논문 00-25-12A-20 한국통신학회논문지 '00-12 Vol.25 No.12A

IEEE 1149.1을 이용한 내장된 자체 테스트 기법의 구현

정회원 박 재 홍*, 장 훈*, 송 오 영**

Implementation of Built-In Self Test Using IEEE 1149.1

Jae Heung Park*, Hoon Chang*, Oh Young Song** *Regular Members*

요 약

본 논문에서는 내장된 자체 테스트(BIST: Built-In Self Test) 기법의 구현에 관해 기술한다. 내장된 자체 테스트 기법이 적용된 칩은 영상 처리 및 3차원 그래픽스용 부동 소수점 DSP 코어인 FLOVA이다. 내장된 로직 자체 테스트 기법은 FLOVA의 부동 소수점 연산 데이터 패스에 적용하였으며, 내장된 메모리 자체 테스트 기법은 FLOVA에 내장된 데이터 메모리와 프로그램 메모리에 적용하였다. 그리고, 기판 수준의 테스팅을 지원하기 위한 표준안인 경계 주사 기법(IEEE 1149.1)을 구현하였다. 특히, 내장된 자체 테스트 로직을 제어할 수 있도록 경계 주사 기법을 확장하여 적용하였다.

ABSTRACT

In this paper, we describe the implementation of BIST(Built-In Self Test) technique, which is adopted to FLOVA, a floating point DSP core used for image processing and 3D graphics. Logic BIST is applied to data path of floating point module and Memory BIST is applied to both of data memory and program memory which are embedded in FLOVA. Furthermore, boundary scan(IEEE 1149.1) technique is adopted to support board-level testing and the BIST logic we implement.

I. 서 론

오늘날 시스템의 고성능화, 고기능화 및 소형화 요구와 함께, 공정 기술의 발달에 힘입어 칩에 대한 집적도는 급속하게 증가되고 있다. 제한된 면적 안에 더 많은 소자를 집적시킬 수 있게 됨에 따라 주입력(primary input)을 통한 테스트가 불가능하게 되었고 결과적으로 칩의 품질을 보장할 수 없게 되었다. 따라서, 이러한 복잡한 칩의 높은 품질을 보장하기 위해 잘 짜인 테스트 설계 전략을 필요로 하였고 이에 따라, 내장된 자체 테스트(BIST: Built-In Self Test) 기법이 등장하였다.

BIST 기법은 테스트가 칩의 동작 주파수에 의해 수행 가능하므로 테스트 소요시간이 적게 걸리며, 테스트 응답의 비교를 위해 부수적인 테스트 장비가 필요하지 않다는 장점을 가지고 있다. 또한, BIST 기법은 내장된 로직 자체 테스트 기법(Logic BIST)과 내장된 메모리 자체 테스트 기법(Memory BIST)으로 나눌 수 있는데, Logic BIST는 테스트 패턴을 내부에서 생성하고 테스트 결과를 외부에 알려주는 매우 효율적인 테스트 기법이고, Memory BIST는 입·출력 신호를 칩의 외부에서 제어하거나 관찰하기 어려운 내장된 메모리를 테스트하는데 매우 효과적이다.

이러한 장점으로 인하여 대부분의 많은 반도체 회사에서는 내장된 자체 테스트 기법을 적용하고

* 숭실대학교 컴퓨터학과 ** 중앙대학교 전자전기공학부

논문번호 : 00230-0623, 접수일자 : 2000년 6월 23일

※ 본 연구는 산업자원부와 과학기술부 및 정보통신부에서 시행하는 주문형 반도체 개발사업의 지원을 받아 수행되었습니다. 본 연구에 사용된 H/W 및 S/W는 부분적으로 IDEC의 지원에 의한 것입니다

있다. 인텔의 경우, 80386, 80486, Pentium 칩에 내장된 자체 테스트 기법 적용하고 있다[1]. Super SPARC과 SuperSPARCII의 경우도 내장된 자체 테스트 기법을 적용하였다[2,3].

본 논문에서는 부동 소수점 DSP 코어인 FLOVA (FLOating-Point VLIW Architecture) 칩에 적용된 내장된 자체 테스트 기법을 소개한다. 테스트 용이도가 낮은 부동 소수점 연산 관련 데이터 패스 회로에는 Logic BIST를, 내장 메모리인 프로그램 메모리와 데이터 메모리에 대해서는 Memory BIST 기법을 적용하였다. 그리고, 기판상의 여러 칩들의 테스팅을 용이하게 하기 위하여 경계 주사 기법을 구현하였다. 뿐만 아니라, 경계 주사 기법의 명령어를 확장하여 적용함으로써 칩에 적용된 자체 테스트 회로를 제어 할 수 있도록 하였다.

본 논문은 다음과 같이 구성되어 있다. 2장에서는 칩에 적용된 자체 테스트 설계 기법들에 대하여 설명한다. 3장에서는 적용된 기법들의 실험 결과를 설명하고, 4장에서 결론에 대하여 설명한다.

Ⅱ. 내장된 자체 테스트 기법의 적용

2.1 경계 주사 기법(IEEE 1149.1)

기판 수준의 테스팅을 지원하기 위하여 IEEE 1149.1을 사용한다. IEEE 1149.1은 5개의 입·출력 핀(TAP: Test Access Port)과 TAP 제어기, 여러 종류의 레지스터, 그리고 명령어 디코더와 같은 기타 회로로 분류할 수 있다[4,5,6]. TAP에는 TCK(Test Clock), TMS(Test Mode Select), TRST(Test Reset), TDI(Test Data Input), TDO(Test Data Output)등이 있으며, TAP 제어기는 TMS와 TCK에 의해 동작하는 동기 유한 상태기로서, IEEE 1149.1 이 동작에 필요한 여러 신호 값들을 만들어 낸다. 또한, 레지스터에는 TDI 포트를 통하여 명령어를 래치할 수 있는 명령어 레지스터, TDI와 TDO 사이의 최단 경로를 제공하여 기판 수준의 테스트에서 테스트 시간을 줄이기 위한 bypass 레지스터, 시스템 핀과 회로 사이에 존재하여 시스템 핀의 상태를 관측하고 시스템 회로로 입·출력되는 값을 조절하거나 관측하는 경계 주사 레지스터가 있다. 그리고 사용자 선택 사항으로 32비트 크기를 갖고 버전, 부품 번호, 제조업자 식별에 관한 정보를 가지고 있는 디바이스 식별 레지스터가 있다.

IEEE 1149.1에서 지원하는 명령어는 public 명령어와 private 명령어로 구분할 수 있다. Public 명령어는 표준안에서 반드시 지원하도록 지정한 명령어이고, private 명령어는 사용자에 의해 추가되는 명령어를 말한다. Public 명령어에는 BYPASS, SAMPLE/PRELOAD, EXTEST가 있으며, 선택사항으로 IDCODE, USERCODE가 있다. 그리고 권장 명령어로 INTEST가 있다.

