Letter No.V-34564,Reg.533/2007-2008 INDIAN JOURNAL OF RESEARCH(2012)6,
ANVIKSHIKI ISSN 0973-9777 Advance Access publication 20 July. 2012

NOC DESIGN DECISIONS FOR EACH OSI LAYER

CHANDRA SHEKHAR

Research Scholar, Singhania University, Pacheri Bari, Jhunjhunu, Rajasthan

Declaration

*The Declaration of the author for publication of Research Paper in The Indian Journal of Research Anvikshiki ISSN 0973-9777 Bi-monthly International Journal of all Research: I **CHANDRA SHEKHAR,** the author of the research paper entitled **NOC DESIGN DECISIONS FOR EACH OSI LAYER** declare that, I take the responsibility of the content and material of my paper as I myself have written it and also have read the manuscript of my paper carefully. Also, I hereby give my consent to publish my paper in Anvikshiki journal, This research paper is my original work and no part of it or it's similar version is published or has been sent for publication anywhere else. I authorise the Editorial Board of the Journal to modify and edit the manuscript. I also give my consent to the Editor of Anvikshiki Journal to own the copyright of my research.*

Abstract

On-chip communication architectures offer a vast design space. Designing a complete network thus appears as a very complex and difficult problem.
A network is indeed characterized by a wide set of parameters (topology, routing algorithm etc.). Designing a network thus simply consists in performing decisions on the value of each parameter, usually to optimize a cost function.
To be able to deal with the complexity of the design problem, it should ideally be split into smaller independent sub-problems. Each sub-problem consists in taking a design decision.
The order in which independent decisions are taken does not matter. Dependent decisions, on the other hand, simply cannot be performed separately. Real design decisions are however generally neither completely independent nor fully dependent of each others. In this paper we throw the light on decisions for each OSI layer.

1. INTRODUCTION

OSI divides telecommunication into seven layers. The layers are in two groups [9]. The upper four layers are used whenever a message passes from or to a user. The lower three layers are used when any message passes through the host computer. Messages intended for this computer pass to the upper layers. Messages destined for some other host are not passed up to the upper layers but are forwarded to another host.
The on-chip communication architecture [1][3][4] characteristics have been split following the OSI stack model in which networks can be viewed at different abstraction layers.
The seven layers of the OSI network model can be identified in an on-chip communication architecture: application, presentation, transport, network, link and physical layers.

2. APPLICATION LAYER DECISIONS

The application layer offers high-level communication services to the IP cores. High-level communication services are then translated in communication primitives based on the architecture of the lower layers.

* Research Scholar, Singhania University, Pacheri Bari, Jhunjhunu, Rajasthan

Decisions that can be performed at the application layer mainly concern the choice of the service offered to the applications.
Distributed Inter-Process Communication (IPC) is a well known subject studied for long in the context of distributed network computing and multi- tasking operating systems. The specificity of the on-chip communication context is that the network interfaces have to deal with both software and hardware tasks. Just as for software IPs, hardware components thus have to be separated from the network by a wrapper which allows the hardware component to be unaware of the presence of a network.
Different inter-process communication models co-exist in the distributed computing context.

Inter-process communication styles
There are basically two types of inter-process communication styles:
• the shared memory programming model
• the message passing programming model

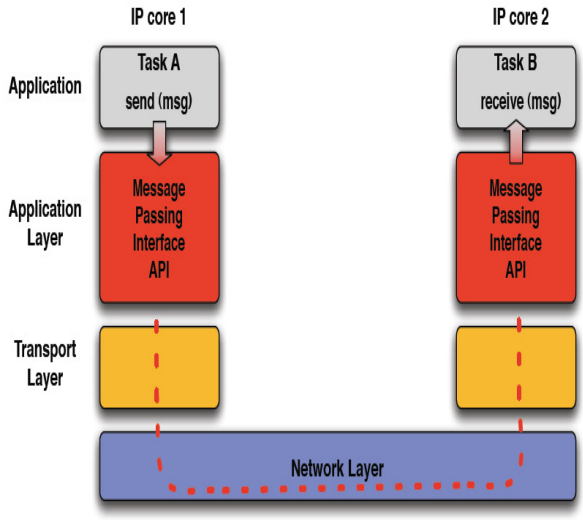

Figure 1: Inter-process communication between tasks A and B respectively located on IP cores 1 and 2. Task A is explicitly sending the message “msg” to task B through the “send” API provided by the application layer. The message is then sent through the transport layer implemented in IP Core 1’s network interface and then through the network layer to the IP Core 2. The later explicitly receives the message from IP core 2 through the “receive” communication API.

