문제를 일으키기 쉬운 BGA 패키지의 디버깅 로직

Debugging logic in BGA packages can be problematic

Tim Allen, Altera
번역 : **한선형**

핀의 개수가 증가하면서 점점 BGA 패키지를 많이 이용하고 있다. 하지만 BGA는 디바이스의 핀을 직접 액세스할 수 없기 때문에 디버깅이 쉽지 않다.

지금까지 칩의 패키지 기술은 칩의 게이트 수 증가와 보조를 맞추어 발전해 왔고 연결핀의 개수와 속도가 증가하면서 Dual Inline Package(DIP)나 Plastic Leadless Chip Carriers(PLCC), Quad Flat Packs(QFP)에서 ball-grid array(BGA) 패키지로 발전하게 되었다. 하지만 BGA 패키지를 이용하게 되면 제조 과정이나 테스트, 디버깅시에 문제가 발생할 소지가 늘어나게 된다. 특히, 디바이스의 핀이 말 그대로 패키지 밑에 묻혀 있기 때문에 디버깅에 대한 문제가 심각해진다. BGA에서는 핀을 직접 액세스할 수 없기 때문에 원하는 핀을 찍어 확인하였던 기존의 방법을 더 이상 사용할 수 없다.

QFP 패키지는 핀이 주변 장치에 노출된 형태이기 때문에 핀의 개수가 250개 이상이 되면 그 크기가 엄청나게 커진다. 304핀을 갖는 RQFP 패키지의 크기는 2.8인치인데 이것은 324핀의 Fine-Line BGA의 크기가 0.5인치인 것에 비해 약 6배 이상 큰 것이다. 이러한 추세에 맞추어 올해 출시될 상업용 기기들은 900개 이상의 핀 개수를 가질 것으로 예상된다. 따라서, BGA 패키지와 BGA 패키지에 묻힌 핀에 대해 심각히 고려해야 한다.

BGA 패키지의 제조 기술은 BGA 패키지를 포함한 보드를 테스트하고 검증할 수 있도록 매우 빠르게 발전해왔다. 예를 들어, JTAG이라는 경계-스캔 표준을 사용하면 모든 핀의 연결 상태를 확인할 수 있기 때문에 핀 개수가 매우 많은 ASIC에서도 네 개의 JTAG 테스트 액세스 포트와 경계-스캔 기능을 사용하고 있다. 연결 상태의 강도를 검증하기 위해 다양한 X-선 기법도 사용한다. 핀을 육안으로 볼 수 없다고 하더라도 X-선을 통해 연결상태를 확인하는 방법이다(놀랍게도 정확도가 매우 높다).

칩을 검증된 설계에 의해 완전히 테스트했다고 하더라도 이것을 이용할 보드를 제작하는 것은 또 다른 문제이며 프로세서의 디버깅과도 다르다. "시스템 디버깅"은 제품 개발 스케줄에서 가장 많은 부분을 차지하기 때문에 많은 신호들이 BGA 패키지에 묻혀 있는 경우라면 디버깅은 보다 어렵고 시간이 많이 소비되는 작업이 된다.

기존의 디버깅 툴

디지털 하드웨어를 디버깅하기 위한 가장 일반적인 툴은 오실로스코프와 로직 애널라이저이다. 요즘 젊은 엔지니어들이 사용하는 현대적 오실로스코프에는 많은 기능이 포함되어 있긴 하지만 50년 전에 사용되는 오실로스코프와 그 방식이 비슷하다. 스코프와 애널라이저

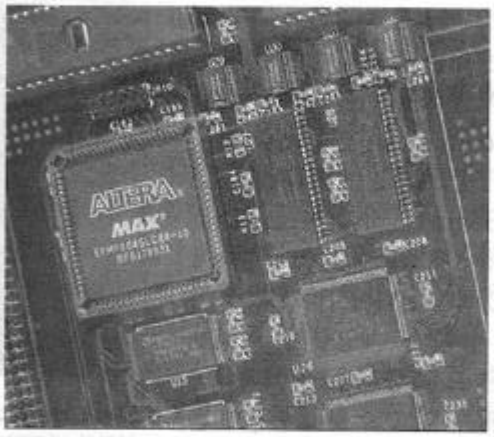
는 모두 노출된 금속면으로 원하는 신호를 액세스 할 수 있어야 사용할 수 있다. 하지만 칩의 집적 기술이 발전하면서 보다 많은 신호들이 패키지의 내부로 숨어들었으며 BGA 패키지의 출현으로 패키지의 핀마저 원하는 대로 찍어 볼 수 없게 되었다.

