International Journal of Computer Applications (0975 – 8887)
Volume 81 – No 14, November 2013

Fair Chance Round Robin Arbiter

Prateek Karanpuria
B.Tech student, ECE branch
Sir Padampat Singhania University
Udaipur (Raj.), India

ABSTRACT

With the advancement of Network-on-chip (NoC), fast and fair arbiter as the basic building block for high speed switches/routers gained attention in recent years. In this paper I propose the fair chance round robin arbiter (FCRRA), a high speed, low power and area efficient RRA for NoC applications. The FCRRAG tool propose in this paper can generate a design for bus arbiter, which can handle the exact number of bus masters for both on chip and off chip buses within one short cycle.

General Terms

i. Arbiters are electronic devices that allocate access to shared resources.

ii. Virtual Output Queues (VOQs) [4]: there are VOQs in a switch to remove possible output port contention (Head of Line (HOL) blocking).

iii. Bus Arbiter resolves bus conflicts when multiple bus masters request a bus in the same cycle.[4]

Keywords

Round Robin arbiter, Fair chance round robin arbiter, Network on Chip, iSLIP, arbiter.

1. INTRODUCTION

The growing demand for optimized channel width utilization fosters the need for high speed packet switches and routers. There are various major aspects in the design and implementation of a high speed packet switch and routers:

1) Cost effective scheme for multiple input packets demands for sharing same channel for output.

2) A switch scheduling algorithm that chooses which packets to be sent from input ports to output ports, and

3) A fast mechanism that generates control signals for switching elements of the switching fabric. [1]

4) Fixed length switching technology: Variable-length IP packets are segmented into fixed-length “cells” at inputs and are reassembled at the outputs. [2]

Earlier packet switches were based on Input Queuing (IQ) are desirable for high speed switching, since the internal operation speed is only slightly higher than the input line. However, an Input Queuing switch has a critical drawback, the throughput is limited to 58.6% due to the head-of-line (HOL) blocking phenomena. Output Queuing (OQ) switches have the optimal delay throughput performance for all traffic distributions, but the N times speed-up in the fabric limits the scalability of this architecture. [2]

Today networks require switching fabrics that can deliver hundreds of gigabits or terabits per second of throughput. Nowadays, modern switches had replaced the idea of a single FIFO queue per input port with multiple virtual output queues (VOQs) per input port. One VOQ exists per output port for which input traffic is destined. Therefore, an input port can have several packets queued in several different VOQs, assuming several different output ports. Each of these VOQs are eligible to be serviced during a clock cycle. It also combines the advantages of an Input Queuing switch and an Output Queuing switch. In a VOQ switch, each input maintains N queues, one for each output. By using VOQ, no additional speedup is required and HOL blocking can be eliminated. But, even with VOQs, there is still the potential for contention across the crossbar fabric; this is true in two ways:

- Multiple inputs could be requesting access to the same output. (An output can only send one packet per clock cycle.)
- Multiple VOQs on the same input could be granted access to an output. (An input can only send one packet per clock cycle.)

A fair chance round robin arbiter based on the iSLIP algorithm is required that manages these contentions, scheduling traffic in such a way that the crossbar fabric is able to achieve maximum throughput within one short cycle. A network switch based on fair chance round robin scheduling takes traffic in on one port (ingress), and sends it out another port (egress). Each request is being given equal chance to send the data. There is no condition for starvation or idle state.

2. RECENT WORK

Current design in Network on chip (NoC) typically use standard round robin token passing scheme for arbitration [3, 4]. In computer network packet switching, previous research in round robin algorithms have reported results on an iterative round-robin algorithm (iSLIP) and a dual round-robin matching (DRRM) algorithm [2]. Furthermore, Chao et al. describe a design of a round-robin arbiter for a packet switch [5]. Chao et al. refer to their hardware design as a Ping Pong Arbiter (PPA). In general, the goal of a switch arbiter in a packet switch is to provide control signals to the crossbar switch fabric as shown in Figure 1. In a packet switch design, one must keep in mind that each input port can potentially request connections to all output ports (e.g., in the case of broadcast). Theoretically, to avoid the HOL block problem, in a packet switch with M input ports and N output ports, each input is allocated N VOQs (one per output) for a total of N^2 VOQs in the packet switch. In general, an MxN switch can have fewer VOQs than N to save cost and area at some slight cost of occasional HOL blocking. However, we assume V=N VOQs in this paper. [4]

