디지털 회로 디버깅을 위한 IEEE 1149 JTAG 및 Enhanced JTAG 기술

박 희 동*

1. 서 론

하드웨어나 소프트웨어 개발 단계에서 IC 칩이나 보드, 그리고 시스템에 이르기까지 개발된 시제품의 기능 동작을 확인하고 잘못된 경우 이를 수정, 보완하는 디버깅은 현재 필수적인 단계로 인식되고 있다.

보통 디지털 시스템의 개발 시, 70년대 중반에는 직접 PCBs(Print Circuit Boards)에 접촉하는 방식으로 보드를 테스트하였다. 이런 방법은 현재에도 지그라는 테스트 장비를 사용하면서 계속되고 있다. 그런데 보드에 탑재된 부품들 단자 사이의 간격이 점차 좁아짐에 따라 테스트가 점점 어렵게 되었고 다층 기판 보드를 많이 사용함에 따라 위의 방법은 거의 불가능하게 되었다. 개발된 보드의 테스트는 하드웨어에 손상을 주면 안되기 때문에 점차 여러 가지 문제점들이 발생하게 되었다. 따라서 비접촉 방식의 테스트 기술이 절실하게 되었고, 80년대 중반에 JETAG(Joint European Test Access Group)란 단체가 결성되었으며 그들은 논리적으로 각 부품의 가장자리에 존재하는 직렬 쉬프트 레지스터라는 개념으로 발전시키게 되었다.[5]

1990년대에 들어 IEEE는 IC 칩과 보드를구체적으로 테스트할 수 있는 방법을 채택하였는데, 이것이 IEEE 1149.1-1990이란 것으로 부품들과 소켓사이에서의 기존의 bed-of-nails, logic probes, test module 등과 같은 기술을 한층 발전시킨 디버깅 기술이었다.

이 JTAG(Joint Test Action Group) 디버깅 기술을 통해 마이크로프로세서와 ASIC을 이용한 기술 개발자들은 각 부품의 동작에서부터 보드의 테스트 지점에 대한 각종 정보를 보다 쉽고 구체적으로 제공 및 수집할 수 있게 되었다. IEEE 1149 표준은 "Standard Test Access Port and Boundary-Scan Architecture"로[1] 디지털 회로의 핀들에 대한 신호 레벨을 액세스 또는 제어할 수 있는 5개의 핀을 통한 직렬 프로토콜을 제시하고 있다.

일반적으로 JTAG 와 Boundary-Scan이란 말을 함께 많이 사용하는데 JTAG은 칩 내부에 Boundary Cell 이란 것을 두어 외부의 핀과 1:1로 연결시켜 프로세서가 할 수 있는 동작을 중간에 셀을 통해 모든 동작을 인위적으로 수행할 수 있어 여러 가지 하드웨어 테스트나 연결 상태 등을 체크할 수 있기 때문이다.

보드나 디바이스가 점차 소형화되고 내부적으로 복잡해짐에 따라 추가적인 디버깅 기술이 필요하게 되었다. 지금까지 이 JTAG에다 기능들을 추가한 여러 가지 확장된 디버깅 규격이 나왔는

* 중부대학교 정보통신대학 정보통신S/W공학과 부교수

데, 대표적인 것이 IBM/Motorola PowerPC 프로세서들을 위한 COP(Common On-chip Processor) 기술, NEC 의 N-wire/N-Trace, IBM 의 RISCWatch, Motorola IC 들에 사용되는 BDM (Background Debug Mode)등이 있다. 이들은 기존의 JTAG을 확장한 EJTAG(Enhanced JTAG)으로 생각할 수 있으며 JTAG 핀들의 디버깅 기능을 덧씌우고 수행시의 추적이나 제어를 위해 핀을 추가한 기술들이다.

본 논문에서는 JTAG 기술에 대해 전반적으로 살펴보고, 이를 기반으로 한 EJTAG 기술 중 하나인 COP 기술, 그리고 디버거 도구인 VisionICE와 PowerTap 에 대해 간단히 기술하기로 한다.

2. JTAG 기술

현재는 고성능 마이크로프로세서나 복잡한 ASIC과 같은 VLSI 디바이스가 많이 사용됨에 따라 이들의 테스트 기능은 매우 중요하게 되었다. 즉, 이들 IC들을 보드에 장착한 채 그 기능과 연결성을 테스트해야 하는 요구 때문에 비접촉형 테스트인 JTAG이 표준화 되었으며 이의 필요성이 중요하게 되었다. 이는 BIST(Built-In Self-Test)라는 형태의 개념을 적용하게 되었으며 디바이스의 기능은 물론 납땜이 잘 되었는지와 부품 장착 경로가 정확한지도 알 수 있게 되었다.

Non-JTAG 디바이스들과 비교해서 JTAG 지원 디바이스들은 비용이 올라가고 성능에서도 영향을 주게 된다. 즉, 추가적인 핀들과 회로, 설계 기간 등이 증가되고 얼마정도의 성능 저하가 일어난다. 디바이스의 복잡도와 기능성에 따라 JTAG을 추가하면 사이즈는 약 10% 정도 증가하고 그 성능(속도)은 대략 10% 정도 감소하는 걸로 알려져 있다. 상용 시험을 위한 보드 레벨에서의 JTAG을 사용하면 자동 테스트 패턴 생성(ATPG : Automatic Test Pattern Generation) 소프트웨어와 검사 패키지를 위한 추가적인 비용이 소요된다. 이러한 비용은 IC의 가격뿐 아니라 생산 시스템의 비용도 증가하게 만든다.

