Universidad Politécnica de Valencia

Redes en Chip

The Spidergon Network-on-Chip

Author:
Andrew Fecheyr Lippens

June 17, 2010

Abstract

TODO: write an abstract

Contents

1 Introduction

A network-on-chip or NoC has become one of the key components in digital chips. As the number of components and transistors per chip grows continuously, most chips have become multi-core. These chips consist of a number of separate cores, often with fixed-function accelerators, memory arrays and input/output units.

Historically buses or crossbar switches were used to connect the functional components of a chip. Above four cores, however, the limited bandwidth of a bus becomes a major limitation, and above 8 or 16 cores, crossbar switches become very expensive. Only a NoC provides the scalability needed for emerging multicore chips. The NoC bounds the performance potential of these chips and accounts for a substantial fraction of the power and area consumption.

TODO: introduce Spidergon

2 Background

2.1 Technical Terms

This paper uses technical terms from the domain of computer networks. Some of them are explained here to help the reader.

System on Chip (SoC) refers to the integration of all components of a computer or electronic system into a single integrated circuit (chip) [6]. SoCs are often used in space and power constrained devices.

A Network on Chip (NoC) is a packet-switched on-chip micronetwork. It is the natural evolution of traditional circuit-switched bus solutions.

An Interconnect Processing Unit (IPU) is a configurable and extensible communication component that implements system services and core communication. It is essentially an on-chip communication network with hardware and software elements which jointly implement key functions of different SoC programming models through a set of communication and synchronization primitives and provide low-level platform services to enable advanced features in modern heterogeneous multi-core SoCs.[1]

A Chip multiprocessor (CMP) is a multi-core processor in which the distinct cores are integrated onto a single integrated circuit die.

IP block: In electronic design a semiconductor intellectual property core or IP block is a reusable unit of logic, cell, or chip layout design that is the intellectual property of one party. They are often used as building blocks in SoC designs.

2.2 Bus vs. Network-on-Chip

SoC communication has traditionally been done using on-chip busses. A diagram of an early bus architecture is depicted on figure 1. This allowed designers of early SoCs to select IP blocks, place them onto the silicon, and connect them together with a standard on-chip bus.

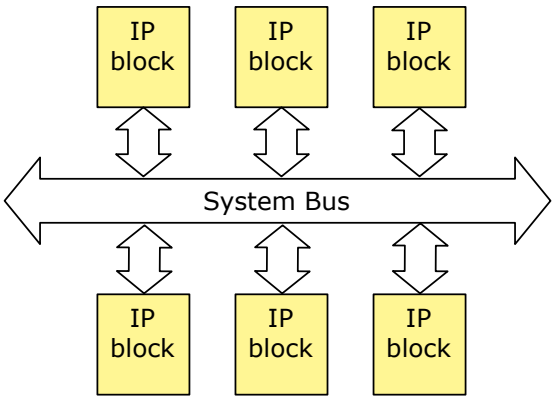

Figure 1: Early SoC structure based on a single shared bus

Buses are well understood and have been successfully implemented in many complex SoCs focusing on both application-based and processor-specific features, thus providing compatibility with most available IP and processor cores, and SoC communication requirements. Examples of more modern on-chip buses include ST Microelectronics' STBus [4], ARM's AMBA AHB and IBM's Core-connect [3].

However, buses do not scale well. With the rapid rise in the number of blocks to be connected and the increase in performance demands, today's SoCs cannot be built around a single bus. Instead, complex hierarchies of buses are sometimes used, with sophisticated protocols and multiple bridges between them. Communication between two remote blocks can go via several buses, and every section of every path must be carefully verified [2].

Moreover, the increase in the number of connected IP and processor cores causes the bus to become a communication bottleneck due to sharing of aggregate bandwidth by all attached units, and worsens arbiter delay, thus introducing

variation, limiting scalability and making the arbiter instance-specific. Additional concerns regarding reachable clock frequency and time closure arise, since every bus transaction involves all connected cores and unnecessarily consumes significant power [1].

