

논문 2010-47SD-5-10

점유율을 고려한 버스중재 방식

(Bandwidth-Award Bus Arbitration Method)

최 항 진*, 이 국 표**, 윤 영 섭**

(Hang-Jin Choi, Kook-Pyo Lee, and Yung-Sup Yoon)

요 약

전행적인 버스 시스템 구조는 공용버스 내에 여러 개의 마스터와 슬레이브, 아비터 그리고 디코더 등으로 구성되어 있다. 복수의 마스터가 동시시간대에 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, Lottery 방식 등이 연구되고 있는데, 버스 우선권이 주로 고려되어 있다. 본 논문에서는 마스터별 버스 점유율을 연산하는 블록을 이용하는 버스중재 방식을 제안하고, TLM(Transaction Level Model)을 통해 다른 중재 방식과 비교하여 성능을 검증하였다. 성능분석 결과, 기존의 Fixed Priority 방식과 Round-Robin 방식은 버스점유율을 설정할 수 없었으며 기존의 TDMA, Lottery 중재방식의 경우에는 100,000 사이클 이상에서 사용자가 설정한 버스점유율과 비교하여 각각 최대 50%, 70%의 오차가 발생하였다. 반면에 점유율 고려 중재방식의 경우에는 약 1000 사이클 이후부터 사용자가 설정한 버스점유율과 비교하여 1% 이하의 오차를 유지하였다.

Abstract

The conventional 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 a role in arbitrating the bus. The efficiency of bus usage can be determined by the selection of arbitration method. Fixed Priority, Round-Robin, TDMA and Lottery arbitration policies are studied in the conventional arbitration method where the bus priority is primarily considered. In this paper, we propose the arbitration method that calculates the bus utilization of each master. Furthermore, we verify the performance compared with the other arbitration methods through TLM(Transaction Level Model). From the results of performance verification, the arbitration methods of Fixed Priority and Round-Robin can not set the bus utilization and those of TDMA and Lottery happen the error of 50% and 70% respectively compared with bus utilization set by user in more than 100,000 cycles. On the other hand, the bandwidth-award bus arbitration method remains the error of less than 1% since approximately 1000 cycles, compared with bus utilization set by user.

Keywords : bandwidth, AMBA, SoC, bus architecture, arbitration policy

I. 서 론

반도체 제조 공정의 발달에 힘입어 서로 다른 기능을 수행하는 여러 컴포넌트들을 하나의 칩에 집적시키는 SoC(System on Chip)기술이 회로설계에 널리 이용되고 있다.^[1] 이러한 SoC는 시스템 설계와 구현 시 칩 크

기와 설계비용을 절감시키며, 구동 시 저전력 소모와 실시간 처리 능력 및 시스템의 유연성 면에서 향상된 성능을 제공할 수 있다.^[2-3] SoC의 구성요소는 복수의 마스터(Master)와 아비터(Arbiter), 복수의 슬레이브(Slave) 및 마스터와 슬레이브 간의 데이터 송신을 위한 공용 버스(Shared Bus)로 이루어져 있다. 마스터는 프로세서, DMA, DSP를 의미하며, 슬레이브는 SDRAM, SRAM과 같은 메모리와 USB 및 UART와 같은 입/출력 수단을 의미한다. 하나의 칩에 여러 가지의 기능이 집적화된 SoC 칩의 경우, 하나의 버스에 마

* 학생회원, ** 정회원, 인하대학교 전자공학과
(Dept. of Electronics Engineering, Inha University)
※ 이 논문은 인하대학교의 지원에 의하여 발간되었음.
접수일자: 2009년11월25일, 수정완료일: 2010년4월1일

스터, 아비터, 슬레이브 등이 연결된 버스 시스템이 필요하다. 다수의 마스터들이 하나의 버스를 동시에 이용하고자 하는 경우, 아비터에 의해 버스 이용 허가 신호를 받은 마스터들이 순서대로 버스를 이용하여 데이터를 슬레이브로 전송하거나, 슬레이브로부터 데이터를 수신하게 된다. 이를 위해, 아비터는 다수의 마스터들이 동시에 버스 이용을 요청할 때 우선순위를 결정하게 되는데, 아비터의 우선순위 결정 방식에 따라 칩의 성능이 달라지게 된다.

