논문 2004-41SD-8-12

JTAG 기반 테스트의 성능향상을 위한 PIDM(Preceding Instruction Decoding Module)

(Preceding Instruction Decoding Module(PIDM) for Test Performance Enhancement of JTAG based Systems)

윤 연 상*, 김 승 열*, 권 순 열*, 박 진 섭*, 김 용 대**, 유 영 갑**

(Yeonsang Yun, Seungyoul Kim, Soonyoul Kwon, Jinsub Park, Yongdae Kim, and Younggap You)

요 약

본 논문에서는 IEEE 1149.1 표준인 JTAG 기반 테스트 성능향상을 위한 preceding instruction decoding module(PIDM)을 제안하였다. PIDM은 test access port(TAP) 명령어 디코딩과정을 TAP 제어회로(TAP-controller) 이전에 수행하여 클럭회수를 최소화하였으며 테스트 타겟 안에서 test mode select(TMS) 같은 신호를 생성할 수 있게끔 설계되었다. CORDIC 프로세서의 테스트 시뮬레이션 결과 PIDM은 non-PIDM에 비해 15% 정도의 성능향상을 나타내었으며 TAP 제어회로의 게이트 수는 기존에 비해 48% 이상 감소하였다.

Abstract

A design of a preceding instruction decoding module(PIDM) is proposed aiming at performance enhancement of JTAG-based test complying to the IEEE 1149.1 standard. The PIDM minimizes the number of clocks by performing test access port(TAP) instruction decoding process prior to the execution of TAP-controlled test activities. The scheme allows the generation of signals such as test mode select(TMS) inside of a target system. The design employing PIDM demonstrates 15% performance enhancement with simulation of a CORDIC processor and 48% reduction of the TAP-controller's circuit size with respect to the conventional design of a non-PIDM version.

Keywords : TMS signal generator, JTAG boundary scan test, CORDIC

I. 서 론

현재 SoC 기술의 발전으로 인하여 시스템의 디버그를 위한 테스트 환경이 칩 안으로 옮겨지고 있다. JTAG 기반의 경계 스캔 테스트방식은 타겟(테스트하고자 하는 보드 또는 칩) 내부에 테스트 회로를 내장하기 적합한 구조로 이루어져 있으며, 이미 SoC 분야에서 시스템의 디버그를 위한 필수요소로 인식되었다[1]. 기존의 bed-of-nails와 같은 보드용 테스트 방식은 점차 소형화 및 집적화 되어가는 시스템의 테스트로는 그 효율성이 떨어지고 있다.

32비트 RISC 프로세서인 ARM의 경우 코어 내부에 별도의 디버깅 및 테스트 회로를 내장하여 사용자의 편의를 제공하고 있다[2]. 또한 DSP와 아날로그 분야에서도 JTAG 기반의 테스트 환경이 사용되고 있다[1]. JTAG 기반 테스트는 별도의 장비 없이 타겟과 호스트와의 인터페이스만을 고려하면 모든 테스트 환경이 완성된다는 점에서 비용절감의 효과가 크다. 하지만 동작

* 학생회원, ** 정회원, 충북대학교 정보통신공학과
(Department of Information & Communication Engineering, Chungbuk Nat'l University)
※ 본 연구는 산업자원부의 지역혁신 인력양성사업의 연구결과이며 충북대학교 컴퓨터정보통신연구소의 일부지원으로 수행되었음.
접수일자: 2004년2월9일, 수정완료일: 2004년7월26일

하고 있는 타겟의 실시간 테스트가 어렵다는 점은 JTAG 기반 테스트분야에서 해결해야할 과제로 남아있다. 최근의 JTAG 기반 방식은 테스트가 시작됨과 동시에 코어의 파이프라인을 정지시킨 후 테스트 모드에서 파이프라인을 수행하도록 설계되고 있다[3]. 테스트 모드는 정상적인 시스템 모드에 비해 파이프라인 수행 속도가 느리다는 단점이 있다. 최근 이러한 단점을 극복하기 위한 노력으로 테스트 수행 시 디버그 모듈에서의 게이트 카운트를 감소시킨 연구 결과가 나오고 있다[4-5]. 이들은 ARM 프로세서의 EmbeddedICE 와 같은 디버그 모듈의 구조를 개선함으로써 성능향상을 꾀하고 있다.

본 논문에서는 JTAG 기반 테스트의 성능향상을 위하여 디버그 모듈이 아닌 기존의 경계 스캔 방식을 효율적으로 변형하였다. 제안된 preceding instruction dec-oding module(PIDM)은 IEEE 1149.1 표준에서 제시하는 test access port(TAP) 명령의 디코딩 방식을 간략화 하였다. 그리고 디버거 제작 시 어려움을 주었던 TMS신호의 생성을 타겟 내의 TMS신호발생기가 생성하도록 제안하였다.

본 논문은 다음과 같이 구성되었다. II장은 JTAG 기반 경계 스캔 방식을 소개하고 개선할 사항을 파악하였으며 III장은 본 논문에서 제안한 PIDM의 설계과정을 설명하였다. IV장은 PIDM 모듈을 CORDIC 프로세서에 적용하여 테스트한 결과와 성능분석을 다루었고 마지막으로 V장을 통해 본 논문의 결론을 지었다.

II. JTAG 기반 테스트

본 절은 JTAG 기반 테스트 시스템의 전반적인 내용을 소개한 뒤 테스트 성능향상을 위해 개선되어야 할 점을 지적하였다.

