

32-bit RISC 마이크로프로세서를 위한 버스 설계 및 구현

양 동 훈, 콧 승 호, 이 문 기

연세대학교 전기전자공학과

전화 : 02-2123-4731 / 핸드폰 : 016-282-0257

Design and Implementation of Bus for 32-bit RISC Microprocessor

Dong Hun Yang, Seong Ho Kwak, Moon Key Lee

Dept. of Electrical and Electronic Engineering, Yonsei University

E-mail : ysydh@spark.yonsei.ac.kr

Abstract

This paper purpose design and implementation of system bus for the effective interconnection between peripheral device and 32-bit microprocessor.

The designed system bus support general bus protocol. Also, it is optimized for 32-bit microprocessor. It is divided into two system : high performance system bus and peripheral system bus.

I. 서론

최근에 개발되는 마이크로프로세서는 다양한 용도로 사용되고 있다. 시스템의 제어에서부터 PDA 및 멀티미디어 관련 제품에 이르기까지 다양한 기능을 지원하고 있다. 이러한 마이크로프로세서가 다양한 기능을 제공하기 위해서는 주변기기를 통해 그 기능을 확장할 필요가 있다. 마이크로프로세서와 주변장치와의 효과적인 연결은 전체 성능과 확장성을 높이는데 있어서 결정적인 역할을 한다.

마이크로프로세서에는 On-chip 메모리, 외부 메모리 인터페이스, DMA, LCD 컨트롤러 같은 고성능을 필요로 하는 주변기와 타이머, UART, 병렬 인터페이스, 스마트카드 인터페이스, 인터럽트 제어기 같은 고성능을 필요로 하지 않는 주변 기기들을 연결할 수 있다. 이러한 주변기기들을 마이크로프로세서가 효율적으로 운용하기 위해서는 기능에 따른 성능을 만족시키는 동

시에 저전력 또한 고려해야 한다.

본 논문에서는 RISC 마이크로프로세서의 주변장치의 효과적인 연결과 확장성을 위한 버스 시스템 구현 및 설계를 소개 한다. 본 버스는 일반적인 버스 프로토콜을 지원하며 또한 SPK610¹⁾ 32bit RISC 마이크로프로세서에 최적화 하여 설계하였다. 버스 시스템을 고성능 시스템 버스와 주변기기 시스템 버스로 구성하였으며, 이를 연결하는 브리지와 Arbitration 동작을 하는 아비터를 설계하였다[1,2,3].

II 장에서는 고성능 시스템 버스(High-performance System Bus : HSB)의 신호와 동작에 대해서 알아보고, III 장에서는 주변기기 시스템 버스(Peripheral System Bus : PSB)의 신호와 동작에 대해서 알아본다. IV 장에서는 고성능 시스템 버스와 주변기기 시스템 버스의 인터페이스를 제어하는 아비터와 디코더에 대해서 설명하고, V 장에서는 HSB와 PSB를 연결하는 브리지에 대해서 설명한다.

II. 고성능 시스템 버스(HSB)

HSB는 고성능을 요구하는 시스템을 위한 시스템 버스이다. 빠른 데이터의 전송 요구를 충족시킬 수 있는 시스템 버스로 Burst 전송 및 Split 전송을 지원한다.

Wait State를 두어서 HSB의 빠른 버스 응답 요구와

1) 연세대학교 VLSI&CAD 연구실에서 설계한 32bit RISC 마이크로프로세서

주변기기의 느린 응답에 대한 시간적인 차이로 인한 문제를 해결하였다.

Burst 전송은 데이터를 연속으로 전송하는 것으로 최대 1024비트까지 지원한다. Burst 전송은 캐시 또는 DMA의 채널에서 데이터 전송 시 사용될 수 있다.

Split 전송은 버스 슬레이브가 Split transaction을 지원하는 경우, 응답이 느린 주변기기에서 일부 데이터를 슬레이브 처리가 완료되는 시점에 전달하는 것으로 중간에 다른 버스 마스터를 Arbitration 할 수 있다. 이를 통해 데이터 전송의 완료를 위해 기다리는 시간의 낭비를 줄일 수 있게 된다.

HSB는 파이프라인 방식의 전송을 지원함으로써 하나의 데이터가 전송되는 과정에도 다음 데이터의 전송을 위한 어드레스가 인가 될 수 있다. 물론 여러 개의 데이터를 연속적으로 버스에 실을 경우 어드레스의 인가로 인한 추가 클럭이 필요 없다.

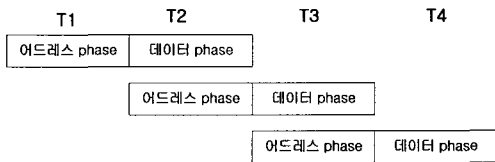


그림 1. 파이프라인 방식의 전송

III. 주변기기 시스템 버스(PSB)

PSB는 저전력에 대한 버스 프로토콜 수준의 설계가 고려되었으며 인터페이스가 단순하다. 파이프라인 전송 방식은 지원하지 않지만, 낮은 데이터 전송 성능만으로도 충분한 주변기기와의 연결이 가능하다.

