5th International Conference on Control Engineering&Information Technology (CEIT-2017)
Proceeding of Engineering and Technology –PET
Vol.32 pp. 97-102

A Mixed-Priority Scheduler for Network-on-Chip Wormhole Routers

Ridha SALEM, Yahia SALAH and Mohamed ATRI
Laboratory of Electronics and Microelectronics
Faculty of Sciences, University of Monastir
Monastir, 5000, Tunisia
Email: Ridha.Salem@fsm.rnu.tn

***Abstract*—In this paper, we propose a mixed-priority scheduler using static and dynamic scheduling algorithms to resolve conflicts between traffic connections in a router for on-chip packet-switched networks. For the scheduler implementation, we consider pseudo adaptive routing and input virtual channels queuing. The synthesis of the scheduler is performed with the Xilinx ISE 14.1 tool targeting Xilinx Virtex-6 technology with XC6VLX760 FPGA device. Simulation and implementation results show that our scheduler design enables a high operating frequency with low request processing time and minimal hardware constraints.**

Keywords— Packet-Switched Networks-on-Chip; Router; Mixed-Priority scheduling; Simulation; Hardware implementation.

I. INTRODUCTION

Networks-on-Chip (NoCs), have emerged as a new design paradigm in the last decade to overcome the limitation of traditionally on-chip communication infrastructure, and are increasingly important in today's System-on-Chip (SoC) designs. This architecture includes physical interfaces and communication mechanisms, which allow the communication between on-chip components. It is able to accommodate a large number of intellectual property (IP) cores, providing separation between the computation and communication, and facilitate a truly modular and scalable design approach for SoC [1].

Since the introduction of the NoC paradigm, various approaches have been proposed to address the complex chip design and timing closure problems of future generations of chip multiprocessors. Several of them use the circuit-switching scheme [2-4], and the vast majority of NoC proposals are based on packet-switched networks [2, 5, 6] to achieve these objectives. Most on-chip packet-switched network implementations have used a regular mesh topology, static routing algorithm and wormhole switching strategy to cope with the question of system flexibility and to reduce area and energy overheads [2, 7, 8]. Nevertheless, in many situations, the timing behavior of the communication part in SoC is unpredictable because there is no mechanism to guarantee the end-to-end delivery time of data. Unpredictable delays for an application with hard time constraints are unacceptable to building reliable embedded systems like mobile-phone, digital-camera, etc. Providing quality-of-service (QoS) – such as guaranteed throughput and end-to-end transfer delivery of messages – is essential for the efficient construction of these systems.

A large range of methodologies and algorithms have been proposed by different groups of NoC research (both by academic and industrial research groups) for satisfying performance constraints and solving the SoCs design problems. The significant challenge is the design of a suitable on-chip interconnection architecture to provide adequate QoS ensuring certain bandwidth and latency bounds for inter-module communication, but at a minimal power and silicon area costs. For those reasons, we try in this paper to propose hardware architecture of a mixed-priority (MP) scheduler for wormhole routers to enable efficient and predictable global on-chip communication. The organization of this paper is as follow: section II highlights some previously published on-chip communication proposals that consider the effect of packet schedule on the QoS performance metrics and cost factors. Section III presents the application and NoC structure models. Section IV describes the proposed MP scheduler that handles best-effort (BE) and guaranteed-services (GS) traffic flows. Experimental results are presented in Section V. Finally, section VI summarizes conclusions.

II. PREVIOUS WORKS

Router component represents the key factor in determining NoC performance. Thus, the router needs to satisfy the different QoS requirements of various traffic classes on the same chip. As packet arrival is unpredictable, so contention cannot be avoided in a router-based network with wormhole switching. It is resolved dynamically by scheduling schemes in which data items are sent in turn. The way to schedule the traffic flows between the heterogeneous resources on a single chip becomes an extremely challenging task. There have been several researches on traffic flow scheduling onto NoC architectures. QNoC [5], proposed by Bolotin et al, is a system based on asynchronous routers with dynamic virtual channel (VC) allocation. It groups all traffic in four service levels based on the on-chip communication requirements and assigns different priorities. The priority based round-robin (RR) scheduling criterion is employed for transmission of flits. L.-F. Leung and C.-Y. Tsui [9] propose an optimal link scheduling and a shared-buffer router architecture which strive to

