

논문 2011-48SD-9-7

# 밴드위스 고려 버스중재방식의 성능분석

(Performance Analysis of Bandwidth-Aware Bus Arbitration)

이 국 표\*, 윤 영 섭\*

(Kook-Pyo Lee and Yung-Sup Yoon)

## 요 약

전형적인 버스 시스템 아키텍처는 마스터, 아비터, 디코더, 슬레이브와 같은 성분으로 구성되어 있다. 아비터는 여러 마스터가 동시에 버스를 사용하지 못하므로 선택된 버스중재 방식에 따라 버스를 중재하는 역할을 한다. 고성능을 위해 사용되는 일반적인 우선순위 방법에는 고정 우선순위, 라운드 로빈, TDMA, 로터리 방식 등이 있다. 일반적인 버스 중재 알고리즘은 버스 점유율을 고려하지 않고, 버스중재를 실시한다. 본 연구에서는 각각의 마스터 블록에서 버스 점유율을 계산한 버스 중재방식에 대해 제안하고 있다. TLM 성능분석 방식을 통해 제안하는 방식과 기존의 다른 버스 중재방식의 성능을 분석하였다. 성능검증 결과에서 일반적인 고정우선순위와 라운드로빈 방식은 버스점유율을 설정할 수 없었으며, TDMA와 로터리 중재방법은 100,000 cycle의 시뮬레이션에서 각각 50%와 70%의 버스점유율 오차가 발생하였다. 그러나, 제안하는 점유율 고려방식에서는 1,000cycle이상에서부터 99%이상 정확도를 보였다.

## Abstract

Conventional bus system architectures are composed of several components such as master, arbiter, decoder and slave modules. The arbiter plays a role in bus arbitration according to the selected arbitration method, since several masters cannot use the bus concurrently. Typical priority strategies used in high performance arbiters include static priority, round robin, TDMA and lottery. Typical arbitration algorithms always consider the bus priority primarily, while the bus utilization is always ignored. In this paper, we propose an arbitration method using bus utilization for the operating block of each master. We verify the performance compared with the other arbitration methods through the TLM(Transaction Level Model). Based on the performance verification, the conventional fixed priority and round-robin arbitration methods cannot set the bus utilization. Whereas, in the case of the conventional TDMA and lottery arbitration methods, more than 100,000 cycles of bus utilization can be set by the user, exhibiting differences of actual bus utilization up to 50% and 70%, respectively. On the other hand, we confirm that for the proposed arbitration method, the matched bus utilization set by the user was above 99% using approximately 1,000 cycles.

**Keywords :** Arbitration, Bus architecture, SoC

## I. INTRODUCTION

Based on the development of semiconductor manufacture, the technology of SOC (System On Chip) which controls many different components is widely used in circuit design. The SOC not only can

reduce the design cost and decrease the chip size for the system design, but can also provide advanced performance with low power consumption and real-time handling capacity as well as system flexibility. The SOC consists of one arbiter and additional masters and slaves. Between the master and slave is a shared bus, which is used for transmitting data. Generally, the master is a CPU, DMA (Direct Memory Access) or DSP (Digital Signal Processor). However, with a slave such as an

\* 정회원, 인하대학교 전자공학과

(Dept. of Electronics Engineering, Inha University)

※ 이 논문은 인하대학교의 지원에 의하여 발간되었음  
접수일자: 2011년3월4일, 수정완료일: 2011년8월17일

SDRAM memory similar to an SRAM and USB (Universal Serial Bus) or UART (Universal Asynchronous Receiver Transmitter) can translate data between parallel and serial forms of data. If the SOC is designed for chip integration with many kinds of functions, it needs a bus system that connects to a master, arbiter and slave, etc. If many masters want to use the bus concurrently, they will need to be allowed to transmit/receive data to/from the slave in turn, after receiving the grant signal from the arbiter. Thus, the arbiter should create priorities determining which master is allowed to initiate data transfers, but the chip function can be changed by using different arbitration methods.

