



개요

Open Core Protocol (OCP)는 시스템 레벨의 통합이라는 현대 과제에 대응하기 위해 생성된 최초의 오픈 라이선스 방식의 코어 중심 프로토콜이다. OCP는 칩 내 서브시스템의 통신을 위해 효율적이고, 버스(bus)에 의존하지 않으며, 구성 가능하고, 높은 확장성을 가진 인터페이스에 대해 포괄적으로 정의하고 있다. 폭 넓은 업계의 지원 및 협력을 바탕으로 OCP International Partnership (OCP-IP)은 현재 2.2 버전의 사양을 제공하고 있으며, 높은 성능의 다중처리, 동기 방식 및 단일 명령에 대한 다중 데이터 전송 등 날로 중요성을 더해가고 있는 분야에서의 그 기능을 확장시켜가고 있다. OCP 데이터 전송 모델은 파이프라인 명령-응답을 통해 단순한 명령-전송의 핸드셰이킹 방식에서 명령 범위를 넘어선 복잡한 동작까지 다양한 범위에 걸쳐 사용될 수 있다.

기존의 IP 코어를 쉽게 OCP와 연동할 수 있게 하는 한편, 새로운 구현에 있어서도 확장 기능을 최대한 이용할 수 있어, 설계자는 코어의 특정 데이터, 제어 및 테스트 구성을 포함한 기능과 신호를 선택하기만 하면 된다. OCP를 사용한 코어에 대한 정의는 완전한 시스템 통합 기술을 요약하여 재작업 없이 코어와 테스트 벤치를 재사용할 수 있도록 하였다. OCP는 코어 개발자 및 시스템 온 칩 (SoC)으로 집적화 하는 자에 대한 설계 책임을 명확히 기술할 뿐 아니라, 검증 기술자와 자동화 소프트웨어 역할을 명확히 분리하는 것에 대해

특징

OCP는 IP코어 재사용을 촉진하고, SoC 설계의 기간, 위험 및 제품 비용을 절감할 수 있게 한다. 상호 접속 토폴로지 및 기타 특정 응용분야의 집적에 관한 선택을 사전에 하지 않고, IP 코어 인터페이스 연결에 중점을 두고 있다.

- 시스템 구조 및 응용 영역에 의존하지 않는 IP 코어 생성이 가능
- 근접 코어 간 통신에 대해 기술
- 코어가 필요로 하는 기능만을 OCP 인터페이스 내에 구성함으로써 다이(die) 면적을 최적화함
- 타이밍 카테고리를 지정하여 코어 상호 연결성 확보
- 신속한 플러그 앤 플레이 방식의 IP 집적 촉진

장점

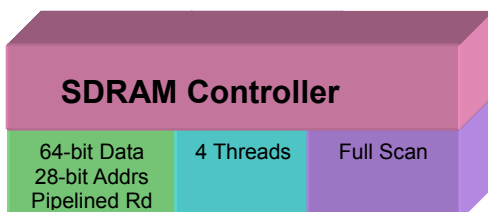
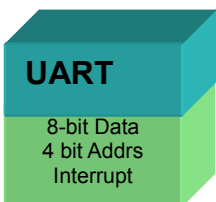
- 업계 전반에 걸쳐 지원되는 de facto 공개 표준
- 인터페이스 프로토콜의 (재)정의, 검증, 문서화 및 지원 등 연속적인 업무를 해소
- 새로운 코어 기능을 지원하기 위한 뛰어난 연동성
- 테스트 벤치 이동성으로 (재)검증 간소화

OCP-단일 프로토콜에 의해 모든 범위의 코어 신호 처리가 가능

성능

OCP는 시스템 중재 및 어드레스 맵 등을 제한함이 없이 모든 코어의 특성을 획득하게 된다.

- 적은 필수 신호에 광범위한 임의 신호를 포함
- 단일 방향 동기 신호로, 구현, 통합 및 타이밍 분석에 대한 간소화가 가능
- 어드레스와 데이터의 워드(Word) 폭 설정 가능
- 사이드밴드(sideband) 신호 처리를 위한 구조화 방법: 높은 레벨의 흐름 제어, 인터럽트, 전원 제어, 디바이스 구성 레지스터, 테스트 모드 등
- 처리량 향상을 위해 임의의 depth까지 전송의 파이프라인화 가능
- 효율성 향상을 위한 Burst 전송 옵션
- 복잡한 명령 실행을 완료하기 위해 다중 동시 전송시 thread 식별자를 사용
- 차별화된 서비스 품질을 위해 접속 식별자에 따라 End-to-end 전송 식별 방식을 제공
- 동기 프리미티브(Primitive)에 자동 테스트 셋, 연장 동기



