

풀커스텀(full-custom)방식의 17x17b 곱셈기의 설계와 효율적인 테스트

정회원 문상국*, 문병인*, 이용석*

Full-Custom Design of a Compact 17x17b Multiplier and its Efficient Test Methodology

Sangook Moon*, Byungin Moon*, Yongsurk Lee* *Regular Members*

요약

본 논문에서는 두 개의 17비트 오퍼랜드를 radix-4 Booth's 알고리즘을 이용하여 곱셈 연산을 수행하는 곱셈기를 설계하고 효율적인 풀커스텀 디자인에 대한 테스트 방법을 제안하였다. 클럭 속도를 빠르게 하기 위하여 2단 파이프라인 구조로 설계하였고 규칙적인 레이아웃을 위해 4:2 CSA(Carry Save Adder)를 사용하였다. 회로는 LG 반도체의 0.6-um 3-Metal N-well CMOS 공정을 사용하여 칩으로 제작되었다. 새로운 개념의 모듈 레벨 고착 고장 모델을 제안하였고 제안한 테스트 방법을 사용하여 관찰해야 하는 노드의 수를 약 88% 줄여 효율적인 고장 시뮬레이션을 수행하였다. 설계된 곱셈기는 9115개의 트랜지스터로 구성되며 코어 부분의 레이아웃 면적은 약 1135*1545 um² 이다. 제작된 칩은 전원전압 5V에서 약 24MHz의 클럭 주파수로 동작한다.

ABSTRACT

In this paper, we implemented a 17x17b binary digital multiplier using radix-4 Booth's algorithm and proposed an efficient testing methodology for the full-custom design. A two-stage pipeline architecture was applied to achieve higher throughput and 4:2 adders were used for regular layout structure. Several chips were fabricated using LG Semicon 0.6um 3-Metal N-well CMOS technology. A novel modular level stuck-at-fault model was proposed and did fault simulations efficiently using the proposed test method resulting in the reduction of the number of faulty nodes by 88%. The chip contains 9115 transistors and the core area occupies 1135*1545 um². The functional tests using ATS-2 tester showed that it can operate with 24 MHz clock at 5.0 V at room temperature.

셈기를 따로 두어 연산을 빠르게 하는 추세를 보이고 있다.

1. 서론

지속적으로 정보화 사회로 변해가는 과정과 함께 전반적인 시스템의 고성능화가 병행하고 있다. 이렇게 시스템이 고성능화가 될수록 복잡한 수학 연산이 많이 요구되는데 이에에는 기본적으로 고속의 곱셈 연산이 많이 필요하게 되어 대부분의 시스템에서는 속도가 느린 bit-serial 곱셈기보다는 전용 곱

본 논문에서는 32x32 비트 2의 보수 연산을 지원할 수 있도록 17x17 비트 2의 보수 연산을 수행할 수 있는 곱셈기를 설계하였다. 32비트 곱셈기보다 면적이 1/4정도 되는 17비트 곱셈기를 2개 사용하여 17번째 비트에 0이나 1을 할당하여 제어해 주면 효과적으로 32비트 곱셈 연산을 수행할 수 있다.

연세대학교 전기전자공학과(lizking@dubiki.yonsei.ac.kr)

논문번호 : 00407-1023, 접수일자 : 2000년 10월 23일

* 본 연구는 1998년 한국학술진흥재단 대학부설 연구과제의 학술연구비에 의하여 지원되었음.

곱셈기는 풀커스텀(full-custom) 방식으로 설계를 수행하였다. 풀커스텀 방식으로 회로를 설계하다 보면 많은 경우들을 트랜지스터 레벨에서 생각해야 하고 이에 따르는 고장 모델은 고착(stuck)-on/off 모델이다^[1]. 하지만 VLSI 회로의 응용 분야가 점점 고성능화됨에 따라 최신형 마이크로프로세서같은 경우에는 수백만개의 트랜지스터를 한칩에 집적하고 이를 트랜지스터 노드 개수에 대해서 전부 테스트 한다는 것은 시간적으로나 인력적으로 실제로 불가능하고 결정적으로 time-to-market을 놓쳐버리기가 매우 쉽다.

