

동적 재구성이 가능한 SoC 3중 버스 구조*

Dynamically Reconfigurable SoC 3-Layer Bus Structure

김규철[†], 서병현[‡]

Kyuchull Kim, Byunghyun Seo

Abstract

Growth in the VLSI process and design technology is resulting into a continuous increase in the number of IPs on a chip to form a system. Because of many IPs on a single chip, efficient communication between IPs is essential. We propose a dynamically reconfigurable 3-layer bus structure which can adapt to the pattern of data transmission to achieve an efficient data communication between various IPs. The proposed 3-layer bus can be reconfigured to multi-single bus mode, and single-multi bus mode, thus providing the benefits of both single-bus and multi-bus modes. Experimental results show that the flexibility of the proposed bus structure can reduce data transmission time compared to the conventional fixed bus structure. We incorporated the proposed bus structure in a JPEG system and verified that the proposed structure achieved an average of 22% improvement in time over the conventional fixed bus structure.

요 약

집적회로의 공정기술 및 설계기술이 발전함에 따라 많은 IP가 하나의 반도체 칩에 집적되어 하나의 시스템을 구성하는 SoC 설계가 많이 이루어지고 있다. 본 논문에서는 다양한 IP 간에 효율적인 데이터 통신이 이루어지도록 버스 상의 전송 특성에 따라 버스모드를 동적으로 재구성하는 SoC 3중 버스 구조를 제안한다. 제안된 버스는 다중-단일버스 모드, 단일-다중버스 모드로 재구성이 가능하며 따라서 단일버스 모드와 다중버스 모드의 장점을 모두 갖는다. 실험결과 제안된 버스구조는 기존의 고정된 버스구조보다 독립적이며 데이터 전송시간을 단축시킬 수 있음을 확인하였다. 그리고 제안된 버스구조를 JPEG 시스템에 적용한 결과 다중버스구조보다 평균 22%의 전송시간 단축을 얻을 수 있었다.

Key words : multi-layer bus, dynamically reconfigurable, SoC, arbiter, bridge

1. 서론

SoC는 하나의 칩 내부에 프로세서, 버스, 메모리, 레지스터 등을 포함하여, 설계자가 원하는 동작을 수행하는 IP(Intellectual Property)와 이를 연결하는 통신 구조로 구성되어 있다. IP를 연결하는 통신 구조는 보편적으로 버스 구조를 사용하고 있다. 버스 구조는 사용되는 버스의 수에 따라서 단일버스 구조와

다중버스 구조로 구분된다. 다중버스를 구성하기 위한 버스 확장 방법에는 브리지를 이용한 버스 확장구조와 내부매트릭스를 이용한 구조가 있다[1]-[4]. 브리지를 이용한 버스 확장구조의 경우 모든 마스터와 슬레이브 간의 전송이 가능하지만, 브리지를 사용하기 위한 추가적인 시간이 요구된다. 내부 매트릭스를 이용한 구조의 경우 매트릭스에 연결된 슬레이브로 두 개의 마스터가 동시에 전송을 할 수 없기 때문에 레이턴시(latency)가 증가할 수 있으며, 매트릭스에 연결되지 않은 서로 다른 버스 상의 마스터와 슬레이브 간의 전송이 불가능하다.

버스구조는 마스터들이 높은 버스 대역폭을 요구하는 경우 이를 해결할 수 있어야 한다[5][6][7].

* 이 연구는 2008년도 단국대학교 대학연구비 지원으로 연구되었음.

[†] 단국대학교 공과대학 컴퓨터학부

[‡] 단국대학교 대학원 전자컴퓨터공학과

일반적으로 대역폭은 하나의 버스구조에 연결된 마스터들에게 분배되기 때문에 버스의 수가 증가하면, 높은 대역폭과 낮은 레이턴시를 갖게 된다[8]. 그러나 전역전송이 지역전송보다 많아질 경우 브리지의 레이턴시 때문에 다중버스 구조의 효율성이 떨어진다. 따라서 높은 대역폭과 낮은 레이턴시가 요구되는 경우에는 다중버스 구조가 적합하며, 낮은 버스사용률을 보이며 전역전송이 지역전송보다 많이 발생하는 경우에는 단일버스 구조가 더 효율적이다. 이미 결정된 전송만을 수행하는 시스템에서는 버스사용률, 대역폭과 레이턴시 등을 고려하여 적절한 버스의 수를 결정하고 마스터를 배치하여 효율적인 시스템을 구현할 수 있다. 그러나, 다양한 패턴의 전송이 요구되는 경우 고정된 버스구조는 효율성에 한계가 있다. 이러한 문제를 해결하기 위하여 재구성성이 가능한 버스구조가 많이 연구되고 있다[9]-[13].