◆ 경계 주사 회로의 구현

FLOVA에 적용된 IEEE 1149.1은 15 비트의 명령어 레지스터를 사용하고 있으며, public 명령어인 BYPASS, SAMPLE/PRELOAD, EXTEST 명령어를 완벽하게 구현하였다.

경계 주사 레지스터에는 여러 가지 종류가 있지만 FLOVA에서는 단방향 핀을 위한 레지스터만을 사용하였다. 따라서, 양방향 핀은 입력 데이터, 출력 데이터, 제어 신호로 구성되어 있기 때문에 단방향 핀을 위한 레지스터를 그림 1과 같이 적용하였다.

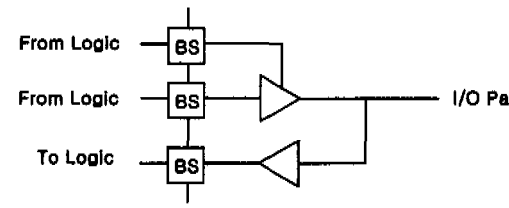


그림 1. 양방향 핀의 경계 주사 레지스터 적용

그리고 FLOVA 칩의 부동 소수점 연산 데이터 패스에는 내장된 로직 자체 테스트 회로를, 내장된 메모리에는 내장된 메모리 자체 테스트 회로를 적용하였는데, 이 회로들의 제어를 위하여 IEEE 1149.1을 이용하였다. 이를 위하여 IEEE 1149.1에 총 16개의 private 명령어를 추가하였으며, 테스트 결과값들을 제어하기 위하여 일부 회로들을 첨가하였다. 그림 2는 IEEE 1149.1과 다른 테스트 회로와의 전체적인 인터페이스를 보여 준다.

그림 2에서 LogicBIST는 부동 소수점 연산 데이터 패스를 위한 자체 테스트 회로이고, MemBIST는 메모리를 위한 자체 테스트 회로이다. IEEE 1149.1은 데이터 패스와 메모리를 테스트하기 위한 명령어가 들어오면 LogicBIST와 MemBIST에 필요한 제어 신호와 값들을 보내준다. 그러면 제어 신호와 값들을 가지고 LogicBIST와 MemBIST는 테스트를 수행하게 된다. 테스트가 끝나면 결과값이 IEEE 1149.1로 보내지고 되고, IEEE 1149.1은 들어온 결과 값들을 TDO를 통해서 칩 외부로 내보내

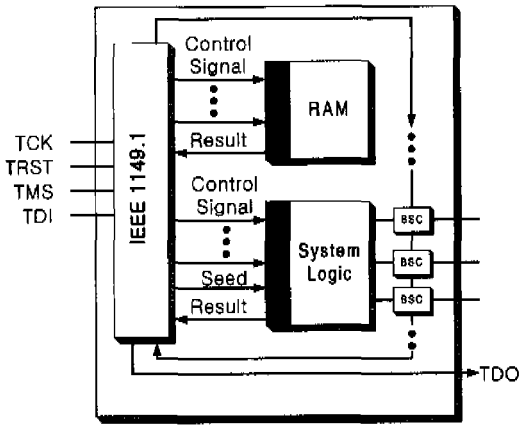


그림 2. 경계 주사 회로와 다른 테스트 회로와의 인터페이스

게 된다.

표 1은 LogicBIST와 MemBIST를 제어하기 위하여 IEEE 1149.1에 추가된 private 명령어들을 보여주고 있다. LogicBIST의 명령어들 중 Seed로 시작하는 명령어는 테스트 패턴 생성기에 seed를 적재하기 위한 명령어이며, Bist로 시작하는 명령어는 테스트 패턴을 생성하기 위한 명령어이다.

2.2 부동 소수점 연산 데이터 패스를 위한 내장된 자체 테스트 기법

내장된 자체 테스트 기법은 정상 동작 시에는 정상 입력(normal inputs)과 정상 출력(normal outputs) 단자를 통해 회로가 동작을 하게 되고, 테스트 동작 시에는 회로에 내장되어 있는 테스트 패턴 생성기(pattern generator)로 테스트 패턴을 생성하여 회로를 테스트한다. 그 후 결과를 내장되어 있는 테스트 응답 분석기(output response monitor)로 결과를 분석하여 회로의 고장 유무를 파악하게 된다[7].

FLOVA 칩에서는 테스트 용이도가 낮은 부동 소수점 연산 데이터 패스에 대하여 내장된 자체 테스트 회로를 적용하여 높은 고장 탐지율을 구할 수

표 1. 자체 테스트를 지원하기 위해 추가된 명령어

회 로	명 령 어	코 드	내 용
LogicBIST	SeedFaluEx1	15'b0000000000011101	FALU 모듈 테스트
	BistFaluEx1	15'b0000000000011001	
	SeedFaluEx2	15'b0000000000011110	
	BistFaluEx2	15'b0000000000011010	
	SeedFaluEx3	15'b0000000000011111	
	BistFaluEx3	15'b0000000000011011	
	SeedFmulEx1	15'b0000000000010101	FMUL 모듈 테스트
	BistFmulEx1	15'b0000000000010001	
	SeedFmulEx2	15'b0000000000010110	
	BistFmulEx2	15'b0000000000010010	
	SeedFmulEx3	15'b0000000000010111	
	BistFmulEx3	15'b0000000000010011	
	SeedFrec	15'b0000000000001101	FREC 모듈 테스트
	BistFrec	15'b0000000000001001	
MemBIST	MemBist	15'b0000000000000110	데이터 메모리 테스트
	PmemBist	15'b0000000000000111	프로그램 메모리 테스트

있게 함으로써 칩의 신뢰도를 향상시켰다. 고장 모델로는 회로의 물리적인 결함을 대체로 잘 반영하고 벡터 생성이 용이하여 널리 사용되고 있는 고착(stuck-at) 고장 모델을 사용하였다[8]. 그리고 의사 무작위 패턴 생성기(PRPG: PseudoRandom Pattern Generator)로는 최대 2^n-1개의 패턴을 생성하는 LFSR(Linear Feedback Shift Register)를 사용하였으며, 테스트 응답 분석기로는 다중 입력 테스트 응답 분석기(MISR: Multiple Input Signature Register)를 사용하였다.

