Power and Area-Efficient Innovative Design of Dynamic Virtual Channel For BiNoC Router

Mr. Ashish Khodwe,
Department of
Electronics
Priyadarshini college
of Engineering ,
RTMNU, Nagpur,
India.

Prof. C. N. Bhoyar,
Department of
Electronics
Priyadarshini college
of Engineering ,
RTMNU, Nagpur,
India.

Abstract

Small optimizations in NoC router architecture can show a significant improvement in the overall performance of NoC based systems. Power consumption, area overhead and the entire NoC performance is influenced by the router buffers. Both NoC performance and energy budget depend heavily on the routers' buffer resources This paper introduces a novel unified buffer structure, called the Dynamic Reconfigure Virtual Channel, which dynamically allocates Virtual Channels (VC) and buffer resources according to network traffic conditions. The insertion of VCs also enables to implement policies for allocating the physical channel bandwidth, which enables to support quality of service (QoS) in applications. This paper presents a VHDL based cycle accurate register transfer level model for evaluating the, Area and power of Dynamically virtual channel BiNoC architectures. 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.

1. 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.

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 64% of the total node (router + link) leakage power for all process technologies, which makes it the largest power consumer in any NoC system [16]. Moreover, buffers are dominant for dynamic energy consumption [17]. It is better to transmit packets instead of storing them because more power consumption is expected in storing them as compared to the transmission [18]. Thus, reductions in number and size of buffers with increase in utilization affect the system performance and impact area and power efficiency.

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 remainder of this paper is organized as follows. In Section 2, we will discuss some of the background materials for NoC architecture and prior related research. In section 3, presents the motivation and requirements for presented router architecture. Our Dynamic virtual channel architecture and how it can contribute to reduce the area and power consumption without affecting throughput. will be given in Section 4. further section 5, router pipeline. in section 6,Overview of a Virtual-Channel Router. Finally, in Section 7, Finally experimental results and In last section 8, brief statements conclude this paper.

2. RELATED WORK

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% [16].

Lan et. al [19] addresses the buffer utilization by making the channels bidirectional and shows significant improvement in system performance. But in this case, each channel controller will have two additional tasks: dynamically configuring the channel direction and to allocate the channel to one of the routers, sharing the channel. Also, there is a 40% area overhead over the typical NoC router architecture due to double crossbar design and control logic. We approach the problem with very simple control logic.

Soteriou et. al [20] introduced distributed shared buffer (DSB) NoC router. The proposed architecture shows a significant improvement in throughput at the expense of area and power due to extra crossbar and complex arbitration scheme.

Kodi et. al [21] illustrates the impact of repeater insertion on inter-router links with adaptive control and eliminating some of the buffers in the router. The approach saves appreciable amount of power and area without significant degradation in the throughput and latency. But there is still some scope to increase the buffer utilization inside the router by using the architecture which we propose here. Neishabouri et. al [22] propose the router architecture with Reliability Aware Virtual Channel (RAVC). In this approach, more memory is allocated to the busy channels and less to the idle channels. This dynamic allocation of storage shows 7.1% and 3.1% latency decrease under uniform and transposes traffic patterns respectively at the expense of complex memory control logic. Though this solution is latency efficient but not area and power efficient, this was not discussed by the authors.

As the NoC design complexity rises, more communication mechanism issues are raised as well. Wormhole flow control have been proposed to reduce the buffer requirement 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 [23]. This introduces the problem of deadlock and livelock in wormhole scheme.

Virtual Channels (VCs) are used to avoid deadlock and livelock. a typical virtual channel router architecture [24]. 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 packets to be bypassed. The drawback of using VCs stands in a more complex control protocol, as data corresponding to different messages which is multiplexed on the physical channel must be eventually separated [23]. Another important issue that needs attention is the utilization tradeoff. VCs are proposed to increase the utilization of physical channels. By inserting the VC buffers, we increase the physical channel utilization but utilization of inserted VC buffers is not considered. It can be observed that if there is no communication on some channel at some time instant and at the same time, neighboring channel is overloaded, free buffers of one channel cannot contribute for congestion control by sharing the load of neighboring channel. Adaptive routing technique provides a solution to these issues but introduces some other problems like packet reordering.

