ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 2, Issue 4, April 2013

Design of Reconfigure Virtual Channel Regulator for BiNoC Router

Mr. Ashish Khodwe , Prof. C.N. Bhoyar

Abstract— **Network-on-Chip (NOC) has been proposed as an attractive alternative to traditional dedicated wire to achieve high performance and modularity. Power and Area efficiency is the most important concern in NOC design. This paper introduces a novel unified buffer structure, called the Dynamic Reconfigure Virtual Channel Regulator, which dynamically allocates Virtual Channels (VC) and buffer resources according to network traffic conditions. It maximizes throughput by dispensing a variable number of VCs on demand. Dynamic Reconfigure Virtual Channels ability to provide similar performance with half the buffer size of a generic router is of paramount importance. This paper presents a VHDL based cycle accurate register transfer level model for evaluating the, Area of Dynamically self Reconfigurable BiNoC architectures. We implemented a parameterized register transfer level design of the BiNoC architecture elements. The design is parameterized on (i) size of packets, (ii) length and width of physical links, (iii) number, and depth of virtual channels, and (iv) switching technique. The paper discusses in detail the architecture and characterization of the various BiNoC components. The characterized values were integrated into the VHDL based RTL design to build the cycle accurate performance model.**

***Index Terms*— Interconnection networks, multiprocessor, systems-on-chip (MPSoCs), networks-on-chip (NoCs), on-chip communication, reconfigurable architectures**

I. INTRODUCTION

The recent technology advances in deep sub-micron technology has enabled higher integration of functional modules within a single chip. This state-of-art technology introduced a new paradigm in chip design methodology and many recent high performance chips are developed based on such multi-core concepts [1]. While this has proven beneficial in terms of overall performance, there are still many challenges posed by this new technique mainly due to the reduced feature size in deep sub-micron technologies. Particularly, the interconnection between functional modules (IP blocks) becomes problematic since on-chip traffic increases dramatically and the traffic behavior becomes more complicated as the number of IP blocks increases. As a result, the on-chip interconnects turn into a critical bottleneck in terms of performance and power consumption. A recent study showed that up to 77% of the overall delay in a SoC chip can come from the interconnect in the 65nm regime [2]. Traditional on-chip interconnects have been implemented mostly using shared bus architecture but due to its limited scalability, it becomes less suitable in meeting the requirements of the future multi-core environment. As an alternative, Network-on-Chip (NoC) architectures have been recently introduced, where a packet-based network infrastructure provides interconnection among IP blocks, allowing concurrent transfer in the network [3, 4]. However, NoCs suffer from their inherent constraints such as limited area and power budget. Such limitations also bound the flexibility in network configuration such as routing algorithms, buffer size, and arbitration logic. Many researchers have focused on several aspects of the NoCs proposing efficient router pipeline design [5-7], fault-tolerant techniques [8, 9], deadlock-free routing algorithms [10-12], and thermal-aware low-power designs [13-15], etc.

State-of-the-art NoC designs often use packet-switched routers to support high bandwidth traffic. Under this model, it often takes multiple hops for messages to reach their destinations, and the energy/delay associated with packets traversing through routers is the dominating factor. There have been several proposals for reducing the performance penalty, such as router bypassing [16]–[18] and enhancing router pipeline design [5]–[19]. There also exists a large body of work on reducing router energy consumption, which corresponds to a large portion of NoC energy [20], [21].

This paper presents a VHDL based cycle accurate register transfer level model for evaluating the dynamic, Area and leakage power consumption of Dynamically self Reconfigurable BiNoC architectures. We implemented a parameterized register transfer level design of the BiNoC architecture elements. The design is parameterized on (i) size of packets, (ii) length and width of physical links, (iii) number, and depth of virtual channels, and (iv) switching technique. The paper discusses in detail the architecture and characterization of the various BiNoC components. The characterized values were integrated into the VHDL based RTL design to build the cycle accurate performance model. The rest of this paper is organized as follows. In Section II, we will discuss some of the background materials for NoC architecture and prior related research. In section III, Motivation. Further section IV, Baseline of NoC Router. a bidirectional network on-chip (BiNoC) architecture will be given in Section V. further section VI, router pipeline. in section VII , Overview of a Virtual-Channel Router.

Finally, in Section VIII, experiment results comparing the performance of the proposed BiNoC architecture against the

Manuscript received April 2013

***Mr. Ashish Khodwe**, Department of Electronics Priyadarshini college of Engineering ,RTMNU,Nagpur,India.*
***Prof.C.N.Bhoyar**, Department of Electronic Priyadarshini College of Engineering, RTMNU, Nagpur, India*

conventional NoC architecture are provided. In last section, brief statements conclude this paper.

II. RELATED WORK

Importance of buffer size and organization

Buffer size and management are directly linked to the flow control policy employed by the network; flow control, in turn, affects network performance and resource utilization. Whereas an efficient flow control policy enables a network to reach 80% of its theoretical capacity, a poorly implemented policy would result in a meager 30% [22]. Wormhole flow control [23] was introduced to improve performance through finer granularity buffer and channel control at the flit level instead of the packet level (a flit is the smallest unit of flow control; one packet is composed of a number of flits). This technique relaxes the constraints on buffer size at each router, allowing for a more efficient use of storage space than store-and-forward and virtual cut through [24] switching. However, the channel capacity is still poorly utilized; while the buffers are allocated at the flit level, physical paths are still allocated at the packet level. Hence, a blocked packet can impede the progress of other packets waiting in line and may also cause multi-node link blocking (a direct consequence of the fact that the flits of a single packet are distributed across several nodes in wormhole routers). To remedy this predicament, Virtual Channel (VC) flow control [25] assigns multiple virtual paths (each with its own associated buffer queue) to the same physical channel. It has been shown that VC routers can increase throughput by up to 40% over wormhole routers without VCs [22].

