

오픈 - 소스 EDA 도구 활용 "내 칩 (MyChip) 제작"

반도체 설계 교육의 혁신

국일호

goodkook@gmail.com

<https://github.com/GoodKook/ETRI-0.5um-CMOS-MPW-Std-Cell-DK>

반도체 설계 및 MyChip MPW 워크숍
부경대학교 반도체 특성화대학 사업단
2025년 4월 4일

“내 칩 (My Chip) 제작 서비스”

- 0.5um Si-CMOS/2-Poly 3-Metal 표준 공정
- 풀 - 커스텀 설계
- **공정과 패키지 (28-pin SOP) 무료**

반도체 설계인력양성을 위한

6인치 Si CMOS 설계검증 플랫폼 개발 및 MPW 지원

『내 칩(My Chip) 제작 서비스』

2024년 9월 11일

반도체 설계인력양성을 위한

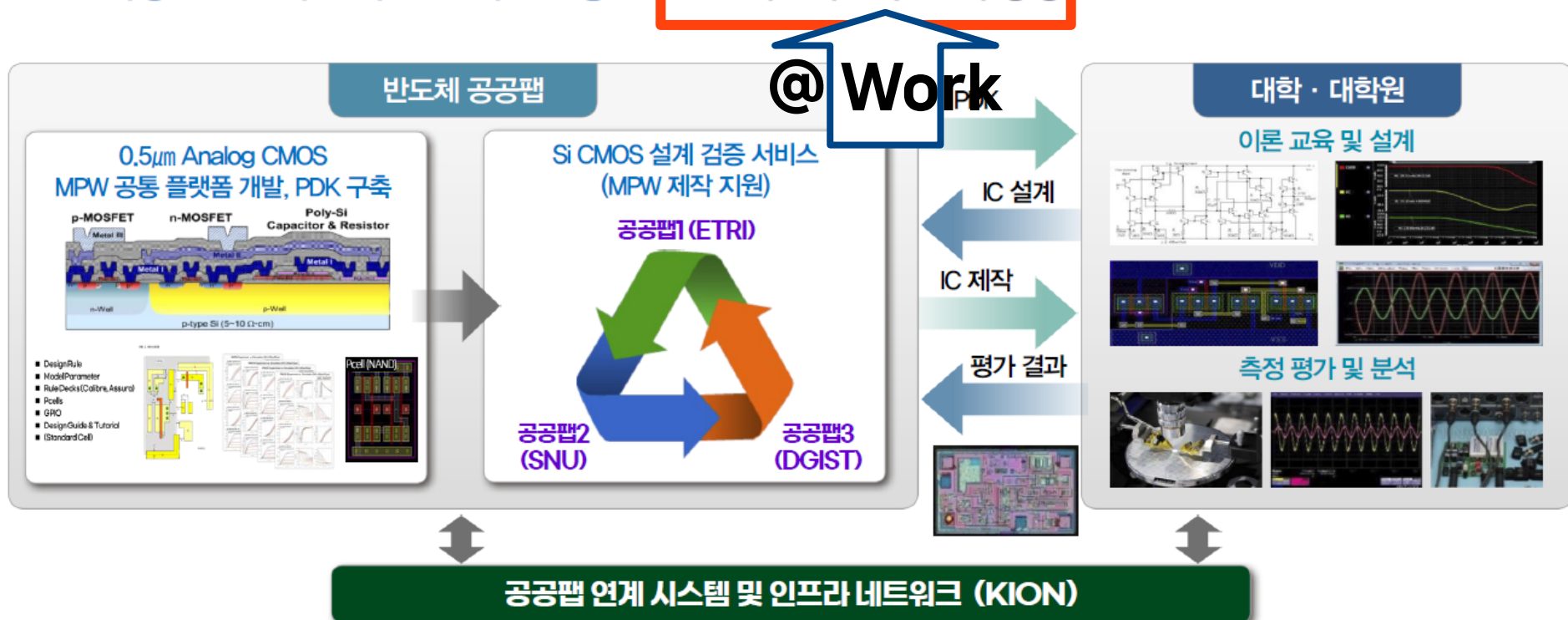
6인치 Si CMOS 설계검증 플랫폼 개발 및 MPW 지원

『내 칩(My Chip) 제작 서비스』

2024년 9월 11일

내 칩(My Chip) 제작 서비스

- ✓ 반도체 설계 분야의 학부생 · 대학원생에게 공공팹을 활용한 6인치 CMOS 공정 기반 MPW 제작 지원
→ 학생들 본인의 설계 칩을 직접 검증 → 반도체설계 실무 인력 양성

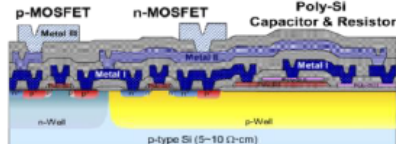


내 칩(My Chip) 제작 서비스

- ✓ 반도체 설계 분야의 학부생 · 대학원생에게 공공팹을 활용한 6인치 CMOS 공정 기반 MPW 제작 지원
→ 학생들 본인의 설계 칩을 직접 검증 → 반도체설계 실무 인력 양성

Design & Test

0.5 μ m Analog CMOS
MPW 공통 플랫폼 개발, PDK 구축



- Design Rule
- Model Parameter
- RuleDecks (Calibre, Assura)
- Pcells
- GPO
- DesignGuide & Tutorial
- Standard Cell



0.5 μ m CMOS 설계 검증 서비스
(MPW 제작 지원)

공공팹1 (ETRI)



PDK

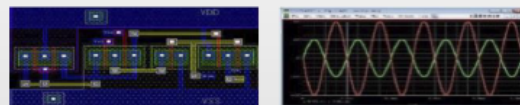
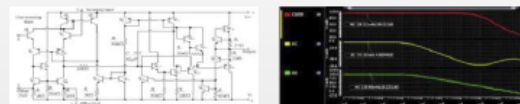
IC 설계

IC 제작

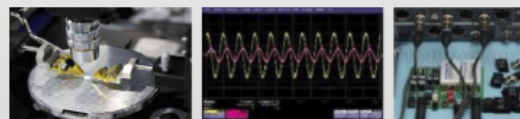
평가 결과

대학 · 대학원

이론 교육 및 설계



측정 평가 및 분석



공공팹 연계 시스템 및 인프라 네트워크 (KION)

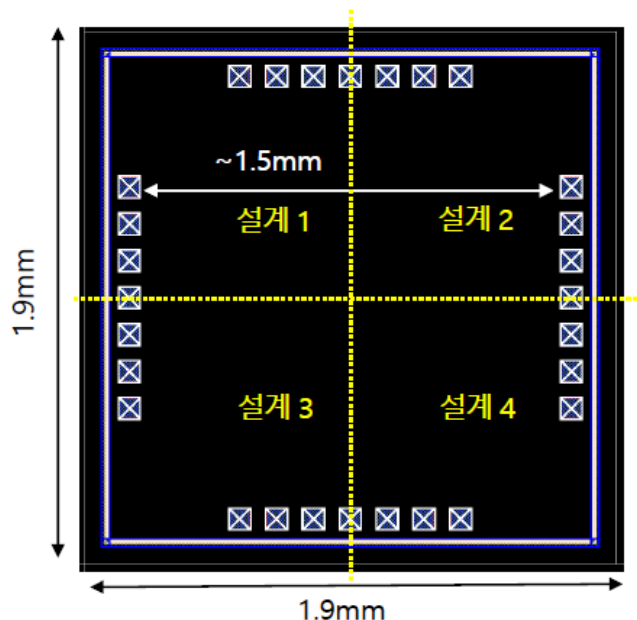
MPW 칩 구성

- ✓ Chip size : 1.9mm x 1.9mm, 1팀당 설계 면적 1.5mm x 1.5mm 활용, GPIO 사용시 설계 면적 1.0mm x 1.0mm 활용
- ✓ 팀(칩)당 구성인원 제한은 없으나 4명 내외 구성 권장
- ✓ 칩 크기가 다를 경우 별도 협의

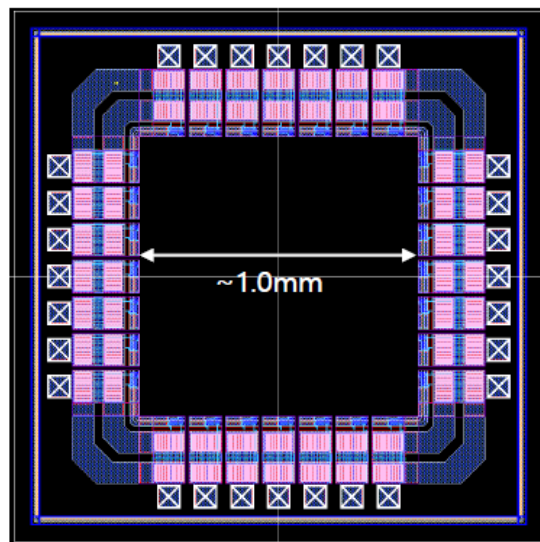
Power-Performance-Area

설계 제한 극복 (Core & Pad)

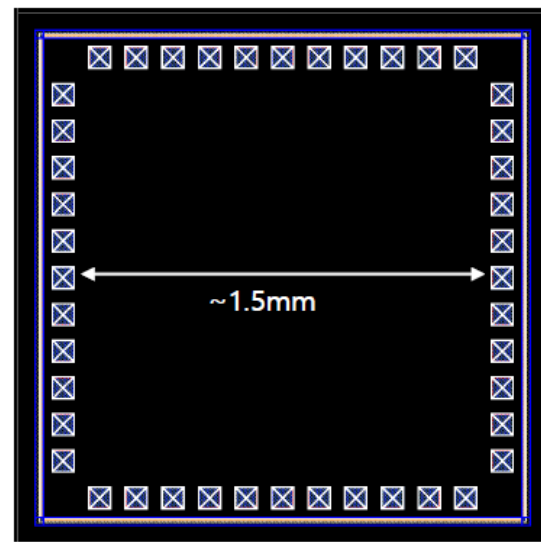
■ PAD 기본 정보



■ GPIO사용 및 패키지 반영 경우 (예)



■ 패키지 반영하지 않는 경우 (예) → PAD 자유로이 배치



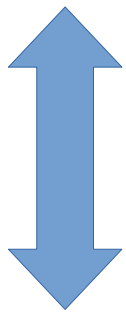
“내 칩 (My Chip) 제작 서비스”

- 0.5um Si-CMOS/2-Poly 3-Metal 표준 공정
- 풀 - 커스텀 설계
- 공정과 패키지 (28-pin SOP) 무료

반도체 설계 교육의 과제

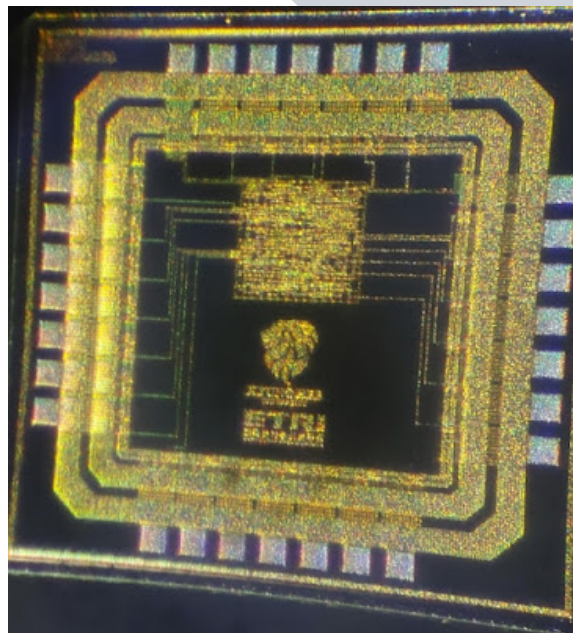
- “추상성 격차”의 극복

(최신 기술에 노출된 학생들)
AI 반도체 ?



동기부여
반도체 설계는 재밌다 !

트랜지스터 !
(기초 기술의 **풀 - 커스텀**)



Chip Art

PDK (Process Design Kit)

■ Design Rule, Design Guide & Tutorial

■ Model Parameter

- ✓ Devices: CMOS, Sub-BJT, Diode, Resistor, Capacitor
- ✓ DC, Capacitance, Temperature Parameter (25°C ~ 115°C), Corner Model (MOSFET)
- ✓ (미제공) RF Parameter, Noise Parameter

■ Rule Decks

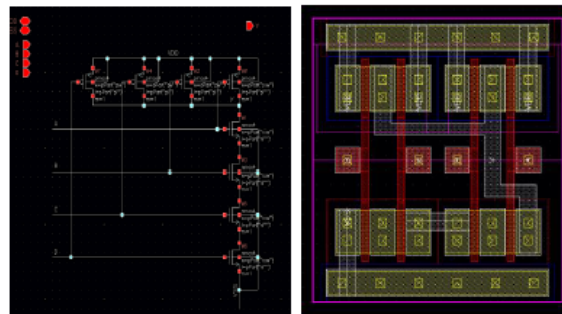
- ✓ Calibre (DRC, LVS, PEX), Assura (DRC, LVS) → Assura PEX 및 Rule Deck 보완 중

■ Pcells, GPIO

- ✓ Pcell: CMOS, BJT, Diode, Cap, Res, Logic Gate (INV, NAND2,3,4, NOR2,3,4)
- ✓ Analog/Digital IC용 IO-PAD 9종 *ESD에 의한 칩 손상 방지를 위해 PGIO사용 권장

■ (Standard Cell) → 구축 예정

〈Pcell 예: nand4〉



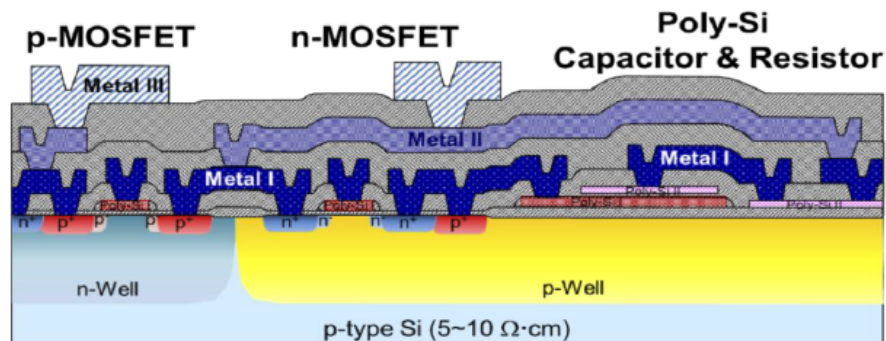
〈MOSIS 제공 Standard Cell〉

The IIT cell library has been developed to provide a free, non-proprietary library for MOSIS SCMOs rules.		List of Cells	
It supports both commercial and public domain EDA tools.		AND2X1	INVX4
The library is available for free upon request from http://vlsi.ece.iit.edu/scells		AND2X2	INVX8
Supported MOSIS Technologies		AOI21X1	LATCH
AMI 0.5μm (includes pads)		AOI22X1	MUX2X1
AMI 0.35μm (includes pads)		BUF2	NAND2X1
TSMC 0.25μm		BUF4	NAND3X1
TSMC 0.18μm		CLKBUF1	NOR2X1
		CLKBUF2	NOR3X1
		CLKBUF3	OAI21X1
		DFNEXX1	OAI22X1
		DFPOSX1	OR2X1
		DFFSR	OR2X2
		FAX1	TBUF1
		FILL	TBUF2
		HAX1	XNOR2X1
		INVX1	XOR2X1
		INVX2	

MPW 지원 공정: 0.5 μ m CMOS (5V), 공공팩 공정 호환, 동일 특성

주요 공정: 2-Poly, 3-Metal 공정

- Diffused Twin Well : 1150°C, 120min
- LOCOS Isolation: 950°C, 5200Å
- Gate Oxide = 110Å (Wet Oxidation)
- Poly Gate; Undoped Poly + POCl₃ Doping (or Doped Poly)
- PIP Capacitor; HTO (or LPCVD-TEOS) 700Å (0.5fF/ μ m²)
- Poly2 Resistor; 125 Ω /sq.

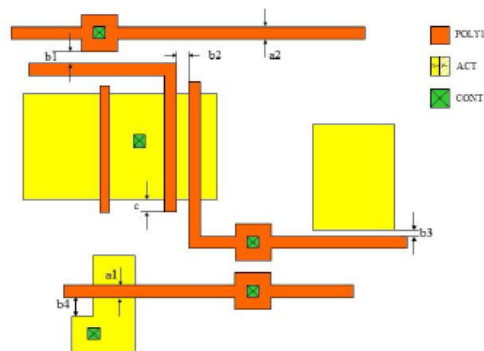


주요 디자인 룰

- Minimum Gate (POLY1) Width/Space = 0.5 μ m/0.6 μ m
- Minimum Active Width/Space = 0.8 μ m/0.8 μ m
- Minimum Transistor(N/PMOS) width = 1.6 μ m
- Minimum Contact Size/Space = 0.6 μ m/0.6 μ m
- Minimum Via1 & Via2 Size/Space = 0.8 μ m/0.8 μ m (Cont, Via Stack 불가)
- Minimum Metal1, 2, 3 Width/Space = 0.8 μ m/0.8 μ m, 1 μ m/1 μ m, 1 μ m/1 μ m

4.3 POLY1 Rules

Rule No.	Description	Label	Value (μ m)
3.PL1.W1	POLY1 width over ACT (NMOS or PMOS)	a1	≥ 0.5
3.PL1.W2	Minimum POLY1 width (Transistor Channel Length)	a1	≤ 100.0
3.PL1.W3	POLY1 width for interconnect	a2	≥ 0.5
3.PL1.S1	POLY1 to POLY1 space in field	b1	≥ 0.6
3.PL1.S2	POLY1 to POLY1 space on ACT	b2	≥ 0.6
3.PL1.S3	Space from POLY1 in field to ACT	b3	≥ 0.5
3.PL1.S4	ACT space to POLY1 in field	b4	≥ 0.5
3.PL1.O	Overlap of POLY1 extended into field wide	c	≥ 0.5
3.PL1.ANT	Minimum ratio of field poly area to gate poly for a given transistor (antenna rule)		≤ 200

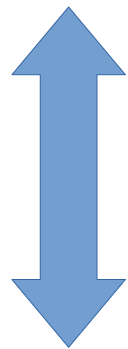


반도체 설계 교육의 과제

▪ “추상성 격차”의 극복

(최신 기술에 노출된 학생들)

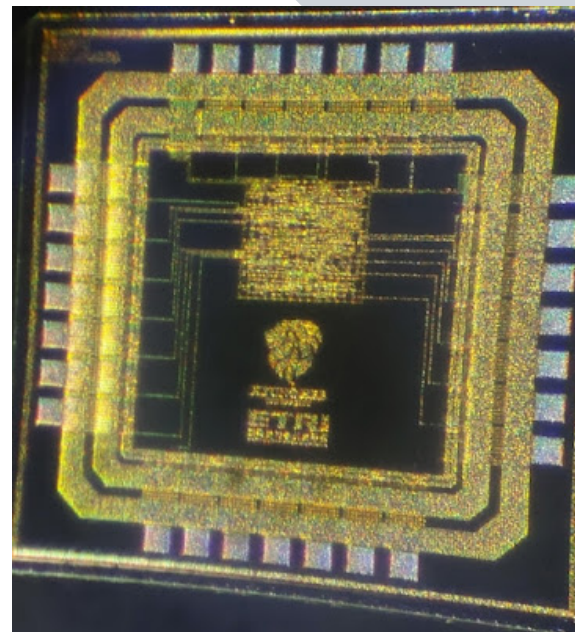
AI 반도체 ?



동기부여
반도체 설계는 재밌다 !

트랜지스터 !

(기초 기술의 **풀 - 커스텀**)



Chip Art



PDK (Process Design Kit)

■ Design Rule, Design Guide & Tutorial

■ Model Parameter

- ✓ Devices: CMOS, Sub-BJT, Diode, Resistor, Capacitor
- ✓ DC, Capacitance, Temperature Parameter (25°C ~ 115°C), Corner Model (MOSFET)
- ✓ (미제공) RF Parameter, Noise Parameter

■ Rule Decks

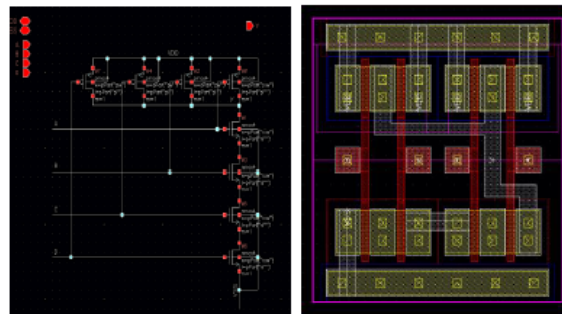
- ✓ Calibre (DRC, LVS, PEX), Assura (DRC, LVS) → Assura (미제공) 클리닝

■ Pcells, GPIO

- ✓ Pcell: CMOS, BJT, Diode, Cap, Res, I/O @ Highest Abstraction Level
- ✓ Analog/Digital IC용 IO-PAD 9종 *ESB 사용 권장

■ (Standard Cell) → 구축 예정

〈Pcell 예: nand4〉



〈MOSIS 제공 Standard Cell〉

The IIT cell library has been developed to provide a free, non-proprietary library for MOSIS SCMOS rules.		List of Cells	
It supports both commercial and public domain EDA tools.		AND2X1	INVX4
The library is available for free upon request from http://vlsi.ece.iit.edu/scells		AND2X2	INVX8
		AOI21X1	LATCH
		AOI22X1	MUX2X1
		BUF2	NAND2X1
		BUF4	NAND3X1
		CLKBUF1	NOR2X1
		CLKBUF2	NOR3X1
		CLKBUF3	OAI21X1
		DFFNEX1	OAI22X1
		DFFP0X1	OR2X1
		DFFSR	OR2X2
		FAX1	TBUF1
		FILL	TBUF2
		HAX1	XNOR2X1
		INVX1	XOR2X1
		INVX2	
		INVX3	

Supported MOSIS Technologies

- AMI 0.5μm (includes pads)
- AMI 0.35μm (includes pads)
- TSMC 0.25μm
- TSMC 0.18μm

