Indian Journal of Science and Technology, *Vol 9(29), DOI: 10.17485/ijst/2016/v9i29/91236, August 2016*

ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

High Throughput Pipelining NoC using Clumsy Flow Control

S. Anu Karpaga[1*] and D. Muralidharan[2]

[1]School of Computing, SASTRA University, Thirumalaisamudram, Thanjavur - 613401, Tamil Nadu, India; sanukarpaga@gmail.com
[2]School of Computing, Information Technology, SASTRA University, Thanjavur - 613401, Tamil Nadu, India; murali@core.sastra.edu

Abstract

Network on-Chip (NoC) is a novel technology which is used to make the interconnections between the components available in the System on Chip (SoC) design. This technology of NoC is defined and used in two varied forms as buffered and bufferless NoC. Bufferless NoC, a predominant type of network on chip, is used to reduce the cost efficiently by removing input buffers of the router. However, it is evident that this performance gets jammed at high loads because of the increase in the network contentions and deflection of packets in huge amount. To reduce the amount of deflection and to buttress the flow, Clumsy Flow Control (CFC) is used in the bufferless NoC. In this paper, a novel proposal has been propounded into the pipelining mechanism keeping in mind the flawless flow control needed for the bufferless NoC to decrease the impact of deflection routing and to increase the throughput with high injection rate. Employing the pipelining technique into the existing flow control increases the frequency which in turn, is responsible for high throughput. Implementation of the aforementioned pipelining mechanism is done in two stages in the bufferless NoC which helps to increase the throughput as well as the injection rate. Finally, the application with its pipelined implementation, as proposed, will be mapped onto the NoC architecture by using the CFC.

Keywords: Bufferless NoC, CFC, Network on Chip, Pipeline, Router

1. Introduction

Due to the varied intricacies in using multi-core network, it has now become difficult to provide point to point communication link between every two blocks that communicate in a chip. This has led to the emergence of Network-on-Chip (NoC) in chip level communication systems. As the number of cores keeps on increasing, NoC becomes a significant aspect of the multi core architecture. Two types of NoC are available in the architecture - buffered NoC and bufferless NoC. Bufferless NoC provides benefits such as reduced cost because of the absence of input buffers. This makes the usage of bufferless NoC more rational and cost effective. In this paper, we have proposed the concept of applying pipelining mechanism in the NoC using the CFC with the motive of increasing the throughput. There would be a palpable hike in the injection rate of the packet, if the throughput is maintained high which would further substantiate the desirability and effectiveness of this proposal.Thus in this paper we have put forth a new pipelining mechanism and have detailed its application in NoC using the CFC.

Flow control, as defined in both the buffered and the bufferless network[1], says how the allocated network resources are consumed between the links in the network. Buffered flow control is the most commonly used flow control. The entire technique is about buffers which are used to store the packets. As a result, the aggregate area consumed is much higher due to the placements of buffers. Buffered NoC works based on the Credit based flow control technique.With the advent of Bufferless NoC, there is a considerable reduction in the cost of the network which is attributed to the reduction in the usage of redundant buffers, which has appropriated huge space and cost in

**Author for correspondence*

Buffered NoC network[2,3]. In this flow control, packets are not buffered instead they can either be misrouted or be dropped and retransmitted. This bufferless flow control technique results in decrease in the throughput and hence could possibly turn awry when the load of the network increases. To overcome these insidious limitations, we have proposed to introduce the pipelining concept and the flow control into the bufferless network. This technique of flow control is termed as the CFC[4]. The proposed CFC is based on destination as it transmits the data if and only if there is a vacancy in the memory controller. If in case more packets are transmitted which is much higher than the obtainable buffer space, then theother set of packets will be deflected and not essentially dropped. To streamline the better flow of packets, Pipelining is included as a viable option in the paper so that the rate of injecting the packet into the network gets boosted-up. In this paper, pipelining mechanism is envisioned to include two stages which tend to increase the throughput and injection rate of the packets.

2. Clumsy Flow Control (CFC)

2.1 CFC Explanation

In thistechnique offlow control, each router which is flowing upstream maintains a value as a credit, which is then used to denote the number of free buffer entries flowing downstream. When the data is made to flow in a particular direction, the correspondingcredit is supposed to flow in the direction opposite to it. This is defined in the Figure 1. In general, all the packets are partitioned into one or more flits while it is set for transmission and width of the buffer entry represents the flit width. The above concept ofCFC is introduced into a bufferless NoC.This is called as an inexact destination and credit based flow control which is used to avoid the blocking between the network and also to reduce the deflection.