본 논문에서는 고정된 버스구조보다 효율적인 전송이 이루어지도록, 버스 상의 전송특성에 따라 다중버스 모드, 단일버스 모드로 실시간 재구성이 가능한 버스구조를 제안한다. 제안된 버스구조는 세 개의 버스를 사용하여 다중-단일버스 모드, 단일-다중버스 모드인 혼합형 모드로 재구성이 가능하여 단일버스와 다중버스의 장점을 모두 갖는다.

II. 본론

1. 제안된 버스구조

본 논문에서는 버스에 대한 접근빈도가 높은 마스터를 디폴트(default) 마스터로 정하여 IDLE 상태에서 버스 사용 권한을 갖게 함으로써 마스터의 버스 접근 시간을 줄일 수 있는 버스 구조를 제안하였다. 이러한 방식을 통해서 서로 다른 버스에 놓인 마스터와 슬레이브 간의 통신에서 브리지를 통과하며 발생하는 추가적인 버스접근 시간을 제거할 수 있다. 즉, 다중버스 구조에서 단일버스 구조와 동일한 동작이 가능하다.

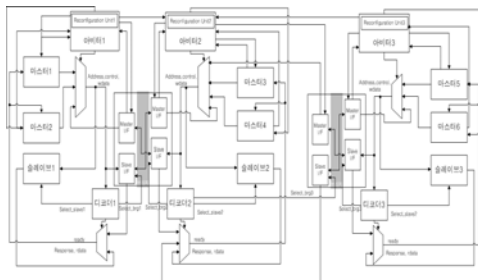


Fig. 1. Proposed 3-layer bus structure

그림 1. 제안된 3중 버스 구조

그림 1은 제안된 3중 버스구조를 보이고 있다. 제안된 버스는 일반 다중버스와 유사한 구조를 갖는다. 기존의 구성요소 이외에, 버스모드 변경을 위해서 버스에 대한 접근 빈도가 가장 높은 마스터를 결정하는 재구성장치(Reconfiguration_Unit)를 사용하였다. 다중버스 구조에서 단일버스 구조와 같은 동작을 수행하도록 바이패스 브리지(Bridge)를 사용하고, 중재기(Arbiter)의 기능을 확장하였다.

가. 브리지

그림 2는 브리지의 인터페이스를 나타낸다. 브리지는 마스터 인터페이스와 슬레이브 인터페이스로 구성된다. 마스터 인터페이스는 슬레이브와 연결되고 슬레이브 인터페이스는 마스터와 연결된다. BRIDGE_BYPASS는 브리지의 바이패스 기능을 활성화하기 위한 제어신호이다.

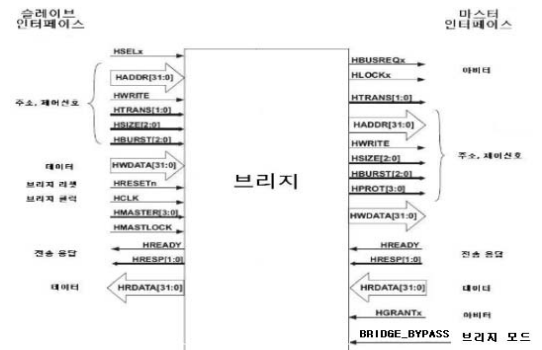


Fig. 2. Bridge interface

그림 2. 브리지 인터페이스

브리지의 동작은 다음과 같이 이루어진다. 입력된 데이터를 브리지의 슬레이브 인터페이스에 저장하고, 마스터 인터페이스를 통하여 전송하고자 하는 버스의 중재기에게 버스접근권한을 요청한다. 버스권한을 부여받으면 최종 목적 슬레이브로 전송을 시작하고, 슬레이브로부터 받은 응답을 마스터로 전송한다.

