A New Router Architecture for High-Performance Intrachip Networks

Everton Carara, Ney Calazans, Fernando Moraes

Faculdade de Informática - Pontifícia Universidade Católica do Rio Grande do Sul (FACIN-PUCRS)
Av. Ipiranga, 6681 – P. 32 – 90619-900 – Porto Alegre – BRAZIL
e-mail: {carara, calazans, moraes}@inf.pucrs.br

ABSTRACT

For almost a decade now, Network on Chip (NoC) concepts have evolved to provide an interesting alternative to more traditional intrachip communication architectures (e.g. shared busses) for the design of complex Systems on Chip (SoCs). A considerable number of NoC proposals are available, focusing on different sets of optimization aspects, related to specific classes of applications. Each such application employs a NoC as part of its underlying implementation infrastructure. Many of the mentioned optimization aspects target results such as Quality of Service (QoS) achievement and/or power consumption reduction. On the other hand, the use of NoCs brings about the solution of new design problems, such to the choice of synchronization method to employ between NoC routers and application modules mapping. Although the availability of NoC structures is already rather ample, some design choices are at base of many, if not most, NoC proposals. These include the use of wormhole packet switching and virtual channels. This work pledges against this practice. It discusses trade-offs of using circuit or packet switching, arguing in favor the use of the former with fixed size packets (cells). Quantitative data supports the argumentation. Also, the work proposes and justifies replacing the use of virtual channels by replicated channels, based on the abundance of wires in current and expected deep sub-micron technologies. Finally, the work proposes a transmission method coupling the use of session layer structures to circuit switching to better support application implementation. The main reported result is the availability of a router with reduced latency and area, a communication architecture adapted for high-performance applications.

Index Terms: Networks on Chip, Switching Modes, Virtual Channels, Session Layer.

1. INTRODUCTION

Important performance figures used to evaluate interconnection infrastructures such as shared busses and NoCs are latency, throughput and jitter. NoC performance is a function of design choices concerning switching mode, physical channel allocation and arbitration policies, buffering strategy and routing algorithms.

Buffering strategies, routing algorithms and arbitration policies are related to router structural parameters. On the other hand, switching mode and physical channel allocation reflect the way data transmission occurs between routers.

Most NoC proposals employ layered stacks similar to the OSI reference model of ISO [1] . The three lower OSI protocol layers (physical, link and network) are often implemented in hardware. The physical protocol layer is responsible for providing the electrical medium specifications to connect routers among them or routers to processing elements (PEs). The link protocol layer is responsible for the reliable transport of packets from one router/PE pair to another router/PE pair through a series of links while applying control strategies like handshake or credit based flow control. The network protocol layer is responsible for path determination and logical addressing (routing algorithms). The transport and session protocol layers respond for end-to-end connection, assembling and disassembling of messages, and end-to-end error handling. The transport and session protocol layers are not usually part of NoCs infrastructures.

This work has two main objectives. The first one is to discuss performance trade-offs for switching modes and physical channel allocation policies. The second objective is to propose increasing the number of protocol layers addressed by NoC infrastructures, by adding to it a session protocol layer. This comprises the inclusion of concerns about managing several simultaneous transmission sessions at router external links. The ultimate goal is to improve the overall NoC performance and utilization. Buffering strategies, routing algorithms and arbitration policies are not further discussed in this work.

The rest of this paper is organized as follows. Section 2 discusses switching modes employed in NoCs, assessing pro et contra arguments for circuit and packet switching modes. Section 3 revises channel multiplexing strategies for NoCs, discussing time division multiplexing and spatial division multiplexing, justifying the use of the last in current and future submicron technologies. Section 4 contains the main contribution of this work, the proposition of adding a session protocol layer to NoC infrastructures, coupled to the use of circuit switching. Section 5 presents some experimental results supporting the paper claims and Section 6 ends the paper, by presenting a set of conclusions and directions or further work.

2. SWITCHING MODES IN NOCS

Messages are data that have to be sent from a sender to a receiver through a network. Messages can be transformed into packets, by encapsulating all or part of each message with network control information. Alternatively, messages can be sent after a connection establishment between the sender and the receiver. This defines the two basic modes for message transmission in networks, packet switching and circuit switching, respectively.

Wormhole packet switching is the switching mode most commonly employed in NoCs [2]. Packet-switched networks often allow for high aggregate system bandwidth, as each packet can be distributed across a subset of network nodes at any given instant [3]. However, such networks generally require congestion control and packet processing, which include the need for buffers to queue packets awaiting the availability of routing resources. Correct buffer sizing is a fundamental parameter to optimize NoC performance. Small buffers increase network congestion and large buffers increase area and latency overhead. This switching mode supports well best-effort (BE) services [4], being efficient for traffic with short and frequent packets. HERMES [5], Xpipes [6], MANGO [7] and SoCIN [8] are examples of NoCs employing wormhole packet switching.

