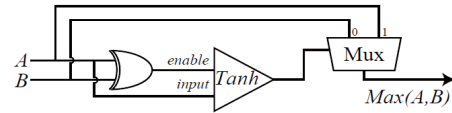


그림 8. Max unit의 SC 구현.

유의해서 설계해야 한다. 다행히 신경망에서 서로 다른 뉴런에서는 같은 난수 발생기를 공유해서 사용해도 큰 문제가 생기지 않는다. 그러나 한 뉴런 내에서 같은 난수 발생기를 여러 군데에 사용하면 신호들 사이의 상관관계(correlation)이 높아져서 정확도가 많이 떨어지니 조심해야 한다. 이밖에도 여러 가지 최적화 기술을 적용하면 더욱 좋은 결과를 얻을 수 있다. 자세한 내용은¹⁵⁾를 참고하기 바란다.

5. 결론

Stochastic computing은 영상 처리와 같은 특정 응용이나 낮은 정밀도를 사용해도 되는 시스템 구현에는 매우 효과적으로 사용될 수 있는 방법이다. 그러나 그렇지 않은 경우에는 덧셈의 비효율성, bit-stream 길이, 난수 발생기의 비용 등의 문제가 있어서 아직 많은 연구를 필요로 하는 방법이기도 하다. 한편 SC 회로를 합성하는 것도 여러 가지 어려움 중 하나이지만 여기에서 소개한 바와 같이 그 해법을 찾는 것도 가능함을 보았다. 또한 SC를 이용한 CNN의 구현도 기본적인 어려움이 있음에도 불구하고 다양한 기술을 고안함으로써 보다 효율성을 높일 수 있음을 알 수 있다.

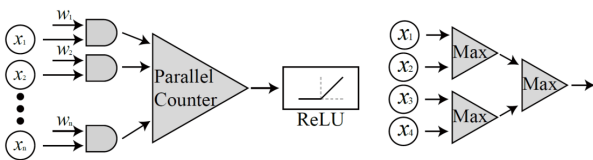


그림 6. CNN의 SC 구현 (a) MAC 연산과 ReLU 연산 (b) MAX 연산.

sReLU(stochastic ReLU)는 [그림 7] (a)와 같이 FSM(finite state machine)으로 구현하고, 마치 스파이킹 신경망의 IF(integrate and fire) 모델처럼 동작시킨다. 즉, 병렬 카운터의 출력 값만큼 상태를 오른쪽으로 천이시키고, 그 결과가 가운데 상태(SN/2-1)를 넘어가는 경우에는 1을 출력한 후(Y=1) 총 state 수의 반(threshold)만큼 왼쪽으로 천이시킨다. 이를 반복하면 출력은 [그림 7] (b)와 같이(saturated) ReLU의 출력과 유사하게 된다. 또한 [그림 6] (b)에서 사용된 Max unit은 [그림 8]과 같은 회로로 구현할 수 있다.

SC의 또 다른 문제는 난수를 많이 생성해야 하고 이로 인한 비용과 에너지 소비가 크다는 것이다. 따라서 난수 발생기의 수를 줄이는 것이 매우 중요하다. 그런데 이를 위해 난수 발생기를 공유해서 사용하면 신호들이 서로 독립적이지 못해서 정확도가 떨어지게 되므로 이 또한

참고문헌

- ① J. von Neumann, First draft of a report on the EDVAC, 1945.
- ② J. Ahn et al., "A scalable processing-in-memory accelerator for parallel graph processing," ISCA, 2015.
- ③ D. S. Jeong et al., "Memristors for energy-efficient new computing paradigms," Adv. Electron. Mater., Sep. 2016.
- ④ J. von Neumann, "Probabilistic logics and the synthesis of reliable organisms from unreliable components," in Automata Studies, C. E. Shannon and J. McCarthy, Ed., Princeton Univ. Press, 1956.
- ⑤ B. R. Gaines, "Stochastic computing," Spring Joint Computer Conference, 1967.
- ⑥ W. J. Poppelbaum et al., "Stochastic computing elements and systems," Fall Joint Computer Conference, 1967.
- ⑦ S. T. Ribeiro, "Random-pulse machines," IEEE Trans. Electronic Computers, June 1967.
- ⑧ A. Alaghi et al., "Stochastic circuits for real-time image-processing applications," DAC, 2013.
- ⑨ M. H. Najafi and M. E. Salehi, "A fast fault-tolerant architecture for Sauvola local image thresholding algorithm using stochastic computing," IEEE Trans. VLSI Syst., Feb. 2016.
- ⑩ J. P. Hayes, "Introduction to stochastic computing and its challenges," DAC, 2015.
- ⑪ A. Alaghi, et al., "A spectral transform approach to stochastic circuits," ICCD, 2012.
- ⑫ Z. Zhao, et al., "A General Design of Stochastic Circuit and Its Synthesis," DATE, 2015.
- ⑬ K. Kim and K. Choi, "Synthesis of multi-variate stochastic computing circuits," VLSI-SoC, 2017.
- ⑭ J. Yu et al., "Accurate and efficient stochastic computing hardware for convolutional neural networks," ICCD, 2017.

