Increase the efficiency of Network on Chip using buffer sharing mechanism

Farid Daneshgar[1],Majid Taghipoor[2]

**[1]Computer Engineering and Information Technology, PNU University
Orumieh; Iran**

**[2] Computer Engineering and Information Technology, University of Applied Science Technology
Orumieh; Iran**

Abstract

This paper proposes DAMQSV and DAMQS algorithms for systems on chip applications that require an interconnection network. The proposed schemes are based on a DAMQ buffer hardware design.

DAMQSV and DAMQS are excellent schemes to optimize buffer management providing a good throughput when the network has a larger load. They can utilize significantly less buffer space without sacrificing the network performance. Implementing the proposed schemes in hardware requires minor modifications to early implementations of the self-compacting buffer. DAMQS provides an excellent approach to optimize buffer management providing a good throughput when the network has a larger load. DAMQSV scheme lets virtual channels from different physical channel share free buffer space. We have discussed simulation results by comparing with SAMQ and DAMQ algorithms.

Extensive simulations were performed on ModelSim to evaluate the performance of these buffer schemes. The simulations results on different network configurations, such as different traffic mode, network size, virtual channel number, buffer size per virtual/physical channel and routing protocols, were presented to show that these novel buffer schemes especially the DAMQSV scheme are efficient buffer organization methods to be used in the network on chip.

Keywords: *Shared buffer, Virtual channel, Network on chip, Functional ability in networks on chip.*

1. Introduction

System on chip (SoC) designs is becoming widely used in telecommunication, consumer electronics and multimedia areas. [1]. SoCs are anticipated to have a number of components (or modules) that will interact to compute a solution. These components will need to communicate to transfer data and/or control information. Having dedicated connections between any given modules could be extremely complex as the number of modules increase. One of the major problems for future SoC designs is the non-scalable global wire delays [9].An alternative could be use an interconnection network within the chip. An interconnection network on chip needs be restricted in terms of area due to the constraints of being in a single chip. Thus, it is extremely important to use schemes that require less hardware resources as well as provide a good performance.

In order to accommodate these increasing network performance demands, many efforts have been made toward improving interconnection networks. One such improvement is the introduction of true fully adaptive buffer schemes.

Buffers are the instrumental elements in router input and output channels. They were largest leakage energy consumers in NoCs. On balance, the effective and resourceful management of buffers and hence input and output channels in NoC routers has a crucial effect in performance and efficiency of interconnection networks.

In this paper we present a novel shared buffer scheme that is based on a dynamically allocated multiple queue (DAMQ) buffer. This scheme provides similar performance as statically allocated multiple-queue (SAMQ) and DAMQ buffers using less and balance buffer space and, therefore requiring less hardware.

This paper presents the implementation and management of multiple queues and of sophisticated flow control information in hardware, inside a high-speed VLSI router . Result simulations were performed on ModelSim to evaluate the performance of these buffer schemes.

2. Overview in ATLAS

The ATLAS environment automates the various processes related to the design flow for some of NoCs (e.g. Hermes, Mercury) proposed by the GAPH Group[4] and eventually by other groups with which we collaborate. ATLAS is written in Java, allowing its execution in different hardware/software platforms.

The ATLAS environment is composed by six tools:

- Traffic Generation: The Traffic Generation tool produces different traffic patterns, for different injection rates and source/target pairs (e.g. random and complement)
- NoC Simulation: The NoC Simulation tool invokes an external VHDL simulator: ModelSim.
- Performance Evaluation
- Power Evaluation
- Noc Debug
- Tester: Tester is a developer tool that allows verifies if the Atlas project generation (NoC + traffic) is correct.

3. Existing routing algorithm

Routing algorithms are used to determine the messages path from source to destination. They can be implemented in two ways, which are deterministic and adaptive ways**.**
Deterministic routing protocol chooses the path for a message only by its source and destination.
Deterministic routing protocol chooses the path for a message only by its source and destination. All packets with the same source and destination pair will follow one single path. The packet will be delayed if any channel along this path is loaded with heavy traffic, and if a channel along this path is faulty the packet cannot be delivered. Thus the deterministic routing protocols are prone to suffer from poor use of bandwidth, blocking when alternative paths are available.
Adaptive routing protocols are proposed to make more efficient use of bandwidth and to improve fault tolerance of interconnection network. In order to achieve this, adaptive routing protocols provide alternative paths for communicating nodes. Thus it can overcome the congested areas in the network. Several adaptive routing algorithms have been proposed, showing that message blocking can be considerably reduced, thus strongly improving throughput.[14] Among them, routing algorithms based on Duato’s design methodology [16] are very popular.

