Networks on Chip (NOC): Design Challenges

Maria Elisabete Marques Duarte

Departamento de Informática, Universidade do Minho
4710 – 057 Braga, Portugal
Elizabete.m.duarte@telecom.pt

Abstract. With the growth of the number of SOC components it is necessary to revalue the design technology and architecture. This work overviews the field of Networks on Chip (NOC), and addresses the distinguishing features of the several architectural designs of a NOC (Octagon and Eclipse). The limitations of the interconnect technology are discussed as well as how his technology has been scaled down to meet systems requirements.

1 Introduction

A System-on-chip (SOC) combines several processing elements into a single silicon chip. The present reality of VLSI projects in market is differentiated by small time-to-market, high complexity and high performance. While the time-to-market is very important, complexity and performance cannot be committed; otherwise it may reach the market with a product that is neither feasible nor competitive. The use of cores (prefabricated modules) is one way to reduce the complexity of current digital systems. Often, these components are technological products, software and knowledge that are subject to patents and costs of intellectual propriety (IP). Thus the digital system will contain some cores that will implement complex functions. One strategy to reduce design time and therefore meet time-to-market requirements is through the use of reusable core or Intellectual Property (IP).

In the future, SOCs will have dozens or even hundreds of processing elements. According to the International Technology Roadmap this will grow to 4 billion transistors running at 10 GHz. Such benefits as system performance improvements, reductions in costs, size and power dissipations and reduced design turn-around-time can be achieved through the use of SOCs. Therefore, on-chip communication architectures will need to be scalable due to the need of interconnecting a greater number of on-chip components into network processors and other SOCs. On the other hand implementing and organizing all the processing elements of a SOC as a single processor with many functional units is not convenient (it is difficult to extract large amounts of instruction level parallelism –ILP- from a single instruction stream).

Future SOCs need to overcome the limiting factor of on-chip interconnections:

- physical constraints (which reduces functional unit utilization and slows down inter-communication);
- limited bandwidth inter-resource;
- inefficient synchronization schemes;
- access-pattern-dependent throughput;
- inability to hide the latency of the internal network;
- poor parallel-computing models and
- energy consumption.

Thus designers to project future chips have to find the appropriate design and process technologies, as well as the ability to interconnect the existing components - including

processors, controllers, and memory arrays - providing the functionally correct and reliable operations conducive to the proper interaction of the components.

This communication emphasizes the importance of a new SOC design approach, the NOC. Section 2 describes NOC schemes and design methodologies, section 3 explain two NOC architectures, the Octagon and the Eclipse.

2 The Approach to Networks

To consider a SOC as a micro-network, with a group of interconnected processing elements amongst themselves, allows the use of techniques and design tools used in generic networks facilitating the design of the complex SOCs. Table 1 shows the difference between a micro-network and a general network.

Table 1. Comparison between General Networks and Micro-Networks

Characteristics	General Network	Micro-Network
Proximity	Apart	Close
Non-determinism	Great	Small
Energy constraints	Not relevant	Relevant
Design-time specialization	Not relevant	Relevant
General purpose communication	Emphasize	Less restrictive
Modularity	Emphasize	Less restrictive
Compatibility constraints	Strongly influenced	Less restrictive
Standardization constraints	Strongly influenced	Less restrictive[1]
Specific end applications	Decoupled from	Less restrictive

The network is the abstraction of the communication between components and must satisfy quality-of-service requirements—such as reliability, performance, and energy limits—under the limitation of intrinsically defective signal transmission and considerable communication delays on wires.

Network on a chip schemes came to solve future SOC architectural and design productivity issues. These issues are overcome through the capability of a NOC to provide a regular connection network connecting multiple resources and standardizing the management of various inter-resources communications needs. Other significant motivations for the NOC schemes are: reusability of existing IP blocks, physical-architectural-level design integration, and platform-based design methodology. [1]

Benini and Micheli propose using the micro network stack paradigm, an adaptation of the protocol stack to abstract the electrical, logic, and functional properties of the interconnection scheme. Every layer is specialized and optimised for the target application domain in a vertical design flow.

2.1 Interconnection Network's Topology

This section specifies the interconnection network's topology and physical organization.

The physical layer specifies the connection wire links. It must transmit a signal without errors and with a low level of energy consumption satisfying competing quality metrics and making available a complete abstraction for the micro-network layers above.

1 In SOC networks, these constraints are less restrictive because developers design the communication network fabric on silicon from scratch. Thus, only the abstract network interface for the end nodes requires standardization. Developers can tailor the network architecture itself to the application, or class of applications, the SOC design targets.

