ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Irregular Network-on-Chip Architectures: System-level exploration and CAD tools

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΙΑΣΩΝ-ΙΩΑΝΝΗΣ Κ. ΦΙΛΙΠΠΟΠΟΥΛΟΣ

Επιβλέπων: Δημήτριος Σούντρης
Επ.Καθηγητής Ε.Μ.Π.

Αθήνα, Φεβρουάριος 2010

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Irregular Network-on-Chip Architectures: System-level exploration and CAD tools

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΙΑΣΩΝ-ΙΩΑΝΝΗΣ Κ. ΦΙΛΙΠΠΟΠΟΥΛΟΣ

Επιβλέπων: Δημήτριος Σούντρης
Αν.Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την …. Φεβρουαρίου 2010.

....	....	....
Δημήτριος Σούντρης	Κιαμάλ Ζ. Πεκμεστζή	Γεώργιος Οικονομάκος
Επ. Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.	Λέκτορας Ε.Μ.Π.

.

Αθήνα, Φεβρουάριος 2010

....

ΙΑΣΩΝ-ΙΩΑΝΝΗΣ Κ. ΦΙΛΙΠΠΟΠΟΥΛΟΣ

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Περίληψη

Αντικείμενο της παρούσας διπλωματικής εργασίας αποτελεί η ανάπτυξη ενός εργαλείου προσομοίωσης, το οποίο με την κατάλληλη μεθοδολογία προσομοιώσεων θα δίνει την δυνατότητα για βελτιστοποίηση της κίνησης των πακέτων σε τοπολογίες δικτύων σε ψηφίδα (network-on-chip). Η βελτιστοποίηση επικεντρώνεται στην ταχύτητα και την κατανάλωση ισχύος και η μεθοδολογία επικεντρώνεται στον εντοπισμό βέλτιστης τοπολογίας από ένα γενικευμένο σύνολο μη-κανονικών τοπολογιών (irregular topologies).

Στο κεφάλαιο 1, παρουσιάζονται τα βασικά χαρακτηριστικά και ο τρόπος λειτουργίας, καθώς και μια πλειάδα ήδη υλοποιημένων τεχνικών για την επικοινωνία σε δίκτυα σε ψηφίδα. Η γνώση αυτών των χαρακτηριστικών θα οδηγήσει στην ανάπτυξη κατάλληλων μεθόδων και τεχνικών για προσομοίωση αυτών των συστημάτων.

Στο κεφάλαιο 2, μελετάται η ροή σχεδιασμού (design flow) που χρησιμοποιείται για δίκτυα σε ψηφίδα. Αναλύονται τα διάφορα στάδια της ροής από την ανάλυση των εφαρμογών που θα εκτελεστούν στο δίκτυο, μέχρι την υλοποίηση της πλατφόρμας στο φυσικό επίπεδο. Έμφαση δίνεται στα εργαλεία προσομοίωσης που απλοποιούν τον σχεδιασμό και ειδικότερα στην μοντελοποίηση στο επίπεδο συστήματος, στο οποίο βρίσκει εφαρμογή αυτή η εργασία.

Στο κεφάλαιο 3, παρουσιάζεται ο προσομοιωτής ο οποίος αναπτύχθηκε για τις ανάγκες αυτής της εργασίας. Γίνεται αναφορά των χαρακτηριστικών και των δυνατοτήτων του, καθώς και του τρόπου λειτουργίας του.

Στο κεφάλαιο 4, περιλαμβάνεται μια σειρά προσομοιώσεων πραγματικών εφαρμογών DSP, οι οποίες μελετώνται και υλοποιούνται σε ενσωματωμένα συστήματα.

Τέλος, στο Κεφάλαιο 5, παρουσιάζονται τα συμπεράσματα της εργασίας, καθώς και κάποια θέματα και ιδέες για διερεύνηση και μελλοντική έρευνα.

Λέξεις Κλειδιά

Δίκτυα-σε-Ψηφίδα, επίπεδο συστήματος, ροή σχεδιασμού, εφαρμογές επεξεργασίας ψηφιακού σήματος

Abstract

The purpose of the present diploma thesis is the development of a simulation tool that along with a systematic design methodology will perform traffic optimization on network-on-chip topologies. The key metrics are delay and power consumption on the network and simulation results aim to reveal the optimum irregular topology for each traffic scheme.

In Chapter 1, we make an introduction to the basic properties and the function of Network-on-Chip. Knowledge presented in this chapter is necessary in order to implement specific methods and techniques for simulation of NoC.

In Chapter 2, there is a presentation of design flow for NoC. There is a detailed description of each design stage from application mapping to synthesis and validation of the system. We focus on simulation tools that simplify various stages of the design flow and justify the need for the simulation tool developed in this thesis.

In Chapter 3, there is an in depth analysis on the simulation tool implemented. We present its features and extensions, which developed during this work, and also its functionality.

In Chapter 4, we present a large number of experimental results based on real applications used in embedded systems.

Last, in Chapter 5, we present the findings of this work along with some topics and ideas that need future research and study.

Keywords

Network-on-chip (NoC), system level, design flow, application mapping, irregular topologies exploration, buffer sizing, energy optimization

Ευχαριστίες/Acknowledgments

Για την εκπόνηση της παρούσας διπλωματικής εργασίας θα ήθελα να ευχαριστήσω τον καθηγητή κ. Δ. Σούντρη για την αμέριστη συμπαράστασή του, την εμπιστοσύνη του και τις πάντοτε καίριες και χρήσιμες συμβουλές του που με καθοδήγησαν στη μελέτη αυτή. Επίσης, η εργασία αυτή δεν θα είχε έλθει εις πέρας χωρίς την πολύτιμη συμβολή και τις γνώσεις και τις παρεμβάσεις του υποψήφιου διδάκτορα Ηρακλή Αναγνωστόπουλου και του διδάκτορα Αλέξη Μπάρτζα που βοήθησαν να ξεπεραστούν όσα σημαντικά προβλήματα προέκυψαν κατά τη διάρκεια της διπλωματικής.

Table of Contents

Chapter 1. Network on Chip, NoC

Chapter 2. Current NoC Design Flow

Chapter 3. CAD Tool Noxim++

Chapter 4. Simulation results

Chapter 1

Network on Chip, NoC

1.1 Introduction

The scope of this chapter is to give a brief review of techniques, referring to on-chip and off-chip communication, implemented over the years. The evolution of these trends justifies the use of the communication approach introduced by Network-on-chip (NoC) architectures.

According to the Moore's law, the number of transistors that can be placed inexpensively on an integrated circuit has doubled approximately every two years. Although originally calculated as a doubling every year, Moore later refined the period to two years and the actual increase, as shown in Figure 1, is very close to predicted values.

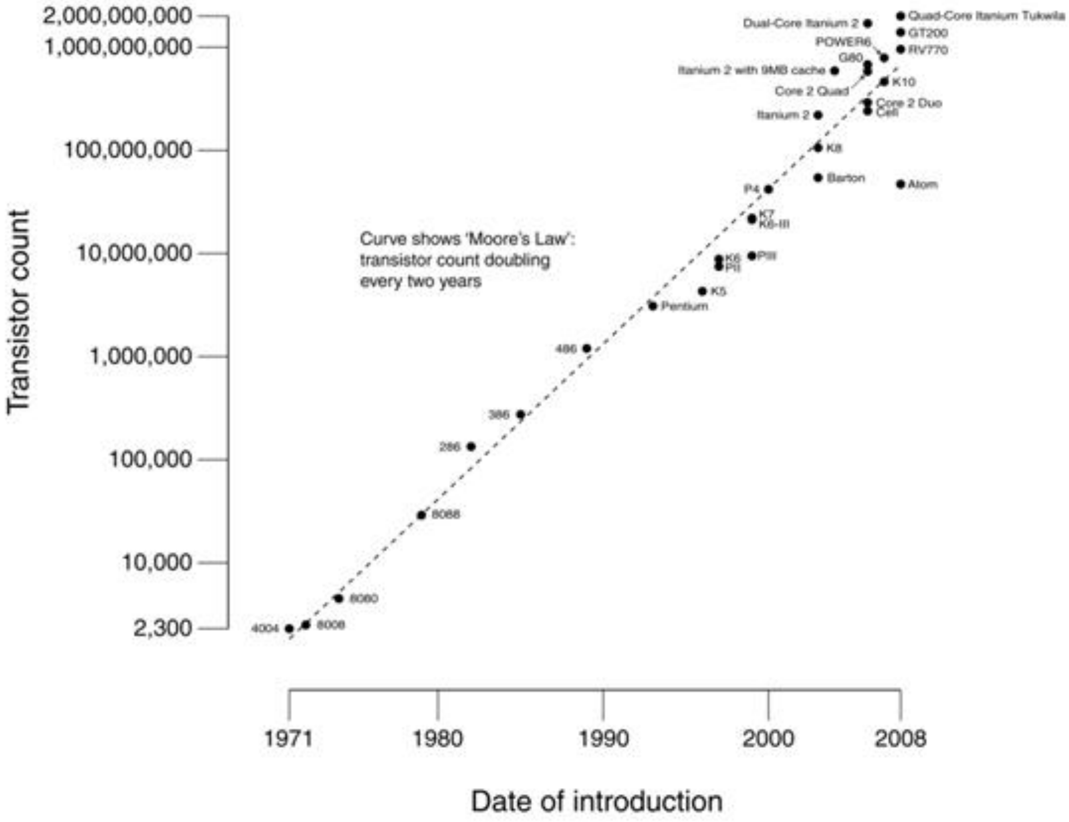


Figure 1. CPU Transistor count 1971-2008

Even though this rate is projected to slow down during the next decade, the available processing power will continue growing exponentially (BJE, 2006). In addition, the scale of system integration becomes bigger and bigger, from LSI at 70’s to VLSI at 80’s and from ULSI at 90’s to multimillion transistor chips, including more than 100M transistors, nowadays. As a result, a chip is becoming capable of including more components and executing more complex tasks. So, the technology moves from LSI and VLSI systems. In LSI systems, a chip was a component of a system module (e.g., a bitslice in a bitslice processor), in VLSI systems, a chip was a system-level module (e.g., a processor or a memory), and in ULSI systems, a chip constitutes an entire system (hence the term System-on-Chip).

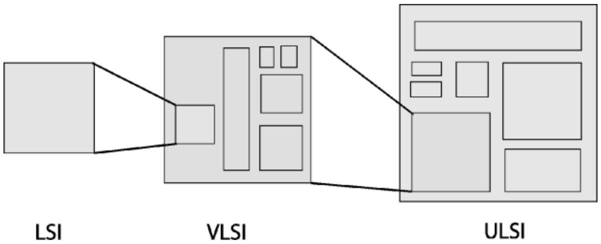


Figure 2. Evolution of technologies (BJE, 2006)

1.2 SoC, MPSoC

Nowadays there is a growing demand for new devices, that will be able to fulfill on-board objectives, and thus more complex chips with lots of components will be implemented. Most of these devices are based on a System-on-chip (SoC). For example, it is essential for a modern mobile phone to offer Bluetooth and PDA functionality, mp3 decoder, video camera with a good resolution and other services. Indeed, modern mobile phones already have up to eight processors, including one or several Reduced Instruction Set Computer (RISC) processors for user-interfaces, protocol stack processing, a Digital Signal Processor (DSP) for voice processing; an audio processor for music playback; a picture processor for still image camera functionality and even video processors for new video on- phone functionality (MAR, 2007). Implementations with many processors like the above are often called multiprocessor System-on-Chip (MPSoC).

SoC platforms are superior to previous technologies in terms of a) performance, b) power consumption and c) reliability. Better performance is achieved by combining the power of many general use processors and application specific processors, like DSP. By dividing a task to many units, faster computation can be achieved without designing new processors that have to work on higher frequencies, which is a hard obstacle to overcome. Low-power consumption is achieved by operating SoCs at low voltage, in combination with voltage (and frequency) scaling and gating (MIC, 2009). Of course, low frequency will result to performance drop, so multiprocessing is essential also. In addition, MPSoC can offer better reliability, because a crash on one processor does not result on a system crash. Also, MPSoC reliability is achieved by redundancy, which is the provision of multiple interchangeable components to perform a single function in order to cope with failures and errors. For example, different processors can be used to execute a task and then check that both came up with the same result.

1.3 Network-on-chip

In order to exploit the advantages of SoC mentioned above, communication between all the components of the chip is crucial. Although, the primary objective for the first SoC designers was the improvement of computation, as the number of components and their performance continue to increase, the design of the communication network architecture becomes essential in order to achieve the desired performance, and energy consumption of the overall system (MAR, 2009). As a result, the Network-on-Chip (NoC) approach was introduced to handle communication aspects of chip design, due to its better scalability, flexibility, energy efficiency, testability and differentiated services (MAR, 2007).

The most common communication structures in SoCs, are shown in Figure 3. SoC systems are traditionally architected around shared buses, ad-hoc p and point-to-point interconnects. The network approach is the evolution of buses and point-to-point links for on-chip communication. Bus based architectures were abandoned for complex designs mainly because of the delay factor (bus lead to a bottleneck when many components are connected) and the big amount of energy consumption. As more units are added to the system, the power usage per communication event grows as well due to more attached units leading to higher capacitive load.. Point-to-point solution is not viable for chips with many components, because the number of connections lead to a great waste of space and energy on chip.

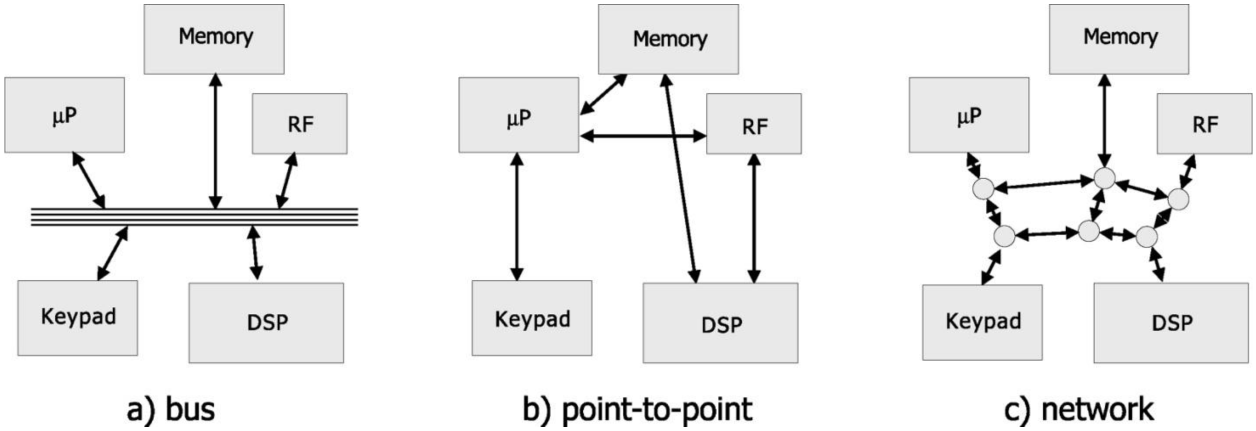
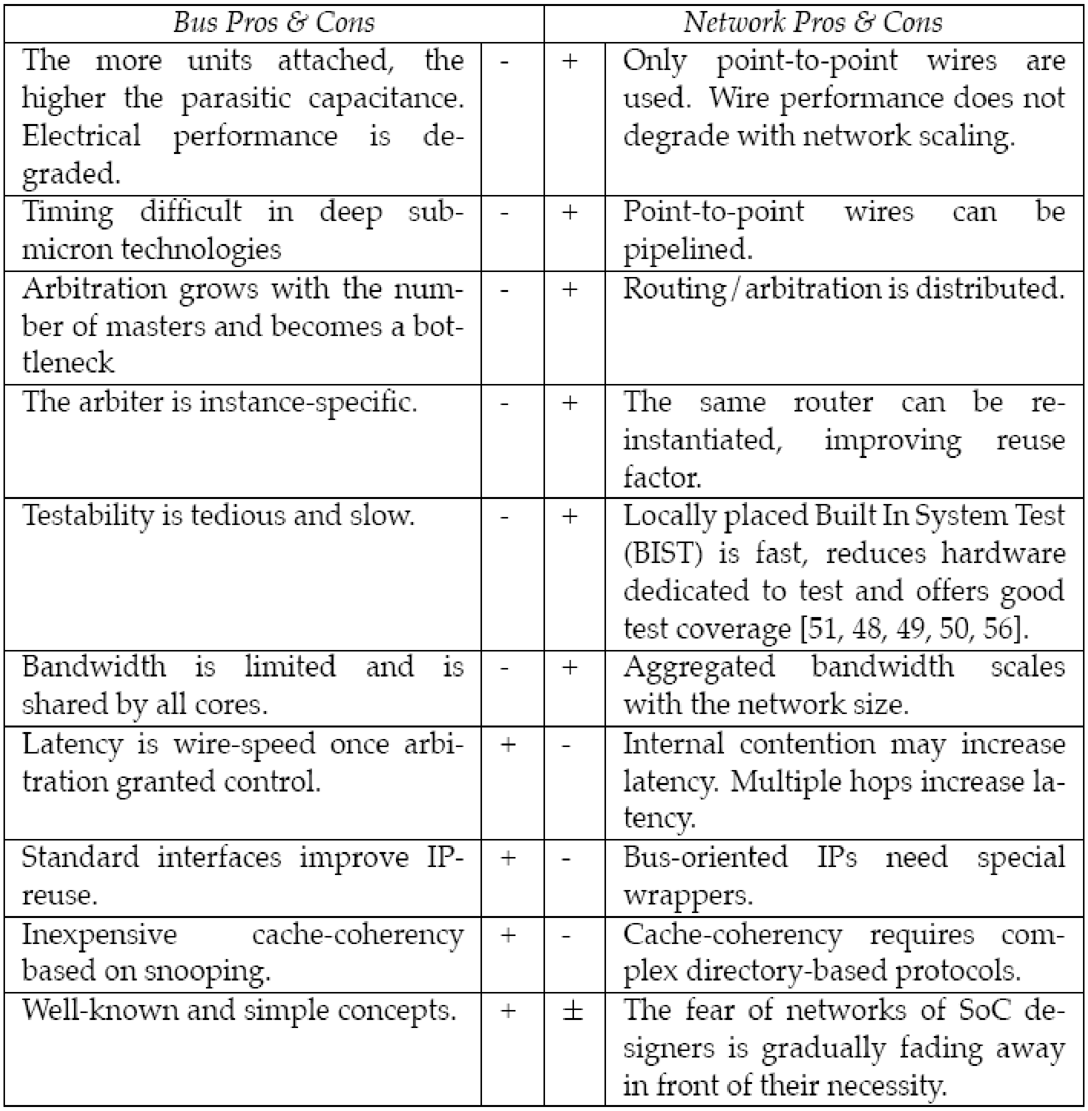


Figure 3. Common communication structures in SoC (BJE, 2006)

On the other hand, network approach achieves better performance for many cores, because connections between components are relatively fast for any size of chip, assuming a few hops between components. A NoC can be characterized by several parameters such as topology, network protocol, structure, and control of a router. Instead of busses and dedicated point-to-point links, a more general scheme is adapted, employing a grid of routing nodes spread out across the chip, connected by communication links. Network adapters (NI) implement the interface by which cores connect to the NoC. Their function is to decouple computation (the cores) from communication (the network). Advantages and disadvantages of these two communication approaches are listed in Table 1. Overall, network communication seems more suitable, because of the growth of on-chip components, but designers have to tackle disadvantages like complex memory coherency.



Bus Pros & Cons		*Network Pros & Cons*	
The more units attached, the higher the parasitic capacitance. Electrical performance is degraded.	-	+	Only point-to-point wires are used. Wire performance does not degrade with network scaling.
Timing difficult in deep submicron technologies	-	+	Point-to-point wires can be pipelined.
Arbitration grows with the number of masters and becomes a bottleneck	-	+	Routing/arbitration is distributed.
The arbiter is instance-specific.	-	+	The same router can be reinstantiated, improving reuse factor.
Testability is tedious and slow.	-	+	Locally placed Built In System Test (BIST) is fast, reduces hardware dedicated to test and offers good test coverage [51, 48, 49, 50, 56].
Bandwidth is limited and is shared by all cores.	-	+	Aggregated bandwidth scales with the network size.
Latency is wire-speed once arbitration granted control.	+	-	Internal contention may increase latency. Multiple hops increase latency.
Standard interfaces improve IP-reuse.	+	-	Bus-oriented IPs need special wrappers.
Inexpensive cache-coherency based on snooping.	+	-	Cache-coherency requires complex directory-based protocols.
Well-known and simple concepts.	+	±	The fear of networks of SoC designers is gradually fading away in front of their necessity.

Table 1. Comparison between bus and network communication (MAR, 2007).

A simple NoC with focus on a tile, which consist of a router with buffers and one processing element with its own memory, is shown in Figure 4 (MAR, 2009).

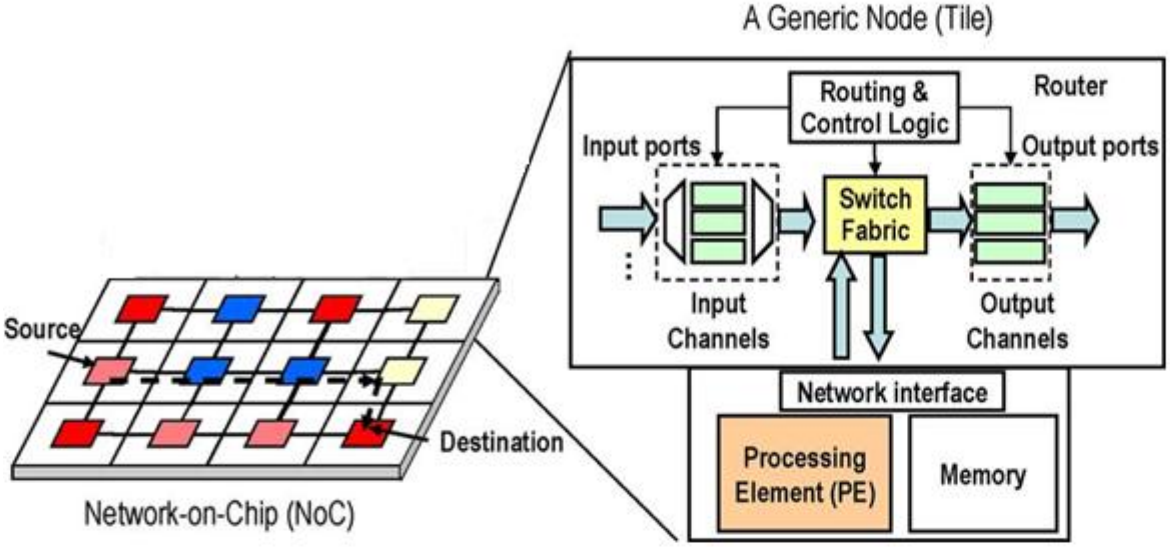


Figure 4. Network-on-chip and structure of its tile (MAR, 2009)

The major parts that participate in communication are the network interface, the router with all of its components and the links. Network interface, as shown in the figure, implements the communication between one intellectual property (IP), here IP consists of a processor and a memory, and the router. Router, whose main components are the switch and the buffers, implements the communication between different tiles. Links are point-to-point wires that connect routers. In this simple the NoC each tile has only one IP and links to its neighbors, but this work focuses on more complex designs.

Each router in a mesh is connected to its four neighboring routers via a bi-directional channel and an embedded core is attached to the router. The core communicates with the router through NI parsing or by making packet headers. The header contains information like the destination or origin of the packet. In wormhole switching, a packet is broken up into flits and they are transported in a pipelined manner. Based on the OSI (Open System Interconnection) model used for computer networks. The communication procedure in NoCs can be divided to 4 layers, according to T. Bjerregaard and S. Mahadevan (BJE, 2006), and following there is a brief description for each one of them.

