Network Interface Design with Advance Network Functionalities for NoC

[1]Rekha, [2]Ajeesh A V
[1] PG Student [VLSI and ES] [2] Assistant professor,
Department of ECE, TKM Institute of Technology, Kollam
[1]rekhasunil1123@gmail.com, [2] ajeeshavi@gmail.com

Abstract: **Network-on-chip (NoC) is an evolving design technology used for growing a packet switched communication infrastructure which contain hundreds of intellectual property (IP) cell in single mutli-processor system on chip (MPSoC). Network interface (NI) is one of the building block which make intellectual property macrocell to be associated to on-chip communication backbone. This work propose the design of network interface macrocell which take care of data packetization/ depacketization to and from NoC and promises successful end to end data delivery. It also include the advanced network functionalities like store and forward (S & F) transmission, error management, ordering handling, security with hardware support. The basic characteristics like flit size, IP bus data size, payload & header FIFO size and frequency & size conversion support can be configured. The features can be added on the top of a basic network interface (NI) core. The work is coded by using verilog and is simulated by Xilinx ISE 13.2.**

Index Terms— **Network on chip, Network interface, Network Functionalities.**

I. INTRODUCTION

Network-On-Chip (NoC) is an emerging design paradigm which builds a scalable packet-switched communication infrastructures, connecting hundreds of IP cells in a Multi-Processor System-on-Chip (MPSoC). In a System-on-a-Chip (SoC), to designing the communication subsystem between IP cores approach a Network-On-Chip. IP Cores can be general purpose processors, memories or peripherals, Application Specific Instruction set Processors (ASIP) and Digital Signal Processors (DSP). NoCs gives a methodology for designing an interconnect architecture which is independent from the connected cores. Parallelization, scalability and reusability all benefit in the design flow can be obtained from this approach .

NoCs will also be a key component for the success of future 3D SoC. Synchronous and asynchronous clock domains or unclocked asynchronous logic can be span by NoCs. It applies networking theory and compared to other designs, it improves the scalability of SoCs and the power efficiency of complex SoCs. Traditionally, integrated circuit (ICs) has been designed with dedicated point-to-point connections, each signal are dedicated with one wire. For large designs, this has several limitations from a physical design point of view. The wires occupy large area of the chip . in nanometer CMOS technology ,an interconnects gives priority for both performance and dynamic power dissipation. Due to the signal propagation in wires across the chip which needs multiple clock cycles. NoC links can be reduce the complexity of designing wires for predictable speed, reliability, power, noise etc.

An NoC based on Spidergon topology, a Ring structure which reduce network crossing latency an additional across link for each node is needed, highlighting its hardware building blocks: connected IP cores, NIs(Network Interface), links and Routers (R).

A. Network Interface

Network Interface (NI) is a key element of a NoC, which allows the IP cores to be connected to the on-chip communication. The NIs are the peripheral building blocks of the NoC. Basically, the NI will take care for packetization/depacketization to and from the NoC. It provides protocol abstraction by encoding the header of the packets, all data are guarantee to be successfully deliver between IP cores (transport layer). NoC packet includes a header and payload (data) which split physically in units called flits. All flits of a packet are passes through the same path over the network. The header field consist of both Network Layer Header (NLH), according to the node map network configuration whose content which determined by the NI and Transport Layer Header (TLH) containing information used by the NIs for end-to-end transaction of data.

In this paper, a Network interfaces architecture offers more integrating features such as store and forward transmission, error management unit, order handler and security. NI is intended for systems on chip (SoC) and it adds some network functionalities which provide the data transmission in a better way. Such NI has been conceived as a scalable architecture. The advanced features can be added on the top of a basic NI core, implementing data packetization between the connected IP core and the NoC.

II. LITERATURE REVIEW

For the performance improvement in NoC, many approaches have been introduced. The paper which discussed about a high speed Network Interface for packet based NoC[1]. NI which can support serial-link packet-based transmission model. It can reduce the link loading, provides transmit data in serial and avoid the data transmission skew. This approach has some demerit which consume more area and more power. A novel pipelined NI architecture between router of NOC and IPs. These network interfaces allow designers to send data from IPs to NOC, and vice versa with low latency [2]. It use AMBA AHB IPs standard at the IP side and use the most three used flow control in NoC. A new approach of NoC Interface known as Micronswitch Interface (MSI) [2] designed for message-passing communication on Micronmesh MPSoC platform with a Micron Message-Passing (MMP) protocol. The MSI provides mechanisms for fault-tolerant communication and efficient buffer management will be necessary for reliable and efficient operation for the MPSoCs. An AXI compliant Network Interface for NoC, which deals with the reordering problem and also support the adaptive routing [3]. A novel reordering mechanism based on look up table is discussed; it can guarantee globally ordering of the response transactions. The NI supports the master and slave core together and also support out-of-order and in-order transaction. Another approach presents a Spidergon topology in network-on-chip (NoC) infrastructures, where a configurable Network Interface (NI) macrocell to be integrated and addresses the problem of its functional verification[4]. In complex designs, the state explosion problem reduce model checking and the cost of theorem proving becomes forbidding. Such issues can be overcome, by a constrained-random coverage-driven approach is presented where customized to be applied to the novel NI as Design Under test (DUT).

