Dynamic Packet Injection Mechanism to Control Congestion in Conventional NoC

Prof. P.Ezhumalai[1], M. Sriram[2], A. Saranya[3]
Department of Computer Science and Engineering,
R.M.K College of Engineering and Technology,
Anna University, Chennai, Tamil Nadu, India.
ezhumalai.es@gmail.com[1] sriramm17@gmail.com[2], saranya.thj@gmail.com[3]

***Abstract*— Congestion problems not only show its influence in a traditional Computer network system, but also in On-Chip Communication system. The On-Chip system on the phase of reliability, results in congestion due to its necessity of making the packets to wait in queue when congestion arises. When the buffer storage exceeds its capacity, congestion occurs and this may lead to packet discard. To avoid this, packets are made to wait. But this leads to decrease in throughput and increase in latency. A mechanism is proposed to control this effect and also to maintain the Performance of the system as equal as the congestion free NoC. The error control mechanism works on the principle among workload, latency and throughput. The concept behind the proposed dynamic packet injection mechanism is when workload is reduced, latency gets decreased. This causes more number of packets to get transferred and throughput increases. This method is followed when the delay occurs at the buffer end.**

***Index Terms* - Congestion, Buffer delay, Conventional NoC, Fault injection, Dynamic packet injection**

I. INTRODUCTION

Network on Chip (NoC) a new technology came into existence to replace the existing System on Chip technology. The New technology follows Globally asynchronous and locally synchronous approach [1]. This approach has a major impact on increased scalability, decreased execution time in the working environment of NoC systems. Network on Chip (NoC) architectures is a proposed scalable solution to the global communication challenges in nano-scale Systems-on-Chip (SoC) designs. The NoCs with standardized interfaces makes use of previously-designed and third-party-provided modules in new designs [2]. Since reliability is a concerned factor in On-Chip Communication Networks the packets are not preferred to get discarded. If a particular packet suffers from congestion, it is not allowed to get discarded. It is made to wait in queue to get processed. This causes delay in processing of all packets.

Throughput gets lowered because of congestion problem in Conventional NoC and Latency gets increased with the increased running time [3]. To control these problems, a mechanism of dynamic packet injection is proposed.

This mechanism is applied to the buffers that causes delay and the resulting unit has less execution time and high throughput. The proposed techniques involves a NoC system that can recover from Congestion and act as a congestion free unit.

II. CONGESTION IN CONVENTIONAL ON-CHIP NETWORK

The On-Chip Network system is designed in such a way that if the buffer overflows, the packets are not discarded and are placed in the queue to get processed by buffer. In case of traditional network system, the tolerance level is high for dropped packets. But the On-Chip system has a very low tolerance level for the dropped packets [4]. Since there are different types of IP Cores in the system, each flit contributes to a sequential function in the system for a functional unit. The missing of a flit would cause whole system's performance to get reduced at a very low level. So, there is a need for buffers to process each and every flit, even though if overflow occurs. This resulting in reduced throughput and increased execution time of

Conventional NoC. The latency is also increased at this instance.

III. FAULT INJECTION TO SIMULATE THE DELAY IN BUFFER OF CONVENTIONAL NOC

The Conventional NoC that suffers from Congestion problem due to buffer overflow is certain to the discarding of packets[4]. To avoid this, the packets are made to wait for getting processed by the buffer. To implement an error control mechanism, there is a need to simulate the congestion problem in the Conventional NoC.

The fault injection of buffer delay is performed, where delay rate in 3% of total packets is done. The fault injection is performed at the Switch and NI Buffers.

The switch and NI Buffers are connected with hardware modules and PU Cores. The external functional unit represents the other part of system than the fault injection part. Fig 1 specifies the fault injection at the buffer end to simulate the congestion problems in Conventional NoC.