아비터에서 사용되는 우선순위 결정 방식에는 Fixed Priority 방식과 Round Robin 방식, TDMA 방식, Lottery 방식 등이 있다.^[4-7]

Fixed Priority 방식은 다수의 마스터들에 대해 미리 정해진 우선 순위에 따라 버스 사용권을 주는 방식이다. 즉, 각각의 마스터에 대해 우선 순위를 프로세서, DSP, DMA1, DMA2의 순서로 1, 2, 3, 4 순위를 순차적으로 정해 놓고, 마스터들이 동시에 접근했을 경우, 우선순위가 높은 마스터에게 버스 사용권을 주게 된다. 이러한 Fixed Priority 방식은 마스터가 처리하게 될 데이터의 특성의 고려 없이 단순히 마스터의 종류에 따라 우선 순위를 결정하기 때문에 데이터 처리 시간 측면에서 효율적이지 못하고, 우선 순위가 낮은 특정 마스터에 스타베이션(Starvation)이 발생할 수 있는 문제점이 있고, 버스 대역폭(bandwidth)에 대한 고려는 전혀 없다.

Round Robin 방식은 마스터의 버스 우선순위를 시간에 따라서 다르게 변화시켜 버스 사용권을 골고루 주는 방식이다. 즉, 마스터에 일정시간을 할당하고, 할당된 시간이 지나면 그 마스터는 잠시 보류한 뒤 다른 마스터에게 버스 사용권을 주고 해당 할당 시간이 지나면 또 그 다음 마스터에게 버스 사용권을 주게 된다. 이러한 Round Robin 방식은 고정된 우선순위 없이 마스터에 할당된 시간 정보를 참조하여 우선순위를 정하기 때문에 빨리 처리해야 할 중요한 특정 마스터의 처리가 늦어지는 문제점이 있으며, 우선순위 없이 모든 마스터에게 균등하게 버스점유권을 부여하는 방식이기 때문에 마스터와 관계없이 버스 대역폭이 동일한 단점이 있다.

TDMA 방식은 마스터에게 각각 슬롯수를 다르게 배분하여 스타베이션 현상을 방지하면서 동시에 중요한 마스터의 데이터 처리를 높이는 방식인데, 2순위 중재에 의한 예상치 못한 버스사용률을 보일 수 있는 단점을 가지고 있다.

Lottery 방식의 경우는 마스터들에게 버스 점유권을 확률적으로 주는 방식이다. 중요한 마스터의 경우는 좀 더 많은 버스 점유권을 주고 그렇지 않은 경우는 좀 더 적게주는 방식으로, 최근에 TDMA 중재방식을 개선하는 방식으로 제안되고 있다.

마스터의 특성을 고려하여 데이터 전송을 수행함과 아울러, 특정 마스터의 스타베이션의 발생을 방지함으로써 데이터 전송 효율을 향상시킬 수 있는 버스 중재장치 및 방법의 개발이 요구되는데, 기존의 중재방식들은 버스 우선권을 주로 고려하였으며, 실제 버스 상에 점유율은 고려되지 않았다. 만약 용도에 따라 마스터당 버스점유율을 사용자가 할당할 수 있다면, 마스터의 성능을 직접 제어할 수 있어서 사용자가 칩을 효율적으로 관리할 수 있다. 그러나 현재 버스 대역폭을 고려한 버스 중재방식은 아직 개발되어 있지 않다.

