

FPGA Hardware Implementation and Evaluation of a Micro-Network Architecture for Multi-Core Systems

Yahia Salah, Med Lassaad Kaddachi, Rached Tourki

Abstract—This paper presents the design, implementation and evaluation of a micro-network, or Network-on-Chip (NoC), based on a generic pipeline router architecture. The router is designed to efficiently support traffic generated by multimedia applications on embedded multi-core systems. It employs a simplest routing mechanism and implements the round-robin scheduling strategy to resolve output port contentions and minimize latency. A virtual channel flow control is applied to avoid the head-of-line blocking problem and enhance performance in the NoC. The hardware design of the router architecture has been implemented at the register transfer level; its functionality is evaluated in the case of the two dimensional Mesh/Torus topology, and performance results are derived from ModelSim simulator and Xilinx ISE 9.2i synthesis tool. An example of a multi-core image processing system utilizing the NoC structure has been implemented and validated to demonstrate the capability of the proposed micro-network architecture. To reduce complexity of the image compression and decompression architecture, the system use image processing algorithm based on classical discrete cosine transform with an efficient zonal processing approach. The experimental results have confirmed that both the proposed image compression scheme and NoC architecture can achieve a reasonable image quality with lower processing time.

Keywords—Generic Pipeline Network-on-Chip Router Architecture, JPEG Image Compression, FPGA Hardware Implementation, Performance Evaluation.

I. INTRODUCTION

CONTINUING advances in semiconductor technologies allow the implementation of ever larger and more complex systems on a single chip. This concept is referred to as System-on-Chip (SoC). These systems usually contain several intellectual property (IP) cores reuse such as general-purpose processors, on-chip memories, dedicated hardware components, and also new technologies. Cores do not make up SoCs alone; they must include an interconnection architecture and interfaces to peripheral devices. Traditionally on-chip communication has been conducted via ad-hoc point-to-point interconnections or shared-bus structures. These interconnect and their derivatives are particularly problematic, because they are not scalable with respect to speed; also they quickly become the bottleneck of a multi-cores embedded system. A key challenge in the design of the future complex SoCs is the

choice of on-chip interconnection networks. An architecture that is able to accommodate a large number of cores, providing flexible and scalable design approach, and satisfying the need for communication and data transfers is the on-chip micro-network (NoC) architecture [1]-[5].

The wide majority of NoC researchers predict that packet-switched on-chip interconnection networks will be essential to address the complexity of future SoC designs and can meet various quality-of-service (QoS) requirements [2], [6]-[8]. Nevertheless, there are many design choices that must be made when designing an on-chip interconnect. We can consider explicitly the topology and the strategies used for buffering, switching and routing. While these mechanisms are important issues in the NoC design, contention resolution (at routers level) and flow control (between two neighbor routers) are considered to be critical issues, especially, for applications having different demands. Few NoCs have implemented both contention resolution and flow control to provide QoS guarantee. In particular, Mango [9] and Æthereal [10] offer complete solutions. Other NoCs have extended the control mechanisms with end-to-end or chained link-level flow control techniques, and they offer the same level QoS guarantees. For instance, the Nostrum [11] is a well-know example of this.

In our work, we apply the round-robin scheduling strategy to solve output port contentions in the router, and we employ a virtual channel flow control to avoid the head-of-line blocking problem. These techniques must be implemented and evaluated via real experiments in order to determine their impact on the performance metrics of the NoC. In this context, the implementation of techniques and network components is performed in register transfer level (RTL) hardware description using the Virtex FPGA technology, NoC design verification is done on ModelSim simulation tool, and evaluation method is realized across a multi-core image processing example.

The remainder of the paper proceeds as follows. Section II deals with a number of issues that arise when designing such an on-chip network, and gives the details of the router as it is the central component of the NoC architecture. Section III shows some experimental results obtained by our router and NoC RTL models; it covers the simulation, synthesis and analysis results. Section IV presents the performance evaluation of the NoC through an on-chip JPEG compressor for multimedia applications, and also the impact of a zonal processing approach on image quality as well as processing

Yahia Salah, Med Lassaad Kaddachi, and Rached Tourki are with the Department of Physics, Laboratory of Electronics and Microelectronics, Faculty of Sciences, Monastir, 5000, Tunisia (corresponding author; phone: 216-558-72233; e-mail: Yahia.Salah@fsm.rnu.tn, Lassaad.Kaddachi@isigk.rnu.tn, Rached.Tourki@fsm.rnu.tn).

time is discussed. Finally, Section V summarizes the conclusions of the paper.