In an arbiter, some priority methods are frequently used such as the fixed priority method, the round robin method, the TDMA method, the lottery method etc.<sup>[1~6]</sup>

Fixed priority is a method that all masters have a fixed priority for obtaining access rights .i.e., each master has a certain process priority, and the order of DSP, DMA1 and DMA2 are fixed to 1, 2 and 3. When some masters want to access the bus concurrently, the master with the highest priority will be permitted access. This fixed priority method is that master does not consider the processed data character, but due to the master's fixed priority data process time cannot be achieved. This is due to some disadvantages such as the fact that a master with low priority will suffer from starvation in spite of the bus bandwidth usage.

A round robin is an arrangement for choosing all elements in a group equally in some rational order, usually from top to bottom of a list, and then choosing again from the top of the list etc. (i.e., one method for dealing with different masters that take turns using the bus is to limit each process to a certain short time period, then suspend it in order to give another process a turn). The round-robin has no fixed priority, only referencing the time allocation to ensure priority, a master with vital information can be granted timely access without priority, because all

of the masters have the same access rights and the bandwidth appears to be the same. TDMA is a method that can quickly process the data owned by a vital master by distributing every slot differently and preventing starvation. However, it cannot well handle bandwidth.

The lottery is a method that can provide the master bus with an access probability. It can provide the vital master more access. Otherwise, an unimportant master will have less access rights. This method was proposed by developing the TDMA method.<sup>[7]</sup>

Because of the master types and data transmission, as well as the master starvation problem, we need an arbitration method for the bus system that can transmit data efficiently. The basic arbitration method mainly considers bus priority, but the bandwidth is not considered. If a user can distribute the bus access rights to a master according to the differences in usage, the performance of the master can be controlled and the chip can be managed well by the user. But so far, no such arbitration method considering bandwidth has been developed.

This paper presents an arbitration method that uses a counter to count bus cycle and occupied bus rate, by comparing the occupied rate the master can obtain with a priority. Though the TLM (Transaction Level Model), a character comparison between bandwidth-aware arbitration and the other methods can shown<sup>[8]</sup>.

## II. Concept of bandwidth-aware bus arbitration

Fig. 1 depicts the signal's timing of the AMBA arbiter block. Every master of AMBA is confirmed by the cycle dimension of the occupied rate of bus HMASTER[x]. This paper proposes an arbiter method which counts the master's clock and its bus occupation rate.

Fig. 2 depicts a block diagram of bandwidth-aware bus arbitration. In this block, master [0] counter to

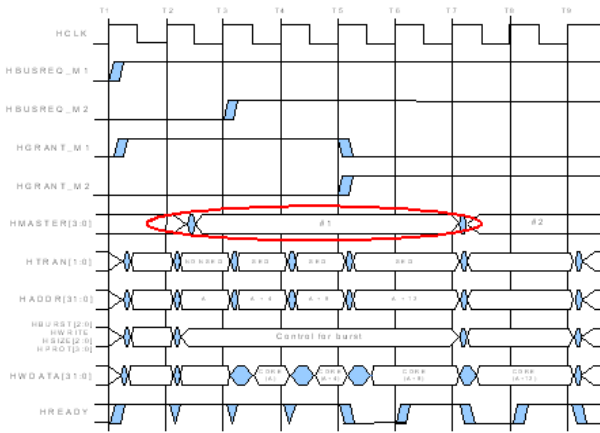


그림 1. AMBA 아비터 블록의 입력, 출력 신호의 타이밍도  
Fig. 1. Timing diagram of input and output signals in AMBA arbiter block.