반도체 설계 교육의 애로점

반도체 설계 교육 과정에서 과목간 연속성

- 학과 이수 체계 vs 반도체 설계 플로우

설계 도구의 확보 / 실습 환경

- 고가의 도구 사용료
- 공정 자료의 비공개
- 공정이 포함된 실습의 고 비용

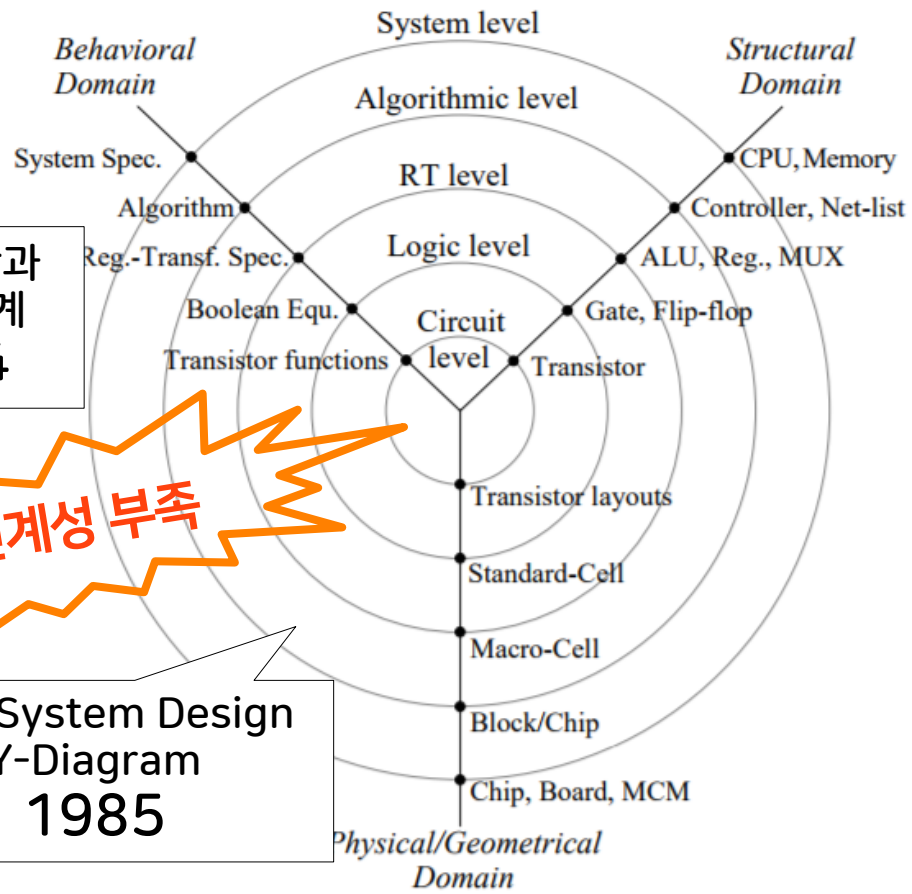
2024학년도 전자공학부 전공공학전공

학년/학기		1-1	1-2	2-1	2-2	3-1	3-2	4-1	4-2	
이수구분/단계		전공기초 및 MSC				전공핵심		전공심화		
전공능력	① 전자회로 설계능력	전자공학 세미나		A 디지털공학 및 실습	A 전자기학	A 반도체공학	B RF기초 및 회로설계		C 시스템 네트워크	
		C 기초 전자공학	A 창의적공학 설계	A 회로이론1	A 회로이론2	B HDL 및 실습	B FPGA 설계실습	C 디지털 시스템설계	C 차세대 디스플레이	
					A 전자회로 및 실습 1	A 전자회로 및 실습2		전자공학 ESP	C 지능형 반도체개론	
									전자공학 ESP	
								종합설계 기획	종합설계1	종합설계2
	② 임베디드 SW능력	프로그래밍 언어	프로그래밍 언어	A 데이터 구조론	A 컴퓨터 구조론	B 마이크로 프로세서1	C 마이크로 프로세서2	C 안드로이드 프로그래밍	C 고급 머신러닝 실습	
				A 객체지향 언어	A 파이썬	C 머신러닝 실습	C 리눅스 시스템설계			
	③ 신호 및 데이터 처리능력			A 신호 및 시스템	A 머신러닝의 이해	B 네트워크 기초	B 인터넷 프로토콜 설계			
		선형대수학	공업수학1		A 통신이론	B 디지털 통신1	B 디지털 통신2		B 이동통신 공학	
					확률및 통계학	A 디지털 신호처리	B 영상신호 처리	B DSP시스템 및 실습	C 멀티미디어 프로그래밍 실습	

전자공학과
이수 체계
2024

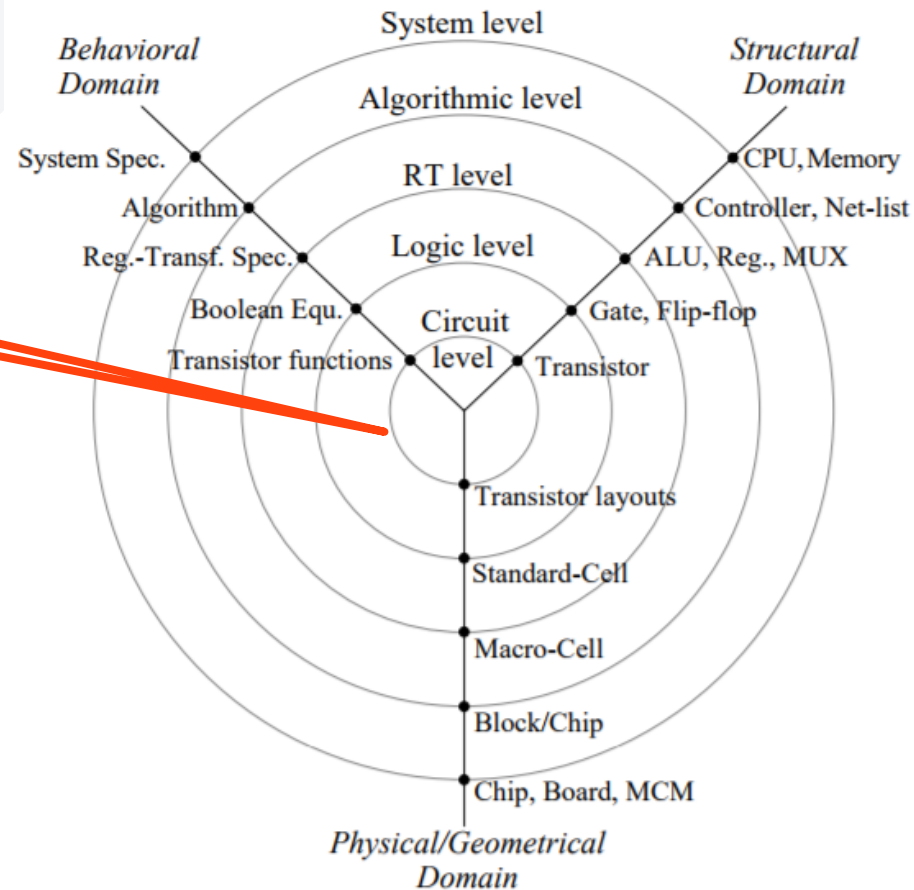
연계성 부족

VLSI System Design
Y-Diagram
1985



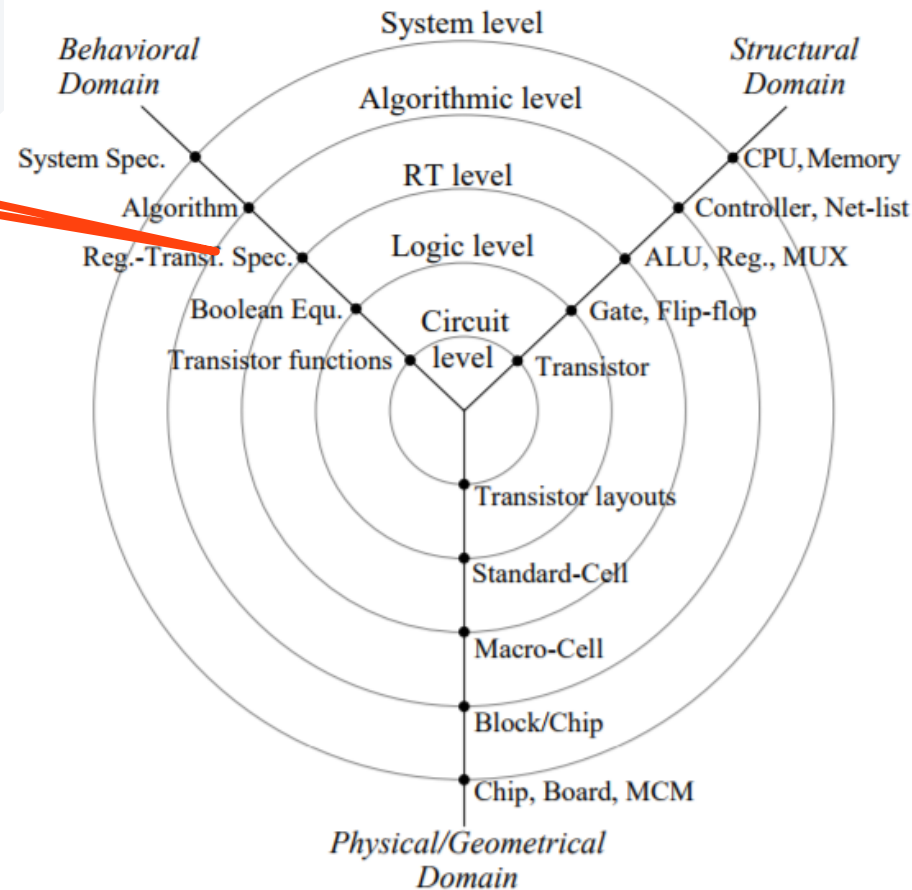
2024학년도 전자공학부 전공공학전공

학년/학기		1-1	1-2	2-1	2-2	3-1	3-2	4-1	4-2
이수구분/단계		전공기초 및 MSC				전공핵심		전공심화	
전공능력	① 전자회로 설계능력	전자공학 세미나		A 디지털공학 및 실습	A 전자기학	A 반도체공학	B RF기초 및 회로설계		C 시스템 네트워크
		C 기초 전자공학	A 창의적공학 설계	A 회로이론1	A 회로이론2	B HDL 및 실습	B FPGA 설계실습	C 디지털 시스템설계	C 차세대 디스플레이
				A 전자회로 및 실습 1	A 전자회로 및 실습 2			전자공학 ESP	C 지능형 반도체개론
									전자공학 ESP
	② 임베디드 SW능력	프로그래밍 언어	프로그래밍 언어	A 데이터 구조론	A 컴퓨터 구조론	B 마이크로 프로세서1	C 마이크로 프로세서2	C 안드로이드 프로그래밍	C 고급 머신러닝 실습
				A 객체지향 언어	A 파이썬	C 머신러닝 실습	C 리눅스 시스템설계		
③ 신호 및 데이터 처리능력			A 신호 및 시스템	A 머신러닝의 이해	B 네트워크 기초	B 인터넷 프로토콜 설계			
	선형대수학	공업수학1		A 통신이론	B 디지털 통신1	B 디지털 통신2		B 이동통신 공학	
				확률및 통계학	A 디지털 신호처리	B 영상신호 처리	C 멀티미디어 프로그래밍 실습		
						B DSP시스템 및 실습			



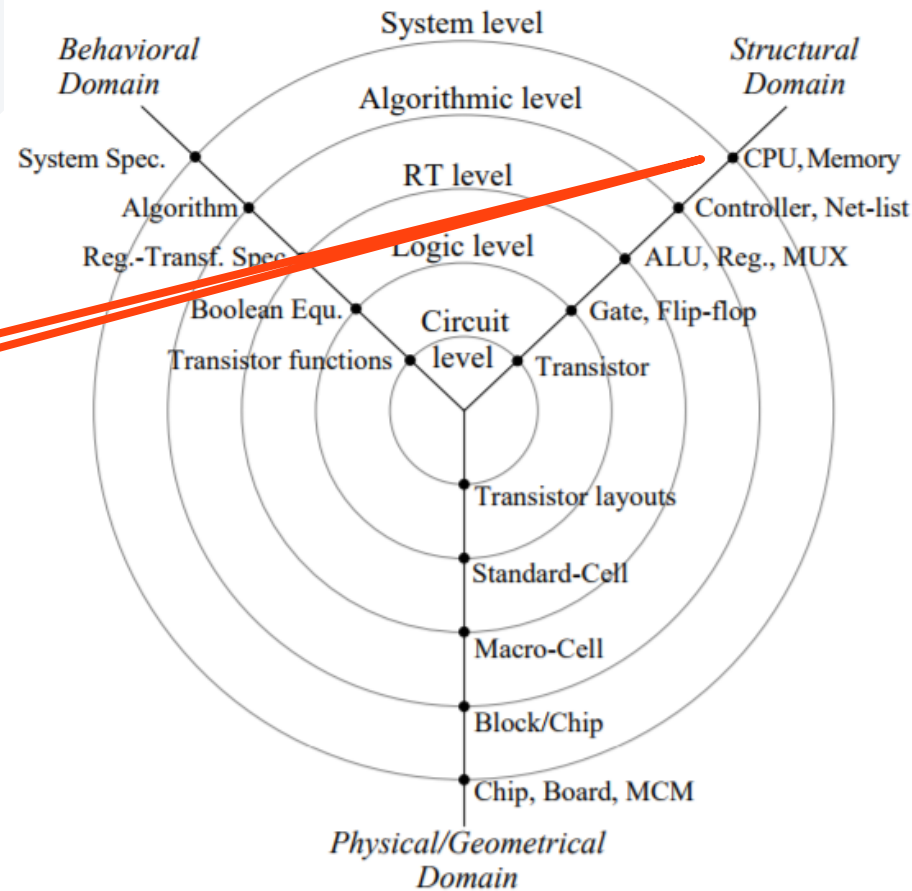
2024학년도 전자공학부 전공공학전공

학년/학기		1-1	1-2	2-1	2-2	3-1	3-2	4-1	4-2
이수구분/단계		전공기초 및 MSC				전공핵심		전공심화	
전공능력	① 전자회로 설계능력	전자공학 세미나		A 디지털공학 및 실습	A 전자기학	A 반도체공학	B RF기초 및 회로설계		C 시스템 네트워크
		C 기초 전자공학	A 창의적공학 설계	A 회로이론1	A 회로이론2	B HDL및실습	B FPGA 설계실습	C 디지털 시스템설계	C 차세대 디바이스 설계
					A 전자회로 및 실습 1	A 전자회로 및 실습2		전자공학 ESP	C 지능형 반도체개론
									전자공학 ESP
	② 임베디드 SW능력	프로그래밍 언어	프로그래밍 언어	A 데이터 구조론	A 컴퓨터 구조론	B 마이크로 프로세서1	C 마이크로 프로세서2	C 안드로이드 프로그래밍	C 고급 머신러닝 실습
				A 객체지향 언어	A 파이썬	C 머신러닝 실습	C 리눅스 시스템설계		
	③ 신호 및 데이터 처리능력			A 신호 및 시스템	A 머신러닝의 이해	B 네트워크 기초	B 인터넷 프로토콜 설계		
		선형대수학	공업수학1		A 통신이론	B 디지털 통신1	B 디지털 통신2		B 이동통신 공학
					확률및 통계학	A 디지털 신호처리	B 영상신호 처리	C 멀티미디어 프로그래밍 실습	
						B DSP시스템 및 실습			



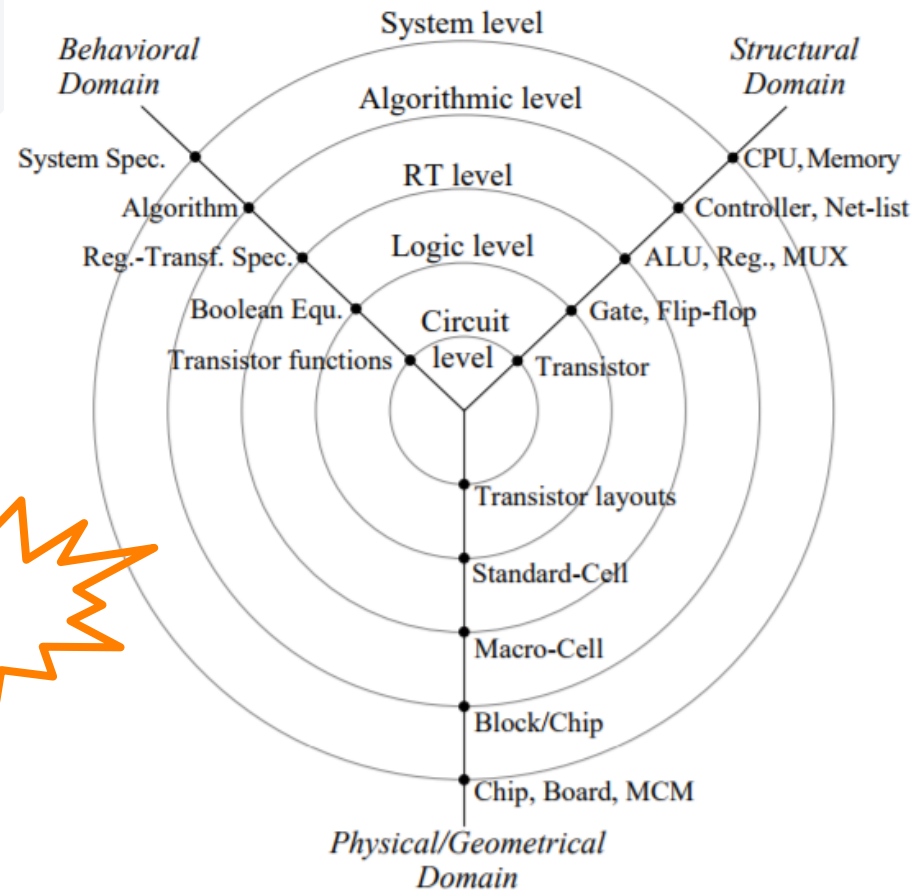
2024학년도 전자공학부 전공공학전공

학년/학기	1-1	1-2	2-1	2-2	3-1	3-2	4-1	4-2
이수구분/단계	전공기초 및 MSC				전공핵심		전공심화	
전공능력								
	전자공학 세미나		A 디지털공학 및 실습	A 전자기학	A 반도체공학	B RF기초 및 회로설계		C 시스템 네트워크
	C 기초 전자공학	A 창의적공학 설계	A 회로이론1	A 회로이론2	B HDL및실습	B FPGA 설계실습	C 디지털 시스템설계	C 차세대 디스플레이
① 전자회로 설계능력				A 전자회로 및 실습 1	A 전자회로 및 실습2		전자공학 ESP	C 지능형 반도체개론
								전자공학 ESP
							종합설계 기획	종합설계1
② 임베디드 SW능력	프로그래밍 언어	프로그래밍 언어	A 데이터 구조론	A 컴퓨터 구조론	B 마이크로 프로세서1	C 마이크로 프로세서2	C 안드로이드 프로그래밍	C 고급 머신러닝 실습
			A 객체지향 언어	A 파이썬	C 머신러닝 실습	C 리눅스 시스템설계		
③ 신호 및 데이터 처리능력			A 신호 및 시스템	A 머신러닝의 이해	B 네트워크 기초	B 인터넷 프로토콜 설계		B 이동통신 공학
	선형대수학	공업수학1		A 통신이론	B 디지털 통신1	B 디지털 통신2		
				확률및 통계학	A 디지털 신호처리	B 영상신호 처리	C 멀티미디어 프로그래밍 실습	B DSP시스템 및 실습



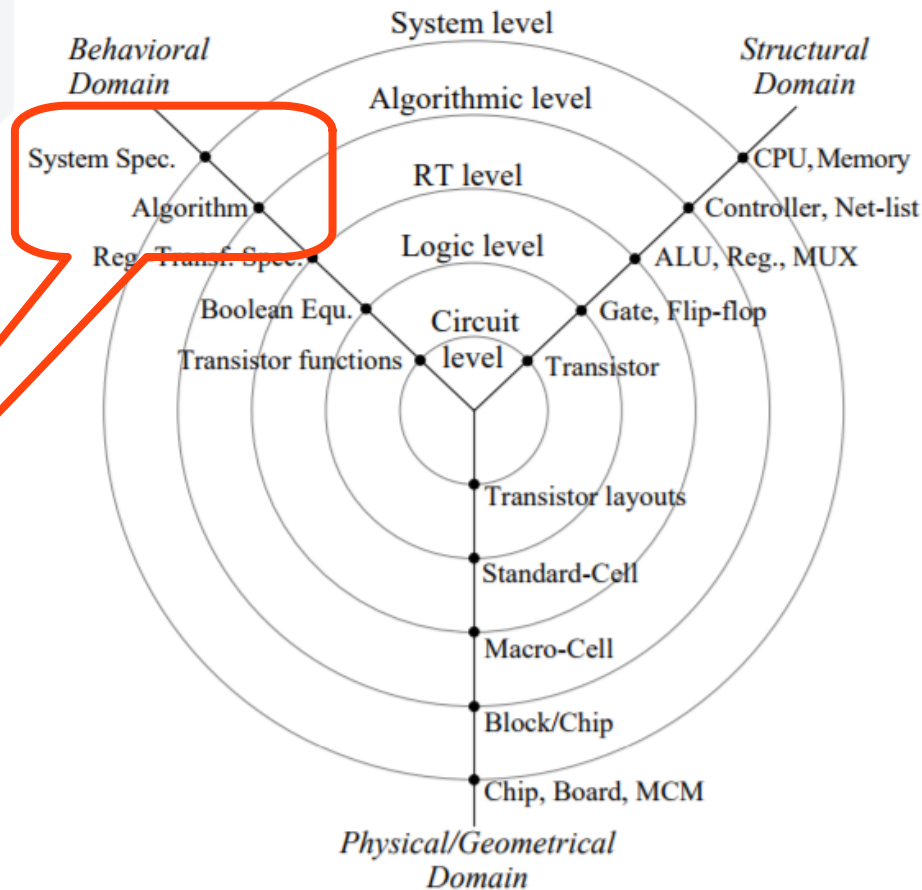
2024학년도 전자공학부 전공공학전공

학년/학기	1-1	1-2	2-1	2-2	3-1	3-2	4-1	4-2
이수구분/단계	전공기초 및 MSC				전공핵심		전공심화	
전공능력								
① 전자회로 설계능력	전자공학 세미나		A 디지털공학 및 실습	A 전자기학	A 반도체공학	B RF기초 및 회로설계	C 시스템 네트워크	
	C 기초 전자공학	A 창의적공학 설계	A 회로이론1	A 회로이론2	B HDL 및 실습	B FPGA 설계실습	C 디지털 시스템설계	C 차세대 디스플레이
② 임베디드 SW능력	프로그래밍 언어	프로그래밍 언어	A 데이터 구조론	A 컴퓨터 구조론	B 마이크로 프로세서1	C 마이크로 프로세서2	C 안드로이드 프로그래밍	C 고급 머신러닝 실습
			A 객체지향 언어	A 파이썬	C 머신러닝 실습	C 리눅스 시스템설계		
③ 신호 및 데이터 처리능력			A 신호 및 시스템	A 머신러닝의 이해	B 네트워크 기초	B 인터넷 프로토콜 설계		
	선형대수학	공업수학1	A 통신이론	A 통신이론	B 디지털 통신1	B 디지털 통신2	B 이동통신 공학	
			확률및 통계학	A 디지털 신호처리	B 영상신호 처리	C 멀티미디어 프로그래밍 실습		
					B DSP시스템 및 실습			



