ACHIEVING HIGH PERFORMANCE ON CHIP NETWORKS
WITH ROSHAQ AND FAULT ANALYSIS

Kousika.T1, Asst.prof.Navaneethan.S 2
1,2Department of Electronics and Communication Engineering,
Sri Shakthi Institute of Engineering and Technology, Coimbatore.

Abstract—On-chip routers generally have dedicated buffers to their input or output ports for storing packets temporarily if variance occurs on output physical channels. Buffers consume router area and it leads to increase in power budgets. While running in a traffic trace, however, all input ports of routers have incoming packets need to be simultaneously transferred. Therefore, in a network large number of buffer queues is empty and other queues are busy. This paper motivates us to design router architecture with RoShaQ (shared queues), router architecture that maximizes buffer usage and allows the sharing multiple buffer queues among the various input ports. When the network load becomes heavy RoShaQ, makes the buffers usage more efficient and achieves higher throughput. On the other hand, at light traffic load, our router achieves low latency and allowing packets to bypass the shared queues. Experimental results on a 65-nm CMOS standard-cell process show that RoShaQ has 17% less latency and 18% higher saturation throughput than a typical virtual channel (VC) router. Because of its higher performance, RoShaQ consumes 10% less energy per packet than VC router with the same buffer space capacity. RoShaq using round robin algorithm, shared queue has 30% lower latency than the virtual channel and it consumes 32% lower energy per packet.

Keywords— Network on chip, shared queues, router architecture, synthetic traffic

I. INTRODUCTION

In a SoC design which uses multi-million gates the designers should face various problems such as, heavy power consumption concerns and increase in testability challenges. Network on Chip (NoC) is a new paradigm to make the interconnections inside System on Chip (SoC) system. As SoCs [2] grow in complexity and size, on-chip communication is becoming increasingly important. Some of the optimization problems may occur such as communication delays and latencies across the chip start dominating computation delays.

Now a days a simple bus based communication architecture is not enough. So we enter into the NOC(Network On Chip),the network on chip can give some network based communication architecture and it also gives the design flow the communication design for SOC’s(System On Chip)should face some various number of unique challenges because of wide range of architectures the same file gives the new opportunities for optimization based on the application specific nature of the system. The corresponding NoC communication design flow that enables rapid design space exploration through design automation should support a wide range of applications then only it achieves the required productivity gain.

SYSTEM on chip(SOC) use the multicore design to speed up the system performance through increased parallelism and the power wall limits will the increase of the clock frequency [10],[5]. Networks on chip are easy to use and feasible and it supports a large number of processing elements and then point-to-point interconnect wires or shared buses [17]. By using 2D mesh network of the routers a multi core can communicate with the processor which is shown in Fig.2. Each router has five ports that connect to four neighboring routers and its local processor. A network interface (NI)
which is shown in Fig.1 is used to transforms the messages into packets and vice versa. It is located in between the processor and its router.

In a router architecture, there is input buffer for each input port for temporarily storing the packets if the output channel is busy. The WH (Wormhole) router has a buffer in single queue and a VC (Virtual Channel) routers has a buffer in a multiple queues as in parallel [18]. These buffers, consumes significant amount of area and power that can be more than 50% of the whole router [19]. Without Buffer in the routers architecture it saves amount of area [15], [13]; But, their performance becomes poor and the packet transmission rates are high. Because of having no buffers, previous router designs proposed to drop and retransmit packets or to deflect them once network variance occurs and that consumes higher energy per packet than a router with buffers [8]. in this router architecture its latency is high and throughput of this router is low.

