

BIST를 지원하는 경계 주사 회로 자동 생성기

정희원 양 선 응*, 박 재 흥**, 장 훈***

Automatic Boundary Scan Circuits Generator for BIST

Sunwoong Yang*, JaeHeung Park**, Hoon Chang*** *Regular Members*

요 약

본 논문에서 구현한 GenJTAG은 기관 수준의 테스트를 위한 정보와 BIST(Built-In Self Test)에 대한 정보를 입력으로 받아 verilog-HDL 코드로 기술된 경계 주사 회로를 자동 생성해 주는 설계 자동화 툴이다. 대부분의 상용 툴들은 생성된 회로를 게이트 수준의 회로로 제공하기 때문에 사용자가 선택적으로 사용할 수 있는 BIST 관련 명령어를 회로에 추가하기가 어려운데 반해, 본 논문에서 구현한 툴은 사용자가 정의한 정보에 의해 BIST 관련 명령어를 지원할 수 있는 behavioral 코드의 경계 주사 회로를 생성하여 준다. 또한 behavioral 코드를 제공함으로써 사용자에게 의한 수정을 용이하도록 하였다.

ABSTRACT

In this paper, we implemented the GenJTAG, a CAD tool, which generates a code of boundary scan circuit supporting a board level testing and a BIST(Built-In Self Test) written in verilog-HDL. A boundary scan circuit code that supports users' own BIST instructions is generated based on the informations from the users. Most CAD tools hardly allow users to add their own BIST instructions because the generated code is described in gate-level. But the GenJTAG generates a behavioral boundary scan circuit code so users can easily make a change on the generated code

1. 서 론

집적도의 증가에 의한 칩 테스트의 어려움은 점점 증가하고 있으며, 이러한 문제점을 해결하기 위해서 테스트 용이화 설계(Design For Testability) 기법을 적용한 칩 설계 기법이 도입되게 되었다. 테스트 용이화 설계 기법은 칩 내부 노드들의 관측 용이도(observability)와 조절 용이도(controllability)가 향상되도록 설계하는 방법으로 스캔 기법, BIST(Built-In Self Test) 기법, Ad-hoc 기법 등이 있다^[1,2]. 특히, BIST 기법은 칩의 동작 주파수에서 테스트가 수행 가능하므로 테스트 소요시간이 적게

결리며, 테스트 응답의 비교를 위해 부수적인 테스트 장비가 필요하지 않다는 장점을 가지고 있다. 이러한 장점으로 인하여 많은 칩에서 BIST 기법을 적용하고 있다^[3,4].

이러한 칩들을 이용하여 하나의 시스템을 구성할 때, 기관 수준의 테스트 또한 필요하다. 기관 수준 테스트는 표면장착(surface mount) 기술의 발달로 칩간의 연결선들이 표면으로 드러나지 않은 경우가 증가함에 따라 테스트의 어려움도 증가하게 되었다. 따라서, 이를 해결하기 위한 설계 기법이 필요하게 되었으며 1980년대 초부터 유럽과 북미의 산학연 협동체인 JTAG(Joint Test Access Group)이 기관 수준에서의 테스트를 지원할 수 있는 테스트 용이

* 숭실대학교 대학원 컴퓨터학과 컴퓨터구조 연구실 (swyang@watt.ssu.ac.kr)

** 숭실대학교 대학원 컴퓨터학과 컴퓨터구조 연구실 (jhpark@watt.ssu.ac.kr)

*** 숭실대학교 컴퓨터학부 (hoon@comp.ssu.ac.kr)

논문번호 : K01163-0721, 접수일자 : 2001년 7월 21일

* 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.

화 설계 기법을 연구하게 되었다. 이를 기초로 하여 IEEE에서는 IEEE 1149.1이라는 표준안을 제정하게 되었다^{5,6}. 많은 칩에서 기판 수준 테스트를 위해 IEEE 1149.1 표준안, 즉 경계 주사 기법을 적용하고 있으며^{7,8}, 요즘은 널리 사용되고 있는 PCI 버스 표준에서도 경계 주사 기법을 지원하기 위한 신호를 정의하고 있다⁹. 또한 경계 주사 기법은 기판 수준의 테스트 뿐만 아니라 사용자의 요구에 따라 다른 테스트 기법을 지원하도록 회로를 확장할 수 있게 하였다.