본 논문에서는 각 마스터별 버스사용 사이클을 계산하는 카운터를 이용하여 마스터별 버스점유율을 계산하고 목표 버스점유율과 비교하는 방법에 의해 우선순위를 결정하는 점유율 버스중재 방식을 제안하고, TLM(Transaction Level Model)을 통해 기존의 다른 중재 방식과 비교하여 성능을 검증하였다.^[7]

II. 본 론

1. 점유율 고려 버스중재방식 개념

그림 1은 AMBA 아비터 블록의 입출력 신호에 대한 타이밍 도를 보여주고 있다. AMBA의 경우 마스터별 버스 점유율은 HMASTER[N:0]의 사이클 크기에 의해

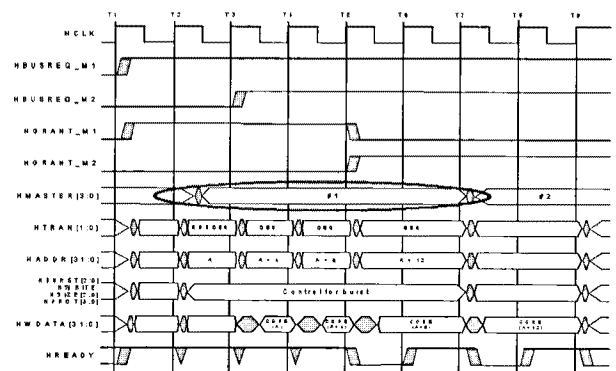


그림 1. AMBA 아비터 블록의 입출력 신호에 대한 타이밍도

Fig. 1. Timing diagram of input and output signals in AMBA arbiter block.

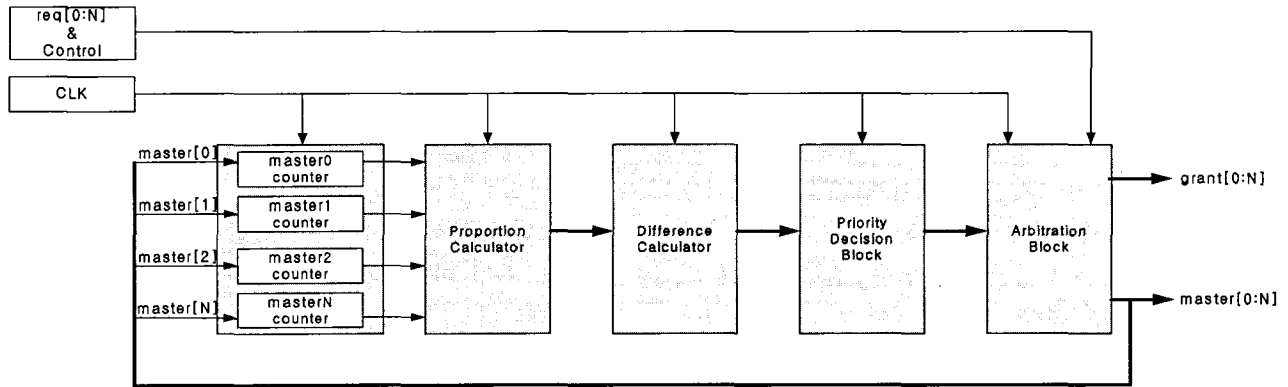


그림 2. 점유율 고려 버스중재방식 블록도
Fig. 2. Block diagram of bandwidth-award bus arbitration.

결정된다. 본 논문에서는 마스터별 HMASTER[x]의 클럭 수를 카운트하여 마스터의 버스 점유율을 계산하고, 이를 버스중재에 반영하려고 한다.

그림 2에는 점유율을 고려한 버스중재방식의 블록도가 나타나 있다. 마스터별 버스 사용 사이클 계산은 master[0], master[1], ..., master[N]의 버스요청 입력신호를 받는 master[0] counter에서 master[N] counter의 블록에서 담당한다. 카운터부 각 마스터들의 버스점유 신호(Master[x])에 따라 클럭신호(HCLK)를 카운팅하여 각 마스터들의 버스 점유 횟수를 산출한다. 각 마스터들의 버스 사용 시간을 정확하게 계산하는 카운터를 이용하여 각 마스터들의 버스 점유율을 정확하게 산출할 수 있다.

