풀커스텀 (full-custom) 고속 곱셈기 회로의 효율적인 테스트 방안

(An Efficient Test Method for a Full-Custom Design of a High-Speed Binary Multiplier)

문상국

목원대학교 정보전자영상공학부

Sangook Moon

Mokwon University, Division of Information-Electronics-Image Engineering

E-mail : smoon@mokwon.ac.kr

요 약

본 논문에서는 두 개의 17비트 오퍼랜드를 radix-4 Booth's algorithm을 이용하여 곱셈 연산을 수행하는 곱셈기에 대한 효율적인 풀커스텀 디자인에 대한 테스트 방법을 제안하였다. 클럭 속도를 빠르게 하기 위하여 2단 파이프라인 구조로 설계하였고 Wallace tree 부분의 레이아웃을 규칙적으로 하기 위해서 4:2 CSA(Carry Save Adder)를 사용하였다. 회로는 하이닉스반도체의 0.6-um 3-Metal N-well CMOS 공정을 사용하여 칩으로 제작되었다. 제안된 테스트 방법을 사용하여 관찰해야 하는 노드의 수를 약 88% 줄여 효율적으로 고장 시뮬레이션을 수행하였다. 설계된 곱셈기는 9115개의 트랜지스터로 구성되며 코어 부분의 레이아웃 면적은 약 1135*1545 um2 이다. 칩은 전원전압 5V에서 약 24MHz의 클럭 주파수로 동작한다. 제안된 테스트 방법은 풀커스텀 방식의 곱셈기를 비롯한 대부분의 커스텀 설계 회로에 적용이 가능하다.

ABSTRACT

In this paper, we implemented a 17x17b binary digital multiplier using radix-4 Booth's algorithmand 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 in the Wallace tree partition. Several chips were fabricated using LG Semicon 0.6um 3-Metal N-well CMOS technology. We 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 mm2. The functional tests using ATS-2 tester showed that it can operate with 24 MHz clock at 5.0 V at room temperature.

키워드

Test method, full-custom

Ⅰ. 서 론

32x32 비트 곱셈을 수행하기 위해서는 곱셈기의 설계 방식을 면적과 처리시간의 trade-off를 고려하여 설계하여야 하는데, 2의 보수 연산을 지원할 수 있도록 17x17 비트 2의 보수 연산을 수

행하는 곱셈기는 면적을 최소화하는 방향의 설계이다. 이는 32비트 곱셈기보다 면적이 1/4정도 되는 17비트 곱셈기를 2개 사용하여 17번째 비트에 0이나 1을 할당하여 제어해 주면 효과적으로 32비트 곱셈 연산을 수행할 수 있다.

곱셈기는 풀커스텀(full-custom) 방식으로 설계를 수행하였다. 풀커스텀 방식으로 회로를 설계하다 보면 많은 경우들을 트랜지스터 레벨에서 생각해야 하고 이에 따르는 고장 모델은 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 테스트한다는 것은 시간적으로나 인력적으로 실제로 불가능하고 결정적으로 time-to-market을 놓쳐버리기가 매우 쉽다.

복잡한 트랜지스터 레벨보다 한 계층 상위, 즉 게이트 레벨에서 에서 고장 모델을 생각하는 것이 최근까지 일반적으로 사용되는 고착(stuck-at) 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.

II. 테스트 방법

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

논리 테스트의 검증으로는 모든 노드의 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 stuck-at-fault 모델을 사용하는 것이 일반적이다. 그런데 full-custom회로에서는 모든 내부 회로를 게이트 레벨에서만으로는 생각할 수가 없다. 왜냐하면 회로 내의 모든 로직이 트랜지스터 레벨에서 정의되고 기술되기 때문이다. 따라서 full-custom에서 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 수만개에서 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.

1. Modular level stuck-at fault

그림 1은 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 Testabillity)를 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 stuck-at-fault 모델이 기존에 정의된 게이트 내에서는 고장이 전혀 없다고 가정하는 것과 마찬가지로, 전체 회로의 기본 빌딩 블록을 개개의 모듈로 보고 고장 모델을 각 모듈의 경계(boundary)에서만 생각하는 것이다. 그림 2는 모듈 레벨 고착 고장(modular level stuck-at-fault)의 개념도이다. 기능 블록 내의 회로를 모두 모듈화하고 모든 모듈들의 입출력을 제어할 수 있고 관찰할 수 있는 테스트 패턴을 주 입력(PI ; Primary Input)에 가해줄 경우 각 경계에서의 논리값이 모두 토글(toggle)될 수 있고 만일 세부적인 모듈 내에서 관찰할 수 있는(observable) 고장이 있다면 각 모듈들의 출력 경계에서의 값이 기대치와는 다른 값이 되어 고장을 검출할 수 있을 것이다. 만일, 개개 모듈 내에

서 어떤 고착 고장이 있어서 경계에서 검출되지 않는 특수한 경우의 고장일지라 하더라도 경계 상에서 모든 테스트 값들이 정상적인 값들을 가지게 되면 이 모듈은 논리적으로는 올바른 동작을 하는 회로라 할 수 있다.

