A Novel Architecture for Inter-FPGA Traffic Collision Management

Atef Dorai, Virginie Fresse, El-Bay Bourennane, Abdellatif Mtibaa

A Novel Architecture for Inter-FPGA Traffic Collision Management

Atef DORAI* †‡, Virginie FRESSE*, El-Bay BOURENNANE†, Abdellatif MTIBAA‡
*Hubert-Curien Laboratory, UMR CNRS 5516, University of Lyon, 42000 Saint-Etienne, France
†Le2i Laboratory UMR CNRS 6306, University of Bourgogne, 21078 Dijon, France
‡EmicroE Laboratory LR99ES30, University of Monastir, 5019 Monastir, Tunisia
Email:{atef.dorai},{virginie.fresse}@univ-st-etienne.fr

***Abstract*—with the increasing complexity of various communications and applications, Network-On-Chip (NoC) is one of the most efficient communication structures. Multi-FPGA platforms are considered as the most appropriate experimental solutions to emulate a large size of MPSoCs (Multi-Processor System-on-Chip) based on a NoC. The deployment of the NoC into several FPGAs requires the use of inter-FPGA communication links. The number and performance of external links restrict the bandwidth of communication. Currently, the number of inter-FPGA signals is considered as a substantial problem in NoC implemented on Multi-FPGA architectures.**

In this paper, we propose the integration of the collision management architecture connected to the NoC. Two collision avoidance algorithms are proposed in the structure to balance the load injected between all routers connected with one external link. This architecture leads to high timing performances in multi-FPGA system communications. The results demonstrate the efficiency of the collision management structure connected to the NoC. The collision management algorithm is chosen according to the type of inter-FPGA communication requirements.

***Index Terms*—Traffic Collision, NoC, Inter-FPGA, Multi-FPGA.**

I. INTRODUCTION

The large-scale integration of embedded cores in MPSoCs design will increase in the future years [1]. For example, Intel has developed a co-processor called knights corner has 50 cores [2], NVIDIA has announced a multicore-chip called Fermi that has over 512 CUDA cores [3]. To support the numerous parallel data transmission and high bandwidth, communication infrastructure on-chip is needed.

MPSoCs using a NoC are the most efficient architectures for many core applications (i.e. applications with hundreds of cores). The verification of such complex architectures can be handled with prototyping process on programmable devices such as FPGA. The resources of one FPGA are not enough to handle the complete MPSoC based NoC architecture. Multi-FPGA platforms are then required. Implementing a large size of NoC on such platforms requires to partition the NoC into subnetwork and to connect each FPGA with external links. The number of communication protocols (when using a plat-form) or the available I/O pins (when using the FPGA only) is restricted. Several nodes of the NoC must share external links, reducing the bandwidth and creating collisions when several nodes require the access at the same time. For example, the virtex-6 XC6VLX195T FPGA has 600 I/O pins. Prototyping a NoC can lead to connect 15 routers with 16-bit data or 8 routers with 32-bit data. If the number of routers to connected is higher, sharing the external links is required. This reduces the bandwidth between each FPGA creating collisions. However, network topologies have high impact on the size of NoC prototyped. The mesh topology with the higher level of interconnect is not easily scalable on multi-FPGA [4].

In this paper, we propose a collision management structure containing two kinds of collision management algorithm inspired from the computer network. In this structure, the external access is done using an Access Point (AP) and an Access Protocol block. The collision management algorithms are the Backoff algorithm and the weighted round robin algorithm adapted from computer network to NoC on multi-FPGA architectures. Some experiments are conducted to compare both approaches and to help the designer to select the most appropriate algorithm according to the communication requirements.

This paper is organized into five sections. Section II presents the background and related works related to the deployment of the NoC on multi-FPGA platform. Section III presents the proposed architecture dedicated to deal with congestion phenomena. This section also details both types of collision management used and the strategies. Section IV presents the experimental results. Comparisons between the Backoff algorithm and weighted round-robin arbiter are also presented in this section. Finally, Section V concludes the paper.

II. BACKGROUND

A. Related work

NoC communication on multi-FPGA architecture provides high scalability, high performances and low power consumption [5]. Few works have been achieved to evaluate the performances of a large NoC. Explorations made in [6] show the limitation of the size of a basic NoC on a single FPGA. Multi-FPGA platform must be considered in a system containing hundreds of cores. Two different approaches for the deployment have emerged. The first approach consists in partitioning the 2D NoC on each FPGA and replaces inter-FPGA links by external serial or parallel buses [7] [8]. This approach ensures the evaluation of communication performance between two FPGAs as each link remains. However, it is difcult to deploy a large NoC as there is a restricted number of external links. In

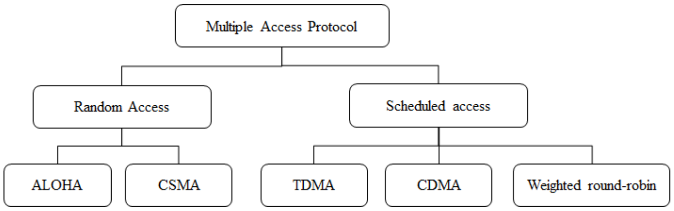


Fig. 1. Access-protocols in computer network

[12] the NoC system contains two parts (the network and the processing system). Each partition is mapped onto different FPGA. This work used the region based partition where each region is composed of a set of processing cores and their routers. The resource chip is composed of four surrounding FPGAs and the network FPGA is the middle FPGA.

The second approach consists in developing a structure dedicated based on a hierarchical 3D concept [9] [10]. Another solution in [11] presents the hierarchical network architecture intended for multi-FPGA. This proposition is based on the tree topology. It consists of three levels of hierarchy (local network, cluster and system). The synthesis results show a loss of 20% of the occupied hardware for the NoC in case of the implementation on the Xilinx Spartan-6 chip. Another methodology is proposed in [6] to deploy the 2D NoC onto multi-FPGA platforms with a restricted number of external links. The deployment and the associated adaptations depend on both the number of FPGA routers (NR) to be connected to the external links and the number of inter-FPGA links (NLI). For a large size of NoC, the number of routers to be connected to one external link is huge, creating collision problems when several routers want to send data at the same time in external link.