◆ 부동 소수점 연산 데이터 패스의 구조

FLOVA 칩 코어에는 부동 소수점 연산에 관련된 3개의 모듈, FALU(Floating Point ALU), FMUL (Floating Point Multiplier), FREC(Floating Point

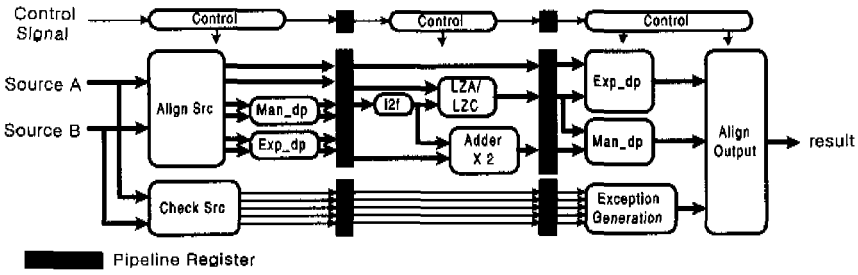


그림 3. FALU 모듈의 구조

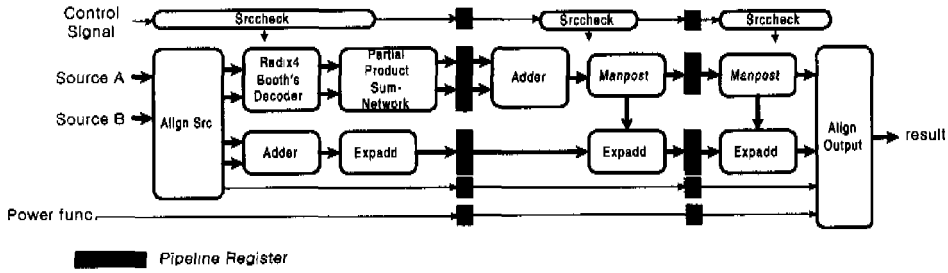


그림 4. FMUL 모듈의 구조

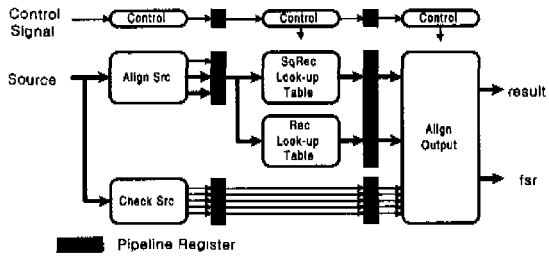


그림 5. FREC 모듈의 구조

Reciprocal)이 있다. 그리고 이 모듈들은 클럭의 falling edge에서 동작하는 파이프라인 레지스터를 갖는 구조로 되어 있다. 특히 FMUL과 FALU는 각각 32비트의 데이터를 처리할 수 있는 똑같은 모듈이 high와 low로 쌍을 이루고 있다. 그림 3, 그림 4, 그림 5은 FALU, FMUL의 high 모듈과 FREC 모듈의 구조를 보여주고 있다.

◆ 내장된 자체 테스트 회로의 구현

부동 소수점 연산을 담당하는 FALU, FMUL 모듈은 다른 고정 소수점 연산을 담당하는 모듈에 비해 회로가 복잡하고 크기 때문에 모듈 전체를 테스트했을 때에는 높은 고장 탐지율을 구하기가 어렵다. 따라서, 비교적 크기가 작은 FREC 모듈을 제외한 FALU 모듈과 FMUL 모듈은 각 파이프라인 단계별로 테스트를 수행함으로써, 적은 테스트 패턴으로 높은 고장 탐지율을 얻을 수 있도록 하였다. 이를 위하여 파이프라인 레지스터를 테스트 패턴 생성기(PRPG)와 테스트 응답 분석기(MISR)로 동작할 수 있도록 파이프라인 레지스터의 구조를 변경하였다. 그리고, 테스트 모드에서 비효율적인 테스트 패턴의 사용을 가급적 줄이고 테스트 패턴의 효율성을 극대화시키기 위해서 reseeding 방법을 사용하였는데, 이 방법을 사용하기 위하여 PRPG와 MISR로 변경된 파이프라인 레지스터에 shift 레지스터 기능과 현재의 값을 유지시키는 기능을 수행할 수 있도록 회로를 추가하였다. 즉 패턴 생성 기능, 패턴 응답 기능, shift 레지스터 기능, 현재값 유지 기능을 수행할 수 있도록 두 개의 파이프라인 레지스터를 변경하였다. 그림 6은 변경된 파이프라인 레지스터의 구조를 보여 주고 있다.

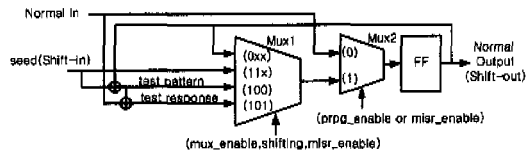


그림 6. 테스트를 지원하기 위해 변경된 파이프라인 레지스터

그림 6에서 Mux2는 정상 모드 데이터와 테스트 모드 데이터를 선택하는 mux이다. Mux1은 현재 플립플롭에 래치되어 있는 값, 테스트 패턴 생성기에 적재될 새로운 seed, 테스트 응답 분석기의 feedback 입력, 생성된 테스트 패턴 중 하나를 경우에 따라서 적절히 선택하는 mux이다. Mux1과 Mux2의 제어 신호는 IEEE 1149.1에 의해 생성된다. Normal In의 값은 회로에서 정상 동작시에 들어오는 값이고, seed 값은 IEEE 1149.1의 TDI를 통하여 외부에서 들어오는 값이다. 표 2과 표 3는 각각 파이프라인 레지스터가 PRPG로 동작할 때와 MISR로 동작할 때의 제어 신호들을 보여 주고 있다.

표 2. 테스트 동작시 PRPG로 동작할 때의 제어신호

mux_ enable	Shifting	misr_ enable	기 능
0	x	x	현재 F/F이 가지고 있는 값 유지
1	1	x	F/F에 새로운 seed 래치
1	0	0	테스트 패턴 생성

표 3. 테스트 동작시 MISR로 동작할 때의 제어신호

mux_ enable	Shifting	misr_ enable	기 능
0	x	x	현재 F/F이 가지고 있는 값 유지
1	0	1	테스트 응답 분석

◆ 테스트 과정

변경된 파이프라인 레지스터를 이용한 각 파이프라인의 단계별 테스트 과정이 그림 7, 그림 8, 그림 9에 나타나 있다. 그림 7, 그림 9와 같이 EX1이나 EX3을 테스팅하기 위해서는 Pipeline2가 패턴 생성기로 작동을 하고 Pipeline1이 응답 분석기로 작동한다. 그리고 그림 8에서는 EX2를 테스팅하기 위해 패턴 생성기로 Pipeline1이, 응답 분석기로 Pipeline2가 작동한다.