A well designed network exploits available resources to improve performance [25]. So, a tradeoff between system performance and resource utilization is needed. Our motivation and innovation is to propose a router architecture with enhanced utilization of VC buffers without affecting the utilization of physical channel to reduce system latency, power consumption and silicon area.

3. 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].

4. BiNoC ARCHITECTURE

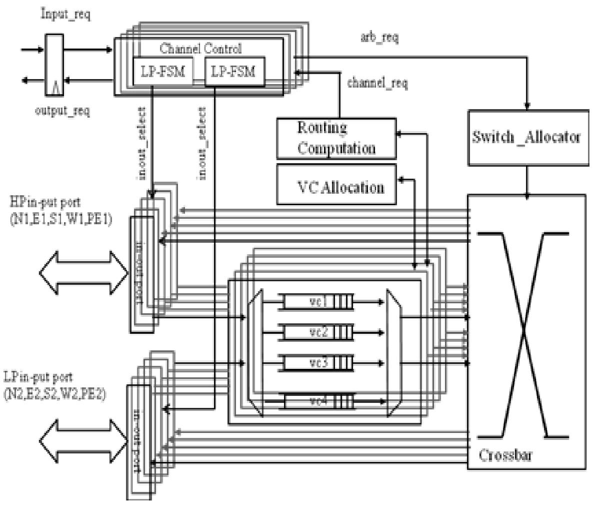


Fig.1: Modified four-stage pipelined router architecture for our proposed BiNoC router with VC flow-control technique.

Fig.1 shows the microarchitecture of A bidirectional channel network-on-chip (BiNoC) virtual channel (VC) router is modelled [26]. 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 [26]. 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.

5. Router Pipeline

A generic on-chip router consists of multiple atomic pipeline stages shown in fig.2; Routing Computation (RC), Virtual Channel Allocation (VA), Switch Allocation (SA), and Switch Traversal (ST) as shown in Figure 3. 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 [27, 28, 29].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.

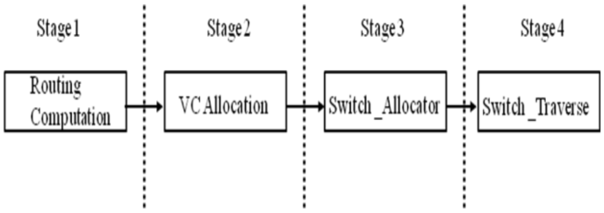


Fig.2: Typical four stage pipelined router design based on VC flow control.

Head flit	BWRC	VA	SA	ST	
Body/tail flit		BW	bubble	SA	ST

Fig.3:.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 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.

6. Overview of a Virtual-Channel Router

fig.1 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) 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.

d) 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.

e) 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 and is known as look-ahead routing.

f) 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 4. 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 4 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.

g) Switch Allocation

Individual flits arbitrate for access to physical channels via the crossbar on each cycle. Arbitration may be performed in two stages [30]. 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 4 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.

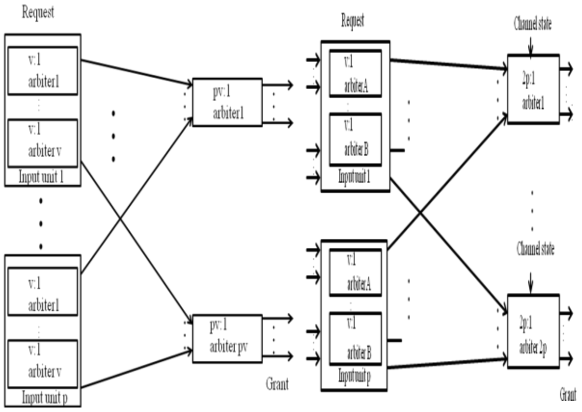


Fig.4: (a) VC allocator in a BiNoC router. (b) SA in a BiNoC router.

h) 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 [31] and Chien [32] 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.

