

# 온칩 인터코넥트를 위한 최소 채널을 가지는 Point-to-Point 프로토콜

송진욱, 이재봉, 박인철  
한국과학기술원 전기및전자공학과  
e-mail : *jinoosong@gmail.com, jblee@ics.kaist.ac.kr, icpark@ee.kaist.ac.kr*

## Point-to-Point Protocol Having Minimum Channels for On-Chip Interconnect

Jinook Song, Jae-Bong Lee and In-Cheol Park  
Department of Electrical Engineering  
KAIST

### Abstract

This paper proposes a simplified point-to-point protocol for on-chip interconnect. The proposed point-to-point protocol has 2 channels and distinguishes two phases in a channel. We implement and synthesize a system having four masters and two slaves.

### I. 서론

Customer의 요구에 만족하기 위해 Mobile Handset의 기능이 다양하고 복잡해지고 있다. 이는 많은 수의 Intellectual Properties (IPs)를 하나의 칩에 집적해야 하며 SoC가 상당히 복잡해지고 있다. 여러 IP를 묶는 방법으로 전통적으로 AHB [1]와 같은 버스를 사용한다. 하지만 버스의 특성상, 하나의 마스터만이 버스를 차지하고 있을 수 있기 때문에 많은 IP가 동시에 동작해야 하는 시스템일 경우 Customer가 요구하는 Throughput을 맞추기가 어렵다.

기존 버스의 개념을 개선하여 Point-to-Point (P2P) 형태의 온칩 인터코넥트가 상업적으로 사용하고 있다. 대표적으로 AXI [2]가 있다. SoC를 구성하는 마스터와 슬레이브가 각각 P2P 형태의 온칩 인터코넥트를 가지며 마스터와 슬레이브의 인터코넥트는 서로

어떠한 방법으로 구매 받지 않고 구현될 수 있다. 다시말해 그림 1. 과 같이 마스터와 슬레이브 사이의 연결은 Blackbox와 다름이 없다. Blackbox를 구현하는 방법은 여러 가지가 가능하다. Network 방식, 버스 방식 등이 가능하다. 따라서 P2P 형태의 온칩 인터코넥트는 두 개 이상의 IP가 동시에 동작 가능하다.

P2P 프로토콜은 기존의 버스에는 없는 채널을 가지고 있으며 각 채널은 서로 독립적으로 동작한다. 상업적으로 널리 쓰이고 있는 AXI는 5개의 채널을 가지고 있으며 Wire의 수가 매우 많은 단점이 있다. SoC에 있는 하나의 IP를 연결함에 있어서 많은 수의 Wire가 필요하게 되면 여러 IP를 집적하는데 어려움이 생긴다. 따라서 본 논문에서는 기존에 정의되어 있는 AXI 보다 훨씬 적은 수의 Wire를 필요로 하는 P2P 프로토콜을 제안하며 간단한 시스템에 대하여 구현하여 보았다.

본 논문은 다음과 같이 구성한다. Section II는 기존 P2P 프로토콜을 검토하고 Section III는 제안하는 P2P 프로토콜을 설명한다. Section IV는 제안하는 P2P를 구현한 결과를 보여주며 결론은 Section V에서 맺는다.

### II. 기존 P2P 프로토콜

하나의 마스터와 슬레이브가 P2P를 이용해 연결되기 위해서는 최소한 두 개의 채널이 필요하다. 마스터가 정보를 요구하거나 슬레이브의 반응을 요구하는 Request 채널과 마스터의 Request 채널에 대응하는 슬