II. MICRO-NETWORK ARCHITECTURE DESCRIPTION

The on-chip interconnection network, as is exemplified by Fig. 1, consists of a set of routers (R) and point-to-point links interconnecting routers in a structured way. In the figure, the SoC comprises 16 IP cores, a 4x4 2D-Mesh/Torus NoC, and 16 network interfaces (NIs). The NI resource is needed to decouple the computation from the communication, enabling IP cores and interconnect to be designed in isolation, and to be integrated more easily. Its main function is the packetization/depacketization of data send over the interconnect. The basic element in the communication infrastructure of NoC is the router with a set of bi-directional ports, connecting to other neighboring routers and to a local IP core. It is responsible for forwarding and routing packets throughout the network from source to destination. In the following paragraphs, we illustrate the main issues of a micro-network in terms of topology, buffering strategy, routing algorithm and switching technique. We then detail the router structure, and also describe specific architectural mechanisms to the router design for supporting QoS.

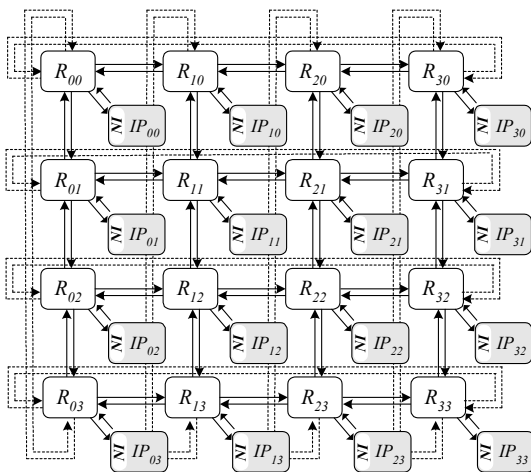


Fig. 1 A 4x4 2D-Mesh/Torus NoC architecture with connected IPs cores

A. NoC topology

Network topology comprises of an arrangement and connectivity of the routers. The selection of the appropriate topology has an effect on all NoC parameters, such as, network latency, routing cost, area and power performance. The 2D-Mesh is currently the most robust regular topology used for on-chip networks in core/tile-based architectures, because it perfectly matches the 2D silicon surface [12]. In addition to this, 2D-Mesh network topology can provide an acceptable wire cost and reasonably high bandwidth, for its simplicity of the XY routing strategy and modularity. Having a relatively high network diameter is a drawback of the mesh topology. The use of a 2D-Torus interconnect architecture reduces the 2D-Mesh network diameter, but requires

potentially costly wraparound NoC links along every dimension. The major reason to choose the two networks was scalability. A mesh or torus topology can be expanded to any size system by adding links and routing elements. Extending the network increases the aggregated throughput.

B. Buffering Strategy

The buffering strategy determines the location of buffers inside the router. Our NoC model adopts input queuing at routers and employs virtual channels (VCs) to improve the network bandwidth and latency. To avoid deadlock, the NoC architecture requires sharing each physical link by a number of virtual channels, and a scheduler determines at which times which queues are connected to which output ports, such that no contention occurs. The dimension of the queue is a significant parameter which influences directly packet latency and the switch area. It is very important to minimize the amount of buffering space. An optimal number of VCs in the NoC depends on the application and the size of the network.

C. Routing Algorithm

The routing algorithm is another important design choice. A dimensional ordered routing algorithm has been implemented in the NoC design because it offers deadlock-free and livelock-free operations with deterministic behavior. Besides, it is usually an easy way to keep the order of the network traffic. With the simplified router logic and interaction between routers, deterministic XY routing algorithm always provides low latency for the regular 2D mesh or torus NoCs [13]. Using XY routing, the packet follow the horizontal dimension first and then along the perpendicular dimension towards its destination. In each router hop, it compares the actual router address (X_L, Y_L) to the target router address (X_T, Y_T) of the packet, stored in the header flit. Network traffic is thus distributed non-uniformly over the mesh/torus links, but each link's bandwidth is adjusted to its expected load, achieving an approximately equal level of link utilization across the chip. This scheme is relatively simple and inexpensive to implement in hardware.

D. Switching Technique

The switching technique specifies how data and control are related. The need for efficient communication at low cost leads to adopting wormhole switching as the dominant switching technique in the NoC [14]. In wormhole routing, each packet consists of multiple fixed length control flow units, called flits. A flit is the smallest unit over which is performed the flow control. The flit size varies between the packet size and channel width according to the topology, architecture and protocol of NoC. The first flit of a packet, so called header, includes the routing information to establish a path between source and destination. Only the header flit needs to be routed. If it goes through a router successfully, the subsequent flits just follow it in pipeline fashion and without any more routing. So NoC guarantees in order packet's flits delivery, this contributes to simplify network interface structure. However, the wormhole scheme may produce the head-of-line (HOL) blocking problem when packets block

each other in a circular fashion in case of traffic congestion. In addition, it is more sensitive to deadlock and generally results in lower link utilization. Virtual channel concept, static routing and dynamic scheduling schemes can be used to avoid these problems.