The collision avoidance and management for large NoC based SoC on FPGA has not been yet proposed in literature. The contribution of the paper is to propose some collision management solutions to distribute the shared external links reducing the collision and increasing the bandwidth.

B. Management of collision in computer network

In the OSI reference model, the Media Access Control (MAC) protocol is positioned at the second layer in computer network. It plays a very important role to control the access for the external users. The MAC protocol can be classified into two majors groups as shown in Figure 1. The first group contains methods access random using ALOHA and Carrier Sense Multiple Access (CSMA) based on Backoff algorithm. The second group contains scheduled access methods. Common access-protocol supporting scheduling mechanisms for computer network are: Time Division Multiplexing, Round-Robin and Weighted Round-Robin [14]. Two multiple access protocols will be used in our work, a random access algorithm based on the Backoff algorithm and a scheduled access algorithm based on the Weighted round-robin algorithm.

a) Backoff algorithm (BO): It is one class of collision resolution algorithm used in the medium access control. When different users compete to access a shared link at the same time, the collision can happen. The backoff algorithm is based on a priority system and on the timing computation to manage transmissions/retransmissions.

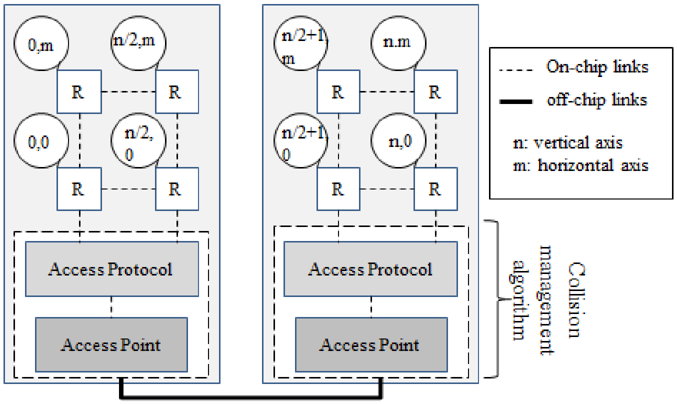


Fig. 2. Collision management algorithm based-NoC on multi-FPGA

b) Weighted-Round-Robin (WRR): This scheduling is an extension of Round-Robin (RR) scheduling. The Weighted Round-Robin (WRR) scheduling algorithm is based on the round-robin and priority scheduling algorithms. The WRR retains the advantage of round-robin in eliminating starvation and also integrates priority scheduling.

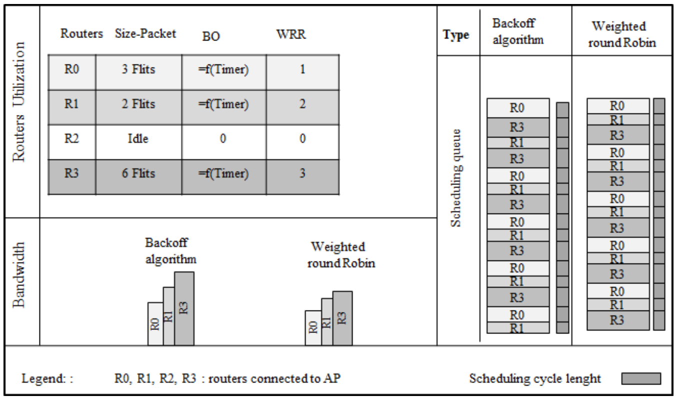
III. COLLISION MANAGEMENT STRUCTURE FOR NOC ON MULTI-FPGA

A large NoC is considered to be deployed on the multi-FPGA platform. Many nodes (i.e. routers) sharing one external link compete to have access to the external link. Some collisions can occur if several routers want to send data to the external link at the same time. A collision management structure is therefore proposed to avoid such collisions and to efficiently distribute the access to the external link. The collision management structure proposed is inserted between the NoC and the external FPGA link. It can control and manage the access several cores connected to one external link. This structure is constituted of two main blocks: the access-protocol and the Access-Point (AP), as depicted in Figure 2.

A. Access-Point (AP)

The Access Point (AP) is an interface that provides connectivity between the access-protocol and the external link. It integrates unidirectional buffer for send and receive data. If a router wants to send packets to the external link, the Access-Point checks if the external link is free. If the link is free, the packet is sent to the AP for transmission. If several routers want the access to the external link, the AP receives the packet from the router selected by the Access Protocol block.

The AP is adapted to the type to the external link used. The communication between two AP uses the handshake flow control in the experiments conducted in the paper. Reducing the number of APs can lead to the optimization of the number of pins used.



Routers	Size-Packet	BO	WRR
R0	3 Flits	=f(Timer)	1
R1	2 Flits	=f(Timer)	2
R2	Idle	0	0
R3	6 Flits	=f(Timer)	3

Fig. 3. Bandwidth communication for both access-protocol

B. Access-protocol

The access-protocol block is inserted between the routers and the AP. It contains one block dedicated to select the router amongst many routers requesting the access. The access-protocol use packet-switching flow control, this presents an attractive solution for communication intensive when the number of router is as high as 8 [15]. This block integrates a collision management algorithm extracted from computer network and adapted to NoC on multi-FGPA. Two algorithms are proposed: the backoff algorithm (BO) and the Weighted Round Robin (WRR) algorithm. Both are used in the experiments for comparisons and to help the designer to select the most appropriate algorithm. The Access Protocol allocates the time for each router when it has access to the AP [13]. The allocated time depends on the algorithm and the packets to be sent. The access-protocol contains an intelligent data allocation for time-varying and for monitoring the real-time load traffic for each packet. The timing vector allocation is first detailed, then the collision management algorithms are presented.

c) Vector allocation time-varying: The flows from the NoC to the AP are time-varying. It is therefore necessary to allocate the time inside AP for each packet incoming from routers. Each packet inside the NoC contains a header and a payload (containing data). Each packet to be sent to the AP has different size (depending on the payload, the header is identical) arriving at different rate. The information in the header of packet is extracted to set the allocation phase. Each packet is stored in the local input port of the router, all the received packets connected to the AP goes first to the Allocation process (given in algorithm 1) for the timing allocation.