복잡한 트랜지스터 레벨보다 한 계층 상위, 즉 게이트 레벨에서 고장 모델을 생각하는 것이 최근까지 일반적으로 사용되는 고착 고장 모델이다. 이 고착 고장 모델은 게이트 레벨에서 정의가 되어 있기 때문에 풀커스텀 설계 측면에서는 회로를 일일이 게이트 레벨로 맵핑하기가 불가능하여 사실상 고착 고장은 생각할 수가 없다. 회로의 규모가 커질수록 설계의 많은 부분에 풀커스텀 방식이 사용될 수밖에 없는데 회로를 보다 상위레벨, 즉 기능적인 모듈 레벨에서 생각하면 기능적인 면에서 모듈별로 입출력을 맵핑할 수가 있다.

본 논문에서는 모듈 레벨에서의 고장 진단 모델을 제안하고 그에 따라 가장 적은 수의 테스트 벡터를 사용함으로써 효율적으로 고장 진단을 검증하고 논리 검증도 수행하는 방법을 제시한다. 논문의 II장에서는 설계된 곱셈기의 전반적인 구조와 사양을 설명하고 III장에서는 테스트 방법에 대해서 설명한다. IV장에서는 칩 테스트 결과에 대해서 고찰하고 V장에서 결론을 맺는다.

II. 곱셈기의 구조

2.1 블록 다이어그램

설계된 곱셈기의 전체 블록 다이어그램은 그림 1과 같다. 전체 블록은 크게 나누어 Booth 인코딩을 하여 Wallace 트리까지 연산하는 부분과 최종 합 벡터와 캐리 벡터를 처리하는 덧셈기 단으로 나누어 볼 수 있다.

전체 동작은 2단 파이프라인으로 구성되는데, 클럭 시작과 함께 입력 A와 입력 B가 플립플롭에 저장되어 입력 A는 Booth 인코더 블록으로 들어가고

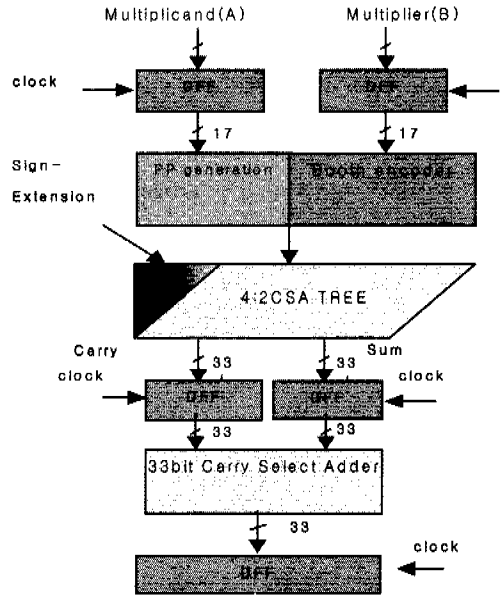


그림 1. 17x17 곱셈기의 전체 블록 다이어그램

입력 B는 Booth 선택기(selector) 블록으로 들어가게 된다. 17비트 입력 A에 대한 부분곱은 radix-4 Booth's 알고리즘에 의해 입력 B의 시퀀스에 따라 결정되어 각각 9개의 18비트 부분곱이 생성된다^[2]. 생성된 부분곱은 4:2 CSA로 구성된 Wallace 트리로 보내어져 33비트 합 벡터와 33비트 캐리 벡터로 표현된 다음 두번째 플립플롭 단으로 넘겨진다. 여기서 부호 확장 때문에 생기는 과도한 계산량을 줄이기 위해서 sign-generation이라는 방법을 사용하여 불필요한 하드웨어를 사용하지 않도록 하였다^[3].

표 1. 가상적으로 생성되는 b17 비트의 값

b17	b16	b15	Select
0	0	0	0A
1	1	0	-1A
0	0	1	+1A
1	1	1	0A

최종적으로 생성된 캐리 벡터와 합 벡터는 33비트 캐리 선택 가산기(carry-select adder)에서 처리되는데 기본 덧셈기 블록으로 캐리 체인 덧셈기(carry chain adder)를 사용하였다^[4].

