

JTAG 인터페이스를 이용한 GUI 디버거 기술 동향

김병철* · 박희동**

1. 서론

지금까지 JTAG을 이용한 마이크로프로세서, 디지털 회로 및 시스템의 설계부터 개발, 생산까지 필요한 테스트 작업을 위해 많은 디버깅 도구들이 출시되었다.

IEEE 1149.1으로 알려진 JTAG(Joint Test Action Group) 디버깅 기술은[1] 그 자체적으로는 하드웨어 및 시스템 레벨의 테스트 규격이지만 이를 실제 사용자들이 활용하기에는 다양한 응용 인터페이스가 필요하게 된다. 따라서 하드웨어적인 디지털 회로에 대한 신호 레벨을 액세스 또는 제어할 수 있어야 하는데, JTAG의 5개 핀을 통한 직렬 프로토콜을 이용하여 사용자가 보다 편리하고 간단하며 쉽게 사용하는 GUI(Graphic User Interface) 기능을 접목시켜 디버깅을 할 수 있는 제품들이 많이 나오고 있다.

이는 반도체나 시스템 보드 기술이 발전되고 점차 디바이스도 소형화되면서 내부적으로 복잡해짐에 따라 시스템적으로 처리해주는 레벨과는 별도로 사용자와의 친숙한 인터페이스를 가진 높은 수준의 테스트와 디버깅 인터페이스는 필수적으로 요구하게 된 것으로 기존의 기술에 추가적인 기능을 포함하여 지속적으로 발전하고 있다.

CPU의 종류가 다양하면서도 고성능을 요구하게 되면서, 특히 인터넷의 급속한 발전으로 인한 통신 시스템 기술의 발전 속도에 맞추어 지속적으로 새로운 네트워크 장비의 출시가 중요해지고 있다. 따라서 이들 장비들이 시장에 나오는데 걸리는 시간이 점차 줄어들면서, 개발 및 생산 기간의 단축 요구는 시스템 테스트 및 디버깅 방식에도 많은 발전을 요구하게 된 것이다. 이러한 요구에 부응하여 기존의 JTAG 기능을 활용한 그래픽 사용자 인터페이스의 강화를 기본으로 하는 다양한 개발 도구가 나오게 되었다.

본 논문에서는 JTAG 기술을 이용하여 디바이스, 보드, 시스템 테스트와 디버깅을 위한 GUI 기술을 위주로 기존의 제품과 기술적인 동향에 대해 기술하기로 한다.

2. JTAG 디버거 구조 및 기능

일반적으로 JTAG을 이용한 테스트 및 디버거 도구는 그림 1과 같이 소프트웨어와 하드웨어를 함께 사용하는 시스템 구조와 그림 2와 같은 소프트웨어가 대부분인 구조가 있다. 이들은 디버거의 속도(성능)와 사용의 편리성, 호환성, 비용 등을 고려하여 구성된다.[3]

이런 구성인 경우에 JTAG 인터페이스에 필요한 대부분의 하드웨어적인 신호는 POD에서 처리

* 경상대학교 자연과학대학 컴퓨터과학 석사과정

** 중부대학교 정보통신대학 정보통신S/W공학 부교수

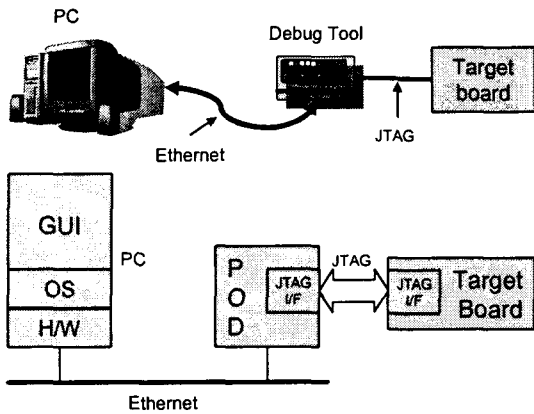


그림 1. 하드웨어 POD 디버거 구조

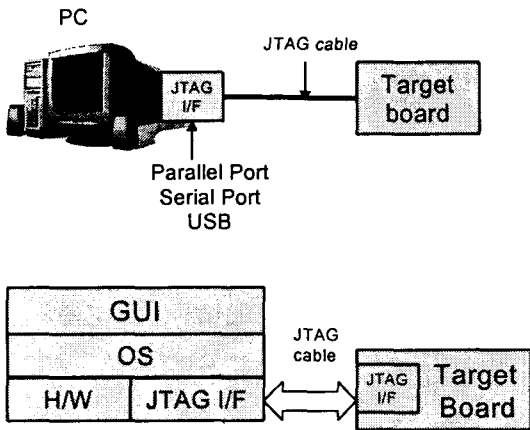


그림 2. 소프트웨어 디버거 구조

된다. 즉, PC에서 요청된 정보는 Ethernet을 통해 JTAG 직렬 프로토콜을 처리해주는 POD로 전달되고 이 데이터는 여기에서 처리되어 타겟 보드로 전달된다. 타겟 보드에서 실행된 결과는 다시 JTAG I/F를 통해 POD로 전해지고 POD는 이 데이터를 처리하여 Ethernet을 통해 PC로 전달하여 사용자가 쉽게 이해할 수 있는 형태로 변환되어 화면에 표시해준다. 따라서 POD는 JTAG 관련 디버깅 관련 처리만 수행하고 JTAG 전용 칩을 사용하기 때문에 그 속도나 성능이 높아지게 된다. 또한 타겟 보드의 종류가 달라지더라도 POD만 거기에 맞는 것으로 변경하면 되므로 좀 더

유연성을 가질 수 있다.