저자정보



최기영 교수

소속
서울대학교 전기컴퓨터공학부

주 연구분야
EDA, computer architecture, machine learning

E-mail kchoi@snu.ac.kr

Homepage <http://dal.snu.ac.kr>

Near-Threshold Voltage 회로 환경에서 발생 가능한 테스트 이슈 및 방법론

오형교 박사과정, 강성호 교수 | 연세대학교 전기전자공학과

1. 서론

최근 스마트 기기의 발전에 따라 차세대 기술로 각광받고 있는 사물 인터넷과 스마트 웨어러블 컴퓨팅 기술은 일반적으로 센서로부터 지속적인 모니터링이 요구되기 때문에 오랜 시간동안 동작하기 위해서는 회로의 전력 효율성이 매우 중요시 된다. 따라서 이러한 디바이스에 내장되어 있는 핵심 기술인 스마트 센서 system-on-chip(SoC)의 전력 효율성을 높이기 위한 초저전압 기술이 이슈가 되고 있다. 일반적으로 동적 전력 소모는 동작 전압의 제곱에 비례하기 때문에 동작 전압을 감소시킴으로써 전력 효율성을 높일 수 있다. 하지만 트랜지스터의 문턱 전압(V_{th})보다 낮은 동작 전압에서는 지연시간이 크게 증가함에 따라 누설전류(leakage current)에 의한 전력 소모가 차지하는 비중이 동적 전력 소모보다 더욱 커지게 되어 전력 효율성이 나빠진다. 최근 업계와 학계에서의 연구 내용에 따르면, 디지털 기반의 회로는 near-threshold voltage(NTV) 영역에서 가장 높은 전력 효율성을 가진다고 하며¹⁻³ 이를 위해 최근, 회로의 동작 전압을 기존의 super threshold voltage(STV) 영역에서 트랜지스터의 문턱 전압 근처까지 낮추어주는 NTV 회로 설계 기법이 연구되고 있다^{4,5}.

하지만 에너지 효율을 위해 회로의 동작 전압을 NTV 영역에서 동작할 경우, 실제 동작 전압이 문턱 전압과 차이가 나지 않기 때문에 STV 영역에서 동작하던 회로에서는 발생하지 않던 고장이 발생하게 되거나 발열 등으로 인한 신뢰성 문제가 함께 야기될 수 있다. 이러한 경우, NTV 동작 회로 특성 때문에 기존의 STV 회로에서 적용되던 테스트 방법들로는 해결되지 않을 수 있으며 이는 반도체 양산 과정의 수율 문제와 직결되어 기업 입장에서 비용적 측면에서 많은 손실을 야기할 수 있다. 또한, 매우 높은 수준의 신뢰성을 요구하는 애플리케이션(의료, 차량용 반도체 등)에서 이러한 문제가 발생될 경우 인명 피해 등 심각한 문제로 연결될 수 있다. 따라서 NTV 회로를 설계하고 이를 테스트 할 때 NTV 환경에서 특히 발생 가능한 여러 테스트 이슈들이 연구되고 있으며 기존의 STV 회로 테스트 시에 고려되던 테스트 방법들과는 다른 NTV 회로의 테스트만을 위한 여러 테스트 방법론들이 연구되고 있다.

2. NTV 환경에서의 합선 고장을 고려한 테스트 방법론

2-1. 합선 저항 분석을 통한 NTV 환경에서의 합선 고장 발생 문제 분석

합선 고장(bridge fault)이란 공정 상의 문제로 두 도선 사이에 이물질 등으로 인해 발생한 결함이 회로 내 두 도선이 합선된 것처럼 관측되는 고장이다. 일반적으로 합선 고장은 두 노드 사이의 합선 저항으로 모델링 될 수 있다⁶. 두 도선이 합선되었다면 합선 저항의 크기는 0일 것이며 이상적인 경우라면 합선 저항은 무한대로 모델링이 될 것이다. 그러나 실제 상황에서는 합선 저항의 크기에 따라 합선된 결함이 고장으로 발현될 수도, 잠재적 결함으로 남아 고장으로 발현되지 않을 수도 있다. 정상적인 동작을 위해서는 합선 저항의 크기가 어느 정도 커야 하는데, 합선 저항의 값이 생겨도 정상적인 회로 동작으로 고려할 수 있는 최소 저항을 임계 저항(critical resistance)이라고 한다. <그림 1>은 두 개의 NAND 게이트 출력에 합선 고장이 발생했을 때를 가정하여 게이트 레벨, 트랜지스터 레벨로 모델링 하는 과정을 보여준다. 결함으로 인해 합선 저항이 생겼을 경우 일반적으로 다른 노드에 영향을 주는 노드를 aggressor, 영향을 받는 노드를 victim이라고 한다.

