The design methodology and the implementation of MPSOC based on Delta MINs on FPGA

Ramzi Tligue
CES Laboratory
National Engineering School of Sfax
Email:
ramzitligue@ieee.org

Yassine Aydi
CES Laboratory
National Engineering School of Sfax
Email:
yassine.aydi@oous.rnu.tn

Mouna Baklouti* and Mohamed Abid
INRIA* & CES Laboratory
Email:
baklouti_mouna@yahoo.fr
mohamed.abid@enis.rnu.tn

Jean-Luc DEKEYSER
LIFL and INRIA-Futurs
University of Lille,France
Email:
dekeyser@lifl.fr

Abstract— **MPSOC integrated a variety of heterogeneous components which require a communication between them. A solution to flexibility and reconfigurability of interconnects is the use of Network on Chip (NoC). These latter are likely proposing efficient solutions with the complex problems of the embedded system integrations.**

Multistage interconnection networks have been frequently proposed as connection means in classical multiprocessor systems. They are generally accepted concepts as on-chip communication platform. We describe in this paper the design methodology and the implementation of a Delta multistage interconnection network on chip. Also, we propose a flexible and an efficient model of MPSOC architecture based on Delta MIN. Finally, the effectiveness of the proposed design methodology is shown through parallelized applications on MPSoC architecture.

Index Terms—**MIN, MPSOC, Performance of Delta MIN**

I. Introduction

Applications require more and more intensive computations, especially multimedia applications. In order to improve the performance of current embedded systems, Multiprocessor System-on-Chip (MPSoC) are a recent solution for increasing computational system performance. Current research in this area points to a large number of processors in a single chip. MPSOC have to be energy efficient, cheap, reliable, and must offer sufficient computing power for advanced and complex applications. So an efficient interconnection network intra-chip [3][4], with low latency and high bandwidth, is fundamental for high communication throughput among the IPs. Network-on-Chip (NoC) is an emerging paradigm for communications within MPSoC systems implemented on a single silicon chip.

Multistage interconnection networks (MINs) with the banyan property are proposed to connect a large number of processors to establish a multiprocessor system [1]. They are used to connect processor to processor and/or memory modules [5]. They are also used as interconnection networks in ATM switch [2]. Such systems require high network performance.

The design methodology and the implementation of MPSoC architecture based on Delta MINs are investigated in this paper. Section 2 outlines some related work in this area. Next, MIN architecture is introduced. Section 4 details the modeling and implementation of Delta MIN's architecture on FPGA. Then an integration of our NoC in MPSoC architecture is described in section5. Hardware synthesis and Simulation results are also detailed in section 4 and section 5. Finally, we conclude the paper and we give directions for future work.

II. RELATED WORK

The design of future homogenous and heterogeneous Multi-Processor System-on-Chip [6] introduces many difficult problems, being communication, one of the most challenging ones. Communication architectures have been the focus of much research over the past several years because of their significant impact on system performance [7]. We briefly describe some communication architecture designs proposed recently. Kumar [8] has proposed a reconfigurable Multi Processor Network on Chip architecture in which both the network and the processing nodes are fully configurable, and the network is designed to match the application requirements. A design of an adaptable FPGA-based Network on Chip dynamically reconfigurable has been proposed by pionteck et al[9]. Reconfiguration can be done without stopping or stalling the NoC by using dynamic routing tables. BASIC-NoC [10], a NoC-based multiprocessor SoC was designed and prototyped. The MINs are often used to connect a large number of modules. In [11], the authors present a model of reconfigurable MIN described in SystemC and an estimation of its performances.

We proposed an efficient model of MPSoC architecture based on Delta MINs that will be designed and implemented on a Xilinx virtex4.

III. Basic MIN

In this section, we present an overview of the networks used for the design of the interconnection platform for MPSoC on programmable circuits.

A. Banyan MIN

The common multistage interconnection networks (MINs) used, have N inputs and N outputs nodes and are built using $r \times r$ switches. Such MINs have N/r switches at each stage, and $log_r N$ stages of switches denoted d. The different stages are connected by links generated by applying permutation functions. Banyan MIN is a multistage interconnection network characterized by one and only one path between each source and destination. A banyan MIN of size N×N consists of r×r crossbars. An interesting subclass of Banyan MINs is composed of Delta networks. Let denote by: o_i the ith output of a crossbar in a MIN, and by C_j, a crossbar belonging to the stage j. So, the Delta property can be defined as follows: if an input of C_j is connected to the output o_i of C_{j-1}, then all other inputs of C_j must be connected to the stage (j-1) on outputs with the same index i.

