
데이터 트래픽 집중에 따른 버스 아키텍처의 성능분석

이국표* · 고시영**

Performance Analysis of Bus Architecture Due to Data Traffic Concentration

Kookpyo Lee* · Si-Young Koh**

요 약

일반적인 버스 시스템 구조는 공용버스 내에 여러 개의 마스터와 슬레이브, 아비터 그리고 디코더 등으로 구성되어 있다. 복수의 마스터가 동시간대에 버스를 이용할 수 없으므로, 아비터는 이를 중재하는 역할을 수행한다. 아비터가 어떠한 중재방식을 선택하는가에 따라 버스 사용의 효율성이 결정된다. 기존의 중재 방식에는 **Fixed Priority** 방식, **Round-Robin** 방식, **TDMA** 방식, 로터리 방식 등이 연구되고 있다. 본 논문에서는 데이터 트래픽 집중에 따른 **Fixed Priority**, **Round Robin**, **TDMA**, 로터리 방식에 대하여 성능을 분석해 보고, 성능개선을 위한 방법을 제안하려고 한다.

ABSTRACT

The general bus system architecture consists of masters, slaves, arbiter, decoder and so on in shared bus. As several masters can't use a bus concurrently, arbiter plays an role in bus arbitration. In compliance with the selection of arbitration method, The efficiency of bus usage can be determined. Fixed Priority, Round-Robin, TDMA, Lottery arbitration are studied in conventional arbitration method. In this paper, we draw the performance analysis of Fixed Priority, Round Robin, TDMA and Lottery bus arbitration policies due to the data traffic concentration and propose the methods of performance improvement.

키워드

AMBA, SoC, 버스아키텍처, 버스 중재

Key word

AMBA, SoC, bus architecture, arbitration

* 정회원 : 영진전문대학 전자정보통신계열 (주저자)

접수일자 : 2012. 07. 20

** 종신회원 : 경일대학교 전자공학과 (교신저자)

심사완료일자 : 2012. 09. 07

I. 서론

현대 사회가 점점 정보화 사회로 진보하는 데에는 반도체 산업이 매우 중요한 역할을 하고 있다. 인터넷 시대의 등장과 함께 텔레비전, 냉장고와 같은 가전에서부터 노트북, 휴대용 정보기기 및 스마트 전자기기에 대한 수요가 증가하면서 전자회로의 임베디드(Embedded) 시스템화, 소형화, 저전력화되어 가고 있다. 또한 반도체 공정 기술의 발달로 기존의 여러 다른 기능을 하는 칩들을 하나에 집적시키는 SoC(System on a Chip) 기술이 각광 받고 있다[1-3].

SoC시스템의 경우, 마스터와 슬레이브, 아비터, 디코더로 구성되어 있다. 마스터는 CPU, DMA, DSP 등과 같은 프로세서들을 말하며, 슬레이브는 마스터와는 다르게 DRAM, SRAM과 같은 메모리를 의미한다. 또한, 아비터는 여러 개의 마스터가 동시시간에 버스를 이용할 수 없기 때문에 이를 중재하는 역할을 수행하고 중재하는 방식에 따라 버스의 효율적인 중재가 가능하기 때문에 전체 시스템의 성능 향상을 위해 많이 연구되고 있는 분야이다. 마지막으로 디코더는 마스터로부터 나오는 어드레스의 상위 비트를 가지고 적절한 슬레이브를 선택해주는 역할을 한다.

기존의 아비터 중재 방식에는 fixed priority 방식, round-robin 방식, TDMA 방식, 로터리 버스 방식 등 여러 가지가 있다[4~9]. 본 논문에서는 트래픽 조건에 따른 최적의 버스중재방식 설정방법에 도움을 주기 위하여 트래픽의 집중에 따른 버스 아키텍처의 성능을 분석해보고 버스 중재방식에 따른 특성차이를 파악해 보고자 한다.