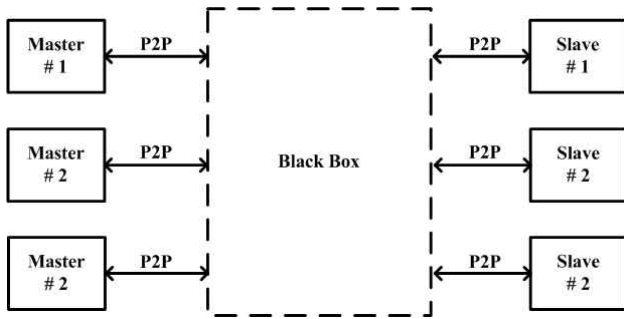


그림 1. P2P 형태로 연결된 온칩 인터코넥트

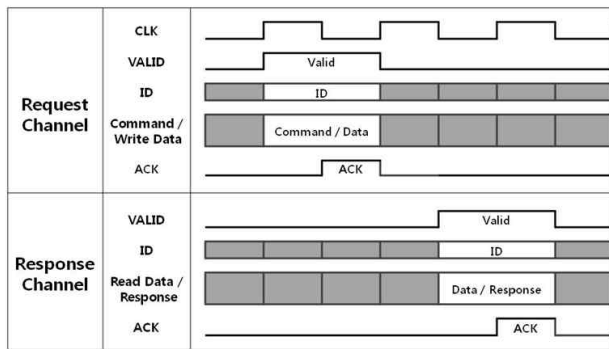


그림 2. Request 채널과 Response 채널

레이브의 Response 채널이 있다. 채널은 그림 2.와 같이 한 Transaction에 대한 ID와 Handshake 방식의 통신을 위한 Valid 신호와 ACK 신호가 있다. ID는 서로 다른 Transaction을 구분하기 위해 존재한다. Request 채널과 Response 채널의 ID는 서로 공유되기 때문에 마스터가 한 ID와 함께 Request 채널을 통해 슬레이브의 정보를 요구한다면 슬레이브는 동일한 ID로 Response 채널을 통해 마스터에 응답해야 한다.

### 2.1 AXI

AXI는 ARM 사의 AMBA Advanced eXtensible Interface의 준말로써 AHB 버스 프로토콜을 발전 시켜서 다섯 채널로 구성된다. 5채널은 쓰기 주소 채널, 쓰기 데이터 채널, 읽기 주소 채널, 읽기 데이터 채널, Response 채널로 구성한다. Handshake 방식으로 채널 내에서 Transfer가 이루어지기 때문에 Register Slicing이 가능하다. 각 채널은 ID를 서로 공유하고 있기 때문에 읽기와 쓰기에 대해서 모두 Out-of-Order Transaction이 가능하다. 여러 Transaction이 Out-of-order로 완료 될수 있지만 하나의 Transaction 내에서는 In-Order로 완료되어야 한다. 읽기와 쓰기에 대

해 모두 Out-of-Order Transaction이 되는 것은 IP간 Data Transfer가 매우 효율적으로 이루어진다고 생각할 수 있다. 하지만 Out-of-Order Transaction이 가능한 마스터와 슬레이브를 설계하는 것은 쉽지 않다. 이는 기 개발된 IP들을 Reuse하는 관점에서 비효율적일 수 있다. 뿐만 아니라 각 IP는 204bit나 되는 많은 수의 온칩 인터코넥트를 위한 포트가 필요하다.

### 2.2 OCP

OCP는 Open Core Protocol로서 [3] 여러 회사에서 차용하고 있는 P2P 프로토콜이다. OCP는 주소 채널, 쓰기 데이터 채널, 읽기 데이터 채널 등 총 세 채널로 이루어져 있다. AXI와 마찬가지로 Handshake 방식으로 Transfer가 이루어지기 때문에 Register Slicing이 가능하고 Transaction ID가 부여되기 때문에 Out-of-order Transaction이 가능하다. Burst Transfer가 가능한데 Increment, Wrapping, Packing, Streaming되는 모드를 지원한다. Increment와 Wrapping은 주소값을 설정하는 것으로 시작 주소값만 받고 매사이클 증가하는 모드 (Increment), 특정 범위 내에서 주소값이 Cyclic되는 모드 (Wrapping)이다. Packing은 Transfer하는 데이터 사이즈가 작고 주어진 데이터 채널의 bitwidth가 클 경우 두 개 이상의 데이터를 묶어서 보내는 방식이다. Streaming은 주소값이 증가하지 않고 한 위치에 데이터를 읽기 쓰기를 반복하는 것이다. 이는 해당 주소 위치에 메모리가 아닌 FIFO로 이루어져 있으면 효율적인 Transfer일 수 있다. Packing은 구현하는 시스템의 특성에 직접적으로 관련있는 바이기 때문에 프로토콜로 구현하는 것은 적합하지 않다.