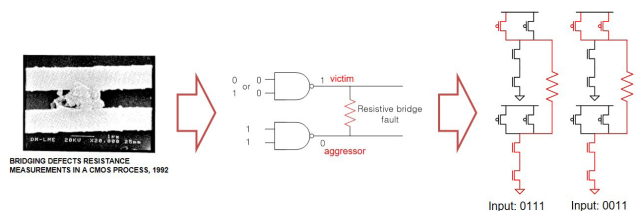


그림 1. 합선 고장 모델링 과정 (게이트 레벨 모델링, 트랜지스터 레벨 모델링)

NTV 환경에서는 동작 전압이 문턱 전압 근처까지 내려감에 따라 트랜지스터에 가해지는 게이트-소스 전압이 문턱 전압 값과 비슷해지게 되고 두 노드 사이에 발생 가능한 임계 저항 값이 상승하게 된다. 이를 검증하기 위해 <그림 1>에서 두 개의 NAND 게이트 사이에 모델링 된 합선 고장을 HSPICE를 이용하여 시뮬레이션을 진행했다. <그림 2>는 해당 시뮬레이션에 대한 결과를 나타낸다. 시뮬레이션은 삼성 65nm 공정 라이브러리를 사용했으며 STV의 동작 전압을 0.7V, NTV의 동작 전압을 0.5V로 설정하여 aggressor에 들어가는 패턴을 0, victim에 들어가는 패턴을 1로 들어가게끔 패턴을 다양하게 적용했을 때 두 노드 사이의 합선 저항 값에 따른 전압 값을 측정했다. 이러한 방식으로 STV와 NTV 각각의 환경에서 정상 동작을 가능하게 하는 임계 저항 값을 측정한 결과, 이전의 STV 환경 대비 NTV 환경에서 최대 26배까지 임계 저항이 상승하는 것을 볼 수 있었다. 이는 실제 두 노드 사이에 결함이 발생하였을 때 STV 환경에서는 결함이 발생하더라도 합선 고장으로 발현되지 않았을 결함이 NTV 환경에서는 합선 고장으로 발현될 수 있음을 의미한다. 따라서 NTV 환경에서는 STV 환경보다 더욱 합선 고장에 대한 테스트를 세밀하게 진행해야 할 필요성이 있다.

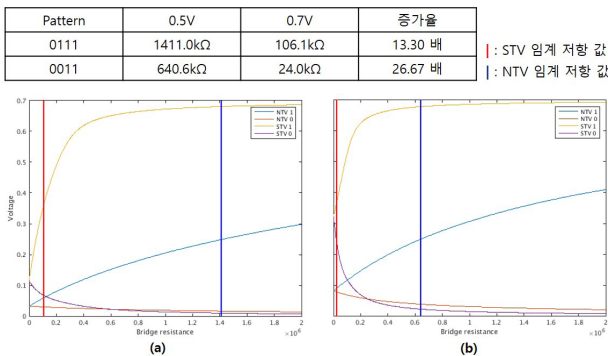
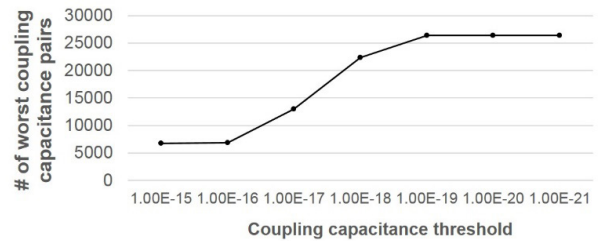


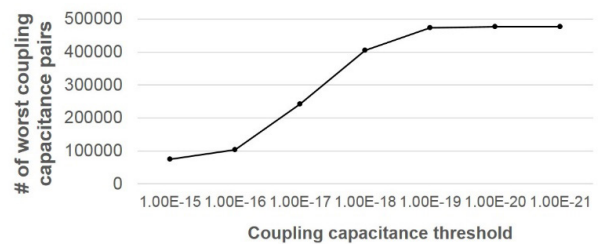
그림 2. HSPICE를 이용하여 2NAND 게이트 사이 합선 고장의 임계 저항 측정 (a) 0111 패턴 (b) 0011 패턴

기존의 STV 환경에서 정해 놓은 기준보다 더 낮은 수준의 임계 값까지 고려하여 합선 고장 리스트를 생성할 필요가 있다.

