Extension of the LISNoC (Network-on-Chip)*

Rakotojaona Nambinina, Daniel Onwuchekwa, Sabikun Nahar, Darshak Sheladiya, Roman Obermaisser
Chair for Embedded Systems , University of Siegen
Andrianoelisoa.Rakotojaona@uni-siegen.de

***Abstract*—Over the years, Network-on-Chip (NoC) has undergone a rapid evolution which urges the performance of NoCs to be analyzed thoroughly. Several NoC solutions exist, but the performance of these NoCs are tied to application requirements. Therefore, it has become practical to extend existing NoCs to satisfy particular application requirements. The LISNoC is one such NoCs that is open-source and provides an easily adaptable implementation to extend its features to satisfy different application requirements. In order to satisfy several key features such as the wormhole switching, source-based routing, support for a virtual channel and an Advanced eXtensible Interface (AXI) based network interface, the LISNoC is improved. Mainly, this work extends the LISNoC to support source-based routing and equips the LISNoC with a new AXI-based network interface.**

***Index Terms*—Network on chip, LINoC, Network interface, Source base routing**

I. Introduction

Nowadays, there is a trend in chip design where components with millions of transistors are integrated and made to operate efficiently. The System-on-chip (SoC) designs implement such an integrated solution for various complex applications. One of the essential considerations in SoC design is the communication architecture between different components. Most of the communication architectures in current SoCs are bus-based. However, the bus architecture has its inherent limitations, and these include large load per data bus line, long delay for data transfer, large energy consumption, and low bandwidth [8]. In addition, latency, noise, power consumption, signal integrity, and synchronization are now seriously considered issues in SoC design. As a result, Networkon-Chip (NoC) has been proposed in [7] to exchange messages between communicating components within an SoC. NoCs implement a high-performance communication infrastructure and is a means for integrating many IP cores for SoC implementation. In addition, the NoC paradigm uses a routerbased network for packet-switched communication between on-chip cores, and it provides the possibility for reusability since the communication infrastructure can be easily integrated to a new product. The primary goal of the communication centric design and NoC paradigm is to reduce manufacturing difficulties, transfer delay, wiring issues, power consumption and reliability to enhance the design productivity and performance [17], [23], [24]. Also, three critical challenges for NoC are power, latency, and CAD compatibility [16]. Several key features are considered when designing an NoC to overcome these challenges. These features include the switching technique, routing algorithm, support for virtual channels, and on-chip interconnection architecture.

Several NoC switching techniques exist, such as store-andforward, virtual cut-through, and wormhole switching [20]. The wormhole switching has proven desirable due to its low latency, implementation simplicity, and performance for a relatively low traffic workload [15].

Generally, routing algorithms can be categorized into three main groups: deterministic, Adaptive, and Stochastic. In terms of implementation, routing algorithms can be implemented by two different techniques, such as source-based routing and distributed routing [20]. Source-based routing has shown to minimize congestion on links and reduce packet delay. In addition, the inclusion of virtual channels in an NoC design provides a way to avoid deadlock and optimize the bandwidth of physical channels. Several protocols for On-chip interconnection exists, such as IBM's core connect [9], Silicore corporation WISHBONE [19], and the Advanced eXtensible Interface (AXI) protocol [21] . The AXI is considered in this work due to it's widespread use, simplicity, and wide compatibility features.

There are currently several NoCs such as LISNoC [5], ANOC [3], Nostrum [12], GALS NoC [6], Hermes [13], AEthereal [2], and SPIN [1]. None of these NoC fulfils the following collective requirements; open source, wormhole switching, source-based routing, virtual channel support, support for mesh topology, and AXI interface. However, among all the NoCs mentioned above, only AEthreal supports source-based routing. Nevertheless, the LISNoC is open source and provides an easily adaptable implementation. This work, therefore, uses the LISNoC and contributes the following to satisfy the NoC challenges above:

- Extends the LISNoC to support source-based routing in which the packet format is also modified.
- Extends the LISNoC with an Advanced eXtensible Interface (AXI) based network interface.
  AXI protocol provides a point to point interconnection to avoid bus sharing and therefore allow higher bandwidth and lower latency.