4. Existing schemes buffer

There are two major buffer schemes, SAMQ and DAMQ, that in following we will compare each other of them.

4.1. SAMQ

Queue is allocated for each router. For a router with four input channels and four output channels and four crossbars to sixteen buffered crossbar is needed. There are four pointers to read and to write four pointers. (Figure 1) This approach has the following problems:
A) The available memory space is divided into static.
B) It only uses partial of the buffer space used by SAMQ. [13]

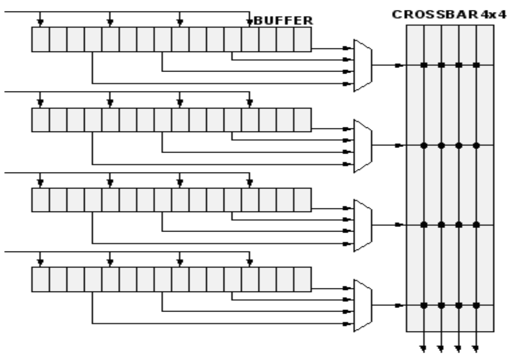

Fig. 1 Allocation Static Queue

4.2. DAMQ

This scheme can allocate the buffer for each router. For a router with four input channels and four output channels and four crossbar to sixteen buffered is needed. (Figure 2)
Four pointers for reading and writing have four pointers. In this model, the memory space is divided dynamically between the ports.

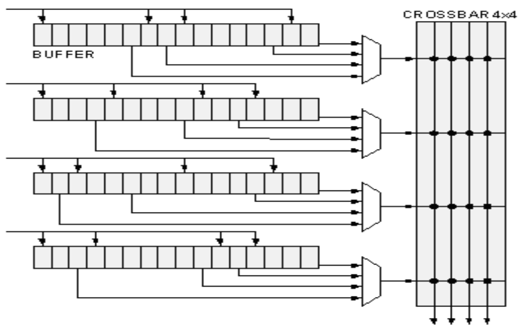

Fig. 2 Allocation Dynamic Queue

4.3. Linked LIST BUFFER SCHEME

In order to let multiple queues of packets share a DAMQ buffer, linked lists can be used to implement the buffer scheme [18] [17] [9]. The basic idea of this approach is to maintain (k+1) linked lists in each buffer: one list of packets for each one of the (k-1) output ports, one list of packets for the end node interface and one list of free buffer blocks. Corresponding to each linked list there is a head register and a tail register. The head register points to the first block in the queue and the tail register points to the last block. In each output queue, next block information also must be stored in each buffer block to maintain the FIFO ordering.

4.4. DAMQS and DAMQSV Buffer Schemes

For optimizing the use of buffer and decreasing delayed flits we suggested two solutions. In first solution we use shared buffer and in second solution we used virtual channel.

4.5. Shared buffer dynamic allocation scheme

DAMQ dynamically allocate buffer blocks according to the packet received. Compared with statically allocated buffer scheme, the advantage of DAMQ is that it uses efficiently the buffer space by applying free space to any incoming packet regardless its destination output port .[17]
Since there is no reserved space dedicated for each output channel, the packets destined to one specific output port may occupy the whole buffer space thus the packets destined to other output ports have no chance to get into the buffer.

