International Journal of Engineering Research & Management Technology ISSN: 2348-4039

March- 2015 Volume 2, Issue-2

Email: editor@ijermt.org www.ijermt.org

Design and Performance Analysis of Efficient Bus Arbitration Schemes

Samrat Nandi
M.tech Scholar
Department of ECE
SRM University NCR Campus
Modinagar

Mr. Amit
Assistant Professor
Department of ECE
SRM University NCR Campus
Modinagar

ABSTRACT—
On-chip communication architecture plays an important role in determining the overall performance of the System-on-Chip (SoC) design. In the recourse sharing mechanism of SoC, the communication architecture should be flexible to offer high performance over a wide range of traffic. The LOTTERYBUS architecture was designed to address the following limitations of current communication architectures: (i) lack of control over the allocation of communication bandwidth to different system components or data flows (e.g., in static priority based shared buses), leading to starvation of lower priority components in some situations, and (ii) significant latencies resulting from variations in the time-profile of the communication requests (e.g. in time division multiplexed access (TDMA) based architectures), sometimes leading to larger latencies for high-priority communications. We present two variations of LOTTERYBUS the first is a low overhead architecture with statically configured parameters, while the second variant is a more sophisticated architecture, in which values of the architectural parameters are allowed to vary dynamically. Our experiments investigate the performance of the LOTTERY- BUS architecture across a wide range of communication traffic characteristics. The results demonstrate that the LOTTERYBUS architecture is (i) capable of providing the designer with fine grained control over the bandwidth allocated to each SoC component or data flow, and (ii) well suited to provide high priority communication traffic with low latencies (we observed upto 85.4% reduction in communication latencies over conventional on-chip communication architectures). In this project dynamic lottery bus arbiter architecture for a system on chip is discussed.
The architecture is based on a probability bus distribution algorithm and uses an adaptive ticket value method. The architecture is model in Verilog HDL and some of the simulation results are presented.

Keywords—System-on-Chip (SoC), Bus Arbiter, Verilog HDL, TDMA

1. INTRODUCTION

Fast growing automatic design technology and manufacture process of semiconductor and market demand for miniaturization of electronics systems, leads different functional systems put into one chip called System-on-Chip (SoC). Therefore SoC Technology is actively adapted to a design field. SoC is a technology that integrates homogeneous or heterogeneous system components (Microprocessor, Memory, and DSPs etc.) into a single chip. The IP reuse technology focuses on a standardization of design, and researcher found some green are in the IP reuse and HW/ SW co-design.

The communication architecture topology consists of a combination of shared buses and dedicated channels to which various SoC components are connected. These include: 1) components that can initiate a communication transaction, called masters (examples are CPUs, DSPs, DMA controllers) and 2) components that respond to transactions initiated by a master, called slaves (examples include on-chip memories and peripheral devices). When the topology consists of multiple buses, bridges are used to interconnect the necessary buses.

Since buses are often shared by several SoC masters, bus architectures come with mechanisms or protocols to manage access to the bus, which are implemented in centralized (or distributed) bus arbiters. Approaches for managing access to shared buses include round-robin, priority-based, and time-division multiplexing. Arbitration is one of the tasks performed by the on-chip communication protocol, which is also responsible for other communication functions. For example, it may limit the maximum number of bus cycles for which a master can use the bus through a maximum burst transfer size, or it may split transactions when slave devices are slow to respond to requests from a master.
In this survey a dynamic bus distribution algorithm based on lottery bus algorithm is discuss. Dynamic lottery bus distribution algorithm solves a conventional bus distribution problem and guarantees low latencies.

1.1. System-on-Chip Communication Architectures: Background

In this section, we introduce concepts and terminology associated with on-chip communication architectures and describe some popular communication architectures used in commercial SoC designs. The communication architecture topology consists of a network of shared and dedicated communication channels, to which various SoC components are connected. These include (i) masters, components that can initiate a communication transaction (e.g., CPUs, DSPs, DMA controllers etc.), and (ii) slaves, components that merely respond to transactions initiated by a master (e.g., on-chip memories). When the topology consists of multiple channels, bridges are employed to interconnect the necessary channels.
Since buses are often shared by several SoC masters, bus architectures require protocols to manage access to the bus, which are implemented in (centralized or distributed) bus arbiters. Currently used communication architecture protocols include round-robin access, priority based selection, and time-division multiplexing. In addition to arbitration, the communication protocol handles other communication functions. For example, it may limit the maximum number of bus cycles for which a master can use the bus, by setting a maximum burst transfer size. Another factor that affects the performance of a communication channel is its clock frequency, which (for a given process technology) depends on the complexity of the interface logic, the placement of the various components, and the routing of the wires.