GEFÖRDERT VOM Bundesministerium für Bildung und Forschung

The remainder of this paper is organized as follows. Section II discusses the related work, and section III describes the NoC. Section IV discusses the extension of LISNoC . Section V discusses the experimental setup and results , and finally, section VI concludes the paper.

II. RELATED WORK

The LISNoC is introduced in [5], and it uses wormhole packet switching for flit transmission within the NoC. The LISNoC's operation has been used three flow control signals: flit, valid, and ready. It supports virtual channels, and virtual channels are defined as flow control signals. The LISNoC, on the other hand, only supports Mesh topology and lacks a network interface. As a result, we extended the LISNoC with source-based routing and developed and Network Interface (NI) that used AXI to interconnect the processing elements with the NI.

The work carried out in [3] has proposed an Asynchronous NOC architecture that combines Quality of Service (ANOC) and Transaction-Level Modeling. The data transmission through the network uses the wormhole packet switching technique. In addition, it uses virtual channels to increase efficiency and reduces latency for prioritised packets. ANOC supports the 2D mesh network, and each node is connected to a network interface that contains a Globally asynchronous locally synchronous (GALS) interface to perform synchronisation between the synchronous and asynchronous domains. In contrast, our design relies on source-based routing to choose the path to a destination, ensuring minimal congestion and low packet latency. In addition, our design provides an AXI-based network interface connection.

In [13], the authors reviewed the state of the art in Network-on-Chip. An infrastrure called Hermes is described, that implements packet switching, mesh topology, and related interconnection architectures. Herme's main component is a switch with five bidirectional ports that is linked to four other switches and a local IP core. Input queuing is used by the switch, which uses an XY routing algorithm. The primary goal of the design was to create a small size switch. The design validation of the Hermes switch and the network-on-chip is presented in the paper. In contrast , we use source-based routing to ensure low packet latency and minimal congestion. In addition, unlike the Hermes NoC our proposed design supports the use of a virtual channel that can be used to minimized the latency in the NoC.

The GALS NoC architectures is introduced in [6]. To solve the clock distribution problem in GALS NoC, asynchronous routers are linked to synchronous blocks. In this architecture, the routing method is wormhole packet switching with an XY routing algorithm. Mesh topology is supported by GALS NoC. Unlike the GALS NoC, our design selects the best path to the destination based on predefined schedule using source based routing algorithm, ensuring conflict of messages and reduced the congestion and the delay when transmitting messages withing the NoC.

The AEthereal NOC architecture is introduced in [2]. To reduce buffering costs, data is buffered using the wormhole switching technique. Virtual channel support is used to improve efficiency and ensure low latency and minimum bandwidth. Mesh topology is supported by AEthereal. The AEthereal includes a Network Interface (NI), which divides the system into a fixed kernel and variable shells. Each NI shell converts read and write transactions from a specific IP protocol. There is no AXI interface linked to NI. Instead, it employs a Multi-layer AHB (ML-AHB) interconnection architecture, which allows for parallel access between multiple masters and slaves, increasing overall bus bandwidth and flexibility in the system architecture.

SPIN (Scalable Programmable Interconnection Network) NOC architecture [1] is introduced by Adriahantenaina et al. Two on-chip communication templates are examined and compared in this paper, based on bus and NoC approaches. In SPIN, packets are forwarded as soon as a router receives their headers, which is a wormhole switching technique. SPIN has adaptive and distributed routing, and SPIN supports the fat-tree topology. A tree structure holds routers on nodes and terminals on leaves, except that every node contains identical fathers. RSPIN router is the basic building block of a SPIN network, containing eight ports, each with two input and output channels.