II. 시뮬레이션 모델

일반적인 버스 아키텍처의 경우는 단일 버스에 여러 개의 마스터와 슬레이브 그리고 아비터, 디코더로 구성되어 있다. 버스의 개수가 오직 하나이기 때문에 데이터 전송을 수행할 때, 동시시간대에 마스터들이 버스를 이용할 수 없다. 그렇기 때문에 여러 마스터들이 버스를 이용하기 위해 서로 경쟁을 하고 아비터는 이를 중재하는 역할을 수행한다[4~9].

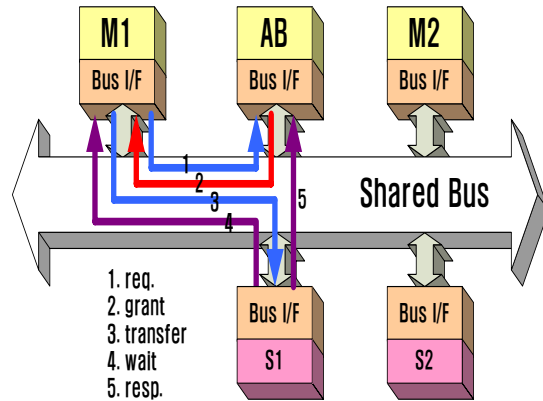
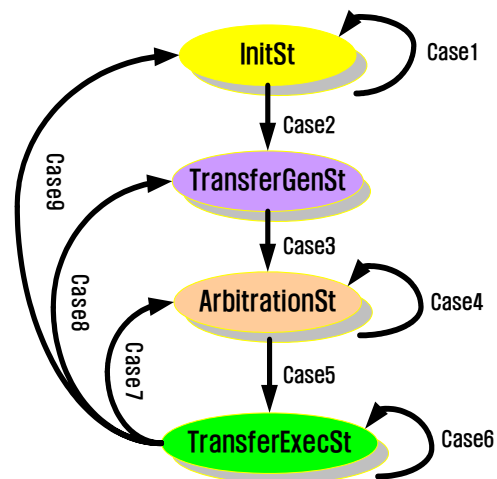


그림. 1 일반적인 버스 아키텍처의 데이터 트랜잭션
Fig. 1 Data transaction flow in general bus architecture.



- Case1 : If(idle_cycle[all master]>0)
- Case2 : If(idle_cycle[some masters]==0)
- Case3 : Always
- Case4 : During arbitration cycle
- Case5 : After arbitration cycle
- Case6 : During transaction cycle
- Case7 : If(Req[some masters]==True)
- Case8 : If(Req[all masters]==False) and (idle_cycle[some masters]==0)
- Case9 : If(idle_cycle[all masters]!=0)

그림. 2 버스 구조 모델의 상태도
Fig. 2 State machine of bus architecture model.

그림 1은 일반적인 버스 아키텍처의 데이터 트랜잭션을 보이고 있다. 마스터 M1과 M2는 각각 동시에 버스 사용 요청(req.)을 할 수 있다. 또한 요청한 신호가 아비터 AB에게 전송되면, 아비터 AB는 고유의 중재방식에 따라 선택된 마스터에게 버스 사용을 허가(grant)한다.

사용 허가를 받은 마스터는 해당 슬레이브 S1 또는 S2에 데이터 전송을 하고, S1 또는 S2는 wait 신호와 resp 신호를 통해 마스터와 아비터에게 데이터 처리에 관한 응답을 한다. 성능분석을 위해 그림 2와 같은 상태도를 기본 개념으로 하여 C++로 자체개발한 AMBA(Advanced Microcontroller Bus Architecture)를 사용하고 TLM(Transaction Level Model) 방법을 적용하였다[10].