Another technique is by sharing the buffer queues that allows utilizing idle buffers [11] or imitating an output buffer router to obtain higher throughput [14]. Our work differs from those router architecture the input packets at input ports are allowed to bypass shared queues so, it can achieve lower zero-load latency. In addition, the proposed router architecture has simple control circuitry making it dissipate less packet energy than VC routers and achieving higher throughput and the network load becomes heavy the queues share workload.
Other approach, such as dynamic voltage and frequency scaling [16],[9],[12] and globally asynchronous locally synchronous [1],[2],[7] can be used for reducing router power and energy. These techniques, are orthogonal to our work which focuses on architectural designs of on-chip routers. These techniques can be applied to this paper for further reducing the power consumption.

The main contributions of this paper are:

1) exploring and analyzing shared-queue router architectures that maximize buffer usage for maximize network throughput;
2) proposed router architecture if the load is low the input packets need not to be waited in the queues this reduces the zero-load packet latency;
3) Evaluating and comparing the proposed router with VC routers the latency, throughput, power, area and packet energy are reduced in both the uniform and synthetic traffic pattern.

This paper gives some analysis about the live lock and deadlock and it supports some routing algorithms and network topologies. We also add more experimental results and comparison on both synthetic traffic patterns and real application traces with the performance, power, and energy data are based on cycle-accurate simulations. The paper is organized in five sections. Section II describes the background and motivation about the wormhole and virtual channel. Section III presents the proposed router architecture with all its component details and section IV shows detecting and correcting the faults. Section V presents the experimental results, and finally section VI shows the conclusions.

II. BACKGROUND AND MOTIVATION

We first discuss about on-chip router architectures with brief description of their performance, and then we discuss about our new router architecture using shared queues.

A. wormhole Router Architectures
A worm hole router architecture with pipeline stage shown in Fig.4
Fig. 4 Four-Stage WH router architecture

1) when input port receives the packet the depends upon input buffer queue that is Queue write(QW)
2) The routing process completely based upon the look ahead routing computation(LRC) multiple packets may enter into the input queue some of the packets have same output port that is known as switch allocation(SA)
3) If the output wins the (switch allocation) SA, it will traversals across the crossbar. This step is called crossbar traversal or switch traversal(SA)
4) After that, it then traverses on the output link toward next router. This step is called link traversal (LT).

In this wormhole router architecture the line blocking problem may occur. These problems are solved by virtual channel router architecture [18].

B. Virtual channel Router Architectures

VC router architecture avoids line blocking problems, to solve these virtual channel use multiple queues to avoid congestion of the packet. The queues are designed parallel to avoid line blocking problem which is shown in Fig.5.

![Virtual channel router architecture diagram]

Fig.5 Five-Stage Virtual channel router architecture

the line blocking problem are occurred at the head of the FIFO in the current router state that is not able to pass the packets to the next router .at this stage the output ports are empty .in these cases virtual channel flow control is used to decouples the buffer. So the congestion of the packet is reduced .the network throughput is improved. The virtual channel router has higher throughput the wormhole router with same number of input the input port has multiple VC queues, each packet has to choose a VC of its next routers input port before arbitrating for output switch. Granting an output VC for a VC allocation is performed in parallel with the LRC. hence the router has five stages shown in fig().therefore although a VC router achieves higher saturation throughput than a WH router while having the same number of buffer entries per input port.

III ROUTER ARCHITECTURE WITH SHARED QUEUES (ROSHAQ)

A. Initial Idea
Fig. 6 Sharing buffer queues in a router

To increase the queue usage the input ports should share the queues which is shown in Fig.6, the input ports receives the packet and passed through any shared queues. This architecture has some drawbacks that is:

- no buffer at input ports
- it should handle many external requests

Because of this drawback the latency is high.

Fig. 7 Input port contains only one queue architecture

To reduce the latency the input port is dedicated to buffer a queue which is shown in Fig.7. It is similar to wormhole router. By using buffers at input ports that reduce the latency. Here the packets are buffered before being sent to the output ports .but it is unnecessary when the net work load is low.

B. Shared Queue Router Architecture

The shared queue router micro architecture which is shown in Fig.8, here the input ports receives the packets and calculates the output port by using look ahead routing table .each input port is dedicated to the buffer queues. In this figure the SQA and OPA is an important block.

