Design of complex integrated systems based on networks-on-chip
− Trading off performance, power and reliability −

Thesis submitted in partial fulfillment of
the requirements for the degree

of **Doktor-Ingenieur (Dr.-Ing.)** at the

Faculty of Computer Science and Electrical Engineering,
University of Rostock, Germany

Submitted by
Claas Cornelius,
from Rostock

Rostock, July 9, 2010

urn:nbn:de:gbv:28-diss2013-0161-5

Doctoral Advisor:

Prof. Dr.-Ing. Dirk Timmermann — University of Rostock, Germany

Additional Reviewers:

Prof. Dr.-Ing. Yiannos Manoli — University of Freiburg, Germany

Prof. Dr. sc.techn. Andreas Herkersdorf — Technische Universität München, Germany

Date of submission:

July 9, 2010

Date of defense:

March 25, 2011

Preface

This thesis originated during my time as scientific coworker at the Institute of applied Microelectronics and Data engineering (Institute MD), University of Rostock. Therefore, my sincere gratitude goes first and foremost to Prof. Dirk Timmermann who called my attention to the exciting field of microelectronics and who gave me the chance to write this thesis. With him as my doctoral advisor, I found myself in an excellent environment to develop own ideas while being attended with constructive criticism. This has been a great inspiration for my scientific and personal development. Moreover, I would like to thank all other professors and colleagues at the Institute MD for numerous fruitful discussions and assistant advice. Two of those colleagues shall be named in particular who already guided me as a student and who have been of invaluable help ever since. Thus, without Dr. Frank Grassert and Dr. Frank Sill this thesis would potentially not have been possible – but definitely not as exciting as it has been.

Beyond that, I would like to gratefully acknowledge those companies that provided detailed insights into their work and organizational structure. In conjunction with the interdisciplinary experiences within the post graduate program, this has remarkably opened my views of both research and society. Finally, I was in the fortunate position to receive feedback and helpful support due to diverse student theses. Hence, I would also like to thank those students who had the confidence in me as the tutor for their respective projects.

Last but not least, my sincere thanks go to my family and friends for their continuous support and patience. The time with them has prepared the ground to keep my ambitions, especially in stressful times. However, there are no words to express the significance of and gratefulness for my parents Karin and Dieter. With no doubt, it is their love and care that has actually brought me here. They not only gave me the freedom to follow my personal believes, but always encouraged me by any conceivable means.

Rostock, June 23, 2010 Claas Cornelius

Contents

INDICES

Chapter 1

Introduction

The concept of digital data has severely influenced humankind by enabling the widespread use of recording, manipulation, transmission and storage of large amounts of information. Corresponding common appliances range from mainframe and personal computers to mobile devices. Beyond that, more and more non-digital application areas are also being implemented by digital electronics due to their advantages in cost and functionality. This shift has opened new markets and generated products that are part of everyday life by now −like the compact disc, digital photography or telecommunication. Thus, such microelectronics is omnipresent in today's society and can be found in plenty of application domains covering for instance automation, computing, entertainment, communication, automotive and medical engineering as well as home appliances.

The major constituent of microelectronics is the **Integrated Circuit** (IC), which is a miniaturized electronic circuitry fabricated on a single substrate. Only half a century after the development of the first prototype, integrated circuits have exponentially increased in functionality and performance, which is recognized most evidently using the example of microprocessors and memory chips. No other industry has gained such a tremendous success, representing a multi-billion dollar business that sells more microprocessors every year than there are inhabitants on the Earth [Sem09]. However, consumer expectations have grown similarly to the boost of performance of integrated circuits. Thus, to develop ever faster computers or multi-purpose mobile devices −such as personal digital assistants and mobile phones− remains a highly challenging task for technologists and designers in the future.

1.1 Motivation and objectives

The development of the first integrated circuit in 1958 has been the inception of the trend to manufacture ever smaller electronic components. This miniaturization has led to faster and less

power-hungry devices while at the same time more components could be integrated on a single substrate at reduced costs. The empirical observation that the number of components per integrated circuit grows exponentially over time was formulated in Moore's Law in 1965 [Moo65] and has become a self-fulfilling prophecy and the driving force of the semiconductor industry. As a result, current commercial products comprise billions of microelectronic components and feature the computing power of former supercomputers. The technical foundation of such products bases on planar manufacturing technology and Complementary Metal Oxide Semiconductor (CMOS) logic. Both concepts were commercially adopted in the 1980s and have not been fundamentally changed ever since. Even though this approach has proven to be successful over the last decades, it is connected with an increasing number of technical problems [Che06]. These issues arise due to the tiny size of current technology with particular dimensions in the range of a few atomic layers. Hence, the further downsizing of microelectronic components is limited and threatens the continuous improvement of integrated circuits. Furthermore, alternative technologies are not at hand to overcome today's severe difficulties of limited performance, growing power and reliability issues as well as the dramatic boost of costs for development and manufacturing. Thus, the investigation of new design methodologies and system architectures is a primary concern of the semiconductor industry [Itr07a].

