Floorplan-Aware Hierarchical NoC Topology with GALS Interfaces

Debora Matos¹, Cezar Reinbrecht¹, Gianluca Palermo², Jonathan Martinelli¹, Altamiro Susin¹, Cristina Silvano², Luigi Carro¹
¹UFRGS Institute of Informatics, Porto Alegre, Brazil
{debora.matos, cezar.reinbrecht, jrmartinelli, susin, carro}@inf.ufrgs.br
²Politecnico di Milano - Dipartimento di Elettronica e Informazione, Milano, Italy
{gpalermo, silvano}@elet.polimi.it

Abstract— Networks-on-chip has been seen as an interconnect solution for complex systems. However, performance and energy issues still represent limiting factors for Multi-Processors System-on-Chip (MPSoC). Complex router architectures can be prohibitive for the embedded domain, once they dissipate too much power and energy. In this paper we propose a low power hierarchical network topology with GALS interfaces, allowing each cluster operates in a specific frequency. The clusters are composed by crossbar devices and the number of cores allocated for each cluster is defined considering floorplan information. Experimental results show that our strategy can reduce the power dissipation in up to 58% and the latency in up to 56% for the benchmarks analyzed when compared with a packet-switched mesh network-on-chip.

I. INTRODUCTION

Technology scaling has permitted a large integration capacity, allowing engineers to build in a single chip, many processing elements (PEs), the called Multi-Processors System-on-Chip (MPSoCs) [1]. Several heterogeneous elements are composing these systems, with different bandwidths and quality-of-service (QoS) requirements, supporting a wide number of applications being executed concurrently in the same system [2]. Hence, these systems should be very efficient in terms of power and performance.

One of the design challenges of such systems concerns the communication costs for such applications. Over the past years, Networks-on-Chip (NoCs) designs have been studied as an appropriate solution for such complex hardware systems, due to scalability, parallelism and quality-of-service. However, due to the complexity of MPSoC designs, the systems are requiring other alternatives of interconnection instead of large and somehow inefficient networks-on-chips.

The performance and energy consumption of a system is totally dependent on the network topology and how the cores are mapped in the NoC [3]. One way to obtain high performance is to connect the PEs using a circuit-switching mechanism. Another requirement of complex systems is to provide different levels of communications in a hierarchical manner.

In this paper we present the architecture of ASHiNoC (Application-Specific Hierarchical NoC) that uses a hierarchical interconnection based on crossbar clusters to explore the communication locality and attend the rates required by the PEs. The eventual communications among clusters is done in a higher hierarchical level by a mesh packet-switching NoC. The scalability of this proposal is achieved not by the increase of the ports of a crossbar switch, but rather by replicating the crossbar structure with a limited granularity. The primary justifications for not using a crossbar because of its intrinsic power dissipation have lost arguments. It is clear that there is a limit in the size of a crossbar, and this is justified in a clustered communication approach, as we present in this paper. We have evidenced from different experiments that crossbars with up to 16 ports can be defined to provide high communication performance without surpassing the power consumption of a conventional router. The results of this work were obtained using an optimal mapping for each application taking into account the floorplan information. With this analyzes we can verify the advantages in terms of power and performance of our solution. In this case, an appropriate tool verifies the cores that can be grouped in the same cluster considering the core communication graph and the core sizes of the application.

In the next section we will present the related works. The hierarchical architecture is presented in section III and the GALS interfaces in section IV. In section V we present the experimental results obtained with our solution and finally we conclude the paper in section VI.

II. RELATED WORKS

Many recent proposed routers have showed a complex architecture with virtual channels (VC) [2], [4-6], four or more pipeline-stages and expensive controls in order to increase the performance. However, with the addition of such resources, the router power consumption could be impractical for the embedded domain [6]. With our proposal we can reduce the power consumption since the crossbar switch is used in a hierarchical manner to compose the clusters. The advantages of this topology are obtained since the clusters use small crossbars to interconnect the cores.

