NoC Design Flow for TDMA and QoS Management in a GALS Context

Samuel Evain,[1] Jean-Philippe Diguet,[1] and Dominique Houzet[2]

[1] *LESTER, UBS/CNRS, Centre de recherché, BP 92116, Lorient Cedex 56321, France*
[2] *IETR, INSA/CNRS, 20 Avenue des Buttes de Coësmes, Rennes Cedex 35043, France*

This paper proposes a new approach dealing with the tedious problem of NoC guaranteed traffics according to GALS constraints impelled by the upcoming large System-on-Chips with multiclock domains. Our solution has been designed to adjust a trade-off between synchronous and clockless asynchronous techniques. By means of smart interfaces between synchronous sub-NoCs, Quality-of-Service (QoS) for guaranteed traffic is assured over the entire chip despite clock heterogeneity. This methodology can be easily integrated in the usual NoC design flow as an extension to traditional NoC synchronous design flows. We present real implementation obtained with our tool for a 4G telecom scheme.

1. INTRODUCTION

Network-on-Chip (NoC) provides designers with a systematic, flexible, and scalable framework to manage communication between a large set of intellectual property (IP) blocs [1]. It can also reduce IP connection wires and optimize their usage. The dynamic reconfigurability of communication paths responds to the fluctuating processing needs of embedded systems [2]. To comply with synchronous design rules, the traditional NoC implementation is based on a global system clock distributed with a limited skew. This design assumption is inconsistent with large future SoCs where multicycle communication paths and varying delays [3] are unavoidable. Thus, NoC designers must consider these issues in order to allow for manageable and controllable communication on such chips.

Another shortcoming of traditional NoC implementation is its unadaptability to multiclock domain SoCs to which the large SoCs are converging. Different reasons can cause such situations, for instance (1) voltage/frequency dynamic scaling embedded SoCs that optimize power consumption with variable computing and communication loads, (2) reconfigurable and/or adaptive architecture chips, (3) chips with right access areas for security purpose [4]. Finally, multicore chips or multiboard systems intrinsically mean disjoint clock domains. Those technologically inherited conditions complicate and induce challenges to the task of designing NoCs for large chips where the need for flexible, scalable, manageable, and controllable communications is increasing.

We discuss in the state-of-the-art section some different, relevant NoC solutions to cope with these issues. The rest is divided into 4 parts. In Section 3, we ask the question of NoC guaranteed traffic (GT) with different time domains and describe how this new technique is inserted in our NoC synthesis framework. In Section 4, we detail our architecture solutions. In Section 5, we present a case study for different possible configurations.

Finally, we conclude with perspectives based on this work.

2. STATE OF THE ART

NoC can be divided into different categories, regarding clock management, flexibility, and communication QoS. Three kinds of clock schemes can be proposed: synchronous SoC, GALS with heterogeneous clocks, and fully asynchronous SoC (clockless). Our main observations are related to the tradeoff between guaranteed traffic and heterogeneous clock management.

Traditional approaches have always considered synchronous NoC as having a single clock domain. The concept of virtual channel (VC) [5] has been introduced to

bring Quality-of-Service. In QNoC [6] for instance, it offers four traffic priority ranks. Guaranteed traffic is not flexible since it is based on traffic traces and paths that are computed according to simulation. The question of real latency and bandwidth has been solved by Ætheral [7] and Nostrum [8] with the use of time division multiplexing access (TDMA) technique over pipelined circuits. This scheduling prevents any conflicts between packets while network travelling. This scheduling means that all network elements are synchronous. We know that such an assumption is not compliant with wide chip or NoC distributed over several chips (e.g., ASIC and FPGA). Usual techniques implemented to solve skew of multicycle paths issues are based on pipelining [9]. These post-routing decisions require design revision to resolve incoherency of path time slot allocations; this is a very time-consuming process. Furthermore, a single global TDMA table may lead to oversized time slot table size and consequently impact latency, design complexity, and buffer depth growth.

The second approach consists in using asynchronous communications with synchronous components (GALS). In [10], an asynchronous cross-bar is presented but real-time constraints are not considered. An industrial solution is provided by Arteris [11] where intercluster communications are implemented as asynchronous packet exchanges across a network. The intercluster media are a network that transports packets without guaranty; however, packets can be tagged with priorities. The main advantages of this solution are reliability and low-cost switches. In an ANOC [12], a GALS solution is also presented, it is based on the use of hand/check protocols at flit level. Authors use two virtual channels to differentiate high and low priority communications. However, traffic can be really guaranteed only with additional constraints. A first solution consists in using spatially distinct GT paths. A second solution is based on simulations when unambiguous static and deterministic scheduling schemes can be extracted from the target applications.

The third approach is the ultimate stage of the GALS concept; it is based on clockless asynchronous communications. This solution is implemented in [12, 13]. In [12], the authors use virtual channels and low latency routers, however, the NoC architecture does not offer latency and bandwidth guaranties. This shortcoming is solved in [13] with the ALG scheduling discipline. These guaranties are obtained under the condition that the maximum flit transmission delay is known and that the interflit intervals are bounded. This approach is promising but it seems that the number of VC has a strong impact on the cost of the router implementation. It means that the implementation of several guaranteed traffics may lead to a very costly solution. Actually, there is probably a tradeoff between a synchronous TDMA-based solution and an asynchronous one depending on the number of GT communications. The main advantages of clockless solutions are delay insensitiveness, dynamic power optimization, and communication speed since communication can run as fast as possible according to local link characteristics. However, such solutions require a specific implementation

usually based on a four-phase dual rails protocol where each bit is coded with two wires. This is the case in [12] and in [13] where the implementation examples are given for a single router. The drawback of asynchronous techniques is the increase of gate area and interconnect wires, as static power is directly related to area, it may be a serious issue with technologies lower than 90 nm.

Another point is the question of flexibility; it means that the NoC features (e.g., paths, time slot allocation) may be programmable in order to adapt the characteristics of NoC links to the variations of CPU load, and communication variations as depicted in [2]. Source routing appears as the appropriate solution to cope with this issue, it has been implemented with the different approaches previously described with synchronous TDMA [7], asynchronous [12]. Our approach also provides flexibility compliant with time division (TDMA), path allocation but also with sub-NoC composition.