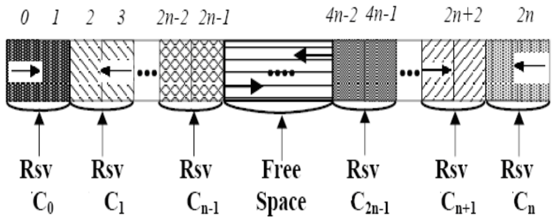

Fig. 3 Dynamic allocation shared buffer scheme

In figure 3, a new scheme to reserve space for all physical channels is defined. As you can see in Figure 3, each channel has its own space to share. In scheme, space is reserved for each port considered.[18]
The buffer space is occupied as opposed to. The registry is used to refer to the beginning and the registry is used to refer to the end of buffer space .If Ci and Cj show two channels, after the buffer Ci was filled. Channel Ci can use the channel buffer Cj.
In Figure 4, both buffers are moving toward each other. If the buffer was saturated, Another buffer is checked.

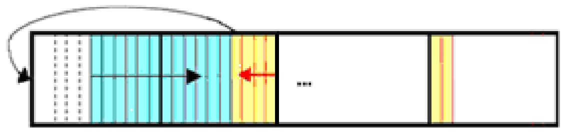
Fig. 4 Shared dynamic allocation of buffer after filling first buffer

4.6. Dynamic allocation buffer scheme with virtual channel:

In the scheme you can also use the virtual channel sharing mechanism.
Because of the wormhole routing algorithm that uses virtual channels, then traffic is not distributed evenly across all buffer space and the number of virtual channels a large part of the capacity remains empty and unused. One approach to solving this problem, allowing the use of the virtual channels of each other.
If Flits, fill your virtual channels,From virtual channels used the same physical channel.If whole virtual channels related to physical channels be filled From physical channel endpoint is used. In Figure 5 is shown in one From your suggestions.

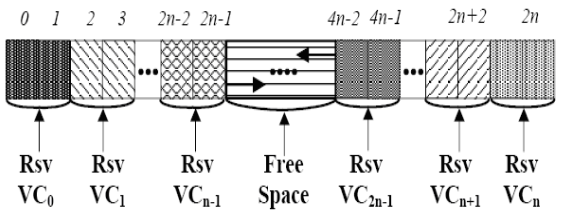

Fig. 5 Dynamic allocation buffer space with virtual channel

In this figure, for each virtual channel has been considered in two places. If fill a virtual channel, from virtual channel is without are used to reserve space.

5- Valuation of efficiency

This section presents the results of simulation experiments conducted to evaluate the performance of our novel buffer organization schemes proposed in previous sections.
Used in the simulation From Hermes routers. The Hermes router has centralized switching control logic and five bi-directional ports. The Local port establishes a communication between the router and its local core. The other ports of the router are connected to neighbor routers. A physical channel, including the local port, may support multiplexed VCs. Each input port has a depth d buffer, for temporary flit storage. When n VCs are used, a buffer with d/n depth is associated to each VC.
Figure 6 shows the eastern gate of the Flit received send flit delay is 580 nano seconds.

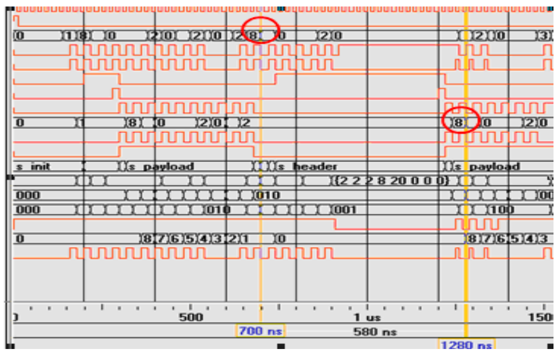

Figure 6- Simulation of router without virtual channel and shared buffer

6. Configuring simulated network

This section presents a model for easily predicting average message latency of traffics in a network on chip where wormhole switching and fully adaptive routing protocols is used.
We use a 2D torus network as our analysis target to illustrate the model for simplicity. Fig. 7 shows a 4-ary 2-cube (torus) message exchanging network

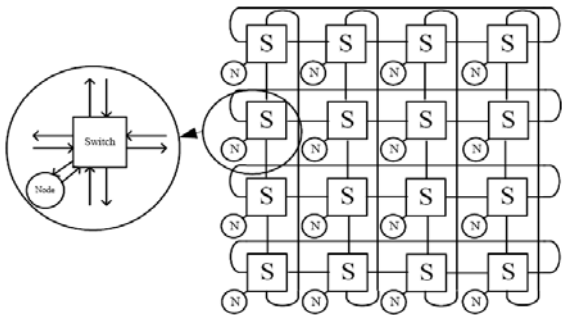

Fig. 7 Mesh network

Simulation parameters are including items below:

- Channels are two-way.
- Each node has a buffer depth of 8 and flit size 16 bit
- Flow Control: Handshaking
- Traffic pattern: matrix transpose
- Number of Virtual Channels: Two
- Routing algorithms: XY
- Techniques are applied to the switch: Wormhole

Figure 8 shows that the matrix transpose traffic pattern is shown in the example, node 31sends to 13and vice versa.