이 구조는 PC에서 대부분의 JTAG 인터페이스 처리가 이루어지고 그 신호가 PC에 장착된 JTAG I/F를 통해 타겟 보드에 전달된다. 보드에서 처리된 결과는 바로 PC로 전달되어 PC에서 JTAG 처리 결과를 해석하여 화면에 출력해주는 구조이다. 이런 구조는 JTAG 관련 기능이 거의 대부분 PC에서 소프트웨어로 처리되기 때문에 속도나 성능 면에서 떨어지게 된다. 또한 JTAG I/F 역시 저속의 인터페이스를 사용하므로 인해 병목현상이 발생할 수도 있다. 하지만 최근에는 USB와 같은 고속의 인터페이스를 사용하기 때문에 이로 인한 병목 현상은 점차 줄어들고 있다.

JTAG 디버거에서는 일반적으로 다음과 같은 기능들이 지원된다.

- ▶타겟 보드 초기화
- ▶메모리 내용 출력 및 변경
- ▶메모리 내용 검사, 비교 및 이동
- ▶레지스터 내용 출력 및 변경
- ▶Breakpoint 설정 및 해제
- ▶소스(Source) 레벨 디버깅
- ▶플래시 메모리 지움 및 프로그래밍
- ▶하드웨어 검사

여기서 소스 레벨 디버깅은 여러 가지 기능이 혼합되어 구현되는데, 일반적으로 다음과 같은 기능들이 주로 지원된다.

- ▶Step: 일반 명령문일 때는 한 라인씩 처리해 주고 함수를 콜하는 부분일 경우에는 그 함수로 진입하여 하나 하나씩 전부 수행한다.
- ▶Next: 일반 명령문일 때 한 라인씩 처리해 주고 함수를 콜할 때에도 하나의 명령어 라인처럼 다음 라인으로 넘어가면서 처리해준다.
- ▶Finish: 현재 처리하고 있는 함수를 다 처리하고 빠져 나온다.

- ▶ *Continue*: 현 위치를 기준으로 다음 break-point가 나올 때까지 함수나 명령어를 구분 없이 연속적으로 처리해준다.
- ▶ *Step ASM*: source 화면에서 실행되고 있는 source를 어셈블리어 명령어 체계로 변경한 후 한 라인의 명령어씩 처리하고 콜하는 부분에서는 그리로 진입하여 처리한다.
- ▶ *Next ASM*: source 화면에서 실행되고 있는 source를 어셈블리어 명령어 체계로 변경한 후 콜하는 부분이나 명령어나 구분 없이 한 라인씩 처리한다.
- ▶ *Down Stack Frame*: 한 함수에서 다른 함수를 콜하고 또 다른 함수를 콜하여 처리하는 과정을 함수 이름을 프레임으로 스택 처리한 후 처리한 프레임을 한 단계 아래로 내려 실행한다.
- ▶ *Up Stack Frame*: 함수 이름을 기준으로 처리한 스택 프레임을 현 프레임에서 한 단계 올려서 실행한다.
- ▶ *Go to Bottom of Stack*: 함수 이름을 기준으로 처리한 스택 프레임을 가장 아래로 내려서 실행한다.

이들 각각의 기능들은 JTAG의 기본 기능을 이

용하여 복잡한 절차를 만들어 전달되어 처리된다. 보통 하나의 기능을 수행하기 위해서는 수천 바이트의 데이터가 여러 번 전달되고 그 속도는 JTAG의 TCK의 속도와 그 전송 속도에 연관된다. 따라서 하나의 명령어를 처리하기 위해서 많은 정보의 송수신에 따른 속도와 성능에 대해 고려하여야 하므로 인터페이스의 전송 속도가 높은 USB나 Ethernet을 사용하는 도구가 증가하고 있다.

3. GUI 디버거 도구

지금까지 JTAG에다 필요한 기능들을 추가한 여러 가지 확장된 디버깅 기술 규격이 있는데, 대표적인 것이 IBM/Motorola PowerPC 프로세서들을 위한 COP(Common On-chip Processor), NEC의 N-wire/N-Trace, IBM의 RISCWatch, Motorola IC들에 사용되는 BDM(Background Debug Mode)등이 있다. 이들 기술들을 활용하여 각종 마이크로프로세서와 디지털 시스템을 테스트할 수 있는 상용 도구들은 많이 나와 있다.

여기서는 현재 이들 도구 중, 임베디스 시스템 개발 도구로 주로 사용되는 대표적인 VisionICE [12], VisionProbe[13], 그리고 PowerTAP[14] 이

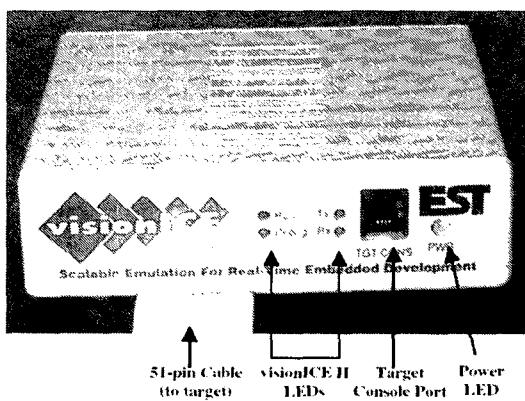


그림 3. VisionICE 전면부

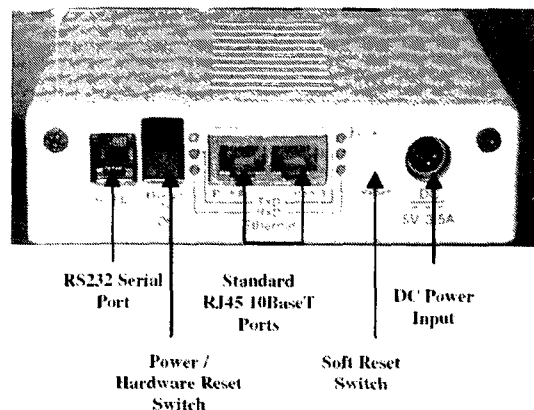


그림 4. VisionICE 후면부

라 불리는 도구의 인터페이스 특징을 기술하고 비교하도록 한다.

