

A Real-time Guaranteed and High-speed Asynchronous Fixed Priority Arbitration Algorithm

Ruizhen Wu^{*1}, Yintang Yang², Li Zhang³

^{1,2,3}Institute of Microelectronics, Xidian University, 2 South Taibai Road, Xi'an, Shaanxi, PR China

^{*1}wuruizhen_1985@163.com; ²yangyt@xidian.edu.cn; ³xidianzhangli1988@gmail.com

Abstract-In shared System-on-Chip (SoC) bus systems, masters may have requirements of priority and real-time guarantee when there exists bus contentions. Based on Fixed Priority Arbitration Algorithm, a real-time guaranteed and high-speed asynchronous fixed priority arbitration algorithm is proposed and implemented from an asynchronous circuit perspective. In NINP (NonIdling and NonPreemptive) model, the proposed algorithm has advantages in bandwidth utilization, power compared and can decrease the delay at least 8% with the commonly-used FP, RR and Lottery algorithms through verifications on Xilinx Virtex5 XC5VLX70T of 65nm CMOS technology and -2 speed grade. The proposed algorithm is suitable in various environments of SoC applications.

Keywords- Arbitration Algorithm; Asynchronous Circuits; Soc; Real-Time; Priority

I. INTRODUCTION

Advances in semiconductor manufacturing technology make feasible integration of an increasing number of Intellectual Properties (IPs) and shared resources onto a single System-On-Chip (SoC). The communication architecture plays a key role in SoC because it facilitates the integration of many IPs. Such communication architecture is commonly built by shared buses which serve as the communication channels between IPs [1].

There are two major components: masters and slaves, in SoC bus architecture. Those IPs who initiate a communication transaction are called masters, such as microprocessors, microcontrollers and so on. Unlike what masters do, those IPs who respond to transactions initiated by a master are called slaves, such as memories, peripheral devices and so on. In many applications, masters may have real-time requirements on requests. Such examples include multimedia players, cellular phones, medical devices, radar, and flight/missile control systems. A master with a real-time requirement demands its transactions accomplished within a fixed number of clock cycles. It will function incorrectly when it is not ensured a real-time guarantee. When several masters issue requests simultaneously, only one master can access the bus, so arbitrations are conducted to deal with this problem [2]. Therefore, the algorithms should be designed carefully in high performance systems.

The advantages of arbitration algorithms depend on the following factors: 1. fairness: Each master has opportunities to access bus and there should not be any starvations and monopolization of bus. 2. Bus utilization: When there exists any requests, bus is not in idle state. 3. Good reusability: The algorithms are applicable to various communication traffics. 4. Power. 5. Speed [3, 4].

Arbitration algorithms commonly used for buses include FP (Fixed Priority), RR (Round-Robin) and Lottery algorithms. In this paper, we propose a real-time guaranteed and high-speed asynchronous fixed priority arbitration algorithm, which meet both priority and real-time demand.

We compare the proposed algorithm with commonly-used algorithm, FP, RR and Lottery algorithms on Virtex5 from Xilinx with 65nm CMOS technology. The experiment results show that the proposed algorithm can handle real-time and priority requirements of each master and has advantages of higher speed, lower power, better bus utilization and smaller area over other algorithms.

The remainder of this paper is organized as follows: The classic algorithm is introduced in Section II. Section III presents the proposed algorithm and working manner of its arbiters. Section IV includes simulations and analysis of performance. Section V concludes this paper.

II. RELATED WORK

In this section, three commonly-used algorithms will be introduced: FP arbiters, RR arbiters, and Lottery arbiters.

A. Fixed Priority Algorithm

It is a common scheduling mechanism. In fixed priority algorithm, each master is assigned a fixed value as its priority. When several masters request simultaneously, the master with the highest priority will be granted. There are many ways to illustrate the logic pattern, as [5] shows, it assumes the request signal of master i is Req_i and master i has the lower priority than $i-1$ and so on. The grant signal of master i is $Grant_i$ and it can be described by the following equation:

$$Grant_i = \overline{Req_0} \times \overline{Req_1} \cdots \overline{Req_{i-1}} \times Req_i \quad (1)$$

The advantages of this arbitration are its simple implement and low hardware cost. However, the masters with lower priority could be severely starved by higher priority ones, and Fixed Priority arbitration is limited in providing a real-time guarantee.