Fig. 8 Shared Queue Router Architecture

Shared queue allocator: SQA is a shared queue allocator which is used to split up the work load to the empty buffers. If the network load is heavy the input ports send network load is heavy the input port send the request to SQA.if SQA accepts the request from the input port it shares work.
output port allocator: OPA is output port allocator. If the network load is low the input port sends the request to OPA. If it accepts the request the input packets need not to be waited in a queue. It directly bypasses the shared queues.

C. ROSHAQ DATAPATH PIPELINE:

In the Fig.9 which shows the data path pipeline in the 1st cycle the packet is passed to the input queue and automatically in the second cycle it performs the three operation that is: LRC, SQA, and OPA. When there is a low network load the OPA accepts the grant request from input ports because of low traffic.

![Fig.9 Light Load Pipeline Characteristics](image)

In the second cycle which is shown in Fig.10 there is heavy network load the OPA does not accepts the request from the input ports but in these cases SQA accepts the request from the input ports for this case we are using shared queues to shares the work load, for sharing the work load we can improve the throughput.

![Fig.10 Heavy Load Pipeline Characteristics](image)

D. Shared Queues Properties

A. Shared queue router architecture is deadlock-free. When network load is heavy, the shared queue allocator shares the work load and if the network load is low the output port allocator bypass the shared queues in these cases the shared queues router architecture will act as WH router architecture so the network load is a deadlock free.

B. Shared queue router architecture is live lock-free. The shared queue allocator(SQA) and output port allocator(OPA) uses round robin manner. So each packet from the input port reaches their destination without any congestion. So the network is live lock free.

C. Shared queue router architecture supports any adaptive routing algorithm: the shared queue router architecture is similar to WH router architecture. The algorithms which is used in WH is also suits for shared queues. Here the output ports depends upon input ports only not depends on shared queues.

D. Shared queue router architecture can be used for any network topology. We can hide the design details inside the shared queues at that time only one buffer at input ports is visible so here we can change the hide details of the shared queues according to the network topology.

E. Latency and Throughput
In a uniform random traffic pattern the source node choose its destination randomly with uniform distribution. In other patterns the destination has been decided according to the location. The performance of router has been analyzed based on the simulation. The processor consumes the packet according to the packet length. The operation are performed according to routing algorithm. Here we use the same XY dimension and routing algorithm for all routers in the processor. Packet latency is measured according to the time consumption from source to destination. The average packet latency of networks corresponding to six router configurations over uniform random traffic is shown in Fig. 12. As shown, even having the same number of buffer entries, Virtual channel has higher saturation throughput. Increasing the number of crossbar input ports improves throughput significantly. Shared queues achieve a saturation throughput of 0.30 flits/cycle which is 5% higher than Virtual channel. Shared queues achieve 0.40 flits/cycle throughput that is 4% higher than Virtual channel. More importantly, the shared queues have zero-load latency and is similar to a WH router that is 15% lower than all VC routers with and without a full-degree crossbar.

Therefore, packets in shared queue routers often bypass shared queues to achieve both lower latency and higher throughput than both virtual channel routers. Such that the shared queue have lower latency than the virtual channel at the same time the shared queue router architecture is similar to the wormhole router architecture because in a wormhole router architecture have lower latency than the virtual channel. So ROSHAQ is similar to WH router architecture.

F. Real Application Communication Traffic:

Many DSP and embedded applications can be represented by a communication task graph where each task can be mapped onto one or multiple processing units (processors, accelerators, or memory modules) [3]. The task graph of a video object plan decoder (VOPD) application [6] VOPAD is a intertask application.