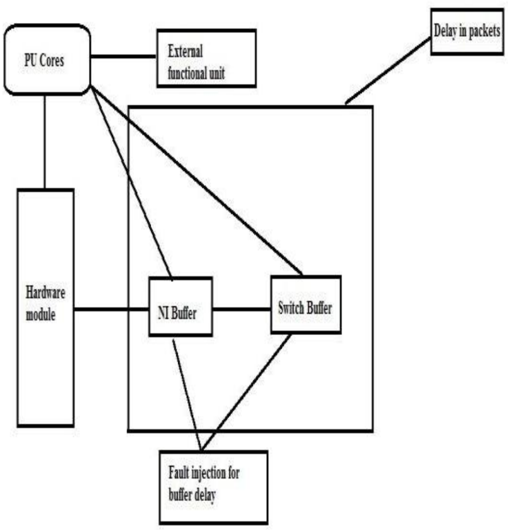

Fig. 1. Fault injection to simulate buffer delay in Conventional NoC

IV. DYNAMIC PACKET INJECTION MECHANISM TO CONTROL CONGESTION

The mechanism of dynamic packet injection is proposed in order to overcome the congestion problems caused in the On-Chip system. With the dynamic rate of packet injection, the throughput is made to increase with a reduced execution time. In the On-Chip communication system, the packet injection rate is based on the inter arrival message time. It is the time interval of cycles at which a packet is injected to the system. If the rate of packet injection is increased, the time interval between the generations of two packets also gets increased. This causes the workload of the system to get reduced. This means the Congestion gets reduced and throughput gets increased [5]. This principle is used for dynamic packet injection to improve the performance of congestion occurred Conventional NoC.

A cycle is a sequence of process executed to perform an instruction and this cycle normally has an execution time. On the basis of architecture of the On-chip-Network system, the execution time varies. In the congestion occurred Conventional NoC whenever the buffer delay occurs, the execution time of the particular cycle increases. When the execution time exceeds the normal rate, the packet injection rate is increased. On the increase of packet injection rate, the time interval between two packets gets increased. The workload is reduced and the succeeding packets also get transmitted Congestion free. This causes the throughput to get increased with a reduced latency. If packet injection rate is increased more, then throughput gets decreased significantly. So the packet injection is performed only at the cycles, which takes more time than the normal execution time. Dynamic packet injection mechanism is performed in Conventional NoC whenever the cycle's execution time exceeds normal rate. This causes workload to be reduced and the latency gets reduced. Throughput increases since the workload gets reduced. Fig 2 specifies the dynamic packet injection mechanism in Conventional NoC

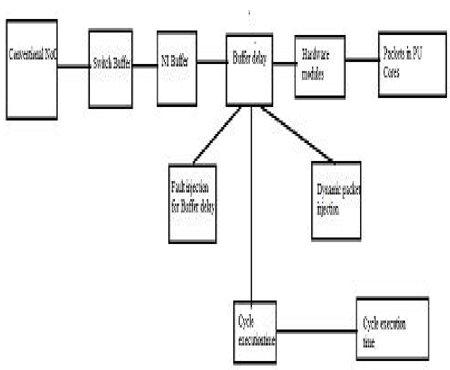

Fig. 2. Dynamic packet injection in Conventional NoC

V. EFFECTUATION OF DYNAMIC PACKET INJECTION IN CONGESTION OCCURRED CONVENTIONAL NOC

A. Fault injection at the Buffer en

The fault injection is done at the buffer end. An average of 3% delay occurs in the buffers of On-Chip systems. The fault injection is performed at the buffer end with 3% delay in packet transmission out of total packets transmitted. As a result of this fault injection, a real time On- Chip performance is obtained. The obtained system shows very high run time, since delay occurs at the buffer end. The throughput obtained is also low.

