

논문 2009-46SD-3-9

하이브리드 버스중재방식의 설계 및 버스효율성과 요청시간에 대한 분석

(Design of Hybrid Arbitration Policy and Analysis of Its Bus
Efficiency and Request Time)

이 국 표*, 윤 영 섭*

(Kook-Pyo Lee and Yung-Sup Yoon)

요 약

우리는 fixed priority 방식에서 나타나는 스타베이션 현상 방지와 각 마스터들에게 우선순위대로 효율적으로 버스 점유권을 할당하도록 하는 fixed priority 방식과 round-robin 방식을 혼재한 하이브리드 중재 방식을 제안하였다. 제안한 중재방식과 기존의 중재방식은 Verilog와 하이닉스 0.18um 기술의 설계라이브러리를 통해 수행되었으며, 게이트 수와 설계 오버헤드를 비교하였다. 성능 분석 결과, 설계 복잡도, 타이밍 마진, 버스 점유율, 스타베이션 방지, 요청사이클 등에서 하이브리드 중재 방식이 다른 중재방식보다 성능이 우수하였음을 확인하였다.

Abstract

We propose the novel Hybrid bus arbitration policy that prevents starvation phenomenon presented in fixed priority and effectively assigns a priority to each master by mixing fixed priority and round-robin arbitration policies. The proposed arbitration policy and the others were implemented through Verilog and mapped the design into Hynix 0.18um technology and compared about gate count and design overhead. In the results of performance analysis, we confirm that our proposed policy outperforms the others in the aspect of design complexity, timing margin, bus utilization, starvation prevention, request cycle and so on.

Keywords : Arbitration, starvation, bus architecture

I. 서 론

전자산업은 현재 우리의 일상생활 속에 깊이 자리 잡고 있다. 그야말로 전자공학의 발전은 언제 어디서나 의 의미로 사용되는 유비쿼터스(Ubiquitous) 세상에 한 걸음 더 다가가게 만들었다. 유비쿼터스 세상이 실현되기 위해서는 전자제품의 소형·경량화가 이루어져야 하며, 성능이 향상되어야 한다. 이를 실현하고자 칩은 SoC(System on a Chip)로 발전하고 있다^[1]. SoC는 단일 버스 아키텍처에 여러 개의 마스터, 슬레이브, 아비

터, 디코더로 구성되어 있다. 마스터는 CPU, DMA, DSP 등과 같이 데이터 트랜잭션을 발생시키는 블록이고, 슬레이브는 SRAM, SDRAM, 레지스터 등과 같이 데이터 트랜잭션에 응답하는 블록이다. 또한 아비터는 마스터가 동시간대에 버스를 이용할 수 없기 때문에 이를 중재하는 역할을 수행하는데, 어떠한 중재 방식을 선택하는가에 따라 SoC의 성능이 크게 바뀔 수 있다. 기존의 중재 방식에는 fixed priority 방식, round-robin 방식, TDMA 방식 등이 있다^[2-5].

Fixed priority 방식은 우선순위에 따라 버스 점유권을 차등 부여하는 특징이 있는데, 가장 우선순위가 낮은 마스터의 버스 점유권을 보장할 수 없는 단점을 가지고 있다. Round-robin 방식은 모든 마스터에게 동등하게

* 정회원, 인하대학교 전자공학과
(Dept. of Electronics Engineering, Inha University)
접수일자: 2008년12월2일, 수정완료일: 2009년2월4일

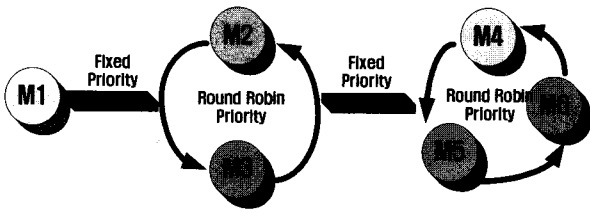


그림 1. 하이브리드 중재 방식의 예
Fig. 1. Example of Hybrid arbitration policy.

