International Journal of Engineering Technology Science and Research
IJETSR
www.ijetsr.com
ISSN 2394 – 3386
Volume 4, Issue 8
August 2017

A Survey on Arbitration Techniques for SoC Design

S.Sundara Mahalakshmi,
PG scholar., National Engineering College
Mr. S.Cammillus,
Assistant professor., National engineering college

ABSTRACT--- Network-on- Chip is a distinguished communication for System-on- Chip (SoC). Arbiter is a heart of the Network-on - Chip router to transfer the data from source to destination based on the priority. The arbitration algorithm contains Index format of round robin arbiter to gain lower power consumption and chip area and critical path delay for on - chip bus communication. Better performance can be accomplished with this arbitration structure results indicated . Based on the performance study , to meet the design necessities, importance arbitration scheme can be custom tuned. Round robin arbiter, Matrix arbiter and Index- based Round Robin (IRR) arbiter mechanism are implemented in this paper. The arbiter was implemented on FPGA and synthesized by XILINX 14.2 version.

KEYWORDS--- Arbiter micro-architecture, System-on-Chip, Index-based Round Robin, Network-on- Chip.

I. INTRODUCTION

A rapid progress in Very Large Scale Integration (VLSI) in the past recent years has resulted in the fabrication of millions of transistors on a single silicon chip. With the current CMOS technology it is possible to implement a design with approximately one billion transistors on a single chip. System on Chip (SoC) contains programmable components such as processor cores or application specific intellectual property cores (IP), On chip memory, I/O devices [2].

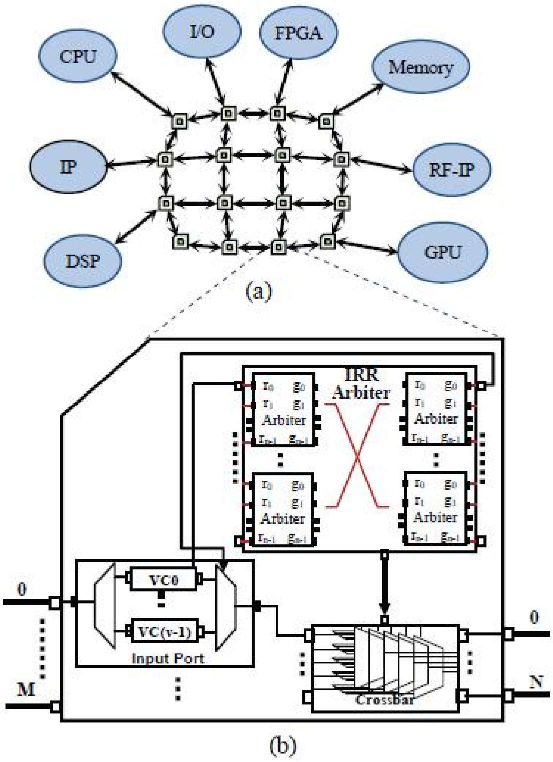

Figure 1. (a) 2D SoC mesh. (b) NoC router.

The most common usage of arbiters is the shared-bus arbitration of a bus-based system where multiple master modules can initiate their transactions [1]. Arbiter is a controller and it controls the arbitration of the ports. Arbiters are used to dispense and access shared resources such as a buffer, channel or switch-port. It knows the information of all ports. Packets from different input ports may have to go to the same output port. Arbiter decides in which order the packets will be served and to which ports. Arbitration works only when there is more than one request for the same port otherwise data flow occurs normally [2].

The advantages of NoC's are architecture, performance, reusability and scalability than bus-based system-on-chip. When multiple packets from different input ports compete for the same output port the NoC's switches should provide high speed and cost-effective contention resolution scheme.

The design of embedded systems following the quick technological progress. Network-on-Chip (NoC) here many IPs (Intellectual Property) such as processor cores, memories, DSP processors and peripheral devices are placed together, on a single die.

In fig.1, new centralized arbiter used in arbitration of a crossbar switch in NoC and shared resources. The Structure of an n × n switch and a crossbar switch fabric implemented as switches between input ports and output ports. The arbiter work is to decide a set of contention-free connection between input and output ports [3].

II. THE MECHANISM OF ROUND-ROBIN ARBITER

The functionality of a round-robin arbiter explained as a request that is just granted request will have the lowest priority on the next arbitration cycle. The round robin arbiters are simple, starvation free and easy to implement. In router, the number of input ports varies with the interconnect architecture [5].

Flits may reach at the same time more than one virtual channel but an arbitration mechanism is to allow only one virtual channel to access a single physical port. Round-robin, Matrix-arbiter, Index-based Round Robin (IRR) arbiter and the other different arbitration mechanisms are applied when consider the proposals of arbitration in the router.

