The Evolution of Network Topologies and their Usage in Modern-day Multiprocessor Networking

Team 5

David Lindskog - W1033581

Amanda Holl - W0783488

Akashdeep Singh - W1166723

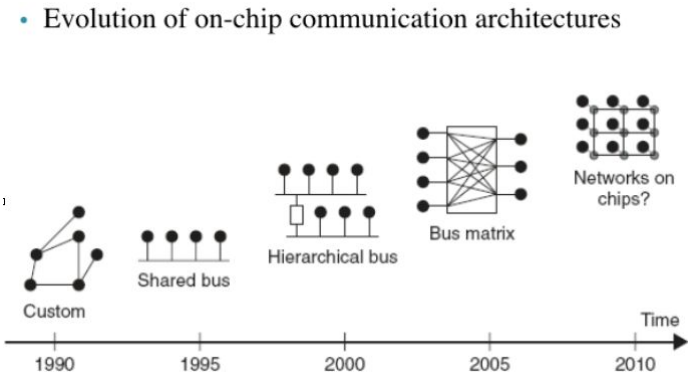
Kedar Hindupur - W1108315

Introduction

To keep up with the needs of increasingly computation-intensive applications and low-power, high-performance systems, the number of computing resources in a single chip has significantly increased, which VLSI technology can support. Although, with the addition of the many computing resources, such as multiprocessors, digital signal processors, memory, input/output

logic control, necessary to build a System-on-Chip (SoC), the interconnection between these resources becomes a challenge.

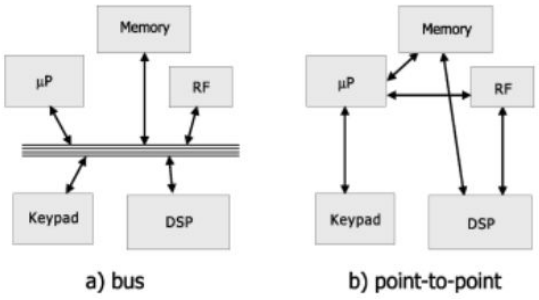
In most SoC applications, a shared bus interconnection, which needs logic to serialize several bus access requests, is adopted for communication between each integrated processing unit since it is low-cost and has simple control characteristics. A shared bus interconnection, however, is limited in its scalability because only one master can utilize the bus at a time, which means that all of the bus accesses should be controlled by an arbitrator. Consequently, in an environment where the number of bus requesters is large, and their required bandwidth for interconnection is more than the current bus, some other interconnection methods should be considered.



Network-on-Chip (NoC) technology can be address this issue, thus it is often called “a front-end solution to a back-end problem.” (Arteris)

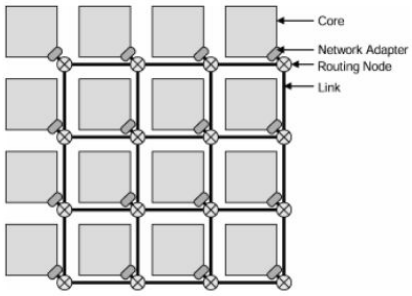
Transfer Signal in System-on-chip and need for NoC

For communication between several cores in an SoC environment, some prevalent mechanisms are bus-based architectures and point-to-point communication methodologies. For simplicity and ease-of-use, bus-based architectures are the most common. However, in a bus-based architecture, there is a fundamental limit in bandwidth, because as the number of components attached to the bus is increased, the physical capacitance on the bus wires grows, as does its wiring delay.



To overcome this limitation and wiring delay, current technological developments and research have been focusing on adopting network-like interconnections, forming what is known as a NoC

architecture. According to NoC white-papers published wi ar advantages over traditional busses, such as improved s e, increased scalability, aggregated bandwidth and easier routi le below. (Guerrier and Greiner)



BUS	Network
Bandwidth is limited and shared by all units attached.	Aggregated bandwidth scales with the network size.
Every unit adds capacitance, therefore electrical performance degrades with growth.	Only point-to-point one-way wires are used, thus local performance is not degraded when scaling.
The arbitration delay grows with the number of masters.	Routing decision are distributed, if the network protocol is made non-central.
The bus arbiter is instance-specific.	The same router may be re-instantiated, for all network sizes.

