

# 메모리 셀렉터를 이용한 AHB1-AHB2 다중버스 아키텍처 구조 구현

이근환, 이국표, 윤영섭  
인하대학교 전자공학과

e-mail : *lkkhhj@nate.com, kplee@yjc.ac.kr, ysyoon@inha.ac.kr*

## Implementation of AHB1-AHB2 Multi-Bus Architecture Using Memory Selector

Keun-Hwan Lee, Kook-Pyo Lee, Yung-Sup Yoon  
Dept. of Electronic Engineering  
Inha University

### Abstract

In this paper, several cases of multi-shared bus architecture are discussed and in order to decrease the bridge latency, the architecture introducing a memory decoder is proposed. Finally, a LCD controller using DMA master is integrated in this bus architecture that is verified due to RTL simulation and FPGA board test. DMA, LCD line buffer and SDRAM controller are normally operated in the timing simulation using ModelSim tool, and the LCD image is confirmed in the real FPGA board containing LCD panel.

### I. 서론

단일 칩으로 여러 기능을 동시에 사용하고자 하는 사용자의 요구를 충족시켜주기 위해, 여러 프로세서, 각종 용도의 DMA, DSP 등을 하나의 칩에 구현하려는 노력이 지속적으로 진행되고 있다[1][2].

그러나 사용자의 다양한 요구의 충족과 여러 시스템이 단일 SoC 칩으로 집적화되는 발전 속도에 따라, 마스터의 개수와 종류가 다양해 질 수밖에 없다.

결국, 다중버스에 여러 마스터들이 구성되어지게 되며, 복잡도는 크게 증대하게 된다[3][4].

본 연구에서는 버스 아키텍처의 구조에 따른 장단점을 분석해 보고, 3단 이상의 다중버스 아키텍처를 설계하여 특징을 분석하였다.

### II. 본론

#### 2.1 다중버스 아키텍처 구성

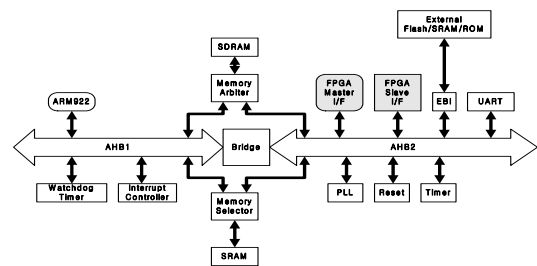


그림 1. 수정된 다중 AHB 버스 아키텍처

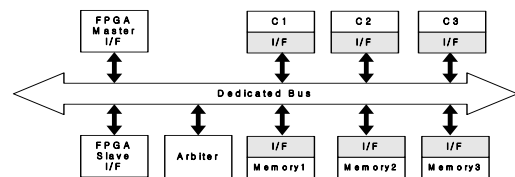


그림 2. 사용자가 추가로 설계할 마스터와 슬레이브를 위한 전용 버스 아키텍처

본 연구에서는 브릿지를 통과에 의한 성능저하를 개선하기 위하여 공용버스 아키텍처에 메모리 셀렉터를 도입하여 그림 1에 나타내었다.

버스 아키텍처에서 브릿지를 통과할 때 발생하는 레이턴시를 줄일 수 있으며, 버스1과 버스2에서 동시에 메모리를 사용할 경우, 메모리 셀렉터에 의해 중재된다. 그림 1에 표현된 것처럼 본 연구의 버스 아키텍처는 AHB1 버스에 ARM922 프로세서, watchdog 타이머, 인터럽트 컨트롤러 등을 구성하였으며, AHB2 버스에는 고속을 요구하지 않는 PLL, 타이머, UART, 외부버스 인터페이스 EBI, FPGA 인터페이스 등을 구성하였다. 그리고 AHB1, AHB2 버스에 대해서 메모리 셀렉터에 의해 공용 SDRAM, SRAM 메모리가 조절되고 있다. FPGA 인터페이스 블록에 의해서 추가적으로 버스를 구성할 수 있는데, 이는 그림2에 나타나 있다. 마스터 컴포넌트 C1, C2, C3를 구성할 수 있으며, 메모리, 슬레이브 블록, 아비터 블록도 추가할 수 있다. SoC 버스 아키텍처에서 필수적으로 사용하는 기본 블록들은 AHB1, AHB2 버스를 통해 구성하고, 추가적인 버스구조를 통해 새로운 블록을 추가할 수 있도록 해서, 버스구조를 조직적으로 만들었을 뿐만 아니라, FPGA를 통해 손쉽게 조절 가능한 버스구조를 확보하였다.