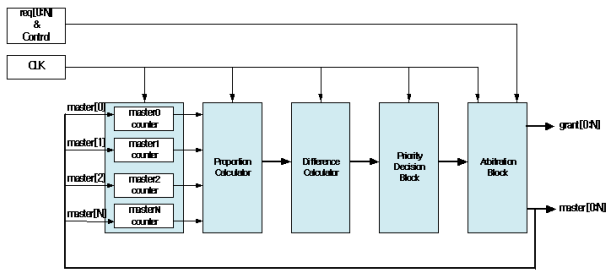


그림 2. 밴드위스 고려 버스 중재방식의 블록도  
Fig. 2. Block diagram of bandwidth-aware bus arbitration.

master [N] counter simply involves receiving a bus request signal from master [0] to master [N], which counts the bus cycles used. In the counter block, the HCLK signal is counted through Master[X] which is the bus access signal of the master. Then the number of master’s bus accesses can be counted. A proportion calculator uses the number from the counter to calculate the bus proportion. Equation (1) indicates how to calculate the proportion, and the value can be rounded off to one decimal place.

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

In this equation, M[X] is the number of master X’s occupied cycles. T is the sum of each master’s (e.g., master 0(M0), master 1(M1), master 2(M2), master 3(M3)) occupied bus cycles.

A difference calculator block is a block that gets a minus value between the proportion calculator value

and the standard proportion value. Then it can give the minus value to a priority decision block.

A priority decision block can determine the master’s priority through the minus value from the difference calculator block, and give the result to the arbitration block. When the minus values are the same, the priority decision block determines the priority by the type of master set up in advance. This master’s priority will be applied to each master’s request in the next cycle.

An arbitration block not only responds to the bus used request of each master, but also provides a bus use right signal (grant[x]) by the priority from the priority decision block. When one master requests a bus for use, it is unrelated to the priority which forms the priority decision block, but when two or more masters concurrently request a bus for use, it will be granted by different priorities from the priority decision block.

For example, Fig.3 shows when master 0(M0)’s occupied bus number is “9”, master 1(M1)’s occupied bus number is “7”, master 2(M2)’s occupied bus number is “4”, master 3(M3)’s occupied bus number is “3”. The proportion calculator block can calculate the occupied bus rate of each master; master 0(M0)’s is “39”, master 1(M1)’s is “31”, master 2(M2)’s is “17”, master 4(M3)’s is “13”. This data will be

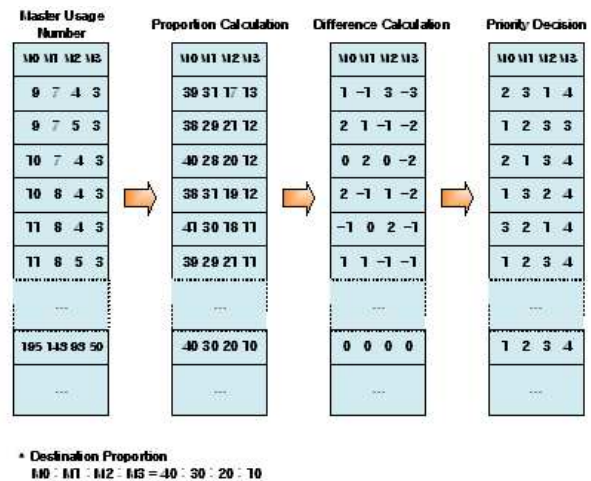


그림 3. 각각 마스터의 버스비율의 스테이트 머신 예  
Fig. 3. State machine example of each master’s bus proportion.

provided in the difference calculator block.

The standard occupied proportion is set by a user who is aiming for a different occupied bus proportion from master 0 to master 3 of 40:30:20:10.

The standard occupied proportion is set by a user who is aiming for a different occupied bus proportion from master 0 to master 3 of 40:30:20:10.

The priority decision block determines the priority depending on the master's type in advance. For these difference values, master 0(M0)'s is "1", master 1(M1)'s is "-1", master 2(M2)'s is "3" and, master 3(M3)'s is "-3", where master 2(M2) has the highest priority, followed in turn by master 0(M0), master 1(M1), and master 3(M3). Each master's priority will be applied to each master's request in the next cycle.

### III. Performance analysis

#### 1. Comparison of occupied rate and request cycle

In order to measure the occupied bus rate and perform the performance analysis, we used the AMBA TLM (Transaction Level Model) which was developed using C++.[8] The data generated in the master is of single or burst type, and the burst data supported is of length 4, length 8 and length 16. The data type and length can be generated randomly using a random function. New data will be generated and a random function will be used for the burst data length and idle cycle value.

