IP Based Design 2003

Session : OPEN FORUM ON DESIGN METHODOLOGY

An Analysis and Implementation of High Fairness Arbitration Mechanism by Using Level-table and Static Priority Orders in Shared Bus Architecture

Jisuhn Suh, Jongsun Kim, Hoi-Jun Yoo
System Integration & Intellectual Property Authoring Center
Korea Advanced Institute of Science and Technology
Daejeon Korea

Abstract :

In this paper a high fairness arbitration mechanism is proposed for shared bus architecture in a system-on-chip, and analyzed in terms of bandwidth and throughput. The proposed scheme is compared with other arbitration mechanisms, such as static priority arbitration and round-ring priority arbitration. We conclude the high fairness arbitration mechanism is better to control bandwidth and to manipulate for reuse the arbiter than other arbitration mechanisms. We also provide an efficient method for hardware implementation, in which the high fairness arbiter has level-table for rate of grant per request and a change of static-priority order simply. VisualElite from Summit Design and ModelSim from Mentor are used for communication architecture design and for HDL simulation, respectively.

Introduction :

System-on-chip is getting more complex and requires more multi-processor architecture. As an SoC era of 100M gates is coming up, the numbers of IPs used in this SoC are about 1000 of size of ARM7.

In an multi-processor SoC with shared communication architecture, the scheduling for restrictive communication channel is very important to implement a high performance SoC.

The arbiter is a core IP that schedules the shared bus. It has three main functions.

i) No contention among masters which use the shared bus. ii) Guarantee the bandwidth of data flows of each master. iii) Improvement of the performance of throughput or latency in shared bus. The arbitration mechanisms for no-contention among masters are classified into three categories i) static priority mechanism ii) round-ring priority mechanism [1] iii) ordered priority mechanism. The arbitration mechanisms to guarantee the bandwidth of each master are also classified into three categories i) token-ring mechanism ii) time-division multiplexed mechanism [2] iii) lottery mechanism [3]. The arbitration mechanism to improve the performance by scheduling data flows are i) pipeline mechanism ii) multi-layer mechanism iii) network mechanism.

Many SoC designers have said that the fairness of arbiters is important in arbitration mechanism. However, there is no analysis or implementation in detail for fairness arbitration mechanism. Therefore we analyze fairness arbitration compared with other arbitration mechanism. The performance metrics are bandwidth of each master and total throughput.

To analyze the fairness arbitration in shared bus architecture, we use the VisualElite from Summit design Inc., C++, and Rendezvous protocol.

Fairness arbitration mechanism :

Fairness property guarantees that a request is granted after finite numbers of other requests are granted. We define the fairness ratio as shown in formula (1)

$$\textit{Fairness Ratio} = GR_{min} / GR_{max} \quad \text{--------- (1)}$$

Fairness Ratio = 1 : ideal fairness
Fairness Ratio = 0 : very poor fairness

$$GR_i \textit{ (grant ratio)} = (\textit{number of grant})_i / (\textit{number of request})_i$$

i : master ID (ex. i = 0 to 7)
GR_{min}***: GR of master that has minimum GR value***
GR_{max}***: GR of master that has maximum GR value***

To get ideal fairness ratio, an arbiter uses the policy to grant to the master that has the smallest GR among requesting masters. C-code for priority

decision made by fairness arbitration policy is shown below.

```
int fair_arbitrate (bool *Request[], float GR[])
{
   int i, min, count;
   min = -1;
   count = 0;
for (i = 0; i < number_max_masters ; i++ )
 {
  if (*(Request[i]) == true)
    {  count++;
          if (count == 1) min = i;
          else if (GR[min] > GR[i]) min = i;
    }
 }
 return (min);
}
```

Analysis of fairness arbitration mechanism :

The block diagram of control flow with arbiter in a shared bus is shown in figure 1.

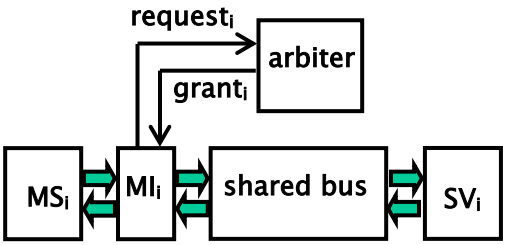

Fig. 1 Block diagram of control flow with arbiter in shared bus

We use 8 masters and 2 slaves for simulation of the arbitration mechanism. Request signals of masters are generated by source from VisualElite library with Gaussian distribution. Clock frequency of a bus is 200MHz, the fastest average request frequency of processor is 20MHz and the slowest is 2MHz. We use 3 types of arbitration mechanisms such as static priority arbitration, round-ring arbitration and fairness arbitration.

