MOTIM – An Industrial Application Using NOCs

Fernando G. Moraes[1]   Everton A. Carara[1]   Daniel V. Pigatto[2]   Ney L. V. Calazans[1]
moraes@inf.pucrs.br   carara@inf.pucrs.br   daniel@datacom.ind.br   calazans@inf.pucrs.br

[1] PUCRS - FACIN - Av. Ipiranga 6681- Porto Alegre - 90619-900 - Brazil
[2] DATACOM TELEMÁTICA - Av. França 735 - Porto Alegre - 90230-220 - Brazil

ABSTRACT

High-speed networks used to interconnect computers advance at an extraordinary pace, driven by the evolution of several contributing technologies. Due to the ever-increasing complexity of designing parts and equipments for these networks, design complexity management makes scalability and reusability more important issues than performance, in most cases. This paper describes MOTIM, a scalable and reusable architecture enabling the implementation of Ethernet switches with low latency and high throughput. The architecture is built around a network-on-chip-based switch fabric, which guarantees scalability. The architecture has been validated by functional simulation and prototyped in FPGAs. The experimental results show that even under severe traffic conditions the architecture achieves packet transmission with low latencies.

Categories and Subject Descriptors

B.7.1 [**Integrated Circuits**]: Types and Design Styles – advanced technologies, algorithms implemented in hardware, VLSI (very large scale integration).

General Terms

Design, Experimentation, Performance, Verification.

Keywords

Networks on Chip, Ethernet switch, Prototyping, FPGA.

1. INTRODUCTION

High-speed networks used to interconnect computers advance at an extraordinary pace, driven by the evolution of several contributing technologies. The most prominent of these are optical interconnects using optical fiber [1], very large scale integration (VLSI) silicon chips [2] and standard communication protocols. Optical carrier levels defined for synchronous optical networks (SONET and SDH) are the basis for equipments in the market that drive 40 Gb/s (OC-768) lines and some expected to drive 160 Gb/s (OC-3072) lines in a near future. Current VLSI technologies (e.g. 90 nm and 65 nm) permit packing more than 10^9 transistors in a single silicon chip, enabling systems on a chip (SoCs). On the communication protocols side, Ethernet constitutes the most accepted standard for Local Area Computer Networks (LANs). Ethernet has also been used in larger networks (Wide Area Networks or WANs).

Modern high-speed network parts may contain several types of complex functional units, e.g. line interfaces, network processors, packet buffers, switch fabrics and system processor(s) [3]. Designing these parts implies combining VLSI chips with state of the art optical technologies and recently defined (thus unstable) high-speed communication protocols. This can indeed be daunting, if undertaken from scratch. Thus, in all technologies mentioned above, design complexity management makes scalability and reusability more important issues than performance, for most designs.

The main goal of this paper is to describe a scalable and reusable architecture useful for the construction of Ethernet switches, named MOTIM, the initial architecture of which is briefly presented in [4]. The main requirement of MOTIM is to allow achieving low latency and high throughput with a generic structure that can be easily scaled. In order to make the architecture scalable, its design is based on the use of a network on chip (NoC), a concept recently proposed for enhancing SoC interconnect design [5] [6] [7]. NoCs stand as a good compromise between silicon cost and performance scalability, easing to attain design requirements. Minkenberg et al. recently identified a set of trends arising in packet switch design and discussed their consequences [1]. The most important of these trends indicates that the aggregate throughput will grow by increasing the amount of ports in switches, rather than by increasing port speed. This imposes a demand for larger crossbars, a structure that do not scale well. Scalable NoCs are a feasible alternative to implement switches with fully interconnected ports.

The rest of this paper comprises seven sections. Section 2 discusses works related to switch design. Section 3 describes the proposal of MOTIM and describes the main characteristics of the architecture. Sections 4, 5 and 6 present the main modules of MOTIM. Validation and prototyping data about MOTIM is the subject of Section 7. The paper ends presenting a set of conclusions and directions for future work, in Section 8.

2. RELATED WORK

Many degrees of freedom can be exploited for designing high-speed network parts and equipment. Examples are the switch topology, scheduling, arbitration approaches, and buffering strategies. This Section gives an in-breadth sample of the state of the art in intra-chip and extra-chip switch and router design. The sample is by no means exhaustive.