TABLE I
COMPARATIVE VIEW FOR DIFFERENT NETWORK ON CHIPS

NoCs	Switching Techniques	Routing Algorithm	Virtual Channel	Topology	Network Interface
LISNoC	Wormhole	Dimension order	Yes	Mesh	NA
ANoC	Wormhole	Dimension order	Yes	Mesh	Contains GALS interface
Hermes	Wormhole	Dimension order	No	Mesh	NA
GALS NoC	Wormhole	XY	Based on priority scheduling	Mesh	No
AEthereal	Wormhole	Source based	Yes	Mesh	AMBA High speed bus
SPIN	Wormhole	Adaptive and distributed	No	Fat-tree	No
Extended LISNoC	Wormhole	Source based	Yes	Mesh, Torus	Yes

III. NETWORK ON CHIP

Network-on-Chip (NoC) is a network-based communication system that is built on an integrated circuit, such as the System-on-Chip. Figure 1 depicts an example of a NoC based MPSoC. It is made up of several components such as routers, links, Processing ELements (PEs), and storage elements such as memory blocks [20]. The Network Interfaces connect all of the IPs to

network adapters (NIs). By enabling PEs, communication via the NoC is achieved. These PEs send and receive packets over a network of switches that are linked together by physical links or channels. Links are typically made up of two uniderectional, point to point buses that run in opposite directions.

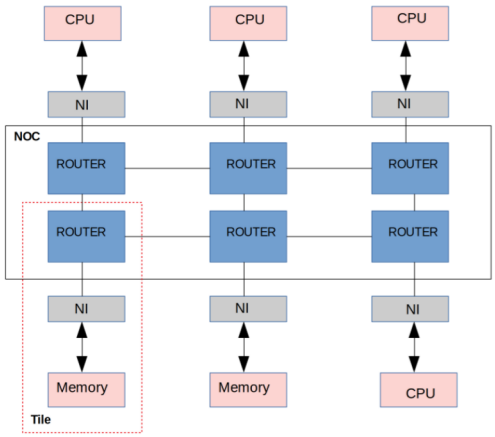

Fig. 1. NoC based MPSoC

In NoC, data packet transmission takes place between source PEs and sink PEs. Depending on the router's decision, data packet transmission is forwarded on the network step by step. The router has switches and buffers, and the packet is first received and stored in the input buffer. Flits are temporarily stored in input/output buffers. The output port allocator selects the output port for each flit/packet. The physical connection is made by the switch via a link from the input port to the output port and the control logic is in charge of overall synchronization. Routers and switches serve as network connectivity devices.

IV. LISNoC EXTENSION

A. *LISNoC*

LISNoC is an open source Network-on-Chip implemented in Verilog mainly used for academic purpose [4]. The main features of LISNoC include support for a virtual channel, flexible router configuration, wormhole routing and strict ordering, and round-robin arbitration for link multiplexing [5]. The LISNoC flow control consists of three signals, namely, Flit, Valid, and Ready signal. The transfer of flits within the LISNoC occurs when the valid and ready signals are high during a clock edge. Virtual channels are represented by the flow control signals in the LISNoC. Data is transmitted as messages which are split into packets. These packets are divided into flow control units (flits) and physical units (phits). Almost all NoCs follow packet-based communication. Data packets have a header portion where all the information regarding the destination processing elements is stated. The amount of a packet transmitted in one clock is called the flit. Wormhole routing is used for packet routing. Wormhole routing does not store any packet. Frames are forwarded to the correct port based on the information contained in the head flit.

B. *Architecture of Extended LISNoC*

The LISNoC is extended with an AXI capable network interface and source-based routing capability. Figure 2, shows the detailed design of the NI, which stores message into the register unit and sends it to the destination address. The NI is designed as a bridge between a core and the NoC switching fabric. The NI is used to connect the PEs to the NoC. The NI is responsible for packetizing and depacketizing the messages from the PEs.

