View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by INRIA a CCSD electronic archive server

FERONOC: FLEXIBLE AND EXTENSIBLE ROUTER IMPLEMENTATION FOR DIAGONAL MESH TOPOLOGY

Elhajji Majdi, Brahim Attia, Abdelkrim Zitouni, Rached Tourki, Samy Meftali, Jean-Luc Dekeyser

To cite this version:

Elhajji Majdi, Brahim Attia, Abdelkrim Zitouni, Rached Tourki, Samy Meftali, et al.. FERONOC: FLEXIBLE AND EXTENSIBLE ROUTER IMPLEMENTATION FOR DIAGONAL MESH TOPOLOGY. Conference on Design and Architectures for Signal and Image Processing, Nov 2011, Tampere, Finland. inria-00609117

HAL Id: inria-00609117

https://hal.inria.fr/inria-00609117

Submitted on 18 Jul 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

FERONOC:FLEXIBLE AND EXTENSIBLE ROUTER IMPLEMENTATION FOR DIAGONAL MESH TOPOLOGY

Majdi Elhajji,Brahim Attia,Abdelkrim Zitouni,Rached Tourki

UNIV. Monastir
Laboratory of Electronics and Micro-Electronics
Monastir 5019, Tunisia

Samy Meftali, Jean-luc Dekeyser

Univ. Lille 1
LIFL,CNRS, UMR 8022, INRIA
Villeneuve d'Ascq 59650, France

ABSTRACT

Networks on Chip (NoCs) can improve a set of performances criteria, in complex SoCs, such as scalability, flexibility and adaptability. However, performances of a NoC are closely related to its topology. The diameter and average distance represent an important factor in term of performances and implementation. The proposed diagonal mesh topology is designed to offer a good tradeoff between hardware cost and theoretical quality of service (QoS). It can contain a large number of nodes without changing the maximum diameter which is equal to 2. In this paper, we present a new router architecture called FeRoNoC (Flexible, extensible Router NoC) and its Register Transfer Level (RTL) hardware implementation for the diagonal mesh topology. The architecture of our NoC is based on a flexible and extensible router which consists of a packet switching technique and deterministic routing algorithm. Effectiveness and performances of the proposed topology have been shown using a virtex5 FPGA implementation. A comparative performances study of the proposed NoC architecture with others topology is performed.

Index Terms— SoC, NoC, RTL , FeRoNoC.

1. INTRODUCTION

Modern applications of specific SoCs in signal audio and video processing require the increase of computation capabilities. Thus, Intellectual properties (IPs) have been more and more integrated which makes communication very complex in these systems. Consequently, according to a set of works in state of the art [1, 2, 3, 4, 5] interconnection architecture based on shared busses shows its limits for the communication requirements of future SoC. In this context NoCs appear as a best solution to provide communication in the Chip. Due to the following characteristics; reliability, scalability of bandwidth and energy efficiency NoCs are emerging to replace busses.

However many applications, especially video encoder like H.264, have some performances requirements. Thus, a major goal in the SoC design is therefore,to ensure performances and QoS required by the application using the minimum available resource.

NoCs seems to be today the most appropriate communication solution for integrating many cores in a system and guaranteed QoS for applications. Indeed, the implementation of high NoC performances has become one of the most important challenges of designers. NoC is generally composed of three basic components: network interfaces (NIs), routers and links. Router which is an element of the NoC topology, implements routing function, switching technique and the flow control algorithms.Topology of NoC defines the connectivity or the routing possibilities between nodes, thus having a fundamental impact on the network performance as well as the switch structure (number of ports and port width). The tradeoff between generality and customization becomes then an important issue when choosing a network topology.

Another major factor that has an important impact on the NoC is the performance of router. The router is characterized by its degree, frequency, power consumption and latency. The degree of a router determines the number of its neighbors. Obviously, designing a router with a higher degree leads to the difficulties of VLSI (Very Large Scale Integration) implementation that can affects the performance of the SoC in term of used resources.