7. 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 Area, energy consumption .We describe our experimental methodology, and detail the procedure followed in the evaluation of these architectures.

b. Simulation Platform

In this section, we evaluate the reconfigurable virtual channel BiNoC in terms of power dissipation, area overhead and overall network performance. We consider 4-stage pipelined router design. Each router has P = 5 input ports (4 for each direction and 1 for the PE). The baseline design considered has 4 VCs per input port, with each VC having 4 flit buffers in the router, for a total of 80 flit buffers (= 5 × 4 × 4). Each packet consists of 16 flits and each flit is 128 bits long. In each case, the design is implemented in VHDL language on RTL level and synthesized using the Xilinx ISE 13.1 tool.

c. 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.

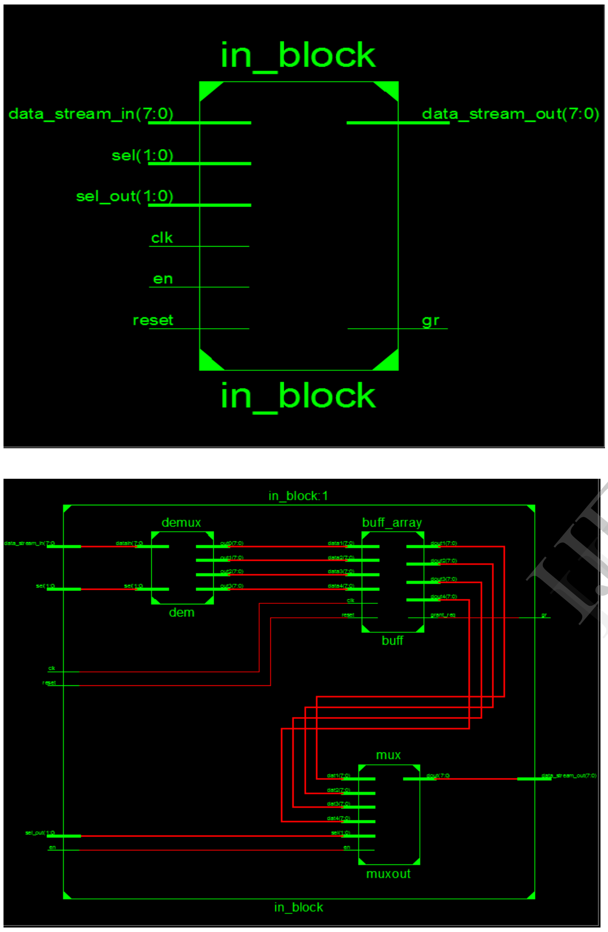


Fig.5: RTL simulation view of virtual channel

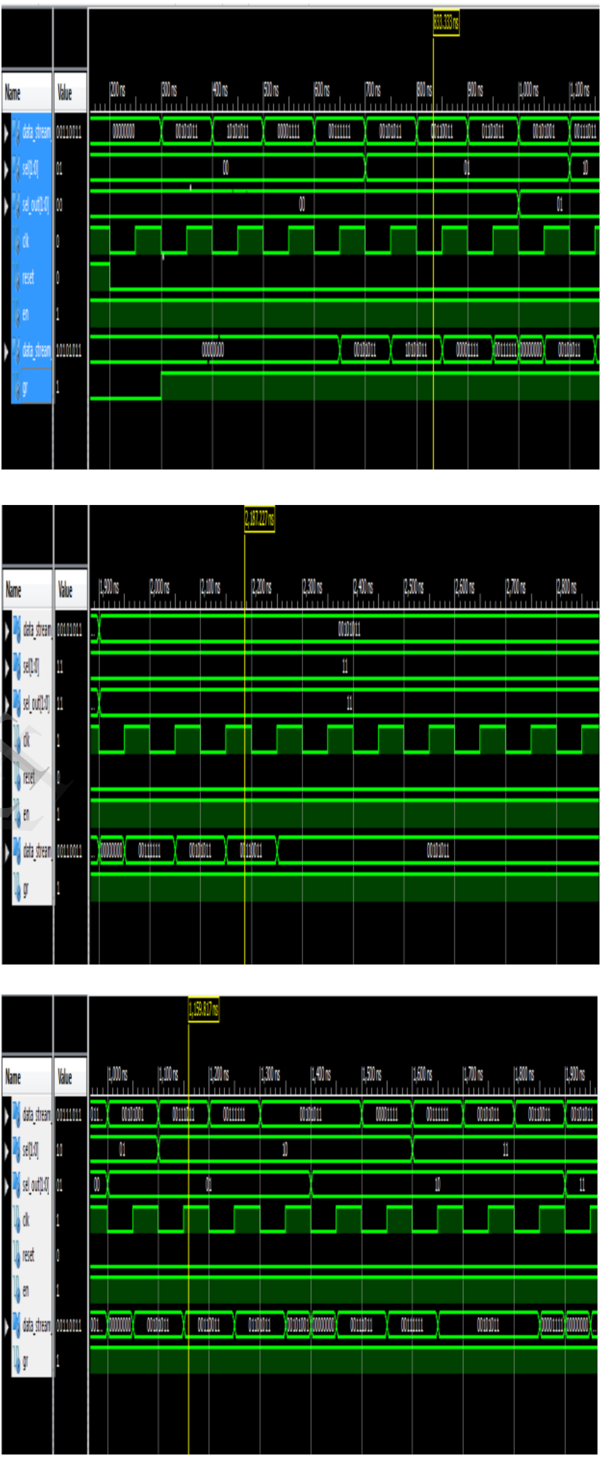
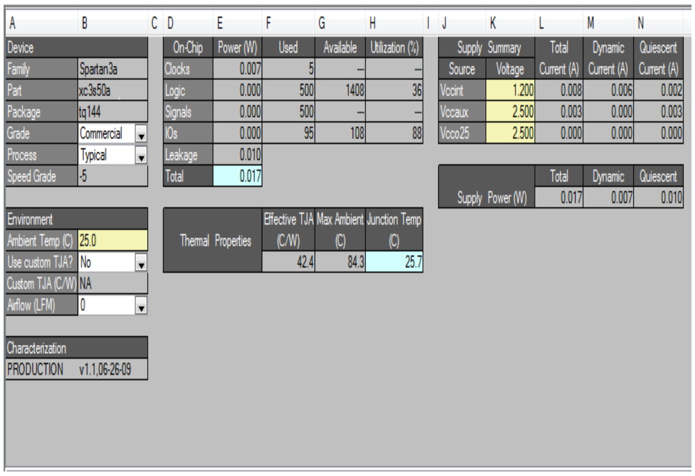


Fig.6:Virtual channel waveform simulation