하지만 이러한 비용 증가와 성능 감소는 이를 상쇄할 수 있는 장점이 있기에 큰 문제가 되지는 않는다. 즉, 쉽고 자동화된 테스트에 따른 설계와 생산 기간의 단축으로 인해 빠른 시간 안에 시장에 출시할 수 있게 되었고(Time To Market), 기존의 매우 고가인 ICT(In Circuit Tester) 장비를 저렴한 PC 기반 테스터로 대체할 수 있었기 때문이다.[3]

JTAG이 제공하는 기능을 먼저 살펴보면 프로세서(CPU)의 상태와는 상관없이 디바이스의 모든 외부 핀을 구동시키거나 값을 읽어 들일 수 있는 기능을 제공한다.

- 디바이스 내에서 모든 외부와의 연결점을 가로챈다. (즉 외부로 나가는 각각의 핀들과 일대 일로 연결)
- 각각의 셀은 직렬 쉬프트 레지스터(Boundary Scan 레지스터)를 형성하기 위해서 서로 연결되어 있다.
- 전체적인 인터페이스는 5개의 핀에 의해서 제어된다. (TDI, TMS, TCK, $\overline{\text{TRST}}$, TDO)
- 회로의 배선과 소자의 전기적 연결상태 테스트
- 디바이스간의 연결상태 테스트
- 플래시 메모리 프로그래밍

JTAG 에는 디바이스마다 하나의 스캔체인(Scan chain)으로 구성된 래치들로 내부에 존재하며, 스캔체인의 길이는 디바이스마다 다를 수 있다. JTAG 은 5개의 신호선을 통해 그림 1과 같이 연결되는 형태로 나타난다.

일반적인 JTAG 을 위한 TAP 구현은 다음과 같은 기능을 지원한다.

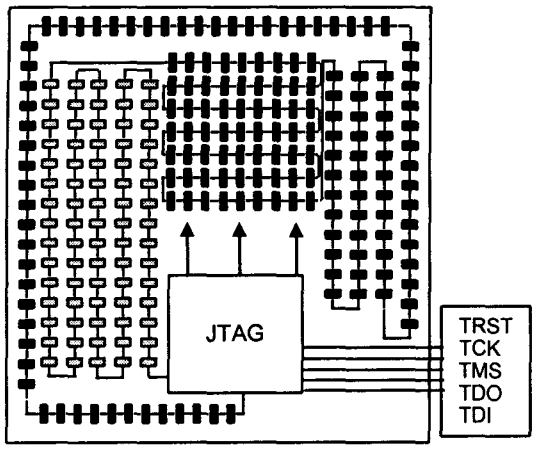

그림 1. JTAG 신호선과 디바이스 접속

- 보드의 전기적 연결성을 확인하기 위한 Boundary scan 동작 기능
- 마이크로프로세서의 Boundary scan 레지스터를 단일 셀로 바이패스 기능.
- Boundary scan 레지스터의 결과를 동작중에 읽어낼 수 있는 기능
- 테스트 중 외부 구동 신호를 disable 하는 기능

JTAG에서 TAP(Test Access Port)는 외부 JTAG 하드웨어와 연결되어 있어 그 상태를 액세스할 수 있도록 되어 있다. 이 포트는 설계자나 테스터(Tester)들이 생산 조립 라인이나 소비자에게 가기 전에 미리 칩에 그 구성(Configuration)을 만들 수 있도록 해 준다. 또한 플래시 메모리 접속을 가진 것이라면 이런 방식으로 프로그램이 가능하다. 이러한 양방향 특성을 가진 디버깅/프로그래밍 기능은 각 JTAG 가능 디바이스들이 어떻게 동작할 것인가를 기술해주는 언어인 BSDL (Boundary-Scan Definition Language)을 사용하여 규정하게 된다.

보통 개발 보드의 테스트에서, 시스템 레벨에서 수행되지 않은 경우에는 제일 먼저 스캔 경로 검증(Scan path verification)이 이루어진다. 이 테스트는 스캔 경로에 있는 디바이스들이 올바르게 통신할 수 있는지와 "Stuck bit"가 없는지를 검증하는 것이다. 이것은 경로를 통해 데이터 또는 명령어 스캔을 사용해 일련의 데이터를 보내면서 수행하는 것으로, 신호의 충돌이나 손상된 디바이스의 사용을 방지하지하기 위해 모든 스캔 가능한 디바이스에 대해 올바른 통신은 필수적인 사항이다.

다음으로 JTAG 디바이스의 내부 동작을 확인하기 위한 컴포넌트 테스트(Component testing)가 이루어진다. 이는 복잡한 ASIC 또는 CPU들에게 Boundary Scan이 적합하지 않은지를 확인하기 위한 것으로, 내부 동작을 시뮬레이션 해주고 그 결과를 확인하기 위한 테스트 파일이 사용된다. 이는 보드에 불량 부품이 장착되는 것을 막아주고 조립과정에서 손상을 입지 않도록 도와주는 역할을 한다.

상호연결 테스트(Interconnection testing)는 신호들의 연결이 적절히 되어 있는가를 테스트하는 것으로, 모든 JTAG 디바이스 사이에서 수행될 수 있다. 이러한 신호망들은 개방과 단락이 검사되고, 나중에 설명하는 EXTEST 명령을 통해 하나의 디바이스 외부로 데이터 패턴을 주거나, SAMPLE 명령으로 목적 디바이스의 패턴을 읽을 수 있다.[2,8]

2.1 TAP 구현

TAP 인터페이스는 JTAG에서 보아온 바와 같이 다음의 신호들이 있다.[4]

- Test Clock (TCK) - 테스트 로직을 동기화하기 위한 시험 클럭 입력.
- Test Mode Select (TMS) - 시험 모드 선택 입력(내부적으로 pull-up됨.)으로 TAP의 상태를 순서화하기 위해 TCK의 상승 에지에서

샘플링 된다.

