

# ETRI 0.5 $\mu$ m 공정 4bit 부스 곱셈기 설계 및 칩 테스트

지하은

경희대학교 전자공학과 소속  
e-mail : [eunhajosu@khu.ac.kr](mailto:eunhajosu@khu.ac.kr)

## Design and verification of 4bit booth multiplier with my chip service

Ji, haeun

Kyung Hee University

### I. 서론

학부생의 신분으로 실제 칩을 설계, 검증하는 경험을 할 수 있다는 기대가 없었다. 파운드리(Foundry)에서 학부생의 설계를 받아주지 않을 것이라는 막연한 생각과 받아준다고 해도 칩을 받아보려면 큰 돈을 들여야 했기 때문이다. 그러던 중 『내 칩(MY Chip) 서비스』를 접했다.

반도체설계검증 인프라활성화사업 『내 칩(MY Chip) 서비스』는 학부생을 위한 MPW 무료 제작 서비스이다. MPW 제작에 지원되는 공정은 한국전자통신연구원(ETRI)의 0.5 $\mu$ m Analog CMOS 공정이다. 설계에 필요한 『내 칩(MY Chip) 서비스』 전용 PDK를 제공할 수 있다.

금전적인 문제가 해결되었으니 제출할 디자인만 있으면 나만의 칩을 받아볼 수 있게 되었다. 흔치 않은 기회라고 생각해 사업에 참여했다. 이 글에선 오픈소스 EDA 환경에서 4bit 부스 곱셈기를 설계하고 실제 칩을 수령해 테스트한 전과정을 담았다.

### II. MPW 설계

#### 2.1 설계 환경 구축

학부생의 설계를 받아줄 foundry도 생겼고 비용도 무료이다. 하지만 직접 반도체를 설계하려고 보니 여전히 막막했다. 이유는 반도체 설계에 필수적인 EDA(Electronic Design Automation) 도구가 필요하기 때문이

다. 판매용 반도체에 사용되는 상용툴은 매우 고가일 뿐더러 설계 절차와 절차에 필요한 EDA에 대한 지식도 부족했다.

이런 고민을 오픈소스 EDA인 Qflow를 사용하는 것으로 해결했다. Qflow는 OpenCircuitDesign의 툴 플로우 도구이다. Qflow에는 설계 절차마다 필요한 EDA가 오픈소스 EDA로 구비되어있으며 이 도구들을 통합해 관리해준다. 『내 칩(MY Chip) 서비스』용 PDK를 Qflow에 적용해 설계한 디자인을 생산가능한 형태인 레이아웃(Layout)으로 변환하였고 Foundry에 제출했다.

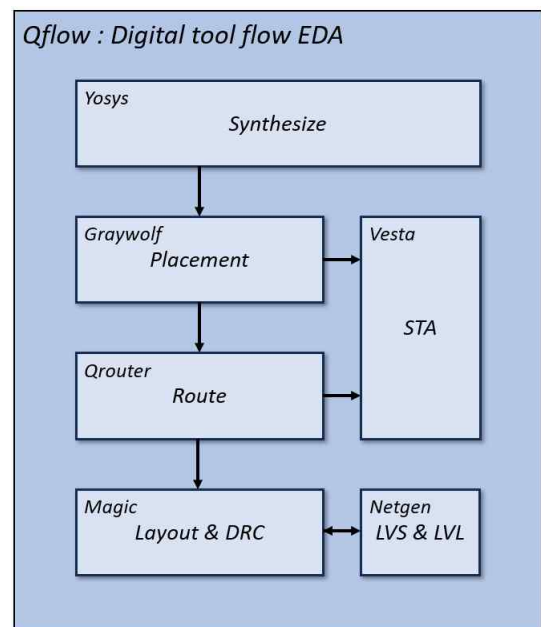


그림 1. Qflow 소속 EDA와 흐름도

## 2.2 RTL 설계 및 검증

부스 곱셈(Booth's Multiplication)은 비트에 연속된 1을 변형하여 부분곱을 줄인 알고리즘이다. 4bit 입력을 가지는 부스 곱셈기를 HDL(Hardware Description Language)인 베릴로그(Verilog)를 사용해 레지스터 전송 수준(Register-Transfer Level)로 설계하였다.

해당 디자인은 두 개의 4bit 입력을 가지므로 가능한 입력 조합을 전부 확인할만 하다고 생각했다. 하지만 RTL 시뮬레이터를 통해 나오는 파형으로 모든 입력 조합에 대한 출력을 확인하는 것은 쉽지 않은 일이라고 생각했다.

베릴레이터(Verilator)는 verilog로 기술된 디자인을 시스템씨(SystemC) 모델로 변환해준다. SystemC는 하드웨어 설계 검증을 위한 C++ 라이브러리이다. 부스 곱셈기 verilog 모델을 systemc 모델로 변환하여 검증을 진행했다. C++는 파일입 출력이 간편하므로 테스트벤치(Test Bench)에서 모든 입력조합을 파일형태로 제공할 수 있다. 출력 결과는 파형이 아닌 시험기준(Golden reference)과 비교하여 일치 여부만을 확인한다.