전체 Open Protocol 사양서는 다음
홈페이지를 참고하십시오
www.ocpip.org

Open Core Protocol 주요 기능

기본적 OCP 상호 운용성

- 단방향 신호의 마스터/슬레이브 인터페이스
- OCP 블록의 상승 에지(Edge)에서 구동 및 샘플링
- 완전 동기화하여, 멀티 사이클 타이밍 패스가 없음 □
- 모든 신호가 Point-to-point 방식 사용(Clock과 reset 신호는 제외)
- 단순한 요청/응답 프로토콜 사용
 - 모든 clock cycle에서 데이터 전송 가능
 - 마스터 또는 슬레이브를 통해 전송 속도 제어 가능
- 코어 고유 데이터와 어드레스 버스 정의
 - Byte 및 non-byte를 지향하는 데이터 버스
 - 읽기 전용 및 쓰기 전용 인터페이스
 - 대역 내 데이터 태깅 (패리티(Parity), EDC 등)
 - 대역 내 명령어 태깅 (프로토콜 확장 등)
- 공지되지 않은 쓰기 기능을 포함한 파이프라인 및 차단 명령 제공
- 보안 액세스 권한은 모든 요청의 일부가 될 수 있습니다
- 코어의 특성, 인터페이스(신호, 타이밍, 구성) 및 성능을 나타내는 명확한 언어 포맷
- 타이밍 카테고리 사양
 - 레벨 2 - 최고 성능 인터페이스 타이밍
 - 레벨 1 - 쉬운 통합을 위한 기존 타이밍
 - 레벨 0 - 타이밍 규정이 없는 프로토콜 (시뮬레이션 /검증 툴 등에 특히 유용함)

간단한 확장 성능

- 완전한 처리를 위한 전송관련 Burst 그룹
- Burst 처리 지원
 - 순차적인 처리(규정 길이 또는 불특정 길이)
 - 스트리밍 (e.g: FIFO)
 - 특정 코어 (e.g: cache line)
 - 용량이 큰 burst를 분할 제어
 - 2차원 블록 시퀀스
- 파이프라인(데이터 앞부분의 cmd/add) 입력
- 정렬되거나 불규칙한 Byte enable 명령
- 읽기 및 쓰기 데이터 흐름 제어
- multi-address-segment 용 어드레스 공간 정의
- 데이터 phase 당 하나의 명령 또는 단일 요청/다중 데이터

축파대 확장 사용자 신호화

- 코어 고유의 사용자 정의 신호:
 - 시스템의 event 신호(e.g.: 인터럽트, 에러 메시지)
 - 2개의 동기 리셋 정의: 마스터에서 슬레이브 또는 슬레이브에서 마스터
 - 데이터 전송 조정(e.g.: 고도의 흐름 제어)
- 디버그와 테스트 인터페이스 확장
 - 구조화된 전체 또는 일부 스캔 테스트 환경 지원
 - 사전 설계된 하드 코어 또는 최종 사용자가 삽입한 소프트웨어 코어를 위한 내부 스캔 기술에 관련된 스캔
 - Clock 제어를 사용하여 여러 개의 Clock 도메인을 포함한 스캔 테스트와 디버그를 실현
 - IEEE 1149를 이용한 JTAG Test Access Port 장착 코어 지원
 - MIPS®, ARM®, TI® DSP, SPARC™ 등의 코어에 대한 JTAG- 및 Enhanced-JTAG 기반의 디버그

복합 확장 동시 지원

- Thread 식별자를 이용하여 다음을 실현:
 - Interleave burst transaction
 - Out-of-order transaction 완료
 - 차별화된 서비스 품질
- 철저한 Thread 흐름 제어 정의를 통한 중단현상 방지
- 연결 식별자를 통해 다음을 실현: □
 - End-to-end system initiator 식별
 - 시스템 목적에 따른 서비스 우선 관리
- 태깅은 순서가 바뀐 트랜잭션에 대해 공유된 흐름 제어를 제공합니다

CoreCreator®

OCP-IP는 OCP 호환 가능한 코어 구축, 시뮬레이션, 검증 및 패키징 작업을 자동화하기 위한 EDA 툴 CoreCreator를 회원에게 제공한다. 또한 CoreCreator 는 OCP 사양에 대한 호환성을 확인하기 위한 Protocol Checker를 제공한다. 코어 모델, 타이밍 파라미터, 합성 스크립트, 검증 세트 및 테스트 벡터를 통합하여 IP 코어 제품을 안전하게 컴포넌트화 할 수 있도록 하였다.