2.2 전용 DMA를 이용한 TFT-LCD 구현

우리는 2.1의 다중 AHB 버스 아키텍처에 전용 DMA를 장착한 TFT-LCD 컨트롤러를 설계하였다. 전형적인 LCD 컨트롤러를 사용한 TFT-LCD 구동방법의 동작 순서는 표 1에 잘 표현되어 있다. 그리고 DMA를 내장한 TFT-LCD 컨트롤러를 사용한 TFT-LCD 구동방법은 표 2에 나타나 있다.

표 1. 전형적인 TFT-LCD 구동방법

순서	내용
1	프로세서가 LCD 영상 데이터를 메모리로부터 읽기
2	데이터를 프로세서 레지스터에 저장
3	저장된 레지스터 데이터를 TFT-LCD 컨트롤러에 전달
4	TFT-LCD 컨트롤러의 데이터를 TFT-LCD 패널에 전달

표 2. DMA를 사용한 TFT-LCD 구동방법

순서	내용
1	프로세서가 TFT-LCD 컨트롤러의 "Enable" 레지스터에 값을 기입하여 TFT-LCD를 동작시킴
2	TFT-LCD 컨트롤러의 레지스터에 저장된 명령에 맞게 DMA가 메모리와 통신하며 데이터를 라인버퍼에 저장
3	라인버퍼 데이터가 LCD 드라이버를 거쳐 TFT-LCD 패널에 전달

위의 두 방법에 대한 특징적인 차이는 TFT-LCD 영상데이터를 전송하는 주체이다. 프로세서가 직접 데이터를 전송할 수도 있으며, DMA를 통해 독자적으로 전송하는 것도 가능하다. 예전에는 TFT-LCD 구동하는 칩의 프로세서는 TFT-LCD만 담당하는 형태라서, 프로세서가 직접 영상데이터를 전송하였다. 그러나 최근의 추세는 하나의 프로세서가 여러 가지 구성요소를 모두 제어하는 SoC 칩 형태로 발전하고 있다. 결국 프로세서는 전체적인 시스템을 운영해야 하며, TFT-LCD 구동하는데 전담할 수 없게 되어 DMA 마스터 등을 통해 업무를 분담시켜야 한다.

III. 결론

많은 컴포넌트를 하나의 칩에 구현하기를 희망하는 사용자의 요구에 따라 단일 shared bus 구조는 다중 shared bus 구조로 급속하게 변화하고 있다. 본 연구에서는 여러 다중 shared bus 구조에 대해 고찰해 보고 브릿지의 레이턴시를 줄이기 위해서 메모리 셀렉터를 도입한 구조를 제안하였다. 그리고 위의 버스구조에 DMA 마스터가 장착된 LCD 컨트롤러를 집적시켜서 정상 동작됨을 확인할 수 있었다. VHDL 코드를 이용하여 설계된 DB를 ModelSim 툴로 타이밍 시뮬레이션한 결과에서 AHB 구조 하에서 DMA, LCD 라인버퍼, SDRAM 컨트롤러 등이 잘 동작되었으며, LCD 패널이 장착된 실제 FPGA 보드에서 이미지를 확인하였다.

참고문헌

[1] R. Lu and C.-K. Koh, "SAMBA-Bus: A High Performance Bus Architecture for System-on-Chips", IEEE Trans. on VLSI Systems, vol. 15, no. 1, pp.69.79, 2007.

[2] M. Anders, N. Rai, R. Krishnamurthy and S. Borkar, "A transition-encoded dynamic bus technique for high-performance interconnects", IEEE J. Solid-State Circuits, vol. 5, no. 4, pp.444.454, 1997.

[3] K. Lahiri, A. Raghunathan, and S. Dey, Dey, "Design Space Exploration for Optimizing On-Chip Communication Architectures", IEEE Trans. Computer-Aided Design, vol. 23, pp.952.961, June, 2004.

[4] K. Lahiri, A. Raghunathan, and S. Dey, "System-Level Performance Analysis for Designing On-Chip Communication Architectures", IEEE Trans. Computer-Aided Design, vol. 20, no. 6, pp. 768.783, June 2001.