Fayyazi and Kaeli [9] exploited two alternative regular topologies for Ethernet Layer-2 switching. The topologies are deemed to be scalable and programmable, and assume the use of multiple network processors and memory modules accessible by these. The ring topology switch (ES-Ring) displays low cost at moderate performance, while the mesh-like ES-Mesh displays high performance at higher costs. Looking for flexibility associated to high performance, the switch architecture HIPIQS [10] capitalizes on the use of a

modified version of dynamically allocated multiple queues (DAMQ) approach. The proponents of HIPIQS demonstrate the flexibility of their architecture along a range of message sizes and required throughput and latency figures, using simulation.

As a derived effect of the higher densities of VLSI chips, higher power densities in switch fabrics increasingly plague packet switching. Based on this fact, Dua et al. [11] propose the Power Managed Input Queued (PMIQ) switch architecture to achieve a better power-delay trade-off, based on the use of dynamic voltage scaling (DVS) and dynamic power management (DPM). The approach trades different speed modes against the switch loading conditions. The problem of scheduling cycles in the switch is formulated using dynamic programming and solved using heuristics.

García et al. [3], on the other hand, center on the problem of memory system organization for future (OC-3072) high-speed routers. The Authors in fact propose an enhancement of the techniques used to dimension the mix of SRAM and DRAM used to built packet buffers to support line rates on the order of 160Gb/s, reducing the amount of needed SRAM. Wolf and Turner [12] propose another design targeting high performance scalable routers, based on the use of a switch fabric associated to specialized port processors and processing engines. Each processing engine is in fact an on-chip multiprocessor with local memory. The work investigates the scaling of the proposed architecture, showing that in nine years its external memory bandwidth is expected to grow exponentially, the same happening to the number of instructions available to process a single byte of a packet in a high-speed (2.4Gb/s) link.

On the commercial equipment side, Ishihara et al., from Fujitsu [13] describe the implementation of a carrier-grade Ethernet switches designed to substitute leased lines in enterprise-wide networks at a lower cost and with larger bandwidth. The proposed switches add efficient redundancy and QoS features to traditional Layer-2 Ethernet switches leveraging these switches with regard to carrier equipment using e.g. SDH/SONET technologies.

3. MOTIM OVERVIEW

The MOTIM switch design derived initially from an industry-academy cooperation targeting the implementation of an Ethernet-SDH multiplexer. In this multiplexer, the switch works with 24 bidirectional Ethernet ports working at 100Mb/s and 1 to 4 high-speed (1 or 10Gb/s) ports. Figure 1 details the internal structure of the MOTIM instance used in the multiplexer. The current version contains only the Fast Ethernet ports. 1-GbE and 10-GbE ports are currently under implementation. Four module types compose the architecture: Ethernet MACs, Packet-Cell (PC) modules, Network Interfaces (NI), and the Network-on-Chip (NoC).

The MAC Ethernet module is an adaptation of an IP Core available at Opencores.org [8]. The PC module fragments Ethernet packets into fixed-size cells and reassembles cells into packets. The NI module provides an interface to the NoC and executes the routing of Ethernet packets, translating MAC destination addresses into NoC physical port destination addresses. The data sending part (NI→NoC) of the NI module stores cells, translates addresses and sends cells through the NoC. The NI module data receive part (NoC→NI) forward cells to the PC module and stores the relation between NoC router origin and MAC source addresses.
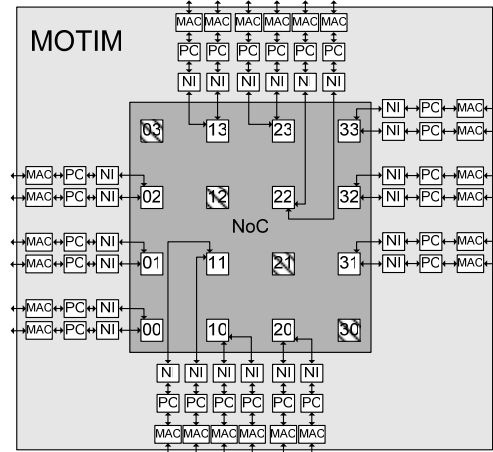

Figure 1 – MOTIM architecture instance. Modules NI, PC and MAC are instantiated 24 times and connect to the NoC module. The main diagonal routers of the NoC are reserved for special blocks.