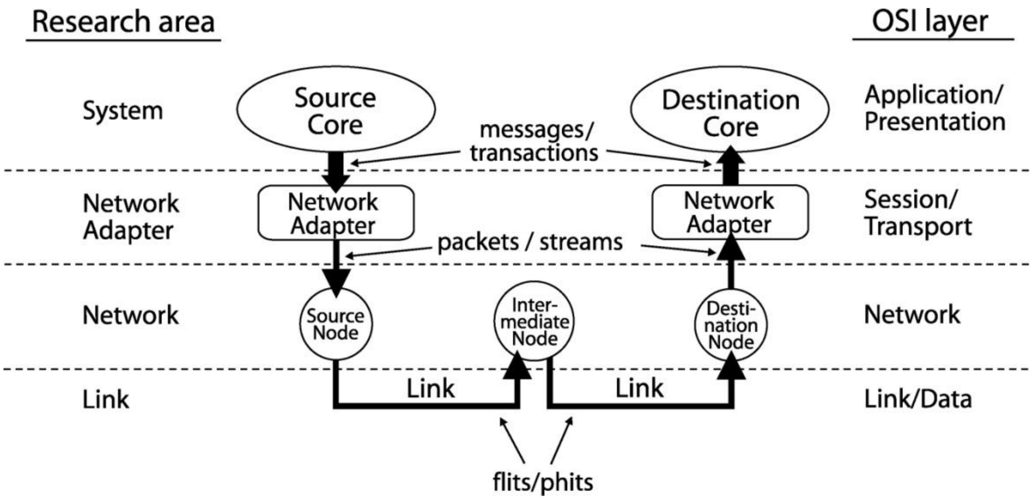


Figure 5. Communication layers on NoC (BJE, 2006)

1.3.1 System Layer

The system layer is the Application Program Interface (API) that allows communication between application messages that have to be transmitted between nodes. In this high level of communication the data format of message use whatever form is desired by the application that execute at the processor at that time. The system layer is not aware of the underlying network hardware and the next levels are responsible to produce packets that comply with the computation and communication rules for the network hardware. A bridge is used in order to produce suitable forms of data for network interface layer. Also, the bridge should be aware of the programming model of the platform, which may be message-passing, shared memory or a combination, and adapt to it (MAR, 2007).

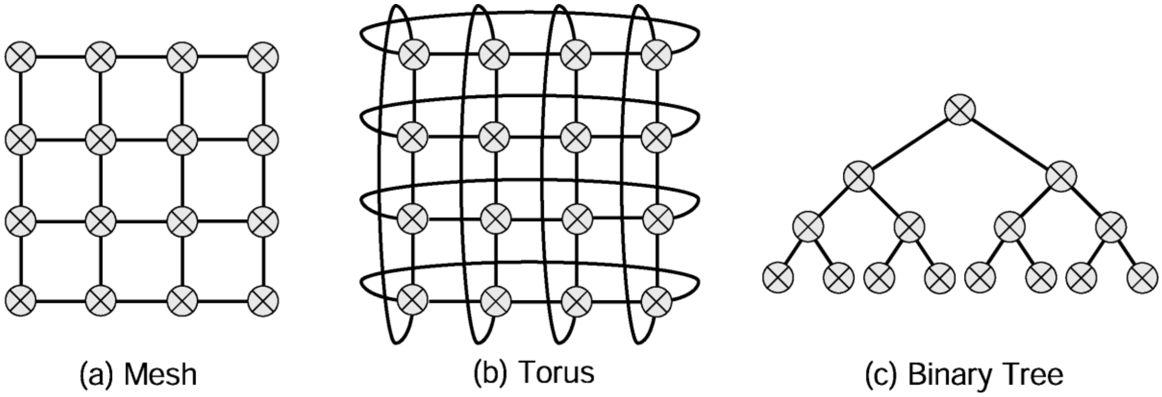
1.3.2 Network Interface Layer

The Network Interface (NI) isolates the core and the network, which is essential, because the processor uses a faster clock than the network. Moreover, the data produced by the core have to be encapsulated by the source network interface and decapsulated by the destination network interface before passed to the destination core. Furthermore, network adapter is also responsible for breaking the message to pieces and monitor the whole transmission process. Generally, a message from previous layer or a packet of this layer is transmitted as a sequence of flits. The first one is called header and the last one is called tail. Information about source, destination and network protocol are included in the header and are needed for the NI to choose the transmission path. Also, the network interface layer is responsible for establishing permanent connection between nodes for the whole transaction, although connection-less communication is in this work. Different levels for quality of services (QoS) depend on choices for transmission and retransmission strategies in this level.

1.3.3 Network Layer

The network is dedicated to delivering messages from source node to destination node (BJE, 2006). The most important factors that have to be defined in network layer are the topology of the network and the transmission protocol that it uses.

Topology determines the connections between nodes on chip. The most common regular topology is the mesh, in which every node connects to its closest neighbors. Other regular topologies are shown in Figure 6. **Examples of regular topologies**



(a) Mesh

(b) Torus

(c) Binary Tree

Figure 6. Examples of regular topologies (BJE, 2006)

The next thing needed for the right operation the network layer is the protocol. The protocol determines the path that data should follow, from source to destination node. Designer of the network protocol can choose between: (a) dedicated paths for each route (connection oriented) or dynamic forwarding from source to destination (connectionless) (b) routing of packets according to predefined routing strategies (deterministic routing) or dynamic routing by checking the availability of links in time of transmission (adaptive routing) (c) control of data flow, that can be done locally on each node or globally by one or more dedicated nodes (d) shortest path routing or not (e) delayed arrival or rejection of a packet after a number of hops and others. In this work, the topology chosen is irregular and minimal routing with connectionless links and locally control, with local routing tables in every node, is implemented.

1.3.4 Link Layer

The link layer consist of the point-to-point wires between tiles and it implements the necessary circuit-level and physical details from the higher layers of the NoC (MAR, 2007). In this layer the communication is achieved through flits, which mentioned before. Link layer aims to reduce energy consumption, through mechanisms like wire driving, bus encoding and serialization, and speed up transmission, for example with use of Globally Asynchronous Locally Synchronous (GALS) approach. In addition, link layer should always check if a transmission can be operated, by ensuring that buffers have free slots, wires are connected etc.

1.4 Irregular NoC

The evolution from bus-based off-chip communication to ad-hoc on-chip communication led to better performance and lower power consumption, as discussed previously. Regular NoC topologies are more suitable for general-purposed on-chip multiprocessors, which consist mostly of homogeneous set of processing and storage arrays and can benefit from spatial locality to achieve higher performance (MIC, 2009). However, there are NoCs manufactured for very specific applications or a group of applications and use many heterogeneous components to fulfill their specific tasks. These heterogeneous components, like processors, controllers, DSPs, and hardware accelerators, can cooperate better in irregular topologies and using special protocols to maximize their specific communication needs. Irregular topologies can take advantage of application’s specific characteristics. Some cases in which irregular topologies are superior to regular in terms of performance are a) the Aethereal architecture (KEE, 2005), b) the

BONE series of chips (LEE, 2003), c) the SPIN network (Scalable Programmable Integrated Network) (ADR, 2003) and others. Also, the Xpipes Compiler (JAL, 2004) gives an optimum solution for three video applications that is based on irregular NoCs.

Irregular NoC term describes a free topology, in which each node, including a router and one or more IPs, is possible to have a link with as many nodes as desired by the designer. Irregular topologies may be the combination of some regular topologies, because regular topologies have been well studied and so modified versions of these can easily boost performance. Designers have put great research effort in regular NoC topologies and a lot of problems have been solved in a very efficient way. Of course, designer should be aware of the processing task that SoC have to execute and study carefully the communication scheme of the application, before choose NoC architecture. In other words, the topology is application depended. The topology of the system is chosen to fit well with the application it runs, as it is illustrated in the next chapter. Two examples of irregular NoC topologies are shown in Figure 7.

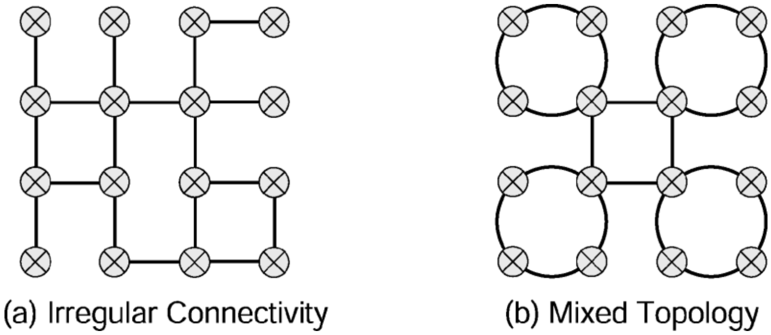


Figure 7. Examples of irregular NoC topologies (BJE, 2006)

Chapter 2

Current NoC Design Flow

2.1 Introduction

Building and designing an MPSoC system is a quite challenging task, as it involves lots of different processing elements that need to cooperate in an efficient way. Also, careful decisions need to be taken in all four network layers described in chapter 1. Moreover, a great number of engineers with a quite wide spectrum of expertise have to cooperate simultaneously in different stages of the final product, such as high level programming for applications running on the SoC, and low level communication links. Furthermore, there is a great market demand for new products and new designs are developed at a high rate. So, the design flow is crucial for every research and development department and it has to ensure that complex embedded systems will work as expected and will be produced according to the time schedule so as to maximize the time-to-money factor

In this chapter, there is a description of design flow for NoCs, divided in three broad sections according to Marculescu et al. (MAR, 2009), as shown in Figure 8.

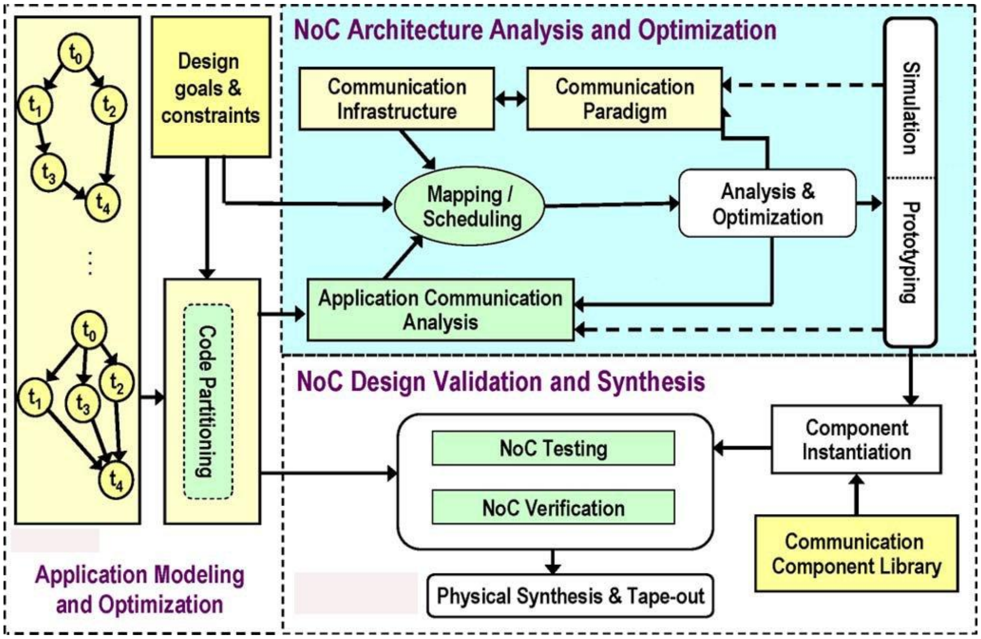

NoC Architecture Analysis and Optimization
Design goals & constraints
Communication Infrastructure
Communication Paradigm
Simulation
Mapping / Scheduling
Analysis & Optimization
Prototyping
Application Communication Analysis
Code Partitioning
t_0
t_1
t_2
t_3
t_4
NoC Design Validation and Synthesis
Component Instantiation
NoC Testing
NoC Verification
Communication Component Library
Physical Synthesis & Tape-out
Application Modeling and Optimization

Figure 8. Design flow for NoCs (MAR, 2009)

It is important to mention here that software design tools are crucial in exploring and testing the proposed solutions-algorithms. Obviously, there is great demand for CAD tools and right now there are enough tools only for regular NoC topologies. Clearly there are more research fields that need careful attention and suck kind of software tools are the cornerstone to the research. The following analysis belongs to design technologies for NoC, but designers should always keep in mind the limitations that arise from available manufacturing technologies.

2.2 State of the art NoCs

Before design flow analysis, there is a presentation of some state of the art NoC implementations, in this chapter.

STNoC (COP, 2004) is a packet-switched NoC, which uses deterministic routing, wormhole switching and output buffering. QoS is provided by a fair bandwidth allocation scheme and STNoC is based on an interesting (patented) choral ring topology Spidergon topology used by STNoC (© ST Microelectronics)(Figure 9), that is deemed to be a good trade-off between performances and area/energy cost for practical SoCs (BON, 2006).

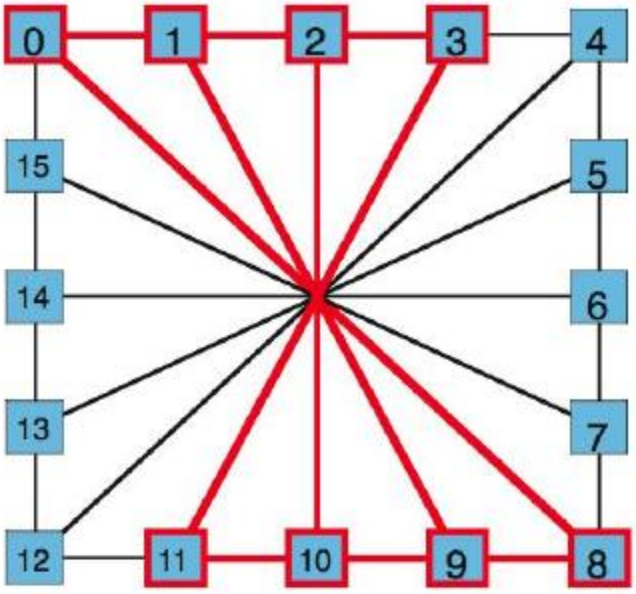


Figure 9. Spidergon topology used by STNoC (© ST Microelectronics)(COP, 2004)

SPIN (Scalable Programmable Interconnect Network) (GUE, 2000) is an academic packet-switched NoC, which was one of the very first NoCs that implemented in hardware. SPIN uses a fat-tree topology (32-bit links) to minimize the network diameter (Figure 10), Virtual Cut Through (VCT) switching and 4-phits flits for deeper pipelining and Virtual Component Interfaces (VCI) to connect IPs and its scope is to reduce latency on the network.

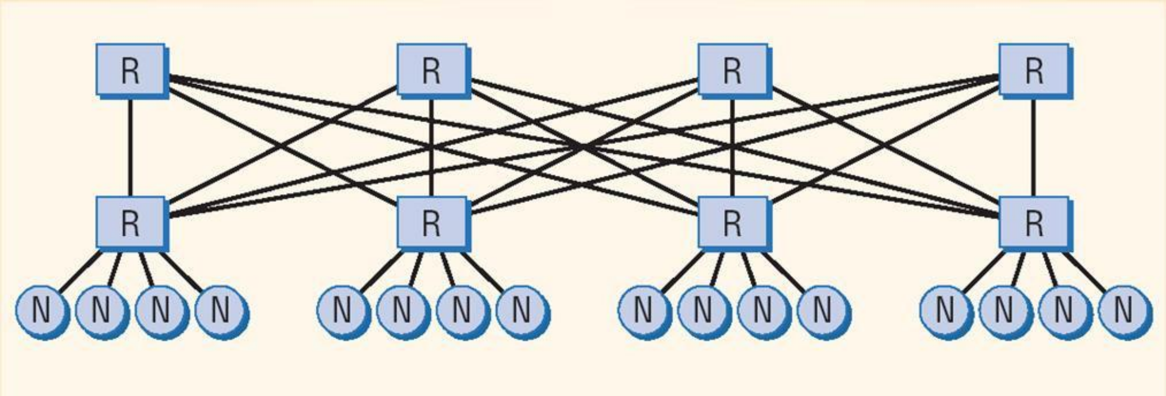


Figure 10. Fat-tree topology of SPIN (© Guerrier)

MANGO (BJE, 2005) (Message-passing Asynchronous NoC providing Guaranteed services over Open Cores Protocol-OCP interfaces) is a clockless asynchronous NoC that mixes a best-effort (BE) router and a guaranteed services (GS) router to provide both best-effort and real-time guarantees. Connection-oriented GS are provided by reserving virtual channels through the NoC and connection-less BE services are used to setup GS connections at run-time. Network adapters provide OCP-based standard socket interfaces and synchronize the clocked OCP interfaces to the clockless network Figure 11.

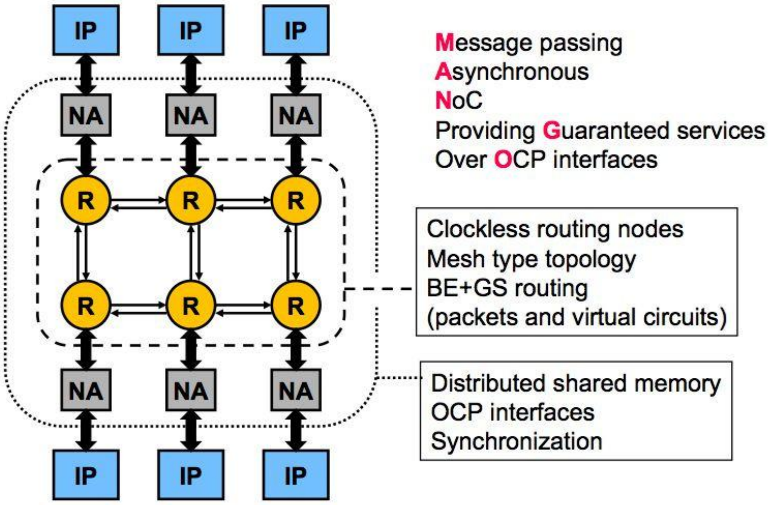


Figure 11. The MANGO communication architecture (© Bjerregaard)

Nostrum mesh architecture built by KTH, is a packet-switched NoC using hot-potato switching that offers connection-less best-effort services and connection-oriented guaranteed throughput and latency (MIL, 2004). Hot-potato is a deflection routing algorithm that upon congestion may choose to deflect packets from their optimal (shortest path) path in the network, but also drop packets that loose arbitration. Signals are used from routers a) to notify their neighbors of congestion ahead and b) to inform that they drop a packet.

Æthereal (GOO, 2005) by NXP Research is a NoC architecture that supports both best-effort (BE) and guaranteed services (GS) QoS. GS guarantee hard real-time bounds on both throughput and latency. Also, Æthereal uses deterministic routing and wormhole switching and guarantees in-order packet delivery without packet dropping. Buffering is done in custom-made hardware FIFOs to optimize the router area. Several versions of Æthereal have been developed over the years. Another innovative feature of Æthereal is its automated design flow that allows either high-level (TLM) SystemC simulation or VHDL generation of a NoC for custom MP-SoC platforms (DIE, 2005). The input to the design flow is an XML description of the network requirements, that contains a description of the modules and their traffic requirements in terms of bandwidth, latency and QoS. NoC topology, node mapping, traffic balancing and time-slot table generation is automated thanks to the UMARS (Unified MApping, Routing and Slot allocation) design-flow (HAN, 2005).

2.3 Application Modeling and Optimization

According to Figure 4 application modeling and optimization is the first stage in the design flow and it focuses on applications' high level analysis of. A careful and precise analysis of applications' characteristics and mainly the communication bandwidth needed by them is essential in order to make a rough plan about the design space and components needed by it. So, in this stage the decisions determine the number of processors, memories and other components and also provides a basic idea about the topology that will be used.

Full application mapping is a difficult task, especially for applications that have tasks executed by many components, as those for multicore chips. Benchmarks and task graphs are used to offer good results in short time. Traffic modeling can be assumed to be a common distribution, like random, bit reversal or butterfly distribution, or special benchmarks, like

SPLASH. The objective is to stress the NoC and find an efficiency topology, without conjunction points and with an acceptable trade-off between minimum delay and energy consumption. Finding a corresponding traffic model will simplify the whole design process by choosing an optimum topology at early stage of design flow.

2.4 NoC Architecture analysis, optimization and evaluation

In this stage of the design flow, the simulation tool attempts to optimize at design-time the use of the communication resources available from decisions made in the previous section (MAR, 2007). The goal is to choose an efficient routing algorithm and a flow control mechanism, including specifications of the routers and their number of connections with PEs that will ensure that the on-chip traffic will be handled, according to the desired specifications. The key performance metrics here are the average and maximum packet latency, the throughput of the network and the communication bandwidth, while important cost metrics are power consumption of communication on network and its overhead in the overall consumption(MAR, 2009).

The differences between simulation and testing results in this stage are mainly ought to limited communication virtualization and estimations of the behavior of the application and the platform performance. Analysis cannot be precisely, because simulation software tools use simplified models for links, routers and other components. However, optimum solutions can be found by comparing the results for different communication strategies. In the end of this stage of the flow, designers make their choice on network topology and routing algorithm, according to the simulation results. Thus, a reliable CAD tool and elegant and powerful power/delay models are crucial for building the communication platform, which has a huge impact on design costs, power and performance (MAR, 2009). In other words, the main goal of this step is to have a clear view of the platform's metrics and characteristics. According to the result the designer uses this feedback in order to trigger some more features.

The CAD tool described to chapter 3 belongs to this stage of the design flow. The motivation of this work is the development of a CAD tool that will be useful in finding optimum communication solutions, including topology and routing parameters, for a given system and a given communication graph for it. Assuming that application's modeling and optimization has been done, so that the components of the system are well-defined and the traffic model for their communication is available, the scope of the software simulation tool is the exploration of possible topologies and the choice of the optimum one.

2.5 NoC Design Validation and Synthesis

The final part of design flow involves validation and synthesis of the NoC platform which will result in a prototype system. The final-prototyped design, which strongly depends on choices made on previous stages, has to be tested. Behavior in testing will show, if the results are acceptable or changes have to be done. Of course, major changes will be time consuming and costly, reasoning why design choices in early stages are so important. The certain thing is that the feedback from testing gives us a lot of important information about the whole design phase. Based on that feedback changes may be made not only in the final stage but also to the previous ones. In conclusion, CAD tools are necessary in NoC analysis, because their simulation results provide valuable knowledge to designers, leading them to reasonable decisions that will result to effective NoC designs in short time.

Chapter 3

CAD tool

3.1 Introduction

In chapter 2 the problem of designing NoCs from high-level specifications, while incorporating with network specialization and optimization of hardware components and protocols, was illustrated. CAD tools are used in all stages of the design flow to simplify the process. This chapter describes a CAD tool designed for communication in NoC. A simplified version of a NoC tool flow is shown in Figure 12.

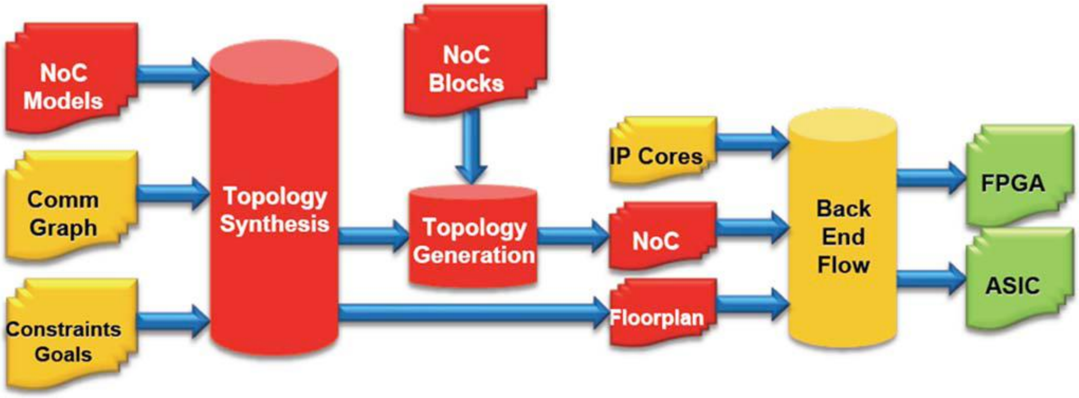


Figure 12. Generic NoC tool flow (MIC, 2009)

In Figure 12, the combination of NoC models, application’s communication graph, relevant constraints and desired goals, leads to topology synthesis. By introducing NoC blocks and other components designers can use a Field-Programmable Gate Array (FPGA) or/and an Application-Specific Integrated Circuit (ASIC) to test their system. Noxim++ is a SystemC fully customizable tool that provides information about the performance of our platform in terms of energy and delay.