B. Logical flow for Fault injection

```
1 InputVCBuffer_addbuffer()
2 BEGIN
3 Input flit
4 Input vcID
5 Input curcycle
6 FOR cycle Starts at 0,
cycle<numberofcycles,Increment cycle
7 IF randomvalue() > 0.97,THEN
8 Perform delay in buffer_add_part_flit and
Process flit after delay time ends
9 ELSE
10 Execute flit without delay
11 ENDIF
12 ENDFOR
13 STOP
```

```
1 randomvalue()
2 Return math.random()
3 EXIT
```

The logical flow of implementation of packet loss recovery mechanism is shown in section IV. Fault injection is performed at the part (7,10) of logical flow in section IV, where a random number function is called to simulate the buffer delay in conventional NoC .

The same logical flow is followed in the InputVCBuffer_removebuffer(),OutputVCBuffer _addbuffer() and OutputVCBuffer_removebuffer().

C. Implementing Dynamic packet injection mechanism in Conventional NoC

The dynamic packet injection mechanism adapted to the Conventional NoC would change the packet injection rate when there is a delay occurring at the buffer end. The threshold of 1mS is kept at the controller part. If a delay is said to occur, it will cross 1 ms of execution time. On the basis of architecture, that is topology, the packet injection rate is changed. If the mesh topology or torus topology is followed, the packet injection rate is increased by 0.005.If the fat tree topology is followed, the packet injection rate is increased by 0.004.The packet injection rate is increased by 0.0055.The proposed rate of increase is done after several cycles of test for a better throughput and less execution time.

Table 1 depicts the Comparison case of Congestion occurred Conventional NoC when dynamic packet injection is not implemented and when dynamic packet injection is implemented. Here, in all four architectures the throughput is quiet low in Conventional NoC, when compared to the NoC with dynamic packet injection. Since the throughput increases, the packet injection rate is high resulting in more number of packets received. The key point to be noted is the execution time of Error controlled NoC. It is far less than the execution time of ordinary NoC. With the high throughput in less execution time this dynamic packet injection mechanism upgrades Conventional NoC. The implementation of dynamic packet injection has resulted in increase of Throughput of Conventional NoC .

D. Logical flow for Dynamic Packet injection

```
1 Dynamicpacketinjection()
2 Input cycle
3 Input numberofcycles
4 Input interarrivalrate
5 FOR cycle Starts at
0,cycle<numberofcycles,Increment cycle
6 IF cycletime>1 AND Topology=="MESH"
THEN
7 interarrivalrate=interarrivalrate+0.005
8 ELSE
9 IF cycletime>1 AND Topology=="TORUS"
THEN
```

```
10 interarrivalrate=interarrivalrate+0.005
11 ELSE
12 IF cycletime>1 AND Topology=="FATTREE" THEN
13 interarrivalrate=interarrivalrate+0.004
14 ELSE
15 IF cycletime>1 AND Topology=="EXFATTREE" THEN
16 interarrivalrate=interarrivalrate+0.0055
17 ELSE
18 interarrivalrate=interarrivalrate
19 ENDIF
20 ENDFOR
21 STOP
```

The dynamic packet injection mechanism is involved in the system only when the normal rate of cycle time exceeds 1 ms, Here the threshold value for cycle time is calculated on the basis of average of number of simulation cycles in a simulation run. Each topology has a different packet injection rate, which is based on its architecture. The logical flow of implementation of dynamic packet injection is depicted in (5,20) of logical flow in section D of effectuation.

In all architectures, the throughput is low for Congestion occurred NoC. It is evident in Fig 3 that shows throughput performance in congestion occurred NoC and Congestion controlled NoC In Congestion free NoC, all the architectures shows high throughput and Fat tree shows exceptional performance. The point to be noted is the increased throughput in all architecture of Congestion free NoC with the reduced running time.

After implementing the dynamic packet injection, the system is Congestion free and producing more number of packets than the uncontrolled Congestion system. The key thing is the execution time. Since the packet gets delay due to Congestion, the time taken to get executed by the system was very high. As, this problem is controlled the system produces more packets in a very low execution time. The desired system would be the one which process more number of packets in less execution time and it is done with the proposed system Fig4 and 5 shows the increased number of packets processed in a reduced execution time when implementing dynamic packet injection in congestion occurred Conventional NoC.