비율산출부(Proportion Calculator)는 각 카운터로부터 제공되는 각 마스터들의 버스 점유 횟수를 이용하여 각 마스터들의 버스 점유 비율을 산출한다. 이때, 카운터로부터 제공되는 각 마스터들의 버스 점유 횟수를 이용하여, 아래의 식 1을 통해 각 마스터들의 버스 점유 비율을 산출한다. 비율산출부는 소수점 첫째 자리에서 반올림 연산을 수행할 수 있다.

$$R(M[x])_{occupied} = \frac{M[x]}{T} \times 100 \quad (1)$$

마스터별 점유비율을 나타내는 식 1에서, M[x]는 마스터 x의 버스 점유 사이클 수를 나타내고, T는 마스터 1(M0), 마스터 2(M1), 마스터 3(M2), 마스터 4(M3) 각각의 버스 점유 사이클 수의 합을 나타낸다.

비교부(Difference Calculator)는 비율산출부로부터 제공되는 각 마스터들의 버스 점유 비율을 기준 점유 비율과의 차이 값을 산출하여 우선순위 결정부(Priority

Decision)에 제공한다.

우선순위 결정부는 비교부로부터 제공되는 차이 값의 크기에 따라 각 마스터들의 우선순위를 결정하여 버스중재부(Arbitration Block)에 제공한다. 이때, 우선순위 결정부는 비교부로부터 제공되는 차이 값이 큰 순으로 각 마스터들의 우선순위를 결정한다. 여기서 제공되는 차이 값이 동일할 경우 우선순위 결정부는 미리 설정된 마스터의 종류에 따라 우선순위를 결정하게 된다. 이렇게 결정된 각 마스터들의 우선순위는 다음 사이클에 각 마스터들로부터 요청되는 버스 사용 요청시에 적용된다.

버스중재부는 각 마스터들의 버스 사용 요청에 응답하여 우선순위 결정부로부터 제공되는 각 마스터들의 우선순위에 따라 버스 사용 권한 신호(grant[x])를 생성하여 해당 마스터에 제공한다. 이때, 우선순위 결정부는 복수의 마스터 중 하나의 마스터가 버스 사용을 요청할 경우에는 우선순위 결정부로부터 제공되는 각 마스터들의 우선순위에 상관없이 버스 사용을 요청한 마스터에 버스 사용 권한을 부여하고, 복수의 마스터 중 적어도 2개의 마스터가 버스 사용을 동시에 요청할 경우에는 우선순위 결정부로부터 제공되는 각 마스터들의 우선순위에 따라 마스터들의 버스 사용 권한을 부여한다.

예를 들어, 그림 3에 도시된 바와 같이 마스터 1(M0)의 버스 점유 횟수가 "9", 마스터 2(M1)의 버스 점유 횟수가 "7", 마스터 3(M2)의 버스 점유 횟수가 "4", 마스터 4(M3)의 버스 점유 횟수가 "3"일 경우, 비율산출부는 식 1을 통해 마스터 1(M0)의 버스 점유 비율을 "39", 마스터 2(M1)의 버스 점유 비율을 "31", 마스터 3(M2)의 버스 점유 비율을 "17", 마스터 4(M3)의 버스 점유 비율을 "13"으로 산출하게 된다. 그 다음에 이 데

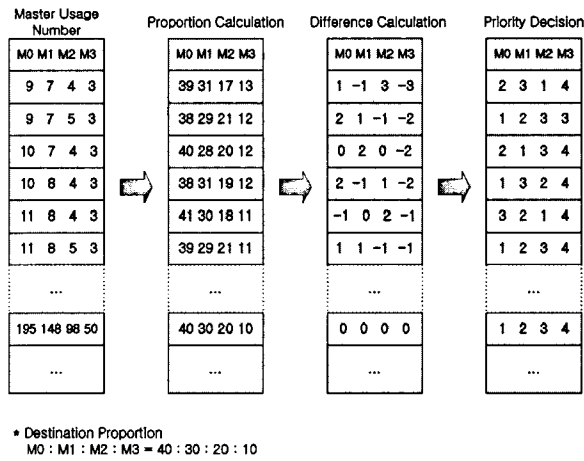


그림 3. 각 마스터의 버스 비율 상태 도 예시
Fig. 3. State machine example of bus master proportion of each master.

이터 값을 비교부에 제공하게 된다.