The router allocation depends on the algorithm used as depicted in Figure 3. An example of 4 routers connected to the AP is presented (R0 to R3). Each packet has a specific size of packet and the number of packet to be sent is different. The variability of packets depends on the algorithm mapped onto the NoC. The size of packets is extracted and used to allocate the required time in AP (considering that the timing allocation is given to send one complete packet). At a specific time, some routers do not require to send packet to the AP (R2 is in idle state as there are any packet available in the local port), other with different size of packets do. The difference between both algorithm is the scheduling queue in the AP as the allocation value differs from each other.

Algorithm 1: Pseudo code for mechanisms allocation

```
Require: P: Input packet
Ensure: Time: Time allocate for each packet Initialization;
  for $i \leftarrow 0$ to $n-1$ do
    $FlitNumber[i] \leftarrow \emptyset$;
    $NumberFlit[i] \leftarrow \emptyset$;
  end
  for $i \leftarrow 0$ to $n-1$ do
    if (rx[i]='1') then
      $FlitNumber[i] \leftarrow FlitNumber[i] + 1$;;
      if (FlitNumber[i]=1) then
        $NumberFlit[i] \leftarrow conv\_integer(P[i]) + 2$;
        $Time[i] \leftarrow NumberFlit[i]$;
      end
      if FlitNumber[i]=NumberFlit[i]+2 then
        $FlitNumber[i] \leftarrow \emptyset$;
        $Time[i] \leftarrow \emptyset$;
      end
    end
  end
```

The algorithm describes the process of the time allocation as shown in algorithm 1. The array of variables called FlitNumber is initialized to zero. In this example, the routers connected to the AP are numbered 1 to n, i is an index whose value is between 0 and (n-1). A new request maintains the list of routers that are connected to the access-protocol by the input signal (rx). The information is in the header of the packet and is extracted by the process. This algorithm analyzes the data and extracts the size of packet. This time allocation value is sent as an input of the access-protocol. However, if the output port of one router connected to the access-protocol is empty, the FlitNumber variable becomes equal to NumberFlit at the end of the transmission. The FlitNumber is set to zero and the index i is moved to another router that will start transmission.

d) Backoff algorithm: The structure of the access-protocol using the Backoff algorithm is depicted in Figure 4. It consists of two blocks, one for transmission and another one for reception. The transmission block contains a priority arbiter, a multiplexer, a counter for the flits of packets and the Backoff block. The reception part includes the demultiplexer, the counter and the routing algorithm.

Each router that wants to send packets to the AP, sends a request signal to the access-protocol. If there is only one request from one router, the arbiter receives the request signal and sends the allocation-time associated to the packet to the AP. The time is set using this allocation time to send the full packet to the AP. The accesss-protocol transmits the packet to the AP by configuring the multiplexer. When the counter value reaches zero, the connection between the transmitted node and

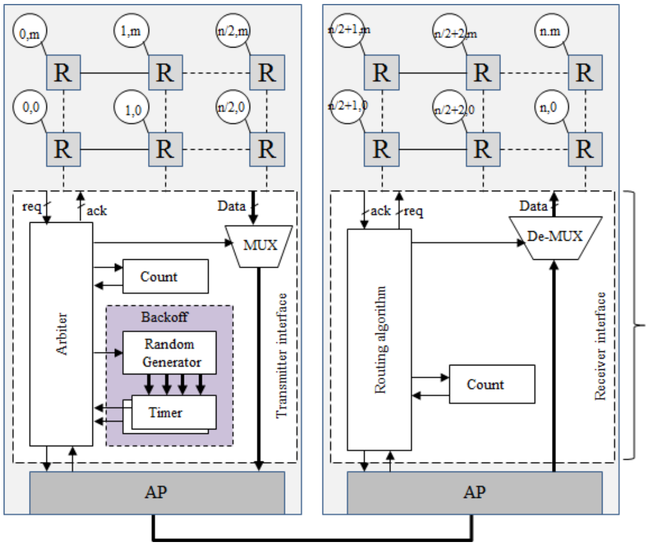


Fig. 4. Structure of Backoff algorithm based NoC multi-FPGA

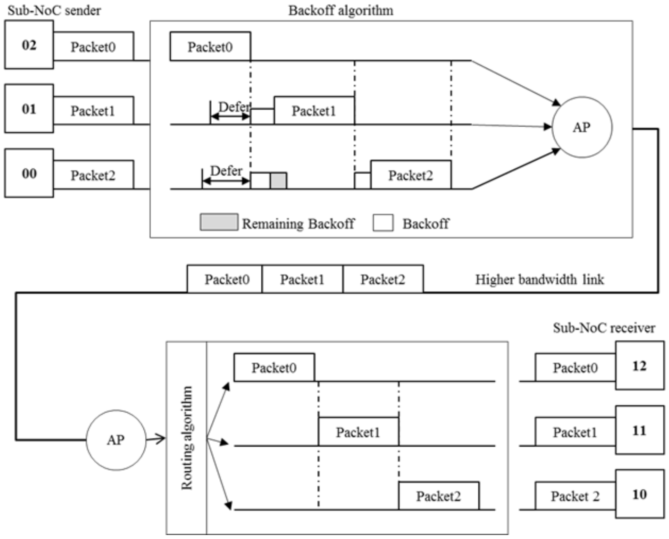


Fig. 5. Backoff procedure

the AP is closed and the arbiter will return to the idle state, waiting for new requests incoming from routers.

The Backoff algorithm model contains 2 major modules which are a Random generator and a Backoff-Timer. The Backoff-Timer accepts the values of unsigned integers from Random generator module. If several routers request the access to the AP, the random generator supplies random values to each requesting router. The Backoff-Timer module generates random values of integer from 1 to 255. This module is based on the concept of linear-feedback shift register (LFSR). The total number of random states generated on LFSR depends on the feedback polynomial. As it is a simple counter, so the maximum value is (2^n-1) by using maximum feedback polynomial, where n is the number of shift registers used in this design. Thus, an LFSR is most often a shift register whose its input bit is driven by the exclusive-or (XOR) of some bits of the overall shift registers value.