Fixed priority arbiter in the design of network on chip, has no limit for waiting time of the request with lower priority [6, 7]. The input request gets the resource has lowest priority in Round-robin arbiter, so that the Round-robin arbiter has strong fairness. The arbiter is processing the priorities among many different request inputs in a given period of time and there was many request inputs the same output or resource. The output port of the arbiter is connected to the crossbar when the last flit has finished transmission. So that other waiting packages could use the output by the arbitration.

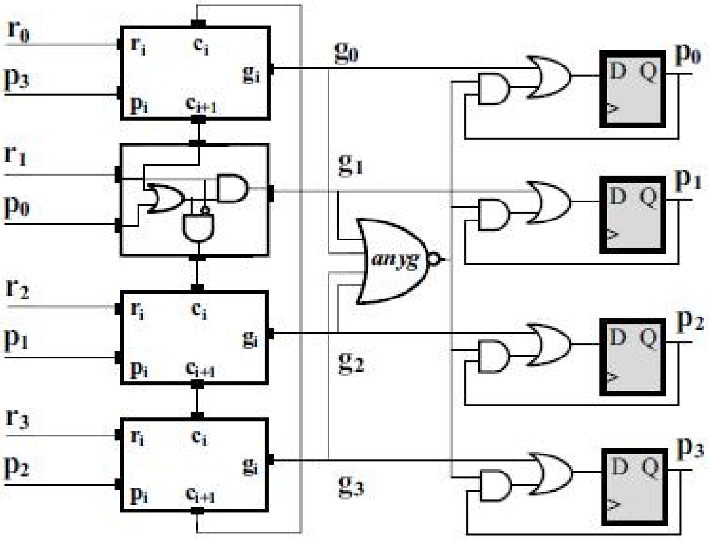

Figure 2. RoR arbiter architecture.

The arbiter is required to access the resource to one request at a time. For example, n-input arbiter is used to arbitrate the resource, such as the virtual channels (VCs) connected to that input port. By asserting its request, each virtual channel that has a flit to be send requests accesses to the input port. We assume there are 8 virtual channels, such as VCI, VC2 ... VC8 separately, if VC2, VC4, VC6 send the requests simultaneously, at this time, assuming VC4 has higher priority than the VC2 and VC6, then the resource should be allocated to it, VC 2 and VC6 will be in waiting state and send the requests in the next clock cycle. A round-robin arbiter operates on the principle that a request which was just served should have the lowest priority on the next round of arbitration [5].

III.THE MECHANISM OF MATRIX-ARBITER

Matrix Arbiter works on the principle of least-recently-served policy. It is the strong fairness arbiter. Matrix arbiter has four inputs. Solid boxes denoted the six flip flops maintained the state in the upper triangular portion of matrix. Each of the shaded boxes represents the inverted output of the diagonally symmetric solid boxes in the lower triangular portion of matrix. Matrix Arbiter implement a least recently served order of priority and it maintains a triangular array of state bits pmn. The element pmn in row m and column n indicates that request m takes priority over request n.

The 4-input gate architecture of a matrix arbiter is shown in figure3. In the upper triangular portion of the matrix, each block with dotted line describes the S-R latch.T he state is maintained in the 6 S-R latches denoted by dotted line blocks [5]. The complementary output of the diagonally symmetric solid box represented in each of the dotted line blocks in the lower triangular portion of the matrix.

For example r0 request is asserted and bit p02 is set then the signal dis 2 will be asserted for disable the lower priority request 2 as shown in Figure 3. If it is not disable,the request travel to the corre sponding grant output through a single and gate.