B. Delta networks

The difference between each of the existing MINs is the topology of interconnection links between the crossbar stages. A study of equivalence of a variety of Delta MINs (Fig.1) has been detailed in [13].

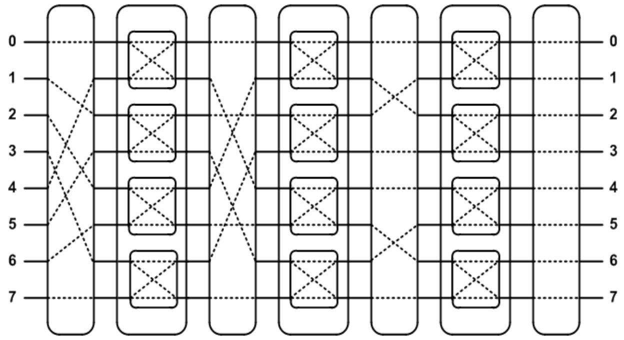

Fig.1. Delta network (8, 2)

IV. MIN's architecture

A. Modeling of MIN architecture

In this section we will deal with the part which relates to the design of the Delta multistage networks on chip dedicated to multiprocessor architectures on reconfigurable platforms FPGA and their methodology.

Our architecture is primarily composed of the following modules: Routing, Arbitration, Memorizing and Connection Module. Then, we detail the internal architecture of each component by explaining its operating process.

1) Data packaging **:**

The data exchanged between MIN nodes are fragmented into packets; the latter depends on the components integrated in the NoC design. The package is composed of three parts: (Fig.2)

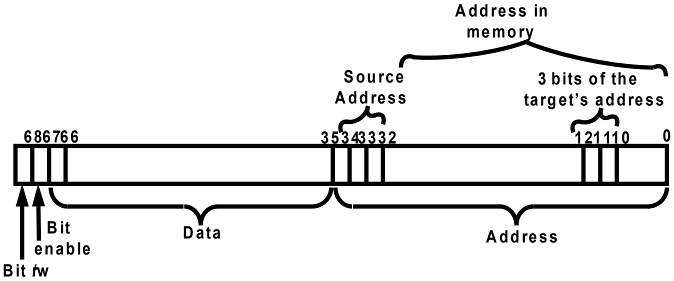

Fig.2.packet sent

- Header: Depending on the information in the packet header, the switch decides if it's an enabled read or write operation.
- message: the data to be exchanged.
- queue: contains the address of the source (processor), the target's address (memory) and the address in the memory.

The packet format depends primarily on the processor type used in the architecture (16-bit, 32-bit…).

2) Switch

It is the most important component of our MIN.

Fig.3 represents a crossbar with r equals 2. As shown in that Fig, the switch module is composed of couple of FIFO, a Scheduler which decides when data is sent from particular inputs to their desired outputs. The incoming packets are buffered in FIFO and held until they will be sent to their destinations.

The scheduler's decision is determined by a scheduling algorithm which has to be fast, efficient, easy to implement in hardware, and fair in serving all the inputs.

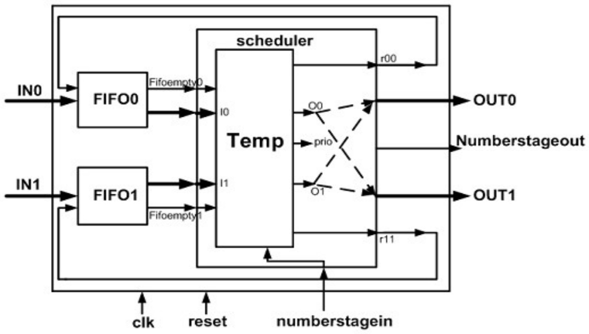

Fig.3. Switch architecture

There are various scheduling algorithms [12]. We choose to use round robin algorithm. The scheduler plays the role of a router, arbiter and a temporary memorizing. He has 2 Inputs which receive the packets from FIFOs. These latter will be held and stored until being forwarded according to their destination within a minimum time. The routing module in the Scheduler determines the output port to which the packet will be sent using his self routing algorithm. If two packets destined for the same output port arrive simultaneously at different input ports, the arbiter allows only one packet at a time to be transferred to output port (the other packet will be held in the temporary memory).