Figure 1 shows a 32x32 network switch with thirty-two input ports and thirty-two output ports. Each input port can request between zero (none) and thirty-two (all) connections to output ports. To accomplish this, thirty-two 32x32 Switch Arbiters (SAs), shown in the bottom right hand side of Figure 2(a), take as input 32x32 requests (req (0, 0), req (0, 1), …, req (31, 30), req (31, 31) - 32 requests per input port, or one request per VOQ) and translates those requests into 32x32 grant signals (one grant signal per possible VOQ to output connection) where at most one grant signal per output port is set to '1' on each clock cycle (thus, of the 322 grant signals, at most 32 are set to '1' each clock cycle).[4]

Each SA grants one request out of at most 32 requests from thirty two VOQs. Each input of the 32x32 SA in Figure 2(a) is connected to a specific VOQ (one per input port) which may request output port 0. The thirty-two outputs of the 32x32 SA are grant signals indicating which of the 32 VOQs is granted output port 0 (note that if no VOQ requests the output port, then all grant signals will be '0' in this case).[4]

3. FAIR CHANCE RRA DESIGN

3.1 BASICS

In fair chance RRA design (FCRRA) there is no chance of request die because of remaining in idle state as not given attention (no chance of starvation).Each and every request is given a fair chance according to the token bus priority. It also allows any unused time slot to be allocated to a master whose round-robin turn is later but who is ready now. A reliable prediction of the worst-case wait time is another advantage of the round-robin protocol. The worst-case wait time is proportional to number of requestors minus one. [4]

3.2 ALGORITHM

The protocol of a fair chance round-robin token passing bus or switch arbiter works as follows (refer Figure 2). In each cycle, one of the request (in round-robin order) has the highest priority (i.e., owns the token) for access to a shared resource. If the token-holding request does not need the resource in this cycle, the request with the next highest priority who sends a request can be granted the resource, and the highest priority master then passes the token to the next master in round-robin order.

3.3 FAIR CHANCE BUS ARBITER

Figure 3 shows a FCBA generated to handle four requests. To generate a FCBA, FCRRAG takes an input the number of requests and produces synthesizable Verilog code at the RTL level within one short cycle.

- In the BA, there are four priority logic blocks to handle four request to produce four grant request according to the token employed through the ring counter.
- In priority logic 0, the req [0] is assigned the highest priority then req [1] then req [2] and finally req [3] is assigned the lowest priority. These priorities rotate in a circular fashion. Like for priority logic 1, req [1] is at the highest priority and req [0] is at the lowest priority and so on.
- To implement FCBA we use token ring in the network. The possession of the token allows the priority logic block to be enabled. Thus only one priority block is enable at one token value which asserts the grant signal.
- In this FCRRA design, each request waits for no longer than M-1 time slot, where M is the number of input.This protocol guarantees the dynamic priority assignment to the requestors without starvation.

3.4 FAIR CHANCE SWITCH ARBITER

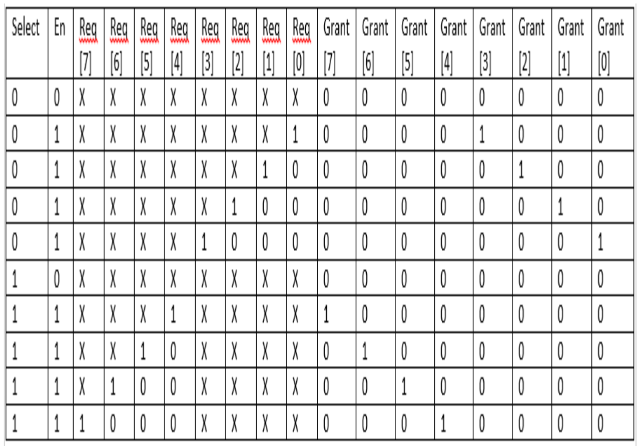
FCRRA switch can be generated by FCRRAG tool. It uses 4x4 FCRRA switch arbiter blocks to implement an MxM FCRRA switch arbiter. FCRRAG is most efficient when the inputs are power of 2 or multiple of two.