The motivation of this work has been addressing the demand for optimized infrastructure communication by proposing an optimal topology that provides a good performances. In the proposed diagonal mesh topology an even number of routers are connected by a links to the neighboring routers in clock wise and counter clock wise direction plus a central connection. The key characteristics of this topology include good network diameter that equal to 2, vertex symmetry, deterministic routing, generic number of routers and low degree for peripheral routers that equal to 3. High router degree reduces the critical path length but increases complexity.

This paper is organized as follows: Next section presents some related works. Section three presents the architectures and the routing algorithm of FeRoNoC. In section four, simulation and Implementation results for router on FPGA are presented and then some comparison and discussions takes

place in section five. Finally we conclude the work.

2. RELATED WORKS

There are many works which offer new architectures of NoC like STNoC [6] and GeNoC[7]. These NoCs are based on flexible and evolutionary packets. Moreover, they are based on the Spidergon and the Octagon topologies respectively and they have low costs of silicon implementation for the router and network interface. On the other side, most of the NoC use 2DMesh topologies [8, 9, 10, 11]. In [8] SoCIN is presented, which is extensible based on XY routing. The basic router called RASoC(Router architecture for SoC) have a configurable FIFO. A SoC(adaptive system on chip)[9] is a scalable architecture, flexible and modular communication between routers. The AEtheral NoC[10] is based on the ATM network and adopts a fixed size packet technique which is oriented to a real time application. The disadvantage of this NoC is the fact that reception of the packets is not guaranteed when flits take different paths. HERMES [12] is a 2D-Mesh NoC topology satisfies the requirement of implementing a low area and low latency communication for system on chip modules. Deterministic routing and Wormhole switching [13] are the dominant approach for NoCs researches.Deterministic routing algorithms are usually used because they require a low cost in term of logic compared to the adaptive algorithms. In Wormhole technique a packet is divided into flits, it implements the functionality with lower buffer requirement[14].Thus, it is an interesting solution compared to packet-based circuit switching and virtual cut-through.Others switch technique like packet switching and virtual cut-through switching require enough buffers for saving a whole packet at each intermediate router. In [15] authors have compared different NoC architectures. Based on different point of views including performances such as latency, area and power consumption. In another work, Bononi and Concer[16] compared ring, mesh and Spidergon topologies. Their paper showed that the spidergon topology outperforms the mesh and the ring topologies.

3. ROUTER ARCHITECTURE

The proposed Mesh diagonal topology generalizes and improves performance of the well known STOctagon network processor topology by a simple bidirectional ring with a central router. This network is constituted by (N+1) routers including a central element which is connected with all peripheral routers via links. Each peripheral router provides four input/output ports enable to connect left/right neighbors (routers), central router and the local IP. The central router contains N ports for the connection of each peripheral router and additional one permitting the connection with its local IP as is shown in figure 1:

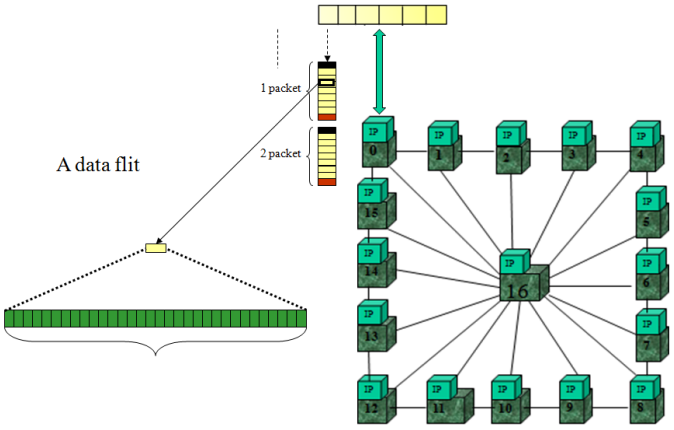

Fig. 1. The NoC topology.

The communication packet used in our NoC is composed by three basic messages called flits which consist of:

- Header: composed by one bit (BOP)indicating the beginning of a packet,
- Body: contain the data to be transmitted on 32 bits,
- Tail: one bit indicating the end of a packet (EOP).

Both of the two kinds of routers composing the diagonal mesh topology use packet switching technique and Wormhole flow control mechanism.In packet switching, packets are transmitted without any need for connection establishment procedure.It requires the use of a switching mode, which defines how packets move through the switches.The described router has a controlled logic and bidirectional ports. Each port has an input buffer for temporary storage of information and a local port enabling communication between router and its local IP core. This module contains a set of components that are described below. Figue 2 shows a general block diagram for the router.