본 논문에서는 기본적인 경계 주사 기법을 지원 하면서, DFT를 위하여 많이 사용되고 있는 BIST를 지원할 수 있는 경계 주사 회로를 자동 생성해 주는 GenJTAG에 대하여 소개한다. 현재 경계 주사 회로를 자동 생성할 수 있는 상용 툴들이 많이 개발된 상태이다. 그러나 이러한 툴들의 일부는 게이트 수준의 회로를 생성하여 줌으로써, 사용자가 회로의 확장을 필요로 할 때 경계 주사 회로에 반영하는 것이 용이하지 않게 되어있다. GenJTAG에서는 behavioral 코드를 생성함으로써 사용자의 요구 사항이 변경된 경우 경계 주사 회로에 쉽게 반영할 수 있도록 하였다.

본 논문은 다음과 같이 구성되어 있다. 2장에서는 경계 주사 기법을 간단히 소개하고, 3장에서는 경계 주사 회로의 설계 자동화 툴인 GenJTAG에 대하여 설명한다. 4장에서는 구현한 GenJTAG과 GenJTAG에 의해 생성된 회로의 동작 검증에 대하여 기술한다.

II. 경계 주사 기법

경계 주사 기법은 기판 수준의 테스트를 지원하기 위해 1990년도에 IEEE에서 제정한 표준안이다. 경계 주사 회로는 TDI(Test Data Input), TDO(Test Data Output), TCK(Test Clock), TMS(Test Mode Select), TRST(Test Reset), 총 5개의 입출력 핀을 가지고 있다. 각 핀의 기능은 표 1에 나와있다.

경계 주사 회로는 TAP 제어기, 명령어 레지스터, bypass 레지스터, 명령어 디코더, 경계 주사 셀, 그리고 기타 회로들로 구성되어 있다. TAP 제어기는 TMS와 TCK에 의해 동작하는 동기 유한 상태기로서 경계 주사 회로의 동작에 필요한 여러 제어 신호들을 생성한다. 명령어 레지스터는 TDI 포트를 통하여 입력되는 경계 주사 회로를 위한 명령어를 래치하는 레지스터이다. Bypass 레지스터는 TDI와 TDO 사이의 최단 경로를 제공하여 기판 수준의 테

스트에서 테스트 시간을 줄일 수 있게 해주는 레지스터이다. 경계 주사 셀은 테스트를 수행할 때 주사 체인을 형성하는 레지스터이다. 그리고 32비트 크기를 갖고 버전, 부품번호, 제조업체 식별에 관한 정보를 갖고 있는 디바이스 식별 레지스터와 사용자가 정의한 레지스터를 필요에 따라 추가할 수 있다. 그림 1과 그림 2는 기본적인 경계 주사 회로의 구조와 TAP 제어기의 상태를 보여주고 있다¹⁰.

경계 주사 기법에서 지원하는 명령어는 public 명령어와 private 명령어로 구분할 수 있다. Public 명령어와 private 명령어로 구분할 수 있다. Public 명령어는 표준안에서 지정한 명령어이고, private 명령어는 사용자에 의해 추가되는 명령어를 말한다. Public 명령어에는 반드시 지원해야 하는 BYPASS, SAMPLE/PRELOAD, EXTEST 명령어가 있으며, 선택사항으로 IDCODE, USERCODE 명령어가 있다. 그리고 권장 명령어로 INTEST 명령어가 있다.