버스 점유권을 줄 수 있지만, 중요한 마스터의 데이터 처리를 빠르게 수행할 수 없다. TDMA 방식은 마스터에게 각각 슬롯수를 다르게 배분하여 스타베이션 현상을 방지하면서 동시에 중요한 마스터의 데이터 처리를 높이는 방식인데, 2순위 중재에 의한 예상치 못한 버스 사용률을 보일 수 있는 단점을 가지고 있다.

본 논문에서는 fixed priority 방식에서 발생하는 스타베이션 현상 방지와 각 마스터에게 효율적으로 버스 점유권을 할당할 수 있는 하이브리드 중재 방식을 제안하였다. 하이브리드 중재 방식의 개념은 fixed priority 방식과 round-robin 방식을 혼합하여 사용한 방식이다.

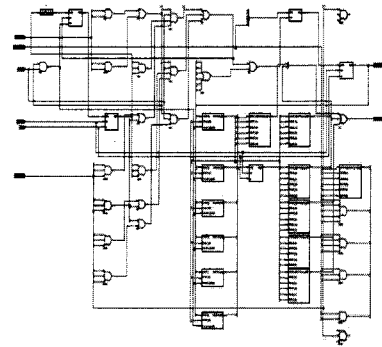
예를 들어, 그림 1을 보면 마스터 M1의 우선순위가 가장 높음을 알 수 있으며, 이는 fixed priority 방식을 적용한 부분이다. 또한, 마스터 M2와 M3의 우선순위는 동일하다. 이는 round-robin 방식을 적용한 부분으로 마스터 M4, M5, M6의 경우도 동일하다. 그리고 마스터 M1과 M2 그리고 M4는 fixed priority 방식의 관계에 있다.

본 논문에서는 첫째로, 중재 방식을 분석하기 위한 TLM 방법을 적용하였으며, 두 번째로는 다양한 중재 방식에 따른 최대 지연시간과 설계 오버헤드를 비교하였다. 셋째로는 하이브리드 중재 방식을 다른 중재 방식과 비교하여 성능을 평가하였고, 마지막으로 제안한 하이브리드 중재 방식의 효율성을 분석하였다.

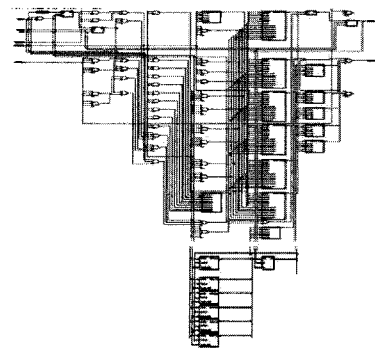
II. RTL 설계 및 분석

1. RTL 설계

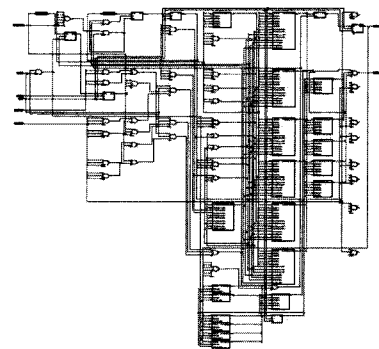
본 논문에서 우리는 임베디드시스템에 많이 사용되는 AMBA 2.0^[6] 버스 모델 인터페이스로 Verilog를 이용하여 다양한 버스중재블록을 설계하였다. 단일버스에 마스터의 개수를 6개로 두었고, 고성능 버스인 AHB 버스를 이용하였다. 기본 아비터의 면적과 최대주파수의 특성을 비교하기 위한 목적으로 설계하였으며, 스플릿



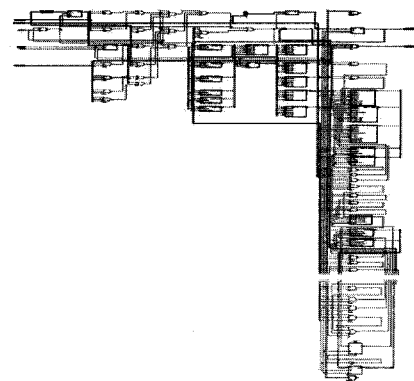
(a) Fixed priority 방식



(b) Round-robin 방식



(c) TDMA 방식



(d) Hybrid 중재 방식

그림 2. 다양한 중재 방식에 대한 로직 블록도
Fig. 2. Logic block diagram about various arbitration policies.