TABLE 1
CONVENTIONAL NOC WITH AND WITHOUT DYNAMIC PACKET INJECTION

CONVENTIONAL NoC WITHOUT DYNAMIC PACKET INJECTION				
PARAMETER & TOPOLOGY	THROUGHPUT [NET]	LATENCY [avg packet delay]	PACKET RECEIVED / RUN	EXECUTION TIME (ms)
MESH	0.1641	105.61	4293	15709
TORUS	0.1617	96.98	4228	11633
FAT TREE	0.1606	126.29	4252	8713
EXT FAT	0.1601	84.14	4153	9416
CONVENTIONAL NoC WITH DYNAMIC PACKET INJECTION				
PARAMETER & TOPOLOGY	THROUGHPUT [NET]	LATENCY [avg packet delay]	PACKET RECEIVED / RUN	EXECUTION TIME (ms)
MESH	0.1650	105.08	4209	9494
TORUS	0.1624	92.72	4283	11461
FAT TREE	0.1699	130.79	4343	7386
EXT FAT	0.1638	77.33	4317	8822

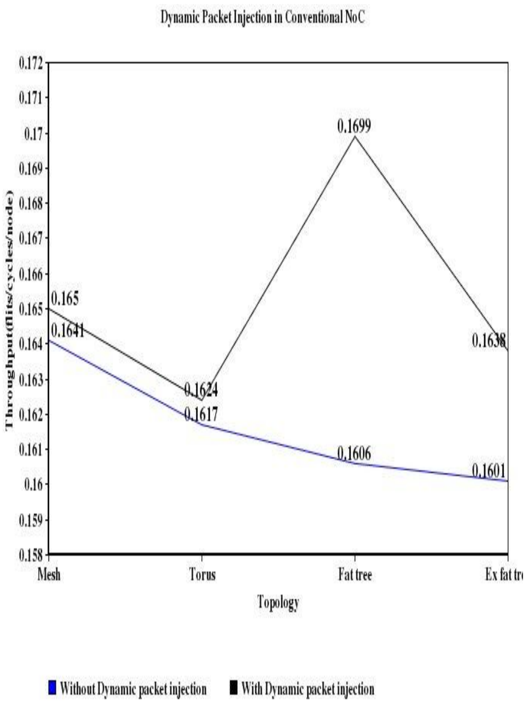

Fig 3. Throughput in Congestion occurred Conventional NoC and Congestion free Conventional NoC

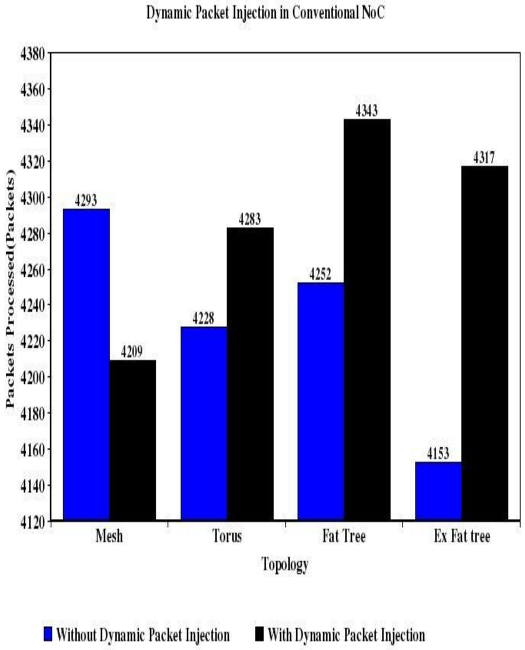

Fig 4. Packets processed in Congestion Occurred NoC and Congestion free NoC, dynamic packet injection.