3.1 VisionICE

WindRiver사의 VisionICE 는 VisionProbe에 비해 월등한 성능을 보여주고 있다. 현재 대부분의 개발 도구에 적용되고 있는 PC와 POD 사이의 Ethernet으로 연결이 되며, 개발 PC와 타겟 보드 간에 모니터링을 위해 직접 직렬포트 연결을 하지 않고 VisionICE 에 직렬포트를 연결하여 원격으로 모니터링을 할 수 있다. 아래 그림에서와 같이 전면 부에는 타겟 보드와 연결되는 51 핀 케이블, 상태/전원 LED, 콘솔 포트가 구성되어 있으며, 후면 부에는 VisionICE의 모니터링을 위한 RS232-C 포트(9600bps), 전원 스위치, 10/100 이더넷 포트, 전원 입력부로 구성되어 있다.

VisionICE의 특징으로는 위에서도 언급한 바와 같이 개발 PC와의 10/100 Ethernet 연결을 지원한다. 그리고 RTOS를 지원하며 하드웨어 진단 기능도 지원한다. 임베디드 시스템에서는 타겟 보드에 있는 플래시 메모리에 프로그래밍도 지원한다. 그 외 추가적인 기능으로는 외부 트리거, 소스 레벨 디버깅, 캐쉬 메모리 액세스를 지원하게 된다. 또 다른 특징으로는 하드웨어가 모듈화 되어 있어 추가적인 기능이 필요할 때는 VisionICE에 모듈화 된 하드웨어를 부착함으로써 이를 해결할 수 있게 되어 있다.[3,4]

개발 도구에 있어서 중요한 점 중에 하나는 열

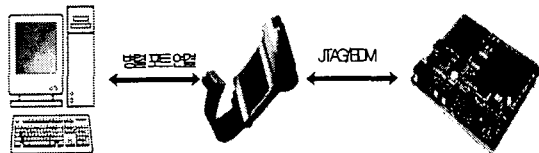


그림 5. VisionProbe의 구성 및 연결

거나 다양한 종류의 CPU를 지원하느냐 하는 것인데, VisionICE는 다양한 CPU를 지원함으로써 이러한 문제점을 해결해 준다. 어떠한 개발 도구킷은 처음 구입 시 제공되는 CPU만을 지원하는 경우가 있는데 이는 경제적으로 볼 때, 좋은 개발 도구라고 말할 수 없을 것이다.

VisionICE에서 지원하는 CPU는 다음과 같으며, 현재 임베디드 시스템에서 많이 사용되고 있는 CPU는 거의 모두 지원하고 있다.

- ▶ Motorola/IBM PowerPC architecture
- ▶ Motorola ColdFire architecture
- ▶ MIPS32/MIPS64 architecture
- ▶ ARM architecture
- ▶ Intel XScale

Vision-ICE는 자체적으로 내부 OS가 있으며, telnet 접속과 시리얼 접속을 지원한다. 이러한 기능은 개발자로 하여금 개발 도구의 상태를 좀더 효율적으로 관리할 수 있을 뿐만 아니라, 이로 인해 개발 기간도 단축시킬 수 있는 효과가 있다.

3.2 VisionProbe

VisionICE와 VisionProbe의 가장 큰 차이점은 호스트 PC와 연결 시 Ethernet 포트가 아닌 병렬 포트를 이용한다는 것이다. 또한 그 기능상의 차이도 존재한다. 이러한 차이점은 개발자에게는 어느 정도의 불편함을 남겨둘 소지가 있다. 예를 들어 VisionProbe와 같이 호스트 PC와 병렬 포트로 연결 시에는 항상 호스트 PC <--> VisionProbe <--> 타겟 보드 식의 순으로 연결이 불가피 하지만, VisionICE 같은 경우에는 VisionICE 자체에 IP 주소가 할당 되므로 네트워크가 연결만 되어 있다면, 호스트 PC와 VisionICE의 연결은 어느 정도 자유로워 질 수 있다 하겠다.

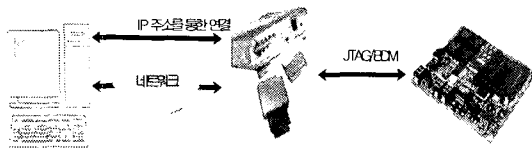


그림 6. VisionICE의 구성 및 연결

VisionProbe의 특징은 위에서도 언급한 대로 병렬 포트를 사용하여 호스트 PC와 연결하며, VisionICE에서 지원하는 telnet/시리얼을 통한 개발 도구의 상태를 알아 볼 수 있는 기능은 포함되어 있지 않다. 이 두 가지를 제외한 기능은 Vision ICE와 흡사한 기능을 가지고 있다.