B. Round-Robin Algorithm

Round-robin algorithm coordinates the use of shared resources among multiple requesters in a fair fashion. It is fair to its request by repeatedly rotating the highest priority pointer to the position immediately next to the second requester picked. As [6] considers a system consisting of a single bus with four masters that contend for access, 0101 is the request vector in the first clock cycle, where the current highest priority pointer points to Master A. Priorities go down as we go right from the current highest priority pointer and wrap around. At the end of the first cycle, the master B granted the bus so the highest priority pointer updated to Master C. Fig. 1 is the working manner of RR arbitration.

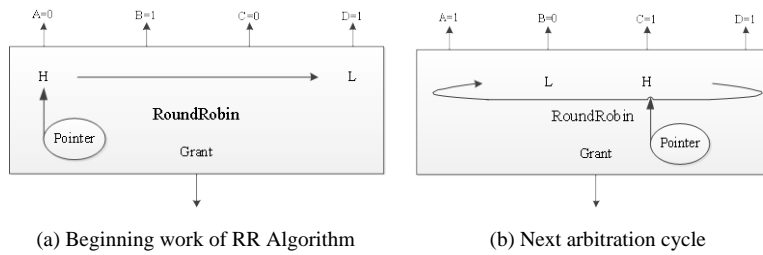


Fig. 1 Working manner of Round-Robin arbitration

RR algorithm guarantees fairness among masters compared with FP algorithm and there will not be any starvations and monopolization of bus. But it is difficult for them to meet priority requirements and cannot always meet real-time demand.

C. Lottery Algorithm

The algorithm stochastically grants one of the contending masters according to the ticket assignment statically. The algorithm generates a pseudo random number by LFSR (Linear Feedback Shift Register). The master with the ticket number matched to this number is granted to use the bus. The master to be granted is chosen with probability as follows, in which the r_i and t_i in the equation means the request and tickets hold by each master respectively [7].

$$P_i = \frac{r_i t_i}{\sum_{j=1}^n r_j \times t_j} \quad (2)$$

The lottery manager is able to meet complex priority management and ordinary real-time demand. Nevertheless, if the bus access behaviors are very diverse among masters, the actual bandwidth ratio would not conform to the priority ratio. To meet the real-time requirement, the tickets of the master with the minimum request ratio should be much larger than the others. However, it is really hard to formulate a proper tickets assignment to each master if there are multiple masters with diverse real-time and priority requirements. Furthermore, more area is required for the efficient random number generation. This assumption may be true for network switch design [7], however, it is unlikely to be true for many other SoC applications.

III. PROPOSED ALGORITHM

We assume a single On-Chip Bus (OCB), on which an arbitration algorithm arbitrates all requests from a number of bus masters in a NonIdling and NonPreemptive (NINP) [8] fashion. By NINP, we mean the algorithm must grant at least one request when any exists, and an already granted request cannot be interrupted before it has been completely served. This section describes the proposed algorithm.

A. Real-Time Algorithm

Bus masters are divided into two disjoint sets: RT masters and NRT masters. RT masters issue real-time constraints requests. And NRT masters issue non real-time constrained requests.

To meet the real-time constraints, we set a real-time counter for each RT master according to their real-time requirements. Each RT master's real-time counter (rt_i) has an initial value and when the request signal is valid, the value is decremented by 1 every time cycle until its master is granted. Besides initial value, there is a dead-line (dl_i) value for each RT master's real-time counter. When the value of real-time counter is reduced to less than dead-line value, its master will be granted the bus to ensure the real-time constraints. And the upper bound on the maximum number of cycles for a burst is set to 78 according to

[8]. The real-time constraints (Rc_i) are made by:

$$Rc_i = rt_i - dl_i \quad (3)$$

Fig. 2 shows an example of real-time algorithm's operation. We assume that M_a has $rt_a=60$; M_b has $rt_b=50$. M_a has $dl_a=55$; M_b has $dl_b=44$.