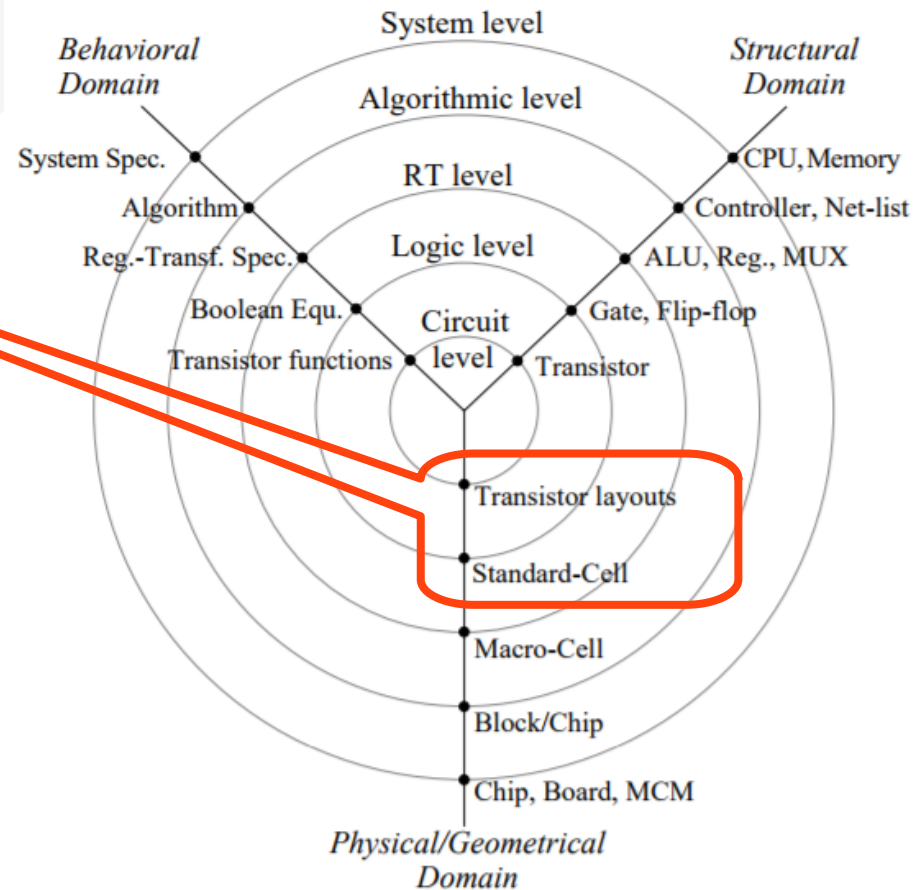
2024학년도 전자공학부 전공공학전공

학년/학기		1-1	1-2	2-1	2-2	3-1	3-2	4-1	4-2
이수구분/단계		전공기초 및 MSC				전공핵심		전공심화	
전공능력	① 전자회로 설계능력	전자공학 세미나		A 디지털공학 및 실습	A 전자기학	A 반도체공학	B RF기초 및 회로설계		C 시스템 네트워크
		C 기초 전자공학	A 창의적공학 설계	A 회로이론1	A 회로이론2	B HDL및실습	B FPGA 설계실습	C 디지털 시스템설계	C 차세대 디스플레이
					A 전자회로 및 실습 1	A 전자회로 및 실습2		C 지능형 반도체개론	
								전자공학 ESP	전자공학 ESP
								종합설계 기획	종합설계1
	② 임베디드 SW능력	프로그래밍 언어	프로그래밍 언어	A 데이터 구조론	A 컴퓨터 구조론	B 마이크로 프로세서1	C 마이크로 프로세서2	C 안드로이드 프로그래밍	C 고급 머신러닝 실습
				A 객체지향 언어	A 파이썬	C 머신러닝 실습	C 리눅스 시스템설계		
	③ 신호 및 데이터 처리능력			A 신호 및 시스템	A 머신러닝의 이해	B 네트워크 기초	B 인터넷 프로토콜 설계		
		선형대수학	공학수학1	A 통신이론		B 디지털 통신1	B 디지털 통신2		B 이동통신 공학
					확률및 통계학	A 디지털 신호처리	B 영상신호 처리	C 멀티미디어 그래픽 실습	



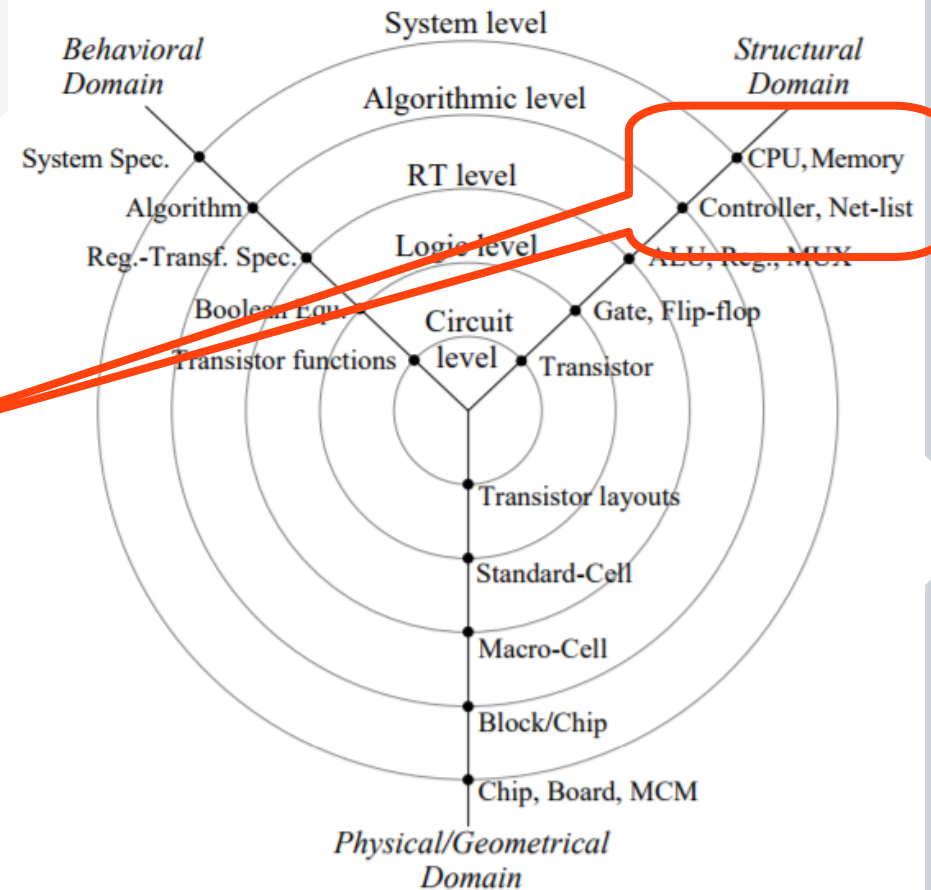
2024학년도 전자공학부 전자공학전공

학년/학기	1-1	1-2	2-1	2-2	3-1	3-2	4-1	4-2
이수구분/단계	전공기초 및 MSC				전공핵심		전공심화	
전공능력								
① 전자회로 설계능력	전자공학 세미나	디지털공학 및 실습	전자기학	반도체공학	디지털 회로 설계	시스템 네트워크		
	기초 전자공학	창의적공학 설계	회로이론1	회로이론2	HDL 및 VHDL 설계	디지털 시스템설계	차세대 디스플레이	지능형 반도체개론
			전자회로 및 실습	전자회로 및 실습2	전자회로 및 실습	전자회로 및 실습	전자공학 ESP	
② 임베디드 SW능력	프로그래밍 언어	프로그래밍 언어	데이터 구조론	컴퓨터 구조론	마이크로 프로세서1	마이크로 프로세서2	안드로이드 프로그래밍	고급 머신러닝 실습
			객체지향 언어	파이썬	머신러닝 실습	리눅스 시스템설계		
③ 신호 및 데이터 처리능력	선형대수학	공업수학1	통신이론	통신이론	디지털 통신1	디지털 통신2	이동통신 공학	
			신호 및 시스템	머신러닝의 이해	네트워크 기초	인터넷 프로토콜 설계		
			확률 및 통계학	디지털 신호처리	영상신호 처리	DSP시스템 및 실습	멀티미디어 프로그래밍 실습	



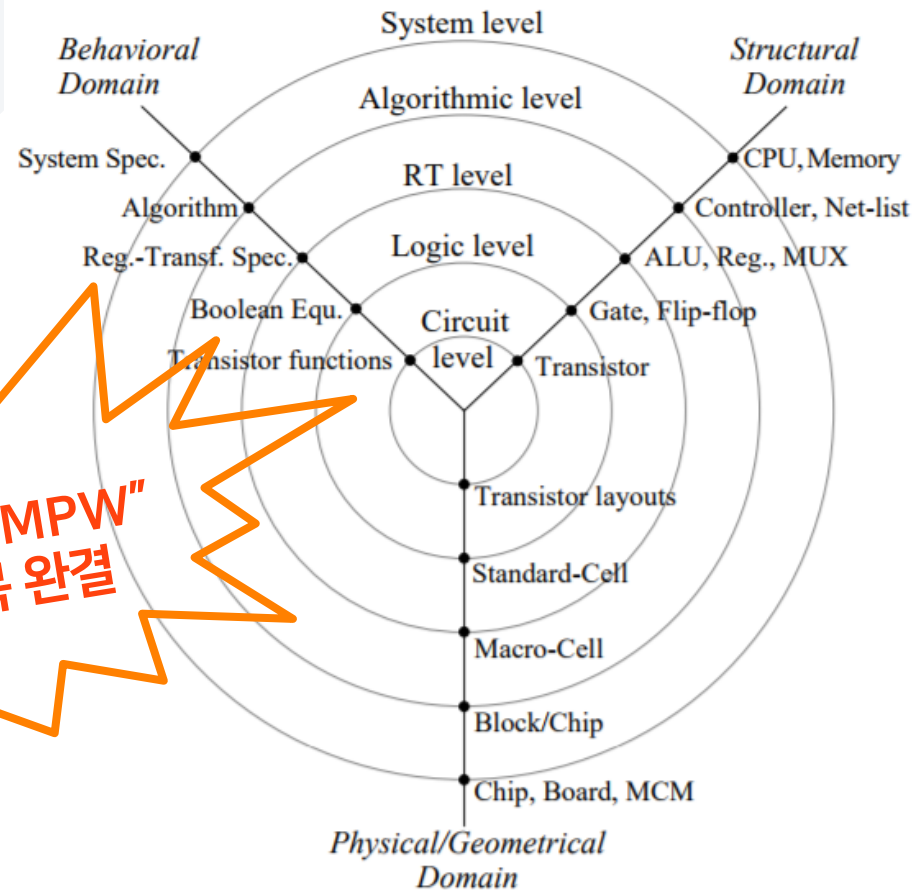
2024학년도 전자공학부 전자공학전공

학년/학기	1-1	1-2	2-1	2-2	3-1	3-2	4-1	4-2	
이수구분/단계	전공기초 및 MSC				전공핵심		전공심화		
전공능력									
① 전자회로 설계능력	전자공학 세미나		A 디지털공학 및 실습	A 전자기학	A 반도체공학	B RF기초 및 회로설계		C 시스템 네트워크	
	C 기초 전자공학	A 창의적공학 설계	A 회로이론1	A 회로이론2	B HDL및실습	B FPGA 설계실습	C 디지털 시스템설계	C 차세대 디스플레이	
			A 전자회로 및 실습 1	A 전자회로 및 실습 2	A 전자회로 및 실습 2		C 지능형 반도체개론		
							전자공학 ESP	C 전자공학 ESP	
							종합설계 기획	종합설계1	종합설계2
② 임베디드 SW능력	프로그래밍 언어	프로그래밍 언어	A 데이터 구조론	A 컴퓨터 구조론	B 마이크로 프로세서1	C 마이크로 프로세서2	C 안드로이드 프로그래밍	C 고급 머신러닝 실습	
			A 객체지향 언어	A 파이썬	C 머신러닝 실습	C 리눅스 시스템설계			
			A 신호 및 시스템	A 머신러닝의 이해	B 네트워크 기초	B 인터넷 프로토콜 설계			
③ 신호 및 데이터 처리능력	선형대수학	공업수학1	A 통신이론	B 디지털 통신1	B 디지털 통신2			B 이동통신 공학	
			확률및 통계학	A 디지털 신호처리	B 영상신호 처리	C 멀티미디어 프로그래밍 실습			
					B DSP시스템 및 실습				



2024학년도 전자공학부 전공공학전공

학년/학기		1-1	1-2	2-1	2-2	3-1	3-2	4-1	4-2	
이수구분/단계		전공기초 및 MSC				전공핵심		전공심화		
전공능력	① 전자회로 설계능력	전자공학 세미나		A 디지털공학 및 실습	A 전자기학	A 반도체공학	B RF기초 및 회로설계		C 시스템 네트워크	
		C 기초 전자공학	A 창의적공학 설계	A 회로이론1	A 회로이론2	B HDL및실습	B FPGA 설계실습	C 디지털 시스템설계	C 차세대 디스플레이	
					A 전자회로 및 실습 1	A 전자회로 및 실습2		전자공학 ESP	C 지능형 반도체개론	
									전자공학 ESP	
								종합설계 기획	종합설계1	종합설계2
	② 임베디드 SW능력	프로그래밍 언어	프로그래밍 언어	A 데이터 구조론	A 컴퓨터 구조론	B 마이크로 프로세서1	C 마이크로 프로세서2	C 안드로이드 프로그래밍	C 고급 머신러닝 실습	
				A 객체지향 언어	A 파이썬	C 머신러닝 실습	C 리눅스 시스템설계			
③ 신호 및 데이터 처리능력		선형대수학	공업수학1	A 통신이론	A 확률및 통계학	A 디지털 신호처리	B 영상신호 처리	C DSP시스템 및 실습		



"My Chip MPW" = 전 과목 완결

반도체 설계 교육의 애로점

반도체 설계 교육 과정에서 과목간 연속성

- 학과 이수 체계 vs 반도체 설계 플로우

설계 도구의 확보 / 실습 환경

- 고가의 도구 사용료
- 공정 자료의 비공개
- 공정이 포함된 실습의 고 비용

PDK (Process Design Kit)

■ Design Rule, Design Guide & Tutorial

■ Model Parameter

- ✓ Devices: CMOS, Sub-BJT, Diode, Resistor, Capacitor
- ✓ DC, Capacitance, Temperature Parameter (25°C ~ 115°C) Corner Model (MOSFET)
- ✓ (미제공) RF Parameter, Noise Parameter

■ Rule Decks

- ✓ Calibre (DRC, LVS) @ Hardware-Description Language 및 Rule Deck 보완 중

■ Pcells, GPIO

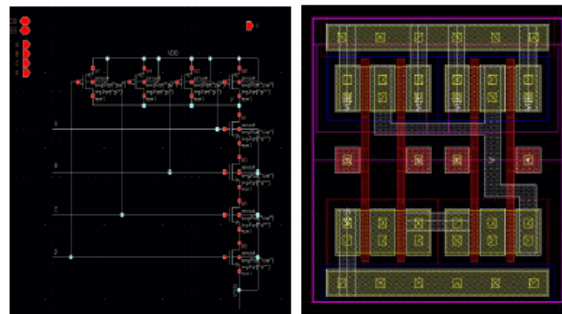
- ✓ Pcell: CMOS, BJT, Diode, Cap, Res, Log (AND2,3,4, NOR2,3,4)
- ✓ Analog/Digital IC용 IO-PAD 9종 *ESD0 칩 손상 방지를 위해 PGIO사용 권장

■ (Standard Cell) → 구축 예정

Algorithms
Hardware-Description Language

오픈 - 소스

〈Pcell 예: nand4〉



〈MOSIS 제공 Standard Cell〉

The IIT cell library has been developed to provide a free, non-proprietary library for MOSIS SCMOs rules.

It supports both commercial and public domain EDA tools.

The library is available for free upon request from <http://vlsi.ece.iit.edu/scells>

Supported MOSIS Technologies

AMI 0.5μm (includes pads)
AMI 0.35μm (includes pads)
TSMC 0.25μm
TSMC 0.18μm

List of Cells

AND2X1	INVX4
AND2X2	INVX8
AOI21X1	LATCH
AOI22X1	MUX2X1
BUF2X1	NAND2X1
BUF4	NAND3X1
CLKBUF1	NOR2X1
CLKBUF2	NOR3X1
CLKBUF3	OAI21X1
DFFNEX1	OAI22X1
DFFPX1	OR2X1
DFFSR	OR2X2
FAX1	TBUF1
FILL	TBUF2
HAX1	XNOR2X1
INVX1	XOR2X1
INVX2	

오픈 - 소스 EDA 도구 활용 디자인 킷

- 시스템 수준 모델링 / 자료 시각화

SystemC/C++, GNU Science Lib, Python, Simple Directmedia Layer

- 시뮬레이션 도구

ngSPICE, iVerilog, Verilator

- RTL 합성 과 자동 배치배선

Yosys, Graywolf, Qrouter

- 레이아웃 편집 및 GDS 생성

Magic, KLayout

- 사인 - 오프

DRC, LVS, 적층 비아 검사

- Co-Simulation & Co-Emulation/ 칩 테스트

FPGA(Gowin), Arduino/Raspberry Pi

오픈 - 소스 EDA 도구 활용 디자인 킷

- 시스템 수준 모델링 / 자료 시각화
SystemC/C++, GNU Science Lib, Python, Simple Directmedia Layer
- 시뮬레이션 도구
ngSPICE, iVerilog, Verilator
- RTL 합성 과 자동 배치배선
Yosys, Graywolf, Qrouter
- 레이아웃 편집 및 GDS 생성
Magic, KLayout
- 사인 - 오프
DRC, LVS, 적층 비아 검사
- Co-Simulation & Co-Emulation/ 칩 테스트
FPGA(Gowin), Arduino/Raspberry Pi

"My Chip MPW"
전과목 완결

Structural Domain

Behavioral Domain

Processor

Register
ALU

Leaf Cell

Transistor

Mask

Cell Placement

Module Placement

Chip Floorplan

Finite State Machine

Module Description

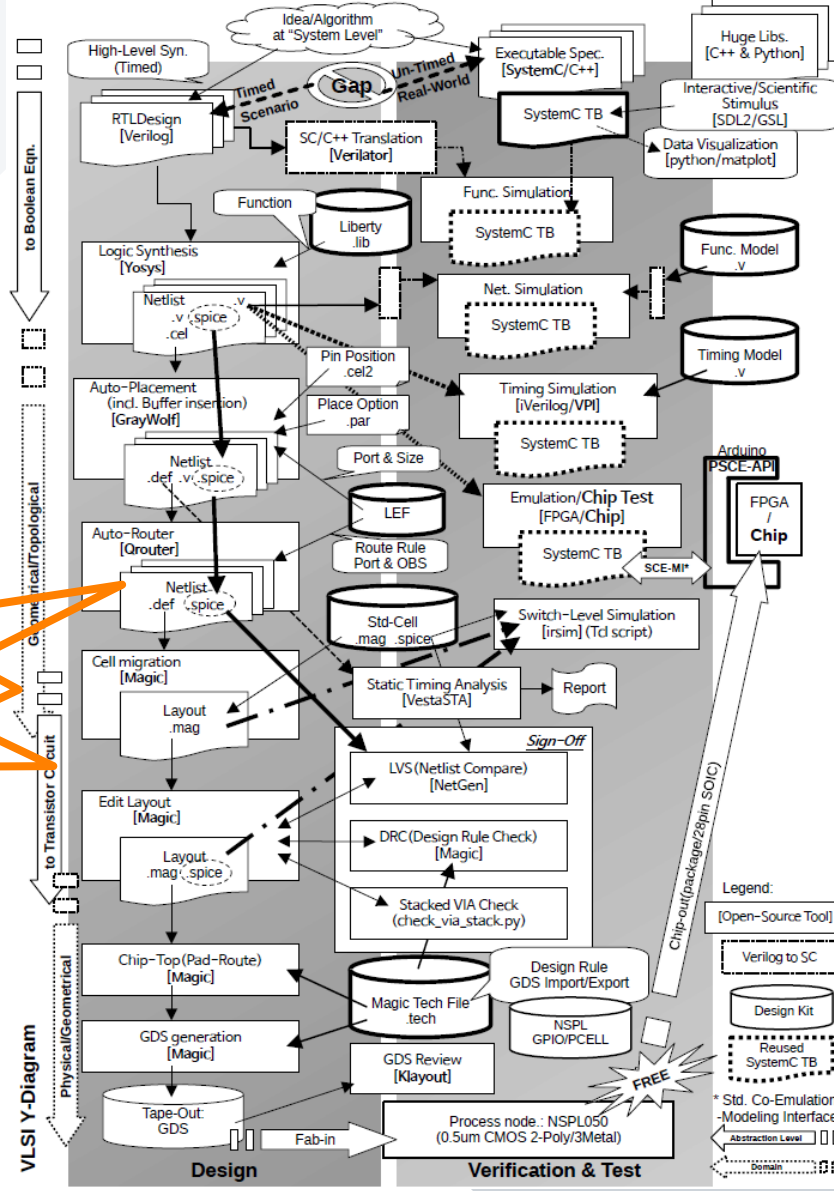
Boolean Equation

Y Chart

Geometrical Layout Domain



체계적 학습



NSPL 0.5u CMOS Std-Cell DK

Qflow: ETRI 0.5um CMOS 공정 추가에 필요한 파일들 [참조]

- 표준 셀 레이아웃 /NSPL 0.5um CMOS 디자인 룰 (VIA1, VIA2 주의)
- 합성기용 리버티 라이브러리 (지연 특성 무효)
- P&R 용 LEF 및 파라미터 파일 (Magic 테크 파일 포함)
- 네트 시뮬레이션 용 베릴로그 모델 (지연 특성 무효)
- LVS 용 표준 셀 SPICE 네트리스트 (트랜지스터 수준 LVS)

공정 특이 사항 점검 및 툴 플로우 빈 곳 채우기

- VIA1 과 VIA2 의 크기 확대 (DRC), 금지된 적 VIA 검사
- 레이아웃 수동 수정 후 LVS 재검사
- 입출력 패드 라이브러리와 코어의 디자인 룰 검사 분리 (주의 :Poly2)

검증과 평가 방법 포함

- 오픈 - 소스 하드웨어 활용 (아두이노, 라즈베리 파이 ...)
- FPGA 에뮬레이터 (PSCE/Poorman's Standard Co-Emulator)

[참조]

- OSU(Oklahoma State Univ) VLSI Computer Architecture Research Group, <https://vlsiarch.ecen.okstate.edu/flows/> :SCMOS MOSIS AMI 0.5um (2-Poly, 3-Metal)

반도체 오픈 - 소스 운동 , 갑자기?



THE MOSIS SERVICE

오픈 - 소스 운동 : 소프트웨어 > 하드웨어 > 실리콘



FOSS 130nm Production PDK
github.com/google/skywater-pdk

efabless.com



TINY TAPEOUT

Free Silicon Conference 2024

FSiC2024

Free Silicon Conference

OpenROAD

Democratizing Hardware Design

The OpenROAD™ project attacks the barriers of Cost, Expertise and Uncertainty (i.e., Risk) that block the feasibility of hardware design in advanced technologies.

[Read More](#)

*Recommendations and Roadmap for
Open-Source EDA in Europe*
<https://fossi-foundation.org/resources/eu-roadmap>



Custodian of the **Free and Open Source Silicon** movement.