As a side bonus, virtual channels can also help with deadlock avoidance [26]. The work in this paper assumes, without loss of generality, the use of VC-based flow control, which suits the low buffer requirements of NoC routers. The way virtual channels – and hence buffers – are organized within a router is also instrumental in optimizing performance. The number of VCs per physical channel and the VC depth are two parameters that form an elaborate interplay between buffer utilization, throughput and latency. Researchers in the macro-network field have identified the decisive role of virtual channel organization in overall system performance [26, 27]. Detailed studies of the relation between virtual channels and network latency indicate that for low traffic intensity, a small number of VCs can suffice. In high traffic rates, however, increasing the number of VCs is a more effective way of improving performance than simply increasing the buffer depth [28]. Under light network traffic, the number of packets traveling through a router is small enough to be accommodated by a limited number of VCs; increasing the number of VCs yields no tangible benefits. Under high traffic, many packets are contenting for router resources; increasing VC depth will not alleviate this contention because of Head-of-Line (HoL) blocking. Increasing the number of VCs, though, will allow more packets to share the physical channels. This dichotomy in VC organization implies that routers with fixed buffer structures will either be underutilized or will underperform under certain traffic conditions. This objective function can only be achieved through the use of efficient management techniques which optimize buffer utilization. Since size and organization are design-time decisions, they cannot be dynamically changed during operation based on observed traffic patterns. However, the use of a carefully designed buffer controller can significantly affect the efficiency of storing and forwarding of the flits. Therefore, the throughput of a switch can be maximized through dynamic and real-time throttling of buffer resources.

III. MOTIVATION

A. Virtual Channel

The design of a virtual channel (VC) is another important aspect of NOC. A virtual channel splits a single channel into two channels, virtually providing two paths for the packets to be routed. There can be two to eight virtual channels. The use of VCs reduces the network latency at the expense of area, power consumption, and production cost of the NOC implementation. However, there are various other added advantages offered by VCs.

B. *Network deadlock/livelock*:

Since VCs provide more than one output path per channel there is a lesser probability that the network will suffer from a deadlock; the network livelock probability is eliminated.

C. *Performance improvement*:

A packet/flit waiting to be transmitted from an input/output port of a router/switch will have to wait if that port of the router/switch is busy. However, VCs can provide another virtual path for the packets to be transmitted through that route, thereby improving the performance of the network.

D. *Supporting guaranteed traffic*:

A VC may be reserved for the higher priority traffic, thereby guaranteeing the low latency for high priority data flits [29], [30].

E. *Reduced wire cost*:

In today's technology the wire costs are almost the same as that of the gates. It is likely that in the future the cost of wires will dominate. Thus, it is important to use the wires effectively, to reduce the cost of a system. A virtual channel provides an alternative path for data traffic, thus it uses the wires more effectively for data transmission. Therefore, we can reduce the wire width on a system (number of parallel wires for data transmission). For example, we may choose to use 32 bits instead of 64 bits. Therefore, the cost of the wires and the system will be reduced.

Bjerregaard and Sparso have proposed the design and implementation of a virtual channel router using asynchronous circuit techniques [29], [30].

F. *Buffer Implementation*

A higher buffer capacity and a larger number of virtual channels in the buffer will reduce network contention, thereby reducing latency. However, buffers are area hungry, and their use needs to be carefully studied and optimized. Zimmer et al. and Bolotin et al. proposed a simple implementation of a

buffer architecture for NOC [32],[33]. Zimmer et al. implemented buffers using 0.18 µm technology to estimate the cost and area of buffers needed for NOC. The Proteo implementation of a buffer architecture has been described in [34]. Gupta et al. studied the trade-off between buffer size and channel bandwidth to secure constant latency. They concluded that increasing the channel bandwidth is preferable to reducing the latency in NOC.

IV. BASELINE NOC ROUTER

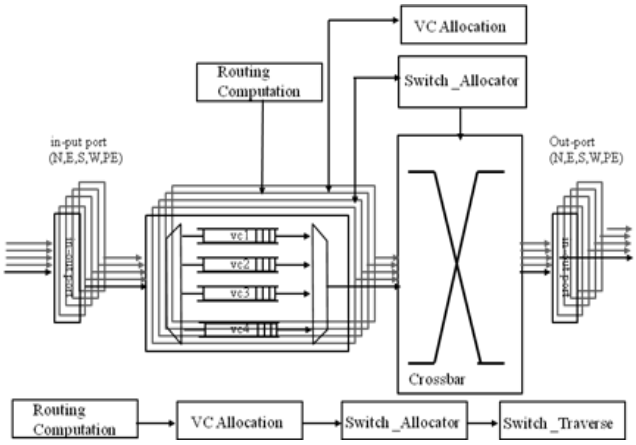


Fig.1 Typical four stage pipelined router design based on VC flow control.