If the external link is busy while routers request a transmision, their transmissions are deferred until the end of the current transmission and a randomly selected Backoff-Timer period. Figure 5 shows an example of a data transmissions scenario in the proposed Backoff algorithm. The NoC in this example have 3 concurrent routers resquesting accesses to the AP. Multiples routers are deferred and selected according to the random Backoff. The router having the smallest random Backoff-Timer will win the transmission as the waiting time is smaller. Once the current router had finished the transmission, the remained routers start decrementing their Backoff-Timer. The router which finish to decrement its Backoff-Timer, can start its transmission. This mechanism is repeated until all routers have accessed one by one to the external link.

e) weighted round-robin: Each router initiates a request signal to get access to the external link and deactivate the request signal when the transmissions is finished. If more than one router request for the external link at the same time, the access is granted on the round-robin basis. For each packet, a weight is set, this weight indicates the time allocated to its associated router inside the AP. The weight depends on the size of packets (number of flits inside a packet). The weight 0 is set to routers that do not request to send packet (idle state).

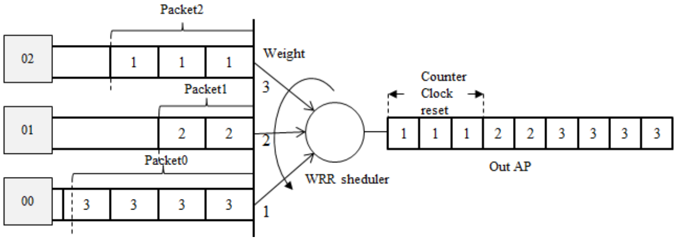


Fig. 6. Weighted Round-Robin procedure

An example is given in Figure 6. All routers resquest the access to the AP. The weights are given according to the position of the router (weight 1 for first router, weight 2 for the second and weight 3 for the third). The access is given to the first router with a time depending on the size of the packet. The first packet contains 3 flits, the counter will count 3 times to send one flit to let the AP to send the complete packet.

This procedure ensures that any independent router is locked out while another router has the continuous access. The continuous access is granted to any router for a period of time. The weighted round-robin scheduler is designed to serve different routers capacities. In WRR, each router-queue has a counter that specifies the number of its that can be sent from it. The use of the AP is optimized as the allocation is based on the size of packets.

IV. Experiments

In this section, some results are presented by using the collision management algorithms. They are based on the

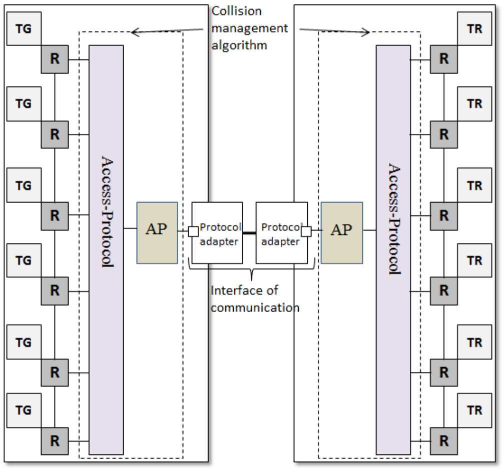


Fig. 7. Emulation platform designed on multi-FPGA

analysis of resources and timing on multi-FPGA platform. These experiments are conducted to analyze the impact of the deployment on the BO/WRR-based NoC in terms of resources and timing.

A. The prototyping platform

Experiments for the collisison management algorithm based of both access-protocols are done on a multi-FPGA platform using ML605. The ML605 boards contains a Virtex-6 FPGA, containing 301 440 registers, 150 720 LUTs (Look-Up-Table) and 600 IOBs (Input Output Blocks). The simulation and Place and Route (P&R) use respectively ModelSim 6.5 and Xilinx ISE 13.1 design Tools (with XST synthesis tool). The Hermes NoC [16] is used to validate the new architecture. This is based on a mesh topology with the wormhole flow control. The determinist XY routing algorithm is used. The architecture of sub-NoC used in each FPGA have the size 1xN, where N = 2, 4, 8, 12, 16 is the number of nodes in the vertical axis. The inter FPGA communication is done by the collision management algorithm that based on a packet switching. Resource utilization experiments are done for routers (16-bit flit, 16-flit buffer).

The proposed collision management algorithm was evaluated for synthesis traffics and real application patterns. The emulation platform is designed. It is based on the Hermes NoC deployed on 2 ML605 platforms and the collision avoidance architecture inserted between the sub-NoC. Emulation blocks are inserted in each local port of the sub-NoCs. Emulation block are traffics generators (TG) and traffics receptors (TR), as shown in Figure 7. The TR can send packets to the NoC from each local port. The number and size of the packets is parameterized. The data injection rate (idle time) between packets can be adapted to. The traffic receptor inserts the clock cycles number in each packet when the packet is sent. The traffic receptor receives packets, extracts the timestamp and defines the latency of the communication from the traffic generator to the traffic receptor. The performance metrics used are the minimal and maximal latency and the average latency. The minimal latency is the time elapsed from the moment the first message (i.e. all packets) is received by the destination node. The maximal latency is the time required for last router to get the latest message. The average latency for all routers is also defined.