The NoC performs all Ethernet data transport. Sixteen routers compose the NoC, interconnected as a 4x4-mesh topology. Each router contains two bidirectional ports to external modules, totalizing 32 external NoC ports. Of these, data connections use 24 ports for Ethernet packets. The remaining 8 ports are used for modules such as control processor, bulk memory and system supervision.

3.1 Cell Structure and Session Control

Fast Ethernet packets display two features leading to NoC sub utilization: low bandwidth with regard to the NoC and variable size. For instance, a NoC with 8-bit physical channel width operating at 100 MHz has 800 Mb/s bandwidth, 8 times bigger than Fast Ethernet packets rate. Low latency NoC packet transmission requires NoC resource reservation. Large packets transmitted as a unit would reserve NoC resources for long periods, causing NoC blocking, thus reducing NoC capabilities.

With the goal to optimize NoC utilization, Ethernet packets are partially buffered in the NI module. Once a pre-determined amount of data is available, this data transmitted in burst though the NoC. This exact amount of data is called a *cell*. Its size on the MOTIM instance is 128 bytes. Sending cells as bursts allows the NoC to operate at full speed. The constant size adds predictability to latency figures. The cell structure appears in Figure 2.
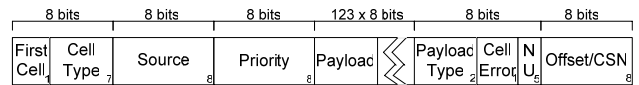

Figure 2 – Cell structure used in MOTIM instance.

Five of the cell bytes are used for control purposes: (*i*) the first bit signals if the cell is the first in an Ethernet packet or not and the next 7 bits define the cell type; (*ii*) the second byte indicates the cell origin router address; (*iii*) the third byte defines the packet priority; (*iv*) byte 126 uses 2 bits to indicate the payload type (first, last or middle cells) and 1 bit to signal errors; (*v*) byte 128 indicates either the cell sequence number (CSN) or, for the last cell in a packet, the number of significant bytes in the payload (offset). The 123 other bytes are payload.

Another relevant concept employed in MOTIM is session control [15]. Since Ethernet packets are fragmented into cells, it is necessary to reserve the target set NI-PC-MAC during transmission of all cells

in a packet. This reservation may lead to resource blocking, reducing system performance. The solution to avoid this blocking is to include *m* receive buffers in each PC module, for storing cells of up to *m* Ethernet packets received from distinct origins in parallel. The first cell from a packet reserves a target buffer, allocating it until its last cell is transmitted. Thus, up to *m* (*m*=4 in MOTIM) distinct sources may simultaneously send data for a same destination, without blocking.

4. NETWORK-ON-CHIP MODULE

The NoC module is based on HERMES [7], a parameterizable infrastructure to implement low area overhead wormhole packet switching NoCs with 2D mesh topology. The HERMES router employs input buffers, centralized control logic, an internal crossbar and five bi-directional ports. The Local port establishes a communication between the router and its local IP core. The other ports of the router connect to neighbor routers. A centralized round-robin arbitration grants access to incoming packets, and a deterministic XY routing algorithm selects the output port. With the goal to achieve low latency and high throughput requirements in the MOTIM NoC, the HERMES infrastructure has been enhanced in four ways: (*i*) support to circuit switching; (*ii*) physical channels duplication; (*iii*) support to session control; (*iv*) support to broadcast. Figure 3 illustrates the MOTIM NoC topology and depicts the NoC-NI interface. The upper right part contains signals used for cell sending, and the lower left part shows signals for cell reception.
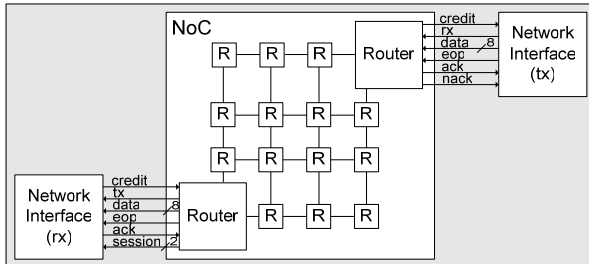

Figure 3 – NoC module topology and its interfaces to module NI.

Cell sending is synchronous. The NoC asserts the *credit* signal to indicate it can receive data. While *credit* is active, the NI signals data presence using *rx*, sending one byte per clock cycle. Cell receiving is similar, using signals *credit*, *tx* and *data*. The NoC does not control cell size, supporting any size. The last cell byte is indicated by signal *eop* (end of packet).