E. QoS Capable Router Architecture

Our router architecture is generic in order to be adaptable with different QoS parameters. It is independently parameterisable in the number of input and output ports, size of flit and packet units, number of virtual channels, buffers and their depth. Furthermore, it is designed to be simple as possible to reduce area and to speed up the data forwarding process. Fig. 2 shows a block diagram of the router architecture.

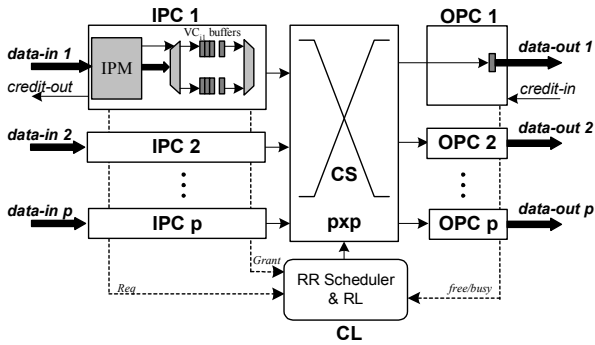


Fig. 2 Generic on-chip router architecture

The router essentially consists of an input port controller (IPC), an output port controller (OPC), a crossbar switch (CS) and a centralized control logic module (CL) performing flits scheduling and reconfiguration. Each IPC block has input VCs make possible the separation between different traffics despite the use of a common physical medium. These VCs provide the ability to deliver guaranteed communications throughput. Further, each IPC possesses its one input process module (IPM) as shown in Fig. 2. At the reception of an incoming flit, the IPM performs the XY routing algorithm to determine its appropriate output port, inserts the flit in the proper VC buffer, then sends a request to the round-robin (RR) scheduler to schedule this flit. The CS block provides data path between input and output ports. The access to the crossbar is controlled by the RR scheduler and reconfiguration logic modules. The OPC block maintains the status of output VC (free/busy) and keeps track of credits available in the down-stream router's VC buffer using credit based flow control. In our implementation, each OPC module contains one lane of one flit size.

The flow control mechanism between routers dominates the communication of the whole network. To avoid overflow in the router input queues (VC_{ip} buffers), a link-level flow control scheme is implemented by the flow control strategy and the reconfiguration process. These two mechanisms manage flit exchanges between neighbor routers. The VC implementation employs "credit-based" flow control strategy,

due to the advantages over handshake. Here, each router is initialized with the amount of free buffer space in the connected routers. Every time a flit is sent to a next router, the free buffer spaces counter (credit-in) corresponding to that destination port is decremented. When a router schedules a flit for the next cycle, it signals its predecessor that the free buffer spaces counter can be incremented (credit-out). The numbers of available flit cycles in the buffers of the next inputs ports are stored in a next buffer state table of the reconfiguration logic (RL). When a space is available, the RR scheduler module schedules flits that are buffered at the input ports and waiting for transmission to their appropriate output ports. The algorithm of the Fig. 3 provides more details on the proposed contention resolution mechanism for router output ports based on the RR technique that composes scheduler module.

Algorithm: RR packet scheduling to avoid conflict between traffic flows

```

p           : number of IPCs per router
Req         : sequence of IPC requests
Grant       : sequence of IPC grants
VCPr       : initial set of IPC priorities
t           : index time
1  begin RR scheduler
2  t ← 1
3  loop
4  //initialize
5  for i=0 to p-1 do
6  Grant(i) ← false           //no grant for VC-Queues i initially
7  end for
8  //find IPC with highest priority
9  j ← 0
10 for i=1 to p-1 do           //at least on VC-Queue i is not empty
11  if VCPr(i) > VCPr(j) then
12  j ← i
13  end if
14 end for
15 // start scanning for an IPC request from IPCs with highest priority
16 i ← 0, k ← j
17 while Req(j) ≠ true and i < p do
18  i ← i+1
19  j ← (j+i) mod p
20 end while
21 // update grants and VCPr
22 if Req(j) = true then
23  Grant(j) ← true
24  for i = 0 to p-1 do
25  VCPr(i)(t+1) ← (VCPr(i)t + 1 + j - k) mod p
26  end for
27 end if
28 t ← t+1
29 end loop
30 end RR scheduler

```

Fig. 3 Pseudo-code of RR scheduling policy

The algorithm operates on the principle that the IPC which was granted access to the shared OPC resource should have the lowest priority in the next round of arbitration. The IPCs can be pictured as being placed in a ring, where the priority of each IPC decreases linearly from the IPC with highest priority.