Many works discuss solutions using complex routers [4- 5], in order to reach the performance requirement by the system. However, these solutions use excessive area resources, since in order to obtain the mentioned advantages, they at least duplicate the area when compared with a more simple and traditional network-on-chip. Moreover, these proposals use many pipeline
stages in the router and large use of resources, as tables to define the routing, allocators, complex arbiters and controls. Therefore, the use of VCs is only advantageous if the buffers are appropriately shared among the ports [6].

In the proposed interconnection, simpler components are used in each hierarchy level and so, it is possible to reach the performance requirements. It occurs because the higher communications occur in the clusters. Thus, the power reduction is obtained once the clusters are composed by crossbars that use only arbiters and multiplexers, consuming less power than buffers and registers that compose the routers.

In the literature some hierarchical NoCs has been proposed but no one shows the same strategy of ASHiNOC. HiNoC [7] proposes a network with two hierarchical levels, where the first level is a mesh and the second level is formed by fat-trees. The switching of the second level of HiNoC is done with packets, and in top level it uses an asynchronous mesh structure. In our strategy, the local level prioritizes the high bandwidth communication, for this reason, circuit-switching is used in this level.

In [8] eight different topologies were analyzed, three of them being hierarchical networks: CrossTorus, CrossRing and MultiCross. In this comparison, the authors observed that hierarchical NoCs show an area efficient solution, being the MultiCross the topology with the best results among the ones compared. However, MultiCross does not show the same gains when the number of clusters and the communication among them increases, once the top level is a ring–based topology, with an obvious bottleneck. The same problem is observed in the proposal presented in [9].

Another hierarchical topology was proposed in [3]. The reported topology is formed by a mesh in the global network and a bus in the local network. Each bus consists of a local bus interface with an arbiter and an FIFO buffer to connect to the global network. GigaNoC is another proposal where a packet-switched wormhole NoC is used at the global level and each router is connected to a cluster through a bus with 4 coupled cores [10]. These proposals can have some disadvantages when the concentration of the PEs increase in each cluster, since using a bus, only one core can send data for a given time.

In [3] the authors commented that crossbars present large power consumption, however this considerations were made for a crossbar with channel width equal to 512 bits. In our proposal we suggest to use a smaller crossbar connected in each global router with the link width between 16 to 64 bits. With this features, ASHiNOC is able to largely reduce the power consumption. Our strategy allows building a simple but efficient interconnection architecture.

### III. Hierarchical Topology

Considering future systems with several cores, different small applications will be able to run in parallel, and only few interactions are need among them. In another possible scenario, in the same application, different bandwidths will be required by different regions. In these heterogeneous scenarios, the communication among these regions usually has a smaller communication rate. For this reason, a hierarchical topology is fundamental for the future of many-core systems.

ASHiNOC is a hierarchical NoC architecture composed by crossbar switches and mesh routers. The local level is basically a crossbar that is integrated to NoC. Each cluster of this architecture can have coupled different number of PEs and the size of the clusters depends on the application needs. In this case, each application needs to have an appropriate mapping for ASHiNOC and an adequate crossbar granularity. In results section we will detail about the mapping used for this topology.

An example of the proposed topology is illustrated in Fig. 1 and the architecture of the crossbar switches and mesh routers are illustrated in the Fig. 2. The use of the proposed crossbar switch instead of the conventional NoC routers brings many advantages, since the PEs are directly connected to each other. In such case, one does not need to use buffers and extra controls like in a conventional NoC router. As the local level does not have buffers, the hierarchical interconnection here proposed can greatly reduce the power dissipation of the NoC. Moreover, with the proposed topology, the performance of the system is improved, since when the cores are in the same cluster, the latency to transfer the packets for many routers is removed, and a simple switch protocol is used in this case.

![Figure 1: The proposed topology illustration composed by 38 cores.](Image)

![Figure 2: Crossbar switch and mesh router architecture which constitute the proposed hierarchical NoC (ASHiNOC).](Image)

The mesh routers contain five ports, four to connect to the neighbor routers and one to connect to the local network. This is similar to a conventional mesh NoC, but in this case, instead of a PE to be coupled to a local channel, the local channel has a connection with a cluster port. As in the global level the communication rates are very smaller than the local communication, the router design can have a simpler implementation, as for example, without the use of virtual channels, once rarely the network will present contention in the top level.

