

파이프라인 구조를 갖는 회로를 위한 내장된 자체 검사 설계에 관한 연구

양선웅*, 한재천, 진명구, 장훈
숭실대학교 대학원 전자계산학과

A Study on Design of BIST for Circuits with Pipeline Architecture

SunWoong Yang*, JaeCheon Han, MyungKoo Jin, Hoon Chang
Dept. Computer Science, Soongsil Univ.

Abstract - In this paper, we implement BIST to efficiently test circuits with pipeline architecture and JTAG to control implemented BIST and support board level test. Since implemented BIST is designed to be initialized using new seed, hard-to-detect faults are easily detected. Besides, to optimize area overhead, it uses JTAG instead of BIST controller and modified pipeline register instead of added test pattern generator and signature generator. And, to optimize pin overhead, it uses pins of JTAG. Function and efficiency of implemented BIST is verified by simulation.

1. 서 론

집적 회로가 발달된 이후 그 사용 범위와 용용 범위가 점점 넓어지고 있으며, 이러한 경향은 칩의 높은 집적화와 높은 신뢰도를 요구하고 있다. 과거 몇 개의 게이트 혹은 간단한 기능 블록을 갖고 있는SSI와 MSI 수준의 칩은 편을 통한 테스트가 가능하였기 때문에 칩의 신뢰도를 보장할 수 있었다. 그러나 1980년대에 접어들면서 칩의 밀도는 급격히 증가하여 VLSI로 디지털 회로가 구현됨에 따라 편을 통한 테스트가 불가능하게 되었고, 이로 인하여 칩의 신뢰도를 보장할 수 없게 되었다.

이러한 문제점을 해결하기 위하여 테스트 용이화(Design For Testability) 기법이 등장하게 되었다. 테스트 용이화 기법은 크게 회로의 한 부분이 회로 자체를 테스트하는 내장된 자체 검사 회로, 회로의 플립플롭을 스캔할 수 있는 구조로 설계하는 스캔 기법, 기판 수준의 테스트를 지원하기 위한 경계 주사 기법, 그리고 회로의 관찰가능도(observability)와 제어가능도(controllability)를 높이기 위해서 테스트 포인트를 삽입하는 Ad-hoc 방법 등이 있다. 이러한 방법들은 실제 상용화에 성공한 칩에 적용되고 있다. 인텔의 경우, 80386은 내장된 자체 검사 기법, 80486은 스캔 기법, 그리고 Pentium에서는 경계 주사 기법까지 적용하고 있다[1]. SuperSPARC과 SuperSPARCl의 경우도 내장된 자체 검사 기법, 스캔 기법, 그리고 경계주사 기법을 사용하였다[2,3].

본 논문에서는 DSP 코어의 구조에 맞도록 테스트 용이화 기법인 내장된 자체 검사 회로와 경계 주사 회로를 Verilog을 이용하여 설계 구현하였다. 그리고 구현된 회로는 VerilogXL을 이용하여 동작 검증을 하였으며, Sunrise의 고장 시뮬레이터(fault simulator)를 이용하여 고장 탐지율을 구하였다.

본 논문은 다음과 같이 구성되어 있다. 2절에서는 본 논문에서 사용한 DSP 코어의 구조에 대하여 기술하-

본 연구는 통상산업부와 과기처에서 지원하는 IDEC 주관 사업인 ASIC 기반기술과제의 지원으로 수행되었음.

고, 3절에서는 DSP 코어에 적용된 내장된 자체 검사 기법과 경계 주사 기법에 대하여 설명한다. 그리고 4절에서는 실험 결과를 보여준다.

2. DFT 기법의 적용

본 논문에서는 파이프라인 구조를 갖는 회로에 테스트 용이화 기법인 내장된 자체 검사 기법을 적용하였으며, 이러한 테스트 회로의 제어 및 기판 수준의 테스트를 지원하기 위한 경계주사 기법을 적용하였다.

2.1 회로의 구조

본 논문에서 사용된 회로는 두 단계의 파이프라인(EX1, EX2) 구조로 모두 3개의 파이프라인이 존재한다. 입력 단에 레지스터 파일1(Rf1)은 32비트 레지스터 4개로 구성된다. EX1과 EX2 사이에는 40비트 레지스터 4개로 구성된 파이프라인 레지스터가 있다. 그리고 마지막으로 EX2의 출력 단에는 40비트 레지스터 4개로 구성된 레지스터 파일2(Rf2)가 있다. 레지스터 파일2의 출력 중 일부는 귀환(feedback)되어 EX1과 EX2의 입력으로 사용된다. 그림1은 파이프라인 구조를 갖는 회로의 구조를 보여준다.