- Test Data In (TDI) - 시험 명령과 데이터를 위한 직렬 입력으로 TCK의 상승 에지에서 샘플링된다.
- Test Data Out (TDO) - 시험 명령과 데이터를 위한 3-state 직렬 출력으로 TCK의 하강 에지에서 명령 레지스터(IR) 또는 데이터 레지스터(DR)의 내용을 이동시킨다.
- Test Reset (TRST) - TAP 제어기의 초기화를 위한 비동기 리셋 기능을 한다.

예를 들어 그림 2는 PowerPC 8xx 계열의 TAP 로직을 보여준다.[4]

이는 일반적인 JTAG 구조와 같은 기능을 가지고 있다. PowerPC MPC8xx 계열 마이크로프로세서에서 TAP 구현은 그림과 같이 TAP 제어부, 4비트 명령 레지스터 그리고 2개의 테스트 레지스터(1비트 비이패스 레지스터와 475비트 boundary scan 레지스터)를 포함하고 있다.

JTAG의 TAP 제어부의 상태 다이어그램은 일반적으로 그림 3과 같이 익히 보아오던 형태로 나타낼 수 있다. 내부적으로 TAP 제어부는 명령을 해석하고 모든 JTAG 기능에 접근하는데 사용된다. 즉, JTAG 명령들을 명령 레지스터에 로드하고, 데이터를 프로세서에서 가능한 스캔체인

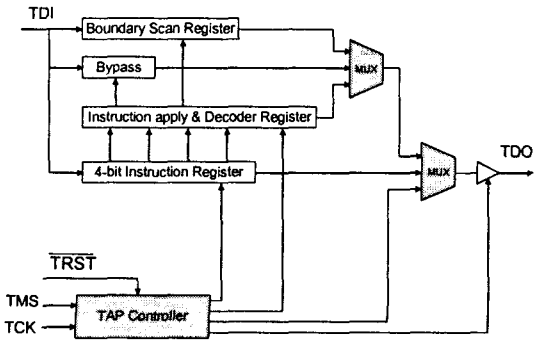

그림 2. PowerPC 계열 TAP 로직

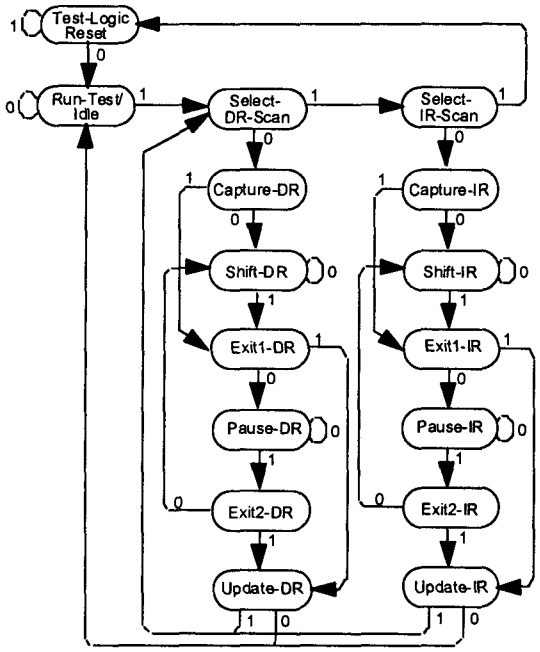

그림 3. JTAG에서의 TAP 상태도

(Scan chain)으로 주거나 받는 기능, 그리고 테스트를 수행하고 JTAG 인터페이스를 리셋하는 기능을 한다.

TMS 입력은 TAP 제어부의 상태 천이에 영향을 주는데, 상태 천이는 TCK의 상승 에지시의 TMS 값에 의해 결정된다.

TMS는 0일 때 idle 상태로 되고, 레지스터나 스캔체인으로부터 데이터를 읽거나 거기에 쓰기 준비가 될 때까지 1의 상태를 가진다. 예를 들어, Run-Test/Idle 상태로부터 시작하여, TMS는 인터페이스를 통해 1-1-0-0의 비트값을 보내면 TAP 제어부가 Shift-IR 상태로 가게 된다. 이때의 명령 로드시의 신호 파형은 그림 4와 같이 나타낼 수 있다.

JTAG에서의 명령 레지스터에 들어가는 명령에는 공개(public) 명령과 사설(private) 명령을 지원하는데, 공개명령에는 BYPASS, SAMPLE, PRELOAD, EXTEST가 있고, 사설명령은 디바

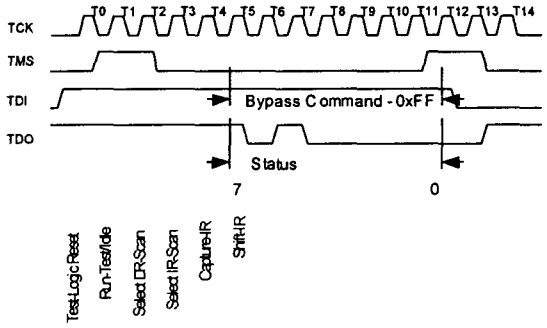

그림 4. TMS의 명령 로드시의 신호파형

이스 생산자가 제공하는 명령이다. 또한 선택적으로 CLAMP, HI-Z, IDCODE, INTEST, USERCODE, RUNBIST 등이 있다. 이들 명령중 중요한 것에 대한 코드와 기능은 다음과 같다.

1) EXTEST
- TDI와 TDO를 Boundary Scan 레지스터에 연결해준다.
- 디바이스의 외부회로에 대한 테스트 즉, 디바이스 외부에 상호 연결된 디바이스의 동작을 테스트하는데 사용된다.