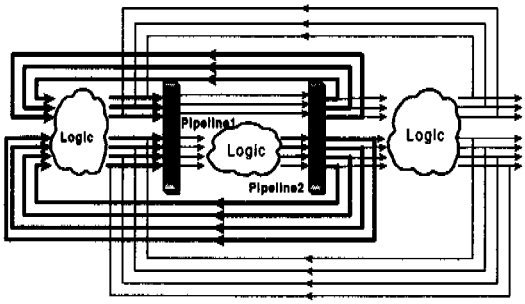


그림 7. EX1 단계 테스트 과정

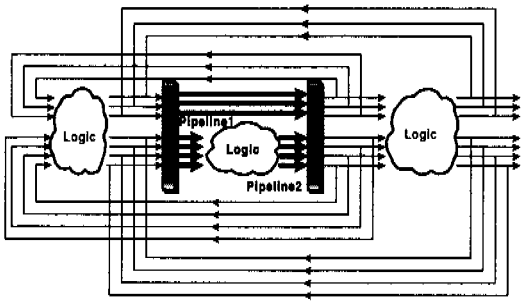


그림 8. EX2 단계 테스트 과정

예를 들어, Falu 모듈의 EX2 단계를 테스트 할 경우를 생각해 보자. 먼저, IEEE 1149.1에 Seed FaluEx2 명령어가 인가되면, Pipeline1은 PRPG로, Pipeline2는 MISR로 동작하도록 신호를 생성한다. 그러면, Seed가 그림 6의 shift-in으로부터 shift-out

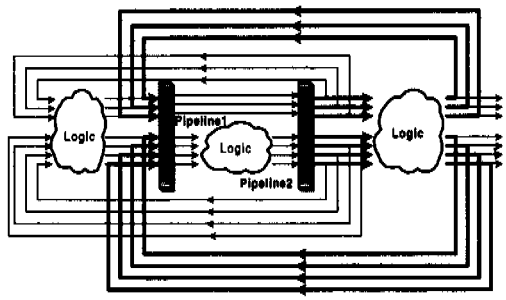


그림 9. EX3 단계 테스트 과정

을 통하여 shift 동작으로 PRPG에 적재된다. 그 후, BistFaluEx2 명령어가 다시 IEEE 1149.1에 인가되어서, PRPG가 패턴을 생성하도록 하는 신호와 MISR이 결과를 압축하도록 하는 신호가 생성된다. 그러면, PRPG는 그림 6에서와 같이 shift-in으로 들어오는 값과 이전의 값을 이용하여 랜덤한 테스트 패턴을 생성하고, MISR은 이전에 결과값과 회로를 테스트하고 Normal In을 통하여 들어오는 결과값을 압축하여 저장하게 된다. 이러한 방법으로 EX2가 테스트된다. 테스트가 완료되면 IEEE 1149.1의 TDO를 통하여 MISR에 저장되어 있는 결과를 내보낸다.

설계된 테스트 패턴 생성기를 이용해 테스트 패턴을 생성하는 절차를 살펴보면 그림 10과 같다. 먼저 쉽게 검출되지 않는 고장에 대해 ATPG (Automatic Test Pattern Generation)를 수행한다. 생성된 패턴은 패턴 생성기의 seed로 사용한다. seed가 입력되면 패턴 생성기가 테스트 패턴을 생성한다. 테스트 패턴 생성기로 생성된 패턴을 고장 시뮬레이션을 통해 휴리스틱한 방법으로 고장 탐지

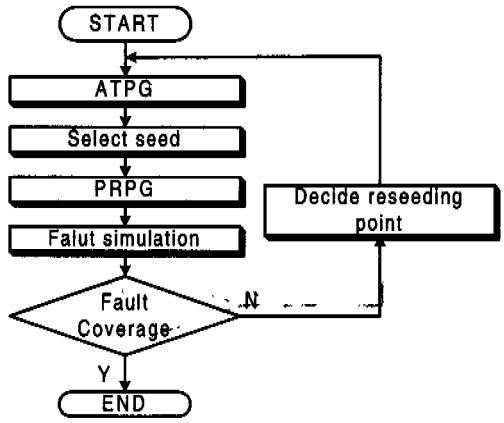


그림 10. 테스트 패턴 생성 과정

율의 증가율이 급격히 떨어지는 지점을 찾는다. 이 지점 이후의 남아있는 고장 리스트 중 가장 검출하기 어려운 고장에 대해 다시 ATPG를 수행한 후, 생성된 패턴을 테스트 패턴 생성기의 새로운 seed로 인가한다. 이러한 과정을 원하는 고장 탐지율에 도달할 때까지 반복하여 최종 테스트 패턴을 구한다.

2.3 내장된 메모리를 위한 자체 테스트 기법

칩의 집적도가 증가함에 따라, 예전에는 칩 외부에 배치되었던 메모리들이 이제는 하나의 칩에 내장되는 추세이다. 이와 같이, 내장된 메모리는 칩의 외부에서 제어하고 관찰할 수 없기 때문에 고집적화된 칩의 테스트에 있어서 가장 어려운 부분 중에 하나로 여겨지고 있다. 따라서 내장된 메모리를 테스트하기 위하여 내장된 자체 테스트 기법이 필요하게 되었다[9].

FLOVA 칩에는 데이터 메모리와 프로그램 메모리가 내장되어 있으며, 이들을 테스트하기 위하여 13N March 알고리즘을 이용하여 내장된 메모리를 위한 자체 테스트 회로를 구현하였다. 13N March 알고리즘은 주소 디코더 고장(ADF: Address Decoder Fault), 고착 고장(SF: Stuck-at Fault), 천이 고장(TF: Transition Fault), 결합 고장(CF: Coupling Fault)을 완벽하게 검출할 수 있다. 그리고, FLOVA의 메모리는 워드 단위로 읽기와 쓰기를 하므로 워드 내에서 발생될 수 있는 error masking 문제를 고려하여 배경 데이터라고 불리는 비트 패턴을 사용하였다[9,10,11]. 또한, 데이터 메모리와 프로그램 메모리가 크기만 다르고 동작 원리는 같다는 점에 착안하여 메모리 BIST 회로를 데이터 메모리와 프로그램 메모리를 위하여 개별적으로 설계를 하지 않고 공유하였으며, 구현된 자체 테스트 회로를 위한 제어기를 따로 구현하지 않고 IEEE 1149.1에 내장 메모리 테스트를 위한 명령어만을 추가함으로써 면적 오버헤드를 최소화하였다. 또한, 테스트 결과를 IEEE 1149.1의 TDO포트로 출력하게 함으로써 추가적인 핀 오버헤드가 발생하지 않았다.