3.2Programming language and specifications

The objective of almost every CAD tool, including Noxim++ simulation tool, is to produce fast simulation results with acceptable accuracy. The chosen programming language for this tool is C++ and its special library SystemC, which extends the capabilities of C++ by enabling modeling of hardware description. C++ together with SystemC combines system level design and RTL models, in a way chosen by the programmer. So, low level of abstraction is used to describe important aspects of simulation, for example the communication links, and high level of abstraction is used for describing general specifications , for example operation of the processors. The parts of code written in usual C++ describe high level behavior of these components, while parts of code written in SystemC describe cycle-accurate design models of components.

System level design offers a way to have a fast executable specification of the design that can be used to validate the system concepts and also verify the design by having bus-cycle accurate models for faster simulation early in the development process (BHA, 2002).On the other hand, register transfer level (RTL) or gate level design are more precise, but also much more time consuming. Also, RTL models are difficult to be written and understood, but system level models are manageable and sufficient for comparing different architectures and topologies. Figure 13 illustrates the differences in terms of speed and size of code for system level and RTL, which justify the choice of C++ with SystemC library.

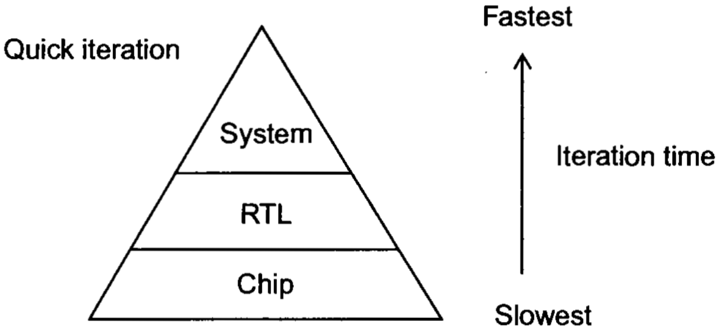


Figure 13. Different levels to explore alternate architectures

3.3 Default features structure, operation and features

The basic components of the Noxim simulator are a) routers, b) processing elements, c) buffers and d) links. The topology is a mesh architecture, in which user defines x and y dimensions. Besides topology, user can optional define routing strategy and/or traffic generated on NoC. By using these options user can get multiple tests in desired designs.

The simulation is based on a SystemC defined clock, which is connected to all components, so the operation is synchronous. There are options about length of runtime and warm-up time of the NoC. During the simulation period there is a collection of various stats and optional printing of information messages at every cycle, as well. At the end, simulation tool offers a brief or detailed presentation of statistics based on user options. Both global results and statistics for every single component are available, revealing possible hot spots and weaknesses of the design.

The main metrics are delay, throughput, power consumption and number of packets transmitted. These metrics are counted for every node separately and for global communication, as average values. Delay values, like aggregated average delay, average delay for communication, maximum delay, minimum delay and others, are presented in clock cycles and are counted for every operating route inside the NoC.

3.3.1 Routers and processing elements

Processing elements are the generators of traffic inside the NoC. Every processing element transmits packets according to its packet injection rate, which can be defined by the user as a command line option. The traffic is generally random, if not defined otherwise, produced by a pseudo-random generator. However, user is free to define exactly the traffic produced by PEs. This can be done by writing a traffic table, in which every line gives all the necessary information for a single transmission. The syntax of a traffic table has the following form:

Source	Destination	Packet injection rate	Probability of retransmission	Time in cycles transmission begins	Time transmission ends	Period between two successive transmissions

Every PE has a port for clock signal and one for reset signal and connects to one router by using a group of ports, as explained in the links section.

Routers are the most important components, referring to communication of NoC. Their objective is to receive packets, read the appropriate fields on head flit of every packet, decide the next hop, check availability of links, reserve links and forward packets. Routers also have reset and clock ports. In the default version of tool every router was connected to one local PE and four neighbors in directions north, south, east and west with an input buffer for every connection.

3.3.2 Links and buffers

The ports of routers and PEs are connected through signals, available on SystemC library. These low level implementations of links enhance the reliability of simulation results. Every connection consists of a group of signals. These signals are:

- Outgoing request signal, send by a component requesting a transmission
- Incoming request signal, received by a component when its neighbor request transmission
- Outgoing flit signal, the bus used to transmit one flit
- Incoming flit signal, the bus used to receive one flit
- Outgoing acknowledge signal
- Incoming acknowledge signal
- Free slots signal
- Free slots neighbor signal
- Outgoing data signal
- Incoming data signal

Every connection has a buffer, whose size can be defined globally for all buffers or locally inside topology table. Routers always check the availability of buffers before sending packets to them.

3.3.3 Routing protocol

Noxim tool supports several routing algorithms, like XY, West-First, North-Last, Negative-First and Odd-Even routing. Also, user can define his custom routing policy as an input routing file. However, building routing tables in the default version of the tool was a difficult task, because user have to define for each router, five routing strategy for every destination node. The syntax of default routing table has the following form:

Source node of the flit	Direction from which the flit came in form X->Y	Destination node of the flit	Chosen routing direction in form X->Y

For example, if user wants node 1 to forward a flits with destination node 2 that arrive to node 1 through link 0 (local), he has to add the following line to routing table:

1	0->1	2	1->2

Accordingly, user needs to add four more lines with incoming directions X->1, where X is the node id for all its four neighbors. Also, this procedure will be followed for every destination node and the result will be a long routing table, which can be generated only by an external script. There is a much easier way to build routing table in the new version of the tool explained later.

3.3.4 Energy models

Statistics are based on normalized energy models that refer to energy consumption on communication. Energy is counted per bit while following events occur:

- Transmission on link
- Selection of next hop
- Routing of packet
- Idle router

Global and local simulation statistics are presented to user after finishing of simulation. Local detailed statistics for every link of every router are available. The following metrics are estimated during simulation:

- Transmitted packets and flits (global and local)
- Latency (global and local, maximum, minimum and average)
- Throughput (global and local, per flit/cycle/router)
- Energy consumption (global and local)

3.4 Extensions

Several extensions have been added to default Noxim tool, in order to support irregular NoC topologies. The user is able to build custom NoCs by combining multi-port routers and testing the design. The general flow for the new simulation tool is shown in Figure 14. User can define his custom topology and the graph of the application running on custom topology. Optional the routing strategy can be included. Also, runtime options, like number of running clock cycles, warm-up period, minimum and maximum number of flits per packet, are given as input from the user.

Low level SystemC models are available for components, including routers, links, processing elements and buffers. These models are aware of energy consumption. User defined tables and predefined components are used by simulation tool to implement the design. Designing algorithms are written in C++. Also, if no routing table is available, shortest path is assumed. Dijkstra algorithm is used to find shortest paths between routers. The generated topology is tested and various statistics are counted. After simulation detailed results will be presented. Figure 14 shows the simulation flow with colored boxes indicating user's input.

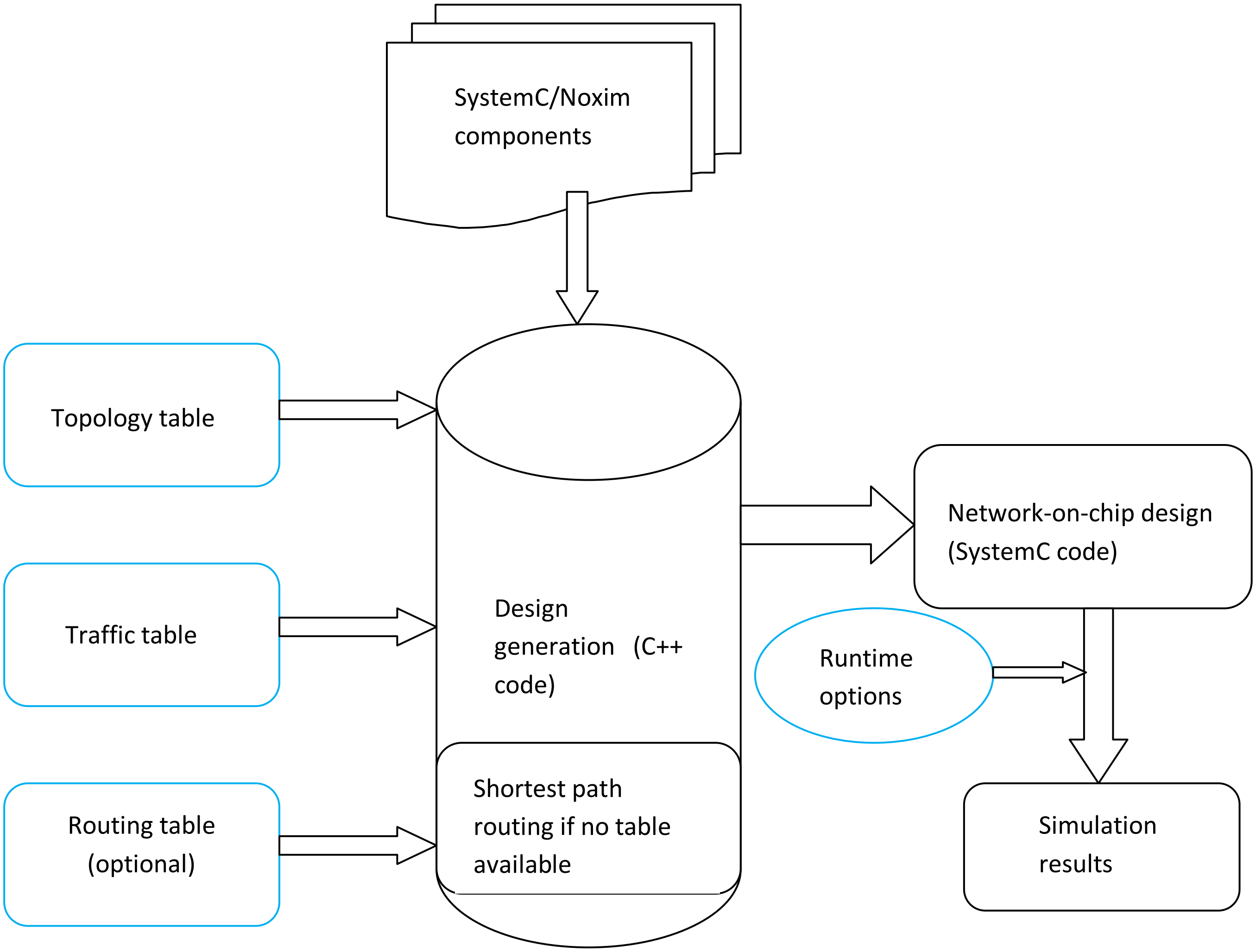

SystemC/Noxim components
Topology table
Traffic table
Routing table (optional)
Design generation (C++ code)
Shortest path routing if no table available
Network-on-chip design (SystemC code)
Runtime options
Simulation results

Figure 14. Simulation flow

3.4.1 Multi-port routers and routing protocol

The most important extension to the default version of simulation tool was the increase on number of ports per router. Exploration of irregular topologies demands different number of connections for every router to maximize performance. New routers implemented, which can handle up to ten connections. These connections can be either links with other routers on the NoC, or connections with processing elements. Each connection contains all the necessarily ports described above, like request and acknowledge flit ports. Also, router contains an input buffer for every connection.

This extension offers the opportunity to test different design approaches. For example, assuming that designer wants to test an application that uses ten processing elements. Using the default version of simulation tool designer can only build mesh architectures, and particularly a 1x10 or a 2x5 or a 3x4 with some inactive routers. In contrast, extended version can simulate every possible topology, starting from one router with all processing elements connected to it, to ten routers with multiple connections between them. The first topology with only one router will minimize power consumption, while architecture with ten routers and every router connected to one processing element and the rest of the routers will minimize delay. Designer can choose the most important performance and cost metric or preferable a balance in trade-off between metrics by running multiple simulations.

3.4.1.1 Automatic generated routing tables

Routing strategy for custom irregular topologies can be defined either by the user with a new simplified routing table or by simulation tool with execution of Dijkstra algorithm to find the shortest path. The second option is preferable, because can be implemented without any extra effort by the designer, but routing table is essential when a special routing strategy have to be tested. In the new routing table one line for each router defines the port, in which the router will forward the flit, according to the destination defined in the header flit. So, the number of lines in routing table is equal to the number of routers and the number of columns is equal to number of processing elements. An element of the table in place (i, j) is the port number to which router i has to forward a flit, which has destination processing element j.

3.4.2 Topology

The extended version of Noxim simulation tool developed during this thesis focuses on custom topologies. The user defines the number of routers and PE’s, the connections between them and the size of the buffer for every connection in a topology file. The following example illustrates the syntax of the topology description, acceptable from simulation tool.

The topology file:

```
%Topology table file
%4 Routers, 8 PE's
%port in, port out, buffer in, buffer out, Router, layer
R0: 1,2,4,2,R1,0 2,6,3,4,R2,0 ; 3,9,4,P7,0 ;
R1: 1,3,4,8,R3,0 ; 3,2,8,P5,0 4,2,8,P4,0 5,2,8,P3,0 ;
R2: 9,5,4,5,R3,0 ; 2,4,4,P6,0 ;
R3: ; 6,4,4,P0,0 4,4,4,P1,0 9,5,5,P2,0 ;
```

describes the following NoC topology:

- Router 0 (R0) has connections with router 1(R1), router 2 and processing element 7.
- The connection R0-R1 is established between port 1 of R0 and port 2 of R1. The buffer dedicated to traffic from R1 to R0 has 4 slots and for flits arriving from R0 the R1 there is a 2 slots buffer in R1.
- For the connection R0-R2, there is a 3 slots buffer on port 2 on the R0 side and a 4 slots buffer on port 6 on the R2 side.
- R0 connects with P7 using its 3rd port and a buffer with 9 slots. Processing elements have only one port and no buffers, that’s why number 4 previous to P7 doesn’t declare anything.
- All the connections take place on layer 0 (not implemented yet)
- Similarly, R1 connects to R3 and three processing elements (P3, P4, P5), besides the connection with R0 declared in the previous line.
- File syntax: comment lines start with (%), a semicolon (;) is used to separate connection with routers and connection with PE’s and another (;) at the end of each line.

The NoC defined by the previous topology description is shown in Figure 15. For the first line there is a colored representation, in which each color on the text line corresponds to the color of the figure produced by the line.

R0: 1,2,4,2,R1,0 2,6,3,4,R2,0 ; 3,9,4,P7,0 ;

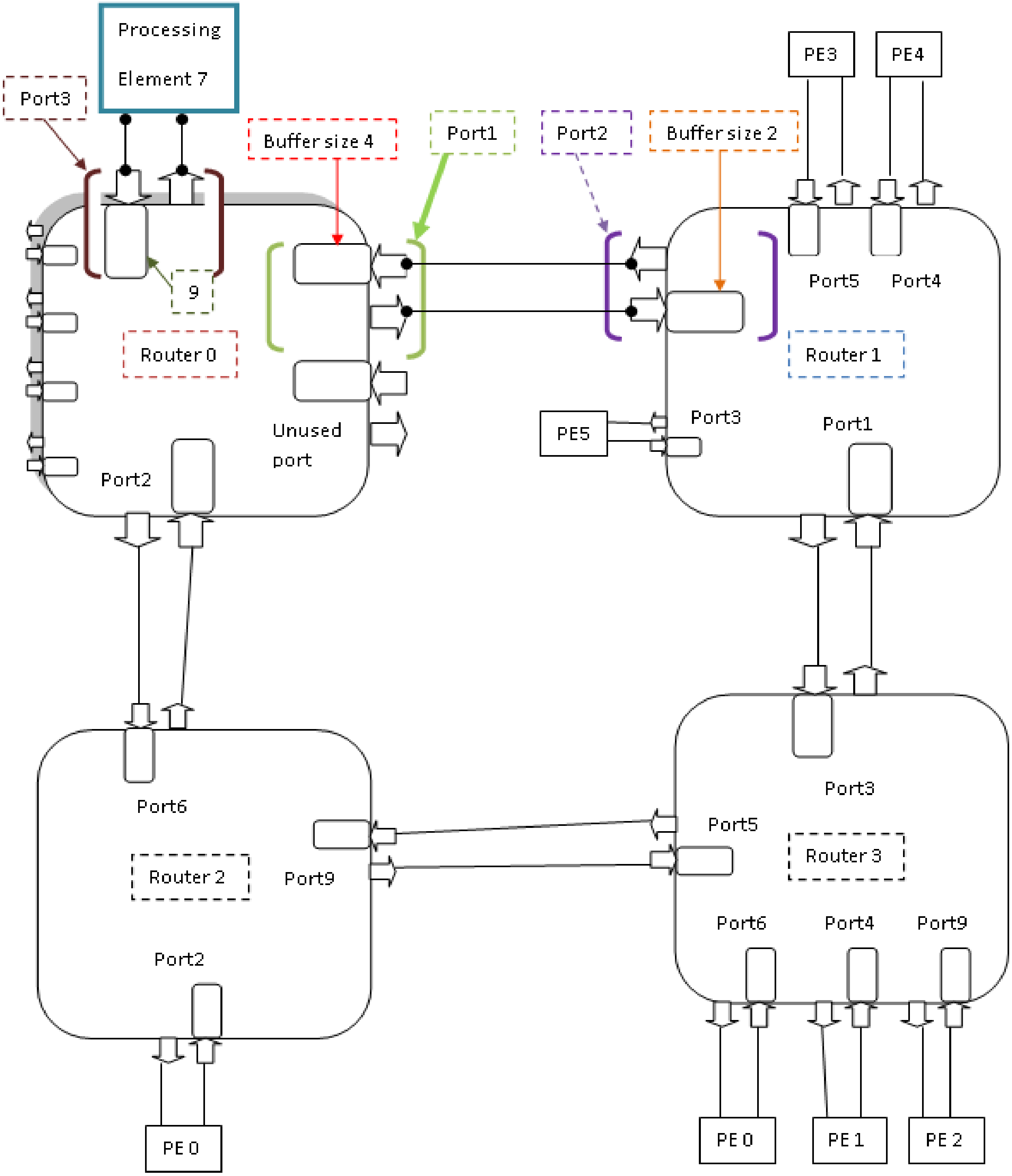


Figure 15. Topology generation example

3.4.3 Buffer sizing

Another extension of the simulation tool is the ability to define input and output buffer size of every link on the network by using the topology table input described above. This option along with the new energy model gives more realistic simulation results.

3.4.4 Buffer energy assumption

A more complex buffer energy model is necessary in the extended version of the tool, because the size of the buffer in every link. We assume a linear growth in energy consumption as buffer size increase. For example, a buffer with four slots consumes half of the energy compared to an eight slots buffer for a single operation, like reading or writing.

3.4 Comparison with Wormsim (extended), Noxim

Wormsim (ANA, 2008) (BAR, 2007) simulator is another innovative NoC simulator. Some features of Wormsim, Noxim and extended version of Noxim (Noxim++) simulators are summarized in table 2.

	Wormsim (extended)	Noxim	Noxim++
Accuracy	High level models	Combination of high and low level models	Combination of high and low level models
Speed	Fast simulation	Quite fast simulation, but slower that Wormsim	The same simulation period as Noxim
Routing protocols	Many	Many	Many
Custom routing	Yes	Yes	Yes

3D- support	Yes	No	No
Custom topologies	No	No	Yes
Routing energy models	Yes	Yes	Yes
Buffering energy models	Yes	No	Yes
Random traffic	No	Yes	Yes
Preset traffic models and custom traffic	Yes	Yes	Yes
Maximum number of connections per router	5	5	10
Automatic shortest path routing option (Dijkstra)	No	No	Yes
Wormhole switching	Yes	Yes	Yes
Auto-find appropriate buffer size	Yes	No	Yes
Maximum number of PE’s per router	1	1	10

Table 2. Wormsim, Noxim, Noxim++ comparison

3.6 Comparison with other tools

There are also other well-know simulation tools and several publication referring to NoC simulation. In this section there is a brief comparison between Noxim++ and simulation tools described in (LU, 2006), (HU, 2002), (PEN, 2006), (BER, 2005) and (PAN, 2005), in terms of NoC topology and evaluation criteria of simulation (Table 3). Generally in current literature, the most studied topologies are bus and mesh and the most common metrics, which are obtained mostly via simulation rather than synthesis, are area, runtime, latency and power consumption (SAL, 2007).

Authors Evaluation criteria		(HU, 2002)	(LU, 2006)	(PEN, 2006)	(BER, 2005)	(PAN, 2005)	Noxim++
Topologies	Single bus	√					
	Mesh		√	√	√	√	√
	Point-to-point				√		
	Custom	√					√
Metrics	Power	√		√	√	√	√
	Latency		√		√	√	√
	Throughput		√			√	√
	Area				√	√	√

Table 3. Comparison with other tools

Chapter 4

Simulation results

4.1 Introduction

This chapter includes simulation results for four popular hardware implementations. The simulated systems are (a) an MPEG-4 hardware encoder/decoder, (b) a video object plane decoder (VOPD), (c) a multimedia system (MMS) and (d) a Multi-Window Display (MWD). Necessary definitions for system designs were discussed in chapter 1 and design flow was presented in chapter 2. In chapter 3 there was a description of the simulation tool, which applies concepts and knowledge from the previous chapters, and simulation results in this chapter were produced using Noxim++.

The goal of this chapter is the design exploration of these four applications in hardware, in order to find the optimal solutions in terms of delay and energy. Generally, minimization of delay results to growth of energy, because more routers with several buffer slots needed, and vice versa. So, the method used for the simulation tries to find the best balance between these metrics. Firstly, there is a topology exploration for every application, that reveals the optimum number of routers that should be used and the components included in each partition. Afterwards, energy optimization will decrease the amount of energy needed for the optimum topology found before. Section 4.2 and 4.3 describe the methodology used in simulations.

4.2 Topology exploration

For every application there is a simulation for the mesh topology, based on optimum mapping from previous works, and proposed irregular topologies that also aware of optimum division into partitions. So, components of every studied DSP application are divided into partitions in a way that components that frequently communicate with each other share the same router. Also, components that produce heavy traffic are connected to a router with few other elements, if possible. These partitions for every application were produced by CHACO tool, which offers minimization in communication between different partitions that causes significant delays.

In simulation results presented the main metrics are global average delay on the NoC, global throughput, maximum delay of packet in whole simulation and total energy consumption. The energy values are normalized to energy consumption of mesh topology, so comparison can be done easier. Traffic traces for every application are available and are tested for three different numbers of flits. So, proposed NoCs are tested for different loads. Simulation results give a good outlook of advantages and disadvantages in every partition and help designer view the optimum topology for every application and every load.

4.2 Power optimization

Power consumption can be decreased by reducing the size of the buffers in each router. As described in section 3.4.3 the size of buffer is important factor to its power consumption. To minimize the buffer energy consumption without downgrading NoC performance router should contain the maximum number of buffer slots used, as proposed in (ANA, 2008). So, during the first simulation each time, results are collected about buffer slots used assuming that routers have a nine slot input buffer for each of their connection. In the next simulation for the same topology and the same packet size, we change the buffer slots for every router to the maximum number of slots used by that specific router. In that way each router has sufficient buffer slots, maintaining the same performance, but energy is reduced every time a router needs less than nine slots. In the next level, we optimize the buffer size for every connection link in the NoC. So, the minimum number of slots is included in every buffer and lower energy consumption is achieved without performance loss. At the rest of the chapter the terms used to describe the buffer slots are:

- Global: All routers on the NoC have buffers with nine slots on each link
- Router: Each router uses the same number of buffer slots in each of its links. This number is defined as the maximum number of slots in use in the more stressed of its links.
- Link: Each link has exactly as many buffer slots as needed, so no delay occurred due to lack of buffer slots. The only limitation is that no link can have more than nine buffer slots.

4.3 MPEG-4

MPEG-4 is a broadly used protocol for audio and video encoding. A hardware encoder and decoder consist of many components, so a NoC approach is suitable, and have studied in regular NoC before. The tested MPEG-4 includes twelve processing elements, such as a video unit, an audio unit, a risc processor, a med cpu, a binary alpha block and three srams. Different partitions produced by CHACO and including two to ten routers are tested. Also, simulation results for an optimized mapped mesh topology are obtained and values are normalized to the values of mesh topology. Three, four and five flits per packet choices are simulated, which stress the NoC more or less. These range of workloads results in different solutions.

