

멀티 코어 – 디버깅을 위한 새로운 도전 과제

Jens Braunes, pls Development Tools

소개

다음은 한 사용자의 일화입니다.

임베디드 응용 프로그램을 위한 대규모 소프트웨어 프로젝트에서 한 함수가 변칙적인 모습을 보였습니다. 함수 실행 중에 변경되어서는 안 되는 변수가 변경되었습니다. 함수 자체에서 변수를 작성한 것은 아니지만, 다른 프로세스인 인터럽트 서비스 루틴 또는 DMA 컨트롤러로 인해 올바르게 작성된 것이 아니었습니다. 원인을 확인하기 위해 비용이 많이 드는 모니터링 방식을 통해 함수의 소스 코드를 배열했습니다. 그리고 변수를 덮어쓴 시점을 찾기 위해 행마다 그리고 인접 분기마다 모니터 코드를 삽입했습니다. 또한 쓰기 프로세스를 확인하기 위해 중요 시간에 수행된 모든 시스템 작업을 확인하는 것도 해결해야 할 문제였습니다. 이는 여러 개의 코어와 버스 그리고 버스 마스터 등이 탑재된 복잡한 시스템에서는 결코 만만한 작업이 아니었으며 문제를 확인하는 데 일주일도 걸렸습니다. 문제는 잘못 구성된 DMA 컨트롤러 때문으로, 회사에서 코어와 버스 트랜잭션에 대한 가시성을 제공하는 온칩 멀티 코어 디버그 시스템이나 하드웨어 플랫폼을 사용하지 않는 것이 큰 원인이었습니다. 프로젝트에 사용할 수 있는 하드웨어 디버그 시스템이 있었다면 즉시 문제를 분리할 수 있었을 것입니다. 이 일화를 통해서 알 수 있듯이, 강력한 하드웨어와 대규모 응용 프로그램을 동시에 갖춘 복잡한 시스템 개발 작업에서 디버깅을 위한 온시스템 스페닝 온칩 지원을 통해 많은 이점을 얻고 있습니다. 복잡한 SoC에서는 단일 코어를 관찰하고 제어하는 것만으로 부족합니다. SoC를 구성하는 여러 개의 코어와 버스 그리고 주변 장치의 상호 작용도 최적의 성능을 위해 시스템 동작을 프로파일링하거나 소프트웨어 문제를 탐지, 추적하며 제거하는 데 필요합니다.

멀티 코어 시스템에 적합한 온칩 디버그 시스템을 결정하려면 사용자 요구 사항을 파악하는 것이 중요합니다. 전체 기사를 보려면 ocpip.org 를 참조하십시오.