The MOTIM NoC supports two switching modes: packet and circuit switching. Packet switching is used to establish connections between origin and destination. For each cell stored in the NI, the existence of a free path is verified, by sending a connection establishment control packet. If a path exists, the destination returns an acknowledge (*ack*) signal (see Figure 3). Next, the cell is sent in burst, using circuit switching. If no available path exists, a *nack* signal is sent, and the NI retries establishing connection after a time proportional to the cell size (currently 200 clock cycles).

The competition for using internal NoC channels by different data flows is inherent to the adopted topology. Only a full crossbar is free of blocking, but crossbar do not scale well. Physical channels can be multiplexed in several ways, to allow sharing by multiple flows. Examples are temporal multiplexing and spatial multiplexing. The first, also called time division multiplexing (TDM), sends data in predefined temporal *slots*. Space division multiplexing (SDM) [14] divides multi-wire physical channels into *n* groups, allowing *n* si-

multaneous parallel flows. MOTIM option is for SDM, with *n*=2 in the described instance. This ensures higher throughput than a TDM approach. In practice, MOTIM showed that the area cost of SDM is inferior to an approach using virtual channels, because the increase in router size to use the last approach more than compensates the additional connections among neighbor SDM routers.

MOTIM routers are also responsible for session control. When the connection establishment packet reaches the destination router, the latter verifies if a free session exists. If this is the case, a session is allocated to the origin router and an *ack* signal is back propagated. When the destination router is receiving a cell, the allocated session is indicated to NI using the signal *session* in Figure 3.

Figure 4 illustrates four distinct routers sending data to the same destination. Cells are stored in PC buffers. After storing a whole Ethernet packet in some buffer, the PC allows the MAC module to send this packet on. Note that path reservation proceeds cell by cell, while session reservation occurs for each Ethernet packet. After a cell leaves the NoC and before the next cell of the same packet enters the NoC, the internal NoC channels can be used by other flows. Considering that the NoC bandwidth is 8 times the rate of packets, cells generate traffic employing 12.5% of the available bandwidth.
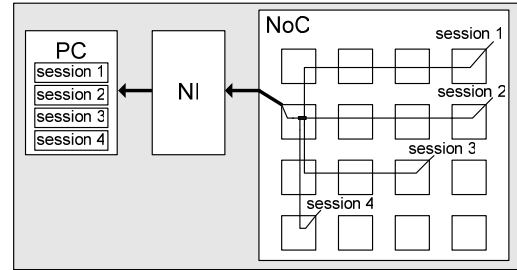

Figure 4 – Four simultaneously active sessions for the same destination router of MOTIM NoC. Each session stores data of a given Ethernet packet in a separate buffer.

5. NETWORK INTERFACE MODULE

The NI module has three functions: (*i*) receive cells from the PC, require session establishment to the NoC, and send cells to the destination router; (*ii*) forwarding cells from the NoC to the PC, with session indication; (*iii*) determining the destination router address based on the destination MAC address of the packet.

Figure 5 illustrates the NI external interfaces. The sense PC→NI→NoC employs the already described synchronous protocol. Whenever a complete cell is stored in the *cell buffer*, the NI requires a NoC circuit establishment. Once this is accomplished, cell sending occurs in burst. When active, the *offset* signal indicates to NI that the *data* lines contain either CSN or the offset. In case the cell buffer is full, packet loss may occur. When this happens, the NI asserts the *error* signal to the PC, which then starts to discard all bytes of the packet being sent. The *cell error* bit (Figure 2) is then asserted in the last stored cell, so that the destination port can discard all previously received cells of that packet. Once session establishment has reserved a buffer in PC, cells from the NoC to the PC are simply forwarded across the NI. When asserted, the *active* signal indicates that cell transmission in under way.
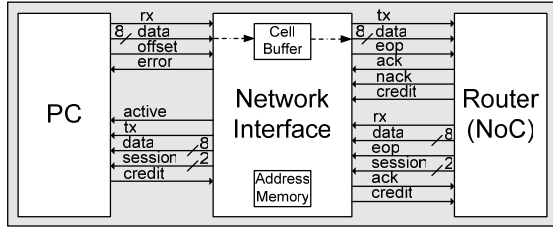
Figure 5 – NI module and its connection to PC and NoC.