기준 점유 비율은 사용자의 기준 점유 비율 설정값에 따라 설정되는 것으로 마스터 1에서 마스터 4까지 각각의 기준 점유 비율을 40:30:20:10으로 가정하기로 한다.

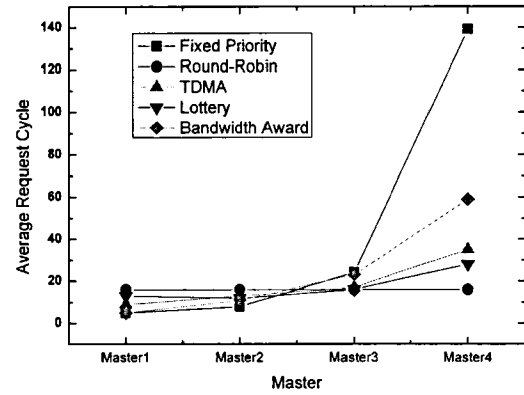
결국 비교부는 마스터 1(M0)에 대한 차이 값으로 "1", 마스터 2(M1)에 대한 차이 값으로 "-1", 마스터 3(M2)에 대한 차이 값으로 "3", 마스터 4(M3)에 대한 차이 값으로 "-3"을 산출하게 된다.

우선순위 결정부는 미리 설정된 마스터의 종류에 따라 우선순위를 결정하게 된다. 마스터 1(M0)에 대한 차이 값이 "1", 마스터 2(M1)에 대한 차이 값이 "-1", 마스터 3(M2)에 대한 차이 값이 "3", 마스터 4(M3)에 대한 차이 값이 "-3"일 경우, 마스터 3(M2)을 제 1 우선순위, 마스터 1(M0)을 제 2 우선순위, 마스터 2(M1)를 제 3 우선순위, 마스터 4(M3)를 제 4 우선순위로 결정하게 된다. 결정된 각 마스터들의 우선순위를 토대로 다음 사이클에 각 마스터들로부터 버스 사용 요청시 적용된다.

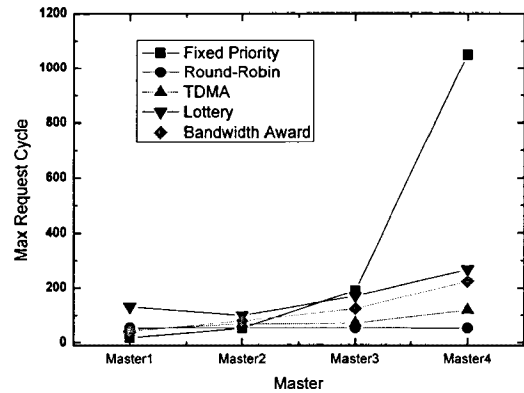
2. 성능분석

2.1 점유율과 요청사이클 비교

버스점유율과 성능분석을 위해 C++로 자체개발한 AMBA TLM(Transaction Level Model)^[8]을 사용하였다. 마스터에서 발생하는 데이터는 싱글 데이터와 버스트 데이터 타입이 있으며, 버스트 데이터 길이는 4, 8, 16까지 지원하는데, 마스터에서 발생하는 데이터 타입과 데이터 길이는 랜덤함수를 적용하여 무작위로 발생하도록



(a)



(b)

그림 4. 버스중재방식에 따른 (a) 평균 버스요청사이클과 (b) 최대 버스요청사이클

Fig. 4. According to the arbitration method (a) Average bus request cycle (b) Max bus request cycle.

하였다. 그리고 idle 사이클 지연 후 새로운 데이터를 발생시키는데, 버스트 데이터 길이와 idle 사이클에 대해서 랜덤 함수를 이용하였다.