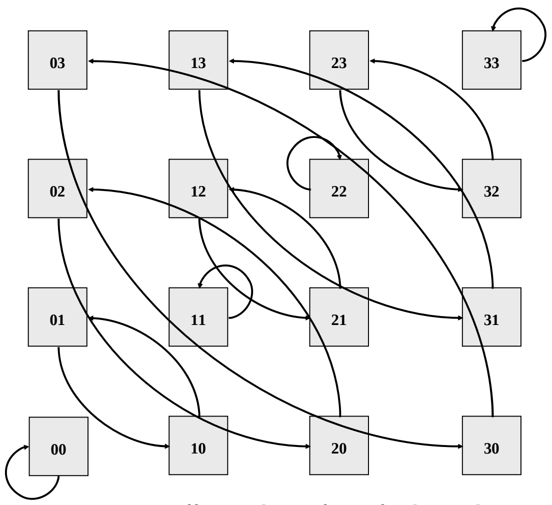

Fig. 8 Traffic template in form of orthogonal .matrix

7- Results of Simulation

The target in first directed experiment, is defining size of the inputting buffer for ideal mode. The experiment performs in this order. We send a file includes 100 packets to IPs that are in different distance from source switch. Distance of destination switch is changing between 1 to 4 hops.
When a flit enters from one of the gates of a switch to exit from another gate of the same switch, switching delay is calculated with the following formula:

$$latency = TF - TI$$

TF is Received time by the outputting gate And TI is the time to receive by the gate.
Figure 9 shows the switching delay. The key point is that for drawing diagram, we used the switch that it's traffic is more than the others.

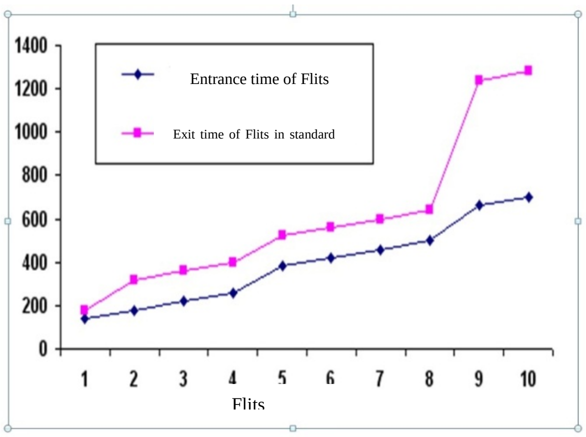

Fig. 9 Comparing Delays

Schemes are the different types compare in terms of latency. Suppose you have two of the buffers.

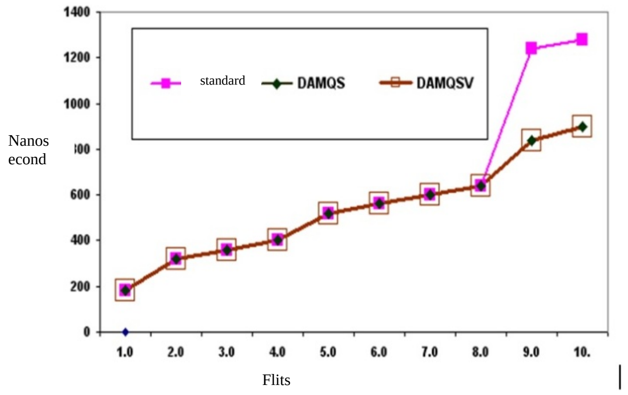

Fig. 10 Comparing Delays

According to figure 10; flits number 8 is the same output. From flit number 9 to next in static allocation scheme flits exit at 840 nanosecond and in dynamic allocation scheme at 840 nanosecond. According to cursor 1 and cursor 2 in figure 11 and 12, the results of the simulation are verified.