In recent literature some NIs, that added to the basic IP-NoC interface functionalities some features such as order handling transactions, error transactions detection , secure memory access control , QoS management and NI programmability. However, the literature does not presents a design which integrating all the above advanced features which mention in the same NI with limited complexity overhead. The latest research frontier on NI architecture design aims at integrating more advanced networking functionalities features which directly support in hardware . The challenge in doing so to keeps NI power, latency overheads and area as low as possible with respect to the connected IP cores. To overcome the limits of the state of the art, this work presents the design and the characterization in deep submicron CMOS technology of a NI architecture directly implementing in hardware advanced networking features such as: store & forward transmission, error management, ordering handling, security. Such NI has been conceived as a scalable architecture. On top of a basic NI core, the advanced features can be added which implementing data packetization/depacketization and conversion of protocols, data size and frequency between the connected NoC and the IP core. The network interface can be configured to reach to the desired trade-off between circuit complexity and supported services.

III. PROPOSED SYSTEM

In a NoC infrastructure, numerous IP cores are used which is commonly classified into Master and Slave IPs : the former (e.g. a processing element) which generates request transactions and receives responses, the latter (e.g. a memory) which receives the requests and the proper responses are send back. Initiator NIs are connected to Master IPs and the target NIs are connected to slave IPs. Apart from the basic NoC functionalities, the advanced networking functionalities can be added on top of the core NI architecture which increase the further performance enchancement.

A. Design of core NoC Interface

The block diagram of modular (layered) NI architecture is shown in the Fig.1. It is a three layer architecture.In both NI types(Initiator and Target NI), both can be identified the Shell, the Kernel and NoC specific. Each one having its own peculiar functionality and interfaces. Initiator NI convert IP request transactions into NoC traffic and converts the packets received from NoC into IP response transactions. Target NIs present a mirrored architecture: requests are decoded from NoC and responses are encoded. Conversion features must be implemented in the two directions as shown in Fig.1 called request path (from Master to Slave IPs) and response path (from Slave to Master IPs) respectively.

The NI architecture showed in Fig.1 is split into three part Shell, Kernel and NoC interface respectively. The Shell part deal with the handshake rule and also builds the Network and Transport Layer headers, needed by successive NoC components (i.e., routers and target NIs) for forwarding the packet. It includes header encoder adopts gray encoding for the robustness. The TLH is a transport layer header which defines the source address and NLH is a network layer header which define a destination address. NI encodes both the TLH and the NLH. The kernel part which is used for buffering services. Shell which received the encoded data are stored in two FIFOs, an header FIFO (holds TLH and NLH) and a payload FIFO (holding actual data from the core). The Kernel is interfaced to the Shell by means of a FIFO like interface. It consist of Header FIFO , Payload FIFO and OFSM.

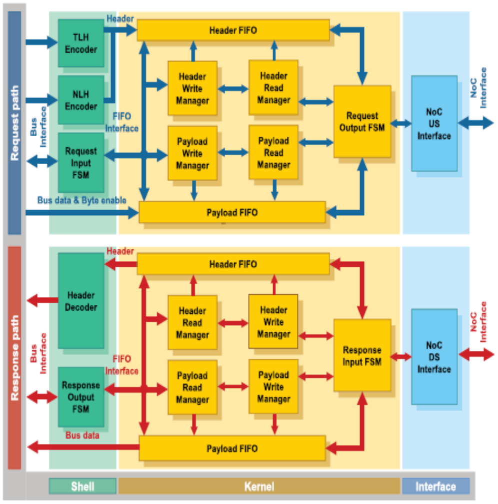

Fig.1 Main blocks in the NI micro-architecture

B. *Flow of Data in Network Interface*

The work follows the Fig.1 to design the core of NI. The Fig.2 shows the top module which defines the flow of data in NI Initiator.

IP Core is of the most critical components of SoC design is the integration of pre developed pieces of functionality called Intellectual Property (IP). These IP blocks can offer a huge differentiation to designers building SOC designs for various applications and helps reduce development cycle time significantly. Here, IP Core is designed as a memory where the data get store.