## 2.3 합성 및 FPGA 에뮬레이션

설계한 RTL을 합성을 통해 추상 단계를 낮춘다. 합성의 결과로 논리 게이트 사이의 연결관계가 기술된 ASIC(Application-Specific Integrated Circuit)용 넷리스트(Netlist)를 얻게된다. Yosys 합성기를 통해 합성을 진행했다. Yosys가 합성한 netlist의 검증을 두 가지로 진행했다.

첫 번째는 RTL 검증 때와 동일하게 verilog netlist를 systemc 모델로 변환하여 검증하는 것이다. 이 때 이전에 사용했던 systemc test bench를 그대로 재사용하였다(Test Bench Reuse).

두 번째는 FPGA(Field Programmable Gate Array) 에뮬레이션이다. Yosys 합성기는 합성 결과로 FPGA용 netlist도 만들어준다. 같은 합성기를 사용하는 것으로 ASIC용 netlist와 등가의 FPGA용 netlist를 얻을 수 있다. FPGA에 netlist를 프로그래밍하고 아두이노 Due보드를 통해 PC와 통신했다. 이때 systemc의 디자인 모듈은 아두이노 Due를 통해 연결된 FPGA를 감싸는 싸개(Wrapper) 파일을 사용한다. 따라서 검증 대상만 외부에서 구현되었을 뿐이므로 test bench reuse가 가능하다.

## 2.4 Layout 생성

Netlist의 검증이 마무리 되었으므로 layout을 생성한다. 생성된 layout의 유효성을 검증하기 위해서 LVS

(Layout vs. Schematic)을 진행한다. 이때 기준이 되는 회로도(Schematic)은 합성기를 통해 얻은 netlist이다.

만들어진 layout에 대하여 DRC(Design Rule Check)를 진행한다. 『내 칩(MY Chip) 서비스』 제공 공정은 레이어(Layer)간 연결 통로인 비아(Via)의 적층을 금지한다. 따라서 적층된 via들을 layout에서 찾아내고 수정한다. 또한 다른 DRC 문제들도 수정한다.

수정된 레이아웃은 더 이상 원본 schematic과의 유효성을 보장할 수 없다. 따라서 LVS를 통해 검증된 layout과 DRC를 완료한 layout과 비교한다(LVL, layout vs. Layout). 해당 작업을 반복하는 것으로 layout 생성을 완료할 수 있다. 최종 layout에 패드를 연결하고 GDS 파일로 변환하는 것으로 설계를 마무리하게 된다. 이때 남는 공간이 있다면 자신만의 그림을 반도체 기판에 그려넣는 칩 아트(Chip Art) layer를 추가할 수도 있다. 아래는 최종 layout으로 디자인 코어와 chip art를 확인할 수 있다.

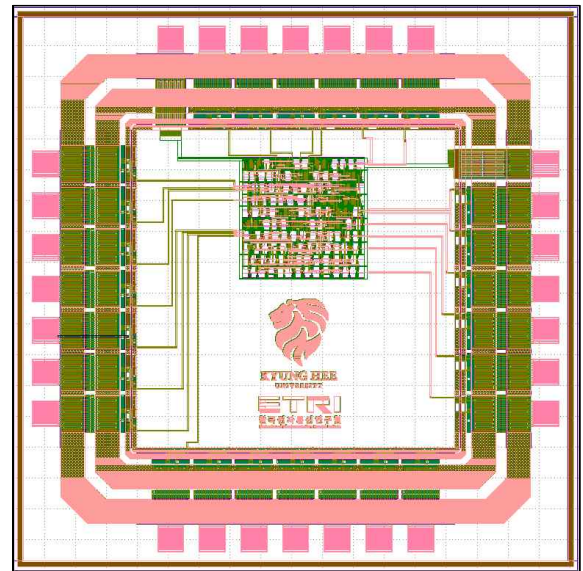


그림 2. 최종 Layout GDS파일

## III. 완성 칩 테스트

### 3.1 칩 제원

제출한 디자인에 대해 제조 공정이 진행되고 실물칩을 받아볼 수 있었다. 베어 다이(Bare Die)와 SOP패키징된 28pin 칩을 순차적으로 받아볼 수 있었다. 수령한 칩의 제원은 다음과 같다.


제원표	
설계명	4bit Booth Multiplier
	Spec.
	section figure note
	Process SOP28 -
	IO voltage 5V -
	port information
	port name port type bit width
	VDD, VSS Power pin -
	clk input clock 1bit
	reset, start input ctrl 1bit 2EA
	X, Y input data 4bit 2EA
	Z output data 8bit
	valid output trl 1bit

표 1. 제원표

### 3.2 칩 검증 및 결과 분석

칩을 수령하고 우선 VDD, VSS에 전류를 흘려 보내 칩을 테스트한다. 이때 칩의 문제로 과전류가 흘러 칩이 망가질 수 있으므로 전류를 제한해 칩 내부의 단락 여부를 테스트한다. 테스트 결과 미약한 전류도 흘러 들어가지 않는 것을 확인했다. 내부 단선이 의심되는 결과였다.