In the diagonal mesh, topology is presented in this work, each switch has a different number of ports, depending on its peripheral position or central router.

3.1. Peripheral Routers

In our NoC, we used a synchronous router with four input/output ports (local, clockwise, counter clockwise and across). Each port is connected to a bi-directional exchange bus. Each switch has a unique address and the switching technique used is packet switching. The data flowing through the network is a Wormhole routing. We made this choice to reduced number of buffers required per node and the simplicity of the communication mechanism. The diagonal mesh topology uses credit based flow control strategies. This later presents interesting advantages over handshake. In credit-based protocol, when the receiver is free, the transmitter

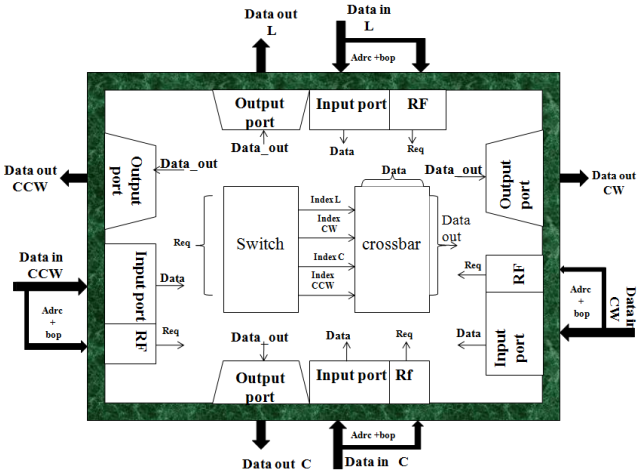

Fig. 2. The router architecture.

sends a new data at each cycle of the clock and the receiver indicates its availability by a signal named credit. The peripheral router is composed by a bidirectional ports numbered starting from zero. They connect the router with its local IP and these neighboring bidirectional ports. The connection is defined as follows:

1. The first port is connected to neighbor in the clockwise direction.

2. the second port connected to the central router.

3. the third port connected to neighbor in counter clockwise.

The internal architecture of the peripheral router is composed by several components, such as input/output controller), routing function and switch allocator. We use in each input port of our router an input component. This latter is divided into three components named FIFO, input controller and output controller. The main function of the input controller (IC) is to create an interface between output controller (OC) of the source router and IC of the current router. The IC is activated when the current router receives the beginning of packets (BOP). Thus, it allows the reception of packet flits from the output ports of the adjacent routers, storage in the FIFO buffer and the management of the flow control (credit based) between adjacent routers. The IC can accept a new packet when the previous one is not entirely switched. This part of the router allows the reception of packets sent by the neighboring OC and writes them in the FIFO. The block diagram of this component is shown in figure 3. This figure describes signals and recommended hardware implementation for the router. The physical data bus width is 32 bits. The IC in the figure is composed by the following signals: (1) BOP control signal indicating the beginning of received packet, (2) EOP: indicating the end of packet;(3) REQ: control signal indicating data availability, (4) $credit_{out}$: control signal indicating that FIFO is not full; (5) data: indicating data to be received; (6)$data_{in}$: data to be written in the FIFO, (7) write: control signal indicating the writing in the FIFO. Thus as explained above, the IC aims to establish a connection between entities of initiator and destination routers.

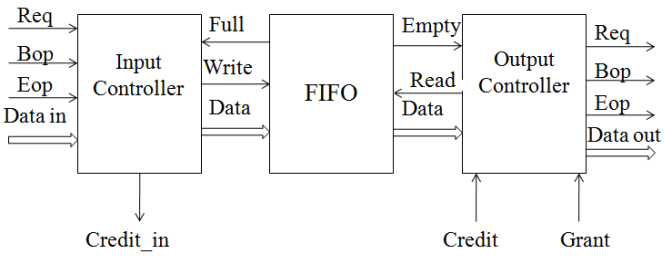

Fig. 3. The Input module.

To complete the description of input block, an explanation of the functionality of the output controller (OC) is mandatory in order to finish the construction of the input block of peripheral router.