ISSN 2356-5608

minimize the overall execution time and improve network resources utilization of the applications that have hard real-time requirement. S. Stuijk et al. [10] present three scheduling strategies that minimize resource usage by exploiting all scheduling freedom offered by NoCs. The first strategy typically gives a solution quickly for a set of problems of a conflict. The second strategy adds backtracking to the greedy approach. And finally, the third strategy tries to avoid conflicts by estimating a priori the usage of all links. D. Bui and E.A. Lee [11] propose an earliest deadline first (EDF) work-conserving packet scheduling discipline for hard real-time NoCs, which provides guaranteed service without sacrificing high consistent average performance. They also derive sufficient buffer sizes for their scheduling discipline.

Almost all previous work focuses on maximizing the performance through the scheduling process, but implementation complexity of algorithms significantly affects the cost of the SoC. Only a few of these researches consider priority scheduling solution with minimal communication overhead [12, 13]. In [12], J. Hu and R. Marculescu present an energy-aware scheduling algorithm which statically schedules application-specific communication transactions and computation tasks onto heterogeneous NoC architectures by considering the network congestion, as well as the IP's heterogeneity. Their proposed algorithm allows minimizing the overall energy consumption of the system while guaranteeing the real-time deadlines imposed on tasks. S. Ould-Cheikh-El-Mehdi [13] propose a novel hybrid scheduling approach, which combines the optimality of EDF algorithm and the simplicity of first-in first-out (FIFO) method to meet the stringent time constraints of real time flows and to allow reducing EDF's implementation complexity. We note that mapping and scheduling problems can be considered jointly to greatly impact both performance and energy consumption of the NoC. The research work reported by P-F. Yang and Q. Wang [14] is one of the current approaches in this area.

Our scheduler architecture presents a mixed approach using the static and dynamic scheduling algorithms. Our model will be clearly developed in the next sections. Unlike most traffic flow scheduling scheme onto NoC architectures, our proposed method is optimized to achieve the best tradeoffs between complexity of dynamic scheduling and hardware constraints. The solution is appropriate for wormhole-switched NoCs using static or dynamic routing and input virtual channels queuing.

III. PLATFORM CHARACTERISATION

A. NoC Structure Model

The architecture platform consists of a set of routers which are connected to each other in an arbitrary topology. Regular topology is a popular NoC architecture due to its predictability and ease of design [15]. In our modeling, we used the mesh-based NoC architecture showed in Fig. 1; each tile connected via its network interface (NI) and the data communications between the IPs of different tiles are packetized and transferred through the links and the routers. A NI can connect more than one IP core to the switched network, and support two types of communication services to IPs: best-effort (BE: in-order packet delivery QoS) and guaranteed-services (GS: guaranteed timing delivery QoS).