Another switching mode employed in NoCs is circuit switching. It can provide guaranteed throughput and latency bounds for individual packets, since an exclusive path is allocated to data transfers between source and target PEs. In addition, the buffering requirement is typically a single register instead of a FIFO buffer, since when the circuit is established the NoC acts like a pipeline where each router acts as a stage. The disadvantages of circuit switching are the channel bandwidth underutilization when data is transmitted at lower rates and the setup latency to establish a circuit, which depends on the traffic in the path during circuit establishment. This switching mode is more efficient for traffics with long packets transmitted at high rates, with requirements for throughput and latency guarantees. Representative circuit-switching NoCs are: PNoC [3], Æthereal [9], SoCBUS [10] and Octagon [11]. Æthereal employs circuit switching only for traffic with QoS requirements, while BE traffic uses wormhole packet switching.

Table I summarizes the main advantageous and inconvenient features of circuit switching and wormhole packet switching.

This paper proposes the use of circuit switching with fixed size packets, using the cell concept used e.g. in ATM [12]. A cell is first buffered and then transmitted to its target using circuit switching. The advantages of using circuit switching with buffered cells are:

1. A cell is sent to its destination if and only if a path exists between source and target PEs, avoiding network congestion;
2. A cell is transmitted at the network rate, not at the PE rate, improving channel bandwidth allocation (burst transmission);
3. Buffering in routers is reduced, due to the use of circuit-switching.

Cells do not only bring advantages. One problem with them is that, due to burst transmission, the source PE must have a buffer to store at least one cell, which may increase packet latency.

A trade-off between the discussed switching modes can be the strategy proposed in the Æthereal [9] NoC, i. e. to combine the two switching modes, with packet switching employed for BE traffic while the cell-based circuit switching deals with QoS traffic.

Table I. Advantages and disadvantages of NoC switching modes.

Features → Switching Mode	Advantageous	Inconvenient
Circuit Switching	- Guaranteed throughput and latency - Single register instead of FIFO buffers	- Static path reservation and possibly wasted bandwidth
Wormhole Packet Switching	- Shared NoC resources, enabling to distributes multiple flows simultaneously along routers	- Under heavy traffic, flits may block an important number of routers - Wasted bandwidth when the traffic initiator rate is slower than the channel rate

3. ROUTER CHANNEL MULTIPLEXING

Most NoCs can be modeled as a graph $G = <R, L>$, where the vertex set R is a set of routers, and the edge set L represents its bidirectional communication links. Each link normally represents two unidirectional channels in opposite directions, enabling the communication between neighbor routers. Channels can be multiplexed, allowing the use of a same channel by different flows in the same direction, improving the NoC performance. NoC literature describes the use of time and spatial division multiplexing techniques.

Time division multiplexing (TDM) is the sharing of physical channels in time, dividing these into logical channels (also called virtual channels or VCs) [13]. In this scheme, at each side of a physical channel, multiple data sources and data sinks area available to be connected through the channel. Each data source connected to a data sink through the channel at any given moment constitutes a *logical channel* or a VC. Time is usually partitioned into equally-sized periods called *time-slots.* During a time-slot, the available bandwidth is exclusively dedicated to a given logical channel. A given packet may need several time slots to be transmitted through a logical channel, and these time slots may be interspersed with time slots used by other packets flowing in different logical channels. TDM reduces NoC overall congestion, and consequently improves NoC performance. The insertion of VCs also allows the use of special policies to reserve time slots for certain flows, i. e. bandwidth reservation, enabling QoS support. However, separate buffers are required for each VC, and a time slot table is required to store VCs allocation. Such table is required when a priority scheme is employed to guarantee QoS. The additional buffers and the time slot table increase the dissipated power and the silicon area [14]. Æthereal [9] and Nostrum [15] are two early NoCs that suggested the use of VCs.

Figure 1 illustrates a typical TDM router architecture. The main router components are: (*i*) a switch control, responsible for arbitration and routing; (*ii*) a crossbar, to connect the input ports to the output ports; (*iii*) input FIFO buffers, for temporary flit storage. It is important to observe in this Figure the presence of de-multiplexers at the input ports, and multiplexers at the output ports, which may significantly increase the router area.

In current technologies, a phit[1] size equal to 32 or 64 bits underutilizes the amount of wires that can be implemented to connect neighbor routers. Consider for example a SoC region as depicted in Figure 2, using a 90 nm technology, with 140 nm wire pitch and 0.1 mm² router area [16]. Even considering the use of only one metal layer, each router could be connected to its neighbor using up to 2258 wires. Such a scenario clearly favors the use of some kind of spatial multiplexing, instead of TDM.

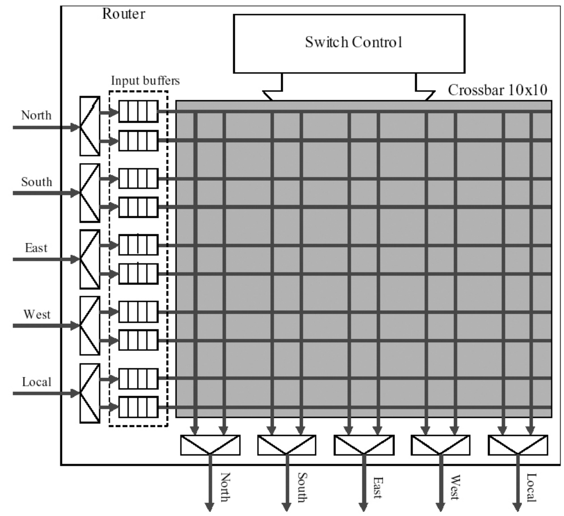

Figure 1. Basic structure of a router with virtual channels using TDM.