A typical NoC system consists of processing elements (PEs) ,network interfaces (NIs), routers and channels. The router further contains switch and buffers. Buffers consume the largest fraction of dynamic and leakage power of the NoC node (router + link) [4] [3]. Storing a packet in buffer consumes far more power as compared to its transmission [35]. Thus, increasing the utilization of buffers and reduction in number and size of buffers with efficient autonomic control enhances the system performance and reduces the area and power consumption. Wormhole flow control has been proposed to reduce the buffer requirements and enhance the system throughput. But on other hand, one packet may occupy several intermediate switches at the same time. In typical NoC architectures, when a packet occupies a buffer for a channel, the physical channel cannot be used by other channels, even when the original message is blocked [25]. This introduces the problem of deadlock and livelock in wormhole scheme. Virtual Channels (VCs) are used to avoid deadlock and livelock. Fig.1 Typical four stage pipelined router design based on VC flow control [37]. VC flow control exploits an array of buffers at each input port. By allocating different packets to each of these buffers, flits from multiple packets may be sent in an interleaved manner over a single physical channel. This improves the throughput and reduces the average packet latency by allowing blocked packets to be bypassed. By inserting the VC buffers, we increase the physical channel utilization but utilization of inserted VC buffers is not considered.

Router architecture illustrated in Fig.1 The router has P input and P output channels/ports. In most implementations, P=5; four inputs from the four cardinal directions (North, East, South and West) and one from the local Processing Element (PE). The Routing Computation unit, RC, is responsible for directing the header flit of an incoming packet to the appropriate output Physical Channel/port (PC) and dictating valid Virtual Channels (VC) within the selected PC. The routing is done based on destination information present in each header flit, and can be deterministic or adaptive. The Virtual channel Allocation unit (VA) arbitrates amongst all packets requesting access to

the same VCs and decides on winners. The Switch Allocation unit (SA) arbitrates amongst all VCs requesting access to the crossbar and grants permission to the winning flits. The winners are then able to traverse the crossbar and are placed on the respective output links. So far, as a result of scarce area and power resources and ultra-low latency requirements, on-chip routers have relied on very simple buffer structures. In the case of virtual channel-based NoC routers, these structures consist of a specified number of FIFO buffers per input port, with each FIFO corresponding to a virtual channel. This is illustrated in Fig.1

Hence, each input port of an NoC router has v virtual channels, each of which has a dedicated k-flit FIFO buffer. Current on-chip routers have small buffers to minimize their overhead; v and k are usually much smaller than in macro networks [35]. The necessity for very low latency dictates the use of a parallel FIFO implementation

V. BINOC ARCHITECTURE

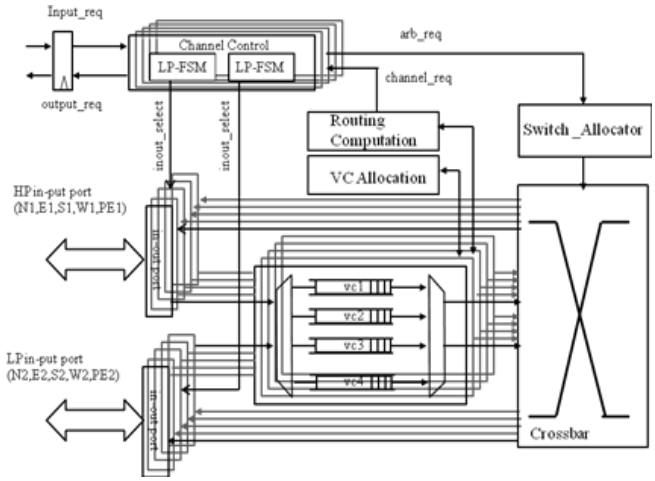


Fig.2 Modified four-stage pipelined router architecture for our proposed BiNoC router with VC flow-control technique.

Fig.1 shows the microarchitecture of A bidirectional channel network-on-chip (BiNoC) virtual channel (VC) router is modeled [43]. This section to enhance the performance of on-chip communication. In a BiNoC, each communication channel allows itself to be dynamically reconfigured to transmit flits in either direction. This added flexibility promises better bandwidth utilization, lower packet delivery latency, and higher packet consumption rate. Novel on-chip router architecture is developed to support dynamic self-reconfiguration of the bidirectional traffic flow. The flow direction at each channel is controlled by (CDC) a channel-direction-control protocol [43]. Implemented with a pair of finite state machines. This channel-direction-control protocol is shown to be of high performance, free of deadlock, and free of starvation.

VI. Router Pipeline

A generic on-chip router consists of multiple atomic pipeline stages shown in fig.3; Routing Computation (RC), Virtual Channel Allocation (VA), Switch Allocation (SA), and Switch Traversal (ST) as shown in Figure 2. Many researchers have proposed router architectures that reduce the router pipelines along the critical path by parallelizing some of these stages, thereby achieving low latency routers [36, 37, 38].The BiNoC architecture assumed in this paper is a four stage pipelined router which allows the RC, VA, and SA stages to execute in parallel.

In such designs, each packet arriving at an ingress port is immediately queued in a VC buffer, and forwarded via five steps: route computation (RC), virtual channel allocation (VCA), switch allocation (SA), and switch traversal (ST), sometimes implemented as separate pipeline stages for efficiency. All flits in a packet are forwarded contiguously, so the first two stages (RC and VCA) only perform computation for the head flit of each packet, returning cached results for the remaining flits.