2. 1 Quality of Service

QoS consists in providing a certain level of guarantees on given communication parameters (generally on bandwidth and latency) that a communication architecture can offer to an IP core.
In the on-chip communication context, the goal of the QoS is to provide communication guarantees to the IP cores in order to tackle the intrinsic unpredictability of the on-chip network communications.

Quality of service generally consists of:

Guaranteed Service (GS) is a performance high predictability service (usually required for real-time systems) based on deterministic values of the network parameters. The only way to

implement this kind of service is to logically decouple the GS traffic from the traffic of other services[8].

Statistical Guaranteed Service is the kind of service generally provided by computer networks. It is based on statistical values of the QoS parameter.

Best Effort (BE) is a service for which no performance guarantees can be given. However, the BE service generally includes a correctness service and transaction completion commitment. It aims at optimizing the average network resources usage.

The concept of Guaranteed Service (GS) implies that a connection has to be established between the sender and the receiver. The IP cores can negotiate the guaranteed services of their connection through their network interfaces.

Guaranteed services includes:
• correctness
• completion
• performance bounds (latency, bandwidth, etc.)

The application layer only defines the quality of services. The actual implementation will be performed by the lower-layers parameters.
The application layer provides an interface to the application to negotiate Quality of Service. The application task asks for a certain QoS requirement to the network application layer. The application layer transmits those requirements to the transport layer which has a more detailed view of the end-to-end network communications. The transport layer itself interrogates the network layer to ensure that the resources required by the application task are available. If resources are available, the resources are reserved in the network and a reservation acknowledgment is propagated back to the application layer. If resources are not available, the QoS must be re-negotiated with the application till the required QoS and available QoS match.

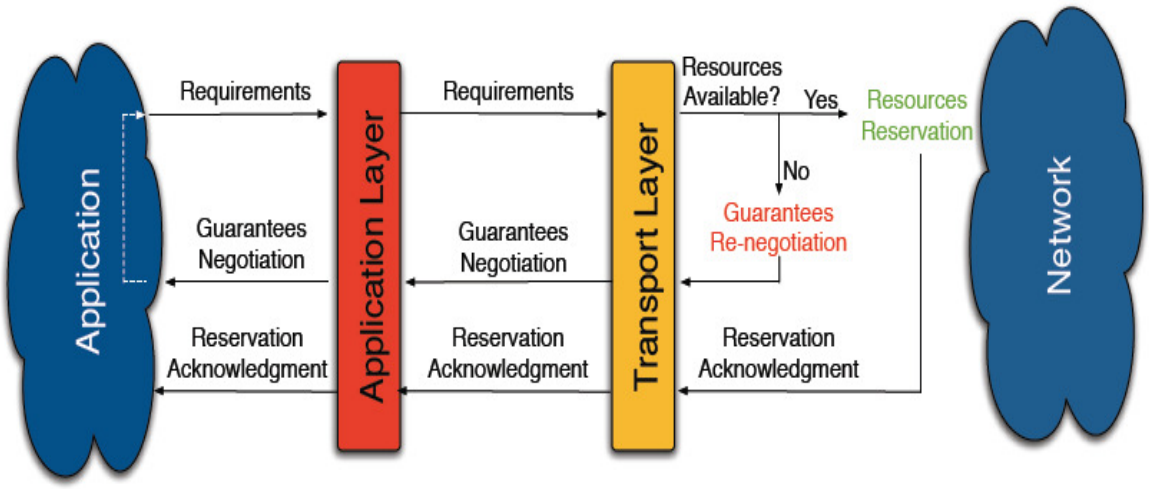

Figure 2 : Quality of Service Negotiation: the application layer provides an interface to the application tasks to negotiate the Quality of Service

3. TRANSPORT LAYER DECISIONS

The transport layer offers message transport services to the application layer. The length of a message can be variable and is not necessarily known in advance.
The transport layer is based on protocols that can guarantee the end-to- end transport of data from source to destination. The lower-level network layer will implement the actual decisions that have to be performed to convey data in the network.
The transport layer describes the high-level communication services that can be provided by the communication architecture. It mainly includes the end-to-end connection management and flow control services.