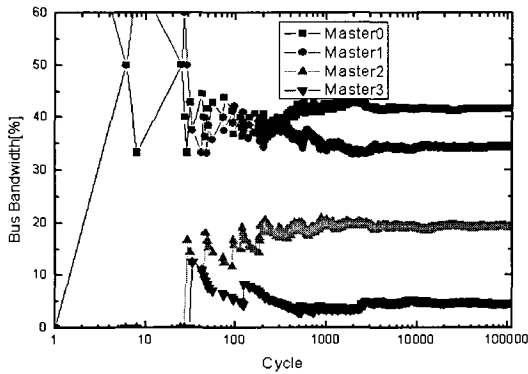
시뮬레이션 모델은 4개의 마스터와 SDRAM, SRAM, 레지스터로 구성된 4개의 슬레이브, 단일 공용버스로 구성하였으며, 복잡한 트래픽을 생성하기 위해서 마스터 트랜잭션 간의 idle 사이클은 평균값 5를 기준으로 랜덤함수를 적용하였다. 정확도를 높이기 위해서 시뮬레이션은 10,000,000 사이클까지 수행하였다.

그림 4에 버스중재방식에 따른 (a) 평균 버스요청사이클과 (b) 최대 버스요청사이클이 나타나 있다. 버스요청사이클은 버스요청을 위해 대기하는 시간을 의미하며 SoC 시스템 성능에 큰 영향을 미친다. 본 연구에서는 평균 요청사이클과 최대 요청사이클을 구분하여 결과를 비교해 보았다. Fixed Priority 중재방식은 우선순위가

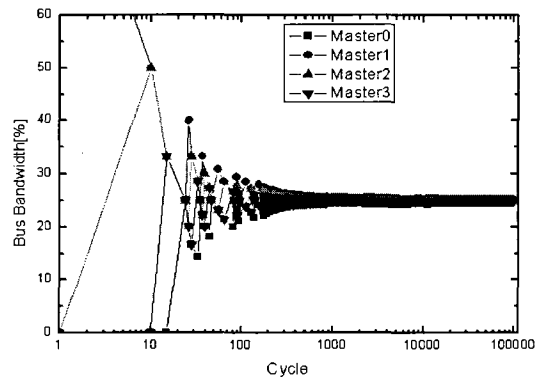
낮은 마스터4에서 요청사이클이 크게 증가하였으며, 점유율 고려 중재방식과 그 외의 중재방식은 거의 유사한 요청사이클을 나타내었다.

그림 5와 그림 6에는 시간에 따른 버스점유율 변화를 보여주고 있다. 그림 5에서 보듯이 Fixed Priority 방식

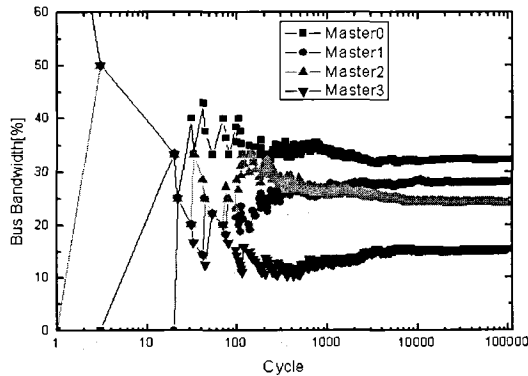
과 Round-Robin 방식은 버스점유율을 전혀 설정할 수 없으며 TDMA, Lottery 중재방식의 경우에는 100,000 사이클 이상에서 사용자가 슬롯수와 점유확률에 의해 각각 설정한 버스점유율과 실제 버스점유율이 최대 50%, 70% 차이가 나타났다. 반면에 그림 6의 우리가



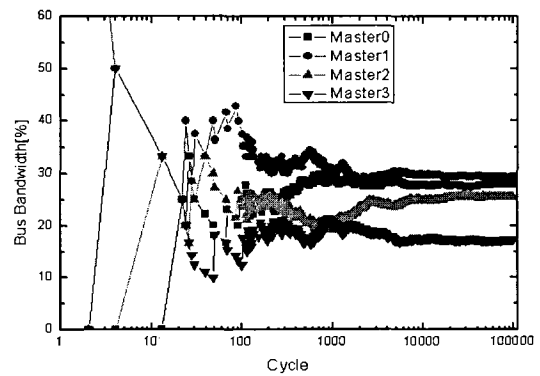
(a)



(b)