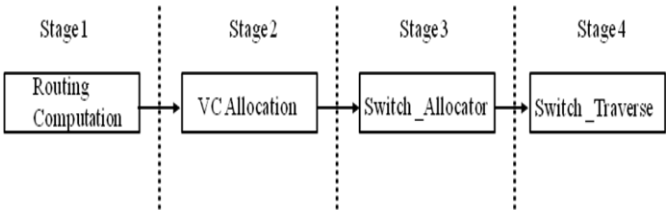


Fig.3 Typical four stage pipelined router design based on VC flow control.

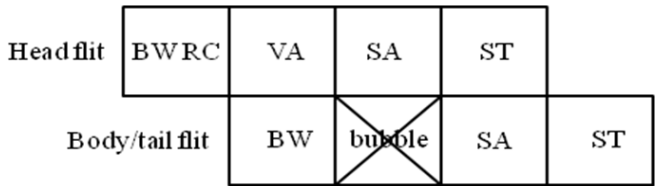


Fig4. Router Pipeline

On-chip designs need to adhere to tight budgets and low router footprints. Every VC has its own private buffer and its size can be specified at runtime. A head flit on arriving at an input port, first gets decoded and gets buffered according to its input VC in the buffer write (BW) pipeline stage shown in fig 4. Every VC has its own private buffer. In the same cycle, a request is sent to the route computation unit (RC) simultaneously, and the output port for this packet is calculated. The header then arbitrates for a VC corresponding to its output port in the VC allocation (VA) stage. Upon successful allocation of an output VC, it process to the switch allocation (SA) stage where it arbitrates for the switch input and output ports. On winning the switch, the flit moves to the switch traversal (ST) stage, where it traverses the crossbar. This is followed by link traversal (LT) to travel to the next node. Body and tail flits follow a similar pipeline except that they do not go through RC and VA stages, instead inheriting the VC allocated by the head flit. The tail flit on leaving the router, deallocates the VC reserved by the packet.

Keeping in mind on-chip area and energy considerations, single-ported buffers and a single shared port into the crossbar from each input were designed. Separable VC and switch allocators as proposed in [3] were modeled. This was done because these designs are fast and of low complexity, while still providing reasonable throughput, making them suitable for the high clock frequencies and tight area budgets of on-chip networks. The individual allocators are round-robin in nature.

VII. Overview of a Virtual-Channel Router

Fig.2 illustrates the major components of a BiNoC virtual-channel router. The router has P input ports and Output ports, supporting V virtual-channels (VCs) per port. Virtual-channel flow control exploits an array of buffers at each input port. By allocating different packets to each of these buffers, flits from multiple packets may be sent in an interleaved manner over a single physical channel. This improves both throughput and latency by allowing blocked packets to be bypassed. The basic steps undertaken by a virtual-channel router are enumerated below:

a) *Routing*

The first flit of a new packet arrives at the router. The routing field is examined and a set of valid output virtual-channels upon which the packet can be routed is produced. The number of output VCs produced by the routing logic will depend on the routing function. Possibilities range from a single output VC to a number of different VCs potentially at different physical channels (i.e. adaptive routing). The selection of an output VC can also be influenced by the class of the packet to be routed. Packets from particular classes will often be restricted to travelling on a subset of virtual-channels to avoid message-dependent deadlock. A common practise is to provide separate request and reply virtual-networks.

b) *Virtual-Channel Allocation*

An attempt is made to allocate an unused VC to the new packet. A request is made for one of the virtual-channels returned by the routing function. Allocation involves arbitrating between all those packets requesting the same output VC.

c) *Switch Allocation*

Each packet maintains state indicating the availability of buffer space at their assigned output VC. When flits are waiting to be sent, and buffer space is available, an input VC will request access to the necessary output channel via the router's crossbar. On each cycle the switch allocation logic matches these requests to output ports, generating the required crossbar control signals.

d) *Crossbar Traversal*

Flits that have been granted passage on the crossbar are passed to the appropriate output channel. The following sections describe in more detail each of the router's components.

e) *Input Buffer and Bypass*

Each new incoming flit is stored in the VC buffer designated by its VC identifier. This identifier is appended to every flit in the previous router stage. If the VC buffer is empty and the flit is able to access the crossbar immediately, a bypass path is required to expedite its journey.

f) *Routing Logic*

In order for virtual-channel and switch allocation to take place the routing function must first be evaluated to determine which virtual-channel(s) at which output port(s) the packet may request. To ensure that this computation does not lie on the router's critical path, the computation may be performed in the previous router in preparation for use in the next. The idea that the route may be calculated one step by the SGI routing chip [38] and is known as look-ahead routing.

g) *Virtual-Channel Allocation*

Peh and Dally detail the complexity of both virtual-channel (VC) allocation and switch-allocation logic in [5]. The following two sections provide a brief overview of these schemes. The complexity of VC allocation is dependent on the range of the routing function. In the simplest case, where the routing function returns a single VC, the allocation process simply consists of a single arbiter for each output VC. As any of the input VCs may request any output VC, each arbiter must support P x V inputs. If the router function returns multiple output VCs restricted to a single physical channel, an additional arbitration stage is required to reduce the number of requests from each input VC to one. The winning request at each virtual channel buffer then proceeds to the second stage as described above. The complexity of such a scheme is illustrated in Figure 3. The routing function determines the output port and VCs that may be requested prior to VC allocation. A VC which is free to be allocated is then selected by the first stage of arbitration. The result of this first stage of arbitration is a request for a single VC at a particular output port. This request is subsequently sent to the appropriate second stage arbiter. While this scheme does not guarantee to allocate all free output VCs to potential waiting input VCs in a single cycle, there is no performance penalty as only one flit may be sent per cycle on an output channel. In the most general case where the routing channel may return any of P x V VCs, the number of inputs to the first stage of arbiters must now be increased from V to P x V illustrated in fig 5 a). In this case some performance degradation may be expected as the scheme makes little effort to perform a good matching of requests to free output VCs.