The fundamental shift towards new design paradigms is most evident by the abandonment of increasing clock frequencies in microprocessor design. That is, spatially distributed and concurrent computation is favored now over the acceleration of local and temporal computation. This abstract definition is apparent in commercial products by stagnating clock frequencies but increasing numbers of computation cores – such as Intel's Xeon, AMD's Opteron or Sun's Niagara series. The communication between the various cores within complex integrated circuits bases primarily on proprietary and application-specific communication backbones or bus-based shared media. However, both approaches are not fully scalable, which means that their system characteristics do not change proportional to the number of communication participants. For instance, the average period of use per participant decreases in a shared medium as more participants access the communication backbone. Accordingly, both existing approaches are not appropriate against the background of an increasing number of cores and their requirements to the on-chip communication.

A promising design approach is the application of a **Network-On-Chip** (NOC) to overcome or to mitigate the challenges of current and future technologies. NOCs offer a packet-oriented communication scheme and are characterized by their modular structure and their concurrency of computation as well as communication. However, to this day very few commercial products based on NOC have been released as such a communication-centric design paradigm comes along with a set of new and unsolved questions that have neither been finally answered by the industry nor by the academic community – e.g. design space exploration, reliability or system monitoring and control.

Objectives: This thesis aims at contributing to an advanced understanding and an improved implementation of networks-on-chip in nanotechnology. Its special focus is to demonstrate the

interlocking characteristics of performance, power consumption and reliability since system functionality in the presence of failures has not yet been addressed in-depth. For this purpose, the individual components of on-chip networks will have to be conceptually designed, implemented and characterized. On this basis, improvements shall be developed that demonstrate and exploit the fact that the various abstraction layers of a system are also intertwined and can rarely be regarded separately. Thus, this approach is to prove that efficient system design requires a cross-layered development ranging from the underlying hardware platform to the architecture and the system design.

1.2 Organization of this work

Following the motivation and objectives of the thesis in this chapter, the key aspects of importance are introduced in chapter 2 together with a classification of critical issues and existing design solutions. Moreover, the characteristics of Networks-On-Chip (NOCs) are discussed and analytically compared to conventional bus-based design. After that, the implementation of the fundamental components of NOCs is described in chapter 3 covering the links and routers. Thereby, reference components are derived as well as several approaches to improve system design. Based on the individual components, chapter 4 determines system behavior against the background of different architectures and algorithms. The achieved results lead to various advancements that are also presented and discussed. The subsequent chapter 5 specifies the requirements for the simulation and development of complex systems based on NOC, and

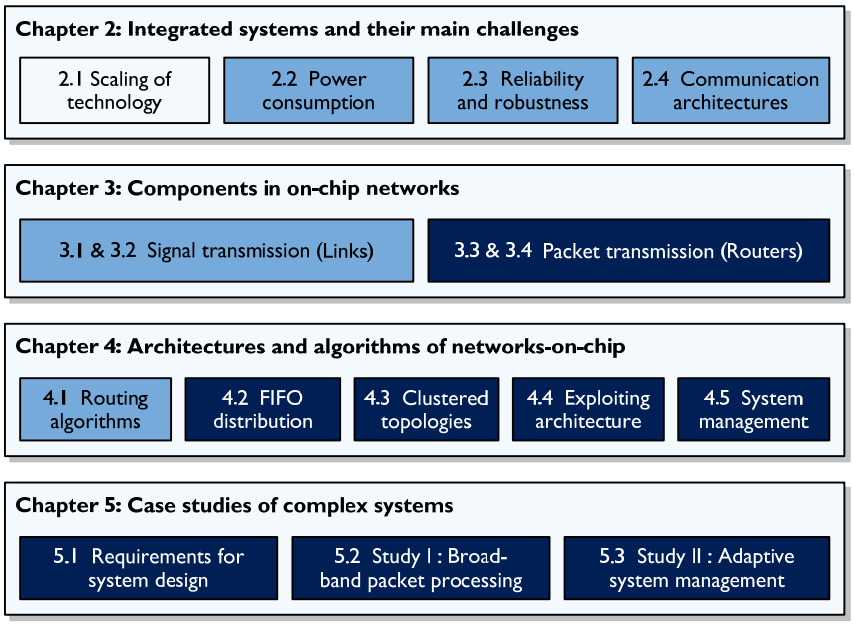

Figure 1-1 : Structure of this work with those sections highlighted that contain own contributions (Legend: ■ Contains considerable own contributions, ■ Partly contains own contributions)

presents different case studies considering the findings of the previous chapters. Lastly, chapter 6 concludes this work and gives a brief outlook on future developments. Figure 1-1 illustrates the structure of this work in further detail and highlights the sections containing own contributions.