The output controller (OC) is the last block of the peripheral router. Their main roles consist in communicating with the IC, reading data, storing data in the FIFO and sending them after getting the control signal credit. When the output port, indicated by the routing function, is allocated in the output controller, the OC starts sending packet. This component is composed by control and data signals. The functional description of these signals is similar to that of the IC. Signal Grant indicates that the OC must send the packet to the corresponding output port. The association of an input and output controller constitutes the input module of the peripheral router. The FeRoNoC input module is described in VHDL and validated by a functional simulation. Its behavior can be summarized by the following steps:

1. When BOP=1, sending flits begins.

2. EOP=1, when the last flit is received.

3. Req=1 Sending.

4. Credit=1, the OC reads the data.

5. Full=1 the IC saves data in the FIFO.

3.2. Routing function and arbitration

A routing function defines the path followed by each message or packet through the NoC. It describes also how data are forwarded from sender to receiver by interpreting the destination address field of the header flit. The choice of a routing algorithm depends on several metrics like power,logic and routing

table,increasing performance and maximizing traffic utilization of the network. In the FeRoNoC router there are two modules implementing the control logic which are routing and arbitration. The routing module is presented in figure 4.

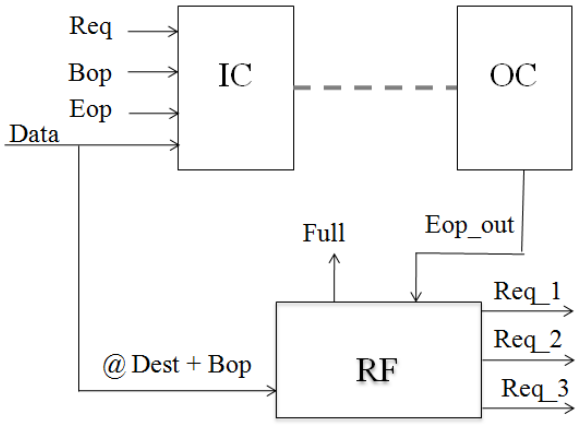

Fig. 4. Routing module.

In our case, routing is deterministic and the communication between ports is not usually established. Indeed the local port can communicate with all ports. The central router has a direct connection with all the routers of the NoC. The routing function extracts the destination address of the packet, calculates the path to follow and generates a req_p signal, where p indicates the number of destination of ports as represented in figure 4.

In our design, each input port of peripheral routers has its specific routing function. The routing algorithm is distributed inside the router as described in the following steps: The routing function of the local port can request three other output ports clock wise, Counter clock wise and across. To provide a correct message transfer, the following algorithm describes how to calculate the direction of packet.

1: $numberofjump = (@dest - @curr)mod(N+1)$
2: **if** number of jump=0 **then**
3: local port direction
4: **else if** number of jump= 1 v 2 **then**
5: clockwise direction
6: **else if** number of jump =14 v 15 **then**
7: counter clockwise direction
8: **else**
9: central router
10: **end if**

In the other side the clockwise port can request only two output ports Counter clockwise and local, thus the algorithm can be described as follow:

1: $numberofjump = (@dest - @curr)mod(N+1)$
2: **if** number of jump=0 **then**
3: local port direction
4: **else if** number of jump =14 v 15 **then**
5: counter clockwise direction
6: **end if**

The routing function of the counter clockwise port can request only two ports (clock wise and local).

1: $numberofjump = (@dest - @curr)mod(N+1)$
2: **if** number of jump=0 **then**
3: Local port direction
4: **else if** number of jump= 1 v 2 **then**
5: Clockwise direction
6: **end if**

Finally, routing function of across port can request only the local port.

1: $numberofjump = (@dest - @curr)mod(N+1)$
2: **if** number of jump=0 **then**
3: Local port direction
4: **end if**

The EOP control signal allows the activation of the arbitration. When it is equal to 1 the arbitration is executed. Indeed the output port is allocated to an input port, thus the correct transfer is provided. The routing function sends control signal to the Switch Allocator of the suitable router. The hardware description of the Switch Allocator is presented in Figure 5.