B. MIN implementation

A multistage network is mainly composed of stages of switches connected by blocks of links defining topology. These blocks describe the link permutation between stages of switches. The multistage network allows connecting N processors to N modules.

C. Performances Estimation

After having implemented each component, we released the performances. These performances are focused primarily on FPGA area and latency.

We have used VHDL and Xilinx ISE 9.1i to synthesize and target our components for Xilinx virtex4 XC4VLX200-FF1513 device.

1) Switch performances

Two types of switch were conceived. The first operates in asynchronous mode and the second in synchronous mode. Moreover we varied the depth of the FIFO used for the two operating processes (2, 4 and 8 by FIFO) to evaluate the switch area consumption.

It was noted that the most adequate solution for our switch is to use a FIFO of depth 8 because it has less FPGA area. This is explained by the use of FIFO blocks prefabricated by XILINX.

TABLE I
Synthesis and latency of Switch in asynchronous mode

Logic Utilization					Latency
Depth of FIFO	Number of Slices	Number of Slice Flip Flops	Number of 4 input LUTs	Number of FIFO16/ RAMB 16s	
2	1073	719	1590	0	2 cycles
4	1081	731	1606	0	
8	756	587	1349	4	
Target : xc4vlx200-11ff1513					

TABLE II
Synthesis and latency of Switch in Synchronous mode

Logic Utilization					Latency
Depth of FIFO	Number of Slices	Number of Slice Flip Flops	Number of 4 input LUTs	Number of FIFO 16/ RAMB 16s	
2	812	713	1182	0	3 cycles
4	826	719	1208	0	
8	568	587	929	4	
Target : xc4vlx 200-11ff1513					

The first switch which operates in asynchronous mode is characterized by minimal latency and important area consumption on FPGA (TABLEI) unlike the second switch which operates in synchronous mode is characterized compared to the first one by less FPGA area and more latency.

2) MIN performances

The performances estimation of the MIN is based on the two types of switch which we already mentioned. There are two types of MIN then: the first operates in asynchronous mode and the second in synchronous mode. TABLE III, TABLE IV

The results were obtained by varying the number of the processors (4, 8.16 and 32)

TABLE III
Synthesis of MIN in Asynchronous mode

Target : xc4vlx200-11ff1513		Logic Utilization						
Number of processor	Number of Slices	%	Number of Slice Flip Flops	%	Number of 4 input LUTs	%	Number of FIFO16/ RAMB 16s	%
4	3538	3	2344	1	6521	3	16	4
8	10718	12	7036	3	19673	11	48	14
16	33199	37	19024	10	61337	34	128	38

TABLE IV
Synthesis of MIN in Synchronous mode

Target: xc4vlx200-11ff1513		Logic Utilization						
Number of processor	Number of Slices	%	Number of Slice Flip Flops	%	Number of 4 input LUTs	%	Number of FIFO16/ RAMB16s	%
4	2281	2	2364	1	3993	2	16	4
8	6839	7	7092	3	12257	6	48	14
16	18220	20	18912	10	33057	18	128	38
32	37566	42	48560	27	66657	37	320	95

V. Integration of MIN in MPSoC architecture

A. MPSoC architecture

We saw in last section that DELTA MIN presents a good solution to connect N processors to N memories. The objective of this section is to implement our MIN in a multiprocessor architecture, and then we vary topologies of Delta MIN in order to test the reliability and the effectiveness of the NOC and to compare the performances of the various types of the networks.

First of all, we will be interested to specify the MPSoC architecture. Then an example of applications will be described to test its functionality.