2.2. Booth 인코더

Booth's 알고리즘에 따르면 입력 B의 시퀀스에 따라서 입력 A의 형태에 대한 +2A, +1A, 0A, -1A, -2A 중 하나를 택해야 하는데 이러한 선택이 이루어지는 블록이다. 입력이 2개의 17비트 데이터이기 때문에 9개의 부분곱을 구해야 하는데 첫번째부터 8번째 부분곱을 구하는 회로까지는 B 입력의 이전 비트와 현재 두 비트를 참고하기 때문에 8개의 인코더가 동일하지만 9번째 부분곱에 대한 인코더는 표 1에서 보듯이 b15와 b16만 참고하면 가상적으로 부호 확장을 시켜 b17을 알 수 있기 때문에 회로를 보다 간단하게 할 수 있다.

2.3. 부분 곱 생성 블록

입력 A가 첫번째 플립플롭 단을 거친 다음 입력 B의 시퀀스에 따라서 부분곱을 생성하는 블록이다. 1에서 8번째 부분곱을 생성하는 블록에서는 5:1 mux를 사용하고 9번째 부분곱을 생성하는 블록에서는 3:1 mux를 사용하였다. mux의 구조는 그림 2에서와 같이 회로의 면적을 줄이기 위하여 weak pull-up PMOS를 사용하였다^[3].

입력 B의 시퀀스에 따라서 +2A, +1A, 0A, -1A, -2A중 어느것을 선택할 것인지가 정해지면 선택 비트 중 하나만이 일정 시간에 활성화 되고 그에 따른 입력 A의 값들이 각각에 해당하는 형태로 쉬프트되거나 보수화가 되어서 출력되어 Wallace 트리로 들어가게 된다.

2.4. Wallace 트리

이전 단에서 생성된 9개의 부분곱들을 고속으로 처리하기 위하여 CSA(Carry Save Adder) 구조를 사용하는데 3:2 덧셈기를 사용한 트리의 형태는 역삼각형 구조를 가지는 것에 비해 4:2 CSA를 사용한

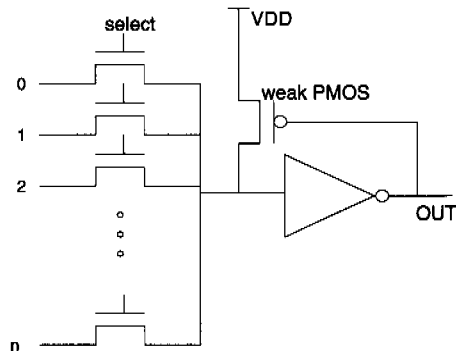


그림 2. weak PMOS를 사용한 mux의 구조

형태가 레이아웃 상에서 규칙적인 구조를 이루기 때문에 본 논문에서는 4:2 CSA를 사용하였다. Wallace 트리 안에서는 부호비트 확장때문에 생기는 불필요한 계산을 하지 않기 위해서 sign-generation method^[3]을 사용하여 면적을 최소화하고자 하였다.

2.5. Carry-select adder

Wallace 트리에서 계산되어 나온 캐리 벡터와 합 벡터는 일단 플립플롭에 담겨진다. 플립플롭은 간단한 형태의 마스터-슬레이브 형태로 구성하였다. 다음 사이클에서 플립플롭의 출력들은 33비트 CSA로 들어가게 되는데 덧셈기의 기본 셀로는 캐리 체인 덧셈기^[4]를 사용하였다. 덧셈기의 최종 출력은 원하는 결과값이 되어 출력 버퍼를 통해 패드로 전해지게 된다.

2.6. 레이아웃 및 예상 결과

전체 설계의 front-end는 Cadence의 Composer와 HSPICE로 작업을 수행하였다. 레이아웃은 전원과 확산 영역(diffusion)을 가능한 한 공유하도록 하여 회로 면적을 최소화 하였다. 전체 회로의 레이아웃은 그림 3과 같다. 레이아웃은 CADENCE의 Virtuoso editor를 사용하였다. 회로를 추출 하기 전까지의 예상되는 지연 경로는 다음과 같다.