오픈 - 소스 반도체 설계 도구들

알고리즘 / RTL 에서 GDS 까지

"democratization of hardware design"

-- OpenROAD: Toward a Self-Driving, Open-Source Digital Layout Implementation Tool Chain

알고리즘 개발 (Un-Timed/Timed)



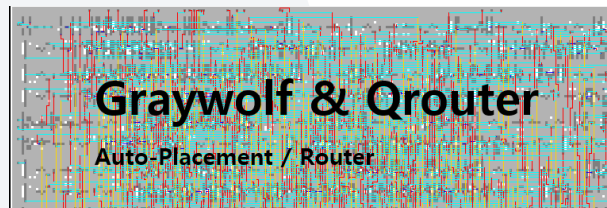
RTL 시뮬레이션 (Pre/Post-Sim)



RTL 합성 (Synthesis)



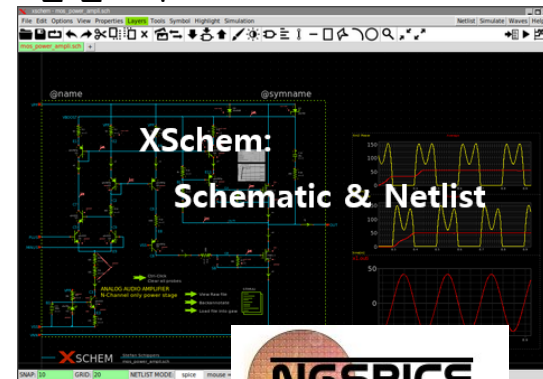
표준 - 셀 자동 배치 및 배선 (P&R)



레이아웃 편집, DRC, GDS 생성



표준 셀 제작



LVS (SPICE Netlist)

Netgen 1.5

Netgen version 1.5 netlist comparison (LVS)

표 1. 디자인 킷에 활용된 오픈 소스 도구 및 라이브러리 목록

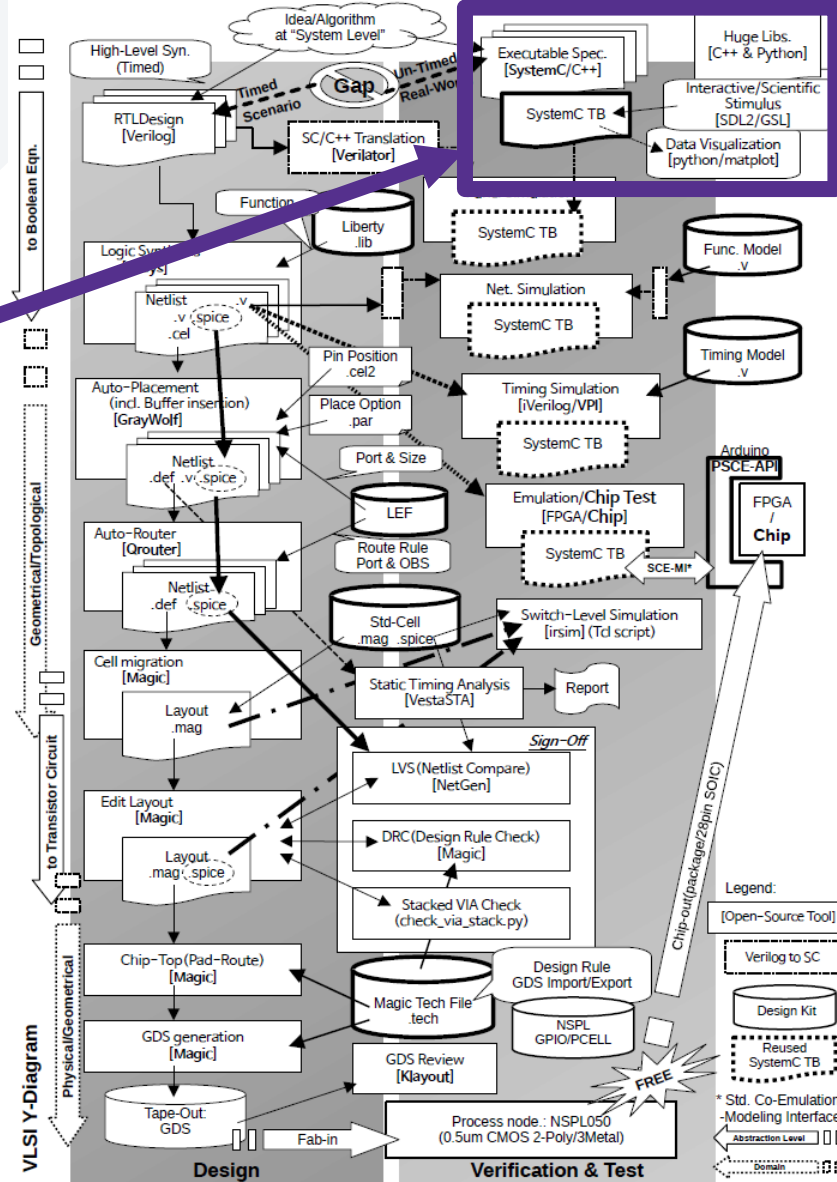
도구 명	용도
SystemC ^[9]	시스템 수준 모델링 C++ 클래스 라이브러리
GSL ^[10]	C/C++ 과학 라이브러리 (GNU Scientific Library) 시뮬레이션 데이터 생성 및 분석
SDL2 ^[11]	멀티미디어 C/C++ 라이브러리 (Simple Directmedia Layer). 대화형 테스트 벤치
python ^[12]	시뮬레이션 데이터 분석 및 시각화
iVerilog ^[13]	베릴로그 시뮬레이터
irsim ^[16]	스위치 레벨 시뮬레이터
ngSpice ^[15]	SPICE 회로 시뮬레이터
XSchem ^[18]	회로도 작성
Verilator ^[19]	베릴로그-C++ 변환기
Yosys ^[20]	RTL 베릴로그 합성기
GrayWolf ^[22]	자동 배치 배선기. 버퍼 삽입 기능 포함
Qrouter ^[23]	자동 배선기
Magic ^[24]	레이아웃 편집, DRC, GDS 생성
NetGen ^[25]	네트리스트 비교 방식 LVS
QFlow ^[26]	설계 플로우 관리 도구

표 2. 디자인 킷의 설계 자동화 용 표준-셀 구성과 파일 형식

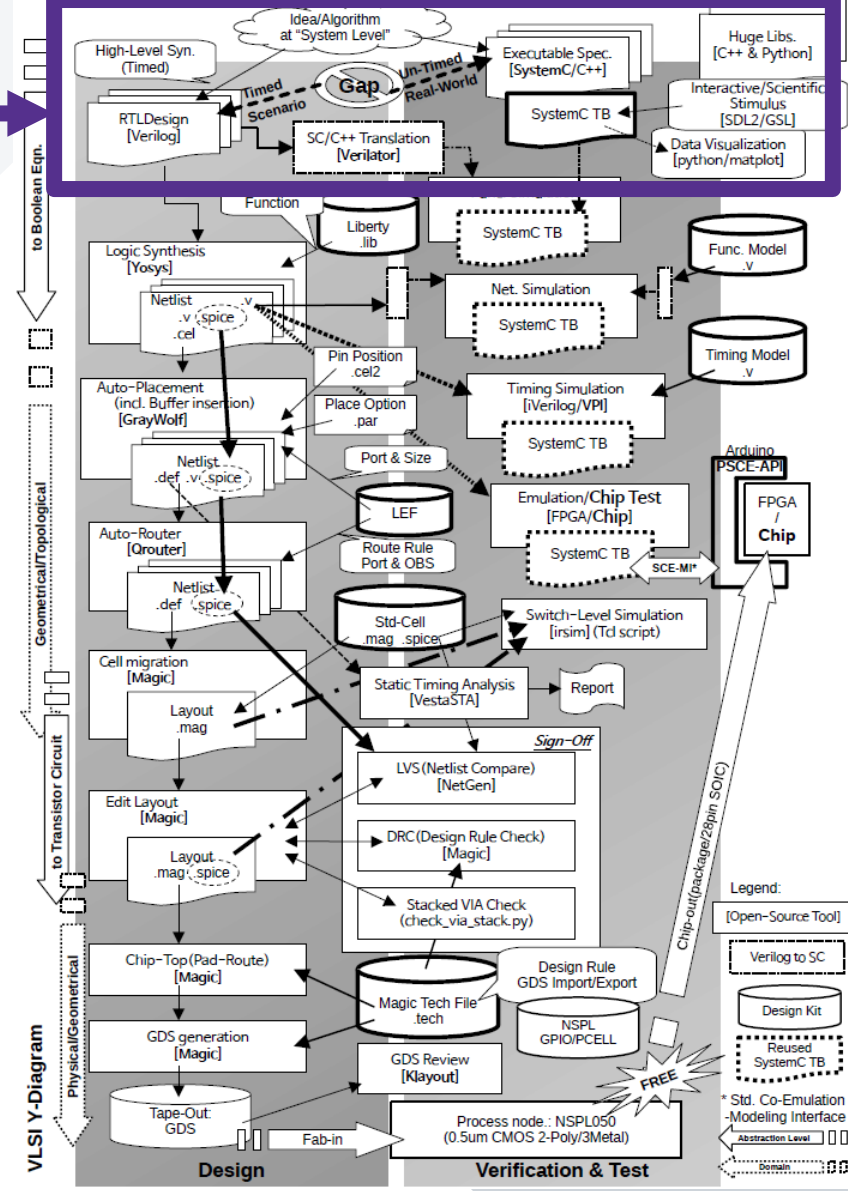
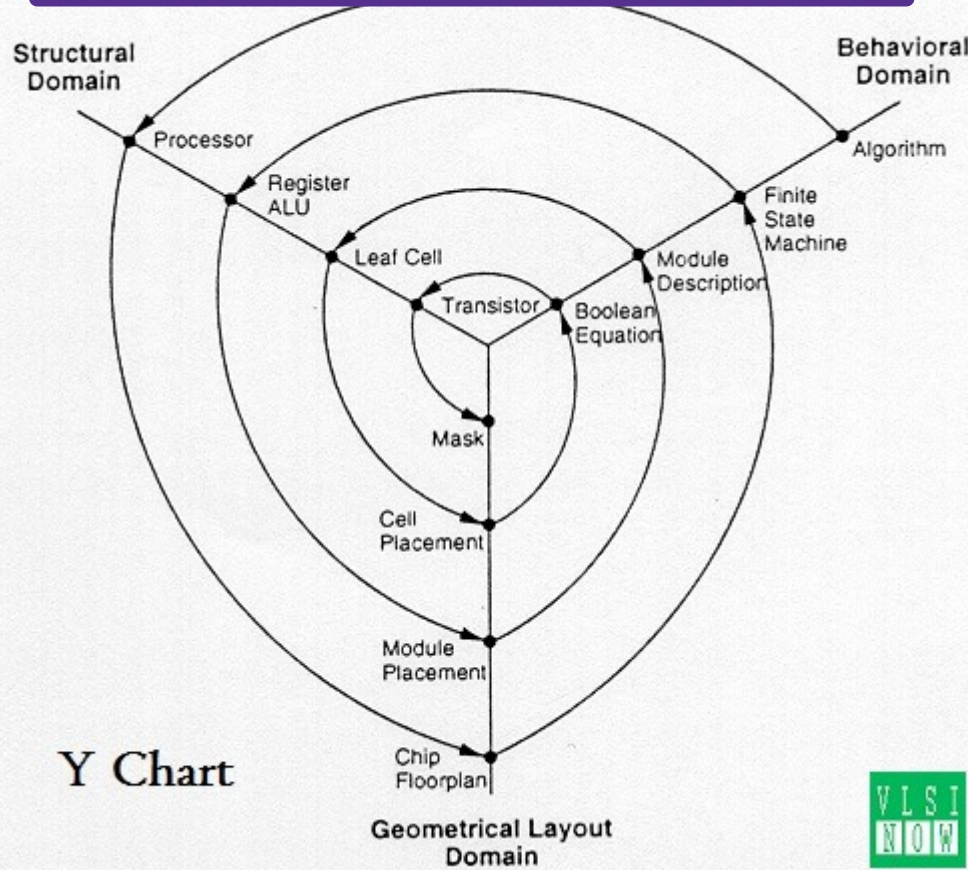
파일 명	형식
	관련 도구(용도)
etri050_stdcells.lib	리버티(liberty)
	RTL 합성(function)
etri050_stdcells.lef	LEF
	배치(PIN, SIZE)
	배선(PIN, OBS)
etri050_stdcells.sp	SPICE
	LVS
SCN3ME_SUBM. 30.ETRI.tech	Magic Tech.
	DRC, 회로 추출 GDS 들여오기/내보내기
etri05_stdcells.v	베릴로그
	기능 시뮬레이션 모형
etri05_stdcells_func.v	베릴로그
	타이밍 시뮬레이션 모형
*.mag	표준 셀 레이아웃

표 1. 디자인 킷에 활용된 오픈 소스 도구 및 라이브러리 목록

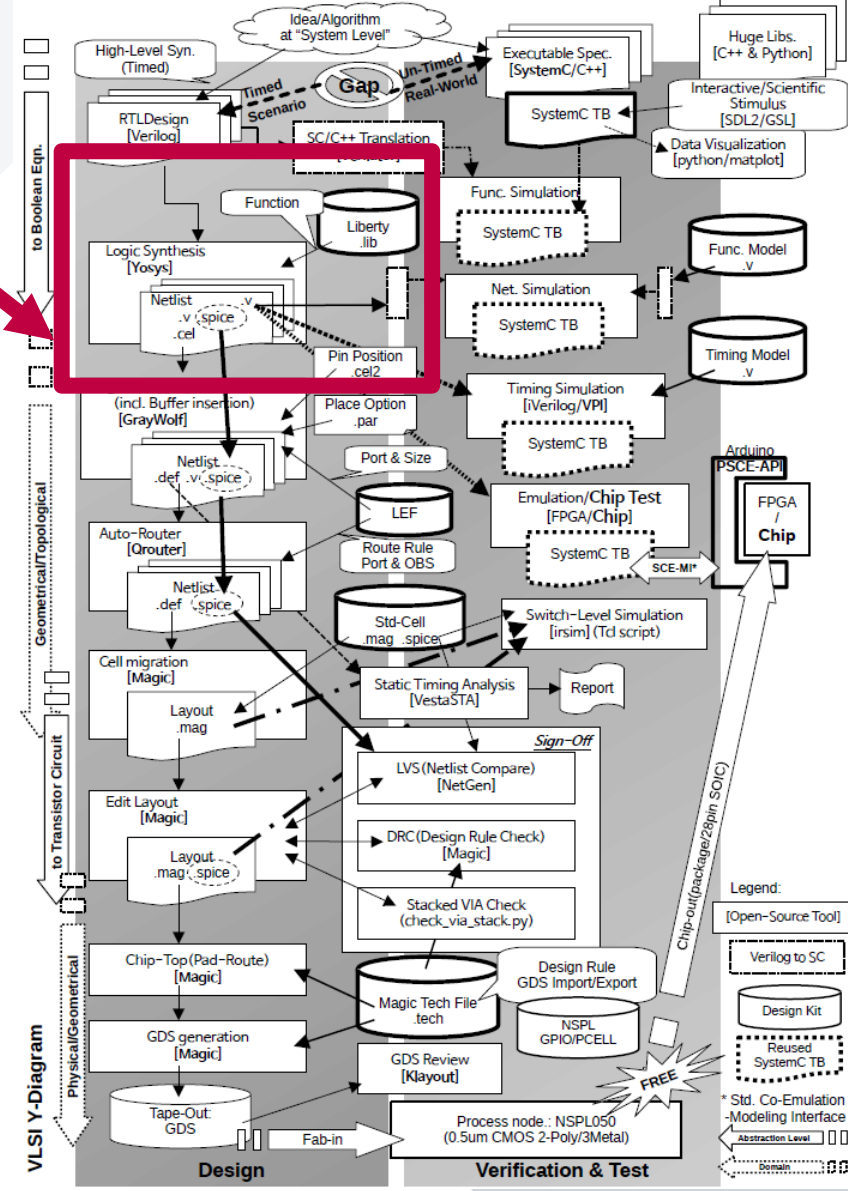
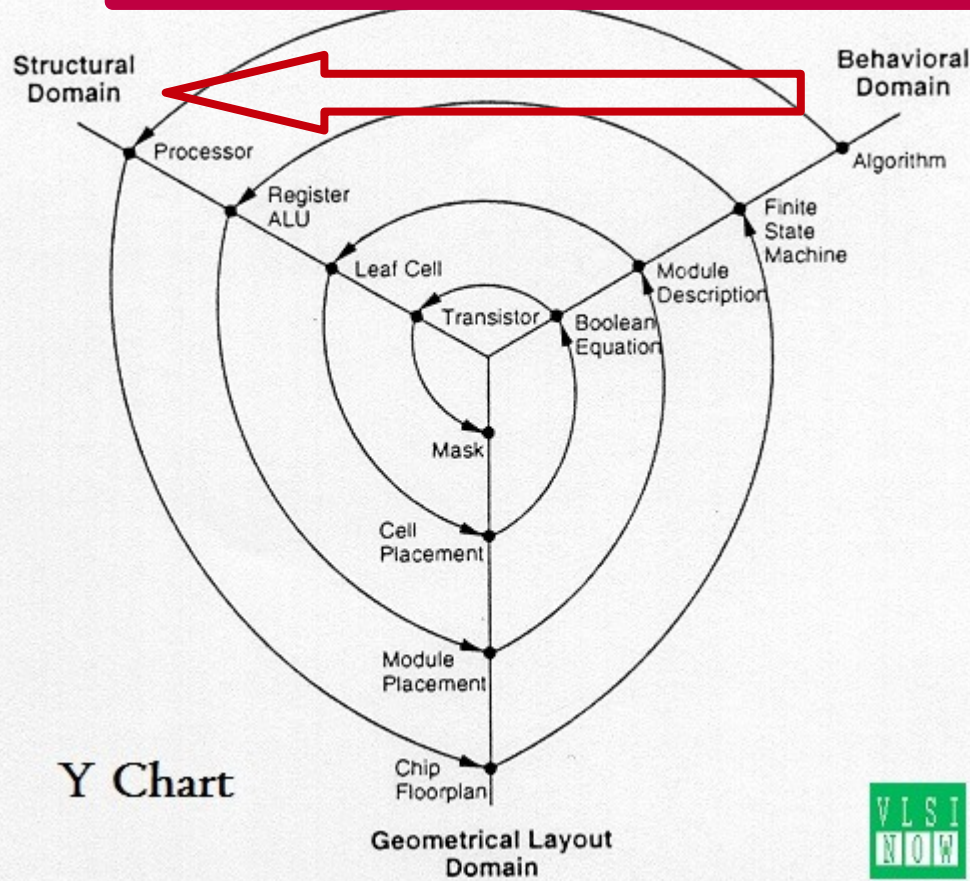
도구 명	용도
SystemC ^[9]	시스템 수준 모델링 C++ 클래스 라이브러리
GSL ^[10]	C/C++ 과학 라이브러리(GNU Scientific Library) 시뮬레이션 데이터 생성 및 분석
SDL2 ^[11]	멀티미디어 C/C++ 라이브러리(Simple Directmedia Layer). 대화형 테스트 벤치
python ^[12]	시뮬레이션 데이터 분석 및 시각화
Verilog ^[13]	배열로그 시뮬레이터
irsim ^[16]	스위치 레벨 시뮬레이터
ngSpice ^[15]	SPICE 회로 시뮬레이터
XSchem ^[18]	회로도 작성
Verilator ^[19]	베릴로그-C++ 변환기
Yosys ^[20]	RTL 베릴로그 합성기
GrayWolf ^[22]	자동 배치 배선기. 버퍼 삽입 기능 포함
Qrouter ^[23]	자동 배선기
Magic ^[24]	레이아웃 편집, DRC, GDS 생성
NetGen ^[25]	네트리스트 비교 방식 LVS
QFlow ^[26]	설계 플로우 관리 도구



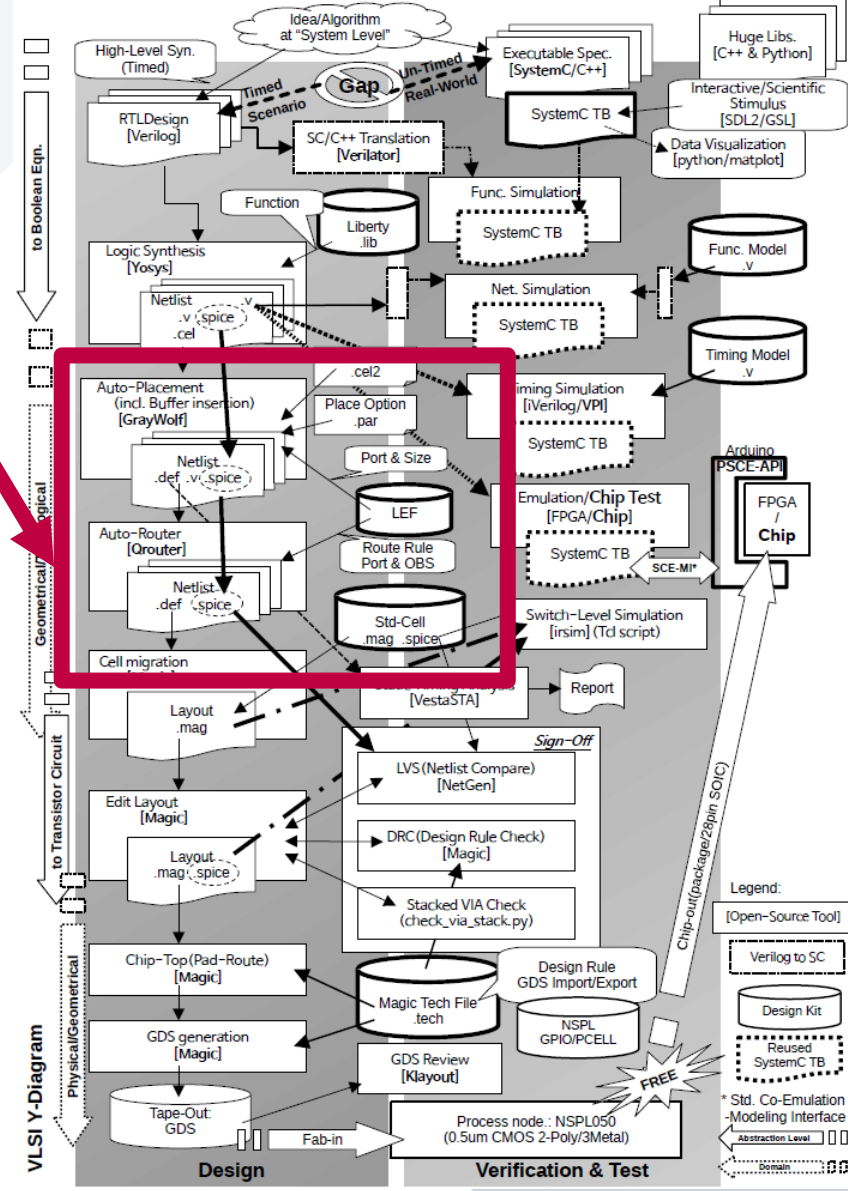
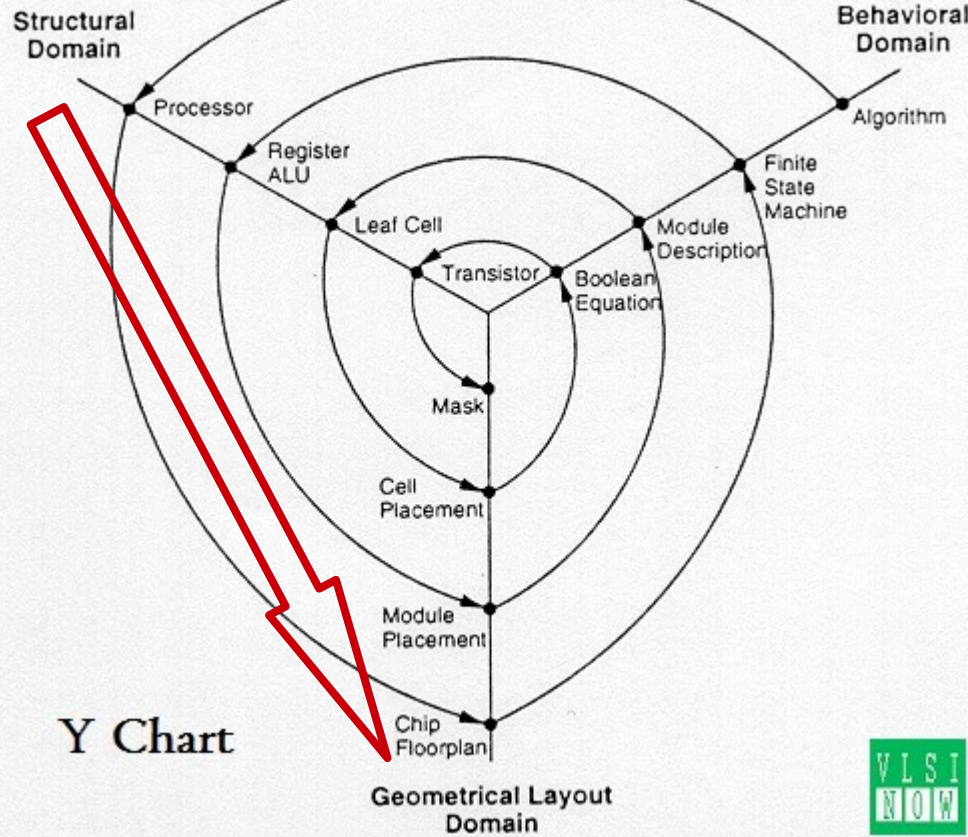
System Level/Algorithm (SystemC, GSL, SDL2, Python)