According to Fig. 2, each crossbar switch is composed by a set of multiplexers and one arbiter for each output port. Thus, when a core connected in the same cluster needs to communicate with another internal core, it sends a request for the arbiter with the ID of the required destination core.
The implementation of the RR algorithm takes only two cycles to verify the requisition of each port, independently of the crossbar switch size. In the first cycle, the arbiter verifies all requisitions of the input ports of a crossbar for an output port and in the second cycle it attends the next requisition. The crossbar switch design is totally configurable, allowing one to define the number of ports of the crossbar and the data link width.

Some studies were obtained to verify the crossbar limitations. A problem related to the clusters are the long wires required to interconnect the PEs in the crossbar, since considering a floorplan, the longest link in each cluster needs to cross by many PEs. Fig. 3(a) shows the crossbar latency analysis for 65nm technology process. This graph considers the crossbar size and the longest link required to do the interconnections among the cores of a regular floorplanning (square or rectangular) and in this case, we have considered cores with area around 1mm². In this graph can be verified the impact of the wire in the crossbar interconnections. Another analysis about the link length is presented in fig. 3(b) with regard to the maximum frequency. According to the link length, a maximum operating frequency can be considered. From these two studies, one can verify that crossbar size up to 16x16 can be adopted without compromise the latency and maximum frequency of the cluster (assuming that for this crossbar size will be required a maximum wire length around 4mm). These parameters were modeled and integrated in the tool used to obtain the results for the comparison between the hierarchical topology considering different cluster sizes can be verified in [18].

The need to define different clock regions in a system has been evidenced for several years and now this requirement is been emphasized with the use of NoCs [11-15]. In order to allow that the cores in the same system can operate with a globally asynchronous locally synchronous (GALS) strategy, synchronization interfaces are required. GALS interfaces are proposed to solve problems related to distribution of a skew-free synchronous clock over the whole chip. In GALS systems other problem is the possibility of synchronization failure between two different clock domains (metastability). Many GALS interfaces have been proposed in the literature [11-15] to cope with these problems and the majority of them are based on the use of asynchronous FIFOs with dual clock domains [14-15].

The use of GALS interfaces is extremely important mainly when a hierarchical network is used. The proposed hierarchical NoC allows that each cluster operates with the ideal frequency. Moreover, as the clusters are heterogeneous, each cluster will have a different maximum operating frequency, since this value depends on the number of crossbar ports and the maximum link length required for interconnecting the cores. The clock synchronizer integrated in the bridges uses a bi-synchronous FIFO like proposed by [14]. The synthesis results for the synchronizer module using the Cadence RTL Compiler tool for 65nm of process technology are shown in table I.

Fig. 4 depicts the bridges used in the proposed architecture. The bridges contain a bi-synchronous FIFO to synchronize the frequency between the two hierarchical levels. Besides, the bridges have the function to adapt the protocol used in each level. In this case, a bridge from crossbar switch to router direction (bridCR) is used to remove the controls of the local protocol. The header of the global level contains the routing information of the mesh topology (we have used a XY algorithm in this case) and a sub-header of the local level, with destination core ID. When a message arrives in the destination cluster, a bridge from router to crossbar switch direction (bridRC) is used to remove the controls and header parts of the packet used in the top level. The bits referring to the cluster header are transfered for the correct controls of the cluster. The bridge bridCR adapts the handshake protocol to allow the sending of data by the network. In this manner, the bridges have a similar function of the network interfaces.