◆ 메모리 BIST 회로의 구조

메모리 BIST회로는 제어 회로(CL: Control Logic), 주소 생성 회로(AGL: Address Generation Logic), 데이터 생성 회로(DGL: Data Generation Logic), 데이터 비교 회로(DCL: Data Comparison Logic)로 구성된다. 그림 11은 메모리 BIST 회로의 구조를 보여준다.

제어 회로(CL: Control Logic): 테스트 진행 과정 중에 자체 테스트 회로의 각 모듈의 동작을 제어하는 회로이다. 이 회로는 테스트의 시작과 종료를 판단하고, 테스트가 진행되는 동안 BIST 회로의 각 모듈로 적절한 제어 신호를 보낸다.

데이터 생성 회로(DGL: Data Generation Logic): 테스트 알고리즘의 진행 순서에 따라 테스트될 메모리에 배경 데이터를 공급해 주는 회로이다. 제어 회로로부터 제어 신호를 받아 각 테스트 단계별로 적절한 테스트 데이터를 생성해 낸다.

주소 생성 회로(AGL: Address Generation Logic): 테스트될 메모리의 주소를 생성하는 회로이다. 13N 알고리즘에는 주소가 증가하는 방향과 감소하는 방향이 있다. 증가하는 방향은 증가 카운터를 사용하였으며, 감소하는 방향은 증가 카운터에서 나오는 값의 보수를 취하여 구현하였다.

데이터 비교 회로(DCL: Data Comparison Logic): 메모리에서 읽어들인 테스트 결과값과 예상되는 결과값을 비교하여 메모리의 고장여부를 판단하는 회로이다. 비교된 결과는 레지스터에 따로 저장된다.

◆ 테스트를 위한 FLOVA 메모리 구조 변경

FLOVA 칩에는 각각 16K의 데이터 메모리와 8K의 프로그램 메모리를 가지고 있다. 데이터 메모리는 두 개의 뱅크(2K x 32)로 구성되어 있으며, 각 뱅크는 2K x 8의 메모리 셀 4개로 구성되어 있다.

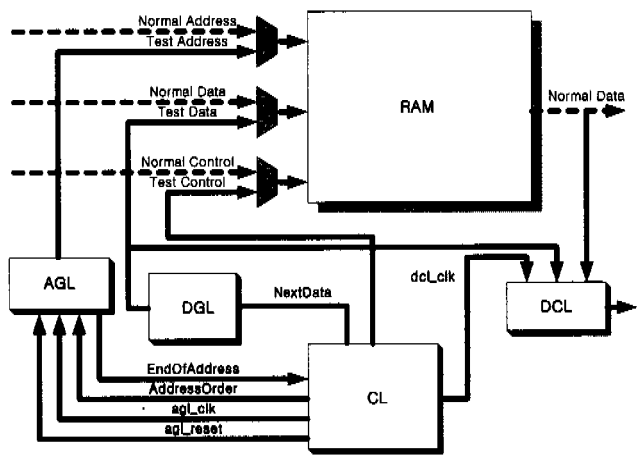


그림 11. 메모리 BIST 회로의 구조

어드레스는 모두 11비트로 최상위 비트가 0일 경우 KDST0 뱅크가, 1인 경우 KDST1 뱅크가 선택된다. 프로그램 메모리는 태크(0.5K x 23) 및 프로그램(0.5K x 64 x 2)을 저장하기 위한 메모리로 구성되어 있다. 각 메모리의 일반적인 구조는 아래의 그림 12, 그림 13과 같다.

데이터 메모리를 위한 제어신호로는 WEB, OE, CEB, RAMEN이 있으며, 주소값을 가지고 있는 address와 데이터의 입력값과 출력값을 가지고 있는 dataio가 있다. RAMEN 신호는 논리 1일 때, 읽기나 쓰기가 활성화되고, WEB 신호는 논리 1일 때 메모리에서 값을 읽는 동작을, 논리값 0일때는 메모리에 쓰기 동작을 한다. CEB 신호가 논리 0일 때 OE 신호에 의해 읽기 동작시 출력을 활성화시킨다. 프로그램 메모리는 주소값을 나타내는 PA와 데이터의 입력값과 출력값을 나타내는 DIN, DOUT이 있으며, 데이터 메모리와 같은 동작을 하는 WEB, RAMEN 신호가 있다. 프로그램 메모리는 항상 출력값이 나오고 있으므로 OE 신호는 필요하지 않다.

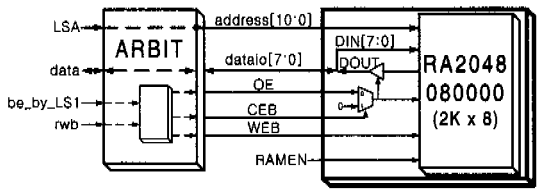


그림 12. ARBIT와 데이터 메모리의 일반적인 구조

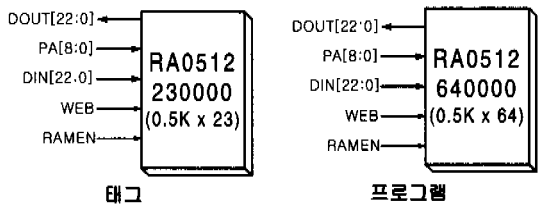


그림 13. 프로그램 메모리의 일반적인 구조

데이터 메모리와 프로그램 메모리가 같은 주소에 대하여 읽기와 쓰기를 할 때, 각 제어신호의 타이밍이 그림 14에 나와 있다. 데이터를 읽을 경우는 먼저 읽을 데이터의 유효한 주소가 address값에 인가되어 있어야 하고, WEB, OE 신호가 논리값 1이 되어 있어야 한다. 이 값들이 안정된 후에 RAMEN 신호가 논리값 1이 되면, 메모리에서 새로운 데이터를 읽어서 DOUT에 인가하게 된다. 그리고 데이터를 쓸 경우는 WEB 신호가 논리값 0으로 되고, OE, RAMEN 신호는 값을 유지하고 있어야 하며, 쓸 데이터가 DIN에 인가되어 있어야 한다.