NI Module which describe the NI initiator structure and its sub module in detail based on the Fig.2. The network features added on top of the NI structure. The data given by the IP core through a handshaking rules to the shell part of NI. It includes security block, order handler, EMU, header encoder, FIFOs for header and payload. The address is checked by the security block and moves to order handler. After encoding the header, data will moves to kernel part which include both header and payload. The error management unit which check the addresses. Through FSMs, the Kernel is connected to the NoC interface stage. In the request path, an output FSM (OFSM) which reads the headers and payloads and according to the NoC protocol it converts them into packets. The Fig.3 shows the reading of header and payload information as per the credit and request signal. The finite state machine initiates and helps for the packing and unpacking of the messages. The OFSM control and schedule the transaction from NI to the NoC.

At the NoC interface, the physical link is concerned as the hardwired lines for the request path. It takes the packet which defines its flit_id and its deals with the credit management flow.In credit-based flow control, credit and valid signals are used. A flit is sent only when there is room enough to receive it, neither retransmission nor flit dropping is allowed. Since the upstream interface sends flits only if the connected downstream interface can receive them and there are no pending flits on the link wires. This approach allows the virtual channel for flit-level interleaving. So that same physical link can be share by separate virtual networks.

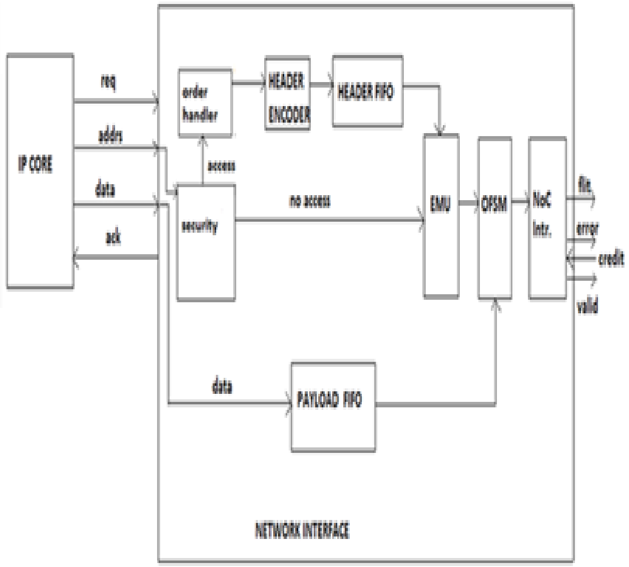

Fig.2 Insight of NI initiator request path

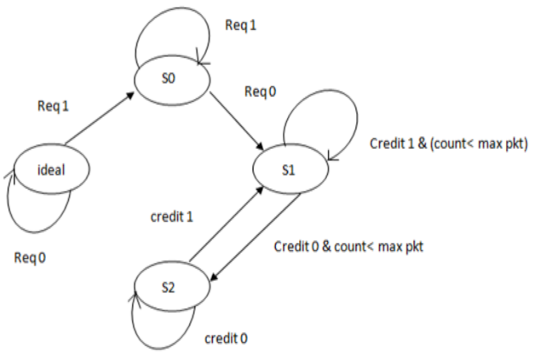

Fig.3 OFSM in NI

IV. ADVANCE FUNCTIONALITIES

The advance feature which is added on the top of the NI architecture. The superior network functionalities which includes:

A. *Store and Forward transmission*

Kernel FIFOs in both Request paths contain flits, either received from NoC and be decoded towards the IP bus or encoded from the bus level and to be transmitted over the interconnect. Behavior of a default NI is that a flit is extracted from the FIFO as soon as it is available. At NoC-to-bus level, it is possible to enable the per-packet S & F. While different S & F options can be selected from bus-to-NoC level, storing a whole bus packet or storing an entire compounded transaction, it mean a collection of sequential packets tied together by setting the appropriate bus fields. The mechanism for the per-packet S & F implementation is too simple. After the completion of a packet, the FSM controlling the FIFOs which reads in a state where only the header FIFO is checked, to extract the beginning of a new packet.

B. *Error Management Unit*

The Error Management unit(EMU) is an optional stage that can be instantiated between the Kernel and the interface to the NoC. EMU can handle bad address errors .When the address of the Master IP transaction is not in the range of assigned memory map or when the request transaction is trying to access a protected memory zone without having the permission, the packet will indicate as an error packet and it will discard the packet.

C. *Order Handler*

Typically the bus protocol rules force the transactions generated by a single Master IP get their responses with the same order of that requests. In NoC platforms, it may happen that some responses are reordered by the interconnect due to the existence of alternative paths or the paths of different length between the Master and its reachable Slaves. Each transaction generated by a Master is characterized by a destination address and an identification number. The destination address identifies to access a specific Slave . The identification number characterizes the Master itself. This information is used by Target NIs to encode response packet to be routed back.

D. *Security*

The security service act as a hardware firewall mechanism. It introduces a set of rules that a request transactions coming from the Master IP must satisfy to get access to the network. Security rules are applied in the Security block of the network interface during the packet encoding. If a test fails the security check, the corresponding transaction is marked as an error and it is detected by the EMU, which must be activated get as well to properly manage security violations.