3.3 PowerTAP

PowerTAP 도구는 Applied Microsystems Corporation(AMC)사의 제품군 중 하나로 CodeTAP/PowerTAP/PowerTAP PRO로 대표될 수 있다. AMC사의 제품군 중 가장 최신의 개발 도구는 PowerTAP PRO로서, Motorola PowerPC RISC Processor를 지원하며, 기존 CodeTAP 에서 지원하던 MPC8xx Family와 PowerTAP에서 지원하던MPC82xx/7xx/74xx/6xx, IBM405GP/405CR/7xx 계열을 모두 지원한다. 기존 PowerTAP에서는 16MHz의 JTAG 클럭 속도를 지원했으나, PowerTAP PRO는 64MHz의 JTAG 클럭 속도를 지원하고 100Base-T의 Ethernet을 지원한다.

주요 특징으로는 기존에 PowerTAP에서 사용되어진 MWX-ICE를 사용하지 않고 CodeWarrior 디버거를 사용한다. 그리고 ROM/Cache/RAM에서 디버깅이 가능하며, VisionICE와 마찬가지로 하드웨어 진단 기능이 지원된다. MPC7xx를 위해 MMU/L1/L2/L3를 지원하며, Chorus/GNU, CYGNUS/GNU, DIAB, Greenhills, Metawave, CodeWarrior, MRI 또는 Tornado/GNU 컴파일

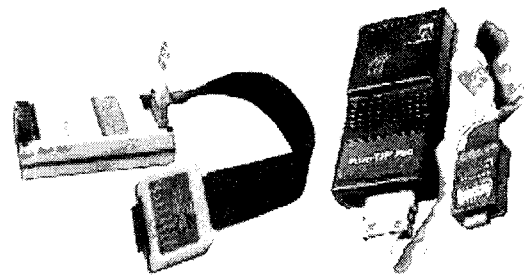


그림 7. PowerTap/CodeTap

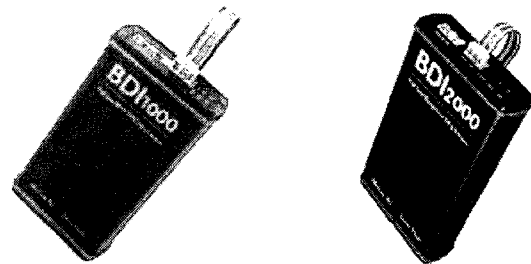


그림 8. BDI 시리즈 외관

러를 지원한다. 플래시 메모리 프로그래밍이 가능하며 주로 AMD, Intel, Sharp사의 제품을 지원하게 된다.

3.4 BDI 시리즈

BDI1000과 BDI2000[8, 9]은 Abatron사의 제품군으로, 주요 특징은 Motorola의 CPU 12/16/32/32+, PowerPC 5xx/8xx, ColdFire의 BDM Port와 ARM, M-CORE, PowerPC 6xx/7xx/82xx, TriCore의 JTAG 포트를 모두 지원하는 디버깅 도구이다. 호스트 플랫폼과의 연결은 시리얼 또는 Ethernet으로 통신하며, CodeWarrior, Ada, Wind River, XRAY, X-Tools 등 사용자가 사용하기 편리한 디버깅 소프트웨어를 소프트웨어적인 인터페이스 모듈을 변경하여 사용할 수 있다. 또한 별도의 bdiPro 소프트웨어를 이용하여 Intel, AMD 등의 플래시 디바이스를 고속으로 프로그

람 할 수 있다.

1) X-Tools

GNU X-Tools[15]은 21개의 마이크로프로세서를 지원하는 도구이며, 각각의 프로세서에 대한 컴파일러, 어셈블러, 디버거, 라이브러리와 유틸리티 등을 포함하고 있다. 또한 코드 에디터, 소스 브라우저를 갖춘 그래픽 IDE 프레임워크를 제공한다. 이러한 GNU X-Tools은 GNU Public License (GPL)에 의거 모든 소스 코드를 함께 제공한다.

X-Tools는 Cygwin을 이용한 윈도우 버전과 리눅스 버전이 존재하며, GUI는 Tcl/tk로 동작하고, 에뮬레이터와의 통신은 시리얼과 Ethernet을 지원한다. X-Tools의 디버거는 GNU 디버거인 GDB를 기반으로 동작하며, Redhat사의 Insight를 기반으로 개발되었다.

GUI의 구성은 다음과 같다.

- ▶타겟 선택
- ▶레지스터 표시 및 설정
- ▶메모리 표시 및 설정
- ▶소스 표시
- ▶Breakpoint

타겟 선택(Target Selection)이란 GNU의 GDB 원격 디버깅과 마찬가지로 GUI는 클라이언

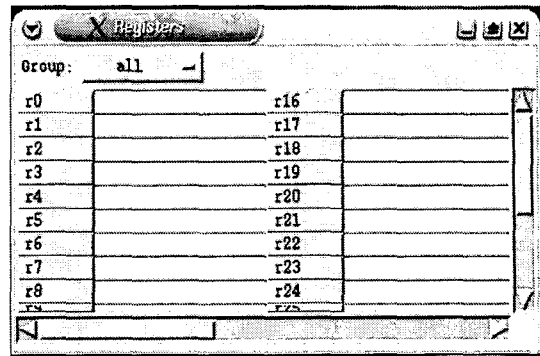


그림 10. 레지스터 표시 및 설정 화면

트로 동작하며, GDB 서버로 접속하기 위한 연결 방법에 대한 설정을 할 수 있다. 만약 "Remote/TCP"를 이용한다면 GDB 서버의 IP 주소와 포트 번호 등을 입력하게 된다.