In this method, we transform the required bandwidth on each intertask connection into the injection rate of the corresponding sending task. A task which requires large sending bandwidth also has large injection rate, and vice versa. Let \( bw_i \) and \( bw_j \) be the required bandwidth of tasks \( T_i \) and \( T_j \) to other tasks in the communication graph, then injection rates of tasks \( T_i \) and \( T_j \) on the corresponding links follow the equation \( f_{ir_i}/f_{ir_j} = bw_i/bw_j \). Therefore, if given an injection rate of any task, we can easily calculate injection rates for all other tasks on all links in the graph.

For evaluation of these embedded applications, we fix the injection rates of all tasks, then application running latency is measured after a total of one million packets are successfully transferred. For each application, we assume that the most busy task spends 50% times for execution and 50% times for communication which means it aggressively executes one cycle and then sends one output to the downstream task in next cycle, repeatedly. With this assumption, the most busy task in each application has an injection rate of 0.5; the injection rates of other tasks are computed according to their required bandwidth given in the graph using the method described above.

For clear comparison, we normalize the latency of each application running on different routers to the latency when running on the typical VC router which is shown in Fig. 11. As shown, RoShaQ has lower latency than both VC routers in all ten applications. On average, RoShaQ has 26% and 12% less latency than VC and VC-fullXbar routers, respectively.

F. Power, Area, and Energy

Here we use two router models that is virtual channel and shared queues are synthesized by using verilog. The power, area and energy of these routers are compared based on the simulation result. According to the comparisons we can analyze the buffer usage, power consumption, area and energy...
of both router models. Which is shown in Table 1.

<table>
<thead>
<tr>
<th>CONTENET</th>
<th>VIRTUAL CHANNEL</th>
<th>SHARED QUEUES</th>
<th>VIRTUAL CHANNEL Vs SHARED QUEUES</th>
</tr>
</thead>
<tbody>
<tr>
<td>POWER</td>
<td>58.26</td>
<td>53.08</td>
<td>1%</td>
</tr>
<tr>
<td>AREA</td>
<td>0.091</td>
<td>0.058</td>
<td>1.7%</td>
</tr>
</tbody>
</table>

**Table.1 Comparisons result of VC and RoShaQ**

In this figure it indicates the buffer usage, crossbar, allocator, and others where in virtual channel buffers are occupy 50% area and consume 75% power of whole router architecture and crossbar occupies 8% only. The shared queues have two crossbar so its crossbar are 50% larger and consumes 30% higher power than the virtual channel. For these case we use shared queue allocator and output port allocator so the shared queue consumes 8% lesser power and area than the virtual channel. But in the virtual channel there are no shared queues so at the heavy traffic it does not shares their workload to other empty buffers so the energy consumption of virtual channel is high than the shared queue router architecture. We can calculate the area, power and energy by using the mathematical expressions

The power consumption (P) of the circuit.  \[ P = \alpha C V^2 f \]

Where  
\( \alpha \)=> circuit switching activity factor  
C=> circuit capacitance  
V=> supply voltage  
f=> clock frequency  

if the supply voltage and clock frequency of the circuit are same then it consumes the low power even in the large capacitance. The total energy consumed by a router:

\[ E_r = \sum_{all \ i} (n_{ri} P_i T_{clk}) \]

\( P_i = \text{synthesis power} \)  
\( n_{ri} = \text{no: of cycles in components I of router } r \)  
\( T_{clk} = \text{clock period} \)

The average packet energy spent on each router in the network:

\[ E_p = \frac{1}{(N_p N_r)} \sum_{all \ r} E_r = \frac{T_{clk}}{(N_p N_r)} \sum_{all \ r} \sum_{all \ i} E_r P_i \]

\( N_r = \text{number of routers} \)  
\( N_p = \text{total number of consumed packets} \)

From these expressions we can analyze the power, area and energy.