### 2.3 VCI

VCI는 여러 회사가 Alliance를 맺고 정의한 온칩 인터코넥트로서 [4] Request 채널과 Response채널로 구성된다. Request 채널은 주소, 쓰기 데이터를 포함하고 Response 채널은 읽기 데이터를 포함한다. VCI 역시 Handshake 방식으로 Transfer를 하기 때문에 Register Slicing이 가능하다. 또한 Transaction 마다 ID를 부여하기 때문에 Out-of-order가 가능함에도 불구하고 Request 채널은 In-order이고 Response 채널은 Out-of-Order로 Transfer가 가능하다.

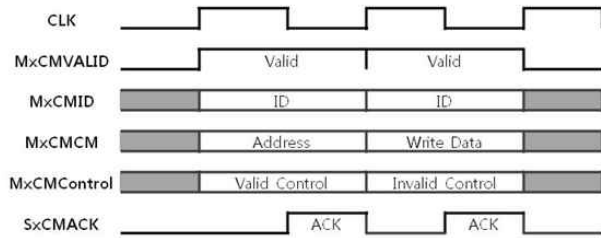


그림 3. 제안하는 Command 채널

### III. 제안하는 P2P 프로토콜

Section II에서 살펴본 바와 같이 상용 P2P 프로토콜의 공통점은 Handshake 방식의 Transfer와 Transaction 별로 ID를 부여하는 점이다. AXI의 다섯 채널로 정의함에 따라 IP 간에 연결하기 위해 많은 포트가 필요하게 되고 그림 1의 Blackbox의 구현이 복잡하게 된다. 따라서 Blackbox의 복잡도를 낮추기 위해서는 OCP와 VCI와 같이 세 채널 또는 두 채널로 낮추게 되면 복잡도가 낮아질 것이다. 하지만 VCI의 두 채널로 정의한 방식은 실제로 OCP의 세 채널에 비해 포트수가 크게 줄지 않는다. 왜냐하면 쓰기 데이터 채널은 쓰기 데이터 비트 수, ID 비트수, VALID, ACK 로만 이루어져있기 때문에 ID 비트수가 4 라면 총 6비트가 줄어드는 효과만 있을 뿐이다. 다시 말해 VCI의 프로토콜을 이용하여도 그림 1의 Blackbox 복잡도는 OCP를 이용한 것에 비해 줄어들지 않는다. 따라서 본 논문에서는 Blackbox의 복잡도를 줄이기 위해 적은수의 포트를 갖는 P2P 프로토콜을 제안한다.

두 채널 중 Request 채널을 주소 및 제어 Phase와 쓰기 데이터 Phase로 나누는 방식을 그림 3과 같이 제안한다. 이와 같은 방식을 갖는 Request 채널을 본 논문에서는 Command 채널이라 부르겠다. Command 채널에서 주소 또는 쓰기 데이터가 Command 이다. Command에 필요한 비트수는 일반적으로 표현하자면 주소의 비트 수와 쓰기 데이터의 비트 수 중 큰 값과 같다. 특히 주소의 비트수와 쓰기 데이터 비트수가 서로 동일할 때 Command 채널의 사용율이 가장 높을 것이다. 일반적인 SoC에서 주소에 필요한 비트수와 쓰기 데이터에 필요한 비트수가 32 bit 로 동일함을 고려하면 주소와 쓰기 데이터를 공통된 포트에 Phase를 나누어 Transaction이 이루어지는 것이 채널을 효율적으로 사용하는 것이다.

	Signals	Description
Command Channel	MxCMVALID	High: Valid
	MxCMID	Ex) 4 bits
	MxCMLK	High: Lock
	MxCMWT	High: Write
	MxCMSZ[1:0]	00/01/10: 8/16/32 bits
	MxCMRB[3:0]	Required Beats
	MxCMMOD[1:0]	00/01/10: Inc/Wrp/Fix
	MxCMCM	32 bits Address Phase and Data Phase
SxCMACK	High: ACK	
Response Channel	SxRVALID	High: Valid
	SxRID	Ex) 4 bits
	SxRDT	XXXD/XXDD/DDDD Ex)32 bits
	SxRRSP	01/00: Read OK/Error 11/10: Write OK/Error
	MxRACK	High: ACK