입력 B -> 플립플롭 -> Booth 인코더 -> 부분 곱 생성기 -> Wallace 트리 -> 두 번째 플립플롭단의 입력

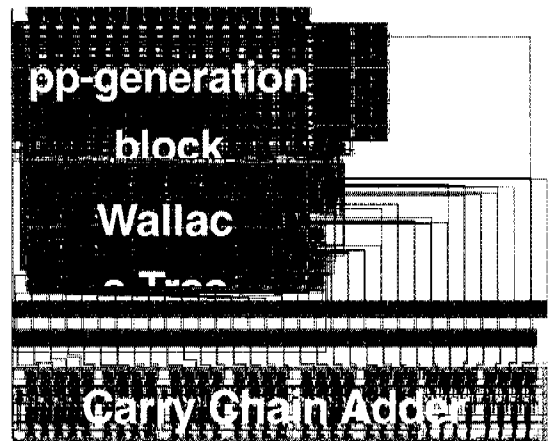


그림 3. 칩 코어의 마스크 레이아웃

위와 같은 경로로 해서 HSPICE에서 시뮬레이션 하여 예상해 본 지연시간은 다음과 같았다.

플립플롭(1.3ns) + Booth 인코더(0.7ns) + 부분 곱 생성(1.2ns) + Wallace 트리(2.1ns*3) = 9.5ns

위의 예상되는 지연시간은 각각의 회로에 출력단에 해당하는 로드의 커패시턴스를 예상하여 HSPICE 시뮬레이션을 수행한 값이다. Wallace 트리 단에서의 계산치는 4:2 CSA의 지연시간을 2.1ns 라고 측정한 다음 최대 지연 경로가 4:2 CSA를 3 단으로 통과한다고 생각했을 때의 예상치이다.

다음으로, LVS(Layout Versus Schematic)를 통과한 레이아웃에서 커패시턴스와 레지스턴스를 추출한 다음 출력단을 토글시킬 수 있는 몇 개의 테스트 패턴을 넣고 시뮬레이션하여 보았다. 그 결과로 얻은 예상 결과는 그림 4와 같았다. 최대 지연 시간의 저항치와 커패시턴스를 모두 고려한 예상치는 14.8ns였다. 이를 주파수로 환산하면 약 67MHz가 된다. 예상했던 약 100MHz 주파수와 차이가 나는 이유는 부분곱 발생 블록에서 Wallace 트리로 값을 전달하는 wire의 길이가 실제 레이아웃 상에서 예상했던 것 보다 상당히 길게 그려져 gate의 부하가 초과되었기 때문이다.

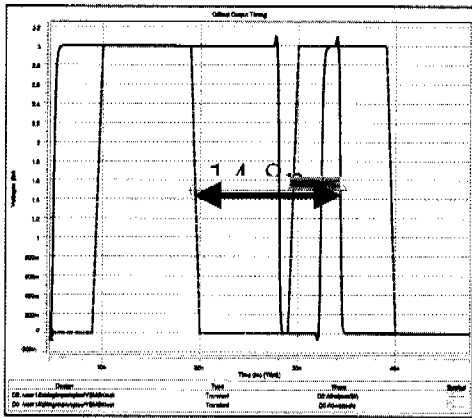


그림 4. post-simulation 결과 최대 출력 지연 시간

III. 테스트 방법

테스트는 크게 두 가지로 분류할 수 있는데, 논리 테스트와 회로 내의 고장 테스트이다. 현대의 집적 회로들이 SOC(System-On-a-Chip)화 되어 가면서 보다 테스트가 용이한 디자인으로 회로를 설계해 가는 추세이다.

논리 테스트의 검증으로는 모든 노드의 토글(toggle) 여부를 확인하고 그에 대한 출력 논리 값의 진위를 판정하는 것으로써 검증의 타당성을 부여할 수 있고 회로 내의 고장 테스트의 검증으로는 일반적으로 게이트 레벨 모델링에서 사용하는 고착 고장 모델을 사용하는 것이 일반적이다. 그런데 풀커스텀 회로에서는 모든 내부 회로를 게이트 레벨에서만으로는 생각할 수가 없다. 왜냐하면 회로 내의 모든 로직이 트랜지스터 레벨에서 정의되고 기술되기 때문이다. 따라서 풀커스텀에서 가장 정확히 고장을 판단할 수 있는 고장 모델은 트랜지스터 레벨에서 생각할 수 있는 것이다.