The simulation model consists of four masters and SDRAM, SRAM, and four slaves comprising the register. In order to generate a complex traffic, a random function was used for the idle cycle value, and an average value of 5 between master transactions was applied. In order to accurately confirm the results, the final value is set to more than 10,000,000 cycles in our simulation.

Fig.4 (a) shows average bus request cycle value according to arbitration methods. Fig.4 (b) shows max bus request cycle value according to arbitration methods. The bus request cycle value is the wait time of a bus request; it seriously influences the SOC

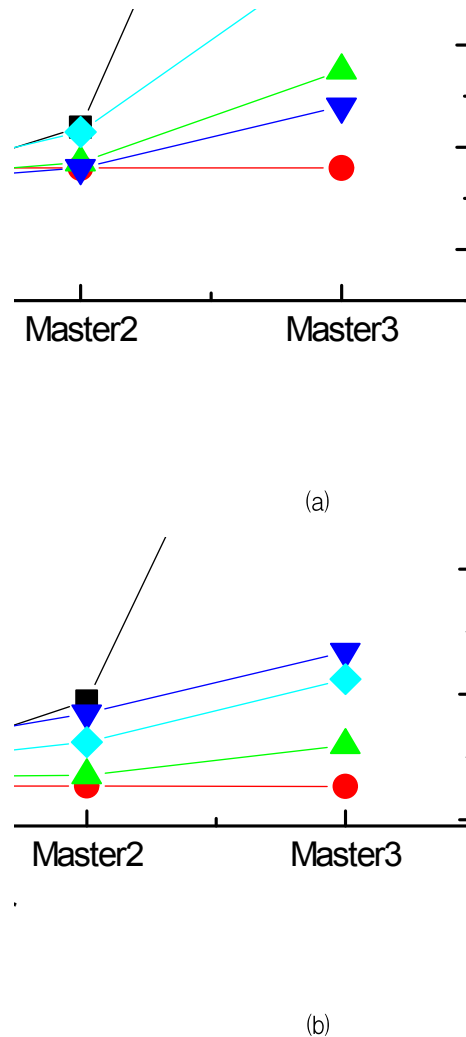


그림 4. 버스 중재방식에 따른 (a) 평균 버스 요청 사이클과 (b) 최대 버스 요청 사이클

Fig. 4. Arbitration method according to (a) Average bus request cycle value (b) Max bus request cycle value.

system performance. In this paper, we show comparison results for the average and max bus request cycle values. In fixed priority arbitration, master 3 has the lowest priority, and the request cycle value is increased significantly. Bandwidth-aware arbitration and the other methods have similar request cycle values. Fixed occupied bus rate and actual occupied bus rate of 50% and, 70% respectively. On the other hand, Fig.7 shows that the proposed arbitration method can 99% match the occupied bus rate after 1,000 cycles.

2. Throughput analysis

In order to understand the performance of the proposed bus arbitration method, the IEEE 802.11 network SOC was used. For the performance analysis, we removed blocks unrelated to performance and simplified, as shown in Fig. 5<sup>[9]</sup>.

In order to analyze the performance of Ethernet MAC, although the transmission performance of TX/RX should be considered, we should also consider the shared bus performance, which is the reason the main character is dropped. When the ARM940T processor orders the I/D cache, a simulation is performed with I/D cache in the ON state. While the cache is in the ON state, its data size is fixed to 32bits and the burst dimension is fixed to 8bit.

In this simulation, for a high performance data transaction, the Ethernet MAC's data size and burst dimension are set as 32bit and 8bit respectively. A random function with an average value of 5 for all idle cycles of all masters is applied; therefore the bus traffic of the shared bus increases.