표 1. 경계 주사 회로의 입출력 핀의 종류 및 기능

핀 이름	기능
TDI	테스트 데이터와 명령어를 인가하기 위한 핀으로 TDI로 인가되는 값은 TCK의 rising edge에서 유효하다.
TDO	테스트된 결과나 TDI로 인가된 값을 출력하기 위한 핀으로 TCK의 falling edge에서 유효하다.
TCK	IEEE 1149.1에서 정의한 테스트 회로에 클럭을 인가하기 위한 핀이다.
TMS	테스트 동작을 제어하는 TAP 제어기의 상태 제어 신호를 인가하기 위한 핀으로 TMS 포트에 인가되는 값은 TCK의 rising edge에서 유효하다.
TRST	TAP 제어기의 비동기 초기화를 위한 optional 핀으로 TRST에 논리 0이 인가되면 TAP 제어기는 Test-Logic- Reset 상태가 된다.

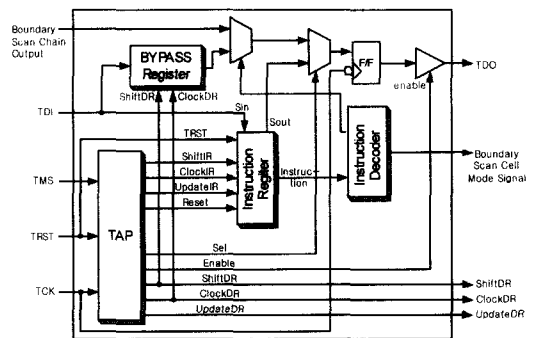


그림 1. 경계 주사 회로의 구조

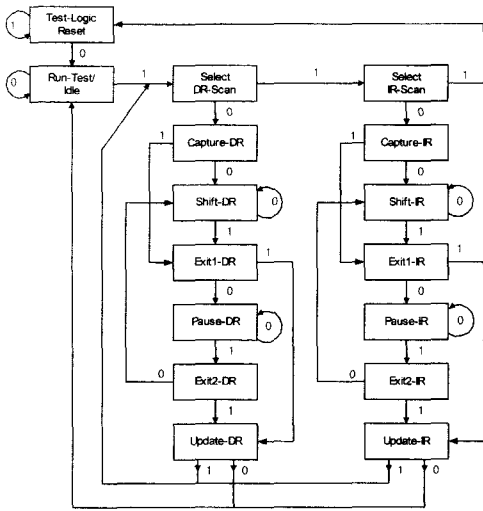


그림 2. TAP 제어기의 상태도

III. GenJTAG 구현

본 논문에서 구현한 GenJTAG은 IEEE 1149.1에 제정된 public 명령어와 private 명령어인 BIST 명령어를 모두 지원할 수 있는 확장된 경계 주사 회로를 자동으로 생성한다. 생성된 경계 주사 회로는 verilog-HDL을 이용하여 behavioral 코드로 기술되어 있다.

1. GenJTAG 구성

본 논문에서 구현된 GenJTAG은 생성할 경계 주사 회로에 대한 정보를 입력받아 설정파일을 생성하고, 생성된 설정 파일을 이용하여 verilog-HDL로 기술된 경계 주사 회로를 생성한다. 그림 3은 GenJTAG을 이용한 경계 주사 회로의 생성 흐름을 보여주고 있다.



그림 3. GenJTAG을 이용한 경계 주사 회로 생성 흐름도

GenJTAG은 사용자가 GUI(Graphic User Interface)를 이용하여 경계 주사 회로에 필요한 public 명령어와 BIST를 동작시키는 private 명령어를 정의할 수 있게 하였다. 그림 4는 경계 주사 회로에 필요한 정보가 입력되어 있는 GenJTAG의 GUI 화면을 보여주고 있다. 그림 4의 ①은 생성될 경계 주사 회로의 모듈 이름을 입력하는 부분이며,