Device	
Family	Spartan3a
Part	xc3s50a
Package	tq144
Grade	Commercial
Process	Typical
Speed Grade	-5

On-Chip	Power (W)	Used	Available	Utilization (%)
Clocks	0.007	5	---	---
Logic	0.000	500	1408	36
Signals	0.000	500	---	---
IOs	0.000	95	108	88
Leakage	0.010			
Total	0.017			

Supply Summary		Total	Dynamic	Quiescent
Source	Voltage	Current (A)	Current (A)	Current (A)
Vccint	1.200	0.008	0.006	0.002
Vccaux	2.500	0.003	0.000	0.003
Vcco25	2.500	0.000	0.000	0.000

	Total	Dynamic	Quiescent
Supply Power (W)	0.017	0.007	0.010

Environment	
Ambient Temp (C)	25.0
Use custom TJA?	No
Custom TJA (C/W)	NA
Airflow (LFM)	0

Thermal Properties	Effective TJA (C/W)	Max Ambient (C)	Junction Temp (C)
	42.4	84.3	25.7

Characterization	
PRODUCTION	v1.1,06-26-09

Fig.7: Power breakdown for 4VCs BiNoC

d. Power breakdown

The total dynamic power consume for a 128-bit flit in the buffer is estimated to be 17 mW shown in fig. 7

e. Area breakdown

The architecture was prototyped on a Spartan 3 FPGA, the hardware occupancy of the system in terms of FPGA slices has been provided in table I.

