Improvement Throughput of Multicast GALS in Synchronous and Asynchronous NoCs

[1]B.Rajasekhar, [2]Dr.K.Rasadurai
[1]PG Scholar, Department of ECE, Kuppam Engineering College, Kuppam, Andhra Pradesh, India.
[2] Associate professor, Department of ECE, Kuppam Engineering College, Kuppam, Andhra Pradesh, India

Abstract: **Networks-On-Chip (NoCs) have become the standard for communication in many-core processors where the system performance and power not only depend on the computing efficiency but are also governed by the on-chip interconnects. A key challenge for modern NoCs is to efficiently support new traffic patterns, common in advanced parallel architectures eg., Multicast (i.e., one-to-many) is is defined as sending the same packet from a single source to an arbitrary subset of destinations, widely used in the parallel computing domain: 1) in cache coherence protocols, to send write invalidates to multiple cores, e.g., in *HyperTransport* and *Token Coherence* protocols, 2) in shared operand networks, to deliver operands to multiple instructions; and 3) in multithreaded applications, to notify barrier synchronization to multiple processors. Also, multicast is used in testing of NoCs for fast delivery of test packets to multiple routers. Multicast is also gaining importance in all inherent forms of communication in emerging NoC technologies: wireless, surface-wave, photonic, and CDMA in applications like neuromorphic computing, both in spiking neural networks (SNNs) as well as in deep convolutional neural networks. A parallel multicast asynchronous NoC with a 2-D mesh topology is proposed which is the general-purpose asynchronous NoC to support multicast in 2-D meshes with a critical feature as the use of a new continuous-time replication strategy, where the flits of a multicast packet are routed through the distinct outputs of the router according to each output's own rate, in parallel, and in continuous time. This NoC has asynchronous continuous-time replication which is not discretized to clock cycles and can handle subtle variations in network congestion and exploit "subcycle" differentials in operating speeds. For diverse multicast benchmarks, the new parallel multicast network is achieved significant latency and throughput gains over a serial baseline. Interestingly, consistent latency improvements were observed for unicast, in spite of the extra instrumentation.**

Index Terms**: Asynchronous circuits, multicast communication, networks-on-chip (NoCs).**

1. INTRODUCTION

Networks-On-Chip (NoCs) have been the standard for communication in many-core processors or applications like in cache coherence protocols, to send write invalidates to multiple cores, e.g., in HyperTransport and Token Coherence protocols. In shared operand networks, to deliver operands to multiple instructions; and multithreaded applications, to notify barrier synchronization to multiple processors. Also, multicast is used in testing of NoCs for fast delivery of test packets to multiple routers. Multicast is also gaining importance in all inherent forms of communication in emerging NoC technologies: wireless, surface-wave, photonic, and CDMA (Code Division Multiple Access) in applications like neuromorphic computing, both in spiking neural networks (SNNs) as well as in deep convolution neural networks.

There has been significant research on multicast in synchronous NoCs [5], [22]–[24]. These techniques can be divided into two categories: serial path-based multicast and parallel tree-based multicast. In the path-based, a multicast packet is serially routed from the source to its first destination, from there to the next, and so on [22], [23], [25]. This technique is simple but can incur significant latency overheads for a large number of destinations. The tree-based multicast is more widely used, where a packet is first routed on a common path from the source toward all destinations. When this common path ends, the packet is replicated. The new copies also follow a recursive tree approach, replicating multiple times to reach the destinations [5], [7], [24], [26]–[28]. Several works use the tree approach for high-performance multicast, but these can still incur significant cost overheads. Early tree-based approaches have used multiple unicast packets to set up paths for a multicast packet. This pre configuration phase can be expensive in terms of network latency, extra congestion, and power [5], [26]. Recent approaches avoid setup entirely and dynamically compute the multicast tree paths based on the destinations. However, these approaches lead to complex router designs due to highly customized route computation, multiple virtual channels (VCs) per port, and turn prohibitions to avoid deadlocks [24], [29].

Finally, a recent high-performance NoC [7] extends an earlier unicast-only SMART NoC [30] to support multicast. This new NoC achieves full broadcast in 2 cycles for an 8 × 8 2-D mesh using an early arbitration and channel preallocation approach, facilitated by a high-speed monitoring network that shadows the datapath. Multicast is supported in two ways:

1) if the number of destinations is large, a full-chip broadcast is first performed, followed by dropping packets at non destinations and

2) if the number of destinations is small, multicast packets are broken into several small unicast packets and injected and routed serially to each destination. Both of these approaches can have serious energy overheads.

In contrast, the proposed asynchronous NoC performs multicast to all destinations using a parallel transmission. Moreover, the monitoring network in the multicast SMART NoC also adds significant area and power costs; wide monitoring channels are used, with 24 distinct SMART links emanating from each router, where each such link can be 2–4 bit wide. As a result, a number of these switches have several dozen extra wires (e.g., 48–96) devoted solely to monitoring.

