

Spatial Division Multiplexing: a Novel Approach for Guaranteed Throughput on NoCs

A. Leroy^{*}, P. Marchal, A. Shickova, F. Catthoor[†], F. Robert[‡], D. Verkest[§]
IMEC vzw, Kapeldreef 75, Leuven, Belgium
Anthony.Leroy@ulb.ac.be

ABSTRACT

To ensure low power consumption while maintaining flexibility and performance, future Systems-on-Chip (SoC) will combine several types of processor cores and data memory units of widely different sizes. To interconnect the IPs of these heterogeneous platforms, Networks-on-Chip (NoC) have been proposed as an efficient and scalable alternative to shared buses. NoCs can provide throughput and latency guarantees by establishing virtual circuits between source and destination. State-of-the-art NoCs currently exploit Time-Division Multiplexing (TDM) to share network resources among virtual circuits, but this typically results in high network area and energy overhead with long circuit set-up time.

We propose an alternative solution based on Spatial Division Multiplexing (SDM). This paper describes our first design of an SDM-based network, discusses design alternatives for network implementation and shows why SDM should be better adapted to NoCs than TDM for a limited number of circuits.

Our case study clearly illustrates the advantages of our technique over TDM in terms of energy consumption, area overhead, and flexibility. SDM thus deserves to be explored in more depth, and in particular in combination with TDM in a hybrid scheme.

Categories and Subject Descriptors

B.4.3 [Input/Output and Data Communications]:
Interconnections (Subsystems)

General Terms

Design

Keywords

Network-on-Chip, Spatial Division Multiplexing

^{*}also Ph.D. student at the Université Libre de Bruxelles, Brussels, Belgium

[†]Also Professor at the Katholieke Univ. Leuven, Belgium

[‡]Professor at the Université Libre de Bruxelles, Brussels, Belgium

[§]Also Professor at the Katholieke Univ. Leuven, Belgium and at the Vrije Universiteit Brussel

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CODES+ISSS'05, Sept. 19–21, 2005, Jersey City, New Jersey, USA.
Copyright 2005 ACM 1-59593-161-9/05/0009 ...\$5.00.

1. INTRODUCTION

Traditional on-chip communication architectures based on buses will no longer be adequate for future Systems-on-Chip because of the high bandwidth requirements and dynamic characteristics of next-generation applications (e.g. multimedia codecs). The research community has therefore proposed Networks-on-Chip (NoC) as a good alternative to buses [4] [9] [2].

In most NoCs, IP-blocks are connected to their own router through a network interface. Routers are interconnected to each other by point-to-point links to form a given network topology (e.g. mesh, torus, ...). Their role is to forward the data from the source to the destination IP.

In real-time systems, many IP-blocks are subjected to performance/throughput constraints. One very simple way of providing guarantees on throughput and latency between two IP blocks consists of establishing a virtual circuit. This virtual circuit is exclusively dedicated to communication between them. Multiple virtual circuits can share the same physical communication resources (e.g. links). This concept is known as *Switched Virtual Circuit (SVC)*.

The best-known approach to implement SVC is Time Division Multiplexing (TDM). In this scheme, the time is discretized in equally long periods of time called time-slots. During a time-slot, the available bandwidth is exclusively dedicated to a given virtual circuit. Network resources are thus shared consecutively in time among the different circuits.

Fig. 1 (a) presents a local view of a TDM-based SVC network. IP1 and IP2 are connected to their own router through their Network Interfaces (NI). In addition to the NI port, routers R1 and R2 have four other ports (North, East, South and West) connected to adjacent routers. The focus of the figure is on the 8-bit link between router R1 and router R2. Several circuits of different bandwidth requirements are present on Fig 1: circuit A requires half of the link bandwidth, circuit B, a quarter and circuit C and D, one eighth. Assuming an 8 time-slots TDM, the link is dedicated exclusively to circuit A for time-slots 4 to 7, to circuit B for time-slots 2 and 3, to circuit C for time-slot 1, and to circuit D for time-slot 0. For each time slot, router R1 looks in its Output Reservation Table (ORT) to determine which port has exclusive access to the R1-R2 link (East port). Thereafter, it configures its internal switch to perform the interconnection between the corresponding input port and the East output port.

The main problem with TDM is precisely that the switching configuration of the router has to be updated for each time-slot. Thus, local configuration memories have to be implemented within routers resulting in high area and energy overhead. As we will see, TDM also imposes tight scheduling constraints on the reservation of circuits.