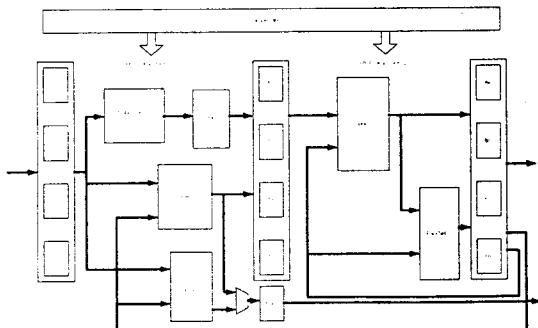


그림 1 DPU의 구조

2.2 내장된 자체 검사 회로

내장된 자체 검사 회로는 테스트 패턴을 생성하는 회로와 테스트 결과를 압축하는 압축치(signature) 생성기로 구성된다.

내장된 자체 검사 기법은 테스트 패턴 생성 방법과 인가 방법에 따라 분류할 수 있다.

테스트 패턴 생성 방법에 따라 분류하면 전경우(exhaustive), 의사 전경우(pseudo-exhaustive)[4, 5, 6], 무작위(random)[7], 의사 무작위(pseudo-random)[8, 9, 10, 11]로 분류할 수 있다.

테스트 패턴 인가 방법에 따라 분류하면 test-per-

clock과 test-per-scan방법으로 분류할 수 있다. 본 논문에서는 제안한 방법은 의사 무작위 test-per-clock, 그리고 test-per-clock 기법의 장점들을 반영하고 있다. 기존의 의사 무작위 기법에서는 인가될 초기값을 칩 내부에 저장하거나 특정 패턴을 생성하기 위한 부가적인 하드웨어가 사용되었다. 그러나 제안된 방법은 초기값을 칩의 핀을 통해서 외부에서 인가할 수 있게 함으로써 이러한 하드웨어의 오버헤드를 줄인다. 또한 초기값을 인가할 경우에만 test-per-scan으로 동작하고 테스트 패턴을 인가할 경우에는 test-per-clock으로 동작하게 함으로써 테스트 수행 시간을 단축한다. 그림 2는 설계된 패턴 생성기의 합성된 결과이다.

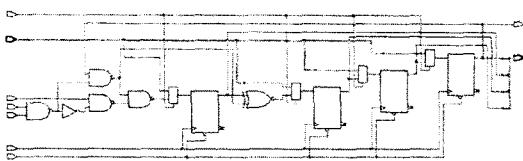


그림 2 합성된 테스트 패턴 생성기(4 bits)

설계된 테스트 패턴 생성기를 이용해 테스트 패턴을 생성하는 절차를 살펴보면 다음과 같다. 먼저 쉽게 검출되지 않는 고장에 대해 ATPG(Automatic Test Pattern Generation)을 수행한다. 생성된 패턴을 패턴 생성기의 초기값으로 사용하여 패턴을 생성한다. 고장 시뮬레이션을 통해 고장 탐지율의 증가율이 급격히 떨어지는 포인트를 찾는다. 이 포인트 이후의 남아있는 고장 리스트 중 가장 검출하기 어려운 고장에 대해 ATPG를 다시 수행한다. 생성된 패턴을 테스트 패턴 생성기의 새로운 초기값으로 인가한다. 이러한 과정을 원하는 고장 탐지율에 도달할 때까지 반복한다. 본 논문에서 제안한 방법을 통해 구해진 고장 탐지율은 결론에 나와있다.

그림3은 위에서 언급한 방법으로 설계된 테스트 패턴 생성기를 적용한 후의 구조를 보여준다.

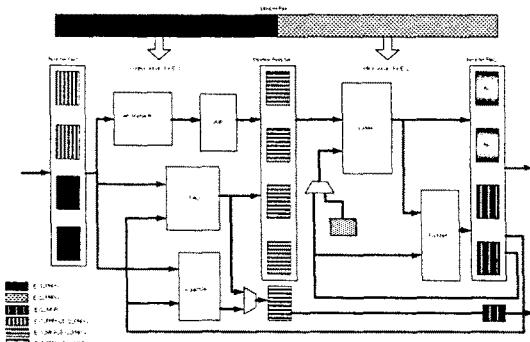


그림 3 테스트 용이화 기법이 적용된 DPU

그림3에서 알 수 있듯이 EX1을 테스트하기 위한 패턴을 생성하기 위해 DpuDecPipe의 일부 레지스터, 레지스터 파일1, 레지스터 파일2의 Ax, Bx가 사용된다. 레지스터 파일2의 Ax, Bx는 Rf2로 부터 PALU와 BSHIFTER에 feedback되는 경로를 끊기 위해 사용된다. EX1의 압축치를 생성하기 위해서 파이프라인 레지스터와 플래그 레지스터가 사용된다.