This paper is organized as section II describes Existing System and in section III discussed Proposed Method and

Section IV describes in Simulation results and Section V concludes the paper followed by references.

2. EXISTING SYSTEM

(i) SERIAL PATH BASED MULTICAST

In the path-based, a multicast packet is serially routed from the source to its first destination, from there to the next, and so on as shown in figure 1. Even though it is simple but can incur significant latency overheads for a large number of destinations.

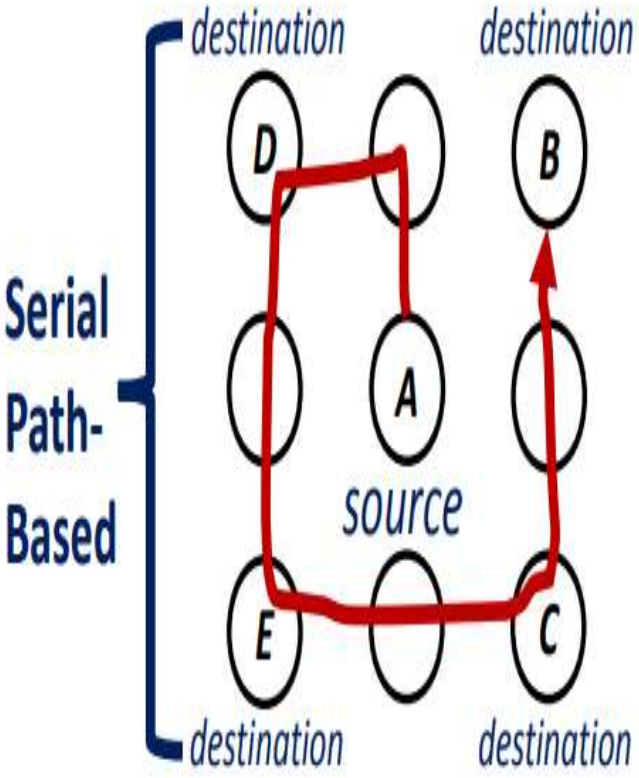

Figure 1: Serial path based Multicast NOC

(ii) TREE BASED MULTICAST

The tree-based multicast is more widely used, where a packet is first routed on a common path from the source toward all destinations. When this common path ends, the packet is diverged and replicated. It has drawback i.e, the new copies also follow a recursive tree approach, replicating multiple times to reach the destinations.

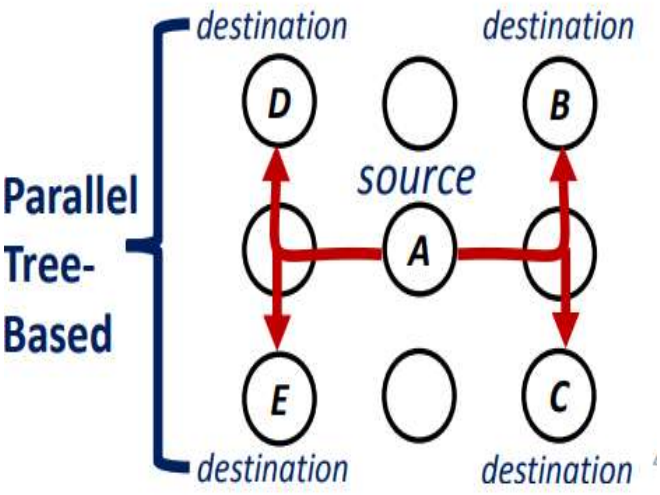

Figure 2: Parallel tree based multipath NOC.

It is usually widely used high-performance design of NOC. The Earlier works set up tree in advance using multiple unicasts but recent works do not use unicast-based set up: tree constructed dynamically. The other design includes the new IPM design, which supports the continuous-time replication strategy, along with the details on its RCU and the CMR buffer as shown in fig. 3. The IPM has one input channel that connects to the upstream router and four output channels toward the OPMs through a crossbar. There are three components in the new IPM: an RCU, the CMR buffer, and four address modifier units (AMUs) on each output direction. The RCU stores only the header addressing and selects the correct OPMs for routing. The CMR buffer, however, stores all the flits, which can be accessed by the OPMs in parallel using its four decoupled read ports. Finally, an AMU is present at each output of the CMR buffer, which modifies the header address such that there is always a unique path for the multicast packet to reach each destination, preventing sending multiple copies of the packet to the same destination through different paths.