In static-priority arbitration mechanism, the starvation problem seriously occurs when the priority number is under 5. In round-ring arbitration mechanism, maximum distribution of bandwidths is about 3, although the maximum distribution of number of requests are about 8. However maximum distribution of bandwidths is about 6 multiples in fairness arbitration mechanism as shown in table 1. This results shows the fairness arbitration provide the similar grant distribution with request distribution.

	Request	Grant by Ring	Grant by Fair
Distribution among masters (max / min)	8.3	2.8	6.2

Table 1. Distribution of request and grant among masters

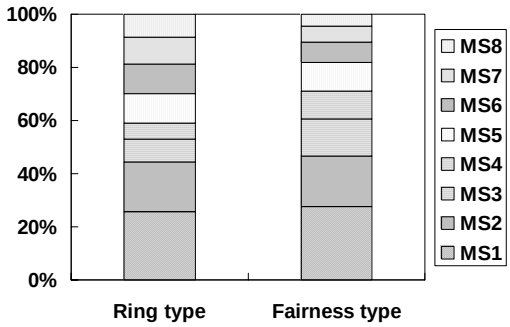

Fig. 2 Bandwidth of ring arbitration and Fairness arbitration

The bandwidths of masters are shown in figure 2. Figure 2 shows that the bandwidth resulted by ring arbitration mechanism is not guranteed the request bandwidth, however fairness arbitration mechnism is guranteed.

The time of communication channel, T_channel_comm, is defined as shown in figure 3. Figure 4 shows throughput and T_arb at each master with ring arbitration mechanism and fairness arbitration mechanism respectively. The arbitration time, T_arb, of masters having low frequent request, MS6-8, is large enough in fairness arbitration mechanism. This is why access time for granting a bus increases to fit high fairness ratio.

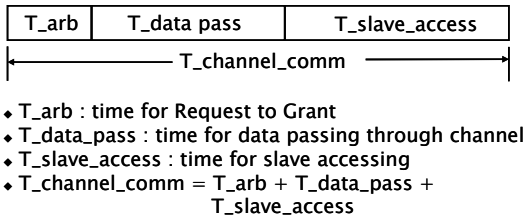

Fig. 3 Definition for time of communication channel

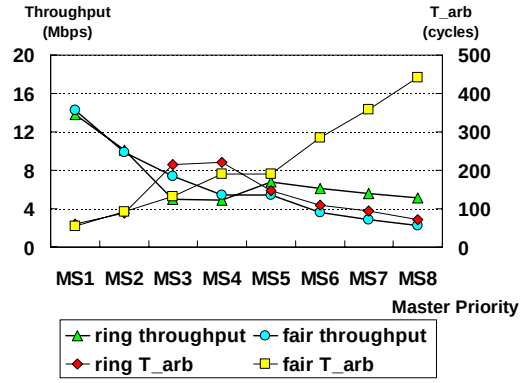

Fig. 4 Throughput and T_arb of each masters having arbiter of ring and fairness arbitration

Figure 5 shows throughput and T_arb with number of masters. T_arb is increased with 8 masters in fairness arbitration, so the throughput of data is decreased by 10%.

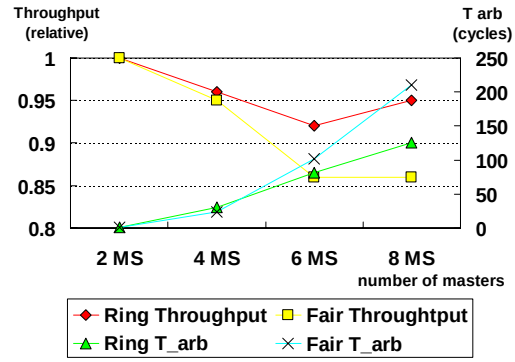

Fig. 5 Throughput and T_arb with number of masters

The bandwidth for masters that requires high priority is not guaranteed in round-ring arbitration mechanism. To overcome the lack of control over allocation of bandwidth, TDM and Lottery mechanism are provided. The TDM mechanism has the reserved time slot for a unique master. And the Lottery mechanism probabilistically chooses a master by assigned number of tickets. However, all of two methods require careful control of time slots or the number of tickets according to placement of masters in SoC architecture. However, fairness arbitration mechanism is good to guarantee the required bandwidth by the amount of requests. Also it is not required to control parameter of arbiter to guarantee the bandwidth. Fairness arbiter is easy to use for the design of SoC that have many masters and to reuse the arbiter in the platform architecture design. However the increase of arbitration time decreases the total throughput of system that have more than 6 masters.