기존의 지연 고장(delay fault)을 위한 테스트 패턴은 일반적으로 일정 이상의 지연 고장과 상당수의 고착 고장(stuck-at fault), 합선 고장을 잡아낼 수 있기 때문에 테스트 패턴 생성 시 가장 먼저 생성하는 테스트 패턴이다. 하지만 NTV 환경에서 합선 고장이 증가함에 따라 지연 고장을 위한 테스트 패턴만으로는 잡을 수 없는 합선 고장의 수가 점점 증가하게 된다. 또한 기존의 고착 고장을 위한 테스트 패턴으로도 증가하는 합선 고장을 잡는 데에 한계가 발생하게 된다. 이러한 문제점을 해결하기 위해 기존의 고착 고장과 지연 고장의 검출율을 유지 하면서 NTV 환경에서 새롭게 발생할 수 있는 합선 고장을 고려한 새로운 테스트 패턴 생성 방법이 연구된 바 있다. <그림 4>는 기존의



(a)



(b)

그림 3. 임계 값에 따른 worst coupling capacitance pair의 수 (a) ISCAS c7552 (b) ISCAS s38584

2-2. NTV 환경에서의 합선 고장 검출 향상을 위한 테스트 패턴 생성 방법

합선 고장을 실제 환경과 유사하게 테스트하기 위해서는 회로에 place and route를 진행한 레이아웃 정보를 이용하여 각 노드 간 coupling capacitance 값을 추출하고 이 중 임계 값을 넘는 값을 지닌 회로 내 노드들을 합선 고장 리스트로 생성하여 고장 시뮬레이션 진행 및 테스트 패턴을 생성하는 과정을 거쳐야한다. <그림 3>은 ISCAS 벤치마크 회로의 임계 값에 따른 worst coupling capacitance를 갖는 노드의 수를 계산한 그래프이며 이 노드들은 차후 고장 시뮬레이션 진행 시 합선 고장을 일으킬 수 있는 가능성이 있는 노드들로 정해진다. 본 시뮬레이션은 Synopsys 사의 SAED 32nm 라이브러리와 Design compiler, IC compiler, STAR RC 툴을 이용하여 합성, place and route, RC 값 추출 과정을 거쳐 진행되었다. 앞서 말한대로 STV 환경에서는 합선 고장으로 발현되지 않았을 노드들이 NTV 환경에서는 합선 고장으로 발현될 수 있기 때문에 합선 고장 시뮬레이션을 진행하기 위해서는

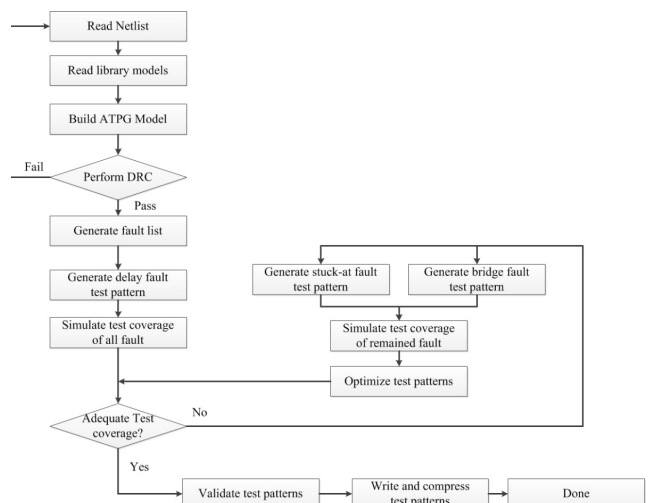


그림 4. NTV 환경에서 합선 고장을 고려하는 테스트 패턴 생성 방법 과정

테스트 패턴을 생성하는 방법에서 NTV 환경에서 합선 고장 검출을 향상할 수 있도록 하는 테스트 패턴 생성 과정이 추가된 테스트 패턴 생성 방법이다. 해당 방법에서는 기존의 테스트 패턴 생성 방법과 동일하게 지연 고장을 위한 테스트 패턴을 생성하여 해당 패턴으로 걸러낼 수 있는 고착 고장과 합선 고장 리스트를 제외한다. 이후 남아있는 고착 고장과 합선 고장을 위한 테스트 패턴을 모두 생성한 후, 각각의 테스트 패턴으로 남아있는 고장에 대한 테스트를 진행한 뒤 목표 테스트 커버리지에 맞는 최적의 전체 테스트 패턴 수를 유지할 수 있는 테스트 패턴을 생성해낸다. 이러한 방법으로 기존 방법 대비 고착 고장과 지연 고장의 검출율을 그대로 유지하면서 낮은 임계 수치에서 추가적으로 발생 가능한 합선 고장에 대한 검출율을 상승시킴으로써 전체적인 회로의 신뢰성을 보장할 수 있게 된다.