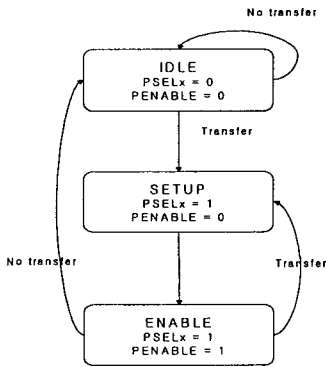


그림 2. PSB의 State 다이어그램

PSB 동작은 그림 2와 같이 IDLE, SETUP,

ENABLE 세 단계의 State를 갖는다. PSB가 사용되지 않는 경우에는 IDLE State에 있게 된다. 이 때, PSEL과 PENABLE이 Low이다. SETUP은 이 다음 단계로서 데이터 전송이 필요할 때, PSEL이 High가 된다. 마지막 단계로 PENABLE이 High가 되면 ENABLE State로 넘어간다.

PSB는 Write와 Read 두개의 전송 모드만을 제공한다. Write 데이터는 PWRITE가 High 일 때, SETUP State에서 버스에 실리게 되며, Read 데이터는 PWRITE가 Low 일 때, ENABLE State에서 버스에 실리게 된다. 그림 3과 4는 PSB의 Write와 Read 전송 타이밍을 보여주고 있다.

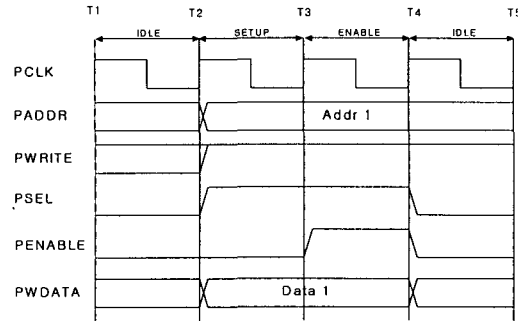


그림 3. PSB의 쓰기 전송 타이밍

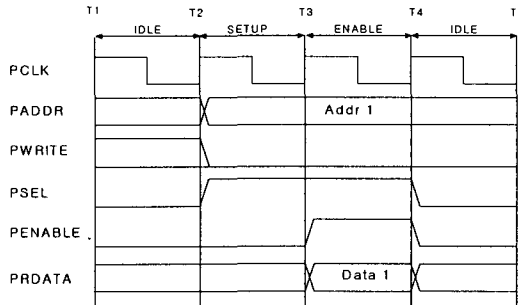


그림 4. PSB의 읽기 전송 타이밍

PSB는 느린 주변기기를 위한 버스이기 때문에 기능의 단순화를 통한 버스 프로토콜 수준에서 저전력을 구현할 수 있다. PSB는 PADDR, PWRITE의 신호를 다음 신호가 들어오기 전까지 Transition을 하지 않으므로써 전력의 소모를 줄인다. 이것이 가능한 이유는 PSB가 ENABLE 상태에서 실제 Write, Read 동작이 되고 그 전단계인 SETUP에서 PADDR과 PWRITE 신호가 바뀌기 때문이다. 따라서 위의 transition으로 신호의 클리치나 conflict는 발생하지 않는다.

IV. 버스 인터페이스 제어기

버스 인터페이스의 제어기는 아비터와 디코더로 구성된다. HSB는 최대 16개의 버스 마스터와 연결되므로 여러 마스터 중 버스를 사용할 수 있는 마스터를 선택하기 위해서 아비터가 필요하다. 디코더는 여러 개의 버스 슬레이브 중 하나를 선택하는 것으로 주어진 주소에 따라 버스 슬레이브를 결정한다.

디코더에 의한 어드레스의 디코딩은 메모리 뱅크를 통한 어드레스 범위 결정 값에 따른 선택 신호를 내보내는 단순한 방식을 채택하였다.

HSB는 아비터와 디코더의 제어에 따른 여러 개의 Multiplexer로 연결된다. 그림 5는 Multiplexer로 연결된 버스 인터페이스의 제어기와 내부 구조를 보여주고 있다.

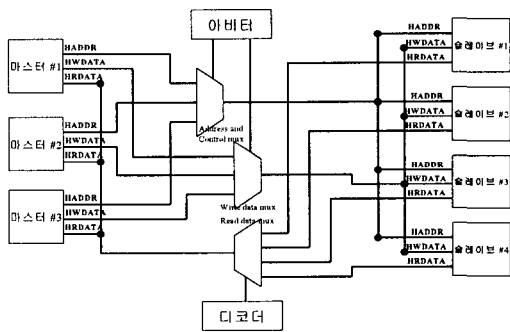


그림 5. 버스 인터페이스의 내부 구조

V. PSB 브리지

HSB와 PSB와의 연결을 위해서 PSB 브리지가 필요하다. PSB 브리지는 HSB와 PSB의 서로 다른 버스 프로토콜과 인터페이스를 중계하는 역할과 PSB 슬레이브에 대한 디코더 역할을 한다. 이를 위해 PSB 브리지는 HSB의 슬레이브로써 동작하여야 한다.