칩의 핀은 엔지니어가 시스템을 디버깅할 때 칩의 동작을 확인할 수 있는 주요 통로이다. SSI 로직에서 문제의 원인을 찾아내려면 오실로스코프를 이용해 로직을 한 단계 한 단계씩 역추적해 나가야 한다. 게이트의 개수가 아무리 많은 칩이라도 대부분의 디버깅 작업은 입·출력 핀을 찍어서 확인하게 되는데 BGA 패키지에서는 말 그대로 찍어 볼 핀이 없는 것이다. 몇몇 기기에서는 핀이 pc 보드의 테스트 지점에 연결될 수도 있지만 이것은 커다란 BGA 기기 중 작은 일부분에만 해당된다. 따라서 테스트 포인트를 유용하게 이용할 수는 있지만 기존의 스코프나 로직 애널라이저로 이 작업을 수행하려면 엄청난 시간이 소요될 것이다. 테스트 포인트는 보통 로직 설계자가 기대하는 신호를 나타내는 것으로 버그는 그 성격상 불가피하게 기대하지 않은 지점에서 발생한다. 따라서 한 손에 보드를 잡고 다른 손에 스코프를 대고 신호를 찍어 보아도 버그의 원인을 찾기는 거의 불가능하다.

많은 핀을 사용하는 패키지는 보통 ASIC, PLD, 프로세서의 세 가지 형태의 기기에 사용된다. 이 중 PLD의 디버깅이 많은 ASIC 설계 검증 과정의 한 부분으로 프로그램 로직에 프로토타입을 만들고 테스트하기 때문에 ASIC의 검증과 같아지고 있다.

프로세서 디버깅

마이크로프로세서에 기반한 시스템은 수행하는 소프트웨어를 디버깅하기 위해 많은 기술들이 사용되어 왔다. 초기 디버거는 프로세서의 I/O 핀이 트리거 되면 시스템을 "멈추고" 프로그램의 수행을 중지시켰다. 이렇게 하면 디버거가 그 명령어의 주소 값에서 명령어 패치 동작을 위한 버스를 확인하여 프로그램 명령어에서 문제가 된 지점을 찾아낼 수 있다. 그러나 이러한 기법은 데이터와 명령어 메모리가 내장 캐시로 옮겨질 때는 무용지물이 된다. 이렇게 되면 현재 어떤 명령어가 프로세서에서 수행되고 있는지 I/O 핀에서 디버거를 신뢰할 수 없게 된다. 반면, 요즘의 프로세서에서는 시스템



많은 핀을 갖는 BGA 패키지가 사용되고 있다. 올해 출시될 몇몇 칩은 900개 이상의 핀을 갖게 될 것이다. 따라서 내장 디버깅 기능은 시스템의 동작을 효율적으로 확인할 수 있는 유일한 방법이다.

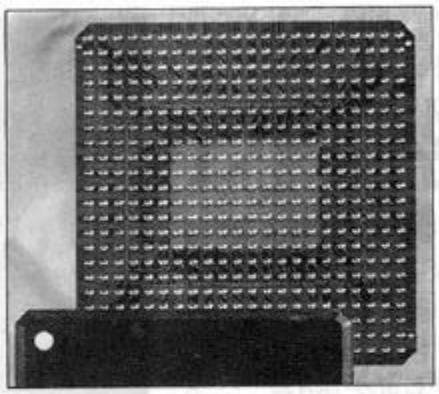
의 내부 상태를 확인하기 위해 "백-도어(back-door)" 방식을 사용한다. 이것의 한 예로 Motorola의 BDM (Background Debug Mode)이 있으며 여기서는 프로세서의 동작을 확인하기 위해 내부의 정교한 디버깅 레지스터와 로직을 사용한다. 즉, 내장된 디버깅 로직은 편리한 백 도어(JTAG 테스트-액세스 포트)를 통해 구성하고 확인할 수 있다. JTAG 포트를 통하여 액세스된 정교한 내장 로직을 사용하는 디버깅 마이크로프로세서 시스템은 현재 PLD에서 사용되는 기술의 한 선례이다.