4.3.1 MPEG-4 delay

Global average delay as a function of topology is presented in figures 16 and 17. In the horizontal axis there are the number of partitions and the mesh topology at the end and corresponding values of delay are on the vertical axis. Lower delay is achieved for lower traffic load. Also, the delays are not acceptable for more than four flits per packet. The best performance is achieved for two partitions when there are three flits per packet and four partitions when there are four flits. Minimum delay for three flits is 0.65 and for four flits 0.79. Minimum delay when every packet consists of five flits is 1.27 for partition 12 and it is more than the average delay of the mesh topology. The average of average delay for three flits is 0.72, for four flits 0.83 and for five flits 8.46.

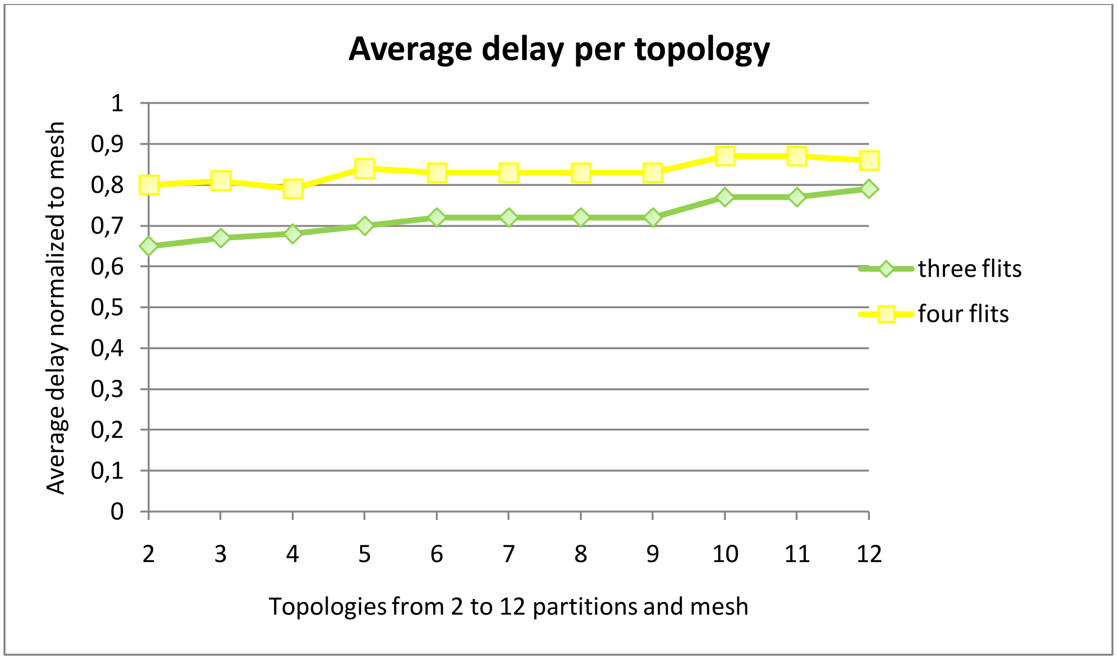


Figure 16. Average delay for three and four flits per packet

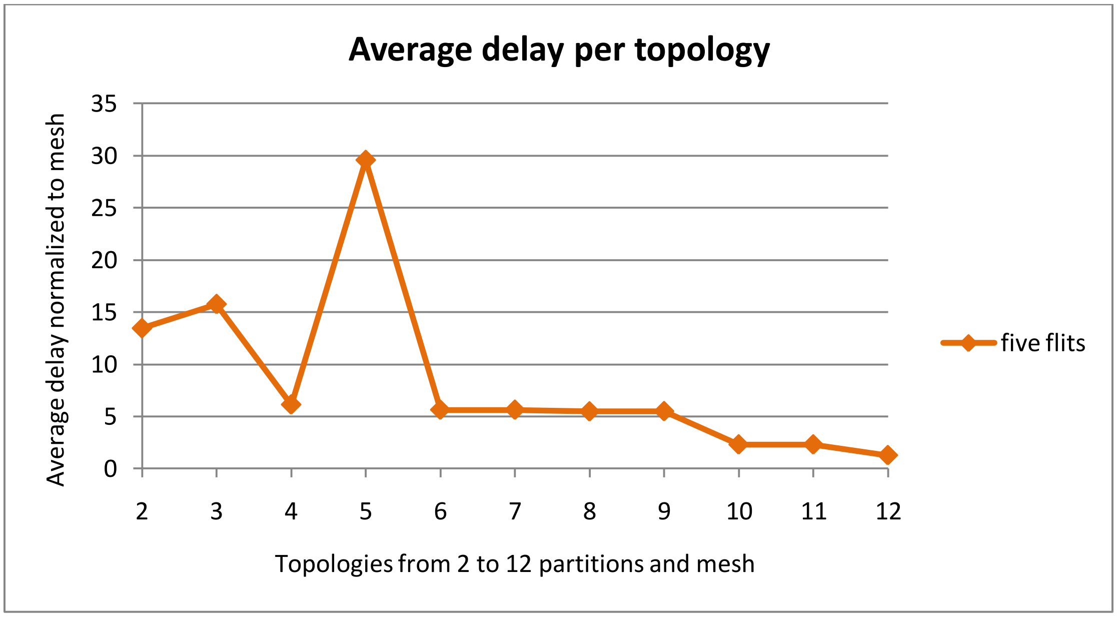


Figure 17. Average delay for five flits per packet

4.3.2 MPEG-4 throughput

Also, better throughput is achieved when few partitions and thus few routers are used as shown in Figure 18. Maximum throughput is 0.81, 1.08 and 1.33 for three, four and five flits curves accordingly. Average values of throughput are 0.45, 0.60 and 0.75. Again for each curve values are normalized to the mesh topology of the curve.

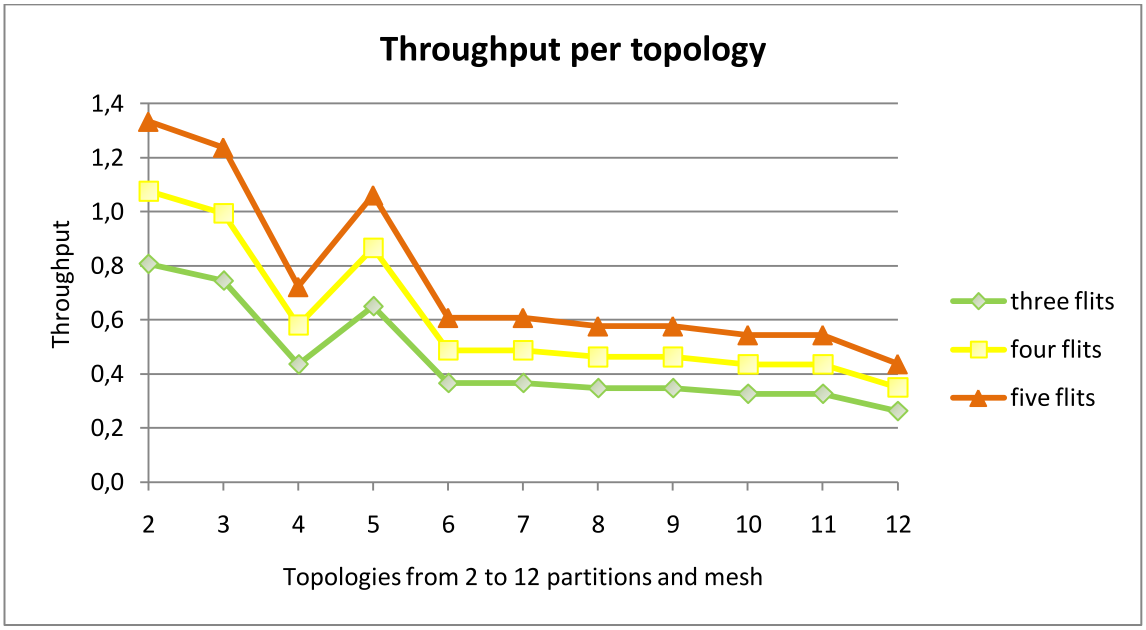


Figure 18. Average throughput for three, four and five flits per packet

4.3.3 MPEG-4 maximum delay

Maximum delays counted in clock cycles in figures 19 and 20 can be used to check if unacceptable delays occur during simulation time. Interestingly, for four flits simulation the four partitions result in the higher maximum delay, although the average delay in minimum, according to figure 16. Maximum values in charts are 28, 45 and 3342 clock cycles for three, four and five flits accordingly.

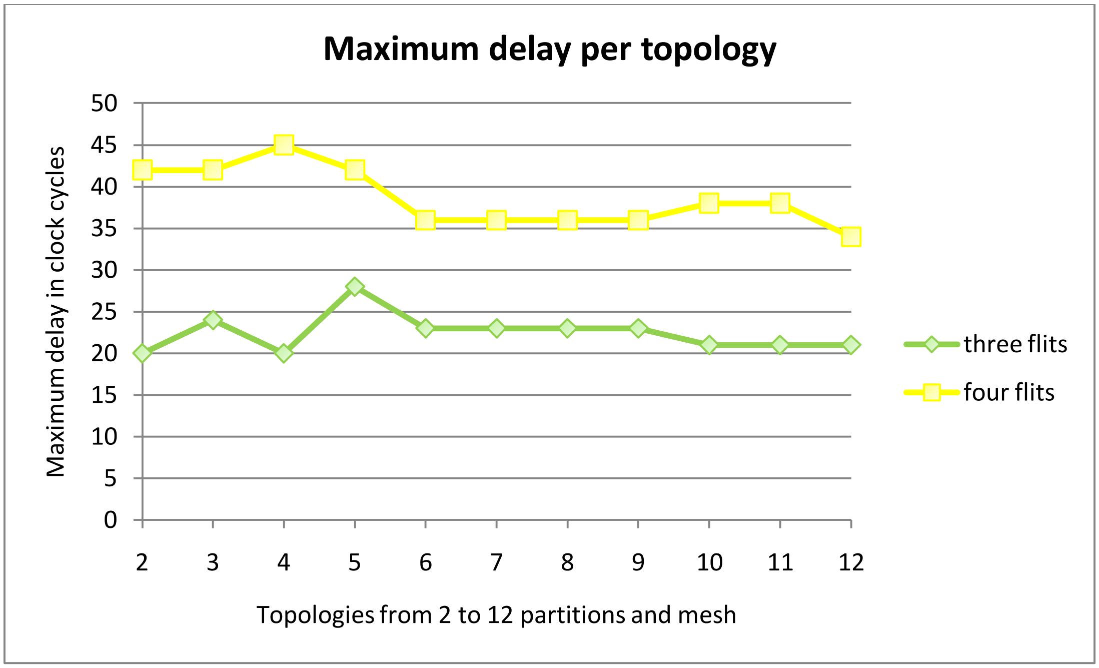


Figure 19. Maximum delays for three and four flits

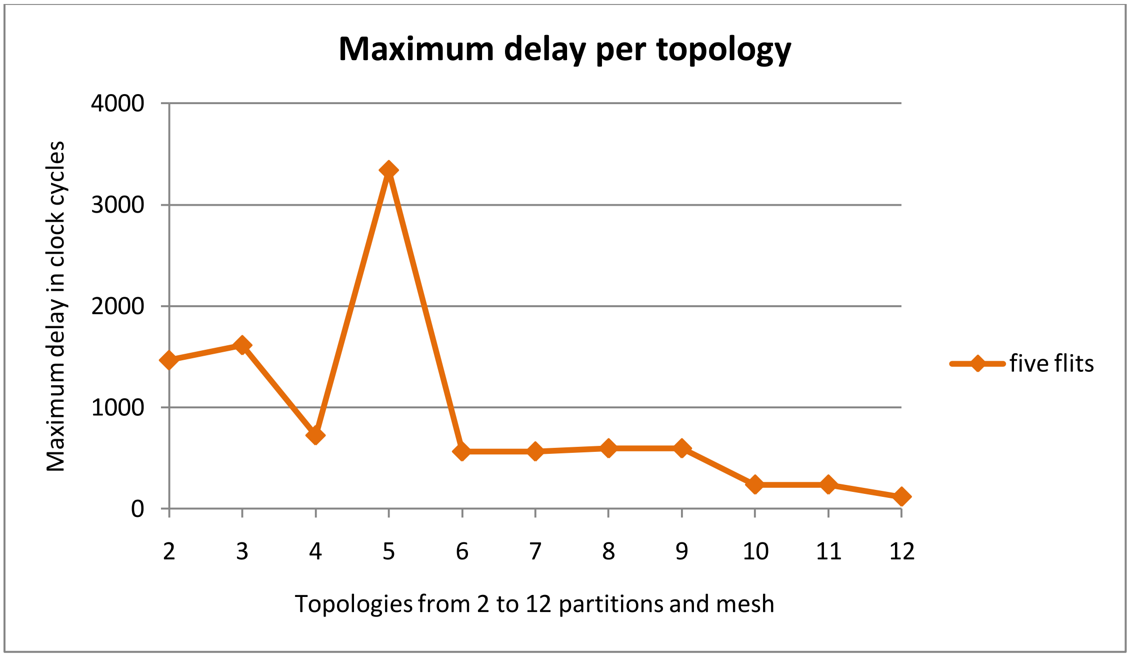


Figure 20. Maximum delays for five flits

Generally, as the number of routers increases the number of hops for each packet increases also, so expect higher delay for higher partitions. However, if the traffic load is significant heavy it cannot be supported by a few routers. This is the reason why six or more partitions/routers are needed to achieve acceptable delays for packets with five flits.

4.3.4 MPEG-4 energy consumption

Energy consumption values for every packet size are figured in figures 21, 22 and 23. The values are normalized to the value of energy consumption on mesh topology with nine slots buffer. The global column presents the energy when nine slots are assumed per buffer; the router column presents the energy for optimized number of buffer slots per router and the link column optimizes buffer slots for each connection on the NoC. The difference between router and link optimization is that in the first case the maximum number of slots of every link on the router is used for each router, as explained in section 4.2. Minimum value 0.235 is achieved for partitioning number 9 for three flits, 0.382 for partitioning number 8 for four flits and 0.58 for partitioning number 8 for five flits. Average values are 0.43, 0.56 and 0.68 accordingly.

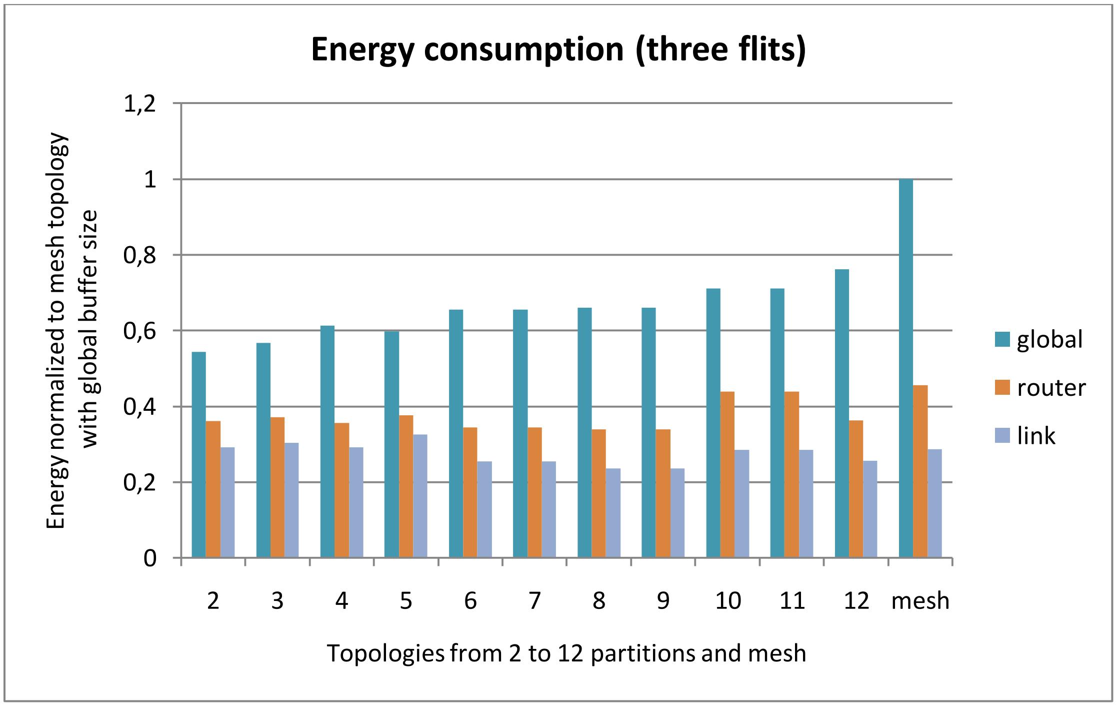


Figure 21. Energy consumption for different buffer slots optimizations

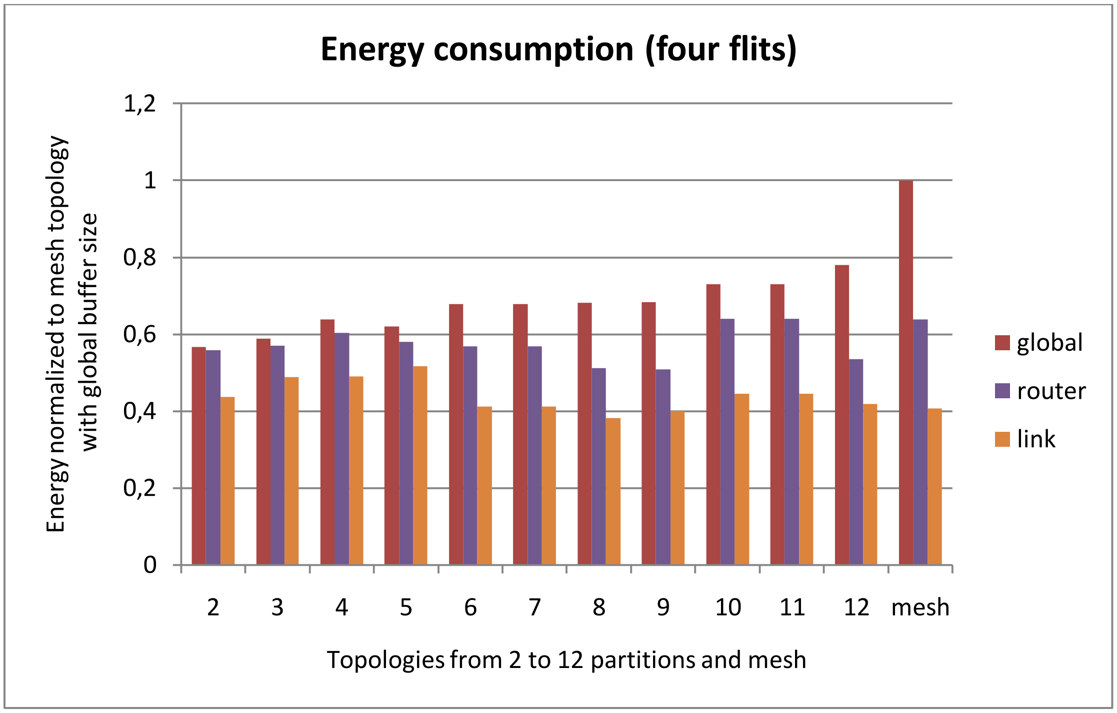


Figure 22. Energy consumption for different buffer slots optimizations

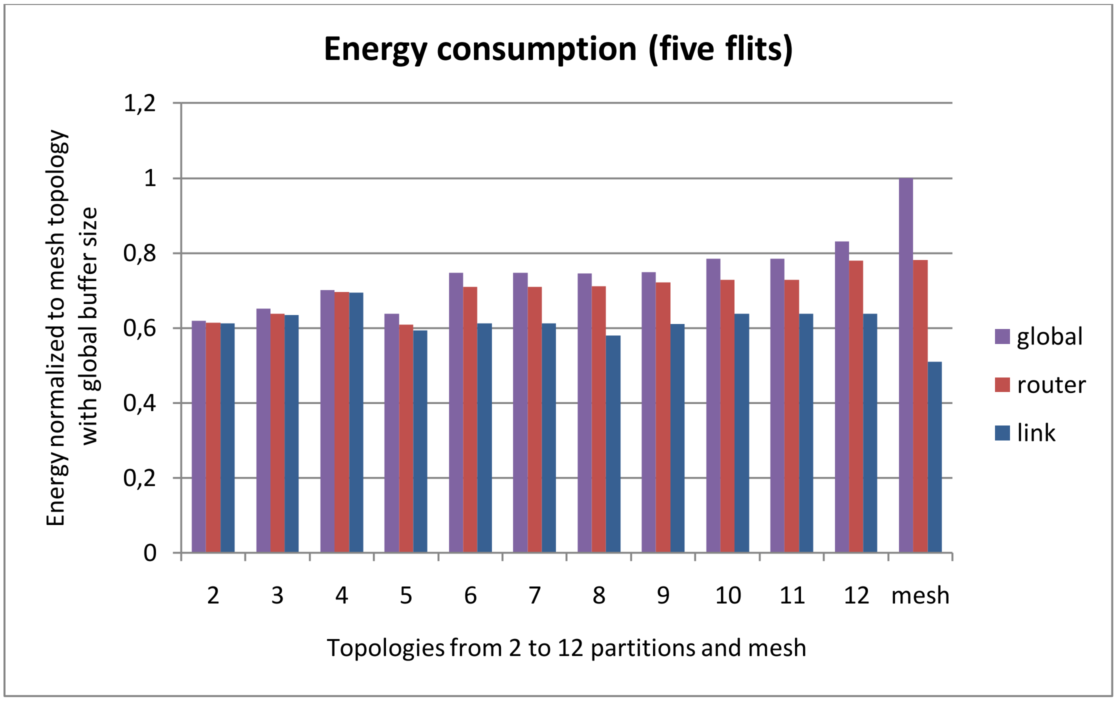


Figure 23. Energy consumption for different buffer slots optimizations

4.3.5 MPEG-4 power gain

Routers consume power for reading and writing buffers along with switching and forwarding. So, reading and writing energy is reduced for smaller buffers as expected. Also, more routers will result to higher energy consumption. We can achieve great reductions, when the traffic is relatively low to the number of slots in the buffers, like the first case in which three flits normally cannot fill most of the buffer slots. So, optimizations result in more than 50% reduction in many cases. On the other hand heavy traffic loads, when five flits are produced for each packet, fulfill the buffer slots and optimization through minimization of buffer slots is not so impressive. A further removal of buffer slots will improve energy gain, but also increase delay on the network, which is not acceptable in this study. Power gain by removal of unused slots can be up to 66.32% for three flits (figure 24), 46.45 % for four flits (figure 25) and 23.22% for five flits (figure 26) for link optimization. Average gains are 57.04, 33.47% and 13.48% accordingly for link optimization. When optimization is achieved only per router the maximum (and average) gains are 52.38% (42.58%), 31.50% (14.14%) and 7.18% (4.26%) for three, four and five flits.