The success of the NoC design depends on the interfaces between processing elements of the NoC and the interconnection fabric. The interconnection of a SoC established procedure has some weak points, such as slow bus response time, energy limitations, scalability problems, and bandwidth limitations. A bus interconnection composed of a large number of elements in a network interface can cause slow response times when sharing the bus. In addition, the interconnection has the defect that power consumption is high when all objects are connected in communication. Moreover, it is impossible to increase the number of connections between the elements infinitely because of the limitation of bandwidth of a bus. Consequently, the performance of the NoC design relies greatly on the interconnection paradigm.

Topologies used in NoC

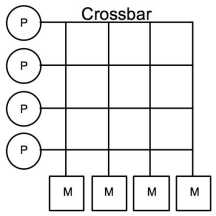
In designing NoC systems, there are several issues to be concerned with, such as topologies, routing algorithms, performance, latency, complexity and so on. Though no one factor can determine an NoC architecture, several different topologies have been proposed in recent years that improve on existing network topologies and their limitations, as will be discussed later. For example, Guerrier and Greiner have proposed a generic interconnect template called SPIN, where a fat-tree architecture is used to interconnect IP blocks. In this fat-tree, every node has

four children and the parent is replicated four times at any level of the tree. Kumar has proposed a mesh-based interconnect architecture called CLICH (Chip-Level Integration of Communicating Heterogeneous Elements).

An Overview of Standard Network Topologies

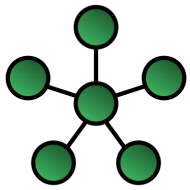
Before exploring in-depth proposed means of designing and improving NoC systems, it is worthwhile to review some standard network topologies that have influenced NoC systems, such as crossbar, star, mesh, and tree topologies. A network topology is the arrangement of the many components, such as the links, nodes, and more, of a network.

Crossbar



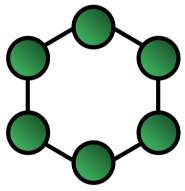
A crossbar, pictured on the right, is a multistage, fully connected network. Each node is connected to all of the other nodes, so messages can be sent between any two nodes in only one pass through the network. A crossbar network topology is useful for nonblocking and high-bandwidth communication, while still being relatively simple, which is why it has been implemented in shared memory multiprocessors. (3.4 Topology)

Star



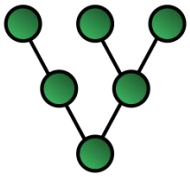
In this topology, each node connects to a central computational resource. It is a common network arrangement, but has a key disadvantage that makes it unstable. The central resource is a single point of failure, since all of the nodes in the network depend on it to function, thus if the central resource fails, all of the nodes are disconnected and the network is down. (Santra and Acharjya 523)

Ring



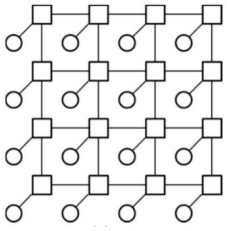
A ring network topology is comprised of a sequence of nodes connected together in a circular formation. As can be seen in the figure, each node is only directly connected to two neighboring nodes, so in network communication for this topology, intermediate nodes pass along messages until the messages arrive at the destination node. (Santra and Acharjya 523)

Tree



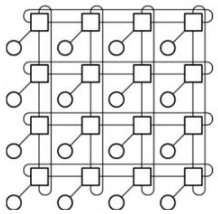
In this type of topology, each node has a coordinate (level, position) where the level is the node's vertical placement in the tree and the position is its horizontal placement from left to right. (Reddy et al. 2) A tree topology functions best when the network is widely distributed and split into numerous branches. (Santra and Acharjya 523) In NoC architectures, a common variation is the Fat Tree topology, where nodes are only linked to leaves in the trees and there are a greater number of links near the root because the bandwidth requirements are higher. (Pasricha and Dutt)

Mesh



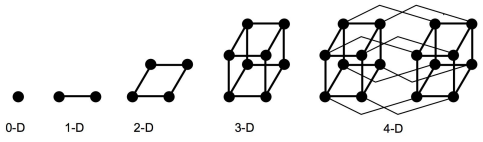
In a mesh NoC topology, pictured on the right, each node, except for those at the edges, is connected by communication channels to other nodes and a computational resource. The mesh topology is common for NoC architectures because it enables many cores to be connected in a regular shape structure. (Reddy et al. 2)