2) BYPASS
- TDI와 TDO를 BYPASS 레지스터에 연결해 준다.
- JTAG의 효율을 높이기 위한 것으로 시험에 요구되는 시간을 상당히 줄이는데 사용된다.

3) SAMPLE/PRELOAD
- TDI와 TDO를 Boundary Scan 레지스터에 연결해준다.
- SAMPLE 명령은 시스템의 상태를 실시간으로 감시하기 위한 기능이며, PRELOAD 명령은 소자핀에서 신호들을 제어하기 위해 사용된다.

4) IDCODE
- TDI와 TDO를 ID CODE를 갖는 Identification Register에 연결해준다.
- 32-bit의 디바이스 정보를 얻는데 사용한다.

5) HIGHZ
- TDI와 TDO를 Bypass Register에 연결해 준다.
- HIGHZ 명령은 Instruction Register로 로더되고, 모든 외부 핀들의 출력들을 비활성 구동(고임피던스) 상태로 한다. 이 상태는 프로세서의 손상 없이 회로상의 에뮬레이터가 프로세서 출력이 일반적으로 구동하는 연결에 신호를 구동하도록 해준다. 즉, EXTEST와 같은 기능을 하지만, 다른 점은 디바이스의 JTAG 기능을 이용하여 외부 핀들의 상태를 제어하는 것이 아니라, 테스트 지그와 같은 장비를 통하여 시험하는 방식처럼 외부에서 직접 신호를 주고, 그 응답 특성을 알아보기 위한 명령이다.

2.2 아날로그 혼합 신호에의 적용

최근의 기술은 JTAG 버스를 FPGA(Field Programmable Gate Array)와 플래시(Flash) 메모리에도 적용하게 되었다는 것이다. FPGA와 CPLD (Complex Programmable Logic Device) 제조사들은 이미 JTAG을 이용한 ISP(In-System Programming) 기능을 제공하고 있으며, 또한 이를 위한 하드웨어 및 소프트웨어 도구도 제공되고 있다. JTAG이 테스트를 위한 목적이었지만 프로그램 할 수 있는 디바이스에 적용하는 것은 전혀 비용이 들지 않기 때문에 매우 유리하게 되었다. 또한 시스템 운영 중에 재구성하여야 하는 경우에도 간단히 JTAG을 이용하여 Upgrade 할 수 있다는 것이다. 이러한 JTAG에 대한 장점이 계속 활용됨으로 인해 JTAG 관련 기술은 지속적으로 발전될 것이다.

JTAG의 활용성에 힘입어 디지털 영역에서의 테스트를 떠나 이제는 아날로그 및 혼합된 신호에 대한 테스트 기술이 선보이고 있다. 이는 IEEE 1149.4 로 알려진 것으로(Dot 4라고도 불린다.) 최근에야 완성된 규격이다. 디지털 신호에 대해서는 테스트 패턴을 직관적으로 발생할 수 있었고 그 입력과 결과도 '0' 아니면 '1' 이었다. 하지만 아날로그 신호는 연속적인 신호이기 때문에 IEEE 1149.1 인 JTAG 보다 훨씬 더 복잡하고 활용성이 많다는 것이다. 더구나 이는 수동 소자인 저항이나 캐패시터 들에 대한 측정도 가능하다는 것이며, 각 핀들은 실시간 측정처럼 동작한다는 것이다.[3]

Dot 4를 아날로그 회로에 추가한다는 것은 기존의 JTAG을 디지털 회로에 적용하는 것보다 훨씬 복잡한 설계의 어려움이 존재한다. 이는 기존의 JTAG 핀에다 2개의 제어를 위한 핀이 추가되었으며 2개의 on-chip 테스트 버스를 추가하여 적용 및 측정을 가능하게 하였다. JTAG과 마찬가지로 Dot 4도 대상 회로의 기능이나 동작에 영향을 주지 않고 디바이스에 통합되도록 하여야 하는데, 이것은 고성능 아날로그 회로의 특성 상 미세한 캐패시턴스나 회로 패턴의 변화가 매우 민감하게 영향을 준다는 점에서 대단히 어려운 기술 도전이라 할 수 있다.

그렇다면 디지털에서의 비용과 장점에 대한 공식이 아날로그에는 어떻게 비교될 수 있을까? 기본적으로는 JTAG에서의 경우와 동일할 것이며, 전체 회로 설계 비용이라는 측면에서 보면 많은 핀을 가진 경우나 복잡한 혼합 신호를 가진 경우에는 제한이 줄어든다는 것이다. 이것은 디지털 영역에서와 같이 마이크로프로세서나 복잡한 VLSI 디바이스인 경우에 적용되는 것과 같은 의미로 생각할 수 있다.

Dot 4에 대한 잠재적인 이점은 대부분의 시스템이 디지털 회로이고 아날로그 회로는 적지만 매우 민감한 부분인 경우(일반적으로 90%는 디지털이고 10% 정도만 아날로그인 경우가 대부분이다.) 디지털 회로를 기존의 JTAG을 이용하여 ATPG 소프트웨어를 이용하여 테스트하고 검사를 수행하게 된다. 보통 아날로그 신호를 디지털 신호로 변환하거나 그 반대로 하는 회로가 많은데, 디지털 회로의 테스트가 완료되었다고 하더라도 아날로그 회로에 대한 테스트가 완료되지 않으면 시스템의 테스트가 완료되었다고 할 수 없다. 더구나 아날로그 회로는 잡음, EMI 등 여러 외부 요인에 매우 민감한 특징을 가지고 있기 때문에 실제로는 전자파 차폐 기능을 가진 곳에서 테스트해야 하는 매우 어려운 문제를 안고 있다.

