Journal of The Institute of Electronics and Information Engineers Vol.52, NO.6, June 2015 http://dx.doi.org/10.5573/ieie.2015.52.6.070
ISSN 2287-5026(Print) / ISSN 2288-159X(Online)

논문 2015-52-6-10

FPGA 경계 스캔 체인을 재활용한 FPGA 자가 테스트 회로 설계

(A Design of FPGA Self-test Circuit Reusing FPGA Boundary Scan Chain)

윤 현 식*, 강 태 근*, 이 현 빈**

(Hyunsik Yoon, Taegeun Kang, and Hyunbean Yi©)

요 약

본 논문은 FPGA 내부의 경계 스캔 체인을 자가 테스트 회로로써 재활용하기 위한 FPGA 자가 테스트 회로 설계 기술을 소개한다. FPGA의 경계 스캔 체인은 테스트나 디버깅 기능뿐만 아니라 각 셀에 연결되어 있는 입출력 핀의 기능을 설정하기 위해서도 사용되기 때문에 일반적인 칩의 경계 스캔 셀보다 매우 크다. 따라서, 본 논문에서는 FPGA 경계 스캔 셀의 구조를 분석하고 소수의 FPGA 로직과 함께 테스트 패턴 생성과 결과 분석이 가능하도록 설계한 BIST(built-in-self-test) 회로를 제시한다. FPGA의 경계 스캔 체인을 자가 테스트를 위하여 재사용함으로써 면적 오버헤드를 줄일 수 있고 보드상에서 프로세서를 사용한 온-라인(on-line) 테스트/모니터링도 가능하다. 실험을 통하여 오버헤드 증가량과 시뮬레이션 결과를 제시한다.

Abstract

This paper introduces an FPGA self-test architecture reusing FPGA boundary scan chain as self-test circuits. An FPGA boundary scan cell is two or three times bigger than a normal boundary scan cell because it is used for configuring the function of input/output pins functions as well as testing and debugging. Accordingly, we analyze the architecture of an FPGA boundary scan cell in detail and design a set of built-in self-test (BIST) circuits in which FPGA boundary scan chain and a small amount of FPGA logic elements. By reusing FPGA boundary scan chain for self-test, we can reduce area overhead and perform a processor based on-board FPGA testing/monitoring. Experimental results show the area overhead comparison and simulation results.

Keywords : FPGA, Built-in self test, BIST, Boundary-scan chain, IEEE 1149.1, I/O block

Ⅰ. 서 론

반도체 제조공정의 고집적화로 반도체 칩(ASIC (Application Specific Integrated Circuit) 또는 SoC (System-on-Chip))의 크기와 전력 소모가 줄고, 따라서 가격도 낮아졌다. 하지만, 칩 내부의 소자 간 간격이 좁아져 간섭에 의한 오류가 증가하고 있다. 정상 동작 중에도 동작환경에 의한 외부간섭과 긴 동작시간에 의한 스트레스로 인하여 갑작스런 오동작이 발생할 수 있다

* 학생회원, ** 정회원, 한밭대학교 컴퓨터공학과
(Department of Computer Engineering, Hanbat National University)
© Corresponding Author(E-mail: bean@hanbat.ac.kr)
※ 이 논문은 2014년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No. NRF-2014R1A1A1002661).
Received : March 25, 2015 Revised : April 20, 2015
Accepted : May 31, 2015

[1, 2]. 이러한 문제를 해결하기 위하여 칩이 시스템에 장착된 후에도 스스로 고장을 모니터링 할 수 있는 기술이 많이 연구되고 있다[3~4]. 프로그램이 가능하고 개발 시간을 단축시킬 수 있어 여러 분야에서 사용되고 있는 Field Programmable Gate Array(FPGA) 역시 경쟁적으로 최신의 반도체 제조공정을 사용하여 제작되므로 위에서 언급한 신뢰성을 저하 시키는 상황에 노출되어 있다.