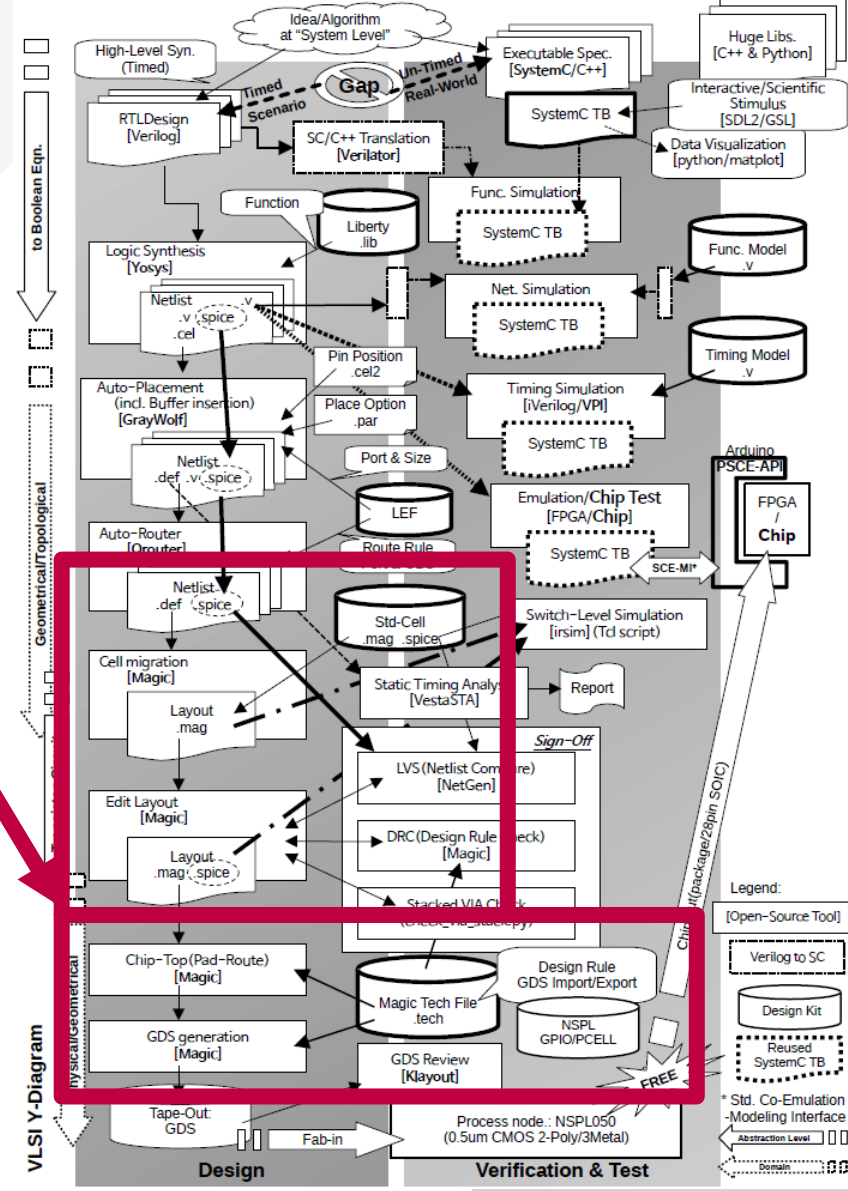
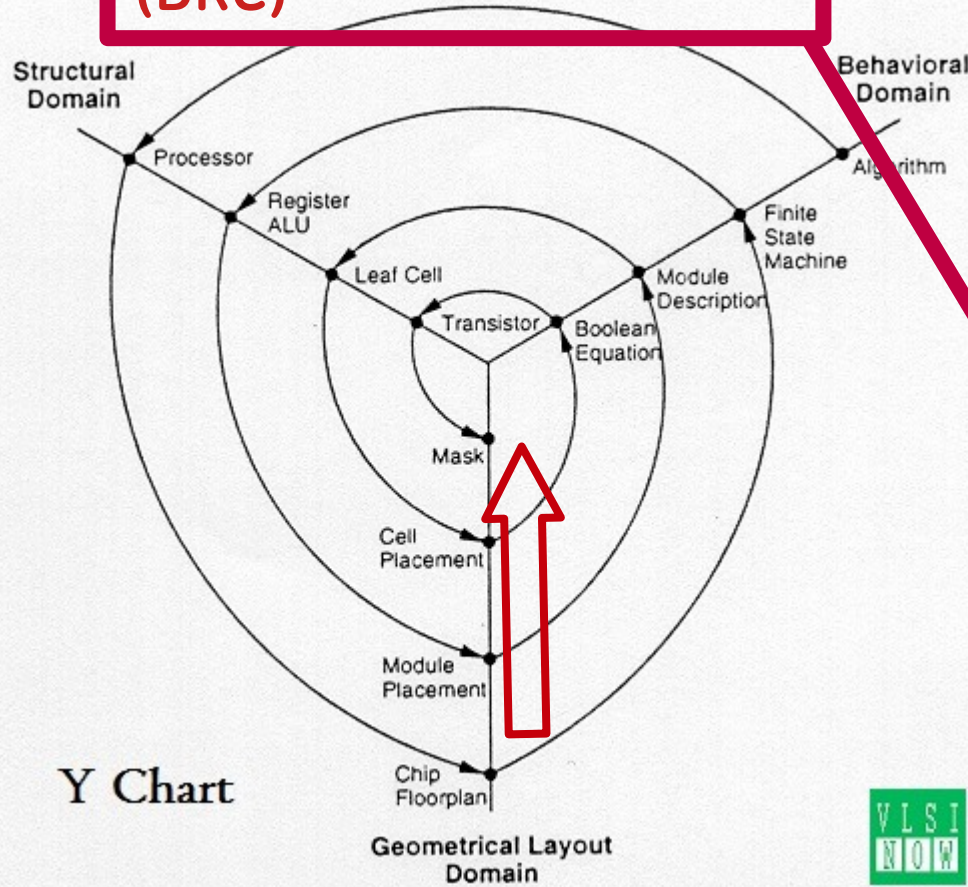
Yosys Synthesizer (HDL to Netlist)

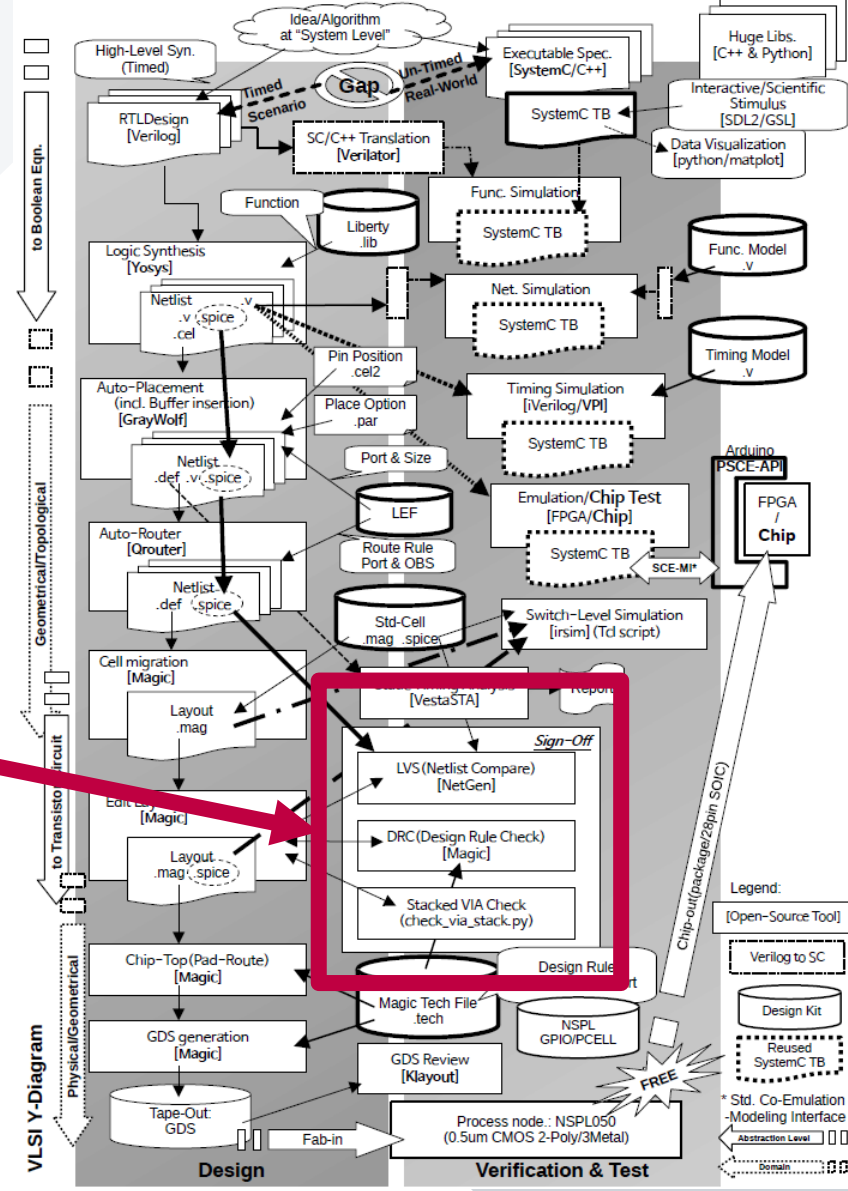
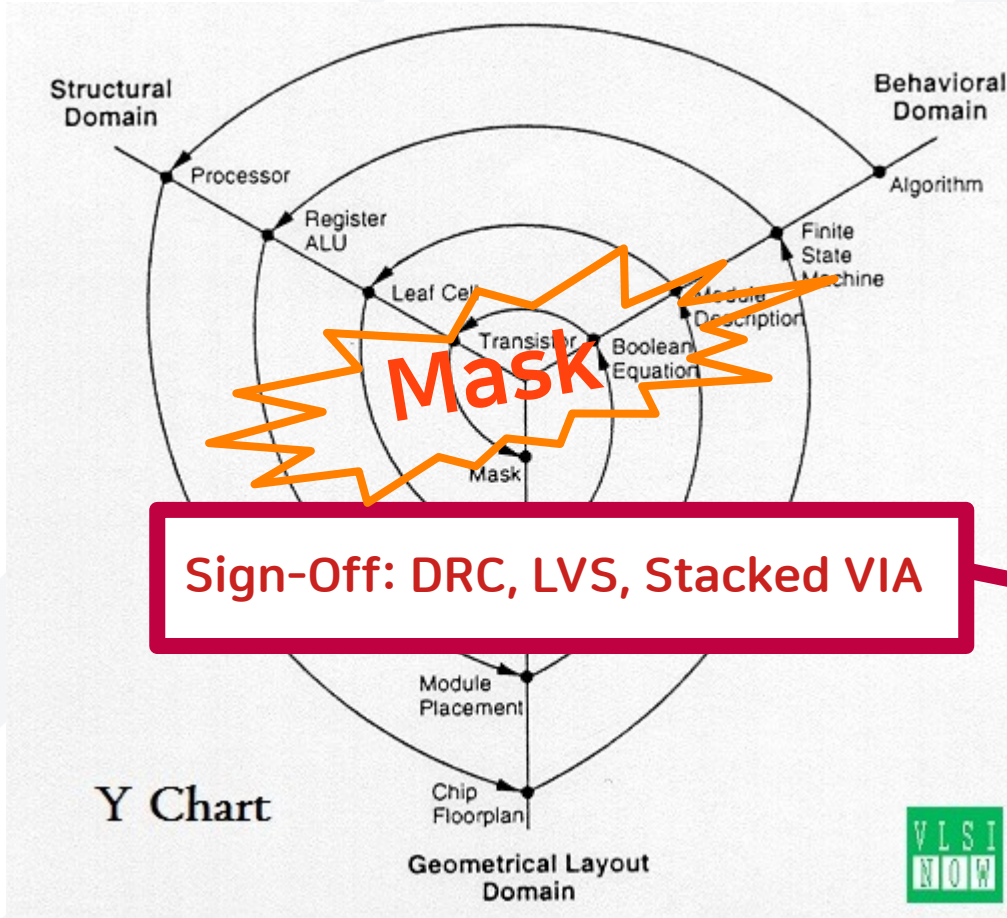


P&R: GrayWolf, QRouter (Routing Rule/DRC, Obstruction)



Layout: Magic, KLayout (DRC)



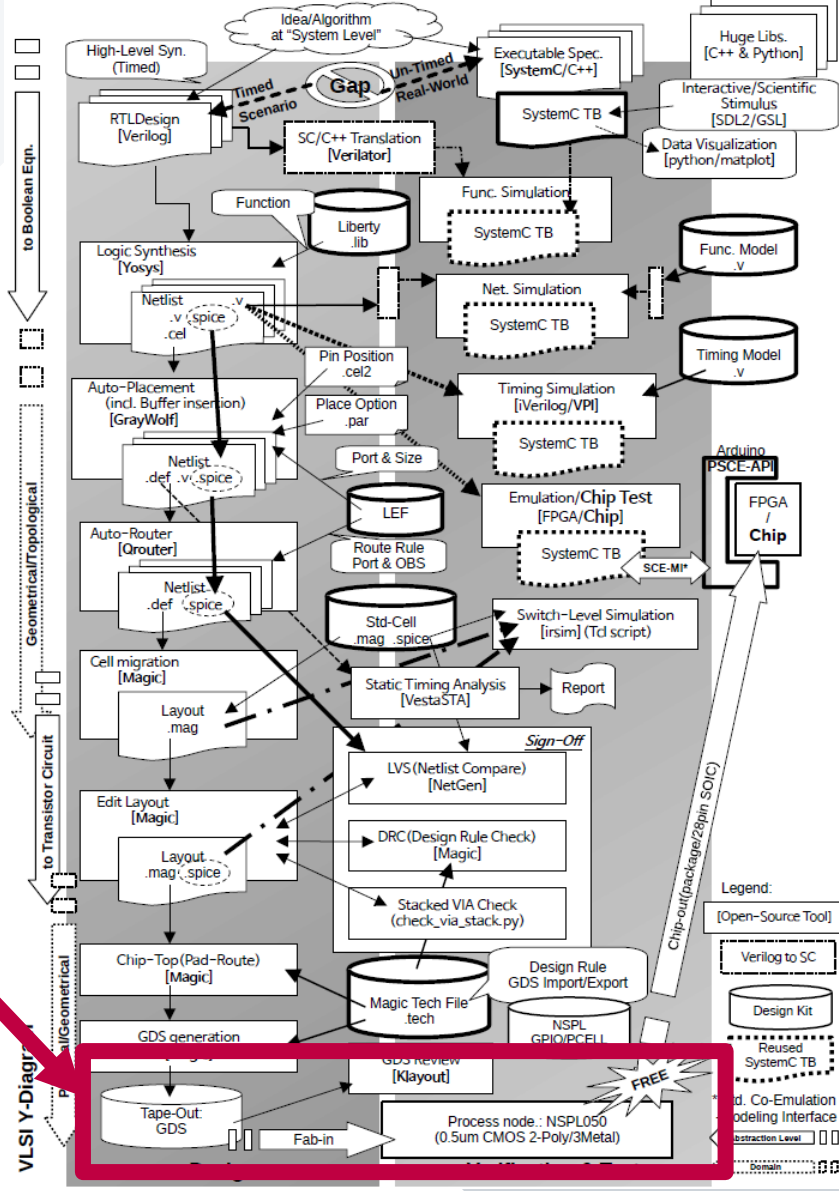


Structural Domain

Behavioral Domain



My Chip MPW Service
Chip Out /w 28Pin SOP



Structural Domain

Behavioral Domain

Processor

Register
ALU

Leaf Cell

Transistor

Module
Description

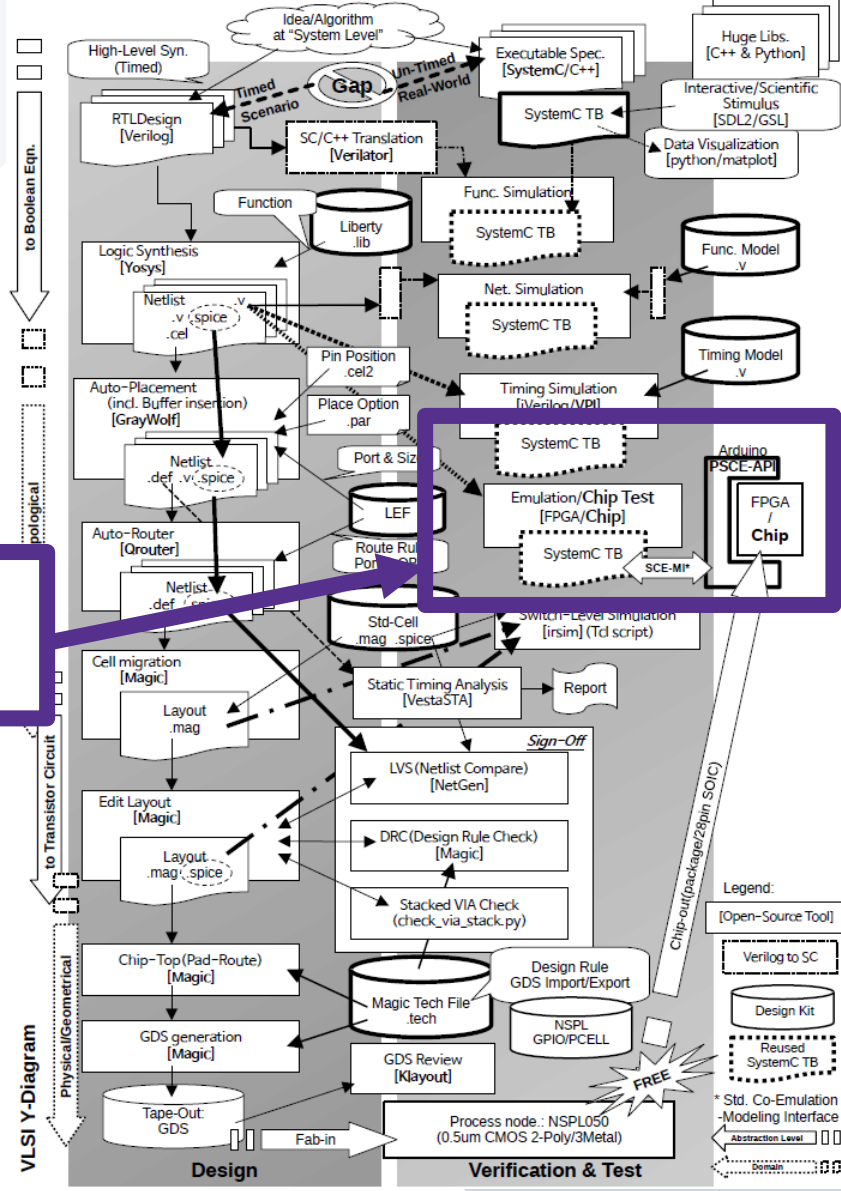
Boolean
Equation

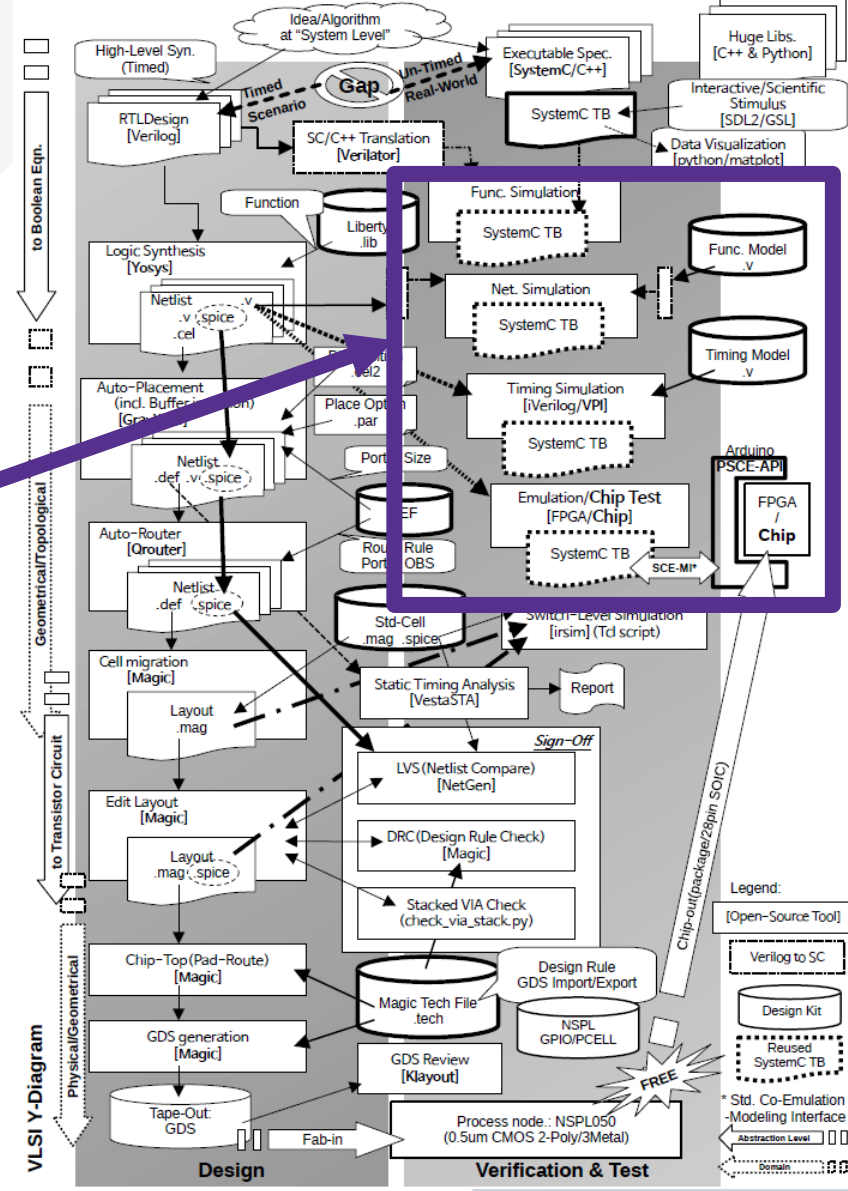
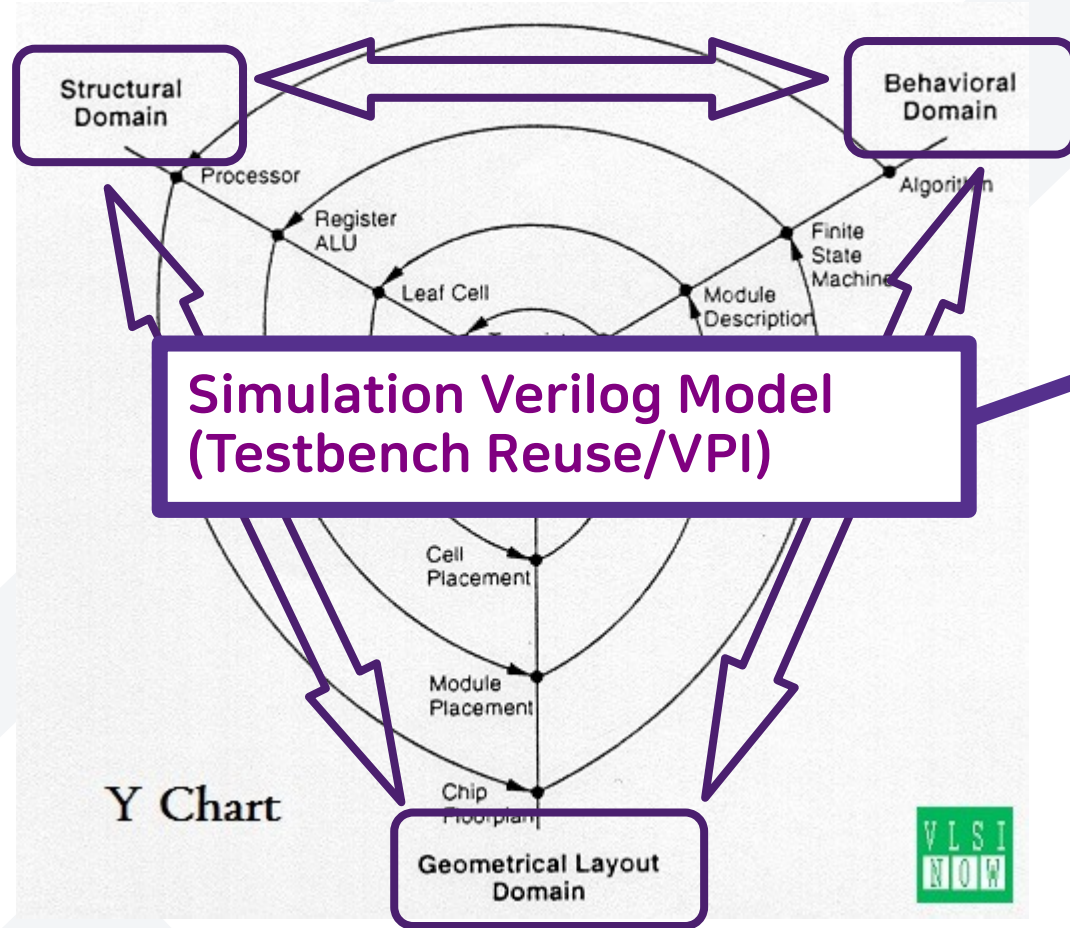
Finite
State
Machine