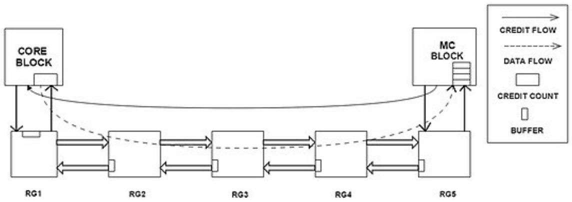

Figure 1. Clumsy Flow Control (CFC).

The credits, which are available in CFC, are continued by cores and these credits represent the availability of the buffer at destination. With every request for the injection of packets into the network, the count maintained by the credit goes down and the counter move, of increasing credit, happens whenever there is a reply from the memory controller. Thus by controlling the flow of the data into the network through CFC, the amount of request is reduced also the deflection associated with it also gets reduced. Thus CFC is said to be destination based as the count maintained by the credits denote the availability of the buffer at the destination as presented in Figure 1.

2.2 Algorithm

In accordance with the process described above the algorithm for the CFC is designed and described as follows in Algorithm 1 which is referred from CFC paper[4]. Each shaded core (*m*) retains two counts for the memory controller MC (*n*), which we denote as r_{mn} and w_{mn} for read and write requests respectively. In the initial stage, $r_{mn} = r$ and $w_{mn} = w$, where *w and r* represents the initial value allotted to every core for write and read requests. For a request for read from the core *m* to the MC (*n*), the core is made to inject into the network and the value of r_{mn} gets decremented. If $r_{mn}=0$, the requests are controlled until the value of the credit is non zero. When the reply is obtained from the MC (*n*)back to the core *m*,the value of r_{mn} gets incremented. Similarly, for the write requests, the value of w_{mn} is used. These values should be greater than 0. As the values of *r* and *w* increases and leads to infinity, CFC represents the bufferless router network without the flow control.

Algorithm 1

```
At each core $m$
for every read request sending to MC core $n$
do
if$r_{mn}$> 0 then
inject the requests ;
$r_{mn}$--;
end if    --end of the if statement
end for
for each response or reply from the MC core $n$ do
if reply comes then
$r_{mn}$ ++;
else
$w_{mn}$ ++;  --count represents the credit value
end if    --end of the if statement
```

```
    end for
```

3. Overview of Router

3.1 Bufferless Router

There are two types of router - buffered router and bufferless router. In this paper, Bufferless (BLESS) router is used which uses the concept of deflection. In the bufferless routers there is a complete absence of buffers and so this deflection router will forward every arriving packet to the output port. Thus, buffer less deflection routing has been implemented in NoC topologies. For further deliberations we have scrupulously considered two-Dimensional (2D) mesh topology as this is conspicuously the most widely used topology, among the various other topologies, as it also gives improved scalability, flexibility and regularity[5]. This 2D topology has been composed of five ports which are defined in Figure 2.

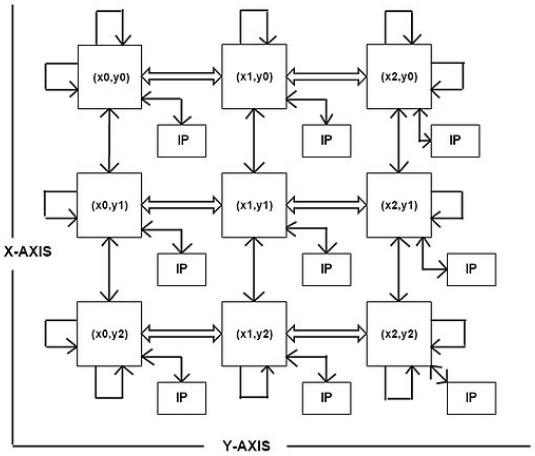

Figure 2. 2D mesh topology network.

The four ports are used to connect all its neighboring routers and the fifth port is the local port which is used to eject or inject the flits from (or) to the network respectively. Each router has been assigned two component considering the unique Y and X co-ordinates of the routers in the 2D networks.