본 논문에서는 보드(board)에 장착되어 있는 FPGA의 고장 유무 검사를 보드 자체에서 할 수 있는 온-라인(on-line) 테스트/모니터링 구조를 제시한다. FPGA의 자가 테스트를 위해서, Built-in self test(BIST)의 사용을 고려할 수 있다. 일반적인 BIST는 Test Pattern Generator(TPG)를 통하여 생성된 테스트 패턴을 테스트 대상 회로(circuit-under-test, CUT)에 인가하고 그 회로의 출력인 테스트 결과를 Output Response Analyzer(ORA)를 통하여 압축 또는 분석하여 기대 결과와 비교함으로써 자가 테스트를 수행한다. 하지만, FPGA내에 TPG와 ORA 회로를 추가적으로 설계해야 하고 FPGA 내부 또는 외부에 테스트 제어 메커니즘도 필요하기 때문에 많은 설계 및 면적 오버헤드가 발생한다. 따라서, 본 논문에서 프로세서와 FPGA가 함께 장착되어 동작하는 보드 상에서 프로세서를 이용한 BIST 제어와 일반 동작 중에는 사용되지 않는 FPGA내의 Boundary-Scan Chain(BSC)을 재활용한 BIST 회로 설계 방법을 제시한다. BSC를 사용하여 TPG와 ORA 회로를 구현함으로써 오버헤드를 감소할 수 있으며 외부 제어 없이 보드상의 프로세서를 통해 FPGA의 자가 진단할 수 있도록 함으로써 온-라인 테스트가 가능하다.

본 논문의 구성은 다음과 같다. Ⅱ장에서는 Altera FPGA의 구조와 기존 FPGA 자가 테스트 방법을 소개하고, Ⅲ장에서 제안하는 프로세서 기반 온-보드 FPGA 자가 테스트 구조와 및 FPGA내에 BIST 회로 설계 방법을 설명한다. Ⅳ장에서 실험결과를 제시하고, Ⅴ장에서 결론을 맺는다.

Ⅱ. 관련 연구

1. FPGA 구조

그림 1은 SRAM 기반의 Altera FPGA의 구조로써, n x n의 Logic Array Block(LAB)과 수직/수평의

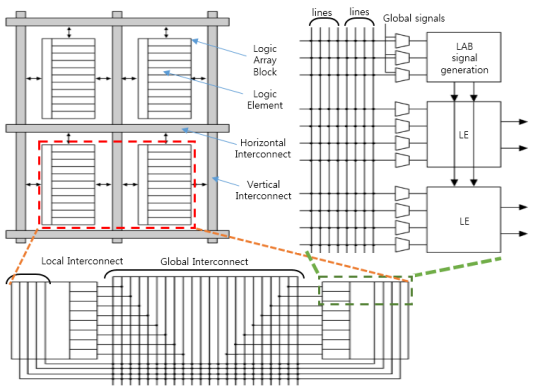


그림 1. FPGA 내부 구조
Fig. 1. FPGA Internal Architecture.

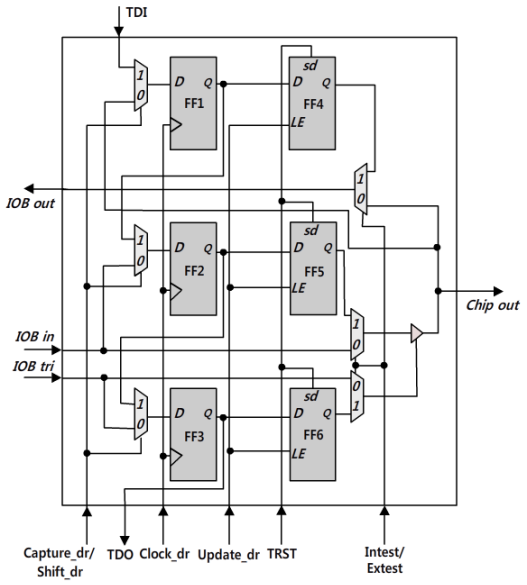


그림 2. 입력 출력 블록(IOB)
Fig. 2. Input Output Block(IOB).