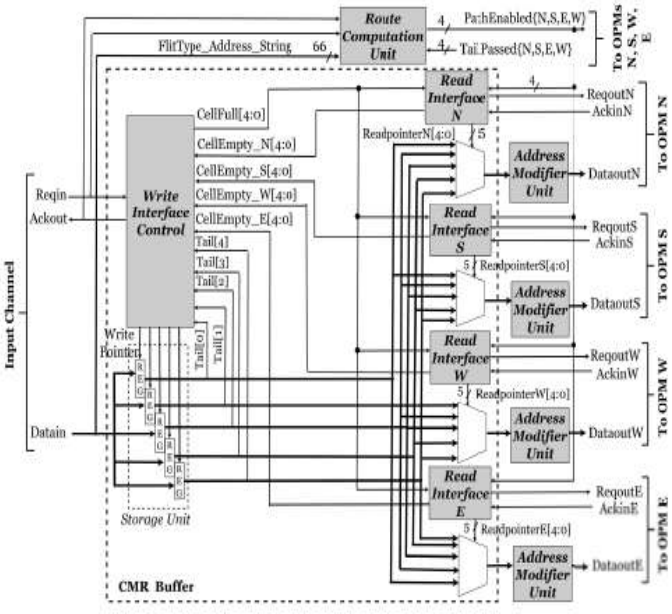

Figure 3. IPM micro architecture with CMR buffer.

Its operates as when a new packet header arrives, it is stored in the CMR buffer, and concurrently, its address is stored in

a small buffer in the RCU to start route computation. After the header is stored in the CMR buffer, it is speculatively broadcast to all read interfaces. The write interface, next, generates an Ackout on the input channel, which is used to:

1) advance the buffer's write pointer and

2) close the buffer in the RCU, disabling it for the remainder of the packet to save energy. A similar write protocol is followed for the body flits until the tail arrives and is stored.

The Ackout for the tail is sent only after the tail has been read by all the correct OPMs of the buffer. This Ackout also reactivates the RCU. Read operations on the buffer are performed in para Read operations on the buffer are performed in parallel with the write. The header is first speculatively read out of all the read interfaces, with AMU address modifications, and sent to all the OPMs along with Reqouts. After the RCU finishes the route computation, the correct OPMs are selected using PathEnabled, which are also used by the read interfaces to throttle the copies on the wrong paths. Each read interface on the correct paths receives an Ackin from its OPM after the header has been routed through the OPM and advances its individual read pointer. Similar read operations are performed for the body and tail flits, where each OPM on the correct path reads these flits independently

Before sending the header to the OPMs, the AMU modifies the header addressing, through a masking operation, to guarantee a unique path for the multicast packet from the current router to each destination. This perturbation is only performed on the header address, keeping the body/tail flits unchanged. The address modification is implemented using a network partitioning approach, where four 64-bit partition bit strings are used, corresponding to each output direction, which has a bit for every node: 1 if the node is reachable through this direction using XY routing, 0 otherwise. At each AMU, a bitwise AND is then performed between the header address bit string and the corresponding partition bit string; the result is a new address bit string with some destinations masked that are not reachable by XY routing through this direction.

This efficient parallel multicast capability in asynchronous NoCs is introduced which exploits asynchronous and GALS NoCs which fulfills a critical architectural need to enable advanced computing systems. But still it has drawbacks like it is suited only for Mesh Topology and it uses Time Replication Strategy.

3. PROPOSED METHOD

The Multicast GALS NoC that includes both Synchronous and Asynchronous Transmission with a slight change in IPM and OPM Architecture to support both synchronous and asynchronous transmission and reception of data packets. It has several advantages like it supports efficient many-to-one traffic, it is suitable for any topology and has improved Throughput.

4. SIMULATION RESULTS

The designs are modelled in Verilog HDL and are functionally verified by using Xilinx ISIM Simulation Tool. The designs are synthesized for Spartan3E FPGA by suing Xilinx ISE 14.5 Tools for the device XC3S500E with a package of FG320 and a speed grade of -5.

The simulation waveform for tree based multicast NOC is shown in figure 4 where clearly the packet is routed to every other node using tree based structure. This design is converted to register transfer logic by the Xilinx synthesizer as shown in figure 5 and to 90nm CMOS technology based LUT mapped schematic as shown in figure 6. The tree based multicast NOC design is extracted for FPGA along with its sample routing without imposing constraints is shown in figure 7.