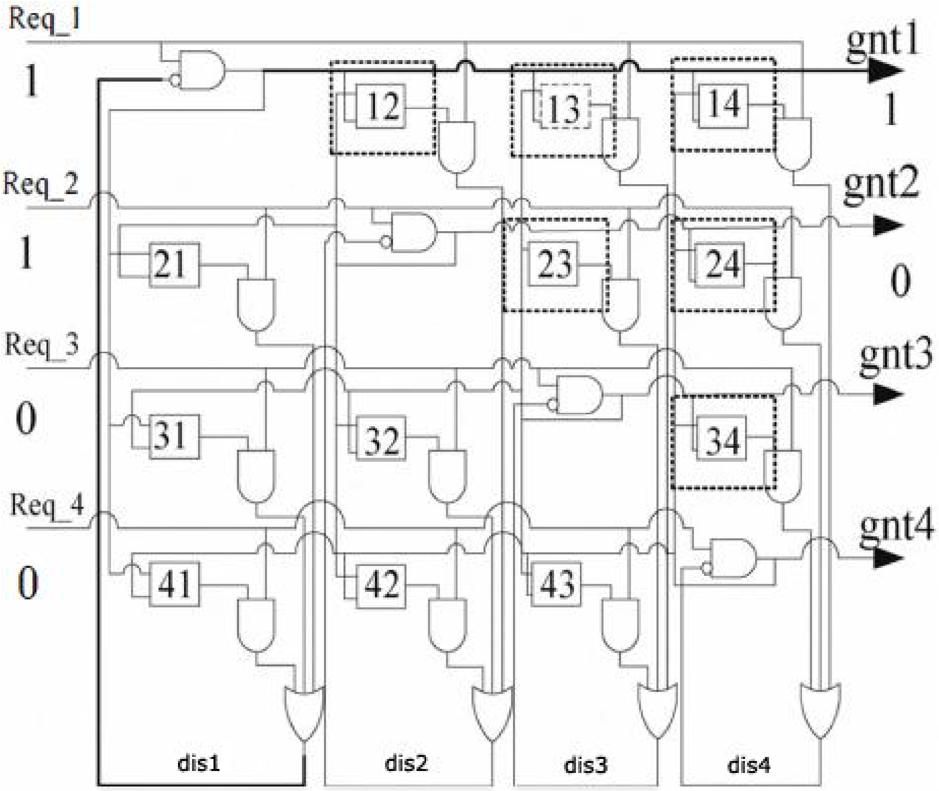

Figure 3. Matrix-arbiter architecture

IV. THE MECHANISM OF INDEX-BASED ROUND- ROBIN ARBITER

Index-based Round Robin (IRR) arbiter that employs a least recently served priority scheme and achieve strong fairness arbitration [5]. This arbiter has smaller arbitration delay, lower chip area and it also consumes less power as compared to the aforementioned arbiters. The fig.4 shows IRR arbiter. IRR arbiter takes one clock cycle for arbitration. It reads the request line and grant the signal only when the input request is having highest priority and highest priority request is not equal to last recently served request.

The IRR arbiter receives the requests and generates grant and grant-id signals. All the arbiters have an output array, grant whose width is the same as that of input width. The index of grant signals, g_id is also generated that is used to address the granted request in some other components, such as control tables, multiplexers and memories used in NoC routers. Router crossbar is made of multiplexers. The g_id can be connected to selection port of multiplexer to switch the granted input to the requested output port. Due to the critical use of g_id in NoC design, all the arbiters covered to generate both grant and g_id as outputs [5].

If the next index of granted request is chosen for the next priority selection, the current granted request receives the least priority, and its next request receives the highest priority among all the requests. To accomplish it, the *g_id* array is stored in a register, and the output of the register is incremented and connected to the selection port of multiplexer, *MP*. To keep the priority unchanged, the output of priority register, *next_g* is fed back through the SF multiplexer to the register to serve for no request.