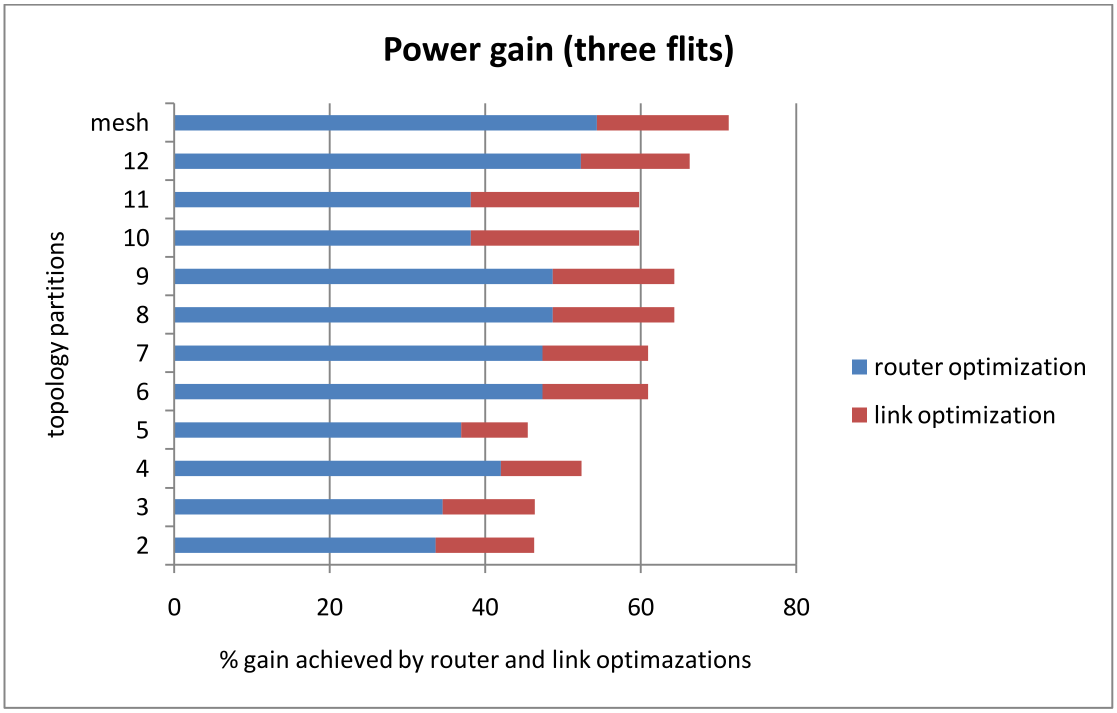


Figure 24. Power consumption reduction through buffer slots optimization

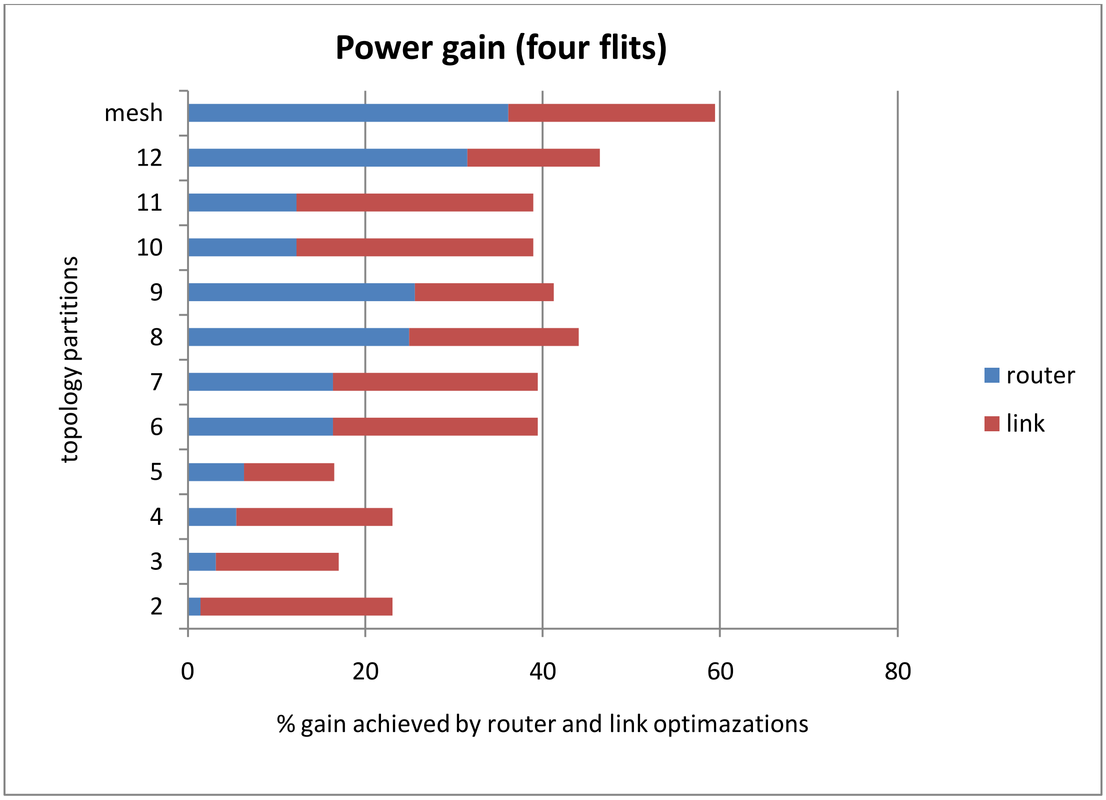


Figure 25. Power consumption reduction through buffer slots optimization

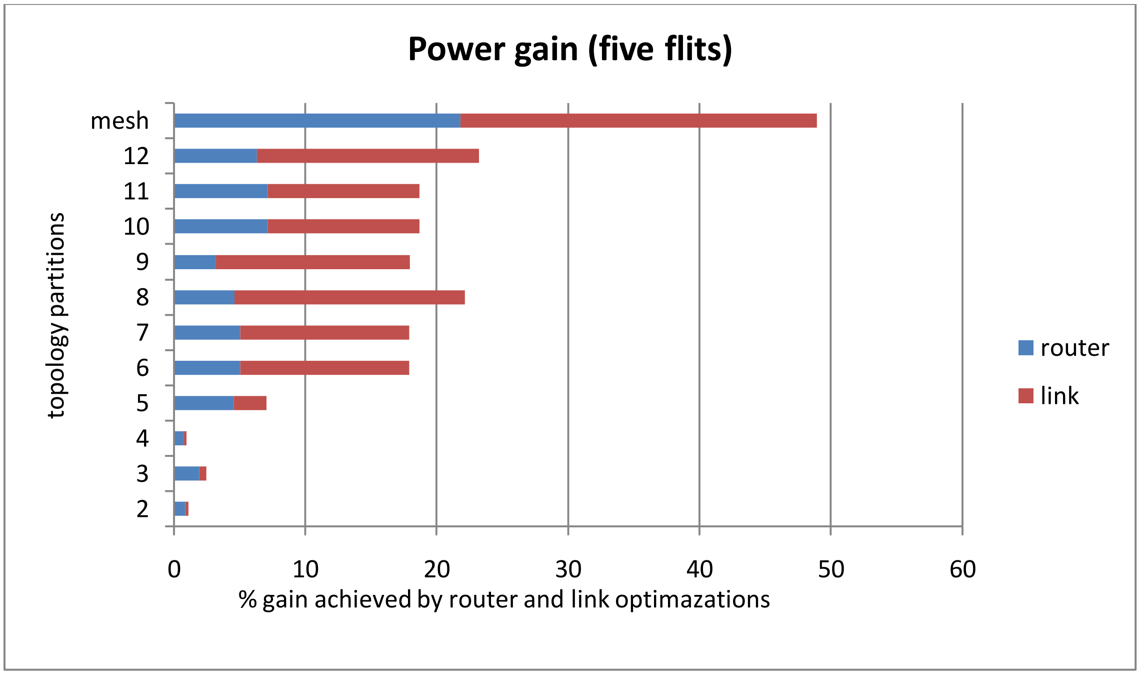


Figure 24. Power consumption reduction through buffer slots optimization

4.4 VOPD

Video object plane decoder is another digital signal processing application that has been proposed for use on NoC and studied before (MUR, 2005). VOPD offers quality video transition with decent bandwidth performance. The tested VOPD decoder includes twelve processing elements, such as two length decoders, an AC-DC prediction, an ARM processor, two memory components and a VOP reconstructor. Different partitions produced by CHACO and designs with two to twelve routers have been tested. Also, simulation results for an optimized mapped mesh topology are obtained and values are normalized to the values of mesh topology. Four, five and six flits per packet choices are simulated, which stress the NoC more or less. These range of workloads results in different solutions.

4.4.1 VOPD delay

Global average delay as a function of topology is presented in figure 25. In the horizontal axis there are the number of partitions and the mesh topology at the end and corresponding values of delay are on the vertical axis. Lower delay is achieved for lower traffic load. All delay values are lower than delays on mesh topology. The best performance is achieved for two partitions for every load. This happens, because the traffic can be efficiently handled by only two routers so more routers delay the packet transition as more hops are done between routers. Minimum delay for four flits is 0.52, for five flits 0.54 and for six flits 0.62. We also notice that partitioning number 8 gives poor performance compared to the rest. The average of average delay for four flits is 0.68, for five flits 0.69 and for six flits 0.72.

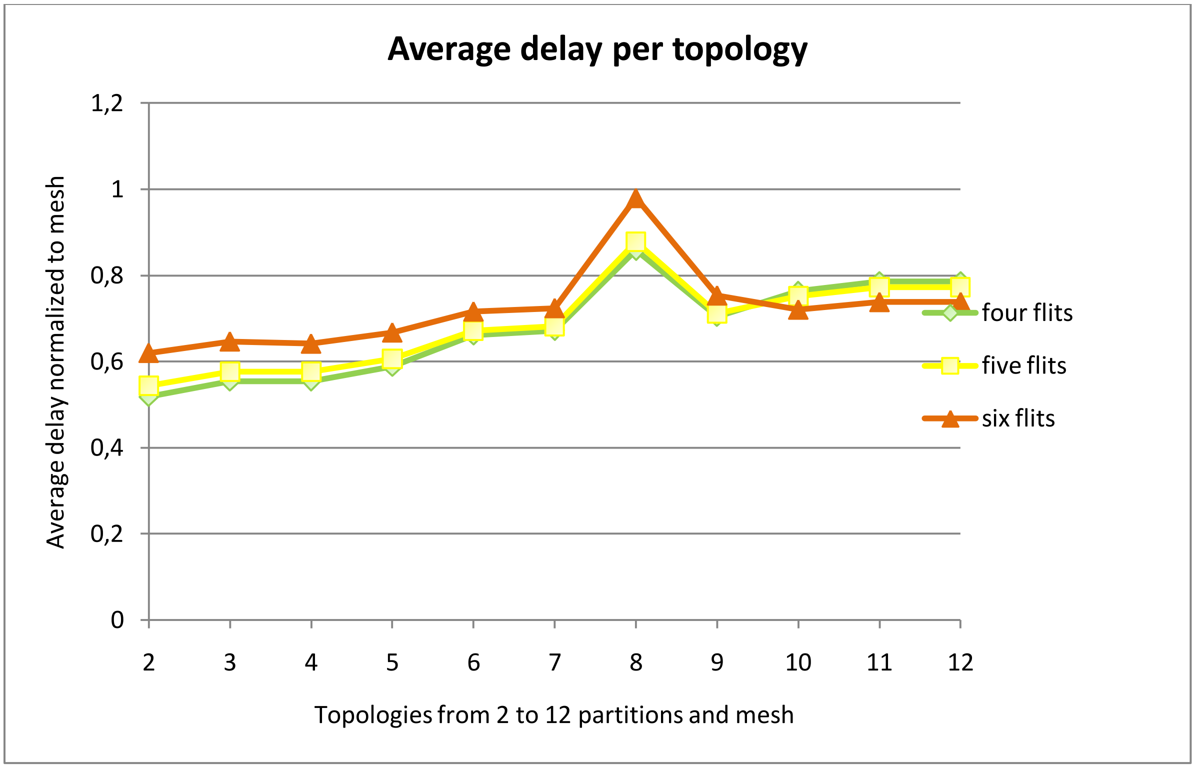


Figure 25. Average delay for four, five and six flits per packet

4.4.2 VOPD throughput

Also, better throughput is achieved when few partitions and thus few routers are used as shown in Figure 26. Again for each curve values are normalized to the mesh topology. Highest throughput is achieved for two partitions for every load and maximum values are 0.74, 0.93 and 1.11 for three, four and five flits. Average values of throughput are 0.39, 0.48 and 0.58 accordingly. As expected the throughput is higher when more flits are transmitted for every packet.

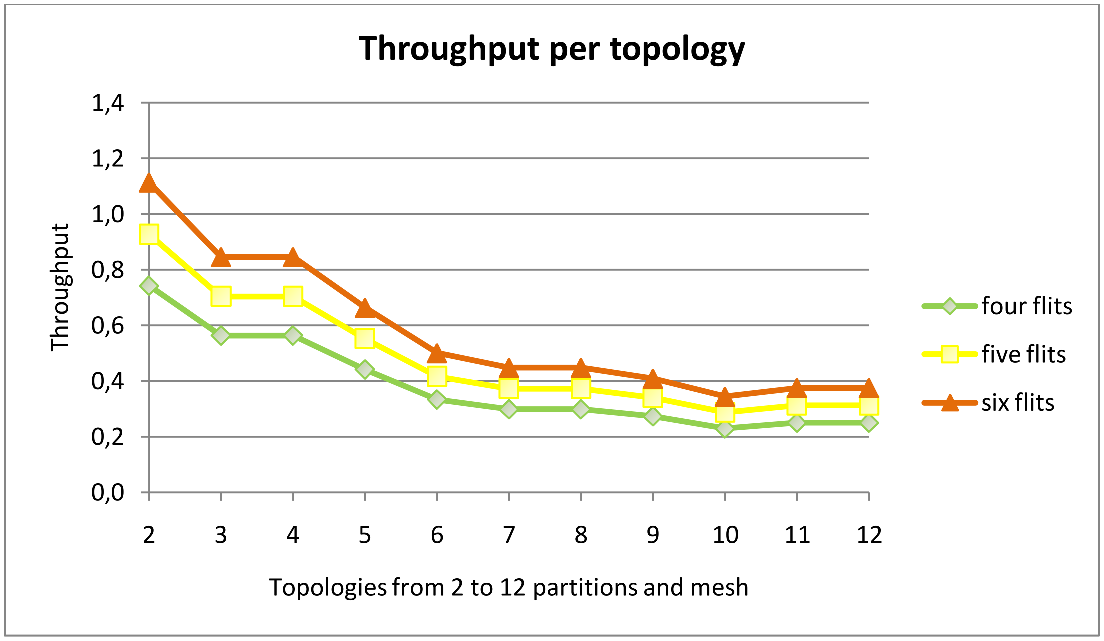


Figure 26. Average throughput for three, four and five flits per packet

4.4.3 VOPD maximum delay

Maximum delays counted in clock cycles in figure 27 can be used to check if unacceptable delays occur during simulation time. The maximum values on the chart are 24, 28 and 72 and average maximum delay is 20, 25.1, 38.27 for four, five and six flits. Generally, as the number of routers increases the number of hops for each packet increases also, so expect higher delay for higher partitions. However, partition 8 again causes high values of delay, especially in the case of six flits when the network is stressed the most.

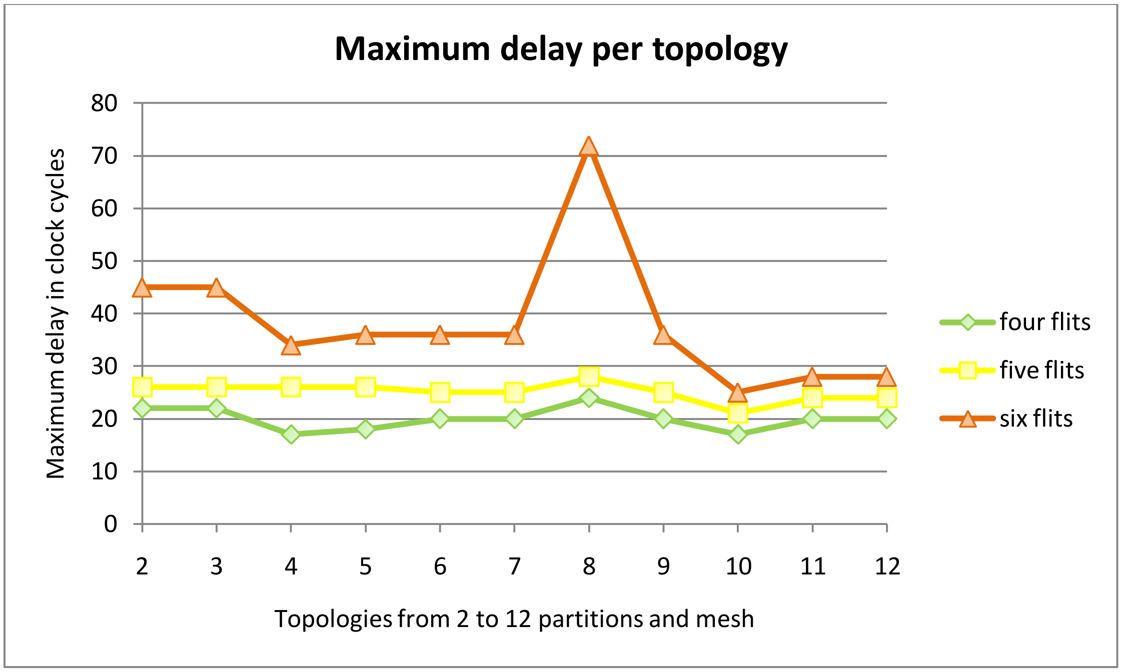


Figure 27. Maximum delays for four, five and six flits

4.4.4 VOPD energy consumption

Energy consumption values for every packet size are figured in figures 28, 29 and 30. The values are normalized to the value of energy consumption on mesh topology with nine slots buffer. The global column presents the energy when nine slots are assumed per buffer; the router column presents the energy for optimized number of buffer slots per router and the link column optimizes buffer slots for each connection on the NoC. The difference between router and link optimization is that in the first case the maximum number of slots of every link on the router is used for each router, as explained in section 4.2. Minimum value 0.170 is achieved for partitioning number 2 for four flits, 0.320 for partitions 11 and 12 for five flits and 0.257 for partitions 11 and 12 for six flits. Average values are 0.40, 0.55 and 0.48 accordingly. Once again, when more flits are transmitted higher partitions offer better performance, while only two partitions minimize the energy when four flits are transmitted per packet.

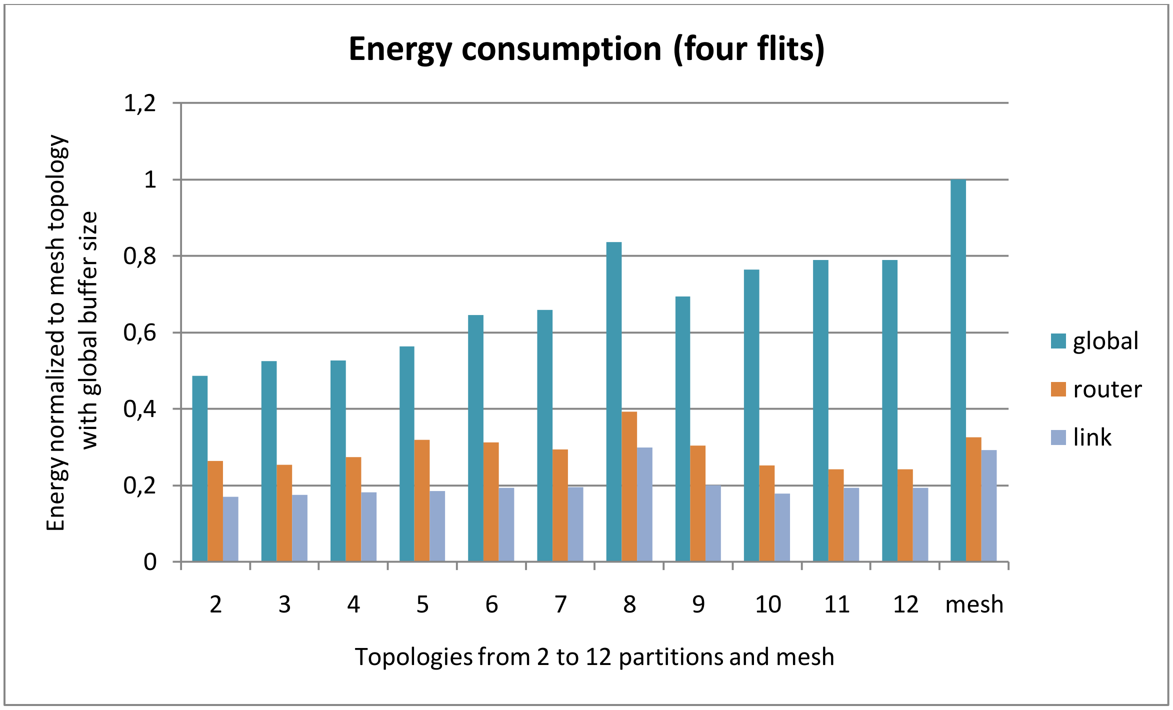


Figure 28. Energy consumption for different buffer slots optimizations

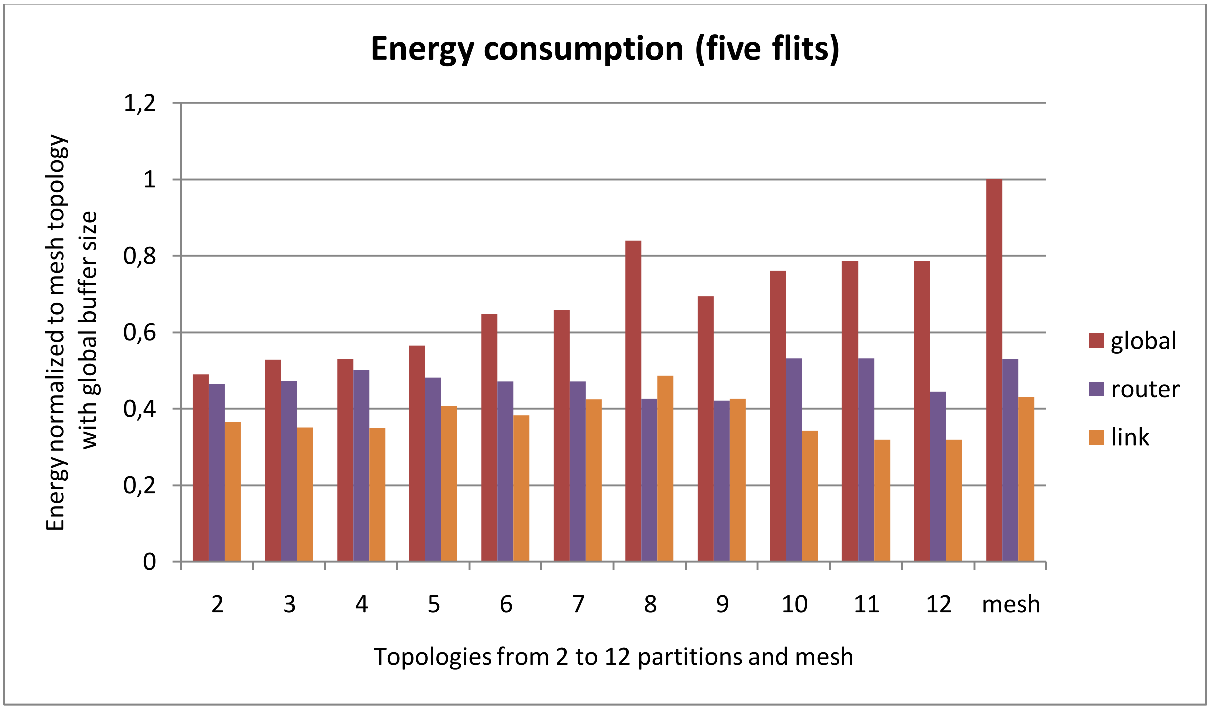


Figure 29. Energy consumption for different buffer slots optimizations

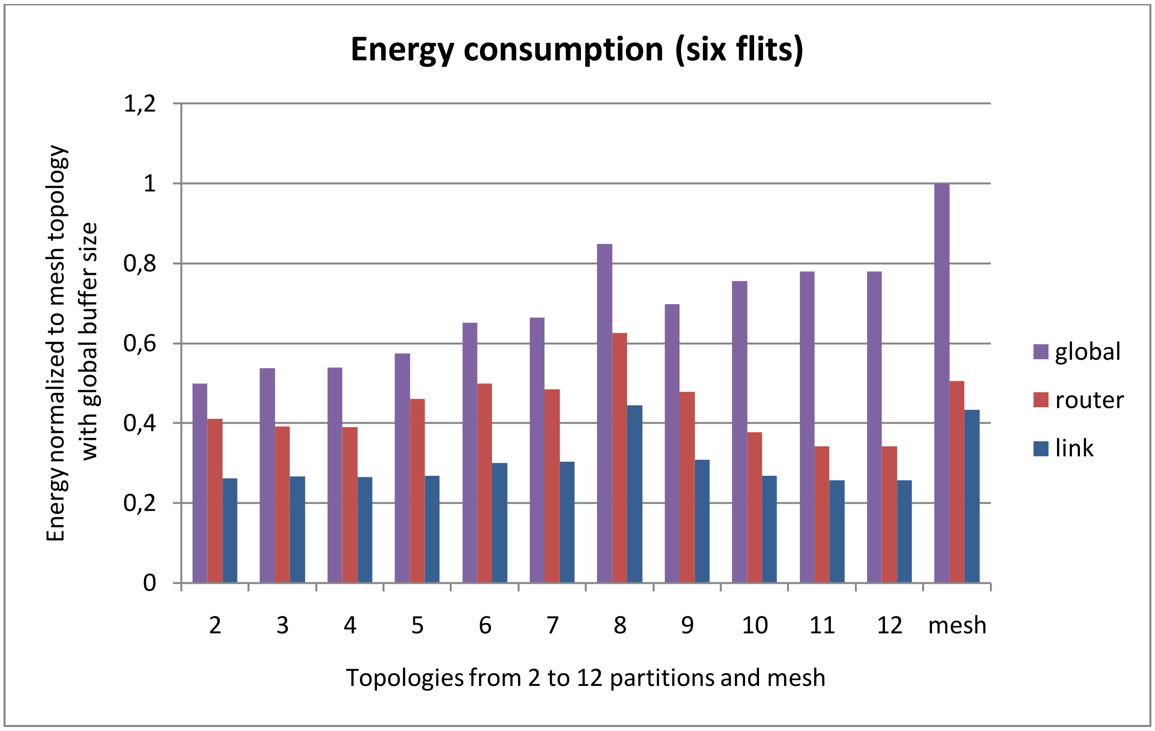


Figure 30. Energy consumption for different buffer slots optimizations

4.4.5 VOPD power gain

Figures 31, 32 and 33 present the power gains in every case. Maximum power gains through router optimization are 69.36, 59.31 and 56.25 for four, five and six flits. Accordingly, maximum gains through link optimization are 76.72, 68.19 and 66.99. Higher gain for four flits is justified, because when the traffic load is low more buffer slots can be removed, so less energy is consumed. Average gains per router (and per link) are 55.47% (69.75%), 41.02% (61.33%) and 33.04% (55.70%) for four, five and six flits accordingly.