Our solution is GALS-oriented since it enables communication between locally synchronous sub-NoCs with heterogeneous clock domains which means that no global clock is required. The main motivation of our work is to find solution to really guaranty traffics (latency and throughput) within an NoC with heterogeneous clocks. TDMA techniques are interesting, firstly, for bandwidth allocation and latency control, secondly, for FIFO sizing. Actually, TDMA technique organizes communication in such a way that conflicts are avoided and consequently, routers require minimum FIFO resources. On the network interface (NI) side, FIFO requirements strongly depend on TDMA size and path lengths. In practice, very short TDMA tables are required and a 3D (x, y, t) path search space increases chances of finding the shortest paths. However, usual TDMA techniques require a global clock which is not acceptable within future large chips. We also notice that asynchronous techniques require complex solutions [13] or imply NoC underused with constraints such as paths nonoverlapping [12]. We do not use asynchronous techniques [12, 13] but dual clock FIFOs for interfacing different clock domains. There are different reasons for this. First, the advantage in terms of power consumption does not appear to be that decisive, since the dynamic power reduction is balanced by the static power increase. Moreover, synchronous solutions are sufficient and cheap enough for medium-size designs that can also benefit from available power management techniques based on dynamic VDD/Clock/Bias [14]. Finally, we only need protocol wrappers with standards such as OPB or AHB to reuse usual synchronous IP, asynchronous techniques need an additional wrapper at the physical level for asynchronous/synchronous interfacing.

Table 1 synthesizes the pros and cons of these different approaches. Thus, our work is focused on TDMA techniques, we introduced time routing and TDMA synchronisers in order to extend the network interface concept to sub-NoC interfaces. The other advantage of the sub-NoCs approach is related to programmability, basically paths within each sub-NoC can be coded independently from other sub-NoCs.

TABLE 1: GALS techniques pros and cons.

	Skew	GT	Area	Power
GALS + dual clock FIFOs	+	− (Paths must not overlap)	+	+
GALS + clockless com.	+	− (Paths must not overlap)	− (Asynchronous logic cost)	+ Dynamic − static
Synchonous NoC	− (Delay insertion)	+ Scheduling	+	+ Dynamic − static
GALS + TDMA synchroniser	+ Only on identified links	+ TDMA scheduling	+ (TDMA enables accurate FIFO sizing)	− Dynamic (Clock gating) + static

3. PROBLEM FORMULATION AND SOLUTIONS

3.1. μSpider NoC

Our NoC uses a wormhole packet switching technique to carry messages. Routers and network interfaces (NI) are connected by two unidirectional opposite links. Credit-based flow control is implemented for link-level flow control. A packet is a set of FLITs (flow control unit). Communication flow control between routers is based on flit credits. A flit is an elementary packet on which link flow control operations are performed. The width, in bits, of the communication link is a phit (physical unit). A flit is measured in phits.

We use the source routing technique to route packets. Instructions are in the packet header and are proceeded by crossed routers to determine the right output port.

NIs connect IP blocks to the NoC in a way which is quite comparable to the Ætheral approach [15]. Virtual channels are used to carry best effort (BE) and guaranteed traffic. To prevent contention between the multiple GTs, TDMA techniques are used for time slot allocation in NIs [16].

Moreover, our NoC is customizable through an associated CAD tool [17]. Our CAD tool is a decision and synthesis tool to help designers obtain the ad hoc NoC depending on the application and implementation constraints. It is able to configure the various functionalities of our NoC, like topology, routing technique, and so forth. Finally, this tool generates an optimized dedicated NoC VHDL code at RTL level.
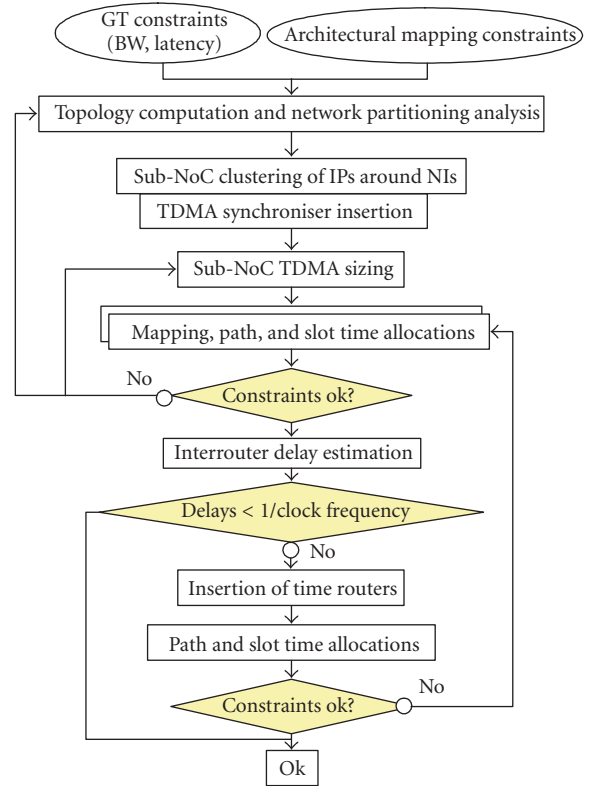
In this paper, we focus on its ability to manage GT on a global network composed of various clock domain sub-NoCs.

3.2. Hierarchical NoC and μSpider flow

Today's chips may have multiple-clocks with different frequencies. Each clock domain may have IPs that need to communicate through an NoC. An NoC in certain clock domains may suffer TDMA coherency problems due to variable multiclock delay links. Even if such a problem does not exist, we may be interested in having more than one NoC to optimize TDMA slot table.

We propose a hierarchical NoC structure. A sub-NoC is a synchronous network area having a single clock frequency and obeying its own TDMA table.

The interconnection of those heterogeneous (different frequencies and TDMA rules) sub-NoCs forms a global NoC.

FIGURE 1: μSpider NoC design flow.

The causes of its network division may be the following.