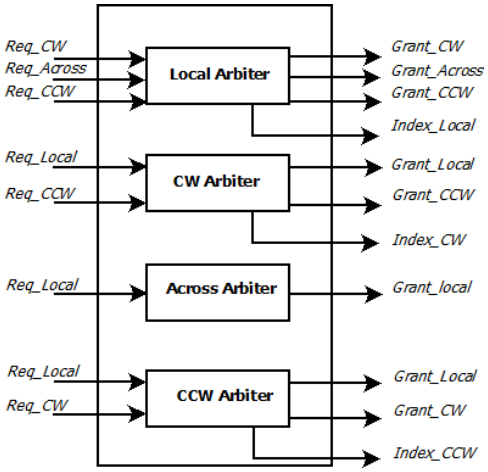

Fig. 5. Switch Allocator.

This module contains four arbitration components that decide the connection between input port and the output port. Its behavior is described by the following algorithm:

1: **if** R_0 allocate the output port **then**
2: priority will be $R_1 > R_2 > R_0$
3: **else if** R_1 allocate the output port **then**
4: priority will be $R_2 > R_0 > R_1$
5: **else if** R_2 allocate the output port **then**
6: priority will be $R_0 > R_1 > R_2$
7: **end if**

As is shown in figure 5 the switch allocator has many control signals such as Req, grant and index. The index signal in-

dicates the number of local input port, grant signal: indicates that the request is achieved. Indeed this information can activate the Crossbar element. The crossbar is a physical switch connecting the inputs to the outputs. It can be represented as a module with N inputs and N outputs. The switching in the crossbar can be achieved via multiplexers. The selection bit of each MUX could be generated by the index signal. Generally the transfer of the packet is performed in many steps. Firstly the state of the FIFO must be sent, secondly, the input controller starts data storage in the FIFO. Then, the routing function sends request signal to allocate the output port and finally follows the arbitration technique. Figure 6 presents a mapped architecture of the routing function and the switch allocator.

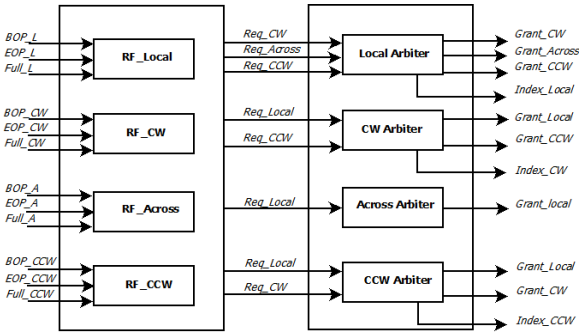

Fig. 6. Interaction between routing function and Switch allocator.

The interaction between modules was described in VHDL and validated by functional simulation as is presented in Figure 7:

The simulation steps are described as:

1. The second port will send to local port.
2. the BOP=1, REQ=1 the IC module starts storage of data.
3. At the same clock the routing function receives the BOP signal and the destination address then specify the suitable output port.

3.3. Central router

The main objective of this router is to minimize the traffic and the load imposed on the peripheral router. Moreover, it allows the routing of data in the network to their destination with a number of required jumps equal to two. The importance of this router appears when the SoC contains a set of IP core. Hardware architecture of this router is the same as the peripheral routers. The only difference appears in the number of input/output port. Figure 8 shows the block diagram of this router.

Each input port of the central router has its specific routing function. The routing algorithm is distributed inside the

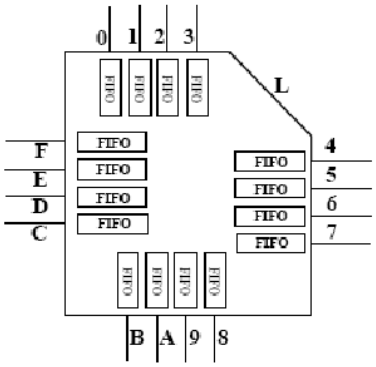

Fig. 8. Block diagram of the central router.

router. The routing function (RF)of the local port can request all other output ports of the central router. RF of the port numbered i can request only output ports from 0 to i - 3 or output ports from i + 3 to N and also the local port of central router.