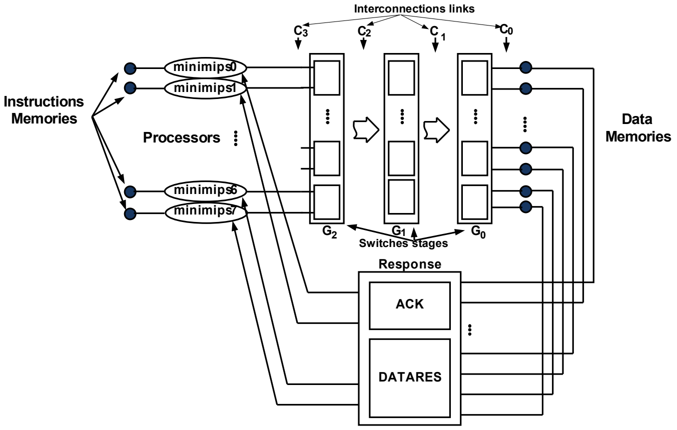

Fig.4.The proposed architecture

Our architecture is composed of N processors miniMIPS, N data memories, Delta MIN, response network and N instructions memories (Fig.4). The clock input is global to all components.

Each memory must send an acknowledgement signal after a processor access; as a consequence we haven't the problem of conflict in response path as in the request path (MIN). The response module has to convey packets from memories in order to join the processors within a minimum time. The response module is composed from two components: ACK and DATARES (transfer data from memories to processors).

B. Application

To test ease of use and performance of the system a FIR-Filter and IIR-Filter applications were implemented.

The difference equation of an IIR Filter is:

$$y[n] = \sum_{k=0}^{n} b_k * C[n-k] - \sum_{k=1}^{m} a_k * y[n-k]$$

If we put a_i=0 we get difference equation of the FIR.

Eight processors are used to illustrate functionality. The impulse response used in this example is: H(n) = { h(0),, h(7) }.

The input vector is: C(n) = {C0,…, C16 } and the generated output is: Y(n) = {y(0),…,y(23)}

H(n) is stored in registers and used by the processors when it is needed in the algorithm.

Each processor sends requests requiring the access to the data memories to bring inputs Ci, in order to carry out arithmetic operations. Finally it returns to data memory to store the obtained results. The distribution of the data in memories is random; consequently it is possible to use different distributions and to compare the performances of the network each time.

C. Results

The size of the implemented system is measured and then compared (Asynchronous Mode, Synchronous Mode), followed by simulation time measurements of the implemented FIR-Filter and IIR-Filter.

TABLE V
Logic utilisation for 4x4 to 16x16 architecture

Target : xc 4vlx 200 -11 ff1513						
	Synchronous Mode			Asynchronous Mode		
Number of processor	4	8	16	4	8	16
Number of Slices	12697	30251	68432	14390	35321	77563
%	14	33	76	16	39	87
Number of Slice Flip Flops	10057	22909	49466	10048	22896	49459
%	5	12	27	5	12	27
Number of 4 input LUTs	23667	56534	127564	26551	65540	142761
%	13	31	71	14	36	80
Number of FIFO 16 / RAMB 16 s	20	56	144	20	56	144
%	5	16	42	5	16	42

Two implementations of Asynchronous and Synchronous architecture are shown in TABLE V

We found for both model 4x4 and 8x8 that Asynchronous mode consumes more FPGA area than Synchronous mode.

In order to test the functionality of our architecture, we used ModelSIM tool to simulate it. Simulations have been done in the two modes (Asynchronous and Synchronous) and for two kinds of application: FIR-Filter and IIR-Filter. Measurements were taken by varying the topologies of the MINs. We used three topologies: Omega, baseline and Butterfly. System clock rate is 50 MHz. Fig.5 shows that Asynchronous mode has shorter latency than Synchronous mode.

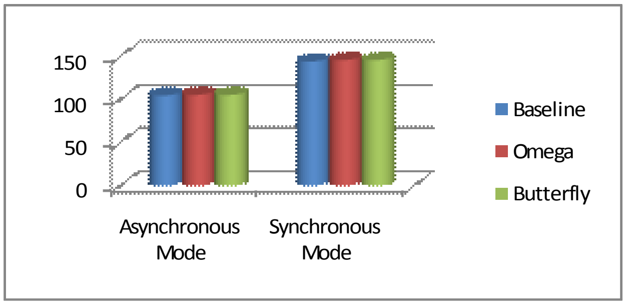

Fig.5.Latency of the application Filter FIR for 4 processors(ns)

In addition, we find that baseline topology is the best in term of simulation time for this application (Fig.6) for both operating modes.