레지스터 표시 및 설정은 타겟의 레지스터들에 대한 표시와 설정이 가능하다. 그림 과 같이 표시 된다. 메모리 표시와 설정은 타겟의 메모리에 대한 표시와 설정이 가능하다. 소스 표시 기능은 소스 레벨 디버깅을 위한 C, C++, 어셈블러 소스를 표시하게 되고, 디버깅 옵션(Run, Continuous, Next, Step, Stop 등)을 이용하여 소스 디버깅이 가능하다. Breakpoint는 소스 레벨 디버깅을 위하여 매우 중요한 기능으로서 소스 윈도우에서 마우스를 통하여 매우 쉽게 설정이 가능하다.

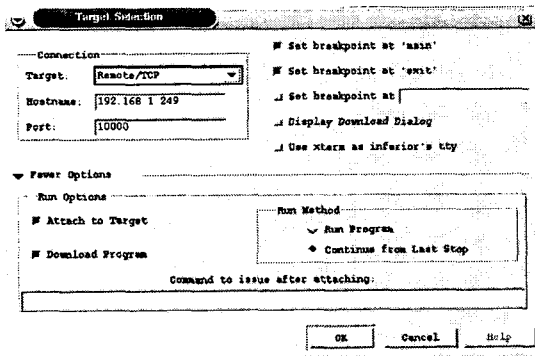


그림 9. 타겟 선택 화면

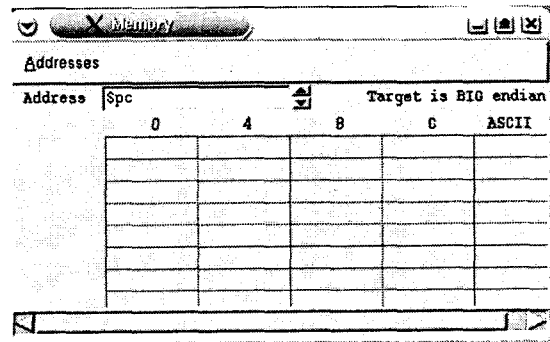


그림 11. 메모리 표시 및 설정 화면

4. 소프트웨어 디버깅 도구

Vision ICE와 VisionProbe는 윈도우 기반의 VisionClick[10]과 유닉스 기반의 VisionXD[11]를 사용하며, PowerTAP은 MWX-ICE[14], CodeWarrior를, PowerTAP PRO는 CodeWarrior를 통해서 디버깅이 가능하다. 본 장에서는 VisionClick과 MWX-ICE의 GUI 특징에 대해서 중요한 몇 가지의 측면에서만 비교 설명하기로 한다.

4.1 디버깅 도구 환경설정

처음 디버깅 도구를 사용하기 위해서는 환경 설정을 해야 한다. 환경 설정 부분은 도구마다 설정 방법과 항목에 많은 차이를 보인다. VisionClick은 프로젝트별로 관리할 수 있는 GUI 방식이며, 반대로 MWX-ICE는 프로젝트별로 관리가 불가능하고 사용자가 설정을 일일이 TEXT 기반으로 해야 하는 번거로움이 있다. 환경 설정 부분에서는 MWX-ICE가 다소 불편한 점이 있다.

그림 14는 VisionClick에서의 환경 설정 화면이다. VisionClick에서의 환경설정 화면은 프로젝트별로 설정이 가능하며, 저장/수정 또한 가능하다는 것을 알 수 있다. 기본적으로 메뉴는 Project Configuration, Load Option, Communication,

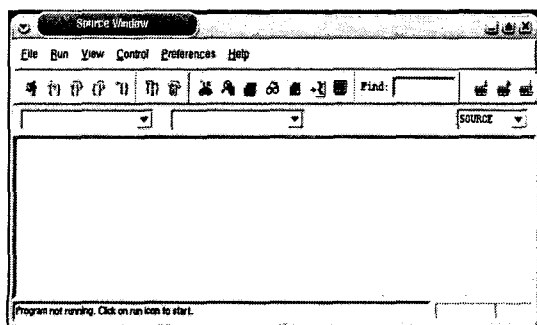


그림 12. 소스 코드 표시 화면

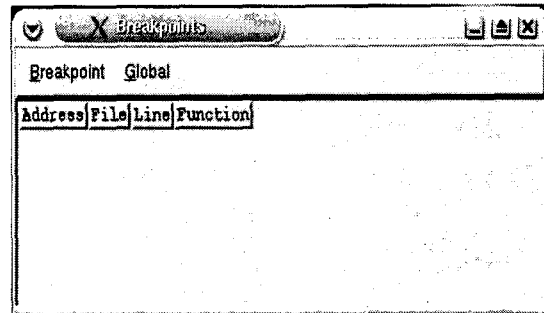


그림 13. Breakpoint 표시 화면

Configuration으로 구성되어 있으며, Project Configuration에서는 프로젝트 전반에 걸친 항목을 설정하고, Load Option에서는 다운로드 받을 이미지가 위치한 경로를 설정하며, Communication에서는 호스트 PC와 ICE간의 통신 방식 및 주소할당을 정하게 된다. 마지막으로 Configuration에서는 타겟 보드에서 사용되는 CPU, Endian Type등 타겟 보드에 관련된 사항을 주로 설정하게 된다.

