보다 간단한

설계 검증을 위한 동반자

Tektronix

Enabling Innovation

목차

1. 소개 및 개요 ----------3
1.1 설계 과정 개요----------3
1.2 설계 프로젝트 개요 5
2. 설계 단계----------6
2.1 효과적인 개발 과정 ----------6
2.2 잠재 문제 영역 식별----------7
2.3 디버그 및 검증을 위한 설계 ----------9
2.3.1 텍트로닉스 설계를 실례로 사용 ----------10
2.3.2 프로빙 결정 사항 ----------12
2.3.3 프로빙 요약 ----------13
2.4 요약----------14

3. 디버그 및 검증 단계 ----------14
3.1 초기 파워 ON ----------15
3.2 기본 기능 검증----------15
3.2.1 전원 장치 분석 ----------17
3.2.1.1 전원 장치 스위칭 손실----------17
3.2.1.2 전원 장치 리플 ----------18
3.2.2 기본 기능 검증----------19
3.2.2.1 마이크로프로세서 재설정 디버그 ----------19
3.2.2.2 마이크로프로세서 부트 디버그 ----------19
3.2.3 요약----------19
3.3 확장 기능 검증----------20
3.3.1 통합 솔루션 사용----------20
3.3.2 신호 무결성 문제를 신속히 탐지하는 지름길 ----------27
3.3.3 요약----------29
3.4 하드웨어/소프트웨어 통합----------30
3.4.1 마이크로프로세서 부트 코드 디버깅 ----------30
3.4.1.1 로직 분석기 소스 코드 창 사용----------32
3.4.2 요약----------32
3.5 특성화----------33
3.5.1 설계자 및 최종 사용자를 위한 사양 ----------33
3.5.2 Setup 및 Hold 테스트----------34
3.5.3 요약----------35
3.6 시스템 테스트 및 최적화 ----------36
3.6.1 요약 ----------37

4. 요약 및 결론----------38
4.1 설계 단계 ----------38
4.2 디버그 및 검증 단계 ----------38
4.3 결론 ----------38

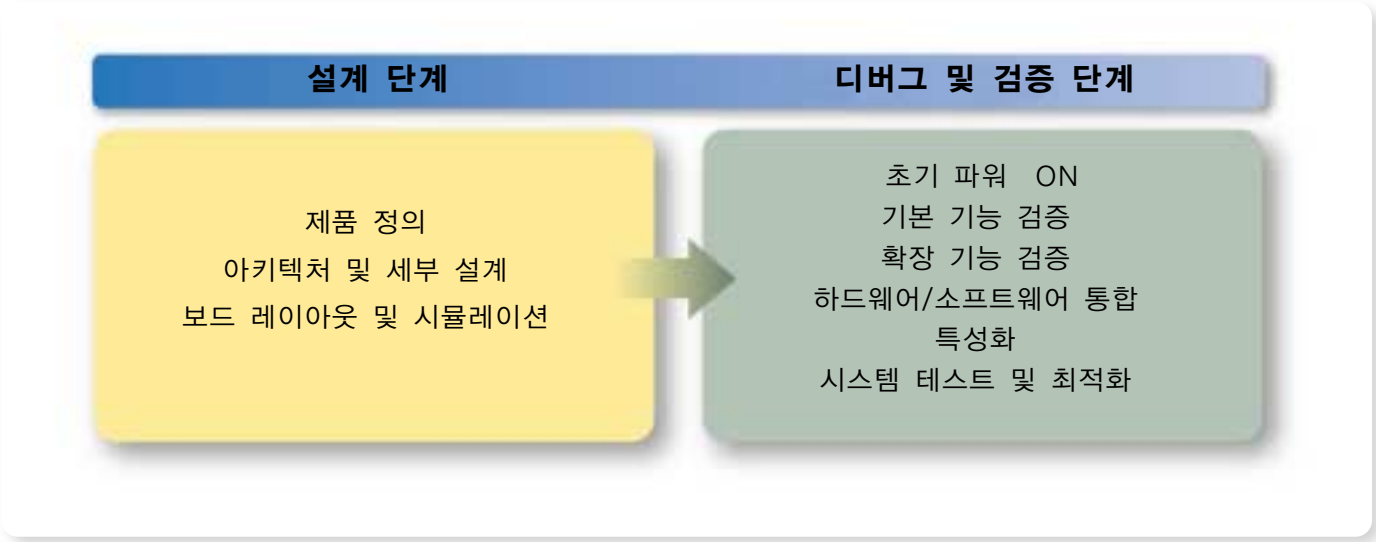


그림1 – *설계 과정 개요*

1 소개 및 개요

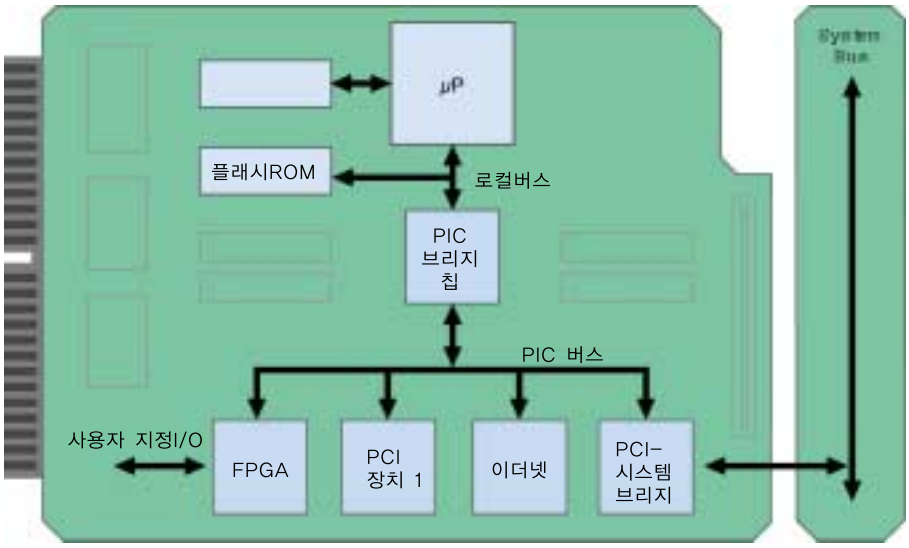
본 입문서는 시간상의 제약과 비용 한도라는 현실적인 조건 때문에 오늘날의 첨단 디지털 시스템에 대한 디버그 및 검증 작업을 합리적으로 수행하는 방법을 배우는 데 관심이 있는 기술 전문가를 위해 꾸며졌습니다.

설계 툴과 EDA(electronic design automation, 전자 설계 자동화) 소프트웨어의 발달로 설계 팀은 복잡한 설계 업무를 효과적으로 진행하는 동시에 설계 소요 시간을 이전과 동일하게 유지하거나 단축할 수 있게 되었습니다. 오늘날과 같이 개발 일정이 빠듯하고 경쟁이 치열한 환경에서 생존하려면 디버그 및 검증 단계에서도 이와 유사한 생산성 증대 효과를 거두어야 합니다. 본 입문서에서는 설계 검증 시 보다 생산적이고 발생한 문제의 디버깅 시 보다 효율적으로 업무를 진행할 수 있도록 돕는 이슈와 기술에 초점을 맞춥니다. 그 개념을 도식화하기 위해 마이크로프로세서 기반의 신형 임베디드 시스템 개발 과정의 개념 설계에서 완제품 제작까지 차례로 짚어 보겠습니다.

1.1 설계 과정 개요

그림 1은 매우 개괄적인 수준에서 시스템을 출시하기까지 필요한 일반적인 단계를 나타낸 것입니다. 설계 단계에서는 개념을 만들어 내고 대안에 가중치를 두어 평가하여 최종 설계의 밑그림을 포착합니다. 디버그 및 검증 단계에서는 설계의 정확성을 검증하고 기능과 신뢰성의 두 가지 측면에서 발견된 문제를 교정하고 설계한 대로 신뢰성 있게 생산 가능한지 평가합니다.

설계 단계 – 설계 단계는 여러 개의 세부 단계로 구성되며, 각 세부 단계에는 일련의 작업과 그 작업을 완료하기 위해 필요한 툴이 주어집니다. 예컨대 시스템 시뮬레이션 툴을 이용하여 제품의 아키텍처를 결정하고, 열 분석 툴과 정교한 모델링 프로그램의 도움을 받아 세부적인 기계 설계 작업을 수행하고, RTL(register transfer language) 코드로 설계 의도를 캡처하고, 도식화된 캡처 소프트웨어로 보드 설계 작업을 수행합니다.



► **그림2 –** *임베디드 시스템 설계 예*

설계 개념이 다듬어지고 추상적인 개념에서 세부 설계로 이어지는 단계를 거쳐 이동하면서 그것이 세부적인 계획을 통하거나 기본적으로 여러 가지 결정을 내리게 되며, 이런 결정으로 인해 설계 검증 소요 시간이 단축되거나 늘어나게 됩니다.

설계 검증 및 디버그 과정의 각 단계에서 어떻게 결정하느냐를 이해하는 것이 매우 중요합니다. 이런 이해를 바탕으로 문제를 조기에 발견하고 재작업을 최소화함으로써 개발 시간을 단축하고 신제품을 정해진 시간 내에 출시할 수 있기 때문입니다. 개발 중인 제품이 출시될 시장에서 경쟁이 매우 치열한 경우에는 수익의 대부분이 가장 빨리 최고의 성능을 제공하는 믿을 만한 제품에게 돌아가게 됩니다.

디버그 및 검증 단계 – 엔지니어로서 가장 희열을 느낄 수 있고 보람이 있는 시간은 새로 개발한 보드를 실험실에서 최초로 작동시키는 순간이며, 며칠되지 않는 그 짧은 시간 동안 시스템의 기능을 하나하나 테스트하면서 마치 시스템에 생명을 불어넣는 듯한 감동을 느낄 수 있습니다. 그런 흥분과 집중적 생산성을 전체 디버그 과정 내내 유지하려면 세심한 계획을 통해 생산성을 저해하는 요소들을 피해야 합니다. 가장 흔히 만나게 되는 문제는 테스트 과정을 문서화하는 작업인데, 이 작업을 철저히 하고 첨단 테스트 툴을 잘 이해하고 있으면 즐거운 흥분이 좌절감으로 바뀌지 않을 것입니다.

1.2 설계 프로젝트 개요

여기서 예로 드는 설계는 통신 인프라 장비, 프린터 및 비디오 장비와 같은 수많은 그리고 복잡한 시스템에서 흔히 볼 수 있는 프로세서 기반 임베디드 시스템입니다. 그림 2에서 나타낸 바와 같이, 이 예에 사용되는 설계는 프로세서, 메모리, 프로세서와 주변장치를 연결하는 내장 통신 버스, I/O 주변장치로 구성됩니다. 비용 요구 사항을 만족시키기 위해 시스템에는 SDRAM, PCI 및 이더넷 인터페이스와 같이 이미 검증되고 널리 사용되는 기술을 내장할 것입니다.

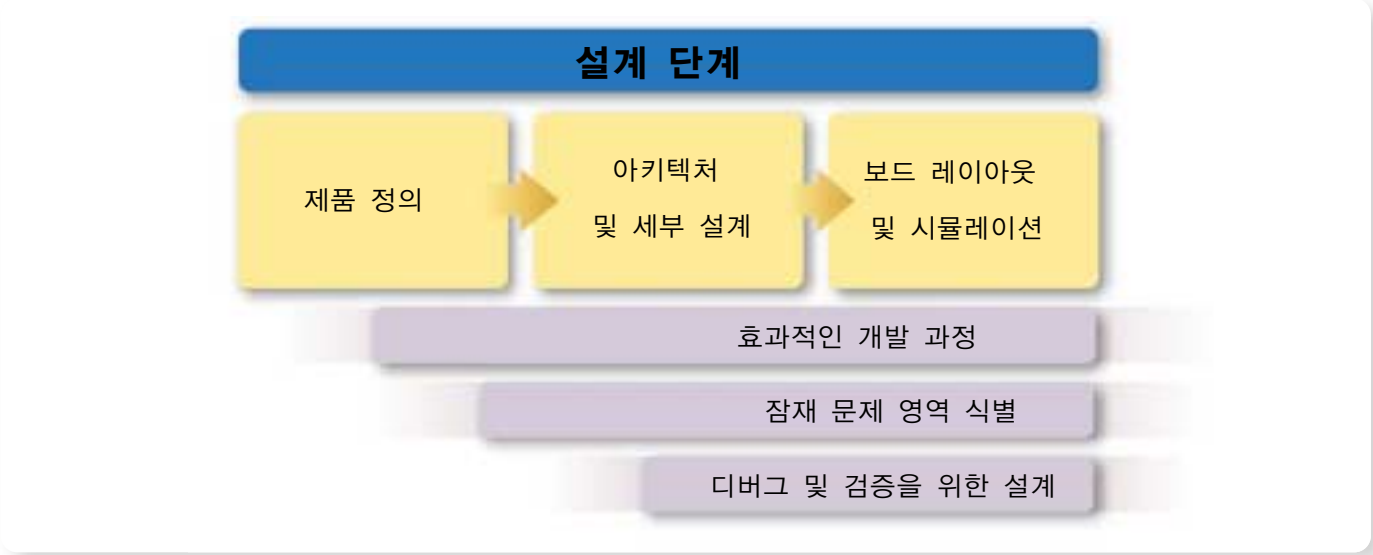
고도로 집적된 프로세서는 166MHz의 속도로 실행되는 SDRAM을 지원하는 내장 메모리 컨트롤러를 제공함으로써 비용을 줄이는 데 일조합니다. 이 프로세서는 필요한 메모리 대역폭을 제공하고 신호 무결성에 대한 염려를 불러일으키기에 충분할 정도로 빠릅니다.

보드 상에서 칩간 버스로는 PCI를 사용합니다.

새로운 PCI Express 표준에 따른 추가 성능과 비용이 필요치 않습니다. 요구되는 성능 목표에 도달하기 위해서 66MHz, 64비트로 구현된 PCI을 사용합니다. 이 로컬 PCI 버스는 시스템 버스와의 연결 고리 역할을 하는 FPGA를 사용하여 시스템 버스와는 따로 떨어져 배치됩니다.

이더넷 포트는 최종 사용자에 이르는 통신 경로와 디버그 단계 중에 사용할 디버그 포트 기능을 모두 제공합니다. 보다 효율적이고 경제적인 10/100 이더넷을 위해 최근에 등장한 기가비트 이더넷 구현 방식을 채용할 것입니다.

이 설계에는 일부 최첨단 기술은 내장되지 않지만 본 신개발 보드의 모든 기능을 검증해야 하고 버그를 찾아서 수정해야 하며 설계 특성화도 수행해야 합니다. 이를 위해서는 유연하면서도 사용하기 쉬운 실시간 오실로스코프, 로직 애널라이저, 그리고 전체 정보를 통찰할 수 있는 기능과 시스템 분석 기능을 제공하는 시스템으로 이 두 계측기를 효과적으로 통합하는 기능이 필요합니다.



► **그림3 –**

2 설계 단계

그림 3에 나타낸 바와 같이, 설계 단계에서는 추상적 개념을 이끌어내고 왼쪽에서 오른쪽으로 설계 흐름으로서 개념을 따라가도록 점진적으로 내용을 다듬습니다. 각 단계에는 보다 세부적인 내용이 추가됩니다. 즉 아키텍처 블록을 정의하고 개별 블록을 세부적으로 설계하고 회로 기판의 레이아웃을 정합니다.