In order to indicate a high character, the ARM940T controlling the entire system is set in advance to obtain a 40% bus occupied bus rate, and the other three Ethernet MACs are set in advance to obtain a 20% occupied bus rate. In the TDMA arbitration method, the slot number of the ARM940T processor and Ethernet MAC are given as 4, 2, 2, 2, and for the Lottery bus arbitration method, the bus request probability is given as 40%, 20%, 20% and 20%. Finally, we proposed a bus bandwidth-aware

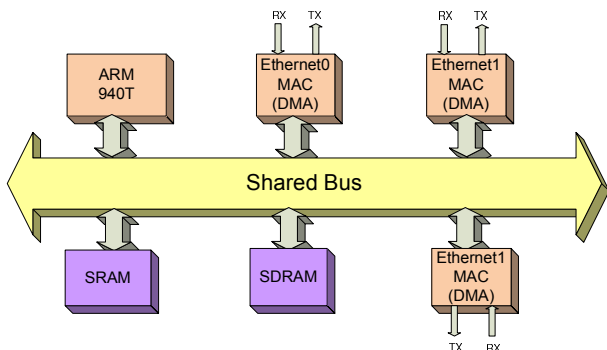
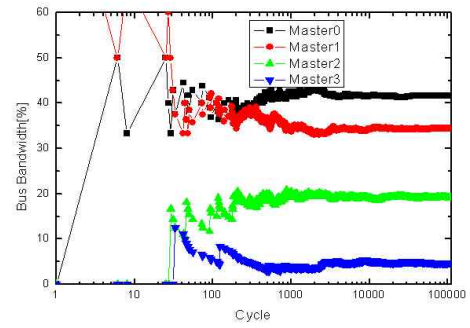
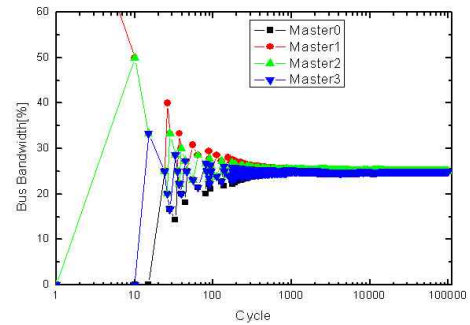


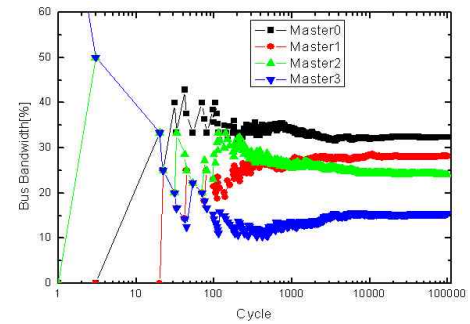
그림 5. 응용 환경  
Fig. 5. Application environment.



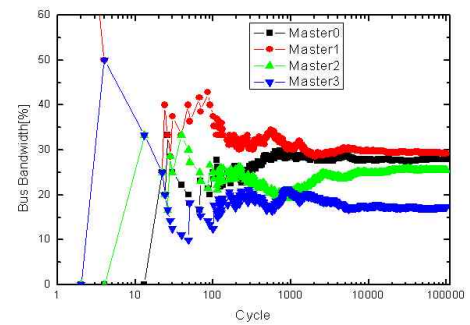
(a)



(b)



(c)



(d)

Fig. 6. 일반적인 버스 중재 방식에 대한 버스 점유도 : (a) Fixed priority, (b) 라운드 로빈, (c) TDMA(4:3:2:1), (d) 로터리 버스 (40:30:20:10)  
Fig. 6. Bus utilization about conventional bus arbitration method cycle: (a) fixed priority, (b) round-robin, (c) TDMA(4:3:2:1), (d) lottery bus (40:30:20:10).