브리지의 모드제어 신호는 중재기로부터 받는다. 다중버스 모드에서는 보편적인 크로스바 브리지와 동일한 동작을 수행한다. 단일버스 모드에서는 브리지의 마스터 인터페이스와 슬레이브 인터페이스가 직접 연결된다. 중재기로부터 제어신호를 받으면 브리지는 진행 중인 전송을 마친 후 브리지 모드를 변경한다.

브리지의 바이패스 기능이 활성화되어 단일버스 모드로 동작하는 경우, 동일 버스 상에서 전송될 데이터가 인접 버스로 전송되는 것을 방지하기 위해서 브리지는 IDLE 전송을 한다.

나. 중재기

제한된 버스의 중재기는 특정 마스터가 버스권한을 독점하게 되는 문제를 방지할 수 있도록 라운드로빈(Round-Robin) 중재 방식을 취한다. 라운드로빈 중재 방식은 현재 권한을 가지고 있는 마스터에게 가장 낮은 우선순위를 부여하여 우선순위를 순환시킨다.

그림 3은 제안된 버스의 중재기 인터페이스를 보이고 있다. 이 중재기 인터페이스에는 기존의 다중버스 구조의 중재기 인터페이스에 재구성장치로부터 디폴트 마스터 선택 신호를 받는 입력과, 브리지의 모드를 제어하는 출력이 추가되어 있다.

중재기의 기본적인 동작은 기존 다중버스 구조의 중재기와 동일하다. 버스 접근을 요청하는 마스터가 없으면, 재구성장치로부터 받은 디폴트마스터 선택신호를 이용하여 가장 버스접근 빈도가 높은 마스터에게 버스 권한을 부여하여 디폴트 마스터로 정한다. 브리지가 디폴트 마스터로 결정되면, 중재기는 버스를 단일버스 모드로 동작시키기 위해서 브리지로 버스모드를 변경하기 위한 신호(BRIDGE_BYPASS)를 보낸다.

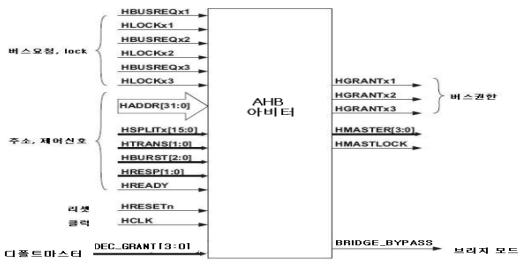


Fig. 3. Arbiter interface
그림 3. 중재기 인터페이스

다. 재구성장치

재구성장치는 전송되는 데이터 패턴을 감시하여 적절한 버스모드를 결정한다. 동일 버스 내의 전송인 지역전송과 서로 다른 버스 사이의 전송인 전역전송의 비율을 확인하여 바이패스 기능의 활성화 여부를 결정한다. 여기에서 고려하여야 할 사항은 버스 모드를 변경하는 시점이다. 예를 들어, 다중버스 구조에 적

합한 데이터를 전송할 때 단일버스 구조를 유지하고 있다면, 모든 마스터들은 순차적으로 전송하고자 하는 데이터를 전송해야하기 때문에 나머지 마스터들은 대기 상태에 놓여 있게 된다. 반대로, 각 마스터가 순차적으로 다른 버스에 놓인 슬레이브로 데이터를 전송하기 위해 버스 사용 요청을 하고 있는 경우, 다중 버스 구조를 유지하고 있다면, 브리지를 통과하는 시간이 필요하기 때문에 효율이 떨어진다. 이와 같은 문제를 해결하기 위해 버스의 데이터 전송을 모니터링 하는 간격을 실시간으로 변경할 수 있어야 한다.

그림 4는 재구성장치의 인터페이스를 보이고 있다. 재구성장치는 중재기와 입력을 공유하므로 동일한 입력 인터페이스를 갖고 있으며, 사용빈도가 가장 높은 마스터에 대한 정보를 중재기로 보낸다.

