

Networking chip subsystems

Giovanni De Micheli
CSL Stanford University
Stanford, CA 94305, USA

1 Introduction

Systems on chip (SoCs) are designed with silicon nanoscale technologies, i.e., with transistor gate lengths shorter than 100nm and decreasing from year to year. Moreover, SoCs integrate subsystems with heterogeneous functionality (e.g., digital, analog, radiofrequency) and heterogeneous technologies (e.g., sensors, optical interfaces, MEMs.) In the coming years, novel technologies may complement and/or substitute for silicon as a substrate for computational or storage subsystems.

Overall, the technology trend shows an increasing device density on chip, and as a result power density and heat extraction will be major design challenges. Energy-efficient design policies will be pervasive. Along this direction, voltage levels on chip will be reduced to the order of a few hundred millivolts. Unfortunately, voltage reduction will adversely affect signal integrity. Signal delays on wires will dominate delays in computational units, and their accurate prediction will be increasingly more difficult.

System on Chip will find application in many embedded systems (e.g., portable communicators, vehicle control systems, health monitoring) where reliability and robustness is a major concern. Thus, new system level design methodologies will be driven both by the end-application requirements as well as by the physical limitations of the underlying technology.

As a result of the increasing complexity of SoC design, future methodologies will rely on the following principles. First, SoCs will be designed using pre-existing components, such as processors, controllers and memory arrays. Design methodologies will support component re-use in a plug-and-play fashion. Second, reliable operation of the interacting components will be guaranteed only by a structured methodology for interconnect design, that relies on networking technology ported to the microelectronic environment.

SoCs differ from wide-area networks because of local proximity and because they exhibit much less non-determinism. Local, high-performance networks (such as those developed for large-scale multiprocessors), have similar requirements and constraints. A few distinctive characteristics are unique of SoC net-

works, namely, energy constraints and design-time specialization.

Whereas computation and storage energy greatly benefits from device scaling (smaller gates, smaller memory cells), the energy for global communication does not scale down. On the contrary, projections based on current delay optimization techniques for global wires show that global communication on chip will require increasingly higher energy consumption. Hence, communication-energy minimization will be a growing concern in future technologies. Furthermore, network traffic control and monitoring can help in better managing the power consumed by networked computational resources. For instance, clock speed and voltage of end nodes can be varied according to available network bandwidth.

Design-time specialization is another facet of the SoC network design. Whereas macroscopic networks emphasize general-purpose communication and modularity, in SoCs networks these constraints are less restrictive. The communication network fabric is designed on silicon from scratch. Standardization is needed only for specifying an abstract network interface for the end nodes, but the network architecture itself can be tailored to the application, or class of applications, targeted by the SoC design.

2 Network Architectures and Protocols

Network design entails the specification of *network architectures* and *control protocols*. The architecture specifies the topology and physical organization of the interconnection network, while the protocols specify how to use network resources during system operation. On chip networks are also referred to as *micro-networks*, to distinguish them from local/wide area networks.

2.1 Architectures

The current dominant on-chip communication paradigm is *shared medium* as exemplified by bus-based architectures. Several existing bus standards (e.g., AMBA) are used successfully today, but their effectiveness is likely to fade as more components are interconnected, making them slow and energy-inefficient communication means.

The *direct* or *point-to-point* network is a network architecture that overcomes the scalability problems of shared-medium networks. In this architecture, each node is directly connected with a subset of other nodes in the network, called *neighboring* nodes. Nodes are on-chip computational units, but they contain a network interface block, often called a *router*, which handles communication-related tasks. Each router is directly connected with the routers of the neighboring nodes. Differently from shared-medium architectures, as the number of nodes in the system increases, the total communication bandwidth also increases. Direct interconnect networks are therefore very popular for building large-scale systems.

Octagon is an example of a direct network on chip. It has been designed by STMicroelectronics for network processors. In an octagon network, eight processors are connected by an octagonal ring and three diameters. Messages between any two processors require at most two hops. Moreover, scalable. If one node processor is used as the bridge node, more Octagons can be tiled together, as shown in Fig. ??.

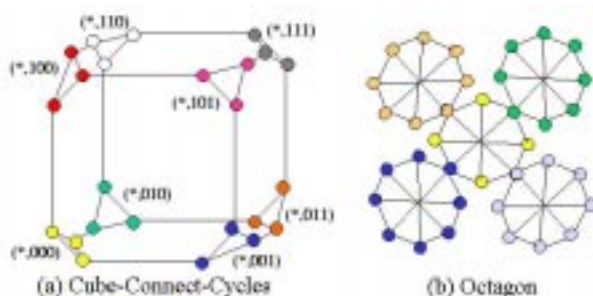


Figure 1: Octagon networks and cube-connected-cycles networks

Indirect or *switch-based* networks are an alternative to direct networks for scalable interconnection design. In these networks, a connection between nodes has to go through a set of *switches*. The network adapter associated with each node connects to a port of a switch. Switches do not perform information processing. Their only purpose is to provide a programmable connection between their ports, or, in other words, to set up a communication path that can be changed over time [?]. As an example, *SPIN* is an indirect network on chip with a *fat tree* topology. Messages reach the processing elements by travelling up and down the routing tree.

FIGURE SPIN

2.2 Protocols

On-chip global wires are the physical support for communication and embody the physical network architecture. Global wires can be seen as *noisy channels*. In micro-networks, noise is the abstraction of the signal disturbances, such as timing errors, cross talk, electromagnetic interference, etc..