Chapter 2

Integrated systems and their main challenges

Early integrated circuits (ICs) comprised the functionality of a specific functional module. Thus, complex systems had to be built by combining different ICs on a printed circuit board. As manufacturing technology advanced, it became possible to integrate more and more system functionality onto a single silicon substrate (also termed die). Tangible examples of such **integrated systems** are today's microprocessors that contain an ever-growing number of diverse functional modules that were previously located in individual dies −as for instance memory controllers, wired and wireless interfaces as well as graphics processors [Amd05, Lia08]. The driving forces to implement such Systems-On-Chip (SOCs) are an overall cost reduction and enhanced system characteristics. Certainly, there is a vast range of applications with diverse requirements including amongst others high-performance computing, portable low-power devices or secure processing. Nonetheless, the International Technology Roadmap for Semiconductors (ITRS) states the following main and general challenges for integrated systems [Itr07a]:

- Manufacturability: This describes the ability to produce refined chips at reasonable costs and feasible schedules. Primary aspects are the continuous scaling and the integration of new materials and devices to sustain the previous performance growth (see section 2.1).
- Power consumption: Considerations have to include all aspects of power consumption (e.g. dynamic power, leakage, power density) across the different design layers ranging from layout over architecture to system management (see section 2.2).
- Reliability: The aim of reliable design is to achieve a system that performs as desired over time and under the influence of temporary and permanent disturbances (see section 2.3).
- Design productivity: As technology progresses towards ever more complex integrated circuits, design productivity has to keep with the pace to maintain design quality and

development costs within a feasible range. Key enablers are system integration, high level of abstraction, reuse and modularity (see section 2.4).

A detailed description of the main challenges and their dependencies among one another follows in the next sections. Lastly, section 2.5 concludes the findings and determines the resulting objectives for this work. Thus, this chapter provides the basis for the subsequent investigations that refer to the contemplated challenges.

2.1 Scaling of technology

The term scaling denotes the downsizing of miscellaneous parameters in semiconductor technology. Even though other device characteristics are also affected, the prime motivations for scaling are cost reduction and performance increase. For example, when you halve the physical dimensions of a microelectronic component you can fabricate four instances within the same given area. Assuming that manufacturing costs are roughly constant, the price per component is greatly cut down to one fourth. The most important component that such scaling refers to is the transistor, which acts like a voltage-controlled switch. This is what makes the transistor so worthwhile for implementing binary digital logic.

Figure 2-1 depicts the spatial arrangement of a **transistor** in a contemporary planar bulk process with its main physical dimensions. Here, the top layer of the structure is the gate electrode, which is a good conductor. The middle layer is a thin insulating film called the gate oxide. Finally, the bottom layer is a doped silicon substrate with two diffusion areas that are contrary doped to the substrate and are named source and drain. This type of transistor operates as follows in the initial state. As the diffusion areas are contrary doped to the substrate, their intersections form reverse-biased diodes so that no current can flow. However, when a voltage is applied to the gate, the charge concentration underneath the gate oxide can be modulated due to the emerging electric field – just as in a capacitor. Thus, when the gate-source voltage V_{gs} is sufficiently large (i.e. larger than the threshold voltage V_{th}), the charge concentration underneath

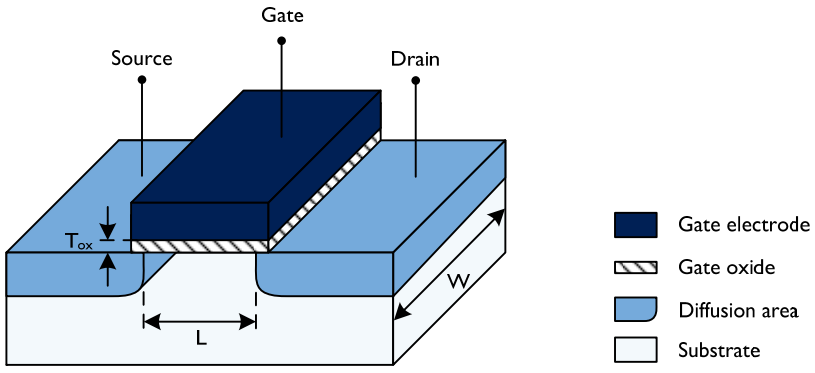

Figure 2-1 : Schematic illustration of the electrical connections and the physical composition of a MOS transistor in a planar bulk process technology