그림 6과 7은 HSB와 PSB사이에서의 Read 전송과 Write 전송의 타이밍이다.

Read 시에는 브리지가 어드레스를 latch 하여 다음 클럭에 PSB 슬레이브에 대한 어드레스를 전송한다. 그러면 이 어드레스에 따라 PSB Read 동작을 수행하고, Read 된 데이터는 바로 HSB에 전달이 된다.

Write 시에는 브리지가 어드레스와 write할 데이터를 holding 한다. 그리고 난 후 다음 클럭에 PSB의 Write 동작을 수행한다.

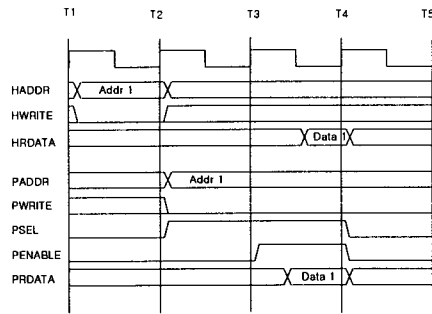


그림 6. PSB 브리지의 읽기 전송 타이밍

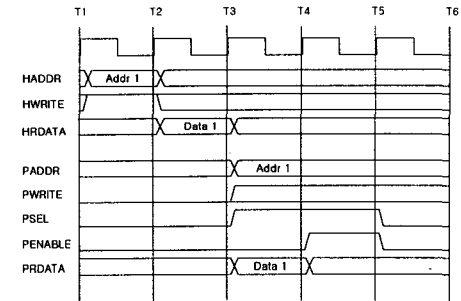


그림 7. PSB 브리지의 쓰기 전송 타이밍

VI. 설계 및 검증

HDL로 설계한 시스템 버스를 검증한 환경은 그림 8과 같다.

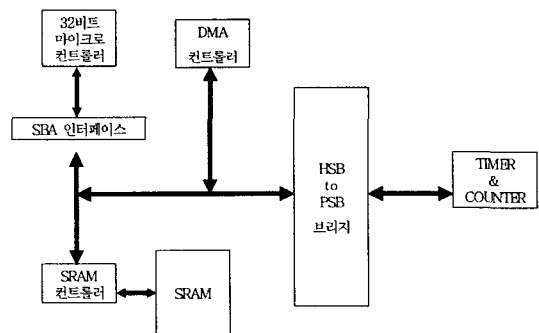


그림 10. 시스템 버스 검증 환경

HSB 버스는 32비트 어드레스 버스와 32비트 데이터 버스로 구성되었으며 마이크로 컨트롤러 코어와 SRAM 제어기 및 SRAM, DMA 컨트롤러 등을 연결

하였다. HSB 마스터를 2개를 두어서 Arbitration 동작이 잘 이루어지는지 검증하였다. 또한 PSB 브리지와 타이머, 카운터를 연결하여 PSB의 Write, Read 동작과 브리지의 여러 동작들을 검증하였다.

설계한 버스 검증을 위해서 SPK610 32bit RISC 마이크로프로세서를 이용했는데, 이를 통해 설계한 버스가 특정 코어에 국한된 시스템 버스가 아닌 독립적인 버스 프로토콜과 인터페이스를 갖는 버스 시스템이 되도록 하였다.

본 버스 시스템을 0.35 μ m CMOS 표준 라이브러리를 이용해 합성하였을 때, 약 7000 게이트가 나왔고, 동작 주파수는 최대 100Mhz였다.

VII. 결론

본 논문은 마이크로프로세서를 위한 시스템 버스의 설계를 주제로 하였다. 이 연구의 주된 초점은 32비트 마이크로프로세서에 가장 적합한 시스템 버스를 설계하는 것이었다. 이를 위해 고성능을 요구하는 시스템을 위한 버스와 인터페이스의 단순화를 통한 일반적인 주변기기와의 연결을 위한 버스를 나누어 설계하였다.

시스템 버스의 인터페이스 제어는 multiplexer를 기반으로 하여 아비터와 디코더를 설계하였고, HSB와 PSB의 서로 다른 프로토콜의 중계를 위해서 PSB 브리지를 설계하였다.

참고문헌

- [1] David Flynn, "AMBA :Enabling reusable on-chip designs", IEEE Micro, 1997
- [2] David Flynn, "Modular bus design supports on-chip testability", IEEE Colloquium: Systems Design for Testability, No. 1995/083, paper 2.
- [3] "AMBA Specification Rev 2.0", <http://www.arm.com>