We propose a solution that implements SVC with *Spatial-*

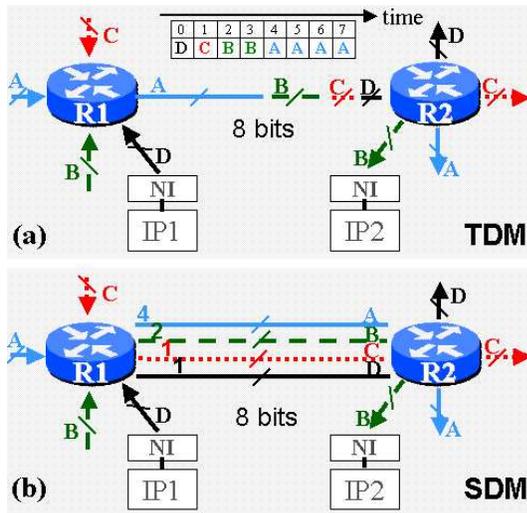


Figure 1: Illustration of the Time Division Multiplexing (TDM) and Spatial Division Multiplexing (SDM) techniques focusing on the link between routers R1 and R2.

Division Multiplexing (SDM). This exploits the fact that on-chip network links are physically made of a set of wires. SDM consists of allocating only a sub-set of the link wires to a given virtual circuit. Messages are *digit-serialized* on a portion of the link (i.e. serialized on a group of wires). The switch configuration is set once and for all at the connection set-up. No inside-router configuration memory is therefore needed and the constraints on the reservation of the circuits are relaxed.

Fig. 1 (b) presents the same configuration as for TDM but implemented with SDM. Four wires are allocated to circuit A, two to circuit B and one wire to circuit C and D. The main difference in this case is that the switch configuration remains the same for the whole circuit lifetime.

The main contribution of this paper is to introduce the SDM technique in the context of NoCs and to propose an architecture for the switch inside the SDM router. This switch is the most critical component of an SDM-based NoC because its size is expected to increase. Indeed, in the extreme case, every input wire from any input port could be connected to any output port wire. Finally, we will validate our technique on a RTL level implementation of the switch with a realistic case study.

The remainder of this paper is structured as follows. Section 2 presents the related work. Section 3 describes the current SVC architectures. Section 4 details our SDM-based implementation focusing particularly on the router. Section 5 presents our experimental setup based on a video application mapped on a realistic NoC platform.

2. RELATED WORK

Related work is divided in NoCs providing only a Best-Effort (BE) service (i.e. no guarantees on latency and throughput) and NoCs also providing a Guaranteed Throughput (GT) service (*Æthreal*, *Nostrum*).

A vast majority of NoC proposals rely only on a Best-Effort service. They are generally based on a *packet-based switching technique*: Dally [4], SPIN [6], Xpipes [8], KTH [7].

The traditional packet-switching technique consists of splitting messages that have to be sent over the network into small independently routed pieces of information called packets. Each packet is

composed of a header containing the control information needed by the routing function and a payload containing the effective data. As no full path pre-establishment overhead is required, packet-switching techniques are well adapted for infrequent short messages but not for long and frequent point to point messages such as encountered in multimedia applications.

Some NoCs also provide a service that ensure predictable and guaranteed communication architecture performances.

Philips was the first to propose a complete solution for a guaranteed throughput (GT) service in addition to a packet-based best effort (BE) service in their *Æthreal* NoC [15]. The GT service guarantees uncorrupted, lossless and ordered data transfer and both latency and throughput over a finite time interval.

The GT service was originally implemented with TDM Switched Virtual Circuit (SVC). During the circuit establishment, time slots are reserved in the output reservation table of each router along the path. The unused time slots can be allocated to the BE traffic [14]. The SVC technique is particularly well adapted for long and frequent messages like multimedia data streams.

However, Philips recently removed the reservation tables from the routers because of their huge area overhead (50%) [5]. They now propose a GT service based on a packet switched technique where resources are reserved by a global scheduler inside the network interfaces. With this technique, the configuration of all routers along the path has to be sent in every packet header. It thus results in some bandwidth waste: in the worst case, one 32-bit header is sent for a 96-bit payload (25% waste). Moreover, each network interface has to centralize all the routing and scheduling information relative to the circuits it has established and it thus become much more complex and hardly scalable. The authors themselves admit that their solution is temporary sufficient for next generation NoCs but not scalable on the long term [5].