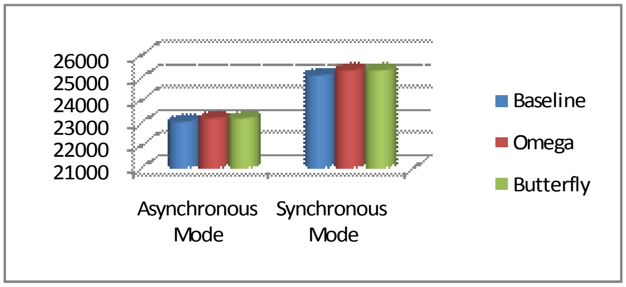

Fig.6.simulation Time of the application Filter IIR for 8 processors(ns)

VI. Conclusion

In this paper, we have proposed the design methodology and the implementation of MPSoC architecture based on Delta multistage interconnection networks on FPGA. Design and modeling of the Delta MIN were realized in three phases.

We proposed two models of Delta MIN networks, an Asynchronous model and another Synchronous and we detailed the characteristics and the advantages of each one.

In the second phase we implemented the Delta MIN in a multiprocessor architecture. The last phase deals with the application in order to evaluate and test the reliability of our interconnection platform and to estimate its performances. For that purpose, two types of Filters were implemented: a Filter FIR and Filter IIR. The performances estimation was focused on the FPGA area, the latency as well as the execution time.

One of future work is to prototype our architecture on a FPGA platform in order to estimate the energy consumption. Another possible prospect is to introduce the conceived chip into a whole industrial chain.

REFERENCES

[1] Gheith A. Abandah and Edward S. Davidson. Modeling the communication performance of the IBM SP2. In Proceedings of the 10th International Parallel Processing Symposium (IPPS96); Hawaii. IEEE Computer Society Press,1996.

[2] Raed Y.Awdeh and H. T. Mouftah. Survey of ATM switch architectures.Computer Networks and ISDN Systems,27:15671613, 1995.

[3] L. Benini and G. D. Micheli, "Network-on-Chip Architectures and Design Methods", IEE Computers & Digital Techniques, Vol. 152, Issue 2, pp.261-272, March 2005

[4] R. Kumar, V. Zyuban and D. M. Tullsen, "Interconnections in Multi-core Architectures: Understanding Mechanisms,Overheads and Scaling", ACM/IEEE International Symposium on Computer Architecture, pp.408-419, 2005

[5] S.Duquennoy, pNOC Design: modeling and IP based SOC Design Conference (IP--SoC 2006), 2006.

[6] Ahmed Amine Jerraya, Wayne Wolf. The Morgan Kaufmann Series in Systems on Silicon, "Multiprocessor Systems-on-Chips", ISBN: 0-12385-251-X, 2005

[7] M.Loghi, F.Angiolini, D.Bertozzi, L.Benini, and R.Zafalon, "Analyzing on-chip communication in a MPSoC environment ", in Proc.DATE,2004, pp.752-757.

[8] A. Kumar, An FPGA design flow for reconfigurable network-based multi-processor systems on chip, in the conference on Design, Automation, and Test in Europe, 2007, pp. 117122.

[9] T.Pionteck, "A Dynamically Reconfigurable Packet-Switched Network-on-Chip", in the conference on Design, Automation, and Test in Europe, 2006, pp. 136-137.

[10] Huynh Viet Thang, Pham Ngoc Nam Prototyping of a Network-on-Chip on Spartan 3E FPGA. Communications and Electronics, 2008. ICCE 2008.Second International Conference on In Communications and Electronics,2008. ICCE 2008. Second International Conference on (2008), pp. 24-28.

[11] Y. AYDI, MEFTALI S., M. ABID and J. DEKEYSER: Design and Performance Evaluation of a Reconfigurable Delta MIN for MPSOC. In 19th International Conference on Microelectronics (ICM'07), Cairo, Egypt, December 2007

[12] N. McKeown and T. E. Anderson, "A quantitative comparison of scheduling algorithms for input-queued switches," Computer Networks and ISDN Systems, vol. 30, no. 24, pp. 2309-2326, Dec. 1998.

[13] C. Kruskal, "A unified theory of interconnection network", Theoretical Computer Science, Vol.48, N°1 Elsevier Science Publishers Ltd, essex, 1986, pp. 75-94.