칩 검증에 사용한 환경은 FPGA 에뮬레이션에 사용한 환경을 활용했다. 단, 칩의 IO 전압은 5V이고 FPGA의 IO 전압은 3.3V이다. 따라서 기존에 활용한 아두이노 Due 보드는 IO 3.3V로 호스트에서 보내는 테스트 벤치의 데이터를 칩으로 온전히 전달시키지 못한다. 따라서 IO 5V를 사용하는 아두이노 Mega 2560 보드를 활용했다. 아두이노IDE는 소스코드를 아두이노 보드에 맞게 컴파일해주기 때문에 보드를 변경해도 기존 코드를 그대로 사용할 수 있다. 최종적으로 설계물의 검증부터 완성칩 테스트까지 테스트 환경과 test bench를 재사용하였다.

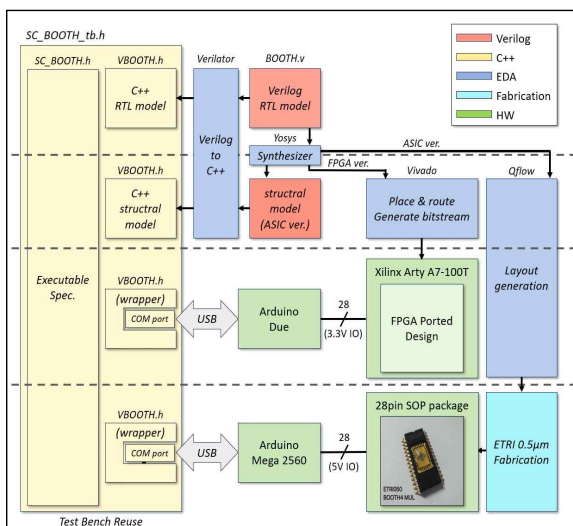


그림 test bench, environment reuse

칩 검증을 진행한 결과 입력에 대해 정상적인 출력을 칩으로부터 얻을 수 없었다. 원인은 IO PAD Library를 오픈소스 EDA로 불러들이는 과정에서 Poly2 layer를 제외한 것이다. Poly2 layer가 IO PAD에서 저항소자로서 사용되고있는 것을 모르고 디지털 공정에 사용되지 않으므로 제외한 것이 원인이었다.

## IV. 참여 소감 및 향후 계획

『내 칩(MY Chip) 서비스』 참여 이전에는 회로 수업의 회로도, 공정 수업의 반도체 레이아웃 그리고 HDL 코드 사이에 관계에 대하여 깊게 생각해 본 적이 없었다. 하지만 프로젝트를 진행하면서 EDA 도구를 활용해 디자인의 추상성을 낮추는 과정을 진행했다. 그러면서 자연스럽게 각 개념들의 관계에 대하여 알 수 있었다. 『내 칩(MY Chip) 서비스』에 참여하지 않았다면 알았다면 각기 다른 과정에서 배우는 반도체 설계 개념들을 통합하는 것이 언제가 되었을지 모른다. 프로젝트 초기에 접한 Y diagram이 이제는 이해가 간다는 것이 기쁘다.

『내 칩(MY Chip) 서비스』에 참여하면서 오픈소스 도구를 사용했다. 오픈소스를 사용하면서 처음 느낀 것은 검은 화면에 입력한 커맨드 라인으로 뭔가 결과물을 얻기는 하는데 내가 어떤 파일을 주는지 그 사이에 무슨일이 일어난 것인지 파악하지 못하고 있다는 것이다. 그 사이에 일어난 일을 파악하기 위해서 해당 오픈소스 EDA의 매뉴얼을 읽어보게 되었고 생성된 로그 파일들을 열어볼 생각을 하게되었다. 프로젝트가 마무리된 지금 각 툴들 간에 어떤 파일을 주고 받으며 툴이 만들어낸 주요 파일들의 목적을 알 수 있게 되었다. 오픈소스를 경험한 덕분에 평생 관심을 가지지 못 할뻔한 내용을 배울 수 있었다.

향후 목표는 다음과 같다. 우선 디지털 설계를 위해 standard cell library를 제공 공정에 맞게 제작하였다. 하지만 PAD적용 과정의 실수로 cell library를 실리콘 검증하지 못했다. 따라서 다음번 참여에서 모든 cell을 검증할 수 있는 칩을 제작할 예정이다. 제출한 4bit 부스 곱셈기는 사용하지 않는 공간이 너무 많았다. 처리하는 데이터 크기가 작기 때문이다. 향후 제출에서는 IO interface를 달리해 더 큰 데이터 입출력을 가지는 설계물을 제출할 예정이다.

## 참고문헌

[1] <http://opencircuitdesign.com/qflow/>