1) Cycle 40(the Left Table in Fig. 2)

As M_a issues a request at this cycle, the real-time counter of M_a is set to its rt_a , 60. At this time, M_b already enters its real-time constraints ($rt_b-dl_b \leq Rc_b$) and thus it is granted first.

2) Cycle 48(the Right Table in Fig. 2)

M_b's request is a 8-beat burst, and therefore the request is finished at cycle 48. At this time, M_a already enters its real-time constraints ($rt_a-dl_a \leq Rc_a$) and thus it is granted first, and M_b's real-time counter is set back to its real-time value $rt_b, 50$.

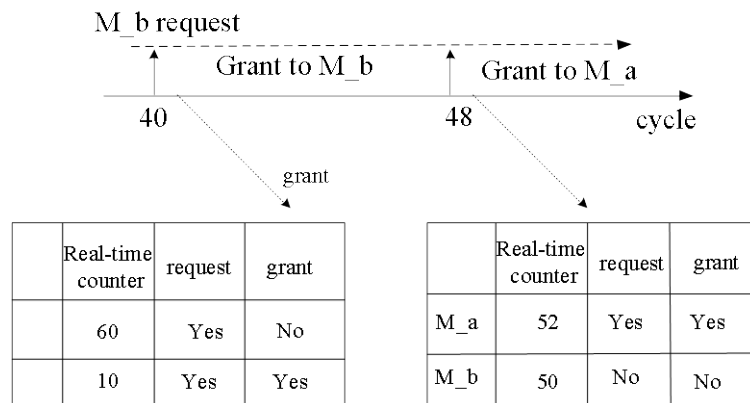


Fig. 2 An example of real-time algorithm's operation

To meet all real-time requirements in any circumstances, dl of each master must be carefully set. Our target is to provide, simultaneously, an RT guarantee and proper priority allocation.

B. The Proposed Arbitration Algorithm

The real-time algorithm is set to satisfy the demands of RT masters, but the priority demands of NRT masters have to be met with another algorithm. FP algorithm can easily satisfy the priority demands and it has an advantage of quick response, so we use the FP as the second level's arbitration algorithm. Fig. 3 shows the flow of proposed arbitration algorithm.

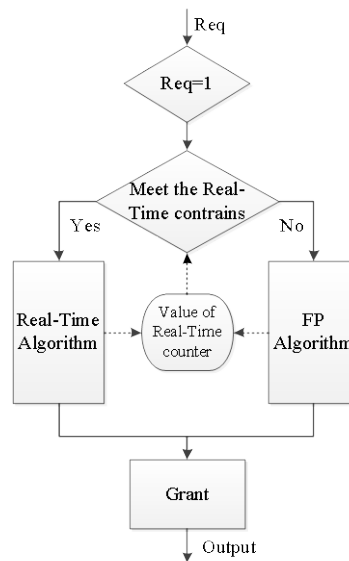


Fig. 3 Flow of proposed arbitration algorithm

For some embedded systems, the real-time guarantee must be satisfied, priority must be supported under the condition that the real-time guarantee is strictly ensured. So there is a real-time constraints judgment module to decide whether to use the real-

time algorithm or FP algorithm at first. When the real-time counter value is no more than the dead-line value, it means that the real-time guarantee must be satisfied at first. Therefore the grant signal is decided by real-time algorithm in this case. Then the real-time value of the master which got the bus will be reset. But if all the real-time counter value is bigger than dead-line value, it means the real-time constraints are not urgent now. Then the arbitration works in FP arbitration way and the grant signal is decided by different priorities. Different from former working manner, the real-time value will decrease 1 to update the value in this case.

The operation mode can switch freely in different case to satisfy different need with the proposed arbitration algorithm.

FP algorithm has the advantages of simple implement, quick response and low hardware cost. But real-time constraints have to be met by adding several registers as real-time counters. It will cause extra area overhead, power loss and slow down the working speed. Asynchronous circuits are triggered by valid requests rather than clock and have advantages of high speed and low power consumption. The four-phase dual-rail protocol which is one of asynchronous multiple protocols is utilized in the proposed algorithm. Compared with common synchronous algorithms, it reduces static power consumptions of similar priority algorithms effectively and electromagnetic compatibility of the circuits is improved. The four-phase handshake protocol is illustrated in Fig. 4 [9].