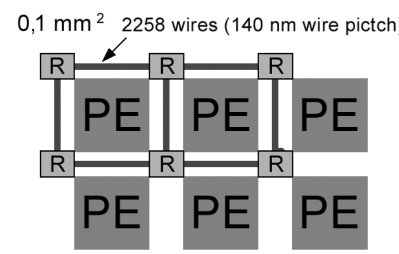

Figure 2. An example of a NoC-based SoC region, showing the maximum number of wires connecting routers (R), in current technologies. It assumes a 90 nm technology, with 140 nm wire pitch and 0.1 mm2 routers.

Early NoC designs that pledge spatial multiplexing may employ either *spatial division multiplexing* (SDM) [14] or *lane division multiplexing* (LDM) [17].

Leroy et al. [14] proposed to divide a physical channel in groups of wires. The number of wires assigned to each flow is a function of its required bandwidth. This method allocates each subset of wires for the whole connection lifetime (as in circuit switching). Data must be serialized and de-serialized at the source and target PEs, respectively. Results presented in this work, using as a case-study a video application, show a gain of 8% on energy consumption and 24% router area reduction, compared to a TDM router implementation. However, SDM increases the critical path by 37%.

[1] A *phit* is the smallest piece of data sent through a physical channel in a NoC. Usually it corresponds to the width of the data signal between two routers, or between a router and a PE.

In a similar work, Wolkotte et al. [17] proposed the Lane Division Multiplexing (LDM) technique, also employing circuit switching. Differently from [14], this work divides each channel into fixed size lanes. Results presented by the Authors comparing the circuit switching LDM router to two packet switching routers display lower power consumption, a smaller chip area and higher maximum throughput. The disadvantages of LDM are the lack of flexibility in router design (fixed lane size) and no support for BE traffic.

SDM and LDM allocate wires for a given flow depending on the required bandwidth, requiring control circuitry and serialization and de-serialization modules. This may incur in area, latency and power overheads comparable to using VCs coupled to TDM. Given the abundance of area available for wiring between routers, this paper proposes a router architecture that simply replicates physical channels in all directions (N, S, E, W, Local), avoiding the extra circuitry of VC+TDM, SDM or LDM.

Figure 3 illustrates the basic structure of a router architecture employing replicated physical channels (RC).

Note in Figure 3 the suppression of de-multiplexers and multiplexers (compared to Figure 1), which significantly reduces router area, as will be shown in Section 5. The switch control complexity in both approaches is similar, since its main function is to control the internal crossbar. The input buffers in both approaches have the same size, requiring the same amount of silicon area.

The replicated channels approach doubles the router bandwidth, when compared to the same number of virtual channels. Also, as can observed in Figure 3, the Local port may receive n distinct flows, where

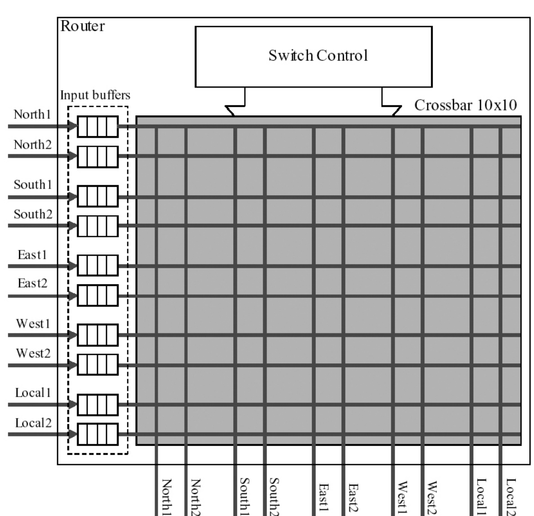

Figure 3. Basic structure of a router with replicated physical channels.

n is the replication degree. This feature allows connecting n PEs to the same router, thus reducing the number of required routers and the total SoC area.

4. SESSION PROTOCOL LAYER

The reasoning behind the use of circuit switching coupled with a session protocol layer resides in the fact that NoC links always display a much higher bandwidth than individual application rates. Consider, for example, a 16-bit 200 MHz router: the available bandwidth per channel is 3.2 Gbps. In contrast, the rate of an application requiring a large amount of bandwidth, such as an HDTV stream (MPEG2), is 15 Mbps.

This paper proposes a transmission method that is able to adapt application rates and NoC physical channel rates. This is achieved by first packing data in a source buffer at the output of the source traffic generator, and then transmitting the packet to the NoC in burst. Figure 4 illustrates an application producing data with a rate inferior to the NoC link rate, and the corresponding packaging before transmission to the NoC. Here, fixed size packets (cells) are adopted. This simplifies buffer sizing and session management. This source buffer ensures that each data transmission (a cell) occurs at the channel rate, avoiding idle time between flits, maximizing the use of channel bandwidth. Interleaving cells from several data flows can then be used to maximize NoC utilization.