Interconnect Resources(IRs)로 구성되며, LAB들은 전역(global) IR을 통해 서로 연결된다. 하나의 LAB는 16개의 Logic Element(LE)로 구성되며, LE들은 Local IR(LAB lines와 Local lines)과 LAB signal generation 블록에 의해 연결 구조와 기능이 설정된다. LE는 Altera FPGA의 기본 로직으로써 입력은 Global IRs 로부터 공유 되고 출력은 독립적인 라인을 사용하며, Look-up table(LUT), 플립-플롭, 다중화기(Multiplexer, MUX)와 간단한 논리 게이트로 구성되어 있다. Altera FPGA 또한 일반적인 칩과 같이 BSC를 가지고 있다. 하지만 FPGA의 입출력 단자는 용도(입력, 출력, 양방향 버스등)에 따라 설정할 수 있어야 하므로 FPGA의 BSC를 구성하는 기본 셀인 Input-Output Block(IOB)

는 일반적인 boundary-scan 셀과는 달리 그림 2와 같이 매우 복잡하다[5]. 하나의 IOB는, FPGA 내부 로직의 입출력 연결(IOB in, IOB out), 3상 버퍼(tri-state buffer) 제어(IOB tri), BSC와 Test Access Port Controller(TAPC)를 통한 테스트/디버깅등의 기능을 수행하기 위하여 총 6개의 플립-플롭과 6개의 MUX로 구성된다. FPGA 종류마다 약간씩 차이는 있으나 이 구조를 기반으로 한다. 이처럼 큰 자원을 재활용하여 BIST 회로의 일부로 활용함으로써 자원 활용도와 자가 테스트 회로의 신뢰성을 높이는 것이 본 논문의 목표 중 하나이다. IOB의 구조적 특징과 활용 방법은 III장에서 자세히 설명한다.

2. FPGA 자가 테스트

BIST는 TPG를 이용하여 생성한 패턴을 회로에 주입하고 ORA를 통해 압축된 패턴의 최종결과를 예상한 결과와 비교하는 방법으로 Automatic Test Equipment (ATE)를 사용한 외부 테스트와 비교하여 많은 테스트 비용을 절감 할 수 있다. [6]은 FPGA에서 기능이 구현된 부분을 Working 영역, 그 외의 부분을 Self-testing 영역으로 정의하고, Self-testing 영역에 테스트 회로를 설계하여 Working 영역을 테스트 하는 구조를 제시하였다. 테스트시에도 정상 동작을 유지하기 위하여, Working 영역을 여러 부분으로 나누고 각 부분(part-under-test, PUT)을 순차적으로 테스트 하되 테스트 시작 전에 해당 PUT의 기능을 이미 테스트 된 영역에 옮겨 Working 영역에 참여하도록 연결을 재구성하였다. 기존의 여러 방식[7~11]에서 이와 같이 FPGA 내부에서 기능에 참여하지 않는 영역을 테스트 회로로 사용한다. 하지만, 기능 이외에 추가되는 면적 오버헤드가 크고 그 만큼 테스트 회로 영역에 오류가 있을 가능성도 높아진다. [12]는 FPGA 내부의 자가 테스트 회로를 로직 컴포넌트가 아닌 내부 메모리를 사용하여 설계하는 방식을 제시하였다. FPGA 내부 메모리 구조의 특징을 분석하여 BIST 회로, 즉, Linear Feedback Shift Register(LFSR)와 Multiple Input Signature Register(MISR) 설계를 위하여 FPGA 내부 메모리를 활용 할 수 있는 방법을 제시하였다. 하지만, 여전히 XOR 연결을 위해 적지 않은 LE가 사용되고 FPGA에 설계되는 기능이 메모리를 많이 사용하는 경우에는 테스트 회로 구현이 불가능하다.

III. 프로세서 기반 온-보드 FPGA 테스트

1. 보드 및 FPGA 내부 테스트 구조

그림 3은 프로세서 기반 테스트가 가능한 보드 구조이다. 일반적으로 보드 상의 프로세서와 FPGA(또는 ASIC)는 테스트나 디버깅을 목적으로 외부에서 접근할 수 있도록 IEEE 1149.1 (JTAG)을 통하여 연결되어 있다. 테스트 제어 신호인 TMS, TCK, TRST는 공유되고 데이터 입출력인 TDI-TDO는 BSC를 통하여 직렬로 연결된다. 이러한 기본 구조에, 본 논문에서는 TAP Signal MUX 블록을 추가한다. External Test 신호에 의해 외부의 접근을 통한 테스트시와 보드 상에서 프로세서에 의한 테스트시 테스트 신호 경로를 변경 할 수 있도록 한다. 이와 같이 보드를 구성함으로써 프로세서가 필드에서도 IEEE 1149.1 신호를 생성하여 FPGA의 BIST 회로를 제어 하는 온-라인 테스트가 가능하다.