Chip Test
(SystemC Testbench Reuse)

Y Chart

Geometrical Layout
Domain





오픈 - 소스 EDA 도구 활용 디자인 킷 개발

- 시스템 수준 모델링 / 자료 시각화
SystemC/C++, GNU Science Lib, Python, Simple Directmedia Layer
- 시뮬레이션 도구
ngSPICE, iVerilog, Verilator
- RTL 합성 과 자동 배치배선
Yosys, Graywolf, Qrouter
- 레이아웃 편집 및 GDS 생성
Magic, KLayout
- 사인 - 오프
DRC, LVS, 적층 비아 검사
- Co-Simulation & Co-Emulation/ 칩 테스트
FPGA(Gowin), Arduino/Raspberry Pi



적절한 예제

디자인 킷 활용 예제 (1)

표준 셀 제작 (예 : D-F/F)

P&R Test
(LEF)

Magic

(LVS)

XSchem

Timing Simulation
(SPICE Model)

Function Sim.
(Verilog Model)

Spec.

(Schematic Entry)

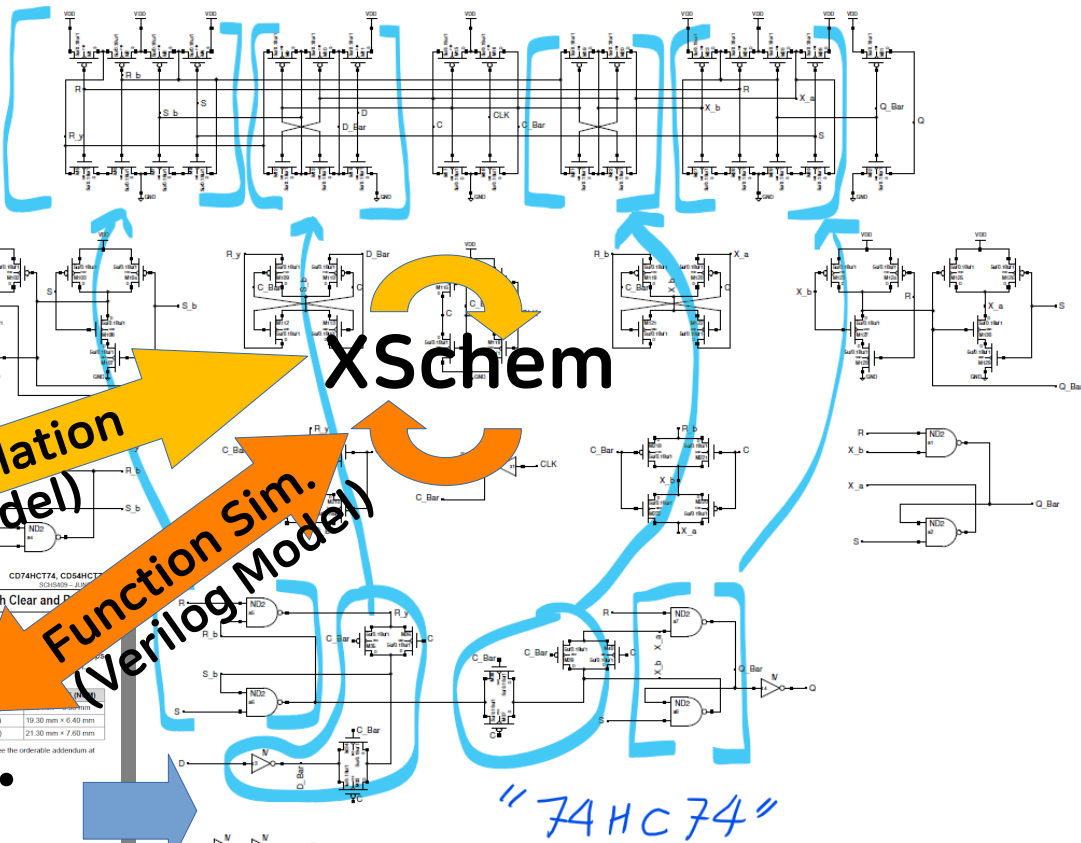
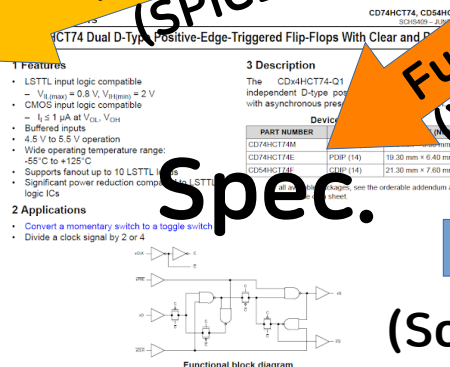
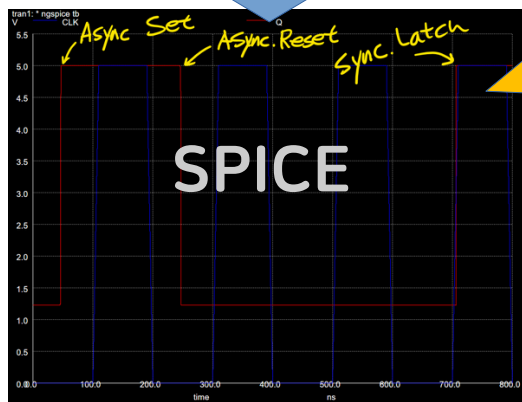
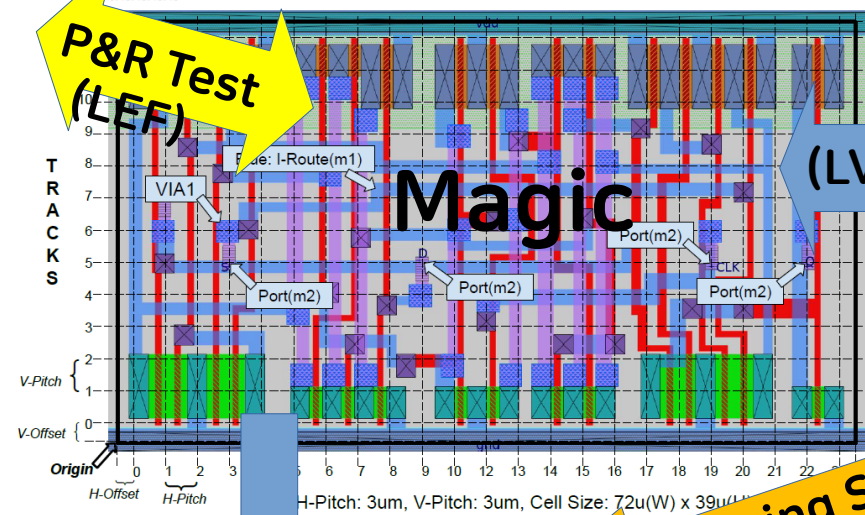


표 3. 디자인 킷의 표준-셀 종류

표준-셀	기능(리버티 형식 ^[29])
AND2X1, AND2X2	(A B)
AOI21X1	(! ((A B)+C))
AOI22X1	(! ((A B)+(C D)))
BUFX2,BUFX4	A
CLKBUF1,CLKBUF2	A
DFFNEGX1	ff() { …… }
DFFPOSX1	ff() { …… }
DFFSR	ff (P0002,P0003) { next_state : "D"; clocked_on : "CLK"; clear : "(!R)"; preset : "(!S)"; clear_preset_var1 : L; } "P0002"
INVX1,INVX2,INVX4	(!A)
MUX2X1	(! ((S A) + (!S B)))
NAND2X1,NAND2X1	(! (A B)),(! ((A B) C))
NOR2X1, NOR2X1	(! (A+B)), (! ((A+B)+C))
OAI21X1,OAI22X1	(! ((A+B) C)),(! ((A+B) (C+D)))
OR2X1, OR2X2	(A+B)

표 1. 디자인 킷에 활용된 오픈 소스 도구 및 라이브러리 목록

도구 명	용도
SystemC ^[9]	시스템 수준 모델링 C++ 클래스 라이브러리
GSL ^[10]	C/C++ 과학 라이브러리 (GNU Scientific Library) 시뮬레이션 데이터 생성 및 분석
SDL2 ^[11]	멀티미디어 C/C++ 라이브러리 (Simple Directmedia Layer). 대화형 테스트 벤치
python ^[12]	시뮬레이션 데이터 분석 및 시각화
iVerilog ^[13]	베릴로그 시뮬레이터
irsim ^[16]	스위치 레벨 시뮬레이터
ngSpice ^[15]	SPICE 회로 시뮬레이터
XSchem ^[18]	회로도 작성
Verilator ^[19]	베릴로그-C++ 변환기
Yosys ^[20]	RTL 베릴로그 합성기
GrayWolf ^[22]	자동 배치 배선기. 버퍼 삽입 기능 포함
Qrouter ^[23]	자동 배선기
Magic ^[24]	레이아웃 편집, DRC, GDS 생성
NetGen ^[25]	네트리스트 비교 방식 LVS
QFlow ^[26]	설계 플로우 관리 도구

표 2. 디자인 킷의 설계 자동화용 표준 셀 구성과 파일 형식

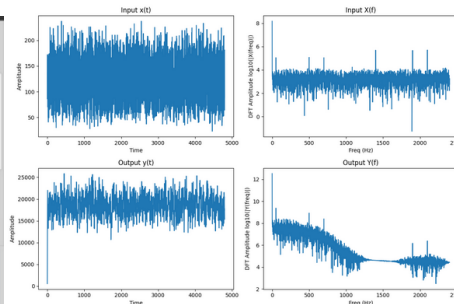
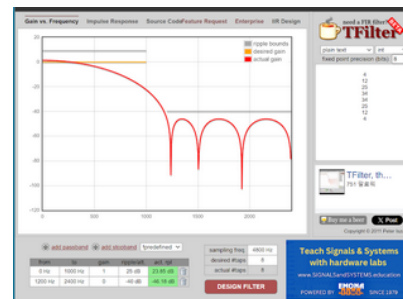
파일 명	형식
	관련 도구(용도)
etri050_stdcells.lib	리버티(liberty)
	RTL 합성(function)
etri050_stdcells.lef	LEF
	배치(PIN, SIZE)
	배선(PIN, OBS)
etri050_stdcells.sp	SPICE
	LVS
SCN3ME_SUBM. 30.ETRI.tech	Magic Tech.
	DRC, 회로 추출 GDS 들여오기/내보내기
etri05_stdcells.v	베릴로그
	기능 시뮬레이션 모형
etri05_stdcells_func.v	베릴로그
	타이밍 시뮬레이션 모형
*.mag	표준 셀 레이아웃

디자인 킷 활용 예제 (2)

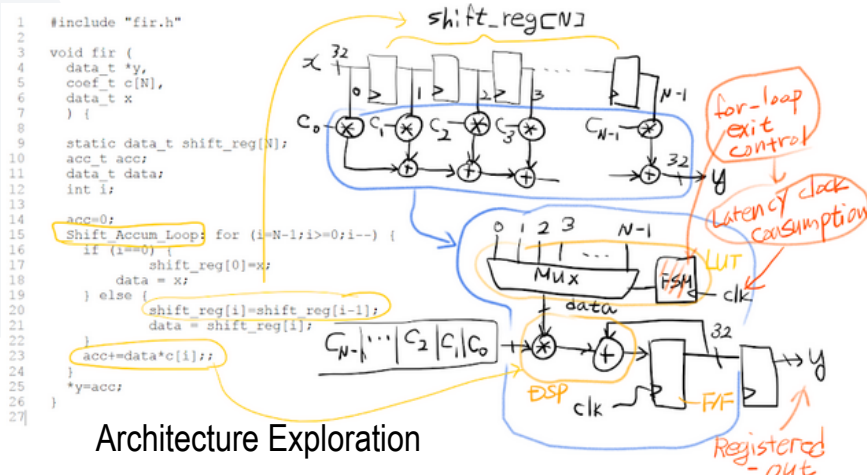
RTL-to-GDS

8-Tab FIR Filter

- 알고리즘 개발 : FIR 계수 (<http://t-filter.engineerjs.com>), 파이썬 활용
- 구조탐색 : 파이프라인 병렬처리 구조 (Array Processing Elements)
- Un-Timed C- 모형에서 RTL 로 변환하는 과정
- “내 칩 MPW” 요건에 맞추기 (칩 면적 , 입출력 핀 수 제한)
- 아두이노 +FPGA 에뮬레이션 / 칩 테스트



Test vector visualization



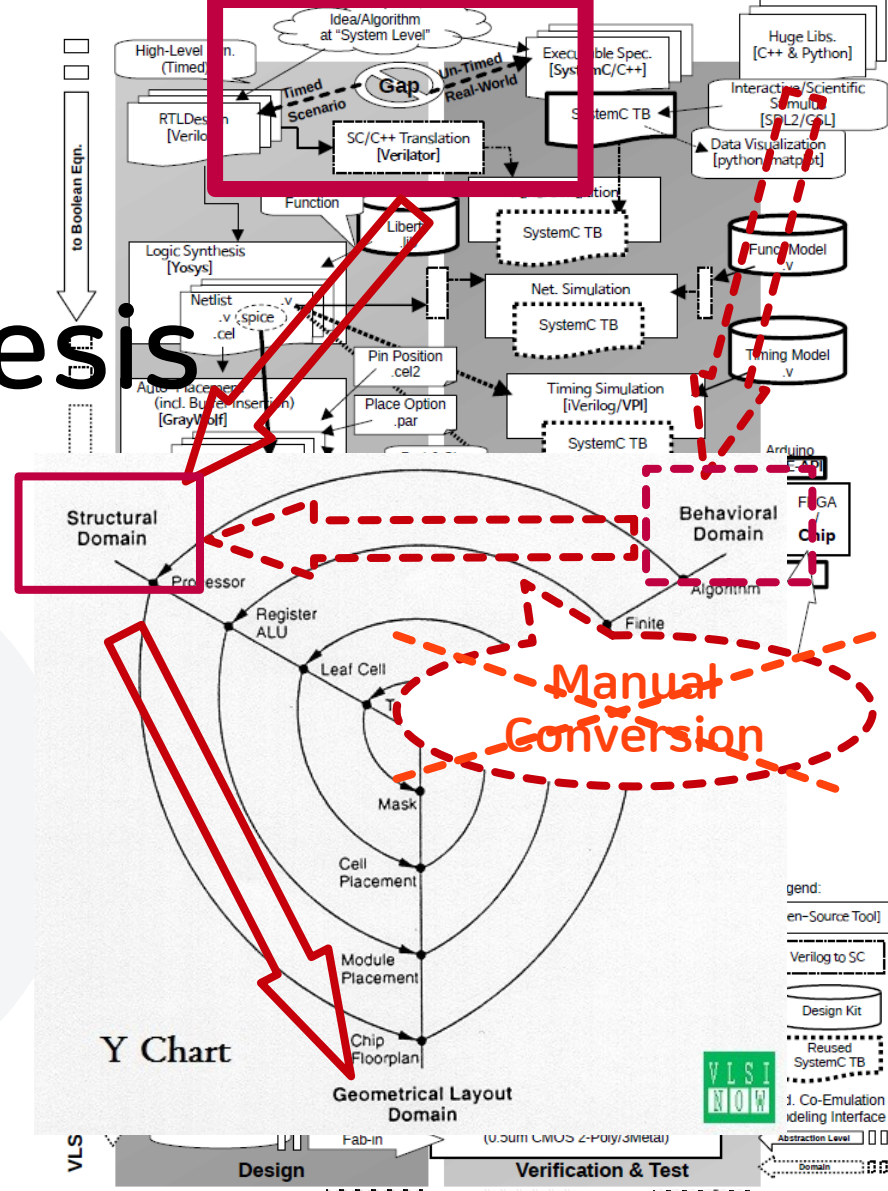
Architecture Exploration

디자인 킷 활용 예제 (3)

High Level Synthesis

Fixed-Point Square Root

- C++ 에서 제공근 구하는 알고리즘
- Fixed-Point Arithmetic 활용 하드웨어 최적화
- Un-Timed C++ 에서 RTL 로 합성 (Xilinx Vitis-HLS)
- “내 칩 MPW” 요건에 맞추기 (칩 면적 , 입출력 핀 수 제한)
- 아두이노 +FPGA 에뮬레이션 / 칩 테스트



칩 제작을 넘어서 ... 검증과 평가

칩 제작 전 “검증”

- 알고리즘 시뮬레이션 (Un-Timed): SystemC TB(Timed)
- 합성전 RTL 시뮬레이션 : Verilator+SystemC TB
- 합성 후 네트 시뮬레이션 : iVerilog+SystemC TB
- FPGA 에뮬레이션 : FPGA/Arduino+SystemC TB
- LVS, DRC, 겹침 VIA 검사

칩 출고 후 기능성 “평가”

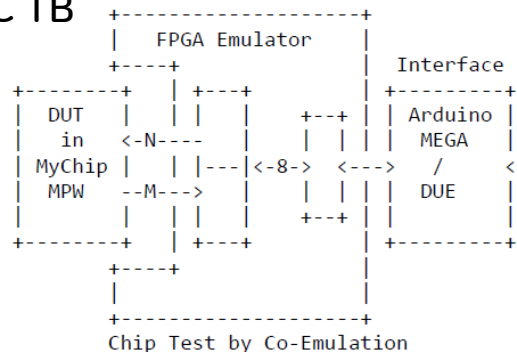
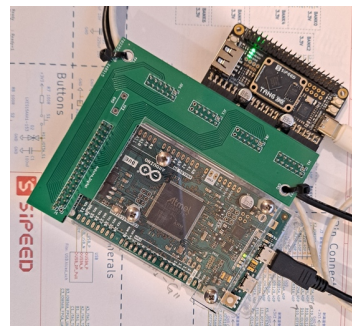
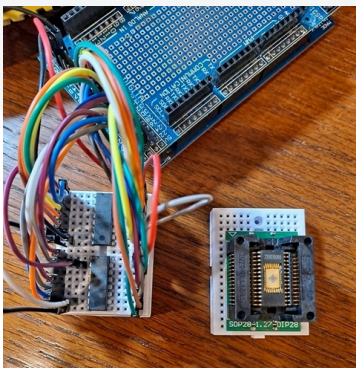
- 테스트 벤치 재사용 : FPGA/Arduino+SystemC TB
- 칩 테스트 장비 제작