Data can be transmitted using connection (circuit switching) or connectionless methods (wormhole packet switching). With wormhole packet switching, cells may be blocked inside the network, increasing the latency. The major benefit on using cells occurs when circuit switching is associated to the use of a session protocol layer. The explanation of the proposed method requires some definitions provided next.

Definition 1: *Physical connection* - Corresponds to the establishment of a circuit between the source and target PEs, *for each* cell of a message. This defines a fixed path for all cells of a message between source and target PEs.

Definition 2: *Session* - Corresponds to the reservation of one of the Local Ports at the target router (target PE) for all cells coming from the source PE. The session is established by the first cell of the

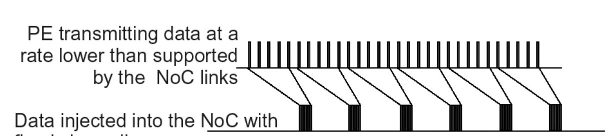

Figure 4. Data packaging in cells, for coupling application and NoC transmission rates.

message, being released after the last cell of the message is received. This reservation is necessary to avoid the interleaved reception of cells belonging to different source PEs at the same port.

For each cell, a *control packet* establishes the *physical connection*. This packet traverses the NoC using BE wormhole packet switching. As successive routers forward the *control packet*, the physical connection resources are allocated. The cell position inside the message is indicated by the control packet and can be one of: (*i*) first cell; (*ii*) middle cell(s); (*iii*) last/single cell. A control packet is composed by two flits, each divided in two fields as Figure 5 depicts, for an 8-bit flit size and 5-bit network addresses. The network address is composed by router (bits 3-0) and local port addresses (bit 4).

When the control packet reaches the target PE, this router back propagates an acknowledge signal (**ACK**), starting the circuit set up definition. After the back propagated signal reaches the source PE, the connection is established. Then, the cell is transmitted, one flit per clock cycle per hop, using circuit switching. A physical connection is broken when the last flit of the cell is transmitted, using a sideband signal named EOP[2]. Note that control packets may find congestion, increasing the time to set up the circuit.

A *session* may be established when the control packet of the first cell of a message arrives at the target PE, requiring a physical connection. If the local port at the target PE is not reserved, the first acknowledge signal sets up both the physical connection and the session. If a session is already established at the target PE, a non-acknowledge signal is back propagated to the source PE indicating that, even if a path exists in the network, the target PE is already receiving data from another PE. The non-acknowledge signal releases all reserved resources between the source and target PEs. If no session is available, the source PE tries to set up a new session after a certain time (in this implementation, a time proportional to the duration of a cell transmission). The session remains active up to the last cell of the message. Figure 6 illustrates the method to transmit messages of any size (e.g. video frames, Ethernet packets, cache blocks).

The protocol comprises three steps:

1. Store data in the source buffer and require a session establishment through a physical connection procedure.
2. Transmit all remaining cells except the last one through physical connections (one per cell), using the active session.
3. Transmit the last cell through a physical connection, closing the active session.

The use of circuit switching, coupled with a session protocol layer improves network performance, because all cells are sent at the network rate. Resource reservation during circuit switching does not reduce performance, since the cell is already stored in the source buffer.

A bottleneck of the proposed approach arises when multiple sources try to simultaneously send messages to the same target PE. This happens, e.g., for a shared memory in an MPSoC. The solution to this problem is to include in the target PE multiple *session buffers*. Each session buffer stores message cells from a unique source. They are placed outside the NoC (at the PE side), in the network interface or inside the PE. Figure 7 illustrates the placement of source and target *session buffers*. Using *session buffers*, the target PE may concurrently receive *k* messages, being *k* the number of *session buffers*. The *session buffers* must be sized to store at least one complete message (for example, an Ethernet packet or a cache block).

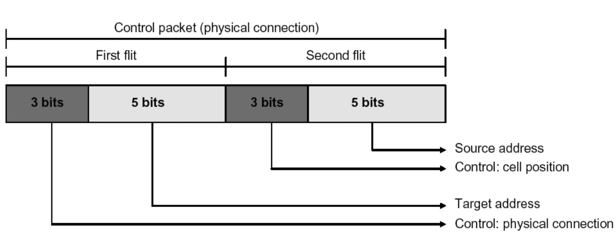

Figure 5. An example of a control packet structure for the proposed router, assuming 8-bit flits and 5-bit network addresses.

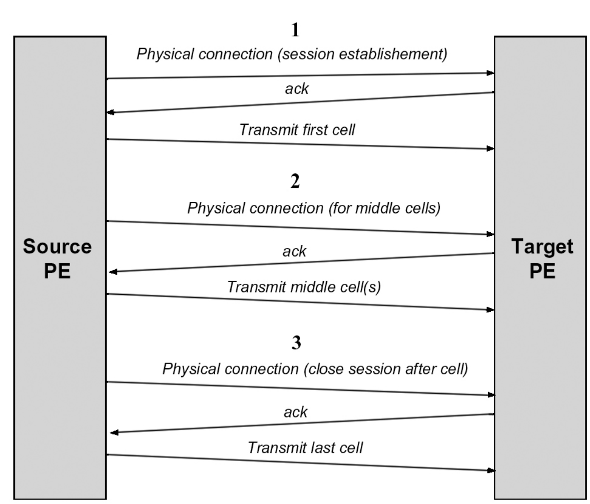

Figure 6. Basic protocol to transmit multi-cell messages.