1. 테스트 시스템

JTAG 기반 테스트 시스템의 개요를 그림 1에서 도시하였다. 테스트 시스템은 크게 호스트와 프로토콜 변환기 그리고 테스트 타겟으로 구분된다[3]. 호스트는 타겟의 TAP(test access point)으로 입력될 신호들을 생성한다. TAP은 TCK, TMS, nTRST, TDI, TDO 이렇게 5핀으로 구성되어 있다.

.TCK는 JTAG 경계 스캔을 위한 클럭 신호이다. TDI와 TDO는 코어 주변의 경계 스캔 체인으로 테스트 입력을 인가하거나 출력을 확인하기 위한 신호이다. TMS 신호는 타겟의 내부에 내장되어 있는 TAP 제어회로를

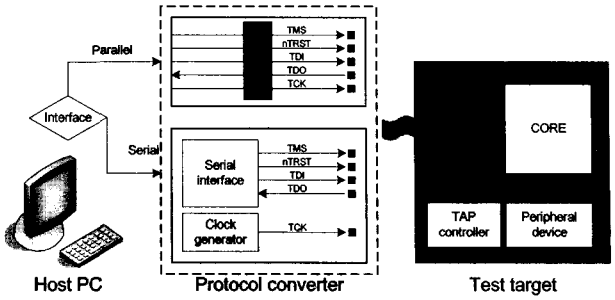


그림 1. JTAG 기반 테스트 시스템
Fig. 1. JTAG based test system.

제어하는 용도로 사용된다. nTRST는 시스템의 초기화를 위해 인가되는 신호이다. 호스트는 TAP 신호들을 모두 생성한다. 직렬전송방식에서는 프로토콜 변환기는 TCK의 발생을 위하여 별도의 클럭발생기가 추가된다.

타겟은 JTAG 기반 테스트를 위하여 별도의 회로를 내장하고 있다. 테스트회로는 경계 스캔 체인과 이들을 제어하는 TAP 제어회로(TAP-controller)로 나누어진다. TAP 제어회로는 호스트로부터 인가된 TMS신호와 TCK에 의하여 동작상태를 결정하게 된다. TAP 제어회로의 상태천이과정을 그림 2에서 나타내었다[6].

TAP 제어회로의 상태는 크게 data register(DR) 테스트 상태와 instruction register(IR) 테스트 상태로 나뉜다. 테스트가 시작되면 TAP명령어를 IR로 저장하기 위하여 Capture-IR, Shift-IR, Update-IR 상태를 지나게 된다. 본 논문에서는 이 과정을 TAP명령어 디코딩과정이라고 정의하였다. TAP명령어 디코딩과정이 끝나면 Capture-DR, Shift-DR, Update-DR 상태를 거쳐 테스트 결과를 TDO로 출력한다. 이상의 과정은 스캔 실행과정이라고 정의하였다. IEEE 1149.1 표준에서는 총 10개의 TAP명령어를 정의하고 있다. 그 중 이용 빈도가 높은 TAP명령어의 종류와 기능을 표 1에서 정리하였다.

2. 개선방향

본 논문에서는 효율적인 JTAG 기반 테스트와 성능향상을 위하여 두 가지의 개선점을 제시하였다. TMS신호를 타겟 내부에서 생성하는 방식과 TAP명령어 디코딩과정을 TAP 제어회로 이전에 수행하는 방식이다.

TMS신호는 TAP명령어에 따라 생성된다. 기존의 테스트 방식에서 호스트는 TAP명령어 디코딩과정이 모두 마무리된 후에도 지속적으로 TMS신호를 TAP 제어회로로 인가해주어야 한다. 또한 디버거 제작자는 TAP 신호들 중 특히 TMS신호를 생성하기 위해 JTA

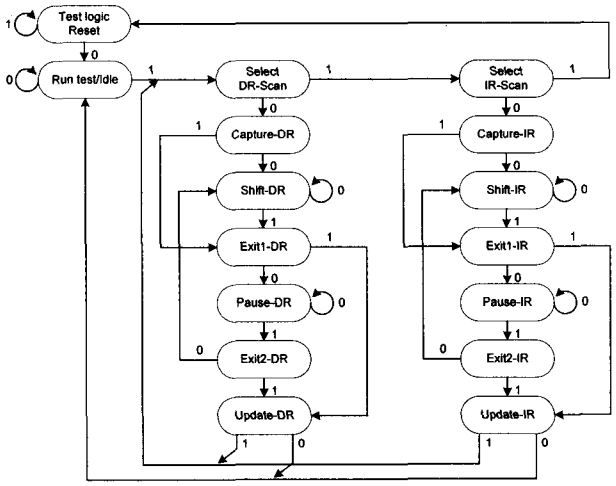


그림 2. 기존의 TAP 제어회로의 상태도
Fig. 2. Conventional state diagram of the TAP controller.

표 1. TAP명령어
Table 1. TAP instruction.

IDCODE	0000	타겟의 고유 ID 확인
BYPASS	0001	지정된 타겟의 테스트 생략
EXTEST	0010	외부시스템과의 입출력 상태 확인
INTEST	0011	내부 동작상태 확인

G 인터페이스와 디버그를 위한 타겟의 하드웨어 사양들을 숙지하여야 한다. TAP명령어 디코딩과정이 끝남과 동시에 내부적으로 TMS신호를 발생시킨다면 호스트는 더 이상 TMS신호를 인가하지 않아도 될 것이다. 또한 디버거 개발에 필요한 시간을 단축할 수 있을 것이다.