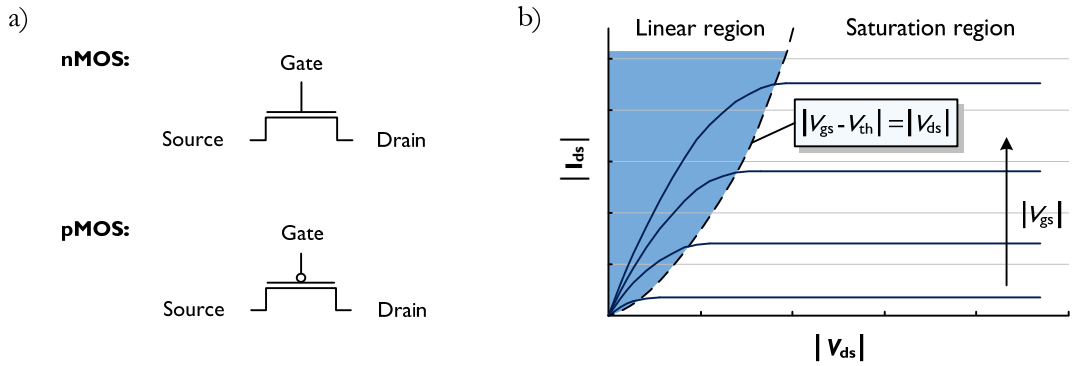

Figure 2-2 : a) Schematic symbols of MOS transistors and b) their idealized I-V characteristic based on the first-order transistor model from Shockley [Sho50]

the gate oxide is inverted and forms a channel between source and drain where an electric current can flow. The name for this class of transistors rests upon their primal materials with a metal gate electrode and their operation based on the electric field: Metal Oxide Semiconductor Field Effect Transistor (MOSFET or MOS in short). Moreover, two types of such transistors need to be distinguished with regard to the kind of applied doping. Firstly, transistors with negative charge carriers due to excess free electrons in the diffusion areas are called nMOS. Secondly, transistors with positive mobile charge carriers are called pMOS, respectively. In principle, the structure and mode of operation is the same for nMOS and pMOS but with different polarities for the associated voltages and currents. In a nutshell, the primary evident parameters for transistor operation are the gate length L, the gate width W, the thickness of the gate oxide T_{ox} (see figure 2-1), the type and strength of doping concentrations as well as the threshold voltage V_{th} and the gate-source voltage V_{gs}.

The schematic symbols of MOS transistors are illustrated in figure 2-2 together with their idealized current-voltage (I-V) characteristic. The diagram relates the drain-source current I_{ds} for different gate-source voltages V_{gs} against the drain-source voltage V_{ds}. The point of origin for the characteristic and the following analytical description is based on the simplified, first-order transistor model for long channel devices from Shockley [Sho50]. Although much more elaborate models are used today for transistor simulations [Sak90], the mentioned model is greatly meaningful and reproduces transistor functionality accurate enough to clarify its mode of operation.

The following description relates to the characteristic of the nMOS transistor, although it also applies analogously for the pMOS transistor with opposite polarities of charges and voltages [Wes05, Rab03]. In any case, three different regions need to be distinguished. Firstly, the transistor is said to be **cutoff** when the gate-source voltage V_{gs} is smaller than the threshold voltage V_{th} and no electric current flows between drain and source (see equation 1). Hence, the other two regions come into play as V_{gs} exceeds V_{th} so that a conducting channel originates between drain and source. Furthermore, the second region is called **linear region** because I_{ds} increases almost linearly with the drain-source voltage V_{ds} (see equation 2). The delineation to the third region is plotted in figure 2-2 b) as a dashed line and is given by $|V_{gs} - V_{th}| = |V_{ds}|$. This

means that for large drain-source voltages V_{ds} the conducting channel is no longer fully inverted in the vicinity of the drain. Hence, the channel between source and drain is pinched off and I_{ds} saturates at a constant level for increasing V_{ds} (see equation 3). Therefore, the third region is termed **saturation region**. Two further parameters appear in the equations that refer to the materials of the transistor: the permittivity of the gate oxide ε_{ox} and the mobility of charge carriers μ_0. Consequently, it can be concluded from the I-V characteristic and the given equations that device dimensions and voltages as well as materials affect transistor operation, and thus need to be considered when technology is scaled.