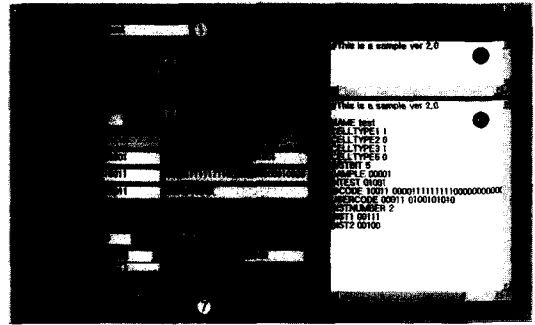


그림 4. GenJTAG의 정보 입력 화면

②는 경계 주사 회로가 제어할 경계 주사 셀의 타입을 선택하는 부분이다. GenJTAG에서는 IEEE 1149.1에서 제정된 7가지 타입 중 1, 2, 3, 5번 타입만을 지원하도록 설계하였다. ③은 public 명령어를 입력하는 부분이다. 먼저 Size 필드 부분에 명령어의 사이즈를 입력한다. 그러면 표준안에 의해 고정되어 있는 BYPASS, EXTEST 명령어는 각각 입력된 명령어 사이즈만큼 각 명령어 필드에 자동으로 000...00, 111...11 이 입력된다. 그리고 SAMPLE/PRELOAD, INTEST, IDCODE, USERCODE 명령어는 사용자가 직접 입력을 하게 되어 있다. 만일, 사용자가 명령어 필드에 값을 입력하지 않으면, 그 명령어는 사용하지 않는 것으로 인식한다. IDCODE 명령어가 지원되어야만 사용 가능한 USERCODE 명령어는 IDCODE 명령어가 입력되어야 USERCODE 명령어를 입력할 수 있도록 제한을 두었다. Private 명령어인 BIST 명령어는 그림 4의 ④부분에서 입력한다. BIST 명령어의 개수를 먼저 입력하면 BIST 명령어의 이름이 자동으로 BIST1, BIST2, BIST3, ... , BISTn으로 설정이 되고, 명령어마다 각각 값을 입력할 수 있도록 하였다. 생성될 경계 주사 회로에 관한 주석문은 ⑤에서 입력한다. 또한, 입력된 정보에 의해 생성되는 설정 파일의 내용을 ⑥부분에서 미리 볼 수 있도록 하였다. ⑦부분은 GenJTAG의 메뉴 버튼이다. Load는 이미 생성된 설정 파일을 읽어드리는 메뉴이고, Save는 입력한 정보를 설정 파일로 저장하는 메뉴이다. Clear 메뉴는 입력된 정보를 제거하는 기능을 하며, Run은 입력된 정보에 의한 설정 파일을 이용하여 경계 주사 회로를 생성하는 메뉴이다. 마지막으로 Quit는 GenJTAG을 종료하는 메뉴이다. 그림 4의 입력 내용을 보면 모듈 이름은 test 이며, 1번과 3번 경계주사 셀을 사용한다. 명령어 사이즈는 5

비트이며 SAMPLE(00001), INTEST(01001), IDCODE(10011), USERCODE(00011) 명령어, 2개의 BIST 명령어(BIST1: 00111, BIST2: 00100)를 사용한다. IDCODE의 값은 0000111111110000000000001000000, USERCODE는 0100101010 이다.

위와 같이 GenJTAG의 GUI에 의해 입력된 정보는 설정 파일로 저장이 된다. 설정 파일의 구성은 아래와 같다.

• 경계 주사 셀 타입

경계 주사 회로에 사용하는 경계 주사 셀 타입을 기술하는 부분이다. 사용하는 타입은 1로, 사용하지 않는 타입은 0으로 표현한다. 타입 1과 3을 사용한다면 아래와 같이 표현된다.

```
CELLTYPE1 1
CELLTYPE2 0
CELLTYPE3 1
CELLTYPE5 0
```

• Public 명령어

사용할 public 명령어에 대하여 기술하는 부분이다. BYPASS, EXTEST 명령어는 표준안에 고정되어 있기 때문에 기술하지 않는다. 사용되지 않는 명령어 또한 설정 파일에는 기술하지 않는다. 명령어의 사이즈가 3이고, 001인 SAMPLE 명령어만 사용한다면 아래와 같이 표현된다.

```
INSTBIT 3
SAMPLE 001
```