ASIC 디버깅

ASIC에서는 그 근본적인 특성상 칩이 완전히 제작된 후에는 디버깅이 거의 불가능하다. ASIC 로직 설계는 실리콘이 만들어지기 전에 수없이 많은 시뮬레이션(혹은 다른 검증 방법)을 통해 디버깅을 수행한다. ASIC 설계에서 시뮬레이션은 시간과 비용이 많이 필요한 작업이다. 다양한 상황을 나타낼 수 있는 테스트 벤치를 작성하고 적절한 개수의 벡터를 통해 확인하는 작업은 ASIC 로직 설계 자체를 위해 시간이 많이 소요된다. 하지만 시뮬레이션/검증을 제대로 하지 못했을 경우 설계 뒷단계에 입게 되는 손실은 더욱 엄청나다. 실리콘에 구현된 ASIC을 디버깅하는 기술은 보다 값이 비싸고 시간도 오래 걸리며 일단 모든 버그를 발견했다고 하더

라도 그 에러를 수정하여 다시 제작하는 데는 수 개월이 걸린다.

에러 분석을 전문으로 하는 업체(Accurel in Sunnyvale, Calif. 등)에서는 FIB(Focused Ion Beam), voltage-contrast E-beam과 같은 기술을 포함하여 ASIC 설계를 디버깅하기 위한 기술적 서비스를 제공한다. 이러한 기술은 보통 한정된 몇 가지에만 적용할 수 있고 그 준비 과정도 간단하지 않다(패키지를 모두 제거하여 다이를 노출시켜야 한다). 그러나 이 기술을 이용하면 기기가 제 속도로 동작하면서 칩의 내부 노드를 확인할 수 있다. 이 장비는 그 기능이 놀랄만 하지만 보통 칩 버그의 원인을 도저히 알 수 없을 때 최후의 수단으로만 주로 이용된다. 공정과 제조상의 결함을 분석하는 데는 칩 외부의 입·출력 포트를 이용한 방법과 "칩 수술" 기법이 보다 자주 사용된다.



FineLine BGA 패키지는 전형적인 BGA 패키지와 비교해 같은 면적에 두 배 이상의 핀을 넣을 수 있지만 디버깅이 어렵고 개발에 시간이 많이 소요된다.

PLD 디버깅

PLD를 이용하면 설계자들은 정확한 시뮬레이션과 칩 제작에 드는 비용 없이 복잡한 로직을 시스템에 넣어 동작을 확인해 볼 수 있다. 구현을 비교적 빠른 시일 내에 할 수 있다는 PLD의 이러한 특성은 매우 큰 장점이다. 극단적인 예로, 설계자는 첫 번째 드래프트만 완성되면 로직 설계를 PLD로 구현할 수 있다. 또한 필요하면 버그를 수정하기 위해 원하는 만큼 설계를 변경할 수도 있다. 하지만 설계자는 반드시 하드웨어의 디버깅 시간을 시뮬레이션에 필요한 시간과 비교해 생각해 보아야 하고 당연히, 하드웨어 디버깅이 시뮬레이션을 통한 검증보다 빠를 때만 이 방법을 택해야 한다. 하드웨어 디버깅은 몇 가지 장점을 가지고 있지만 그만큼 위험한 요소도 많이 가지고 있다. 장점에는 다음과 같은 사항이 포함된다.

· **신뢰도** : 로직을 실제 수행 속도로 실행시켜 볼 수 있기 때문에 매우 정교한 시뮬레이션 테스트 벤치에서도 간과될 수 있는 실제 시스템 조건을 모두 확인해 볼 수 있다.

· **시스템의 속도** : 실제 하드웨어를 수행시키는 클록 사이클이 같은 시간동안 시뮬레이션을 통해 할 수 있는 양보다 많다.