**IV.FAULT ANALYSIS BY USING VITERBI ALGORITHM:**

**A.FAULT ANALYSIS**

There are two types of errors are occurred that is
1. Entire packet loss  
2. Single (or) Multibit error
**Entire packet loss error:**
If we send many packets to the router, the congestion may be occurred. If the buffers in the routers are filled, the processor will not be able to pass the packets to their destination. So the packets may be lost. If the packets are lost, the destination does not give the acknowledgement signal. In that case, the processor understands that the packets are lost before reaching the destination. So it retransmits the packets, still getting the acknowledgement signal.

**Single or multi bit error**
The single or multi bit error are occurred inside the packet that is known as bit error. Inside the packet, either one or more bits will get corrupted, which is known as burst error. Burst error indicates that a few number of bits inside the packet are corrupted. These errors have been handled by 2 methods:
1. Error detection
2. Error correction

**Error detection and error correction**
Where error detection is used to detect the error. But it detects only which packets are corrupted, it does not indicate which bit is corrupted inside the packet. When signals are transmitted, noise attenuation, distortion may occur. Some of the data in that signal are corrupted. So it may give some error at the receiving side. The main reason for error occurrence is a frame size and bit error.

**B. Viterbi algorithm**
The most commonly employed decoding technique that can be implemented using either software or digital hardware.
VA uses the trellis diagram (Fig.12) and can theoretically perform maximum likelihood decoding. It finds the most likely path by means of a suitable distance metric between the received sequence and all the trellis paths.

**Fig.11 Trellis diagram**

**BMU:** BM are computed from introduced input data
**ACSU:** PMs of all states are updated according to equation (1)
**SMU:** The stored decisions are employed in the SMU to build a unique decoded output

\[
PM[i][t+1] = \min ( PM[k][t] + BM([k][i])(t) ) \quad (1)
\]

**PM[k][t]:** Path metric corresponding to state k at instant t
**BM([k][i])(t):** Branch metric of the transition from state k at t to state i at t+1
Fig. 12 basic computation units of Viterbi

In this fig.13 represents the BMU the branch metric units, ACS And SMU (survivor path memory unit) the Branch Metrics is a deep linking and business analytical platform offering developer tools for large and independent mobile application companies Branch Metrics is credited with both developing the original core technology and coining the term. Mobile application developers can use these tools in conjunction with Branch's drop-in SDK to personalize the install experience for each new user referred or acquired through digital advertising. The Branch Metrics platform dashboard tracks and displays such data as user behavior, K-factor, campaign performance, and LTVs. The ACSUs and SPU are known to be critical parts in a hardware implementation. In particular, the algorithm used for the SPU affects the overall memory requirement and latency of the decoder, two important aspects in today's communication systems. SPU algorithms rely on the fact that the survivor paths are expected to have merged with sufficiently high probability after a certain decoding depth.

V. SIMULATION RESULTS

Several simulations have been performed in ModelSim6.5e software. Here we have three simulation results are shown in fig() that is virtual channel and shared queues outputs and fault analysis. In the virtual channel workload are not shared because it does not have the (SQA), so the latency is high. The virtual channel needs many grant signals that is it requires grant for each bits. But in the shared queue the workload is shared to the empty buffers and if there is a low load the packets need not to be waited in the queue. It requires only one grant signal. The latency of shared queue is low and efficiency is high and fault analysis by using viterbi algorithm. It is used to detect and correct the errors automatically. It can detect the multibit errors inside the packets.
Fig. 13 and 14, show a comparison of how the works are shared on both the existing and proposed model is being carried on.

Fig. 15 show a automatic bit error detection and corrections

VI. CONCLUSION

This project work presents a modelsim simulated result of virtual channel and shared queue router architecture. The shared queue allocations can provide performance improvement similar to wormhole router configuration for on chip network but much less area overheaded. The performance of interconnection networks was improved by organizing the buffers associated with each network channel into shared queues rather than a single FIFO queue. Associating several lanes with each physical channel decouples the allocation of physical channel to flits that leads to 8% lower power and 17% lesser area.

REFERENCES