Torus



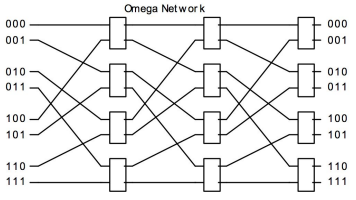
This type of topology is similar to a mesh topology for a NoC system, except the nodes at the edges are connected to the nodes at the opposite edge via folded channels. Each node has a port connecting it to the computational resources and nodes in close proximity. The torus topology is useful when all of the nodes in the system need to have the same characteristics, which the nodes in mesh topology do not have, and when the network needs lower latency. (Reddy et al. 2)

Cube



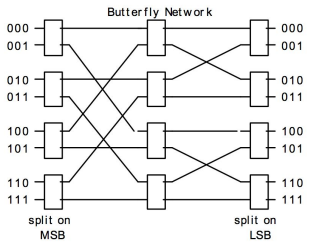
A cube network topology is "An n-dimensional interconnect with 2^n nodes, requiring n links per switch (plus one for the processor) and thus n nearest-neighbor nodes." (Patterson and Hennessy 538) Examples of 0 to 4 dimensional cube network topologies are shown above. This topology has been implemented in parallel processors sold commercially, though in practice it is optimized to increase performance and reliability.

Omega



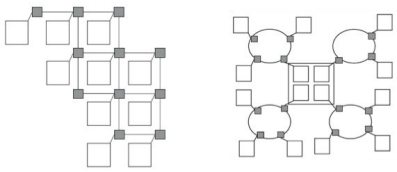
An omega network is a multistage, fully connected network, so each node has a small switch and every processor has a bidirectional connection to all of the other processors. This type of network topology requires less hardware than a crossbar network does because it uses only $2n \log_2 n$ switches, but unlike in a crossbar network, there can be a conflict between messages being passed.

Butterfly



A butterfly network is a logarithmic, multistage network. In a n-dimensional butterfly network topology, there are (n+1)2^n nodes. Each node can be represented by coordinates (x, i) where x is an n-bit binary number and i is the level, between 0 and n, in the network. A butterfly network topology has been implemented in systems with shared memory multiprocessors because in this type of network, all of the processors can connect to different memory resources and thus be active, with no blocked requests. (Lampis)

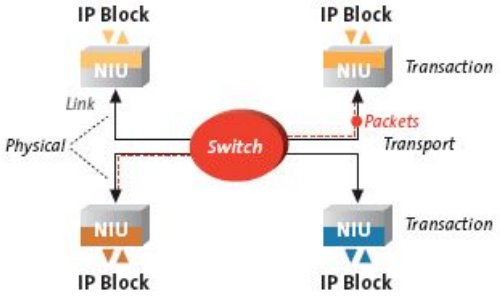
Ad Hoc Network Topologies



These topologies are custom designed for an application. They typically include a combination of topologies such as shared bus, mesh, or crossbar topologies. An example of an ad hoc network topology is a reduced mesh, cluster-based hybrid topology. (Pasricha and Dutt)

Scalability of NoC

Some of the topologies mentioned above, such as ad hoc and butterfly network topologies, have been used in NoC architectures for their scalability. There are two notable design factors that make NoC architectures scale well: a layered architectural approach and overall modularity. The purpose behind the layers is that the functionality of the communication is broken up. Each step of the communication can be handled independently, creating flexibility. The exact number of layers, however, may vary depending on the designing company's preference. We will use Arteris as an example company for one way to set up the layers of a NoC.



Transaction Layer

The first layer is the transaction layer. This layer defines the way that data is exchanged between NIUs (NoC interface units) for a particular transaction. An NIU in this instance is an interface to a particular module in the NoC. Packets are sent out and received through the NIUs. The responsibility of this layer to prepare all of the NIUs for a transaction. This includes translating data from an external protocol to an internal one or defining the primitives that will be used in a transaction.

Transport Layer

The second layer is the transport layer. The responsibility of this layer is to manage the traversal of packets through the switch fabric. By modifying the headers of packets, the transport layer can control the flow of packets, deal with congestion and increase reliability. The transport layer does not need much information about each packet, and therefore, is unaffected by purpose of the data. It only needs to know where the data is going. This adds to the flexibility of the system, allowing for other layers to make changes leaving the transport layer largely unaffected.