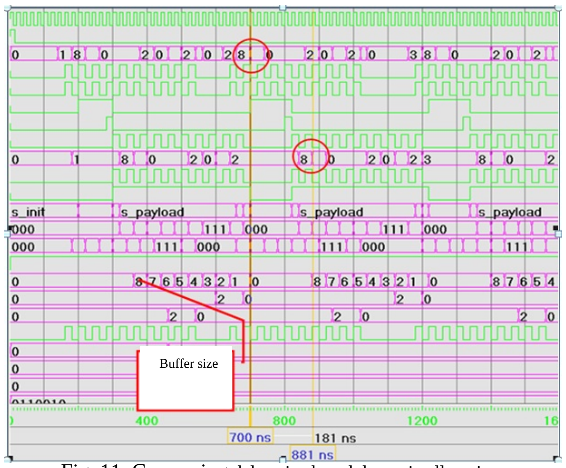

Fig. 11 Comparing delays in shared dynamic allocation

The reason of equivalence of flits exit is that, although this diagram related to switch with high traffic but, because in virtual channel shared dynamic allocation, Switching time from buffering gate to other buffering gate happen less than nanosecond, and in shared dynamic allocation, virtual channel is almost empty and switching is done rarely, so virtual channel shared dynamic allocation has a little delay, next a formula for calculating routing delay and judgment to the other routers is presented.
In each message, numbers of four cycles for judgment and routing to other router, is ST hour cycle each writing action take along two hours cycle and considering that subject flit should be saved inside of buffer to process, so ideal pipeline should work with this formula.[10]

$$all\_time\_packet = (ST + (NF - 1) * 2) * NP$$

NF: Number of Flits
NP: Number of packets (here is 100)
Considering the formula, calculated delay is:(table1)

Table 1: delay calculated

SAMQ	DAMQSV	DAMQS	Normal
98	63	61	98

This formula is used in ideal mode and for buffer size 6 and higher in normal switch, ST is about 10. By placing those values in above equation, the whole charged time for delivering 50 packets with 39 flits is equal with 4300 hour clock.[10]

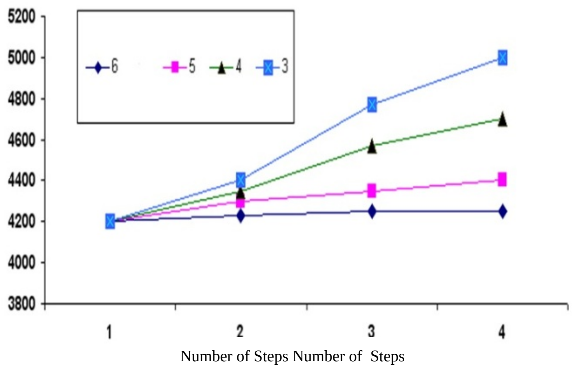

Fig. 12 entire time for delivering 50 packets for different buffer sizes

Then if the depth of the buffer decreases to 6 flit, in the general mode it makes a little change in efficiency of the implemented model.
But more than it, the amount of time ST increases to 10.5 to 11 and gets the time of simulation for carrier packet 39 flits, from 4300 second about 4550 seconds.[10]

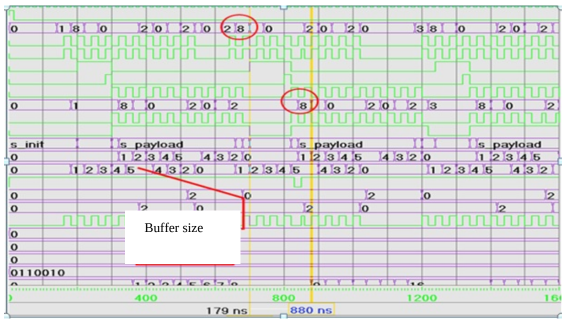

Fig. 13 entire time for delivering 50 packets for different buffer sizes.

Then we can decrease the size of the buffers in each virtual channel, from 8 units to 6 units, as you see in figure 13, there is very little impact in latency of switch and in the general mode the latency of the entire tested model depends on confirmation of the source.