Network protocols are designed in layers. The data-link layer abstracts the physical layer as an unreliable digital link. The main purpose of data-link protocols is to increase the reliability of the link up to a minimum required level, and to regulate the access to a shared-medium network, where contention for a communication channel is possible.

Error detecting and *correcting codes* (ECCs) are used in different ways to provide for signal transmission reliability. When only error detection is used, error recovery involves the retransmission of the faulty bit or word. When using

error correction, some (or all) errors can be corrected at the receiving end. Error detection and/or correction requires an encoder/decoder pair at the channel's end, whose complexity depends on the encoding being used. Obviously, error detection is less hardware intensive than error detection and correction. In both cases, a small delay has to be accounted for in the encoder and decoder. Data re-transmission has a price in terms of latency. Moreover, both error detection and correction requires additional (redundant) signal lines.

An effective way to deal with errors in communication is to packetize data. If data is sent on an unreliable channel in packets, error containment and recovery is easier, because the effect of errors is contained by packet boundaries, and error recovery can be carried out on a packet-by-packet basis. In this case, the redundant data lines can be avoided by adding the redundant information at the tail of the packet, thus trading off space for delay.

Error correction can be complemented by several packet-based error detection and recovery protocols, such as *alternating-bit*, *go-back-N*, *selective repeat*, which have been developed for macroscopic networks [?, ?]. Several parameters in these protocols (e.g., packet size, number of outstanding packets, etc.) can be adjusted depending on the goal to achieve maximum performance at a specified residual error probability and/or within given energy consumption bounds.

As an example, the SPIN micro-network [?] defines packets as sequences of 36-bits words. The packet header fits in the first word. A byte in the header identifies the destination (hence, the network can be scaled up to 256 terminal nodes), and other bits are used for packet tagging and routing information. The packet payload can be of variable size. Every packet is terminated by a trailer, which does not contain data, but a checksum for error detection. Packetization overhead in SPIN is 2 words. The payload should be significantly larger than 2 words to amortize the overhead.

The protocol network layer implements the end-to-end delivery control in advanced network architectures with many communication channels. Key tasks are *switching* and *routing*. Switching algorithms can be grouped in three classes: *circuit switching*, *packet switching*, and *cut-through switching*. With circuit switching, a path from the source to the destination is reserved prior to the transmission of data, and the network links on the paths are released only after the data transfer has been completed. Circuit switching is advantageous when traffic is characterized by infrequent and long messages, because communication latency and throughput on a fixed path are generally very predictable. With circuit switching, network resources are kept busy for the duration of the communication, and the time for setting up a path can produce a sizable initial latency penalty. Hence, circuit switching is not widespread in packet networks where atomic messages are data packets of relatively small size: communication path setup and reset would cause unacceptable overhead and degrade channel utilization.

As an example, the SPIN micro-network adopts cut-through switching to minimize message latency and storage requirements in the design of network

switches. However, it provides for some extra buffering space on output links to store data from packets that are blocked. It is interesting to notice that the fat tree network architecture in SPIN is non-blocking if packet size is limited to a single word. Blocking in SPIN is a side effect of cut-through switching alone, because packets can span more than one switch.

The protocol transport layer decomposes messages into packets at the source. It also resequences and reassembles them at the destination. Packetization granularity is a critical design decision, because the behavior of most network control algorithms is very sensitive to packet size. In most macroscopic networks, packets are standardized to facilitate internetworking, extensibility and compatibility of networking hardware produced by different manufacturers. Packet standardization constraints can be relaxed in SoC micro-networks, which can be customized at design time.

In micro-networks, the size of the packets has a direct impact on both performance and energy consumption. Thus, an interesting problem is the search for optimal packet size. To some extent, optimal packetization depends on the network architecture and on the system function (including application software).

ONE PARAGRAPH ON MIDDLEWARE

3 Conclusions

The challenges of designing SoCs in 50-100nm technologies available in the second part of this decade include coping with design complexity, providing reliable, high-performance operation and minimizing energy consumption. Starting from the observation that interconnect technology will be the limiting factor for achieving the operational goals, we envisioned a communication-centric view of design. We focused on energy efficiency issues in designing the communication infrastructure for future SoCs. We described several open problems at various layers of the communication stack, and we outlined basic strategies to effectively tackle the energy efficiency challenge for on-chip communication networks.

References

- [1] L. Benini and G. De Micheli, "Networks on Chips: A New SoC Paradigm," *IEEE Computers*, January 2002, pp. 70-78.
- [2] D. Bertozzi, L. Benini and G. De Micheli, "Low-Power Error-Resilient Encoding for On-chip Data Busses," *DATE, International Conference on Design and Test Europe* Paris, 200, pp. 102-109.
- [3] P. Guerrier, A. Grenier, "A Generic Architecture for On-chip Packet-switched Interconnections," *Design Automation and Test in Europe Conference*, pp. 250-256, 2000.

- [4] T. Theis, "The future of Interconnection Technology," *IBM Journal of Research and Development*, Vol. 44, No. 3, May 2000, pp. 379-390.
- [5] H. Zhang, V. Prabhu, V. George, M. Wan, M. Benes, A. Abnous, J. Rabaey, "A 1-V Heterogeneous Reconfigurable DSP IC for Wireless Baseband Digital Signal Processing," *IEEE Journal of Solid-State Circuits*, vol. 35, no. 11, pp. 1697-1704, Nov. 2000.