각 단계에서는 전체 생산성에 영향을 주는 결정들이 이루어집니다. 편리하게 신호에 액세스할 수 있는 기능을 제공한다든지, 소프트웨어 디버그 툴에 대한 연결 기능 등과 같이 어떤 것은 명백합니다. 하지만 기술 선정, 구성 요소 선택, 기계적 패키징 개념과 같이 그다지 명백하지 않은 문제도 있습니다. 최고 생산성은 다음과 같이 얻을 수 있습니다.

- 효과적인 개발 과정 마련
- 문제 발생 가능성이 있는 곳 파악
- 디버그 및 검증을 위한 설계

2.1 효과적인 개발 과정

모든 제품 개발 팀에는 저마다 문서화된 개발 과정이 있습니다. 이 과정에서는 해당 제품이 고객의 요구를 만족시킬 것이라는 확신을 제고하기 위해 필요한 점검 사항과 보호 대책을 정의합니다.

고객 요구 사항을 캡처하고 검증하는 메커니즘을 정의할 필요가 있습니다. 제품 단가와 개발 비용을 철저히 검토함으로써 원하는 투자수익률을 얻을 수 있을지 확인해야 합니다.

아키텍처 검토, 설계 검토 및 코드 검토를 위한 과정에서 설계자는 팀이 엔지니어링 모범 사례에 따를 수 있도록 고안된 질문을 해야 합니다.

개발 과정을 고수하는 것이 처음에는 시간적, 재정적 비용을 늘려 생산성을 저해하는 것처럼 보일 수 있습니다. 하지만 실제로 잘 정의되고 유연성 있는 개발 과정은 최초 설계로 고객의 요구를 정확히 캡처할 확률을 높이고, 아키텍처를 명확하게 정의하게 하고, 운영 체제를 제작하는 데 필요한 보드 수와 소프트웨어 릴리스 수를 줄임으로써 생산성을 대폭 개선하는 효과가 있습니다.

2.2 잠재 문제 영역 식별

미래를 예견하여 문제의 해답을 찾아낼 수만 있다면 얼마나 좋겠습니까!

하지만 현실은 그렇지 않아 설계자가 설계 디버그 중에 문제가 어디서 발생할지 아는 경우는 거의 없습니다. 과거의 실수로부터 배운 경험이 이런 어려운 상황을 타개하는 데 도움이 됩니다. 통찰력이란 이런저런 설계를 해보는 과정에서 점차 얻어지는 것입니다. 하지만 이번이 처음으로 해보는 설계 작업이라면 어떻게 해야 할까요? 혹은 신호 무결성이 문제가 되는 수준의 빠른 클럭 주파수나 에지 속도가 포함되는 설계 작업을 처음으로 해보는 경우라면 어떻게 해야 할까요? 이런 경우 어떤 문제를 예상해야 할까요?

설계 작업에서 맞닥뜨리게 되는 문제의 유형은 크게 두 가지로 바로 기능 문제와 신호 무결성과 관련된 문제입니다.

기능 문제 – 기능 문제는 구입한 소자의 작동에 대한 오해, FPGA에서 구현된 지적재산 RTL 수준에 있어서의 오류 또는 부정확한 하드웨어/소프트웨어 상호 작용 등 여러 가지 다양한 이유로 인해 발생합니다.

효과적인 개발 과정을 갖춘 팀은 설계 검토와 보드 수준의 시뮬레이션 작업 중에 수많은 기능 문제를 잡아냅니다. 양방향 버퍼 방향 제어 신호에 대해 반전된 로직 레벨과 같은 일반적인 문제를 포착하는 것뿐만 아니라, 배전과 클럭 분포를 검사하는 데에는 도면 설계 검토가 매우 유용합니다. 하지만 도면 검토만을 통해서는 모두 발견하기 어려울뿐만 아니라 검토자가 아주 부지런하고 꼼꼼해야만 파악할 수 있는 문제도 많이 있습니다.

게이트 수가 1백만 개 이상인 효과적인 설계 툴로서 출시기간을 앞당길 수 있는 이점을 지닌 FPGA가 첨단 시스템에 많이 이용되며 이 툴은 보기보다 훨씬 더 많은 기능이 구현되어 있습니다.

설계 툴의 발달로 보다 개괄적인 수준에서 추상 설계 작업을 수행하고, 복잡한 설계를 보다 빨리 통합하고, 짧은 시간 내에 주기를 배치 및 라우팅하는 작업을 완료할 수 있습니다. 이와는 대조적으로, 테스트 벤치 설계, 자극 모델 작성 및 테스트 사례 관리는 생산성에 대한 제한 요소로 볼 수 있습니다. 실제로는 설계 단계에서 발견된 문제와 버그를 수정하기가 더 쉽고 비용도 적게 듭니다. 가능한 한 일찍 발견 가능한 문제를 더 많이 발견할수록 디버그 시간과 개발 비용을 줄일 수 있습니다.

하지만 시뮬레이션에는 한계가 있습니다. 예를 들어 클럭 경계를 통과하는 신호의 동기화와 관련된 문제는 찾아내기 힘들고, 테스트 사례가 불완전한 경우가 종종 있고, 복잡한 하드웨어/소프트웨어 상호 작용은 모니터링 하기가 어렵습니다.

신호 무결성 문제 – 신호 무결성의 개념은 아날로그 영역에서 신호를 손상시킬 수 있는 노이즈, 왜곡 및 변형에 관계된 것입니다. 신호 경로 설계, 임피던스 및 로딩, 송신 라인 효과, 심지어는 회로 기판 상의 배전 등 많은 변수가 신호 무결성에 영향을 미칠 수 있습니다. 처음부터 그런 문제를 최소화하고 문제가 나타날 때 이를 교정하는 것이 설계자의 책임입니다.

신호 품질이 저하되는 데는 두 가지 근본적인 원인이 있습니다.

- 디지털 문제 – 일반적으로 타이밍과 관련된 문제입니다. 버스 경합, Setup 및 Hold 위반, 불안정성 및 경합 조건이 버스 또는 장치 출력에서 발생하는 이상 신호 동작의 원인일 수 있습니다.
- 아날로그 문제 – 진폭이 낮은 신호, 느리거나 빠른 변환 시간, 글리치, 오버슈트, Crosstalk 및 노이즈와 같은 현상들은 회로 기판 설계나 신호 종단에 그 원인이 있지만 다른 원인도 있습니다.

디지털 및 아날로그 신호 무결성 문제들 사이에는 상호 작용과 상호 의존 정도가 높습니다. 예를 들어 게이트 입력에서의 느린 상승 시간으로 인해 출력 펄스가 지연되고, 따라서 디지털 환경에서의 버스 경합이 다운스트림 쪽으로 더욱 치우치는 원인이 됩니다. 신호 무결성 계측과 문제해결을 위한 철저한 해결책은 디지털 및 아날로그 툴이 모두 관련됩니다.

경험이 풍부한 엔지니어는 설계 과정 중에 늘 신중함을 유지한 결과로서 신호 무결성을 달성할 수 있다는 점을 잘 압니다. 신호 무결성 문제는 설계가 전개되어 나감에 따라 복잡한 양상을 띠고 추적하기 어려워지기 십상입니다. 첫 프로토타입 보드에서는 눈에 띄지 않는 작은 착오가 보드를 다른 보드와 병합할 경우 전체 시스템이 작동 중단되어버리는 원인이 될 수도 있습니다.

이런 현실에서 신호 무결성은 어디서 시작되는 것일까요? 설계자는 설계 단계의 맨 처음부터 신호 무결성 작업을 시작할 필요가 있습니다. 모든 임베디드 시스템 설계에 공통적인 사항이 몇 가지 있습니다. 첫째, 알맞은 클럭 분포가 매우 중요합니다. 클럭이 생성되어 보드 주변에 분포되는 방식이 전자기 간섭(EMI)에서부터 타이밍 요구 사항을 충족하는 마진(또는 부족함)에 이르기까지 영향을 미칩니다. 아키텍처 정의 단계에서 이루어진 결정 사항이 영향을 미칩니다. 소자 선택도 영향을 미칩니다. 보드 전체에 클럭을 분산시키려고 스큐를 제거하기 위해 일반 버퍼 장치 또는 PLL이 내장된 특수 IC를 사용합니까?

신중하게 계획해야 하는 시스템 설계의 마지막 측면은 배전 설계입니다. 배전 설계에는 전원 장치 설계, 국부 전압 조절, 중요한 아날로그 섹션을 위한 무중단 전원 공급 및 회로 기판 구조의 모든 측면을 망라할 필요가 있습니다.

2.3 디버그 및 검증을 위한 설계

설계 작업 중에 만나게 되는 문제의 유형을 잘 이해한 상태에서 검증 및 테스트 계획 수립에 착수할 수 있습니다. 이 계획에서는 다음과 같은 방법을 통해 예측치 못한 상황과 잠재적 장애물을 제거합니다.

- 테스트 대상 기능과 기능이 수행되는 방식 식별
- 검증 필요성이 있는 인터페이스 및 신호 식별
- 수행 필요성이 있는 계측 유형 식별

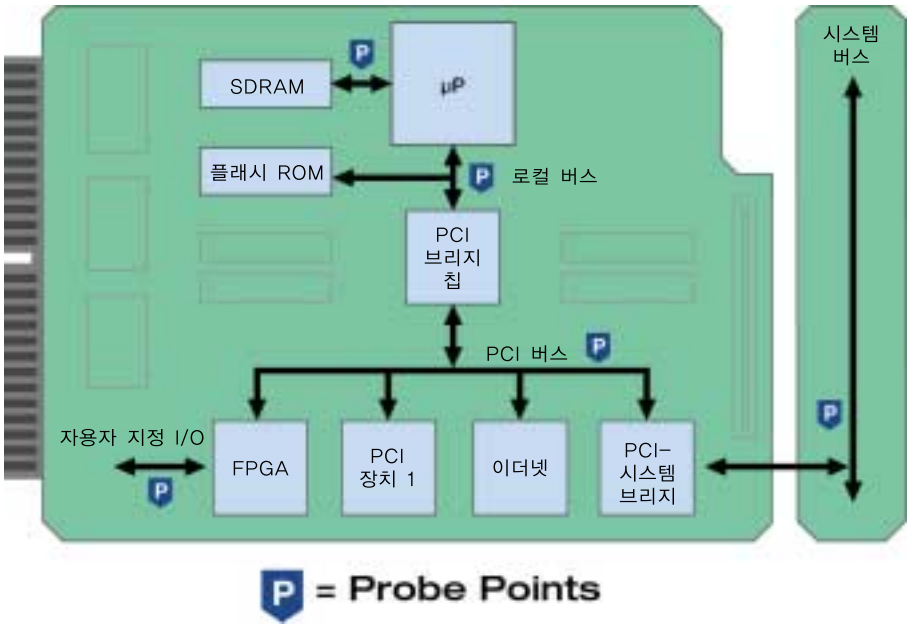
이 계획은 설계 단계 중에 개발되어야 합니다.

최악의 시나리오는 설계 단계에서 디버그 및 검증 필요성을 고려하지 않고 효과적으로 디버깅할 수 있는 능력을 제한하거나 아예 없애버리는 것입니다. 오늘날의 복잡한 설계에 있어 문제해결과 디버그 작업을 할 필요가 없을 정도로 완벽한 엔지니어는 없습니다.

그렇다면 어떻게 하면 설계 디버그 작업을 더 간단하게 할 수 있을까요? 한 가지 방법은 로직 애널라이저와 오실로스코프 모두에 적합한 편리하고 쉬운 프로브 포인트를 제공하는 것입니다. 하지만 이것은 너무 문제를 단순화한 것입니다. 그렇다면 실제로는 어디에 액세스 포인트를 제공해야 할까요? 모든 곳에 액세스 포인트를 배치하는 것이 좋겠지만 대부분의 경우 이는 비실용적입니다. 회로 기판에서 사용할 수 있는 제한된 공간과 디버그시 요구되는 공간을 잘 검토해야만 합니다.

그것은 테스트 연결부를 추가하는 것은 말할 것도 없고 할당된 공간에 필요한 기능만이라도 채워넣기 위한 힘겨운 몸부림이 될 수도 있습니다.

이 충돌을 어떻게 해결할 수 있을까요?



► **그림4 –** *로직 애널라이저 프로브 포인트*

2.3.1 텍트로닉스 설계를 실례로 사용

텍트로닉스 설계를 실례로 사용하여 다음과 같은 일련의 질문을 시작해보겠습니다.

- 마이크로프로세서 가시성이 얼마나 중요합니까? 하드웨어 디버그에 대한 가시성에만 초점을 맞춰야 합니까, 아니면 소프트웨어 디버그도 문제가 됩니까?
- 내장 버스 중 어떤 것을 볼 필요가 있습니까? 오실로스코프 테스트 포인트로 이렇게 할 수 있습니까, 아니면 로직 애널라이저 테스트 액세스도 필요합니까?
- 어디서 설계 마진이 문제가 됩니까? 그것을 어떻게 검증합니까? 온도와 기타 환경 요소에 대해서는 어떻습니까?
- 신호를 보지 못하는 경우에는 어떤 일이 일어납니까? 이것이 개발 일정에는 어떤 영향을 미치게 됩니까? 보드 설계 작업을 다시 해야 하는 것입니까?
- 테스트 포인트와 프로브는 신호와 어떻게 상호 작용합니까? 지나친 용량성 부하 때문에 이들이 회로의 비정상 작동 원인이 됩니까?

출발점으로서 그림 4에 나타낸 것처럼 모든 버스에 대해 로직 분석기 액세스를 제공하는 것이 바람직하겠습니다. 왜 그렇겠습니까?

로컬 버스 – 로컬 버스에 액세스하면 부트 문제를 모니터링하고 디버그할 수 있습니다. 이 곳이 알 수 없는 하드웨어와 소프트웨어가 처음으로 한데 모이는 곳입니다.

SDRAM 인터페이스 – 시스템 데이터 구조와 시스템 코드는 SDRAM에 저장됩니다. 가시성이 있으므로 소프트웨어 실행 상황을 실시간으로 추적해서 볼 수 있습니다. 이 기능 덕분에 소프트웨어 성능 분석뿐만 아니라 하드웨어/소프트웨어 상호 작용의 디버그까지 가능합니다.

PCI 버스 – 성능 목표를 달성한다는 것은 사용 가능한 PCI 대역폭을 현명하게 사용할 수 있도록 함의 뜻합니다.

► **그림5 –** *범용 로직 애널라이저 프로브*

► **그림6 –** *고밀도 로직 애널라이저 프로브*

로컬 PCI 버스에 대한 테스트 액세스가 손쉽고 편리하여 Throughput 문제가 해결될 뿐만 아니라 시스템 내 두 개의 FPGA 기능 디버깅도 가능합니다.

사용자 지정 I/O 인터페이스 – FPGA 하나가 텍트로닉스 제품을 타 경쟁사 제품과 차별화하는 기능을 구현합니다. 시뮬레이션만으로는 이 FPGA에 있는 기능 및 타이밍 관련 문제를 모두 잡아내지 못합니다.

시스템 버스 – 이 전용 버스는 시스템의 핵심입니다. 시스템 버스를 모두 캡처하는 능력으로 시스템 문제를 보다 쉽게 해결합니다.

이들 포인트에 로직 애널라이저를 어떻게 연결할까요?

실제로는 두 가지 옵션이 있습니다. 첫째, 범용 프로브(그림 5 참조)를 사용하여 보드에 연결할 수 있습니다. 이 접근 방식은 비교적 소수의 신호를 검사해야 하는 경우에 편리하지만 PCI와 같은 버스에 이런 프로브 여러 개를 끼워 맞춰야 하는 번잡함과 성능 문제를 고려해봐야 할 것입니다.