h) *Switch Allocation*

Individual flits arbitrate for access to physical channels via the crossbar on each cycle. Arbitration may be performed in two stages [5]. The first reflects the sharing of a single crossbar port by V input virtual-channels, this requires a V-input arbiter for each input port. The second stage must arbitrate between winning requests from each input port (P inputs) for each output channel. The scheme is illustrated in Figure 5 b). The request for a particular output port is routed from the VC which wins the first stage of arbitration. In order to improve fairness, the state of the V-input the second stage of arbitration.We assumes this organization wherever multiple stages of arbitration are present. This switch allocator organization may reduce the number of requests for different output ports in the first stage of arbitration, resulting in some wasted switch bandwidth.

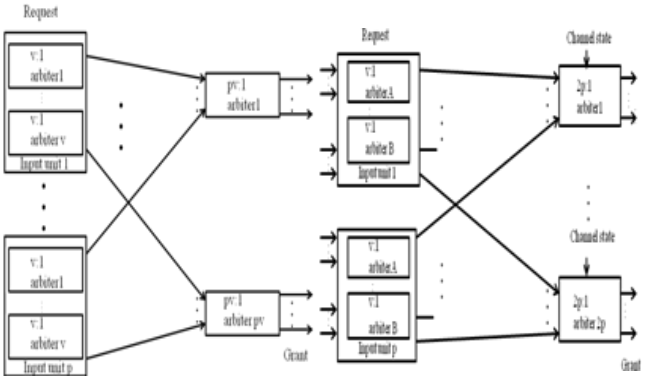


Fig. 5 (a) VC allocator in a BiNoC router. (b) SA in a BiNoC router.

i) *Speculative Switch Arbitration*

Virtual-channel flow control as discussed performs VC allocation and switch allocation sequentially. This guarantees that only packets that have successfully obtained an output VC from the VC allocator can make requests for their desired output channel. Peh and Dally [5] describe how this dependency may be relaxed if we speculate that a waiting packet will successfully be allocated an output VC. In this way both VC and switch allocation can be performed in parallel. To avoid a negative impact on performance the switch allocator in the speculative design must priorities non-speculative requests over speculative ones. This is achieved by implementing two switch allocators, one handling speculative requests (from packets that are also requesting a VC be to allocated) and another for non-speculative requests (from packets which have already been allocated a VC). Only when no non-speculative requests are granted for a particular output port are successful speculative requests granted. In the case that a speculative request is granted we must ensure that the VC has been allocated and it is capable of receiving a new flit (has free buffer space) before the flit is actually sent. Fortunately, such checks may be performed in parallel with crossbar traversal.

j) *Crossbar*

In the architecture illustrated in Figure 2 each input port is forced to share a single crossbar port even when multiple flits could be sent from different virtual-channel buffers. This restriction allows the crossbar size to be kept small and independent of the number of virtual-channels. Dally [25] and Chien [39] suggest that providing a single crossbar input for each physical input port will have little impact on performance as the data rate out of each input port is limited by its input bandwidth.

VIII. EXPERIMENTAL RESULTS

a. *Performance Evaluation*

In this section, we present simulation-based performance evaluation of our architecture, BiNoC router with VC flow-control technique in terms of network latency, energy consumption .We describe our experimental methodology, and detail the procedure followed in the evaluation of these architectures.

b. *Simulation Platform*

A cycle-accurate NoC simulator was developed in order to conduct a detailed evaluation of the router architectures. The simulator operates at the granularity of individual architectural components, accurately emulating the major hardware components. The simulation test-bench models both the routers and the interconnection links, conforming to the implementation of various NoC architectures. The simulator is fully parameterizable, allowing the user to specify parameters such as network size, topology, switching mechanism, routing algorithm, number of VCs per PC, number of PCs, buffer depth, PE injection rate, injection traffic-type, flit size, and number of flits per packet. The simulator models each individual component within the router architecture, allowing for detailed analysis of component utilizations and flit flow through the network. The activity factor of each component is used for analyzing power consumption within the network. We assume that link propagation happens within a single clock cycle. In addition to the network-specific parameters, our simulator accepts Hardware parameters such as power consumption (dynamic and leakage) for each component and overall clock frequency. These parameters are extracted from hardware synthesis tools and back annotated into the simulator for power profile analysis of the entire on-chip network.

c. *Simulation setup*

In this section the synthesis results will be presented, and a cost analysis of area and power consumption will bemade based on the synthesis results. The proposed BiNoC router with VC flow-control technique 5 port router architecture were implemented Is structural Register- Transfer Level (RTL) VHDL. A Router with parametrable flit size and 4 flits buffer depth and five ports have been modeled with VHD language on RTL level. They were simulated and synthesized respectively by using the ModelSim tool and ISE 13.1 tool.

d. Virtual Channel Functional Validation

The virtual channel was described in VHDL and validated by functional simulation. Figure presents a functional simulation for the most important signals and the simulation steps are described below.