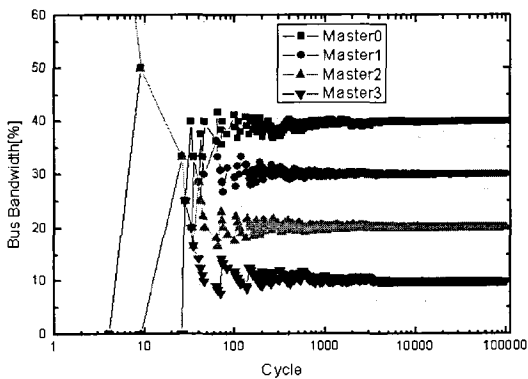
(c)



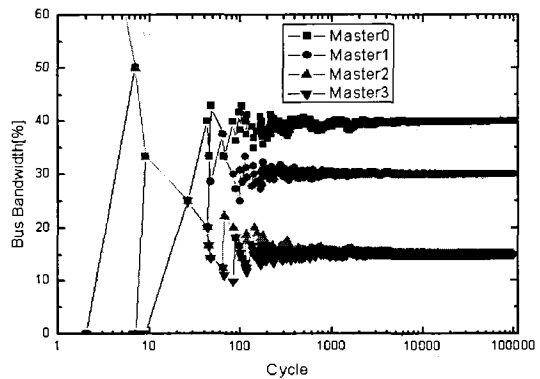
(d)

그림 5. 기존 버스중재방식의 사이클에 대한 버스점유율: (a) Fixed Priority, (b) Round-Robin, (c) TDMA(4:3:2:1), (d) Lottery (40:30:20:10)

Fig. 5. Bus utilization about conventional bus arbitration method cycle: (a) Fixed Priority, (b) Round-Robin, (c) TDMA(4:3:2:1), (d) Lottery (40:30:20:10).



(a)



(b)

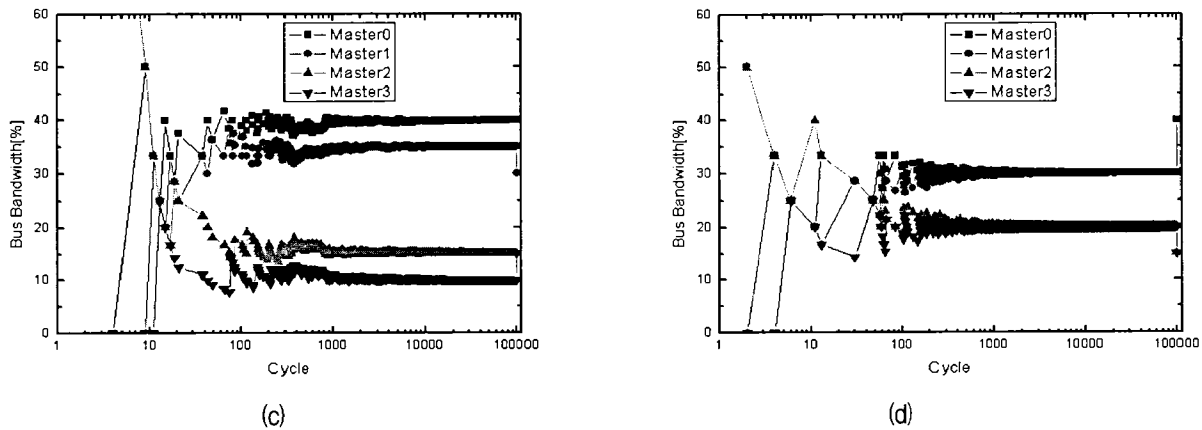


그림 6. 점유율 고려 버스중재방식의 사이클에 대한 버스점유율:

- (a) 버스점유율 설정비율 M0:M1:M2:M3 = 40:30: 20:10
- (b) 버스점유율 설정비율 M0:M1: M2:M3 = 40:30:15:15
- (c) 버스점유율 설정비율 M0:M1: M2:M3 = 40:35:15:10
- (d) 버스점유율 설정비율 M0:M1:M2:M3 = 30:30:20:20

Fig. 6. Bus utilization about bandwidth-award bus arbitration method cycle:
 (a) set the ratio of bus utilization M0:M1:M2:M3 = 40:30:20:10
 (b) set the ratio of bus utilization M0:M1:M2:M3 = 40:30:15:15
 (c) set the ratio of bus utilization M0:M1:M2:M3 = 40:35:15:10
 (d) set the ratio of bus utilization M0:M1:M2:M3 = 30:30:20:20.