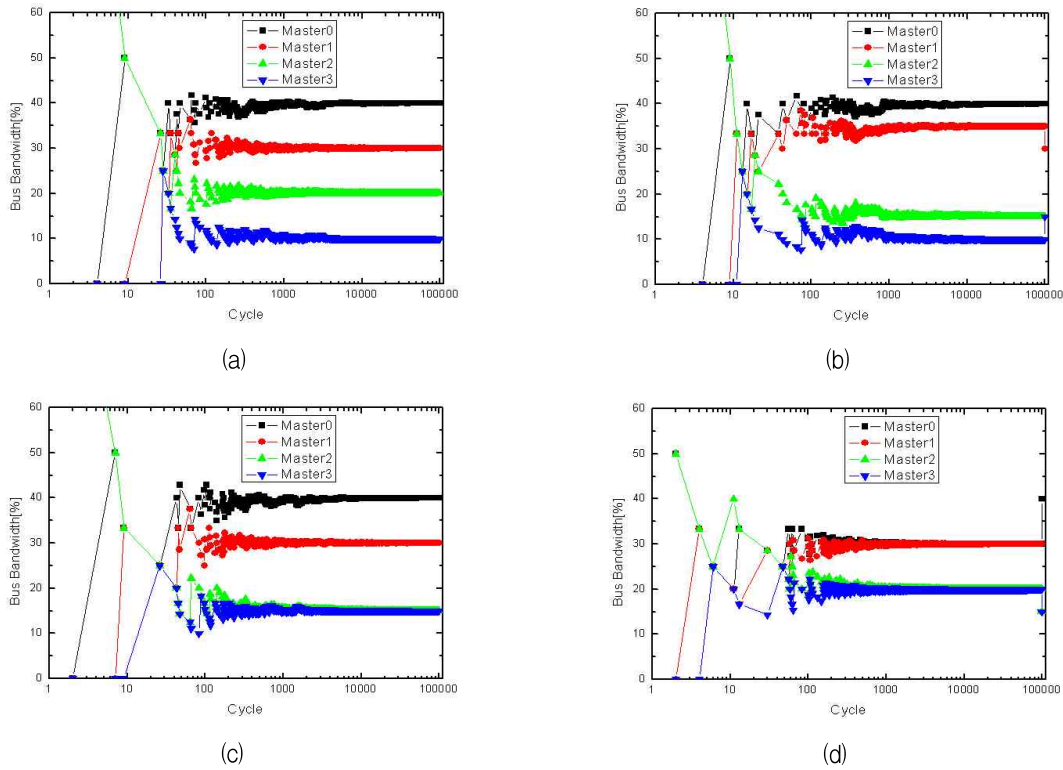


Fig. 7. 밴드위스 고려 버스 중재방식의 점유율 : 버스점유비율 M0:M1:M2:M3 = (a) 40:30:20:10, (b) 40:35:15:10, (c) 40:30:15:15, (d) 30:30:20:20

Fig. 7. Bus utilization about bandwidth-aware bus arbitration method cycle: (a) ratio of bus utilization M0:M1:M2:M3 = 40:30:20:10 (b) ratio of bus utilization M0:M1:M2:M3 = 40:35:15:10 (c) ratio of bus utilization M0:M1:M2:M3 = 40:30:15:15 (d) ratio of bus utilization M0:M1:M2:M3 = 30:30:20:20 .

arbitration method which is set in advance to obtain an occupied bus rate of 40%, 20%, 20% and 20%.

$$Performance [bit/s] = \frac{N_{trans} N_{burst} N_{bit}}{T} \quad (2)$$

Equation (2) describes the performance of the master where  $N_{trans}$  is the total transmitted data,  $N_{burst}$  is the burst dimension,  $N_{bit}$  is the total number of data bits, and  $T$  is the total time.

Fig. 5 is the simulation of the application environment, and its result is shown in Fig.8. For the fixed priority method, the character deviation of each master is too large to transmit successfully.

For the round-robin method, the performance of all masters has an occupied bus rate equal to 0.55Gbps. As a result, the ARM940T processor will achieve an occupied bus rate twice that of other masters, but this goal is unsatisfactory.