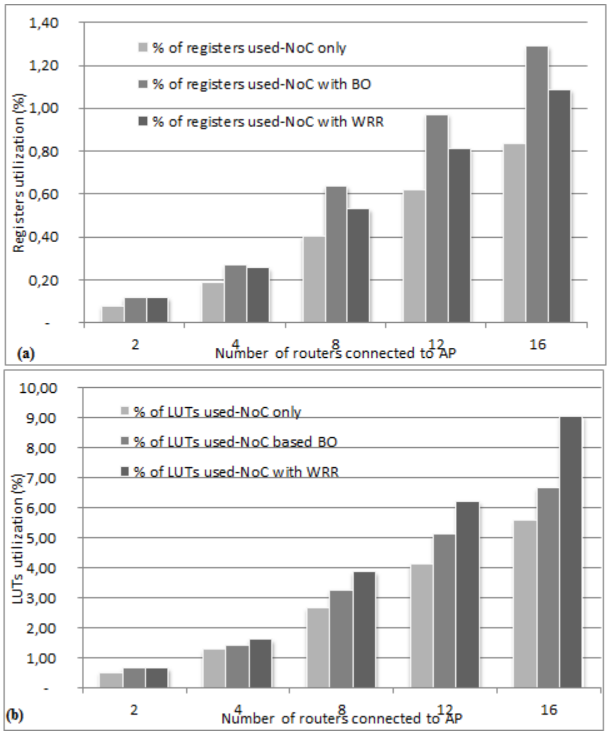


Fig. 8. Percentage of resources on virtex-6 FPGA: (a) registers, (b) LUTs

B. Resource results

1) Synthesis results: Figure 8 represents the percentage of resources used on the Virtex-6 FPGA. The number of resources for all three architectures is small compared to the number of resources available on the FPGA. The advantage becomes obvious with the increase of the sub-Network size. The percentage of registers used increases from approx.0.12% to 1.29% and 0.12% to 1.09% for a 1x2 to a 1x16 sub-NoC respectively with the BO and WRR. The percentage of LUTs increases from approx. 0.68% to 6.67% and 0.68% to 9.07% with the Backoff and the weighted Round-Robin. These extra resources are not significant compared to the total number of the available FPGA resources. The number of resources linearly depends on the size of the NoC whatever the architecture used.

The number of resources added to the NoC when integrating the collision avoidance architecture is depicted in Figure 9. The percentage of added registers to the NoC is from 43% to 58% for the Backoff and is from 30% to 48% for the Weighted Round-Robin (Figure 9.a). The percentage of added registers remains stable for a number of routers higher than 4 and for the WRR architecture. The percentage of added LUTs is 10% and 26% for the 1x4 with the BO and WRR (Figure 9.b). It corresponds to 245 more registers and 185 more LUTs. It uses 54%, 30% of registers and 19%, 62% of LUTs for the size

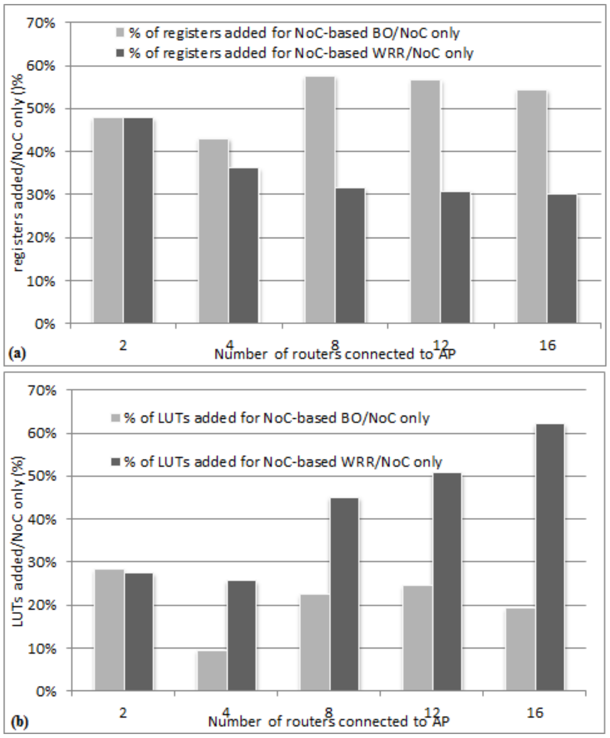


Fig. 9. Added resources compared to NoC only: (a) registers, (b)LUTs

1x16 with the BO and the WRR. It corresponds to 1373 and 5239 more LUTs and 1629 and 762 more registers with the BO and the WRR.

The percentage of LUTs is higher for the WRR algorithm compared to BO. The percentage of registers for the BO compared to the WRR. The percentage of the total added resources is the same for both (the sum of added registers and LUTs is globally the same). Both architectures require more resources than the original NoC itself.

2) NoC partitionning: One access-point is used in the experiments. All routers are connected to this AP. The size of the NoC partitioned can be higher with the use of the AP compared to a traditional link to link connection. Using a NoC partitioned into sub-NoC (without AP), the maximal number of routers that can be connected to external pins is 9 (for 32-bit flits) and 4 (for 64-bit flits) for the ML605 platform. Table I presents the number of routers in the basic partitioning and the number of AP when using a collision avoidance architecture. This is therefore not possible to deploy a large size of NoC without sharing external links. The use of the AP can increase the size of the NoC as several routers can be connected to the AP (in theory the number is unlimited). The limitation is due to the number of FPGA resources and the FPGA communication network. Moreover the number of AP depends on the number of pins available in the FPGA and the size of the external bus connected to the AP. It is therefore possible to increase the number of AP. For the size of flits of 64-bits the maximal number of AP is 4 and 9 for the 32-bit size. The collision management architecture can reduce the number of pins and can increase the number of routers. The use of the AP offers a substantial improvement both compared limitation made by the number of I/O.

TABLE I
INTER-FPGA CONNECTIONS ACCORDING TO THE PARTITIONNING

Size of flits	32-bits	64-bits
Maximum number of AP	9	4
Routers surrounding FPGA before adding AP	8	4

C. Timing results

This section analyses the timing performances obtained with both proposed collision management algorithms and compare these results. The timing performances with the NoC only are not considered as it cannot manage collisions. The objectives are to analyze the impact of the two access-protocols, with several load injected and with several lengths and number of packet when several nodes send packets to AP at the same time. Several experiments with different traffics are done using synthetic traffics and real application traffics. In the synthetic traffic analysis, the evaluation of the communication between TGs and TRs for each link is done using synthetic scenarios. The test scenario is the one to one scenario. In this scenario, each router (0, j) in first FPGA communicates only with the router (1, j) in the second FPGA. Experiments are done on homogeneous and heterogeneous packets. Packets are homogenous if all the packets have the same size and the same T$_{idle}$ between two packets for the same destination (and the same data injection rate). Traffics are heterogeneous when the size of packet is different (with fixed or varying T$_{idle}$). The detailed experiments in a collision context are given as follows:

- Homogenous traffic: all TGs send the same packet (same size and same number) with the same data injection rate (T$_{idle}$ to).
- Heterogeneous traffic: the TGs send the same size of packets, the value of T$_{idle}$ changes and the data injection rate to.
- Heterogeneous traffic: T$_{idle}$ is fixed for all packets and the length of packets changes for each TG (the data injection rate is different for each TG).
- Heterogeneous traffic based on the real application: the traffic parameters are based on the real-application for multispectral imaging. In this application, the number and size of packets and the T$_{idle}$ vary for each TG.