둘째, 보드에 커넥터나 액세스 포인트를 직접 추가하는 방법입니다. 이 방법은 범용 프로브를 사용할 때의 번잡함과 성능 문제는 극복하지만 소중한 보드 공간을 차지하게 됩니다. 그림 6은 전형적인 고밀도 프로브와 이 프로브가 회로 기판에 연결되는 방식을 나타낸 것입니다.

► **그림7 –** *로직 애널라이저용 마이크로프로세서 어댑터(사진 제공: Ironwood Electronics)*

► **그림8 –** *로직 애널라이저 프로빙용 DIMM 인터포저(사진 제공: Nexus Technologies)*

2.3.2 프로빙 결정 사항

검증된 인터페이스의 버스 폭으로 인하여 고밀도 로직 애널라이저 프로브를 사용해야 할 때가 있습니다. 여기서 우리는 보드 공간이 충분치 않다는 것을 실감하게 됩니다. 하지만 임의의 가능한 액세스 포인트를 배제하기 전에 모든 대안을 고려해볼 필요가 있습니다. 아마도 보드 공간에는 영향을 주지 않으면서 프로브 작업을 할 방법이 있을 것입니다.

예를 들어 마이크로프로세서용 프로브 어댑터를 사용할 수 있을 것입니다. 이런 종류의 어댑터는 일반적으로 마이크로프로세서가 있어야 할 자리에서 보드와 결합합니다.

마이크로프로세서는 보통 고속 소켓에 배치되고 보드 주변에 로직 애널라이저 테스트 포인트가 추가됩니다. 그림 7은 일반적인 어댑터를 보여줍니다. 어댑터를 사용하면 보도 공간을 보존할 수 있고 중요한 로컬 버스 및 SDRAM 신호에 대한 액세스를 제공합니다. 불행히도 마이크로프로세서에 바로 사용할 수 있는 어댑터가 없으므로 로컬 버스에 액세스하는 유일한 방법은 고밀도 로직 애널라이저 프로브를 사용하는 것입니다.

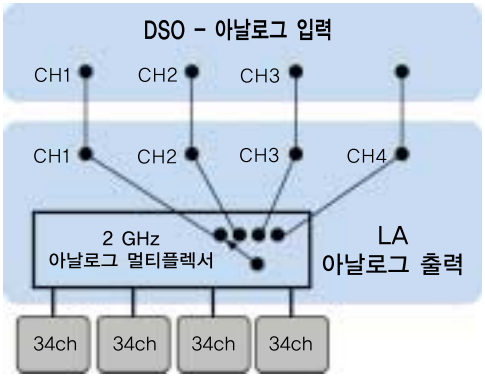
SDRAM 인터페이스에 액세스할 수 있는 두 가지 옵션이 있습니다.

첫째, DIMM 소켓을 기반으로 한 설계의 경우에는 인터포저 카드를 사용할 수 있습니다. 그림 8에 전형적인 인터포저가 나와 있습니다.

► **그림9 –** *로직 애널라이저 프로빙용 PCI 인터포저(사진 제공: Nexus Technologies)*

사용 가능한 DIMM 소켓이 없는 경우에는 보드 설계에 테스트 액세스 포인트를 포함시켜야 합니다. 다행히도 단일 DIMM 소켓을 사용 중이므로 인터포저 보드를 사용하여 SDRAM에 액세스할 수 있습니다.

PCI 버스에 대해 동일한 두 가지 옵션이 있습니다. 그림 9에 전형적인 PCI 인터포저가 나와 있습니다. PCI 버스는 내장 버스이고 PCI 슬롯이 없기 때문에 PCI 인터포저를 사용할 수 없습니다. 내장된 PCI 버스를 보려고 할 경우에는 보드에 테스트 액세스 포인트를 설계해야 합니다.



- 단일 포인트 디지털 및 아날로그 프로빙
- 모든 채널에 2GHz 아날로그 대역폭
- 136개의 로직 애널라이저 입력 채널 중 4개가 아날로그 출력 BNC 4개에 다중화됨
- 아날로그 프로브 출력은 항상 "작동 중" 상태임

► **그림10 –** *iCapture™다중화 툴*

각 버스와 인터페이스를 이와 같은 방법으로 차례대로 검사합니다.

각각에 대해 앞서 개괄한 내용을 자문해볼 필요가 있습니다. 이런 자문에 답한 후 지금 다루고 있는 예제 보드에 대한 디버그 전략을 표 1에 요약합니다.

2.3.3 프로빙 요약

영향을 최소화하기 위한 방법으로 액세스를 제거할 필요가 있는 경우 원하는 액세스 포인트를 수용할만한 보드 공간이 아직도 충분치 않다는 점이 명백해집니다. 다행히도 이 예에서는 위에 개괄한 전략을 충분히 활용하는 데 필요한 보드 공간을 확보하고 있습니다.

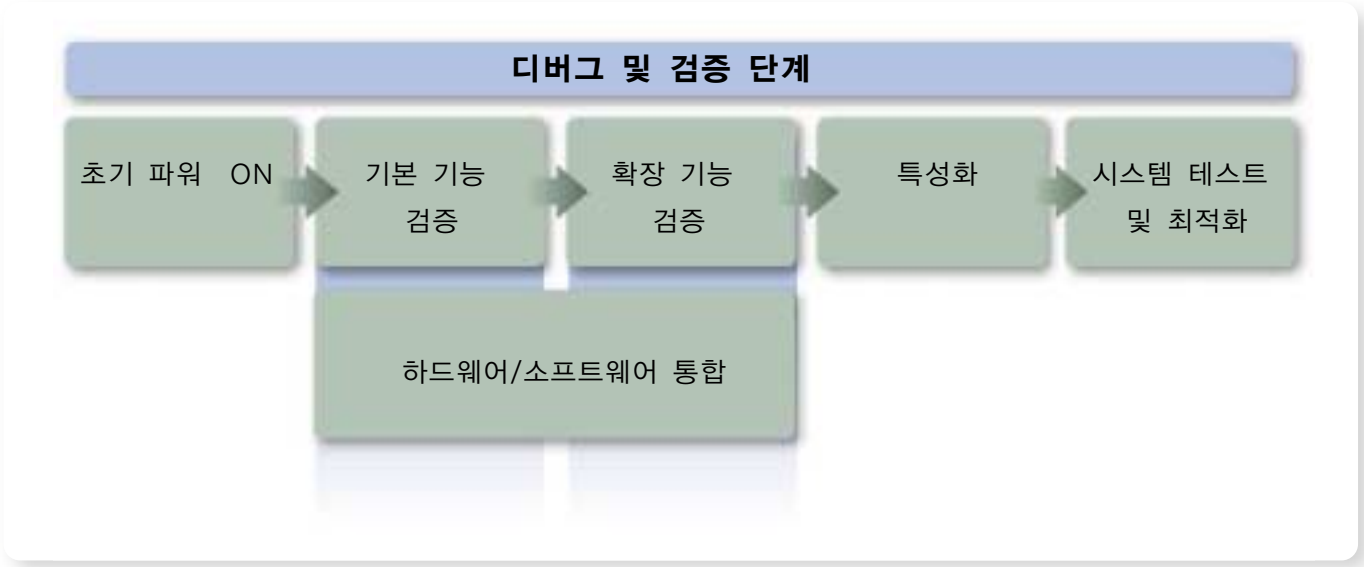
오실로스코프 액세스의 경우에는 어떨까요? 액세스할 필요가 있는 신호들 중에는 이들 인터페이스의 일부에 포함되어 있지 않을 수 있습니다.

고려 사항으로는 전원 장치, 클럭 및 재설정 기능이 있습니다. 오실로스코프 프로브에 대한 그라운드 포인트 수는 쉽게 액세스할 수 있는 수준으로 할 것을 잊지 마십시오.

버스	액세스 방식
로컬 버스	고밀도 로직 애널라이저 프로브 포인트에 설계
SDRAM	DIMM 인터포저 카드
내장형 PCI 버스	고밀도 로직 애널라이저 프로브 포인트에 설계
FPGA I/0	고밀도 로직 애널라이저 프로브 포인트에 설계
시스템 버스	사용자 지정 인터포저 카드

► **표 1 –** ***로직 애널라이저 액세스 포인트 전략***

오실로스코프 프로브 사용에 대한 대안으로서 일부 로직 애널라이저는 디지털 및 아날로그 정보를 모두 캡처하는 프로빙 솔루션을 제공합니다. 사용자는 이런 프로브를 사용하여 아날로그 신호의 경로를 로직 애널라이저에서 오실로스코프로 취사 선택할 수 있습니다. 그 기본 개념이 그림 10에 나와 있습니다.



► **그림11 –** *디버그 및 검증 개요*

2.4 요약

이 섹션에서는 제품 개발 주기의 설계 단계에서 효율적인 디버그 및 검증 과정이 시작되는 것을 살펴 보았습니다. 보드가 조립된 후까지 디버그 및 검증 필요성을 고려하기를 기다리다 보면 디버그 및 검증하는 것이 불가능하지는 않더라도 매우 어려워지는 결과를 낳게 되는 경우가 종종 있습니다.

효과적인 설계 과정은 디버그 및 검증 시간을 최소화하는 데 있어 첫 번째 단계입니다. 팀에서 문서화하여 따르는 설계 과정의 목표는 가급적 빨리 가능한 한 많은 오류를 찾아내어 수정함으로써 디버그 및 검증 단계 후반부에 이런 작업을 하느라 시간과 비용을

낭비하지 않도록 하는 것입니다.

문제가 어디서 발생할 지 파악하는 것이 두 번째 관건입니다. 기능상의 버그와 신호 무결성 문제가 특별히 관심을 끄는 두 영역입니다. 클럭 분배, 재설정 기능 및 전원 장치에 잠재적 문제가 있는지 면밀히 검사해야 합니다. 고속 회로는 신호 무결성 문제를 최소화할 수 있도록 주의를 기울여 배치 및 배선해야 합니다.

마지막으로, 설계에 적당한 테스트 액세스 포인트를 추가하면 디버그 및 검증 작업이 훨씬 쉬워집니다.

디버그 및 검증을 위한 설계는 설계 단계에서 로직을 정확하게 구현하는 것만큼이나 중요합니다.

3 디버그 및 검증 단계

수많은 고된 작업과 신중한 계획 수립 후 마침내 보드가 완성되었습니다. 이제 디버그 및 검증 프로세스를 시작합니다. 그림 11에 나타낸 것과 같이, 디버그 및 검증을 위한 6가지 기본 단계가 있습니다. 우리가 작업 중인 설계에 대해 각각의 단계를 거친 후에야 제품 선적을 시작할 수 있습니다.

다음 섹션에서는 각 단계를 개별적으로 자세히 다루겠습니다.

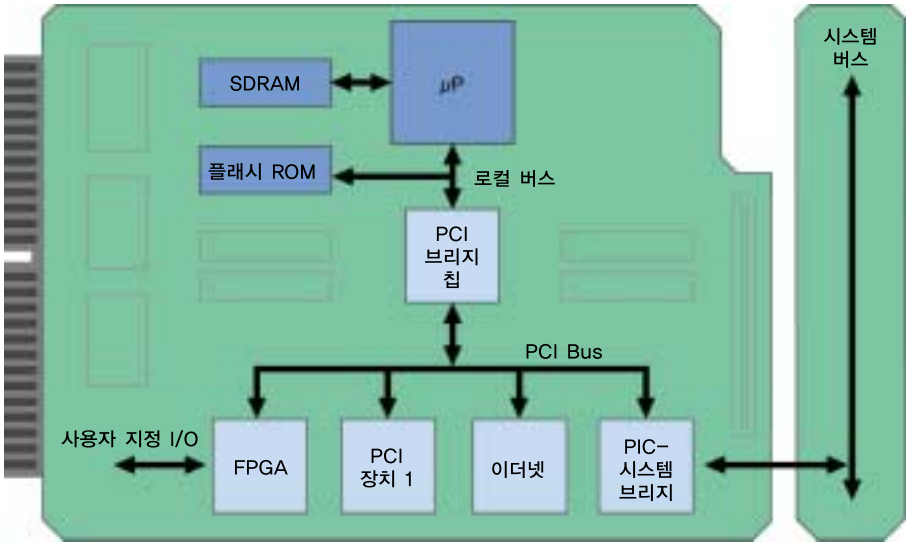


그림12 – *마이크로프로세서 완성을 위한 기본 기능 검증*

3.1 초기 파워 ON

스모크 테스트가 아마도 이해하기 가장 쉬운 단계일 것입니다.

전원 장치와 접지부가 함께 단락되지 않았음을 검증한 후 보드에 전원을 넣습니다. 이때 연기가 나면 분명히 불량품입니다. 연기가 나지 않아야 합격입니다.

3.2 기본 기능 검증

첫 번째 단계는 기본 기능 검증을 수행하는 것입니다.

이 단계는 설계 실행의 핵심이 되는 단계입니다. 마이크로프로세서를 제대로 작동시킬 수 있습니까? 전원이 인가될 때 예상한 대로 작동합니까? 클럭이 작동 중입니까? 재설정 기능이 올바르게 작동합니까? 보드 전체에 전력이 정확히 분배됩니까? 이런 핵심 사항이 검증되고 나면 마이크로프로세서가 올바르게 부팅되는지 확인할 필요가 있습니다. 그림 12에 나타낸 것과 같이, 우리는 의도적으로 시스템의 나머지 부분을 무시한 채 설계의 핵심 기능에 집중하고 있습니다.

이 때 발생할 수 있는 문제가 많기 때문에 발생 가능한 기능 오류와 신호 무결성 문제를 다루어야 할 뿐만 아니라 보드 빌드 오류도 처리해야 합니다.

이 시점에서 문제를 효과적으로 디버그하려면 신호들 사이의 관계를 정확히 관찰해야 합니다. 이 단계에서는 세부 분석까지는 보통 필요치 않지만 신호를 가시화할 수 있어야 합니다. 설계 노트북과 화이트보드에 그린 시퀀스가 일치합니까? 어떤 전압 수준에서 시스템 재설정에 문제가 생기는지 볼 필요가 있습니다. 전원 장치 순차 처리를 관찰하고 마이크로프로세서 부팅 시퀀스 및 로컬 버스 주기를 가시화해야 합니다. 이런 것들을 가시화하는 데는 실시간 오실로스코프와 로직 애널라이저와 같은 툴이 제격입니다.

두 계측기 모두 설계에 대한 우수한 가시성을 제공합니다.

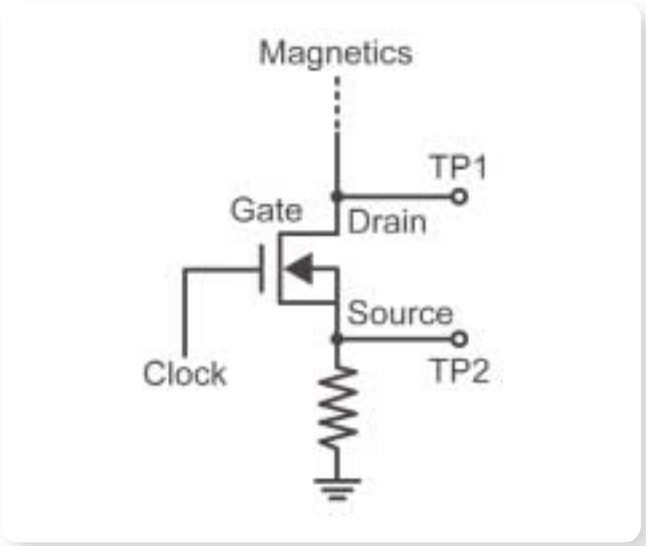
대부분의 엔지니어는 처음에 DSO(디지털 스토리지 오실로스코프)나 DPO(디지털 포스퍼 오실로스코프)를 선택합니다. 이 두 가지 유형의 오실로스코프는 필요한 성능(대역폭, 샘플 속도 등) 뿐만 아니라 다양한 트러거링 선택 및 프로빙 옵션도 제공합니다. 가장 중요한 것은, 이런 실시간 플랫폼은 전원 장치 노이즈에서 고속 신호에 이르기까지, 테스트 포인트를 쉽게 탐지하고 신뢰성 있게 파형을 포착할 수 있게 해준다는 점입니다.