그림 4. 제안하는 P2P 프로토콜의 포트 정의

제안하는 P2P 프로토콜은 Command 채널과 Response 채널로 구성되며, 데이터와 주소 비트수를 32bit, ID의 비트수를 4bit로 가정하였을 때 89bit가 필요하며 정리하면 그림 4와 같다. 이는 AXI에 필요한 비트 수에 비해 60% 줄어든 수치이며 OCP에 비해 30% 줄어든 크기이다. 따라서 Blackbox를 구현함에 있어서 기존의 AXI, VCI, OCP에 비해 그 복잡도가 줄어들 것이다.

### IV. 구현

제안하는 P2P 프로토콜을 이용하여 4개의 마스터 2개의 슬레이브를 포함하는 시스템을 설계하였다. Core-A 32-bit Embedded Processor를 사용하였다[5]. Core-A는 Harvard 아키텍처를 가정하기 때문에 Instruction을 위한 포트와 Data를 위한 포트가 각각 나와 있다. 그래서 Core-A의 Instruction 과 Data 포트를 각각 하나의 Master로 가정하였다. 그리고 구현한 시스템은 두 개의 Core-A를 포함하는 멀티 컴퓨터 시스템이다. 구현한 시스템은 그림 5와 같다.

Core-A와 SRAM의 포트는 P2P 프로토콜과 맞지 않기 때문에 P2P 프로토콜에 맞는 Wrapper를 디자인하였다. 또한 그림 1의 Blackbox는 그림 5에서 Switch를 기본으로 한 Full connection을 구현하였다. 모든 Switch와 Wrapper는 Register를 포함하고 있어서 Core-A가 SRAM에 데이터를 Write하기위해서 적어도

4사이클이 걸린다. Core-A와 SRAM을 제외한 모든 Interconnection에 대해 0.18um에서 3ns로 합성하였고 26k Gate 수로 합성되었다.

재산권 창출 촉진사업의 지원과 반도체설계교육센터 (IDEC)의 도움을 받아 이루어 졌습니다.

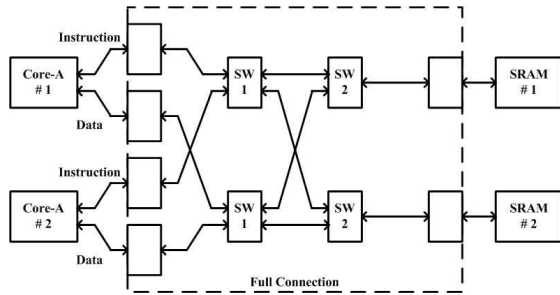


그림 5. 제안하는 P2P 프로토콜을 이용한 시스템

#### IV. 결론 및 향후 연구 방향

많은 수의 IP를 집적해야하는 임베디드 시스템을 위한 SoC를 구현함에 있어서 P2P 프로토콜은 매우 중요하다. 기존의 프로토콜은 많은 수의 포트가 필요하기 때문에 실제로 구현할 때 매우 복잡한 Connection이 나타난다. 제안하는 P2P 프로토콜은 Command 채널을 정의하고 채널 내에서 주소 및 제어 Phase와 쓰기 데이터 Phase를 구분하여 포트의 활용율을 높였다. 제안하는 프로토콜은 기존의 프로토콜보다 간단하고 Connection을 구현함에 있어서 적은 면적으로 구현 가능하다. 앞으로 다양한 애플리케이션에 대해 제안하는 온칩 인터코넥트와 상용 온칩 인터코넥트와 퍼포먼스를 비교하도록 하겠다.

#### 참고문헌

- [1] "AMBA Specification," Rev. 2.0, Axis. Sunnyvale, CA, 1999.
- [2] "AMBA AXI Protocol Specification," Axis. Sunnyvale, CA, 2003.
- [3] "Open Core Protocol Specification," OCP International Partnership, 2001.
- [4] "Virtual Component Interface Standard," VSI Alliance, 2000..
- [5] Core-A, <http://core-a.co.kr>

#### Acknowledgement

본 연구는 특허청 반도체 사무국의 핵심반도체 설계