그림 4는 제안하는 FPGA의 IOB를 재활용한 BIST 회로이다. 패턴 생성과 압축을 위한 LFSR과 MISR를 IOB의 그룹과 그 그룹에 인접한 소수의 LE를 조합하여

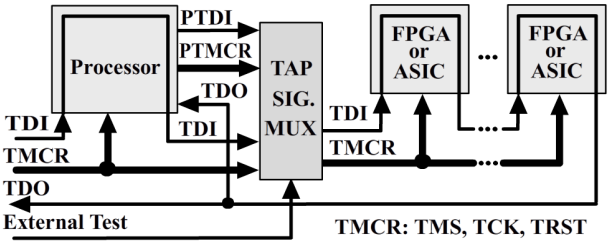


그림 3. 프로세서 기반 온-보드 FPGA 테스트를 위한 보드 구조

Fig. 3. Board architecture for processor based on-board FPGA Test.

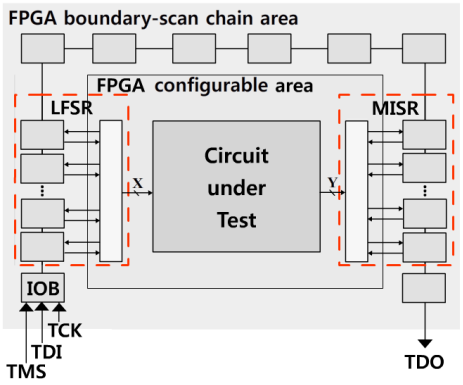


그림 4. IOB를 활용한 FPGA BIST 구조

Fig. 4. FPGA BIST architecture using IOB.

쉬프트 동작, XOR 연산, 피드백 루프(feedback loop)를 설계해야 하는데, IOB가 TAPC 신호에 의해 제어 되므로 그림 5의 IEEE 1149.1 TAPC의 상태천이에 따른 IOB의 동작과 구조의 특성을 파악해야 한다. 그림 2를 바탕으로 LFSR과 MISR의 기능 구현과 연관된 IOB의 특징을 살펴보면 다음과 같다.

(1) 직렬 테스트 데이터 입력(TDI)은 FF1에 저장되고, 그 값은 FF2로 쉬프트(Shift) 되거나 FF4로 업데이트(Update) 되어 IOB Out을 통하여 FPGA 로직으로 인가 될 수 있다.
(2) FPGA 로직의 결과는 IOB In을 통하여 FF2에 캡쳐(Capture) 되거나 FPGA의 외부로 출력(Chip out) 될 수 있다.
(3) FF2에 저장된 데이터는 FF3으로 쉬프트 될 수 있고 FF3에 저장된 데이터는 TDO를 통하여 다음 IOB의 FF1로 쉬프트 될 수 있다.

LFSR은 쉬프트와 업데이트 동작을 반복하고 MISR는 캡쳐와 쉬프트 동작을 반복한다. 하지만, TAPC의 상태천이에 따라 캡쳐(Capture DR)-쉬프트(Shift DR)-업데이트(Update DR)-캡쳐(Capture DR)의 순서를 반복하며 동시에 제어되므로 LFSR과 MISR 각각 불필요한 동작을 거치게 되어 의도하지 않은 결과가 발생할 수 있다. 따라서, 업데이트시에 IOB Out으로 출력된 값을 캡쳐시에 직접 또는 XOR를 통하여 IOB In으로 캡쳐 될 수 있도록 LE를 구성하여 업데이트-캡쳐 동작이