이런 형태의 테스트를 위해서는 2단계로 나누어진 테스트 과정으로 개발 단계의 테스트와 생산 단계의 테스트가 모두 이루어져야 할 것이다. 디지털 부분은 기존의 JTAG 디지털 테스트 기술을 사용하여 테스트하면 되는데, 아날로그 회로는 종래의 ICT 환경에서 외부 테스트 부분들과 사용자 시설 구축, 그리고 ATE에 대한 투자 등이 필요하게 된다.

앞으로 Dot 4 규격에 대한 많은 디바이스의 지원이 되면 많은 부품, 디바이스들이 단일 제조 테스트 플로우로 표준화될 것이다.

3. EJTAG(Enhanced JTAG) 기술

EJTAG은[2] 기존의 JTAG 에다 그 기능을 추가한 형태로 반도체 파트너와 하드웨어와 소프트웨어 개발 도구 개발자들과 공동으로 정한 규격이다. EJTAG은 더욱 많은 시뮬레이션 기능을 제공해주는데, 이는 각 부품마다 고려해야 하는 내용이 많아지고 여러 다른 동작이 발생할 수 있기

때문이다. 따라서 JTAG보다 좀 더 진보된 칩 설계 기술이 필요하고 이는 그 복잡성과 속도 그리고 기능 등에 영향을 받게 된다. EJTAG은 설계자나 개발자에게 하드웨어 및 소프트웨어적인 break-point 설정 기능을 지원해준다. 따라서 이를 이용하면 좀 더 빠르고 향상된 시스템 평가와 디버깅 작업에 도움을 받을 수 있다.

EJTAG의 중요한 부분은 바로 "tracing" 기능이다. 이는 사용자로 하여금 실시간으로 코드를 보게 하고 시간 분석과 사용되지 않는 코드 발견을 통해 성능의 병목 지점을 알 수 있게 해준다. JTAG이 기본적인 디버그 기능을 제공해주지만 EJTAG은 좀 더 구체적인 ICE(In-Circuit-Emulation) 기능을 제공해준다.

EJTAG이 설계되는 경우에는 외부 디버깅 모듈과 통신을 하기 위한 DSU(Decoding Support Unit) 라는 인터페이스가 필요하다. 이를 통해 On-chip 논리는 모든 핀을 통해 동작 테스트를 위한 신호를 보낼 수 있으며, 또한 설계자가 평가할 수 있도록 EJTAG 테스트 장비로 전달되는 on-board 회로로 되돌아오는 실시간 상태 정보를 모든 핀으로부터 읽을 수 있게 된다.

IDT(Integrated Device Technology) 프로세서를 위한 EJTAG은 추가적인 디버깅과 테스트 기능을 위한 규격으로 설계되었다. 처음에 RC32355 고성능 프로세서에 EJTAG 기능이 필요했고, 점차 SoC(System On a Chip) 구조의 발전과 집적 기술의 발전으로 하드웨어 및 소프트웨어적인 시스템 테스트가 그만큼 더 복잡해졌다. 예를 들어, IDT MIPS RISC 프로세서에는 다른 레벨의 EJTAG 기술이 제공되었으며, 그림 5와 같이 RC3000이나 RC40000 코어 프로세서를 기초로 한 시스템 설계에서는 별도로 EJTAG 블록이 추가되었으며 이 EJTAG 포트를 통해 프로세서의

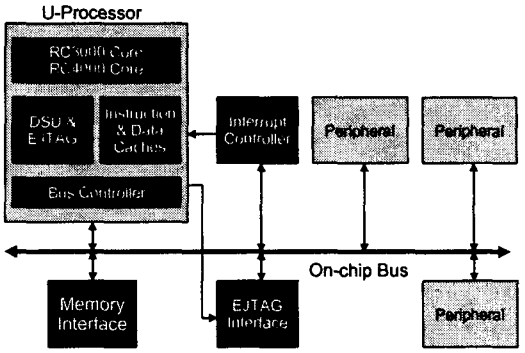

그림 5. RC3000/4000 프로세서의 구조

기능과는 별도로 시스템의 boundary scan이 이루어진다.

만약 EJTAG 블럭이 프로세서 내에 통합되지 않는 경우에는 디버깅을 위해 프로세서를 소켓에서 분리해야 하고 따라서 ICE 장비의 가격이 매우 올라가게 된다. EJTAG 커넥터는 프로세서와 시스템 회로 모두 접속되기 때문에 프로세서를 물리적인 침해없이 모든 시스템 기능과 연결성을 테스트할 수 있게 된다.

그림 6과 같은 RC32355와 같은 프로세서는 CPU 내부에 EJTAG 블록이 포함되어 있고 외부 JTAG 테스트 시스템과 9핀 인터페이스를 이용해 연결하고 있다. 이는 저비용으로 PC 기반 에뮬레이터 시스템을 가능하게 해주며 기존의 고가 ICE를 대체할 수 있도록 해 준다. 이 구조는 시스템 설계자나 시스템 테스터로 하여금 최종 생산까

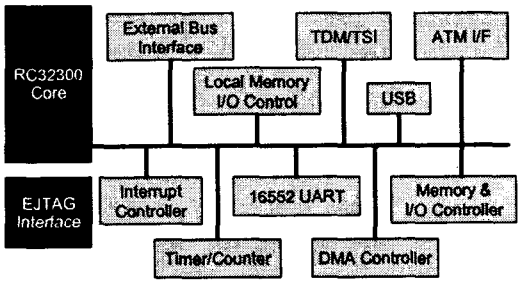

그림 6. RC32355와 EJTAG 인터페이스

지 디버깅하는 과정을 좀 더 편리하게 해주는 가장 적절한 방안이라 할 수 있다.