FPGA

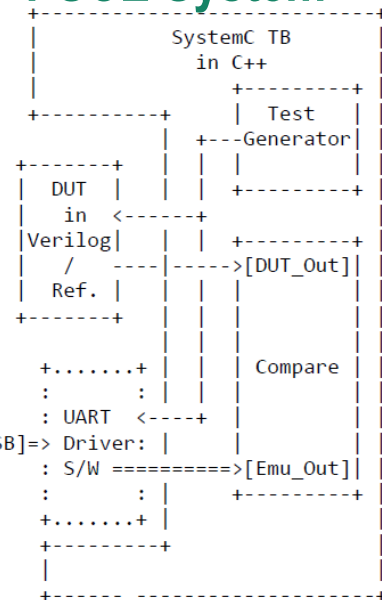
Arduino DUE

Raspberry PI

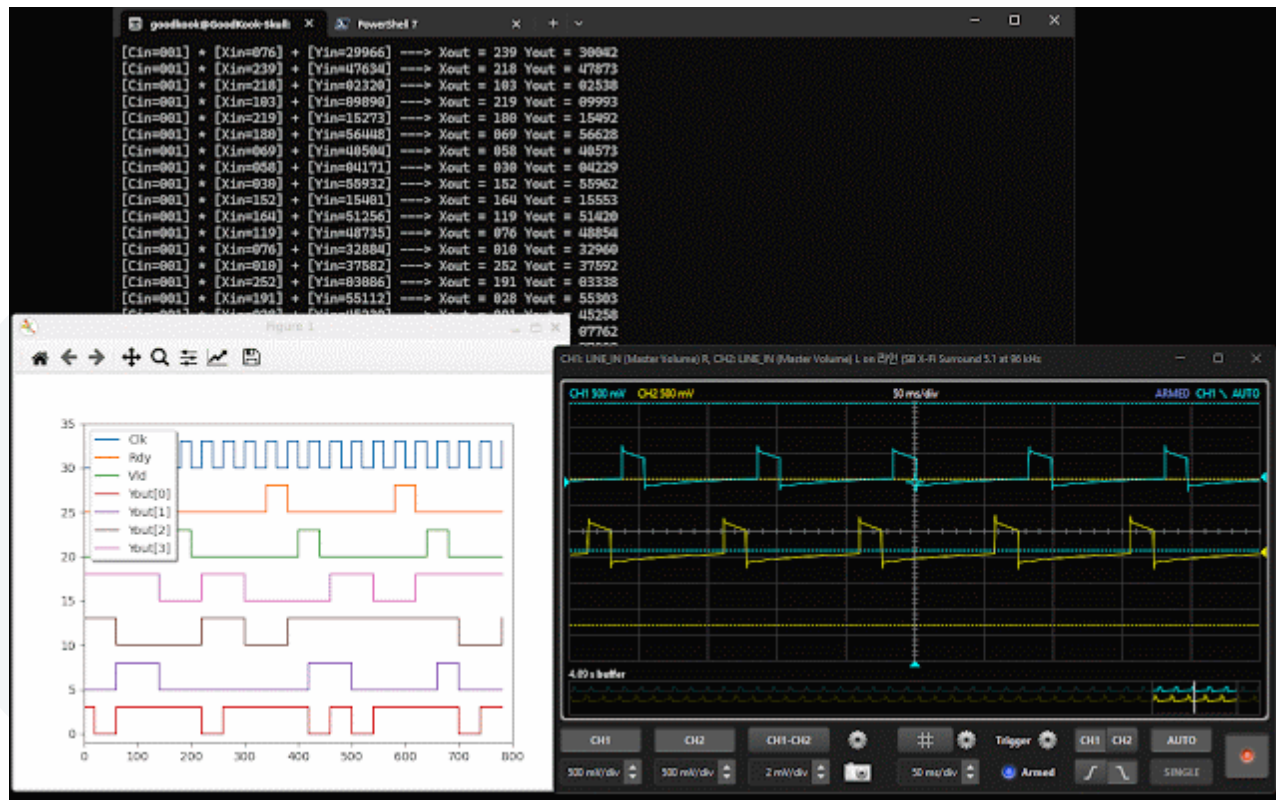
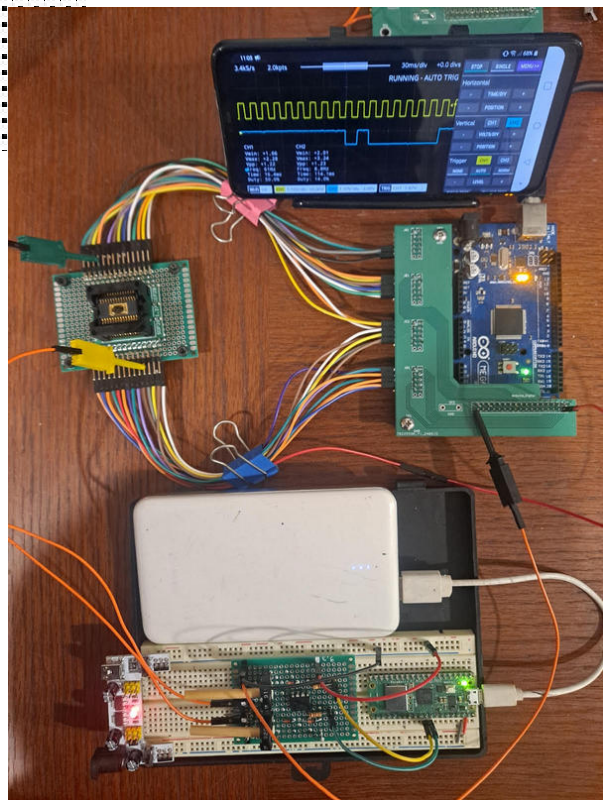
ESP32



GoodKook's PSCE System



칩 제작을 넘어서 ... 칩 테스트



SCE-MI

Standard Co-Emulation / Modeling Interface, <http://www.accelera.org>

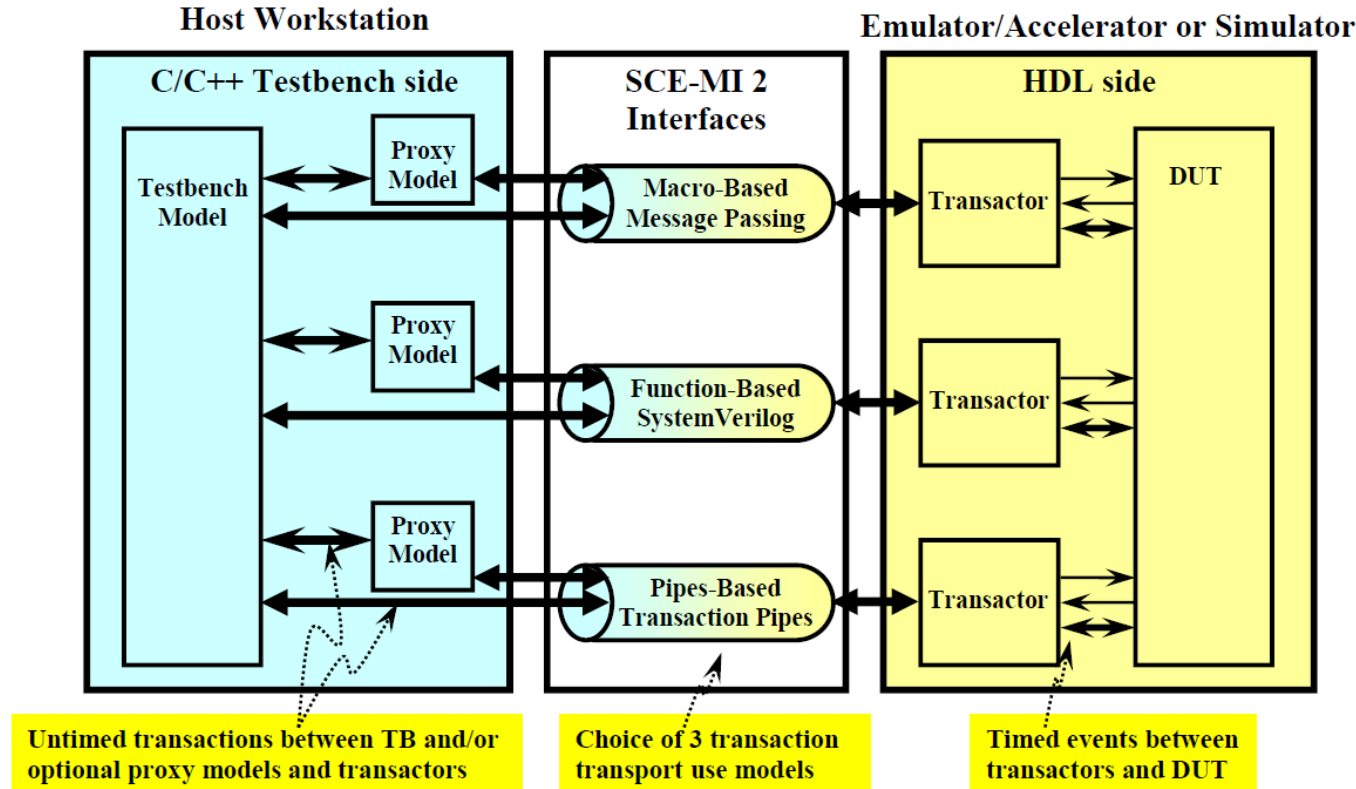
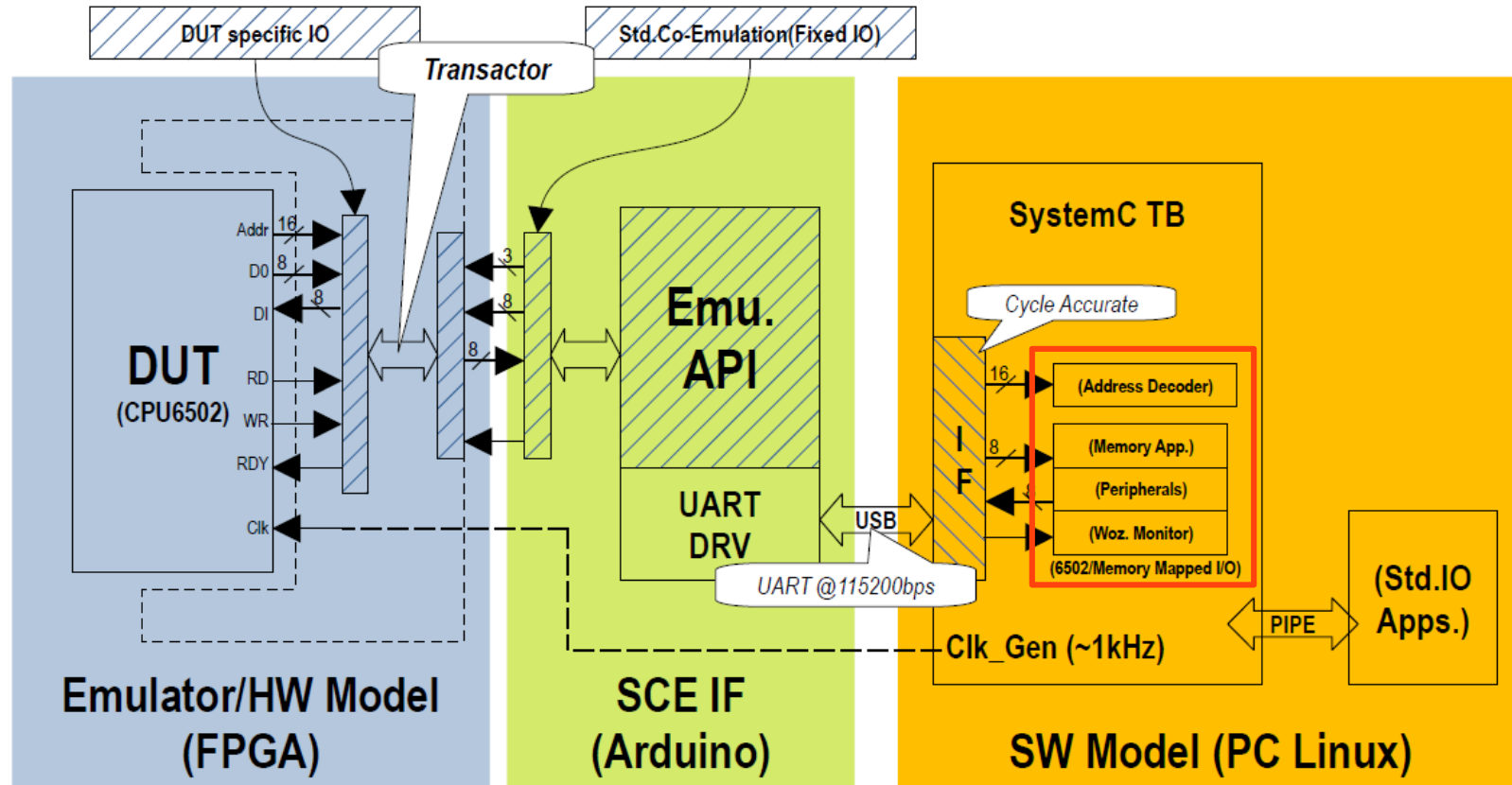


Figure 4.1

Three Interfaces

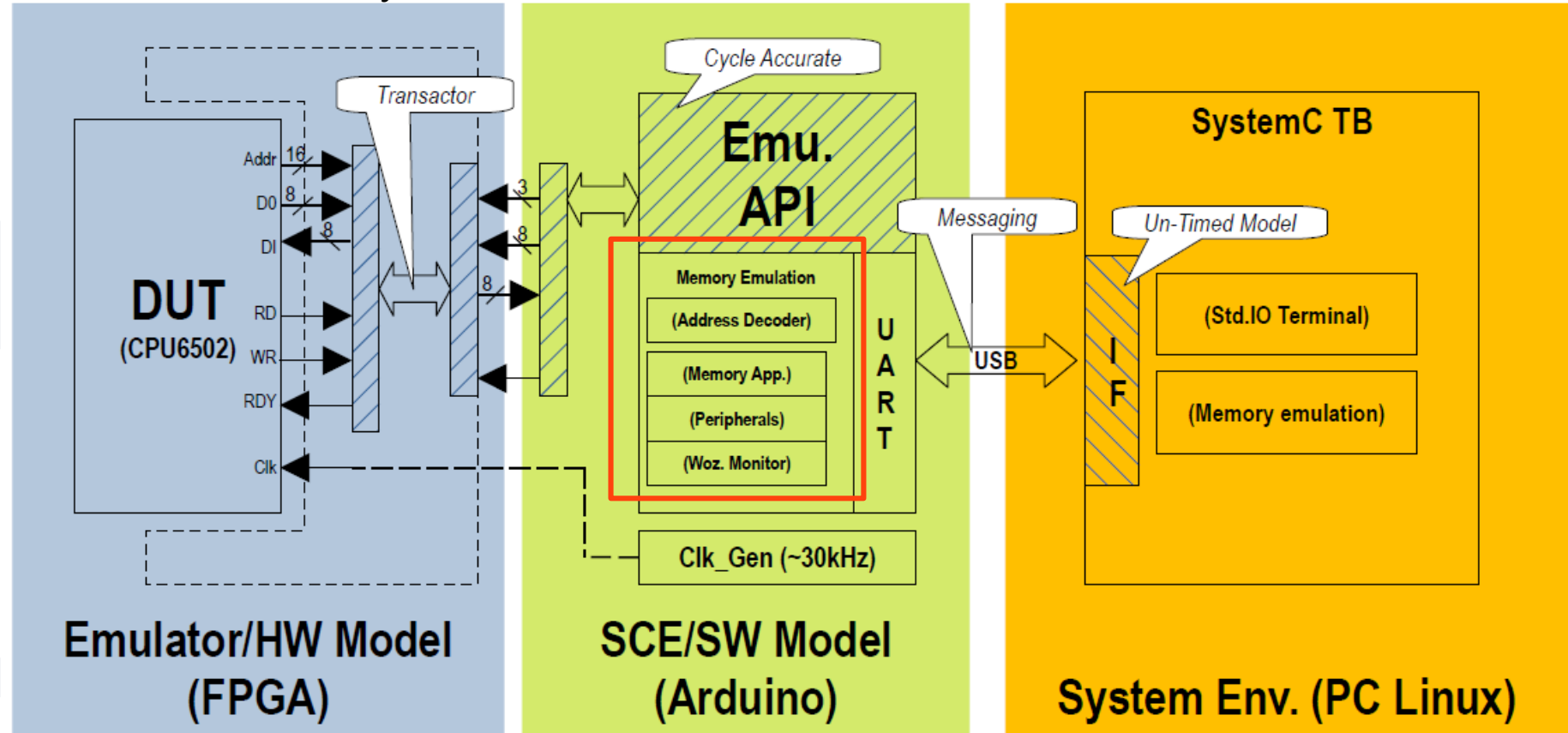
Poorman's SCE-MI(CA)

Emulation MI/API: Cycle-Accurate



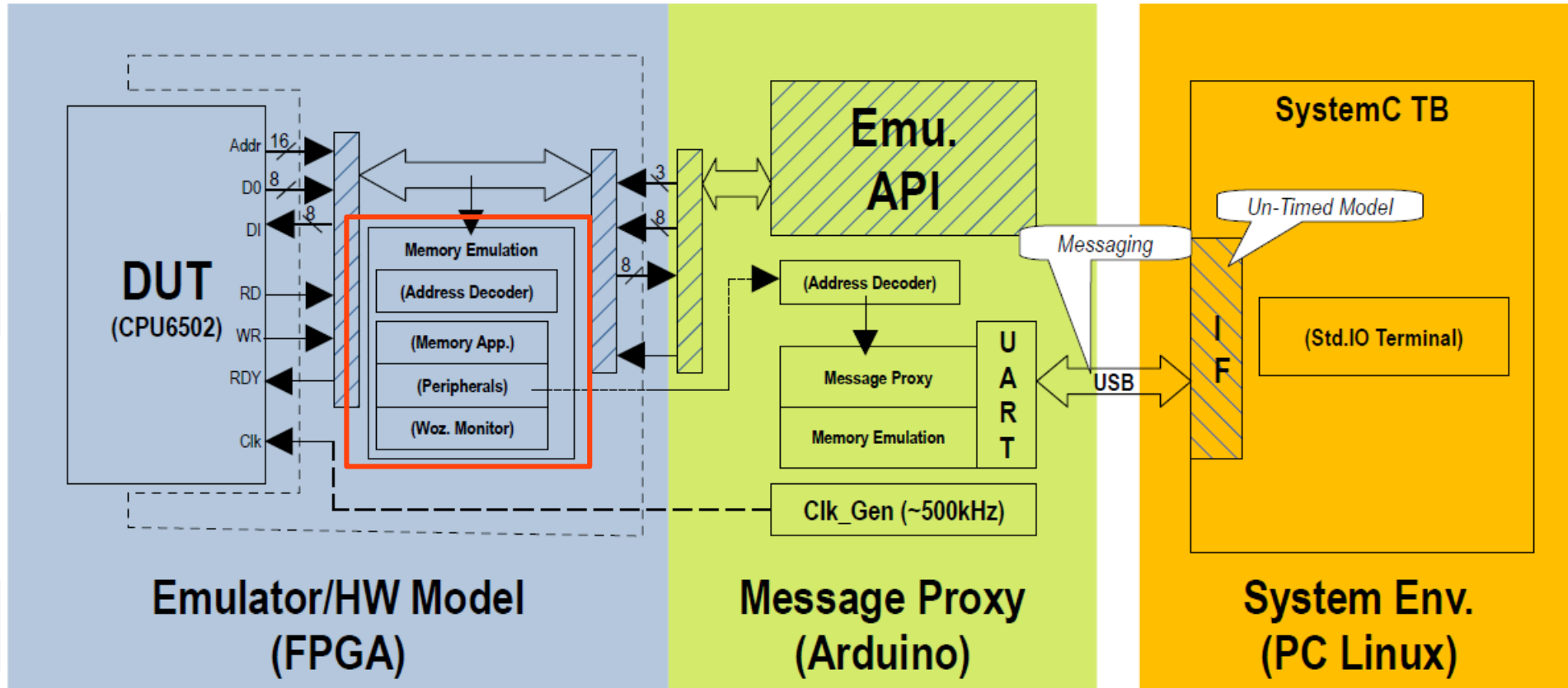
Poorman's SCE-MI(SA)

Emulation MI/API: System-Accurate

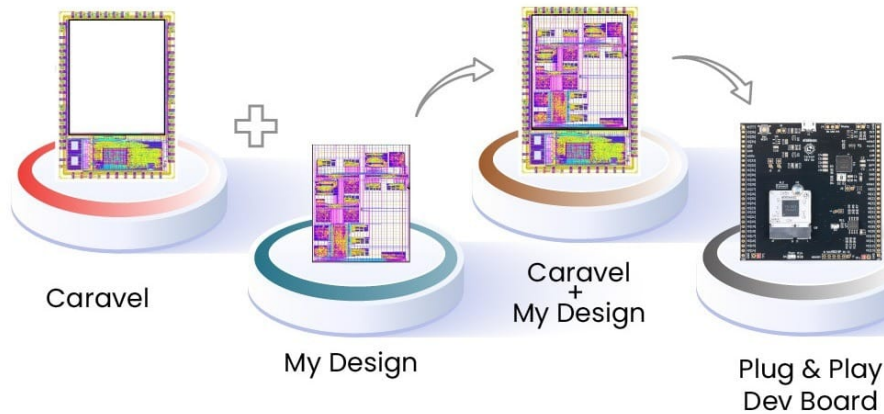


Poorman's SCE-MI(RT)

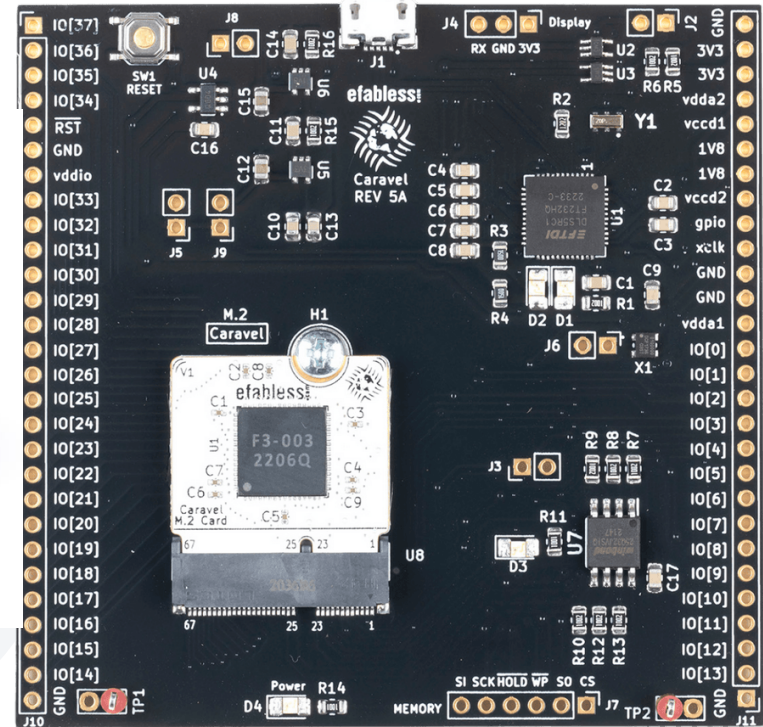
Emulation MI/API: Real-Time



eFabless 평가 보드



<https://efabless.com/chipignite>



마이크로 컨트롤러 칩의 변화

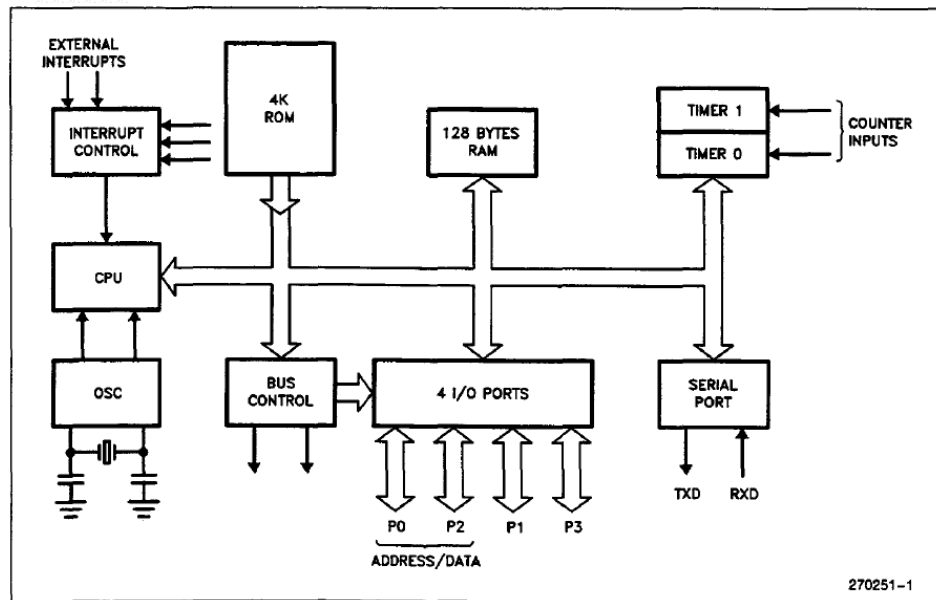
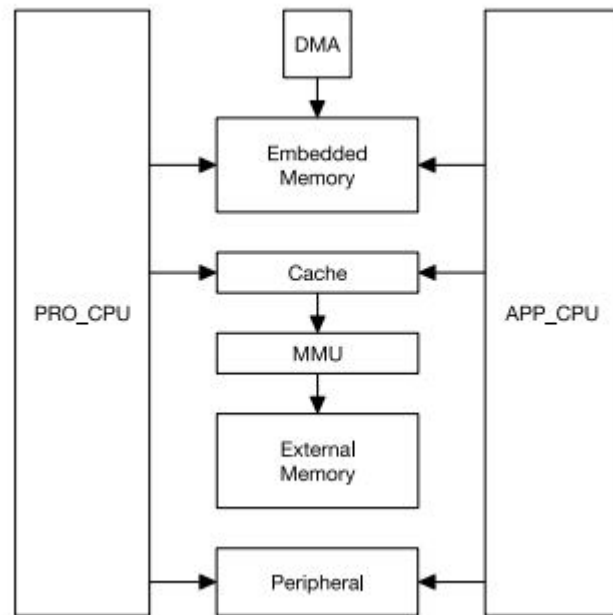