3. NTV 회로의 발열 특성을 고려한 지연 고장 테스트 방법론

3-1. 온도에 따른 NTV 회로의 지연 시간 변화 특성 및 테스트 신뢰성 문제 분석

NTV 회로는 동작 전압이 문턱 전압과 차이가 나지 않기 때문에 온도와 공정 변이에 소자의 문턱 전압이 영향을 받는 경우, 회로의 지연시간에 미치는 영향이 더욱 크게 나타나게 된다. 즉 STV 회로에 비해 NTV 회로는 공정변이와 온도 변화에 민감하게 영향을 받게 된다. 또한, NTV 회로는 높은 온도에서 회로의 지연 시간이 줄어 들고 낮은 온도에서는 늘어나는 경향을 보이기 때문에 STV 회로의 경향성과 반대의 방향성을 나타내는 것이 확인되었다⁷⁾. <그림 5>는 이를 검증하기 위한 NTV 환경에서 온도에 따른 회로 지연 분포 시뮬레이션 결과 및 테스트 시 발생 가능한 이슈에 대한 그래프를 나타낸다. 해당 시뮬레이션에서는 타겟 온도를 50°C로 가정하여 진행하였으며 NTV 회로에서 이에 따른 3sigma에 해당하는 회로 지연 시간은 4.1717(normalized)이다. 일반적으로 지연 고장 테스트에서는 회로 지연시간 분포의

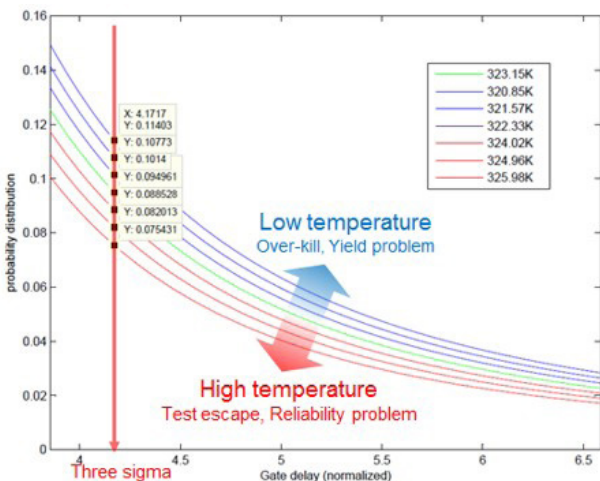


그림 5. NTV 회로에서의 온도에 따른 회로 지연 분포 시뮬레이션 결과

3sigma(표준편차의 3배) 이상의 회로들을 테스트에서 고장으로 판별하기 때문에 해당 수치를 사용했다. 발열 특성에 따른 STV와 NTV 회로의 지연 분포 차이 때문에, 테스트 시 발열 특성을 고려할 때에 기존의 테스트 방법과는 다른 테스트 방법이 NTV 환경에서는 필요하게 된다. 기존의 STV 환경에서 제안되던 테스트 방법론에서는 테스트 시의 온도를 테스트 시에 발생하는 고열에 의하여 회로가 영구적인 손상을 입지 않도록 관리하는 데에 초점이 맞춰진 반면, NTV 회로에서는 전력 소모가 STV 회로에 비하여 획기적으로 줄어들었기 때문에 고온에 의한 손상은 상대적으로 줄어들 것으로 보인다. 하지만 앞서 말한대로 NTV 회로는 온도 변화에 의하여 회로 지연시간이 민감하게 반응하기 때문에 낮은 온도에 의해 회로의 지연시간이 길어져 동작온도에서는 정상인 회로를 추가적인 고장으로 판별하거나 높은 온도에 의해 회로의 지연 시간이 짧아져 지연 고장 테스트를 통해 걸러내어야 할 제품을 걸러내지 못하는 문제가 발생할 수 있다. 따라서 NTV 회로를 테스트 할 시에는 이러한 발열 특성을 고려한 새로운 테스트 방법론이 필요하게 된다.