이들은 다음과 같은 기능들을 포함하게 되는데, 여기에는 UART와 디버그 모니터는 필요하지 않게 된다.

- EJTAG을 통한 코드 다운로드 기능
- UART나 디버그 모니터를 위한 어떤 프로세서 자원도 사용하지 않는다.
- 소프트웨어 및 복잡한 하드웨어 breakpoint 가능
- 실시간 PC trace 출력 기능
- 시스템 메모리와 프로세서 주변기기의 직접 액세스 기능

최근의 디바이스들은 외부 핀들의 수가 증가함으로 인해 이전의 방식으로는 테스트가 점점 어려워지게 되었으며 따라서 생산 라인에서도 검사가 어려워지게 되었다. JTAG이나 EJTAG을 사용함으로써 회로 경로가 쉽게 스캔될 수 있고 테스트 프로그램이 기능을 검증할 수 있게 되어 결함 확률을 획기적으로 줄일 수 있게 되었다.[6]

4. COP 기술

COP(Common On-chip Processor)는 원래 Motorola와 IBM 제품의 디버깅과 테스트를 위한 것이었다. PowerPC 계열의 마이크로프로세서 내에 구현되어 있으며, 이를 통해 여러 가지 유용한 디버깅 동작이 가능해진다. 즉, 프로세서 레지스터들의 값을 읽어오거나 갱신 작업, 메모리 내용 표시와 수정, 명령의 실행 등이 가능하다.

COP의 설계와 사용은 구현된 시스템마다 다를 수 있기 때문에 버전이나 모델에 따라 매우 신중한 접근을 요하게 된다. 특히 이에 대한 공식적인 문서화가 된 것이 없기 때문에 일반적으로 내부를 알기에는 매우 제한적이고, 따라서 Motorola와 IBM 외부에서 COP에 대한 표준이나 규격을 알기에는 상당히 어려운 상태이다.

이것은 산업체 나름대로의 지적 재산권이나 기술 유출에 대한 우려가 있을 수도 있고 시장 지배력을 가진 두 회사의 기술적 우위를 앞세워 복잡한 내부 기술보다는 보다 친숙한 사용자 인터페이스에 초점을 맞춘다는 의미도 있을 수 있을 것이다. 따라서 여기서는 공개된 문서나 제품들에서 알 수 있는 정도로 간략하게만 기술하기로 한다.

4.1 COP 개요

PowerPC 계열의 마이크로프로세서라도 JTAG 인터페이스를 위한 포트는 기본적으로 제공하게 된다. 여기에다 좀 더 많은 기능을 제공하게 위해 내부적으로 COP 기능을 추가하는 것이다.

COP 는 프로세서 레지스터들의 값을 읽어오거나 갱신, 메모리 내용 표시와 수정, 명령의 실행 등과 같은 JTAG 의 기능을 포함하고 있으며, 특히 breakpoint 기능은 프로그램의 디버깅 중에 프로세서의 레지스터들을 액세스할 수 있도록 해준다.

즉, Breakpoint, step, trace 모드는 COP를 통해 바로 초기화될 수 있다. 이러한 기능은 기존의 PowerPC 구조에서 특수 목적 레지스터(Special purpose register)들인 IABR(Instruction Address Breakpoint Register)와 MSR(Machine State Register)에 의해 이미 제공되고 있는 모드이다. 이렇게 COP를 사용하는 가장 중요한 잇점은 JTAG과 마찬가지로 이들 기능들을 비접촉(Non-intrusive)형태로 사용 가능하다는 점이다.

기존의 JTAG 직렬 인터페이스는 COP와의 양방향 통신이 가능하게 되어 있다. JTAG 명령은 보드의 상호연결 테스트에 주로 사용되고 COP

기능은 보드 디버깅과 시스템 하드웨어 및 소프트웨어의 개발에 사용될 수 있다. COP 내에서의 동작은 마이크로프로세서의 정상적 동작 중에 비동기적으로 실행될 수 있다는 것이다.

PowerPC 프로세서에 대한 데이터 문서를 보면 특수 레지스터들을 정리한 부분을 볼 수 있는데, 앞에서 언급한 IABR 등이 디버깅 및 시스템 개발에 유용하게 사용될 수 있는 것이다. 보통 프로세서 코어에 따라 조금씩은 다르지만 거의 동일한 구조를 가진다.

디바이스들의 스캔체인(Scan chain)에서 각 레지스터들의 위치는 버전이나 수정버전마다 다를 수 있고 이들을 액세스하는 방법도 부품마다 다를 수 있기 때문에 이전의 COP 디버그 프로그램이 무용지물이 될 수도 있다다

COP의 기능은 IEEE 1149.1 JTAG과 호환되며, 기존의 JTAG 명령에다가 추가적인 기능들을 더 가지고 있다. 이들은 공개되어 있지 않지만 기존의 PowerPC 디버깅 관련 제품 사용을 하거나 인터넷 등의 매체를 통해 알려진 것을 보면 많은 기능이 추가되어 있는 것으로 유추된다.

즉, JTAG 구조에다 COP 직렬 인터페이스 구조가 추가된 JTAG/COP 구조로 되어 있다. COP 명령은 기존의 JTAG 명령과 호환성이 있는데 이들은 Non-persistant 명령으로 지칭되고 있다. Non-persistant 명령이란 명령이 시작되면 매번 한번만 실행되고 자동으로 리셋되는 명령 형태이다. 이와는 별도로 Persistant 명령은 어떤 테스트 모드로 들어가면 이에 대응되는 리셋 명령이 실행될 때까지 그 모드를 유지하게 되는 명령이다. 이는 JTAG 명령과는 호환되지 않는 것이기 때문에 COP에 밀접한 관련을 가지는 명령이다.