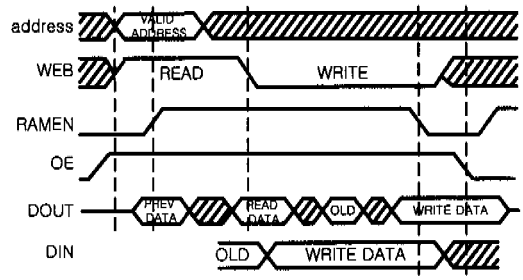


그림 14. 데이터 메모리 읽기/쓰기 동작

그림 12의 일반적인 구조를 가진 데이터 메모리를 테스트하기 위해서는 테스트에 필요한 신호들을 인가할 수 있도록 구조를 변경하여야 한다. 그림 15은 메모리 테스트를 위하여 변경된 데이터 메모리 셀의 구조를 보여준다. 메모리 BIST 회로에서 나오는 값들은 tDADDRESS, tOE, tWEB, tRAMEN이다. 경계 주사 회로에서 나오는 tDMenBistEnable 신호가 논리 1이 되면, 메모리 테스트를 하겠다는 의미이고, 따라서 메모리 BIST 회로에서 나온 값들이 선택되어 메모리에 인가된다. 또한 외부에서 직접 값이 들어오는 TEST_MODE 값에 의하여 tRAMEN 값이 메모리에 인가된다. 이렇게 메모리 BIST 회로에서 나오는 값들이 메모리에 인가되면서 메모리 테스트가 수행되는 것이다. 그림 16은 변경된 프로그램 메모리 셀 구조이다. 데이터 메모리와 마찬가지로, tPMemBistEnable 신호가 논리 1이 되어서, 메모리 BIST 회로에서 나오는 tPA, tDIN, tWEB, tRAMEN 값들에 의해 프로그램 메모리 테스트가 수행된다.

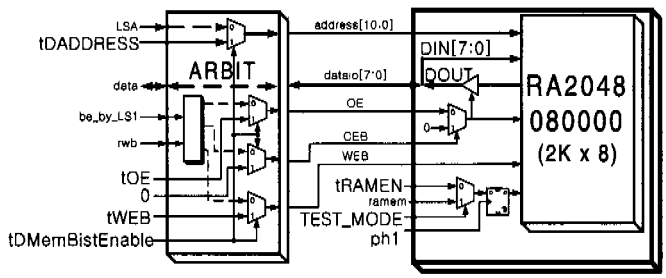


그림 15. 메모리 테스트를 위하여 변경된 데이터 메모리 구조

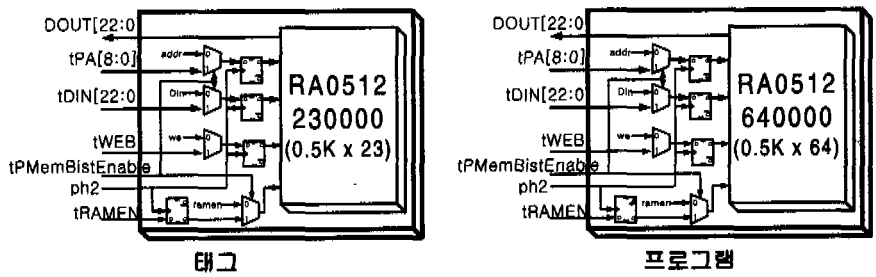


그림 16. 메모리 테스트를 위하여 변경된 프로그램 메모리 구조

그림 17은 데이터 메모리와 프로그램 메모리, 메모리 BIST 회로의 전체적인 구조를 보여준다. 메모리 BIST 회로인 MemBIST에서 메모리 테스트에 필요한 모든 값들을 생성한다. DRAM은 그림 15의 메모리 셀로 구성되어 있으며, PCACHE는 그림 16의 메모리 셀로 구성되어 있다.

Ⅲ. 실험결과

본 논문에서 구현한 기법들은 Verilog-HDL을 이용하여 설계하였으며, Synopsys 툴과 현대 hcb60 라이브러리를 이용하여 합성하였다. 그리고 회로 검증을 위한 시뮬레이터는 Cadence사의 Verilog-XL을 사용하였으며, ATPG와 폴트 시뮬레이션은 sunrise 툴을 사용하였다.

3.1 경계 주사 회로(IEEE 1149.1)

그림 18과 그림 19은 IEEE 1149.1 회로의 TAP 제어기 상태의 변화와 TAP 제어기의 출력 파형을 보여주고 있다. 그림 18은 테스트 데이터 레지스터에 관련된 TAP 제어기의 상태 변화와 출력 신호를 보여주고 있다. 그림 18의 UpdateDR 신호는 TAP 제어기의 상태가 Update-DR 상태에 들어갔을 때 falling edge에서 ~TCK의 파형을 생성하고 있다. enable 신호는 TAP 제어기의 상태가 Shift-DR 상태에 들어갔을 때 falling edge에서 논리 1이 된다. 즉 TDO 포트로 출력되는 값은 falling edge에서 유효하게 됨을 알 수 있다. sel 신호는 Select-DR Scan 상태가 되었을 때 rising edge에서 논리 0이 됨으로써 테스트 레지스터의 출력이 TDO 포트로 연결되게 한다. ClockDR 신호는 Capture-DR 상태