3.1 Impact of topology choice on the network cost function

This section briefly expose the impact of the topology choice on network performance and performance guarantees, physical lay-out, network through- put, reliability and energy consumption.
Impact of topology on performance and performance guarantees
For small size systems (16 nodes), the minimum latency is offered by low- dimension networks. For bigger systems (16 to 256), higher dimension

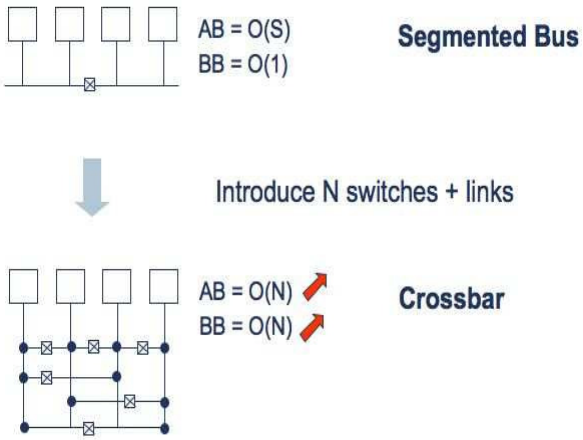

Figure 3: From a segmented bus to a crossbar
Networks are performing better. This is the motivation for networks exploiting octagonal or fat-tree based topologies (SPIN and Octagon).

Broadcasting is easier on topologies that exploits lots of shared-medium (e.g. buses). Topologies that offer limited sharing of links need to copy data resulting in high aggregated throughput.

3.2 Impact of topology on physical lay-out

Some topologies are difficult to map on a 2D floorplan as they lead to many links crossing. Topologies like the 2D mesh, on the other hand, are particularly easy to implant and lay-out on a chip due to its 2D-structure. This explains why most communication architectures proposed by the academic community and the industry are based on this topology.

3.3 Impact of topology on network throughput

The topology has a direct impact on the aggregated bandwidth of the net- work. The

aggregated bandwidth of a bus is fixed for any number of nodes while the aggregated bandwidth of a crossbar grows in $O(N^2)$ with the number of nodes.

3.4 Impact of topology on reliability

Some topologies are more fault-tolerant than others depending on the communication concurrency/redundancy they offer. A bus is for example very sensitive to faults while a distributed communication architecture like a Network-on-Chip is much more reliable.

3.5 Impact of topology on energy consumption

High dimension networks embedded in the plane lead to longer wire lengths for the inter-router physical channels and therefore are generally less energy- efficient than low-dimension networks.

4. LINK LAYER DECISIONS

Link layer decisions are related to the way data are sent between the communication resources[5].
Important link layer decisions concern the choice of:

- the data forwarding technique
- the arbitration
- the multiplexing technique
- the data link encoding technique

4. 1 Data Forwarding Technique

The link layer most important decision concerns the manner that the data are forwarded on the network (once the routing decision has been per- formed).
The network literature generally describes data forwarding technique as one unique network design decision (usually referred to as switching technique).

- the distance between control information and data
- the buffering
- the flow control granularity

4.1.1 Distance between control information and data

The “distance” that separates control information from data is also one important parameter of the link layer decision. The distance between control and data can be spatial or temporal.
Spatial distance simply consists in providing separate physical channels for data and control. As we will see in chapter 4, this is generally the case for many shared-bus architectures. On the contrary, control and data can be multiplexed on the same channel. This is generally the case for many packet-based architectures and for some bus-based architectures. Moreover, nothing prevents buses from using packet-based scheme in which control information is regrouped in a header that is sent along with the data over the bus.
Temporal distance is the duration separating the injection of the control information in the communication architecture and the injection of the corresponding data (which corresponds to a number of hops or physical distance in the network due to propagation). This parameter affects the run-time adaptivity of the path taken by the

data. More distance generally allow more flexibility.

Pipelined bus transactions

A bus transaction consists of a set of successive phases. To increase the bus efficiency, phases of consecutive transactions which use separated physical signals can be treated in parallel in a pipeline-way.