하지만 VLSI 시스템 내부의 모든 회로를 트랜지스터 레벨에서 고장 시뮬레이션을 수행한다는 것은 실제적으로 불가능하다. 트랜지스터의 개수가 수만 개에서 수백만개에 이르는 트랜지스터의 모든 노드에 대한 고장 모델링을 실행한다면 테스트에 투자되는 비용과 시간이 너무나 엄청나기 때문이다.

이러한 이유로 해서, 본 연구에서는 설계한 곱셈기가 2단 파이프라인으로 되어 있지만 기능적으로는 조합회로라는 점에 착안을 하여 보다 간단하고 빠른 방법으로 테스트를 수행할 수 있는 방법을 제안한다.

3.1. Modular-level 고착 고장

그림 5는 전체 회로가 수 개의 기능별 블록으로 나누어지고 각 기능별 블록이 또 세부적인 모듈로 나누어진다고 했을 때 피드백(feedback)이 없는 순수한 조합회로와 피드백이 있는 회로를 나타낸 것이다. 고착 고장 모델을 생각할 경우 어느 점에 고장이 있는지 알아내는 방법은 그 노드가 제어가 가능하고(controllable) 관찰이 가능한 경우(observable) 그 노드에 예상되는 고장값의 반대값을 할당하고 그 값이 제대로 출력까지 전달되는지의 여부에 따라서 고장 진단을 할 수가 있다. 하지만 피드백이 있는 회로에서는 같은 규모의 조합회로에 비해 매우 큰 탐색 공간을 가지고 회로 내부상태의 설정 및 관측이 어려울 뿐더러 회로 내에서는 단일고장이라 하더라도 시간전개한 반복어레이 조합회로로 고쳐서 생각하면 이를 다중고장으로 취급하여야 한다. 이러한 문제를 해결하기 위해서 피드백이 있는 회로에서는 스캔패스(scan path) 설계방식 등 테스트 용이화 설계(DFT ; Design For Testability)를

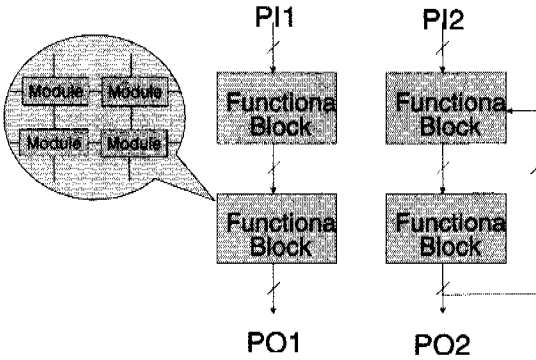


그림 5. 순수 조합 회로와 피드백이 있는 회로의 비교 블록도

이용하여 회로를 조합회로의 테스트 문제로 보고 테스트를 수행하기도 하기 때문에 실제 테스트가 매우 번거롭고 여러가지 절차를 많이 거쳐야 한다.

순차회로를 포함하여 피드백이 있는 회로들이 이렇게 테스트하기 어려운 반면, 순수 조합회로만 생각한다면 문제는 매우 간단해진다. 게이트 레벨로 구성된 회로에서 고착 고장 모델이 기존에 정의된 게이트 내에서는 고착이 전혀 없다고 가정하는 것과 마찬가지로, 전체 회로의 기본 빌딩 블록을 개개의 모듈로 보고 고장 모델을 각 모듈의 경계 (boundary)에서만 생각하는 것이다. 그림 6은 모듈 레벨 고착 고장의 개념도이다. 기능 블록 내의 회로를 모두 모듈화하고 모든 모듈들의 입력력을 제어할 수 있고 관찰할 수 있는 테스트 패턴을 주 입력 (PI ; Primary Input)에 가해줄 경우 각 경계에서의 논리값이 모두 토글될 수 있고 만일 세부적인 모듈 내에서 관찰할 수 있는 고착이 있다면 각 모듈들의 출력 경계에서의 값이 기대치와는 다른 값이 되어 고장을 검출할 수 있을 것이다.

