MOTIM – A Scalable Architecture for Ethernet Switches

Érico Bastos[1] Everton Carara[1] Daniel Pigatto[2] Ney Calazans[1] Fernando Moraes[1]
[1] *PUCRS - FACIN - Av. Ipiranga 6681- Porto Alegre - 90619-900 - Brazil*
[2] *DATACOM TELEMÁTICA - Av. França 735 - Porto Alegre - 90230-220 - Brazil*
{ebastos, carara, calazans, moraes}@inf.pucrs.br, daniel@datacom-telematica.com.br

The main goal of this work is to describe a scalable and reusable architecture useful for the construction of Ethernet switches, named MOTIM. 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 [1][2]. 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 [3]. 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 concept of the MOTIM switch architecture derived initially from an industry-academy cooperation targeting the implementation of the 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. Gigabit ports are future work. Four module types compose the architecture: Ethernet MACs, Packet-Cell (PC) modules, Network Interfaces (NI), and the Network-on-Chip (NoC).

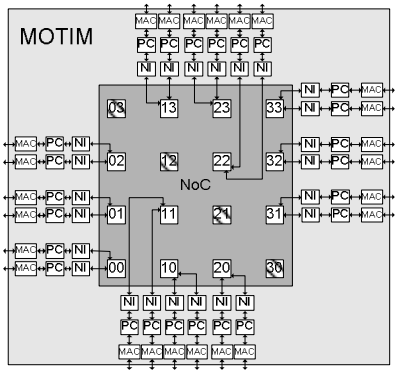

Figure 1 – MOTIM architecture instance. Modules NI, PC and MAC are instantiated 24 times and connect to the NoC module. The main diagonal routers of the NoC are reserved for special blocks.

The MAC Ethernet module is an adaptation of an IP Core available at Opencores [5]. 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 to the NoC. The data receive part of the NI module (NoC→NI) forward cells to the PC module and stores the relation between NoC router origin and MAC destination addresses.

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 other modules, such as control processor, temporary bulk memory and system supervision.

The NoC module is based on HERMES [4], 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 are connected to neighbor routers. A centralized round-robin arbitration grants access to incoming packets, and a deterministic XY routing algorithm is used to select the output port.

Fast Ethernet packets display two features leading to NoC sub utilization: low bandwidth with regard to 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 predetermined amount of data is available, the NoC receives these as a burst. This exact amount of data is called a *cell*. Its size on the MOTIM instance discussed here 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. Five bytes are used for control purposes: (*i*) the first bit signals if the cell is

IEEE Computer Society Annual Symposium on VLSI(ISVLSI'07)
0-7695-2896-1/07 $20.00 © 2007 IEEE

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 cell) 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.

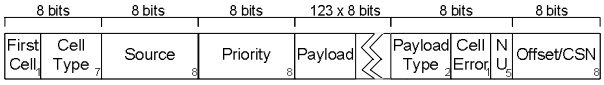

Figure 2 – Cell structure used in MOTIM instance.

Another relevant concept employed in MOTIM is session control. Since Ethernet packets are fragmented into cells, it is necessary to reserve the set NI-PC-MAC during reception 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 from distinct origins. The first cell from a packet reserves a buffer, allocating it until its last cell is received. In this way, up to *m* (*m*=4) distinct sources may simultaneously send data for a same destination, without blocking.

The main test scenario has as objective evaluating: (*i*) the saturation point of the architecture, by injecting a large number of packets simultaneously; (*ii*) the effectiveness of adopting Spatial Division Multiplexing; (*iii*) verify how the architecture behaves under injection of Ethernet packets of several sizes, including maximum size packets (1500 bytes) and less than a cell size packets (80 Bytes).

This traffic scenario includes 12 traffic sources and 12 different traffic targets, injecting Ethernet packets at 100 Mb/s. Traffic is 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 minimum inter packet gap (IPG) for Fast Ethernet is used (0.96 μs), creating worst case simulation. During simulation, 2250 Ethernet packets were transmitted, a volume of 1.432 Mbytes.

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

A reduced version the MOTIM architecture was prototyped in hardware as a NoC 3x3, with five local ports. The MAC module was substituted by a packet generator with better controllability and observability for packets injected in the network. The MAC has been separately prototyped.

Table 1 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 reception 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 1 – 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 whole system, with 24 NI modules, 24 PCs, 24 MACs, and a 4x4 NoC was synthesized for area occupation analysis. Table 2 displays area results, obtained with the Leonardo Spectrum synthesis tool, for the Xilinx Virtex 2VP100 FPGA.

Table 2 – 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%

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 channel, the amount of sessions, the address memory depth, 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. The MOTIM instance described here was synthesized with no major effort during synthesis to achieve the operating requirements. For example, if extra effort is put in the design it should no be 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.

REFERENCES

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

[2] 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.

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

[4] 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.

[5] 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.