Shared Medium Networks: this is the most common of SOC architectures, denoted by the simplest interconnect structures. In this type of architecture all communication devices share the transmission medium. Only one device can drive the network at a time. This type of topology is energy inefficient and not scalable.

Direct and Indirect Network: in a direct or point-to-point network each node directly connects to a limited number of neighbouring nodes. This architecture overcomes the scalability problems of shared-medium networks. In indirect or switch-based networks a connection between nodes must go through a set of switches. The network adapter associated with each node connects to a switch's port.

Hybrid Networks: two examples are multiple-back-plane and hierarchical buses. These architectures cluster tightly coupled computational units with high communications bandwidth and provide lower bandwidth intercluster communications links.

2.2 Micro-Network Control

The protocols specify how to use the network resources during system operation. Network control dynamically manages network resources during system operation, striving to provide the required quality of service.

Data Link Layer: the physical layer is an unreliable digital link in which the probability of bit upsets is non-null. Data-link protocols increase the reliability of the link, up to a minimum required level, under the assumption that the physical layer by itself is not sufficiently reliable. It defines error detection and correction protocols in packet communications. The packet size and the number of outstanding packets can be adjusted in this level seeking maximum performance with a low probability of residual error, within energy consumption constraints.

Network Layer: this layer implements end-to-end delivery control. Switching algorithms can be grouped into three classes: circuit, packet, and cut-through switching. Switching is closely related to routing. Routing algorithms establish the path a message follows through the network to its final destination. Deterministic routing algorithms are best suited for identical or regular traffic patterns providing the same path between a given source-destination pair. In contrast, adaptive approaches use information regarding network traffic and channel conditions to avoid congested network regions. This approach is preferable when dealing with irregular traffic or in networks with unreliable nodes and links. Depending on the application domain, no determinism can be more or less tolerable.

Transport Layer: above the network layer, the transport layer decomposes messages into packets at the source, re-sequences and reassembles the messages at the destination. Packetization granularity presents a critical design decision because most network-control algorithms are highly sensitive to packet size. Packet standardization constraints can be relaxed in SOC micro-networks, which can be adapted at design time. In general, either deterministic or statistical procedures will offer the basis for flow control and negotiation.

2.3 Software Layers

Network architectures and control algorithms constitute the infrastructure and provide communication services to the end nodes, which are programmable in most cases.

System Software: The operating system holds the system programs that maintain SOC operation. Each programmable component will be provided with system software to support its own operation, control its communication with the micro-network, and interact effectively with neighbour components' system software. On-chip communication proto-

cols should be programmable at the system software level in order to adapt the underlying layers to the components' features, supplying an abstraction of hardware platform. The system software must support dynamic power management (DPM) for its components and dynamic information-flow management. DPM enables selecting the suitable component state to service a workload ensuring the minimum energy consumption.

Application Software: The system software provides adequate libraries and facilities to support standard programming languages. SOC application software development should reach portability and offer some intelligence to leverage the dispersed nature of the underlying platform.

3 NOC architectures

This section outlines the NOC architectures Octagon and Eclipse under the following main features:

- Physical architectural topology, the first layer of a NOC (Interconnection topology);
- Resources: computing resources, storage resources, static and dynamic hardware blocks, switch parameters (number of wires connecting a switch to another switch and a switch to a resource);
- Communication protocols, the second layer of a NOC (Micro-Network control);
- Scalability.

3.1 Octagon

The next generation of Internet backbone routers must deliver ultrahigh performance over an optical infrastructure and realize its functions fulfilling the service level agreements (SLA). Current processor architecture in router equipments are RISC based which provides the flexibility to upgrade to new tasks falling short in satisfying the growing speed requirements for complex tasks. Application Specific Integrated Circuits (ASICs) are also finding in current router architectures, these provide the speed but not the required programming flexibility. The Octagon design surges them the need of the designers to develop new high-speed network processors that permit flexible programmability and work at 40 Gbps and process $57*10^6$ instructions per second and exceed the limitations of traditional RISC and ASIC architectures. [2]

Octagon is a new On-Chip communication architecture where its cost, performance, and scalability advantages make it suitable for the aggressive on-chip communication demands of not only networking SOCs, but also SOCs in several other domains.

Physical architectural topology: a basic Octagon architecture contains eight nodes and 12 bidirectional links. Each node connects to the previous, the successive and the following. This architecture guarantees two-hop communication between any pair of nodes, higher aggregate throughput than a shared bus or crossbar interconnect under certain implementation conditions, simple shortest-path routing algorithm, and less wiring than a crossbar interconnect.