그림 15는 MWX-ICE에서 환경설정 방법이며, VisionClick에서와 같이 모든 설정을 GUI로 하는 것이 아니라 타겟 보드에서 사용되는 CPU만 선택 후 include 파일 경로를 설정하는 형태로 되어

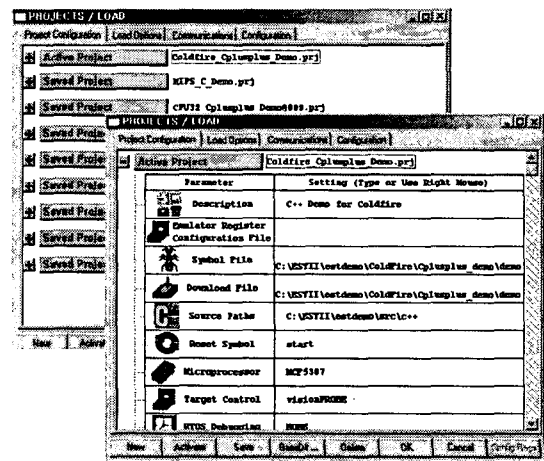


그림 14. VisionClick에서의 환경 설정 화면

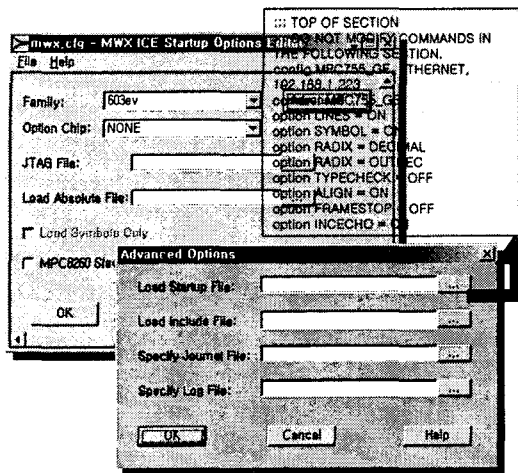


그림 15. MWX-ICE에서의 환경 설정 화면

있다. 여기서 include 파일은 VisionClick에서 설정한 항목과 비슷한 항목을 GUI가 아닌 텍스트 형태로 기술한 후 로드하여 사용하는 형태를 취한다. 이러한 형태의 환경설정은 유용성이 있을 수는 있으나, 처음 개발 도구를 사용하는 사용자 입장에서는 상당히 불편함을 느낄 수 있다.

4.2 레지스터 출력 및 설정

마이크로프로세서 내의 레지스터 표시 및 설정 기능은 디버깅 시 유용하게 사용되는 메뉴 중 하나이다. 이는 소스 코드를 스텝별로 디버깅할 때 관련된 레지스터의 정보를 한눈에 쉽게 볼 수 있고, 수정 및 추가가 용이하여야 한다.

예로 그림 16과 그림 17에서 볼 수 있듯이 레지스터의 각 비트별 설명이 자세히 나와 있음을 알 수 있다. 만약 필요한 레지스터가 누락되거나 삭제되었을 경우에는 사용자 환경에 맞게 추가도 가능하다. 이러한 면에서 VisionClick이나 MWX-ICE 모두 기능적인 면에서 사용하기 쉽게 되어 있음을 알 수 있다. 부가적인 기능으로 타겟 보드가 가지고 있는 현재 레지스터의 값을 파일로 저장

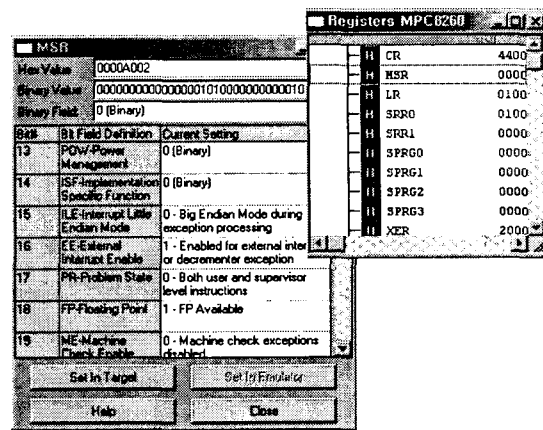


그림 16. VisionClick의 레지스터 표시 화면

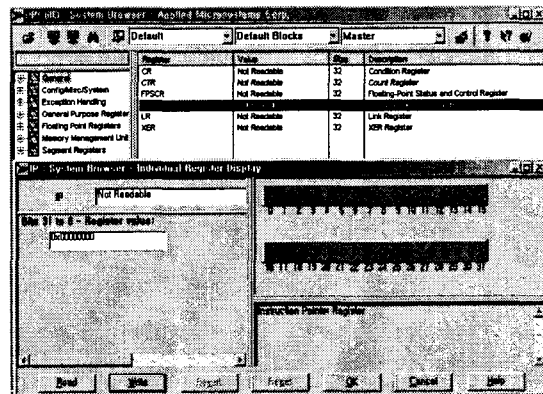


그림 17. MWX-ICE의 레지스터 표시 화면

가능하며, 반대로 레지스터 값을 다운로드하여, 타겟 보드로 전송하는 것도 가능하다.

4.3 플래시 메모리 프로그래밍

일반적인 플래시 메모리에는 타겟 보드를 초기화 시키고 OS나 애플리케이션을 동작시킬 부트로더(BootLoader), 그리고 시스템이 재 시작 되었을 때에도 남아있어야 할 데이터 등을 저장하게 된다. 이때 개발 도구에서 지원하는 플래시 메모리 프로그래밍 기능을 이용하여, 쉽게 프로그래밍할 수 있다. 플래시 메모리에 프로그래밍하기 위

해서는 타겟 보드에 부착되어 있는 제조사와 타입을 정확히 숙지하고 있어야 한다. 예외적으로 각 디버깅 도구에서 지원하지 않는 플래시 메모리가 있을 수가 있다. 이때 VisionClick은 제작사에 의존적이나, MWX-ICE 같은 경우는 플래시 타입과 관련된 파일을 사용자가 타겟 보드에 맞게 수정하여 사용할 수 있는 유연성을 보여준다. 이외 기능은 GUI 측면에서 보면 두 가지 도구 모두 충족할 만한 기능을 제공하고 있다.