(Split)^[6]을 지원하지 않는 기본 AMBA 아비터 스펙을 사용하였다. 그림 2는 RTL을 합성 후 나타난 로직 블록도를 보여주고 있다.

2. 각 중재 방식의 최대 지연시간과 게이트 수 비교
제안한 하이브리드 중재 방식과 기존의 중재 방식과의 타이밍 마진과 설계 오버헤드를 비교하기 위해 Verilog가 사용되었고 Synopsys Design 컴파일러를 이용한 하이닉스 0.18um 라이브러리로 합성되었다.

표 1에는 각 중재방식에 따른 최대 지연시간이 나타나 있다. 중재방식에 따른 최대 지연시간은 1.12ns-1.16ns로 차이가 별로 크지 않다. 이를 주파수로 환산하면, 약850MHz 정도로 충분히 경쟁력있는 성능임을 알 수 있다. 합성된 넷리스트(Netlist)에서 다양한 중재 블록들은 158-260의 게이트 수를 가지고 있다. 그러나 설계면적에 대한 오버헤드는 SoC 전체 모듈을 고려하여 볼 때 상당히 작은 수준이다.

결국 RTL설계 및 합성을 통하여 우리는 제안한 하이브리드 중재 블록이 구현하기에 간단하고, 쉬웠으며, 그것의 설계면적 오버헤드와 타이밍 마진이 다른 중재 블록과 비교하여 충분히 경쟁력이 있음을 알 수 있었다.

표 1. 최대 지연시간과 게이트 수
Table 1. Max delay time and gate count.

해당 블록	최대 지연시간	게이트 카운트
Fixed Priority	1.16ns	158
Round-Robin	1.12ns	210
TDMA	1.16ns	260
Hybrid	1.12ns	220

*2-input nand gate를 1로 환산하고 사용하였음

III. 성능분석

성능분석을 위해 C++로 자체개발한 AMBA TLM (Transaction Level Model)을 사용하였다^[7].

마스터에서 발생하는 데이터는 싱글 데이터와 버스트 데이터가 있으며, 버스트 데이터 길이는 4, 8, 16까지 지원한다. 그리고 idle 사이클 지연 후 새로운 데이터를 발생시키는데, 버스트 데이터 길이와 idle 사이클에 대해서 랜덤 함수를 이용하였다.

시뮬레이션의 목적은 각 마스터의 버스 사용률과 버스 요청 사이클을 기초로 5가지 중재 방식의 효율성을

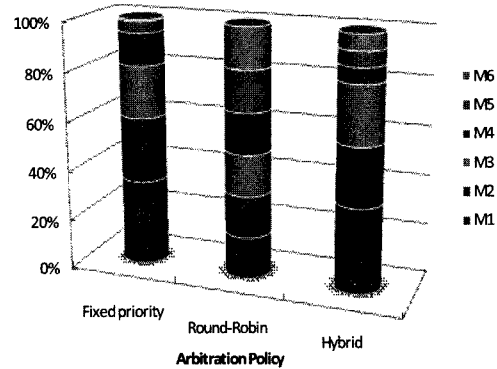


그림 3. 하이브리드 중재 방식과 비교하여 fixed priority 방식과 round-robin 방식의 버스 사용률 (Y축 버스 사용률)

Fig. 3. Bus utilization of fixed priority and round-robin, compared with Hybrid arbitration.