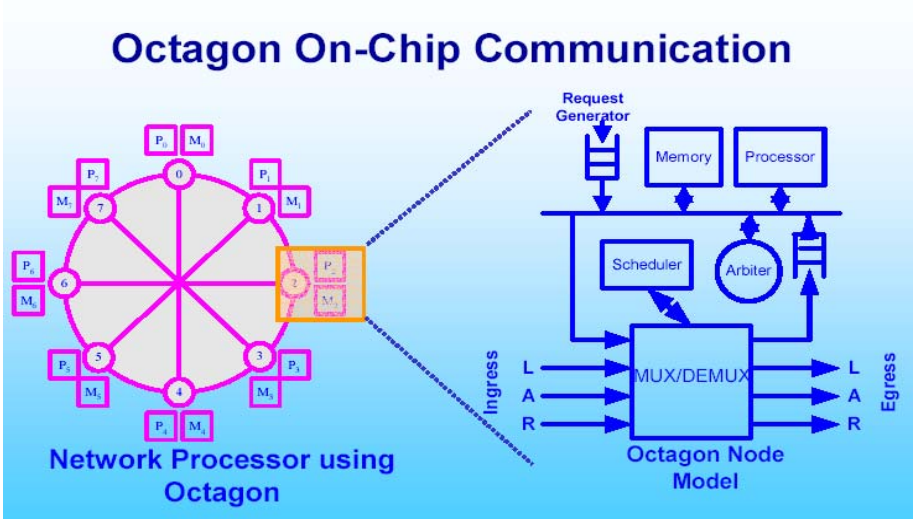

Fig. 1. Basic Octagon configuration

Resources: each Octagon node is a processor with a local memory, a non-blocking switch and maintains three queues of outstanding requests, one for each egress link. For Octagon's best-fit connection scheduling, a node (process) is not blocked if the scheduler cannot schedule its communication request immediately. In its place, the requesting node queues the request in its egress queue.

Communication protocols: Octagon operates in packet or circuit switched mode. In the former the networks nodes buffer packets at intermediate nodes if there is contention at the egress link, in the latter the network arbiter allocates the entire path between source and destination nodes of a communicating node pair for a number of clock cycles.

An Octagon packet can have a fixed or a variable length.

The Octagon can operate with an arbiter where the priority scheme regards to the overall network; this global scheduler gives priority to the head-of-line in arrival time order (lower arrival time implies higher priority). The central scheduler performs switch arbitration. The switch has neither input nor output buffering. These strategies can improve system performance and node utilization more than some communication protocols (especially most bus protocols), which pause the requesting node until its request can be granted. However, each Octagon node must have a large enough queue to avoid packet loss. A system designer can enable a zero packet loss agreement in Octagon by having the packet scheduler refuse requests if the egress queue is at full or near-full capacity, thereby stalling the requesting node (as many existing buses do).

The best-fit scheduler is a connection-oriented communication protocol with cannot contain simultaneously overlapping connections. Karim and colleagues show that a relatively high utilization of 12 communications and at a 10^{-4} packet loss probability, a system using the Octagon architecture requires fewer than 50 packet buffers. Thus, the average queue occupancy is not excessive.

Scalability: the most relevant features of the Octagon architecture is it's to scale linearly, this however requires two different nodes types: bridge nodes connect adjacent Octagons and perform hierarchical packet routing and member nodes attach to only one Octagon. The Octagon's nodes require either three or six wires connecting to its neighbours, resulting in wiring complexity. The maximum distance increases linearly as Octagon grows. In this strategy, the maximum distance between nodes grows much more slowly, but it does remain constant for the crossbar. This is good for SOC where low wire complexity is the principal concern. But, this feature might not suit systems where high

throughput is the dominant consideration. These systems can use a second scaling strategy of Octagon architecture that performs better than the first but has more complex wiring: extending Octagon into multidimensional space. Octagon bridge nodes are linked with each other. The maximum distance between nodes scales much better under the second strategy compared to that of the first one.

3.2 Eclipse

The embedded chip-level integrated parallel supercomputer (Eclipse) is a scalable, high-performance computing architecture for NOCs.
Eclipse features a completely software-based design methodology to support flexibility and general-purpose operation. [3]

Physical architectural topology: the Eclipse NOC is a high-bandwidth acyclic variant of a 2D sparse mesh with separate lines for messages from processors to memories, and from memories to processors; two-level switch organization; simple routing; an efficient synchronization mechanism; and randomised hashing of memory locations over the memory modules.

