Elastic Buffer for Virtual Channels in Heterogeneous Switching Network on Chip

Gowthaman T V[1] Mugilan D[2]
[1,2]Department of Electronics and Communication Engineering
[1,2]K S Rangasamy College of Technology Tiruchengode, India

Abstract— The Network on Chip is the recent pattern, adopted in all types of systems on chip (SoCs) for reducing the complexity of system integration at the IP assembly level and from logical design to physical verification stage. On-chip network also enhances the system performance by partitioning the system cores and parallelizing the process. NOC provides the best solution to the wire delays and hence low latency period and high throughput has been achieved. The architecture of the router buffer is a critical design feature that affects both performance and implementation. Buffer architectures to support multiple Virtual Channels called Elastistore, which minimize the requirements of buffering without waiving the performance. A Virtual channel is the time multiplexed physical channel have different traffic flits, provided that separate buffer space for each flits. Elastistore having only single register per VC, round trip time appeared in NoC link is achieved by a large sized shared buffer. Elastistore integrated NoC gives efficient design at low cost along with similar performance. Area and delay of the router architecture were reduced considerably.

Key words: Network on Chip, Virtual Channel, Buffer, Elastistore, VLSI, Flits

I. INTRODUCTION

Network on Chip is the integrated communication technology used in all recent system on chip (socs). Interconnection between the computing resources such as CPU, Processor etc., to build a single system called system on chip is provided by noc to face the growing high speed processors. Network on Chip replaces the regular hierarchal bus and crossbar interconnects by employing at a system level. Semiconductor transistors sizes are shrinking day by day, millions of transistors and IP blocks are fabricating in a single chip. NOC is adopted to provide the easy time closure, reduce routing congestion and higher operating frequencies.

NOC should be scalable in network performance and its functionality; same time it also less complexity in terms of physical design along with area and power minimization. These requirements can be possible by unifying the VC and buffer based architecture. Elastic Buffer is the fundamental and simpler form of buffering in noc because of its simple ready, valid and handshake operation, so it can be used as a plug and play approaching both input and output sides of noc router. This can also be integrated inside to router [5], this will acts as a buffer replicators in network links [8]. The control logic for elastic buffer was possible by using pipelined flip flops as an elastic buffer with more than one storage locations [10]-[11].

II. ELASTIC CHANNELS AND BUFFERS

The single lane elastic channel between elastic buffer consists of two types of control channels, which is valid and ready wire along with parallel data channel comprises the implementation of EB protocol (Fig. 1a). In this protocol. If the ready control signal is asserted high, EB accepts the input signal and if valid control is asserted high, then it has output data stream. If both the control signals are high in elastic buffer data transmission will occur. Elastic data flow is illustrated in Fig. 1b.

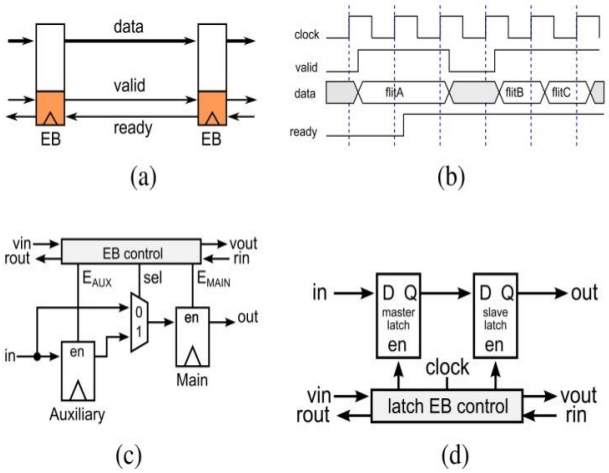

Fig. 1: Elastic Buffer Protocol. The two control wires and a Data wire together forms single lane channel. (a) Elastic Channel. (b) Data flow in EB. (c) Flip-Flop Elastic Buffer. (d) Latch Elastic Buffer

When the elastic buffer chain's output stops, it will propagate at the rate of one stage at each cycle. EB can keep two words at a same time, one for output data and another one for propagated data from next stage is given in Fig. 1c. In Fig. 1d, same design architecture built from latched was illustrated.