• Private 명령어

BIST 명령어에 대한 정보를 입력하는 부분이다. BIST 명령어 개수 및, 각 명령어를 기술한다. 각각 100, 101를 가지는 두 개의 BIST 명령어를 사용한다면 아래와 같이 표현된다.

```
BISTNUMBER 2
BIST1 100
BIST2 101
```

2. 생성된 경계 주사 회로의 구조 및 동작

저장된 설정 파일을 이용하여 GenJTAG은 verilog-HDL로 기술된 경계 주사 회로를 생성한다. 생성된 경계 주사 회로는 public 명령어를 지원할 뿐만 아니라 BIST 회로를 제어하기 위한 명령어들

도 지원한다. 이를 위하여 구현한 GenJTAG에 의해 생성되는 경계 주사 회로는 명령어 디코더 기능을 확장하였고, BIST 회로의 테스트 결과를 출력하기 위한 회로를 추가하였다. 그림 5는 GenJTAG에 의해 생성되는 BIST 명령어를 지원하는 경계 주사 회로의 구조를 보여준다. 그림에서 BISTEnable 신호는 명령어 레지스터에 적재된 BIST 명령어의 디코딩 후에 발생하는 신호로 칩 내의 특정 BIST 회로를 구동시키기 위한 신호이다. BIST 회로의 테스트 결과는 경계 주사 회로의 TDO 포트를 통해서 칩 외부로 출력할 수 있다. BIST 회로의 테스트 결과를 경계 주사 회로에 전달하기 위해 사용되는 포트가 BISTOut 포트이다. 그림에서 BISTOut 포트는 BISTOut_Mux라는 멀티플렉서를 통해서 특정 BIST 회로의 테스트 결과를 TDO로 출력되게 된다. BISTOut_Mux의 제어 신호는 명령어 디코더에서 현재 명령어 레지스터에 적재된 BIST 명령어를 디코딩하여 생성해 준다.

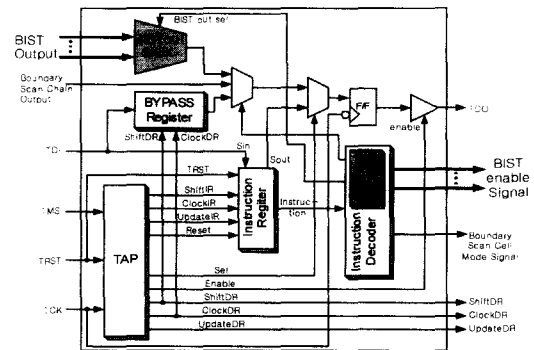


그림 5. BIST 명령어를 지원하는 확장된 경계 주사 회로 구조

그림 6은 BIST 명령어를 경계 주사 회로의 명령어 레지스터에 적재, BIST 회로를 이용한 테스트, 그리고 BIST 회로의 테스트 결과를 경계 주사 회로의 TDO 포트를 통해 칩 외부로 출력하는 과정을 파형으로 보여주고 있다. BIST 명령어는 TAP 제어기의 Shift-IR 상태에서 TDI 포트를 통해 명령어 레지스터에 적재된다. 명령어가 명령어 레지스터에 적재된 후, TMS 값을 이용하여 TAP 제어기를 Update_IR 상태로 이동시킨다. 명령어 레지스터에 적재된 명령어는 Update_IR 상태의 falling edge에서 명령어 디코더에 의해 디코딩되고 BISTEnable 신호가 활성화되고, 이 신호는 명령어 레지스터에 새로운 명령어가 적재되고 디코딩 되거나 TAP 제