8- Conclusion

In this paper we implement a buffer scheme based on Dynamic allocation with compressed and managed buffer . One of the characteristics of implemented both schemes is reducing buffer space. In both of them we can reduce buffer space from 8 to 7 or 6. It means we are saving 12.5 to 25 percentages in using memory.
The best characteristic of first scheme was using a lot of unused gates buffer space and the second scheme uses virtual channels vice versa, shared dynamic allocation and normal mode that use less than them. Increasing of delay was the obvious characteristic of both

9. Reference

[1] [AND03] Andriahantenaina, A.; et al. "SPIN: a Scalable, Packet Switched, On-Chip Micro-network". In: Design Automation and Test in Europe Conference (DATE'2003), 2003, pp. 70-73.

[2] [BEN01] Benini, L.; De Michelins, G. "Powering networks on chips: energy-efficient and reliable interconnect design for Sacs". In: 14th International Symposium on Systems Synthesis (ISSS'01), Oct. 2001, pp. 33-38.

[3] [BEN02] Benini, L. De Michelle. "Networks on chips: a new SoC paradigm**".** Computer, Volume: 35(1), Jan. 2002, pp. 70-78.

[4] [BER01] Bergamaschi, R.; et al. "Automating the design of SOCs using cores**".** IEEE Design & Test of Computers, Volume: 18(5), Sept.-Oct. 2001, pp. 32-45.

[5] [BER92] Bermam, P. E.; et al. "Adaptive Deadlock and Livelock Free Routing with All Minimal Path in Tours Networks**".** In: Fourth Symposium on Parallel Algorithms and Architectures, 1992, pp. 3-12.

[6] [BOL03] Bolotin E. et al. "QNoC: QoS architecture and design process for network on chip". In: Journal of Systems Architecture, 49, Dec. 2003. Special issue on Networks on Chip.

[7] [CAR04] Carara, E. "Arquiteturas para Roteadores de Redes Intra-chip". Trabalho de conclusão, FACIN-PUCRS, Dez. 2004, 62p.

[8] [CUL98] Culler, D.; Singh, J. P. "Parallel Computer Architecture: a Hardware Software Approach". Los Altos, California: Morgan Kaufmann, 1998, 1100 p.

[9] [DAL04] Dally, W. J.; Towels, B. "Principles and Practices of Interconnection Networks". San Francisco: Morgan Kaufmann, 2004, 550 p.

[10] Canais Virtuais em Redes Intra-Chip: Um Estudo de Caso(Prof. Fernando Gehm Moraes2005)

[11] A DAMQ SHARED BUFFER FOR NETWORK ON CHIP(Jin Liu Delgado-Frias2007)

[12] [DAL90] Dally, W. J. "Virtual-Channel Flow Control". In: 17th International Symposium on Computer Architecture (ISCA), 1990, pp. 60-68.

[13] [DAL87] Dally, W. J.; Seitz, C. L. "Deadlock Free Message Routing in Multiprocessor Interconnection Networks". IEEE Transactions on Computers, Volume: 36(5), 1987, pp. 547-553.

[14] [DUA97] Duato, J. et al. "Interconnection Networks". Los Alamitos, California: IEEE Computer Society Press, 1997, 515 p.

[15] Santi, S.et al. "On the Impact of Traffic Statistics on Quality of Service for Network on Chip", ISCAS052349-2352.

[16] Partha Pratim Pande.et al."Performance Evaluation and Design Trade-offs for MP-Soc Interconnect Architectures",IEEE Trans.Computers,vol. 54, no .8,Aug 2005,1025-1040.

[17] M.A.Jabraeil Jamli and A.Khademzadeh." DAMQ-Based Schemes for Efficiently Using the Buffer Spaces of a NoC Router", IJCSI, Vol 4, No.2,2009.

[18] D. U. Becker and W. J. Dally. Allocator implementations for network-on-chip routers. In proceedings of the 2009. ACM/IEEE Conference on Supercomputing, 2009.

[19] C. Bienia. Benchmarking Modern Multiprocessors. PhD thesis, Princeton University, January 2011.