Like for example 8x8 FCRRA switch is designed using two 4x4 FCRRA structures. (Refer figure 4)

3.5 IMPLEMENTATION OF THE DESIGN

8x8 FCRRA is being designed and implemented using FCRRAG tool. In this design two 4x4 FCRRA bus arbiter are used in parallelism to obtain the desired result in Xilinx ISE suite 14.5 .The code is being written and synthesize in Verilog.(refer Figure 5)

Table 1. Truth table for 8x8 FCRRA block



Select	En	Req [7]	Req [6]	Req [5]	Req [4]	Req [3]	Req [2]	Req [1]	Req [0]	Grant [7]	Grant [6]	Grant [5]	Grant [4]	Grant [3]	Grant [2]	Grant [1]	Grant [0]
0	0	X	X	X	X	X	X	X	X	0	0	0	0	0	0	0	0
0	1	X	X	X	X	X	X	X	1	0	0	0	0	1	0	0	0
0	1	X	X	X	X	X	X	1	0	0	0	0	0	0	1	0	0
0	1	X	X	X	X	X	1	0	0	0	0	0	0	0	0	1	0
0	1	X	X	X	X	1	0	0	0	0	0	0	0	0	0	0	1
1	0	X	X	X	X	X	X	X	X	0	0	0	0	0	0	0	0
1	1	X	X	X	1	X	X	X	X	1	0	0	0	0	0	0	0
1	1	X	X	1	0	X	X	X	X	0	1	0	0	0	0	0	0
1	1	X	1	0	0	X	X	X	X	0	0	1	0	0	0	0	0
1	1	1	0	0	0	X	X	X	X	0	0	0	1	0	0	0	0

4. IMPROVEMENT FROM THE OTHER DESIGN

- The FCRRA switch arbiter generated performs better than the ping pong arbiter and programmable priority encoder by the factor of 1.9X and 2.4x, respectively.
- As the number of requestor increases, the design complexity will increase leading to increase in the area and latency.
- This FCRRA can be generated by switch arbiter using 4x4 design instead of 2x2. It has been observed that FCRRA shows an increase in the clock frequency. Employing 4x4 FCRRA switch arbiter design block design gives 16% area saving and 36% gate delay reduction compared to the design obtained by 2x2.
- Ping pong arbiter uses 2x2 switch arbiter block as the basic building block.[5]
- Programmable priority encoder [4] and hierarchical round robin arbiter uses pointer updater design to reallocate the request after the current arbitration. Even though the area saving is 19.53% in HRRA [6] but the delay and the area increases as the design gets larger when M increases which reduces the speed of arbitration.
- There may arise some issues in FCRRA design due to circular priority or due to increase number of inputs but pipelining the RRA may alleviate these issues, the latency introduced at pipelining is not desired, especially in NoC designs.

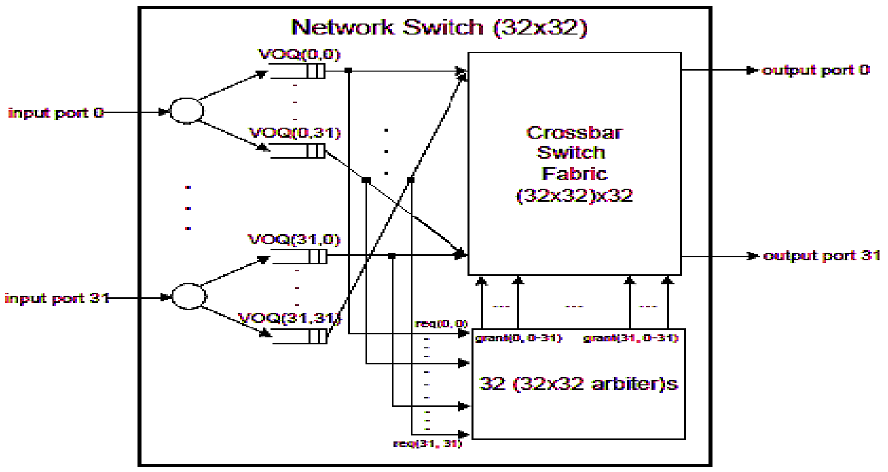


Figure1: Network Switch Arbiter Courtesy By: [4]