2. ARBITRATION TECHNIQUES

There is several arbitration techniques has been developed mention as below.

2.1. Static fixed priority algorithm

Static fixed priority is a common scheduling mechanism on most common buses. In a static fixed priority scheduling policy, each master is assigned a fixed priority value. When several masters request simultaneously, the master with the highest priority will be granted. The advantage of this arbitration is its simple implement and small area cost. The static priority based architecture does not provide a means for controlling the fraction of communication bandwidth assigned to a component. If masters with high priority requests frequently, it will lead to the starvation of the ones with low priority.
Advantages: It is simple in implement & Small area cost.
Disadvantages: In Heavy communication traffic, master that has low priority value cannot get a grant signal.

2.2. TDM/Round-Robin algorithm

Time division multiplexed (TDM) scheduling divides execution time on the bus into time slots and allocates the time slots to adapters requesting use of the bus. Each time slot can span several physical transactions on the bus. A request for use of the bus might require multiple slot times to perform all required transfers. However, in this architecture, the components are provided access to the communication channel in an interleaved manager, using a two level arbitration protocol.

The first level of arbitration uses a timing wheel where each slot is statically reserved for a unique master. In a single rotation of the wheel, a master that has reserved more than one slot is potentially granted access to the channel multiple times. If the master interface associated with the current slot has an outstanding request, a single word transfer is granted, and the timing wheel is rotated by one slot. To alleviate the problem of wasted slots, a second level of arbitration is supported. The policy is to keep track of the last master interface to be granted access via the second level of arbitration, and issue a grant to the next requesting master in a round-robin fashion, at figure 1, the current slot is reserved for M1, but it has no data to communicate. The second level increments a round-robin pointer rr2 from its current position at M2 to the next outstanding request at M4.

Advantages: Easy to implement.

Disadvantages: Leads to the mistake of data transfer.

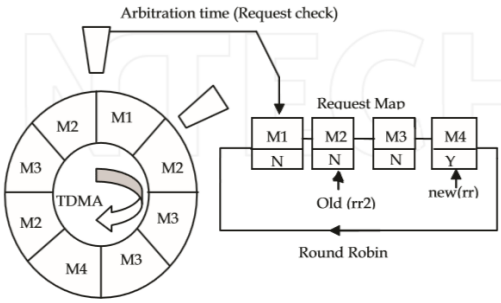

Figure 1.Round Robin based arbiter communication architecture

2.3. Static Lottery Bus Algorithm

The core of the LOTTERYBUS architecture is a probabilistic arbitration algorithm implemented in a centralized "lottery manager" for each bus in the communication architecture. The architecture does not presume any fixed communication topology. Hence, various SoC components may be interconnected by an arbitrary network of shared channels or a flat system wide bus as shown in figure 2.

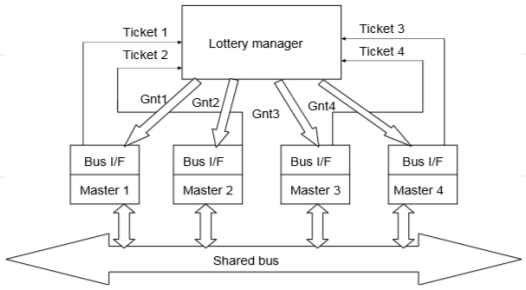

Figure 2.Lottery manager for a bus in a Lottery bus based communication architecture