III. EXPERIMENTS

In this section, we present the design results of both router and NoC architectures. The used design methodology is based on the Very high speed integrated circuit Hardware Description Language (VHDL) at the RTL-level. This

description is simulated until obtaining the expected behavior. The ModelSim software is used to simulate and verify the functionality of our NoC components. ModelSim tool uses, in addition to VHDL description, a test file containing the system input stimuli in order to visualize the output behavior. The timing simulation was performed to verify the functional correctness of the design. Therefore, the RTL simulation waveforms are included in this section.

The synthesis of the NoC router was performed with the Xilinx ISE tool targeting Xilinx Virtex-II technology with XC2V1000 FPGA target device.

In the first part of this section, we present HDL simulation results of the 2D-Mesh/Torus NoC architecture. In the second part, we present the synthesis results of the proposed router architecture.

A. Simulation Results

The developed router can establish more than one connection at the same time. It can simultaneously handle up

to five connections. A request processing requires a control time of 40 ns after reset. One cycle is needed to switch a flit from an IPC block to its corresponding OPC block.

The packet transmission in the 4×4 2D-Mesh/Torus NoC topology of the Fig. 1 was validated by a functional simulation. The choice between Mesh and Torus modes is made by the routing algorithm according to contents of the received packet header information. Torus mode is applied when at least one router (source or destination) is a corner or side type of the Mesh/Torus network. Otherwise, Mesh mode is executed. Fig. 4 illustrates the transmission of this packet from router R_{00} (source) to router R_{32} (destination). In fact, the input and output interface behaviors of routers R_{30} and R_{31} are shown in the simulation results. We notice that the figure illustrates the successive data output of the connection C_{00-32} along 24 clock cycles throughout all the four routers and shows the granted requests at each cycle.

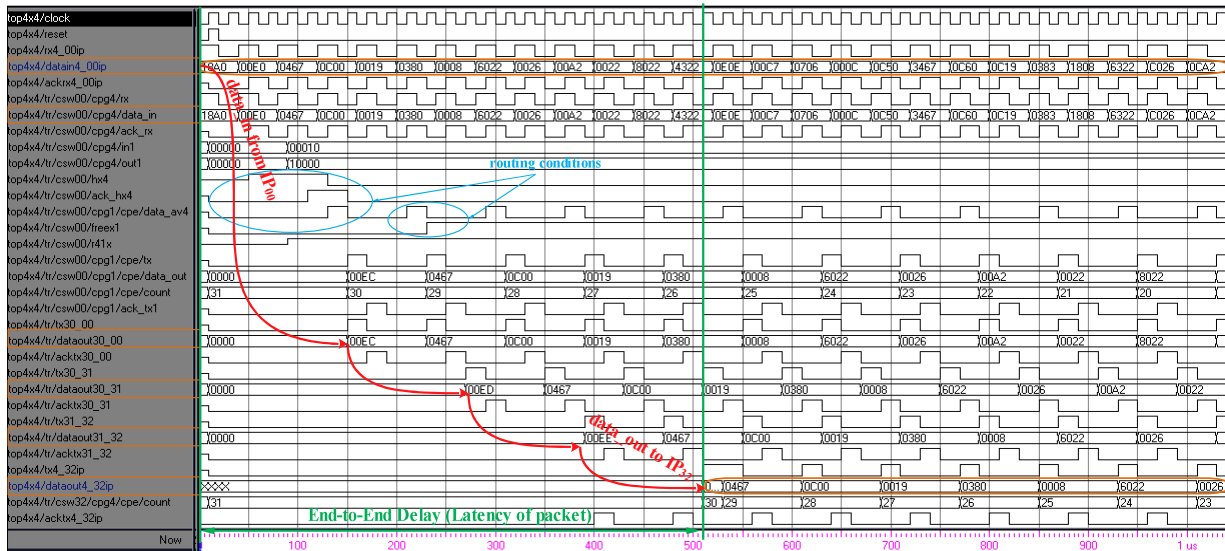


Fig. 4 Simulation waveform during packet transmission in 4×4 2D-Mesh/Torus topology

B. Synthesis Results

The synthesis results of the five ports router architecture are shown in the Table I. In this experiment, the physical network link is multiplexed by four virtual channels according to the desired QoS levels. The depth of each input VC buffer is fixed to 2 flits. The output VC buffer contains one lane with a size of one flit. Thus the total buffer budget of the router is equal to 45 32-bit flits with a NoC link width of 32 bits. The area requirement of our architecture is about 9.16% of the space of our Virtex-II.