(i) Multicore chips separating the networks in several areas, causing long distance wires between them. No classic solution can be used (relay station [9] or link pipelining) because those solutions cannot take place between cores.
(ii) Architectural mapping constraints.
(iii) Reuse of previously designed NoCs implemented as IPs.
(iv) Clock dynamic management for power optimization.
(v) Areas with different security levels.

Figure 1 shows the μSpider design flow modified to take into account different time domains.
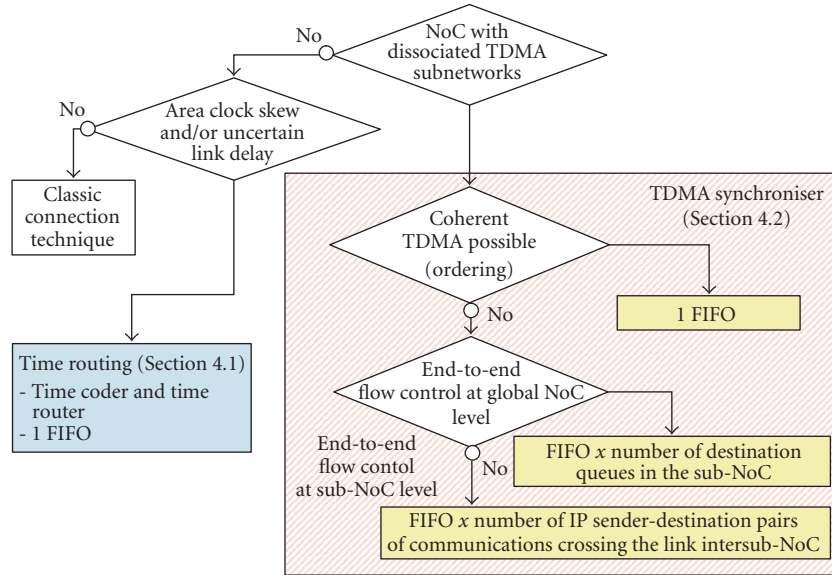
FIGURE 2: Solution tree.

The flow provides two outstanding evolutions from the initial single NoC design flow. First, the designer has the possibility of specifying different sub-NoCs. Sub-NoC management is supported by synchronization techniques explained in Section 4. The decisions of sub-NoC allocations are performed together to avoid local optimal subsolutions but not optimal global solutions. Then, in the design flow, if delays which are potentially larger than one cycle are noticed, it means that synchronous design assumptions are not valid. In that case, time routers (see Section 4) are inserted and path/slot time allocations are recomputed with initial mapping constraints.

3.3. Solution tree

Different solutions may be considered regarding the problem to be solved. Figure 2 shows the associate solution tree. The first question deals with identifying uncertain or long delay links, if they belong to the same TDMA domain, then they are solved with the time routing technique described in Section 4.1. Otherwise, they are intrinsically solved with the TDMA synchroniser technique described in Section 4.2. In this case, some additional characteristics lead to different implementations depending on situations.

Case 1. The coherence between the TDMA tables of the sub-NoCs; namely the data are emitted and completely read with the same order.

Case 2. The implementation of the end-to-end flow control can be local to sub-NoCs or global.

4. μSPIDER ARCHITECTURAL SOLUTIONS

Time routing is a bridge between portions of NoC having the same TDMA but with a possible phase skew.
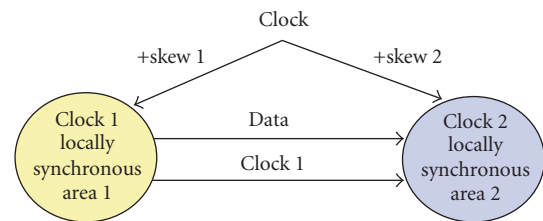

FIGURE 3: Same clock but different clock skews.

TDMA synchroniser is a bridge between sub-NoCs having different TDMA, moreover a phase skew between the sub-NoC clocks is possible.

4.1. Time routing solution

In this first case, we consider router clusters having the same clock but different clock skews and a link with unpredictable but bounded delay.

Figure 3 shows a global clock distributed on both areas. Clock1 and clock2 frequencies are equal but have different skews. Moreover, the data transmission has a delay that is a function of the link's physical characteristics (length, capacity, ...). Skew and delay transmission can cause synchronisation problems in the time division multiplexing over the pipelined path across the considered link.

To be independent from variable skews and data link delays in a given TDMA of a sub-NoC, we have considered that these two variables are bounded with known values. TDMA time slot allocations are computed using the worst case delay and clock skew between those routers. The time coherency between routers is controlled in order to impose this worst case delay for all packets. This control is implemented by integrating the time coder (TC) and the time router (TR) in the sending and the receiving routers, respectively (see Figure 4).
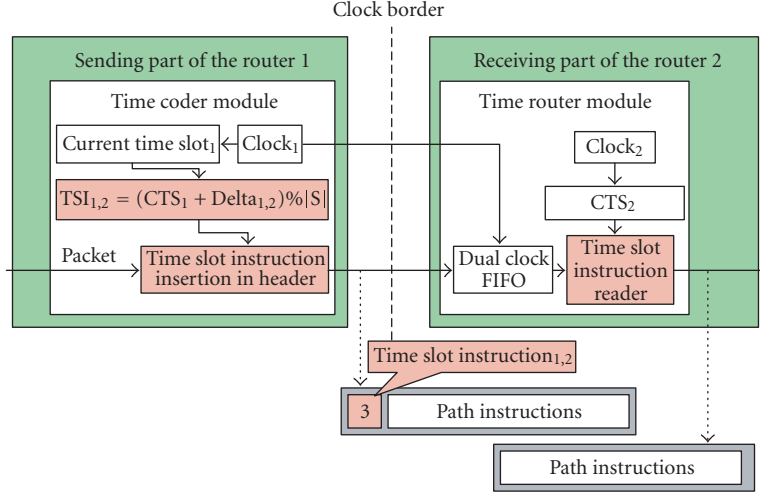
FIGURE 4: Time coder and time router.

The architecture of the dual clock FIFO is not presented here since it is not the main topic of this paper.

The TC adds a time slot instruction in the header path instruction field of passing packets. This time slot instruction indicates to the TR the appropriate time slot number at which it must release this packet. The TR does not use the TDMA table; however, it remains aware of the current time slot with a simple counter. The received packet is stored in an FIFO and is released once the current count time and the time slot instruction are identical.