The lottery manager accumulates requests for ownership of the bus from one or more masters, each of which is (statically) assigned a number of "lottery tickets," as shown in figure 2. The manager pseudo-randomly chooses one of the contending masters to be the winner of the lottery, favoring masters that have a larger number of tickets, and grants access to the chosen master for a certain number of bus cycles. Multiple word requests may be allowed to complete without incurring the overhead of a lottery drawing for each bus word. However, to prevent a master from monopolizing the bus, a maximum transfer size is used to limit the number of bus cycles for which the granted master can utilize the bus Also, the architecture pipelines lottery manager operations with actual data transfers, to minimize idle bus cycles. The inputs to the lottery manager are a set of requests (one per master) and

the number of tickets held by each master. The output is a set of grant lines (again one per master) that indicate the number of words that the currently chosen master is allowed to transfer across the bus. The arbitration decision is based on a lottery. The lottery manager periodically (typically, once every bus cycle) polls the incoming request lines to see if there are any pending requests. If there is only one request, a trivial lottery results in granting the bus to the requesting master. If there are two or more pending requests, then the master to be granted access is chosen using the approach described next.

2.4. Lottery-based arbitration algorithm

Let the set of bus masters be C1, C2, C3, C4 & Let the number of tickets held by each master are t1, t2, t3, t4. At any bus cycle, let the set of pending bus access requests be represented by a set of Boolean variables ri (i=1, 2…, n) where ri=1 if component Ci has a pending request, and ri=0 otherwise. The master to be granted is chosen in a pseudo-random way, favoring components with larger numbers of tickets. The probability of granting component Ci is given by

$$P(C_i) = \frac{r_i * t_i}{\sum_{j-1}^{n} r_j * t_j}$$

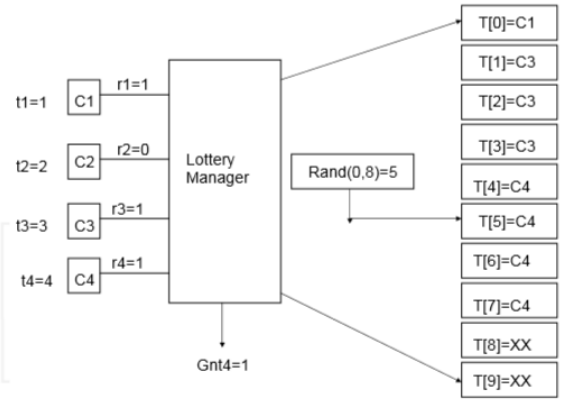

Figure 3. Lottery that determines which master should be awarded ownership of the bus.

Figure 3 shows an example where three out of four bus masters have contending requests, with tickets in the ratio 1:3:4. For this request map and ticket holding combination, the lottery-based approach should result in P(C1)=0.12, P(C2)=0, P(C3)=0.37, P(C4)=0.5, To make an arbitration decision, the lottery manager examines the number of "active" tickets, or the number of tickets in possession of the set of components that have pending requests. This is given by

$$\sum_{j=1}^{n} r_j * t_j$$

Therefore, a random number is generated uniformly in the range (0, 8) In the example, the generated random number is 5, and lies between r0*t0+r1*t1+r2*t2=4 and r0*t0+r1*t1+r2*t2+r3*t3=8. Therefore, it indexes to a ticket owned by component C4.

2.5. Structure of Static Lottery Based arbiter

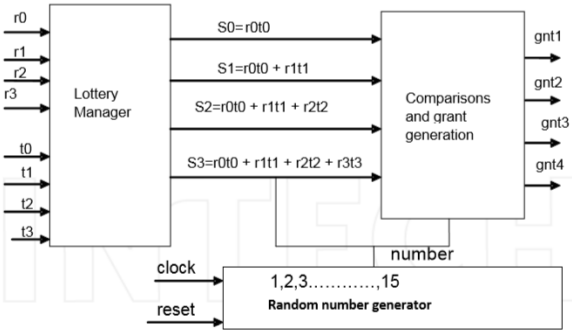

Figure 4.Structure of Static Lottery Based arbiter

As illustrated in Figure 4. Therefore, the bus is granted to component C4 win the very first lottery. Figure 4 shows block diagram of Lottery Bus architecture. It contains three basic blocks. (1)Lottery manager:-In this block r1, r2, r3, r4 are the requests signal of the master and t1, t2, t3 and t4 are the tickets of the master respectively. That will generate the ticket values that are r1t1, r1t1+r2t2, r1t1+r2t2+r3t3, r1t1+r2t2+r3t3+r4t4. (2)Random number generator:- Random number generator is working on the principle of pseudo random binary sequence generator .That will generate the number randomly. (3) Comparison and grant generation hardware:- The random number is compared in parallel against all four partial sums.