3-2. NTV 회로의 발열 특성을 고려한 테스트 스케줄링 방법

기존의 STV 회로에서 테스트를 진행할 때에는 지연 고장과 고착 고장의 테스트 스케줄링 시 고온에 의한 회로 손상을 우려했기 때문에 모든 테스트에 대해 일정 온도에 대한 제한을 두어 테스트를 진행할 필요가 있었다^{8),9)}. 하지만 NTV 회로의 테스트 과정에서는 온도에 의한 지연 변이 정도와 그에 의한 테스트 신뢰성 하락을 방지하는 것이 목적이 되기 때문에 기존의 STV 회로의 테스트 방법론에서 사용된 온도 제한 기법과는 다른 제한 기법을 두어 테스트를 할 필요가 있다. 먼저, NTV 회로의 경우 전력 소모가 낮아 발열량이 매우 작기 때문에 회로에 손상을 줄 정도의 온도가 발생하지 않아 고착 고장 테스트의 경우 온도 제한이 큰 의미가 없게 된다. 또한 회로 테스트 시 발생하는 온도 상승은 기존의 STV 회로에서는 지연 시간을 상승시켰기 때문에 지연 고장 테스트 시 지연 고장으로 판별되지 않을 지연 시간을 갖는 회로도 고장으로 판별되게 하는 overkill 현상에 대해 고려해야 했다면, NTV 회로에서는 지연 시간을 하락시키기 때문에 지연 고장으로 판별될 지연 시간을 갖는 회로가 고장이 아닌 것으로 판별되게 하는 test escape 현상에 대해 고려해야 한다. 즉, 기존의 STV 테스트 방법처럼 고정된 온도 제한을 두고 테스트를 진행하는 것이 아니라 고장 시뮬레이션 및 테스트 패턴 생성 후 패턴 별 회로의 온도 상승 정도를 파악하여 그에 맞는 테스트 스케줄링 기법을 적용하여 테스트를 진행하는 것이 전체 NTV 회로의 테스트를 진행하는 데 있어 훨씬 효율적인 방법이 될 수 있다.

<그림 6>은 NTV 회로를 테스트 하기 위한 온도 제한 기법 및 이를 위한 테스트 스케줄링 알고리즘을 나타낸다. 일반적으로 짧은 패스에 해당하는 지연 고장일수록 더 큰 타이밍 슬랙을 지니게 된다¹⁰⁾. 따라서 NTV 회로의 테스트 시에 긴 패스에 해당하는 지연 고장들은 test escape 현상에 더욱 위험하게 되고 짧은 패스에 해당하는 지연 고장들은 상대적으로 더 안전하게 된다. 따라서 지연 고장 시뮬레이션을 진행 후 테스트 패턴을 생성할 때 긴 패스에 해당하는 고장들을 잡을 수 있는 테스트 패턴들을 low robust pattern, 짧은 패스에 해당하는

고장들을 잡을 수 있는 테스트 패턴들을 high robust pattern으로 분류하여 테스트를 진행하게 되면 테스트 시 발생하는 온도 변화를 고려하면서 효율적인 테스트 진행이 가능하게 된다. 이에 대한 설명이 <그림 6>의 오른쪽 그림에 나타나 있다. 기존의 제한된 온도 환경 하에서 테스트를 진행 시 회로의 온도가 제한된 온도에 다다를 경우, 이를 식혀주는 과정을 진행 후 다시 테스트를 진행하는 열 처리 과정이 필요하다. 하지만 test escape를 유발할 확률이 높은 low robust pattern을 이용하여 먼저 테스트를 진행시킨 후 점차적으로 high robust pattern을 이용하여 테스트를 진행할 경우, 열 처리 과정에 따른 지연 시간이 줄어들게 되고 결론적으로 전체 테스트 시간이 줄어드는 것을 확인할 수 있다.

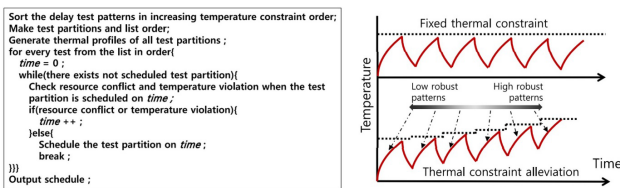


그림 6. NTV 회로의 테스트를 위한 온도 제한 기법 (왼쪽) 및 테스트 스케줄링 알고리즘 (오른쪽)

4. 결론

4차 산업혁명과 관련하여 필수적으로 연계되는 기술인 사물인터넷과 스마트 웨어러블 컴퓨팅 기술에서 가장 핵심적인 부분은 높은 전력 효율성을 지니는 회로 설계 기술이며 이를 위해서는 반드시 NTV 회로 설계가 필요하다. 이러한 NTV 회로를 설계하는 과정에서 특히 수율 향상과 신뢰성을 보장하기 위한 테스트 과정은 이것이 반도체 제조 비용과 여러 안전 이슈들과 직결될 수 있기 때문에 매우 중요하다고 할 수 있다. 앞서 살펴본 대로 NTV 회로는 기존의 STV 회로와 달리 합선 고장의 위험성이 훨씬 크고, 발열에 따른 지연 고장의 측면에서 STV 회로와 정 반대의 다른 특성들을 지니는 부분들이 있기 때문에 실제 NTV 회로가 상용화 되어 양산되기 위해서는 이러한 부분을 고려한 테스트 방법론에 대해 더 많은 연구가 필요할 것으로 예상된다.