Figure 4 shows the architecture of the time coder and time router on both sides of an unpredictable delay unidirectional link. Note that the TC and the TR processings are performed in parallel with the usual router computations so no additional delay is introduced.

The time slot instruction can be arbitrary chosen; the maximum required FIFO depth is equal to the number of reserved slots of communications crossing this link during a complete TDMA table rotation. However, we can adjust the time slot instruction to reduce latency.

Hereafter, we show how to determine the maximum slot time difference and the required FIFO depth.

The following parameters are used to formalize the computation.

(i) Slot size is made of Ls phits.
(ii) The time slot table size is $|S|$.
(iii) CTS_i = current time slot in area i. $0 \leq CTS_i < |S|$, for a communication from area x to area y.
(iv) $TSI_{x,y}$ = time slot instruction computed in area x and proceeded in area y. $0 \leq TSI_{x,y} < |S|$.
(v) $SKEWup_{x,y}$ = upper bound clock skew between both x and y area clocks.
(vi) $SKEWlo_{x,y}$ = lower bound clock skew between both x and y area clocks.
(vii) $Tup_{x,y}$ = the necessary number of periods for a flit to cross over a link considering the longest specified boundary delay.

$$\begin{array}{l} \text{If } (Clo_{x,y} < 0), \\ \text{then } FDEPTH = |S| * Ls, \\ \text{else } FDEPTH = Min((Delta_{x,Y} * Ls) - Clo_{x,y}, |S| * Ls) \end{array}$$

ALGORITHM 1

(viii) $Tlo_{x,y}$ = the necessary number of periods for a flit to cross over a link considering the shortest specified boundary delay.
(ix) Hence the maximum cycle delay difference between area x and area y (in clock periods) is

$$Cup_{x,y} = Tup_{x,y} + SKEWup_{x,y}. \tag{1}$$

(x) The minimum cycle delay difference between area x and area y is

$$Clo_{x,y} = Tlo_{x,y} + SKEWlo_{x,y}. \tag{2}$$

(xi) Thus, the maximum slot time difference between area x and area y (in slots) is

$$Delta_{x,y} = \left\lceil \frac{Cup_{x,y}}{Ls} \right\rceil \tag{3}$$

and time slot instruction for router in area y is

$$TSI_{x,y} = (CTS_x + Delta_{x,y})\%|S|. \tag{4}$$

(xii) Finally, FIFO depth is given by Algorithm 1.

The value of $Delta_{x,y}$ could be added in the time router module instead of the time coder module. Moreover, this value may be static and preconfigured in the time router module. However, this configurability ability offers more flexibility in the time domain to find a TDMA solution.
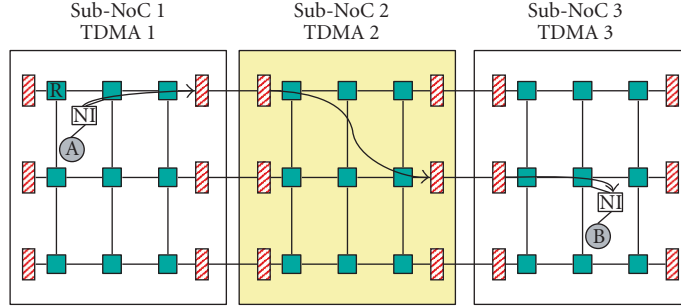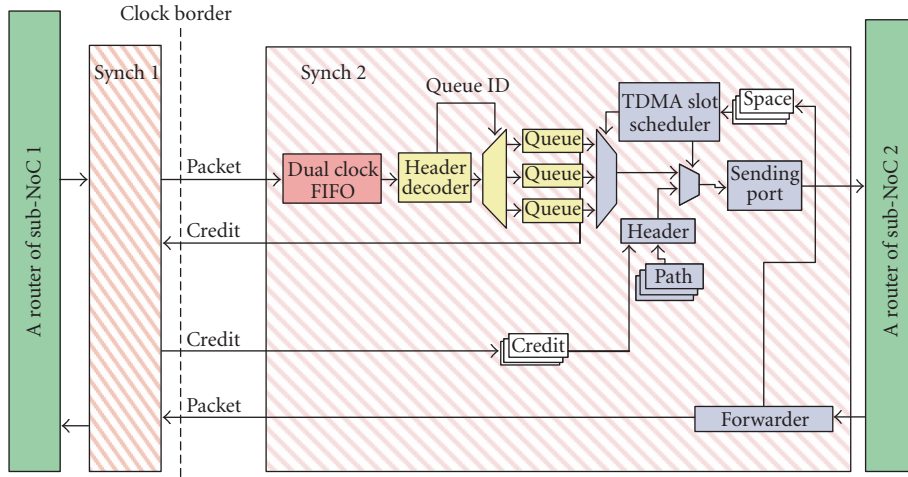
FIGURE 5: NoC made of 3 connected subnetworks.

FIGURE 6: Synchroniser architecture.

4.2. TDMA synchroniser

4.2.1. Concept

In such a case, we consider independent clocks between local sub-NoCs having distinct TDMA tables. To illustrate our concepts, we consider the NoC shown in Figure 5. It is made of three side-to-side joined sub-NoCs. Sub-NoC 2 offers guaranteed traffics to flow through.

A TDMA synchroniser, detailed in Figure 6, is introduced between each pair of communicating sub-NoCs as a bridge between their TDMA slot tables. It is composed of two synchronisers. Each synchroniser is composed of two correlated parts: the first one synchronizes the traffic (traffic synchroniser) while the other one forwards opposite sense traffic (forwarder). Two synchronisers are connected head to tail, that is, the forwarder of synchroniser 1 is connected to the traffic synchroniser of synchroniser 2, and visa versa. The general synchroniser architecture is detailed only for synchroniser 2.

A synchroniser belongs to the sub-NoC to which it sends data, and is seen as a NI to this sub-NoC. As for any classic NI, the synchroniser TDMA table contents are coherent with the sub-NoC TDMA table. Each sub-NoC sees the other one as a classic NI.