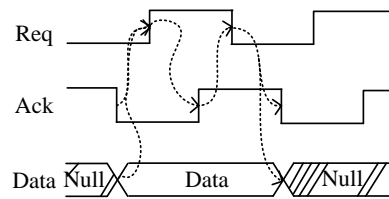


Fig. 4 Four-phase handshake protocol

The data transmission of four-phase handshake protocol consists of four steps: request, acknowledge, remove request, and remove acknowledge. Four-phase handshake protocol is also called RTZ (return-to-zero) handshake protocol. By RTZ, we mean request and acknowledge signals are set to zero and transmission is in Null state after each transmission is completed. Thus, static power can be reduced effectively because of Null state.

The dual-rail encoding added Null state which is different from the binary encoding and it is showed in Table 1.

TABLE 1 DUAL-RAIL ENCODING

Data	Wire0	Wire1
0	1	0
1	0	1
Null	0	0
Invalid	1	1

As table 1 indicates, each bit data is represented by two lines. “01” represents logic 0, while “10” represents logic 1. “00” state is a null state, and “11” state is invalid. In dual-rail operating mode, data transmission switches between valid state and Null state. Furthermore, only one line flips whenever state changes, so the circuits have high robustness.

The four-phase dual-rail protocol is utilized in the proposed algorithm but as we introduced above, there will be more than twice of the loss if each signal is expressed with four-phase dual-rail protocol. To take full advantage of, the proposed algorithm employs four-phase dual-rail protocol only for trigger signals. Fig. 5 shows the circuit's structure of the proposed algorithm.

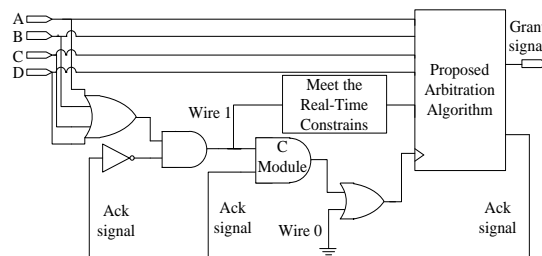


Fig. 5 The circuit's structure of proposed algorithm

Since the entire request signals A, B, C, D are synchronized, they are connected with a quad-input OR gate and the output signal is connected with AND gate with the inverted Ack signal. Here the Ack signal is used to indicate that the proposed algorithm has finished its work. When Ack signal equals to 1, it means the arbitration is working now and 0 means that it has already finished its work and ready for the next round of arbitration. This design can make sure when there is at least one valid request and the algorithm has finished its work. The wire 1 can be a high level signal. The arbitration is triggered by wire 1 to check the real-time constraints before arbitration.

Wire 1 signal and Ack signal is connected with a C module (a special state unit always used in asynchronous circuits) to avoid signals “race and hazard” problem. C module is a state-holding element which is similar with the set-reset latch. The output of the C module reflects the inputs when the states of all inputs match. The output then remains in this state until the inputs all transition to the other state [10]. Th2 [11] gate is used here instead of C module which is implemented by FPGA based on [8] and [12]. Since the algorithm needn’t to work when the request signal is 0, Thus Wire 0 is grounded here. In this working mechanism, the real-time counter and the arbitration algorithm can work only when they needed, which will reduce power consumption and improve the working speed.

IV. EXPERIMENTAL RESULTS

To compare the performances of different algorithms, Xilinx Virtex5 XC5VLX70T of 65nm CMOS technology and -2 speed grade are utilized as the verification platform. Comparisons are made between the proposed algorithm and commonly-used FP, RR and Lottery algorithms.

