

버스충돌 감소를 위한 결합전송 아비터 구조

김일산, 나재호, 한탁돈

연세대학교 컴퓨터과학과

e-mail : {sany, jhnah, hantack}@kurene.yonsei.ac.kr

Transmission Combining Arbiter for Reducing Bus Conflicts

Il-San Kim, Jae-Ho Nah, Tack-Don Han

Media System Laboratory, Department of Computer Science, Yonsei University,

요 약

제안하는 arbiter 구조는 AMBA AHB Protocol 에서 사용하는 표준 arbiter 를 개선하여, master device 들간의 버스 사용에 따른 bus conflict 를 감소시킨 구조이다. 제안하는 arbiter 구조는 인접한 주소를 참조하는 master device 들의 전송을 버스의 대역폭 내에서 한 번에 전송함으로써 버스 전송 횟수 및 데이터 전송량을 감소시킨다. 실험결과, 제안하는 arbiter 구조는 기존의 arbiter 구조에 비해 최대 89%의 전송량이 감소하였다.

1. 서론

AMBA AHB Protocol[1]은 System-On-Chip(SOC) 를 구성하기 위한 open standard, on-chip bus specification 으로서, bus 통해 연결된 다양한 master and slave device 이 통신하는 규약이다. AMBA AHB Protocol 에서는 한번에 한 개의 master 장치만 버스를 사용할 수 있기 때문에 master device 들간의 arbitration 이 필요하고, 잦은 arbitration 은 bus conflict 를 발생시켜 시스템의 성능을 저하시킨다.

2. 제안하는 아비터 구조

제안하는 arbiter 는 버스의 대역폭 내에서 복수 개의 master device 들의 데이터 전송을 결합하여 한 번에 전송함으로써 전송 횟수 및 전송량을 감소시키는 구조이다. 그림 1)은 제안하는 arbiter 구조의 block diagram 으로서, 버스의 사용권한을 할당하는 Master Selector, slave device 의 사용 권한을

할당하는 Ready Selector, master 들의 주소와 제어 신호를 결합하는 Address Combiner, master 들의 write data 를 결합하는 Data Combiner, 그리고 읽 어온 data 를 master 들에게 분배하는 Data Decomposer 로 구성되어 있다.

3. 제안하는 아비터의 처리과정

버스를 통한 데이터 전송을 수행하려면 master device 는 버스의 사용권과 대상장치의 사용권을 받아야 한다. 기존의 arbiter 는 우선권이 높은 master device 의 순서대로 버스의 사용권과 장치의 사용권을 할당 받아 데이터를 전송하였다. 제안하는 arbiter 는 버스 사용을 요청하는 모든 master device 들에게 버스 사용을 허가하고, 허가된 master device 들의 주소를 분석하여 버스의 대역폭 내에서 한 번의 전송으로 처리가 가능한 master device 들에게 장치의 사용권을 허가하여 데이터를 전송한다. 전송과정은 크게 버스 사용

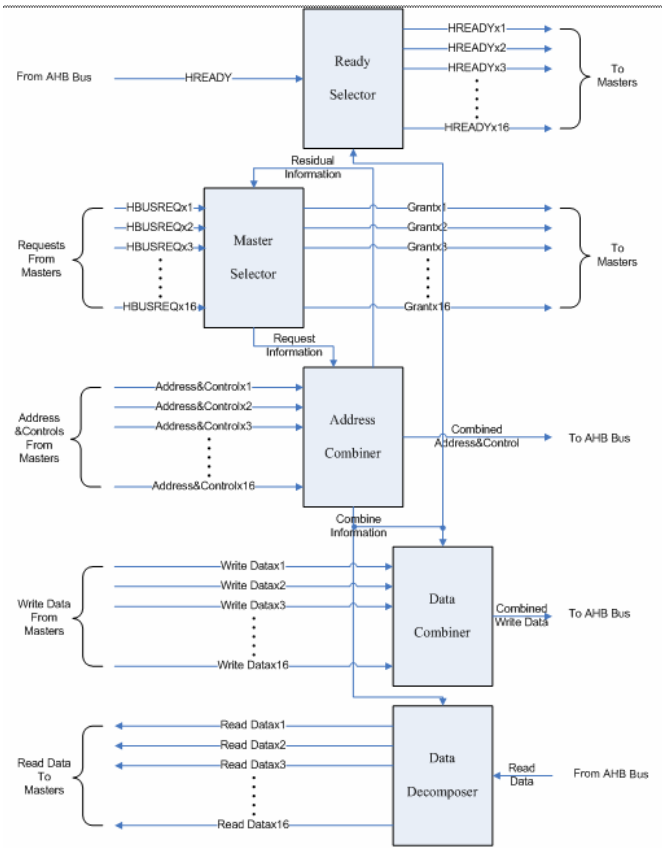


그림 1) 제안하는 아비터 구조

권한을 얻는 요청 단계, 사용할 slave device의 주소 및 제어 신호를 전달하는 주소 단계, 그리고 데이터를 전송하거나 전송 받는 데이터 단계로 나눌 수 있다.