III. 트래픽 집중에 따른 성능분석

자체 개발한 TLM 모델을 이용하여 Fixed Priority, Round Robin, TDMA와 로터리 중재방식의 버스성능을 비교해 보았다. 버스 중재방식의 성능은 데이터 트랜잭션이 많은 상황에서 주요한 차이가 나타나므로, 마스터의 버스요청을 많이 발생하기 위해서 각 마스터 M0~M5의 버스 요청 간격을 랜덤함수를 이용하여 평균 10 사이클, 0에서 20사이클 사이값으로 하였으며, 기타 변수는 기존 논문과 동일하게 설정하였다[10]. Fixed Priority의 경우 M0에서 M5의 순서로 마스터의 우선순위를 설정하였으며, Round Robin의 경우 M0에서 M5의 6개 마스터의 우선순위를 모두 동일하게 설정하였다. TDMA 방식의 슬롯수는 마스터 M0~M5에서 각각 3, 2, 2, 1, 1, 1이며, TDMA의 슬롯배치는 M0, M4, M1, M2, M0, M3, M1, M5, M0, M2 순으로 실시하였다. 로터리 방식의 버스중재 확률은 마스터 M0~M5에서 각각 30%, 20%, 20%, 10%, 10%, 10%로 두 방식의 비율을 동일하게 하였다.

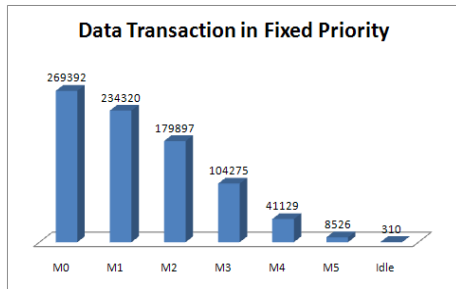
그림 3에 각 버스 중재 방식별 데이터 트랜잭션 사이클이 나타나 있다. 6개의 마스터가 평균 10사이클 이내에서 계속 데이터 전송을 요청하여 트래픽이 집중되는 상황이며, 버스 중재방식 따라 데이터를 전송하지 않고 대기하고 있는 idle사이클이 전체 1,000,000 사이클에서 300 사이클 내외 이었다. 이렇게 데이터 트래픽이 집중되는 상황에서는 버스중재방식에 따라 성능이 비교적 분명하게 나타나게 된다. 그림 3(a) Fixed Priority의 경우 M0 마스터의 경우, 약 27만 데이터가 전송된 반면, M5

마스터는 불과 1만의 데이터도 전송하지 못하고 있다. 그림 3(b) Round Robin 방식의 경우, 모든 마스터에서 거의 동일한 성능을 나타냈으며, TDMA와 로터리버스는 각각 슬롯과 중재확률에 따라 마스터의 성능이 결정되었다.

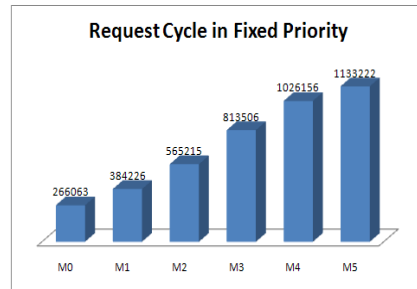
버스 성능 분석의 경우 실제 데이터 전송량과 함께 중요한 성능 지표는 버스요청 대기 시간이다. 임의의 마스터에서 버스 요청 후 버스 데이터 전송까지 걸리는 시간을 의미하며 작을 수록 데이터를 신속하게 전송할 수 있다. 그림 4에서는 버스중재 방식에 따른 버스 요청 사이클이 마스터 별로 나타나 있다. 그림 3의 데이터 전송량이 많을수록 버스요청 사이클이 적으므로 데이터 전송량과 버스요청 사이클은 모든 경우 반비례 관계를 나타내고 있다. 그림 3과 그림 4의 버스중재 방식 별 데이터 전송량, 버스요청 사이클 경향을 참조하여 실제 SoC 시스템의 응용처에 따라 버스중재 방식을 설정하는데 본 자료가 활용될 것으로 기대된다.

다음으로 데이터 트래픽의 강도에 따른 버스 아키텍처의 성능을 비교 분석해 보았다. 일반적으로 버스 성능은 그림 3과 그림 4의 경우와 같은 데이터 트래픽이 극단적으로 많은 상황에서 비교되는데, 실제 SoC 칩의 응용 상황에서는 데이터 트래픽이 많지 않은 경우가 대부분이고, 일정시간 특정한 상황에서만 트래픽이 몰리는 경향이 있다. 그래서, 트래픽이 극단적으로 많은 상황과 더불어 트래픽의 정도에 따른 버스 아키텍처의 성능을 분석하는 것이 중요하다. 본 연구에서는 버스 전송조건을 다양하게 하기 위하여, case1~case4에서 마스터 M0~M5의 버스 요청 간격을 랜덤함수를 이용하여 평균 10 사이클, 15 사이클, 20 사이클, 25 사이클로 변경하여, 그림 5와 그림 5의 결과를 얻었다.