두 번째 개선 방향으로 TAP명령어 디코딩과정을 TAP 제어회로 이전에 수행하는 방식을 제시하였다. 기존의 TAP명령어 디코딩과정의 시뮬레이션 결과는 그림 3과 같다. BreakPT 신호가 인가되면서 테스트는 시작된다. 이 때 TAP명령어를 IR에 인가하기 위하여 Shift-IR과 Update-IR 상태를 지난다. 스캔 실행과정이 시작되기 전까지 13클럭이 소비되었고 updateIR 신호가 발생하여 IR에 저장되기까지는 10클럭이 소비되었다.

일반적으로 직렬로 입력되는 신호를 4비트 레지스터에 저장하기 위해서는 최소 4클럭이 요구된다. 안정적인 저장을 위한 업데이트 클럭까지 고려한다면 총 5클럭이 요구된다. 하지만 기존의 TAP명령어 디코딩 과정은 일반적인 경우에 비하여 5클럭이 더 소비되었다. 그 원인은 TAP 제어회로가 그림 2의 상태를 따라 Run

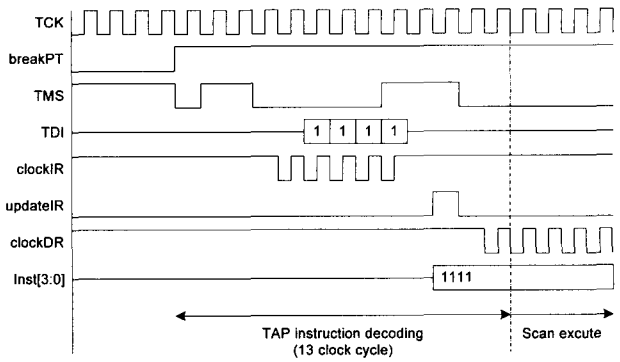


그림 3. 기존의 TAP명령어 디코딩과정 시뮬레이션 결과
Fig. 3. Simulation results of a conventional TAP instruc -tion decoding process.

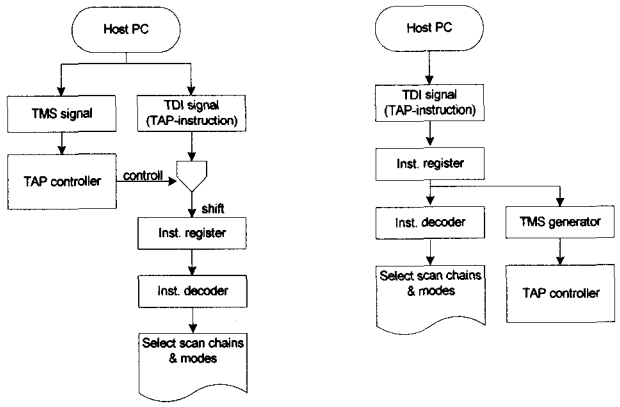


그림 4. TAP명령어 디코딩 순서:
(a)기존의 순서; (b)개선된 순서
Fig. 4. Outline of the TAP instruction decoding process:
(a)conventional outline; (b)advanced outline.

Test/Idle, Select-DR, Select-IR, Capture-IR, Shift -IR로 천이되면서 불필요하게 클럭을 소모하였기 때문이다.

TAP명령어 디코딩과정의 순서를 그림 4에서 나타내었다. 그림 4a와 같이 기존의 TAP명령어 디코딩과정은 TAP 제어회로로부터 제어되어 IR로 저장된다. 하지만 제시된 개선방향은 그림 4b와 같이 TDI 입력이 TAP 제어회로를 거치지 않고 IR로 저장되므로 TAP 제어회로에서의 불필요한 클럭 소모를 피할 수 있다.

III. PIDM 설계

앞서 테스트 성능 개선 방향으로 TMS신호를 타겟 내에서 발생시키는 방식과 TAP명령어 디코딩과정에서 TAP 제어회로를 거치지 않는 방식을 제시하였다. 본 논문에서는 위의 개선될 사항을 모두 만족시키는 PIDM을 제안하였다. 기존의 테스트 타겟의 내부구조를 그림 5에서 나타내었으며 본 논문에서는 이러한 기존

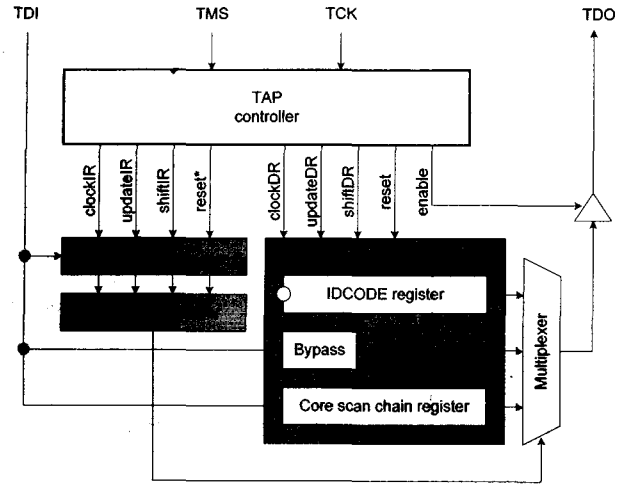


그림 5. 기존의 타겟 구조(non-PIDM 구조)
Fig. 5. Conventional device under test(non-PIDM archi -tecture).