The resource utilization break-down of the router component is shown in Fig. 5. As seen, the IPC units for all the five input ports combined consume around 56% of each router's resources, primarily due to the memorization resources, then to the IPM and control modules for interfacing with other router components. Note that the area occupation of

each IPC unit increases with the number of VCs per port and the depth of each VC buffer in flits. The second largest consumer of router's hardware resources is CS unit that accounts for about 17% of the total router area. CL unit, 13%, is the third largest component of router area. It made up of the scheduler and reconfiguration logic modules. The resources consumed by the CL unit increases with the number of router ports and VCs per port.

TABLE I
HARDWARE IMPLEMENTATION RESULTS OF THE ROUTER

Resources	Used	Accessible	Percentage of use
# Slices	496	5.120	9.16 %
# FFs	424	10.240	4.14 %
# LUTs	877	10.240	8.56 %
# IOBs	88	324	27 %
# GCLKs	1	16	6 %

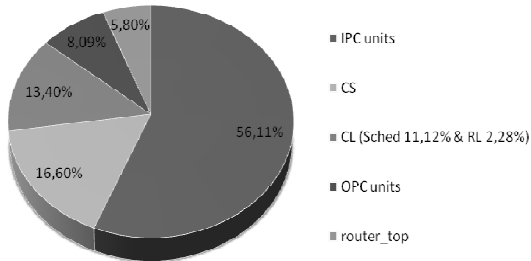


Fig. 5 Pie chart of area distribution for internal components of the router

As the design is pipelined, the block in the critical path with highest delay determines the maximum clock frequency of the circuit. In this particular case it is the centralized CL unit, due to its scheduling complexity, which imposes a cycle time of 0.5 of the total max delay. For the IPC unit we cover a cycle time of 0.25 of the max internal delay, taking into account the impact of the time spent in buffer management. The same delay value of CS unit is needed to forwarding a flit from an IPC to its corresponding OPC unit. For each number p of input-output ports router, Fig. 6 presents the propagation delays of the CL unit, IPC unit, and the critical path delay of the router.

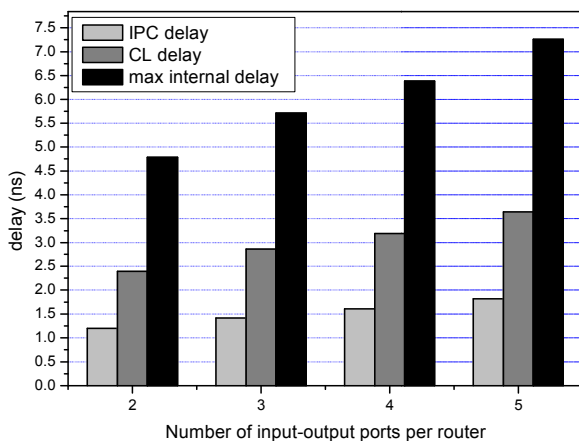


Fig. 6 Critical path delay in function of the number of input-output ports of the router

Summarizing the data, a 2D-Mesh/Torus router design supporting the proposed techniques requires 9.16% of XC2V1000. It takes one clock cycle to buffering and determinate the appropriate OPC of an incoming flit, two clock cycles to model the max internal delay for a header flit and one cycle to forwarding a flit from an input port to its corresponding output port. For a 5-port router with 32 bits-wide links, the critical path delay is 7.14 ns, i.e. the router operates at about 140 MHz allowing a theoretical peak performance of 22.4 Gbits/s. Thus, the comparison between our design results and the others (like work proposed in [6]) shows that our router provides high performances at low cost.

We now proceed to evaluate our micro-network architecture with an example of multi-core system-on-chip.

IV. ON-CHIP JPEG COMPRESSOR

The NoC evaluation is performed on an example of multimedia applications to conform the efficiency of design issues and the capability of the proposed router architecture. In order to achieve this purpose, a JPEG image compression algorithm is used. It consists of nine IPs cores, responsible for compressing, decompressing and transmitting images. Table II describes the main task of each IP core.