DSO는 에지가 빠르거나 펄스 폭이 좁은 저/고 반복 비율의 신호에 이상적입니다. 또한 전원 장치 순차 처리 및 재설정 작업과 같은 단발 신호 이벤트와 과도 응답을 캡처하는 데에도 탁월합니다. 본 설계 프로젝트의 경우 TDS6000 시리즈가 알맞은 솔루션입니다.

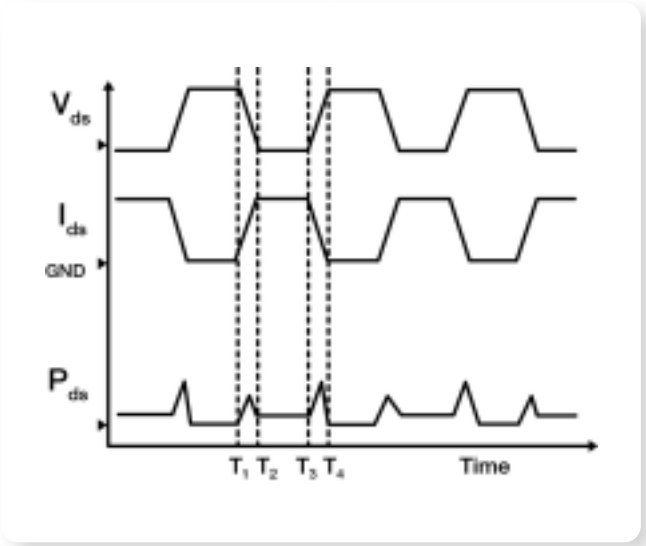
DPO는 디지털 문제해결과 단속 신호를 찾는 데 적합한 툴입니다. DPO의 매우 우수한 파형 캡처 속도는 다른 어떤 오실로스코프보다도 빠르게 정보를 스윕한 후 오버레이함으로써 선명도를 달리하여 발생 주파수 세부 정보를 컬러로 제시합니다. 이 경우 TDS5000B는 디지털 시스템 설계 애플리케이션의 가혹한 요구 조건을 충족하는 DPO 솔루션을 예증합니다.

High Speed 오실로스코프에서는 프로브의 선택이 매우 중요한 요소입니다. 보드에서 신호를 정확히 포착할 수 없는 경우에는 어느 엔지니어라도 문제를 효과적으로 디버그할 수 없을 것입니다. 설계 단계에서 구현한 디버그 솔루션은 신호를 정확히 포착할 수 있도록 핵심적인 역할을 수행하지만 올바른 프로브를 사용하는 것 또한 매우 중요합니다.

오실로스코프와 프로브는 계측 시스템으로서 함께 작동합니다. 가능하다면 프로브는 오실로스코프 그 자체와 같은 대역폭을 제공해야 합니다. 그리고 대역폭 등급 외에도 프로브는 신호에 대해 최소한의 부하 만을 지니고 있어야만 합니다. 이상적인 것은, 프로브의 대역폭을 포함한 계측 시스템 대역폭이 관찰하는 신호의 주파수보다 최소한 3배(3X) 이상이 되는 것입니다. 다음 단계는 전원 장치 검증입니다.



▶ **그림13 –** *SMPS(스위치 모드 전원 장치)*



▶ **그림14 –** *스위칭 장치 작동*

3.2.1 전원 장치 분석

이 설계에 사용되는 DC 전원 장치는 부하를 효율적으로 변경하는 능력으로 유명한 SMPS(스위치 모드 전원 장치)입니다. 전원 "전원 경로" 에는 수동, 능동 및 자기 소자가 포함됩니다. 그림 13은 능동, 수동 및 자기 소자가 있는 전력 변환 섹션을 보여주는 단순화된 SMPS 회로도를 나타낸 것입니다.

SMPS 기술은 MOSFET(metal oxide semiconductor field effect transistors, 금속 산화막 반도체 전계 효과 트랜지스터)와 IGBT(insulated gate bipolar transistors, 절연 게이트 바이폴러 트랜지스터)와 같은 전원 반도체의 스위칭 장치에 달려 있습니다. 이들 장치는 빠른 스위칭 시간을 제공하고 이상 전압 스파이크에 견딜 수 있습니다. 이와 똑같이 중요한 점은 이 장치들이 On 또는 Off 상태에서 매우 적은 전력만을 소비하므로 발열량은 적으면서도 높은 효율을 달성합니다. 대부분의 경우 스위칭 장치가 SMPS의 전체 성능을 결정합니다.

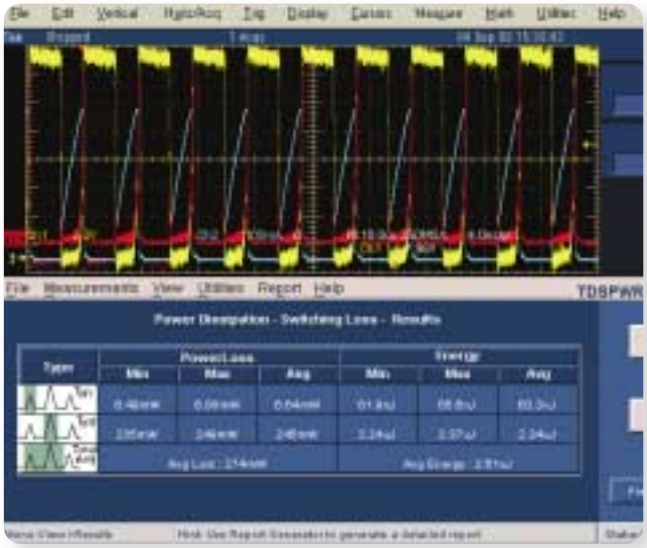
TDS6000 시리즈를 사용하여 스위치 모드 전원 장치의 두 가지 주요 측면인 스위칭 손실과 전원 장치 리플을 정량화 해보겠습니다.

3.2.1.1 전원 장치 스위칭 손실

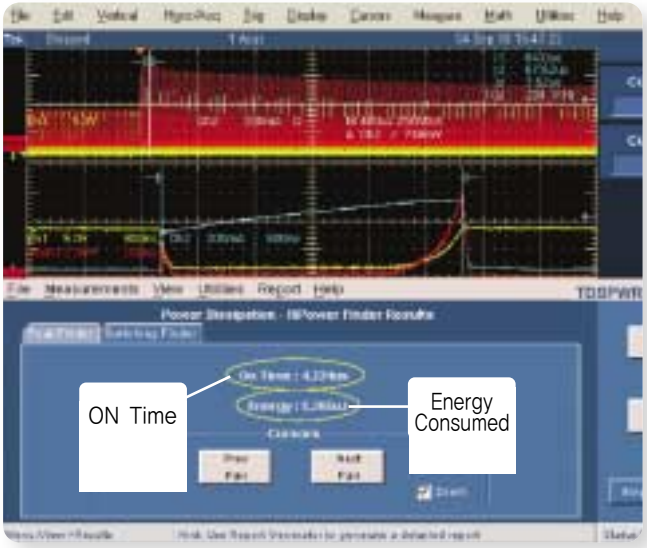
전원 장치에서의 스위칭 손실이 효율을 결정합니다. 따라서 이 계측은 가능한 한 빨리 수행하는 것이 중요합니다. 스위칭 손실은 전원 장치가 안정적으로 작동되는 상태에 있고 동적으로 부하가 변동되는 와중에 분석해야 합니다.

스위칭 장치에서의 전력 손실을 계측하기 위해 우선 그림 14에 표시된 것과 같이 스위칭 장치 신호를 검토해야 합니다.

스위칭 장치에 걸리는 전압은 장치가 꺼져 있는 동안 높을 것이며 전도 시간 중(On 상태)에는 낮을 것입니다(V 포화). 장치가 Off 상태 중에는 전류가 없습니다. 하지만 전도 시 전류는 최대치에 이릅니다. 전력 파형을 보면 변환 중에 최대 순간 전력 손실이 발생합니다. 스위칭 장치가 Off 상태에서 On 상태로 변환하는 동안의 전력 손실을 TOn 손실이라 합니다. On 상태에서 Off 상태로 변환하는 동안의



► **그림15 –** *스위칭 손실 계측*



► **그림16 –** *HiPower Finder 계측*

전력 손실을 T_{Off} 손실이라 합니다. 전도 시간 중의 전력 손실을 전도 손실이라 합니다. 정수 주기의 전력 손실을 총 평균 전력 손실이라 합니다. 위 그림에서 T1 ~ T2 중의 전력 손실이 TOn 손실이고, T3 ~ T4에서는 TOff 손실 그리고 T1 ~ T4에서는 주기 전력 손실입니다.

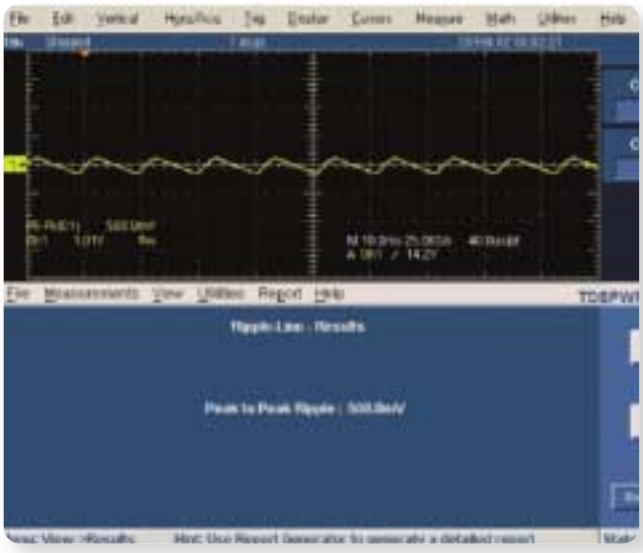
TDSPWR2 애플리케이션과 함께 TDS6000 시리즈는 보다 자동화된 솔루션을 제공하여 버튼만 한 번 눌러 이 계측 작업을 수행할 수 있습니다.

그림 15에 표시된 이 계측 기능은 설계 최적화에 사용될 수 있는 정보를 제공합니다. 예를 들어 T_{On}과 T_{Off}를 알면 총 전력 손실을 줄일 수 있는지 여부를 결정할 수 있고, 총 평균 전력 손실 값을 이용하여 히트 싱크 설계를 최적화할 수 있으며, 전원 장치의 신뢰성은 과도한 전력 손실을 찾아내어 분석될 수 있습니다.

대부분의 시스템에서 전원 장치는 시간의 경과에 따른 부하 변동을 바탕으로 전력을 공급할 필요가 있습니다. 이런 부하 변동 중 스위칭 장치의 스위칭 손실이 변하게 됩니다. 순간 전력 손실이 규정된 한계 이내가 되도록 하려면 설계자는 이벤트를 캡처하고 그것을 분석하여 전력 손실을 파악해야 합니다. TDSPWR2의 HiPower Finder 기능은 이 계측 작업을 자동화함으로써 그림 16에 표시된 것과 같이 필요한 종합적인 정보를 제공합니다.

3.2.1.2 전원 장치 리플

마지막으로, 각 전원 장치의 리플을 분석할 필요가 있습니다. 리플은 출력 전압에 편승해 있는 불필요한 주파수 구성 요소에 지나지 않습니다. 다시 한 번 버튼 하나만 눌러 그림 17에 표시된 것과 같이 원하는 결과를 얻을 수 있습니다.



▶ **그림17 –** *전원 장치 리플 계측*

3.2.2 기본 기능 검증

DSO나 DPO가 기본 기능 검증 단계에서 발생하는 수많은 디버그 문제를 해결할 수 있지만, 때로는 그것만으로 충분치 못한 경우가 있습니다. 오실로스코프만으로 충분하지 않을 때는 언제일까요?

3.2.2.1 마이크로프로세서 재설정 디버그

가장 일반적인 상황은 5개 이상의 신호가 가진 타이밍 관계를 봐야 하는 경우입니다. 예를 들어 마이크로프로세서가 재설정에서 나올 때는 5–7개의 제어 신호에다 주소와 데이터 라인까지 모니터링해야 합니다. 이 범위는 일반적으로 8비트 컨트롤러를 위한 21개의 신호로부터 32비트 마이크로프로세서를 위한 75개의 신호까지입니다. 로직 애널라이저의 독립형 TLA 시리즈는 34, 68, 102 또는 136개의 채널을 포착할 수 있는 모델을 제공하는 이런 유형의 디버그에 적합합니다. 채널이 더 많이 필요한 경우에는 모듈러 TLA 시리즈에서 훨씬 더 많은 채널을 이용할 수 있습니다.

신호 상호간의 타이밍 관계를 가시화하기 위해서는 로직 애널라이저의 비동기 클러킹 모드를 사용해야 합니다. 로직 애널라이저는 비동기 클러킹 기능을 이용하여 테스트 중인 시스템으로부터 데이터를 샘플링하는 데 사용되는 자체적인 내장 클럭 신호를 생성합니다. 모든 샘플은 정기적으로 고정된 간격으로 수집합니다. 비동기 클러킹을 이용한 샘플 신호 포착을 종종 타이밍 포착이라고 합니다.

3.2.2.2 마이크로프로세서 부트 디버그

오실로스코프만으로는 불충분한 또 다른 상황은 마이크로프로세서나 버스 작동을 모니터링할 필요가 있는 경우입니다.

예를 들어 설계한 마이크로프로세서의 부팅 작업을 검증 또는 디버그할 필요가 있을 때 로직 애널라이저가 플래시 ROM 인터페이스에 연결된 마이크로프로세서를 모니터링하고 부트 코드 작동을 표시할 수 있습니다. 로직 애널라이저를 이용하면 데이터를 추출할 수 있으며 이는 파형 디스플레이에 나타나는 75개의 신호를 관찰하여 얻을 수 있는 시스템 작동 관련 정보보다 더 많은 정보를 제공합니다.

마이크로프로세서 부트 시퀀스를 보려면 로직 애널라이저의 동기 클러킹 모드를 사용합니다. 동기 클러킹을 사용하는 경우에는 로직 애널라이저 샘플을 수집할 때 구동하는 데 사용되는 클럭 신호가 보드에서 생성됩니다. 클럭 신호는 고정 주파수이거나 매우 큰 변이를 보일 수 있습니다. 동기 클러킹으로 수집하는 것을 종종 상태 포착이라고도 합니다.

3.2.3 요약

기본 기능 검증 단계에서는 설계의 기능을 검증합니다. 제품의 다른 기능 및 특징과 인터페이스를 검증할 수 있도록 기능적인 신뢰성을 제공할 수 있습니까? 기본 기능 검증에는 종종 전원 장치, 재설정, 클럭 및 주요 제어 신호를 검증하는 과정이 포함됩니다. 오실로스코프와 로직 애널라이저 모두 이 작업을 하는 데 이상적인 툴입니다.