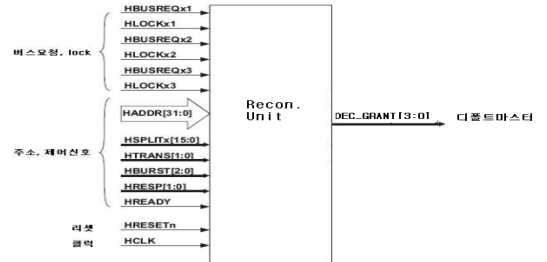


Fig. 4. Reconfiguration_Unit interface
그림 4. 재구성장치의 인터페이스

그림 5는 재구성장치의 동작흐름도를 보이고 있다. 재구성장치는 모니터링 시간(M_time)동안 마스터에서 중재기로 보내는 신호를 관찰한다. 모니터링 시간에 도달할 때까지 디폴트 마스터를 정할 수 있는 최소한의 정보량보다 적은 정보가 들어오면 최대 허용범위(M_time_max) 내에서 모니터링 시간을 두 배로 늘린다. 모니터링 시간에 도달하기 이전에 디폴트 마스터를 정할 수 있는 충분한 정보양이 입력되면 모니터링 시간을 반으로 줄이고 선택한 디폴트 마스터를 중재기에게 알린다. 이와 같은 방법으로 버스접근 빈도가 다양하게 변하는 마스터들에 대해서 효과적으로 디폴트 마스터를 결정할 수 있다.

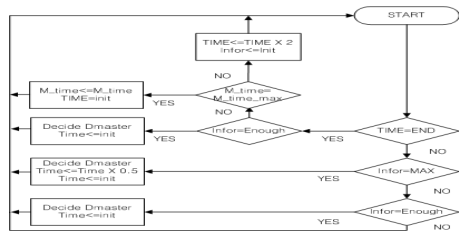


Fig. 5. Flow Diagram of Reconfiguration_Unit
 그림 5. 재구성장치의 동작 흐름도

2. 버스 구조 실험 및 결과

가. 설계 환경

Altera Quartus 4.2에서 VHDL로 제안된 구조를 설계하였다. Synthesis Target Device로는 ARM 코어와 FPGA가 결합된 Altera 사의 Excalibur를 사용하였다. 시뮬레이션에는 Mentor사의 Modelsim6.0d를 사용하였다.

나. 2중 버스 구조

제안된 버스구조의 성능을 실험하기 위해 데이터 발생기를 만들었다. 데이터 발생기는 실제 버스의 구성요소가 아닌, 본 논문에서 제안한 버스 구조를 검증하기 위하여 제작한 블록으로서 각 백엔드 로직을 가진 마스터 인터페이스에 연결되어 전송로드를 부여한다.



(가) 단일버스 모드에 적합한 데이터 전송 (나) 다중버스 모드에 적합한 데이터 전송

Fig. 6. Diagram of the Data Generator(2-layer bus)

그림 6. 데이터 발생기의 다이어그램(2중 버스)

그림 6은 두 개의 버스에서 데이터의 전송 경로를 나타낸다. 데이터 발생기는 그림 6의 데이터 경로를 갖는 상태를 조합하여 버스에 있는 마스터에 동작을 제어한다.

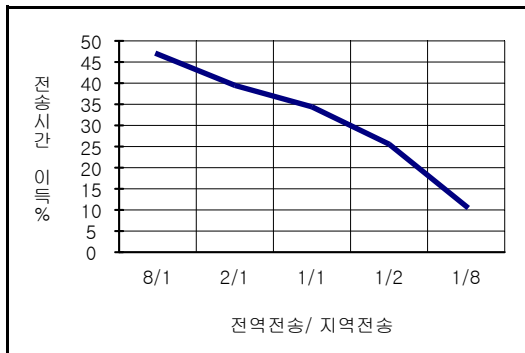


Fig. 7. Time comparison between global and

local transmission

그림 7. “전역전송/ 지역전송”에 따른 전송시간 비교

그림 7은 동일 버스사이의 전송과 서로 다른 버스사이의 전송의 비에 따른 제안된 버스 구조의 다중버스 구조에 대한 전송시간 비율을 보여주고 있다. 전체 전송에서 전역전송이 차지하는 비율이 증가할수록 제안된 버스구조의 전송효율이 증가함을 확인할 수 있다.

다. 3중 버스 구조

그림 8은 세 개의 버스에서의 데이터 전송 경로를 나타낸다. 데이터 발생기는 그림 8의 데이터 경로를 갖는 0부터 F까지의 16개의 상태를 조합하여 버스에 있는 마스터의 동작을 제어한다.