4. SYNTHESIS AND SIMULATION RESULTS

This section presents some simulation and synthesis results. It also explains how to find a compromise between latency and diameter. The router in the FeRoNoC was validated by functional simulation and by the synthesis results. We started by a VHDL simulation, in figure 9 a packet transmission in router is illustrated. The local port sends data to the second port after having sent the flits and the request. The simulation shows some of internal signals defining interconnection.

The target address is the peripheral router number 8. This later is located far from the source, thus the packet crosses the central router. The simulation scenario can be explained as follow:

1. Peripheral Router sends the first flit of the packet (address of the target switch) to the data out signal at it's across port and asserts the req and bop signals in this port.
2. Central Router detects the req signal asserted in its port number 0 and gets the flit in the data in signal. It takes 2 clock cycles to route this packet. Next flits are routed with 1-clock cycle latency.
3. Central Router puts the flit in data out signal and asserts the req and Bop signals of its output port number height. It takes 2 clock cycles to route this packet.
4. Router 8 detects asserted req signal of it's across port. The first flit of the packet is routed to the local port of the router 8 and the source to target connection is now established.
5. The remaining flits contain the payload of packet.

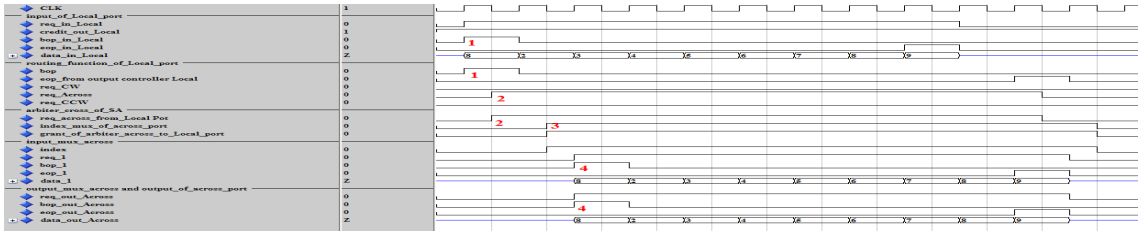

Fig. 7. Simulation of the router.

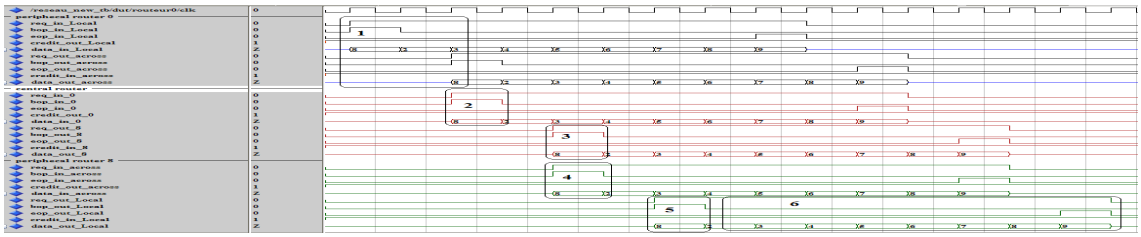

Fig. 9. Simulation of the proposed diagonal Mesh topology

6. After sending all flits, the connection is closed and the inputs and outputs reserved by this packet can be used by other packets.

However, the main objective of the central router is to improve latency in the NoC depending on the distance. In our case study the peripheral router 0 sends data to the peripheral router 8. The simulation shows that minimal latency to switch packet from source at target depends on the diameter of the network. Latency is given by:

$$Latency = (\sum_{i=1}^{n}(R_i)) + P \times clockcycles \qquad (1)$$

where n is the number of routers in the communication path, R_i is the required time of the routing algorithm at each switch, P is the reference packet size and clock cycle presents a required time to send flits. Based on this equation, for our diagonal Mesh topology, the n=2 target and source routers are involved. In this paragraph some synthesis results are presented and a cost analysis of area and power consumption is realized. The router performance has been evaluated in terms of speed, latency and estimated peak performance. The FeRoNoC router was synthesized on Xlinx virtex 2pr xc2vp device using Xilinx ISE 9.1. The simulation was performed using modelSim 6.5 SE tool. The proposed router has been prototyped on 2 different FPGA technologies: Xilinx Virtex5 xc5vlx50-3ff676 and Xilinx virtex 2 pro. Table I presents our synthesis router results with Xilinx virtex 5 in which: area, operating frequency and power consumption results are shown. Table II presents implementation works targeting the same FPGA. The maximum running frequency is about 264 Mhz and the power consumption is 33mw.