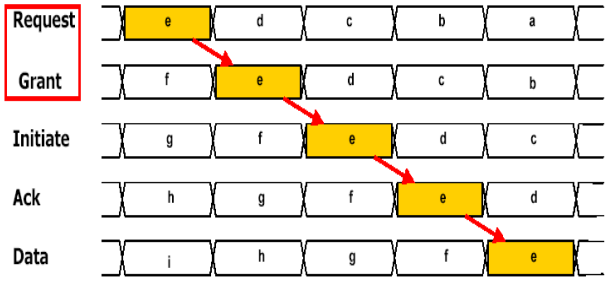

Figure 4: Pipelined bus transactions permits a considerable increase of the bus efficiency

Split transactions

Split transactions scheme consists of considering request and response as two independent parts of the bus transaction. They do not have to be consecutive anymore. That is particularly efficient as resources do not have to be blocked by wait states. Slaves send a signal when they are ready to put their request on the bus.

4.1.2 Buffering

The role of buffers is to temporarily store data in the network to solve potential contention issues in the access of a shared medium (bus, crossbar,...) without loosing data.

Buffer size

The buffer sizing decision depends on the flow control granularity. Indeed, the buffer size cannot be smaller than the flow control granularity.
The buffer sizing decision results of a trade-off between avoiding network traffic contention and the cost in terms of area, latency and energy.
The buffer size affects:

- the maximum router clock frequency and thus the maximum band- width per port
- the possibility of unresolved contention and blocking in the network and thus also reduces the available bandwidth
- the router power consumption

The buffer size also has a considerable impact on the network overhead area. In ×pipes, doubling the buffer size from 3 to 6 flits affects the area overhead by 54% and reduces the maximum frequency by 6%.

Buffer position

Buffers can be placed at different positions relatively to the switching element:

Input Queuing: Input Queuing consists in placing the queues in front of the switching fabric. The control scheme is simple to implement but the main problem is the Head-of-Line (HoL) problem. When the first element of an input queue is blocked, all the following elements are also blocked even if their corresponding outputs are available. This limits the utilization of the switch.

Output Queuing:- Output Queuing consists in placing the queues after the switching fabric. The output queuing is also simple to implement. However, the switching fabric must be clocked faster than for input queueing to deal with the case that all inputs are going to the same output (N times faster if N is the number of inputs).

Central Queuing:- Central Queuing consists in storing data in a large central buffer. Switches are placed in front and behind the central buffer.
Rarely used in on-chip communication architecture because it requires a large and fast multi-port memory to implement the central buffer which is costly both in terms of energy consumption and area overhead.

Virtual Output Queuing (VOQ):- Virtual Output Queuing (VOQ) consists in placing queues in front of the switching fabric .The difference with input queuing is that each input is composed of a set of buffers, one queue per possible output. This solves the Head of Line Blocking problem of the Input Queuing technique.

4.1.3 Flow control granularity

The flow control describes how the data flow is adjusted between communicating devices. Data are generally sent on the communication architectures in multiple times as links bit-width is limited. However, the flow control can be performed at several levels (packet, flit, phit etc.). The flow control granularity defines the granularity at which data are sent through communication devices.
Flow/congestion control strategies at the link level describe the transfer of data from a communication device input port to an output port. The communication device can be a switch or a bus.

Data forwarding decisions applied to shared-buses

Bus burst transfers
To better amortize the cost of bus arbitration (at least 2 cycles per transfer), high performance buses can grant bus accesses to a master for multiple cy- cles, allowing to transfer multiple words per transaction. This is particularly efficient for DMA accesses and block transfers.

Data forwarding decisions applied to networks : the switching techniques
When applied to networks, different combinations of flow control granularity, control-data distance, buffer corresponds to well-known switching techniques such as:

- real circuit switching
- store and forward
- virtual cut through
- wormhole
- mad postman
- switched virtual circuit