High speed packet switching requires the use of mechanisms for routing Ethernet packets, since each MOTIM port may be virtually connected to several equipments with distinct MAC numbers. To deal with this problem, each NI contains an *address memory* with capacity to store up to p routing entries (here, p=256).

As detailed in Figure 6, each address memory entry contains: the MAC destination address, the number of the local port (remember that each router connects to two MACs), the NoC router address and the number of times the line was accessed.
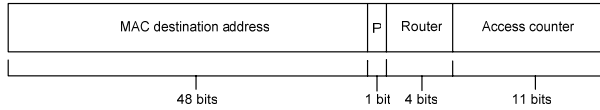

Figure 6 – Structure of an *address memory* entry in NI.

The *address memory* is initially empty. The dynamic filling of this memory confers flexibility to the system, allowing interconnecting new equipments *on-the-fly*. When an Ethernet packet enters MOTIM the NI obtains its MAC destination address. The memory is looked up for a match with the first field of some entry. If a match occurs, the packet is sent to the router with address specified in the third field of the entry, and the access counter is incremented. If no match occurs, the packet is broadcast to all routers.

When a packet arrives from the NoC to the NI, the MAC source address is also obtained. In case this address already exists in the address memory, no action is taken. Otherwise, a new entry is created with the field *access counter* initialized to 0. In this way, MOTIM learns the MAC locations.

Given the usually big size of the address memory, sequential search is avoided. From the MAC address, the application of a hash function generates the access addresses. In addition, the *access counter* field of each entry is used to implement an entry substitution policy that removes entries with smaller values, except those just inserted (with *access counter*=0).

6. PACKET/CELL MODULE

The PC module is responsible for fragmenting and reassembling. The external interfaces of the PC module are depicted in Figure 7. During Ethernet packet reception (MAC→PC) the PC executes a buffer-less fragmentation. Buffering is needless since: (*i*) the reception rate is much lower than MOTIM processing rate, leaving time to insert the three control bytes at the start of each cell; (*ii*) the control bytes relative to the cell size and cell sequence number are added only at the end of each cell.

Ethernet packet sending (PC→MAC) only occurs when a full packet is stored in some PC session buffer (mem0 to mem3, in Figure 7). The PC can interleave cell receptions of up to m distinct origins, storing cells of a same packet in one of the mem*i* buffers. There is a policy defined to detect full packets in these buffers that considers the packet arrival order, to avoid out of order transmission.
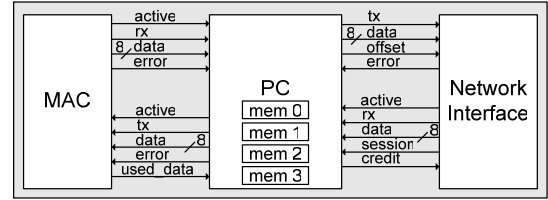

Figure 7 – Packet/Cell module and its connection to MAC and NI.

In case some cell reaches the PC with the *error* field asserted, all stored packet cells are discarded and the session buffer is freed. This avoids that incomplete or incorrect packets be sent to some MAC, and from there to the external world. However, this procedure implies latency proportional to the packet size. This is the biggest source of latency in the MOTIM architecture. Although it is possible to send cells as the PC module receives them, this is not advisable, because long intervals may occur between successive cells of a packet, blocking a MAC for too long, with consequent reduction of NoC utilization, which affects the global performance of the architecture.

7. EXPERIMENTAL RESULTS

This Section describes the functional validation of the MOTIM instance described above and the prototyping of a smaller instance of the architecture, together with initial data on its latency and area occupation.

7.1 Functional Validation

The test scenario depicted in Figure 8 has as objective evaluating: (*i*) the saturation point of the architecture, by injecting a large number of packets simultaneously into the NoC; (*ii*) the effectiveness of adopting SDM; (*iii*) verify how the architecture behaves under injection of Ethernet packets of several sizes, including maximum size packets (1500 bytes) and smaller than a cell packets (80 Bytes).