상태 0, 1은 두 개의 브리지를 통과하는 버스로의 전송, 상태 2~5는 인접한 버스로의 전송이며 (가)에 해당한다. 상태 8은 세 개의 버스에서 동시에 동일버스로의 전송, 상태 9~B는 두 개의 버스에서 동시에 동일버스로 전송으로 (다)에 해당한다. (나)에 해당하는 상태 C~D와 (라)에 해당하는 상태 E~F는 한 버스에서는 동일 버스로의 전송이고 다른 두 버스에서는 인접한 두 개의 버스로의 전송이다. 상태 6~7은 하나의 버스에서 동일 버스 내의 전송으로서 버스의 영향을 받지 않는다.

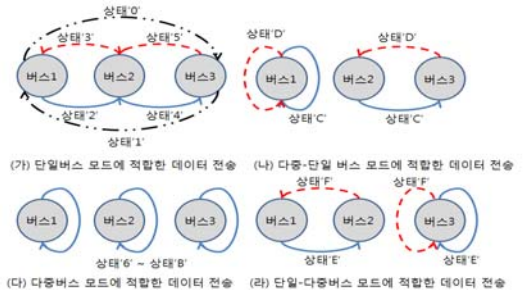


Fig. 8. Diagram of the Data Generator(3-layer bus)

그림 8. 데이터 발생기의 다이어그램(3중 버스)

그림 9는 데이터 발생기의 16개의 상태를 한 번씩 수행하는 데 소요되는 시간을 나타낸다. 상태 6~B는 각 버스 상에서 전송이 일어나므로 두 버스 구조 사이에 처리시간의 차이가 없다. 상태에 따라서 전송 시간의 차이가 다르므로 상태의 분포에 따라 전체적인 전송 효율이 변하게 된다.

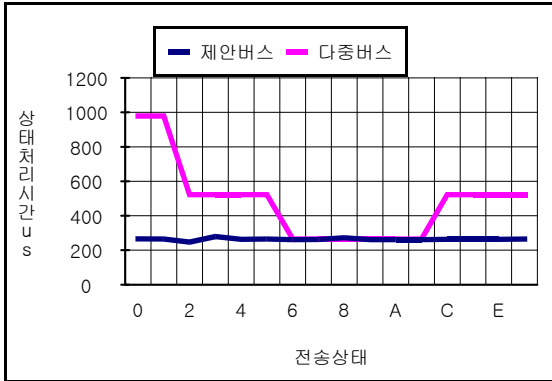


Fig. 9. Transmission time for each state (uS)
 그림 9. 각 상태에 대한 전송 시간 (uS)

그림 10은 총 9개의 마스터 M1~M9가 부속되어 있는 버스에 대해 16개의 상태를 순서대로 반복 수행할 때, 각 마스터들의 10ms 동안의 실행 횟수를 나타내고 있다. M1~M3은 버스1, M4~M6은 버스2, M7~M9는 버스3에 놓여있다. 동일 시간 동안 제안된 버스에서 더 많은 마스터가 동작하고 있는데 이는 제안된 버스구조에서 상태 6~B를 제외한 나머지 상태에서 모두 전송이 빨리 일어나기 때문이다. 위 실험들을 통하여 버스 상의 가능한 데이터 전송경로에 대한 전송시간과 동일 시간동안 마스터의 데이터 처리속도를 확인하였으며 제안하는 버스구조가 버스 상의 데이터 패턴에 따라 적절한 버스모드로 재구성됨을 확인할 수 있다. 이러한 버스의 재구성을 통하여 고정된 다중버스 구조보다 데이터의 전송경로에 따른 손실을 줄여 동일 시간동안 많은 데이터의 전송 및 처리가 가능하다.