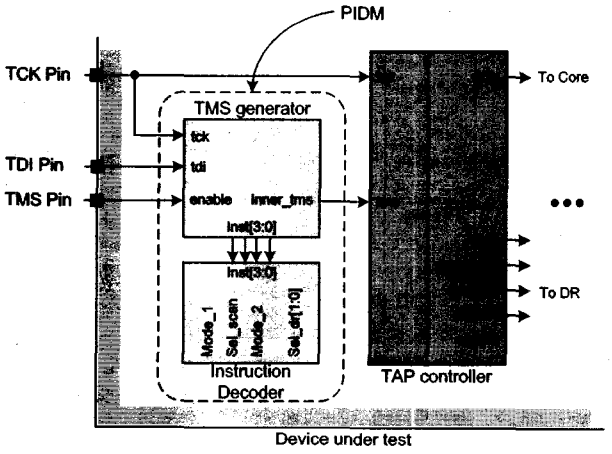


그림 6. PIDM 구조
Fig. 6. PIDM architecture.

의 구조를 non-PIDM 이라고 정의하였다. TAP 제어회로의 clockIR, updateIR, shiftIR 제어 신호는 IR의 스캔 셀로 입력되어 TDI 입력을 IR로 저장시킨다. 명령어 디코더는 최종적으로 테스트하고자하는 DR을 선택한다. 그 후 TAP 제어회로는 Shift-DR 상태로 천이되고 테스트 데이터를 TDO로 시프트하게 된다.

제안된 PIDM의 구조는 그림 6과 같다. PIDM은 TMS신호발생기와 명령어 디코더로 구성된다. 명령어 디코더의 기능은 non-PIDM 구조와 변함이 없다. PIDM 구조에서 기존의 TMS신호는 TAP 제어회로로 인가되지 않고 TMS신호발생기의 enable 입력이 된다. TAP 제어회로는 기존의 외부 TMS신호 대신 PIDM이 내부적으로 생성한 inner_TMS신호를 입력받는다. 그리고 TAP 제어회로의 출력신호 중 IR을 제어하기 위한 신호는 모두 제거되었다.

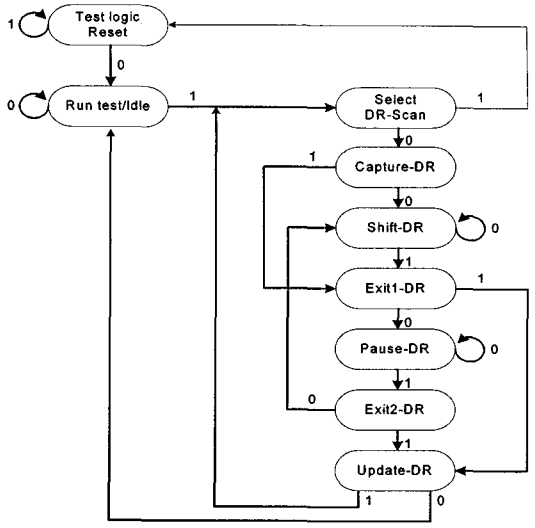


그림 7. TAP 제어회로 상태도(PIDM)
Fig. 7. TAP controller state diagram(PIDM).

PIDM 구조에서는 TAP 제어회로의 상태천이 과정이 그림 7과 같이 축소되었다. 상태의 축소로 인하여 TAP 제어회로의 reset 과 dclk의 출력에 변화가 발생하였다. 이들 신호는 TAP 제어회로의 내부 상태가 각각 Test logic reset, Run test/Idle 상태일 때 high를 출력한다. Reset은 non-PIDM 구조에서 TMS신호가 연속하여 high를 유지해야 하는 시간이 5클럭이었으나 제안된 PIDM 구조에서는 4클럭으로 감소하였다. 이는 테스트 시스템 초기화 시 TMS 신호를 연속하여 4클럭 동안만을 인가함으로서 초기화를 위한 준비시간을 단축시키는 결과를 낳았다.

INTEST는 타겟으로 테스트 벡터를 인가하고 파이프라인을 동작하게 한 뒤 그 출력을 스캔하는 명령이다. 이 때, 파이프라인을 동작시키는 클럭은 테스트 모드 진입 이전의 클럭이 아닌 Dclk을 사용한다[3]. PIDM 구조에서는 Run/state-Idle 상태가 반복되는 최단주기가 3클럭이며 이는 non-PIDM 구조보다 1클럭 단축된 결과이다. 따라서 TAP제어회로의 상태가 Run/state-Idle 일 때마다 출력되는 dclk의 트리거 반복 주기도 1클럭 감소되었다. 이는 INTEST 명령의 파이프라인 동작에서 시간 단축을 가져왔다. 예를 들어, 32비트 프로세서의 경우 파이프라인이 10단으로 구성되어 있다면 PIDM 구조는 총 10클럭의 테스트 클럭을 단축시킬 수 있다. 부가적으로 상태천이 과정이 간략화 되어짐에 따라 TAP 제어회로의 회로 크기가 non-PIDM 구조보다 48% 이상 감소하였다.