[2] The EOP (end of packet) signal enables the use of variable size packets. An exceptionally sized cell in the approach described is the last cell of a message, which can be smaller than the cell size.

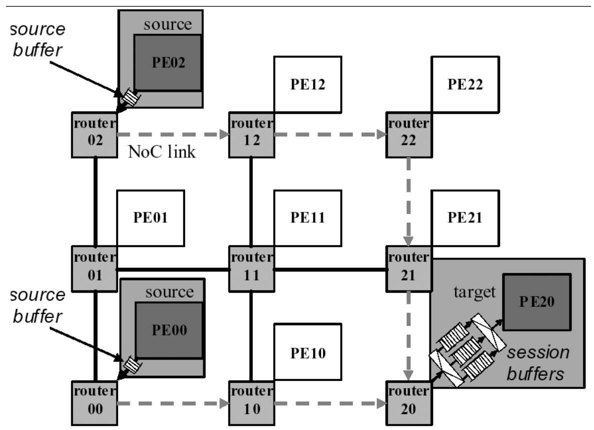

Figure 7. Buffers included in the PE wrappers when allowing multiple sessions per PE.

Multiple simultaneous sessions require session tables inside the routers or PE wrappers. A session table associates an established session (session buffer) with a source PE. Whenever the control packet reaches a target router, the session table is consulted. If there is an established session associated to the source PE indicated in the control packet or a free session, an acknowledge signal is back propagated. Otherwise a non-acknowledge signal is sent back. Only routers that need supporting simultaneous sessions (session buffers) have session tables.

5. RESULTS

A NoC implementation applying the proposed methods is available, and the results of evaluating it are the object of this Section. This NoC implements the methods directly in RTL VHDL, and derives its structure from the *HERMES* NoC [5] infrastructure. Performance figures like latency, throughput and total time to deliver messages derive from the use of RTL simulation.

A. Virtual Channels X Replicated Channels

This Section compares the architectures presented in Figure 1, virtual channels using TDM and Figure 3, replicated channels. Table II presents the common features of both architectures. This experiment evaluates only multiplexing strategies, without employing circuit switching.

Table II. Common features for VC and RC architectures.

Flit/phit size	8 bits
Flow control	credit based
NoC topology	mesh 4x4
Routing algorithm	deterministic XY
Switching mode	packet switching/wormhole

Input buffers have the same size in both architectures. The virtual channel architecture has 16-flit deep buffers for each logical channel. The replicated channel architecture has 16-flit deep buffers for each physical channel.

The goal of using virtual or replicated channels is reducing congestion when different flows compete for the same path inside the network. Figure 8 illustrates the traffic scenario used to evaluate latency and throughput. This scenario is justified by the amount of concurrent flows in the same channel. Lines indicate the path taken by packets from source to target routers. Ellipses highlight channels where two flows compete for a link.

Each source-target pair in Figure 8 transmits 500 257-flit packets (2 header flits and 255 payload flits). Packets enter the network at the channel rate.

Table III presents the average latency to transmit one packet and the average throughput per packet. The latency values include the network latency, proportional to the number of hops, and the packet latency, proportional to the packet size. This latency is computed as the time elapsed, in clock cycles, between the first packet flit injection in the source router and the reception of the last packet flit at the target router local port. The throughput is relative to the channel bandwidth (%). It is defined as the ratio between the packet size (257 bytes) and the time elapsed (clock cycles) between the reception (target router) of the first and last packet flit, multiplied by 100.

It is clear from these results that replicated channels reduce latency by 47.3% in average, and doubles the throughput, when compared to virtual channels. This is an expected result, since the internal NoC bandwidth doubles. Without congestion, both approaches have equal latency.

Table IV presents area consumption for FPGA mapping. For a single router (5-port router columns), a 12% area reduction is observed when using the replicated channels approach. For a 4*x*4 NoC the area reduction is 15%.

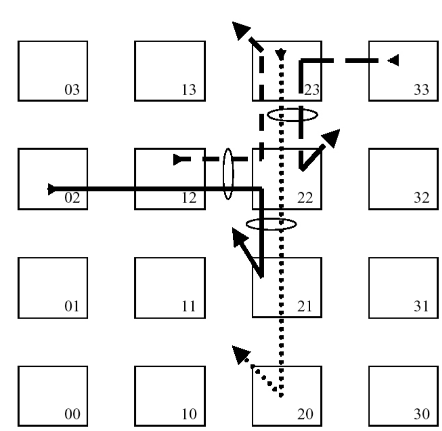

Figure 8. Spatial traffic distribution for latency evaluation when comparing VC and RC architectures.

Table III. Average latency (LT) and throughput (TH) values, for Virtual Channels (VC) and Replicated Channels (RC).