Physical Layer

The physical layer is the third layer of a NoC and it contains all of the physical connections and pathways between NIUs and IP blocks. This layer is responsible for def re physically transferred to an interface. As we discussed earlier, there on topologies that can be implemented, each of which has its own strer s. Various link types can coexist within this layer including tran ly asynchronous/locally synchronous links, chip to chip links etc. The isolati es it possible to modify the links being used without affecting the other layers, which adds to the flexibility of the system.

The separation and independence of these layers is what makes the NoC so scalable. Components can be modified on a layer without affecting the other layers or components around it. Each layer has its own job to do and there is no cross over or shared responsibility between any of the layers.

Energy Efficiency of NoC

As stated earlier, the number of processor cores on a single chip continues to grow. Because of this, the communication subsystem between the cores is quickly becoming the largest factor when it comes to the power consumption of a chip. There are a few factors that come into play with the energy efficiency of a chip: dynamic power consumption and static power consumption.

Dynamic/Static Power Consumption

Dynamic power consumption is affected by the distance data must travel from one point to another. The longer the wire that data must travel over, the more power dissipation there is. Because NoCs are switch based and wired in a highly point to point manner, wire length is minimized. This lowers the necessary load capacitance when sending data and consequently reduces the amount of dynamic power the system consumes. The point to point nature also allows for easy calculation of quiet data pathways, and using quieter paths reduces the amount of power the switches consume. Static power consumption is affected by, and is roughly proportional to, the overall area of silicon the chip covers. The switched, point to point format of NoC allows for a much smaller system area. This is true in general, although the area is affected by the network topology implemented.

Power Saving Techniques

Power can be managed by intelligent selection of pathways through which packets are sent from one block to another, minimizing switching activity. This same idea can be applied even more liberally by partitioning the subsystem and physically shutting down parts of the network that are not being used much. Parts of the subsystems may also be clocked individually and each part kept at its own lowest locally possible clock rates. These techniques are all possible through the highly modular design of the NoC. Because they are closely tied to application requirements, however, quantifying actual energy efficiency improvement over other architectures is difficult.

Advancements in the Design of Network Topologies and NoC Architectures

Modern NoC architectures and multiprocessor network topologies, such as the Optical Network-on-Chip and hybrid topologies, have focused on improving scalability and incorporating the power saving techniques described above.

Optical Network-on-Chip

Optical Network-on-Chip (ONoC) is a new type of NoC for multiprocessor SoC. While traditional NoC relies on electrical signals to transfer information, and is thus called electrical network-on-chip (ENoC), its performance and energy efficiency are bound by the significantly unbalanced scaling of on-chip global metal wires compared to transistors. Optical telecommunications have been successful in many networking domains in replacing electrical telecommunications. Riding on the achievements of photonic technologies, a wide range of studies have been done on ONoC.

However, according to the paper published by the reference stated below, some researchers claim that the optical network on chips are not very efficient when a large number of IP's are connected. "Interconnecting a large number of IPs using optical networks is infeasible due to the current technological constraints. We propose a methodology that enables ONoC designers to overcome this challenge by dividing a large ONoC into multi-ONoCs. At the acceptable price of extra waveguides and electrical routing, the proposed method reduces the number of nOSC, making it possible to interconnect large number of IPs. In our future work, we will evaluate the benefits of this methodology to different topologies, such as ring." (Le Beux et al)

Hybrid Topology

Hybrid topologies are commonly used in Wide Area Networks because, true to the name, they are a combination of two or more different topologies and thus have a far more flexible approach to meeting the requirements of any particular organization or user. A hybrid topology can be designed to achieve the best of the two or more topologies and thereby cater to the specific needs of the user. For example, if there exists a ring topology in one office department, while a bus topology in another department, connecting these two will result in hybrid topology. It is important to note that connecting two similar topologies cannot be termed a hybrid topology. Star-ring and star-bus networks are the most common examples of a hybrid network.

Advantages of Hybrid Network Topology

1. *Reliable*: Unlike other networks, fault detection and troubleshooting is easy. The faulty part can be isolated and rectified without affecting the network.
2. *Scalable*: Its easy to increase the size of network by adding new components.
3. *Flexible*: A hybrid network can be designed with respect to the requirements of the client and by optimizing the available resources.