TMS신호발생기의 구조를 그림 8에서 나타내었다. TMS신호발생기는 크게 IR과 inner_TMS신호발생기로

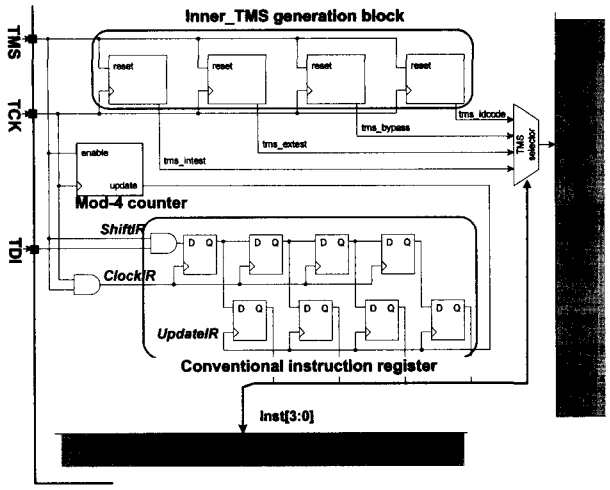


그림 8. TMS신호발생기 구조
Fig. 8. Architecture of the TMS signal generator.

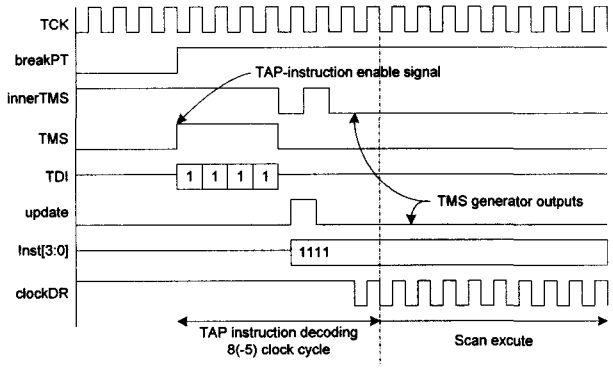


그림 9. PIDM 시뮬레이션 결과
Fig. 9. Simulation results of the PIDM.

구성된다. IR은 non-PIDM의 구조와 같다. 기존의 TAP 제어회로에서 발생되었던 clockIR, shiftIR, updateIR 신호들은 TMS신호발생기 내부적으로 생성하였다. TMS 신호발생기에서 *ClockIR* 제어입력은 TMS와 TCK의 AND연산결과로 *ShiftIR* 제어입력은 외부 TMS 입력신호로 대체하였다. 그리고 *UpdateIR* 제어입력의 발생을 위하여 별도의 4진 카운터를 설치하였다. 4진 카운터는 4비트의 TAP명령어가 모두 시프트 되는 순간 high를 발생시킨다. Inner_TMS신호는 외부에서 입력되는 TMS신호가 high에서 low가 되는 순간 출력되기 시작한다. 발생된 tms_idcode, tms_bypass, tms_extest, tms_instest 신호들은 inst[3:0]의 제어입력에 의해 TMS selector에서 최종적으로 선택되어진다. 선택되어진 inner_TMS신호는 TAP 제어회로로 입력된다.

PIDM을 이용한 TAP명령어 디코딩과정의 시뮬레이션결과를 그림 9에서 보였다. PIDM 구조에서 TAP명령어 디코딩과정은 non-PIDM 구조에 비하여 5클럭이 감소한 8클럭만이 소비되었다.

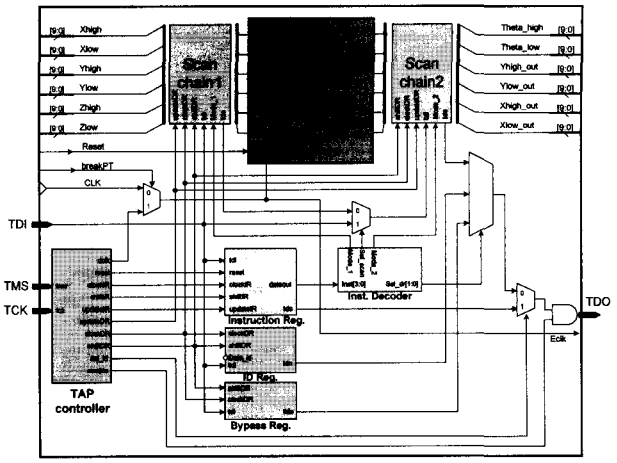


그림 10. CORDIC 프로세서의 테스트 구조(non-PIDM)
Fig. 10. Test architecture of the CORDIC processor (non-PIDM).

Ⅳ. 시뮬레이션 결과 및 성능분석

본 절에서는 제안된 PIDM을 실제로 CORDIC 프로세서에 적용하여 테스트한 결과를 설명한다. 그리고 PIDM 모듈의 성능분석을 다루었다.

1. PIDM을 이용한 테스트 결과 시뮬레이션

사용된 타겟은 CORDIC 프로세서이다[7]. CORDIC 프로세서는 60비트의 데이터 입력과 60비트의 데이터 출력 그리고 clock 과 reset 으로 구성되어 있고 총 9단계의 파이프라인으로 동작된다. Non-PIDM 구조와의 비교를 위해 그림 10과 같이 비교 회로를 구성하였다. 그림 10의 좌측 중앙에 보이는 breakPT 신호는 프로세서의 상태를 시스템 모드에서 테스트 모드로 전환시키는 역할을 한다. BreakPT 신호가 high가 되면 CORDIC 프로세서는 외부 클럭이 아닌 TAP 제어회로에서 생성하는 dclk으로 전환된다. 따라서 CORDIC 프로세서의 파이프라인은 TAP 제어회로에 의해 제어된다. Dclk은 TAP 제어회로의 내부 상태가 Test Run/Idle 상태일 때만 high가 된다.