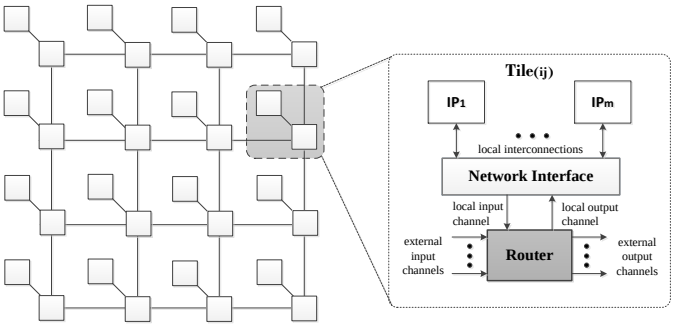

Fig. 1. A Generic NoC Architecture Platform

Routers are the main elements in NoCs: they are responsible for forwarding and routing packets throughout the network from source to destination. As we can see in Fig. 2, the generic wormhole router architecture includes *p* input port controllers (IPC) and *p* output port controllers (OPC) to communicate with its neighbors and with the IP cores of the system. These input-output ports are connected through a *p*×*p* crossbar switch (CS). An IPC block holds an input process module (IPM) and a variable number of virtual channels, a set of VCs for GS flits and the others for BE flits. The IPM unit determines which output port channel is selected for a packet arriving on a given input port channel. It taking into account the information in the packet header and the status of the neighbour's channel of OPC modules. For the sake of simplicity, deadlock and livelock freedom, the pseudo adaptive XY routing scheme [7] is used to direct the packets across the chip. In addition, wormhole packet switching method [8] is employed to increase the on-chip router performance. Hence, data that needs to be transmitted between source and destination cores is partitioned into fixed length packets which are in turn broken down into control flow units (flits). The router then schedules the transmission for each flit on the appropriate OPC module. A scheduler of the router control logic receives requests from the IPM header decoders of the other unit routers and performs arbitration in to enable fair access to the common crossbar fabric for all incoming packets.

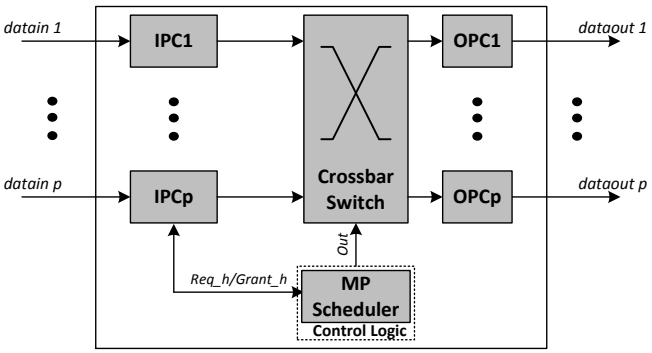

Fig. 2. Generic on-Chip Router Architecture

To reduce area overhead in our implementation, each OPC module contains a credit counter and one lane of one flit size per physical output channel. During router operation, the lane can be in one of the two following states: free – the lane is not used at the moment; or busy – there is a flit of a packet using the lane. The scheduler and reconfiguration logic (RL) modules are used to solve conflicts at the physical output channels.

B. Application Model

We model an application by its tasks flow graph (TFG). A TFG = (T, S), is a directed graph where each vertex in T represents a task T_i and each directed arc (T_i, T_j) models a stream of messages produced by T_i and consumed by T_j. A stream of messages from task T_i to T_j is a sequence of messages (M_1, M_2, … , M_n). The parameters of a message M_i are:

- type (*TT*) : GS or BE traffic,
- identifier (*id*),
- source (*Src*): the source address,
- destination (*Dst*): the destination address,
- traffic (*t*): size in flits number,
- absolute delivery deadline (*d*): the absolute latest time for delivering the message to the destination.

In this work a task corresponds to a transfer of one message through the NoC. For simplicity, we assume that there is only one packet with in one message, so the message and packet mention the same thing in this paper. A flow is a set of packets which possess the same source and target cores, with a packet is a set of flits.

IV. MP SCHEDULER DESIGN

The function of the MP scheduler is to arbitrate between requests from input blocks to output blocks of the router. The scheduling scheme is based on three strategies, which are: Round-Robin (RR), High Priority First (HPF), and Earliest Deadline First (EDF). A scheduling decision is arrived at in three steps: (a) Requests are sent from the input blocks within a router to the scheduler. (b) For each shared output port, the scheduler can choose one of three actions using HPF, RR or EDF scheduling mechanism and selects entry with highest priority. (c) Grant signals are generated by the scheduler and these signals represent the final scheduling decision. This decision is sent back to the input blocks as well as the crossbar switch to enable transfer of packets from the IPC blocks to the OPC blocks.

Fig. 3 depicts the MP scheduler model in a NoC router. It may receive up to |*p*| requests where *p* is the set of IPC units within a router. Each request specifies: the type of the packet GS or BE, the number of data flits in the packet, the deadline constraint if it is a GS, and the destination OPC as computed by the pseudo adaptive XY routing algorithm. Here, a credit based flow-control mechanism is employed to prevent data being sent to an IPC buffer. Each VC has a separate credit-based counter which is decremented when a request is forwarded to the MP scheduler. If the counter is zero the request is blocked until new credits are received from the receiving switch. The numbers of available flit cycles per each traffic class in the VC buffers of the next IPCs are stored in a next-buffer-state table of the reconfiguration logic.