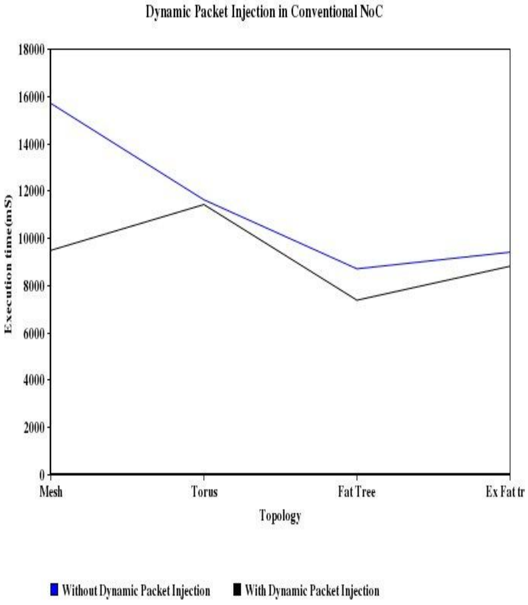

Fig 5.Execution time in Congestion Occurred NoC and Congestion free NoC, dynamic packet injection.

VI. CONCLUSION AND FUTURE WORK

Though the On-Chip Communication system is preferred methodology over the traditional systems, the congestion problems that occurs in the newer technology, makes the implementation of the system less practical. With the error control mechanism adapted for controlling the congestion problem in Conventional NoC, better results are obtained. The principle behind workload, throughput, latency is well effected with the system, and it causes the Congestion problem to get overcome with increasing throughput and decreased execution time. The latency is not showing constant measures for all architectures and a method must be implemented to overcome this problem.

REFERENCES

[1] Miloš Krstić, Eckhard Grass, Frank K. Gürkaynak, Pascal Vivet, "Globally Asynchronous, Locally Synchronous Circuits: Overview and Outlook," IEEE Design and Test of Computers, vol. 24, no. 5, pp. 430-441, Sept. 2007, doi:10.1109/MDT.2007.164

[2] Synthesis of application-specific on-chip networksby *Yan, Shan*, Ph.D., University of California, San Diego, 2009, 118 pages; AAT 3360452

[3] Carvalho, E.; Calazans, N.; Moraes, F.; , "Congestion-Aware Task Mapping in NoC-based MPSoCs with Dynamic Workload,". *IEEE Computer Society Annual Symposium on VLSI, 2007. ISVLSI '07*0,9-11March 2007doi: 10.1109/ISVLSI.2007.32

[4] Dumitras, T.; Marculescu, R.; , "On-chip stochastic communication [SoC applications]," *Design, Automation and Test in Europe Conference and Exhibition, 2003* , pp. 790- 795, 2003 doi: 10.1109/DATE.2003.1253703.

[5] Hossain, H.; Ahmed, M.; Al-Nayeem, A.; Islam, T.Z.; Akbar, M.; , "GpNoCsim - A General Purpose Simulator for Network-On-Chip," *International Conference on Information and Communication Technology, 2007. ICICT '07*. pp.254-257, 7-9 March 2007.

Bibliography:

Ezhumalai Periyathambi received the B.E degree in Computer science and Engineering from Madras University, Chennai, India in 1992 and received the Master of Technology (M.Tech) degree in Computer Science and Engineering from J N T University, Hyderabad, India in 2006. He is working as a Professor in Computer Science and Engineering in Rajalakshmi Engineering College. His research areas are reconfigurable architecture, multicore tech CAD –Algorithm for VLSI architecture, Theoretical Computer Science and mobile computing.

M.Sriram is Pursuing Bachelor of Engineering in the Stream of Computer Science in Rajalakshmi Engineering College, Anna University Chennai, Tamilnadu, India. His research areas are Network on Chips, Cryptography and Network Security.

A.Saranya is Pursuing Bachelor of Engineering in Computer Science in Rajalakshmi Engineering College, Anna University Chennai, Tamilnadu, India. Her research areas are Network on Chips, Open Source technologies and Virtualization.