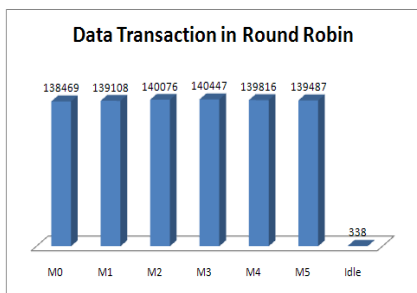
그림 5에서 버스중재 방식과 무관하게 데이터 트래픽이 많은 case1에서 데이터 트래픽이 적은 case4로 갈수록 마스터에 따른 데이터 전송량은 일정해지는 경향이 있다. 이는 우선순위가 높은 마스터가 데이터 요청을 하지 않는 경우가 많아서 우선 순위가 낮은 마스터가 데이터 전송할 기회가 많아져서 나타나는 현상이다. Round Robin 중재방식의 경우 모든 경우에 마스터별 데이터 전송량이 모두 일정했으며, Idle사이클이 약 7만 정도인 case4의 경우 그림 5의 마스터별 데이터 전송량이 Fixed Priority, TDMA, 로터리 버스에서 모두 거의 유사함을 알 수 있다.



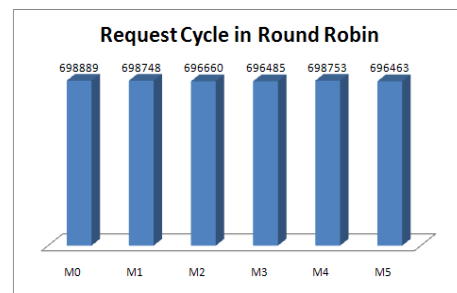
(a)



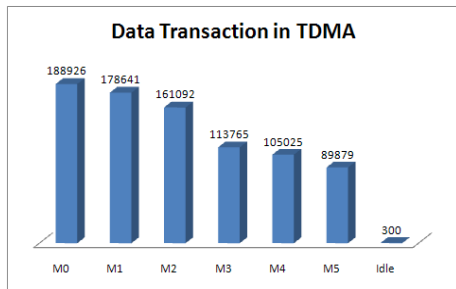
(a)



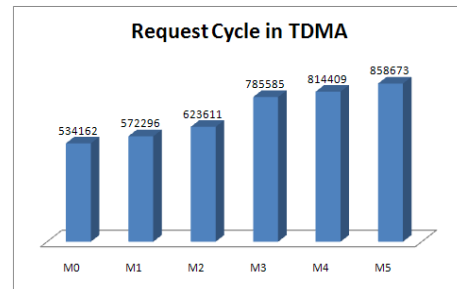
(b)



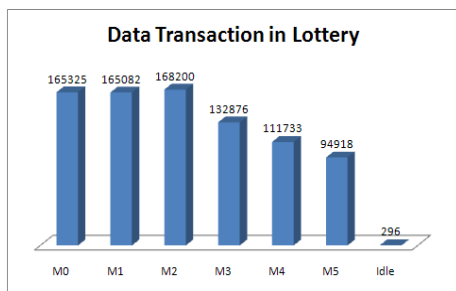
(b)



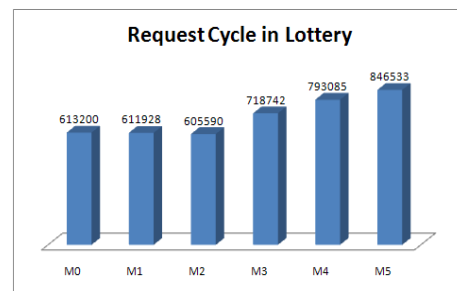
(c)



(c)