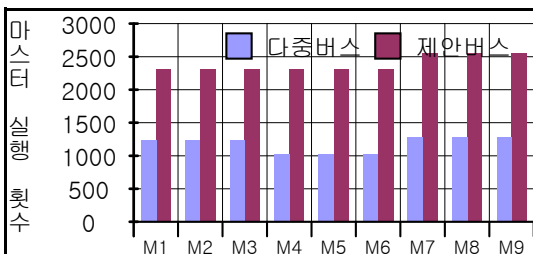


Fig. 10 Transaction counts for a master
 그림 10. 마스터의 실행 횟수

라. JPEG 시스템의 적용

성능을 검증하기 위하여 그림 11의 JPEG 시스템에

제안된 버스 구조를 적용시켰다[14]. JPEG 압축을 위한 블록은 첫 번째 버스에, 복원을 위한 블록은 두 번째 버스에, 그리고 화면에 출력하기 위한 블록은 세 번째 버스에 연결하였다. 버스 1에 놓인 "RGB TO Converter"는 BMP 파일을 읽어서 메모리에 YCbCr로 저장하고 "JPEG Encoder"는 이를 압축한다. 버스 2에 놓인 "JPEG Decoder"와 "YCbCr TO RGB Converter"가 파일을 복원시킨다. 버스 3에 놓인 "LCD Controller"는 복원된 결과를 화면에 출력한다.

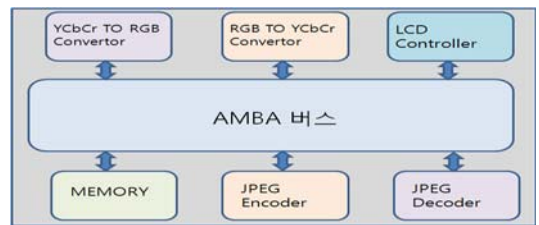


Fig. 11. JPEG system
 그림 11. JPEG 시스템

그림 12는 JPEG 시스템의 버스 상의 전송 부하에 대한 시뮬레이션과 제안된 버스구조의 브리지의 바이패스 제어 신호를 나타내고 있다. 서로 다른 버스로의 전송 시 브리지의 바이패스기능이 활성화 되고 있음을 확인할 수 있다. 버스1, 2, 3에 놓인 슬레이브는 각각 HEX"80xxxxxx", HEX"81xxxxxx", HEX"82xxxxxx"의 주소영역을 갖는다.

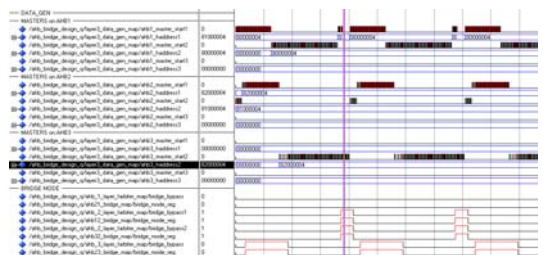


Fig. 12. Simulation for JPEG bus load
 그림 12. JPEG 버스 부하 시뮬레이션

그림 13은 그림 11에 보인 JPEG 시스템을 고정된 다중버스 구조와 제안된 버스구조에 적용시킨 시뮬레이션 결과를 보이고 있다. 시뮬레이션 결과, 제안된 버스 구조에서는 고정된 다중버스 구조보다 평균 22%의 시간이 단축됨을 확인하였다.