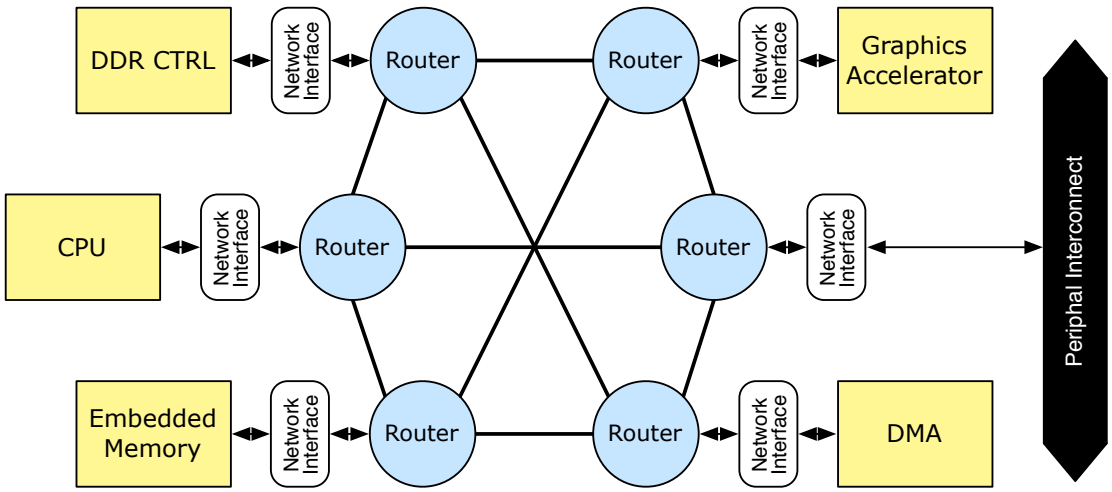

Figure 2: SoC equipped with a Network-on-Chip

3 Spidergon STNoC

As a first attempt at designing a configurable IPU, Spidergon STNoC IPU technology consists of a flexible, pseudo-regular on-chip communication network implementing a set of customizable low-level platform services and a set of communication primitives.

The Spidergon STNoC IPU is a software programmable on-chip communication network that enables system designers to extend communication primitives. These communication primitive extensions are then automatically synthesized, placed, and routed into the Spidergon STNoC communication network. The communication network leverages graph properties of the Spider- gon regular topology discussed, while matching heterogeneity of Multicore SoCs through software programmability concepts, application-specific hardware configurability and extensibility.

Existing NoC architectures are either based on a fixed regular network topology, or they are topology-independent, i.e. they can be customized to a particular application-specific communication graph. The Spidergon STNoC topology fills

the gap between these two approaches, trading off regularity with customizability; for this reason, it is sometimes called a pseudo-regular topology[1]. Using this concept, NoC topology becomes an architectural parameter that can be configured depending on the communication patterns exhibited by the application. Moreover, in order to address Multicore application requirements for feature-rich devices, e.g. speech processing, video, GPS, security, and mobility, Spidergon STNoC technology provides a set of low-level platform services. The most important services defined in Spidergon STNoC are security, power management, and QoS. These services can be instantiated depending on the real target application and may be augmented by customer-specified services.

3.1 Spidergon Topology

The Spidergon topology provides an interesting price/performance trade-off for SoC devices. In the Spidergon topology, all of the IP blocks are arranged in a ring where each IP block is connected to its clockwise and its counter-clockwise neighbour and directly to its diagonal counterpart in the network. This allows the routing algorithm to minimise the number of nodes that a data packet has to traverse before reaching its destination. A particularly important advantage is that the functional diagram (shown below on the left of figure 3 for a network of 16 nodes) corresponds to a simple planar implementation (shown on the right side of figure 3) in which the wiring only needs to cross itself at one point. This is a key benefit in delivering high price/performance [5].

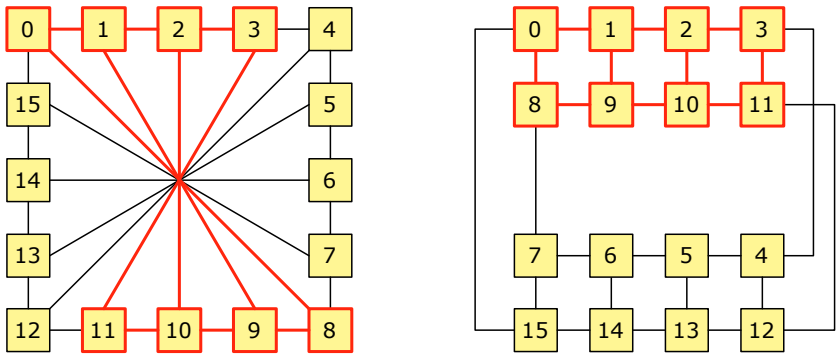

Figure 3: Equivalent representations of the Spidergon topology for $N = 16$

Topologies with increased connectivity, such as 2D mesh, provide very good theoretical metrics, but in most cases, these features cannot be fully exploited due to the nature of communication traffic in Multicore SoC applications. On the other hand, simple topologies, such as rings, are cost-effective in terms of manufacturing

cost, but deliver relatively poor performance, especially as the number of connected cores increases. Spidergon tries to strike a balance in this tradeoff.

Compared to complex topologies, Spidergon offers a small number of links and simple implementation. For current, realistic NoC configurations with up to 60 nodes, the proposed Spidergon graph has a smaller number of edges and a competitive network diameter with respect to fat-tree or 2D mesh topologies.

Figure 4 illustrates different families of topologies supported by the Spidergon STNoC. These topologies are essentially degree 2 or 3 Spidergon subgraphs that range from rings and simple spanning trees to irregular chordal rings. Depending on application traffic requirements and especially mapping of master and slave agents, connection paths can be removed if never used and cross connections customized to provide shortcuts between any pair of nodes in the ring.