하지만 하드웨어 디버깅의 큰 문제는 관측성이다. 시뮬레이션에서는 로직의 모든 신호와 상태, 조건을 확인하고 따라가기가 용이하지만 하드웨어 디버깅에서는 특별한 측정법을 이용하지 않으면 신호를 추적하여 확인하기가 쉽지 않다. 수많은 신호들이 베일에 싸여 있어 그 내부를 확인하기가 거의 불가능하다. 이 문제는 BGA 하드웨어에서 더욱 심각한데, 여기서는 I/O 핀도 액세스할 수 없다.

초기부터 PLD 제조사들은 관측성이 빠른 디버깅을 핵심적 요소로 인식해 왔다. 예를 들면, 1988년에 발표된 Actel (Sunnyvale, Calif.)의 첫 번째 제품인 ACT-1에는 패키지의 I/O 핀을 "찍어" 볼 수 있도록 수행 중에 내부 노드를 연결하는 관측성을 높이기 위한 정교한 기술이 포함되어 있다. 후에 Actel은 이 기능을 완전한 분석 툴인 Silicon Explorer에 확장하였다. Silicon Explorer는 실시간으로 두 개의 내부 노드(두 개의 프로브 핀 사용)와 16개의 외부 I/O에서 데이터를 모은다. 또한 Silicon Explorer에는 인식용 툴과 로직 애널라이저처럼 파형으로 신호를 관찰하는 데 필요한 툴을 조절하기 위해 PC에 기반한 소프트웨어가 포함되어 있다. 소프트웨어를 이용해 문자 그대로 스코프나 로직 애널라이저를 내부의 한 노드에 연결시킬 수 있으면 가장 일반적인 의문인 "이 신호는 어디에 사용되는 거지?"라는 질문에 쉽게 답할 수 있게 된다. 그렇게 되면 웬만한 버그를 해결할 수 있는 중요한 단서를 발견할 수 있다.

지금까지 디버깅 과정에서 내부 노드를 확인하는 데는 패키지의 핀만을 사용해 왔으며 또한 한 노드보다는 보다 많은 비트(예, 멀티 비트 내부 레지스터, 버스)를 한꺼번에 확인하는 것이 훨씬 편리한 경우가 많다. 원칙적으로 디바이스의 I/O 핀 중 상당 부분을 디버그 포트로 할당할 수 있다. 하지만 이것은 BGA를 사용하고자 하는 기본적인 목적에 반하는 것으로 디버깅을 쉽게 하

기 위해 핀의 개수를 늘리면 시스템의 비용이 그만큼 커지고 기존의 I/O 자원을 많이 소비하게 된다.

멀티 비트 신호에 대한 디버깅 문제의 한 가지 해법은 몇 개의 핀을 통해 칩 외부에서 직렬로 많은 비트를 스캔하는 것이다. 대부분의 RAM 기반 PLD는 직렬로 프로그램되고 몇 개의 포트로 검증되기 때문에 많은 메커니즘에서 이미 디버깅을 위해 칩의 내부나 외부에서 멀티 비트 신호를 스캔하는 기법을 사용하고 있다. 최근의 많은 PLD에서도 경계-스캔 테스트에서 사용되는 같은 규격의 4핀 JTAG 테스트 액세스 포트를 적용하고 있다. 디버깅 액세스에 같은 규격의 JTAG 포트를 사용하면 관측성을 위해 필요한 I/O의 비용을 거의 0으로 만들 수 있다. JTAG 테스트 액세스 포트는 디바이스의 구성과 디버깅 과정의 관측성 모두를 위해 많은 백도어를 마련한다.

하지만 직렬 스캔 방식에도 한 가지 큰 단점이 있다. 여기서 PLD의 내부 상태는 선택된 멀티 비트 신호가 몇 개의 외부 포트를 통해 천천히 나가고 있을 때 변하지 않아야 한다. 따라서 칩의 내부 상태의 한 단면을 확인하는 것은 가능하지만 실제로 동작하는 동안 회로 동작의 흐름은 확인할 수 없다. 네트워크 데이터 전송 등과 같은 많은 애플리케이션에서 로직 디바이스의 클럭을 멈추면 동작이 전혀 진행되지 않도록 효율적으로 시스템을 죽이게 된다. 이러한 시스템에서 내부 신호를 스캔하기 위해 시스템을 멈추면 결과 값이 필연적으로 바뀌게 된다.