또한 EX2를 테스트하기 위해서 테스트 패턴 생성기는 DpuDecPipe의 일부 레지스터, 파이프라인 레지스터, 그리고 PackNet으로 인가되는 귀환을 끊기 위해

레지스터 파일1의 X0,Y0 레지스터가 사용된다. 압축치 생성기로는 레지스터 파일2가 사용된다.

이상에서 살펴본 바와 같이 각 레지스터는 테스트 세션에 따라 테스트 패턴 생성기 또는 압축치 생성기로 동작하도록 설계함으로써 부가적인 회로의 추가를 최소화하였다. 또한 이러한 회로의 동작을 경계 주사 회로에서 생성되는 제어 신호와 클럭 제어기를 이용하여 제어 힘으로써 내장된 자체 검사 회로 제어기가 필요하지 않게 하였다. 클럭 제어기의 구조는 그림4에 나와있다.

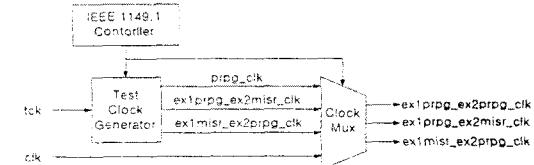


그림 4 클럭 제어기의 구조

2.3 경계 주사 회로

기판 수준의 테스트를 지원하기 위한 경계 주사 회로는 IEEE 1149.1 표준안에 따라서 설계 및 구현되었으며 내장된 자체 검사 회로를 위한 명령어가 추가되었다. 구현된 경계 주사 회로는 5개의 테스트 데이터 레지스터, 4비트의 명령어 레지스터, 5개의 핀, 16 상태의 테스트 포트 제어기(TAP:Test Access port)로 구성된다.

5개의 테스트 데이터 레지스터는 1비트의 bypass 레지스터, 138비트의 경계 주사 레지스터와 3개의 테스트 패턴 생성 및 압축치 생성 회로로 구성된다. 그림5는 경계 주사 회로와 기능 블록과의 인터페이스를 보여준다.

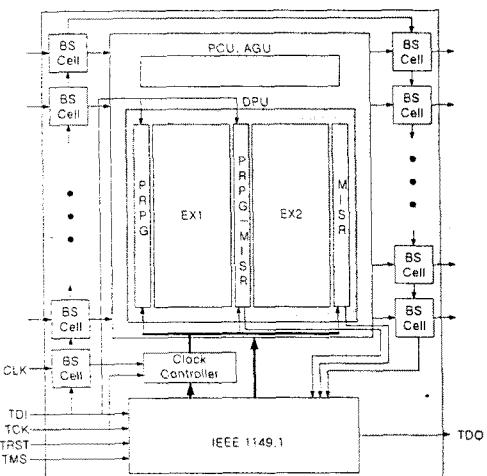


그림 5 경계 주사 회로와 기능블록의 인터페이스

구현된 경계 주사 회로는 3개의 표준에서 지정한 명령어(public instruction)와 내장된 자체 검사를 위해 추가된 4개의 명령어(private instruction)를 지원한다. 표1은 사용된 명령어들을 보여준다.

내장된 자체 검사를 위해 추가된 명령어는 내장된 자체 검사 회로를 위해 추가된 명령어이다. EX1_RESEED와 EX2_RESEED 명령어는 테스트 패턴 생성기에 새로운 seed를 인가하기 위해 사용하는 명

령어이다. EX1_PRPG와 EX2_PRPG는 테스트 패턴 생성기에서 테스트 패턴을 생성시키기 위해 사용하는 명령어이다.

Public 명령어	SAMPLE/PRELOAD	0001
	EXTEST	0000
	BYPASS	1111
Private 명령어	EX1_SEED	1101
	EX1_PRPG	1001
	EX2_SEED	1110
	EX2_PRPG	1010

표 1 경계 주사 명령어

3. 결 론

본 논문에서는 파이프라인 구조를 갖는 회로를 테스트하기 위한 내장된 자체 검사 회로와 경계 주사 회로는 Verilog을 사용하여 설계하였다. 그리고 회로 합성은 Synopsys를 이용하였으며, 동작 검증은 VerilogXL을 사용하였다. 구현된 회로의 효율성을 검증하기 위해서 Sunrise의 고장 시뮬레이터를 사용하였다.