CORDIC 프로세서의 입력과 출력 단에 스캔 체인을 설치하였다. 스캔 체인은 IEEE 1149.1 표준에 따라 normal 모드, capture 모드, shift 모드, update 모드를 지원하도록 설계되었다. 스캔 체인1은 총 60개의 스캔셀로 구성되었다. Capture-DR 상태가 되면 입력 핀으로부터 인가되는 신호들이 동시에 스캔 셀의 시프트 레지스터에 저장된다. 그리고 Shift-DR과 Update-DR 상태를 지나 프로세서로 테스트 데이터가 인가된다. 스캔 체인2

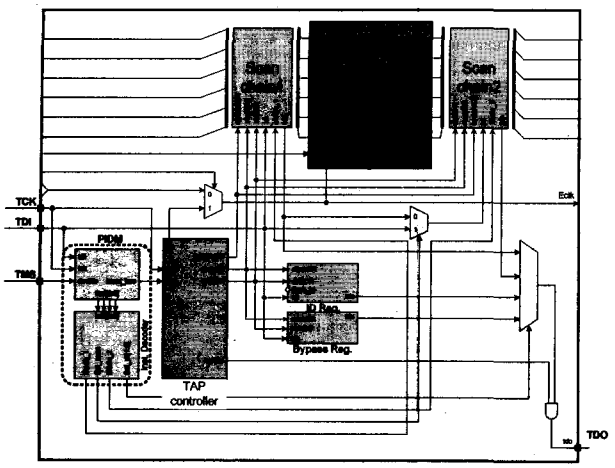


그림 11. CORDIC 프로세서의 PIDM을 이용한 테스트 구조

Fig. 11. Test architecture of the CORDIC processor using PIDM.

는 스캔 체인1과 같은 구조를 갖는다. 다만 프로세서의 출력이 Capture-DR 상태에서 시프트 레지스터로 인가되고 Shift-DR 상태에서 TDO를 통해 출력된다. 명령어 디코더는 표 1에서 제시한 명령어를 지원한다. TAP 제어회로는 IR과 DR의 스캔 셀들을 제어하는 신호를 발생시키며 그림 2의 상태천이 과정을 따른다.

PIDM을 이용한 CORDIC 프로세서의 테스트 구조를 그림 11에서 제시하였다. 그림 10과 비교할 때 TAP명령어 디코딩을 위한 블록들이 모두 TAP 제어회로 이전으로 이동하였음을 확인할 수 있다. 그리고 그림 7의 상태만을 고려하였으므로 TAP 제어회로의 출력신호들은 스캔 실행과정을 위한 제어신호만을 발생시킨다. 그 밖의 CORDIC 프로세서 주변의 스캔 체인1과 스캔 체인2의 구조와 동작은 변함이 없다.

CORDIC 프로세서의 non-PIDM과 PIDM 구조의 테스트 시뮬레이션 결과는 그림 12와 같다. 사용된 TAP 명령어는 INTEST(0011)이다. INTEST는 디버거의 "명령어 한줄 실행"과 같이 코어의 내부 동작의 검증에 사용되는 TAP명령어이다. CORDIC 프로세서에서의 INTEST의 동작은 TDI 핀을 통하여 테스트 입력을 주고, 프로세서에서 파이프라인을 모두 실행한 뒤 최종 출력을 TDO를 통하여 확인하는 3단계의 동작을 거치게 된다. 즉, INTEST TAP명령어는 디코딩 단계를 포함하여 테스트입력 인가단계, 파이프라인 실행단계, 테스트 출력 단계의 4단계의 절차로 구성된다.

TAP명령어 디코딩 단계는 테스트입력이 인가되기 전까지의 시간이다. CORDIC 프로세서의 테스트 입력은 총 60비트이고 TDI를 통해 인가되기 위해서는 60회의

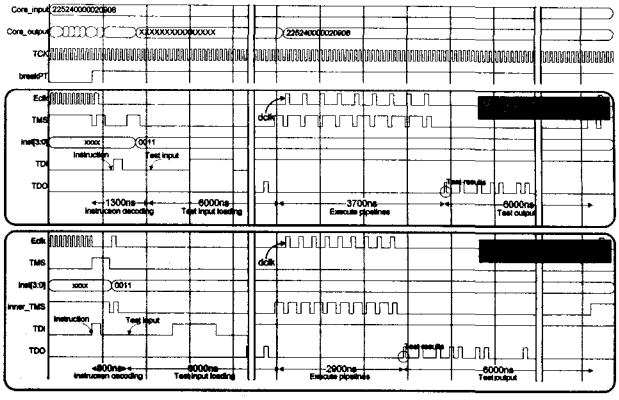


그림 12. CORDIC 프로세서의 시뮬레이션 결과

Fig. 12. Simulation results of the CORDIC processor.

시프트 동작이 요구된다. 테스트 출력 역시 TDO를 통해 출력되기 위하여 60회의 시프트 동작을 수행하여야 한다. 따라서 테스트입력 인가단계와 테스트출력 단계는 non-PIDM과 PIDM 모두 각각 60회의 클럭을 소비한다. 위 시뮬레이션에서 사용된 TCK의 클럭 주파수가 10MHz이므로 시프트 동작 시간은 6000ns가 된다.