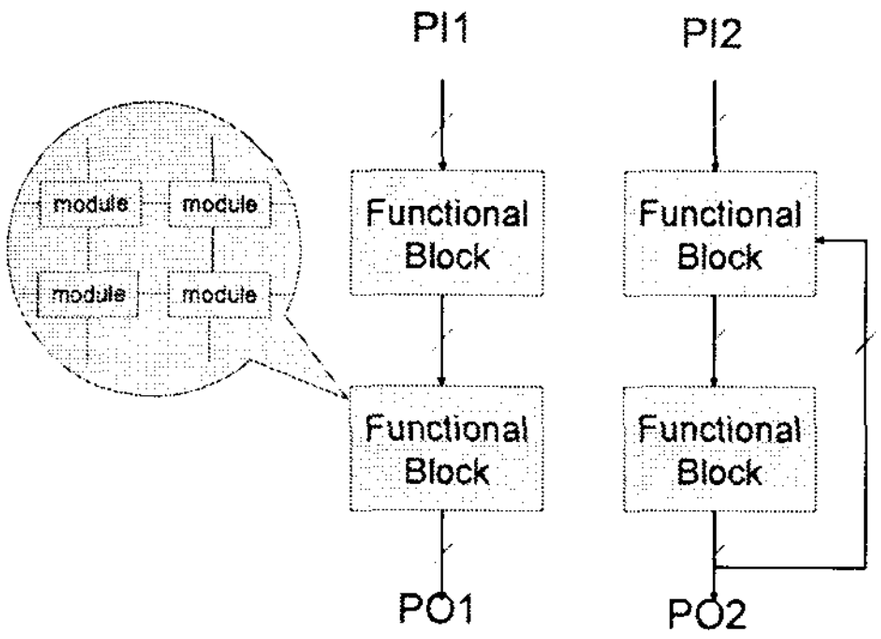

그림 1. 순수 조합회로와 피드백이 있는 회로의 비교 블록도
Fig. 1. Difference between pure combinational circuit and a circuit with feedback

III. 모듈 레벨 테스트

본 논문에서 설계한 곱셈기는 전체 2단 파이프라인으로 이루어져 있다. 그 중간중간에 플립플롭이 있기는 하지만 레지스터의 값이 시간의 진행에 따라 피드백되지 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 4:2 CSA와 전가산기, 반가산기, 9개의 Booth's encoder, 162개의 Booth's decoder 그리고 나머지 세부 로직을 모듈화할 수가 있는데, 위에서 제시한 모듈 레벨 고장 진단 개념을 사용함으로써, 전체 트랜지스터 9115개에 해당하는 stuck-on/off 고장, 즉 9115*2 = 18230개에 해당하는 고장의 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 37개 까지의 테스트 벡터만으로도 표와 같은 결과를 얻을 수 있었다. 테스트 패턴은 임의의 패턴을 발생시키는 C 프로그램을 실행시켜서 계속 발생시키면서 fault dictionary를 계속 관찰하였다. 37번째 테스트 패턴까지 입력시켰을 때 모든 노드들에 대한 정보를 얻을 수 있었다. 표에서untestable은 관찰, 제어할 수 없는 노드를 나타내고 drop detected는 고장이 이미 지정된 횟수만큼 관측되어 더 이상 시뮬레이션에 포함되지 않는 노드를 말한다[5]. Drop potential은 시뮬레이션 결과 정상 값은 known-value이지만 고장 값이 unknown-value인 경우이며 이러한 경우는 이 회로에서 기본 모듈로서 사용한 weak pull-up PMOS를 사용하는 인버터 같은 경우 게이트 레벨로 바꾸어서 생각하면 출력값이 충돌하기 때문에 발생하는 경우이다. Undetected는 기본 모듈의 입력으로 vdd나 vss가 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.

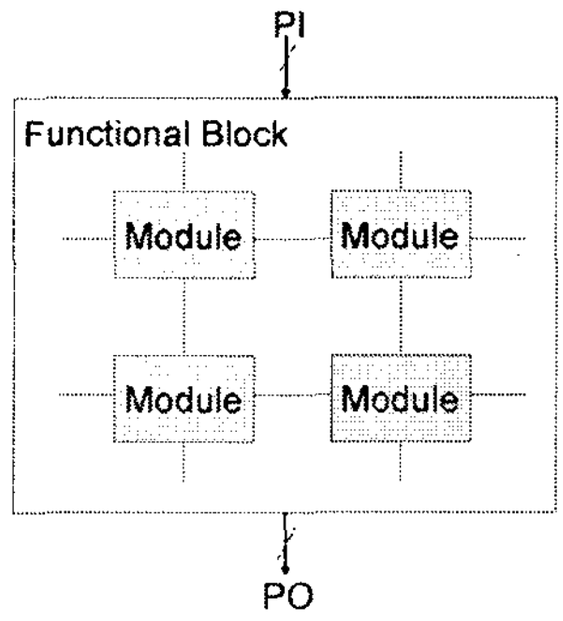

그림 2. 모듈 레벨 고착 고장의 개념도
Fig. 2. The concept of modular level stuck-at fault model

	0		0	
	1913	90.5	1361	88.1
	0	0.0	0	0.0
	0	0.0	0	0.0
	59	2.8	59	3.8
	142	6.7	124	8.0
All	2114		1544	