4.4 소스 디버깅

디버깅 도구에서 소프트웨어에 대해 중요한 기능이 소스 디버깅 기능이라고 할 수 있다. 이 기능에서 중요한 부분은 High Level과 Low Level Mode 동작이 지원되는지인데, 이는 C, C++와 같은 고급 언어와 Assemble 언어가 제대로 표현되는지와 소스코드의 효율적 관리 기능, 데이터 추적 기능을 제대로 나타내는지 등이다. 이들 기능에 대한 지원은 두 가지 도구 모두 High /Low Level 언어들에 대한 화면 표시는 비슷한 측면을 보여주고 있다. 하지만 소스 코드의 효율적 관리 측면에서는 MWX-ICE가 다소 부족하며 데이터 추적 기능도 다소 부족한 모습을 보여주고 있다. 그림 18은 VisionClick에서 소스를 모듈/함수/심블로 나누어 효율적으로 관리하는 것을 볼 수 있다. 특히 그림 19에서 보는 바와 같이 C와 같은

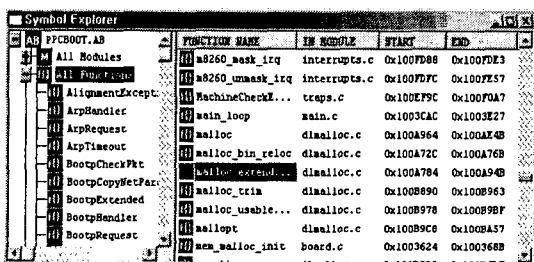


그림 18. VisionClick에서 소스 관리 화면

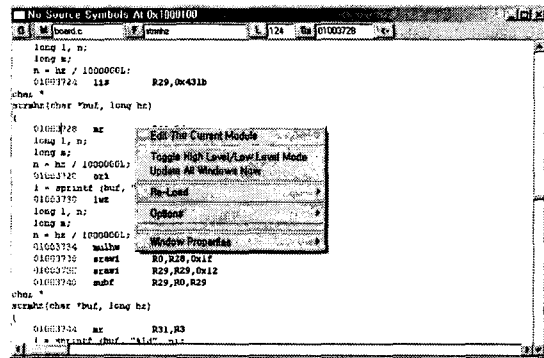


그림 19. VisionClick에서 고급 언어와 어셈블러 표시 화면

고급 언어와 어셈블러를 동시에 표시해주는 기능이 있어 고급 언어와 어셈블러와의 관계를 명확하게 알 수 있게 해준다.

4.5 Breakpoint 지원 기능

VisionClick 도구에서 지원하는 Breakpoint는 소프트웨어 및 하드웨어 Breakpoint를 지원하며, 내부 하드웨어와 하드웨어 Breakpoint는 VisionICE에 VisionContorl과 VisionEvent 모듈을 추가 시에 사용할 수 있으며, 기본적으로 소프트웨어 Breakpoint를 사용할 수 있다.

Breakpoint 설정은 일반적인 디버깅 도구에서와 같이 소스 윈도우 창에서 설정할 수 있으며, VisionClick에서 제공되는 Symbol Explorer에서도 설정할 수 있다. 부가적으로 설정한 조건에 맞을 때 Breakpoint를 발생시키는 기능도 포함되어 있다.

그림 20은 MWX-ICE에서의 Breakpoint 설정시의 화면이다. MWX-ICE에서는 액세스/명령 Breakpoint(읽기/쓰기)를 지원한다. 액세스 Breakpoint는 405GD와 740/750 프로세서에서만 지원되는 기능으로 기본적으로 명령 Breakpoint를 사용할 수 있다. MWX-ICE에서 Breakpoint 처리는 다음과 같다.

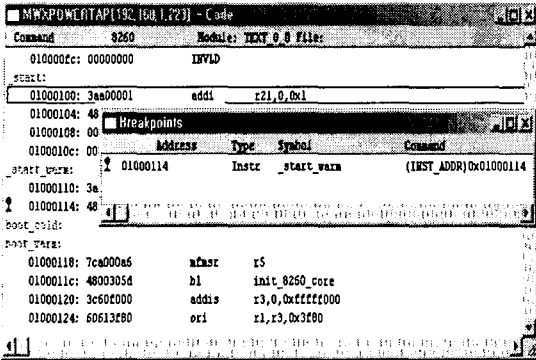


그림 20. MWX-ICE에서의 Breakpoint 설정 화면

- ▶ Breakpoint가 설정된 주소에서 서스펜드 (suspend) 프로그램이 실행된다.
- ▶ Breakpoint가 설정된 주소에서 원래의 명령을 트랩 명령으로 대체한다.
- ▶ Breakpoint에 매크로가 설정되어 있을 때에는 매크로가 반환하는 값에 따라 프로그램을 진행하기도 또는 다시 시작하기도 한다.

프로그램 소스의 메모리 주소와 그 코드, 그리고 소스 등을 표시해 주면서 필요한 번지에 Breakpoint를 설정한 것을 보여준다. 그림 21은 VisionClick에서의 Breakpoint 설정 화면이다.