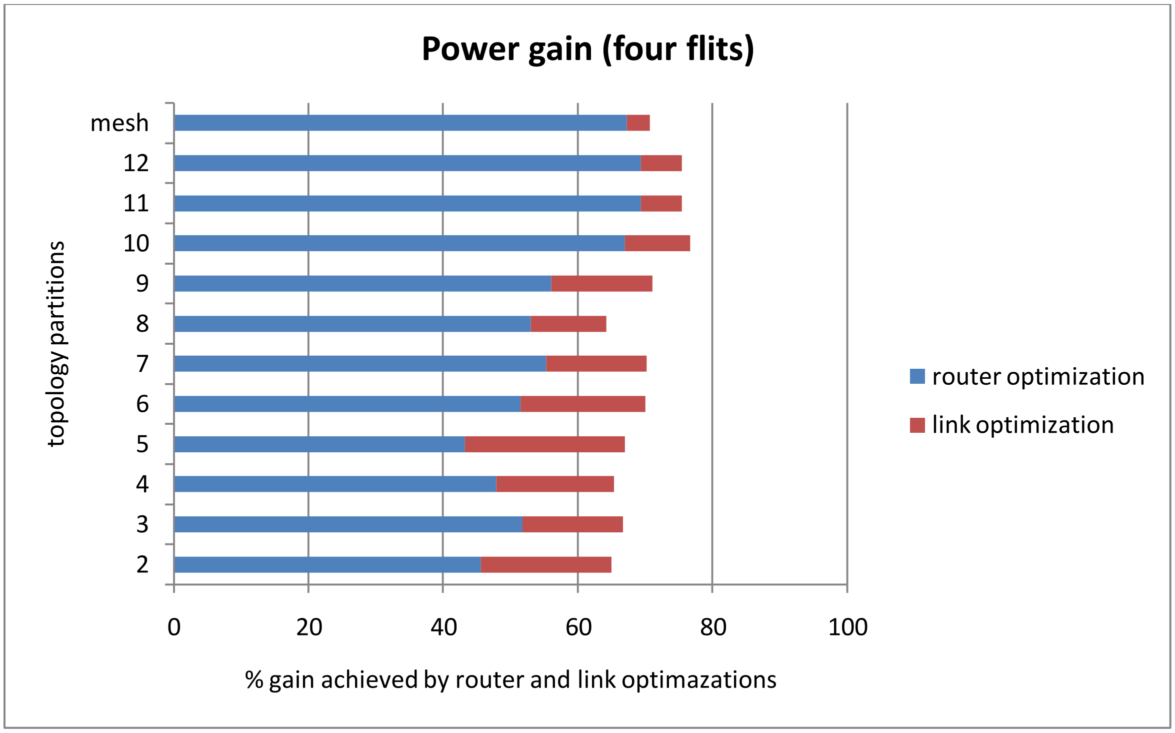


Figure 31. Power consumption reduction through buffer slots optimization

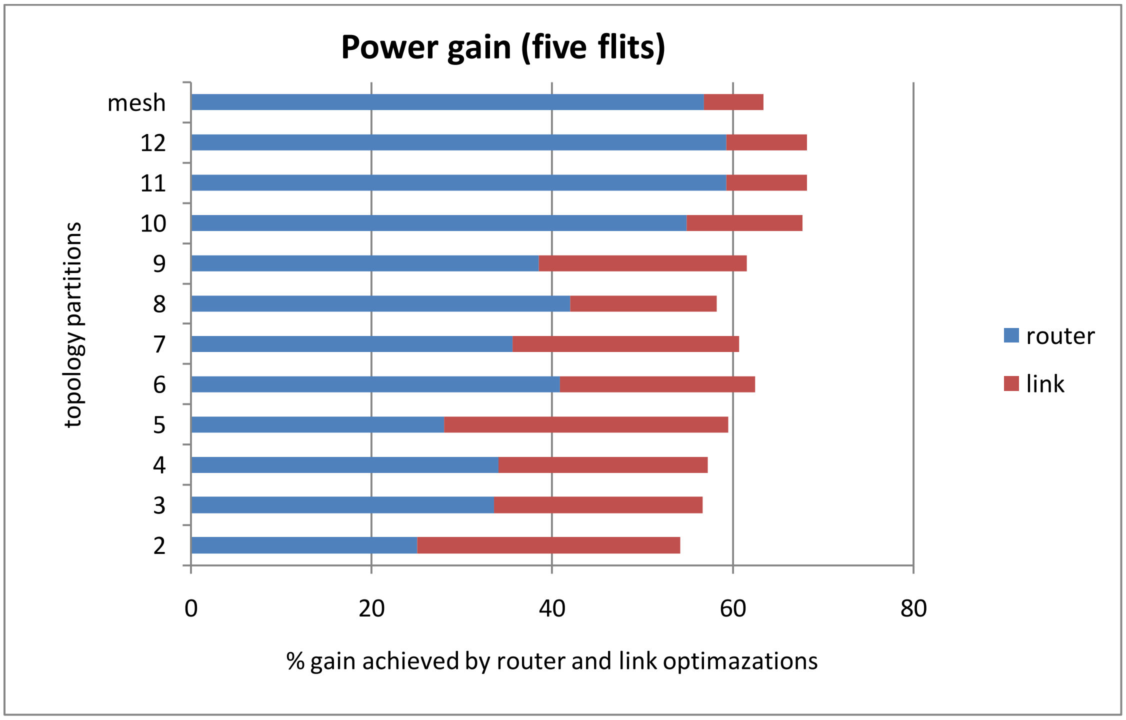


Figure 32. Power consumption reduction through buffer slots optimization

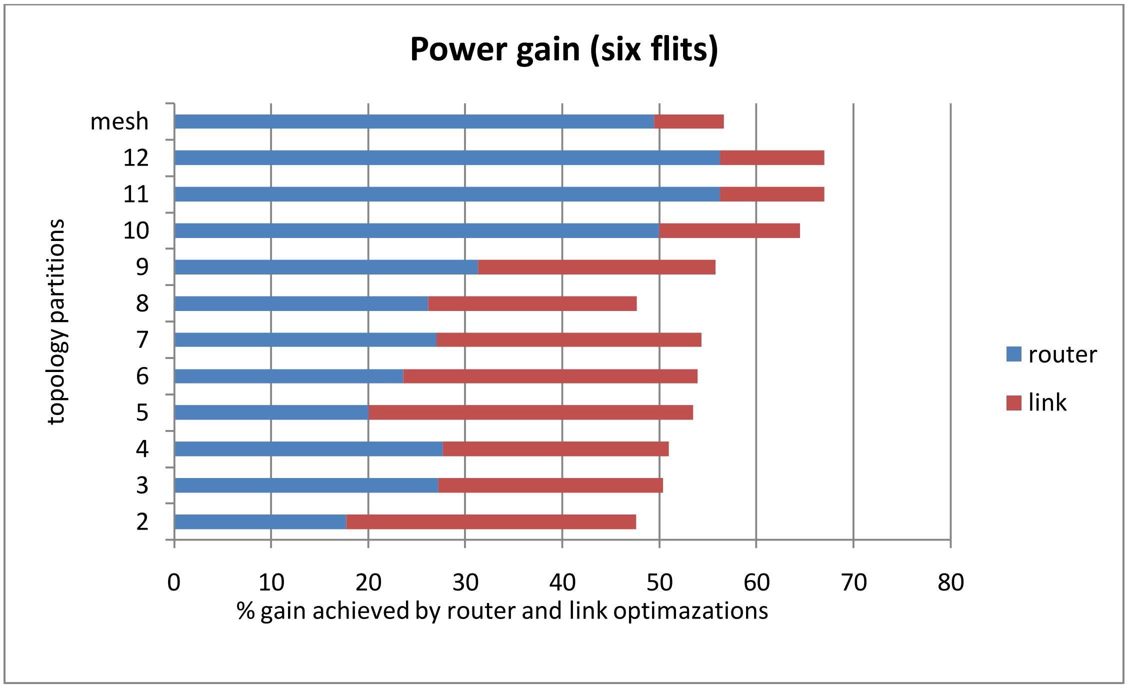


Figure 33. Power consumption reduction through buffer slots optimization

4.5 MWD

Multi window display is another digital signal processing application (TAM, 2005), which is also suitable for NoC architectures and also uses twelve processing elements. The different partitions produced by CHACO and the same method used for the simulation. Again, simulation results for an optimized mapped mesh topology are obtained and values are normalized to the values of mesh topology. Four, five and six flits per packet choices are simulated to collect simulation results.

4.5.1 MWD delay

Global average delay as a function of topology is presented in figure 34. In the horizontal axis there are the number of partitions and the mesh topology at the end and corresponding values of delay are on the vertical axis. Lower delay is achieved for lower traffic load. All delay values are lower than delays on mesh topology. The best performance is achieved when nodes are divided to two partitions for four flits packets and three partitions for five and six flits packets. This happens, because the traffic can be efficiently handled by a few routers as in this application the need for communication is lighter than in previous applications. Minimum delay for four flits is 0.64, for five flits 0.69 and for six flits 0.71. The average of average delay for four flits is 0.76, for five flits 0.78 and for six flits 0.81.

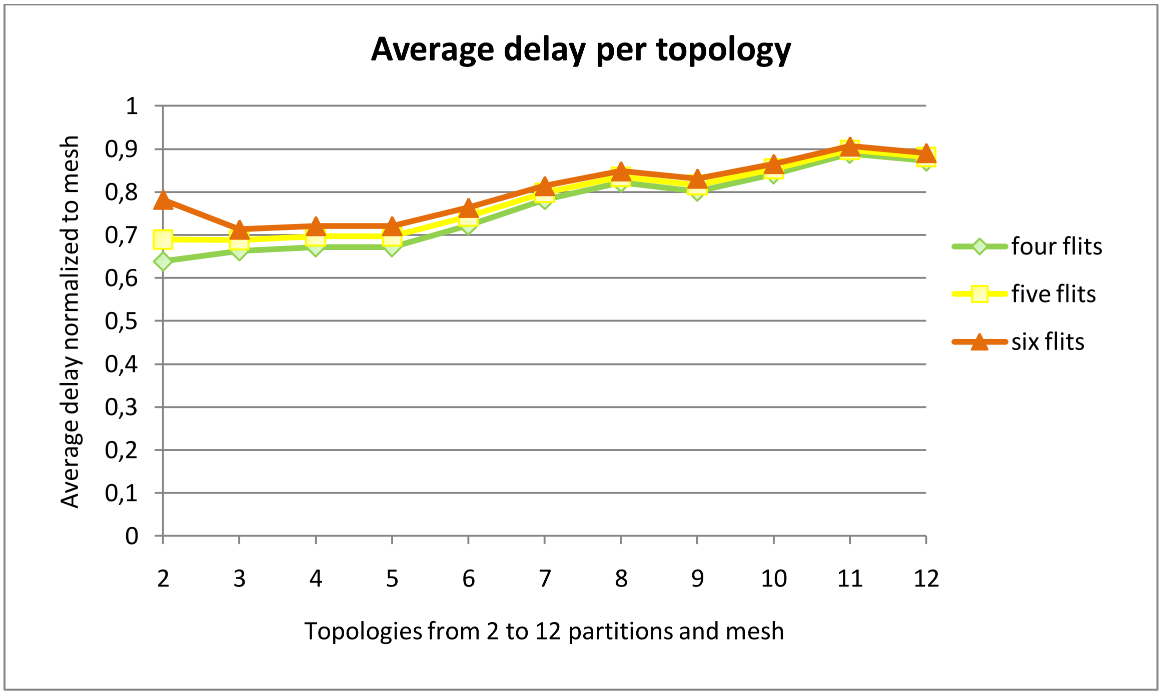


Figure 34. Average delay for four, five and six flits per packet

4.5.2 MWD throughput

Also, better throughput is achieved when few partitions and thus few routers are used as shown in Figure 35. Again for each curve values are normalized to the mesh topology. Highest throughput is achieved for two partitions for every load and maximum values are 0.19, 0.24 and 0.28 for three, four and five flits. Average values of throughput are 0.43, 0.54 and 0.65 accordingly. As expected the throughput is higher when more flits are transmitted for every packet.

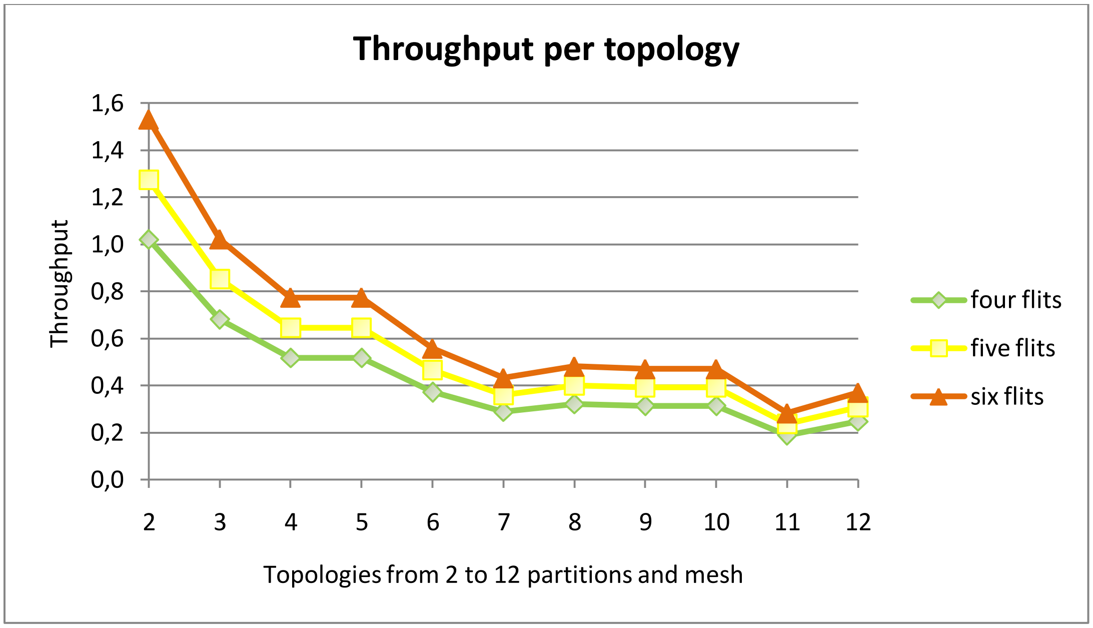


Figure 35. Average throughput for three, four and five flits per packet

4.5.3 MWD maximum delay

Maximum delays counted in clock cycles are shown in figure 36. The maximum values on the chart are 27, 21 and 25 and average maximum delay is 11.73, 13.91 and 19.09 for four, five and six flits. Generally, as the number of routers increases the number of hops for each packet increases also, so expect higher delay for higher partitions.

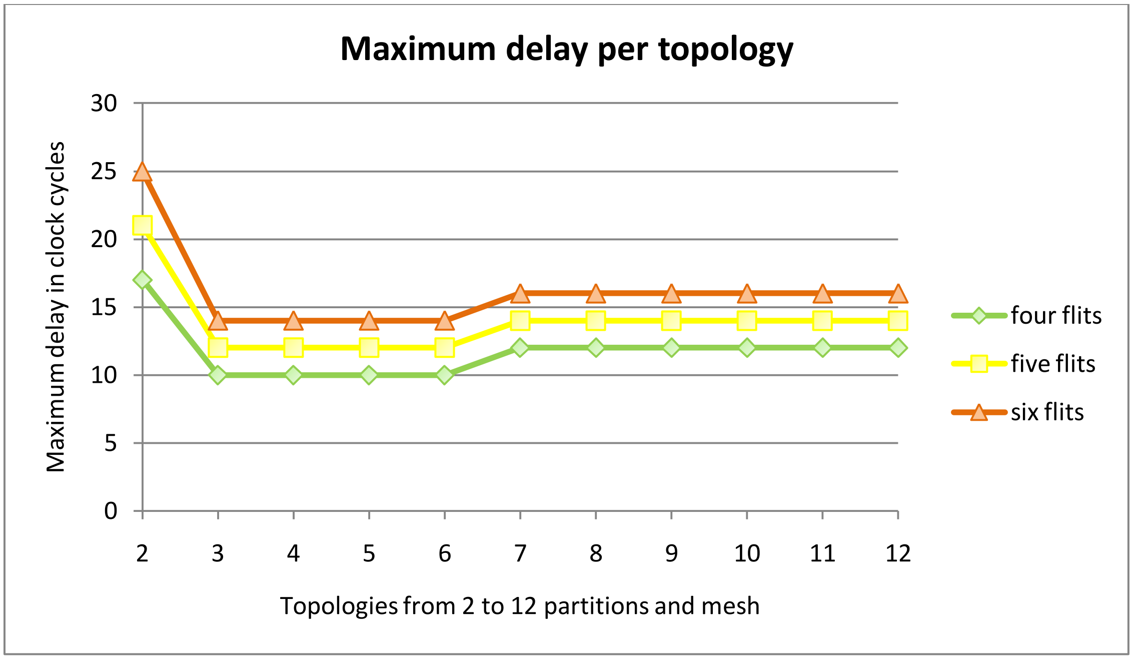


Figure 36. Maximum delays for four, five and six flits

4.5.4 MWD energy consumption

Energy consumption values for every packet size are figured in figures 37, 38 and 39. The values are normalized to the value of energy consumption on mesh topology with nine slots buffer. The global column presents the energy when nine slots are assumed per buffer; the router column presents the energy for optimized number of buffer slots per router and the link column optimizes buffer slots for each connection on the NoC. The difference between router and link optimization is that in the first case the maximum number of slots of every link on the router is used for each router, as explained in section 4.2. Minimum value 0.37 is achieved for partitioning number 6 for four flits, 0.39 for partition 2 for five flits and 0.41 for partition 6 for six flits. Average values are 0.10, 0.13 and 0.15 accordingly.

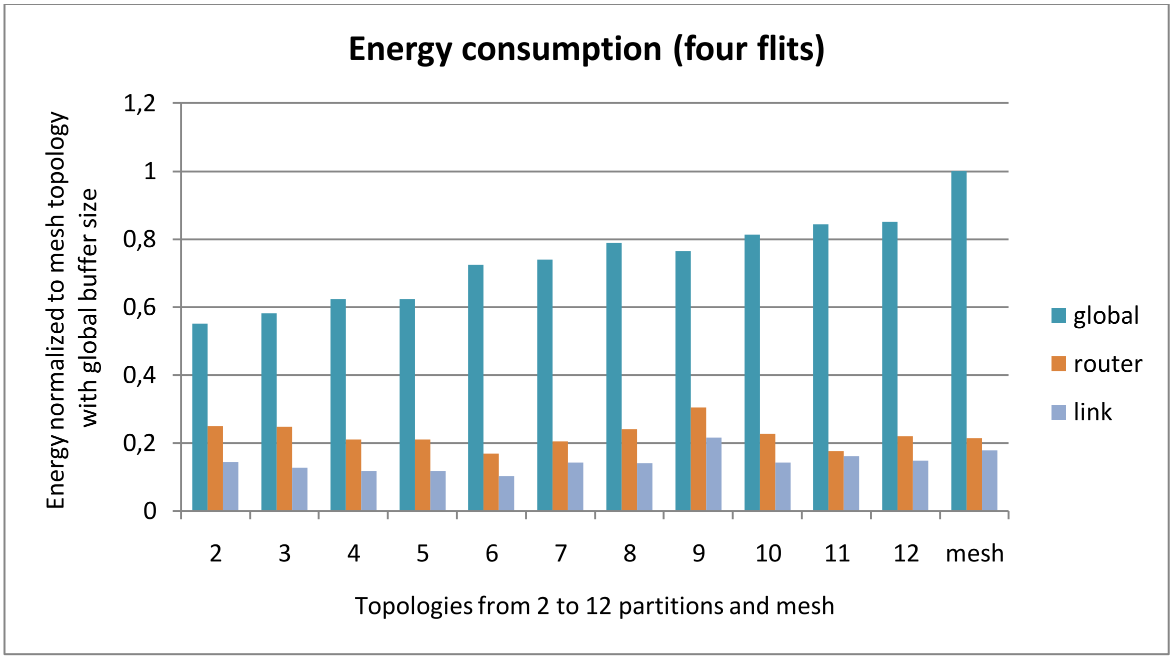


Figure 37. Energy consumption for different buffer slots optimizations

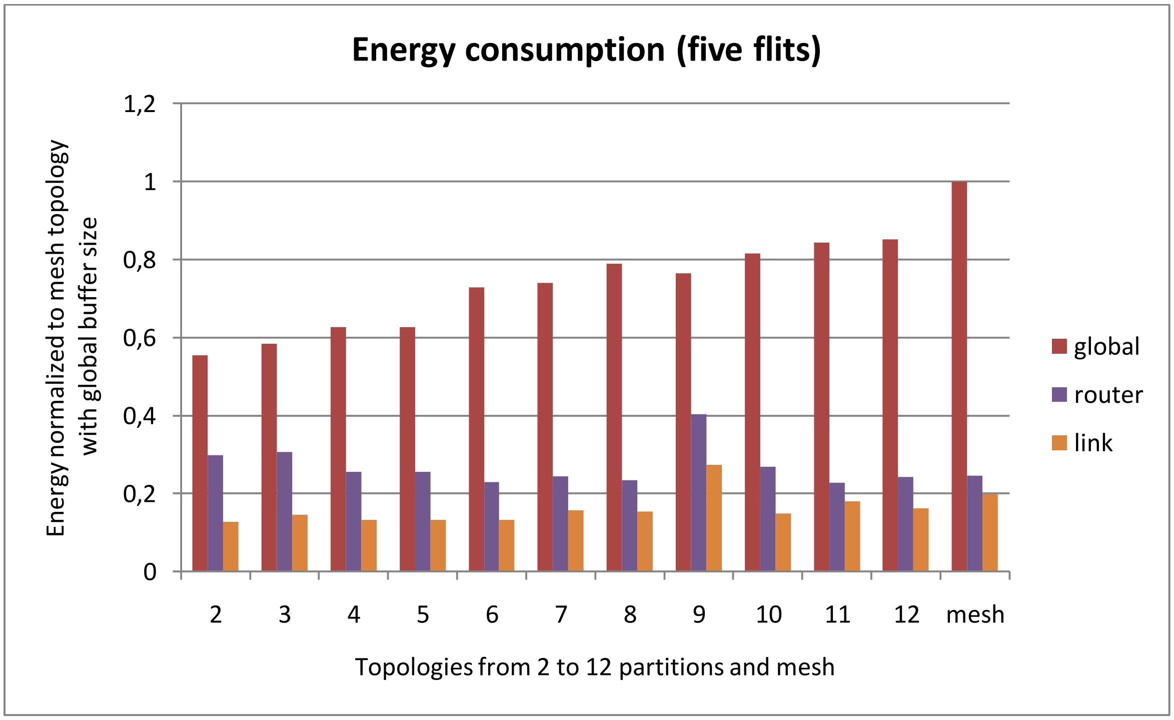


Figure 38. Energy consumption for different buffer slots optimizations

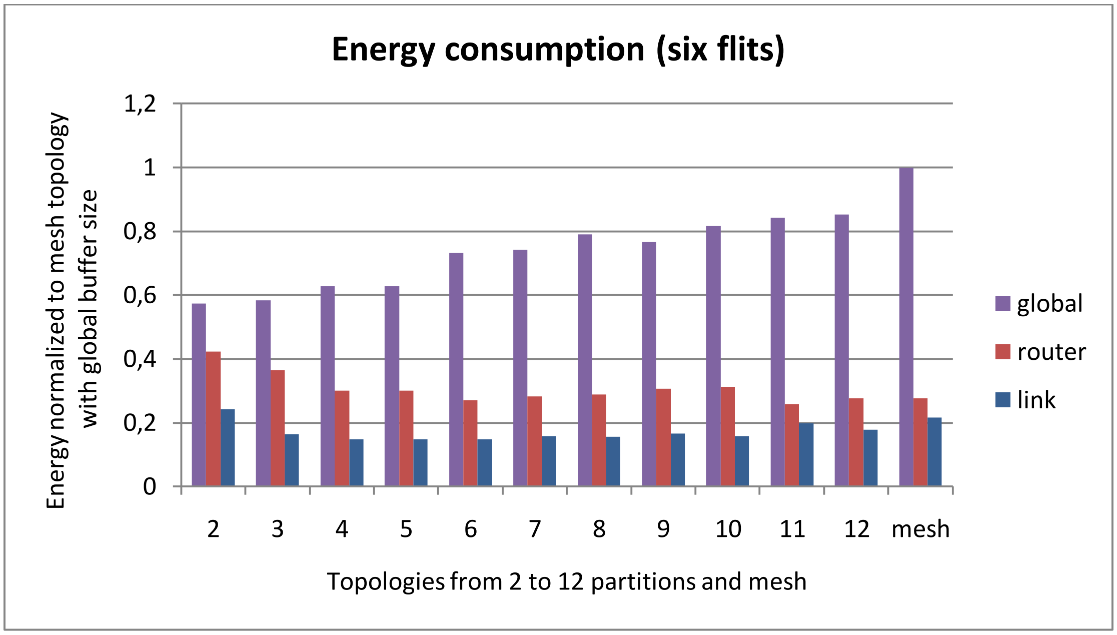


Figure 39. Energy consumption for different buffer slots optimizations

4.5.5 MWD power gain

Figures 40, 41 and 42 present the power gains in every case. Maximum power gains through router optimization are 79.15, 75.47 and 72.39 for four, five and six flits. Accordingly, maximum gains through link optimization are 85.71, 81.84 and 80.75. Higher gain for four flits is justified, because when the traffic load is low more buffer slots can be removed, so less energy is consumed. Average gains per router (and per link) are 68.13% (80.12%), 61.58% (77.85%) and 55.89% (76.00%) for four, five and six flits accordingly.