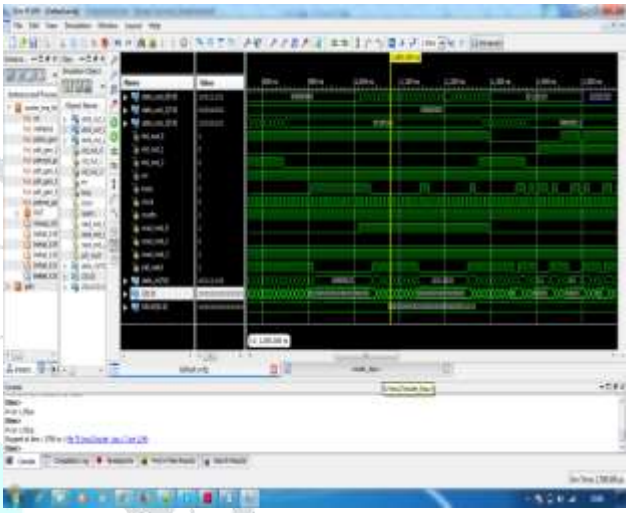

Figure 4: Simulation result of tree based multicast NOC Design

Figure 5: RTL Schematic of tree based multicast NOC Design

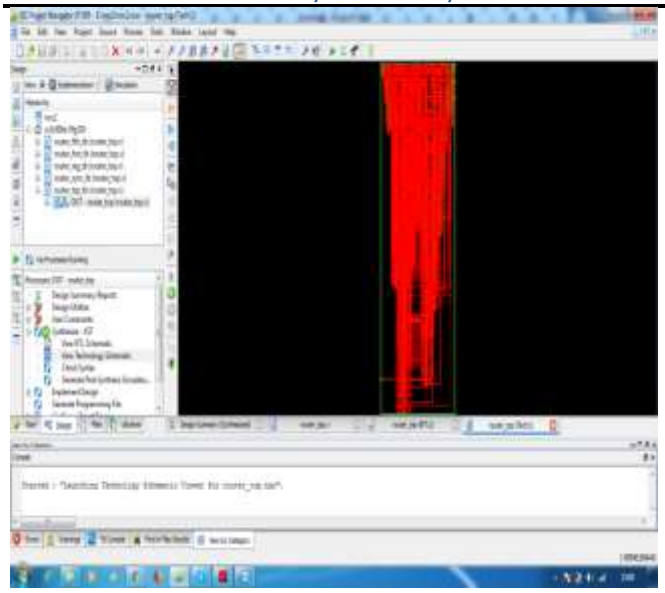

Figure 6: Technology Schematic of tree based multicast NOC Design

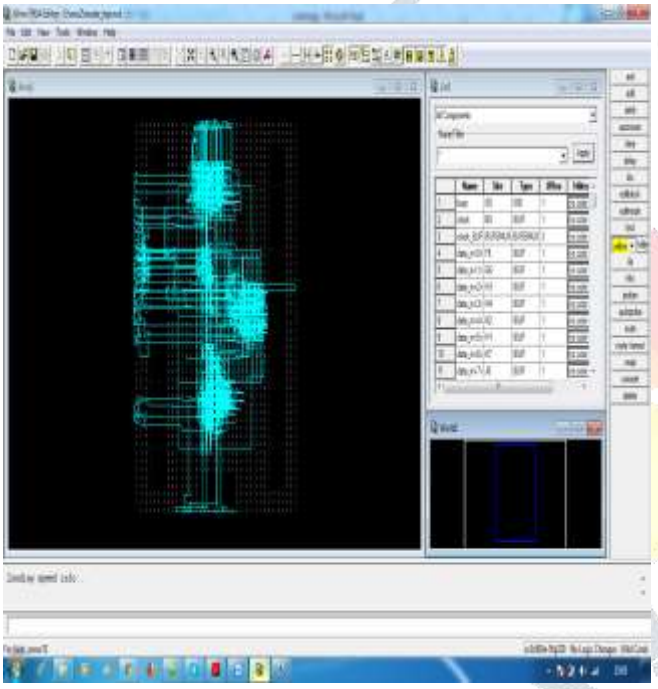

Figure 7: FPGA Implementation of tree based multicast NOC Design

The simulation waveform of IPM and OPM Architecture of existing Router NOC is shown in figure 8 where clearly the packet is routed to every other node using tree based architecture. This design is converted to register transfer logic by the Xilinx synthesizer as shown in figure 9 and to 90nm CMOS technology based LUT mapped schematic as shown in figure 10. The IPM and OPM Architecture of existing Router for multicast NOC design is extracted for FPGA along with its sample routing without imposing constraints is shown in figure 11.