- 요청 단계: Master device들이 버스 사용을 요청하면, Master selector는 버스 사용을 요청한 마스터들에게 사용 허가를 의미하는 HGRANT 신호를 되돌려준다. 그리고 결합을 수행할 master device들을 Address Combiner에 알린다. 만약, Residual Information에 이전에 버스 사용을 요청해서 결합이 수행되어야 할 master device의 정보가 남아 있다면, 해당 master device들의 결합이 완료될 때까지 HGRANT 신호를 전달한다.
- 주소 단계: HGRANT를 받은 master device들은 주소와 제어신호를 전달하며, Address Combiner는 우선권이 가장 높은 master device를 기준으로 버스의 대역폭 내에서 단일 전송으로 처리가 가능한 주소와 제어 신호들을 판단하여 한 개의 버스 신호로 결합한다. 이때, 결합 처리는 단일 전송을 대상으로 하며, 읽기 전송과 쓰기 전송은 분리하여 처리한다. 결합이 되지 못한 주소 및 제어신호들은 다음 결합처리를 위해

내부 레지스터에 저장되며, HGRANT 신호 획득을 위해 Master Selector에 전달된다. Ready Selector는 결합 처리된 master device에게만 slave device의 HREADY 신호를 전달하여 데이터를 전송할 수 있도록 한다.

- 데이터 단계: HREADY 신호를 받은 master device들은 데이터 단계에서 전송을 수행하는데, Data Decomposer는 read transfer를 요청한 master device들에게 버스를 통해 읽어들인 정보를 전달하며, Data Combiner는 master device들의 write data를 결합하여 버스를 통해 전송한다.

4. 시뮬레이션 결과 및 결론

제안하는 구조의 성능을 측정하기 위하여, 우리는 3차원 응용프로그램의 깊이 정보가 3차원 그래픽 가속기의 픽셀파이프라인에서 처리하는 과정을 시뮬레이션 하였다. 픽셀 파이프라인은 arbiter를 통해 버스에 연결되어 깊이 버퍼와 깊이 정보를 주고 받는다. 고려된 버스의 대역폭은 4bytes, 8bytes, 16bytes, 그리고 32bytes 이고, 픽셀 파이프라인의 수는 4, 8, 16 개이다. 사용된 응용프로그램들은 상용 3차원 게임인 Unreal Tournament 2004(UT2004)와 Quake3 로서, 성능측정을 위해 다른 연구들에서도 자주 사용되는 응용프로그램이다[2].

표 1) 벤치마크에 따른 실험결과

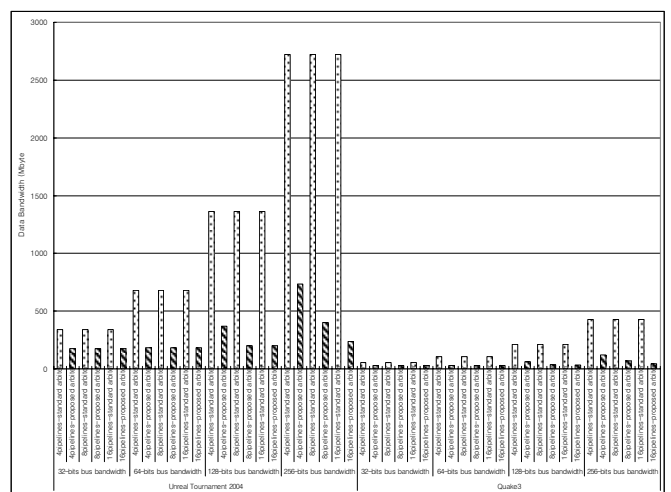


표 1)은 각 벤치마크들에서 10 프레임 동안 버스를 통해 전송된 데이터 전송량과 전송횟수를 실험한 결과로서, Original은 표준 아비터 구조를, Proposed는 제안하는 아비터 구조를 의미한다. 표 1)의 결과에서, 제안하는 구조는 표준 구조에

비해 데이터 전송 횟수 및 전송량이 크게 감소됨을 알 수 있는데, 이는 실험에 사용된 응용프로그램들이 연속적인 메모리 주소를 참조하는 성향이 높기 때문이다. 또한 버스의 대역폭이 클수록 전송횟수 및 전송량이 크게 감소하는데, 이는 버스의 대역폭이 클수록 결합이 가능한 전송요구가 증가하기 때문이다. 따라서 실험결과를 통해 나타나듯이, 제안하는 arbiter 구조는 3 차원 응용프로그램과 같이 연속적인 주소를 참조하는 성향이 높고, 버스의 대역폭이 높은 환경에서 버스를 통한 데이터의 전송횟수 및 전송량을 감소시켜 시스템의 성능을 향상시킨다.

참고문헌

- [1] Advanced RISC Machines Limited(ARM Ltd.), available at <http://www.arm.com/products/solutions/AMBAHomepage.html>
- [2] Woo-Chan Park, Kil-Whan Lee, Il-San Kim, Tack-Don Han, and Sung-Bong Yang, "An Effective Pixel Rasterization Pipeline Architecture for 3D Rendering Processors," *IEEE Transactions on Computers*, Vol. 52, No. 11, pp. 1501-1508, Nov. 2003.