Implementation of high fairness arbiter :

The block diagram of the high fairness arbiter architecture is shown in figure 6.

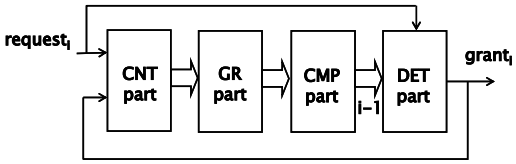

Fig. 6 Architecture of high fairness arbiter

This architecture consists of 4 parts. i) CNT part : counts the number of requests and grants. We used 6 registers for counting the number of requests and grants, respectively. This counting number of 6 provides the 13 level of GR part and a 4bit comparator that is optimal in size and delay [4] ii) GR part : generate the level of grant ratio. To get the ratio of grant per request, we use level-table as shown in table 2. This method reduces size and delay of an arbiter. The important point of level-table is which master has the higher value rather than how master has exact value. Example, difference between level 1 and Level 2 is 3%, and difference between level2 and level3 is 5%. However the step of level 1, 2 and 3 are equal.

Masters which have level 0 will become the highest priority at next request. iii) CMP part : a) compares the Grant Ratio among masters b) compares the static priority among masters if Grant Ratio among requested masters is equal iv) DET part : a) grants to the win master b) changes the static priority of each master in orders of low level at the reset time of request count. The reason to change static priority order of masters is to reduce error rate caused by limited number of register for counting request and grant. The fairness ratio of an arbiter that has the change mode of static priority is maximally 48% higher than that arbiter that does not have change mode of static priority.

To grant the current request, we compare the levels of Grant Ratio which was calculated in previous request time. Therefore there is no latency for comparing the level among masters.

Level	(Num of Grant) / (Num of Request)	Grant / Request (%)
0	0/n	0
1	1/6	17
2	1/5	20
3	1/4	25
4	1/3, 2/6	33
5	2/5	40
6	1/2, 2/4, 3/6	50
7	3/5	60
8	2/3, 4/6	67
9	3/4	75
10	4/5	80
11	5/6	83
12	n/n	100

Table 2. Level-table for calculation of ratio of grant per request

We implemented high fairness arbiter for 4 masters, because the performance of high fairness arbiter that controls 4 masters is optimal. We compare the fairness ratio with other arbitration mechanism. Figure 7 shows the results of fairness ratio. The increase orders of high fairness ratio : static < round-ring < lottery < TDM < fairness.

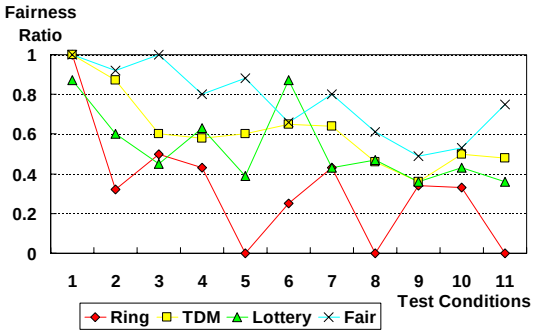

Fig. 7 The comparison fairness ratio with other arbitration mechanism

Conclusion :

SoC which have high Fairness arbiter, less than 4 masters and shared bus architecture represent good performance in bandwidth and throughput. In high fairness arbitration mechanism, the bandwidth among masters have the same distribution as request distribution. So it is not required to control the arbiter parameter for adjusting the bandwidth of masters. The total throughput of SoC that have high fairness arbiter is no loss under 4 masters. However, in SoC which have over than 5 masters, the total throughput is lower than ring arbitration mechanism.

And we implement the high fairness arbiter for 4 masters that have low latency and size. We have a minimim fairness ratio of 0.49 at boundary condition of request counting number. The fairness ratio of implemented by level-table and change mode of static priority are 26% up compare to other arbitration mechanism.

Reference :

[1] E. S. Shin, V. J. Mooney III, G. F. Riley, "Round-robin arbiter design and generation," Proc. in ISSS 2002

[2] Sonics, " Sonics µNetwork Technical Overview," http://www.sonicinc.com/

[3] K. Lahiri, A. Ranhunathan, G. Lakshminarayana, "Lottery: High performance communication architecture for system-on-chip designs," in Proc. DAC, pp 15-20, Jun. 2001

[4] Chung-Hsun Huang, Jinn-Shyan Wang, " High-Performance and Power-Efficient CMOS Comparators," in IEEE Journal of Solid-State Circuits, VOL.38, NO2, Feb. 2003