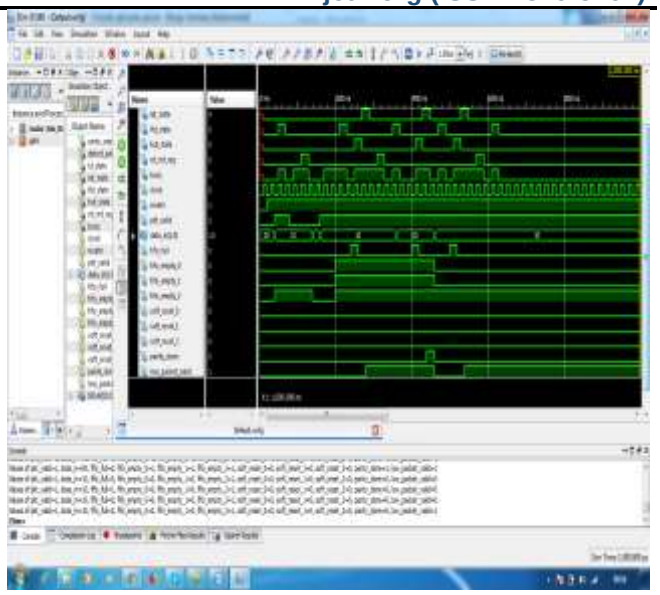

Figure 8: Simulation Waveform of IPM and OPM Architecture of existing Router

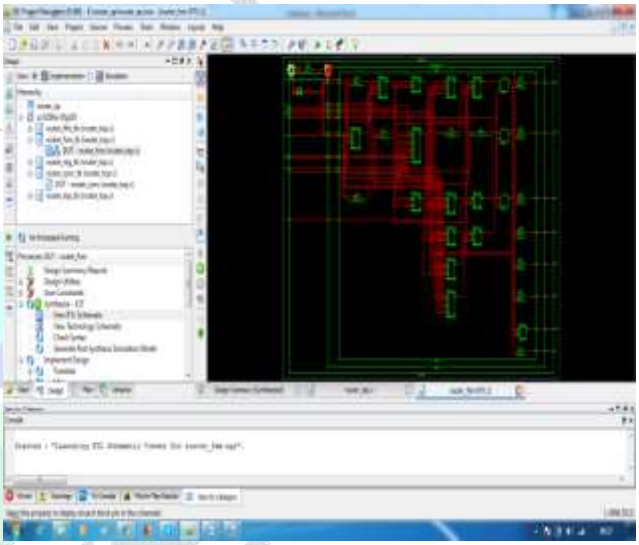

figure 9: RTL Schematic of IPM and OPM Architecture of existing Router

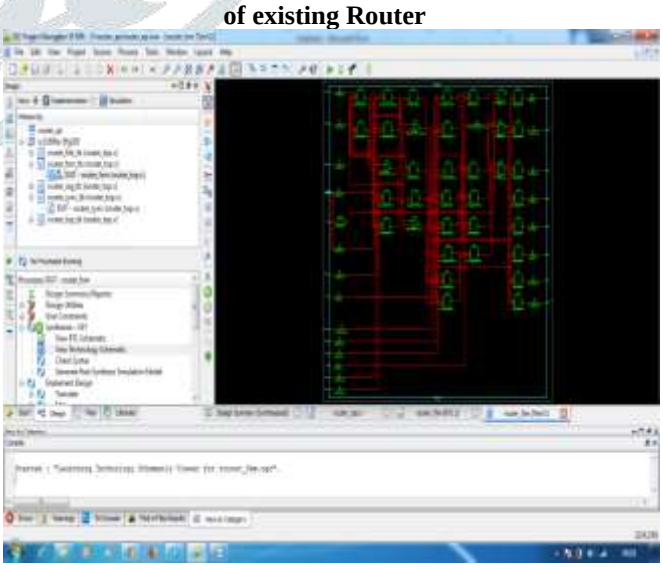

Figure 10: Technology Schematic of IPM and OPM Architecture of existing Router

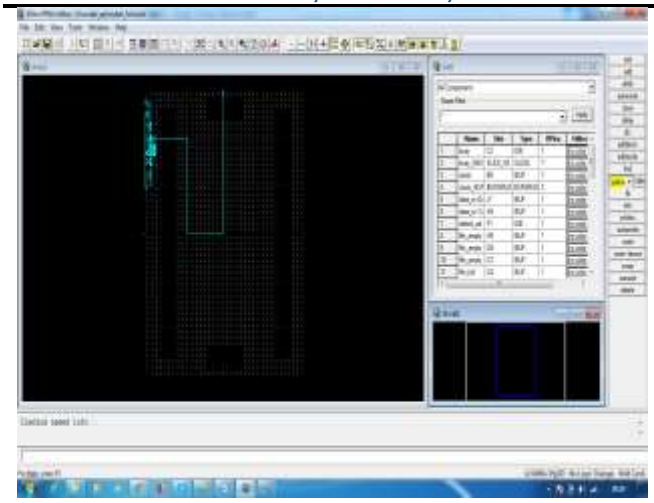

Figure 11: FPGA implementation of IPM and OPM Architecture of existing Router

The simulation waveform of Proposed Router that supports both synchronous and asynchronous routing in NOC is shown in figure 12 where clearly the packet is routed to every other node. This design is converted to register transfer logic by the Xilinx synthesizer as shown in figure 13 and to 90nm CMOS technology based LUT mapped schematic as shown in figure 14. The proposed Router for multicast NOC design is extracted for FPGA along with its sample routing without imposing constraints is shown in figure 15.