$$I_{ds} = \begin{cases} 0 & \text{for} \quad V_{gs} < V_{th} \quad (1) \\ \mu_0 \frac{\varepsilon_{ox}}{T_{ox}} \frac{W}{L} \left(V_{gs} - V_{th} - V_{ds}/2 \right) V_{ds} & \text{for} \quad V_{ds} < V_{gs} - V_{th} \quad (2) \\ \frac{\mu_0}{2} \frac{\varepsilon_{ox}}{T_{ox}} \frac{W}{L} \left(V_{gs} - V_{th} \right)^2 & \text{for} \quad V_{ds} \geq V_{gs} - V_{th} \quad (3) \end{cases}$$

2.1.1 Fundamentals of scaling and their impact on performance

The pertinent questions of scaling are to what extent the miscellaneous parameters are changed and how the continuous scaling affects the operating properties of MOS transistors. The original scaling analysis considered three different models that are based on two independent factors: S refers to the physical dimensions and U applies to the voltages [Rab03, Den74, Bac82]. The first model was **full scaling** whereas dimensions and voltages are reduced by the same factor S. Thus, the electric field across the gate oxide remains constant and the physical integrity of the transistor is ensured across different technology generations. Furthermore, such scaling leads to smaller area usage (i.e. greater device density), higher performance and reduced power consumption. However, this scenario was no viable option, as well-defined levels for component compatibility and noise-error margins needed to be maintained. Consequently, the second model in practice was **fixed-voltage scaling**. Thereby, the dimensions are scaled down but the voltages are kept constant. Therewith, this model became prohibitive with the emergence of short channel effects [Bjö81]. Against this background, fixed-voltage scaling does not give a significant performance benefit over full scaling but comes with a major power penalty and undesirable

Table 2-1 : Summary of the different scaling scenarios (with the scaling factor S being historically roughly $\sqrt{2}$ and $S > U > 1$)

	Full scaling	Fixed-voltage scaling	General scaling	Equivalent scaling
Physical dimensions	$1/S$	$1/S$	$1/S$	$< 1/S$
Voltages	$1/S$	1	$1/U$	$< 1/U$
Materials	Unchanged	Unchanged	Unchanged	Changing
Device structure	Unchanged	Unchanged	Unchanged	Changing

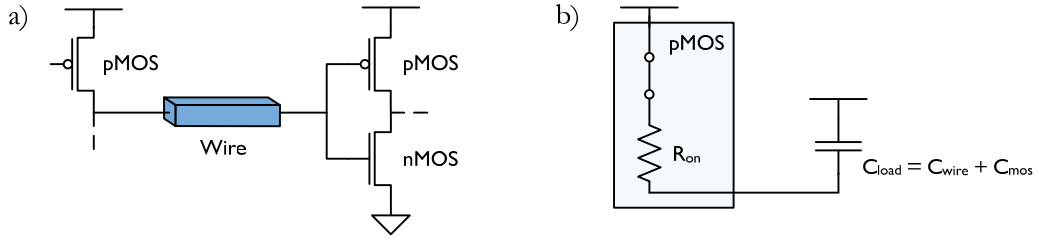

Figure 2-3 : a) Common scenario in a contemporary digital circuit and b) its equivalent RC circuit model for a delay estimate

physical phenomena. A sustainable compromise between the first two models leads to the third model named **general scaling**. Here, physical dimensions and voltages are scaled down by different independent factors whereas the voltages do not drop as fast as the dimensions. Table 2-1 summarizes the introduced scaling scenarios with their appropriate scaling factors.

The further downsizing of physical dimensions and voltages poses exceptionally difficult challenges, as current nanotechnology has reached atomic scales by now. Hence, miniaturization is slowing down which is represented in the last column of table 2-1 by smaller scaling factors. However, in order to retain the previous performance growth, new materials and modified device structures come into consideration. This scenario of **equivalent scaling** enables further performance improvements without such aggressive geometric scaling as seen in earlier technology generations. Examples of corresponding changing materials are the integration of high-$\varkappa$ dielectrics and diverse gate materials [Pig06, Doy06]. Moreover, sophisticated changes of device structures cover for instance elevated and extended diffusion areas, halo implants as well as offset spacer [Cho02, Won04].