4. *Effective*: A hybrid topology is the combination of two or more topologies, so we can design it in such a way that strengths of constituent topologies are maximized while their weaknesses are minimized.

Disadvantages of Hybrid Topology

1. *Complexity of Design:* It is not easy to design this type of architecture.
2. *Costly Hub:* The hubs used to connect two distinct networks, are very expensive. These hubs are different from usual hubs as they need to be intelligent enough to work with different architectures and should be function even if a part of network is down.
3. *Costly Infrastructure:* As hybrid architectures are usually larger in scale, they require a lot of cables, cooling systems, sophisticate network devices, etc.

Conclusion

NoC architectures, implemented to enable communication between the many components on a chip, have evolved in recent years to provide better scalability and energy efficiency. These architectures adapt the network topologies, such as torus and ad-hoc topologies, used in traditional network design to provide better communication than what is provided with a standard bus architecture. Modern NoCs use a layered architecture and modularity to increase scalability and reduce the energy consumed through data traversal and switching activity, an improvement over shared bus architectures. Energy saving techniques are made possible by the flexibility of NoC. Researchers and developers have been focused on employing these techniques, in addition to making other advancements, to improve the communication between components in a NoC. The Optical Network-on-Chip and hybrid topologies are examples of how NoC architectures and multiprocessor network topologies are continuing to evolve to address the needs of contemporary computer architectures.

The introduction to NoCs and multiprocessor network topologies that we put together for this paper allowed us to explore an area of computer architecture that we had little knowledge of before. We learned that many of the NoC architectures and multiprocessor network topologies being researched are primarily theoretical and have a lot of potential for success, but few have been implemented in systems used today. It was, however, challenging to fully understand these topics, let alone clearly summarize this broad field given its highly-technical nature. Overall, we

all have gained a better understanding of how the communication between chip components has and will continue to evolve over time.

References

1. "3.4 Topology." *Oak Ridge National Laboratory.* Web. http://www.phy.ornl.gov/csep/ca/node22.html.
2. "A Comparison of Network-on-Chip and Busses." *Design & Reuse*. 1 Jan. 2005.
3. Bahn, Jun Ho. "Overview of Network-on-Chip." Web. http://gram.eng.uci.edu/comp.arch/lab/NoCOverview.htm.
4. Lampis, Michael. "The Butterfly, Cube-Connected-Cycles and Benes Networks." *NTUA*. Web. http://www.corelab.ece.ntua.gr/courses/parallel.postgrad/Butterflies.pdf.
5. P. Guerrier and A. Greiner, A Generic Architecture for On-Chip Packet-Switched Interconnections, Proc. Design and Test in Europe, pp. 250-256, Mar. 2000.
6. Pasricha, Sudeep, and Nikil Dutt. "Networks-on-Chip." *On-chip Communication Architectures System on Chip Interconnect*. Amsterdam: Elsevier / Morgan Kaufmann, 2008. Print.
7. Patterson, David A., and John L. Hennessy. "Parallel Processors from Client to Cloud." *Computer Organization and Design: The Hardware/Software Interface.* Fifth ed. Morgan Kaufmann, 2014. Print.
8. Reddy, Tetala Neel Kamal, Ayas Kanta Swain, Jayant Kumar Singh, and Kamala Kanta Mahapatra. "Performance Assessment of Different Network-On-Chip Topologies." *International Conference on Devices, Circuits and Systems (ICDCS)* 2 (2014): 1-5. Print.
9. "Routing Congestion." *Arteris*. Web. 12 Mar. 2015. http://www.arteris.com/routing_congestion.
10. S. Kumar et al., A Network on Chip Architecture and Design Methodology, Proc. Intl Symp. VLSI (ISVLSI), pp. 117-124, 2002.

11. Santra, Santanu, and Pinaki Pratim Acharjya. “A Study And Analysis on Computer Network Topology For Data Communication.” *International Journal of Emerging Technology and Advanced Engineering* 3.1 (2013): 522-25. Print.
12. Sebastien Le Beux, Jelena Trajkovic, Ian O’Connor, Gabriela Nicolescu, Guy Bois, et al.. “MultiOptical Network-on-Chip for Large Scale MPSoC.” IEEE embedded systems letters, IEEE, 2010