표 1. 고장 시뮬레이션 결과
Table 1. Result of fault simulation

IV. 고찰 및 결론

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

앞으로 연구되어야 할 내용은 이번 설계시 임의로 발생시킨 테스트 패턴을 일반적인 풀커스텀으로 회로 설계시 자동적으로 테스트 패턴을 생성시켜 주는 방법(ATPG ; Automatic Test Pattern Generation)에 대한 문제, VLSI 회로가 대규모, 고집적이 되어 가면서 점점 복잡해져가는 테스트에 대한 새로운 고장 모델에 대한 정립 등이 필요할 것이다. 본 논문에서는 모듈 레벨에 대한 고착 고장 모델에 대해서 앞으로 연구되어야 할 하나의 방향을 제시하였다. 테스트를 고려한 설계 등 여러가지 다양한 설계 방법들이 많지만 이러한 방법들에는 그에 수반하는 하드웨어의 부담이 반드시 따르게 된다. 만일 게이트 레벨 고착 고장 모델보다 상위의 모델을 본 논문을 발전시켜 표준화 하여 정립한다면 이것은 향후 VLSI 테스트에 굉장히 커다란 도움을 줄 수 있으리라고 기대된다.

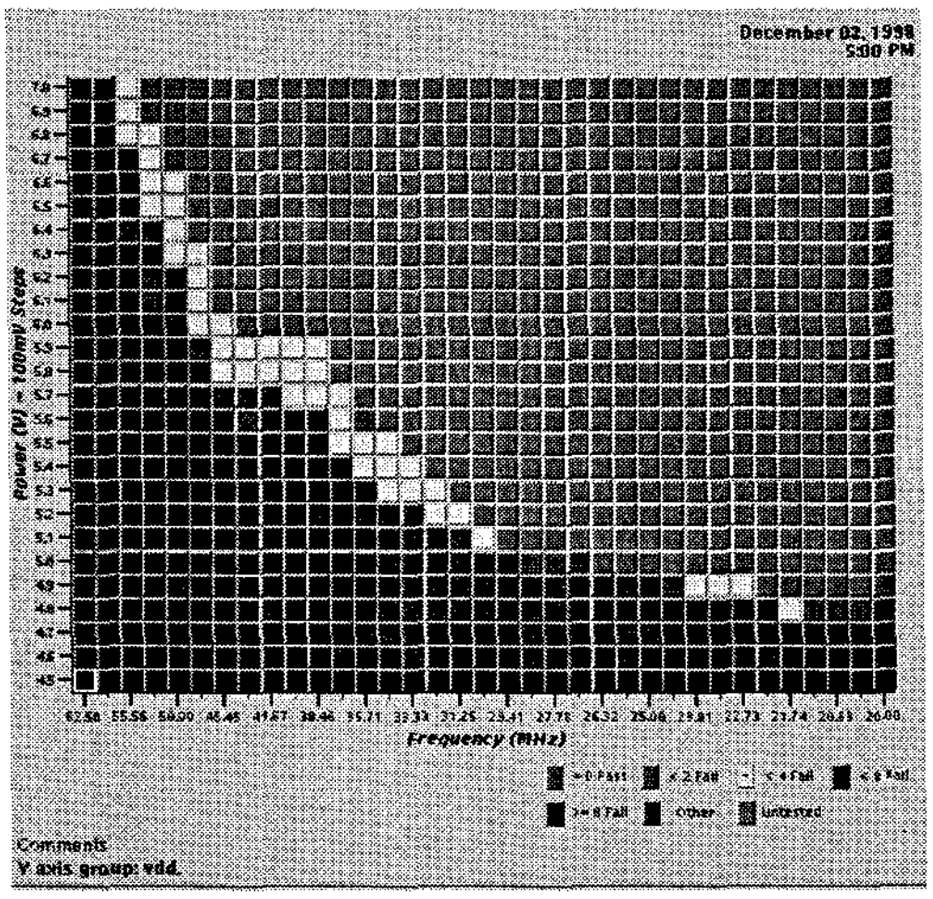

그림 3. 곱셈기 칩의 테스트 결과 Shmoo plot
Fig. 3. Shmoo plot of the multiplier

V. 참고문헌

[1] 공진홍, 김남영, 김동욱, 이재철, "VLSI 설계, 이론과 실습", IDEC 교재개발시리즈 2

[2] ISRAEL KOREN, "Computer Arithmetic Algorithms", University of Massachusetts, Amherst

[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

[9] Verifault-XL Reference Manual, CADENCE, 1997