Tightly coupled to the pursuit of smaller device area and reduced costs is the striving for increased transistor speed. Thus, it is necessary to estimate MOS performance without resorting to complex simulation models. A simple approach resulting in quite accurate delay estimates is to regard transistors as idealized switches in series with resistors and the rest of the circuit as a network of capacitors. A common scenario of a digital circuit is shown in figure 2-3 a) where a pMOS charges the attached node of a wire and a subsequent logic gate −here, an inverter composed of two transistors. The difficulty to obtain the equivalent RC circuit model in figure 2-3 b) is to average the load capacitance C_{load} and the on-state resistance R_{on} that are based on dynamic and non-linear characteristics. However, based on equation 2 and the time constant τ for the RC model, the following expression can be derived for the **delay time of a transistor** t_{MOS} to charge or discharge a capacitor:

$$t_{\text{MOS}} = k_{\text{vr}} \cdot \tau = k_{\text{vr}} \cdot R_{\text{on}} C_{\text{load}} = k_{\text{vr}} \cdot \left(\frac{\partial I_{\text{ds}}}{\partial V_{\text{ds}}} \right)^{-1} C_{\text{load}}$$

$$\Rightarrow \quad t_{\text{MOS}} \approx k_{\text{vr}} \cdot \frac{T_{\text{ox}}}{\mu_0 \, \varepsilon_{\text{ox}}} \frac{L}{W} \frac{C_{\text{load}}}{(V_{\text{dd}} - V_{\text{th}})} \qquad \text{with} \begin{cases} V_{\text{dd}} = V_{\text{gs}} \\ \mu_0 = \mu_0(T) \text{ and } V_{\text{th}} = V_{\text{th}}(T) \end{cases} \tag{4}$$

Whereas k_{vr} is a factor related to the voltage range of the delay time, V_{dd} denotes the supply voltage and the load capacitance C_{load} subsumes the capacitances due to the wire C_{wire} and all transistors C_{mos} involved (with $C_{mos} \propto W \cdot L$). It should be stressed that a large gate width W of the driving transistor, a small threshold voltage V_{th} and a high supply voltage V_{dd} result in a short delay time t_{MOS}. By contrast, enlarging the oxide thickness T_{ox}, the load capacitance C_{load} or the gate length L leads to an increased delay time t_{MOS}. Beyond that, the carrier mobility μ_0 and the threshold voltage V_{th} are subject to a complex dependence on temperature T. Based on these correlations, one can say that transistor performance generally also degrades with higher temperatures [Gut01, Hun05, Tsa00]. Furthermore, the consideration of wire capacitance C_{wire} in the delay estimate already indicates its impact on performance. However, as wires are also affected by scaling, their parasitic effects increase drastically and play a significant role for system characteristics and design decisions. This fact is reflected in the following sections and is investigated thoroughly in sections 3.1 and 3.2. Lastly, even though transistor performance in terms of the delay time t_{MOS} is also the foundation for performance estimates on other design layers, the definition of particular metrics varies considerably – ranging from gate delay to frequency and throughput. Thus, such additional metrics are introduced in the appropriate sections.

2.1.2 Physical limits and their connected issues

Atomic scales of devices in current technology limit the further downsizing both due to obvious geometrical reasons and due to intrinsic material characteristics. For example, the gate oxide thickness is in the range of a few nanometers – which is equivalent to a few atomic layers – and there are just a sparse number of dopant atoms in the transistor channel of less than a hundred. Such tiny structures are not only extremely difficult to manufacture in order to obtain homogeneous devices and operating characteristics, but lead to a large number of new and aggravated issues. Moreover, intrinsic material characteristics as the silicon bandgap or the built-in junction potential can simply not be scaled to keep the physical integrity of transistors. Lastly, to modify device voltages (e.g. supply or threshold voltage) is in addition restricted by physical factors that relate to power consumption and reliability (see sections 2.2 and 2.3). In summary, the given limits of various physical parameters are connected with a wide range of issues that impact all layers from manufacturing to system design [Bor99]. An extract of the most important and relevant ones for this work is given in the following.

Manufacturing technology needs to experience major changes in order to fabricate ever smaller and consistent devices as well as to integrate new materials and refined process steps. In particular, current lithography and its connected requirements (e.g. masks and resists) represent a key limiter due to the used deep ultraviolet light sources with a wavelength of 193 nm, which mainly affects the smallest critical dimension that can be uniformly reproduced. Even though process advances do slightly relieve the issue – such as immersion lithography or optical proximity correction – they cannot completely circumvent the need for a fundamental change in the long term [Itr07c]. Furthermore, additional and refined process steps are prospectively

required to allow following those principals of equivalent scaling (see subsection 2.1.1). This means first and foremost that innovative materials have to be integrated which requires the consideration of decisive chemical and physical side-effects. An example for the former is the unintentional diffusion of interfacing materials, and the thermal and mechanical implications of low-$\varkappa$ dielectrics are examples of the latter [Itr07e]. After all, the introduction of entirely new device structures −like multi-gate MOSFETs and insulated substrates− will also help to mitigate the miscellaneous short channel effects [Bjö81]. Hence, the combined efforts in manufacturing might alleviate the significant issues of scaling that conventional bulk CMOS is facing.