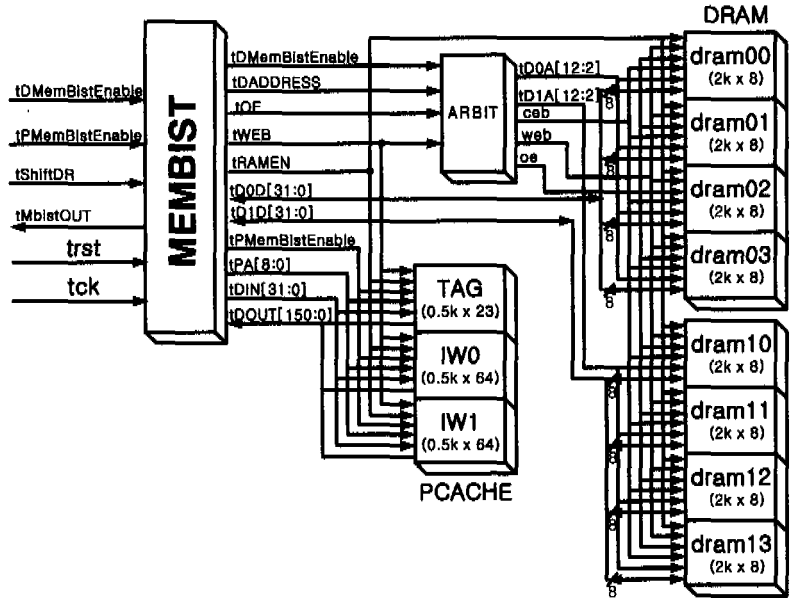


그림 17. FLOVA 칩의 메모리 BIST 회로의 구조

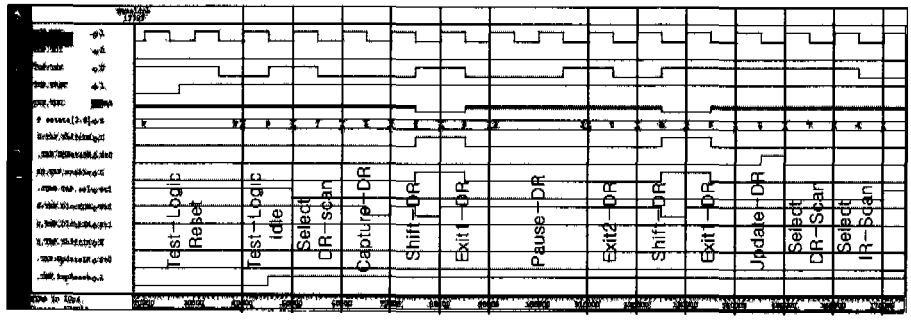


그림 18. 테스트 데이터 레지스터를 위한 TAP 제어기의 동작

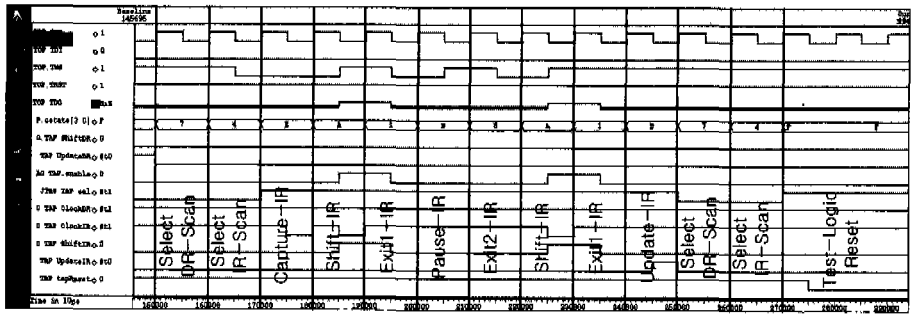


그림 19. 명령어 레지스터를 위한 TAP 제어기의 동작

와 Shift-DR 상태에서 TCK와 같은 위상을 갖는 클럭을 생성한다. 그림 19는 명령어 레지스터에 관련된 TAP 제어기의 상태 변화와 출력 신호를 보여주고 있으며, 명령어 레지스터에 관련된 출력 신호를 생성하는 점을 제외하고 동작은 테스트 데이터 레지스터에 관련된 TAP제어기의 동작과 동일하다.

3.2 데이터 패스를 위한 내장된 자체 테스트 기법

FLOVA 칩의 부동 소수점 연산 모듈의 신뢰도 향상을 위하여 내장된 자체 테스트 기법의 적용과 이를 이용한 95% 이상의 고장 탐지율을 얻을 수 있는 테스트 패턴 생성을 목표로 하였다. 부동 소수점 연산 모듈의 경우, FREC 모듈을 제외하고는 모듈 전체에 대해 테스트를 수행하기에는 회로의 크기가 크고 복잡하기 때문에 파이프라인 단계 별로 나누어 테스트를 수행하였다. 이로 인해 높은 고장 탐지율을 얻을 수 있는 테스트 패턴을 생성할 수 있었다. 표 4는 각 모듈의 단계별 고장 탐지율을 보여준다. 표 4에서 "reseed" 열의 "random"은 무작위 테스트 패턴을 생성하기 위해 테스트 패턴 생성기에 seed로 실린 전체 패턴의 수를 말하며, "deterministic"은 deterministic 패턴을 생성하기 위해 테스트 패턴 생성기에 실린 패턴의 수를 말한다. "패턴" 열은 전체 테스트 패턴의 수를 말한다.

그림 20은 FREC 모듈의 한 파이프라인 단계에 대해 테스트를 수행하는 과정을 보여주고 있다. 수행된 명령어는 SeedFrec와 BistFrec이다. 그림 20에서 1인 구간은 SeedFred 명령어가 IEEE 1149.1 회로의 명령어 레지스터에 적재되는 과정이다. 2인 구간은 테스트 패턴 생성기에 seed가 적재되는 구간이고, 3인 구간은 BistFrec 명령어가 IEEE 1149.1 회로의 명령어 레지스터에 적재되는 구간이다. 4인 구간은 테스트 패턴 생성기가 테스트 패턴을 생성하는 구간이며, 5인 구간은 IEEE 1149.1의 TDO 포트로 테스트 응답 분석기에 래치된 테스트 결과값을 출력하는 구간이다.

3.3 내장된 메모리를 위한 자체 테스트 기법

그림 21은 FLOVA 칩의 내장된 데이터 메모리를 테스트하는 과정을 보여준다. 그림 21의 1인 구간은 데이터 메모리 주소를 3번지로 축소하여 13N 알고리즘을 수행하는 부분을 나타낸다.

그리고 데이터 메모리가 워드 단위이기 때문에 배경 데이터가 필요하다. 구간 1부터 4까지는 각각 배경 데이터를 가지고 13N March 알고리즘을 적용하여 테스트를 수행하는 과정이다.

표 4. Fault grading 결과

모듈명	단계	reseed		패턴	고장 탐지율 (%)	테스트 시간 (clock)	회로 사이즈 (2-input NAND)
		random	deterministic				
FALU	Ex1	7	127	1003	99.69	18691	47351
	Ex2	4	84	840	99.62	12456	
	Ex3	7	153	1233	98.34	22353	
FMUL	Ex1	2	153	336	97.67	20796	54017
	Ex2	2	66	523	99.97	9499	
	Ex3	4	197	518	97.41	27050	
FREC	All	1	127	4789	98.01	21685	5994

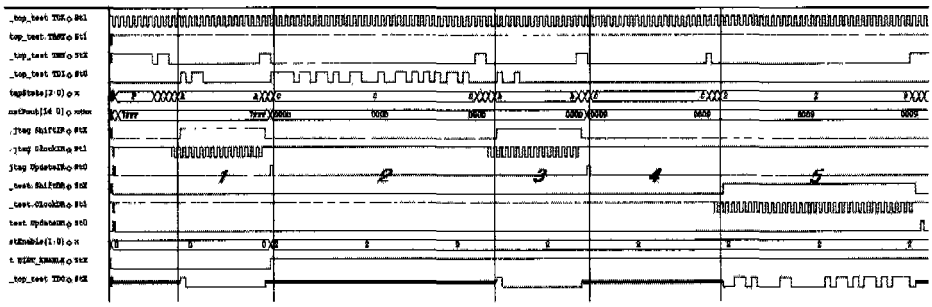


그림 20. FREC 모듈의 테스트 수행 과정

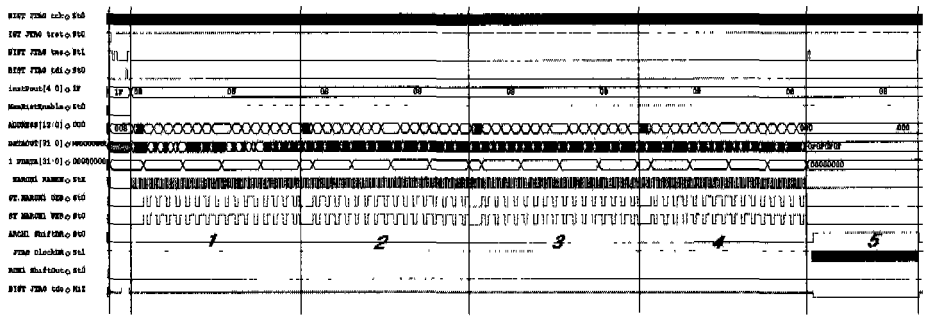


그림 21. 데이터 메모리를 테스트하는 과정

구간 5는 메모리 BIST 회로의 dcl 모듈에 있는 테스트 결과를 TDO 포트로 출력하는 과정을 보여준다. tdo의 값이 모두 0이므로 메모리 테스트가 성공적으로 수행되었음을 알 수 있다.