Source	Target	VC -LT	VC-TH	RC-LT	RC-TH
02	21	580	50,8%	305	100%
12	23	546	51,2%	290	100%
33	22	556	51,8%	302	100%
23	20	570	51,2%	290	100%

Table IV. Area results for Virtual Channels (VC) and Replicated Channels (RC), targeting a Xilinx Virtex2 family XC2VP30 FPGA.

Resource	5-port router		4 x 4 mesh Noc		Available
	VC	RC	VC	RC	
Slices	861	758	10538	8904	13696
LUTs	1722	1515	21075	17808	27392
Flip Flops	455	398	5866	5057	29060

Table V. Area results for Virtual Channels (VC) and Replicated Channels (RC), targeting a 0.35 μm ASIC library.

Resource	5 ports router		4 x 4 mesh Noc	
	VC	RC	VC	RC
Equivalent gates	6709	6416	83952	78759
16*x*16 bits memory blocks	5	5	64	64

Table V presents area consumption data for ASIC mapping (0.35 μm ASIC library), considering the number of equivalent gates and a macro-cell "16x16 bits memory blocks" to implement the buffers (5 per router). For the single router and the 4*x*4 NoC, a 4% and 6.4% area reduction is observed when using the replicated channels approach.

The observed area reduction obtained with the physical channel replication approach instead of multiplexing is due to the elimination of input demultiplexers, output multiplexers and TDM logic responsible for its control (see Figure 1). The area gains obtained in FPGA are higher because the multiplexers are implemented using LUTs, while in ASIC they are implemented using logic gates.

B. Session Protocol Layer Evaluation

The next experiments evaluate the benefits of adding a session protocol layer over circuit switching. The NoC has the same features presented in Table II, except for the switching mode, which is now circuit switching. Input buffers has 16-flit deep. Single links connect routers, with neither VCs nor RCs.

The cell size is a fundamental parameter in the proposed transmission method. Figure 9 depicts the traffic scenario used to evaluate the NoC performance as a function of the cell size for different injection rates. All flows compete with at least another flow for the same resources. All six traffic initiators (represented as shaded squares) send one 1280-byte message. The initiators PE00 and PE10 start transmitting first, inducing blocking situations for the remaining initiators.

As explained before, the goal of using circuit switching coupled with session protocol layer is to reduce internal NoC resources reservation when PEs are transmitting data at rates lower than the NoC rate. The link rate is 400 Mbps, deriving from a NoC frequency of 50 MHz and a flit size of 8 bits. The rate of traffic initiators varies from 66.4 Mbps, obtained from an injection rate of 16.6 % to 160 Mbps, derived from a 40% injection rate.

Figure 10 illustrates the number of clock cycles to transmit all messages as a function of the cell size (*CS*) for different injection rates (*IR*). The obtained results correspond to the traffic scenario of Figure 9, where flows compete for the same links.

For lower injection rates (16.6% and 20%) the time spent transmitting messages increases linearly with the cell size. The idle time between cells in lower injection rates favors the sharing among flows in the same link and reduces the impact of circuit switching

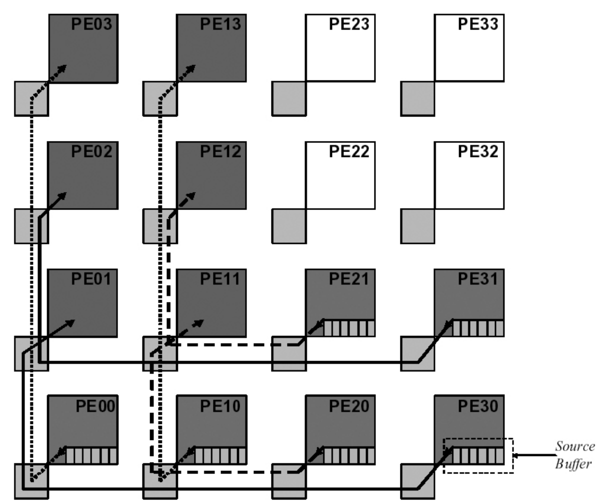

Figure 9. Spatial traffic distribution used to evaluate the NoC performance in function of the cell size.

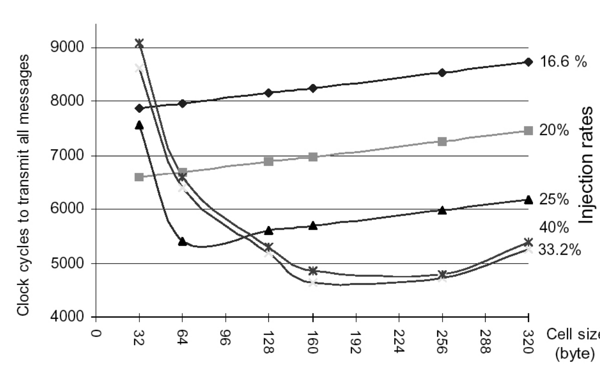

Figure 10. Performance as a function of the cell size for different injection rates.

connection establishment time. The linear growth observed in latency for these injection rates is due to the increase in time for resource allocation by flows using bigger cells, which increases network congestion.