Further physical limits become apparent in circuit design as the propagation speed of charge carriers and thermal noise determine how fast and how reliable signals can be communicated across the die [Rab03]. Hence, as die size does not shrink, neither the length of global wires nor their delay time does decrease. Even local wires in dense logic impact system metrics significantly due to increasing wire resistance and capacitance (see section 3.1). Accordingly, overall system performance is about to being restricted by rather data communication than computation [Nur04]. This fundamental shift has already led to the introduction of multi-level caches and task switching. Such approaches try to mask the latencies due to slow memory access whereas communication delay plays a decisive role. However, the confined signal propagation delay also influences design decisions on system level. For example, higher operating frequencies delimit the distance that can be reached by a signal within a single clock cycle [Liu04]. Thus, fully synchronous designs will be prohibitive and globally asynchronous architectures with latency tolerant components will have to be implemented.

The list of connected issues due to the limits of scaling comprehends further domains. One domain is power consumption that includes considerations not only related to the absolute power dissipation but also secondary effects such as power density and thermal implications. Other domains are reliability and robustness because the increasing number of components as well as additional physical phenomena increases the probability of disturbances or system failure. These three challenges together with convenient approaches are thoroughly discussed in sections 2.2 and 2.3, respectively. Finally, the design process in general is also inevitably affected as all technical changes and new solutions have to be accounted for within the design flow and the corresponding design tools. This vast field comprises manifold aspects including the modeling of diversified components (like digital, RF, analog or memory), test and verification, design for manufacturing and many others. Because of the substantial complexity of these issues, they are only covered where due consideration is necessary.

Concluding, albeit the stated physical limits are diverse and impair miscellaneous aspects, they have all in common that the costs for manufacturing, design and test drastically increase with further scaling of technology. Thus, besides essential individual solutions, new design paradigms, architectures as well as combined efforts are required that also exploit the interlocking characteristics of the different design steps and abstraction layers −as for instance Design For Manufacturing (DFM) is already aimed at.

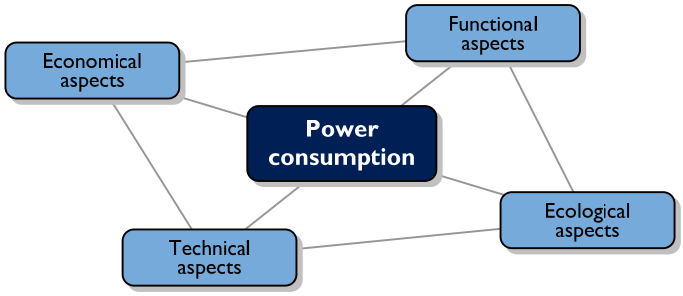

Figure 2-4 : Illustration of various intertwined aspects related to power consumption that affect developers, vendors as well as customers

2.2 Power consumption

The term electric power consumption defines the amount of electrical energy that is transferred by a consumer in a given time –and is thus another critical design parameter alongside performance. Similarly to the performance of integrated circuits, the overall power consumption has seen a dramatic increase over the last decades due to non-ideal scaling and the connected physical constraints (see subsection 2.1.2). Therefore, it is said that "the power problem is the number one issue in the long term for computing" [Pap04]. However, the fact that Information and Communication Technologies (ICT) account for more than 10 % of the entire power dissipation in Germany [Bmw08] emphasizes its relevance not only for computing. In conclusion, power consumption is one of the most challenging issues because it affects diverse aspects for developers, vendors and customers:

- Economical: Direct costs arise for power consumption but also for cooling due to the heating of electronic products.
- Technical: Appropriate materials, devices and solutions are required to avoid performance loss or degraded reliability in the case of high currents and thermal issues.
- Functional: Weight or run-time of mobile devices and the noise of fans for cooling limit market opportunities.
- Ecological: Efficient and eco-friendly usage of natural resources is obligatory.

Figure 2-4 illustrates that the introduced aspects are intertwined. Moreover, such aspects are also linked to further system characteristics besides the absolute amount of power consumption –e.g. instantaneous power or energy. Such parameters are introduced in the following subsections together with a classification of low-power approaches and a selection of viable solutions.