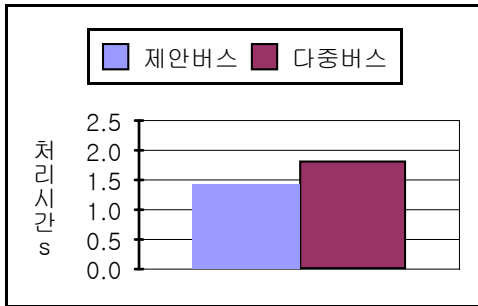


Fig. 13. Simulation result for JPEG system
그림 13. JPEG 시스템의 시뮬레이션 결과

III 결론

기존의 고정된 버스 구조는 하나의 버스구조에 여러 시스템이 구성되면서 발생하는 다양한 전송에 대하여 제한적인 연결을 제공하거나 비효율적인 전송이 발생한다. 본 논문에서는 이러한 문제를 해결하기 위하여 동적 재구성이 가능한 3중 버스 구조를 제안하였다. 동적재구성을 구현하기 위하여 중재기와 브리지의 기능을 수정하고 재구성장치를 추가하였다. 재구성장치는 버스 상의 데이터 전송을 감시하여 버스에 대한 접근빈도가 가장 높은 마스터를 디폴트 마스터로 정의하고 버스모드를 동적으로 변경시킨다. 제안된 버스 구조를 검증하기 위하여 버스에서 전송 가능한 모든 경우를 각 상태로 표현하고 시뮬레이션을 수행하였다. 시뮬레이션 결과 제안된 버스 구조가 동일한 시간동안 기존의 버스 구조보다 더 많은 데이터를 전송할 수 있음을 확인하였다. 실제 시스템에서의 성능을 확인하기 위하여 JPEG 시스템에 제안된 버스 구조를 적용시켜 다중버스 구조와 비교하였다. 그 결과 제안된 버스구조는 고정된 다중버스 구조보다 평균 22%의 전송시간 단축을 보였다. 이는 제한된 버스 구조가 고정된 버스구조보다 데이터 패턴에 따라 버스 모드를 변경함으로써 전송시간을 단축시킬 수 있음을 의미한다.

참고문헌

[1] 공민철, "다중 마스터 설계를 이용한 AMBA 인터페이스 구현", 한국산업기술대학, 2004.
[2] ALTERA, multi_master_reference_design, 2002.
[3] ARM, Multi-layer AHB Overview, 2001.
[4] Baruch Halachmi, "A MULTI-BUS INTERCONNECTION MODEL", Simulation

Conference Proceeding, pp.1138-1142, DEC.1993.
[5] Kyeong Keol Ryu, et al., "A comparison of five different multiprocessor SoC bus architectures", in Proc. of Euromicro Symposium on Warsaw Poland, Digital Systems Design, pp.202-209, SEPT. 2001.
[6] 천제익, 김보관, "SoC를 위한 통신망 설계 동향", 전자공학회지, 제30권, 제5호, pp.992-1001, SEPT. 2003.
[7] S. Han, A. Baghdadi, M. Bonaciu, S. Chae, and A. A. Jerraya, "An Efficient Scalable and Flexible Data Transfer Architecture for Multiprocessor SoC with Massive Distributed Memory.", Design Automation Conf., pp.370-371, 2004.
[8] 조신형, 조경록, "SoC 플랫폼에서 다중구조 공유버스의 레이턴시 모델과 해석", 컴퓨터 정보통신 연구, vol.13, no.1, pp.85-92, 2005
[9] K. Larhiri, A. Raghunathan, S. Dey, "Design of High Performance System-on-Chips Using Communication Architecture Tuners", IEEE Trans. on CAD, vol.23, pp.919-932, 2004.
[10] S. Pasricha, N. Dutt, and M. B. Romdhane, "Fast Exploration of Bus-based On-chip Communication Architectures", Symp. HW/SW Codesign, 2004.
[11] T. Meyerowitz, C. Pinello, and A. Sangiovanni-Vincentelli, "A Tool for Describing and Evaluating Hierarchical Real-Time Bus Scheduling Policies", Design Automation Conf., pp.312-317, 2003.
[12] K. Sekar, K. Lahiri, A. Raghunathan, S. Dey, "FLEXBUS : a high-performance system-on-chip communication architecture with a dynamically configurable topology", Design Automation Conf., Proceedings. 42nd 13-17, pp.571 - 574, 2005.
[13] 서병현, 김규철, "동적 재구성이 가능한 고성능 시스템온칩 버스 구조에 관한 연구", 2007 정보 및 제어학술대회 논문집, pp.369-370, 2007.
[14] IDEC, IDEC NEWSLETTER, DEC. 2007.

 저 자 소 개

김 규 철 (정회원)



1978년 2월 : 서울대학교
자연대학 물리학과 학사
1980년 2월 : 서울대학교
자연대학 물리학과 석사
1986년 2월 위스콘신대학
전기컴퓨터학과 석사
1992년 2월 : 위스콘신대학

전기컴퓨터학과 박사

1993년 : 삼성전자 마이크로본부

1993년 ~ 현재 : 단국대학교 공과대학 컴퓨터학부
<주관심분야> 집적회로 설계 및 테스트, Design
For Testability, SoC 설계, 테스트 스케줄링,
신호처리

서 병 현 (비회원)



2006년 2월 : 단국대학교
전자컴퓨터 공학 학사
2008년 2월 : 단국대학교
전자컴퓨터 공학 석사
2008년 ~ 현재 : LG display,
TV LED 회로설계팀
<주관심분야> Flexible Bus
Architecture , Timing

Controller And LCD Control Circuit