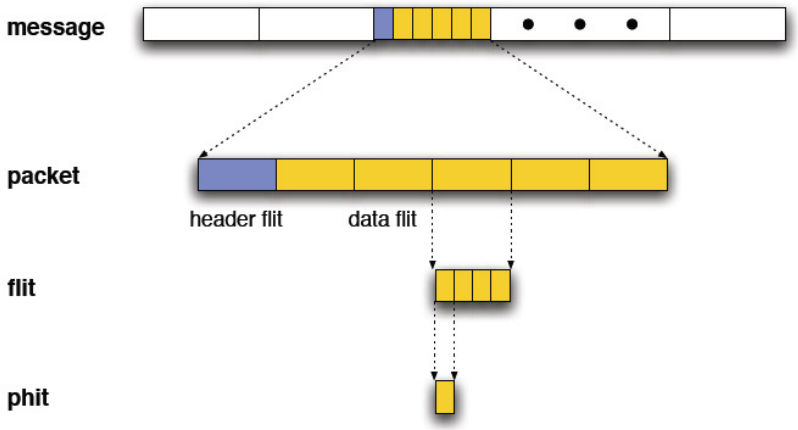

Figure 5: A message can be split into packets. A packet can be itself split into flits (flow control unit). Physically, a flit can be sent on the network as a sequence of phits which size corresponds to the links bit width

Those switching techniques [6] specify the way a message is forwarded on the network but not which path along the network the message should follow which is the role of the routing technique. The switching technique choice considerably affects the network delay. For each technique, the analytical expression of the delay for an L-bit message with W bits phits (=flits), D links between source and destination will be given. tr will designate the routing decision delay; ts , the switching delay and tw , the wire delay, inverse of the bandwidth. The header is assumed to be 1 flit long.

Impact of the transport-layer connection mode on the choice of the switching technique

The choice of the network connection mode influences the choice of the switching technique. Connection oriented mode at the network layer impose the usage of pure circuit-switched or virtual-circuit switched technique.

Complete switching technique design space

Following Figure presents the switching technique design space.

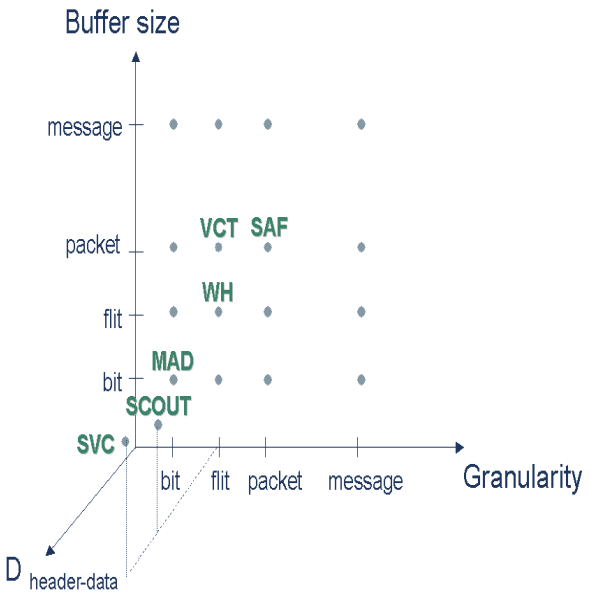

Figure 6: Switching technique design space

Switching technique	Latency	Buffering	Hardware Complexity
Store and Forward	- -	-	+
Wormhole	++	+	- -
Virtual Cut Through	++	-	-
Switched Virtual Circuit	+	++	- -

Table. 1: Comparison of switching techniques

4.2 Arbitration

The arbitration decision defines the contention resolution technique i.e. how possible conflicts between different data that have to access common network resources (e.g. buses, crossbar,etc.). Arbitration technique is influenced by the presence of buffers in the communication architecture.
The arbitration can be performed for the access to the complete on-chip communication architecture (e.g. for shared-buses) or per communication device (e.g. to have access to one router port).

One important characteristic of the arbitration process is its fairness. The fairness is the ability to provide equal access to different requesters of the same priority.

5. PHYSICAL LAYER DECISIONS

The physical layer's options that are presented in this section are only concerning the upper layers of the interconnect stack as on-chip inter-tile communication architectures are implemented by those layers.

Many solutions have been proposed to tackle the problems inherent to DSM technologies, mainly focusing on optimization power, delay and crosstalk.
As DSM problems are critical, optimization efforts at all degrees of abstraction levels must contribute for obtaining an efficient solution.