평가하기 위한 것이다. 시뮬레이션 모델은 6개의 마스터와 4개의 슬레이브, 단일 공용버스로 구성되어 있다. 시뮬레이션 사이클은 10,000,000 사이클로 충분히 수행하여 결과의 정확성을 높였다. 그림 3은 fixed priority 방식과 round-robin 방식 그리고 제안한 하이브리드 중재 방식의 버스 사용률을 비교하고 있다. Fixed priority 방식에서 각 마스터는 마스터 M1, M2, M3, M4, M5, M6 순으로 고유의 우선순위를 가지고 있다. 마스터 M1은 가장 높은 우선순위를 가지고 있기 때문에 처음으로 버스 점유권을 얻을 수 있다. 만약 마스터 M1이 버스 사용을 요청하지 않는다면, 마스터 M2-M6 순서대로 버스 점유권을 얻는다. 특히, 우선순위가 가장 높은 마스터 M1의 데이터 트랜잭션 사이클이 우선순위가 가장 낮은 마스터 M6보다 30배 이상 크다. 결국, 우선순위가 낮은 마스터는 버스 점유권을 얻기가 매우 어렵다. 가장 낮은 우선순위를 가지고 있는 마스터는 항상 다른 마스터와의 버스 점유권 경쟁에서 이기기 어렵기 때문에, 그 마스터는 스타베이션 현상이 발생한다. 반면, round-robin 방식에서 버스 점유권은 각 마스터에게 골고루 분배되어 스타베이션 현상은 발생하지 않지만, 마스터 중 중요한 마스터에게 우선순위를 줄 수 없는 단점이 있다. 그러나 본 논문에서 제안한 하이브리드 중재 방식은 중요한 마스터 M1의 버스 사용률이 fixed priority 방식과 동일하며, 또한 마스터의 스타베이션 현상이 발생하지 않았다. 그리고 round-robin 방식을 혼재하였기 때문에 마스터의 우선순위가 정해져 있으면서도 우선순위가 같은 마스터의 경우는 동일한 버스 사용률을 보였다.

표 2는 조건1, 조건2, 조건3, 조건4 네 가지 경우로 나

표 2 TDMA 방식의 슬롯수와 LOTTERYBUS 방식의 티켓확률
Table 2. Slot number of TDMA and ticket probability of LOTTERYBUS.

조건	슬롯 수			티켓 확률		
	M1	M2, M3	M4, M5, M6	M1	M2, M3	M4, M5, M6
조건1	3	2	1	3/10	2/10	1/10
조건2	5	2	1	5/12	2/12	1/12
조건3	5	1	1	5/10	1/10	1/10
조건4	7	2	1	7/14	2/14	1/14

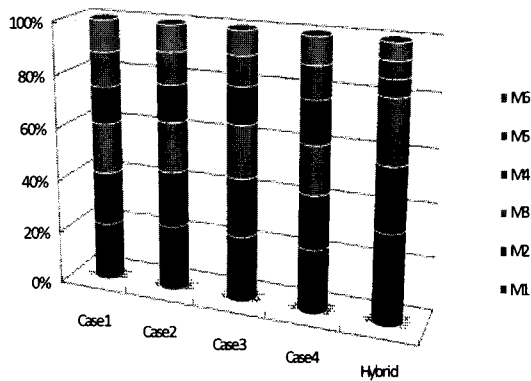


그림 4. 하이브리드 방식과 TDMA 방식의 버스 사용률 (Y축 버스 사용률)

Fig. 4. Bus utilization of Hybrid arbitration and TDMA.

주었을 때의 TDMA 방식의 슬롯수와 LOTTERYBUS 방식의 티켓확률을 보여주고 있다. 예를 들어, TDMA 방식의 슬롯수가 조건1의 경우는 마스터 M1은 3, 마스터 M2, M3은 2, 마스터 M4, M5, M6의 경우는 1로 주었다. 또한 LOTTERYBUS 방식의 티켓확률은 조건1의 경우 마스터 M1은 3/10, 마스터 M2, M3의 경우는 2/10, 마스터 M4, M5, M6의 경우는 1/10을 주었다.