The connection establishment time at higher injection rates (33.2% and 40%) penalizes smaller cell sizes. At high injection rates, small cells are completely stored in the source buffer faster. As soon as a cell is transmitted, there is another cell stored and ready to be transmitted. The control packets used to establish the physical connection are more frequent and this increases network congestion. Hence, connection establishment time becomes larger than the time to transmit the cell itself. For example, the time spent to connect two routers for a 3-hop path is 25 clock cycles, if there is no contention. Transmitting a 32-flit cell requires 32 clock cycles. Consequently, each small cell has its latency doubled due to the connection establishment activity. As cell size grows, these are completely stored in the slower source buffer and the control packets are less frequents. However, for larger cells the link bandwidth is dominated by one flow, increasing again the time to transmit the messages. At an intermediate injection rate like 25%, using small cells (32 and 34 flits) leads to a behavior similar to that observed at high injection rates, while using larger cells produces a behavior similar to that seen at low injection rates.

This experiment points to an intermediate cell size as best compromise, say 128 flits. Smaller cells penalize performance at high injection rates, due to connection establishment time and bigger cells increase congestion inside the NoC for all injection rates.

The previous experiment employed only one session per PE, since all sources have different targets. The next experiment exploits multiple sessions and compares the proposed transmission method against transmission using only circuit switching. Figure 11 illustrates the traffic scenario used to evaluate the NoC performance as a function of the number of simultaneous sessions. Source PEs generate data at a 20% injection rate and store it in the source buffer. Here, PE00, PE10, PE20 and PE30 transmit each a set of 50 1280-byte messages concurrently, each message being divided into 10 128-byte cells. All messages have PE03 as target. Besides having the same target PE, conflicts occur also in internal links. In the transmission using only circuit switching, as soon the first message flit is stored in the source buffer, the PE tries to establish a physical connection with the target. This physical connection allocates the resources until the end of the complete message transmission (1280 bytes). The last message flit releases the resources.

Table VI presents the average message throughput relative to the local port bandwidth, and the total time for PEs to transmit all messages, in clock cycles. Throughput is defined as the ratio between the message size (1280 bytes) and the time elapsed in clock cycles between the reception at the target PE of the first control packet (i.e. physical connection establishment) and the recepetion of the last message flit, multiplied by 100.

In the transmission using only circuit switching (CS column), PE00 establishes the physical connection before the other PEs. Since it has few message flits stored when the connection is established, the message is transmitted as data is generated. Hence, the average message throughput is close to generation data rate (20%). While PE00 transmits in a lower rate than the NoC link bandwidth, others PEs store the message in the source buffer. When another PEs establishes a physical connection, the message is almost completely stored and is transmitted in burst. Hence, the average message throughput is close to 100%.

Using the proposed transmission method with simultaneous sessions (2, 3 and 4), the NoC resources are maximized and efficiently shared by the flows. The average message throughput becomes close to the

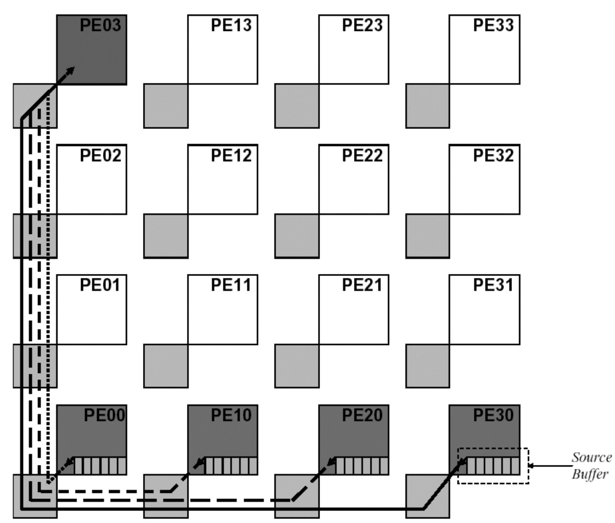

Figure 11. Spatial traffic distribution used to evaluate the NoC performance in function of the number of sessions.

Table VI. Average message throughput and total transmission time (clock cycles). CS = Circuit Switching.

	Source	Target	CS	CS with Simultaneous Sessions		
				2	3	4
Average throughput	PE00	PE03	25%	40%	22%	21%
	PE10	PE03	99%	48%	35%	21%
	PE20	PE03	98%	36%	29%	21%
	PE30	PE03	98%	60%	36%	28%
Total transmission time			**454517**	**384185**	**356500**	**341048**

generation data rate and the total transmission time reduces as the number of simultaneous session increases. Four simultaneous sessions present a 25% performance gain when compared to CS used alone.

If there is only one session per PE, the target PE does not require a buffer session. However, the cost to add simultaneous sessions is one source buffer per session, each sized to the longest possible message size.

In a NoC design, only few PEs are expected to receive simultaneous sessions. For example, in an MPSoC, shared memories may receive simultaneous write messages, or a communication PE may also receive simultaneous Ethernet packets to transmit to the external circuitry. In such situations, simultaneous sessions are a solution to reduce hot spots, and the overall latency.