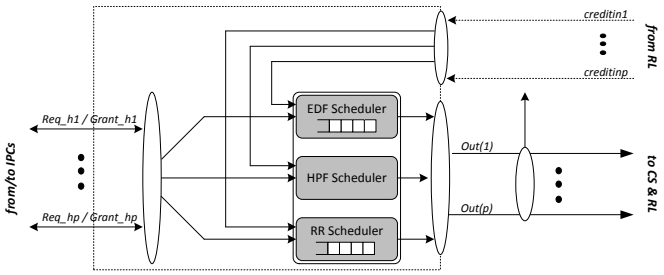

Fig. 3. The scheduler model for mixed-priority scheduling of GS/BE packets in a NoC router

The scheduler first detects the incoming requests of the IPC blocks and identifies their traffic types. If a traffic is of type GS, the scheduler inserts its request in the EDF queue. Otherwise, request is inserted in the RR queue. In the case of some input ports request the same output port simultaneously, the scheduler behaves as follows. GS requests are handled first, and the EDF policy is used to grant one of these requests based on their relative deadlines. For each output port, the scheduler compares the relative deadlines and selects the greatest entry as the highest priority. If the requests are of types BE and GS then the HPF policy is used. If all requests are of type BE then the RR mechanism is used. Note that if a router output port is free and is not shared by multiple input ports, the scheduler will automatically respond to the request with a grant.

In our scheduling scheme, a relative deadline is defined by the composition $edf_{kl}=T^R_{kl}/D^R_{kl}$, where T^R_{kl} is denotes the number of flits remaining to be routed from IPC_k to OPC_l, and D^R_{kl} its deadline. One necessary condition to satisfy the deadline constraint is to have $T_{ij}/D_{ij} \leq 1$, for all *i*, *j*. T_{ij} corresponds to the number of GS flits to be sent from network interface NI_i to NI_j, and D_{ij} is their arrival deadline counted in number of cycles.

The HPF technique assigns static priorities to packets. Priorities are assigned according to the traffic type (BE or GS) of each packet. The finite state machine (FSM) of the HPF scheduler is given by Fig. 4 where *TT* represents the type of the packet and T_{S00} and T_{S01} indicate, respectively, the number of GS flits and the number of BE flits to be routed from IPC_k to OPC_l of the router.

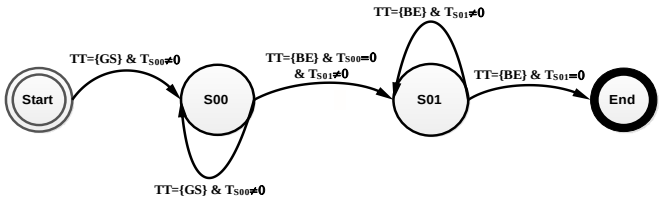

Fig. 4. Finite State Machine of the HPF Scheduler

The priority based RR scheduling criterion is employed for transmission of BE flits. The scheduler sequentially passes control from one flit to another; empty input queues are skipped. The FSM of the RR scheduler is presented in Fig. 5 where T_{S00}, T_{S01}, T_{S10} and T_{S11} indicate the numbers of BE flits to be routed from IPC_k to OPC_l of the router. These numbers are associated, respectively, to the BE requests of the four states *S00*, *S01*, *S10* and *S11*. The FSM 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.