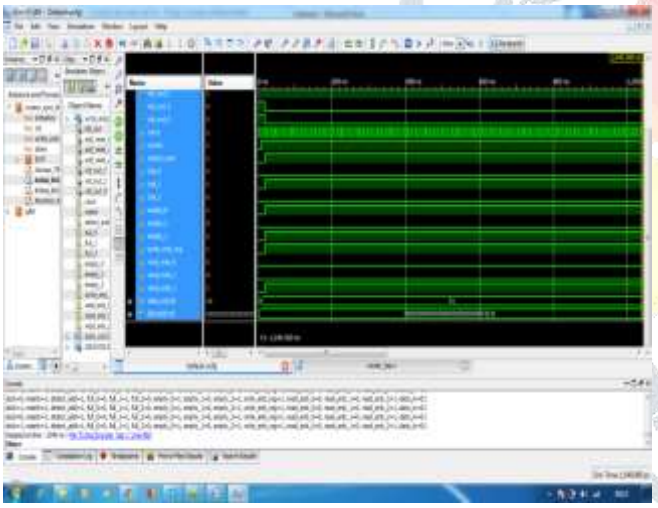

figure 12: Simulation Waveform of Proposed NOC Router

Figure 13: RTL Schematic of Proposed NOC Router

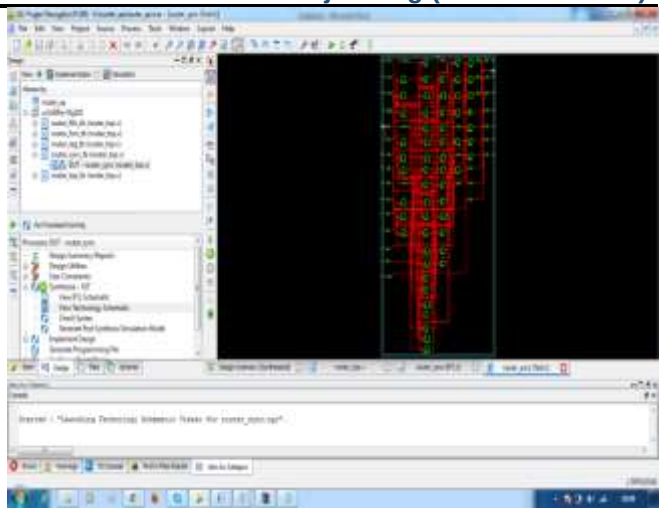

Figure 14: Technology Schematic of Proposed NOC Router

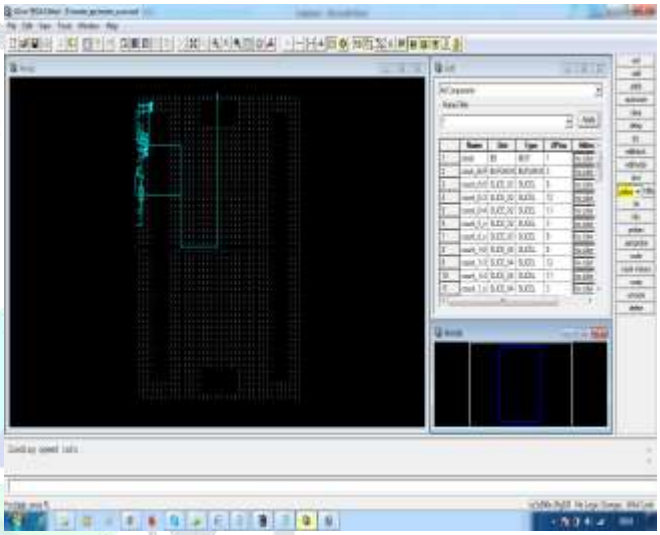

Figure 15: FPGA Implementation of Proposed NOC Router

Table - I: Comparison Table of NOC Routers

Parameters	Tree Based Multicast NOC Design	IPM and OPM based Multicast NOC Design	Proposed Multicast NOC Design
No. of slices	485 out of 4656	16 out of 4656	19 out of 4656
No. of 4-input LUTs	481 out of 9312	31 out of 9312	45 out of 9312
Combinational Path Delay	8.153ns	-	6.031ns
Average Fanout	3.87	2.90	3.12
Logic Power	0.00002	0.00004	0.00004
Signal Data Power	0.00008	0.00007	0.00002
I/O Power	0.00315	0.00843	0.00324

From Table I, it is clearly observed that the average fanout of data is increased for proposed design, the signal data power reduces by 71.4% and I/O Power reduces by 61.5%. But it slightly increases the area occupied so as to support both synchronous and asynchronous data transmissions. Also the delay reduces by 26% when compared to existing design.

5. CONCLUSION

The key challenge for modern NoCs is to efficiently support new traffic patterns, common in advanced parallel architectures eg., Multicast (i.e., one-to-many) destinations, widely used in the parallel computing domain i.e, in cache coherence protocols, to send write invalidates to multiple cores, e.g., in *HyperTransport* and *Token Coherence* protocols, in shared operand networks, to deliver operands to multiple instructions; and in multithreaded applications, to notify barrier synchronization to multiple processors. The applications of it include in emerging NoC technologies like wireless, surface-wave, photonic, and CDMA in applications like neuromorphic computing, both in spiking neural networks (SNNs) as well as in deep convolutional neural networks. A parallel multicast asynchronous NoC with a 2-D mesh topology is verified against the existing parallel tree based routing and the proposed NoC which supports both synchronous and asynchronous transmissions of data packets with the use of a continuous-time replication strategy, where the flits of a multicast packet are routed through the distinct outputs of the router according to each output's own rate, in parallel, and in continuous time. The results prove that the average fanout of data is increased for proposed design, the signal data power reduces by 71.4% and I/O Power reduces by 61.5%. But it slightly increases the area occupied so as to support both synchronous and asynchronous data transmissions. Also the delay reduces by 26% when compared to existing design.