A. Experiment Setup

In this test, the simulated architecture consists of 4 masters, of which master C, D have real-time requirements, while A and B do not. In the proposed arbitration algorithm, the priorities of four masters are set according to the request ratio. If the request proportions of master A, B, C and D satisfy the following equation: $A > B > C > D$, then the priorities of four masters satisfy: $A > B > C > D$, and vice versa. To analyze performance of the proposed arbitration algorithm under various conditions, we set up parameter of T_f to illustrate three categories of traffic pattern: light, heavy and aggressive [13]. T_f is the ratio of total work time and total idle time of n masters per unit time and is shown as follows.

$$T_f = T_{delay} / T_{work} \quad (4)$$

Where T_{work} is the total work time and T_{delay} is the total idle time. In bus-based arbitration, only one master among n masters could get the ownership of bus. In this way, the more time n masters request for occupying bus, the bigger the probability is for n masters requesting to get bus simultaneously, which means the bigger bus contentions are. Thus this setup could verify the capacity of bandwidth allocation. A smaller value of T_f means the proportion of contention of four masters is larger, and vice versa. In this experiment, we use $T_f=1$ to test the aggressive traffic, $T_f=1.5$ to test the heavy traffic and $T_f=2$ to test the light traffic.

Furthermore, we set the request ratio of A, B, C, D is 4:3:2:1, 1:2:3:4 and 1:1:1:1 respectively. Our target is to obtain the bandwidth utilization ratio, real-time violations, max latency of four masters and the bus utilization under various conditions. We ran the execution for 10,000 clock cycles.

B. Performance Analysis

We compare the proposed algorithm with other three algorithms, FP, RR and Lottery under various conditions mentioned above. The experimental results are shown in Table 2 and Table 3.

TABLE 2 REAL-TIME VIOLATIONS OF FOUR ALGORITHMS

Algorithm	Real-time Violations	Max latency(cycle)
FP	60	10000
RR	55	245
Lottery	41	654
Proposed Algorithm	0	84

TABLE 3 BANDWIDTH UTILIZATION RATIO OF FOUR DIFFERENT ALGORITHMS

	$T_f=1$					$T_f=1.5$					$T_f=2$					
	A	B	C	D	IDLE	A	B	C	D	IDLE	A	B	C	D	IDLE	
1:1:1:1	0.40	0.39	0.00	0.00	0.21	0.40	0.39	0.00	0.00	0.21	0.40	0.39	0.00	0.00	0.21	FP
1:2:3:4	0.20	0.36	0.18	0.00	0.27	0.18	0.18	0.18	0.16	0.30	0.15	0.21	0.21	0.14	0.29	
4:3:2:1	0.38	0.36	0.00	0.00	0.27	0.38	0.36	0.00	0.00	0.27	0.38	0.36	0.00	0.00	0.27	
1:1:1:1	0.20	0.20	0.20	0.19	0.21	0.20	0.20	0.20	0.19	0.21	0.20	0.20	0.20	0.19	0.21	RR
1:2:3:4	0.20	0.18	0.18	0.18	0.26	0.20	0.18	0.18	0.18	0.26	0.12	0.21	0.21	0.20	0.26	
4:3:2:1	0.20	0.18	0.18	0.18	0.26	0.20	0.18	0.18	0.18	0.26	0.21	0.21	0.21	0.11	0.26	
1:1:1:1	0.14	0.16	0.18	0.16	0.36	0.14	0.16	0.24	0.12	0.34	0.14	0.16	0.22	0.12	0.36	Lottery
1:2:3:4	0.15	0.18	0.21	0.15	0.31	0.10	0.14	0.22	0.18	0.36	0.10	0.16	0.20	0.18	0.36	
4:3:2:1	0.24	0.18	0.21	0.09	0.28	0.18	0.16	0.22	0.08	0.36	0.18	0.16	0.20	0.06	0.40	
1:1:1:1	0.30	0.27	0.25	0.18	0	0.2	0.22	0.21	0.21	0.16	0.2	0.2	0.2	0.20	0.20	Proposed Algorithm
1:2:3:4	0.17	0.17	0.27	0.30	0.09	0.17	0.18	0.20	0.26	0.19	0.15	0.15	0.18	0.23	0.29	
4:3:2:1	0.45	0.23	0.21	0.11	0	0.40	0.15	0.14	0.13	0.18	0.37	0.13	0.13	0.11	0.26	
	A	B	C	D	IDLE	A	B	C	D	IDLE	A	B	C	D	IDLE	

From Table 2, we can see that FP, RR and Lottery algorithm fail to meet real-time requirements since they do not take real-time requirements into consideration. Max latency is the number of clock cycles that a master has to wait to get the bus ownership. Note that FP is the worst because its max latency is much longer than other arbitration algorithms. Although Lottery is much better in handling real-time requirements than Round-robin, its max latency is much longer. The proposed algorithm can meet real-time requirements and its max latency is the smallest because of the asynchronous working manner.