IP core	Description
IP ₀₀	Binarization
IP ₀₁	Divide to block 8×8
IP ₀₂	Color Transform
IP ₁₂	2-D Zonal DCT
IP ₁₁	Quantization
IP ₁₀	Entropy Coding
IP ₂₀	Decoding
IP ₂₁	Inverse Quantization
IP ₂₂	Inverse DCT

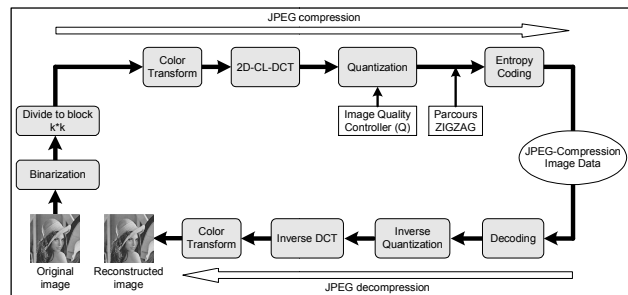


Fig. 7 Block diagram of the images compression/decompression architecture

To achieve compression, we have employed our encoding scheme presented in [15]. The image compression encoder that we propose combines the best lifting DCT (Discrete Cosine Transform) algorithm of the literature with an efficient zonal processing approach in order to optimize the number of operations per coefficient and to reduce the number of coefficients to be processed and encoded. In our approach, we have adopted the Cordic-Loeffler (CL) algorithm in the 2-D 8-point DCT, with 38 additions and 16 shifts, because it offers the best trade-off between the computational complexity and the image quality. In order to decrease even more the computational cost of the CL-DCT, we derived a zonal form of the 2-D 8-point DCT algorithm. Details can be found in [16]. Fig. 7 summarizes all the steps of the images compression and decompression. The proposed architecture for images compression and transmission was partitioned in three main IP blocks. These three blocks are 2-D Zonal DCT, quantization, and RLE/Huffman coder. We have defined an internal Zigzag table which changes radically with the size k used for Zonal DCT [15]. The zigzag table receives as input

the quantized 2-D DCT coefficients and feeds the data to the RLE/Huffman in a zigzag order. Zonal DCT consists in computing only the most significant DCT coefficients, in other words the low frequency coefficients. Note that JPEG decoding performs the reverse of the aforementioned steps.

The major challenge in the use of the proposed zonal approach is the choice of the best value of parameter k which guaranties a reasonable trade-off between processing time and image distortion. In this part we'll focus on the quality of reconstructed images as well as processing time of the proposed compression encoder. We try to select the optimal value of k which offers best trade-off between processing time and visual distortion. In order to speed up the processing of the JPEG images compression/decompression hardware architecture, a 3×3 2D-Mesh/Torus NoC structure is used.

The cores are manually mapped on the network topology. Our NoC uses the minimal XY routing algorithm, wormhole flow control and input virtual channels queuing. Each router contains 4 VCs, and each input VC buffer is two 32-bit flits deep. Further, each router is connected to its neighbor routers by four bi-directional ports, and also connected to a network interface. To increase the compatibility and to ease the reuse of the NoC architecture, the OCP-IP standard is used [17]. Benefiting from the OCP standard, each IP core with OCP interface can be easily connected on our micro-network architecture.

The JPEG SoC is modeled with standard VHDL language at the RTL-level and implemented on a Xilinx FPGA device using ISE 9.2i tool. The target device is a FPGA Virtex-V XC5VL330T.

The synthesis results show that design of the 3×3 2D-Mesh/Torus NoC occupies 9.23% of XC5VL330T and 12.58% of the SoC device area. Each router of the NoC occupies less than 1 % of the SoC area and has an average latency of 4 cycle's clock. For a 32-bit-wide link, it operates at a frequency of 140 MHz that performs five different transfers per cycle. Because the maximum number of routers to traverse is 3 (the source and destination routers are included in the topology), two cores running on the device with a frequency of 133 MHz can send and receive the packets without delay in their execution, if the network is free. The used cores, placed on the device, are large components and they cover 87.42% of the total area used by the SoC. The area results correspond to these cores are shown in the Fig. 8. The largest consumers of SoC's hardware resources are 2-D DCT and Inverse DCT cores that accounts for about 39% of the total SoC area, followed by Quantization, Decoding, and Inverse Quantization, which occupy around 33% of the SoC area.