The traffic scenario of Figure 8 includes 12 traffic sources and 12 different traffic targets, injecting Ethernet packets at 100 Mb/s. Observe that in several links there are multiple simultaneously active flows, causing contention (see for example, the link between routers 4 and 8). Traffic is characterized as depicted in Table 1. Traffic is depicted in the Table as a function of origin address, destination address, hop count (distance between origin and destination), number of injected packets and packet size. To approach saturation, the inter packet gap (IPG, minimum delay between successive packets) for Fast Ethernet is used (0.96 μs), creating worst-case simulations. During the simulation, 2250 Ethernet packets were transmitted, a total volume of 1.432 MBytes.
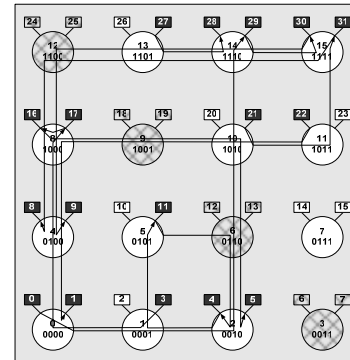

Figure 8 – Test scenario used for latency evaluation of MOTIM.

Table 1 – Traffic injection characteristics used in conjunction with the test scenario of Figure 8 (LP stands for Local Port).

Source	Target	Number of hops	Number of Packets	Packet Size
LP 0	LP 29	6	200	500
LP 3	LP 17	4	100	100
LP 4	LP 11	3	50	1500
LP 11	LP 4	3	200	500
LP 16	LP 5	5	100	1500
LP 17	LP 16	1	50	1500
LP 21	LP 31	3	200	1500
LP 22	LP 1	6	150	500
LP 27	LP 28	2	300	100
LP 29	LP 30	2	300	1500
LP 30	LP 9	6	500	70
LP 31	LP 8	6	400	80

Table 2 presents in the third column the estimated latency for each flow of the Figure 8 test scenario. From the fourth to the sixth column appear the latency values (minimal, average, maximal) obtained by HDL simulation.

Table 2 – Latency values for the test scenario presented in Figure 8, in clock cycles, MOTIM frequency=50MHz.

Source	Target	Estimated Latency	Minimum Latency	Average Latency	Maximum Latency
LP 0	LP 29	**2368**	2366	**2369**	2382
LP 3	LP 17	**614**	587	**618**	633
LP 4	LP 11	**6256**	6278	**6278**	6282
LP 11	LP 4	**2320**	2318	**2321**	2331
LP 16	LP 5	**6288**	6310	**6310**	6318
LP 17	LP 16	**6224**	6251	**6251**	6254
LP 21	LP 31	**6256**	6279	**6279**	6283
LP 22	LP 1	**2368**	2366	**2385**	2546
LP 27	LP 28	**598**	569	**600**	606
LP 29	LP 30	**6240**	6262	**6262**	6269
LP 30	LP 9	**510**	485	**542**	648
LP 31	LP 8	**550**	525	**599**	810

The estimated latency is computed according to Equation 1. Constant 54 corresponds to the fixed latency of the components MAC-NI-PC. The second part of the Equation corresponds to the *buffering latency*, proportional to the packet size. The term BR corresponds to the *bandwidth ratio*.

$$estimated\ latency = \frac{54 + packet\ size * BR\ +}{(hops * RL + 128)\ + latency_{last\_cell}} \quad (1)$$

In the present simulation, BR=4 (MOTIM frequency=50 MHz, Ethernet PHY frequency=12.5 MHz). The third part of the equation is the required time to transmit the last cell (t_{last_cell}), where RL corresponds to the *router latency*, being equal to six clock cycles, and 128 is the size of one cell. The fourth component is named *latency-$_{last\_cell}$*. If the required time to store the last cell ($mod(packet\_size/123)*BR$) is smaller than t_{last_cell}, $latency_{last\_cell}$ is equal to t_{last_cell} - $latency_{last\_cell}$, else it is null. The highest estimated latency observed in Table 2 is 6456 50-MHz clock cycles, corresponding to 129.12 μs.

Although in this scenario there is competition among flows for internal NoC channels and each local port is constantly sending data at maximum rate, no cell has been discarded and latency values displayed only small variations. Comparing the estimated and the measured average latencies, packet congestion created maximum

delays of approximately 50 clock cycles (1 μs). This test scenario demonstrates the effectiveness of the NoC mechanisms for low latency data transmission.