제안하는 중재방식의 경우에는 사용자가 설정한 다양한 버스점유율에 대하여 약 1000 사이클 이후부터 설정한 버스점유율과 99% 이상 일치함을 알 수 있다.

III. 결 론

본 논문에서는 점유율을 고려한 중재방식을 제안하고 그 성능을 분석하였다. 점유율을 고려한 중재방식은 복수의 마스터 각각의 버스 점유율에 따라 버스 사용의 우선순위를 결정하여 버스 사용을 중재함으로써 각 마스터별 버스 점유율을 사용자가 직접 설정하여 제어할 수 있었으며, 각 마스터의 데이터 전송을 효율적으로 관리할 수 있었다. 또한, 제안하고 있는 점유율 고려 중재방식이 기존의 중재 방식들과 비교하여 사이클에 대한 버스 점유율, 평균 버스요청 사이클과 최대 버스요청 사이클 등의 성능측면에서 우수한 특성을 나타내어 새로운 버스중재 방식으로 응용될 수 있음을 확인하였다.

참 고 문 헌

[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] E. Salminen, V. Lahtinen, K. Kuusilinna, and T. Hamalainen, "Overview of bus-based system-on-chip interconnections", in Proc. IEEE Int. Symp. Circuits Syst., pp. II-372-II-375, 2002.
 [3] L. Benini and G. D. Micheli, "Networks on chips: A new SoC paradigm", IEEE Comput., vol.35, pp.70-78, Jan. 2002.
 [4] M. Jun, K. Bang, H. Lee and E. Chung, "Latency-aware bus arbitration for real-time embedded systems," IEICE Trans. Inf.& Syst.,vol .E90-D,no.3,2007.
 [5] 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.
 [6] A. Bystrov, D.J .Kinniment and A. Yakovlev, "Priority Arbiters", in Proc. IEEE 6th internation Symp. ASYNC, pp.128-137, April. 2000.
 [7] K. Lahiri, A. Raghunathan, and G. Lakshminarayana, "The LOTTERYBUS On-Chip Communication Architecture", IEEE Trans. VLSI Systems, vol.14, no.6, 2006.
 [8] K. Lee and Y. Yoon, "Architecture Exploration for Performance Improvement of SoC Chip Based on AMBA System", ICCIT, pp.739-744,

2007.

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

저 자 소 개



최 항 진(학생회원)
 2008년 세종대학교
 광전자공학과 학사졸업.
 2008년~현재 인하대학교
 전자공학과 석사과정.
 <주관심분야 : 반도체, SoC, 디지
 털 회로설계>



이 국 표(정회원)
 1999년 인하대학교 전자재료
 공학과 학사졸업.
 2001년 인하대학교 전자재료
 공학과 석사졸업.
 2005년~현재 인하대학교 전자
 공학과 박사과정.
 (박사수료)

<주관심분야 : 반도체, SoC, 디지털, 아날로그
 회로설계, FPGA 설계, 버스 아키텍처 성능분석>



윤 영 섭(정회원)
 1975년 서울대학교
 금속공학과 학사졸업.
 1977년 한국과학원
 재료공학과 석사졸업.
 1988년 Univ. Southern California
 전자공학과 박사졸업.

1987년~1988년 Oklahoma State University
 대우교수

1988년~1989년 UCLA Device Research Lab.
 연구원

1989년~1992년 삼성전자 기흥반도체연구소
 수석연구원

1992년~현재 인하대학교 전자공학과 교수
 <주관심분야 : ULSI DRAM 을 위한 신물질 개
 발, 강유전성 박막, Pyroelectric 센서, SAW
 device, 회로설계>