6. CONCLUSIONS AND FUTURE WORK

This paper proposed methods to increase the overall performance of NoC routers. Results show significant performance gains, demonstrating the effectiveness of the propositions, even with higher injection rates and flows competing for the same physical channel. Both replicated channels and circuit switching achieve latency reduction through congestion reduction. Replicated channels increase router bandwidth, whereas circuit switching coupled with a session protocol layer maximizes the physical channel utilization.

Channel replication relies on abundance of routing area in deep sub-micron technologies. The method reduces both latency and circuit area, and it is an advantageous alternative to the use of virtual channels for most situations.

A session protocol layer shares the physical channels similarly to virtual channels. The main difference relies in the abstraction level. Virtual channels share the physical channels at the packet level, while a session layer shares the physical channels at the flow level. This technique can also be used to reduce hot spots, since it allows PEs to handle several simultaneous connections.

Future works include evaluating NoCs employing replicated channels together with session layers and the analytical definition of the cell size as a function of message sizes and input rates.

ACKNOWLEDGEMENTS

This research was supported partially by CNPq (Brazilian Re-search Agency), project 300774/2006-0.

REFERENCES

[1] Dehyadgari, M.; Nickray, M.; Afzali-kusha, A.; Navabi, Z. "*A new protocol stack model for network on chip*". In: IEEE Computer Society Annual Symposium on Emerging VLSI Technologies and Architectures, 2006. 3 pp.

[2] Bjerregaard, T.; Mahadevan, S. "*A survey of research and practices of Network-on-chip*". ACM Computing Surveys, 38(1), 2006, pp. 1-51.

[3] Hilton, C.; Nelson, B. "*PNoC: a flexible circuit-switched NoC for FPGA-based systems*". IEEE Proceedings on Computers and Digital Techniques, 153(3), May 2006, pp. 181-188.

[4] Jantsch, A.; Tenhunen, H. "*Networks on Chip*". Kluwer Academic Publishers, 2003, 303p.

[5] Moraes, F.; Calazans, N.; Mello, A.; Möller, L.; Ost, L. "*HERMES: an Infrastructure for Low Area Overhead Packet-switching Networks on Chip*". Integration the VLSI Journal, 38(1), Oct. 2004, pp. 69-93.

[6] Benini, L.; Bertozzi, D. "*Xpipes: A Network-on-Chip Architecture for Gigascale Systems-on-Chip*". IEEE Circuits and Systems Magazine, Second Quarter, 2004, pp. 18-31.

[7] Bjerregaard, T.; Sparso, J. "A *Router Architecture for Connection-Oriented Service Guarantees in the MANGO Clockless Network-on-Chip*". In: Proceedings of the Design, Automation and Test in Europe, DATE'05, 2005, pp. 1226-1231.

[8] Zeferino, C. A.; Susin, A. A. *"SoCIN: a Parametric and Scalable Network-on-Chip*". In: 16th Symposium on Integrated Circuits and Systems Design, SBCCI'03, 2003, pp. 169-174.

[9] Goossens, K.; Dielissen, J.; Radulescu, A. "*Æthereal network-on-chip: concepts, architectures, and implementations*". IEEE Design & Test of Computers, 22(5), Sept.-Oct. 2005, pp. 414-421.

[10] Wiklund, D.; Liu, D. "*SoCBUS: Switched Network-on-Chip for Hard Real Time Embedded Systems*". In: Proceedings of the 17th IEEE International Parallel and Distributed Processing Symposium, Apr. 2003, pp. 113-116.

[11] Karim, F.; Nguyen, A.; Dey, S. "*An Interconnect Architecture for Networking Systems on Chips*". IEEE Micro, 22(5), Sept.-Oct. 2002, pp. 36-45.

[12] Giroux, N.; Ganti, S. "*Quality of Service in ATM Networks: State-of-Art Traffic Management*". Prentice Hall, 1998, 252 p.

[13] Mello, A.; Tedesco, L.; Calazans, N.; Moraes, F. "*Virtual channels in networks on chip: implementation and evaluation on HERMES NoC*". In: 18th Symposium on Integrated Circuits and Systems Design, SBCCI'05, 2005, pp. 178-193.

[14] Leroy, A.; Marchal, P.; Shickova, A.; Catthoor, F.; Robert, F.; Verkest, D. "*Spatial Division Multiplexing: a Novel Approach for Guaranteed Throughput on NoCs*". In: Proceedings of the 3rd IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis, 2005. CODES+ISSS'05, 2005, pp. 81-86.

[15] Millberg, M.; Nilsson, E.; Thid, R.; Jantsch, A. "*Guaranteed Bandwidth Using Looped Containers in Temporally Disjoint Networks within the Nostrum Network-on-Chip*". In: Proceedings of the Design, Automation and Test in Europe, DATE'04, 2004, pp. 890-895.

[16] Leibson, S. "The Future of Nanometer SOC Design". In: Proceedings of the International Symposium on System-on-Chip, SOC'06, 2006, pp. 1-6.

[17] Wolkotte, P. T.; Smit, G. J. M.; Rauwerda, G. K.; Smit, L. T. "*An Energy-Efficient Reconfigurable Circuit-Switched Network-on-Chip*". In: Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium / 12th Reconfigurable Architectures Workshop, IPDPS'05/ RAW'05, 2005, pp. 155a.