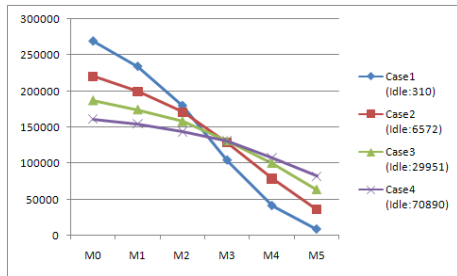
(d)



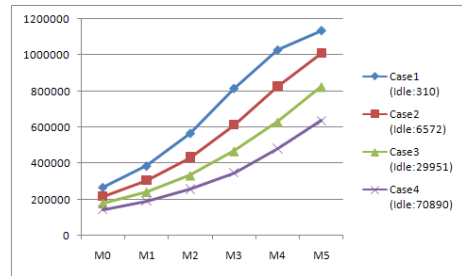
(d)

그림. 3 (a) Fixed Priority, (b) Round Robin, (c) TDMA, (d) 로터리 중재방식에서 데이터 트랜잭션 사이클
Fig. 3 Data transaction cycle in (a) Fixed Priority, (b) Round Robin, (c) TDMA and (d) Lottery arbitration

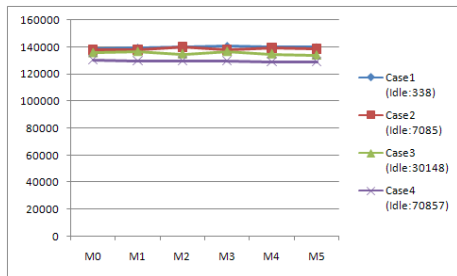
그림. 4 (a) Fixed Priority, (b) Round Robin, (c) TDMA, (d) 로터리 중재방식에서 버스 요청 사이클
Fig. 4 Bus request cycle in (a) Fixed Priority, (b) Round Robin, (c) TDMA and (d) Lottery arbitration



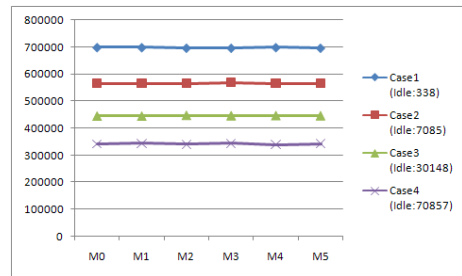
(a)



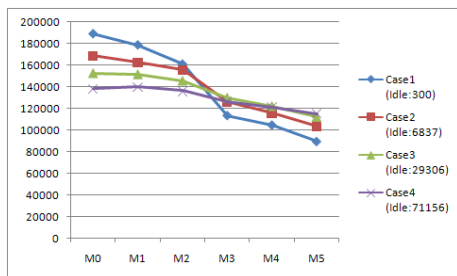
(a)



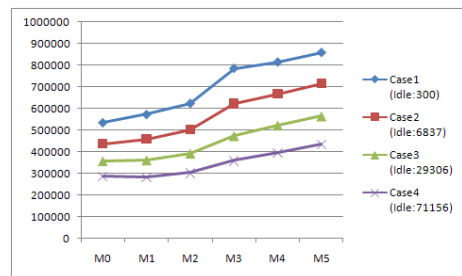
(b)



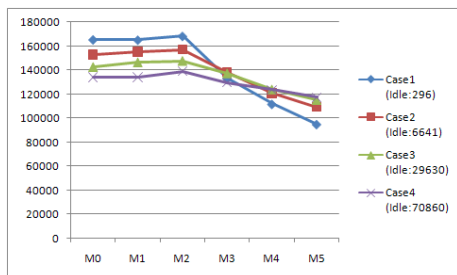
(b)



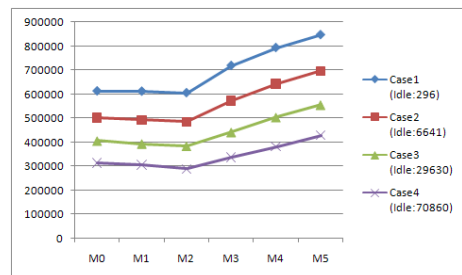
(c)



(c)



(d)



(d)