In Figure 2, the wrapper is the fundamental component for different PEs. Many on-chip networks may not be developed based on the IP cores master interface, a slave wrapper is required to communicate between the IP core, and its associated router [22], [11]. Communications occur via a transactional protocol, which utilizes the master interface of IP core to transmit read and write commands at an address. A slave module that receives those data executes them and may reply with a response message including the status of the operations [10], [4]. As shown in Figure 2, AXI4 Full slave interface is used as a wrapper that allows burst particular mode transactions. AXI4 Lite is not used as it allows only 1 data transfer per transaction. The advantage of burst transfers is that the bandwidth is employed as effectively as possible since the initial address is only sent with some information regarding the burst [18].

The Register Unit (RU) in the block diagram illustrates the Memory Map in NI. It consists of 7 Dual Port RAMs. This register unit aims to store Tx (Transmit) message and NoC configuration information(e.g. Traffic ID, a Destination address, Number of flits) into the memory from the host side. It also stores the Rx (receive) messages and the status information(e.g. Source busy, Sink busy) coming from the receiver. The Trigger FSM is a state machine responsible for controlling the trigger signal to inject messages into the network. The Configuration FSM is a state machine responsible for configuring the NoC events on different memory location. The status Update block receives data from Multiplexer and passes it to the status register memory location. The Source FSM implements a state machine used to transfer the destination address and messages to the NoC. It also performs the packetization of the message. The Sink FSM receives messages from the NoC and performs depacketization.

V. EXPERIMENTAL SETUP AND RESULTS

To evaluate the extended LISNoC implementation, we have performed four sets of experiments with a different number of hops and flit sizes. The experiment begins with a 2x2 mesh network for the NoC. One virtual channel is configured, and the system is fed with a 100 MHz clock. Different flit sizes 8, 16, 24, 32, 40, 48 were taken into account. A message source and sink are connected to different routers with two hops separating them. The setup then utilises a Global Time Base (GTB) implementation to measure the end-to-end latency

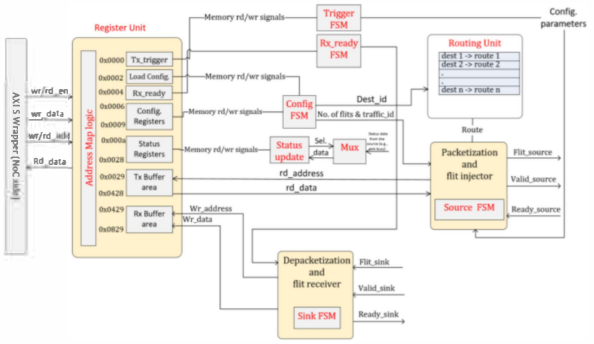

Fig. 2. Architecture of Extended LISNoC

of messages from a source to sink. After which, the experiment is repeated for a 3x3, 4x4, and 5x5 mesh network. The results of the experiment are shown in Table II. Table II shows the impact of the flit sizes on latency. It captures the number of clock cycles required to process the stated hops. A plot of average latency against flit size is shown in figure 3. The results show the existence of a linear relationship between the flit sizes and average latency. It observed that the extended LISNoC for a two-hop configuration needs 13.5 clock cycles to process a flit size of 8.

TABLE II
EXPERIMENT RESULTS

Number of Flits	Latency [ns] (clock cycles)2x2 mesh (2 Hops)	Latency [ns] (clock cycles) 3x3 mesh (3 Hops)	Latency [ns] (clock cycles) 4x4 mesh (4 Hops)	Latency [ns] (clock cycles) 5x5 mesh (5 Hops)
8	135 (13.5)	165 (16.5)	195 (19.5)	225 (25.5)
16	215 (21.5)	245 (24.5)	275 (27.5)	305 (30.5)
24	295 (29.5)	325(32.5)	355 (35.5)	385 (38.5)
32	375(37.5)	405 (40.5)	435(43.5)	465 (46.5)
40	455 (45.5)	485 (48.5)	515(51.5)	545 (54.5)
48	535 (54.5)	565 (56.5)	595 (59.5)	625 (62.5)
56	615 (61.5)	645 (64.5)	675 (67.5)	705 (70.5)
64	695(69.5)	725 (72.5)	755 (75.5)	785 (78.5)
72	775 (77.5)	805 (80.5)	835 (83.5)	865 (86.5)