Figure 1. Block Diagram of the 8051 Core

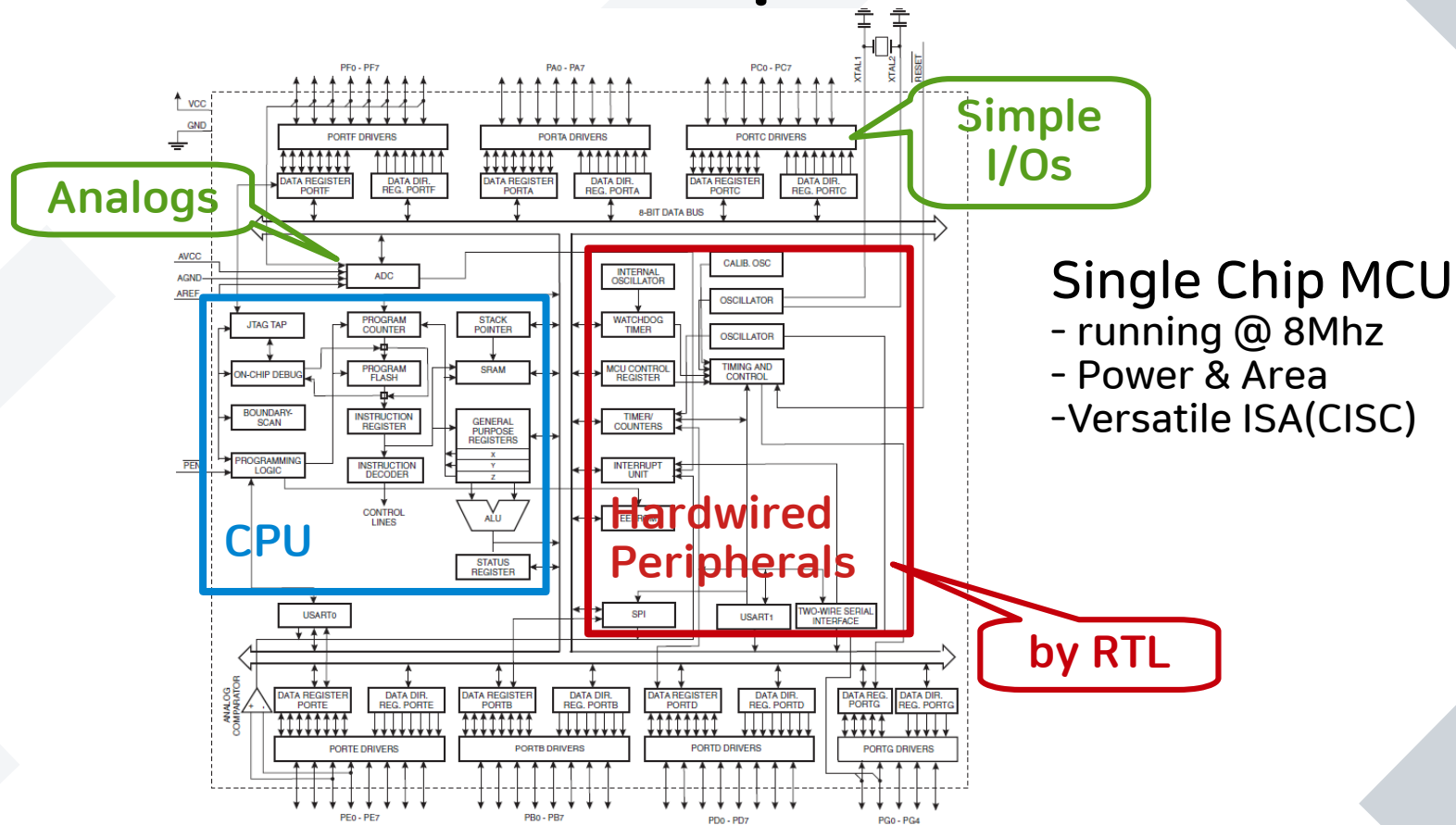
Single Chip MCU



Dual Core MCU

마이크로 컨트롤러 칩의 변화 ...

Hardwired Peripherals

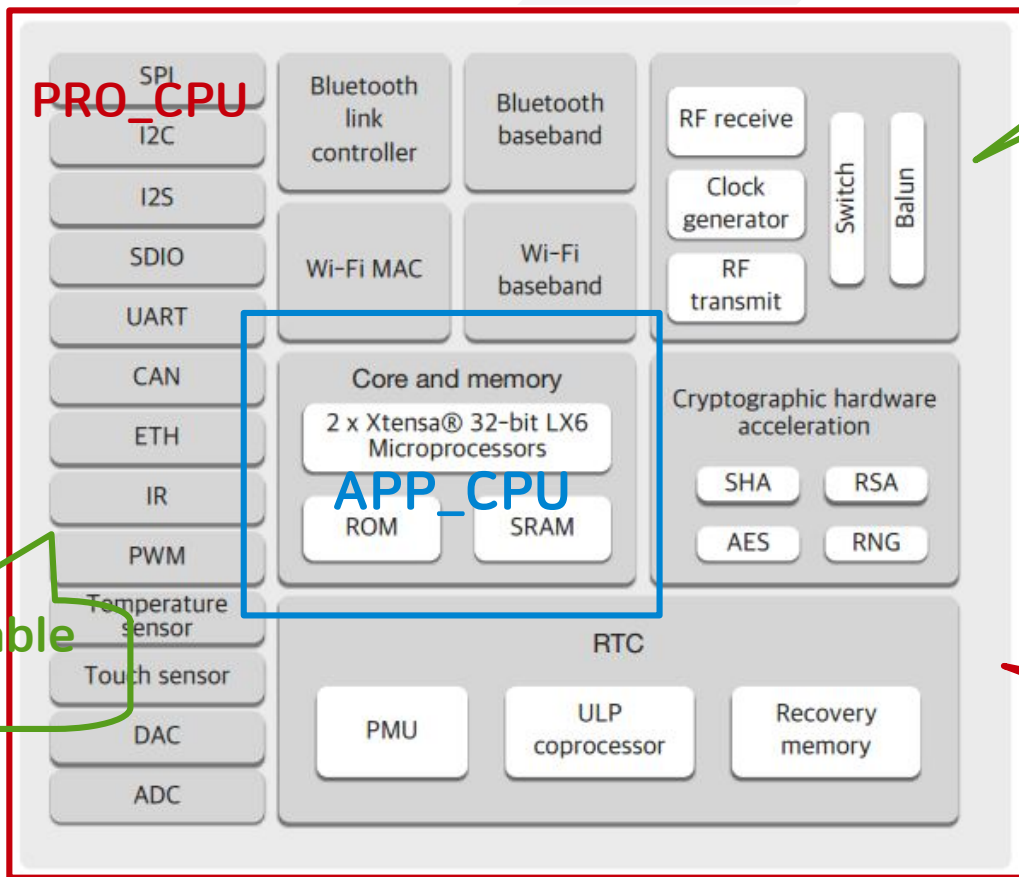


Single Chip MCU

- running @ 8Mhz
- Power & Area
- Versatile ISA(CISC)

마이크로 컨트롤러 칩의 변화 ...

Programmable Peripherals



Analogs /
PHY

Dual Core MCU

- running @ 240Mhz
- Multi-Chip Package(Chiplet)
- PPA Compromise
- Reduced ISA(RISC)

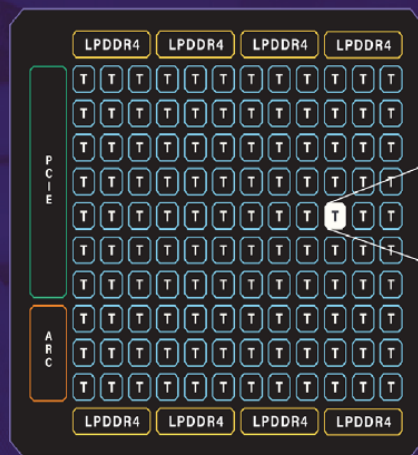
Programmable
I/Os

by RTOS Software

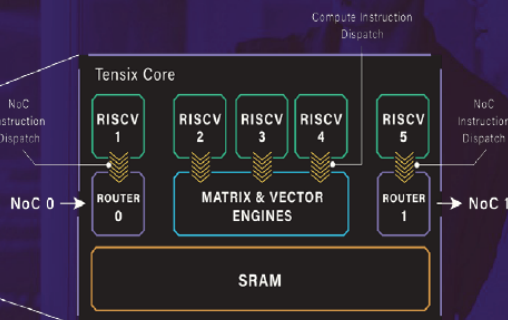
마이크로 컨트롤러 칩의 변화 ...

AI Compute

Scalable Tensix Element



Grayskull: 120 Tensix cores



- Communication Subsystem
 - RISC-V controlled NoC subsystems
- Computation subsystem
 - General computation: "Baby" RISC-V
 - RISC-V controlled Matrix and vector engines
- Collaboration Mechanisms
 - Hardware supported through RISC-V

Parallel Processing

- PPA: High Throughput
- Multi-Chip Package(Chiplet)
- Reconfigurable Processing Element
→ Simple ISA

**Application Specific HW
High Level Synthesis
Area → Energy
HW-SW Co-Design**

Hardware Design
@ High-Level Abstraction

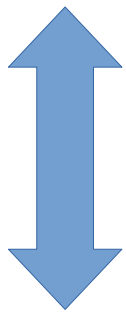


tenstorrent

“내 칩 제작 서비스” 반도체 설계 교육의 혁신

▪ “추상성 격차”의 극복

(최신 기술에 노출된 학생들)
AI 반도체 ?



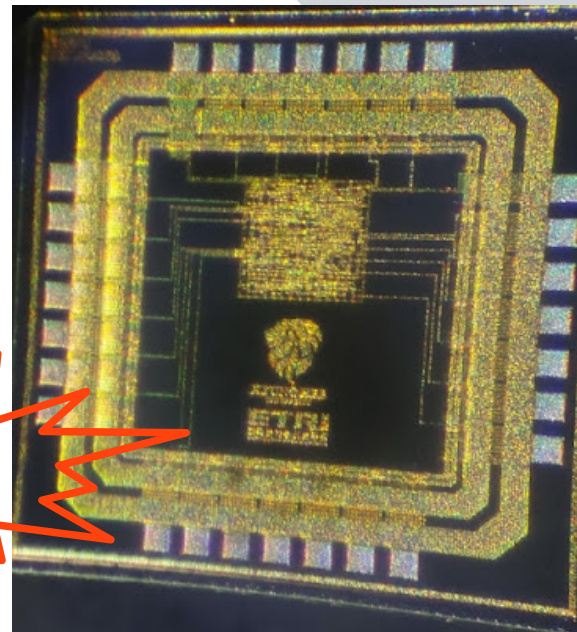
자발적 학습과 재미
동기부여

Beyond
MyChip

트랜지스터 !
(기초 기술의 풀 - 커스텀 공정)



[Volk'01] Used with permission



Chip Art

결론

무료 “내 칩 MPW 서비스”

- 풀 - 커스텀
- 디지털과 아날로그 혼성 신호 설계

오픈 - 소스 도구의 효과

- 스스로 학습 (도구 소스 입수 및 빌드)
- 커뮤니티 참여와 기여 (국내외)

반도체 설계 교육 목표의 설정

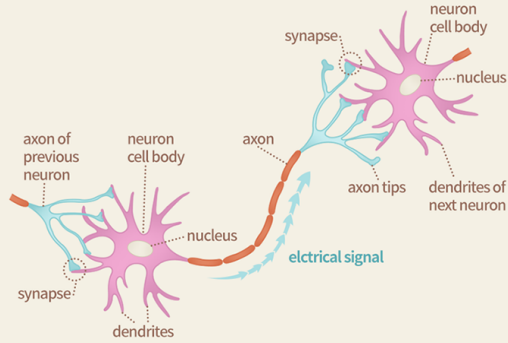
- 반도체 산업 생태계를 이해하고 안목을 기른다.
- 졸업 후 진출 분야 확대 (알고리즘 / 공정 / 설계 / 검증 / 도구)
- 관련 업무의 전문 용어 이해로 빠른 업무 적응력 배양

Open-Source & MOOC !

AI? ML? NN?

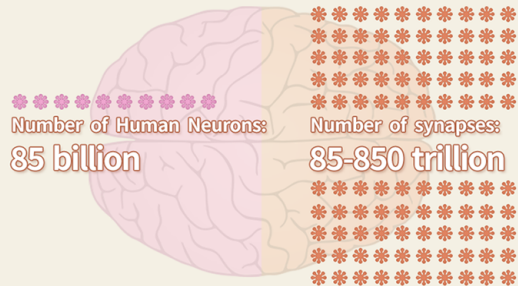
$$a^{(1)} = S \left[\sum_{n=0}^N W_n * a_n^{(0)} + b \right]$$

IF...
ELSE ...

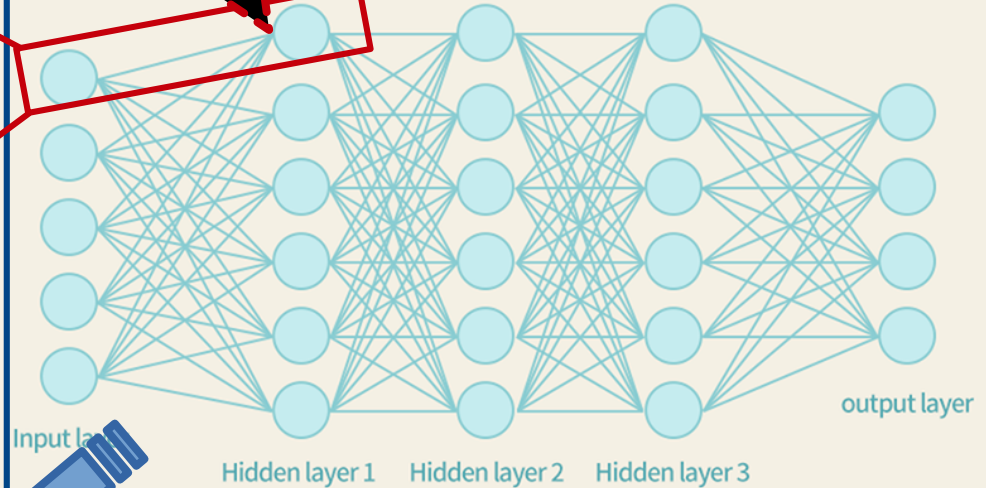


"Mathematical Foundations for Activation Functions in Artificial Neural Networks" (MedRxiv, 2016, Aug 09)
<Source: <https://www.medrxiv.org/content/10.1101/2016.08.09.16080001v1>>

Comparison of Number of Human Neuron and Synapse



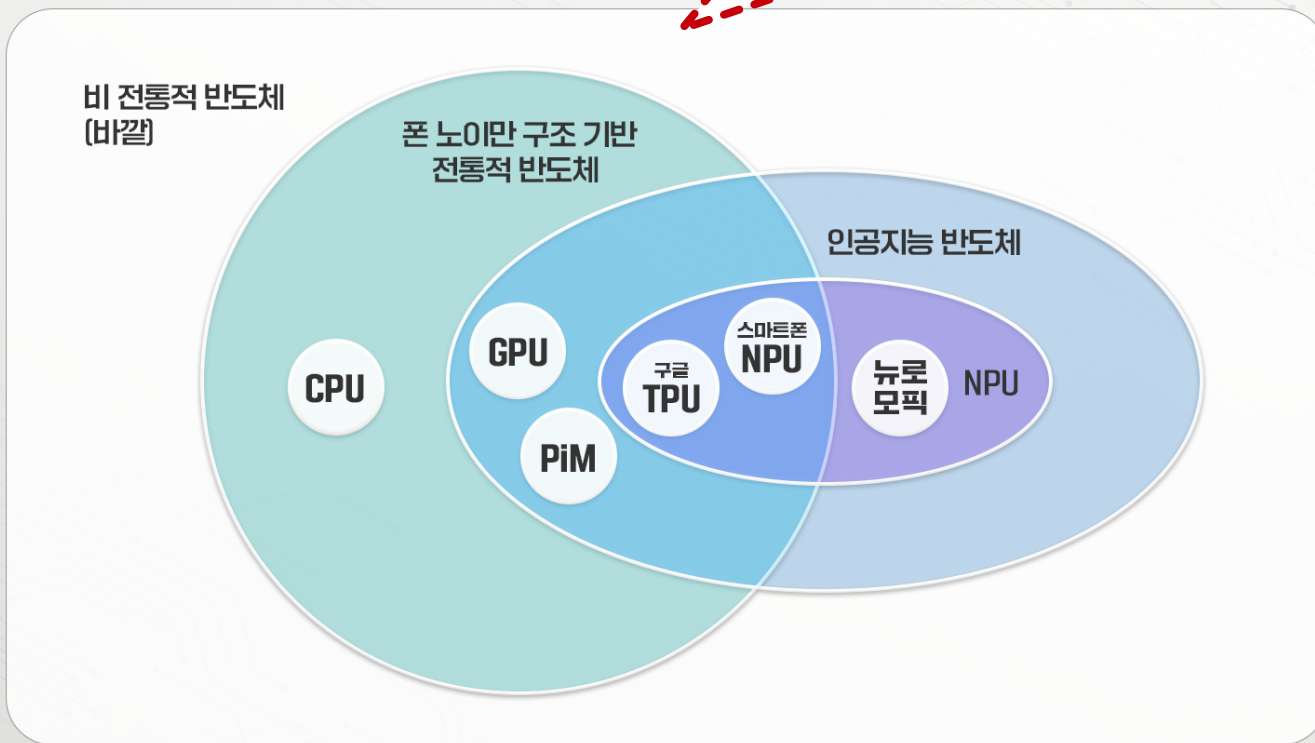
X 1,000 - 10,000



<Source: <https://towardsdatascience.com/training-deep-neural-networks-9fdb1964b964>>

AI 반도체 ?

~~AI Powered~~

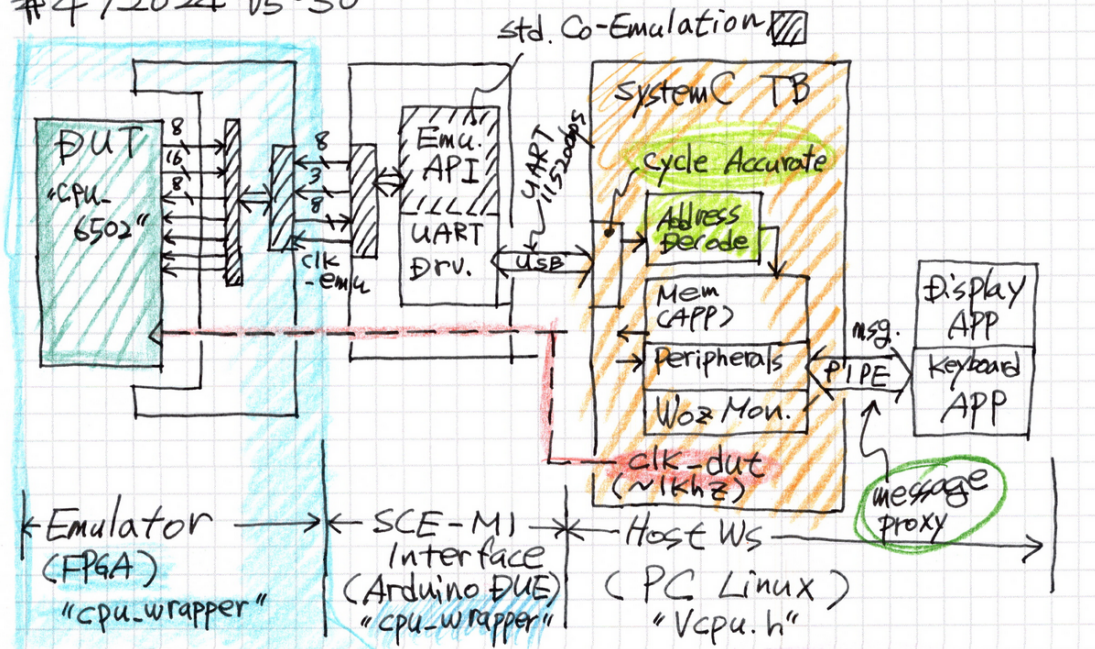


Poorman's SCE-MI(CA)

https://github.com/GoodKook/ETRI-0.5um-CMOS-MPW-Std-Cell-DK/tree/main/PSCE_API

Emulation MI/API: Cycle-Accurate

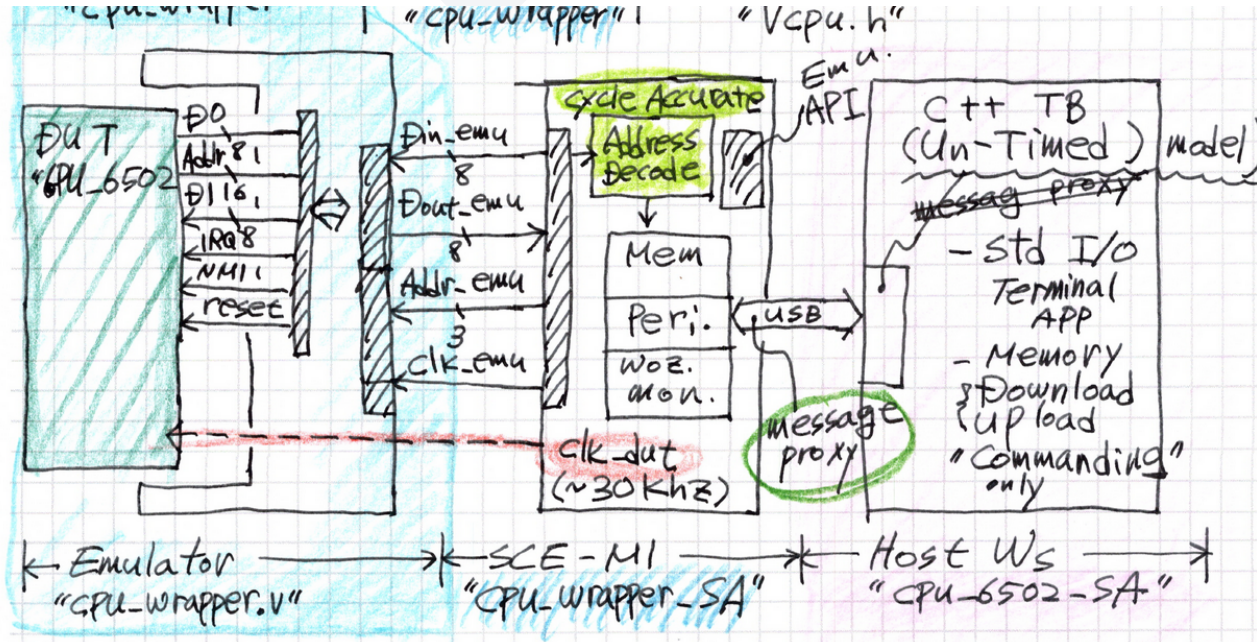
#4 / 2024-05-30



Poorman's SCE-MI(SA)

https://github.com/GoodKook/ETRI-0.5um-CMOS-MPW-Std-Cell-DK/tree/main/PSCE_API

Emulation MI/API: Stand-Alone



Poorman's SCE-MI(RT)

https://github.com/GoodKook/ETRI-0.5um-CMOS-MPW-Std-Cell-DK/tree/main/PSCE_API

Emulation MI/API: Real-Time