The KTH has also proposed a guaranteed bandwidth and latency service in addition to their best-effort packet-switched service for their *Nostrum* Mesh Architecture [12]. This GT service is based on virtual circuits implemented on a packet-based network by exploiting an interesting characteristic of their routing policy (temporally disjoint networks). Compared to *Æthreal*'s original design based on SVC, their solution requires less hardware as no routing tables or input/output queues are needed.

Because they all rely on a TDM-based approach, the main drawback of the above techniques is that the scheduling of communication is rather complex and the energy consumption paid for regularly changing the switch configuration is high.

Our approach based on SDM addresses those problems.

3. MOTIVATION FOR AN SDM-BASED SWITCHED VIRTUAL CIRCUIT

In the Switched Virtual Circuit (SVC) technique, an application establishes a virtual circuit from source to destination and uses it exclusively. This circuit is created by a routing probe injected in the network prior to the data transmission. This probe contains control information like the destination address and the bandwidth required. When a path is found, an acknowledgment probe is transmitted back to the source to initiate the data transmission.

In SVC, the routing information is usually stored in a configuration memory within the router .

This section first describes the architecture and operation of the current TDM-based SVC and motivates the need for an alternative solution. Then, a detailed description of our SDM-based alternative is presented.

3.1 TDM-based SVC networks

The main components of a TDM-based SVC network are the network interfaces and the routers.

The TDM network interface is basically composed of two message queues, a serializer/deserializer and a scheduler (Fig. 2 (a)). The output message queue stores the messages coming from the IP. Those messages are then serialized into smaller data units called flits. Flits are then sent over the network. At the other end of the network, the original message is reconstructed from the incoming flits by a deserializer and is buffered in the input message queue before being delivered to the IP. A scheduler controls the emission of data in the time-slot reserved for this particular circuit. An end-to-end flow control is also generally implemented to avoid buffer saturation at the destination.

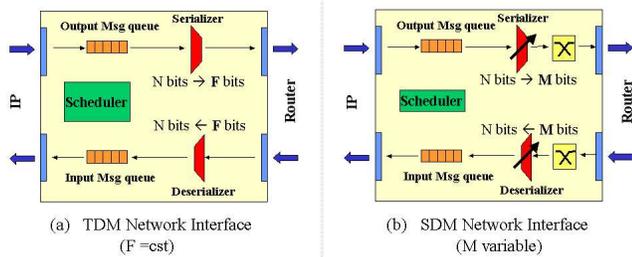


Figure 2: Comparison of the network interface architectures for TDM (a) and SDM (b)

After injecting the message into the network, routers ensure that it arrives at the network interface of the destination IP.

A P-ports TDM router is basically composed of a P×P switch which connects the router input ports to output ports and an Output Reservation Table (ORT) (Fig. 3). The switch is usually implemented with a full crossbar which connects the P n-bit wide input ports to the output ports. The ORT contains the switch configuration for each time-slot based on the decisions performed by the routing algorithm. It is implemented by an SRAM read at each time-slot to set-up the corresponding switch configuration.

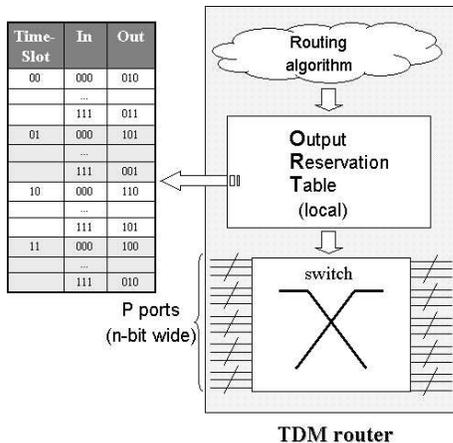


Figure 3: Architecture of a TDM router illustrating the content of the local Output Reservation Table (ORT)

In order to avoid data buffering inside the routers, a constraint is introduced on the time slots allocation. It consists of allocating consecutively time slots for neighboring routers. For example, if time slots T and T+1 have been reserved for a given circuit at router

R1, at the next router R2, the reservation will be made for time slots T+1 and T+2. Any other configuration would require some extra buffering to temporarily store the data until the required time slot.