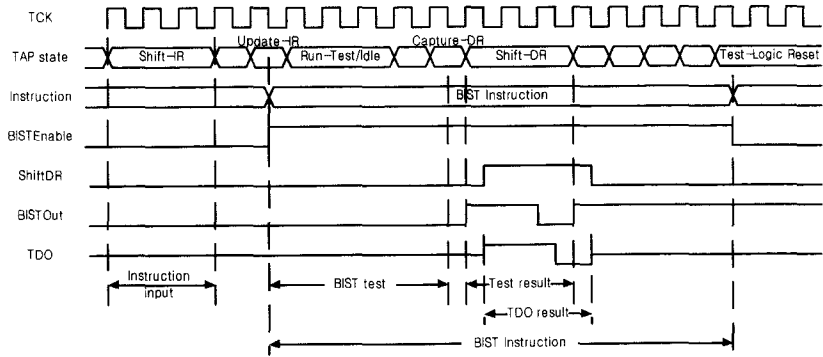


그림 6. BIST 명령어시 동작 파형

여기가 Test-Logic/Reset 상태에 들어갈 때까지 유지된다. BISTEnable 신호가 활성화되면, 선택된 BIST 회로는 TAP 제어기가 Run-Test/Idle, Select-DR, Capture-DR 상태일 동안 테스트를 수행하게 된다. BIST 회로의 테스트 결과는 Shift-DR 상태에서 ShiftDR 신호에 의해 경계 주사 회로의 TDO 포트를 통해서 칩 외부로 출력되게 된다.

IV. 실험 결과

본 논문에서는 그림 4와 같이 입력된 정보를 가지고 GenJTAG의 동작 검증을 수행하였다. 검증 결과 입력된 정보에 대하여 GenJTAG가 설정 파일과 확장된 경계 주사 회로를 정확하게 생성함을 확인하였다. 그림 7은 GenJTAG에 의해 생성된 verilog-HDL 소스 코드의 일부를 보여주고 있다.

생성된 경계 주사 회로는 verilog-XL 시뮬레이터를 이용하여 동작 검증을 하였다. 시뮬레이션을 통하여 모든 public 명령어에 대하여 정상적으로 동작

함을 확인하였다. BIST 명령어에 대한 동작 검증을 위해서는 간단한 BIST 회로를 GenJTAG에 의해 생성된 경계 주사 회로와 연결하여 수행하였다. 테스트에 사용된 BIST 회로는 ISCAS'85 시험 회로 [11]의 c1908, c6288 회로, 랜덤 패턴 생성기인 PRPG(Pseudo-Random Pattern Generator), 다중 입력 압축치 분석기인 MISR(Multiple Input Signature Register)^[12]로 구성되어 있으며, 구조는 그림 8과 같다.