Our approach provides performances in terms of latency, speed and area. The router design is highly modular and adaptable without the use of explicit handshake signal for communication between sub modules of the router. Other metrics evaluating our design is the peak performance which depends on the maximal clock frequency F_{max}, the flit size ($flit_{size}$) and the time (T) in clock cycle for transmitting a flit.

$$PMperport = (F_{max}/T) \times flit_{size} \qquad (2)$$

The credit based control flow used by our router requires one clock cycle for transmitting one flit then $T = 1$ and $flit_{size} = 32bits$

Table 1. Synthesis results

FPGA/Perf.	Virtex2	Virtex5
Slice	5%	4%
Flip Flop	3%	2%
Lut	2%	2%
Frequency(Mhz)	218	264
Power(mw)	97(200Mhz)	33(200Mhz)
Peak.Perf.	6.9 Gbit/s per port	8.44 Gbits/s per port

5. COMPARAISON AND DISCUSSION

The work on flexible and extensible network on chip is an emerging topic. There are many proposed topologies in the literature, as cited in previous section each topology designs offers a different set of tradeoffs in terms of metrics, such

as network degree, network extendibility notice in which a 2D mesh[17] topology provides very good theoretical metrics. Nevertheless, due to the increasing complexity of application, this topology cannot provide a good performance. On the other hand simple topology like ring provides a low cost in term of area but poor performance where the number of cores increases. The diameter and average distance represents an important factor in terms of performance and implementation. The proposed NoC is designed to deliver a good tradeoff between hardware cost and theoretical performance. Due to its higher connectivity our topology out performs ring, mesh and STspidergon in terms of diameter and average distance. In addition, our diagonal Mesh topology can contain a large number of nodes without changing the diameter. Indeed it can deliver a good latency for multimedia application like video encoder (H.264). Thus, it represents a solution for on chip communication in next SoC. The major inconvenience of the proposal appears in the required link. However, based on the evolution of semi conductor technology this problem can be solved.

On the other side, many works are presented in the literature to implement NoC. Authors in [18] describe a 2D meshes Network on chip implementation based on virtex4 and virtex2 FPGA. They describe an open source FPGA based NoC architecture with a high throughput and low latency. In this work, a generic bridge based on packet switching with Wormhole routing, was proposed. As is shown in table II, the data width of this implementation is about 36 bits and the maximum frequency is less that our frequency. This work provides a low cost area. In [19] authors present a packet switched NoC running on a virtex2. Moreover, they show an implementation of packet switched and time multiplexed FPGA overlay networks running at 166 Mhz. The aim of this work was to support designers in choosing between time multiplexing and packet switching. They use a 32 bits data width, as is shown in table II. Our implementation outperforms this work in term of area, latency and maximum frequency.

Table 2. Comparison with other works

Perf./Works	[18]	[19]	Our
Data width	36bits	32bits	32bits
Latency	3	6	2
Slice	431	1464	989
Frequency(Mhz)	166	166	218
Topology	Mesh	Mesh	Diagonal Mesh

6. CONCLUSION

Network on chip presents a most adapted technology to perform communication in complex SoCs. In this work, we present a novel topology named diagonal mesh and related router called FeroNoC.It offers a low latency (2 clock cycles) and high speed (264 Mhz) communication for on chip modules. This paper presents all the details of our NoC such as topology, routing algorithm, dynamic arbiter and router module.This architecture offers a variety of SoC communication services owing to its flexibility and adaptability.The physical parameters of the designed router consist on the width and the depth of the FIFO, the number of the input/output ports, the valence and the maximal diameter of the NoC. Compared with other NoC, the advantage of this architecture, resides in its capacity to handle a suitable cost/performance compromise in the field of NoC. This is due to its wide constant and low diameter, latency of the router, frequency and power consumption. The simulation and implementation of this architecture show its performances and effectiveness.