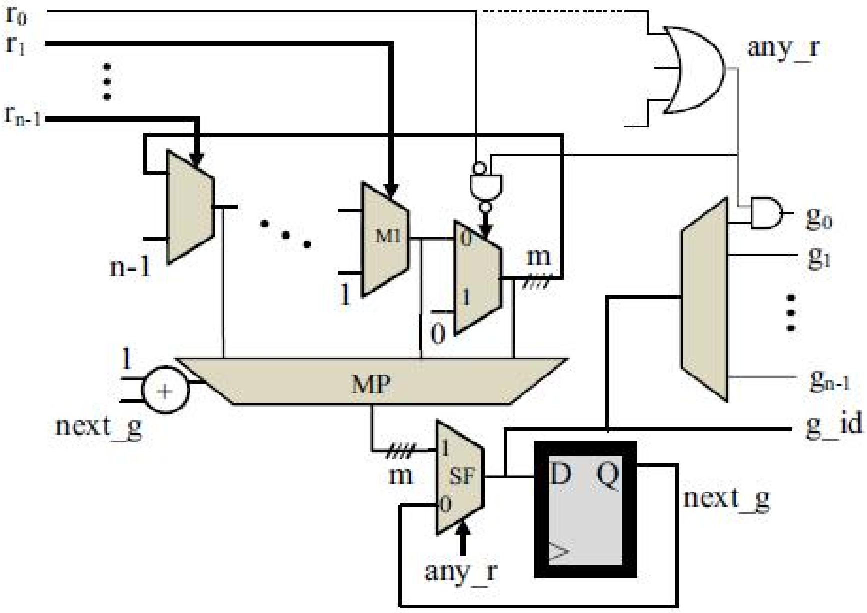

Figure 4. n-input IRR arbiter, *where m=log2 (n).*

V. ANALYTICAL COMPARISION

Hardware analysis performed to compare the expected performance and hardware overhead of the aforementioned Matrix and Round robin arbiters with IRR arbiter. The IRR routers have less chip area, less power consumption, and faster frequency as compared to the Matrix arbiter and Round Robin arbiter. The main advantage of an arbiter circuit are speed, area and power consumption. The maximum clock frequency (Fmax) of an arbiter circuit is the measure for speed. The clock frequency of an arbiter depends on the critical path delay between two registers clocked at the same time.As an Electronic Design Automation software does, we don’t apply any algorithm to optimize the circuits. We calculated the summation of the areas and powers of all the cells of each arbiter to estimate their power and area. The power includes both static and dynamic powers. For speed estimation, the critical path delay between two registers of each circuit is calculated.

HARDWARE CHARACTERISTICS

Table I.Various Arbiters Vs Total Cell Area(μm2)

Input	Design	ASIC design (90 nm Generic Library) (2.5ns)
		Total Cell Area(μm2)
4	IRR	290
	RoR	295
	Matrix	420
8	IRR	702
	RoR	1060
	Matrix	1830
16	IRR	1320
	RoR	1848
	Matrix	7800

Table II. Various Arbiters Vs Power (μW)

Input	Design	ASIC design (90 nm Generic Library) (2.5ns)
		Power (μW)
4	IRR	54
	RoR	72
	Matrix	86
8	IRR	110
	RoR	210
	Matrix	300
16	IRR	152
	RoR	240
	Matrix	1200

Table III. Various Arbiters Vs Critical Path(ns)

Input	Design	ASIC design (90 nm Generic Library) (2.5ns)
		Critical Path(ns)
4	IRR	0.53
	RoR	0.89
	Matrix	0.59
8	IRR	0.67
	RoR	0.80
	Matrix	0.69
16	IRR	1.25
	RoR	1.39
	Matrix	0.65

Table IV.Comparision of Arbiters under various parameters

Design	ASIC design (90 nm Generic Library) (2.5ns)		
	TotalCell Area(μm2)	Power (μW)	Critical Path(ns)
IRR/RoR	20% saving	40% saving	42% Shorter
IRR/Matrix	73% saving	72% saving	6% Shorter

VI.CONCLUSION

There are several arbitration mechanisms can be choose when designing NOC, such as Round-robin arbiter and Matrix arbiter and Index based Round Robin arbiter. However in related literatures, there is lack of analysis, especially in resource, performance and power consumption of them. In this paper, three mechanisms are emulated on FPGA platform. Comparing the difference of them is meaningful for designing arbiters.

The experiment details and the results are presented after comparing IRR with other NoCs utilizing some previous buffering and arbitration approaches. The IRR routers have less chip area, less power consumption, and faster frequency as compared to the Matrix arbiter and Round Robin arbiter. Moreover, the IRR NoC performance shows much higher throughputs and lower average latencies than those of Matrix arbiter and Round Robin arbiter NoCs for various traffic patterns.

REFERENCES

[1] Jagrati Gupta1 and Nidhi Goel2, “Efficient Bus Arbitration Protocol for SoC Design,” in 2015 International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM), pp.396-400, 6-8 May 2015.

[2] Gopika Santhosh, “Design of Network on Chip with an Arbiter”, International Journal of Innovative Research in Computer and Communication Engineering(IJIRCCE), Vol. 3, Issue 9, September 2015.

[3] Masoud Oveis-Gharan and Gul N. Khan, “Dynamic Virtual Channel and Index-based Arbitration based Network on Chip Router Architecture,” in IEEE 2016.

[4] M. Oveis-Gharan and G. N. Khan, "Index-Based Round-Robin Arbiter for NoC Routers," in Proc. IEEE Computer Society Annual Symposium on VLSI, pp. 62-67, Montpellier, July 2015.

[5] Z. Fu and X. Ling, "The design and implementation of arbiters for Network-on-chips," in Proc. Int. Conf. Industrial & Information Systems, pp. 292-295, Dalian, 2010.

[6] K. A. Helal, S. Attia, T. Ismail, H. Mostafa, "Priority-select arbiter: An efficient round-robin arbiter," in Proc. IEEE 13th Int. Conf. on New Circuits and Systems, pp. 1-4, Grenoble France, June 2015.

[7] Chang Hee Pyoun, Chi Ho Lin, Hi Seok Kim, Jong Wha Chong, The Efficient Bus Arbitration Scheme in SoC Environment”, Proceedings of The 3rd IEEE International Workshop on System-on-Chip for Real-Time Applications, 2003.