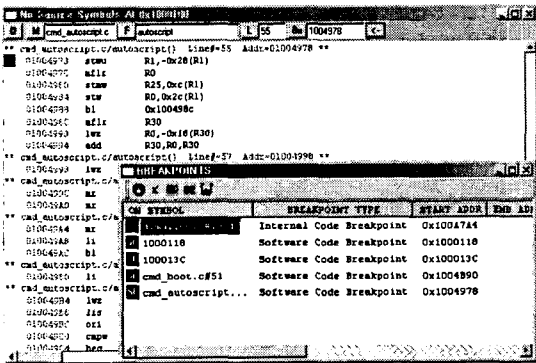


그림 21. VisionClick에서의 Breakpoint 설정 화면

4.6 메모리 내용 표시 기능

메모리 내용은 두 도구 모두 편리하게 표시해

주고 그 값을 변경할 수 있는 기능을 제공해준다. 그림 22는 VisionClick의 메모리 표시 화면이다.

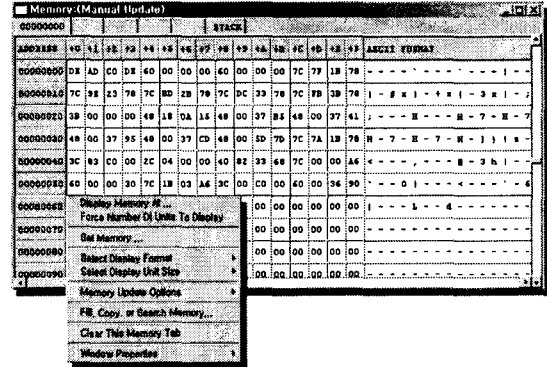


그림 22. VisionClick의 메모리 표시 화면

5. 결론

본 논문에서는 JTAG 인터페이스를 이용한 테스트 및 디버거 기술 중 사용자 편리성에 중점을 둔 GUI 디버거 기술 및 제품 동향에 대해 기술하였다.

인터넷의 발전과 이동 개인기기의 사용이 급격히 늘어나면서 새로운 종류의 시스템이 지속적으로 출시되는 데는 이들 장비들의 개발 및 시험기간의 단축이 그 한자리를 차지하고 있다. 이러한 GUI 디버거들은 거의 대부분 외국 장비들이 차지하고 있으며 국내에서도 몇몇 산업체에서 간단한 디버거를 출시한 적이 있다. GUI 디버거들의 구조도 점차 LAN을 접속하여 원격에서도 작업이 가능한 형태로 바뀌어가고 있으며, JTAG 테스트 기능뿐 아니라 프로그래밍까지 추가하는 확장 기능으로 발전되고 있다.

여기서는 PowerPC 계열과 ARM 계열, 그리고 Intel CPU 등에 주로 사용되는 WindRiver사의 디버거 하드웨어인 VisionICE와 VisionProbe와 AMC사의 PowerTap, 에 대해 간단히 비교 기술하였다. 또한 Abatron사의 BDI 시리즈는 GNU

소프트웨어를 사용하는 것으로 윈도우와 리눅스를 모두 지원하고 대상 CPU도 거의 대부분을 포함하고 있지만 GUI 기능은 좀 부족한 면이 존재한다. VisionICE와 VisionProbe에서는 VisionClick이라는 GUI 인터페이스를 사용하고, PowerTap에서는 MWX-ICE를 사용하지만 이는 점차 CodeWarrior로 대체되고 있다. 이들 GUI 인터페이스들은 사용자들에게 매우 편리하고 강력한 기능을 제공해준다.

이러한 사용자 편리성과 지원하는 기능의 다양성, 그리고 고성능은 많은 GUI 인터페이스가 추구하는 것으로 앞으로 점점 더 강력한 JTAG을 이용한 GUI 도구가 나올 것이다.

참 고 문 헌

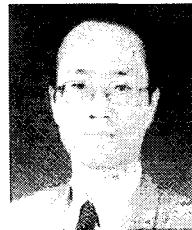
[1] IEEE, "IEEE Standard Test Access Port and Boundary-Scan Architecture", IEEE Std 1149.1-2001, 2001.
 [2] http://www.embedded.com/design_library/esd/d/OEG20021121S0049
 [3] "Hardware Debugging on MMU based Operating Systems", http://www.embedded.com/db_area/motorola2003/artic04.html
 [4] Motorola, "PowerQUICC-II User's Manual", Motorola, 2001.
 [5] <http://www.embedded.com/1999/9910/9910sr.htm>
 [6] <http://www.embedded.com/97/br9711.htm>
 [7] <http://www.embedded.com/1999/9909/9909feat1.htm>
 [8] "Using the Abatron BDI2000 to debug a Linux kernel", Ultimate Solutions, Inc. 2002.
 [9] "Performance BDI1000/BDI2000", Abatron AG, 2002.
 [10] "Vision Click 7.3 User Manual", WindRiver, 2002.
 [11] "Vision Ware/XD User Manual", WindRiver, 2002.

[12] "VisionICE II Product Overview", WindRiver, 2002.
 [13] "VisionProbe II Product Overview", WindRiver, 2002.
 [14] PowerTAP RPO with MWX-ICE", Metro-Werks, 2003.
 [15] "XTools Overview", Embedded Tools Network, 2003.



김 병 철

- 2001년 경상대학교 컴퓨터과학과 학사졸업
- 2002년~현재 경상대학교 컴퓨터과학과 석사과정
- 관심분야 : 운영체제, 병렬프로그램 디버깅, 임베디드 소프트웨어
- E-mail : banggl2@race.gsnu.ac.kr



박 회 동

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