The second test scenario targets illustrating the behavior of the session control policy. Since the output rate is 100 Mb/s the summation of input rates cannot exceed 100 Mb/s, otherwise packet loss may occur. Two situations can be observed, both with 4 origins sending 50 1500-Byte packets to the same destination, but each using a distinct value of IPG.

The first situation, with IPG=300μs, results in an equivalent transmission rate of 28.6 Mb/s, causing the occurrence of packet loss. Table 3 illustrates the obtained latency values and the number of discarded packets. As can be observed in the Table, from the 200 packets transmitted, 25 were discarded. Note also the latency increase, which is superior to the estimated latency, due to the congestion of the NoC internal channels.

Table 3 – Latency values when 4 sources transmit data to the same target, IPG=300μs (50 1500-Bytes Ethernet packets per source).

Source / Target	Estimated Latency	Minimum Latency	Average Latency	Maximum Latency	Discarded Packets
LP 4/ 27	**6288**	12961	**28516**	38391	5
LP 14 /27	**6288**	10090	**27979**	38385	5
LP 16/ 27	**6256**	19004	**31290**	47269	5
LP 22 / 27	**6272**	6918	**24564**	41226	10

The second situation, with IPG=400μs, results in an equivalent transmission rate of 23.1 Mb/s, avoiding packet loss. Table 4 illustrates the obtained latency values and the number of discarded packets. As can be observed in the Table, although latency has increased, no packet has been discarded.

Table 4 – Latency values when 4 sources transmit data to the same target, IPG=400μs (50 1500-Bytes Ethernet packets per source).

Source / Target	Estimated Latency	Minimum Latency	Average Latency	Maximum Latency	Discarded Packets
LP 4/ 27	**6288**	12961	**19004**	25048	0
LP 14 /27	**6288**	12962	**19004**	25048	0
LP 16/ 27	**6256**	19004	**19004**	19005	0
LP 22 / 27	**6272**	6918	**6918**	6919	0

The above experiments demonstrate the effectiveness of the session control mechanism. It allows multiplexing the utilization of the NoC physical channels during packet transmission for a same destination. In case no session control existed, the PC input buffers would discard packets quite often. This would occur because a path would often be allocated to another packet, imposing a wait for its complete transmission.

7.2 Prototyping

To validate the MOTIM architecture in hardware, a reduced version was described and implemented. The characteristics of this version are: NoC 3x3, with five local ports transmitting data. The MAC modules were substituted by a packet generator. These packet generators display better controllability and observability for packets injected in the network than a full MAC. Nonetheless, the MAC has also been separately validated by FPGA prototyping.

Table 5 presents the area consumption for the MOTIM prototype, targeting a Virtex XC2VP30 FPGA device. The number of BRAMs in the set PC-NI is equal to 6, including: 1 BRAM to store the incoming packet, 4 BRAMs for session control, and 1 BRAM to implement the address memory. The traffic generation and traffic re-

ception use 3 BRAMs per router. Note that although not all LUTs have been consumed, almost all CLBs have been used up (99%) due to the flip-flops usage.

Table 5 – Area consumption for the MOTIM architecture prototyped on the Virtex XC2VP30 FPGA device.

Resource	Used	Available	Utilization
Function Generators (LUTs)	18,781	27,392	68%
CLB Slices	13,694	13,696	99%
Block RAMs	45	136	33%

The five local ports transmit Ethernet packets using the source and target addresses presented in the first column of Table 6. Each source transmits 200 500-Byte packets. A timestamp is inserted in the packet when it enters MOTIM. When the last byte of the packet leaves the network, the packet latency is computed, by subtracting the insertion timestamp from the present timestamp. The second and third columns of Table 6 present the estimated and measured latencies. The prototyping scenario does not present competition between flows, leading to similar values for estimated and measured latencies. This test demonstrates the correct operation of the architecture in a realistic hardware environment.

Table 6– Estimated and measured latencies for the MOTIM architecture FPGA prototype.

Source /Target	Estimated Latency	Average Measured Latency
LP 2 / LP16	2336	2334
LP 16 / LP2	2336	2334
LP 3 / LP 4	2304	2306
LP 4 / LP 11	2304	2306
LP 11 / LP 3	2320	2317

The whole system, with 24 NI modules, 24 PCs, 24 MACs, and a 4x4 NoC was synthesized for area occupation analysis only. Table 7 displays area results, obtained with the Leonardo Spectrum synthesis tool, for the Xilinx Virtex 2VP100 FPGA.