For long, only delay optimizations and a bit dynamic energy consumption optimization were the major concerns. Few publications are dealing with static energy consumption reduction but the interest is growing on that topic as the microelectronic community realizes the major impact it could have.
Dynamic energy consumption mainly consists in the energy required to switch the equivalent capacitance Ceq of a line.
The dynamic power consumption can be basically modeled by following equation.

$$P_{dyn} = \frac{1}{2}\alpha f_{clk} C_{eq} V_{dd} V_{swing}$$

As V_{dd} is fixed by the technology node and V_{swing} cannot be reduced much further in future technologies for noise immunity reasons, only two parameters can influence significantly the dynamic power consumption: the activity □, the clock frequency f_{clk} and the equivalent switching capacitance C_{eq}.

The goal is not to get performances at any cost but given performances at the lowest energy-cost. Many publications claim that as technology scales down to deep sub-micron era, global on-chip interconnect will be the most power consuming part of the chip.

Therefore, a serious study of the global interconnect optimizations from an energy point of view is required.

The choice of a relevant cost function is important for any optimization. For interconnect, when delay is critical, the usual figure of merit that is considered is RC. When power is also an important issue, the figure of merit chosen is power$*$RC. When power considerations dominate, like in our case, the figure of merit that is commonly adopted is power$^2 * RC$. Considering only dynamic power consumption, the figure of merit that will be adopted here is:

power$^2 * RC \rightarrow (\alpha f_{clk} V_{dd} V_{swing})^2 RC$ [3]

6. CONCLUSION

The on-chip communication architecture characteristics have been split following the OSI stack model in which networks can be viewed at different abstraction layers.

Some decisions will have to be performed first to guarantee that the decisions with the highest cost impact are performed first. Upper layers decisions will thus impose constraints on the lower layers values of decisions to optimize a given cost function. Within the same layer, decisions may also have to be performed in a specific order to permit constraints propagations from a higher to a lower priority decision for the cost function that is optimized.

We have classified the various network design decisions according to their qualitative impact on various costs (area overhead, energy consumption, end-to-end latency and reliability).

Although the impact of many design decisions on the costs is generally well covered in the literature, we have highlighted the lack of a thorough study of the multiplexing technique design decision.

7. REFERENCES

[1]. DAVIDE BERTOZZI AND LUCA BENINI. XPIPES: A Network-on-Chip Architecture for Gigascale Systems-on-Chip. IEEE Circuits and Systems Magazine, 4,2004.

[2]. D. BERTOZZI, L. BENINI, AND G. DE MICHELI. Error Control Schemes for On- chip Communication Links: the energy-reliability trade-off. IEEE Transac- tions on CAD, 24(6):818–831, 2005

[3]. P. BHOJWANI, R. SINGHAL, G. CHOI, AND R. MAHAPATRA. Forward error cor- rection for on-chip networks. In Proc. of Workshop for Unique Chips and Systems (UCAS-2). March 2006.

[4]. BAS BIJLSMA. Asynchronous Network-on-Chip Architecture Performance Analysis. Master's thesis, Department of Electrical Engineering, Delft University of Technology, 2005.

[5]. AXEL JANTSCH, ROBERT LAUTER, AND ARSENI VITKOWSKI. Power analysis of link level and end-to-end data protection on networks on chip. In Proceedings of the IEEE International Symposium on Circuits and Systems. 2005.

[6]. CLAUDIA KRETZSCHMAR, ANDR´E K. NIEUWLAND, AND DIETMAR MU¨LLER. Why Transition Coding for Power Minimization of On-Chip Buses Does Not Work. In DATE, pages 512–517. 2004.

[7]. FERNANDO MORAES, NEY CALAZANS, ALINE MELLO, LEANDRO MOLLER, AND LUCIANO OST. HERMES: an infrastructure for low area overhead packet-switching networks on chip. Integr. VLSI J., 38(1):69–93, 2004.

[8]. ANDREI RADULESCU, JOHN DIELISSEN, SANTIAGO GONZALEZ PESTANA, OM PRAKASH GANGWAL, EDWIN RIJPKEMA, PAUL WIELAGE, AND KEES GOOSSENS. An Efficient On-Chip Network Interface Offering Guaranteed Services, Shared-Memory Abstraction, and Flexible Network Program- ming. IEEE Transactions on CAD of Integrated Circuits and Systems, 24(1):4–17, January 2005.

[9]. VERSITA, WARSAW ,Applying of security mechanisms to middle and high layers of OSI/ISO network model ,Theoretical and Applied Informatics, Vol 24, Number 1 , Pp- 95-106, 2012