In order to evaluate the impact of the zonal CL-DCT approach on image quality at the receiver level, we have measured the image distortion that the encoders cause using the MATLAB tool. The main metric used to this purpose is the peak signal-to-noise ratio (PSNR). Processing time of the proposed compression algorithm is a second metric that affects the system response time, obtained using the ModelSim simulation tool. It is determined by the processing delay consumed by IP cores that make-up the JPEG images

compressor and the network latency. This last parameter denotes the difference between the time at which the first flit of a packet has been offered to the network and the time the last flit has been delivered to the destination. To demonstrate the efficiency of both NoC and JPEG coding system architectures, the reference Lena image was used. It is 512×512 pixels with 8 bits per pixel. Table III shows the performance reached by the NoC-based encoder for different values of k .

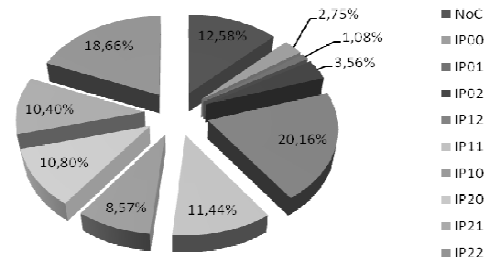


Fig 8 Area breakdown of the NoC-based SoC

TABLE III
IMAGE DISTORTION AND PROCESSING TIME OF LENA ENCODED AT 0.5 BPP FOR DIFFERENT VALUES OF K

	PSNR (dB)	Processing time (s)
$k=2$	22.48	2.23
$k=4$	27.06	2.67
$k=6$	28.02	3.13
$k=8$	28.23	3.55

From the above table, even if the overall visual quality is good, image distortion increases as k decreases. The image distortion caused by the proposed algorithm remains good for values of k from 4 to 8. On the other hand, for large values of k (i.e. 6 and 8 here), the proposed algorithm offers better performance, which is normal since more coefficients are processed.

To discuss the processing time needed for the image compression and packet transmission process, we present, in the same table, the impact of the parameter k on the overall processing time. The table shows that the processing time is significant when we increase the parameter k (i.e. 6 and 8 here). Note that the network latency has a large influence on this metric. The lower NoC latency offers shorter processing time to move data through the network.

The best performance of the proposed image compression algorithm in terms of distortion ratio is mainly obtained by the zonal encoding-based approach, especially for the cases where $k=4, 6$ and 8 . In order to achieve higher picture quality during the packet transmission, we have to use the lower parameter k at higher network throughput (the maximum network data rate is 28.8 Gb/s @ a core clock frequency of 133 MHz).

As a general interpretation of these results, we note that the decrease in PSNR is expected for low values of k . On the other hand, for a higher network data rate and large values of k (i.e. 6 and 8 here), the proposed algorithm offers higher picture quality. This is normal since more coefficients are processed. However, the processing time becomes significant.

Therefore, based on the results obtained from the proposed image encoder, we can conclude that the case where $k=4$ offers best trade-off between image distortion and processing time.

V. CONCLUSION

In the first part of this paper, we introduced the details of our proposed router-based micro-network architecture used for multi-core embedded systems. The key research issues for network-on-chip (NoC) and specific architectural mechanisms to the router design were discussed. Then, we present the results of the implemented NoC components. According to the simulation and synthesis, the proposed router provides high performance for low-cost FPGA implementation. Last, we evaluated both our NoC architecture and the image processing algorithm in a JPEG-based image compression SoC used for multimedia applications. Flexibility of routers makes them adaptable with different quality-of-service (QoS) parameters, which can radically change with the image compression algorithm steps. Moreover, the higher bandwidth and lower NoC latency let speed up the images forwarding process and thus offer shorter processing time to move data through the network. The experiments present a reasonable trade-off between processing time and image distortion.