따라서 TAP 제어부에 관련된 COP 제어부가 존재하며 프로세서 버전에 따라 많이 달라지기 때문에 이를 위한 레지스터, 그리고 외부 메모리를 위한 스캔체인 레지스터들이 추가되어 있다.

COP의 개념적인 구조는 그림 7과 같다.

COP 제어부는 On-chip 테스트를 위한 내부 기능을 위한 것이며, JTAG 과 COP 명령을 해독하고, 입출력 멀티플렉서를 제어하며 클럭에 대한 제어 기능을 제공한다.

또한 그림에서는 나와 있지 않지만 시스템 클럭에 관련된 카운터를 이용하여 내부 클럭 동작을 제어할 수 있는 기능도 있다.

스캔체인은 시스템의 테스트, 디버깅 및 개발 작업을 지원할 수 있으며, Instruction Register 가 인터페이스의 기본 경로로 이용되며, 나머지는 특정 명령에 의해 선택된다.

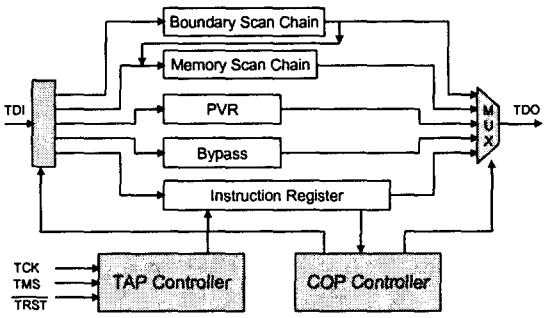

그림 7. COP 구조

4.2 COP 기능

COP는 Power-On-Reset(POS), 인터페이스를 통한 Diagnostic, Boundary scan 및 클럭 제어등의 기능을 제공하며, 특히 PowerPC 계열 프로세서들은 다음과 같은 breakpoint 기능도 추가로 제공한다.

- Single-step instruction trace
- 명령 주소 breakpoint
- COP 디버그 명령

이외에도 몇가지 기능들이 더 지원되는데, COP

명령은 JTAG 명령과 같은 방식으로 동작된다. 즉, TAP 제어부의 Update-IR 상태에서 TCK의 상승 에지에서 실행된다.

COP 명령은 앞에서도 언급한바와 같이 2종류로 나누어지는데, persistent와 non-persistent 그룹이다. 모든 JTAG 명령인 EXTEST, BYPASS, SAMPLE/PRELOAD, HIGHZ, CLAMP 등은 non-persistent 명령으로 분류되며, 이외에도 COP 명령인 EXMEM, HARD_STOP, SOFT_STOP 등이 있고 각각의 명령마다 대응되는 Reset 명령이 존재한다. Non-persistent 명령에는 COP_PVR, COP_SCAN, HALT, RESUME, STATUS 등이 있으며 이들 각각에 대한 구체적인 설명은 생략한다.

일반적으로 디버거 환경으로 들어가려면 현재 목표로 하는 타겟 프로세서에 대한 초기화 절차를 수행해야 한다. 보통 프로세서 종류에 대한 추적과 검사, 그리고 이에 대한 수정 작업등이 수행되어야 하며 다음의 절차를 거치게 된다.

- 프로세서를 중지시킨다.
- 프로세서로부터 메모리와 스캔체인을 읽어온다.
- 이들을 처리하고 변경한다.
- 이들의 값을 프로세서로 전달한다.
- 프로그램을 재동작시킨다.

4.3 디버거 도구

현재 Motorola, IBM의 PowerPC 계열 프로세서의 디버깅이나 시스템 개발 도구들은 라이선스 기반으로 제공되고 있으며, 이들에는 IBM에서 Hewlett packard, Corelis 등이 있으며, Motorola에서 Microtek, Macraigor, Lauterbach, WindRiver, AMC, HMI, ITT, YDC 등이 있다.

대부분의 도구들은 이전의 프린터 포트를 이용한 연결에서 지금은 고속의 정보 전달과 원격 접속 지원을 위해 LAN을 통한 연결을 지원하고 있다. 이들의 대표적인 연결 형태는 그림 8과 같다.

그림 8. 디버거 도구의 연결

여기서는 현재 많이 사용되는 디버깅 도구인 WindRiver사의 VisionICE 시리즈와 AMC의 PowerTap에 대해서 간략히 알아본다.

WindRiver사의 VisionICE와 VisionProbe는 Ethernet을 통해 호스트와 개발 보드와의 연결을 통해 소스 레벨 디버깅 기능과 메모리나 레지스터 값의 읽기/변경 등을 수행할 수 있는 강력한 도구이다. 마이크로프로세서에 내장된 On-chip 디버깅 서비스를 지원하는 하드웨어 지원 도구로서, 완전한 하드웨어와 소프트웨어의 통합을 위한 표준적인 소프트웨어 디버거이다. 또한 플래시 프로그래밍, 개발, 생산, 테스트에 있어 풍부한 기능을 제공한다. 대상 마이크로프로세서 종류는 PowerPC 계열을 비롯하여 MIPS32 계열, ARM 계열, Intel XScale 등 대부분의 프로세서를 지원한다. 다음의 기능들이 있다.

- LAN을 통한 연결 지원
- 실시간 기능 지원
- 내장된 하드웨어 검사 기능
- 플래시 메모리 프로그래밍
- 고속의 JTAG 인터페이스
- 메모리, 레지스터 액세스, 변경
- 가상 메모리 액세스 기능

이 도구는 최근 WIND PowerICE 라는 이름으

로 더욱 더 많은 기능을 제공하는 도구로 발전되었다.