► ***확장 기능 검증***

3.3 확장 기능 검증

안정적이지만 최소 사양으로 구성된 시스템이 작동 중이라면 다른 블록을 검증할 필요가 있습니다. 이상적인 것은 한 번에 한 블록, 한 기능, 한 인터페이스를 검증하는 것입니다. 각각의 검증 작업이 끝나면 다음으로 이동합니다. 설계의 핵심 기능 이외에 검증해야 할 첫 번째 기능은 로컬 버스와 로컬 PCI 버스를 연결하는 브리지 칩입니다. 여기서는 드라이버가 브리지 칩을 정확히 초기화 및 프로그래밍하고 있는지, 적절한 버스 주기가 생성되고 있는지, 타이밍 파라미터가 충족되는지 검증합니다.

3.3.1 통합 솔루션 사용

일정을 맞추려면 문제의 진짜 원인을 빠르게 식별하는 것이 관건이 됩니다. 아날로그 또는 디지털 중 어떤 것을 선택할까요?

이는 두 영역 모두 다룰 수 있는 툴과 문제해결 방식을 사용함을 의미합니다. 선호되는 솔루션은 우리가 이미 사용한 DPO나 DSO 및 로직 애널라이저로 계측기를 편성하는 것입니다. 앞서 봤듯이, DSO가 글리치와 같은 개별 이벤트뿐만 아니라 왜곡, 변환 시간, 중요한 Setup 및 Hold 타이밍 값을 관찰하는 데 최적의 툴입니다.

► **그림18 –** *iLink™ 툴 세트*

로직 애널라이저는 시스템을 따라 움직이면서 기본적인 형태(관련된 타이밍 정보와 함께 이진수 값)로 로직 신호를 캡처합니다. 아날로그와 디지털의 두 영역 간 상호 작용을 캡처하는 것이 효율적인 문제해결에 관건입니다.

일부 첨단 솔루션, 특히 텍트로닉스 로직 애널라이저 시리즈와 TDS 시리즈 DPO에는 트리거와 시간 상관관계 표시 정보를 공유하는, 즉 두 계측기를 통합하는 기능이 있습니다. 그림 18에 나타낸 iLink™툴 세트를 이용하여 로직 애널라이저와 오실로스코프의 장점을 결합할 수 있습니다. 이 섹션에서는 이 두 계측기를 함께 작동시켜 세부적인 설계 문제까지 파고들 수 있는 방법을 알아봅니다.

iLink™툴 세트: 두 강력한 계측 툴을 하나로

로직 애널라이저와 오실로스코프는 디지털 문제해결을 위해 오래 전부터 사용되어왔지만 모든 설계자가 이 두 핵심적인 계측기를 통합함으로써 얻을 수 있는 장점을 경험한 적이 있는 것은 아닙니다.

로직 애널라이저는 디지털 정보 스트림을 분석하여 회로 오류를 트리거하고 관련 이벤트를 캡처함으로써 디버깅 및 검증 시간을 단축합니다.

오실로스코프는 디지털 타이밍 다이어그램 이면의 정보를 간파하여 원시 아날로그 파형을 표시함으로써 신호 무결성 문제를 신속히 드러내어 줍니다.

업계에서도 독보적인 입지를 굳히고 있는 로직 애널라이저/오실로스코프 통합 패키지인 iLink™툴 세트는 여러 텍트로닉스 로직 애널라이저 모델에서 제공됩니다. iLink™툴 세트는 텍트로닉스 로직 애널라이저 시리즈의 파워를 정선된 TDS 시리즈 오실로스코프 모델과 결합시켜 줍니다.

일련의 강력한 iLink™ 툴 세트 기능을 사용하여 시간 상관 디지털 및 아날로그 신호를 로직 애널라이저 디스플레이에 나타낼 수 있습니다. 로직 애널라이저는 신호를 포착하여 디지털 형태로 표시하는 반면, 연결된 TDS 시리즈 오실로스코프는 같은 신호를 아날로그 형태로 캡처하여 로직 애널라이저 화면에 표시합니다.

이 두 가지 뷰를 동시에 보면 예컨대 디지털 영역의 타이밍 문제가 어떻게 아날로그 영역에서 글리치를 일으키게 되는지 쉽게 알 수 있습니다.

iLink™ 툴 세트는 다음과 같이 문제 탐지 및 문제 해결 시간을 단축하도록 설계된 종합적인 패키지입니다.

- iCapture™ 다중화 툴은 단일 로직 애널라이저 프로브를 통해 디지털 및 아날로그 신호를 동시에 포착합니다.

- iView™ 디스플레이는 로직 애널라이저 디스플레이에 시간과 관련된 로직 애널라이저 및 오실로스코프 계측 결과를 통합적으로 표시합니다.

- iVerify™ 분석 툴은 오실로스코프에서 생성된 아이 다이어그램을 사용하여 멀티채널 버스 분석 및 검증 테스트를 실시합니다.

앞서 언급했듯이, 설계 과정에서 만나게 되는 한 가지 공통적인 문제는 단선 및 단락과 같은 보드 관련 문제입니다.

일정을 맞추려면 보드에서 발생한 단선 또는 단락 문제를 재빨리 식별할 수 있어야 합니다. 불행히도 로컬 PCI 버스로 넘어가려 할 때 보드와 관련된 것으로 보이는 문제를 만났습니다.

이런 문제를 식별하기 위한 한 가지 기법은 메모리 또는 디버그 스크립트를 사용하는 것입니다. 이런 간단한 소프트웨어 루틴에서는 주소 범위나 단일 주소에 미리 정의된 데이터 패턴을 쓰고 읽습니다. 한 가지 익숙한 테스트가 바로 1 반복 테스트(alternating 1's test)입니다. 어떤 주소에 0x55와 0xAA라는 패턴을 교대로 계속 쓰면 모든 단일 데이터 비트가 0과 1 사이를 반복적으로 전환하게 됩니다. 루프 실행 중인 상태에서 각 데이터 비트를 오실로스코프로 프로브하여 모두 전환되는지 그리고 어떤 것도 1, 0 또는 제3의 상태로 고착된 것이 없는지 확인할 수 있습니다.

이것이 쉽게 보이는 만큼이나 실행하기 어려운 경우가 종종 있습니다. 단일 데이터 비트를 모니터링하는 프로세스는 다음과 같습니다.

1. 회로도에서 데이터 비트를 찾고 가능한 프로브 위치를 결정합니다.
2. 보드 회로도를 검토하여 가장 쉬운 프로브 위치를 찾습니다. IC 핀에서 프로브해야 합니까? Through a via? 혹은 테스트 포인트를 추가했습니까? 이 경우에는 U34 핀 18에서 PCI_AD0(PCI 버스 주소/데이터 버스 비트 0)를 프로브하겠습니다.
3. 육안 검사나 보드 회로도를 보고 보드에서 U34를 찾습니다.
4. U34의 18번 핀을 찾아 그 핀에 스코프 프로브를 연결합니다. 불행히도 이렇게 하기가 항상 쉬운 것은 아닙니다. 부품 배치가 복잡하고 BGA가 문제를 더욱 복잡하게 만들기 때문입니다.

32비트 버스 하나를 확인하기 위해서는 3분이 걸리는 이 프로세스를 31회 이상 반복하여 총 90분 이상 걸리는 작업이 요구됩니다. 따라서 이 작업을 보다 간단하고 생산적으로 수행할 수 있는 방법이 있어야 합니다.

우리가 세운 디버그 전략의 일부가 PCI 버스에 대한 전용 테스트 포인트를 설계하는 것이었다는 것을 기억하십니까? 이 설계에는 그림 19에 나타낸 P6960 고밀도 로직 애널라이저 프로브용 패드가 포함됩니다. P6960은 값비싼 온보드 커넥터가 없으며 D-Max™ 커넥터리스 프로빙 기술을 사용한 회로 기판의 패드에 연결할 수 있습니다.

► **그림19 –** *D-Max™ 커넥터리스 프로브*

TLA 시리즈 iCapture™멀티플렉서를 통해 로직 애널라이저에 연결된 오실로스코프에 아날로그 신호를 전달하는 것이 중요합니다. iCapture™ 멀티플렉서는 로직 애널라이저가 앞의 그림 10에서처럼 단일 프로브를 통해 디지털 및 아날로그 신호를 동시에 포착할 수 있도록 해줍니다.

이제는 보드에서 32개의 다른 신호를 프로브하기 위해 오실로스코프를 사용할 필요 없이 보드에 P6960 프로브를 한 번만 연결하면 되며 마우스만 몇 번 간단히 클릭하면 모든 32비트가 정확히 전환 중인 것을 검증할 수 있습니다.

그 프로세스는 다음과 같이 간단합니다.

1. 그림 20에 표시된 것과 같이 34채널 D-Max™커넥터리스 프로브를 회로 기판에 연결합니다.
2. 로직 애널라이저 모듈의 CH1 아날로그 출력을 오실로스코프 채널에 연결합니다.
3. TLA 애플리케이션에서 아날로그 공급(Analog Feeds) 대화 상자를 사용하여 관심 있는 신호를 아날로그 출력 중 하나에 지정합니다. 이 경우에는 PCI_AD31:0을 CH1 출력에 지정합니다. CH1에 대해 아날로그 공급 사이클링(Analog Feed Cycling) 확인란을 선택하면 32개의 신호 각각을 오실로스코프로 쉽게 라우팅할 수 있습니다.
4. 오른쪽 화살표 버튼을 클릭하여 다음으로 선택된 신호를 오실로스코프로 라우팅합니다. 오실로스코프 상의 신호를 검사하여 토글 전환되는지 검증합니다.
5. 모든 신호의 검사가 끝날 때까지 4단계를 반복합니다.

32비트 버스 전체를 검사하는 데 90분이 걸렸던 작업이 마우스를 32회만 클릭하고 시간으로 따지면 겨우 90초밖에 걸리지 않는 간단한 작업으로 줄어들었습니다.

이 기술을 사용한 결과 PCI_AD7과 PCI_AD13이 납땜으로 연결되어 있다는 사실을 쉽게 알 수 있었습니다.

그림20 – *D-Max™커넥터리스 프로브와 회로 기판 연결*

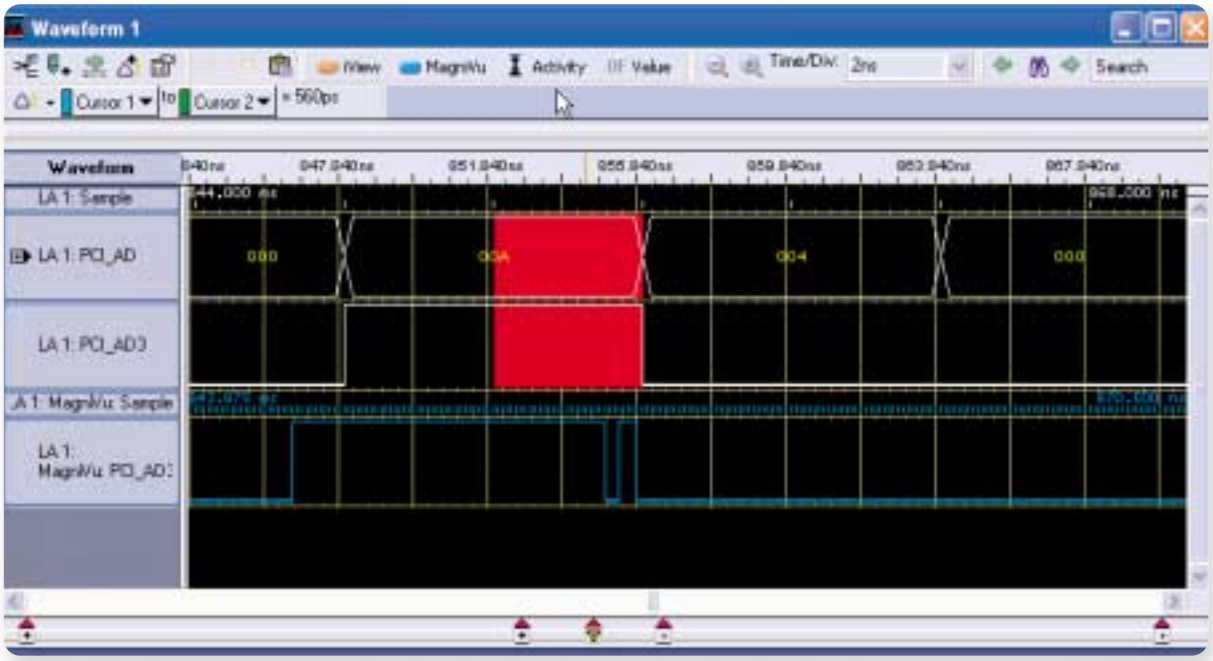
► **그림21 –** *로직 애널라이저가 글리치를 트리거하고 개별 글리치 위치를 플래그로 표시한 모습*

다른 기능 블록으로 이동하면 다른 디버그 문제가 발생합니다. 이 경우에는 시뮬레이션에 나타난 것처럼 데이터가 처음에는 로컬 버스에서 PCI 버스로 안정적으로 전송됩니다. 하지만 몇 차례 트랜잭션이 발생한 후 로컬 PCI 버스에 오류가 나타나기 시작합니다. PCI_AD 버스(PCI 주소/데이터 버스)에서 0x0008이 되어야 하는 값이 한 번도 아니고 반복해서 0x0000으로 나타납니다. 이 문제는 "고착된" 오류나 라우팅이 잘못된 신호로 보이지는 않으며 같은 오류를 계속해서 발생시키고 있습니다. 이 문제가 지닌 변칙성으로 미루어볼 때 일부 단속 이벤트가 올바른 데이터 비트로 잘못 인식되어 16진수 결과 값이 바뀌는 것으로 해석됩니다. 문제가 반복적으로 발생한다는 것은 프로토타입의 레이아웃이나 어셈블리의 오류로 인한 글리치가 있음을 나타냅니다.

PCI_AD 버스 오류가 나타나는 것은 어떤 상황입니까?

이 오류의 실제 특성은 로직 애널라이저의 글리치 캡처 트리거(Glitch Capture Trigger)와 디스플레이 모드를 사용하여 포착을 완료한 후에 그 모습을 드러내기 시작합니다. 글리치를 탐지할 때 로직 애널라이저를 트리거하는 이 모드에서는 타이밍 디스플레이에 그 위치도 플래그로 표시됩니다. 로직 애널라이저는 "글리치"를 샘플 포인트 사이에서 신호 변환이 두 차례 이상 발생하는 것으로 정의합니다. 그림 21은 TLA 시리즈 로직 애널라이저의 결과를 디스플레이한 것입니다.

그림 21에는 두 가지 유형의 "파형"이 표시되어 있습니다. 화면 위쪽에는 각 버스에 대한 워드 값을 나타내는 버스 파형으로 PCI_AD 버스가 요약되어 있습니다. 버스 파형은 수많은 개별 신호의 상태를 한눈에 알아볼 수 있도록 표시되므로 문제해결 시 시간을 절약할 수 있습니다. 또한 각각의 개별 신호 라인을 전개하고 다시 글리치 위치에 플래그 표시하도록 디스플레이를 구성할 수 있습니다.