The consecutiveness of time-slot reservation complicates the case where a circuit reservation is possible. When the network becomes heavily loaded, it may become impossible to make a reservation even though the required bandwidth is actually available. As a result, the routing algorithm can be forced to take a sub-optimal route which will increase circuit latency and energy consumption.

A critical parameter for TDM routers is the *bandwidth allocation granularity*. This parameter represents the ratio between the minimal bandwidth that can be allocated to a circuit and the total link bandwidth. For example, if an audio-stream circuit requires 1 Mbps and the total link bandwidth is 32Mbps, the bandwidth granularity would be 1/32. In TDM, the bandwidth allocation granularity is fixed by the number of individual time-slots that can be allocated. A finer granularity can be obtained at the cost of more time-slots but it also implies bigger ORTs and thus higher energy consumption as this memory is read very frequently, at each time-slot. In our example, using 16 time-slots would result in smaller ORTs but at the cost of a 1Mbps bandwidth waste for the audio-stream.

An important issue in the design of the TDM network is the duration of a time-slot. The larger the time-slot (more network cycles), the larger will be the latency for a message to arrive at its destination. Therefore, the time-slot duration is typically one network clock cycle in order to reduce the end-to-end delay of the circuit.

In conclusion, the TDM implementation suffers from drawbacks resulting from the need to change regularly the switch configuration and the tight constraints on time-slots allocation.

3.2 Spatial Division Multiplexing

The SDM technique consists of allocating a sub-set of the link wires to a given circuit for the whole connection lifetime. This section presents the network interface and router architectures required to implement an SDM-based SVC in NoC.

The SDM network interface is similar to the TDM (Fig.2 (b)). The main differences concern the serialization-deserialization process. In SDM, data is serialized on a number of wires proportional to the bandwidth allocated to the circuit. Therefore, the output bit-width of the SDM serializer has to be parameterizable. A small $(n/m) \times (n/m)$ crossbar is also necessary to select the wires of the n-bit port on which data will be sent.

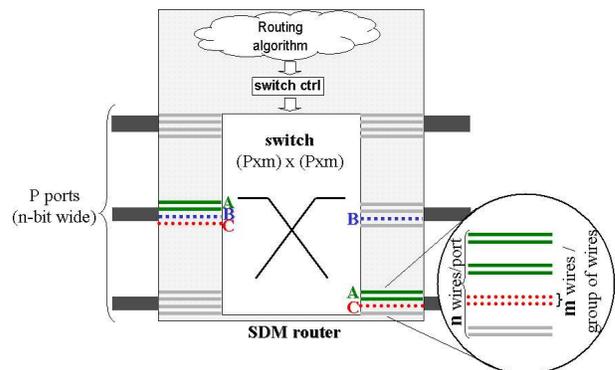


Figure 4: Architecture of a P×P SDM router with 3 virtual circuits: A, B and C.

The SDM router contains a switch and a switch control unit (see Fig. 4). The switch is slightly bigger than in TDM as it must

be able to potentially interconnect any group of wires present at the router input port to another group of wires of any output port. The TDM router offering m time-slots was based on a $P \times P$ n -bit wide crossbar. For SDM, a n -bit port is divided in m individually switchable group of wires. Therefore, for the same bandwidth and number of segments, at the same clock frequency, the number of input and output ports of the switch is increased by a factor m for SDM. However, the ports bit-width is divided by a factor m . The SDM router would thus require a $(P \times m) \times (P \times m)$ n/m -bit wide crossbar.

In contrast to TDM, no particular constraint exists for bandwidth allocation: any available group of wires is suitable. As a result, a shorter connection set-up time and a smaller energy consumption for the routing algorithm are possible. Ultimately, this also leads to finding a shorter path from source to destination.

Another advantage of SDM is that the output reservation table has to be read only once at the circuit establishment as opposed to every time-slot for TDM. As a consequence, it is not necessary to include the ORT inside the router and area can be saved.

Bandwidth allocation granularity is also a critical parameter for SDM routers. In SDM, a finer granularity implies either more wires per link or a bandwidth allocation unit corresponding to less wires. In both cases, it will increase the size of the switch required inside the router, resulting in higher energy consumption. A detailed study of the SDM router energy consumption in function of the bandwidth allocation granularity will be performed in section 5.