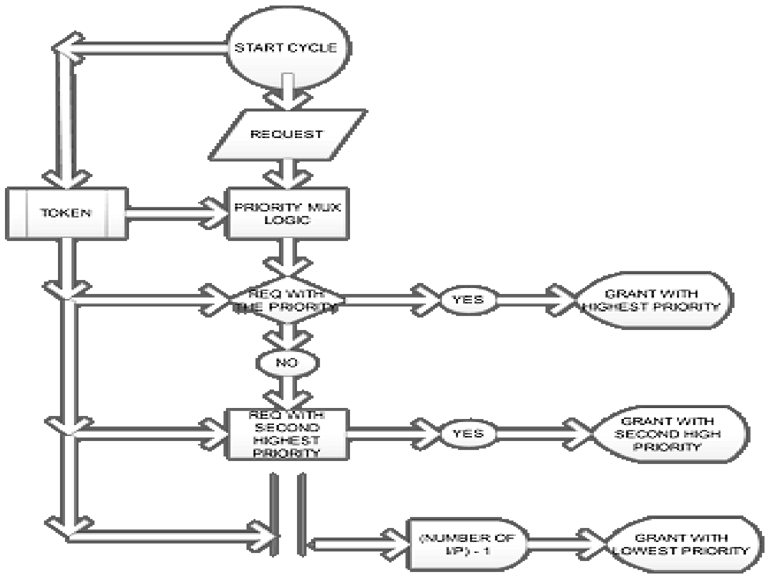

START CYCLE
REQUEST
TOKEN
PRIORITY MUX LOGIC
REQ WITH THE PRIORITY
YES
GRANT WITH HIGHEST PRIORITY
NO
REQ WITH SECOND HIGHEST PRIORITY
YES
GRANT WITH SECOND HIGH PRIORITY
(NUMBER OF I/P) - 1
GRANT WITH LOWEST PRIORITY