VI. CONCLUSION

This work extended the LISNoC with an AXI-based network interface and a source-based routing algorithm. A new packet format was defined for the LISNoC to realise the source-based routing scheme. The results of the extended LISNoC show a linear relationship between the average latencies and the increment in flit sizes. For a 2x2 mesh network, 13.5 clock cycles was required to transmit a flit size of 8. It is planned to reduce the number of clock cycles required for each hop to transmit messages and extends the LISNoC with adaptive features and time triggered systems for safety relevant application in future works.

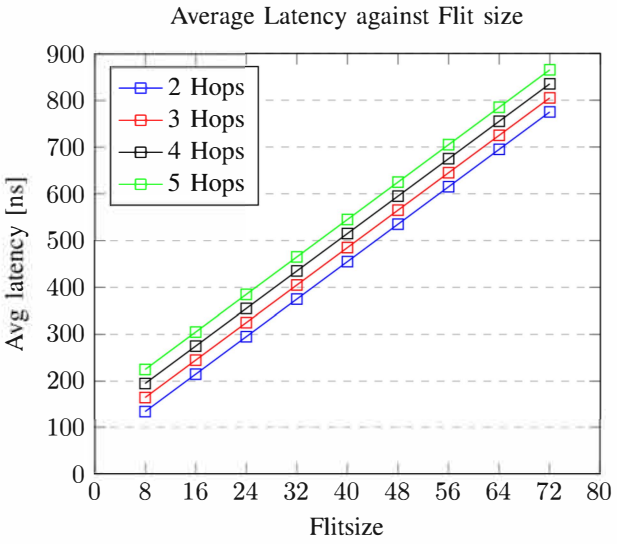

Fig. 3. Average Latency against Flit size

ACKNOWLEDGMENT

This work has been supported by the research project FRACTAL in part by the EC under grant number 877056 and the BMBF under grant number 16MEE015K.

REFERENCES

[1] Adrijean Adriahantenaina, Herve Charlery, Alain Greiner, Laurent Mortiez, and Cesar Albenes Zeferino. Spin: a scalable, packet switched, on-chip micro-network. In 2003 Design, Automation and Test in Europe Conference and Exhibition, pages 70–73. IEEE, 2003.

[2] Chris Bartels, Jos Huisken, Kees Goossens, Patrick Groeneveld, and Jef Van Meerbergen. Comparison of an Æthereal network on chip and a traditional interconnect for a multi-processor dvb-t system on chip. In 2006 IFIP International Conference on Very Large Scale Integration, pages 80–85, 2006.

[3] Edith Beigne, Fabien Clermidy, Pascal Vivet, Alain Clouard, and Marc Renaudin. An asynchronous noc architecture providing low latency service and its multi-level design framework. In 11th IEEE International Symposium on Asynchronous Circuits and Systems, pages 54–63. IEEE, 2005.

[4] lisnoc. http://www.lisnoc.org/. (accessed: 01.03.2021).

[5] Kazem Cheshmi, Jelena Trajkovic, Mohammadreza Soltaniyeh, and Siamak Mohammadi. Quota setting router architecture for quality of service in gals noc. In 2013 International Symposium on Rapid System Prototyping (RSP), pages 44–50. IEEE, 2013.

[6] Kazem Cheshmi, Jelena Trajkovic, Mohammadreza Soltaniyeh, and Siamak Mohammadi. Quota setting router architecture for quality of service in gals noc. In 2013 International Symposium on Rapid System Prototyping (RSP), pages 44–50. IEEE, 2013.

[7] Hsin-Chou Chi and Jia-Hung Chen. Design and implementation of a routing switch for on-chip interconnection networks. In Proceedings of 2004 IEEE Asia-Pacific Conference on Advanced System Integrated Circuits, pages 392–395. IEEE, 2004.