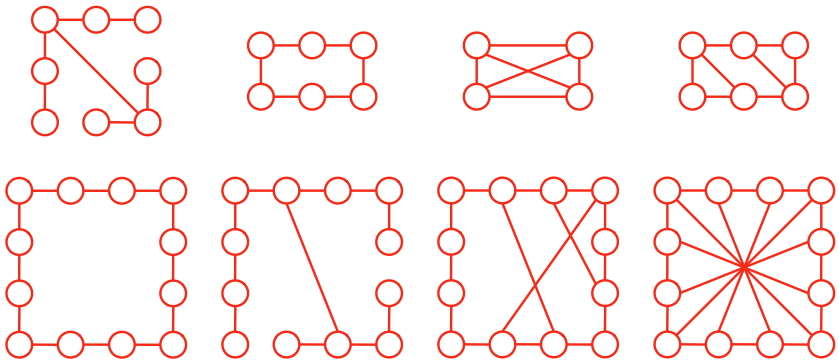

Figure 4: Different topologies supported by Spidergon STNoC

3.2 Topology Comparison

Each NoC topology offers a different set of tradeoffs in terms of metrics, such as:

- Vertex symmetry, which affects routing or scheduling cost and performance,
- Network degree, which affects the operating frequency and the complexity of the routers,
- Network extendibility, which should be as low as possible to maximize flexibility in Multicore designs, and
- Average distance between nodes.

Property	**Ring**	**Spidergon**	**2D $n \times m$ Mesh**
Symmetry	both	vertex	no
Degree	2	3	4
Links	$2N$	$3N$	$4mn - 2m - 2n$
Extendibility	1	2	$\min(m, n) \geq 2$
Node connectivity	2	3	2
Diameter	$N/2$	$N/4$	$m + n - 2$
Average distance	if N even: $N/4$ if N odd: $(N^2 - 1)/4N$	if $N = 4n$: $(2n^2 + 2n - 1)/N$ if $N = 4n + 2$: $(2n^2 + 4n + 1)/N$	$\frac{(m+n)(mn-1)}{3mn}$
Bisection width	4	if $N = 4n$: 8 if $N = 4n + 2$: 10	if max(m,n) even: 2min(m,n) if max(m,n) odd: 2min(m,n) + 2

Table 1: Theoretical metrics for different topologies

Table 1 presents a summary of static topological metrics for three typical NoC topologies: Ring, Spidergon and $m \times n$ Mesh.

The data shows that for $N < 60$, the Spidergon topology despite its constant bisection is very competitive. Due to its higher connectivity, the Spidergon topology always outperforms Rings in terms of diameter and average distance. Spidergon also competes favorably or outperforms 2D Mesh, although outcomes depend on network size. In fact, while Spidergon properties scale linearly with the network size, 2D Mesh behavior is quite irregular. This irregularity is a severe bottleneck in Multicore SoC design, since it complicates design space exploration in terms of cost-performance tradeoffs. For example, a 22 node 2D mesh has smaller diameters and average distance metrics than a 24 node 2D mesh.

3.3 Switching Strategy

The switching strategy refers to how a packet traverses the route. More specifically, the switching technique determines how a message is fragmented and transmitted from an input of the on-chip network to an output resource by defining how channels and possible buffers along the path are allocated. There are four types of switching techniques:

- Circuit switching,
- Store-and-forward,
- Wormhole, and
- Virtual Cut-through.

Spidergon STNoC adopts wormhole routing, which is nowadays commonly used in NoC design. Due to this switching technique, the Spidergon STNoC router has a simple architecture, occupies a small area, and is extremely fast in terms of operating frequency.

4 Evaluation

TODO: look for good evaluation resources

5 Conclusion

TODO: write a conclusion

References

[1] M. Coppola, M.D. Grammatikakis, R. Locatelli, G. Maruccia, and L. Pieralisi. *Design of Cost-Efficient Interconnect Processing Units: Spidergon STNoC*. CRC, 2008.

[2] S. Furber and J. Bainbridge. Future trends in soc interconnect. In *System-on-Chip, 2005. Proceedings. 2005 International Symposium on*, pages 183–186, 2005.

[3] K.K. Ryu, E. Shin, and V.J. Mooney. A comparison of five different multiprocessor soc bus architectures. In *dsd*, page 0202. Published by the IEEE Computer Society, 2001.

[4] A. Scandurra, G. Falconeri, and B. Jego. Stbus communication system: concepts and definitions, internal document. *STMicroelectronics*, 2:2, 2002.

[5] STMicroelectronics. Stmicroelectronics unveils innovative network-on-chip technology for new system-on-chip interconnect paradigm. December 2005.

[6] Wikipedia. http://en.wikipedia.org/wiki/system-on-a-chip.