REFERENCES

- [1] E. Bolotin, I. Cidon, R. Ginosar, A. Kolodny, "QNoC: QoS architecture and design process for network on chip," *Journal of Systems Architecture*, vol.50, no. 2-3, pp. 105–128, 2004.
- [2] M. Ali, M. Welzl, M. Zwicknagl, "Networks on Chips: Scalable Interconnects for Future Systems on Chips," in *Proc. of the 3rd IEEE International Conference on Circuits and Systems for Communications*, 2006, pp. 1–6.
- [3] M.A. Al Faruque, J. Henkel, "QoS-Supported On-chip Communication for Multi-Processors," *International Journal of Parallel Programming*, vol. 36, no. 1, pp. 114–139, 2008.
- [4] B. Grot, et al., "Kilo-NOC: a heterogeneous network-on-chip architecture for scalability and service guarantees," in *Proc. of the 38th ISCA*, 2011, pp. 1–12.
- [5] W.-C. Tsai, Y.-C. Lan, Y.H. Hu, S.-J. Chen, "Networks on Chips: Structure and Design Methodologies," *Journal of Electrical and Computer Engineering*, pp. 1–15, 2012, DOI:10.1155/2012/509465.
- [6] S.A. Asghari, H. Pedram, M. Khademi, P. Yaghini, "Designing and Implementation of a Network on Chip Router Based on Handshaking Communication Mechanism," *World Applied Sciences Journal*, vol. 6, no. 1, pp. 88–93, 2009.
- [7] J.J.H. Pontes, M.T. Moreira, F.G. Moraes, N.L.V. Calazans, "Hermes-A – An Asynchronous NoC Router with Distributed Routing," in *International Workshop on Power and Timing Modeling, Optimization and Simulation*, 2010, pp. 150–159.
- [8] S. Saponara, L. Fanucci, M. Coppola, "Design and coverage-driven verification of a novel network-interface IP macrocell for network-on-chip interconnects," *Microprocessors and Microsystems - Embedded Hardware Design*, vol. 35, no. 6, pp. 579–592, 2011.
- [9] T. Bjerregaard, J. Sparsø, "Implementation of guaranteed services in the MANGO clockless network-on-chip," *IEE Proc. Computers and Digital Techniques*, vol. 153, no. 4, pp. 217–229, 2006.
- [10] K. Goossens, et al., "The Aetheral network on chip: Concepts, architectures, and implementations," *IEEE Design and Test of Computers*, vol. 22, no. 5, pp. 414–421, 2005.
- [11] M. Millberg, E. Nilsson, R. Thid, A. Jantsch, "Guaranteed bandwidth using looped containers in temporally disjoint networks within the Nostrum network on chip," in *Proc. of DATE'04*, 2004, vol. 2, pp. 890–895.
- [12] R.N.R. Mohammad, K. Reza, "Performance Comparison of 3D-Mesh and 3D-Torus Network-on-Chip," *Journal of Computing*, vol. 4, no. 1, pp. 78–82, 2012.
- [13] M. Valinataj, S. Mohammadi, S. Safari, "Fault-aware and Reconfigurable Routing Algorithms for Networks-on-Chip," *IETE Journal of Research*, vol. 57, no. 3, pp. 215–223, 2011.
- [14] M. Tang, X. Lin, "Rqrt: Reduce Querying Routing Table for Mesh-Based Network-on-Chip," *Journal of Circuits, Systems, and Computers*, vol. 20, no. 8, pp. 1529–1545, 2011.
- [15] Med. L. Kaddachi, L. Makkaoui, A. Soudani, V. Lecuire, J.-M. Moureaux, "FPGA-based image compression for low-power Wireless Camera Sensor Networks," in *Proc. of the 3rd International Conference on Next Generation Networks and Services (NGNS 2011)*, December 2011, pp. 68–71.
- [16] L. Makkaoui, V. Lecuire, J.-M. Moureaux, "Fast Zonal DCT-based Image Compression for Wireless Camera Sensor Networks," in *Proc. of the IEEE international Conference on Image Processing Theory, Tools and Applications (IPTA 2010)*, July 2010, pp. 126–129.
- [17] <http://www.ocpip.org>.

Yahia Salah was born in 1979 in Mahdia, Tunisia. He received the Master Degree in Micro-electronics and the Ph.D. in Physics (Electronics option) from the Faculty of Sciences at the University of Monastir (Tunisia) in 2005 and 2012 respectively. He is currently Assistant Professor at the Higher Institute of Industrial Systems of Gabes. His research includes interconnect design for Systems-on-Chips, with particular emphasis on developing quality of services-based design methods for Networks-on-Chip IPs. His interests also include high-level modeling of real-time and embedded systems.

Med Lassaad Kaddachi received his Ph.D. (2012) in Electronic and Electrical Engineering from the Monastir University. He is currently an Assistant Professor at the Higher Institute of Computers Sciences and Management of Kairouan (Tunisia). His research activity includes QoS management in real time embedded system and multimedia applications. He focuses mainly on the design and performances evaluation of hardware solutions for communication systems with multiple constraints.

Rached Tourki was born in Tunis, on May 13 1948. He received the B.S. degree in Physics (Electronics option) from Tunis University, in 1970; the M.S. and the Ph.D. in Electronics from Orsay Electronic Institute, Paris-South University in 1971 and 1973 respectively. From 1973 to 1974 he served as Microelectronics Engineer in Thomson-CSF. He received the Doctorat d'état in Physics from Nice University in 1979. Since this date he has been Professor in Microelectronics and Microprocessors with the physics department in Faculty of Sciences of Monastir. His researches interest, DSP and Hw-Sw co-design for rapid prototyping in telecommunications.