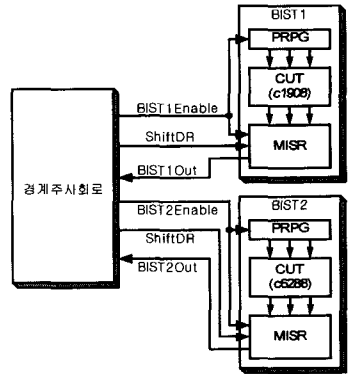


그림 8. BIST 명령어 테스트 회로 구조

PRPG와 MISR은 BISTEnable 신호에 의해 테스트를 수행하며, ShiftDR 신호에 의하여 BISTOut을 통해 테스트 결과를 경계 주사 회로에 전달한다. 경계 주사 회로에 전달된 결과는 경계 주사 회로의 TDO를 통하여 칩 외부로 출력된다. 그림 9는 테스트용 BIST 회로와 경계 주사 회로의 시뮬레이션 결과 파형을 보여주고 있다. 각 BIST 명령어마다 enable 신호가 생성되고 BIST의 결과(in)가 TDO 포트(out)를 통하여 칩 외부로 출력됨을 확인할 수 있으며, 그림 6과 비교하여 동일한 파형임을 알 수

```

module test_jtag (tck, tms, tdi, tdo, update, shiftDR, captureDR, updateDR, modeDR,
modeDR, BISTenable, BISTtest, BISTdisable, BISTtest);
input tck, tms, tdi, tdo, update; input shiftDR; input BISTtest; input BISTtest;
input shiftDR; input captureDR; input updateDR; input modeDR;
output modeDR; output tdo; output BISTenable; output BISTtest; output BISTtest;
wire [3:0] capture; wire [8:0] instPost; wire [8:0] termPost; wire [1:0] BISTenable;
wire enable; wire capture; wire out; wire shiftDR; wire capture; wire update;

tapTap (.tapTap(
.tck(tck),
.tms(tms),
.enable(enable),
.capture(capture),
.update(update),
.term(term),
.shiftDR(shiftDR),
.captureDR(captureDR),
.updateDR(updateDR),
.modeDR(modeDR),
.BISTenable(BISTenable),
.BISTtest(BISTtest),
.BISTdisable(BISTdisable),
.BISTtest(BISTtest)));

wire instPost;
function ONEZERO;
(.capture(capture),
.term(term),
.shiftDR(shiftDR),
.captureDR(captureDR),
.updateDR(updateDR),
.modeDR(modeDR),
.BISTenable(BISTenable),
.BISTtest(BISTtest),
.BISTdisable(BISTdisable),
.BISTtest(BISTtest)));

wire [3:0] capture;
function ONEZERO;
(.capture(capture),
.term(term),
.shiftDR(shiftDR),
.captureDR(captureDR),
.updateDR(updateDR),
.modeDR(modeDR),
.BISTenable(BISTenable),
.BISTtest(BISTtest),
.BISTdisable(BISTdisable),
.BISTtest(BISTtest)));

wire update;
function ONEZERO;
(.capture(capture),
.term(term),
.shiftDR(shiftDR),
.captureDR(captureDR),
.updateDR(updateDR),
.modeDR(modeDR),
.BISTenable(BISTenable),
.BISTtest(BISTtest),
.BISTdisable(BISTdisable),
.BISTtest(BISTtest)));
endmodule
    
```

그림 7. 생성된 verilog-HDL 소스 코드 일부

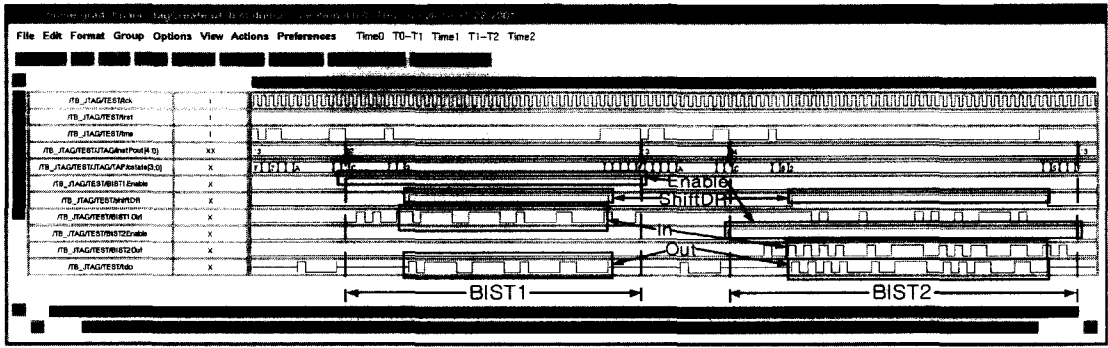


그림 9. BIST 명령어 테스트 회로의 동작 파형

있다. 그리고 Xilinx XCS40 FPGA 칩이 장착된 툴킷을 이용하여 실제 칩에서의 동작 검증도 수행하였다. 툴킷의 FPGA 칩에 GenJTAG에 의해 생성된 경계 주사 회로와 테스트용 BIST 회로를 매핑하였으며, 그림 10은 테스트용 BIST1 회로의 테스트 후 TDO로 나온 결과값 0184F3d(16진수)를 보여주는 그림이다. 그림 10의 결과 값이 그림 9의 BIST1 파형의 결과 값과 일치함으로써 회로가 정확하게 동작함을 확인하였다.