III. ELASTISTORE ARCHITECTURE

The intercommunication between the cores in the critical soc like Quad-core, Octa-core processor is need to be efficient in terms of loss, power, area and physical implementation, noc is the best way to fulfil all this requirements. Elastistore is the one of the best method to reduce the power, area and implementation complexity. In this method, buffer usage is limited by assigning one separate register for each virtual circuit, along with one large shared buffer which is connected to all virtual circuits. Shared buffer is used to get the round trip latency. Baseline method utilizes two buffer spaces for one virtual circuit, this increases the power consumption. The Elastistore architecture with three virtual circuits is illustrated in Fig. 2. This comprises of control unit and shared buffer unit along with single buffer for each virtual circuit. Hence this architecture utilizes only N+1 buffer for N virtual circuit noc.

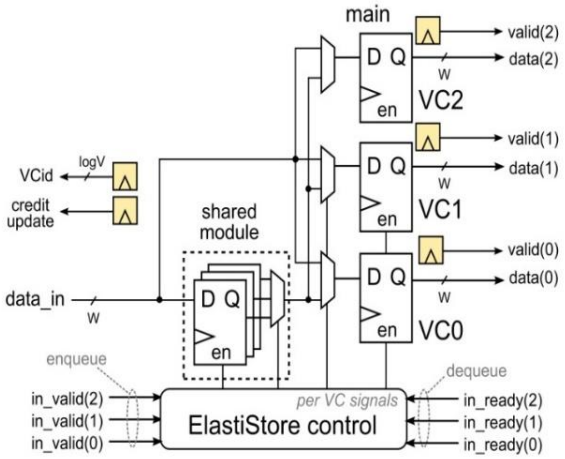

Fig. 2: Elastic Store Organisation for Three vcs. Elastistore consists of Single Register per VC and one Shared buffer dynamically shared with all Virtual Circuits.

A. Operational Details

Elastistore consists of only one register per virtual circuits, which is nothing but the main register along with one shared register which dynamically shared with all virtual circuits in the noc. As in the EB protocol, this consists of valid and ready control signal, Elastistore control manages the operation of whole module. Noc will gets input data only when ready signal is asserted high, this will depends on traffic of the virtual channel i.e., empty buffer or occupied buffer (availability of buffer space), along with the data requirements. Similarly valid signal is asserted high, only when valid output data reaches the output port. Then only, buffer sends the data out. Shared buffer is used to achieve the round trip latency.

NOC Buffers	I-Stage
Baseline	(3V+1)N
ES-IO	(2V+2)N

Table 1: Minimum Buffering Requirements for Baseline and Elastistore noc. (N - No. Of V C)

When a new flow bits belongs to particular Virtual Circuit, there may be a possible of placing the data to main buffer or shared buffer. The data takes the main buffer for particular virtual channel, if that VC is empty or will becomes empty in the current clock cycle, else that data takes the shared buffer for its transmission. Sometimes both main and shared register are filled then in that case control unit does not assert ready signal to high. Ready signal will be in low state till all data's are drained to output side. Similarly valid signal also works like this.

B. Shared Buffer

Shared buffer is the main feature in this architecture. FIFO order should be maintained in the shared buffer within the elastic store to achieve round trip latency. FIFO order can be preserved by shifting the flits forward every time in the shared buffer. But in the real process the flits are not shifted physically. The pointer is shifted, instead of flits shifting. In this manner the FIFO position is maintained throughout the logic. A shared buffer shown in figure consists of two main storage parts. The first one is the shift-register-base FIFO and the second one is the register file. The former part stores the VC ID of a flit and a pointer and the latter part stores the actual flit content.

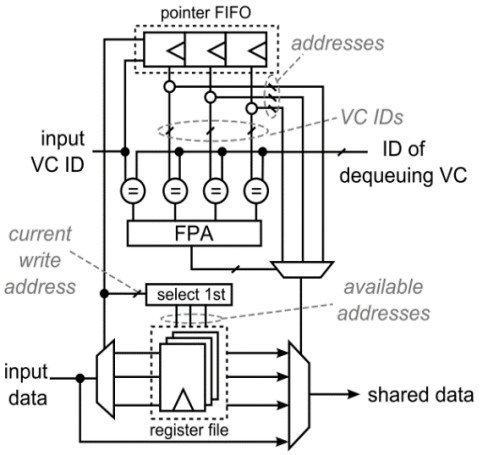

Fig. 3: Shared Buffer of the Elastic Store.