In the extreme case of a unitary granularity i.e. when a circuit can be assigned to only one wire, the router must be able to connect any individual input wire to any output wire. For a 5×5 router with 32 bits-wide links, this would result in a 160×160 switch. To evaluate the size of a switch, a common measure is the required number of cross-points. A cross-point is a small switching element that makes or breaks the connection between one input and one output of the switch. For a $N \times N$ crossbar, the number of cross-points required evolves in $O(N^2)$. Given the number of wires that have to be interconnected in SDM, its area and energy overhead would become unaffordable.

A critical issue is thus that the switch inside the router is bigger than for TDM. The next section explains how to efficiently tackle this problem.

4. DESIGN ISSUES IN BUILDING A SWITCH FOR THE SDM ROUTER

Full-crossbars have a too high complexity to be used as an SDM router's switch. An interesting alternative to crossbars consists of using Multiple stages Interconnection Network (MIN) switches. Those can reduce the cross-points cost down to $O(N \log_2 N)$. The cost of using such a switch is paid either in bandwidth (longer clock cycles) or in delay (pipelined stages, multiple cycles to go through).

A wide variety of MIN switches have been proposed in the literature [10] [3]. As the number of cross-points in MIN switches is reduced, some input-output connections cannot be realized anymore as one cross-point can be simultaneously required by two connections, resulting in a *blocking state* (e.g. in Fig. 5 (b) left-side: circuits of the input ports 1 and 2 cannot reach the requested output port, respectively 2 and 1). Table 1 presents a classification of the MIN switches depending on how easy those blocking states can be avoided. In Strictly Non Blocking (SNB) switches, any new connection from a free input to a free output can always be realized. The same condition applies to Non Blocking (NB) switches but with the restriction of carefully choosing the path taken in the switch. In Rearrangeable Non Blocking (RNB) switches, in cer-

tain situations an internal switch re-routing might be necessary to find a non-blocking solution but a solution always exist. Finally, for blocking switches, some connections can be blocked by others without any alternative solution.

Type	Cost	Example
Strictly Non Blocking (SNB)	$O(N^{1.5})$	Clos
Non Blocking (NB)	$O(N \log^2 N)$	Batcher-Banyan
Rearrangeable Non Blocking (RNB)	$O(2N \log N)$	Beneš
Blocking switches	$O(N \log N)$	Banyan

Table 1: Classification of $N \times N$ MIN switches

Our design space is limited to non blocking switches as blocking switches would result on a big loss of flexibility on the bandwidth allocation when the network is heavily loaded. Among the different implementation possibilities, SNB switches are attractive but their minimum cross-point cost is still big ($O(N^{1.5})$) which would lead to an area overhead comparable to the crossbar's.

To reduce the switch overhead to a minimum, we chose a RNB Beneš switch for our first implementation. The Beneš switch has a cost limited to $O(2N \log N)$ [1]. The Beneš switch is built recursively as shown on Fig. 5 (a). At the top hierarchy level, the $N \times N$ switch is composed of three stages. The first and the last stages consist of $N/2$ 2×2 switches. The intermediate stage is itself composed of two $N/2 \times N/2$ Beneš switch. The building process goes on until $N=4$. The $N \times N$ Beneš switch is thus composed of $2 \log_2(N) - 1$ stages of $N/2$ atomic switches. An atomic switch is a 2×2 m -bit-wide switch that can either forward the 2 input data to the output in the same order or invert them (see Fig. 6 (a)). The structure of an atomic switch is presented on Fig. 6 (b): it is simply composed of two $2m$ -to- m bits multiplexors (m being the segment bit-width) and a 1-bit latch to store the switch state (inversion or not). These switches are thus very small and fast.

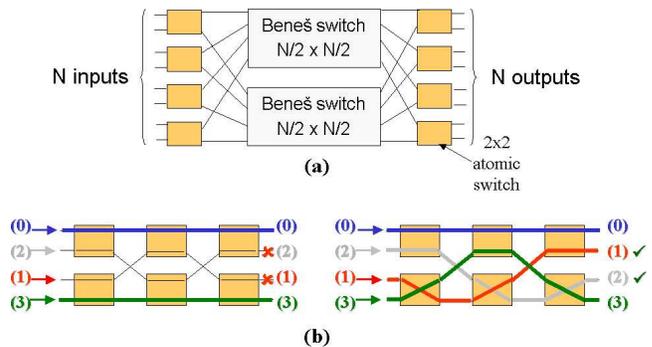


Figure 5: (a) General recursive Beneš switch construction (b) A 4×4 switch instance illustrating a blocking state

Another advantage of MIN switches over crossbars is that only the really required part of the switch is activated for an input-output connection, thus saving energy. The critical path is also almost constant for every input-output couple as the number of activated atomic switches remains always the same for every possible connection and only the interconnect length varies.