Each comparator outputs a "1" if the random number is less than the partial sum at the other input. Since for the same number, multiple comparators may output a "1" (e.g., if r1=1 and the generated random number is smaller than, all the comparators will emit "1"), it is necessary to choose the first one, starting with the first comparator. For example, for the request map 1011 if the generated random number is 5, only's C4 associated comparator will output a "1." However, if the generated random number is "1," then all the comparators will output a "1," but the winner is C1.

The architecture is model using VerilogHDL for three masters. Ticket values are keeping fixed. Figure 2.3.2 shows the simulation results for the discussed architecture. Here t0, t1, t2 & t3 are tickets values and gnt0, gnt1, gnt2 & gnt3 are grant signals of the master processor. Signal n1 is random number generated signal and signal h0, h1, h2 &h3 are calculated value for the master or processor according to it's ticket value and request signal r. As shown in figure 4 the signal r(0), r(1) ,r(2) and r(3) are the request of master 0,master 1,master 2 and master 3 respectively the signal t0, t1 , t2 and t3 are the ticket values of master 0,master 1,master 2 and master 3 respectively. The signal s0, s1, s2 and s3 are the total ticket values of master 0, master 1, master 2 and master 3 respectively. The signal n1 represents the number generated by pseudo random number binary sequence generator. The signal gnt0, gnt1, gnt2 and gnt3 are the grant signal of master 0, master 1, master 2 and master 3 respectively.

2.4. Dynamic Lottery Bus Algorithm

In this architecture, the inputs to the lottery manager consist of a set of request lines (r0r1r2r3), and the number of tickets currently possessed by each corresponding master that are generated by ticket generated by ticket generator. Therefore, under this architecture, not only Range of current tickets varies dynamically but it can take on any arbitrary value (unlike the static case, where it was fixed). Therefore at each lottery, the lottery manager needs to calculate for each component Ci.

This is implemented using a bit wise AND operation and tree of adder, as shown in figure 5. The final result, T=r0t0+r1t1+r2t2+r3t3, defines the range in which the random number must lie. A limitation of this

implementation is that distribution of the resulting random number is not uniform. The rest of the architecture consists of comparison and grant hardware, and follows directly from the design of the static lottery manager.

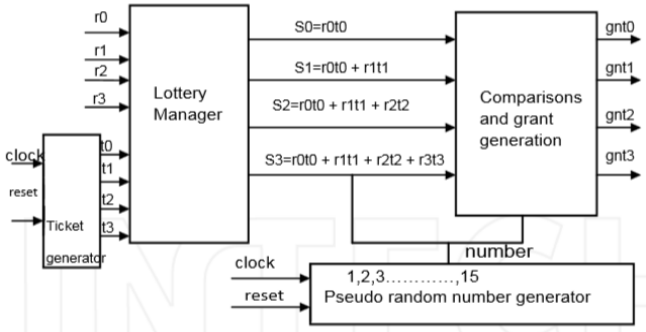

Figure 5.Structure of Dynamic Lottery bus

The architecture is modeled using Verilog HDL. Ticket values are keeping varying. Figure 4 shows the waveforms for the discussed architecture. Here t0, t1, t2 & t3 are tickets values and gnt0, gnt1, gnt2 and gnt3 are grant signals of the master processor. Signal n1 is random number generated signal and signal s0, s1, s2 and s3 are calculated value for the master or processor according to its ticket value and request signal r.

Advantages: All the masters that are requesting gain the control of bus.

Disadvantages: If the pseudo random number is greater than total ticket value then none of the masters will get the grant signal.

3. SIMULATION RESULT OF ARBITER

The architecture is modeled using Verilog HDL. Ticket values are keeping varying. Figure 5 shows the waveforms for the discussed architecture. Here t1, t2, t3 & t4 are tickets values and gnt1, gnt2, gnt3 and gnt4 are grant signals of the master processor. Signal n is random number generated signal and signal s1, s2, s3 and s4 are calculated value for the master or processor according to its ticket value and request signal requests r1, r2, r3 & r4.