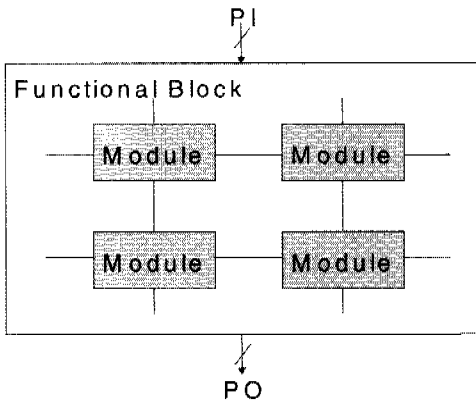


그림 6. 모듈 레벨 고착 고장의 개념도

만일, 개개 모듈 내에서 어떤 고착 고장이 있어서 경계에서 검출되지 않는 특수한 경우의 고착일지라 하더라도 경계 상에서 모든 테스트 값들이 정상적인 값들을 가지게 되면 이 모듈은 논리적으로는 올바른 동작을 하는 회로라 할 수 있다.

3.2. 테스트 결과

본 논문에서 설계한 곱셈기는 전체 2단 파이프라 인으로 이루어져 있다. 그 중간중간에 플립플롭이 있기는 하지만 레지스터의 값이 시간의 진행에 따라 피드백 되지 않기 때문에 이 회로는 순수한 조합회로라고 간주할 수 있다. 곱셈기는 전체 9115개의 트랜지스터로 구성되고 이것을 게이트로 따진다면 약 2300개 NAND 게이트에 해당된다. 다시 이것을 모듈 레벨에서 생각한다면 67개의 플립플롭, 약 80개의 4:2CSA와 전가산기, 반가산기, 9개의 Booth 인코더, 162개의 Booth's 디코더 그리고 나머지 세부 로직을 모듈화할 수가 있는데, 위에서 제시한 모듈 레벨 고착 진단 개념을 사용함으로써, 전체 트랜지스터 9115개에 해당하는 고착-on/off 고착, 즉 $9115 \times 2 = 18230$ 개에 해당하는 고착의 수를 모듈 개념으로 변환시켜 2114개의 모듈 레벨 고착 고장으로 줄여서 시뮬레이션을 수행하였다. 시뮬레이션 결과는 표 2에 나타내었다. 테스트에 사용한 툴로써는 Cadence의 Verifault를 사용하였다. 구현된 곱셈기의 테스트 결과는 37개까지의 테스트 벡터만으로도 표와 같은 결과를 얻을 수 있었다. 테스트 패턴은 임의의 패턴을 발생시키는 C 프로그램을 실행시켜서 계속 발생시키면서 고착 대조표(fault dictionary)를 계속 관찰하였다. 37번째 테스트 패턴까지 입력시켰을 때 모든 노드들에 대한 정보를 얻을 수 있었다. 표에서 untestable은 관찰, 제어할 수

표 2. 고장 시뮬레이션 결과

	total#	total%	prime#	prime%
untestable	0		0	
drop detected	1913	90.5	1361	88.1
detected	0	0.0	0	0.0
potential	0	0.0	0	0.0
undetected	59	2.8	59	3.8
drop potential	142	6.7	124	8.0
all	2114		1544	

없는 노드를 나타내고 drop detected는 고장이 이미 지정된 횟수만큼 관측되어 더 이상 시물레이션에 포함되지 않는 노드를 말한다^[5]. Drop potential은 시물레이션 결과 정상 값은 알려진 값(known-value)이지만 고장 값이 미지값(unknown-value)인 경우이며 이러한 경우는 이 회로에서 기본 모듈로서 사용한 weak pull-up PMOS를 사용하는 인버터(그림 2) 같은 경우 게이트 레벨로 바꾸어서 생각하면 출력 값이 충돌하기 때문에 발생하는 경우이다.