그림 4는 제안한 하이브리드 중재 방식과 표 2에 보인 경우에 따른 TDMA 방식의 버스 사용률을 보여주고 있다. TDMA 방식의 조건1의 경우는 슬롯수를 마스터 M0은 3, 마스터 M1, M2는 2, 마스터 M3, M4, M5는 1로 주었을 경우에 해당한다. 이때 슬롯수를 많이 받은 마스터 M0의 버스 사용률이 약 22%로 다른 마스터에 비해 가장 높게 나타났음을 알 수 있지만 다른 마스터와 비교해도 버스 사용률이 월등히 높지 않았다. 슬롯수를 바꾼 다른 경우에 있어서도 슬롯수가 가장 많은 마스터 M0의 버스 사용률이 25%를 넘지 않았다.

그림 5는 경우에 따른 TDMA 방식의 1, 2순위 중재를 보여주고 있다. 대체적으로 1순위 중재의 데이터 트랜잭션 사이클은 주어진 경우에 따른 슬롯수와 일치하

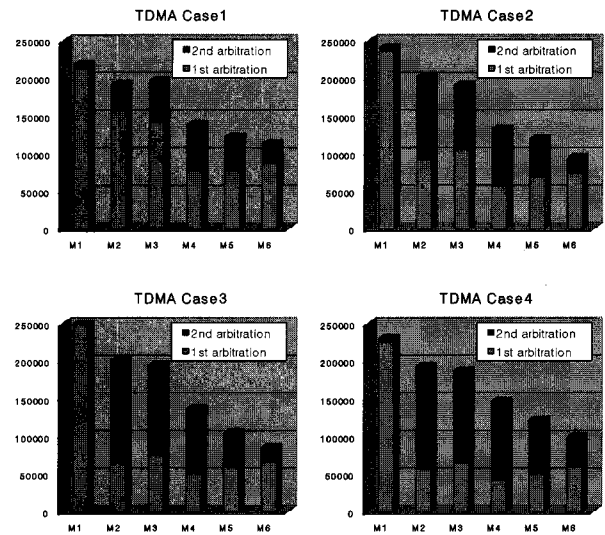


그림 5. 실험조건에 따른 TDMA 방식의 1, 2순위 중재 (Y축 데이터전송 사이클수)

Fig. 5. The 1st, 2nd level arbitration of TDMA due to experiment condition.

는데 비해 2순위 중재의 경우는 마스터 M2-M6에 따라 트랜잭션 사이클의 차이가 크게 나타났다. 이것은 2순위 중재의 특징 때문에 발생하는 현상으로, 슬롯수가 가장 많은 마스터 M1이 버스 요청을 하지 않으면, 마스터 M1에 가장 가까이 있는 마스터 순으로 버스 점유권을 받기 때문에 발생한다.^[8] 결국, TDMA 중재방식에서는 버스 점유율과 슬롯수가 일치하지 않는 경우가 발생할 수 있으며, 버스 유틸리티 제어에 한계를 나타낸다.

그림 6은 제안한 하이브리드 중재 방식과 표 2에 보인 경우에 따른 LOTTERYBUS 방식[8]의 버스 사용률을 보여주고 있다. LOTTERYBUS 방식의 조건1의 경우는 티켓확률을 마스터 M1은 3/10, 마스터 M2, M3은 2/10, 마스터 M4, M5, M6은 1/10로 주었을 경우인데, 마스터 M1의 티켓확률을 많이 주었음에도 불구하고 마스터 M3의 버스 사용률이 높았다. 조건2의 경우는 마스터 M1의 확률을 5/12로 좀 더 크게 준 결과 버스 사용