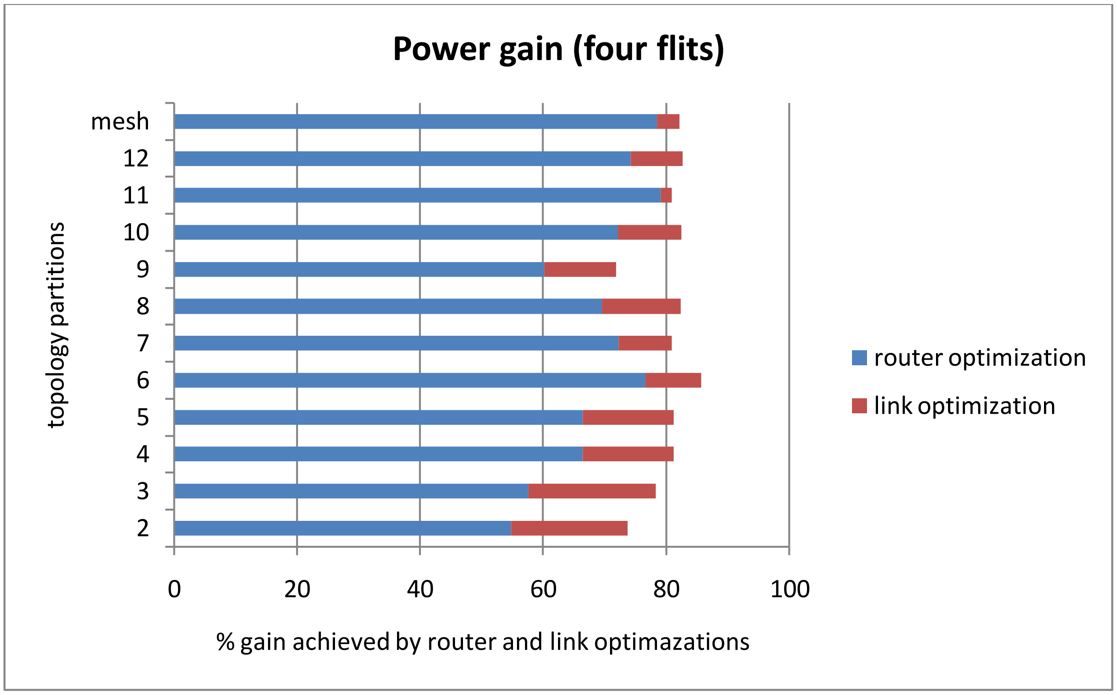


Figure 40. Power consumption reduction through buffer slots optimization

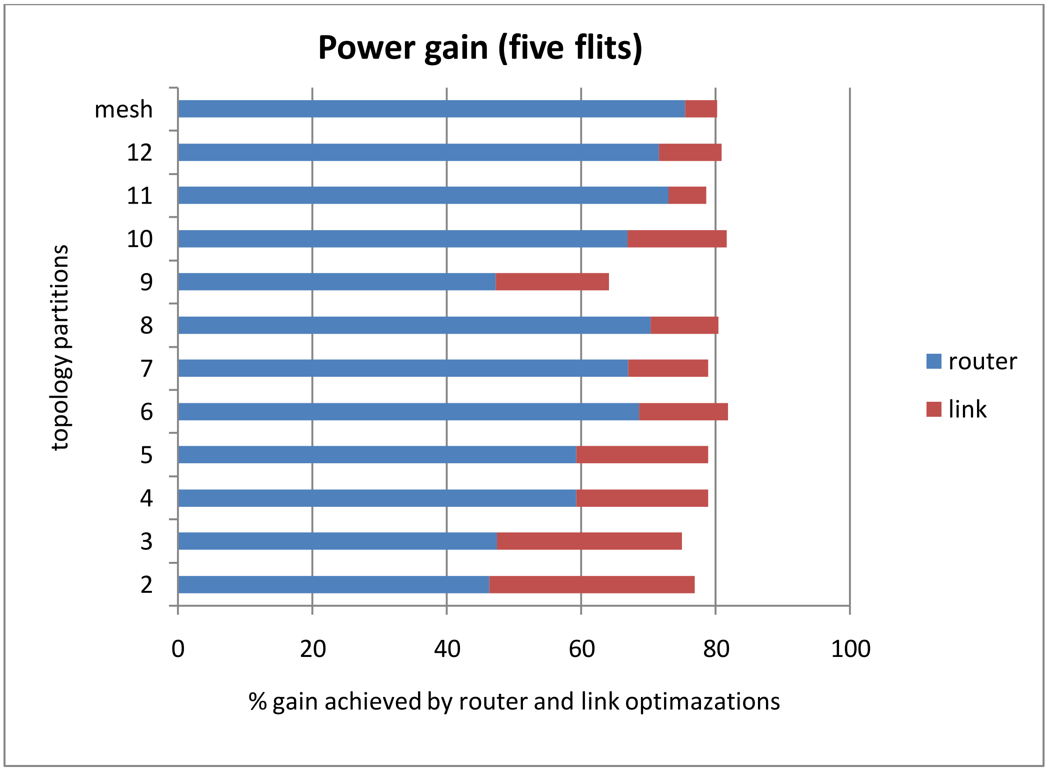


Figure 41. Power consumption reduction through buffer slots optimization

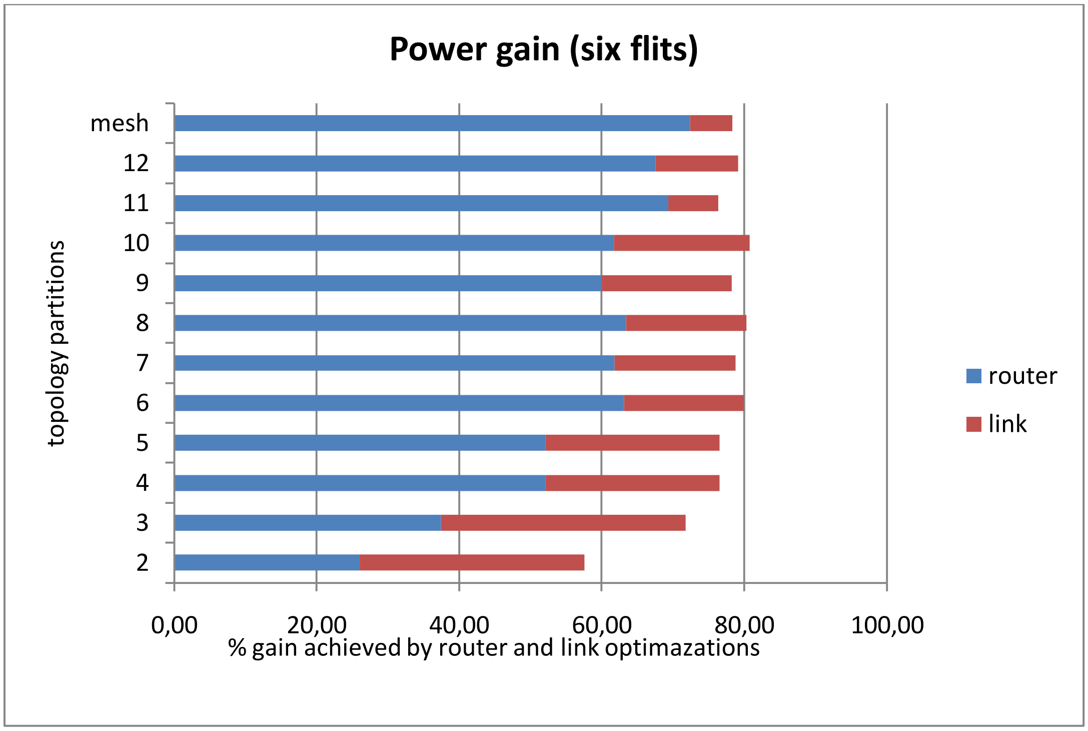


Figure 42. Power consumption reduction through buffer slots optimization

4.6 MMS

In this section a multimedia system (MMS) is tested. The system contains 25 components, including several memories and DSP processors. Most of the different partitions produced by CHACO and range from 2 to 25 are simulated. Again, simulation results for an optimized mapped mesh topology are obtained and values are normalized to the values of mesh topology. Four, five and six flits per packet choices are simulated to collect simulation results.

4.6.1 MMS delay

Global average delay as a function of topology is presented in figure 43. In the horizontal axis there are the number of partitions and the mesh topology at the end and corresponding values of delay are on the vertical axis. Lower delay is achieved for lower traffic load. All delay values are lower than delays on mesh topology. Minimum delay for four flits is 0.52, for five flits 0.56 and for six flits 0.62. The average of average delay for four flits is 0.62, for five flits 0.65 and for six flits 0.70.

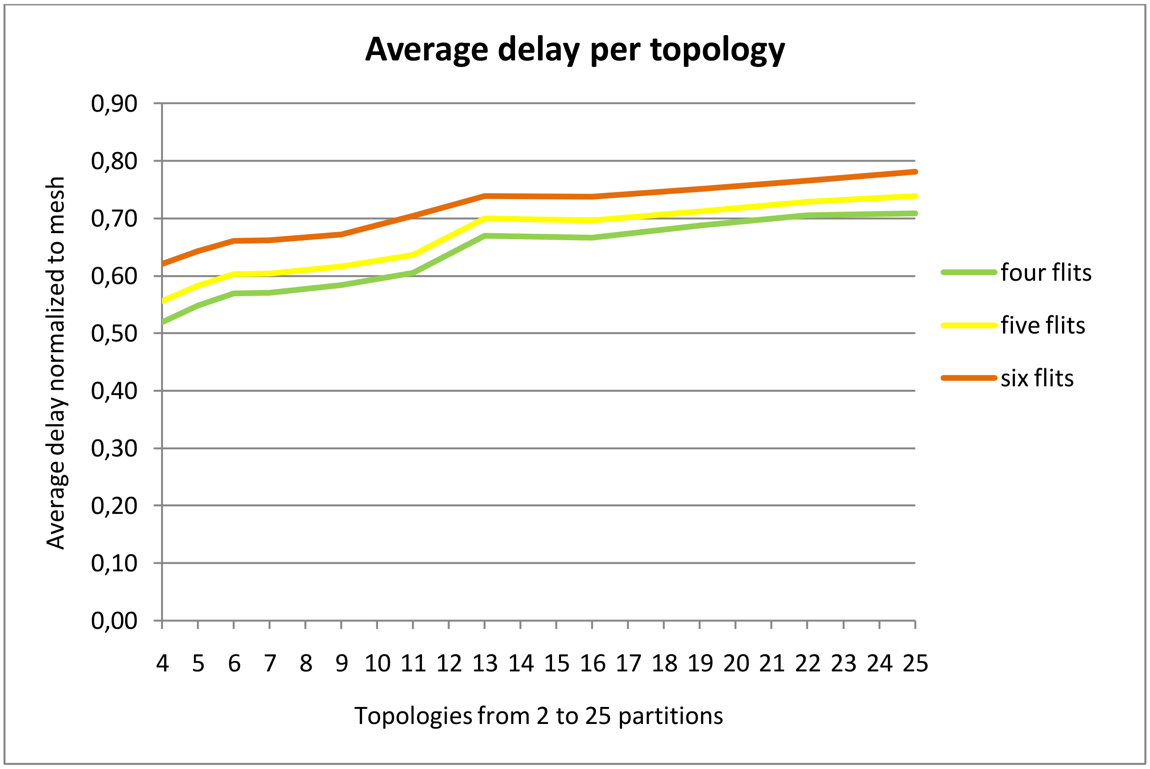


Figure 43. Average delay for four, five and six flits per packet

4.6.2 MMS throughput

Better throughput is achieved when few partitions and thus few routers are used as shown in Figure 44. Again for each curve values are normalized to the mesh topology. Highest throughput is achieved for two partitions for every load and maximum values are 0.36, 0.45 and 0.55 for three, four and five flits. Average values of throughput are 0.20, 0.25 and 0.30

accordingly. As expected the throughput is higher when more flits are transmitted for every packet.

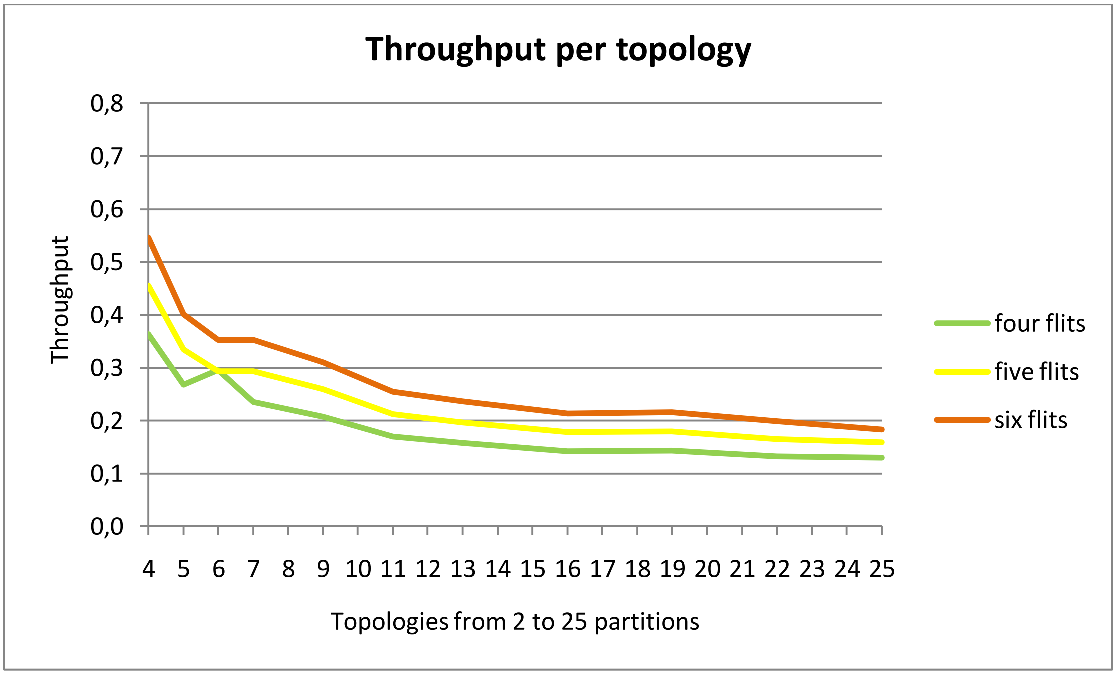


Figure 44. Average throughput for three, four and five flits per packet

4.6.3 MMS maximum delay

Maximum delays counted in clock cycles are shown in figure 45. The maximum values on the chart are 18, 22 and 42 and average maximum delay is 16.45, 20.10 and 27.82 for four, five and six flits. Generally, as the number of routers increases the number of hops for each packet increases also, so expect higher delay for higher partitions.

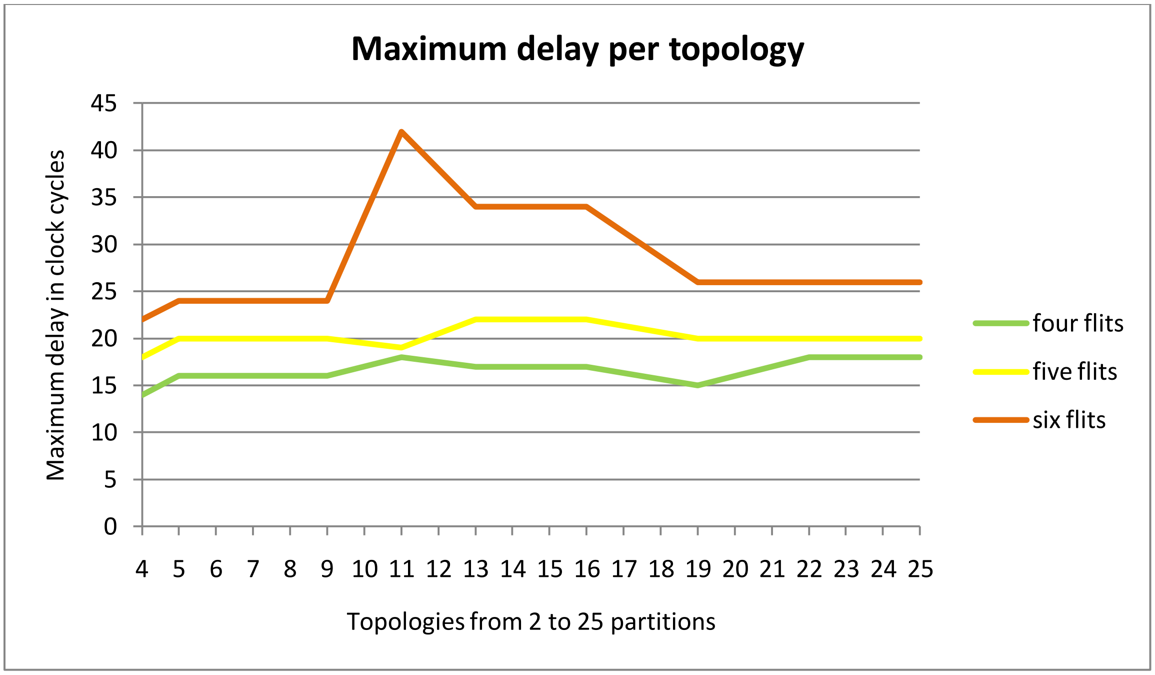


Figure 45. Maximum delays for four, five and six flits

4.6.4 MMS energy consumption

Energy consumption values for every packet size are figured in figures 46, 47 and 48. The values are normalized to the value of energy consumption on mesh topology with nine slots buffer. The global column presents the energy when nine slots are assumed per buffer; the router column presents the energy for optimized number of buffer slots per router and the link column optimizes buffer slots for each connection on the NoC. The difference between router and link optimization is that in the first case the maximum number of slots of every link on the router is used for each router, as explained in section 4.2. Minimum value 0.10 is achieved for partitioning number 5 for four flits, 0.17 for partition 4 for five flits and 0.21 for partition 4 for six flits. Average values are 0.30, 0.34 and 0.37 accordingly.