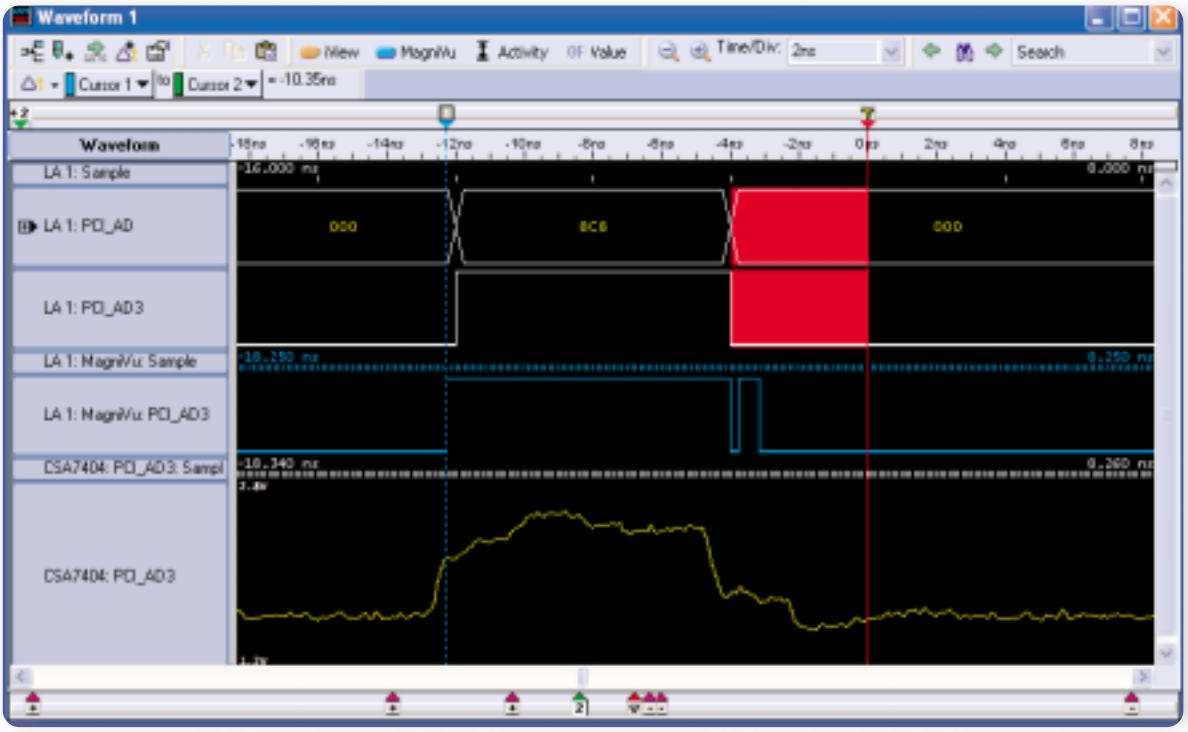
▶ **그림22 –** *로직 애널라이저의 MagniVu™ 포착 디스플레이는 PCI_AD3 신호의 오류 표시*

그림 21에서 클럭 간 주기는 4.00ns입니다. 이 예에서 TLA 시리즈 로직 애널라이저의 MagniVu™ 포착 툴은 트리거 포인트 근처에서 125ps 간격으로 신호 값을 캡처하고 임의의 신호를 별도로 나타내는 고해상도 뷰로 이 정보를 표시할 수 있습니다.

이 기능은 동일한 프로브를 통해 메인 타이밍 데이터와 동시에 고해상도 데이터를 포착합니다.

그림 22는 MagniVu™ 포착 휘선을 추가한 디스플레이를 나타낸 것입니다. 여기에는 125ps 클럭과 PCI_AD3 신호의 상세 뷰가 모두 버스 파형 뷰와 함께 표시됩니다. 이 신호는 주기의 후반부 절반에서 발생하는 짧은 변환을 표시합니다. 이 주기에서 틀린 버스 값이 생성되고 있다는 것을 이미 알고 있으므로 이 변환이 오류의 원인일 가능성이 높습니다. 하지만 잘못된 변환의 원인은 무엇일까요?

디지털 신호 착오는 아날로그 신호 무결성 문제로부터 비롯되는 경우가 자주 있습니다. 앞서 그림 18에 표시된 iLink™ 툴 세트를 사용하면 디지털 변형의 아날로그적 특성을 쉽게 파악할 수 있습니다. 모듈러 TLA 시리즈의 iCapture™ 멀티플렉서는 로직 애널라이저 내의 아날로그 멀티플렉서를 통해 D-Max™커넥터리스 프로브에서 측정한 신호들을 TDS 시리즈 오실로스코프까지 전달합니다. 이 멀티플렉서는 아날로그 및 디지털 신호 모두를 위한 경로를 제공하므로 프로브가 이중 프로빙하는 일과 장치 신호 관련 이중 부하가 없습니다. iLink™ 툴 세트의 또 다른 기능인 iView™은 로직 애널라이저 화면에 결과 시간 상관 디지털 및 아날로그 파형을 표시합니다.



► **그림23 –** *iView™계측 결과 PCI_AD3 상에 나타나는 디지털 오류의 원인이 되는 아날로그 동작이 표시됨*

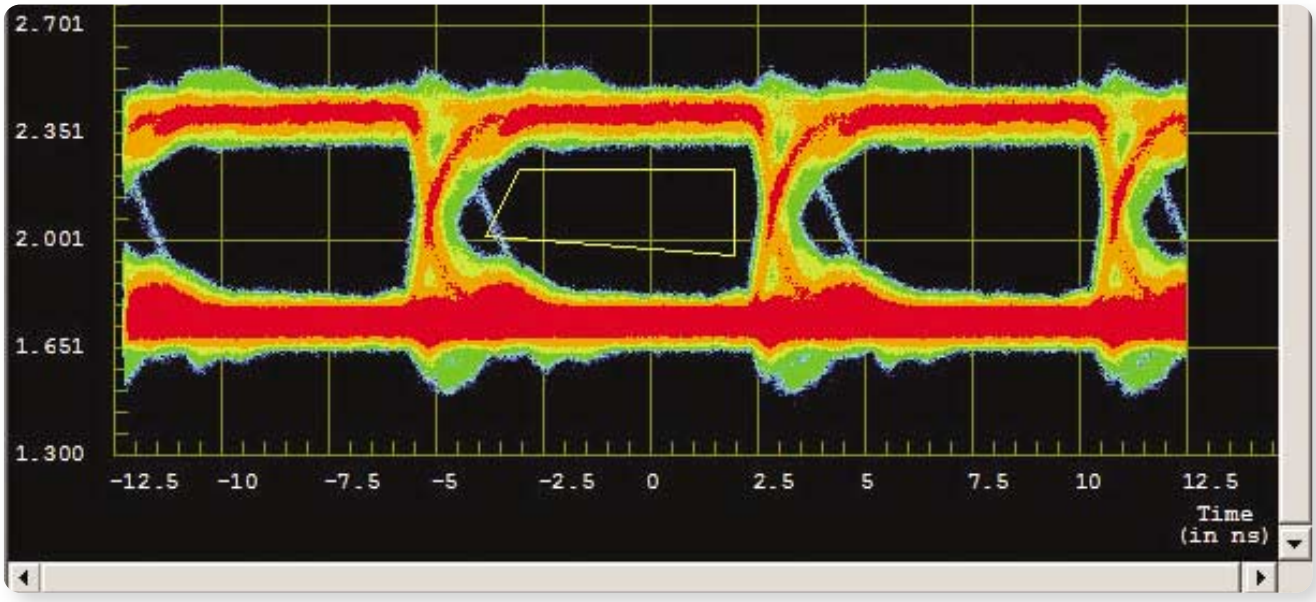
그림 23은 아날로그 신호 PCI_AD3과 이를 디지털 신호로 변환한 파형 모두를 함께 화면 상에 정렬한 것입니다. 이 화면은 전체적인 정보를 보여주는 화면입니다. 즉 디지털 글리치가 발생한 바로 그 순간에 아날로그 신호의 진폭이 로직 임계값 영역에서 저하됩니다.

진폭은 분명히 순간적으로나마 임계 전압 아래로 떨어져 일시적인 "로우(low)" 또는 로직 0 레벨을 만듭니다. 그런 다음 임계값을 넘을 만큼만 증가하여 "하이(high)" 또는 로직 1 레벨로 돌아온 후 주기 경계에서 다시 로직 0으로 전환됩니다.

이런 아날로그 동작이 글리치의 원인이며 버스 상의 16진수 출력에서 발생하는 오류의 원인이 됩니다. 이런 불안정성이 모든 하강 에지에 똑 같은 방식으로 영향을 주는 것은 아니며 많은 펄스가 오류 없이 지나갑니다. 물론, 이 버스 주기에서 파형의 불안정 부분을 전후로 하여 유효 에지가 발생해야 하는 시점을 결정하기 위해 설계 모델을 검토할 필요가 있습니다.

능숙한 엔지니어라면 이 왜곡된 파형에서 실마리를 찾아낼 것입니다. 이와 같이 저하된 로직 레벨은 보통 제대로 종단되지 않은 송신 라인에서 역으로 반사된 결과입니다. 이 설계의 경우에는 신호의 빠른 에지가 신호의 목적지에 종단 저항이 없는 상황에 직면했습니다. 그 결과는 이상하지만 상승 및 하강 에지를 손상시킵니다.

요컨대 로직 애널라이저/오실로스코프 컴비네이션으로 문제해결을 시도하는 것은 아래에 나열된 것과 같이 iLink™툴 세트가 장착된 계측기에서 4가지 신호 형식을 사용하여 개괄적이고 전역적인 뷰에서 화면을 점점 확대하여 면밀하게 각각의 신호를 관찰해가는 과정입니다.



▶ **그림24 –** *iVerify™분석 툴에서는 아이 다이어그램 형식을 사용하여 한 번에 여러 개의 신호와 그 에지 변환을 표시*

– 버스 파형은 버스 상의 어딘가에서 발생하는 문제를 한눈에 알아볼 수 있도록 표시해줍니다.

– Deep 타이밍 파형은 어떤 신호 라인이 관련된 것인지 정확히 보여줍니다.

– 고해상도 MagniVu™타이밍 파형은 더 높은 해상도로 오류 발생 시간을 정확히 표시해줍니다.

– DSO에서 제공되고 iCapture™멀티플렉서와 iView™인터페이스를 통해 연결되어 있는 아날로그 파형은 신호의 아날로그 특성을 캡처하여 잠재적 원인을 보여줍니다.

3.3.2 신호 무결성 문제를 신속히 탐지하는 지름길

iLink™툴 세트가 장착된 로직 분석기에 사용할 수 있는 추가적인 문제해결 툴이 있습니다. 그것은 아날로그 멀티채널 아이 다이어그램 분석 결과를 로직 애널라이저 화면에 표시하는 iVerify™분석 툴입니다.

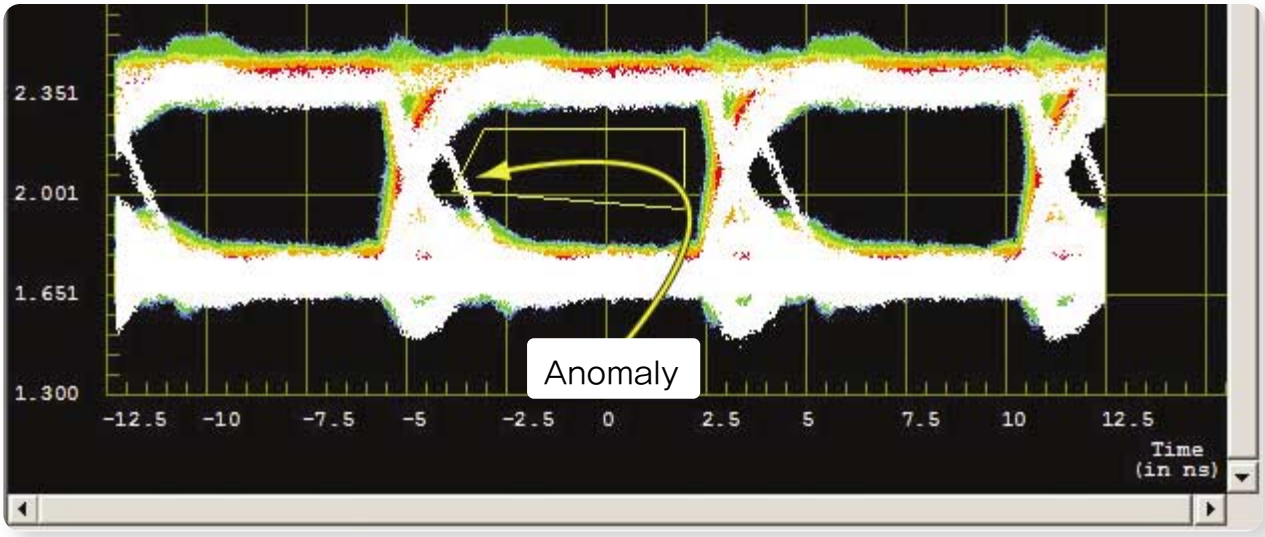
아이 다이어그램은 클러킹된 버스에서 데이터 유효 시간과 일반적인 신호 무결성을 관찰하기 위한 시각적 툴입니다. 이것은 오늘날 사용되는 수많은 버스, 특히 직렬형 버스에 적합합니다. 물론 어떠한 신호 라인도 아이 다이어그램으로 볼 수 있습니다.

iVerify™분석 툴을 사용하면 양의 방향과 음의 방향으로 가는 두 가지 펄스의 리딩 에지와 트레일링 에지를 포괄하는 하나의 뷰로 여러 아이 다이어그램을 통합시킴으로써 문제해결 시간이 단축됩니다.

그림 24는 iVerify™분석 결과를 나타낸 디스플레이입니다.

여기서는 전체 AD 버스에 해당하는 32개의 신호가 겹쳐져 나타납니다.

"한 번에 32개"를 모두 나타낼 때의 장점은 명백한 것이며 32비트 및 64비트 버스에서도 쉽게 적용됩니다. D–Max™커넥터리스 프로브에 연결된 신호 그룹이라면 어떤 것이든 선택할 수 있습니다.



► **그림25 –** *iVerify™ 분석 툴이 잘못된 파형을 앞으로 가져와 강조 표시함으로써 쉽게 평가할 수 있도록 한 모습*

아이 다이어그램은 단일 뷰에 가능한 모든 로직 변환 정보를 제공하므로 신호의 상태도 빠르게 평가할 수 있습니다. 아이 다이어그램은 느린 상승 시간, 과도 응답, 감쇠 수준 등과 같은 아날로그 문제를 보여줍니다. 어떤 엔지니어는 아이 다이어그램을 먼저 본 다음 이상현상을 자세히 추적하는 것으로 평가 작업을 시작합니다.

그림 24의 아이 다이어그램에는 신호 변형이 나타나 있으며, 비교적 드물게 발생하는 변환을 얇은 파란색 줄로 표시합니다. 또한 이 아이 다이어그램에서는 최소한 하나 이상의 신호가 정상 범위를 벗어난 에지를 가지고 있음을 나타냅니다. 마스크 기능을 사용하면 문제의 원인이 되는 특정 신호를 찾아내는 데 도움이 됩니다.

문제를 일으키는 에지가 마스크 영역으로 침투하도록 마스크를 그리면 해당 신호를 분리하여 강조 표시한 상태로 이미지의 전면 레이어로 가져올 수 있습니다. 그 결과가 그림 25에 표시되어 있으며 결함 있는 신호가 맨 앞에 흰색으로 강조 표시되어 있는 것을 볼 수 있습니다.

이 예에서 정상 범위를 벗어난 에지는 PCI_AD3 신호 상의 문제를 나타냅니다. 문제의 원인은 Crosstalk이며 회로 기판 상의 인접한 휘선의 신호에 의해 에지 변화가 유도되고 있는 것입니다.

3.3.3 요약

대부분의 경우 기능 검증 단계에서 여러 가지 문제가 드러나게 됩니다. 이들 문제의 원인은 기능 또는 신호 무결성과 관련된 문제 중 하나일 수 있습니다. 디지털 기능을 테스트할 때는 로직 분석기가 제1의 방어선입니다.