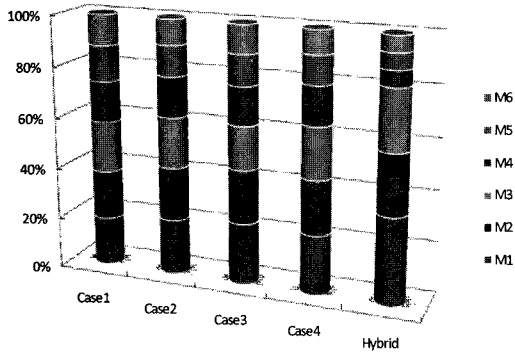


그림 6. 하이브리드 방식과 LOTTERYBUS 방식의 버스 사용률 (Y축 버스 사용률)
 Fig. 6. Bus utilization of Hybrid arbitration and LOTTERYBUS.

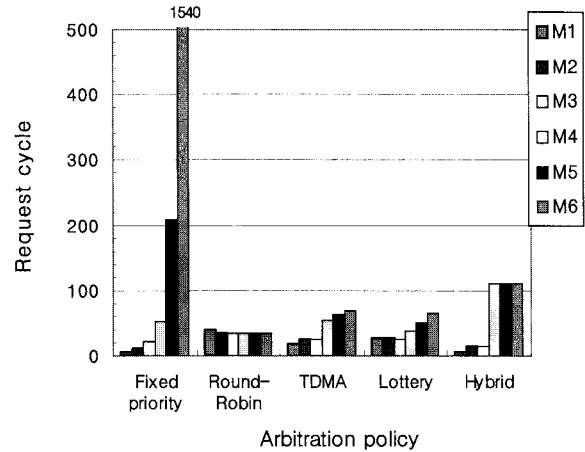


그림 8. 다양한 중재 방식의 평균 요청 사이클
 Fig. 8. Average request cycle of various arbitration policies.

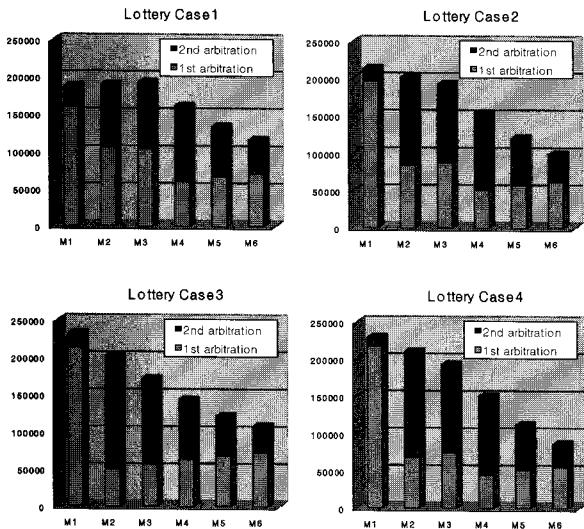


그림 7. 실험조건에 따른 LOTTERYBUS 방식의 1, 2순위 중재 (Y축 데이터전송 사이클수)
 Fig. 7. The 1st and 2nd level arbitration of LOTTERYBUS due to experiment condition.

물이 마스터 M1이 약 22%로 높았지만, 다른 마스터에 비해 월등히 높지 않았다. 조건3의 경우는 마스터 M1의 확률을 5/10로 주고, 나머지 마스터들은 1/10로 주었다. 그 결과 마스터 M1의 버스 사용률은 약 24%로 조건2보다는 2% 높아졌지만, 많이 차이나지는 않았으며, 조건4의 경우는 조건3의 경우보다 마스터 M1의 버스 사용률이 약 23%로 1% 낮았다. 전체적으로 마스터 M1의 확률을 가장 높였음에도 불구하고 마스터 M1의 버스 사용률이 25%를 넘지 못하였다. 그러나 제안한 하이브리드 중재 방식은 가장 높은 우선순위인 마스터 M1의 버스 사용률이 약 35%였다.