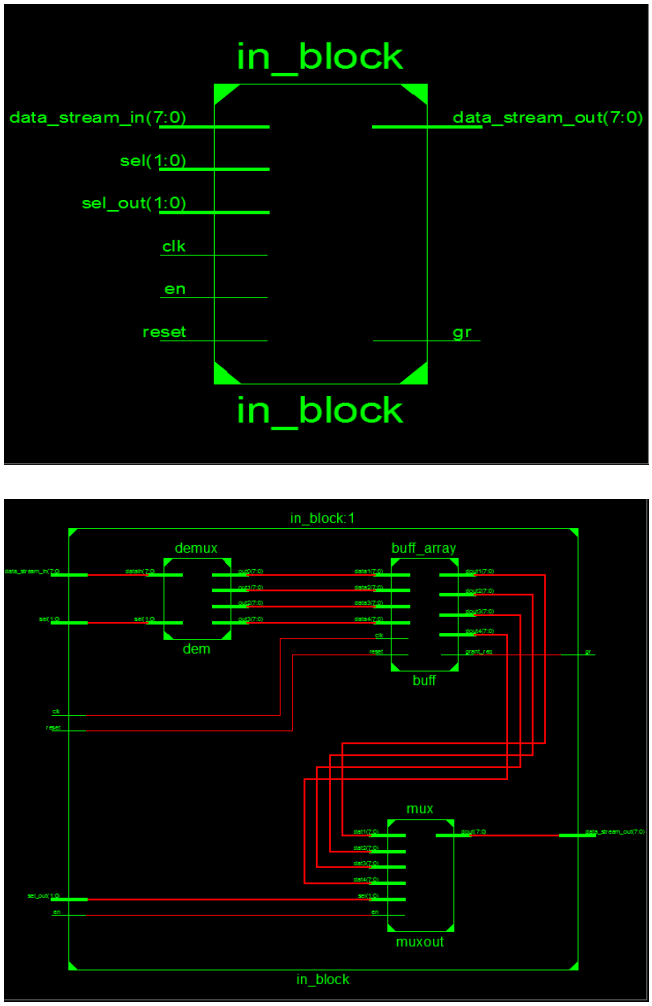
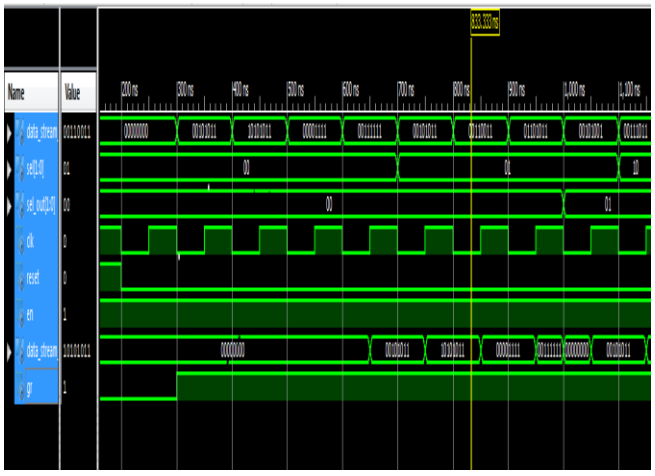


Fig .6 RTL simulation view of virtual channel



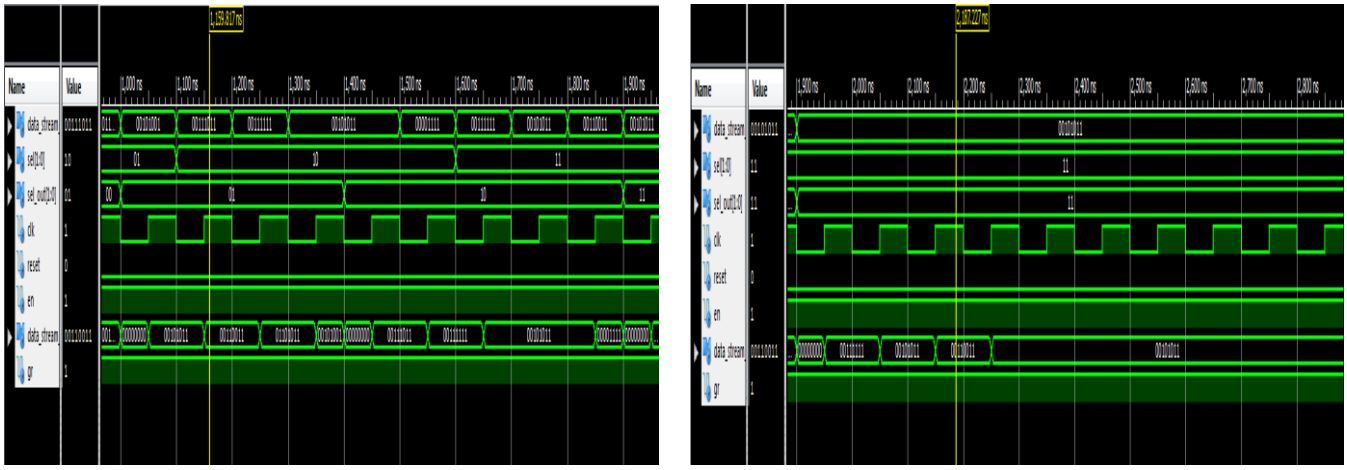


Fig.7 virtual channel simulation

I. Comparison with Existing Architectures

Table I. Comparison with existing NoC router architecture [40]