A packet leaves a sub-NoC by traversing the forwarder module of the local synchroniser. When it arrives to the remote synchroniser, it is stored in the dual clock FIFO. The header decoder uses the queue ID in the packet header to find the right queue into which it will store this packet. The exact number of FIFOs and synchroniser architecture, in general, depends on problem parameters and designer choices that are discussed in the next paragraph. The TDMA scheduler requests read operations from a certain queue according to its TDMA time slot reservation table. Moreover, the path field in the packet header receives the correct path instructions to cross this sub-NoC depending on the communicated queue ID.

This guaranteed traffic service can be offered if and only if the following property is respected:

$$|S_i| \geq \frac{B_i}{F} * |S| + \frac{N_h * L_h}{L_s}, \tag{5}$$

where

(i) $|S_i|$ is the number of reserved slots for a communication i.

(ii) B_i is the specified payload in phits/s for communication i.

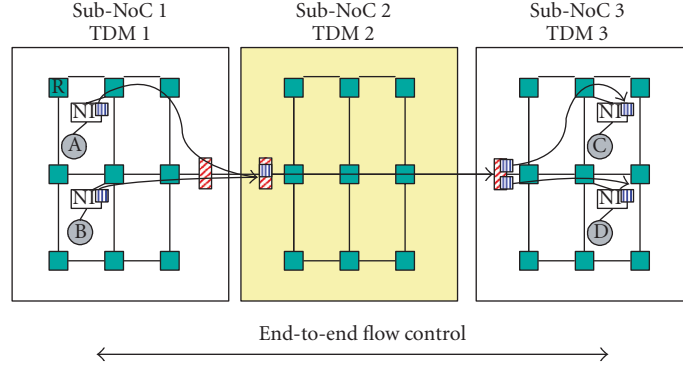(iii) F is the channel link frequency in sub-NoC (Hz).

FIGURE 7: End-to-end flow control at global NoC level.

(iv) N_h is the number of headers during slot table iteration.
(v) L_h is the header size in phit unit.
(vi) L_s is the slot size in phit unit.

4.2.2. End-to-end flow control

To ensure that no overflow can occur in destination queue, we use the end-to-end flow control. At connection setup between a sender and a receiver pair, the full space of the destination queue is allocated to the sender. This queue is called round trip latency hiding FIFO. Then, the sender can only send data to the receiver when it has space credits, credits represent the amount of queue space at receiver. Moreover, the sender decreases this value each time it sends data to this destination. The receiver grants credits to the sender when data have been consumed and so new empty space is available in the destination queue.

End-to-end flow control policy can be introduced at different hierarchical levels, global or local. An end-to-end flow control between NIs across the global NoC is global. Independent end-to-end flow controls in each sub-NoC are local. The choice of this policy depends on a lot of conditions such as reusability and adaptability.

4.2.3. Global end-to-end flow control

In the case of global end-to-end flow control, if the designer selects this option for FIFO optimization reasons, it means that different communication sharing the same resources can be considered as a single communication from the point of view of the crossed sub-NoC as the sub-NoC 2 in the example of Figure 7. In sub-NoC TDMA synchroniser, only one FIFO queue is implemented per destination independently of the original source, moreover, credit and space modules of Figure 4 are removed. End-to-end flow control at global NoC level (across all the NoCs) needs large depth FIFO queue located in the destination NIs. The objective is to hide the round trip latency (of credit at end-to-end flow control level) due to the long distance for which it takes a message to arrive and credits to return. These FIFO queues can be large

in case of long paths. Another pertinent issue is TDMA coherency, basically two sub-NoCs are coherent if packets travelling from one to another are entirely emitted and consumed with the same order. From a sub-NoC point of view, if the designer is able to identify a coherency possibility between two sub-NoCs, then a single FIFO is needed to store data after the removal of queue ID. This implies interesting fee minimization.

4.2.4. Local end-to-end flow control

The other possibility is an end-to-end flow control at sub-NoC level. In a TDMA synchroniser, there is one FIFO queue for each sender-destination pair of communication crossing the link inter sub-NoC. Packets are depacketized when leaving a sub-NoC and repacketized when entering in another sub-NoC.

End-to-end flow control at sub-NoC level means multiple small end-to-end flow controls (Figure 8). Round trips are short but numerous. The FIFO cost for round trip purposes is distributed in the crossed TDMA synchronisers. Moreover, each sub-NoC buffer dimensioning is independent of neighbouring sub-NoCs. So sub-NoCs can be seen as single IPs.

4.2.5. Local versus global end-to-end flow control

The choice between global and local flow controls mainly depends on the nature of constraints. Actually, for a given application, the total FIFO size in the local case is equal or slightly larger than the global case. On the one hand, the local case uses more small distributed FIFOs, so induces a larger control cost (including counters), the other drawback is the decision of FIFO size distribution over the whole set of distributed FIFOs. On the other hand, the local case brings a subdivision of concerns and consequently facilitates design reconfiguration. The drawback is the large number of small FIFOs.

We can extract two extreme cases for which the choice is clear. First, in a case of many different communications
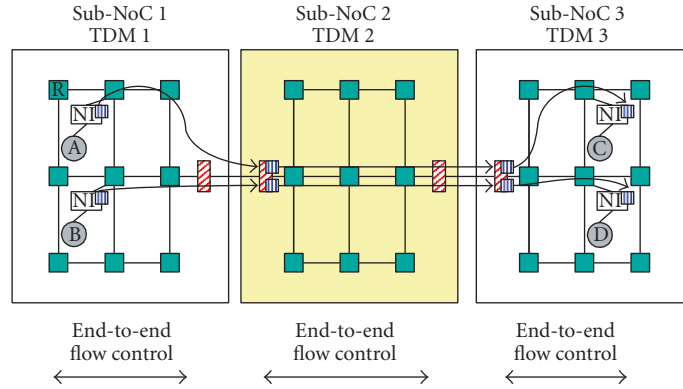
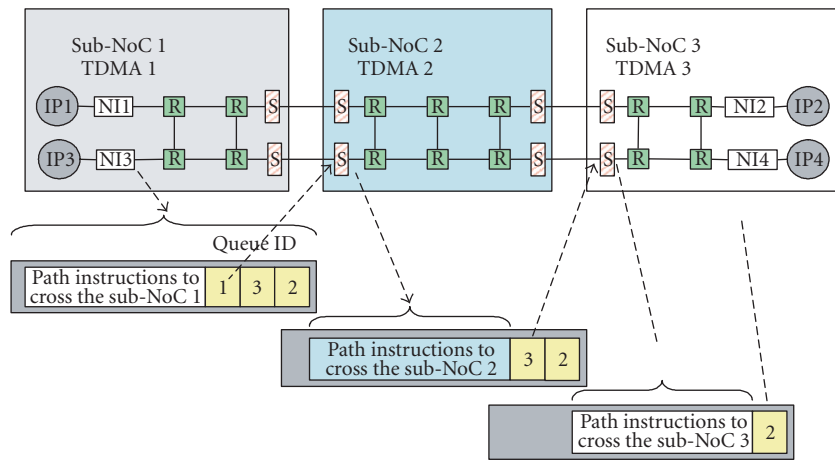FIGURE 8: End-to-end flow control at subnetwork level.