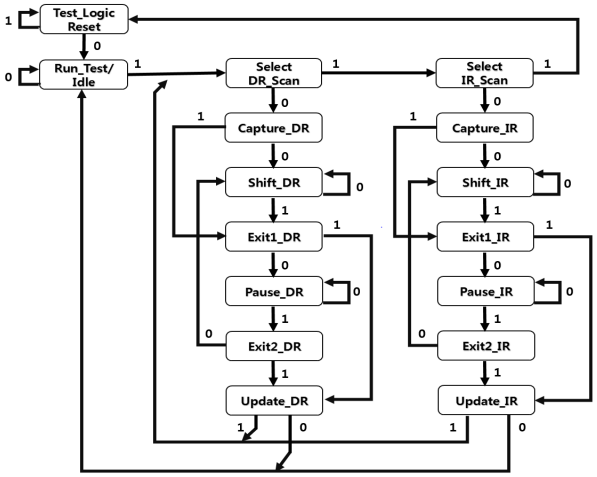
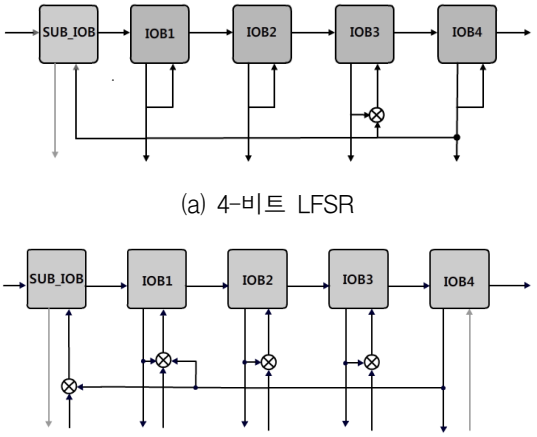


그림 5. TAPC 상태 천이도
Fig. 5. TAPC State Diagram.



(a) 4-비트 LFSR

(b) 4-비트 MISR

그림 6. 4-비트 LFSR과 MISR 예
Fig. 6. Examples of 4-bit LFSR and MISR.

쉬프트와 같은 동작이 수행되도록 함으로써 해결 가능하다. 그림 6은 4-비트 LFSR과 MISR의 예이다. 각 IOB의 출력(IOB Out)이 로직으로 인가됨과 동시에 다시 직접 또는 필요에 따라서 XOR을 통하여 자신의 IOB In으로 연결되어 있다. 마지막 IOB(IOB4)에서 첫 IOB(IOB1)로 값을 전달하는 피드백 루프를 구현하기 위해서는 마지막 IOB의 출력과 첫 IOB의 입력이 연결되도록 내부 IR을 사용해야한다. 그러나 IOB는 'TDI → TDO'의 직렬연결을 갖고, TAPC의 Capture → Shift → Update의 상태 천이를 반복하는 상황에서, 피드백루프의 데이터를 IOB1의 IOB In을 통해 캡쳐 하면, 두 번의 쉬프트 동작으로 IOB2로 이동되고 업데이트시 IOB2 Out에서 출력되어 결과적으로 3bit의 데이터만을 출력하게 된다. 따라서, 한 개의 IOB를 추가하고 추가된 IOB에서 데이터를 먼저 받게 함으로 4 bit의 데이터를 출력하도록 한다. 이를 통해 n-비트 LFSR과 MISR가 n+1개의 IOB와 XOR 연산과 입출력 연결을 위하여 소수의 LE로 구성 될 수 있다.

2. Test Control

프로세서에 의한 FPGA 테스트를 수행하기 위해서 프로세서는 1) BIST회로를 포함한 테스트 연결 구성 데이터를 FPGA에 다운로드를 하고 2) 테스트 프로그램을 구동하여 TAP 신호를 생성하고 3) 테스트 결과와 기댓값을 비교하는 과정을 수행한다. 그림 7은 본 논문에서 제시하는 프로세서 기반 FPGA 테스트 구조의 테스트 과정이다. 테스트 모드가 시작되면 프로세서에 의해