The control of a MIN switch is more complex than the control of a crossbar for which only the input-output ports couple is needed to univocally determine which cross-point has to be activated. Fig. 5 (b) shows on the left the situation in which a Beneš switch can block. Circuits at the input 0 and 3 set the atomic switches in such a configuration that the other connections at the input 1 and

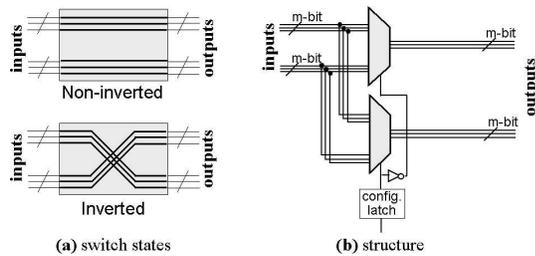


Figure 6: (a) The two states of an atomic switch: inverted or non-inverted inputs (b) Structure of a Beneš atomic switch

2 cannot reach their requested outputs (respectively 2 and 1). The non-blocking configuration is presented on the right of Fig. 5 (b).

A small routing algorithm is thus needed to find a path from input to output port inside the switch and determine which atomic switches to activate. Opferman and Wu have proposed a looping algorithm that avoids any contention in the switch [13]. This recursive algorithm has a better than linear computational complexity in $O((\log_2 N)^2)$. The Beneš switch thus needs a dedicated switch control unit that allows to solve any potential contention within the router, reserve the route within the switch and control the corresponding atomic switches.

Choosing a RNB switch comes at the price of a potential internal switch re-routing. However, if our Beneš switch is not pipelined, it is possible to update the internal switch configuration within the same clock cycle, transparently for the already established connections. In the case of a pipelined switch, the re-routing is a bit more problematic as the switch has to be flushed and some extra-buffering is required.

5. EXPERIMENTAL RESULTS

This section presents our first experimental results concerning the SDM router architectures as the most critical architectural differences between SDM and TDM appear inside those components. Network interfaces are still on-going research but their architectures are very similar for both techniques at the exception of the serializer/deserializer which should be parameterizable in the SDM case. Preliminary results indicate that the impact of SDM NI is very low, but these will be presented in future work.

All delay, energy consumption and area estimations have been performed after synthesis with Synopsys Physical Compiler for the 130nm UMC standard cells technology in average conditions (1.2V, 25C). The energy consumption is obtained with Power Compiler by performing a switching activity annotation of the design during a post lay-out gate-level simulation performed with Mentor Graphics Modelsim.

This section is divided into two parts. The first part evaluates the impact of the choice of granularity on our SDM router for a synthetic workload. The second part presents a proof of our concept based on a detailed comparison of SDM and TDM techniques for a video case study.

5.1 Impact of granularity on SDM router

In this experiment, the energy consumption and area of an SDM router is evaluated for different bandwidth granularities.

We have chosen a synthetic workload corresponding to random traffic and a unitary activity of all the router ports. The router is clocked at 20 MHz, offering a bandwidth of 640 Mbps per port.

Fig.7 describes the evolution of the power consumption and of the area overhead for different choices of granularity for a 32 bit-

wide port. It appears that both power consumption and area are logarithmic functions of the number of circuits per port.

The maximal power consumption is reached for 32 segments per port (unitary granularity) with 1.79mW for an area of $0.135mm^2$.

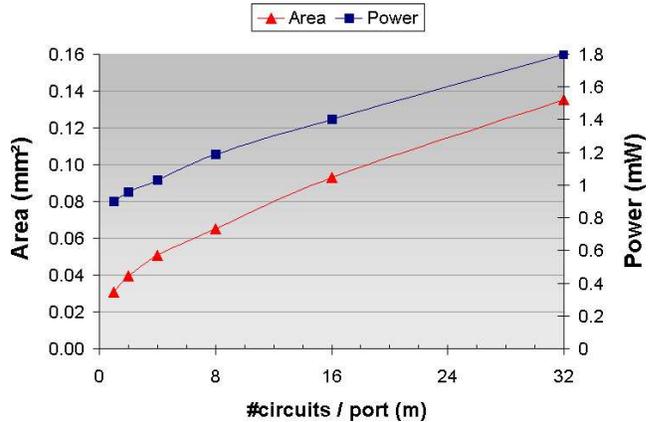


Figure 7: Evolution of the SDM switch area and maximal power consumption in function of the number of circuits that can be allocated per port