그림 7은 경우에 따른 LOTTERYBUS 방식의 1, 2순위 중재를 보여주고 있다. TDMA의 경우처럼 대체적으로 1순위 중재는 티켓확률과 비슷하였으나, 2순위 중재에 의한 데이터 트랜잭션 사이클이 티켓확률이 같음에도 불구하고 각 마스터들이 다름을 알 수 있다.

표 3. 하이브리드 방식과 다른 중재 방식의 비교
 Table 3. Compared results of Hybrid policy and the others.

비교항목	버스 중재방식의 종류				
	Hybrid	Fixed priority	Round robin	TDMA	LOTTERYBUS
설계 난이도	간단	간단	간단	복잡	복잡
타이밍 마진과 설계 오버헤드	우수	우수	우수	우수	보통
동일 그룹에 동일 버스점유율 부여	우수	나쁨	우수	나쁨	나쁨
스타베이션 방지	우수	나쁨	우수	우수	우수
마스터에 따른 충분한 버스점유율의 차이 확보	우수	우수	나쁨	나쁨	나쁨
버스 요청 사이클 최소화	우수	나쁨	우수	우수	우수

그림 8은 다양한 버스 중재 방식의 평균 요청 사이클을 보여주고 있다. Fixed priority 방식의 경우는 마스터 M6의 버스 요청 사이클이 1,540으로 사실상 스타베이션 현상이 발생하였음을 알 수 있으며, round-robin 방식의 경우는 약 32 정도로 모든 마스터가 비슷하였으며, 스타베이션 현상이 발생하지 않았다. TDMA 방식의 경우와 LOTTERYBUS 방식의 경우는 모든 마스터가 버스 요청 사이클이 100을 넘지 않아서 스타베이션 현상은 막았지만, 똑같은 슬롯수 또는 티켓확률을 가진 마스터들의 버스 요청 사이클이 일치하지 않았음을 알 수 있다.

그러나 본 논문에서 제안한 하이브리드 중재 방식의 경우는 마스터의 우선순위에 따른 버스 요청 사이클이 일정함을 알 수 있으며, 스타베이션 현상도 발생하지 않았다.

IV. 결 론

버스 사용률을 효율적으로 제어하기 위해 우리는 하이브리드 중재 방식을 제안하고 성능을 분석하였다. 제안한 중재 방식의 실험 결과는 이 전의 장에서 논의하였으며, 표 3과 같이 요약할 수 있다. 우리는 제안한 하이브리드 중재방식이 설계 관점, 버스효율성관점, 버스 요청 시간 등 모든 측면에 양호한 특성을 나타내어 버스중재 방식으로 응용될 수 있음을 확인하였다.

참 고 문 헌

- [1] L. Benini and G. D. Micheli, "Networks on chips: A new SoC paradigm", IEEE Comput., vol.35, pp.70-78, Jan. 2002.
- [2] 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.
- [3] 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.
- [4] A. Bystrov, D.J .Kinniment and A. Yakovlev, "Priority Arbiters", in Proc. IEEE 6th internation Symp. ASYNC, pp.128-137, April. 2000.
- [5] K. Lahiri, A. Raghunathan, and G. Lakshminarayana, "The LOTTERYBUS On-Chip Communication Architecture", IEEE Trans. VLSI Systems, vol.14, no.6, 2006.
- [6] AMBA TM Specification(AHB) (Rev 2.0), ARM Ltd, May 1999.
- [7] 이국표, 윤영섭, "마스터와 슬레이브에 따른 싱글버스와 다중버스 토폴로지의 성능분석", 전자공학회 논문지, 제45권 SD편 제9호, pp. 96-102, 2008.
- [8] K. Lahiri, A. Raghunathan, and G. Lakshminarayana, "The LOTTERYBUS On-Chip Communication Architecture", IEEE Trans. VLSI Systems, vol.14, no.6, 2006.

저 자 소 개



이 국 표(정회원)
대한전자공학회 논문지
제45권 SD편 제4호 참조



윤 영 섭(정회원)
대한전자공학회 논문지
제45권 SD편 제4호 참조