3.2 Stages of BLESS Router

This bufferless deflection router includes four stages which are listed as follows: router, ejector stage, injector stage and allocation of switch and switching fabric stages. These stages are defined in the diagram as Figure 3. This is done without pipelining concept, that is, these stages are done in a sequential manner (i.e.) one after the other, thereby transmitting the packets from the source to the destination. In the Figure 3, the flit present at the input port is represented by a bit (vin, fin).

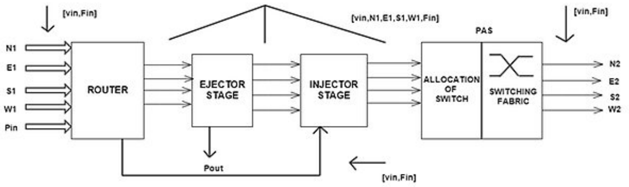

Figure 3. Stages of bufferless router.

The first stage is the router block which is used to check for the arriving packets and transfer the data based on the minimum distance from the source to the destinations. In this stage, each flit has a 4-bit representation which indicates the output port as (n1, e1, s1, w1). The bit is denoted as 1 if the respective output port is active. The second stage is the ejector stage, which is used to remove one of the incoming flits arriving from the network and transfer it to the IP core. This injector stage is used to inject the flit coming from the IP core and passing into the next stage. The final stage is allocation of switch and switching fabric, which is used to connect the set of incoming flits to the set ofthe output ports and depending on this connection the flits are sent to the output port through the switching fabric.

4. Proposed Method

4.1 Pipelining Mechanism

The pipelining mechanism is defined as the mechanism in which the consecutive stages of the process are executed concomitantly, and hence another stage begins before the nextstage could be completed. In this paper, two stages are used, one stage is used for calculating the output link and another stage is used for assigning the link. Of all the four stages that are involved in the bufferless router, the first three stages are bundled together and are considered as the preliminary stage or the first stage for pipelining. The final stage, (i.e.) fourth stage, of the bufferless router is considered as the second stage for pipelining. Thus these two stages are used for the pipelining mechanism.

Pipelining mechanism is used in the bufferless router after including the CFC into the network so as to increase the throughput and increase the injection rate. If the packet is about to be sent from the source to the destination, flits are sent in order. When the first flit is initiated in the first clock cycle, it is made to transmit to the next router depending on the condition and the principle. Then the second flit enters into the router, along side the first flit, and thus this continues until the entire packet is transmitted. This increases the throughput and the injection rate of the packets. Pipe lining concept is generally defined in the Figure 4.

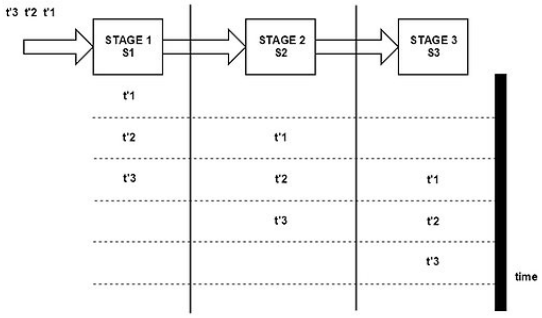

Figure 4. Pipelining stages.

The throughput of the pipelining stage is defined as the measure of the packets sent per clock cycle. When pipelining is included in this NoC, the throughput is expected to improve when compared to the normal one. Thus, pipelining concept is included in this NoC to improvise the throughput.

4.2 Issues

Deadlock and livelock are the most common problems which occur in the network on chip. By using the CFC, the problem of deadlock can be avoided because separate networks are made available for the request and the reply processes. However there is less chance of occurrence of deadlock in the network when the entire network is busy and no write request could be injected. This issue occurs mainly because of the lack of the flow control in this NoC[6].

Livelock is the problem which occurs when more than one process change their state continuously with respect to the changes in other process. The effect is that no process will complete its work. There is a chance of livelock problem in this pipelining technique and this is the drawback that could occur when the concept is included in the paper. As a result of the increasing number of stages in the pipelining mechanism, there occurs a chance of increase in the latency, which is a common problem in the pipelining technique. Latency is the problem which is due to time gap between the stimulus and the result coming out of the router. This is the most common problem in NoC.

5. Experimental Results and Discussion

Results are obtained for both the concepts, that is, with and without pipelining concept. The result inferred from the simulation is the difference in the latency and the throughput. With the help of simulation results for the CFC and CFC with pipeline, the tabulation result is defined in Table 1.