Table I. Area breakdown result of Reconfigurable Virtual Channel BiNoC router architecture

Resources	Mapping to Spartan 3A FPGA Device		
	Used	Available	Utility %
	4 VCs		4VCs
Slices	280	704	39%
Slices+ FF	275	1408	19%
4 input LUT	499	1408	35%
Bonded IOBs	95	144	65%

f. Measurement

NoC router architectures in terms of logic gate count and percentage calculated by synopsis design compiler [26].

I. *Area and Power breakdown of BiNoC_4VC*

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

Component buff. Depth	**BiNoC_4VC(16) 4 flits x 4**	
	Area (gate count)	**Power (mW)**
Input buf. + buf. ctrl	18,722	16.90
Routing computation	669	0.48
VC allocation	12,295	5.76
Switch allocation	2,245	1.75
Switch traversal	4,402	2.35
Bidir. ch. ctrl	1,628	0.68
Total	39,960	27.94

8. CONCLUSION

Flexible BiNoC router design and dynamic virtual channel not only effectively exploit area and power consumption, but also support more advanced features to accommodate various services using an NoC platform. We With multiplex a physical channel using virtual channels (VCs), we anticipate that Dynamic virtual channel has the potential of supporting better congestion control schemes, differentiated services and fault tolerance capability to accommodate more diverse services in the future. The architecture was prototyped on a Spartan 3A FPGA based Reconfigurable Virtual Channel BiNoC system is presented. 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 BiNoC has also been studied. We also have synthesized this router on FPGA to estimate Area and power of each router component. The performance can also be improved by selecting the routing algorithm that best suits the application under observation.

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] N. Banerjee, P. Vellanki and K.S. Chatha. A Power and Performance Model for Network-on-Chip Architectures. Proceedings of the conference on Design, automation and test in Europe (DATE), pp.1250-1255, Vol.2, 2004.

[17] Xuning Chen and Li-Shiuan Peh. Leakage powermodeling and optimization of interconnection networks.Proceedings of International Symposium on Low Power Electronics and Design, pp. 9095, 2003.

[18] T. T. Ye, L. Benini, G. De Micheli. Analysis of power consumption on switch fabrics in network routers Proceedings of the 39th Design Automation Conference(DAC), pp. 524-529, 2002.

[19] Ying-Cherng Lan, Shih-Hsin Lo, Yueh-Chi Lin, Yu-Hen Hu, Sao-Jie Chen. BiNoC: A bidirectional NoC architecture with dynamic self-reconfigurable channel. Proceedings of 3rd ACM/IEEE International Symposium On Networks-on-Chip (NoCS), pp.266-275, May
2009.

[20] V.Soteriou, R.S. Ramanujam, B. Lin, Li-Shiuan Peh. A High-Throughput Distributed Shared-Buffer NoC Router. IEEE Computer Architecture Letters, vol. 8, no. 1, pp. 21-24, Jan.-June 2009, doi:10.1109/LCA. 2009.5.

[21] A. Kodi, A. Louri, J. Wang. Design of energy-efficient channel buffers with router bypassing for network-onchips (NoCs) Proceedings of International Symposiumon Quality of Electronic Design (ISQED), pp.826-832, March 2009.

[22] M. H. Neishabouri, Zeljko Zilic. Reliability aware NoC router architecture using input channel buffer sharing. Proceedings of the 19th ACM Great Lakes symposium on VLSI (GLSVLSI), pp.511-516, 2009.

[23] Luca Benini and Giovanni De Micheli, Networks on Chips., Morgan Kaufmann Publishers, 2006.

[24] Robert Mullins, Andrew West and Simon Moore. Low-Latency Virtual-Channel Routers for On-Chip Networks. Proceedings of the 31st Annual IEEE International Symposium on Computer Architecture (ISCA), pp.188-197, 2004.

[25] James Balfour and William J. Dally. Design tradeoffs for tiled CMP on-chip networks. Proceedings of the 20th annual international conference on Supercomputing (ICS), pp.187-198, 2006.

[26] 26Ying-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.

[27] 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.

[28] 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.

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

[30] 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.

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

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