다음으로 AMC(Applied Microsystems Corporation)사의 PowerTap, CodeTap 시리즈 역시 많이 사용되는 강력한 도구 중 하나이다. 이는 CodeWarrior라는 디버거 기술을 사용함으로 인하여 사용자 인터페이스, 문서화 및 관리를 On-chip 뿐만 아니라 Off-chip 레지스터에서 메모리까지 주변기기와 ASIC 디바이스까지 지원하고 있다.[7] 이들의 기능은 다음과 같다.

- 타겟 보드의 실행을 제어할 수 있다.
- 메모리와 레지스터들의 값을 확인하고 수정할 수 있다.
- Source level 디버깅 기능.
- Diagnostic 기능.
- LAN을 통한 접속을 지원.
- Telnet 접속을 통하여 타겟보드의 직렬 포트를 연결할 수 있는 기능을 제공한다.
- Overlay Memory와 Trace Processor 기능.

PowerTap에서의 Power-Up 과정은 다음과 같이 수행된다.

- 디버거는 CPU에 있는 Product Version Register (PVR) COP 명령을 통하여 읽어온다.이때 603e 코어는 PVR이 0으로 된다.
- 알맞는 명령어 파일을 로드하고 CPU를 확인하기 위해 PVR 값을 이용한다. (이는 디버깅 명령을 이용하여 직접 로드하고 확인할 수 있다.)
- $\overline{\text{QACK}}$을 구동하고 $\overline{\text{HRESET}}$과 $\overline{\text{TRST}}$를 구동한다.
- 프로세서r를 제어하기 위해 "Soft_Stop" 명령을 사용한다.

PowerTap은 3개의 리셋 명령을 지원한다.

- **reset**은 타겟의 $\overline{\text{TRST}}$와 $\overline{\text{HRESET}}$ 핀을 통해 구동된다.
- **sreset**은 JTAG 헤더의 $\overline{\text{SRESET}}$ 핀을 통해 구동된다.
- **treset**은 타겟의 $\overline{\text{TRST}}$ 핀을 구동하며 JTAG/COP 인터페이스의 재동기를 제공한다. 이때 COP 상태를 리셋하기 위해 여러개의 TMS 신호를 발생하게 된다.

이외에도 여러 다른 디버깅 도구가 있지만 현재 PowerPC 계열에서 가장 많이 사용되고 있는 VisionICE 시리즈와 PowerTap 시리즈에 대해서만 기술하였다.

4. 결 론

지금까지 마이크로프로세서와 이를 이용한 시스템 보드에 대한 디버거 기술 중 JTAG과 EJTAG인 COP 기술에 대해 알아보았다.

JTAG은 IEEE 1149에서 규정한 것으로 디지털 회로의 신호 레벨을 액세스 또는 제어할 수 있는 5개의 핀을 통한 디버깅을 지원하고 있다. JTAG 기술은 추가적인 비용 증가와 성능 감소가 있지만 이를 상쇄할 수 있는 자동화된 테스트에 따른 설계 및 생산 기간의 단축, 그리고 기존의 고가 테스트 장비를 저렴한 PC 기반 테스터로 대체할 수 있기 때문에 그 사용이 증가하고 있다.

또한 JTAG 디버깅 기능에다 프로그램 수행시의 추적이나 제어를 위해 여러 기능을 확장한 EJTAG 기술과 이들 중 많이 사용되는 Motorola 사의 COP에 대해 간단히 기술하였다.

디지털 회로와 그 디바이스들이 점점 복잡해지면서 이제는 JTAG과 같은 비접촉 테스트 기술이 없으면 디버깅은 거의 불가능한 수준으로 되었다. 거의 대부분 부품들은 이제 JTAG 인터페이스를

제공하고 있고, 이에 따라 설계 시점부터 개발, 생산까지 테스트가 더욱 쉽게 되었다. 이런 요구를 반영하여 더욱 쉽고 편리한 디버깅 도구가 나와 있고 앞으로는 아날로그 디바이스까지 테스트 뿐 아니라 신호 적용 및 측정까지 할 수 있는 기술인 IEEE 1149.4 지원 도구까지 상용화되면 디지털 회로와 아날로그, 그리고 그 혼합 회로들을 통합적으로 테스트, 디버깅할 수 있을 것이다.

참 고 문 헌

[1] IEEE, "IEEE Standard Test Access Port and Boundary-Scan Architecture", IEEE Std 1149.1-2001, 2001.
[2] IDT, "JTAG/EJTAG Devices", Integrated Device Technology, 2002.
[3] "DFT, JTAG, and the Testability Equation", System Test Access Solutions, 2002.
[4] Motorola, "PowerQUICC-II User's Manual", Motorola, 2001.
[5] www.falinux.com
[6] WindRiver, "Mutiprocessor JTAG Connection Scheme", WindRiver, 2002.
[7] "CodeRarrior's Support for the PowerQUICC II", Metrowerks's White paper, 2002.
[8] "IEEE 1149.1 Testability", 2001.

박 희 동

- 1982년 경북대학교 전자공학과(공학사)
- 1991년 포항공과대학교 전자계산학과(공학석사)
- 1999년 경상대학교 전자계산학과(공학박사)
- 1982~1994년 한국전자통신연구원(ETRI) 선임연구원
- 1994~1999년 마산대학 정보통신과 조교수
- 1999~현재 중부대학교 정보통신S/W공학과 부교수
- 관심분야 : 병렬 환경, 네트워크프로토콜, 운영체제, 임베디드 소프트웨어 등
- E-mail : hdpark@joongbu.ac.kr