Table 1. Comparison ofthroughput

Pipeline Parameters	Without Pipeline	With Pipeline
Throughput	813.43 MBps	1050.9 MBps

From the table, it is inferred that there is a discernible incidence of latency in the mechanism, whenever pipeline stages are included in the concept. With reference to the minimum deflection, the throughput is obtained in the table as defined[7]. However the increase in the throughput leads to the better result of the output. Thus the throughput is obtained 29% better than the existing one. There is a drawback defined in this paper which is latency. This is not an issue since the throughput has been achieved high.

There are many types of flow control mechanisms and only a few has been used in these NoC papers[8,9]. Low latency concept is brought into the NoC so as to reduce latency however there will be drawbacks such as the performance is not considered into account[10]. Flow control was then brought into the NoC papers with the sole objective to improve the performance. A flow control is used on NoC based on the source utility maximization, which addresses the problem of congestion control[10]. This is implemented by using the centralized controller. Performance is not dealt in this paper amply. Another flow control is used which is mainly to control the injection level and this deal with the rate of controlling of injection packets[11]. This explains how the network could be congested and how the traffic could be adjusted in a network. There are many types of flow control and the final one to be discussed here is the flow control which is used to control the fault by using an error free transmission[12]. Many concepts were used in NoC so as to reduce the power dissipation, area as defined in[12] but in this paper it has been proposed to improve the throughput. In the paper, the survey implies that the design of NoC is much better when compared to that of SoC with the increased performance and reduced consumption of power[13]. These are the type of flow controls referred with an intention to bring the pipelining concept in the Clumsy Flow Control (CFC) and to improve the throughput. However, the disadvantage is the latency problem which often occurs in pipelining mechanism.

7. Conclusion

In this paper, the pipelining mechanism is included in the clumsy flow control to alleviate the performance of the bufferless NoC. This is mainly done to increase the throughput. The scrupulous application of this technique is easily discerned from the output table which shows the efficiency.Result implies that the throughput has been increased by 29% and the injection rate also increases which concomitantly proves decrease of the deflection rate.

8. References

1. Dally WJ, Towles B. Principles and Practices of Interconnection Networks. San Francisco, CA: Morgan Kaufmann; 2004 Jan.
2. Moscibroda T, Mutlu T. A case for bufferless routing in on-chip networks. ISCA. 2009 Jun; 196–207.
3. Hayenga M, Lipasti M, Enright JN. SCARAB: A single cycle adaptive routing and bufferless network. 42nd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO); 2009 Dec. p. 244–54.
4. Kim H, Kim Y, Kim J. Clumsy flow control for high-throughput bufferless on-chip networks. IEEE Computer Architecture Letters. 2013 Jul–Dec; 12(2):47–50.
5. Rantala V, Lehtonen T, Plossila J. Network on Chip routing algorithms. TUCS Technical Report No. 779. Turku Center for Computer Science; 2006 Aug. p. 1-38 .
6. Fallin C. CHIPPER: A low-complexity bufferless deflection router. IEEE 17th International Symposium on High Performance Computer Architecture; 2011 Feb. p. 144-55.
7. Stojanovic IZ, Jovanovic MD, Djordjevic GLJ. Low-cost port allocation scheme for minimizing deflections in bufferless on-chip networks. 21st International conference on Telecommunications Forum; 2013 Nov. p. 357-60.
8. Kang YH, Kwon TJ, Draper J. Fault-tolerant flow control in on-chip networks. IEEE 4th International Conference on Networks-on-Chip; 2010. p. 79-86.
9. Talebi MS, Jafari F, Khonsari A. A novel flow control scheme for best effort traffic in NoC based on source rate utility maximization. Modeling Analysis and Simulation of Computer and Telecommunication Systems. 2007 Oct; 381–6.
10. Mullins R, West A, Moore S. The design and implementation of a low-latency on-chip network. Asia and South Pacific Conference on Design Automation; 2006 Jan.
11. Tang M, Lin X. Injection level flow control for network-on-chip. Journal of Information Science and Engineering. 2011; 27:527-44.
12. Selvaraj G, Kashwan KR. Reconfigurable adaptive routing buffer design for scalable power efficient network on chip. Indian Journal of Science and Technology. 2015 Jun; 8(12):1-9.
13. Beulah HS, Vigneshwaran T, Jasmin M. Survey on energy-efficient methodologies and architectures of Network-on-Chip. Indian Journal of Science and Technology. 2016 Mar; 9(12):1-8.