FIGURE 9: Sub-NoC interconnections and header path instructions.

between two sub-NoCs, a global implementation is required. On the contrary, if a very few amount of communications is specified, then a local configuration is more appropriate since the number of FIFO is reasonable. Moreover, the packet resizing overhead is existing only in the global case as explained in Section 3.3.

4.2.6. Header path instructions

Figure 9 shows packet header path instruction fields. The used path in a given sub-NoC is only related to that sub-NoC which means that it can be reconfigured independently from the others. When a packet leaves a sub-NoC, the previously used path instructions are removed. When a packet reaches a synchroniser, its queue ID request is analyzed and its destination is deducted. The sub-NoC automatically inserts the appropriate path to reach this destination inside this sub-NoC.

This path knowledge distribution reduces path instruction lengths and solves the problem of the path size field in the header of packets. With this solution, we keep the main advantages of an NoC, which must be scalable and reconfigurable.

4.3. Packet resizing in case of global end-to-end flow control

4.3.1. Problem formulation

The number of reserved slots (send window) for a same communication may be different in crossed sub-NoCs due to different slot table size and frequency.

This leads to the problem of packet resizing with split and merge operations. Actually, packets may be split or merged according to available send windows. It means a control cost for packet reorganisation.

When a packet must be split in two parts, its header is copied to be the header of the second part. The credit information field is not copied. This leads to an increase of the number of headers, so to a decrease of the available bandwidth. The cost of header insertion is not negligible in case of small packets.

Rebuilt Packet implies the removal of header, after memorizing its credit to be able to add it in the new packet header belonging to an identical transaction. However, packets interleaved with packets belonging to another transaction cannot be pasted together, except if distinct FIFOs are used to

Table 2: Configuration associate rules.

Send window size in sub-NoC$_i$ compared to send window in sub-NoC$_{i+1}$.	Rules
>	– Bandwidth offered in sub-NoC$_{i+1}$ must be sufficiently higher than bandwidth offered in sub-NoC$_i$ to carry additional headers introduced by packet splitting.
= 1	– Packet cannot be split. – Bandwidth offered in sub-NoC$_{i+1}$ must be at least equal to bandwidth offered in sub-NoC$_i$.
≤	– Packet header can be emitted only during the first slot of the send window to be sure it will not be split. – Bandwidth offered in sub-NoC$_{i+1}$ must be at least equal to bandwidth offered in sub-NoC$_i$.

order packets belonging to the same transaction; in usual cases, this approach is too complex.

Packer resizing can introduce bandwidth degradation. In many cases, it may be acceptable; however, simple solutions make avoiding it possible.

4.3.2. Solutions

Consider packet going from sub-NoC$_i$ to sub-NoC$_{i+1}$, in Table 2, we compare send window width between these two neighbouring sub-NoCs and give rules to respect.

A solution to obtain the same number of reserved slots and the same bandwidth for communications in each Sub-NoC consists in conserving the same (frequency/slot table size) ratio. Figure 10 shows an example with two sub-NoCs with different TDMA slot table size, running at different frequencies.

5. CASE STUDIES

5.1. Application context

To point out the different solution costs, we consider a 4G telecommunication application. This is a two-way transceiver implementing MC-CDMA MC-SS-MA baseband communication techniques. The application constraint is 665.6 μs and a frame is composed of 32 symbols. The MC-SS-MA part requires 18 IP ports and the MC-CDMA one needs 32 IP ports. IP ports are grouped into 22 clusters. This application requires 29 unidirectional communications between dedicated hardwares including local memories. We use end-to-end flow control, so communications can be seen as bidirectional.

Required communication bandwidths are very different. Communications close to the antenna in the application graph need bigger bandwidth than decoded data or not yet

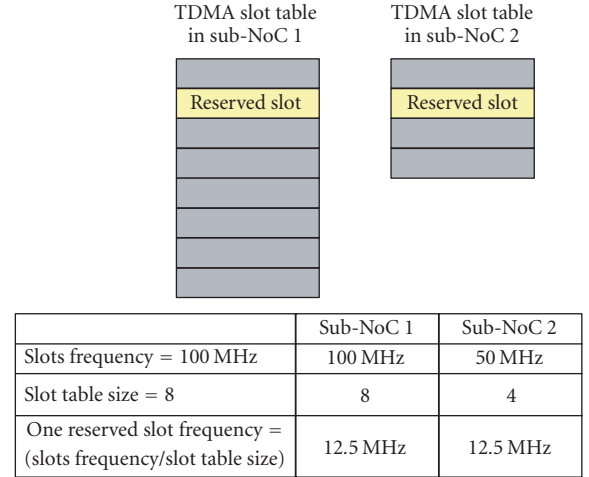|	Sub-NoC 1	Sub-NoC 2
Slots frequency = 100 MHz	100 MHz	50 MHz
Slot table size = 8	8	4
One reserved slot frequency = (slots frequency/slot table size)	12.5 MHz	12.5 MHz

Figure 10: Frequency and slot table size ratio.

coded data. This application is mapped on a topology composed of two parts called areas. Mapping is made to group communications with small bandwidth in left area 1, and large bandwidth communications in right area 2. The topology of our example is shown in Figure 11. Area 1 and area 2 are implemented as one single NoC or two sub-NoCs in the cases described hereafter. Three communications go from area 1 to area 2, they are named interarea communications. Interarea communications represent approximately 10% of the total. This proportion is representative of sub-NoCs composition. Only interarea communications are represented in Figure 11.