하지만 디지털 문제는 여기서 보여준 바와 같이 부적절한 종단 또는 Crosstalk로 인한 에지 저하를 비롯한 아날로그 신호 문제로부터 유래될 수 있습니다. 로직 애널라이저와 오실로스코프를 함께 사용하고 같은 화면에서 시간 상관된 디지털 및 아날로그 신호를 평가함으로써 둘 중 한 가지 영역에 영향을 미치는 문제를 쉽게 해결할 수 있습니다.

► ***하드웨어/소프트웨어 통합***

3.4 하드웨어/소프트웨어 통합

하드웨어/소프트웨어 통합은 실제로는 기본 기능 검증 단계 중에 시작되어 확장 기능 검증 단계가 끝날 때까지 계속됩니다. 하드웨어/소프트웨어 통합 문제는 디버그하기 어렵고 시스템 내의 하드웨어와 소프트웨어를 모두 검사할 수 있는 툴이 요구됩니다. 로직 애널라이저는 특정 신호를 프로세서의 명령 실행 흐름과 상관시키는 기능을 제공합니다. 이 기능을 사용하면 팀의 모든 하드웨어 그리고 소프트웨어 엔지니어들이 왜 특정 명령이 메모리 에러를 일으켰는지 또는 소프트웨어가 오작동을 하는 지 쉽게 찾아낼 수 있습니다.

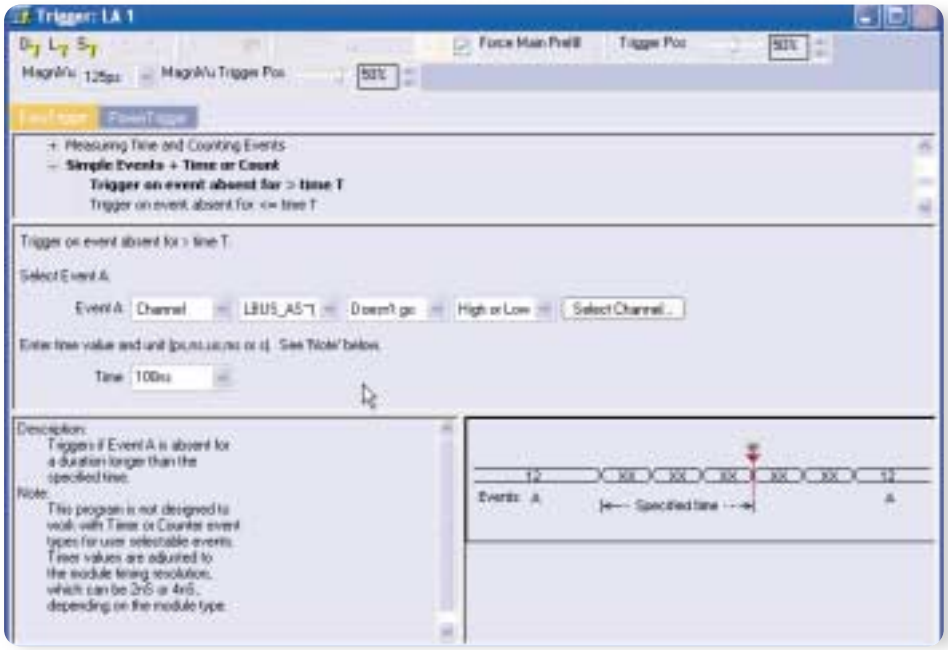
3.4.1 마이크로프로세서 부트 코드 디버깅

임베디드 시스템의 마이크로프로세서를 부팅하는 것이 문제가 되는 경우가 종종 있습니다. 알 수 없는 하드웨어가 알 수 없는 소프트웨어와 처음으로 결합됩니다. 일련의 오작동 등이 일상적일 수 있습니다! 임베디드 시스템은 일반적으로 대상 시스템에 장애를 일으키는 오작동으로부터 보호할 수단이 없다는 점에서 PC와 같은 비임베디드 시스템과는 다릅니다. 강력한 운영 체제에는 대개 시스템이 이상 작동하지 않도록 문제를 격리하는 다양한 메커니즘이 있지만 임베디드 시스템은 그렇지 못한 경우가 많습니다. 따라서 부트 코드가 손상되면 전체 시스템이 다운되고 문제를 디버깅하는 데 유용한 정보를 잃게 됩니다.

현재 검토 중인 프로토타입 예에서는 마이크로프로세서를 부팅하는 데 있어 문제를 만났습니다. 때때로 시스템에는 예기치 않게 장애가 발생합니다. 이런 장애의 근본 원인을 알 수 없었으므로 고장 메커니즘을 트리거하지 못했습니다. 크래시 버그를 디버깅할 때는 고장의 증상이나 발생해야 할 것이 발생하지 않는다면 이를 트리거하는 것이 더 쉬운 경우가 종종 있습니다. 이 경우에는 로컬 버스 신호 중 하나가 없는 것을 트리거합니다.

이 스트로브 신호가 충분히 자주 전환되지 않는다면 마이크로프로세서가 예상대로 작동하지 않은 것입니다.

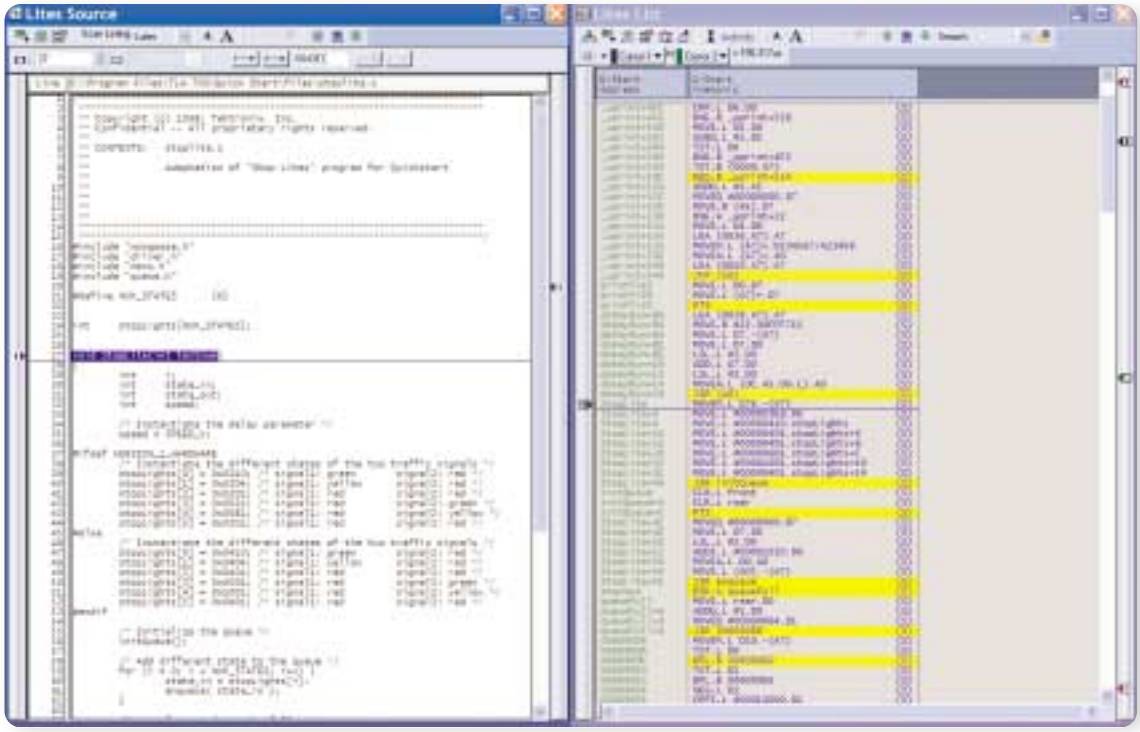
대안으로서 "워치독"이나 "하트비트" 펄스를 시스템에 바로 내장할 수 있습니다. 하트비트가 펄스 작동을 하는 한 시스템은 작동합니다. 하트비트가 멈추면 그 장애가 언제 중대한 문제가 되었는지 알게 됩니다.



▶ **그림26 –** *로직 애널라이저 시간 초과 트리거*

다행히도 로직 애널라이저가 "아무 것도" 트리거하지 않고 시스템 상태를 상세히 표시하도록 설정하기는 매우 쉽습니다. 활동이 없음을 트리거하는 것을 시간 초과 트리거링(Timeout Triggering)이라고 합니다. 로직 애널라이저가 라인이나 라인 그룹을 감시하도록 설정할 수 있습니다.

시간 주기 내에 아무런 일도 일어나지 않고 아무런 로직 변화도 없으면 로직 애널라이저가 트리거하게 됩니다. 그림 26은 TLA 시리즈 로직 애널라이저 EasyTrigger™메뉴의 시간 초과 트리거 화면을 나타낸 것입니다.



► **그림27 –** *소스 코드 창*

3.4.1.1 로직 애널라이저 소스 코드 창 사용

장애 증상이 캡처된 경우 그림 27에 나타낸 것과 같이 로직 애널라이저의 소스 창을 사용하여 캡처된 데이터를 소스 코드에 상관시켰습니다.

첫 번째 캡처된 명령에서 우리가 사용한 방법으로 작업하여 예상치 못한 인터럽트가 발생하기까지 프로세서가 예상대로 실행되는 것을 관찰했습니다. 이로 인해 코드가 초기화되지 않은 메모리 내 위치로 분기되었습니다. 잠시 후 프로세서가 작동을 멈추면서 다운되었습니다.

부팅 중에 발생하는 인터럽트를 차폐하는 간단한 하드웨어 변경으로 이 문제를 해결했습니다. 더 확실히 안전을 보장하기 위해 부팅 시퀀스에서 가능한 한 빨리 소프트웨어의 인터럽트를 차폐하도록 부팅 시퀀스를 재조정했습니다.

3.4.2 요약

하드웨어와 소프트웨어를 통합할 때 만나게 되는 문제 대부분은 장애의 원인을 정확하게 판단하지 못하는 데서 비롯되는 것입니다. 어떤 문제가 하드웨어 문제인지 소프트웨어 문제인지 알 수 있었습니까? 하드웨어 및 소프트웨어 실행을 모두 캡처할 수 있는 로직 분석기는 하드웨어/소프트웨어 통합에 관련된 문제를 중재하고 해결하는 데 이상적인 툴입니다.

전체 기능 집합을 검증하고 나면 바로 특성화 단계로 이동할 수 있습니다.

▶ ***특성화***

3.5 특성화

이제는 성능 마진, 한계 및 허용오차를 결정하기 위해 보드를 특성화해야 할 차례입니다. 이 정보는 제조 가능한 신뢰성 있는 설계가 되었음을 확인해주는 것입니다.

많은 경우에 있어 이런 파라미터는 다양한 온도, 습도, 고도, 진동 등의 조건에서 테스트를 통해 검증해야 합니다. 일반적으로 전력 소비량을 고려하고 많은 유형의 제품에 있어 배터리 수명도 한 가지 고려 요인이 됩니다. 장치의 EMI에 대한 내성뿐만 아니라 EMI를 방출하는 경향성도 평가 및 문서화되어야 합니다.

실시간 오실로스코프, 고대역폭 프로브 및 전문화된 계측 및 분석 소프트웨어와 같이 초기 설계 프로세스 단계에서 사용되는 대부분의 동일한 장비를 사용하여 완제품의 한계를 조사합니다. 프로세스의 이 지점에 이르렀을 때는 외부적 효과(부하, EMI 등)가 편차를 일으키지 않는 한 신호 무결성이 문제가 되어서는 안 됩니다.

특성화에는 세부적인 스트레스 테스트가 포함될 수 있습니다.

이 프로세스에서는 장치가 공급 전압의 변동에 영향을 받는데, 예컨대 이상적인 상황보다 열악한 환경에서 정상 작동하는 능력을 평가합니다.

또 다른 접근 방식은 결함 있는 신호를 고의적으로 입력해보는 방법입니다. 이에 따른 문제로는 여분의 진폭, 느린 상승 시간, 오버슈트와 같은 이상이 있을 수 있습니다. 이런 증상을 만들어내는 데는 텍트로닉스 AWG 시리즈와 같은 AWG(임의 파형 제너레이터)가 많이 쓰이는 툴입니다.

3.5.1 설계자 및 최종 사용자를 위한 사양

대부분의 전자 제품에는 다음의 두 가지 사양 수준이 있습니다.

– 최종 사용자에게 보증으로 제공되는 출판용 사양.
이것은 제품 설명서, 브로셔 등에 나오는 사양입니다.

– 비출판 표준 사양은 설계 프로세스를 지원하고 출판용 사양 이상의 기준을 제시하기 위한 사양입니다. 이런 높은 기준은 정상적으로 제작된 유닛이 출판용 사양의 범위 내에 들도록 보장하기 위해 더 높이 설정되어 있습니다.

두 사양 세트 모두 일반적으로 개발 프로젝트 전체의 지침으로 사용되는 엔지니어링 사양에 요약되어 있습니다. 최종적으로 실시하는 계측 작업을 통해 출판용과 비출판용을 망라하여 사양의 전 범위가 기준에 부합되는지 확인할 수 있습니다. 관련된 장치의 유형에 따라 이 계측 작업에는 Setup/Hold 및 기타 타이밍 허용 오차, 펄스 진폭 및 상승 시간, 클럭 주파수 안정성, 노이즈 특성, 지터, 특정 임피던스 한계 등과 같은 파라미터가 포함됩니다.

이 모든 계측은 본 입문서에서 이미 설명한 계측기의 능력 범위 이내에 충분히 포함되는 것입니다.

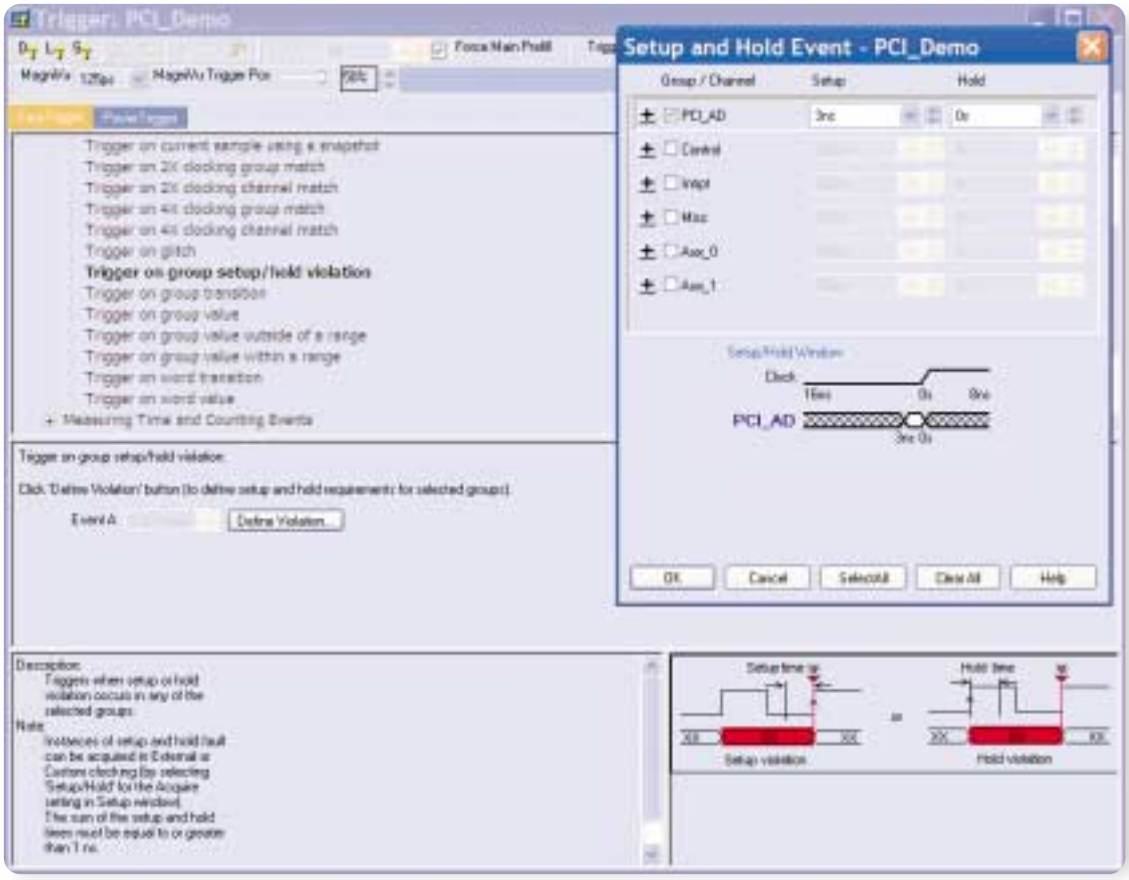
3.5.2 Setup 및 Hold 테스트

부적절한 타이밍 마진은 제조 라인이 다운되는 상황과 현장에서 발생하는 신뢰성 문제에 가장 큰 영향을 미치는 요인 중 하나입니다. Setup/Hold는 가장 중요한 동기 타이밍 파라미터 중 하나입니다. 이는 지금까지의 설계가 적합한 Setup/Hold 마진을 가지고 있는지 검증하는 과정이며 매우 중요한 의미입니다. 불행히도 어떤 설계에서 Setup/Hold 마진을 계측하고 문서화하는 작업은 꽤나 시간이 많이 걸리는 일이 될 수 있습니다.