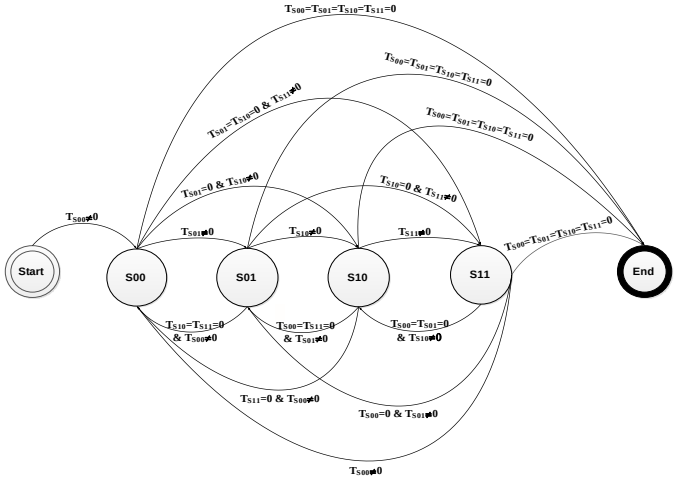

Fig. 5. Finite State Machine of the RR Scheduler

The EDF technique uses the deadline of a GS request as its priority. The request with the earliest deadline has the highest priority, while the request with the latest deadline has the lowest priority. In our scheduler implementation, the EDF queue is sorted by the relative deadlines of requests. The EDF scheduler selects the greatest relative deadline as the highest priority. This allows a suitable strategy to handle applications with strict QoS requirements such as end-to-end delay and throughput. Fig. 6 shows the FSM of the EDF scheduler where edf_{00}, edf_{01} and edf_{10} represent, respectively, the relative deadlines of GS requests of the three states *S00*, *S01* and *S10*.

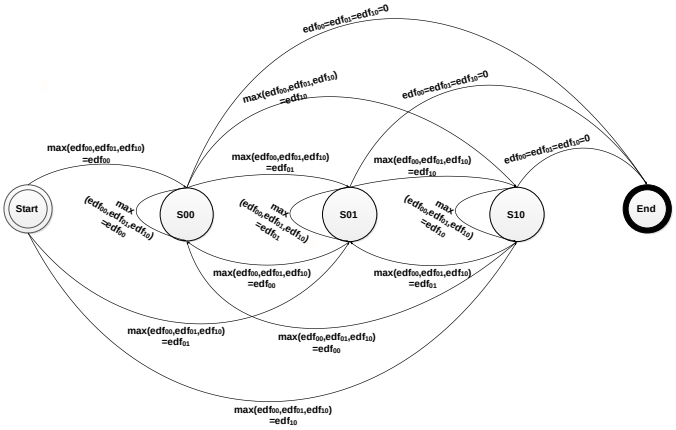

Fig. 6. Finite State Machine of the EDF Scheduler

V. Experimental Results

In this section, simulation and synthesis results are presented to demonstrate the performance of our MP scheduler for NoC wormhole routers.

A. Simulation Setup

Table 1 summarizes default parameters for the evaluation of the distributed MP scheduler module in a NoC-based SoC environment. We employ a traffic generator to replace the original IPs placed in their corresponding network interfaces for testing the efficiency of scheduler, router and NoC. The traffic generator module could generate a set of BE and GS messages with different constraints. We assume that a message coming through an input port is not returned to the output port of the same interface. We modeled the SoC modules in VHDL language, using the register transfer level (RTL) description, allowing accurate measurement of area, latency, critical path delay and maximum frequency values. We simulated the full system with Mentor Graphics ModelSim 10.0c.

TABLE I. Default Simulation Parameters

Simulation parameters	Specifications
Topology	*n*×*n* 2D-Mesh
Number of router ports	5
Routing	pseudo adaptive XY
Number of VCs per channel	2
Buffer depth	4 flits / VC
Flit size	32 bits
Packet size	variable
Message size	1 packet
Traffic pattern	uniform

B. Simulation Results

The developed MP scheduler can simultaneously handle up to five requests. It can also be configured in 1D and 2D NoC routers. A request processing requires a control time of one clock cycle after reset.

Figure 7 shows an example of an application that is modelled by a TFG. This TFG is mapped to IP cores of the SoC: a node represents a task, and an arc represents the amount of flits produced and to be transmitted from one task to another and their deadline if they are of type GS. As shown in Figure 8, we manually mapped the set of tasks to the 9 IP cores of a 3×3 2D-mesh NoC.

Figure 9 illustrates an example of BE/GS traffic connections (C_{kl}) in the router R_{11} of the 3×3 2D-mesh NoC topology of Figure 8. Five connections pass through this router. We notice that there is a contention problem on output OPC_1 for GS/GS traffic connections, and a contention on output OPC_2 for BE/BE traffic connections. The problem appears between circuits C_{01}, C_{21} and C_{41} respectively for messages M_{14}, M_{34} and M_{24}; C_{12} and C_{32} respectively for messages M_{46} and M_{53}. These connections need to be established with free contention. The table 2 gives the principal characteristics of messages sent by each task and an unambiguous representation of connections and contentions. We note that the parameter r of the table 2 represents the

arrival date of a GS or BE request in the router R_{11}. If the message is of type BE then D^{R}_{kl} will be set to symbol "X".