5.2 Case-study : an MPEG2 video pipeline

To evaluate the performance of SDM with a realistic workload and to compare SDM and TDM in a realistic case, we have chosen a workload extracted from a digital video processing chain. It is a representative driver application to illustrate the characteristics of the two multiplexing techniques as many NoCs will be part of a multimedia system. Our comparison is in no way restricted to only this particular case study and setup, but it gives a concrete setting to produce absolute values on power and area.

The video chain consists of a camera interface (CAM), an MPEG2 encoder and decoder (ENC and DEC), an intermediate buffer (BUF) and a display interface (DISP) (Fig. 8).

Each communication link involves different bandwidth and routing requirements. The camera produces a stream of 30 raw frames per second (4-CIF format: 704x576) which are transferred to the MPEG2 encoder. The recent history of the encoded video (a few seconds) is placed in an intermediate buffer, allowing the user to quickly play back a recent scene. The video is then read directly from this on-chip buffer and sent to the display.

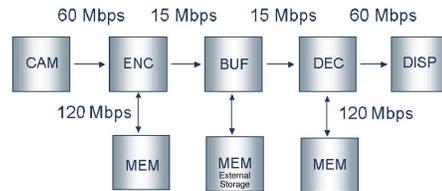


Figure 8: Video chain, with indication of bandwidth requirements

The logical view of our platform (Fig.9) shows the mapping of the video application on 4x4 mesh-based NoC. In this paper, only the particular case of the most activated router R6 will be presented.

For the sake of a fair relative comparison, we have designed RTL-level VHDL models for both a TDM and an SDM implementation of the router.

Our video application bandwidth requirement ranges from 15 Mbps for the compressed video stream between the encoder and the decoder to 120 Mbps for the communication between the processing nodes and their working memories. Both TDM and SDM router are assumed to have 8-bit ports and their clock frequency is set to 15MHz bandwidth to satisfy the top bandwidth requirement of our video application.

A bandwidth allocation granularity of 8 bandwidth allocation units (i.e. time-slots or groups of wires) per link would be optimal as it is the exact ratio between maximal and minimal circuit bandwidth requirements.

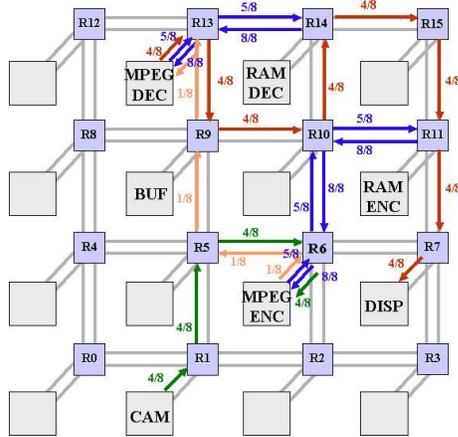


Figure 9: Mapping of the video application on our NoC platform (logical view). with indication of the number of time-slots/groups of wires allocated to each circuit

The TDM router implementation is based on a 8x8 8-bit-ports crossbar. This switch is controlled by an output reservation table implemented by a dual-port 256 bits SRAM (8 time-slots).

The delay and the energy consumption and area breakdown for the TDM router is presented in Table 2. As can be seen, the ORT contributes to a significant part of the overall router power consumption and area overhead (respectively 23.5% and 53%).

	TDM	SDM
Power consumption (μW)	325	301
Output Reservation Table	77	~ 0
Switch and other components	248	301
Area (mm^2)	29433	22410
Output Reservation Table	15536	-
Switch and other components	13897	22410
Critical Path (ns)	0.32	0.44

Table 2: Power, area and delay estimations for router R6 implemented with TDM and SDM (post-layout)

The 8x8 SDM router contains a 64x64 Beneš switch. Each wire of a port can carry a circuit and thus, can be switched independently. The power, area and delay breakdown of router R6 implemented with SDM is presented in Table 2. The contribution of the ORT is almost negligible as it is only accessed once, at the circuit set-up time.

The SDM technique allows a gain of 8% on energy consumption and 31% in area overhead. This comes at the cost of a larger critical path delay (+ 37%).