PIDM 구조의 경우 TAP명령어 디코딩 단계와 파이프라인 실행단계에서 non-PIDM 구조와 비교하여 차이를 나타냈다. TAP명령어 디코딩을 수행하는 동안 non-PIDM 구조는 1300ns, PIDM 구조는 500ns 감소한 800ns의 시간이 소비되었다. 파이프라인 수행시간은 non-PIDM과 PIDM의 경우 각각 3700ns와 2900ns를 나타냈다. 성능향상의 원인은 PIDM 구조에서 CORDIC 프로세서로 인가되는 dclk의 속도가 증가되었기 때문이다. Dclk는 TAP제어회로에서 Run Test/Idle 상태마다 트리거 된다. 이 때, PIDM은 그림 7에서 제시한 상태도와 같이 기존의 그림 2에서 Select-IR Scan 상태가 제거되었으므로 클럭 소비를 1회 단축하게 된다.

2. 성능분석

본 논문에서는 테스트 성능의 기준을 응답시간(response time)으로 정의하였다. 단위는 클럭회수(clock cycles)이다. 응답시간은 TAP명령어가 타겟으로 인가되는 시점부터 TDO를 통하여 테스트 결과가 모두 시프트되는 시점까지로 정의한다. 일반적으로 테스트 목적을 고려할 때 INTEST 명령은 가장 사용 빈도가 높은 명령에 속한다. INTEST 명령의 경우 CORDIC 프로세서의 테스트 시뮬레이션 결과를 표 2에서 정리하였다. PIDM 구조는 non-PIDM 구조보다 7.65%의 성능향상을 나타내었다.

표 2의 성능분석 과정과 같이 표 1의 4개의 명령을

표 2. INTEST 명령 시뮬레이션 결과
Table 2. Simulation results of the INTEST instruction.

(단위 : clock cycles)

Instruction decoding	13	8
Test data input	60	60
Pipeline execution	37	29
Test data output	60	60
Total	170	157

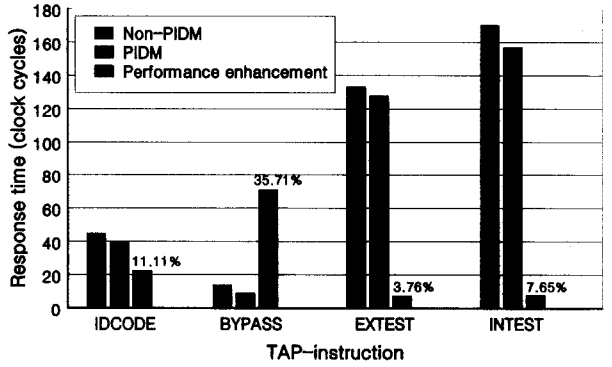


그림 13. CORDIC 프로세서의 테스트 성능
Fig. 13. Test performance of the CORDIC processor.

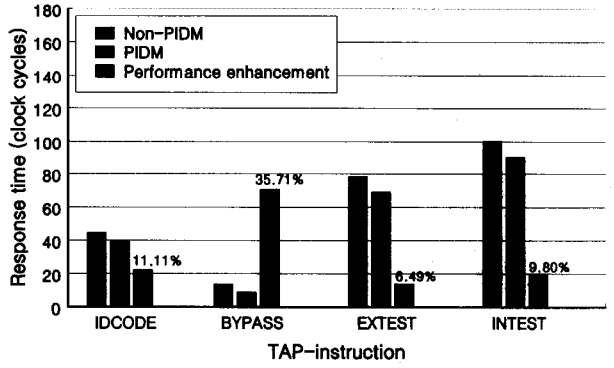


그림 14. 32비트 프로세서의 테스트 성능
Fig. 14. Test performance of a 32bit processor.

시뮬레이션 한 결과 그림 13과 같은 그래프를 얻을 수 있었다. 모든 TAP명령어를 고려할 때 평균 14.5%의 성능향상을 보였다. 시프트 되는 스캔 셀의 수가 1개인 BYPASS 명령은 non-PIDM 구조와 비교하여 35.71% 성능향상을 보였다. 시프트 되는 스캔 셀의 수가 120개인 EXTEST 명령의 경우 3.76%의 성능향상을 보였다.

EXTEST 명령과 시프트 되는 스캔 셀의 수가 같은 INTEST 명령은 7.65%를 나타내었다. 즉, PIDM 구조에서는 시프트 되는 스캔 셀의 개수가 적을수록 그리고 INTEST 명령과 같이 테스트 도중에 파이프라인 수행과정이 있을 경우에 보다 높은 성능향상을 보였다.

본 논문에서는 기본적인 산술·논리명령을 갖춘 32비트 프로세서의 입·출력에 각각 32개의 스캔 셀을 설치한 뒤 제안된 PIDM 구조를 적용하였다. 32비트 프로세서는 5단계 파이프라인 구조를 갖고 32비트의 인스트럭션 입력과 32비트의 출력을 갖는다. 32비트 프로세서의 테스트 회로는 그림 10과 그림 11의 CORDIC 프로세서의 구조와 같다. 다만 스캔 체인1 과 스캔 체인2의 셀의 수가 각각 32개로 감소되었다. 시뮬레이션 결과 그림 14와 같이 테스트 성능의 증가를 나타내었다. EXTEST 명령과 INTEST 명령에서 CORDIC 프로세서보다 약간 증가한 성능을 나타내었다. 모든 TAP명령어를 고려할 때 PIDM 구조는 non-PIDM 구조보다 평균 15.7%의 성능향상을 보였다.