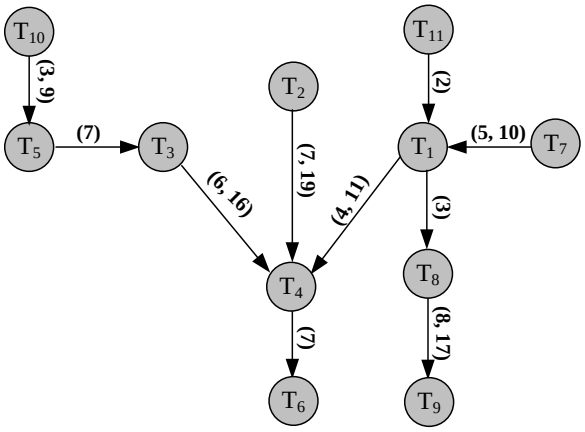

Fig. 7. An example of a TFG with 11 tasks

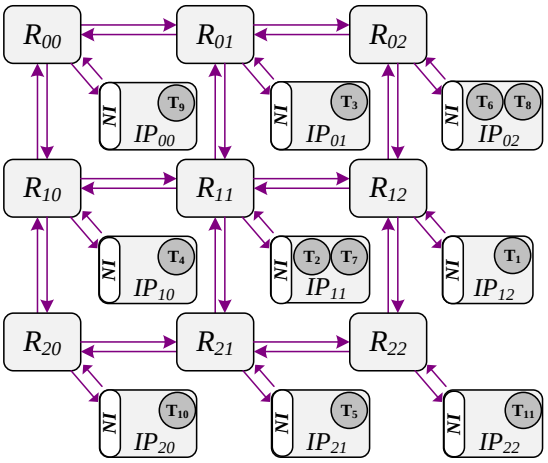

Fig. 8. TFG mapped to 9 IP cores of the 3×3 2D-Mesh NoC

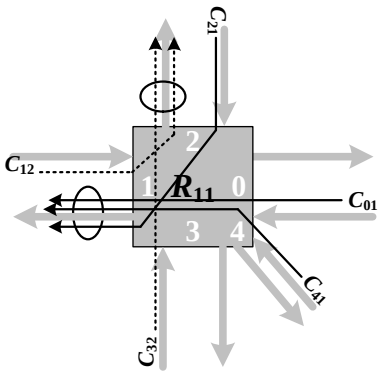

Fig. 9. Connections in a 2D-Mesh NoC router

TABLE II. THE PRINCIPAL CHARACTERISTICS OF MESSAGES

Problem variables									Shared Resources
M_{ij}	IP Src	T_i	IP Dst	T_j	C_{kl}	T^{R}_{kl} [flits]	D^{R}_{kl} [clock cycles]	r	
M_{14}	IP_{12}	T_1	IP_{10}	T_4	C_{01}	4	11	2	OPC_1
M_{34}	IP_{01}	T_3	IP_{10}	T_4	C_{21}	6	16	2	OPC_1
M_{24}	IP_{11}	T_2	IP_{10}	T_4	C_{41}	7	19	2	OPC_1
M_{46}	IP_{10}	T_4	IP_{02}	T_6	C_{12}	7	X	2	OPC_2
M_{53}	IP_{21}	T_5	IP_{01}	T_3	C_{32}	7	X	2	OPC_2

The simulation results of our scheduling approach are provided and illustrated by the figure 10. It shows the successive data output of connections given by Figure 9 along 34 clock cycles throughout the same router R_{11}. For the connections C_{01}, C_{21} and C_{41}, the EDF algorithm is applied to the GS messages M_{14}, M_{34} and M_{24}. It compares the relative deadlines edf_{kl}, selects the greatest entry as the highest priority then grants the corresponding IPC_k. For the connections C_{12} and C_{32}, the RR algorithm is used to the BE messages M_{46} and M_{53}. So, the IPC_1 and IPC_3 can be pictured as being placed in a ring, where the priority of each IPC decreases linearly from the IPC with highest priority.