프로그램 메모리를 테스트하는 과정은 데이터 메모리를 테스트하는 과정과 동일하다.

Ⅳ. 결 론

본 논문에서는 상용화에 목적을 둔 고성능 부동소수점 DSP 코어인 FLOVA(FLOating-Point VLIW Architecture) 칩에 적용된 내장된 자체 테스트 기법을 소개하였다.

기판 수준의 테스트를 위하여 IEEE 1149.1 표준안을 Verilog-HDL을 이용하여 구현 및 검증을 완료하였다. 그리고, 부동 소수점 연산 모듈의 테스트를 위해 14개의 명령어와 내장된 메모리 테스트를 위해 2개의 명령어를 추가하여 기능을 확장하였다.

부동 소수점 연산 모듈에는 Logic BIST를 적용하였다. 이를 위해서 부동 소수점 연산 모듈의 구조를 파악하고, 부동 소수점 연산 모듈을 위한 최적의 내장된 자체 테스트 기법을 구현하였다. 구현한 자체 테스트 회로는 테스트 패턴 생성기의 seed를 칩 외부에서 공급할 수 있게 함으로써 deterministic 패

턴을 생성하기 위한 부가적인 하드웨어의 추가를 없게 하였으며, 비효율적인 테스트 패턴의 사용을 가급적 줄이고 테스트 패턴의 효율성을 극대화시켰다.

FLOVA에 내장된 메모리 테스트를 위하여 Memory BIST를 구현하였다. 구현된 자체 테스트 회로는 메모리의 ADF, 고착 고장(SAF), 천이 고장(TF), 결합 고장(CF)을 검출 할 수 있다. 그리고, 데이터 메모리와 프로그램 메모리의 크기가 다를 뿐 메모리의 동작 원리는 동일하다는 것에 착안하여 자체 메모리 테스트 회로를 공유함으로써 테스트로 인한 면적 오버헤드를 최적화하였다.

특히, 경계 주사 회로를 이용하여 데이터 패스를 위한 내장된 자체 테스트 회로와 내장 메모리를 위한 자체 테스트 회로의 제어를 수행함으로써 자체 테스트 회로의 제어를 위한 부수적인 하드웨어의 추가를 피할 수 있게 하였고, 테스트 패턴의 인가 및 테스트 결과의 출력을 경계 주사 회로의 TDI 포트와 TDO 포트를 이용하게 함으로써 자체 테스트 회로로 인한 핀 오버헤드를 최적화하였다.

참 고 문 헌

[1] Wayne Needham and Nags Gollakota, "DFT Strategy For Intel Microprocessors," *In Proc. IEEE International Test Conference*, pp. 396-409, 1996.

[2] Rajiv Patel and Krishna Yarlagadda, "Testability Features Of The SuperSPARC™ Microprocessor," *In Proc. IEEE International Test Conference*, pp. 773-781, 1993.

[3] Hong Hao and Rick Avra, "Structured Design-for-Debug the SuperSPARC™ II Methodology and Implementation," *In Proc. IEEE International Test Conference*, pp. 175-183, 1995.

[4] IEEE Std 1149.1-1990(including IEEE Std, 1149.1a-1993), *IEEE Standard Test Access Port and Boundary-Scan Architecture.*

[5] Test Technology Standards Committee, *IEEE Standard Test Access Port and Boundary Scan Architecture*, IEEE Computer Society Press, 1990.

[6] H. Bleeker, P. ven den Eijnden, and F. de Jong, *Boundary-Scan Test*, Kulwer Academic Publishers, 1993

[7] H. Fujiwara, *Logic Testing and Design for Teatability*, The MIT Press, 1985.

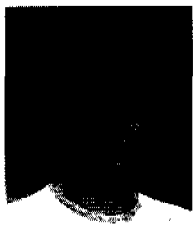
[8] E. Poage, "Derivation of optimum tests to detect faults in combinational circuits," *Int, Proc. Symp. On Mathematical Theory of Automata*, pp.483-528, 1963.

[9] A. J. Goor, *Testing Semiconductor Memories: Theory and Practice*, John Wiley & Sons Ltd., 1991.

[10] R. Dekker, F. Beenker, and L. Thijseen, "A Realistic Self-Test Machine for Static Random Access Memories," *In Proc. IEEE International Test Conference*, 1988.

[11] Pinamki Mazumder and Kanad Chakraborty, *Testing and Testable Design of High-Density Random-Access Memories*, Kluwer Academic Publishers, 1996.

박 재 흥(Jae Heung Park)



1999년 : 숭실대학교 컴퓨터학부 졸업(B.S).
1999년 : 숭실대학교 대학원 컴퓨터학과 석사과정.
<주관심 분야> 컴퓨터 구조, CAD. VLSI 설계. VLSI 테스팅.

장 훈(Hoon Chang)

1987년 : 서울대학교 전자공학과 졸업(B.S.).
1989년 : 서울대학교 전자공학과 졸업(M.S.).
1993년 : University of Texas at Austin 박사학위 취득.
1991년 : IBM Inc. 1993년 Motorola Inc. Senior Membger of Technocal Staff.
1994년 : 숭실대학교 컴퓨터학부 조교수.
<주관심 분야> 컴퓨터 시스템, VLSI 설계, VLSI 테스팅

송 오 영(Oh Young Song)

1980년 2월 : 서울대학교 전기공학과 학사.
1982년 2월 : 한국과학기술원 전기 및 전자공학과 석사.
1992년 2월 : University of Massachusetts at Amherst, 전기 및 컴퓨터공학과 박사.
1982년 3월~1985년 5월 : 국방부 기술연구 사무관.
1991년 9월~1992년 10월 : Intergraph Corp. Electronics 수석연구원.
1992년 1월~1993년 11월 : IBM Microelectronics 수석연구원.
1994년 1월~1994년 8월 : 삼성전자 LSI사업부 수석연구원.
1994년 9월~현재 : 중앙대학교 전자전기공학부 부교수.
<주관심 분야> VLSI시스템 설계 및 테스트