It is clear from Table 3 that the FP algorithm is sensitive to communication traffic. When T_f is larger than 1, there will exist starvations and monopolization of bus. The request ratio has a great influence on the actual bandwidth allocation ratio. Furthermore, it has worse performance when T_f is larger and complex bandwidth allocation ratio cannot be achieved. The RR algorithm has a good performance when the request ratio is 1:1:1:1 or T_f is larger. When T_f is smaller, the output bandwidth allocation ratio is influenced by the request ratio. Furthermore, the RR algorithm is limited in achieving various priorities of masters. Lottery is sensitive to request ratio. We can see that when $T_f=2$, the bandwidth of each master is approximate to it. However, when $T_f=1$, the bandwidth requirement of each master is not met. Because of the real-time constraints and fixed priority, the proposed algorithm is less sensitive to the traffic pattern variations while providing a much better bus utilization. When request ratio differs significantly, the bandwidth utilization of four masters approximates to their priorities. So the proposed algorithm outperforms other three commonly-used algorithms.

IDLE in Table 3 means the total waiting time, and its value decides the entire bus bandwidth utilization. For example, if the value of IDLE is 0.26, the entire bus bandwidth utilization is 74%. The higher the value of IDLE, the lower utilization of the bus bandwidth. From Table 3, it is obvious that the IDLE value of proposed algorithm is the lowest because of its asynchronous working manner, in other words, its entire bus bandwidth utilization is the highest.

C. Power Comparisons

To compare resources utilization of four different algorithms, we choose the situation when request ratio is 1:1:1:1, $T_f=1$ and $V_{cc}=1$. The results are shown in Table 4.

TABLE 4 POWER OF FOUR ALGORITHMS

	FP	RR	Lottery	Proposed Algorithm
Dynamic	0.023	0.027	0.028	0.026
Quiescent	1.435	1.435	1.435	1.431
Total	1.457	1.462	1.462	1.456

From Table 4, we can see that FP has the smallest dynamic power consumption. The proposed algorithm deploys dual-rail protocol to meet real-time constraints based on FP, so there is a small extra consumption because of the real-time registers. But the quiescent power is reduced because of the asynchronous working manner. Thus, the proposed algorithm has a better power performance than the other three.

D. Speed

To compare the speed of four algorithms, post simulations are made of two-input, four-input and six-input algorithms after auto place and route on Xilinx Virtex5. The clock cycle is 10ns. The speed of four algorithms is compared in Fig. 6.

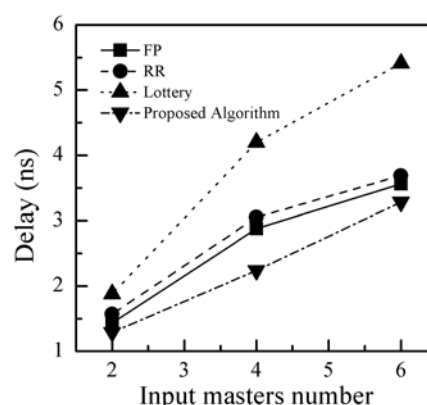


Fig. 6 The speed of four algorithms

The speed of proposed algorithm is faster than other three algorithms. The delay is decreased by 8%-10% compared to FP algorithm and 31.3%-39.3% to Lottery algorithm. But the delay increases with the increases in the number of inputs because the number of data line will increase manifold in dual-rail encoding with the increase in request signals.

V. CONCLUSIONS

Arbitration algorithms play an important role in SoC. In this paper, an arbitration algorithm with real-time constraints is designed from an asynchronous perspective based on FP algorithms with quick response. The proposed algorithm could meet the requirements of priorities and real-time guarantee and has advantages of high speed and low power compared with the commonly-used FP, RR and Lottery algorithms. The proposed algorithm is suitable in various environments of SoC applications.