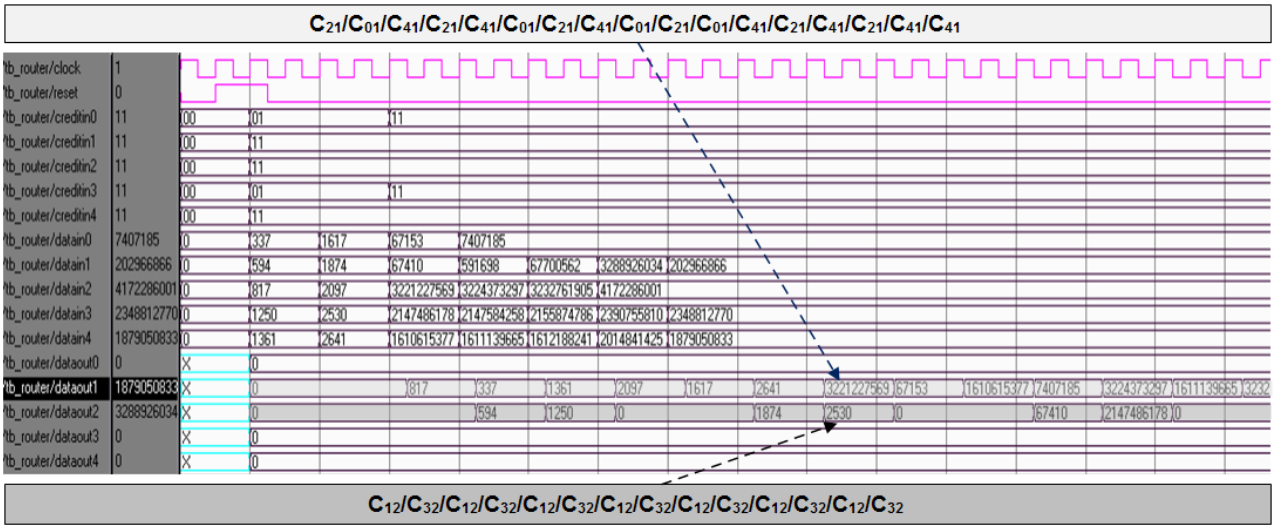

Fig. 10. Simulation results for the contention resolution between BE/BE and GS/GS traffics

The obtained results lead to the conclusion that our scheduling methodology is successfully implemented in the context of our on-chip communication model.

C. Hardware implementation

The mixed-priority scheduler prototype is implemented on Virtex-6 FPGA Xilinx XC6VLX760 target device using Xilinx 14.1 ISE Design Suite and evaluated in terms of area overhead, request processing time and operating frequency. When the input request is 12 bits, the scheduler occupies less than 1 % of the device area (128 out of 474240 slice LUTs used) and has a maximum frequency of 658 MHz. The table 3 shows the comparison between our MP scheduler design and two arbiters (Round-Robin and Matrix) published in [16].

TABLE III. THE COMPARISON BETWEEN OUR DESIGN AND OTHERS

Module	**Performance Analysis**				
	Area			***Delay (ns)***	***Maximum Frequency (MHz)***
	Slice LUTs	***Flip Flop***	***Slice Registers***		
Round-Robin Arbiter [16]	780	460	670	3.3	325
Matrix Arbiter [16]	750	475	520	3.1	335
Our MP Scheduler	128	40	78	1.518	658.610

Round-Robin arbiter and Matrix arbiter receive a number of 8 bits per input request and use the FPGA Xilinx Spartan 3 XC3S400 target device. Among the two designs, our scheduler can handle the requests with a critical path delay of 1.518 ns. Furthermore, the area of our scheduler is lower than the area of the two arbiters presented in [16]. The simplicity of our scheduler architecture offers a low scheduling delay and high speed requests processing with minimum area cost.

VI. CONCLUSION

In this paper, we presented a mixed-priority scheduler design for NoC wormhole routers. The proposed scheduler supports static and dynamic scheduling mechanisms that are used for best-effort and guaranteed service traffic flows to resolve conflicts between router input ports and impact the network performance. We have synthesized the RTL model of the scheduler architecture using FPGA Xilinx XC6VLX760. Under different traffic types, scheduler outputs are verified using simulation results. Compared with other NoC arbiters, our scheduler provides high performance for low-cost FPGA implementation.