REFERENCES

[1] R. Marculescu, U. Y. Ogras, L.-S. Peh, N. E. Jerger, and Y. Hoskote, "Outstanding research problems in NoC design: System, microarchitecture, and circuit perspectives," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 28, no. 1, pp. 3–21, Jan. 2009.

[2] M. Koibuchi, H. Matsutani, H. Amano, and T. M. Pinkston, "A lightweight fault-tolerant mechanism for network-on-chip," in *Proc. NOCS*, Apr. 2008, pp. 13–22.

[3] K. Goossens and A. Hansson, "The aethereal network on chip after ten years: Goals, evolution, lessons, and future," in *Proc. DAC*, Jun. 2010, pp. 306–311.

[4] D. Bertozzi, G. Dimitrakopoulos, J. Flich, and S. Sonntag, "The fastevolving landscape of on-chip communication: Selected future challenges and research avenues," *Des. Automat. Embedded Syst.*, vol. 19,pp. 59–76, Mar. 2015.

[5] N. E. Jerger, L.-S. Peh, and M. Lipasti, "Virtual circuit tree multicasting: A case for on-chip hardware multicast support," in *Proc. ISCA*, Jun. 2008, pp. 229–240.

[6] S. Abadal, A. Mestres, E. Alarcón, A. Cabellos-Aparicio, and R. Martínez, "Multicast on-chip traffic analysis targeting manycore NoC design," in *Proc. PDP*, Mar. 2015, pp. 370–378.

[7] T. Krishna and L.-S. Peh, "Single-cycle collective communication over a shared network fabric," in *Proc. NoCS*, Sep. 2014, pp. 1–8.

[8] D. Xiang, K. Chakrabarty, and H. Fujiwara, "A unified test and faulttolerant multicast solution for network-on-chip designs," in *Proc. ITC*, Nov. 2016, pp. 1–9.

[9] P. Abad, V. Puente, J. A. Gregorio, and P. Prieto, "Rotary router: An efficient architecture for CMP interconnection networks," in *Proc. Int. ymp. Comput. Architecture*, Jun. 2007, pp. 116–125.

[10] A. Adriahantenaina and A. Greiner, "Micro-network for SoC: Implementation of a 32-port SPIN network," in *Proc. Des., Autom. Test Eur. Conf.*, Mar. 2003, pp. 1128–1129.

[11] M. A. Al Faruque, T. Ebi, and J. Henkel, "Run-time adaptive on-chip communication scheme," in *Proc. Int. Conf. Comput.-Aided Des.*, San Jose, CA, 2007, pp. 26–31.

[12] A. M. Amory *et al.*, "A scalable test strategy for network-on-chip outers," in *Proc. IEEE Int. Test Conf.*, Nov. 2005, pp. 591–599.

[13] F. Angiolini, P. Meloni, S. Carta, L. Benini, and L. Raffo, "Contrasting a NoC and a traditional interconnect fabric with layout awareness," in *Proc. Des., Autom. Test Eur. Conf.*, Mar. 2006, pp. 124–129.

[14] F. Angiolini, D. Atienza, S. Murali, L. Benini, and G. De Micheli, "Reliability support for on-chip memories using networks-on-chip," in *Proc. Int. Conf. Comput. Des.*, Oct. 2006, pp. 389–396.

[15] G. Ascia, V. Catania, and M. Palesi, "Multi-objective mapping for meshbased NoC architectures," in *Proc. Int. Conf. Hardware-Softw. Codesign Syst. Synthesis*, Sep. 2004, pp. 182–187.

[16] W. Bainbridge and S. Furber, "Delay insensitive system-on-chip interconnect using 1-of-4 data encoding," in *Proc. Int. Symp. Asynchronous Circuits Syst.*, Mar. 2001, pp. 118–126.

[17] J. Balfour and W. J. Dally, "Design tradeoffs for tiled CMP on-chip networks," in *Proc. Int. Conf. Supercomputing*, Jun. 2006, pp. 187–198.

[18] N. Banerjee, P. Vellank, and K. S. Chatha, "A power and performance model for network-on-chip architectures," in *Proc. Des., Autom. Test Eur. Conf.*, Feb. 2004, pp. 1250–1255.

[19] T. A. Bartic *et al.*, "Highly scalable network on chip for reconfigurable systems," in *Proc. Int. Symp. Syst.-on-Chip*, Nov. 2003, pp. 79–82.

[20] P. Beerel and M. E. Roncken, "Low power and energy efficient asynchronous design," *J. Low Power Electron.*, vol. 3, no. 3, pp. 234–253, Dec. 2007.