V. 결 론

제안된 PIDM은 기존의 TAP명령어 디코딩과정에서 발생되었던 불필요한 클럭소모를 최소화함으로서 테스트 성능향상은 물론 TAP 제어회로의 하드웨어크기를 48% 이상 축소할 수 있었다. PIDM 구조를 CORDIC 프로세서에 적용하였을 경우 14.5%의 테스트 성능증가를 나타내었다. 분석 결과, PIDM 구조를 이용할 경우 테스트 성능은 시프트 되어지는 스캔 셀의 개수가 적을수록 그리고 타겟의 파이프라인 수가 많을수록 non-PIDM 구조에 비해 높은 성능향상을 보였다. 특히 CORDIC 프로세서를 32비트 프로세서로 대체하였을 경우 15.7%로 1.2%정도의 성능향상이 이루어졌다. 위 결과를 바탕으로 현재 상용되는 32비트 마이크로프로세서에서도 마찬가지로 테스트성능의 증가를 기대할 수 있다. 그리고 기존의 호스트나 프로토콜 변환기에서 생성하였던 TMS 신호를 타겟 내부에서 생성함으로써 사용자가 디버거 제작에 소요하는 시간을 크게 단축시킬 전망이다.

참 고 문 헌

[1] K. P. Parker, The Boundary Scan Handbook, Kluwer Academic Publishers, 2003.

[2] S. Furber, ARM System-on-Chip Architecture 2nd Ed., Addison Wesley, 2000.

[3] Advanced RISC Machines, ARM7TDMI Data Sheet, Document Number: ARMDDI0029E, http://www.arm.com.

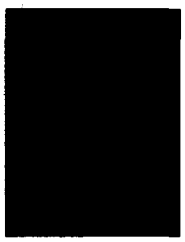
[4] I. Huang, C. Kao and H. Chen, "A Retargetable Embedded In Circuit Emulation Module for Microprocessors," IEEE Design & Test of Computers, vol. 19, pp. 28-38, Aug. 2002.

[5] C. MacNamee, D. Heffernan, "Emerging On-Chip Debugging Techniques for Real-Time Embedded Systems," IEE Computing & Control Eng., vol. 11, no. 6, pp. 295-303, Dec. 2000.

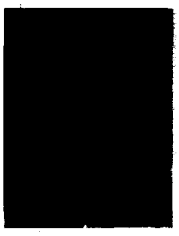
[6] IEEE Std 1149.1-2001, Test Port and Boundary-Scan Architecture, IEEE, 2001.

[7] 김승열, 김용대, 한선경, 유영갑, "Redundant Signed Binary Number에 의한 CORDIC 회로," 대한전자공학회논문지, 40권, CI편, 6호, 317-324쪽, 2003년 11월

저 자 소 개

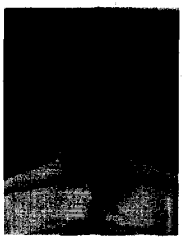


윤 연 상(학생회원)
2004년 충북대학교 전기전자 공학부 학사 졸업.
2004년 현재 충북대학교 정보통신 공학과 석사과정 재학 중
<주관심분야: 디지털 회로 설계 및 테스트, 임베디드 시스템, 암호 시스템>

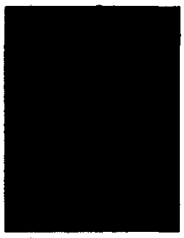


권 순 열(학생회원)
2003년 충북대학교 전기전자 공학부 학사 졸업.
2004년 현재 충북대학교 정보통신 공학과 석사과정 재학 중
<주관심분야: Computer arithmetic, ASIC 설계, Communication system>

김 용 대(정회원)
1990년 충북대학교 정보통신 공학과 학사 졸업.
1993년 충북대학교 컴퓨터공학과 석사 졸업
1989~1998년 신흥기술연구소 팀장
2000년~2004년 현재 충북대학교 정보통신공학과 박사과정
<주관심분야: Computer arithmetic, ASIC 설계, 암호 시스템>



김 승 열(학생회원)
2002년 충북대학교 전기전자 공학부 학사 졸업.
2004년 현재 충북대학교 정보통신 공학과 석사과정 재학 중
<주관심분야: Computer arithmetic, 고속인쇄회로 설계>



박 진 섭(학생회원)
2004년 충북대학교 전기전자 공학부 학사 졸업.
2004년 현재 충북대학교 정보통신 공학과 석사과정 재학 중
<주관심분야: 디지털 회로 설계, 암호 시스템>

유 영 갑(정회원)
1975년 서강대학교 전자공학과 학사 졸업
1975년~1979년 국방과학연구소 연구원
1981년 Univ.of Michigan, Ann Arbor 전기전산학과 석사 졸업
1986년 Univ.of Michigan, Ann Arbor 전기전산학과 공학박사 졸업
1986~1988년 금성반도체 (주) 책임 연구원
1993~1994년 아리조나 대학교 객원교수
1998~2000년 오레곤 주립대학교 교환교수
1988~2004년 현재 충북대학교 정보통신공학과 교수
<주관심분야: VLSI 설계 및 테스트, 고속 인쇄회로 설계, 암호학>