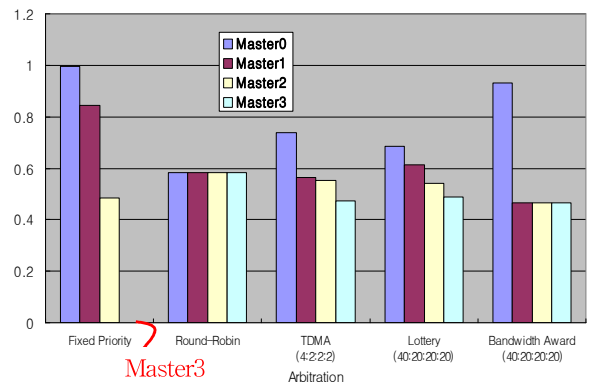


Fig. 8. 버스 중재방식에 따른 성능 비교

Fig. 8. Throughput comparison according to arbitration method.

The simulation results of TDMA and the Lottery bus method are contrary to our expectations. In this paper, our proposed bandwidth-aware method achieves an occupied bus rate of 40%, 20%, 20% and 20% as we expected.

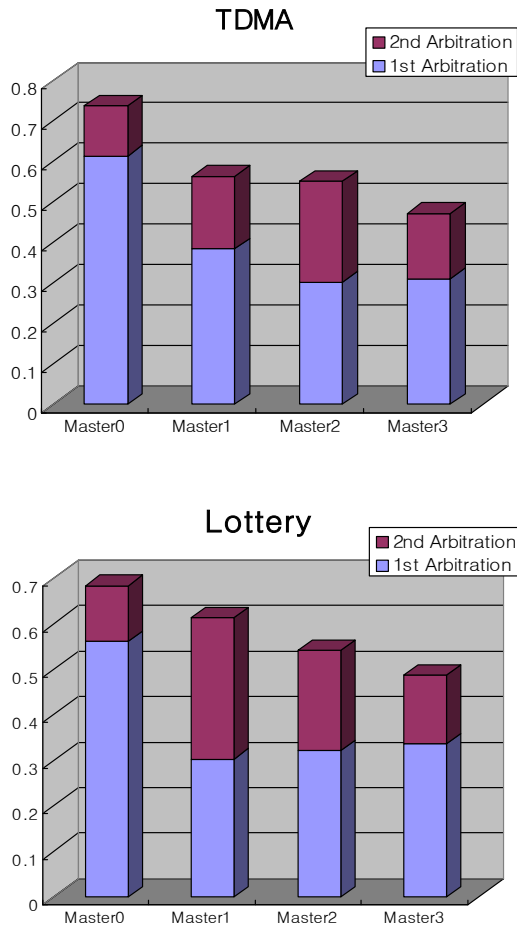


그림 9. TDMA와 로터리 버스 중재에 따른 1순위, 2순위 버스중재 결과

Fig. 9. 1st and 2nd arbitration results of TDMA and Lottery bus arbitration policy.

In order to analyze why TDMA and the Lottery bus methods fail to adjust the occupied bus rate, we present Fig.9, which shows the 1st and 2nd arbitration results of TDMA and the Lottery bus method. The 1st arbitration result determines whether the user sets the slot number or the bus arbitration probability. If the master from the 1st arbitration result does not have a bus request, no user request can generate the 2nd arbitration result.

For TDMA and the Lottery bus shown in Fig.9, the 1st arbitration result's data transaction cycle value is almost the same as the previously fixed slot number (4,2,2,2) and bus arbitration rate (40:20:20:20). However, the 2nd arbitration result indicates a difference between the slot number and bus

arbitration rate. As a result, we can understand TDMA and Lottery bus, as a two level arbitration is difficult to control the bandwidth-aware.

#### IV. Conclusions

In this paper, we propose a method of bandwidth-aware arbitration and analyze its characteristics. The bandwidth-aware arbitration method can determine the bus priority according to the occupied bus rate. The occupied bus rate of each master can be controlled by the user, so the master's data transactions can be managed effectively. The bandwidth-aware arbitration method we proposed not only provides a much better occupied bus rate than other arbitration methods, but also provides a good character etc. It is proved to be a great architecture method with excellent performance.

#### REFERENCES

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

---

— 저 자 소 개 —

---



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



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