[21] 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," in *Proc. Int. Symp. Asynchronous Circuits Syst.*, May 2005, pp. 54–63.

[22] E. Beigne, F. Clermidy, S. Miermont, and P. Vivet, "Dynamic voltage and frequency scaling architecture for units integration within a GALS NoC," in *Proc. Int. Symp. Netw. Chip*, 2008, pp. 129–138.

[23] L. Benini and G. De Micheli, "Networks on chips: A new SoC paradigm," *Computer*, vol. 35, no. 1, pp. 70–78, Jan. 2002.

[24] D. Bertozzi *et al.*, "NoC synthesis flow for customized domain specific multiprocessor systems-on-chip," *IEEE Trans. Parallel Distrib. Syst.*, vol. 16, no. 2, pp. 113–129, Feb. 2005.

[25] D. Bertozzi, L. Benini, and G. De Micheli, "Error control schemes for on-chip communication links: The energy–reliability tradeoff," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 24, no. 6, pp. 818–831, Jun. 2005.

[26] D. Bertozzi, A. Acquaviva, D. Bertozzi, and A. Poggiali, "Supporting task migration in multi-processor systems-on-chip: A feasibility study," in *Proc. Des., Autom. Test Eur. Conf.*, Mar. 2006, pp. 1–6.

[27] P. Bhojwani, J. D. Lee, and R. Mahapatra, "SAPP: Scalable and adaptable peak power management in NoCs," in *Proc. Int. Symp. Low Power Electron. Devices*, Aug. 2007, pp. 340–345.

[28] T. Bjerregaard and J. Sparso, "A router architecture for connectionoriented service guarantees in the MANGO clockless network-on-chip," in *Proc. Des., Autom. Test Eur. Conf.*, Mar. 2005, pp. 1226–1231.

[29] T. Bjerregaard and S. Mahadevan, "A survey of research and practices

of network-on-chip," *ACM Comput. Surv.*, vol. 38, no. 1, pp. 1–51, Mar. 2006.

[30] C. Bienia, S. Kumar, J. P. Singh, and K. Li, "The PARSEC benchmark

suite: Characterization and architectural implications," Princeton Univ., Princeton, NJ, Tech. Rep. TR-811-08, Jan. 2008.

[31] P. Bogdan, T. Dumitras, and R. Marculescu, "Stochastic communication: A new paradigm for fault-tolerant networks-on-chip," *Hindawi VLSI Design*, Feb. 2007.

[32] E. Bolotin, I. Cidon, R. Ginosar, and A. Kolodny, "QNoC: QoS architecture and design process for network on chip," *J. Syst. Architecture: EUROMICRO J.*, vol. 50, no. 2/3, pp. 105–128, Feb. 2004.

[33] E. Bolotin, I. Cidon, R. Ginosar, and A. Kolodny, "Routing table minimization for irregular mesh NoCs," in *Proc. Des., Autom. Test Eur. Conf.*, Apr. 2007, pp. 942–947.

[34] J. W. van den Brand, C. Ciordas, K. Goossens, and T. Basten, "Congestion-controlled best-effort communication for networks-onchip," in *Proc. Des., Autom. Test Eur. Conf.*, Apr. 2007, pp. 948–953.

[35] G. Campobello, M. Castano, C. Ciofi, and D. Mangano, "GALS networks on chip: A new solution for asynchronous delay-insensitive links," in *Proc. Des., Autom. Test Eur. Conf.*, Mar. 2006, pp. 160–165.

[36] L. P. Carloni, K. L. McMillan, and A. L. Sangiovanni-Vincentelli, "Theory of latency-insensitive design," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 20, no. 9, pp. 1059– 1076, Sep. 2001.

[37] V. Catania, R. Holsmark, S. Kumar, and M. Palesi, "A methodology for

design of application specific deadlock-free routing algorithms for NoC systems," in *Proc. Int. Conf. Hardware-Softw. Codesign Syst. Synthesis*, Oct. 2006, pp. 142–147.

[38] J. Chan and S. Parameswaran, "NoCEE: Energy macro-model extraction methodology for network on chip routers," in *Proc. Int. Conf. Comput.- Aided Des.*, Nov. 2005, pp. 254–259.

[39] S. C. Chan, K. L. Shepard, and P. J. Restle, "Design of resonant global clock distributions," in *Proc. Int. Conf. Comput. Des.*, Oct. 2003, pp. 248–253.

[40] M. F. Chang *et al.*, "CMP network-on-chip overlaid with multi-band RFinterconnect," in *Proc. Int. Symp. High-Performance Comput. Architecture*, Feb. 2008, pp. 191–202.

[41] T. Chelcea and S. M. Nowick, "Robust interfaces for mixed-timing systems with application to latency-insensitive protocols," in *Proc. Des. Autom. Conf.*, Jun. 2001, pp. 21–26.