1) Homogenous traffic: The first scenario using homogenous traffics is done on the regular 1x8 sub-network on each FPGA. The timing evaluation is based on sending 50 packets, each packet containing 100 flits. The T$_{idle}$ and the data injection rate are the same for all routers. Figure 10 and Figure 11 present the obtained results for the minimal and maximal latencies with different data injection rates. The data injection rate varies from 10% to 80%. The time spent by routers to transmit one flit is two clock cycles.

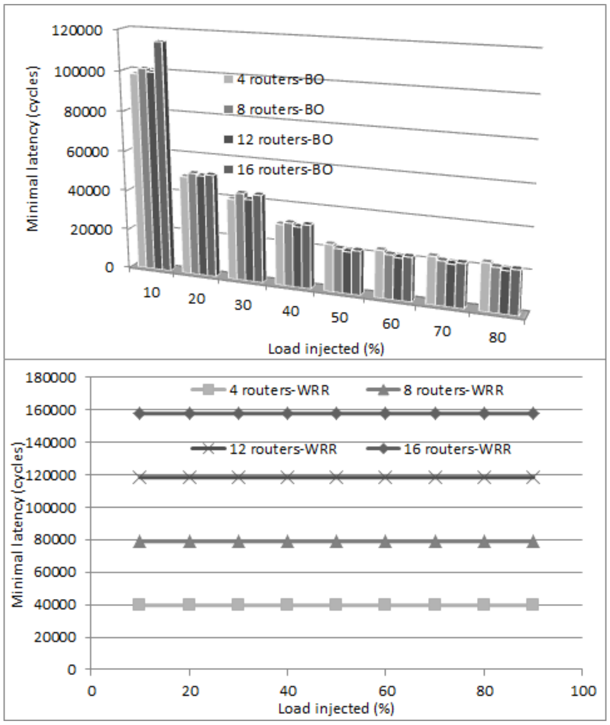


Fig. 10. Minimal latency for : (a) Backoff algorithm (b) Weighted Round-Robin

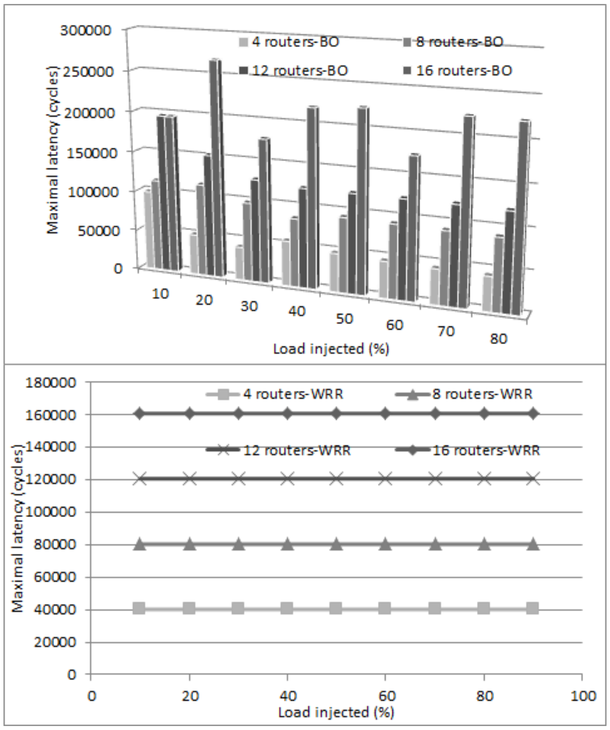


Fig. 11. Maximal latency for : (a) Backoff algorithm (b) Weighted Round-Robin

The minimal and maximal latency strongly depend on the packet scheduling on the access-protocol, the number of routers connected to the AP and the data injection rate. It is observed that the minimal latency for the BO algorithm converges and becomes constant from 50% data injection rate. The minimal latency is the same whatever the number of the routers connected to the AP for the Backoff architecture. The minimal latency changes according to the number of routers for the WRR architecture. The load injection does not have any impact on the minimal latency. The WRR architecture provides better minimal latencies for a data injection rate below 50%. The BO provides better minimal latencies above 50%. Figure 11 depicts the maximal latencies using the Backoff (a) and the Reighted Round-Robin (b) for different number of routers and for different data injection rate. There is a degradation of the maximal latency for BO, the maximal latencies depend on the number of routers and on the load injected. For most scenarios, the WRR provides a better maximal latency. The maximal latency is identical with 4 routers and for a data injection rate higher than 10%. The maximal latency is also identical for 8 or 12 routers with the 40% data injection rate and for 16 routers with the 30% data injection rate.

The maximal latency for BO for some traffics is higher than the WRR solution because of the value of Backoff-Timer generated by the random generator. For the size of the sub-NoC, the minimal latency for the Backoff does not exceed 120.10^3 cycles, while the maximal latency varies between 160.10^3 and 250.10^3 cycles. This variation does not appear with WRR as the minimal and maximal latencies only depends on the length of packet and the number of routers connected to the AP. The idle time between two packets is hidden by the request of the following router. The minimal and maximal latency for the weighted round-robin remains constant until the end of transmission.