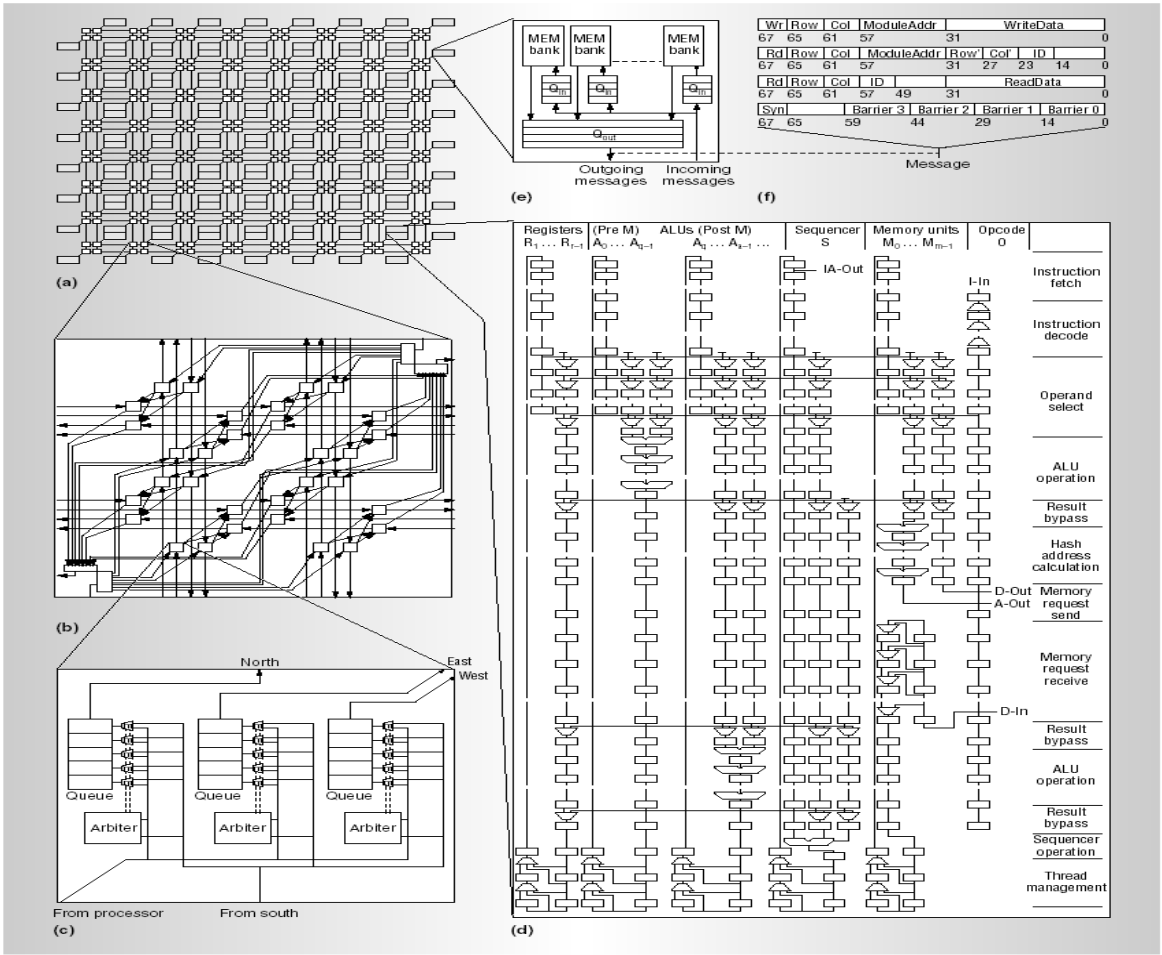

Fig. 2. Block diagrams of an Eclipse(a) superswitch(b) switch(c) MTAC processor(d) memory module(e) message formats(f)

Resources: an Eclipse chip contains a multithreaded architecture with chaining (MTAC) processors with dedicated instruction memory modules, highly interleaved data memory

modules. Eclipse's optimised resources—MTAC processors and interleaved memory modules—can easily produce one message per clock cycle. [2]

MTAC is a multithread processor architecture specifically designed to implement the parallel random-access machine (PRAM) model on physically dispersed memory architectures. The list of used techniques is:

- a VLIW-style instruction set with fixed execution ordering of sub instructions, reflecting the chain-like organization of functional units;
- a hardware-assisted barrier synchronization mechanism;
- the organization of functional units in MTAC aims to exploit ILP during parallel execution supersteps;
- the support of overlapped execution of a variable number of threads, with hazard-free interthread pipeline;
- superpipelining, decreases the clock cycle to a minimum;
- VLIW scheduling used to simplify the structure of the processor.