그림 6은 테스트 패턴 생성기의 초기값 재인가에 따른 고장 탐지율의 변화를 보여주는 그래프이다. 가장 밑의 그래프는 하나의 초기값을 가지고 패턴을 생성한 경우를 나타내며, 밑에서 두 번째 그래프 첫 번째 초기값으로 P1 지점까지의 고장을 검출하고 두 번째의 초기값을 인가한 후의 고장 탐지율 변화를 보여준다.

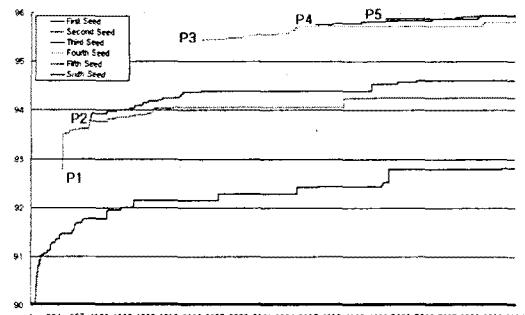


그림 6 고장 탐지율의 변화

표2는 파이프라인 구조를 갖는 회로를 테스트하기 위해 설계된 내장된 자체 검사 회로와 경계 주사 회로를 이용해 위에서 언급한 테스트 패턴 생성 및 인가 알고리즘을 적용하여 EX1과 EX2 단계의 고장 탐지율을 측정한 표이다.

	EX1	EX2
고장검출율	96.13 %	97.13 %

표 2 고장 탐지율

표3은 파이프라인 레지스터를 변형한 내장된 자체

	No Test	Test
면적	67109	71474(+6.5%)

표 3 면적 오버헤드

검사 회로와 경계 주사 회로가 적용된 후의 칩 면적 오버헤드를 보여주고 있다. 첫 번째 열은 테스트 회로가 인가되지 않은 경우의 면적이고, 두 번째 열은 BIST 및 경계 주사 회로가 인가된 후의 면적이다.

파이프라인 구조를 갖는 회로의 경우, 본 논문에서 제안하는 방법으로 테스트 회로를 구현할 경우 적은 면적 오버헤드, 짧은 테스트 패턴, 그리고 짧은 테스트 시간으로 높은 고장 탐지율을 얻을 수 있을 것이다.

향후에는 회로에 인가되는 제어 신호의 특성을 테스트 패턴 생성기에 반영하고, ATPG에 의해 생성된 테스트 패턴을 분석하여 테스트 패턴의 길이를 줄이는 연구와 테스트 패턴 초기값을 재인가할 효율적인 지점을 찾는 방법에 관한 연구를 수행할 예정이다.

(참 고 문 헌)

- (1) Wayne Needham and Nags Gollakota, "DFT STRATEGY FOR INTEL MICROPROCESSORS," IEEE International Test Conference, 1996
- (2) Rajiv Patel and Krishna Yarlagadda, "TESTABILITY FEATURES OF THE SuperSPARCTM MICROPROCESSOR," IEEE International Test conference, 1993
- (3) Hong Hao and Rick Avra, "Structured Design-for-Debug the SuperSPARCTM II Methodology and Implementation," IEEE International Test Conference, 1995
- (4) Eleanor Wu, "PEST:A Tool for Implementing Pseudo-Exhaustive Self Test," IEEE Custom Integrated Circuits Conference, 1990
- (5) Eleanor Wu and Marsha Ramer Moskowitz, "A cost-effective test data compaction scheme," IEEE Custom Integrated Circuits Conference, 1992
- (6) Rajagopal Srinivasan, Sandeep K. Gupta and Melvin A. Breuer, "Novel Test Pattern Generators For Pseudo-Exhaustive Testing," IEEE International Test Conference, 1993
- (7) Hiroshi YOKOYAMA, Xiaoging WEN and Hideo TAMAMOTO, "Random Pattern Testable Design with Partial Circuit Duplication," IEEE Asian Test Symposium, 1997
- (8) Franc Grglez, Clay Gloster and Gershon Kedem, "Hardware-Based Weighted Random Pattern Generation for Boundary Scan," IEEE International Test Conference, 1989
- (9) Samir BOUBEZARI and Bozena KAMINSDA, "Cellular Automata Synthesis based on Precomputed Test Vectors for Built-In Self-Test," IEEE/ACM International Conference on CAD, 1993
- (10) Danial J. Neebel and Charles R. Kime, "Inhomogeneous Cellular Automata for Weighted Random Pattern Generation," IEEE International Test Conference, 1993
- (11) Birgit Reeb, Hans-Joachim Wunderlich, "Deterministic Pattern Generation for Weighted Random Pattern Testing," European Design & Test Conference, 1996