The actual contents are written in the register file, when a flit is pushed to the shared buffer at the same time the first empty slot of the shift register is pushed with the write address and the VC ID of the incoming flit. Router allocation is not a part in the proposed shared buffer, as in the second design rule.

The first appearance of the dequeuing VC ID is read out on the read operation. This operation is done by comparing the pointer FIFO. It checks for the pointer to match. Then this is used to retrieve the actual content from the register file. When there is no content the pointer shifts forward. At the same time, available register file is marked and cane be reallocated to any VC.

IV. RESULTS AND DISCUSSIONS

Elastistore architecture is a add-on, that can be used as a plug and play in input and output of noc routers. As it requires only a single buffer space per virtual circuit, so area efficiency can be achieved. This also contains shared buffer, and hence, Round trip latency can be achieved. The simulation result of Shared Buffer architecture in elastistore was in Figure 4, in which Data_in and Shared_data represents input and output data respectively. Rorw is the control signal, 0 and 1 for Read and Write operation. Vc_idin and vc_iddq is the input VC ID and output VC ID, where there read and write operation take place. Valid signal in the output waveforms confirm that read or write is successful or not, in which 1 is successful operation and vice versa.

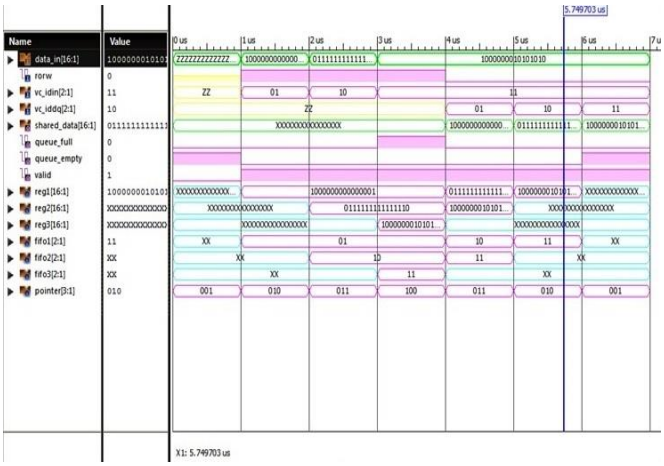

Fig. 4: Output Waveforms for Shared Buffer Architecture with 4 bits data.

The simulation result of elastistore architecture was in Figure 5, in which Data_in and Data_out represents input and output data respectively. Eqordq is the control signal, 0 and 1 for Read and Write operation. Vc_no is the VC ID, where the read and write operation take place. Valid signal in the output waveforms confirm that read or write is successful or not, in which 1 is successful operation and vice versa.

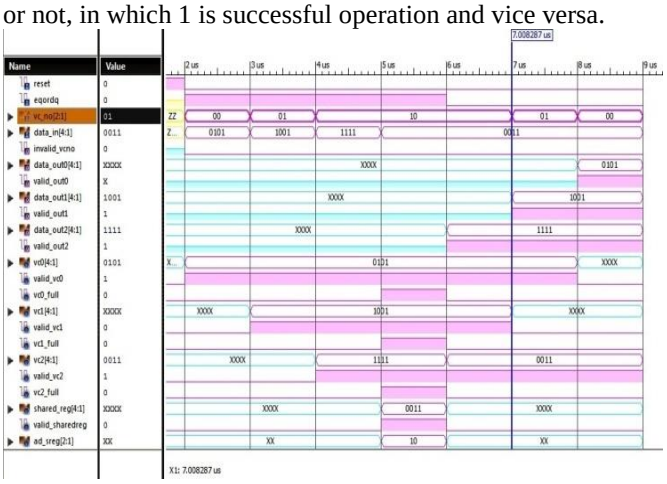

Fig. 5: Output Waveforms for 3VC elastistore Architecture with 4 bits data.

Power is the main concern in all noc routers, so better architecture should take care of the power consumption problem also, in parallel to the area and efficiency.