참고문헌

1. H. Kaul et al., "Near-threshold voltage (NTV) design—Opportunities and challenges," in Proceedings of Design Automation Conference, 2012, pp. 1149–1154.
2. R. G. Dreslinski et al., "Near-threshold computing: Reclaiming Moore's law through energy efficient integrated circuits," in Proceedings of the IEEE, 2010, vol. 98, no. 2, pp. 253–266.
3. 주재욱 외, "웨어러블 기기 UI/UX를 위한 SoC 설계기술 동향," 전자공학 회지, 제41권, 제11호, 20-29쪽, 2014년 11월
4. Yang, Y., Jeong et al., "Single bit-line 7T SRAM cell for near-threshold voltage operation with enhanced performance and energy in 14 nm FinFET technology," IEEE Transactions on Circuits and Systems I, vol. 63, no. 7, 2016, pp. 1023-1032.
5. F. S. Alghareb et al., "Energy and Delay Tradeoffs of Soft-Error Masking for 16-nm FinFET Logic Paths: Survey and Impact of Process Variation in the Near-Threshold Region," IEEE Transactions on Circuits and Systems II vol. 64, no. 6, 2017, pp. 695-699.
6. U. Ingelsson et al., "Process Variation-Aware Test for Resistive Bridges," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 28 no. 8, 2009, pp.1269-1274.
7. Kanda, K. et al., "Design impact of positive temperature dependence of drain current in sub 1V CMOS VLSIs," in Proceedings of Custom Integrated Circuits, 1999, pp. 563-566.
8. Bild, D.R. et al., "Temperature-aware test scheduling for multi-processor systems-on-chip," in Proceedings of International Conference on Computer-Aided Design, 2008, pp. 59-66.
9. N. Aghaee et al., "Process-variation and temperature aware SoC test scheduling technique," Journal of Electronic Testing, vol. 29, no. 4, pp. 499–520, 2013.
10. D. Xiang et al., "Thermal aware small-delay defect testing in integrated circuits for mitigating overkill," IEEE Transactions on Computer-Aided Design Integrated Circuits and System, vol. 35, no. 3, 2016, pp. 499–512.

저자정보



오형교 박사과정
 소속 연세대학교 전기전자공학과
 주 연구분야 SoC 테스트 및 디버그
 E-mail kyob508@soc.yonsei.ac.kr
 Homepage http://soc.yonsei.ac.kr/



강성호 교수
 소속 연세대학교 전기전자공학과
 주 연구분야 SoC 설계 및 테스트 / DFT
 E-mail shkang@yonsei.ac.kr
 Homepage http://soc.yonsei.ac.kr/

EDA Tool 소개 | Scientific Analog 사

GLISTER

개요

A. 목적 (모델 생성 및 시뮬레이션)

아날로그 회로의 동작을 디지털 검증환경인 SystemVerilog 상에서 모델링하고, 이를 포함한 혼성신호 시스템 IC의 동작 및 성능을 빠르고 정확한 시뮬레이션을 통해 검증한다.

B. 구분 (모델기반의 혼성신호 시스템 시뮬레이터)

XMODEL은 모델을 기반으로 아날로그 및 혼성신호 IC 시스템을 검증하는 시뮬레이터라는 점에서 기존의 Verilog-AMS, Matlab/Simulink, CppSim과 유사한 목적과 기능을 가지나, 순수한 디지털 시뮬레이터인 SystemVerilog 상에서 동작한다는 점, 정확도와 실행속도가 10~100배 이상 월등하다는 점, 비트에러율 같은 통계적인 분석이 가능하다는 점, 아날로그-디지털 co-simulation을 지원한다는 점 등에서 차별성을 갖는다.

C. Supported platform and O/S System

Linux RedHat Enterprise 64-bit Release 5.0 이상

D. 특성 및 기능

Scientific Analog 사는 현재 XMODEL, GLISTER 그리고 MODELZEN의 세가지 제품을 보유하고 있다.



XMODEL

XMODEL은 다양한 아날로그 회로들을 기존의 디지털 검증 환경인 SystemVerilog 상에서 쉽게 모델링하고 시뮬레이션할 수 있게 해주는 확장팩키지입니다.



GLISTER

GLISTER는 Cadence® Virtuoso® 환경에서 아날로그 모델을 코드 작성없이 schematic 형태로 쉽게 만들고 시뮬레이션할 수 있게 해주는 GUI 환경입니다.