Table 7 – Resource occupancy for the full MOTIM architecture. on the Virtex 2VP100 FPGA device.

Resource	Used	Available	Utilization
Function Generators (LUTs)	65829	88192	74.64%
CLB Slices	32915	44096	74.64%
Block RAMs	144	444	32.43%

8. CONCLUSIONS AND FUTURE WORK

Scalability was a main concern during the MOTIM architecture development. MOTIM has thus been designed to allow extensive parameterization, including the physical channel width, the number of replicated physical channels (n), the amount of sessions (m), the address memory depth (p), and the NoC dimensions. The operating frequency is a function of the physical synthesis, which depend on the target implementation technology and synthesis tool performance and tuning. For example, if extra effort is put in the design it is not hard to implement a version of MOTIM with 16-bit width physical channels working at 200MHz. The resulting 3.2 Gb/s per channel bandwidth would suffice to support Gigabit Ethernet links. As for the use of NoCs, MOTIM appears as one of the still rare practical application of NoC concepts.

As ongoing work it is possible to cite the prototyping of the full MOTIM architecture, adding the MAC modules to the already prototyped parts. Next, comes the integration of the architecture with other modules to implement the whole system mentioned in Section 3. Also, producing a higher speed MOTIM supporting Gigabit Ethernet links is an activity currently under way.

9. ACKNOWLEDGMENTS

10. REFERENCES

[1] Minkenberg, C. et al. **Current Issues in Packet Switch Design**. ACM SIGCOMM Computer Communications Review, 33(1), January 2003. pp. 119-124.

[2] Saleh, R. et al. **System-on-Chip: reuse and Integration**. Proceedings of the IEEE, 94(6), June 2006. pp. 1050-1069.

[3] García, J. et al. **Design and Implementation of High-Performance Memory Systems for Future Packet Buffers**. In: International Symposium on Microarchitecture, December 2003. pp. 372-384.

[4] Bastos, E.; Carara, E.; Pigatto, D.; Calazans, N.; Moraes, F. **MOTIM - A Scalable Architecture for Ethernet Switches**. In: ISVLSI, 2007, pp. 451-452.

[5] Kumar, S. et al. **A Network on Chip Architecture and Design Methodology**. In IEEE Computer Society Annual Symposium on VLSI (ISVLSI), April 2002. pp. 105–112.

[6] Benini, L. and De Micheli, G. **Networks on Chips: A New Soc Paradigm**. IEEE Computer, 35(1), January 2002. pp.70–78.

[7] Moraes, F. et al. **Hermes: an Infrastructure for Low Area Overhead Packet-switching Networks on Chip**. Integration the VLSI Journal, 38(1), October 2004. pp. 69-93.

[8] Mohor, I. **Ethernet IP Core Design Document**. Revision 0.4, Available at http://www.opencores.org/cvsweb.shtml/ethernet/doc /eth_design_document.pdf, October, 2002. 46 pages.

[9] Fayyazi, M. and Kaeli, D. **Localized Message Passing Structures for High Speed Ethernet Packet Switching**. In: International Conference on Parallel and Distributed Processing Techniques and Applications, June 2002. pp. 1551-1557.

[10] Sivaram, R. et al. HIPIQS: **A High-Performance Switch Architecture Using Input Queuing**. IEEE Transactions on Parallel and Distributed Systems, 13(3), March 2002. pp. 275-289.

[11] Dua, A. et al. **Power Managed Packet Switching**. In: ICC 2007, pp. 357-362.

[12] Wolf. T. and Turner, J. S. **Design Issues for High-Performance Active Routers**. IEEE Journal on Selected Areas in Communications, 19(3), March 2001. pp. 404-409.

[13] Ishihara, T. et al. **Carrier-Grade Ethernet Switch for Reliable Wide-Area Ethernet Service**. Fujitsu Scientific and Technical Journal, 39(2), December 2003. pp 234-243.

[14] Leroy, A. et al. **Spatial Division Multiplexing: a Novel Approach for Guaranteed Throughput on NoCs**, In: International Conference on Hardware Software Codesign, 2005. pp. 81-86.

[15] Carara, E.; Moraes, F.; Calazans, N. **Router Architecture for High-Performance NoCs**, In: SBCCI 2007, pp. 111-116.