FPGA 내에 테스트를 위한 구성(Test Configuration)을 하고, TAPC 제어를 통하여 SEED를 입력 한다. LFSR에 의한 패턴 생성은 반복 횟수는 미리 설정된 반복 횟수(Last SHIFT)가 되면 완료한다. 주입할 SEED가 남아 있으면 다음 SEED의 주입과 패턴 생성을 반복하는데, 본 논문에서 제시하는 구조에서는 LFSR과 MISR가 FPGA BSC의 일부로써 TAPC에 의해 동시에 제어되므로 LFSR에 SEED 주입시 MISR의 압축 결과의 추출과 초기화를 동시에 진행해야 한다. 기본적으로 테스트는 모든 SEED의 주입이 완료되고, 모든 패턴의 결과가 압축 될 때까지 진행한다. 그러나 두 번째 SEED의 입력부터 TDO로 추출되는 MISR의 압축 결과의 관측이 예상한 값과 일치 하지 않으면 테스트 프로그램의 설정에 따라 더 이상의 테스트를 수행하지 않고 중단 할 수도 있을 것이다. 테스트가 완료되면 프로세서는 진단 결과를 사용자에게 알리고 FPGA를 다시 정상 동작을 할 수 있도록 연결을 재구성한다. 그림 8은 IOB내에서의 쉬프트 동작과 IOB간 데이터 이동을 위한 TAPC 신호를 보여준다. 어느 한 IOB의 FF1(그림 2 참조)에 저장되어 있는 데이터는 Update_DR에 IOB out으로 출력되었다가 Capture_DR에 IOB in을 통해 바로 또는 XOR를 통해 FF2로 입력(그림 6 참조)되어 저장된다. 이후 두 사이클의 Shift_DR 상태에 쉬프트 동작이 2회 수행되어 다음 IOB의 FF1로 전달된다.

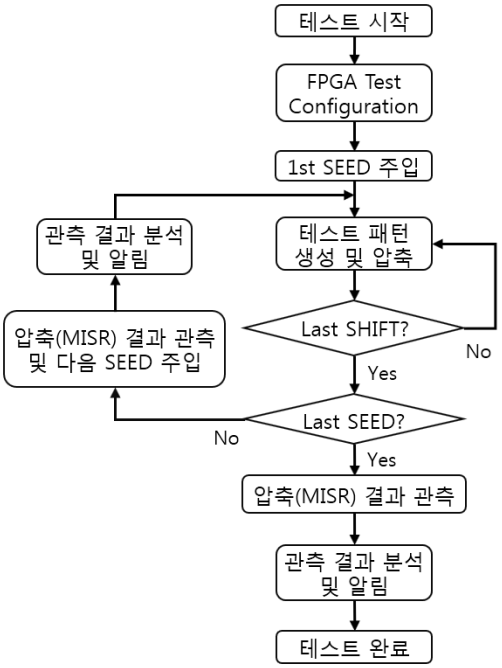


그림 7. 테스트 과정 흐름도
Fig. 7. Test process flow.

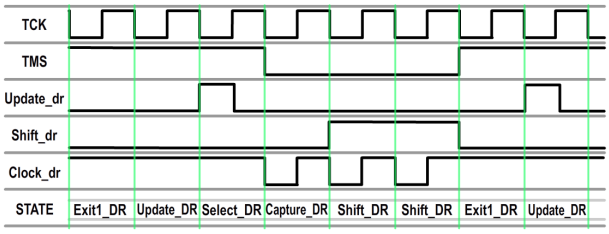


그림 8. 패턴 생성 및 압축 동작의 TAPC 제어
Fig. 8. TAPC control of pattern generation and compression.

IV. 실 험

실험을 위하여 ISCAS-85의 c6288을 벤치마크 회로로 사용한다. c6288은 반가산기(half-adder)와 전가산기(full-adder)의 조합으로 구성된 16x16 곱셈기로, 입력과 출력 비트 수가 각각 32개 이므로 기능 검증 패턴 생성을 위한 32-비트의 LFSR과 MISR가 요구된다. 또한 LFSR은 "000...0"을 제외한 모든 패턴, 즉, 2^{32}-1개의 패턴을 생성할 수 있도록 생성다항식 $1+x^2+x^6+x^7+x^{32}$을 사용한다. BIST 적용으로 발생하는 FPGA의 면적 오버헤드를 비교하기 위해 벤치마크 회로에 기존의 BIST와

표 1. 면적 오버헤드 비교
Table 1. Comparison of area overhead.