Fig 2: Algorithm FCRRA Protocol

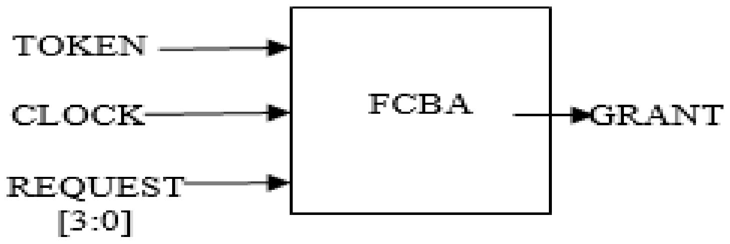


Fig 3: Fair Chance Bus Arbiter

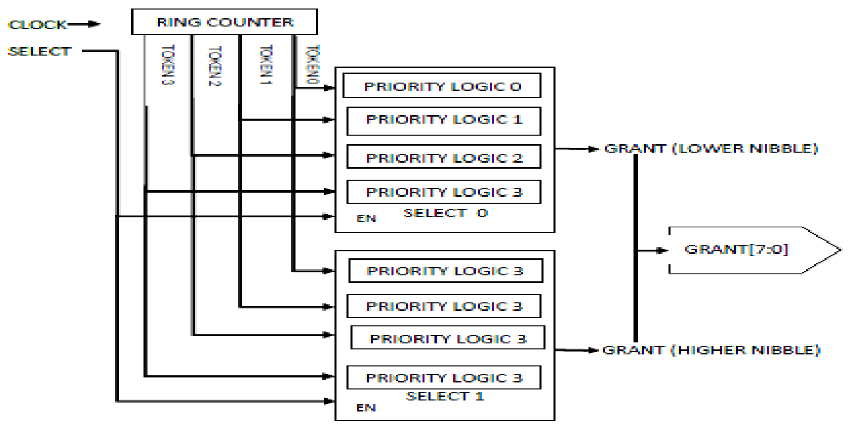


Figure 4: 8x8 FCRRA logic diagram

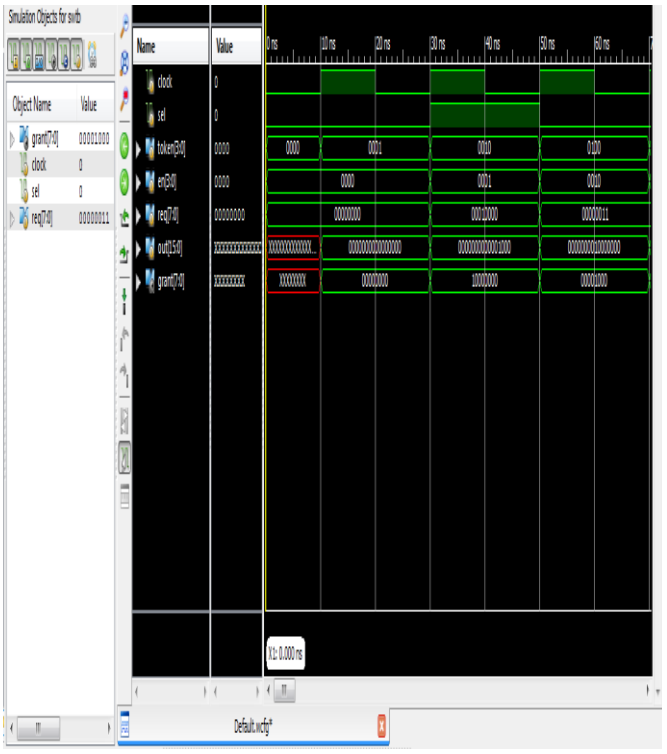


Figure 5: Implementation of 8x8 FCRRA design waveforms

5. CONCLUSSION

A major concern in computer networks today is the design of ultra-high speed switches, which provide a high speed and cost-effective contention resolution scheme when multiple packets from different input ports compete for the same output port. Fair chance round robin arbiter design has been proposed in this paper to produce fair, fast and efficient arbitration to all the requests arriving at the system using cyclic priorities. FCRRAG tool introduced in this paper can generate a BA to handle the exact number of request for both on chip and off chip within one short cycle. We also discussed that FCRRAG is most efficient when the inputs are power of 2 or multiple of two. We also revived the comparative analysis of the different RRA design. And finally we implemented the design 8x8 FCRRA switch arbiter using two 4x4 FCRRA.

6. ACKNOWLEDGMENTS

I would thank the lord almighty Huzoor Saheb Sayed Ali Mira Datar Bapu, Huzoor Mohammed Khan Saheb and my grandparents for bestowing their blessings upon me and filling me with positivity. I am also thankful to God, my family and my parents for their blessings and support.

I am really grateful and indebted to Dr. Ghanshyam Purohit (Dean Research, Sir Padampat Singhania University), Mr.

Arun Kumar (Assistant Professor, Department CSE at Sir Padampat Singhania University), Mr. Arjun Singh (Assistant Professor, Department CSE at Sir Padampat Singhania University), Mr. Vishal Sharma (Assistant Professor, Department ECE at Sir Padampat Singhania University) and Mr. Bhargav Tarpara for their great assistance and valuable comments and suggestions on improving the quality of the paper.

7. REFERENCES

[1] Si Quing Zheng and Mei Yang, "Algorithm Hardware Code sign of Fast Parallel Round Robin Arbiters," IEEE transactions on Parallel and distributed Systems, vol 18, no.1 January 2007.

[2] Yihan Li, Shivendra S. Panwar, H. Jonathan Chao, "The Dual Round Robin Matching Switch with Exhaustive Service."

[3] M. J. Karol, M. Hluchyj, and S. Morgan, "Input versus output queuing on a space-division packet switch," IEEE Trans. on Communications, vol.35, pp. 1347-1356, 1987.

[4] Eung S Chin,Vincent J. Mooney III and George F Riley, "Round Robin Arbiter Design and Generation," ISSS'02, October 2-4 2002 Kyoto, Japan.

[5] H. J. Chao, C. H. Lam, and X. Guo, "A Fast Arbitration Scheme for Terabit Packet Switches," Proceedings of IEEE Global Telecommunications Conference, 1999, pp. 1236-1243.

[6] N.Sertac Artan, Ming Yang and H.Jonathan Chao, "Hierarchical Round Robin Arbiter for High Speed, Low-Power and Scalable Networks –on-Chip.

[7] N. Mckeown, P. Varaiya, and J. Warland, "The iSLIP Scheduling Algorithm for Input-Queued Switch," IEEE Transaction on Networks, 1999, pp. 188-201.

[8] Nick McKeown,"The iSLIP Scheduling Algorithm for Input-Queued Switches" IEEE/ACM transactions on networking, vol. 7, no. 2, April 1999.

[9] H. J. Chao and J. S. Park, "Centralized Contention Resolution Schemes for a Larger-capacity Optical ATM Switch," Proceedings of IEEE ATM Workshop, 1998, pp. 11-16.

[10] P. Gupta and N. Mckeown, "Designing and Implementing a Fast Crossbar Scheduler," IEEE Micro, 1999, pp. 20-28.