Actel의 프로브 핀은 제 속도로 동작되는 몇 개의 내부 신호를 볼 수 있도록 고속의 "디버깅 홀"을 지원하였다. 직렬 스캔 방식으로 멀티 비트 신호를 확인은 할 수 있지만 이것은 반드시 시스템을 멈춘 상태여야만 한다. 어떤 경우에는 시스템을 멈추는 것이 불가능할 수도 있으나, 문제는 시스템을 멈추지 않고 실시간으로 멀티 비트 신호 벡터를 확인할 수 있는 디버깅 기법을 찾는 것이다.

실시간으로 멀티 비트(여러 개의) 신호를 디버깅하기 위해서는 데이터를 빠르게 잡아내기만 하면 된다. 즉 데이터를 칩 외부로 끌어내는 작업까지 빠르게 이루어져야 할 필요는 없다. 한 가지 해법은 실시간으로 칩 내부의 버퍼에 적당한 개수의 샘플을 넣어 두고 후에 확인과 분석을 통해 로그 버퍼를 직렬로 (천천히) 스캔하는 것이다. 버퍼에 신호를 저장하려면 칩 내부에 RAM이 필요하다. Altera의 Apex-20K 디바이스 같은 새로운 PLD에는 내부 RAM을 재구성할 수 있는 블록이 포함되어 있다. Altera의 SignalTap에서는 칩에 내장된 로직 애널라이저가 버퍼를 인식할 때 RAM 블록을 사용한다. 내장된 로직 애널라이저는 멀티 비트 버스나 다른 여러 신호를 포함한 어떤 내부 노드에도 연결될 수 있다. Altera의 개발용 시스템(Quartus)에서 동작하는 소프트웨어를 이용하면 입력 신호를 정하고 조건과 내장된 애널라이저를 위한 버퍼의 크기를 조절할 수 있다. 내장된 로직 애널라이저를 조절하면 완전한 속도로 시스템의 동작에 영향을 미치지 않고 내부 신호를 RAM 버퍼에 저장할 수 있다. 이벤트를 잡아낸 후에 버퍼에 저장된 데이터는 JTAG 테스트 액세스 포트를 통해 스캔되고 호스트 컴퓨터에 파형으로 표시된다.

내장형 로직 애널라이저를 BGA 디바이스에 사용하면 "어디를 찍어 보아야 할 것인가"라는 질문을 효율적으로 해결할 수 있다. 로직 애널라이저의 입력은 소프트웨어로 연결하고 재구성 가능한 배열의 라우팅 자원으로 구현되기 때문에 디바이스의 모든 노드를 액세스할 수 있다.

요즈음 디바이스의 패키지에서 내장 디버깅 기능은 더 이상 단순한 선택 사항이나 편리한 기능에 머물지 않는다. 내장 디버깅 기능은 시스템의 동작을 효율적으로 확인하기 위해 이용할 수 있는 유일한 방법이다. 현대의 PLD에 일반적으로 구비되어 있는 재구성 가능한 라우팅 자원과 RAM을 사용하여 기존의 로직 애널라이저 기능을 칩 내부에 구현하는 것은 바람직할 뿐만 아니라 그리 어렵지도 않다. 이 기술은 BGA 패키지와 고밀도의 하드웨어에서 발생하는 디버깅 문제에 대한 해결책일 뿐만 아니라 디버깅과 시뮬레이션 사이의 "관측성 차이"를 줄이는 방법이다. 복잡한 PLD의 디버깅을 보다 쉽게 처리하면 프로그램 로직을 탑재한 시스템이나 프로토타입과 검증에 프로그램 로직을 사용한 ASIC 기반 시스템 모두의 개발 시간을 단축시킬 수 있다.

게재된 기사는 본지의 웹사이트를 통해서도 보실 수 있습니다.
http: //www. chomdan. co. kr