Circuits	Total Logic Elements	Additional Area Overhead (%)
c6288(benchmark)	503	-
(with) Conventional LRSR/MISR	592	17.69
(with) Proposed LFSR/MISR	537	6.75

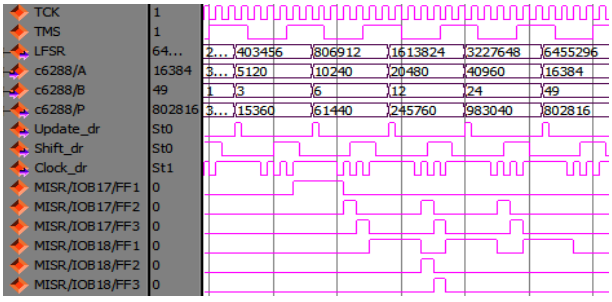


그림 9. MISR에 의한 테스트 결과 압축 과정
Fig. 9. Compression process by the proposed MISR.

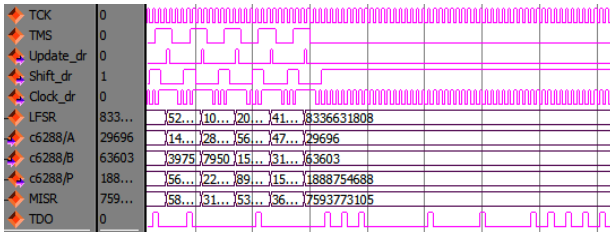


그림 10. 패턴 생성/압축 완료후 TDO를 통한 결과 관측
Fig. 10. Observation of test result through TDO after pattern generation and compression.

제안하는 BIST를 설계하였다. 표 1은 일반적인 BIST와 제안하는 BIST의 면적 오버헤드를 나타낸다. 제안하는 BIST는 32-비트의 입출력의 회로에 적용하기 위해 기존 BIST보다 LFSR과 MISR에 1-비트 씩, 총 2-비트가 추가되지만, BSC를 기반으로 패턴 생성을 위한 XOR과 내부 연결을 제외하고 레지스터를 사용하지 않기 때문에, 모든 회로를 내부에 구성하는 기존 BIST보다 62%의 면적 오버헤드를 감소 할 수 있다.

그림 9는 LFSR에서 생성된 패턴이 두 개의 16-비트 입력인 c6288/A와 c6288/B로 분리되고 곱셈기의 출력 c6288/P이 압축되는 과정을 보여주고 있으며, 그림 10은 테스트 종료 후 MISR에 저장된 압축 데이터가 IOB를 통해 TDO로 쉬프트 되는 과정을 보여주고 있다. LFSR에서 생성된 패턴은 Update_DR 신호와 함께 인가되고(LFSR_out) 벤치마크 회로를 통과한 결과 데이터는 캡쳐 상태에서 MISR로 입력된다. 이러한 과정을 검증하기 위하여 MISR에서 17, 18번째 IOB의 연속된 5개의 플립-플롭(IOB17/FF2-3과 IOB18/FF1-3)의 값을 샘플링 하였다. 결과 데이터는 Capture_DR 상태에서 한 차례 앞선 Update_DR 신호를 통해 출력된 MISR의 초기 값과 압축되어 IOB17/FF2에 입력된다. 이 후, 연속된 두 번의 쉬프트 동작을 통해 다음 IOB의 첫 번째 플립-플롭(IOB18/FF1)으로 전달되고, Update_DR 상태에서 출력 되면, 그 다음 캡쳐 동작에서 새로운 결과 데이터와 XOR 게이트를 통해 압축되어 IOB18/FF2에 저장됨을 확인 할 수 있다. Shift_DR 상태에서 벗어나 그 다음 캡쳐 동작을 수행하기 위하여 Capture_DR 상태로 이동해야 하고, Exit1_DR-Update_DR-Select_DR_Scan 상태를 거쳐야 하기 때문에 IOB의 FF1에 쉬프트 된 후 압축 데이터가 FF2에 캡쳐 되기까지 시간이 걸리는 것을 확인 할 수 있다.

V. 결 론

본 논문에서는 FPGA의 IOB를 BIST 회로로 재활용하는 방법을 제시하였다. 제안하는 BIST는 테스트 실행에 있어, LFSR과 MISR를 위해 XOR과 피드백을 위한 최소의 FPGA 내부로직과 BSC를 이용하는 구조로, 내부 로직이 검증되지 않은 상태에서 BIST가 Fault를 가지고 있을 수 있는 로직을 사용할 가능성을 감소시킴으로써 테스트의 신뢰성을 높인다. 또한 회로의 기능 테스트뿐만 아니라 FPGA의 구조적 테스트에 적용이 가능하며, 칩의 디버깅과 연결구성을 위해 매우 큰 영역을 차지하지만 정상 동작 시에는 사용되지 않은 IOB를 재활용함으로써 FPGA 테스트를 위한 오버헤드를 크게 줄일 수 있다.