The energy consumption of the SDM router could be considerably

improved if proper encoding techniques would be used. Serializing data over the links is indeed dramatically affecting the network traffic pattern and the energy consumption savings due to correlations between bits of consecutive flits will thus be lost. However, this can be efficiently avoided by using coding techniques such as SILENT developed by Kaist [11]. This technique allows up to 50% reduction in power consumption for multimedia data traffic.

As can be seen on Table 2, the SDM increases the size of the switch resulting in a higher power consumption for this component. The TDM suffers from the energy cost of its large frequently accessed ORT memory. The energy savings of SDM thus result of a trade-off between those two effects. As a designer, the most efficient multiplexing technique should be selected after a proper application characterization, especially evaluating the required bandwidth allocation granularity which is the most critical parameter.

6. CONCLUSION

In this paper we have compared two approaches for circuit-switched NoCs. We show that, for our case study, the SDM technique performs better in terms of area overhead and energy consumption than the traditional TDM technique. SDM thus appears as a very valuable alternative to TDM that is worth to be explored in more depth as well as a combination with TDM in a hybrid scheme. More work is ongoing on evaluating the switch control overhead and the cost of the network interface.

7. ACKNOWLEDGMENTS

This work is partially supported by the FRIA (Fonds pour la Formation à la Recherche dans l'Industrie et dans l'Agriculture).

8. REFERENCES

- [1] V. E. Benes, On rearrangeable three-stage connecting networks, *The Bell System Technical Journal*, 41, 5, 1962.
- [2] L. Benini and G. D. Micheli, Networks on chips: A new SoC paradigm, *Computer*, 35(1):70–78, Jan. 2002.
- [3] C. Clos. A study of nonblocking switching networks, *Bell Syst. Tech. J.*, 32:406–424, 1953.
- [4] W. Dally. Route packets, not wires: On-Chip interconnection networks, In *Proceedings of DAC-2001*, p 684–689, New York, Jun. 2001, ACM Press.
- [5] J. Dielissen, A. Rădulescu, K. Goossens, and E. Rijpkema, Concepts and implementation of the Philips Network-on-Chip, In *IP-Based SOC Design*, Nov. 2003.
- [6] A. Greiner and P. Guerrier, A generic architecture for on-chip packet-switched interconnections, *Proc. Design Automation and Test in Europe*, Feb. 2000.
- [7] A. Hemani, A. Jantsch, S. Kumar, A. Postula, J. Oberg, M. Millberg, and D. Lindqvist, Network on Chip: An architecture for billion transistor era, 2000.
- [8] A. Jalabert, L. Benini, S. Murali, and G. D. Micheli, xPipesComiler: a tool for instantiating application-specific NoCs, *Proceedings of DATE'04*, Feb 2004.
- [9] A. Jantsch and H. Tenhunen, Networks on Chip, *Kluwer Academic Publishers*, Feb 2003.
- [10] L. N. Jose Duato, Sudhakar Yalamanchili, Interconnection networks, an engineering approach, *IEEE Computer Society Press*, 1998.
- [11] K. Lee, SILENT : Serialized low energy transmission coding for on-chip interconnection networks, *IEEE International Conference on Computer Aided Design (ICCAD) 2004*, p 448–451, Nov. 2004.
- [12] M. Millberg, E. Nilsson, R. Thid, and A. Jantsch, Guaranteed bandwidth using looped containers in temporally disjoint networks within the nostrum network on chip, *Proceedings of DAC 2004*, p 890–895, 2004.
- [13] D. C. Opferman and N. T. Tsao-Wu, On a class of rearrangeable switching networks; part I: Control algorithms; part II: Enumeration studies and fault diagnosis, *Bell System Technical Journal*, 50(5):1579–1618, May-Jun. 1971.
- [14] E. Rijpkema, K. Goossens, J. D. A. Rădulescu, J. van Meerbergen, P. Wielage, and E. Waterlander, Trade offs in the design of a router with both guaranteed and best-effort services for networks on chip, *IEE Proceedings: Computers and Digital Technique*, 150(5):294–302, Sep. 2003.
- [15] E. Rijpkema, K. Goossens, and P. Wielage, A router architecture for networks on silicon, In *Proceedings of Progress 2001, 2nd Workshop on Embedded Systems*, Veldhoven, the Netherlands, Oct. 2001.