Undetected는 기본 모듈의 입력으로 전압 소스가 직접 입력된 경우 undetected로 처리되는 결과를 보였다. 이 결과로 보면 검증에 사용된 37개만의 테스트 벡터로 90.5% 이상의 고장 검출율을 구할 수 있었고 내부 구조상 충돌되는 값의 검출율(drop potential)을 포함시키면 전체 고장 검출율은 97.2%라고 말할 수 있다. 나머지 undetected 고장들은 테스트 벡터에 의해서 검출할 수 없는 전원이 직접 연결된 고장점들이다.

IV. 칩 테스트 결과

설계된 회로를 LG 0.6-um 3-metal 1-poly CMOS공정으로 제작하였다. 그림 7은 칩의 특성을 보여주는 Shmoo plot이다. 제작된 칩은 5.0V에서 약 24.5MHz의 주파수로 동작하는 것을 확인할 수 있었다. 그림 7의 결과는 제안한 모듈 레벨 고장 시물레이션을 통해 얻은 37개의 테스트 패턴을 포함하여 약 2만개의 테스트 패턴을 입력으로 하여 테스트 한 결과이다. C 프로그램을 이용하여 얻은 곱셈 결과와 칩 테스터에서 결과로 출력된 값을 비교하여 논리 검증할 수 있었다.

칩의 특성을 살펴보면 전체적으로 전원 전압을 충분히 공급하는데 문제가 발생한 것을 알 수 있다. 예를 들어 전원 전압을 7.0V 까지 올렸을 경우에는 약 45MHz의 주파수에서 동작할 수 있지만 5.0V보다 약간만 낮은 전압을 가하면 주파수 특성이 현저히 떨어지는 현상을 보였다. 이에 대한 원인으로서는 우선 부분 곱 생성기와 Wallace 트리 둘레 길이의 반 정도를 약 500um으로 예상하여 입력으로 연결되는 wire의 길이를 정하고 인버터의 크기가 가능한 한 최소 면적을 갖도록 크기를 결정하였는데 칩이 완성된 후 확인해보니 레이아웃상 몇몇 부분의 라우팅 wire의 길이가 800um가 넘는 wire가 사용되었고 이에 따른 인버터의 오버로딩 영향을

칩 테스트 후 결과 분석 후에야 알 수 있었다.

또한 5.0V에서 전체 동작 주파수는 24.5MHz라고 측정되었지만 이 수치는 파이프라인의 첫번째 단인 Booth 인코더를 거쳐 부분곱을 생성한 다음 Wallace 트리를 거친 시간에 해당하는데, 파이프라인의 두번째 단인 두번째 플립플롭에서부터 덧셈기를 거친 출력핀까지의 지연 시간을 측정해 보았더니 6.6ns밖에 걸리지 않았다. 이는 주파수로 환산하면 약 150MHz에 해당하고 이는 파이프라인의 균형이 잘 맞지 않았다는 사실을 보여준다.

차후에 다시 칩을 제작할 경우 Wallace 트리를 구동하는 인버터의 사이즈를 크게 하여 잘 맞추어 준다면 이 설계로 5.0V 근처에서 45MHz 이상의 주파수 특성을 보여줄 것으로 예상된다. 또한, 파이프라인의 균형을 잘 조절하여 구조를 개선시켜 준다면 100MHz 이상의 동작 주파수를 얻을 수 있을 것이다.

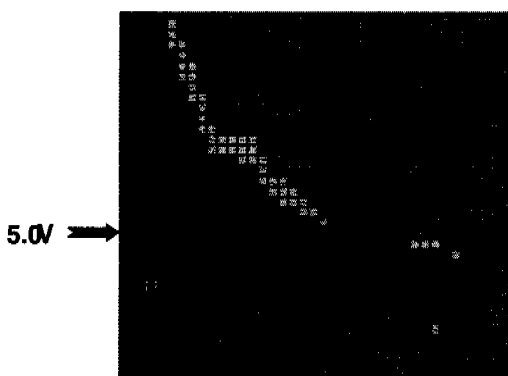


그림 7. 설계된 곱셈기 칩의 테스트 결과 Shmoo Plot

V. 결론

본 논문에서는 17x17b 연산을 수행하는 곱셈기를 칩으로 구현하고 일반적인 풀커스텀 설계에 응용할 수 있는 새로운 모듈 레벨에 대한 테스트 방법을 제안하여 트랜지스터 레벨에서 관찰해야 하는 노드의 수를 약 88% 줄여 시간이나 비용면에서 효율적인 고장 시물레이션을 수행하였다. 제작된 칩은 회로를 97.2%까지 신뢰할 수 있는 테스트 패턴을 입력으로 하여 5.0V에서 24.5MHz의 속도로 동작하였다.