오실로스코프를 사용하는 경우에는 Setup/Hold 시간을 계측할 수 있는 두 가지 방법이 있습니다. 첫째, 클럭과 단일 데이터 라인을 프로브할 수 있습니다. Setup/Hold에 위반되는 이벤트를 트리거하도록 오실로스코프를 설정할 수 있습니다. 이 방법의 분명한 단점은 한 번에 단 하나의 비트만 테스트할 수 있다는 점입니다.

스코프를 이용한 또 다른 접근 방식은 클럭 신호가 최대 3개의 데이터 라인을 추가로 캡처하여 작성할 필요가 있는 애플리케이션을 이용해 데이터를 사후 처리하여 타이밍 위반을 찾아내는 방법입니다. 이 접근 방식을 선택하면 소요 테스트 시간은 약간 줄어들지만 애플리케이션 작성을 위해 소중한 개발 시간을 할애해야 합니다.

실제로는 로직 애널라이저가 많은 신호의 Setup/Hold 마진을 한 번에 검증할 수 있는 이상적인 툴입니다. TLA 시리즈 로직 애널라이저는 모든 신호에 대해 한 번에 사용자가 정의한 Setup/Hold 위반을 트리거하고 표시함으로써 Setup/Hold 위반 검색을 자동화할 수 있습니다. 내장 PCI 버스를 예로 들어보겠습니다.



▶ **그림28 –** *Setup/Hold 트리거*

PCI_AD 라인에서는 3ns의 Setup 시간과 0s의 Hold 시간이 요구됩니다. 고밀도 로직 애널라이저 프로브의 장점을 살리면 로직 애널라이저를 PCI_AD 신호에 쉽게 연결할 수 있습니다. 그런 다음 로직 애널라이저는 그림 28에 나타낸 것과 같이 임의의 Setup/Hold 위반을 트리거하여 강조 표시하도록 구성됩니다. 저녁에 퇴근하면서 로직 애널라이저의 실행/중지(Run/Stop) 버튼을 누른 다음 귀가합니다. 그러면 로직 애널라이저가 어떤 라인에서든 위반 사항이 있는지 PCI_AD 버스를 밤새도록 모니터링합니다. 위반 사항이 발생하는 경우 로직 애널라이저는 그 문제를 트리거하고 캡처합니다. 다음 날 아침에 출근해서 로직 애널라이저를 살펴보면 테스트 상태를 알 수 있습니다. 로직 애널라이저가 트리거한 경우에는 위반이 발생했다는 것이고, 그렇지 않은 경우에는 Setup/Hold 위반이 없었다는 것입니다.

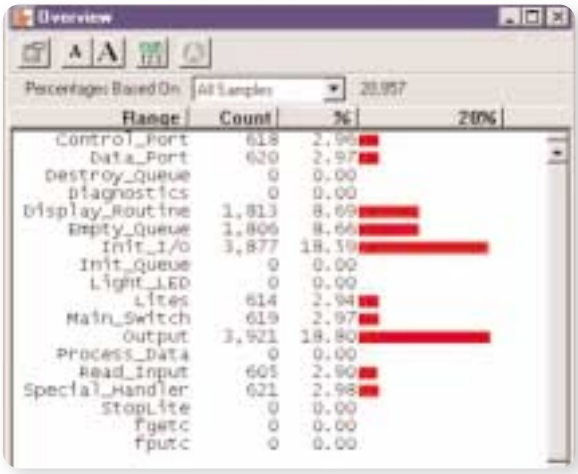
이 접근 방식이 전통적인 오실로스코프 접근 방식보다 훨씬 쉽습니다.

3.5.3 요약

특성화 단계에서 요구되는 작업의 대부분은 수많은 계측 작업을 반복적으로 해야 하는 것들입니다.

예를 들어 Setup/Hold 마진은 여러 가지 다른 작동 온도에서 평가되어야 합니다. 테스트 툴이 발달되면 이런 계측 작업이 대폭 간소화되고 소요 시간도 크게 줄어듭니다.

► ***시스템 테스트 및 최적화***



► **그림29 –** *범위 개요*

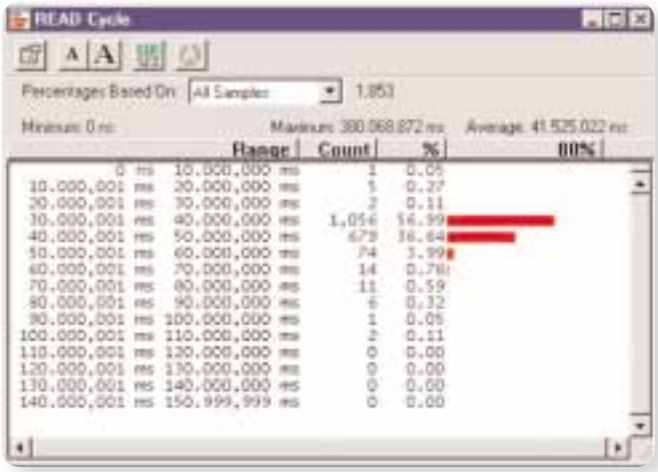
3.6 시스템 테스트 및 최적화

오늘날의 임베디드 소프트웨어 애플리케이션은 점점 더 커지고 복잡해져 전체 흐름의 "큰 그림"과 소프트웨어의 실행 시간을 알기가 어렵습니다. 때때로 임베디드 소프트웨어 개발자는 코드가 제대로 작동하긴 하지만 성능 목표에 미치지 못하는 경우에 부딪히곤 합니다. 이런 경우 일반적으로 프로젝트가 끝날 무렵에 수행되는 성능 최적화 작업이 필요한 것입니다. 자주 인용되는 경험철학에 따르면 "코드의 20%가 실행 시간의 80%를 차지한다"는 것이지만 여기서 큰 문제는 코드 중 20%가 어떤 부분인지 알아내는 것입니다. 제품의 Throughput 요구 사항을 충족시키기 위해 이런 종류의 조정 작업이 종종 필요합니다.

로직 애널라이저는 수백 개의 소프트웨어 모듈 중 어떤 것이 프로세서 실행 시간의 대부분을 소비하는지 표시하는 개요 정보를 제공할 수 있습니다. PA(성능 분석) 툴은 소프트웨어가 시간을 소비하는 곳을 표시해줍니다. 이 정보를 통해 어떤 루틴을 최적화하면 성능을 가장 크게 개선할 수 있을지 신속히 식별할 수 있습니다.

그림 29에 표시된 범위 개요 정보는 포착된 데이터에 대해 히스토그램 형식을 사용하여 각 범위에서의 히트 수를 그룹화하고 표시하는 사용자 정의 범위로 분석한 것입니다.

이 뷰는 어떤 소프트웨어 모듈이 가장 많은 실행 시간을 소비하는지 표시하는 데 매우 유용합니다.



▶ **그림30 –** *단일 이벤트*

인터럽트/예외 처리기와 같은 시간이 가장 중요한 루틴 역시 설계 목표에 부합되는지 검증할 수 있어야 합니다. TLA 시리즈 로직 애널라이저의 단일 이벤트(Single Event)는 로직 분석기의 타이머와 카운터를 사용하여 단일 루틴에 대한 실행 시간/카운트 범위를 표시하는 성능 분석 계측 모드입니다.

그림 30에 나타낸 결과 디스플레이는 단일 이벤트가 여러 차례의 실행을 통해 이벤트 실행에 걸리는 최단, 최장 및 평균 시간을 표시합니다.

소프트웨어 성능 분석 외에도 PCI 버스의 프로파일을 분석하여 어떤 PCI 에이전트가 대역폭을 소비 중인지 판단할 필요가 있습니다. 범위 개요 기능을 사용하면 이 작업을 쉽게 수행할 수 있습니다. 각 에이전트의 I/O 및 메모리 공간에 맞도록 간단히 범위를 정의합니다.

3.6.1 요약

시스템 테스트 단계에서는 설계가 모든 필수 작동 파라미터를 만족하는지 검증합니다. 응답 시간은 충분히 빠릅니까? 시스템이 모든 데이터를 효과적으로 관리하기에 충분한 대역폭을 가지고 있습니까? TLA 시리즈 로직 애널라이저의 성능 분석 툴을 이용해 시스템 병목 지점을 정확히 알 수 있고 하드웨어와 소프트웨어를 모두 최적화하는 데 필요한 정보도 얻을 수 있습니다.

4 요약 및 결론

본 임베디드 시스템에 대한 개발 과정을 거치면서 우리는 계측 작업을 보다 쉽게 완료할 수 있도록 하는 데에 초점을 맞추었습니다.

4.1 설계 단계

설계 단계에서 이루어지는 결정 사항은 기능을 평가하고 기능 및 신호 무결성 문제를 탐지 및 해결하고 시스템 조정 작업을 수행하는 데에 큰 영향을 미칩니다. 우리는 높은 생산성을 달성하기 위해 설계 단계 중 효과적인 개발 과정을 배웠고 문제소지의 가능성이 높은 부분을 고려해야 했으며 디버그를 위한 설계를 해야 한다는 것을 배웠습니다.

4.2 디버그 및 검증 단계

초기 파워 ON이 디버그 및 검증 프로세스의 6가지 단계 중에서 가장 이해하기 쉬운 단계입니다. 하지만 진정한 첫 단계는 시스템의 핵심 요소를 작동시키는 기본 기능 검증 단계입니다.

문제를 디버그하는 데 흔히 첫 번째 계측기로 사용되는 것이 실시간 DSO 또는 DPO입니다. 앞서 살펴본 바와 같이, 이들 실시간 플랫폼은 쉽게 테스트 포인트를 프로브하고 전원 장치 노이즈로부터 고속 신호에 이르는 다양한 파형을 신뢰성 있게 포착할 수 있게 해줍니다. 5개 이상의 신호들 사이의 관계를 가시화해야 하는 경우 또는 소프트웨어 실행을 검토할 필요가 있는 경우에는 로직 분석기가 이상적인 툴입니다.

점점 더 많은 기능 블록을 검증하고 디버그함에 따라 보드 관련 문제나 신호 무결성 문제에 직면하게 될 확률이 커집니다. TLA 시리즈 로직 애널라이저의 iCapture™멀티플렉서가 보드 관련 문제를 식별하는 프로세스를 간소화하고 글리치를 트리거 및 플래그 표시 함으로서 신호 무결성 문제의 탐지 시간을 단축할 수 있다는 점을 살펴봤습니다.

복잡한 하드웨어/소프트웨어 통합 문제를 해결하려면 시간이 많이 걸릴 수도 있습니다. 어떤 오류가 하드웨어 오류인지 소프트웨어 오류인지 판단하는 것이 공통된 문제입니다. 로직 분석기를 사용하여 하드웨어 이벤트를 소프트웨어 실행에 상관시켜 하드웨어/소프트웨어 통합 문제를 해결할 수 있다는 것을 배웠습니다.

또한 Setup/Hold 트리거와 같은 TLA 시리즈 로직 애널라이저의 특정 기능과 성능 분석 툴이 특성화 단계뿐만 아니라 시스템 테스트 및 최적화 단계도 단순화하는 데 도움이 된다는 점도 살펴봤습니다.

4.3 결론

본 입문서에서는 개발 주기 초기부터 디버그와 검증 프로세스를 고려해야 할 필요성을 역설했습니다. 계획을 제대로 세우지 않으면 실제로 언제 발생할지 모르고 반드시 제거한 후 다음 단계로 진행해야 하는 글리치, 변형 , 그리고 손상 등을 밝혀내는 데에 심각한 영향을 미칠 수 있습니다. 첨단 제품을 설계할 때 부딪히는 디버그 관련 난제는 첨단 오실로스코프, 로직 애널라이저 및 신호원과 함께 자동화된 포착 및 분석 툴의 도움으로 신속히 해결할 수 있습니다.

텍트로닉스 연락처:
동남아시아/대양주/파키스탄 (65) 6356 3900
오스트리아 +41 52 675 3777
발칸, 이스라엘, 남아프리카 및 다른 ISE 국가들 +41 52 675 3777
벨기에 07 81 60166
브라질 및 남미 55 (11) 3741-8360
캐나다 1 (800) 661-5625
중앙동유럽, 우크라이나 및 발트국 +41 52 675 3777
중앙 유럽 및 그리스 +41 52 675 3777
덴마크 +45 80 88 1401
핀란드 +41 52 675 3777
프랑스 및 북아프리카 +33 (0) 1 69 86 81 81
독일 +49 (221) 94 77 400
홍콩 (852) 2585-6688
인도 (91) 80-22275577
이태리 +39 (02) 25086 1
일본 81 (3) 6714-3010
룩셈부르크 +44(0) 1344 392400
멕시코, 중앙아메리카 및 카리브해 52 (55) 56666-333
중동, 아시아 및 북아프리카 +41 52 675 3777
네덜란드 090 02 021797
노르웨이 800 16098
중국 86 (10) 6235 1230
폴란드 +41 52 675 3777
포르투갈 80 08 12370
대한민국 82 (2) 528-5299
러시아 및 CIS 7 095 775 1064
남아프리카 +27 11 254 8360
스페인 (+34) 901 988 054
스웨덴 020 08 80371
스위스 +41 52 675 3777
대만 886 (2) 2722-9622
영국 및 아일랜드 +44 (0) 1344 392400
미국 1 (800) 426-2200
기타 지역: 1 (503) 627-7111
최종 갱신일 2005년 6월 15일

추가 정보

Tektronix는 최첨단 기술을 다루는 엔지니어를 지원하기 위해 응용 자료, 기술 문서 및 기타 리소스 등을 총 망라한 방대한 자료를 보유 관리하고 있으며 이를 계속 확장하고 있습니다. www.tektronix.com을 참조하십시오.

9/05 FLG/WOW 52K-18668-1

Tektronix
Enabling Innovation