REFERENCES

[1] T. Bjerregaard et al., "A scalable, timing-safe, network-on-chip architecture with an integrated clock distribution method," in Proc. DATE, 2007.

[2] E. Fernandez-Alonso, D. Castells-Rufas , J. Joven and J. Carrabina, "Survey of NoC and Programming Models Proposals for MPSoC," International Journal of Computer Science Issues (IJCSI), Vol. 9, Issue 2, No. 3, March 2012.

[3] C. Hilton and B. Nelson, "PNoC: a flexible circuit-switched NoC for FPGA-based systems," IEE Proc. Comput. Digit. Tech., Vol. 153, No. 3 (2006), pp. 181-188.

[4] G. Kees and H. Andreas, "The aethereal network on chip after ten years: goals, evolution, lessons, and future," in Proceedings of the 47th Design Automation Conference (DAC 2010), Anaheim, California, USA, pp. 306-311.

[5] E. Bolotin, I. Cidon, R. Ginosar and A. Kolodny, "QNoC: QoS architecture and design process for network on chip," Journal of Systems Architecture, Volume 50, Issue 2-3 (Special Issue on Network on Chip) (2004), pp. 105-128.

[6] M. Ali, M. Welzl and M. Zwicknagl, "Networks on Chips: Scalable Interconnects for Future Systems on Chips," in Proceedings of the 3rd IEEE International Conference on Circuits and Systems for Communications (ICCSC'06), Bucharest, Romania.

[7] M. Dehyadgari, M. Nickray, A. Afzali-kusha and Z. Navabi, "Evaluation of Pseudo Adaptive XY Routing Using an Object OrientedModel for NOC," The 17th International Conference on Microelectronics, 13–15 December 2005.

[8] L. Bin, Z. Li, I. Ravi, P. Li-Shiuan, L. Michael, E. Michael, E.L. Seung and N. Donald, "CoQoS: Coordinating QoS-aware shared resources in NoC-based SoCs," Journal of Parallel Distributed Computer, 2010, pp. 1-14.

[9] L. Lap-Fai and T. Chi-Ying, "Optimal link scheduling on improving best-effort and guaranteed services performance in network-on-chip systems," in Proceedings of the 43rd Design Automation Conference (DAC 2006), San Francisco, California, USA, pp. 833–838.

[10] S. Stuijk, T. Basten, M. Geilen, A.H. Ghamarian and B.D. Theelen, "Resource-efficient routing and scheduling of time-constrained streaming communication on networks-on-chip," Journal of Systems Architecture - Embedded Systems Design, vol. 54, issues 3-4, 2008, pp. 411–426.

[11] D. Bui, E. A. Lee, "Composable Flexible Real-time Packet Scheduling for Networks on-Chip, " Technical Report (EECS-2012-114), Electrical Engineering and Computer Sciences, University of California at Berkeley, May 16, 2012.

[12] J. Hu and R. Marculescu, "Communication and task scheduling of application-specific networks-on-chip," IEE Proc comput Digital Tech 152(5), 2005, pp. 643–651.

[13] S. Ould-Cheikh-El-Mehdi, W. Fawaz and K. Chen, "A novel approach for real time flows scheduling," in Proceedings of GLOBECOM '06, San Francisco, California, USA, pp. 1-5.

[14] P.-F. Yang and Q. Wang, "Effective Task Scheduling and IP Mapping Algorithm for Heterogeneous NoC-Based MPSoC," Mathematical Problems in Engineering, vol. 2014, Article ID 202748, 8 pages, 2014. doi:10.1155/2014/202748.

[15] F.G. Villamón, D. Ludovici, S. Medardoni, D. Bertozzi, L. Benini and G.N. Gaydadjiev, "Designing Regular Network-on-Chip Topologies under Technology, Architecture and Software Constraints," in Proceedings of CISIS'09, Fukuoka, Japan, pp. 68-687.

[16] T. Kavitha, S. Shiyamala and P. Nagarajan, "FPGA implementation of arbiters algorithm for Network-on-Chip", ARPN Journal of Engineering and Applied Sciences,Vol. 11, No. 19, pp. 11451 – 11456, Oct. 2016.