그림. 5 다양한 조건에 따른 (a) Fixed Priority, (b) Round Robin, (c) TDMA, (d) 로터리 중재방식의 마스터별 데이터 트랜잭션 사이클
Fig. 5 Data transaction cycle of each master due to the various conditions in (a) Fixed Priority, (b) Round Robin, (c) TDMA and (d) Lottery arbitration

그림. 6 다양한 조건에 따른 (a) Fixed Priority, (b) Round Robin, (c) TDMA, (d) 로터리 중재방식의 마스터별 버스요청 사이클
Fig. 6 Bus request cycle of each master due to the various conditions in (a) Fixed Priority, (b) Round Robin, (c) TDMA and (d) Lottery arbitration

그리고, 그림 6의 버스요청 사이클은 트래픽과 무관하게 데이터 전송량과 모든 경우 반비례 관계를 나타내었으며, 그림 5의 데이터 전송량 비교와 동일하게 case4로 갈수록 버스중재 방식에 따라 모두 일정해지는 경향이 나타났다.

결국 트래픽이 많지 않는 시스템을 설계할 경우, 기본적인 중재방식으로도 충분한 성능을 나타낼 수 있음을 알 수 있다.

IV. 결 론

본 논문에서는 TLM 모델을 이용하여 트래픽의 정도에 따른 Fixed Priority, Round Robin, TDMA, 로터리 버스 중재방식의 성능을 비교해 보았다. 기존의 성능분석에서는 트래픽이 극단적으로 많은 경우를 기준으로 실시하여 최대 성능을 나타내는 버스 아키텍처를 선택하여 시간과 설계인력 등 많은 오버헤드를 감수해야 했다. 본 연구에서는 트래픽이 극단적으로 많은 경우, 많은 경우, 중간인 경우, 아주 적은 경우 별로 성능을 분석하여 각 버스 중재방식별 특성을 마스터에 따라 비교해 보았다.

본 연구의 데이터를 통하여 버스 중재방식별 성능 분석을 물론 마스터에 따른 성능비교, 버스 트래픽 강도에 따른 비교를 쉽게 할 수 있으며, 이를 통하여 SoC 시스템에서 최적의 조건을 설정하는데 본 자료가 활용되리라 생각된다.

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] Sonics, Inc., Mountain View, CA, "Silicon micronetworks technical overview", 2002.

[3] AMBA TM Specification(AHB) (Rev 2.0), ARM Ltd, May 1999.

[4] L. N. Bhuyan, "Analysis of interconnection networks with different arbiter designs", J.Parallel Distrib. Comput., vol.4, no.4, pp.384-403, 1987.

[5] J. G. Delgado-Frias and R. Diaz, "A VLSI selfcompacting buffer for DAMQ communication switches", in Proc. IEEE 8th Great Lakes Symp. VLSI, pp.128-133, Feb. 1998.

[6] A. Bystrov, D.J. Kinniment and A. Yakovlev, "Priority Arbiters", in Proc. IEEE 6th internation Symp. ASYNC, pp.128-137, April. 2000.

[7] Y. Xu, L. Li, Ming-lun Gao, B.Zhand, Zhao-yu Jiand, Gao-ming Du, W. Zhang, "An Adaptive Dynamic Arbiter for Multi-Processor SoC", Solid-State and Integrated Circuit Technology International Conf., pp.1993-1996, 2006.

[8] K. Lahiri, A. Raghunathan, and G. Lakshminarayana, "The LOTTERYBUS On-Chip Communication Architecture", IEEE Trans. VLSI Systems, vol.14, no.6, 2006.

[9] http://www.samsung.com/global/business/semiconductor/productInfo.do?fmly_id=234&partnum=S3C2510A

[10] 이국표, 고시영, "TLM 방법을 이용한 다양한 중재 방식의 특성 비교", 한국정보통신학회논문지, 제13권 8호, pp.1653-1658, 2009.

저자소개

이국표(Kookpyo Lee)

한국정보통신학회 논문지
제12권 제3호 참조

고시영(Siyoung Koh)

한국정보통신학회 논문지
제12권 제3호 참조