앞으로 연구되어야 할 내용은 일반적인 풀커스텀

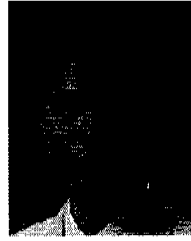
으로 회로 설계시 자동적으로 테스트 패턴을 생성시켜 주는 방법(ATPG ; Automatic Test Pattern Generation)에 대한 문제 해결, VLSI 회로가 대규모, 고집적이 되어 가면서 점점 복잡해져가는 테스트에 대한 새로운 고장 모델에 대한 정립 등이 필요할 것이다. 테스트를 고려한 설계에는 여러가지 다양한 설계 방법들이 많지만 이러한 방법들에는 그에 수반하는 하드웨어의 부담이 반드시 따르게 된다. 본 논문에서 제안된 모듈 레벨 테스트 방법을 발전시켜 기존 게이트 레벨 고착 고장 모델보다 상위의 모델을 표준화 하여 정립한다면 이는 향후 VLSI 테스트 분야의 새로운 방향이 될 수 있으리라 라고 기대된다.

참 고 문 헌

- [1] 공진홍, 김남영, 김동욱, 이재철, *VLSI 설계, 이론과 실습*, IDEC 교재개발시리즈 2I
- [2] Israel Koren, *Computer Arithmetic Algorithms*, John Wiley & Sons, Inc. 1993
- [3] M. Annaratone, W.Z Shen, *The Design of a Booth Multiplier : nMOS vs. CMOS Technology*
- [4] Yong Surk Lee, "A 4 Clock Cycle 64x64 Multiplier with 60MHz Clock Frequency", *대한전자공학회 영문논문지*, 1991, 12
- [5] 최호용, "순차회로의 테스트생성 기술", *전자공학회지* 제 25권 11호, 1998, 11
- [6] G.Goto, et al., "A 4.1ns Compact 54x54b Multiplier Utilizing Sign-Select Booth Encoders", *IEEE Journal of Solid-State Circuits*, 1997, 11
- [7] M.R.Santoro & M.A.Horowitz, "SPIM : A Pipelined 64x64 bit Iterative Multiplier", *IEEE Journal of Solid-State Circuits*, 1989, 4
- [8] Abramovici, Breuer, Friedman, *Digital Systems Testing and Testable Design*, Computer Science Press, 1990, pp. 181-280.
- [9] *Verifault-XL Reference Manual*, CADENCE, 1997

문 상 국(Moon Sangook)

정회원



1995년 2월 : 연세대학교
전자공학과 졸업
1997년 2월 : 연세대학교
전자공학과 석사
1997년 3월~현재 : 연세대학교
전자공학과 박사과정

<주관심 분야> VLSI 설계, 암호용 프로세서 설계

문 범 인(Moon Byung-In)

정회원

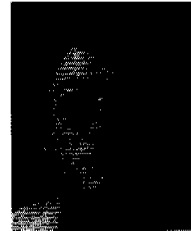


1995년 2월 : 연세대학교
전자공학과 졸업
1997년 2월 : 연세대학교
전자공학과 석사
1997년 3월~현재 : 연세대학교
전자공학과 박사과정

<주관심 분야> 마이크로프로세서 설계, SMT

이 용 석(Lee Yong-Surk)

정회원



1973년 5월 : 연세대학교
전기공학과 졸업
1977년 2월 : University of
Michigan Ann Arbor
석사
1981년 3월 : University of
Michigan Ann Arbor
박사

1993년~현재 : 연세대학교전기전자공학과 교수
재임중

<주관심 분야> 마이크로프로세서 설계