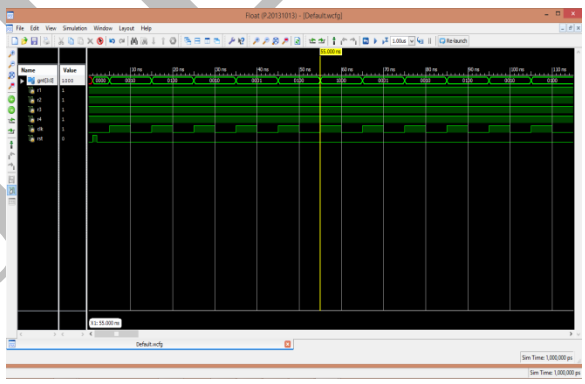

Figure 6.Simulation waveforms for Lottery Arbiter for Varying Request With dynamic tickets.

4. SYNTHESIS RESULT OF ARBITER

.

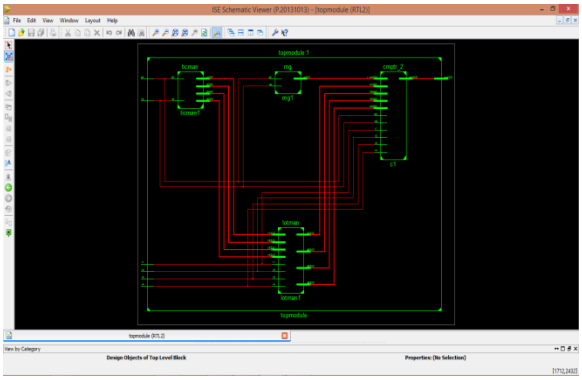

Figure 7.Synthesis result for Lottery Arbiter.

5. CONCLUSION

For the dynamic lottery arbitration tickets are generated dynamically as the requests are dynamic [Figure 5]. The proposed algorithm is flexible, and all masters requesting for the bus will be granted access. It is observed that performance of SoC is slightly improved with respect to latency in the dynamic ticket based lottery algorithm compared to static algorithm. If grant signal g3 for M3 is observed delay is found to be around 55 ns. In the proposed algorithm the waiting time is reduced for the master4 (gnt4).

The architecture is simulated using Model Sim and latency is observed for various components. Latency for master4 is observed to be reduced around 55 ns with clock period of 10 ns in the proposed probability based lottery algorithm. In this algorithm only, master requesting for the bus will be granted and at a particular time only one master will be granted the bus. The arbiter proposed in this paper does not exhibit this phenomenon, resulting in low latencies for high priority components.

Static priority and other conventional architectures had the drawbacks like bus starvation and bus contention when more than one master request for the bus. Hence this dynamic round robin arbiter based on lottery method is proposed. It is observed, latency is improved and attends to all masters requesting for the bus. At a time only one master will be granted the bus. This arbiter provides flexible design for efficient SoC. This algorithm can be implemented in the design of Soc which has its application in the field of communication and can demonstrate its superiority over other conventional architectures.

6. REFERENCES

1. Chang HeePyoun, et. al. (2003).“The Efficient Bus Arbitration Scheme InSoC Environment”, IEEE International Workshop on System-on-chip.
2. K. Lahiri, A. Raghunathan (2006). “The Lotterybus on-chip communication architecture”, IEEE Trans. On VLSI system
3. “Arbitration Schemes for Multiprocessor Shared Bus” by Dr. Preeti Bajaj and Dinesh Padole G.H. Raisoni College of Engineering, Nagpur India
4. “Performance Analysis of On-Chip Communication Architecture in MPSoC” by Professor/HOD, IT Sri Muthukumaran Institute of Technology Mangadu, Chennai-69 & Dr. R.Amutha, P.hd, Professor,ECE SSN College of Engineering, Kalavakkam.
5. “Dynamic Lottery Bus Arbiter for Shared Bus System on Chip: A Design Approach with VHDL” by Neeta Doifode, Dinesh Padole, Dr. Preeti Bajaj, Professor & Head, ECE Dept, G.H. Raisoni College of Engineering, Nagpur, India.
6. KanchanWarathe, Dinesh Padole and Dr.Preeti Bajaj , “A Design Approach to AMBA Bus Architecture With Dynamic Lottery Arbiter”, 2009 IEEE.