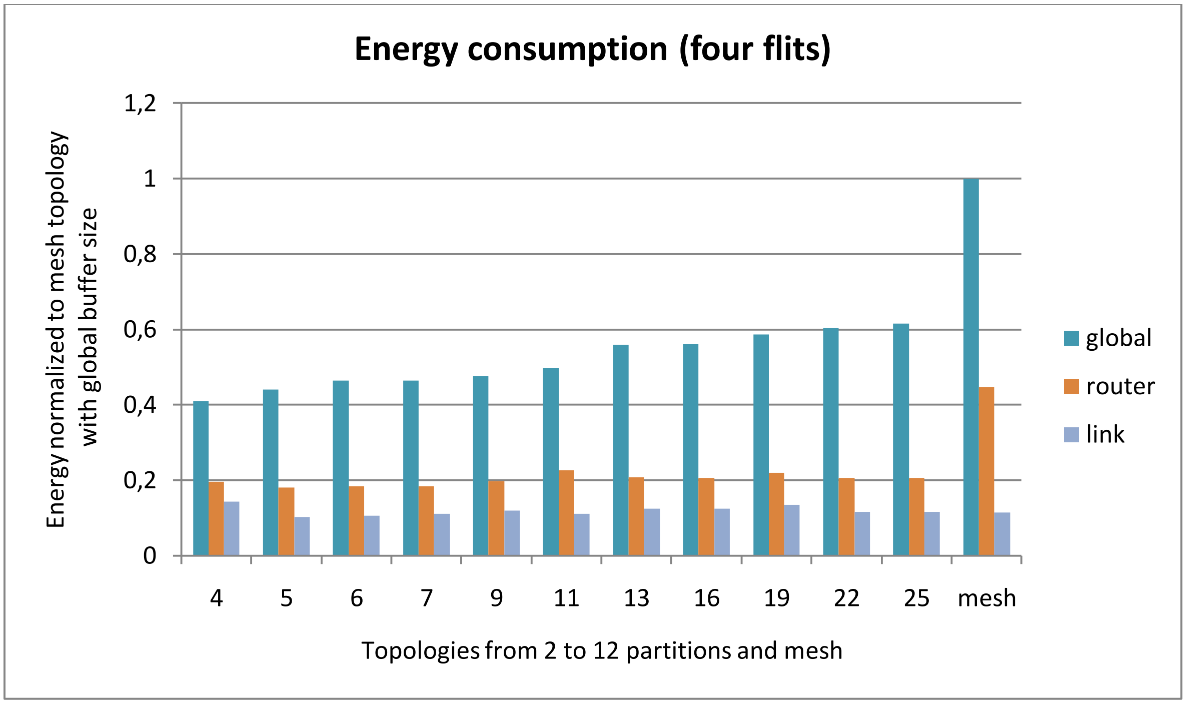


Figure 46. Energy consumption for different buffer slots optimizations

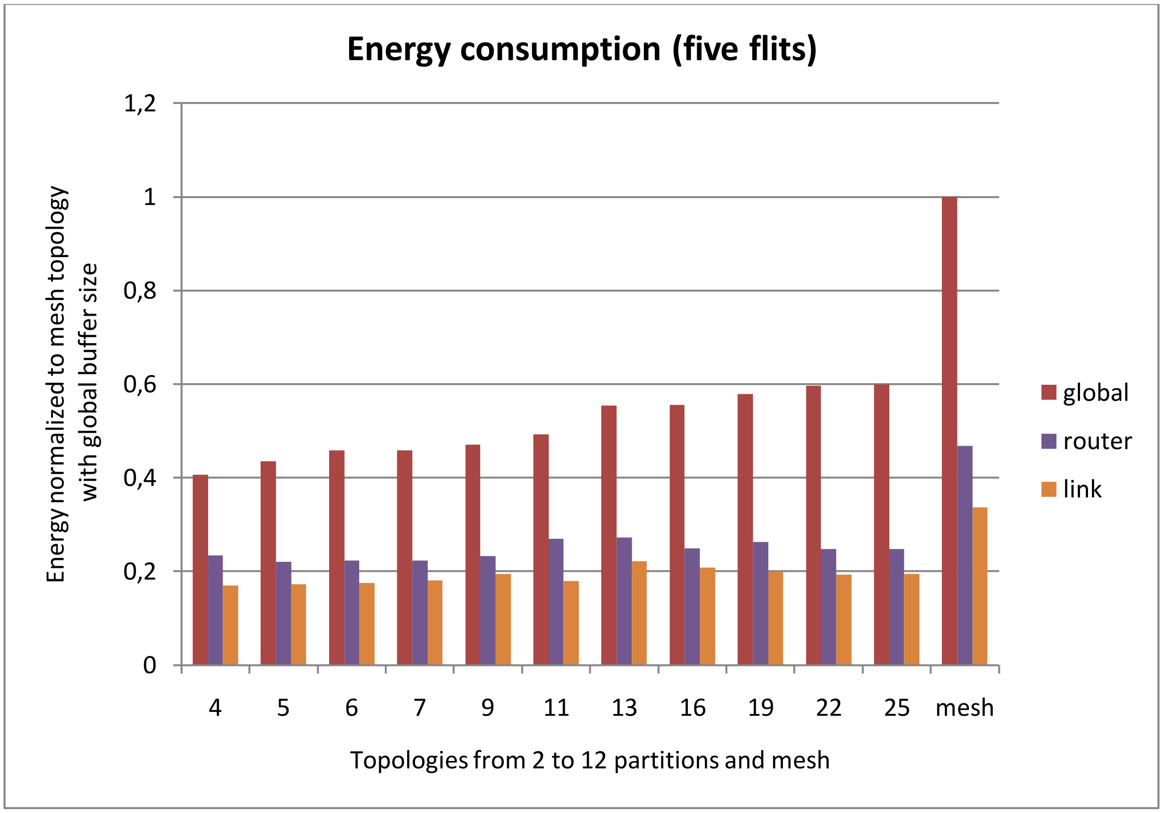


Figure 47. Energy consumption for different buffer slots optimizations

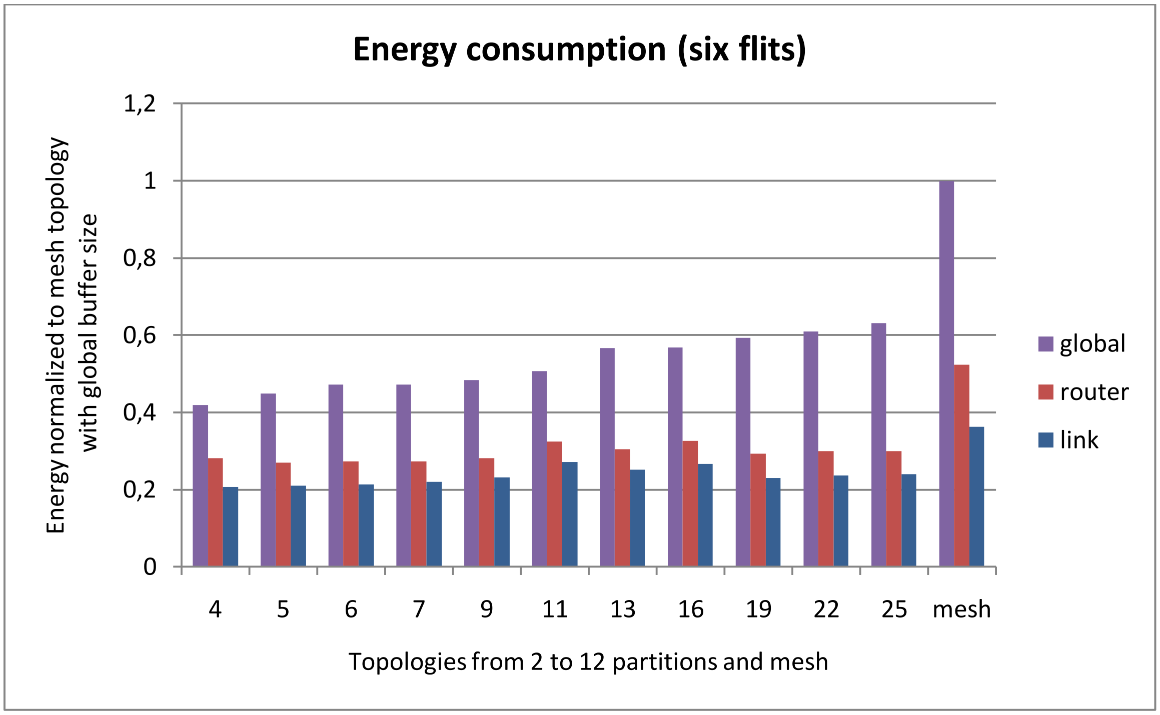


Figure 48. Energy consumption for different buffer slots optimizations

4.6.5 MMS power gain

Figures 49, 50 and 51 present the power gains in every case. Maximum power gains through router optimization are 66.44, 58.62 and 52.49 for four, five and six flits. Accordingly, maximum gains through link optimization are 73.88, 67.61 and 63.69. Higher gain for four flits is justified, because when the traffic load is low more buffer slots can be removed, so less energy is consumed. Average gains per router (and per link) are 60.56% (68.53%), 51.63% (62.33%) and 43.47% (54.83%) for four, five and six flits accordingly.

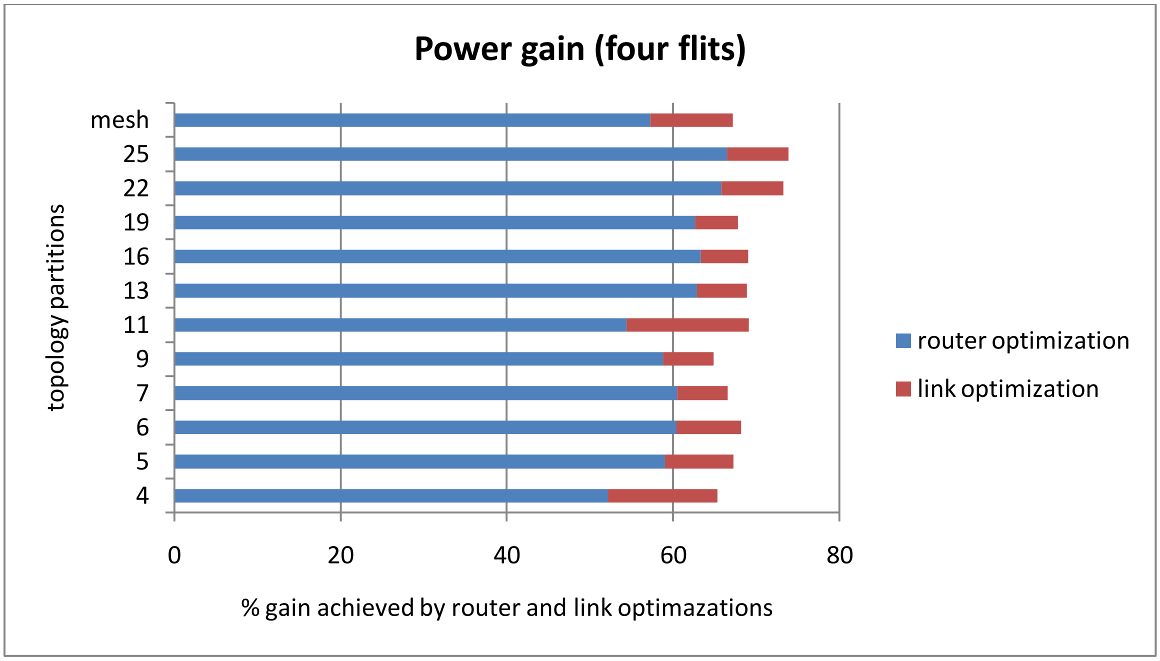


Figure 49. Power consumption reduction through buffer slots optimization

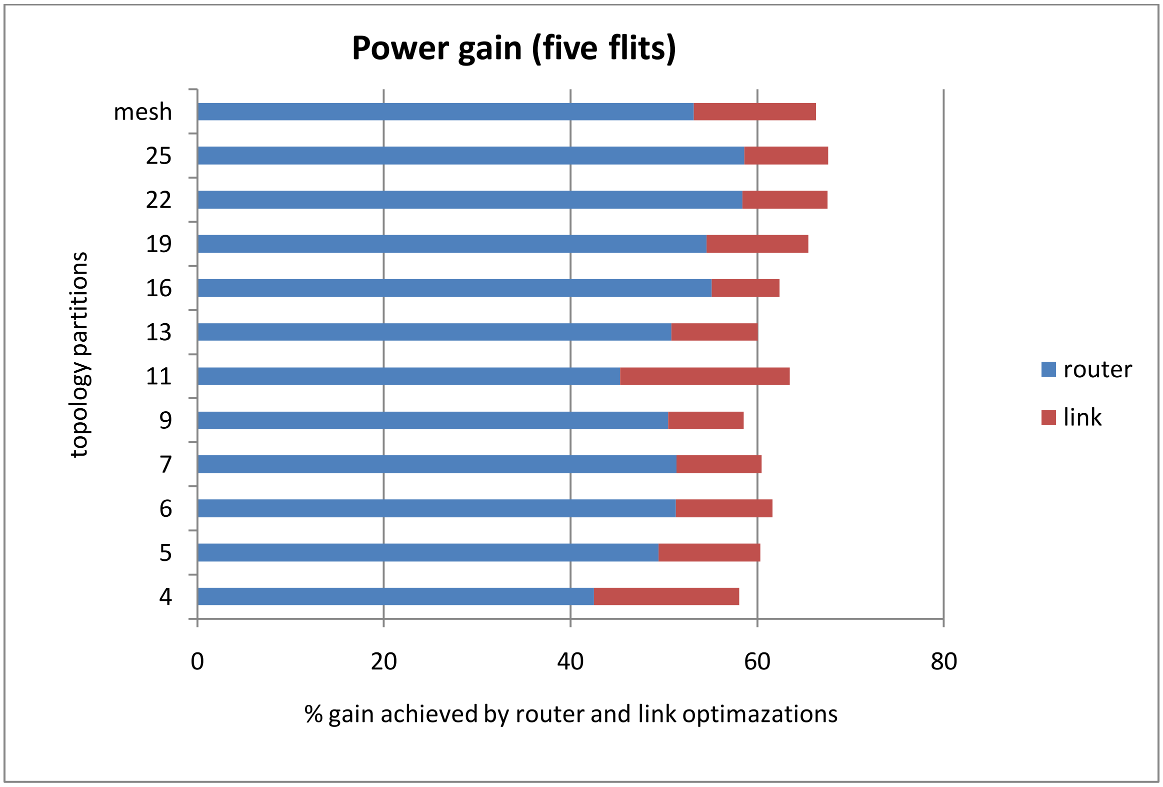


Figure 50. Power consumption reduction through buffer slots optimization

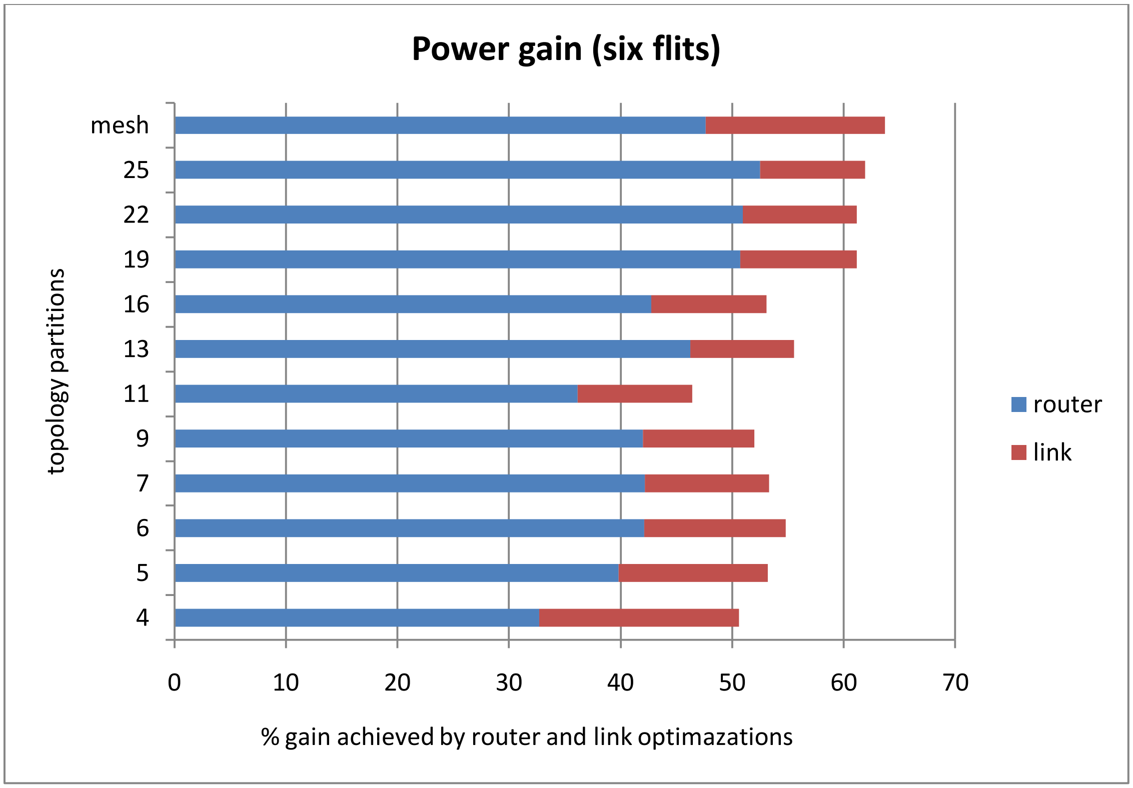


Figure 51. Power consumption reduction through buffer slots optimization

Chapter 5

Conclusions and future work

5.1 Summary

The simulation results make it clear that key metrics, like latency and energy, heavily depend on applications executed on the NoC. Thus, specific NoC can be designed to achieve great performance, but only for a specific application. Generally, when the traffic generated by the nodes of application is well distributed around the network designer can achieve important power gains by buffer optimization. Well distributed traffic can be realized when irregular NoC topologies are used most of the times. However, as shown in the simulation results, for applications like MPEG-4 with 5 flits per packet point to point connections with dedicated router for each node are optimum, because the traffic load is too heavy for the network approach.

Some specific conclusions for each one of the studied applications can be done. For VOPD and MWD applications it is clear that irregular NoCs with many components connected to a few routers offer great advantages compare to regular mesh topologies. Good simulation results are produced for every packet size, because both applications do not produce intense communication on the network. On the other hand, MPEG-4 is an application that stresses the network and thus more routers with many buffer slots are needed in order to handle the produced traffic. This is shown in simulation results especially when five flits are generated for every packet. So, designers should choose more routers in this case to minimize the delay. Lastly, MMS application can be handled easily with irregular NoCs and a few routers. Keeping number of routers relatively small boosts the performance and minimizes the energy consumption in this case.

5.2 Future work

Several topics can be chosen for future work in the field of system-on-chip design. In the following two sections there is a presentation of ideas arise from this work. More complex models can always be proposed in order to offer more realistic simulation results. During this work detailed communication models have been developed and inserted to simulation tool. However, some areas need further study, such as router switching latency which is not counted as routing decisions are made by using high level C++ code. Also, the routing tables are automatic generated in high level, but this could also be done in low level and each router could explore its connections but sending packets to discover its neighbors.

5.2.1 Automatic RTL irregular topology generation

The simulation tool should be expanded to offer output compatible for RTL implementation. As explained in chapter 2, all design stages are connected and simulation results from one stage are used in next stage. The development of simulation tool in this thesis primary focuses on NoC communication architecture analysis and optimization and help the designer choose the optimum topology. RTL irregular topology generation will be useful to the next design stage (NoC design validation and synthesis) and reduce the design time in that stage.

5.2.2 Design space exploration

This thesis focuses on performance metrics and energy consumption of the network. However, an important cost metric of system design is the space needed to implement the design. So, design space exploration during simulation time is vital, in order to achieve several goals in synthesis level, which will follow later. To perform design space simulation, space models should be studied for routers and processing elements and inserted to the simulation tool. Space occupied by each router should be a function of its connections and the buffer slots dedicated to each connection and different space should be awarded to different components, for example different space models should be used for memories and processors.

5.2.3 Thermal exploration on 3D NoC

Another aspect of system design that should be studied is 3D integration of the chip. Several researches have been done during the past years and 3D NoCs seems to offer revolutionary features in design. Moreover, a new methodology can be developed in order to find the optimum mapping of applications to 3D NoC and techniques to overcome temperature and other problems that maybe arise from this design should be studied. One of the most critical challenges for implementing designs in 3D NoCs is the power management, and hence the thermal problem, which has already been studied for 2D architectures. This problem is exacerbated in 3D architectures for two reasons: (i) the vertically stacked layers cause a rapid increase of power density and (ii) the thermal conductivity of the dielectric inserted between device layers for insulation is very low compared to silicon and metal.

Temperature reduction and hotspot minimization is of extreme importance to modern systems because it affects the overall power consumption, performance and reliability (e.g., MTTF) of the device. For these reasons, the reduction of peak temperature is something that the designer should take into consideration. To achieve this on a NoC System, the designer can tweak the interconnection network in order to reduce the number of hotspots.

5.2.4 Mapping techniques

Simulation results reveal the importance of mapping of nodes on the network. Performance boost can be achieved when an application is efficiently mapped to a network, in a way that nodes that communicate a lot with each other are closer than others. However, optimum mapping for one application will make the network ideal only for one application or a group of relatively applications. So, there is a strong connection between software application and hardware designed to run this application. This can set an important limitation in design flow. Future work may solve this issue by developing generic techniques to achieve efficient mapping without much effort from the NoC designer.

References

(BJE, 2006) Tobias Bjerregaard and Shankar Mahadevan. A survey of research and practices of network-on-chip. ACM Comput. Surv., 2006.

(ANA, 2008) Ηρακλής Ν. Αναγνωστόπουλος. Υποστήριξη παροχής ποιότητας υπηρεσιών (Quality of Service - QoS) σε δίκτυα σε ψηφίδα (Network-on-Chip - NoC), 2008

(MIC, 2009) Giovanni De Micheli. An Outlook on Design Technologies for Future Integrated Systems. IEEE transactions on computer-aided design of integrated circuits and systems, vol. 28, no. 6, June 2009

(MAR, 2007) Theodore Michel Marescaux. Mapping and Management of Communication Services on MP-SoC Platforms. Technische Universiteit Eindhoven, 2007.

(NOL, 2006) Vincent Nollet, Theodore Marescaux, Diederik Verkesty. Operating System Controlled Network on Chip. ACM, 2006

(MAR, 2009) Radu Marculescu, Umit Y. Ogras, Li-Shiuan Peh, Natalie Enright Jerger, and Yatin Hoskote. Outstanding IEEE transactions on computer-aided design of integrated circuits and systems, vol. 28, no. 1, January 2009

(JAL, 2004) Antoine Jalabert, Srinivasan Murali, Luca Benini, and Giovanni De Micheli. xpipesCompiler: A tool for instantiating application specific Networks on Chip. In: Design, Automation and Test in Europe (DATE) , Paris, France, February 2004.

Free On-Line Dictionary of Computing

(BHA, 2002) J.Bhasker. A SystemC Primer. Star Galaxy Publishing, 2002.

(KEE, 2005) Kees Goossens, JohnDielissen, and Andrei Radulescu. The æthereal Network on Chip: Concepts, Architectures, and Implementations. IEEE Designand Test of Computers, 2005.

(LEE, 2003) Se-Joong Lee, Seong-Jun Song, Kangmin Lee, Jeong-Ho Woo, Sung-Eun Kim, Byeong-Gyu Nam, Hoi-Jun Yoo, An 800MHz Star-Connected On-Chip Network for Application to Systems on a Chip, Int'l Solid-State Circuits Conference (ISSCC), San Francisco, Feburary, 2003.

(ADR, 2003) Adrijean Andriahantenaina and Alain Greiner. Micro-Network for SoC: Implementation of a 32-Port SPIN Network.DATE,2003.

(BON, 2006) L. Bononi and N. Concer. Simulation and analysis of network on chip architectures: ring, spidergon and 2d mesh. In: Design, Automation and Test in Europe,DATE, 2006.

(COP, 2004) M. Coppola, R. Locatelli, G. Maruccia, L. Pieralisi, and A. Scandurra.Spidergon: a novel on-chip communication network. In: System-on-Chip,Proceedings,International Symposium, 2004.

(GUE, 2000) Pierre Guerrier and Alain Greiner. A generic architecture for on-chip packet-switched interconnections.DATE, ACM Press, 2000.

(BJE, 2005) Tobias Bjerregaard. The MANGO clockless network-on-chip: Concepts and implementation . PhD thesis, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, 2005.

(MIL, 2004) Mikael Millberg, Erland Nilsson, Rikard Thid, Shashi Kumar, and Axel Jantsch. The Nostrum backbone - a communication protocol stack for networks on chip. In: Proceedings of the VLSI Design Conference , Mumbai, India, 2004.

(GOO, 2005) Kees Goossens, JohnDielissen, and Andrei Radulescu. The æthereal Network on Chip: Concepts, Architectures, and Implementations. IEEE Design and Test of Computers, 2005.

(HAN, 2005) Andreas Hansson, Kees Goossens, and Andrei R˘ adulescu. A Unified Approach to Constrained Mapping and Routing on Network-on-Chip Architectures. Int'l Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS) , 2005.

(DIE, 2005) Kees Goossens, John Dielissen, Om Prakash Gangwal, Santiago Gonzalez Pestana, Andrei Radulescu, and Edwin Rijpkema. A design flow for application-specific networks on chip with guaranteed performance to accelerate SOC design and verification.: Proc. Design, Automation and Test in Europe Conference and Exhibition (DATE), 2005.

(MAR, 2004) Radu Marculescu, Umit Y. Ogras, and Nicholas H. Zamora.Computation and communication refinement for multiprocessor SoC design: A system-level perspective. In Proc. of DAC, pages 564–592.ACM Press, 2004.

(LU, 2006) Z. Lu, M. Zhong, and A. Jantsch, Evaluation of on-chip networks using deflection routing, *GLSVLSI*, 2006.

(HU, 2002) J. Hu, Y. Deng, and R. Marculescu, System-level point-to-point communication synthesis using floor planning information, *ASPDAC/VLSI*, 2002.

(PEN, 2006) S. Penolazzi and A. Jantsch, A high level power model for the Nostrum NoC, *DSD*, 2006.

(BER, 2005) D. Bertozzi *et al.*, Noc synthesis flow for customized domain specific multiprocessor systems-on-chip, *IEEE Trans. Parallel and Distributed Systems*, 2005.

(PAN, 2005) P. Pande *et al.*, Performance evaluation and design trade-offs for network-on-chip interconnect architectures, *IEEE Trans. Computers*, 2005.

(SAL, 2007) Erno Salminen, Ari Kulmala, and Timo D. Hamalainen, On network-on-chip comparison, 10th Euromicro Conference on Digital System Design Architectures, Methods and Tools (DSD 2007), IEEE, 2007.

(MUR, 2005) S. Murali and G. D. Micheli, “Bandwidth-constrained mapping of cores onto NoC architectures,” in Proc. of DATE 2005.

(TAM, 2005) D. Bertozzi, A. Jalabert, S. Murali, R. Tamhankar, S. Stergiou, L.Benini, and G. De Micheli, NoC synthesis flow for customized domain specific multiprocessor systems-on-chip, IEEE TPDS, 2005.

(BAR, 2007) A. Bartzas, N. Skalis, K. Siozos and D. Soudris: Exploration of Alternative Topologies for Application-Specific 3D Networks-on-Chip, WASP, 2007