[8] Dakshina Dasari. Timing Analysis of Real-Time Systems Considering the Contention on the Shared Interconnection Network in Multicores. PhD thesis, Universidade do Porto (Portugal), 2013.

[9] Amit Goel and William R Lee. Formal verification of an ibm coreconnect processor local bus arbiter core. In Proceedings of the 37th Annual Design Automation Conference, pages 196–200, 2000.

[10] Pierre Guerrier and Alain Greiner. A generic architecture for on-chip packet-switched interconnections. In Design, Automation, and Test in Europe, pages 111–123. Springer, 2008.

[11] Theodore Marescaux, Erik Brockmeyer, and Henk Corporaal. The ´ impact of higher communication layers on noc supported mp-socs. In First International Symposium on Networks-on-Chip (NOCS'07), pages 107–116. IEEE, 2007.

[12] Mikael Millberg, Erland Nilsson, Rikard Thid, and Axel Jantsch. Guaranteed bandwidth using looped containers in temporally disjoint networks within the nostrum network on chip. In Proceedings Design, Automation and Test in Europe Conference and Exhibition, volume 2, pages 890–895. IEEE, 2004.

[13] M Fernando , C Ney , ,M Aline , M Leandro, and O Luciano . Hermes: an infrastructure for low area overhead packet- switching networks on chip. Integration, 38(1):69–93, 2004.

[14] Tammy Noergaard. Embedded systems architecture: a comprehensive guide for engineers and programmers. Newnes, 2012.

[15] Ioannis Nousias and Tughrul Arslan. Wormhole routing with virtual channels using adaptive rate control for network-on-chip (noc). In First NASA/ESA Conference on Adaptive Hardware and Systems (AHS'06), pages 420–423. IEEE, 2006.

[16] John D Owens, William J Dally, Ron Ho, DN Jayasimha, Stephen W Keckler, and Li-Shiuan Peh. Research challenges for on-chip interconnection networks. IEEE micro, 27(5):96–108, 2007

[17] Shafi Patel, Parag Parandkar, Sumant Katiyal, and Ankit Agrawal. Exploring alternative topologies for network-on-chip architectures. BIJIT-BVICAM's International Journal of Information Technology, 3(2), 2011.

[18] Magdy Saeb. The stone cipher-192 (sc-192): A metamorphic cipher. The International Journal on Computers and Network Security (IJCNS), 1(2):1–7, 2009.

[19] Mohandeep Sharma and Dilip Kumar. Wishbone bus architecture-a survey and comparison. arXiv preprint arXiv:1205.1860, 2012.

[20] Konstantinos Tatas, Kostas Siozios, Dimitrios Soudris, and Axel Jantsch. Designing 2D and 3D network-on-chip architectures. Springer, 2014.

[21] AXI Xilinx. Reference guide, ug761 v13. 1, 2011. URL http://www. xilinx. com/support/documentation/ip documentation/ug761 axi reference guide. pdf, 2011.

[22] Sungjoo Yoo, Gabriela Nicolescu, Damien Lyonnard, Amer Baghdadi, and Ahmed A Jerraya. A generic wrapper architecture for multiprocessor soc cosimulation and design. In Proceedings of the ninth international symposium on Hardware/software codesign, pages 195–200, 2001.

[23] Ashokkumar, N., P. Nagarajan, and P. Venkatramana. "3D (dimensional)—Wired and wireless network-on-chip (NoC)." In Inventive Communication and Computational Technologies, pp. 113- 119. Springer, Singapore, 2020.

[24] R. Nambinina, D. Onwuchekwa, H. Ahmadian, D. Goyal and R. Obermaisser, "Time-Triggered Frequency Scaling in Network-on-Chip for Safety-Relevant Embedded Systems*," 2021 International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON), 2021, pp. 1-7, doi: 10.1109/SMARTGENCON51891.2021.9645782.