In this NoC, a phit is a 32 bit width word; flit size is two phits; a header is one phit.

5.2. Case study descriptions

We have used our tool to find design solutions according to application constraints and usual real-life cases (different clock frequencies and long delay).

Case 1. A single clock NoC without any delay larger than the clock period. It corresponds to a classic NoC case, this is our reference for comparisons.

In the following Cases 2 to 5, we assume a maximum delay between both areas equal to 50 nanoseconds.

Case 2. A single clock NoC with a time routing solution to solve the long delay problem.

For the following three cases, area 1 and area 2 have different clock domains (9 and 100 MHz, resp.). They are implemented as sub-NoC 1 and sub-NoC 2. The sub-NoC 1 slot table size is no longer 6 but 4, all other parameters remain unchanged. Two TDMA synchronizers are introduced on link between the two sub-NoCs. These following three cases correspond to different system conditions.
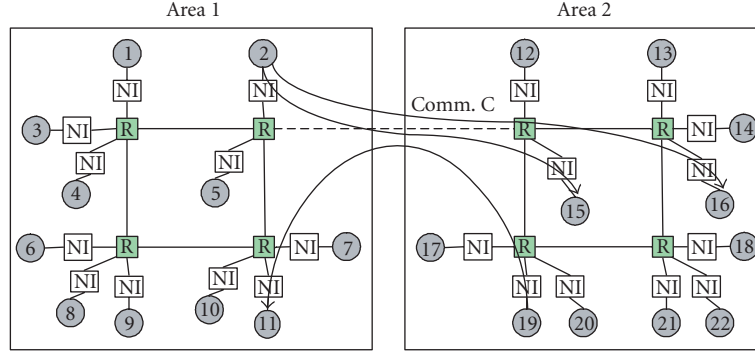
FIGURE 11: Topology and interarea communications.

TABLE 3: Case studies.

Cases	Area 1		Area 2		Comments and solutions
	F MHz	|S|	F MHz	|S|	
1	100	6	100	6	Same TDMA, 1 cycle delay
2	100	6	100	6	Same TDMA, 5 cycles delay Time routing
3	9	4	100	6	Synchronisers with coherent TDMA
4	9	4	100	6	Synchronisers with Global NoC end-to-end flow control
5	9	4	100	6	Synchroniser with Local end-to-end flow control

Case 3. Two sub-NoCs with heterogeneous clock, coherent TDMA, and global end-to-end flow control.

Case 4. Two sub-NoCs with heterogeneous clock, noncoherent TDMA, and global end-to-end flow control.

Case 5. Two sub-NoCs with heterogeneous clock, noncoherent TDMA, and local end-to-end flow control.

Note finally that this application does not have any latency constraints but only bandwidth limitations. Our tool found some solutions that meet the bandwidth constraints for the various schemes without any latency test. However, latency can be easily guaranteed when it is necessary. In practice, the latency and bandwidth are computed and checked simultaneously.

The considered cases are summarized in Table 3.

5.3. Result analysis

Table 4 shows each case relative cost.

In Case 1, the total FIFO depth is due to the sum of decoupling and round trip latency hiding FIFOs in NIs; the global latency for communication C is 200 nanoseconds.

The time delay added in Case 2 increases the latency with 2 slots duration. The increase of FIFO cost is due to the following two reasons.

TABLE 4: Latency and FIFO depth cost.

	Case Studies				
	1	2	3	4	5
Number of FIFO in synchronisers (control cost, FIFO size distribution)	0	2	2	3	3
Sum of FIFO depths in words	323	335	348	348	348
FIFO depths increase for the three inter-area communications	0	12	18	18	18
Latency for communication C (ns)	200	240	1938	1938	1938

(i) The FIFO in the two TRs (12 words).
   Two TRs are added (one for each direction). Each TR has an FIFO. The FIFO depth is constant (6 words), it does not depend on interarea communications.
(ii) The depth of round trip latency FIFOs in NIs depends on interarea communications, they are used to hide the latency of the returned credit for end-to-end flow control (2 words).

In Cases 3 to 5, the reduced frequency in sub-NoC 1 implies an increase of latency (+1578 nanoseconds) for communication travelling in sub-NoC 1. The latency increase is also due to the resynchronization process between TDMAs.

Case 3 benefits from coherent TDMAs into sub-NoC 1 and sub-NoC 2 and thus each synchroniser uses only one FIFO. Case 4 implements incoherent TDMAs with global end-to-end flow control. It needs more FIFOs but the sum of all FIFO depths remains unchanged.

Case 5 is similar to the previous one but uses a local flow control.

To conclude, we observe that the management of fluctuating multicycle delays and/or heterogeneous clock domains has an acceptable cost for the four cases compared to the reference one.

Additionally, our experience shows that the frequency reduction implies an area increase for latency hiding. An accurate study is needed to see if the dynamic power reduction is really interesting compared to the increase of static power due to the area overhead.

6. CONCLUSION AND PERSPECTIVES

In this paper, we have introduced alternative solutions offering guaranteed throughput traffics in the context of NoCs with different clock areas and skew. We have presented two original techniques (time router and TDMA synchroniser) included in a new design approach based on the concept of sub-NoC compositions. This approach meets the design productivity constraints since it is compliant with a classic synchronous single NoC methodology and can be easily inserted in a usual SoC design flow. It has been implemented in our μSpider design tool and applied to a real-life telecom application. Moreover, the solution of sub-NoCs, with independent or disjoint time domains, makes the implementation of local/global NoC manager for power management by means of Vdd/Clock dynamic selection and security monitoring possible. These two points are our current research directions.

REFERENCES

[1] W. J. Dally and B. Towles, "Route packets, not wires: on-chip interconnection networks," in *Proceedings of the 38th Design Automation Conference (DAC '01)*, pp. 684–689, Las Vegas, Nev, USA, June 2001.