Our next objective is to prove the use of the diagonal mesh topology, mapped H.264 encoder will be investigated in a future work. Also the dynamic reconfiguration (in terms of number of nodes which depends on the number of IP of the application) will be studied. Adopting this technology, it is possible to obtain application specific NoCs. Moreover, we are going to design a low power SoC, based on the diagonal mesh topology and H.264 encoder, which includes dynamic reconfiguration.

7. REFERENCES

[1] K Shashi, J Axel, P.S. Juha, F Marteli, M Mikael, Johny, T Kari, and H Ahmed, "A network on chip architecture and design methodology," in *IEEE computer Society Annual Symposium*. IEEE, 2002, pp. 105–112.

[2] M Mikael, N Erland, T Rikard, K Shashi, and J Axel, "The nostrum backbone communication protocol stack for networks on chip," in *VLSI Design*. IEEE, 2004, pp. 105–112.

[3] B Luca and D.M Giovanni, "Powering networks on chips energy-efficient and reliable interconnect design for socs," in *ISSS'01*. ACM, 2002, pp. 105–112.

[4] B Luca and D.M Giovanni, "Networks on chips: A new soc paradigm," *IEEE computer*, vol. 35, pp. 70–78, January 2002.

[5] G Pierre and G Alain, "Ageneric architecture for on-chip packet-switched interconections," in *DEsign, Automation and Test in Europe*. IEEE, 2000, pp. 250–256.

[6] C Marcello, D.G Miltos, L Riccardo, M Giuseppe, and Pieramlisi Lorenzo, *Design of Cost-Efficient Interconnect Processing Units: Spidergon STNoC*, 2008.

[7] J Schmaltz and D Borrione, "A generic network on chip model," Tech. Rep., TIMA Laboratory, Grenoble,France, 2009.

[8] J Liang, A Laffely, S Srinivasan, and R Tessier, "An architecture and compiler for scalable on-chip communication," *IEEE transaction on very large scale integration systems*, vol. 12, pp. 711–726, July 2004.

[9] C.A Zeferino and A Susin, "A parametric and scalable network-on-chip," in *16th Symposium on integrated circuits and system design*. IEEE, 2003.

[10] K Goossens, J Dielissen, and A Radulescu, "Aethereal network on chip: Concepts, architectures, and implementations," *IEEE Design and Test of Computer*, vol. 22, pp. 414–421, September 2005.

[11] F Uriel and R Prabhakar, "Exact analysis of hot-potato routing," in *33rd Annual Symposium on Foundations Of Computer Science*. IEEE, 192, pp. 553–562.

[12] K Goossens, J Dielissen, and A Radulescu, "Hermes: an infrastructure for low area overhead packet-switching networks on chip," *VLSI journal, Elsevier*, vol. 38, pp. 69–93, March 2004.

[13] J Dally and C Seitz, "Deadlock-free message routing in multiprocesor interconnection networks," *IEEE transaction*.

[14] J Dally and B Towles, "Route packets, not wires: On-chip interconnection networks," in *Design Automation Conference (DAC)*. ACM, 2001, pp. 683–689.

[15] P.P Partha, G Cristian, J Michael, I Andre, and S Resve, "Performance evaluation and design trade-offs for network-on-chip interconnect architectures," *IEEE transaction on Computers*, vol. 54, pp. 1025–1040, August 2005.

[16] L Bononi and N Concer, "Simulation and analysis of network on chip architectures: Ring, spidergon and 2d mesh," in *DATE*. IEEE, 2006, pp. 154–159.

[17] T.A Bartic, J.-Y Mignolet, V Nollet, T Marescaux, D Verkest, S Vernalde, and R Lauwereins, "Topology adaptive network-on-chip design and implementation," *IEE Comput. Digit*, vol. 152, pp. 467–452, July 2005.

[18] A Ehliar and D Liu, "An fpga based open source network-on-chip architecture," in *Field Programmable Logic and Applications*. IEEE, 2007, pp. 800–803.

[19] N Kapre, N Mehta, M deLorimier, R Rubin, H Barnor, M.J Wilson, M Wrighton, and A DeHon, "Packet switched vs. time multiplexed fpga overlay networks," in *IEEE Symposium on Field-programmable Custom Computing Machines*. IEEE, 2006, pp. 800–803.