2) Heterogenous traffics: Two scenarios using heterogeneous traffics are presented. Both examples exhibit most intensive communications with heterogeneous packets commonly used in most systems. Destinations for packets are chosen to have a point to point connection (similar than the previous experiment), different T_{idle} and different length with bandwidth requirements are randomly selected. In the first example, T_{idle} is set to 100 cycles and the size of packets is randomly chosen for all transmitting routers (the size of packet also varies). In the second example, all routers send packets with the same size and number but with a varying T_{idle}. Three scenarios are used for both examples. For all experiments, the average, minimal and maximal latency are compared. In the first experiment, the position of the load injected changes from router to another one. For each example we apply the time allocation algorithm. Table II and Table III describes both examples with the different values of T_{idle}, the size of packet.

Figure 12 gives the minimal, maximal and average latencies for the three scenarios using the BO and WRR. The T_{idle} is 100 clock cycles and the size of packets changes as shown in Table II. The average, minimal and maximal latencies are similar with the use of the WRR whatever the scenario. The average, minimal and maximal latencies changes according to the scenario used. The average and minimal latencies are

TABLE II
VARIED SIZE OF PACKET

Path	size of packet			T_{idle}
	scenario0	scenario1	scenario2	
R0 → R1	10	200	200	100
R2 → R3	50	100	50	100
R4 → R5	100	50	100	100
R6 → R7	200	10	10	100
R8 → R9	10	200	200	100
R10 → R11	50	100	50	100
R12 → R13	100	50	100	100
R14 → R15	200	10	10	100

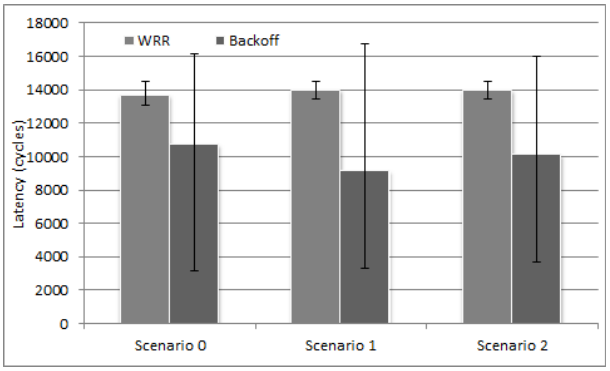


Fig. 12. Different scenario based on the variation of the size of packet (with the T_{idle} constant)

TABLE III
VARIED DATA INJECTION RATE T_{idle}

Path	packet lenght	T_{idle}		
		scenario0	scenario1	scenario2
R0 → R1	20	10	360	160
R2 → R3	20	15	160	95
R4 → R5	20	25	95	15
R6 → R7	20	40	60	360
R8 → R9	20	60	40	25
R10 → R11	20	95	25	60
R12 → R13	20	160	15	10
R14 → R15	20	360	10	40

better with the BO and the maximal latency is better with the WRR. The first packet arrives faster and the last packet arrives later with the BO. In the average, packets arrive faster with the BO compared to the WRR. The packet distribution is more uniform with the WRR. Timing performances can be optimized with the BO architecture for heterogeneous traffics. Figure 13 gives some experiments with the same size of packets (with different value of T_{idle} for each TGs). The T_{idle} are given in Table III. The average timing performance is close for both solutions (around 3000 clock cycles). The minimal latency is better with BO and the maximal latency is better with WRR.

For both experiments, the maximal latency is worse for

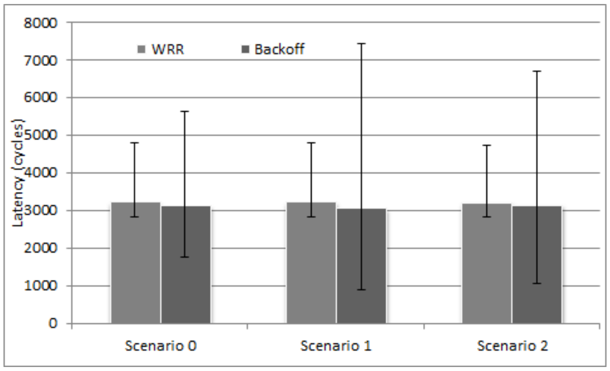


Fig. 13. Different scenario with the variation of data injection rate T_{idle}

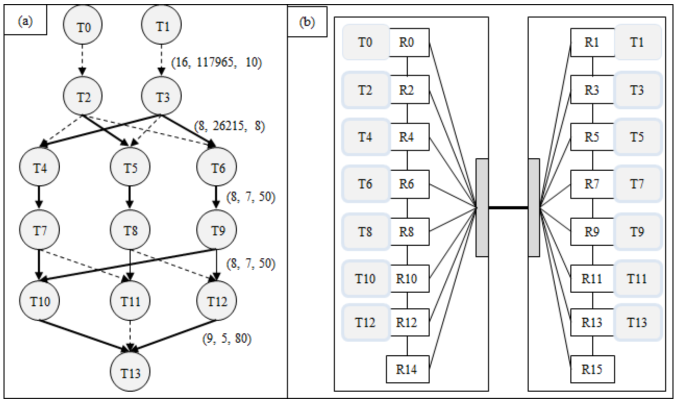


Fig. 14. Task graph of the multispectral imaging application and the task mapping on the NoC

the BO because of the random value used to postpone the transmission. The BO is the most adapted solution when the traffic is fully heterogeneous (different size of packets with different injection rate). Both solutions can be adapted for the same size of packets with different data injection rates.

3) *Results under applications specific traffic:* In the previous scenarios, experiments are done with intensive one to one connections between FPGA with homogeneous and heterogeneous traffics. The evaluation of the proposed architecture is done under a real benchmark application, a multispectral image application developed for art authentication [17]. This application contains 14 tasks with communications given in the task graph depicted in Figure 14.a. The communication requirements of each communication are given by a triplet (x,y,z), indicating the size of packets, the number of packets and the T_{idle}. For example, the message from T3 to T6 is (8, 26215, 8) indicating that 26215 packets with the size 8 is sent and the idle time between packet is 8 clock cycles. The task mapping for each task for the application onto two 1x8sub-NoCs is described in Figure 14-b.