MODELZEN

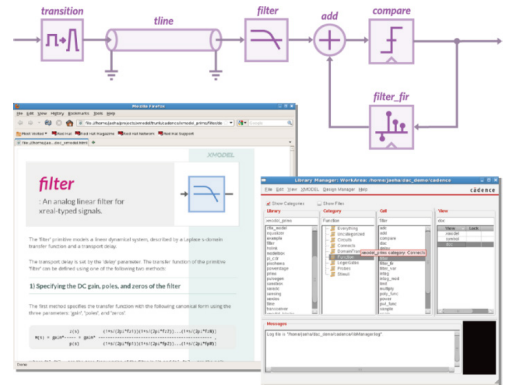
MODELZEN은 임의의 아날로그 회로로부터 SystemVerilog 모델을 자동으로 생성해주는 툴입니다.



GLISTER는 Cadence의 Virtuoso 환경에서 아날로그 회로의 모델들을 한 줄의 코드 작성 없이도 schematic 형태로 쉽게 표현하고, 이로부터 SystemVerilog 모델을 추출하여 시뮬레이션할 수 있게 해주는 GUI 환경이다.

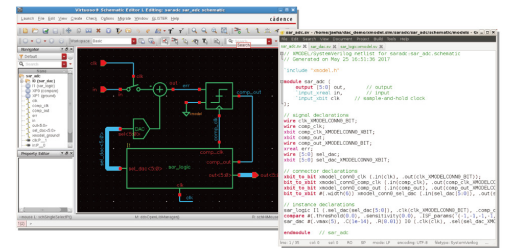
XMODEL Primitive 심볼을 이용한 Schematic 형태의 모델 작성

GLISTER는 XMODEL의 다양한 primitive들을 schematic symbol의 형태로 제공한다. 따라서 사용자는 다양한 아날로그 회로의 모델들을 코딩 작성없이 단순히 schematic 위에 그 symbol들을 배치하고 도선으로 연결하는 것으로 쉽게 표현할 수 있다. 특히, GLISTER 환경에서는 각 primitive에 대한 설명을 쉽게 열어볼 수 있어 모델 작성을 더 용이하게 해준다.



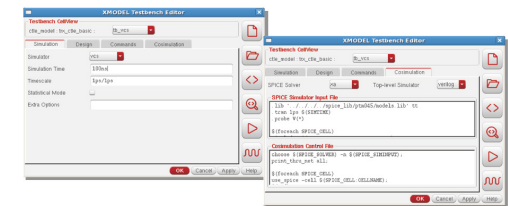
Hierarchical Netlisting을 통한 SystemVerilog 모델 생성

아날로그와 디지털 신호가 혼재하는 모델에서는 schematic 상의 각 도선이 wire, real, xbit 및 xreal 등의 다양한 signal type을 가질 수 있다. GLISTER는 이와 같은 signal type의 차이를 이해하는 유일한 netlister이며, 각 도선이 연결하는 primitive의 종류에 따라 적절한 signal type을 자동으로 인지할 수 있다. 게다가, 서로 다른 type의 신호들 간에 연결이 필요할 때 그 신호의 type을 변환해주는 connect primitive들도 자동으로 삽입할 수 있다.



통합된 Testbench 관리

GLISTER의 Testbench Editor를 사용하면, 대상 design의 hierarchy 구성, 시뮬레이션의 각종 옵션 등 시뮬레이션 테스트 벤치에 해당되는 내용들을 GUI 환경을 통해 정의할 수 있고, 그 내용을 Cadence 설계 database 상의 cellview의 형태로 저장할 수 있다. 또한 GLISTER는 이 테스트벤치를 netlisting할 때, command-line 상에서 Makefile 스크립트를 통해 직접 실행할 수 있는 형태로 시뮬레이션 폴더를 구성하므로 일괄처리도 용이하게 수행할 수 있다.



XMODEL-SPICE Co-simulation 지원

대상 design의 hierarchy에 회로 schematic view와 모델 schematic view들이 함께 있는 경우, GLISTER는 사용자가 XMODEL-SPICE co-simulation을 수행하고자 한다고 판단하고, 그에 알맞는 SystemVerilog 모델 파일, SPICE 또는 Spectre netlist 파일 및 co-simulation에 필요한 추가 파일들을 자동으로 준비해준다. 이러한 대상 design hierarchy는 Cadence의 Hierarchy Editor를 사용하여 선택할 수 있다. GLISTER의 Testbench Editor는 Synopsys의 VCS와 XA, Cadence의 NCVerilog와 APS 등 다양한 시뮬레이터들에 대해서 일관된 사용자 인터페이스를 제공한다.