REFERENCES

[1] Sony Corporation, *Sony Semiconductor Quality and Reliability Handbook*, 1st edition, Oct 2000.
[2] Renesas Electronics, *Semiconductor Reliability Handbook*, Rev.1.01, Nov 2008.
[3] Y. Park, I. Choi, and S. Kang, “IEEE std. 1500 based an Efficient Programmable Memory BIST,” *Journal of The Institute of Electronics Engineers of Korea*, Vol. 50, NO. 2, pp. 114-121, Feb 2013.
[4] Y. Kim, I. Kim, and H. Min, “BIST structure based on new Random Access Scan architecture for Low Power Scan Test,” *ITC-CSCC*, pp. 812-815, July 2009.
[5] “IEEE 1149.1 JTAG Boundary-Scan Testing in Altera Devices”, *Application Note*, no. 39, June 2005.
[6] M. Abramovici, C. Stroud, C. Hamilton, S. Wijesuriya, and V. Verma, “Using Roving STARs for On-Line Testing and Diagnosis of FPGAs in Fault-Tolerant Applications,” *Int. Test Conf.(ITC)*, pp. 973 - 982, 1999.
[7] C. Hsu, and T. Chen, “Built-in Self-Test Design for Fault Detection and Fault Diagnosis in SRAM-Based FPGA,” *IEEE Trans. on Instrumentation and Measurement*, Vol. 58, Issue. 7, pp. 2300-2315, Feb 2009.
[8] P. Gadde, and M. Niamat, “FPGA Memory Testing Technique using BIST,” *IEEE Int. Midwest Symp. on Circuits and Systems*

(MWSCAS), pp. 473-476, Aug 2013.
[9] N. Das, P. Roy, and H. Rahaman, "Built-in-self-test technique for diagnosis of delay faults in cluster-based field programmable gate array," *IET, on Computers & Digital Techniques*, Vol. 7, No. 5, pp. 210-220, Sep 2013.
[10] S. Vemula, and C. Stroud, "Built-in self-test for programmable I/O buffers in FPGAs and SoCs," *IEEE Southeastern Symp. on System Theory*, pp. 534-538, Mar 2006.
[11] S. Rehman, M. Benabdenbi, and L. Anghel, "BIST for Logic and Local Interconnect Resources in a Novel Mesh of Cluster FPGA," *IEEE Int. Symp. on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT)*, pp. 296-301, Oct 2013.
[12] K. Ito, T. Yoneda, Y. Yamato, K. Hatayama, and M. Inoue, "Efficient Scan-Based BIST Architecture for Application-Dependent FPGA Test," *IEICE Tech.* Vol. 113, no. 353, pp. 1-6, 2013.

저 자 소 개

윤 현 식(학생회원)
2013년 한밭대학교 컴퓨터공학과 학사 졸업.
2013년~현재 한밭대학교 컴퓨터공학과 석사 과정.
2013년~2014년 한국전자부품 연구원 연구원.
<주관심분야 : Design for Testability, Digital design, Embedded system software>

강 태 근(학생회원)
2013년 한밭대학교 컴퓨터공학과 학사 졸업.
2015년 한밭대학교 컴퓨터공학과 석사 졸업.
2015년~현재 한밭대학교 컴퓨터공학과 박사 과정.
<주관심분야 : Design-for-Reliability, 오류정정 코드, FPGA 설계>

이 현 빈(정회원)
2001년 한양대학교 전자컴퓨터 공학과 학사 졸업.
2003년 한양대학교 전자컴퓨터 공학과 석사 졸업.
2007년 한양대학교 전자컴퓨터 공학과 박사 졸업.
2002년~2007년 한국전자부품연구원 연구원.
2007년~2009년 미국 UMass 연구원.
2009년~2011년 일본 NAIST 연구원.
2012년~현재 한밭대학교 컴퓨터공학과 교수.
<주관심분야 : Design-for-Reliability, 회로/메모리 노화 모니터링, 오류정정코드>