For the application, packets have to be sent from both sides. Two collision avoidance architectures are integrated between both sub-NoC to transmit data for both sides. Figure 15 shows the communication latencies between routers (the routers depicted in the figure are the receivers). The communications

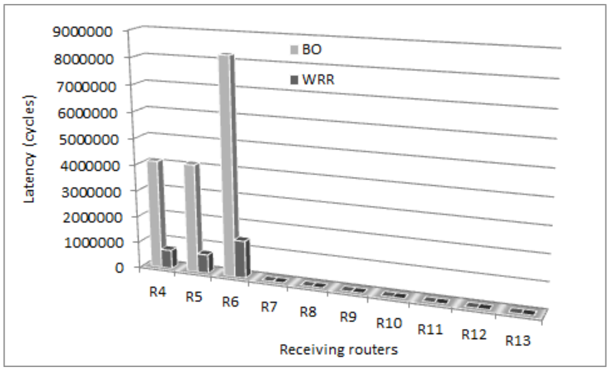


Fig. 15. The communication latency of the imaging multispectral application between APs

between R3-R4 (receiving router R4 in the figure), R2-R5 and R3-R6 are faster with the WRR than BO. The differences between the BO and WRR for the communications give above are high as it represents 50% of added clock cycles. For some communication (receivers R9, R11 and R13), the latency is better with the BO than the WRR. In this case, the WRR architecture is most efficient than the BO architecture (not for all inter-FPGA communications).

V. Conclusion

In this paper a novel collision management architecture dedicated to manage collisions between sub-NoC implemented on different FPGA boards is proposed. The proposed collision management consists of two blocs: the Access Point to send packets between inter-FPGA links, and a Access Protocol to manage the collision and schedule the access to the Access Point. This block integrates two algorithms, the Backoff algorithm that distributes randomly the access and the weighted round-robin algorithms. The use of the AP can decrease the number of external IO (and links) and it is possible to deploy a large a size of NoC on multi-FPGA.

The experiments based on synthetic traffics and on a real application evaluated on an existing NoC demonstrate that both congestion algorithms are efficient. The choice of the algorithm depends on the traffics. Both can optimize the schedule on the Access Point with a number of required added resources on the FPGA.

In this case, the WRR architecture is most efficient than the BO architecture (not for all inter-FPGA communications). The values of the idle time should be explored to fully exploit the collision management algorithm (the previous work selected the idle times using the background of the user).

Acknowledgment

The funding for this project was provided by a grant from la Region Rhone-Alpes.

References

[1] International Technology Roadmap for Semiconductors (ITRS) 2009. [Online]. Available: www.itrs.net/Links/2009ITRS/Home2009.html.

[2] [Online].Available:http://www.intel.com/content/www/us/en/architecture-and-technology/many -integrated-core/intel-many-integrated-core-architecture.html.

[3] Nvidia Fermi: Next Generation Cuda Architecture. [Online]. Available: http://www.nvidia.com/object/fermi architecture.html

[4] A. Lankes, T. Wild, A. Herkersdorf, "Hierarchical NoCs foroptimized Access to Shared Memory and IO Resources". DSD2009, 07 December 2009, pp. 255-262.

[5] B. M. Al-Hashimi, "System-on-Chip: Next Generation Electronics". Circuits, Devices and Systems, 2006.

[6] V. Fresse, J. Tan, F. Rousseau, "From Mono-FPGA to Multi-FPGA Emulation Platform for NoC Rerformance Evaluations". International Conference on parallel Computing,-ParaFPGA,October 2011.

[7] K. M. Abdellah-Medjadji, B. Senouci, F. Petrot, "Large Scale On-Chip Networks : An Accurate Multi-FPGA Emulation Platform". 11th Euromicro Conference on Digital System Design Architectures, Methods and Tools, pp. 3-9, 2008.

[8] L. Xinyu, O. Hammami, "Multi-FPGA emulation of a 48-cores multi-processor with NOC". 3rd international Design and test workshop, pp 205, December, 2008, Tunisia.

[9] C. Seiculescu, S.Murali, L. Benini, "SunFloor 3D: A Tool for Networks on Chip Topology Synthesis for 3-D Systems on Chips". IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol.29, December 2010.

[10] B.S. Feero, P.P. Pande, "Networks-on-Chip in a Three-Dimensional Environment: A Performance Evaluation". IEEE Transactions on Computers, (Vol: 58, Issue:1). January. 2009.

[11] M. Stepniewska, A. Luczak, J. Siast, "Network-on-Multi-Chip (NoMC) for multi-FPGA multimedia systems". 13thIEEE Euromicro Conference onDigital System Design, Methods and tools (DSD). September 2010.

[12] Y.Liua, P.Liua, Y.Jiangb, M.Yangb, K.Wua, W.Wanga and Q.Yaoa, "Building a Multi-FPGA-based Emulation Framework to Support NoC Design and Verification". International Journals of Electronics 2010 (jul) 2010 (Special Issue on Evolutionary synthesis of Network-on-Chip-Based Systems).

[13] D. Stiliadis, A. Varma, "Latency-Rate Servers: A general Model for Analysis if Traffic Scheduling Algorithms, in Proc.". IEEE INFOCOM'96 111-119 (1996).

[14] R. Jurdak, al, "A Survery, Classification and comparative Analysis of Medium Access Control Protocols For Ad hoc Networks". IEEE Communication Survery, First Quarter 2004.

[15] C. Zeferino et al, "A Study on Communication Issues for Systems-on-Chip". in Symposium on Integrated Circuits and Systems Design, Los Alamitos, CA, USA, 2002, pp. 121126.

[16] F. N. Moraes, A. Mello. Calazans, "HERMES: an Infrastructure for Low Area Overhead Packet-switching Networks on Chip". Integration, the VLSI Journal, vol. 38, no. 1. Oct. 2004. Page(s): 6993.

[17] Linlin Zhang,Virginie Fresse, Mohammed A. S. Khalid, Dominique Houzet, Majid Ahmadi, "Evaluation of NoC Dedicated to Multispectral Image Data Communication". International Symposium on Signals, Circuits and Systems, 2009. ISSCS 2009.