[2] T. Marescaux, A. Bartic, D. Verkest, S. Vernalde, and R. Lauwereins, "Interconnection networks enable fine-grain dynamic multi-tasking on FPGAs," in *Proceedings of the 12th International Conference on Field-Programmable Logic and Applications (FPL '02)*, pp. 795–805, Montpellier, France, September 2002.

[3] W. J. Dally, "Interconnect-limited VLSI architecture," in *Proceedings of IEEE International Conference Interconnect Technology*, pp. 15–17, San Francisco, Calif, USA, May 1999.

[4] S. Evain and J.-Ph. Diguet, "From NoC security analysis to design solutions," in *Proceedings of The IEEE Workshop on Signal Processing Systems (SIPS '05)*, Athens, Greece, November 2005.

[5] W. J. Dally, "Virtual-channel flow control," *IEEE Transactions on Parallel and Distributed Systems*, vol. 3, no. 2, pp. 194–205, 1992.

[6] E. Bolotin, A. Morgenshtein, I. Cidon, R. Ginosar, and A. Kolodny, "Automatic hardware-efficient SoC integration by QoS network on chip," in *Proceedings of the 11th IEEE International Conference on Electronics, Circuits and Systems (ICECS '04)*, pp. 479–482, Tel-Aviv, Israel, December 2004.

[7] K. Goossens, J. Dielissen, J. van Meerbergen, et al., "Guaranteeing the quality of services in networks on chip," in *Networks on Chip*, A. Jantsch and H. Tenhunen, Eds., pp. 61–82, Kluwer Academic, Dordrecht, The Netherlands, 2003.

[8] M. Millberg, E. Nilsson, R. Thid, and A. Jantsch, "Guaranteed bandwidth using looped containers in temporally disjoint networks within the Nostrum network on chip," in *Proceedings of Design, Automation and Test in Europe (DATE '04)*, vol. 2, pp. 890–895, Paris, France, February 2004.

[9] L. P. Carloni and A. L. Sangiovanni-Vincentelli, "Coping with latency in SOC design," *IEEE Micro*, vol. 22, no. 5, pp. 24–35, 2002, special issue on systems on chip.

[10] A. Lines, "Nexus: an asynchronous crossbar interconnect for synchronous system-on-chip designs," in *Proceedings of the 11th Symposium on High Performance Interconnects*, pp. 2–9, Stanford, Calif, USA, August 2003.

[11] Arteris, http://www.arteris.net.

[12] E. Beigné, F. Clermidy, P. Vivet, A. Clouard, and M. Renaudin, "An asynchronous NOC architecture providing low latency service and its multi-level design framework," in *Proceedings of the 11th IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC '05)*, pp. 54–63, New York, NY, USA, March 2005.

[13] T. Bjerregaard and J. Sparsø, "A scheduling discipline for latency and bandwidth guarantees in asynchronous network-on-chip," in *Proceedings of the 11th IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC '05)*, pp. 34–43, New York, NY, USA, March 2005.

[14] D. Hillman, "Using mobilize power management IP for dynamic & static power reduction in SoC at 130 nm," in *Proceedings of Design, Automation and Test in Europe (DATE '05)*, vol. 3, pp. 240–246, Munich, Germany, March 2005.

[15] E. Rijpkema, K. Goossens, A. Rădulescu, et al., "Trade offs in the design of a router with both guaranteed and best-effort services for networks on chip," in *Proceedings of Design Automation and Test Conference in Europe (DATE '03)*, pp. 350–355, Munich, Germany, March 2003.

[16] A. Radulescu, J. Dielissen, K. Goossens, E. Rijpkema, and P. Wielage, "An efficient on-chip network interface offering guaranteed services, shared-memory abstraction, and flexible network configuration," in *Proceedings of Design, Automation and Test in Europe (DATE '04)*, vol. 2, pp. 878–883, Paris, France, February 2004.

[17] S. Evain, J.-Ph. Diguet, and D. Houzet, "A generic CAD tool for efficient NoC design," in *Proceedings of International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS '04)*, pp. 728–733, Seoul, Korea, November 2004.

Samuel Evain is an Associate Professor at the UBS University (France) and works at the LESTER Laboratory. His research interests include Network-on-Chip concept and design methodology. He is currently finishing a Ph.D. degree in electronics from the Institut National des Sciences Appliquées (INSA) of Rennes, France.

Jean-Philippe Diguet received the M.S. degree and the Ph.D. degree from Rennes University (France), in 1993 and 1996, respectively. His thesis, within the LASTI laboratory (IRISA/R2D2) addressed the estimation of hardware complexity and algorithmic transformations for high level synthesis. Then he joined the IMEC in Leuven, where he worked as a postdoctoral fellow on memory hierarchy decisions for power optimization. He has been a Member of the LESTER laboratory (Lorient, France) since 1998, where he started research project in design space exploration at both algorithmic and system levels. He has been an Associated Professor at UBS University (France) from 1998 until 2002. In 2003, he initiated a technology transfer and cofounded the Dixip Company in the domain of wireless embedded systems. Since 2004 he has been a CNRS Researcher. His current work focuses firstly on managing the EDA framework project design trotter for design space exploration in the domain of heterogeneous real-time embedded systems. The second topic

is the definition of environmentaware and self-adaptive architectures under QoS and power constraints, it includes RTOS new services, security concerns, NOC, and architecture reconfiguration control.

Dominique Houzet received the M.S. degree in computer sciences, in 1989 from Paul Sabatier University, Toulouse, France, and the Ph.D. degree and HDR degree in computer architecture, in 1992 and 1999, both from INPT, ENSEEIHT, Toulouse, France. He worked at IRIT Laboratory and ENSEEIHT Engineering School from 1992 to 2002 as an Assistant Professor and at IETR Laboratory INSA Engineering School in Rennes from 2002 to 2006 and also as a Digital Design Consultant with SME and large companies. He is now a Professor at LIS-INPG, Grenoble. He has published a number of research papers in the area of parallel computer architecture and SoC design and a book on VHDL principles. His research interests include codesign and SoC design methodologies applied to image processing and radiocommunications. He is a Member of the IEEE Computer Society.