Noc Buffers	4 vcs	
	I-Stage	II-Stage
ES-I	67	69

Table 2: Energy Required per Cycles (In picojules)

As discussed earlier because of its absolute minimum buffer requirement, very less amount of power is needed when compared to the regular one. The cost efficient noc router can be designed with the help of this architecture with less area overhead.

V. CONCLUSION

The buffer architecture required for the noc router is a complicated design, which can alter performance, size and power of the noc. Elastistore is the simplest buffering for VC, uses only a single register per VC along with dynamically shared register used to achieve maximum throughput. This new architecture design gives the way for designing low cost router with power efficiency. Elastistore implemented router architecture will give the same network performance with large area savings.

ACKNOWLEDGMENT

We would like to thank the centre for VLSI Design, Department of Electronics and Communication Engineering, K S Rangasamy College of Technology, Tiruchengode, Tamilnadu, for providing the FPGA kit and Synopsys tools.

REFERENCES

[1] Dominic ditomaso, Avinash Kodi, Ahmed Louri, and Razvan Bunescu,"Resilient and Power-Efficient Multi-Function Channel Buffers in Network-on-Chip Architectures," IEEE Trans. Comput., Vol.:PP, Issue: 99.

[2] Masoud Oveis-Gharan and Gul N. Khan, "Efficient Dynamic Virtual Channel Organization and Architecture for noc Systems," IEEE Transactions on VLSI Systems, Vol.:PP, Issue: 99.

[3] Dong Guoyue Jiang, Zhaolin Li, Fang Wang, and Shaojun Wei, "A Low-Latency and Low-Power Hybrid Scheme for On-Chip Networks," IEEE Transactions on VLSI Systems, Vol.:23, PP:664-677, Issue:4.

[4] Zhiliang Qian, Syed Mohsin Abbas, and Chi-Ying Tsui, "fsnoc: A Flit-Level Speedup Scheme for Network-On-Chips Using Self-Reconfigurable Bidirectional Channels," IEEE Trans on VLSI Systems.

[5] Yaniv Ben-Itzhak, Israel Cidon, Avinoam Kolodny, Michael Shabun, and Nir Shmuel, "Heterogeneous noc Router Architecture," IEEE Transactions on VLSI Systems, Vol.:PP, Issue: 99.

[6] Manuel Cheng Li, Mark Browning, Paul V. Gratz, and Samuel Palermo, "luminoc: A Power-Efficient, High-Performance, Photonic Network-on-Chip," IEEE Trans. Computer-Aided Design of I C and Systems, Vol. 33, No. 6.

[7] Anh T. Tran, and Bevan M. Baas, "Achieving High-Performance On-Chip Networks With Shared-Buffer Routers," IEEE Trans. VLSI Systems, Vol. 22, No. 6.

[8] G. Michelogiannakis and W. J. Dally, "Elastic buffer flow control for on-chip networks," IEEE Trans. Comput., vol. 62, no. 2, pp. 295–309.

[9] Jacob Postman, Tushar Krishna, Christopher Edmonds, Li-Shiuan Peh, and Patrick Chiang (2013), "SWIFT: A Low-Power Network-On-Chip Implementing the Token Flow Control Router Architecture with Swing-Reduced Interconnects," IEEE Trans. VLSI Systems, Vol. 21, No. 8.

[10] Y. J. Yoon, N. Concer, M. Petracca, and L. P. Carloni, "Virtual channels and multiple physical networks: Two alternatives to improve noc performance," IEEE Trans. Comput.-Aided Design IC Syst., Vol. 32, No. 12, pp. 1906–1919.

[11] Ying-Cherng Lan, Hsiao-An Lin, Shih-Hsin Lo, Yu Hen Hu, and Sao-Jie Chen (2012), "A Bidirectional noc (binoc) Architecture with Dynamic Self-Reconfigurable Channel," IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems, Vol. 30, No. 3.

[12] Rajdeep Bondade, and Dongsheng Ma (2012), "Self-Reconfigurable Channel Data Buffering Scheme and Circuit Design for Adaptive Flow Control in Power-Efficient Network-on-Chips," IEEE Trans. Circuits And Systems—I: Regular Papers, Vol. 57, No. 11.