<table>
<thead>
<tr>
<th>Area (um²)</th>
<th>Power (uW)</th>
<th>Frequencies</th>
</tr>
</thead>
<tbody>
<tr>
<td>BridCR</td>
<td>1384</td>
<td>518</td>
</tr>
<tr>
<td>BridRC</td>
<td>1316</td>
<td>702</td>
</tr>
</tbody>
</table>

**TABLE I: SYNTHESIS RESULTS FOR THE SYNCHRONIZER MODULE CONSIDERING TWO FREQUENCY POSSIBILITIES.**

The results were obtained using an analytical model from Hspice simulations and Cadence RTL Compiler synthesis for a 65nm of process technology. The model considers the core distribution in the cluster and the link length required for the interconnections. To estimate the results of the global level (mesh NoC) we have considered the Orion 2.0 tool [16]. The model for the local level was integrated on the tool, allowing estimating the power, area and latency of each cluster. The core allocation was implemented using the NSGA-II multi-objective genetic algorithm [17]. The tool obtains an optimum heterogeneous hierarchical mapping solution considering a decision mechanism based on the product: \((\text{average_power}) \times (\text{average_latency})\). As commented previously, the input parameters of tool are the communication graph and core sizes of the application and we have considered this tool to obtain the results for the comparison between the hierarchical and mesh NoC, as illustrated in Fig. 5. More details about the mapping algorithm and the tool used to obtain the

![Figure 3: (a) Latency estimation for the crossbar considering its interconnections among the cores (b) and the maximum frequency according to the link length.](image)

**V. EXPERIMENTAL RESULTS**

The results were obtained using an analytical model from Hspice simulations and Cadence RTL Compiler synthesis for a 65nm of process technology. The model considers the core distribution in the cluster and the link length required for the interconnections. To estimate the results of the global level (mesh NoC) we have considered the Orion 2.0 tool [16]. The model for the local level was integrated on the tool, allowing estimating the power, area and latency of each cluster. The core allocation was implemented using the NSGA-II multi-objective genetic algorithm [17]. The tool obtains an optimum heterogeneous hierarchical mapping solution considering a decision mechanism based on the product: \((\text{average_power}) \times (\text{average_latency})\). As commented previously, the input parameters of tool are the communication graph and core sizes of the application and we have considered this tool to obtain the results for the comparison between the hierarchical and mesh NoC, as illustrated in Fig. 5. More details about the mapping algorithm and the tool used to obtain the
results of this paper can be verified in [18]. In order to verify the gains for our hierarchical interconnection proposal, we considered three benchmarks: TVOPD [19], TMPEG4 (triple MPEG4 in similar manner to the TVOPD) and the NCS [20] benchmarks. For these benchmarks, the maximum frequency obtained for the clusters were 750MHz and 1GHz, while for the mesh topology the frequency defined was 1GHz. The maximum frequency of each cluster depends on the cluster size, consequently of the crossbar and core sizes and still, of the maximum link length required in the cluster interconnections. In the experiments we defined the link width equal to 32 bits for the cluster interconnections and 16 bits for the NoC. The same algorithm was used to obtain an optimal mapping for ASHiNoC and mesh topologies.

As we can see in Fig. 5, one can largely reduce the average power, the average latency and the network hops when one compares the ASHiNoC solution with a conventional mesh NoC composed only by one core per router. The power reduction for these applications is between 38% and 58% and the latency reduction is between 43% and 56%. The gains obtained with the hierarchical topology are still larger if we compare the average network hops. In these experiments, the average latency is an ideal value, i.e., it considers the time required to send a message without contention in the network but considering the total bandwidth of the communications that share the links. As the tool considers the interconnection area, the total area of the hierarchical network is worse than the conventional mesh. This occurs because the crossbars of the ASHiNoC topology require longer links.

VI. CONCLUSION AND FUTURE WORKS

In this paper we have proposed a hierarchical network topology with bridges composed by a synchronizer module to interconnect the levels. In this manner, we can obtain the advantages of the GALS architectures that can be well applied for the proposed topology. The results show the great advantages in use the ASHiNoC instead of a conventional mesh. With the proposed topology is possible to reduce the power dissipation, the latency and, consequently, the energy consumption.

For future work we plan to propose some adaptability strategies to the hierarchical interconnection in order to allow the use of this architecture for different systems requirements. Besides we intend to extend the ASHiNoC to the 3D domain, since this topological solution could be naturally applied in this new context.

REFERENCES