V. EXPERIMENTAL RESULT

A structural model of NI is developed and an overall flow is showed in the Fig.4 top module which consist IP core, NI module. The simulation of NI module is showed in the Fig.5 contain header encoder, header and payload FIFOs and the NoC interface with network functionalities. The modules are modelled using Verilog in Xilinx ISE Design Suite 13.2 and the simulation of the design is performed to verify the functionality.

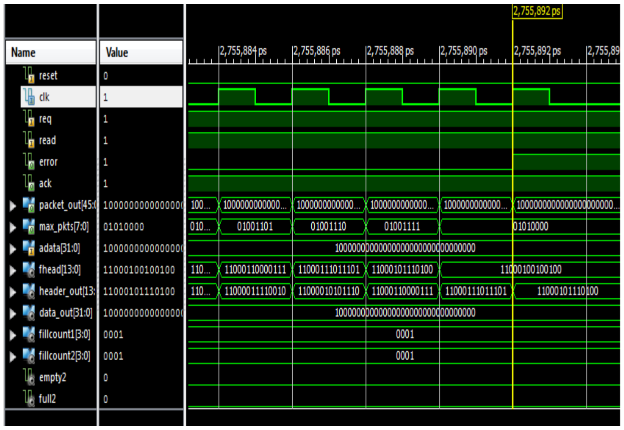

Fig. 4 Simulation results of top module

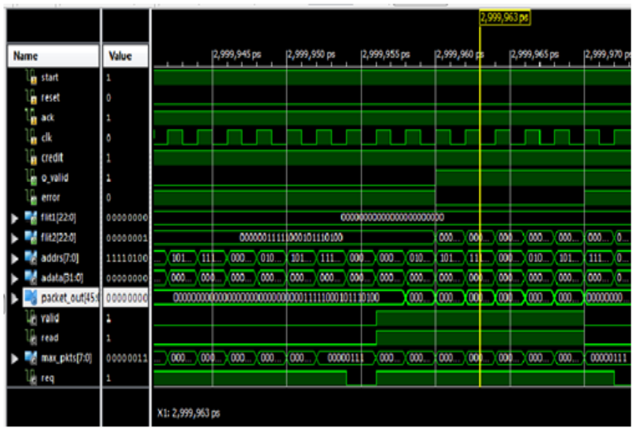

Fig. 5 Simulation results of NI module

VI. CONCLUSION

A novel Network Interface for on-chip communication infrastructure has been designed. NoC aims at providing scalable and flexible packet switched communication infrastructure. Network interface will guarantee a successful end to end data delivery between the IP core. NI is intended to provide communication between two separate interconnect namely NoC and on chip bus. Handshake rule is provided between the IP core and Network interface. In this paper, NI design includes the advanced networking functionalities like store & forward transmission, error management unit, ordler handler, security in a single NI. On the top of the base core architecture of NI all advance features are added which supports the hardware. The NI represents a complete solution that can be customized for different literature, from multimedia to real-time applications just analyzing the system requirements and parameters to obtain an optimized hardware description and tailoring features.

REFERENCES

[1] Sergio Saponara,Senior Member, IEEE,Esa Petri,Member, IEEE,Luca Fanucci,Senior Member, IEEE, and Marcello Coppola, Member, IEEE,"Design of an NoC Interface Macrocell With Hardware Support Of Advanced Networking Functionalities",IEEE Trans.on computers,vol.63, no.3,march 2014

[2] S. Saponara, L. Fanucci, and M. Coppola,"Design and coverage-driven verification of a novel network-interface IP macrocell for network-on chip interconnects", Microprocessors and Microsystems, vol. 35, no. 6,pp. 579 592, 2011.

[3] M. Ebrahimi, M. Daneshtalab, P. Liljeberg, J. Plosila, and H.Tenhunen, "A High-Performance Network Interface Architecture for NoCs Using Reorder Buffer Sharing",Proc. 18th Euromicro Intl Parallel, Distributed and Network-Based Processing (PDP) Conf.,pp. 546-550, 2010.

[4] B. Attia, W. Chouchene, A. Zitouni, A. Nourdin, and R. Tourki, "Design and implementation of low latency network interface for network on chip",in Proc. 5th Int. Design and Test Workshop (IDT), pp. 37 42,2010.

[5] H. Kariniemi and J. Nurmi, "NoC Interface for fault-tolerant Message- Passing communication on Multiprocessor SoC platform," in *Proc. NORCHIP*, 2009, pp. 1–6.

[6] X. Yang, Z. Qing-li, F. Fang fa and L. Cheng, "NISAR: An AXI Compliant On chip NI Architecture Offering Transaction Reordering Processing",Proc. Seventh Intl Conf. ASIC (ASICON 07), pp. 890-893, 2007.