그림 10. Xilinx 툴킷을 이용한 동작 검증

V. 결론

본 논문에서 경계 주사 회로를 자동으로 생성해주는 설계 자동화 도구인 GenJTAG에 대하여 소개하였다. GenJTAG은 표준안에서 정의된 기본적인 경계 주사 기법을 지원함은 물론 사용자의 요구에 따라 BIST를 지원할 수 있도록 하였다. 또한 경계 주사 회로를 behavioral verilog-HDL의 형태로 생성함으로써 사용자의 쉽게 이해할 수 있도록 하였으며, 사용자의 요구 사항이 변경되었을 경우 회로의 직접적인 변경이 용이하게 하였다. 생성된 경계 주사 회로의 동작 검증은 테스트용 BIST 회로와 연결하고 Xilinx 툴킷을 이용하여 실제 FPGA 칩에서

수행하였다. 테스트 결과 Xilinx FPGA 칩에서 정상적으로 동작함을 확인하였다. 구현한 GenJTAG을 사용함으로써 경계주사회로의 쉽게 설계 할 수 있고, 특히 BIST와 경계 주사 회로를 연동시키고자 할 때 많은 시간과 노력을 줄일 수 있다.

참고 문헌

- [1] M. Abramovici, M. A. Breuer and A. D. Friedman, *Digital system testing and testable design*, Computer Science Press, 1990,
- [2] R. Rajsuman, *Digital Hardware Testing : Transistor-Level Fault Modeling and Testing*, Artech House Boston London, 1992.
- [3] Wayne Needham and Nags Gollakota, "DFT Strategy For Inter Microprocessors," *In Proc. IEEE Int'l Test Conf.*, pp. 396-409, 1996.
- [4] R. Raina, R. Bailey, D. Belete, V. Khosa, R. Molyneaux, J. Prado, A. Razdan, "DFT Advances in Motorola's Next-Generation 74xx PowerPCTM Microprocessor," *In Proc. IEEE Int'l Test Conf.*, pp. 131-140, 2000.
- [5] IEEE Standard 1149.1-1990, "IEEE Standards Test Access Port and boundary-scan Architecture," IEEE Standards Board, New York, 1990.
- [6] Test Technology Standards Committee, "IEEE Standard Test Access Port and Boundary-Scan Architecture," *IEEE Computer Society Press*, 1993.
- [7] D.D. Josephson, D.J. Dixon, and B.J. Arnold, "Test Features of the HP PA7100LC Processor," *In Proc. IEEE Int'l Test Conf.*, pp.

764-772, 1993.

- [8] F. Golshan, "Test and On-Line Debug Capabilities of IEEE Std 1149.1 in Ultra SPARCTM III Microprocessor," *In Proc. IEEE Int'l Test Conf.*, pp. 141-150, 2000.
- [9] 박선호, *PCI 버스 해설과 인터페이스 카드 설계*, 국제테크노정보연구소, 1999.
- [10] K. P. Parker, *The Boundary-Scan Handbook*, Kluwer Academic Publishers, Norwell MA, 1992.
- [11] Brglez, F., Fujiwara, H., "A neutral netlist of 10 combinational benchmark circuits and a target translator in fortran," *IEEE International Symposium on Circuits and Systems(ISCAS)*, pp. 677-692, June 1985.
- [12] 홍성제, 박은세, 강성호, 최호용, 장훈, *테스팅 및 테스팅을 고려한 설계*, 홍릉과학출판사, 1998

장 훈(Hoon Chang)

정회원



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 Member of Technocal Staff.

1994년 : 숭실대학교 컴퓨터학부 부교수.

<주관심 분야> 컴퓨터 시스템, VLSI 설계, VLSI 테스팅

양 선 웅(Sun-woong Yang)

정회원

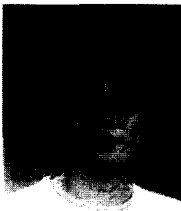


1996년 : 숭실대학교
전자계산학과 졸업(B.S.)
 1998년 : 숭실대학교 대학원
전자계산학과 졸업(M.S.).
 1998년 : 숭실대학교 대학원
컴퓨터학과 박사과정.

<주관심 분야> 컴퓨터 구조, VLSI 설계 및 테스팅, CAD

박 재 흥(Jae-heung Park)

정회원



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