ACKNOWLEDGMENTS

This Project is supported by the National Natural Science Foundation of the China (No. 60725415), the National High-tech Program (No. 2012AA012302).

REFERENCES

- [1] D. Shanthi and R. Amutha, "Design of efficient on-chip communication architecture in MpSoC," 2011 International Conference on 2011 in Recent Trends in Information Technology (ICRTIT), Piscataway, pp. 364-369, 2011.
- [2] L. Zhang, R. Wu, Y. Yang, and F. Lu, "An adaptive dynamic and real-time guaranteed arbitration algorithm for SoC bus communication," *Journal of Computational Information Systems*, vol. 9, pp. 5693-5700, Jul. 2013.
- [3] L. Wang and X. Shen, "The implementation of packet circular priority algorithm on PCI," *Microelectronics & Computer*, vol. 1, pp. 1-4, Jan. 2002.
- [4] A. Kulmala, E. Salminen, and T. D. Hamalainen, "Distributed bus arbitration algorithm comparison on FPGA-based MPEG-4 multiprocessor system on chip," *Computers & Digital Techniques*, vol. 2, pp. 314-325, Jul. 2008.
- [5] S. Abdel-Hafeez and S. Harb, "A VLSI high-performance priority encoder using standard CMOS library," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 53, pp. 597-601, Jan. 2006.
- [6] H. F. Ugurdag, F. Temizkan, O. Baskirt, and B. Yuce, "Fast two-pick n2n round-robin arbiter circuit," *Electronics Letters*, vol. 48, pp. 759-760, Jun. 2012.
- [7] K. Lahiri, A. Raghunathan, and G. Lakshminarayana, "The LOTTERYBUS on-chip communication architecture," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 14, pp. 596-608, Jun. 2006.
- [8] H. K. Peng and Y. L. Lin, "An optimal warning-zone-length assignment algorithm for real-time and multiple-QoS on-chip bus arbitration," *Transactions on Embedded Computing Systems*, vol. 9, Mar. 2010.
- [9] W. Song and D. Edwards, "Survey of asynchronous networks-on-chip," *Journal of Computer-Aided Design and Computer Graphics*, vol. 24, pp. 699-709, Jun. 2012.
- [10] J. Sparso and S. Furber, *Principles of Asynchronous Circuits Design: A Systems Perspective*, Boston: Kluwer Academic Publishers, 2001.
- [11] S. Abdel-Hafeez and S. Harb, "A VLSI high-performance priority encoder using standard CMOS library," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 53, pp. 597-601, Aug. 2006.
- [12] Z. Xu, "The algorithm and application for the symmetrical routing," *2011 International Conference on Transportation, Mechanical, and Electrical Engineering*, Changchun, China, pp. 1508-1511, 2011.
- [13] H. Shah, A. Raabe, and A. Knoll, "Priority division: A high-speed shared-memory bus arbitration with bounded latency," *Automation and Test in Proceedings-Design*, Grenoble, France, pp. 1497-1500, 2011.

Ruizhen Wu received his B.S. degree from Institute of Microelectronics from Xidian University in 2009, and he is currently working toward the MD-PhD degree with the school of Microelectronics, Xidian University, China. His research interests include asynchronous circuits design, network on chips and VLSI design.

Yintang Yang received the Ph.D. degree from the School of Technical Physics, Xidian University, majoring in the specialty of semiconductor. He is now a professor in Microelectronics Institute, Xidian University. He won the title of National Model Teacher and the Chinese Youth Science & Technology Award. He was selected into the "Trans-century Outstanding Talents Program" of the Ministry of Education, the National "Key Talents Program" & "New Century Key Talents Program", and the Shaanxi Provincial "Trans-century Talents" Program. He has more than 100 publications in refereed journals and conferences.

Li Zhang received his B.S. degree from the School of Electronics and Information Engineering of Hebei University in 2011, and she is currently working toward the Master degree with the school of Microelectronics, Xidian University, China. Her research interests include asynchronous circuits design, network on chips and VLSI design.