Architecture Resource	Typical NoC	Typical NoC-double	Reduced BiNoC	Normal BiNoC
Total number of buffers	5	5	5	10
Buffers/Direction	1	1	1	2
Total Channels	5-in 5-out	5-in 5-out	5-inout	10-inout
Channels/Direction	1-in 1-out	1-in 1-out	1-inout	2-inout
Each Buffer Size	32 flit	64 flits	32 flit	16 flit
Total Buffer size	160 flit	320 flits	160 flit	160 flit
Crossbar	5x5	5x5	5x5	10x10

e. Area Measurement

NoC router architectures in terms of logic gate count and percentage calculated by synopsys design compiler [40].

II. Area breakdown of BiNoC_4VC

Table II shows Area breakdown of *BiNoC_4VC* [40]

Component buff. Depth	BiNoC_4VC(16) 4 flits x 4	
	BiNoC_4VC Architectures Area (gate count)	Area (gate count) (%)
Input buf. + buf. ctrl	18,722	46.84
Routing computation	669	1.67
VC allocation	12,295	30.76
Switch allocation	2,245	5.62
Switch traversal	4,402	11.01
Bidir. ch. ctrl	1,628	4.07
Total	39,960	100.00

IX.CONCLUSION

Network-on-Chip (NOC) has been proposed over the years as an attractive alternative to traditional dedicated wire to achieve high performance and modularity. Power and Area efficiency is the most important concern in NOC design. We have implemented an accurate hardware model for reconfigurable virtual channel with VHDL and using it, have measured the performance, Area and power of several routing component. The effect of number of virtual channels on power and performance of NoC has also been studied. We also have synthesized this router on FPGA to estimate Area and power of each router component.

REFERENCES

[1] J. Held, J. Bautista, and S. Koehl, "From a Few Cores to Many: A Tera-scale Computing Research Overview," Intel Research (White Paper), 2006.
[2] P. Rickert, "Problems or opportunities? Beyond the 90nm frontier," ICCAD - Keynote Address, 2004.
[3] P. Guerrier and A. Greiner, "A generic architecture for on-chip packet-switched interconnections," in Proc. of the Design,Automation and Test in Europe pp. 250-256, 2000.
[4] L. Benini and G. D. Micheli, "Networks on Chips: A NewSoC Paradigm," IEEE Computer, vol. 35, pp. 70-78, 2002.

[5] L. S. Peh and W. J. Dally, "A delay model and speculative architecture for pipelined routers," in Proc. of the High Performance Computer Architecture (HPCA), pp. 255-266, 2001.

[6] J. Kim, D. Park, T. Theocharides, N. Vijaykrishnan, and C. R. Das, "A low latency router supporting adaptivity for on-chip interconnects," in Proc. of the Design Automation Conference (DAC), pp. 559-564, 2005.

[7] R. Mullins, A. West, and S. Moore, "Low-latency virtual-channel routers for on-chip networks," in Proc. of the International Symposium on Computer Architecture (ISCA), pp. 188-197, 2004.

[8] R. Marculescu, "Networks-on-chip: the quest for on-chip faulttolerant communication," in Proc. of the symposium on VLSI, pp. 8-12, 2003.

[9] D. Park, C. Nicopoulos, J. Kim, N. Vijaykrishnan., and C. R. Das, "Exploring Fault-Tolerant Network-on-Chip Architectures," in Proc. of the Dependable Systems and Networks (DSN), pp. 93-104, 2006.

[10] J. Duato, "A new theory of deadlock-free adaptive routing in wormhole networks," Parallel and Distributed Systems, IEEE Transactions on, vol. 4, pp. 1320-1331, 1993.

[11] K. V. Anjan and T. M. Pinkston, "An efficient, fully adaptive deadlock recovery scheme: DISHA," in Proc. of the International Symposium on Computer Architecture (ISCA), pp. 201-210, 1995.

[12] J. H. Kim, Z. Liu, and A. A. Chien, "Compressionless routing: a framework for adaptive and fault-tolerant routing," in Proc. of the International Symposium on Computer Architecture (ISCA), 1994.

[13] L. Shang, L. S. Peh, A. Kumar, and N. K. Jha, "Thermal Modeling, Characterization and Management of On-Chip Networks," in Proc. of the International Symposium on Microarchitecture (MICRO), pp. 67-78, 2004.

[14] K. Skadron, M. R. Stan, W. Huang, V. Sivakumar, S. Karthik, and D. Tarjan, "Temperature-aware microarchitecture," in Proc. of the 30th International Symposium on Computer Architecture, 2003.

[15] D. Brooks and M. Martonosi, "Dynamic thermal management for high-performance microprocessors," in Proc. of the High- Performance Computer Architecture (HPCA), pp. 171-182, 2001.

[16] U. Y. Ogras and R. Marculescu, "It's a small world after all: NoC Performance Optimization via Long-range Link Insertion," IEEE Trans VLSI Systems, vol. 14, no. 7, pp. 693–706, 2006.

[17] A. Kumar, L.-S. Peh, P. Kundu, and N. K. Jha, "Express Virtual Channels: Towards the Ideal Interconnection Fabric," in Proc. Int. Symp. Computer Architecture, 2007, pp. 150–161.