Eclipse has two types of memory modules - *data memory* and *instruction memory* modules - that are isolated from each other to ensure uninterrupted data and instruction streams to the MTAC processors. The memory and the processors have different speeds that result in performance loss, to avoid this Eclipse use profound interleaving of memory modules.

Each switch on Eclipse consists of eight switch elements (a simple device in which output queues and arbiters route messages). An arbiter detects messages targeted at a nearby queue and checks whether the queue has space for them. The switches related to each resource pair is grouped into superswitches. This two-level structure allows the sending of a message from a resource to any of the superswitch switches in a single clock cycle and pipeline switching operation naturally.

Communication protocols: execution in Eclipse occurs in supersteps. A superstep consists of a set of independent local computations, followed by a global communication phase and a barrier synchronisation. Each superstep is further subdivided into three ordered phases consisting of:

- computation, locally in each process, using only values stored in the memory of its processor;
- communication actions amongst the processes, involving movement of data between processors;
- a barrier synchronisation, which waits for all of the communication actions to complete, and which then makes the data that was moved available in the local memories of the destination processors.

During a superstep, each thread of each processor alternately executes an instruction and can include at most one shared memory reference sub instruction.

The routing is realized by a simple routing Eclipse routes messages using the simple greedy algorithm with two intermediate targets, a switch on a superswitch. The first are related to the sending resource and the second are related to the target resource.

Eclipse uses an advanced synchronization wave technique. When a processor has sent all messages belonging to a single superstep, it sends a synchronization message. When a switch receives a synchronization message from one of its inputs, it waits until it has received synchronization messages from all inputs, and then forwards the synchronization wave to all of its outputs. When a synchronization wave sweeps over a network, all switches, modules, and processors receive exactly one synchronization message via each input link and send exactly one synchronization message via each output link.

Eclipse serve as a single-instruction multiple data (SIMD) machine, a multiple-instruction, multiple-data (MIMD) machine, or as a combination of several SIMD and MIMD machines.

Scalability: An Eclipse's structure is homogenous, simplifying design and making it easier to incorporate into a larger SOC.

The comparisons made by Forsell show that Eclipse presents better performance than the generic NOC architecture, based on a 2D mesh network with ARM9-style processors. The comparison was based in calculates Bench's execution time as a function of the memory bank cycle time, the area constant, the number of clock cycles per hop, the level of superpipelining, the number of resources and the fraction of dependent parallel portions. The Eclipse performance is somewhat independent of memory speed, switch delay, and fraction of dependent parallel portions. The performance in Eclipse is improved increasing the level of superpipeline while the basic NOC requires an increase in the number of processors to enhance its performance.

4 Conclusions

NOCs offer significant potential for innovation: On-chip micro-network architectures and protocols can be tailored to specific system configurations and application classes. Further, the impact of network design and control decisions on communication energy presents an important research theme that will become critical as communication energy consumption scales up in SOC architectures. A layered micro-network design methodology will likely be the path to mastering the complexity of SOC designs in the next generation of SOC.

Developers have found adequate solutions to the problems of designing SOC processor architecture in routers equipments, the Octagon. Octagon offers a solid network outline and implements a simple communication protocol in SOC.

Eclipse clearly provides a novel approach to SOC design, avoiding the pitfalls of other recently proposed NOC schemes. The proposed architecture promises to bring easy-to-use, truly scalable high-performance computing to chip-level designs.

While selecting an appropriate communication architecture for the target application is an important issue, for a given communication architecture topology, there remains a large design space that needs to be explored in order to optimally map the system's communications onto physical paths of the communication architecture. Equally important are the subject of synchronism of SOCs.

References

[1] Benini, L., Giovanni M.: Networks on Chips: A New Soc Paradigm. IEEE Micro. (2002) 70-77

[2] Karin, F., Nguyen A., Dey S.: An Interconnect Architecture for Networking Systems on Chips. IEEE Micro. (September-October 2002) 36-45

[3] Forsell M.: A Scalable High-Performance Computing Solution for Networks on Chips. IEEE Micro. (September-October 2002) 46-55

[4] Kumar V.P., Lakshman T.V., Stiliadis D.: Beyond Best Effort: Router Architectures for the Differentiated Services of Tomorrow's Internet. IEEE Comm. (May 1998) 152-164

[5] Lahiri K., Raghunathan A., and Dey S.: Evaluation of the Traffic Performance Characteristics of System-on-Chip Communication Architectures. IEEE CS Press. Los Alamitos, Calif. (2001) 29-35