[18] M. F. Chang, J. Cong, A. Kaplan, M. Naik, G. Reinman, E. Socher,and S.-W. Tam, "CMP Network-on-chip Overlaid with Multi-band RFinterconnect,"in Proc. Int. Symp. High-Performance Computer Architecture, 2008, pp. 191–202.

[19] R. Mullins, A. West, and S. Moore, "The Design and Implementation of a Low-Latency On-Chip Network," in Proc. Asia & South Pacific Design
Automation Conf., 2006, pp. 164–169.

[20] H. Wang, L.-S. Peh, and S. Malik, "Power-driven Design of Router Microarchitectures in On-chip Networks," in Proc. Int. Symp. Microarchitecture, 2003, pp. 105–116.

[21] S. R. Vangal, J. Howard, G. Ruhl, S. Dighe, H. Wilson, J. Tschanz, D. Finan, A. Singh, T. Jacob, S. Jain, V. Erraguntla, C. Roberts, Y. Hoskote,
N. Borkar, and S. Borkar, "An 80-Tile Sub-100-W TeraFLOPS Processor in 65-nm CMOS," J. Solid-State Circuits, vol. 43, no. 1, pp. 29–41, 2008.

[22] L. S. Peh and W. J. Dally, "A delay model for router micro architectures," IEEE Micro, vol. 21, pp. 26-34, 2001.

[23] W. J. Dally and C. L. Seitz, "The torus routing chip," Journal of Distributed Computing, vol. 1(3), pp. 187-196, 1986.

[24] P. Kermani and L. Kleinrock, "Virtual cut-through: a new computer communication switching technique," Computer Networks, vol. 3(4), pp. 267-286, 1979.

[25] W. J. Dally, "Virtual-channel flow control," in Proceedings of the 17th Annual International Symposium on Computer Architecture (ISCA), pp. 60-68, 1990.

[26] W. J. Dally and C. L. Seitz, "Deadlock-free message routing in multiprocessor interconnection networks," IEEE Transactions on Computers, vol. C-36(5), pp. 547-553, 1987.

[27] Y. M. Boura and C. R. Das, "Performance analysis of buffering schemes in wormhole routers," IEEE Transactions on Computers, vol. 46, pp. 687-694, 1997.

[28] M. Rezazad and H. Sarbazi-azad, "The effect of virtual channel organization on the performance of interconnection networks," in Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium, 2005.

[29] E. Bolotin, I. Cidon, R. Ginosar, and A. Kolodny, "QNoC: QoS architecture and design process for network on chip", Journal of Systems Architecture, Volume 50, Issue 2-3 (Special Issue on Network on Chip), pp. 105-128, February 2004.

[30] E. Beigne, F. Clermidy, P. Vivet, A. Clouard, and M. Renaudin, "An asynchronous NOC architecture providing low latency service and its multi-level design framework", Proc. 11th International Symposium on Asynchronous Circuits and Systems (ASYNC), pp. 54–63, 2005.

[31] T. Bjerregaard and J. Sparsø, "A router architecture for connection-oriented service guarantees in the MANGO clockless Network-on-Chip", Proc. Of IEEE on Design Automation and Test, vol. 2, pp. 1226-1231, 2005.

[32] E. Bolotin, A. Morgenshtein, I. Cidon, R. Ginosar, and A. Kolodny, "Automatic hardware-efficient SoC integration by QoS Network-on-Chip", Proc. 11th International IEEE Conference on Electronics, Circuits and Systems, pp. 479-482, 2004.

[33] H. Zimmer, S. Zink, T. Hollstein, and M. Glesner, "Buffer-architecture exploration for routers in a hierarchical network-on-chip", Proc. 19th IEEE International Symposium on Parallel and Distributed Processing, pp., 1-4, April 2005.

[34] I. Saastamoinen, M. Alho, and J. Nurmi, "Buffer implementation for Proteo network-on-chip", International IEEE Proceeding on Circuits and Systems, vol. 2, pp. 113-116, May 2003.

[35] R. Mullins, A. West, and S. Moore, "Low-latency virtualchannel routers for on-chip networks," in Proceedings of the International Symposium on Computer Architecture (ISCA), pp. 188-197, 2004.

[36] J. Kim, D. Park, T. Theocharides, N. Vijaykrishnan, and C. R. Das, "A low latency router supporting adaptivity for on-chip interconnects," in Proc. of the Design Automation Conference (DAC), pp. 559-564, 2005.

[37] R. Mullins, A. West, and S. Moore, "Low-latency virtual-channel routers for on-chip networks," in Proc. of the International Symposium on Computer Architecture (ISCA), pp. 188-197, 2004.

[38] M. Galles, "Scalable Pipelined Interconnect for Distributed Endpoint Routing: The SGI SPIDER Chip," in Proc. of the Hot Interconnect Symposium IV, 1996.

[39] A. A. Chien. A cost and speed model for k-ary n-cube wormhole routers. In Proceedings of Hot Interconnects, 1993.

[40] Ying-Cherng Lan, Hsiao-An Lin, Shih-Hsin Lo, Yu Hen Hu, and Sao-Jie Chen, "A bidirectional noc (binoc) architecture with dynamic selfreconfigurable channel," Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on, vol. 30, no. 3, pp. 427 –440, march 2011.