2.2.1 Constituents of power consumption

The instantaneous power $P(t)$ that is drawn from the power supply is defined as the product of the supply current $I(t)$ and the supply voltage V_{dd} (see equation 5). Hence, the average

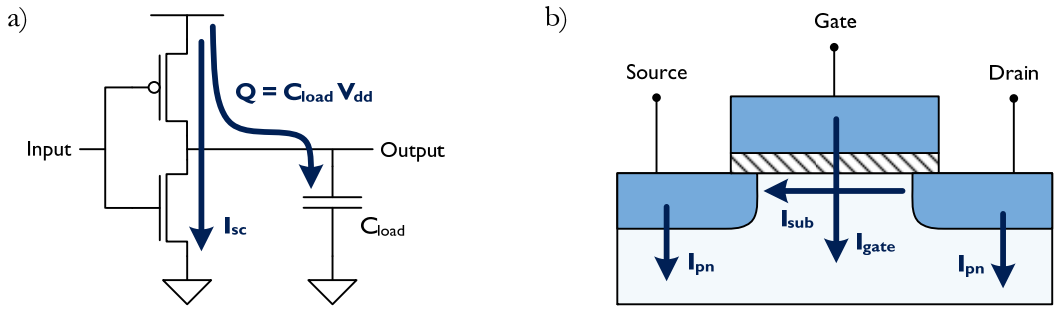

Figure 2-5 : Constituents of power consumption: a) Dynamic and short circuit power of an inverter b) Basic leakage mechanisms of a transistor (here nMOS)

dissipated power $\overline{P}_{avg}$ over a given time interval $[0, t_x]$ can be derived by integrating the instantaneous power (see equation 6). However, it is common practice to assume that the current is constant so that the instantaneous power equals the average power. Given this backdrop, the different constituents that add up to the **total power consumption** P_{tot} (see equation 7) are introduced in the following.

$$P(t) = I(t) \cdot V_{dd} \tag{5}$$

$$\overline{P}_{avg} = \frac{1}{t_x} \int_0^{t_x} I(t) \cdot V_{dd} \, dt \tag{6}$$

$$P_{tot} = P_{dyn} + P_{sc} + P_{leak} \left(+ P_{glitch} + P_{static} \right) \tag{7}$$

The first constituent, the **dynamic power** P_{dyn}, comprises the power that is dissipated when digital data is processed. Figure 2-5 a) depicts such a common scenario of an inverter implemented in static CMOS logic [Wes05, Rab03]. The total charge of $Q = C_{load} V_{dd}$ is transferred there through the pMOS from the supply rail to the load capacitance. In the second cycle, the same charge is transferred to ground, which means that no additional charge is taken from the supply rail. The frequency of such a complete event is expressed by the activity factor α in relation to the clock frequency f of the circuit. Thereupon, equation 8 can be derived for the dynamic power consumption P_{dyn}:

$$P_{dyn} = \alpha \cdot f \cdot C_{load} \cdot V_{dd}^2 \qquad \text{with} \quad \alpha \in [0, 0.5] \tag{8}$$

In an ideal logic gate the output node is connected through transistors to either the supply rail or to ground. However, as the input signal has a finite slope both transistors are conducting for a short period of time t_{sc} when the logic gate is switching. The consequence is a short circuit current I_{sc} between the power rails that does not contribute to charging or discharging the load capacitance – see the illustration in figure 2-5 a). This second constituent is called **short circuit power** P_{sc} and can be computed with equation 9. Thereby the frequency of the occurrence of short circuits is twice as high as for P_{dyn} because a conducting path can emerge every time when

the input signal changes. Furthermore, both the time t_{sc} and the short circuit current I_{sc} depend on the slopes of the input/output signals, the load capacitance and the size of the transistors.

$$P_{sc} = 2\alpha \cdot f \cdot t_{sc} \cdot I_{sc} \cdot V_{dd} \qquad \text{with} \quad \alpha \in [0, 0.5] \tag{9}$$

In contrast to both other constituents, **leakage power** P_{leak} is also dissipated when the circuit is in its idle state. Since a great number of miscellaneous technological causes contribute to the total amount of leakage power, no all-embracing equation exists to model these effects. However, the strongest impact originates from the subthreshold current I_{sub}, the gate oxide current I_{gate} and the junction current I_{pn}, which are illustrated in figure 2-5 b) and which are introduced below. The former is the current I_{sub} between drain and source of a transistor when the device should in fact be cutoff ($V_{gs} < V_{th}$). The second contributor is the gate oxide current I_{gate} that evolves from charge carriers with sufficient kinetic energy to tunnel through the gate oxide. Here, the main physical cause is direct tunneling that results in a current from the gate to both the substrate and the diffusion areas. The third essential leakage current is the junction leakage I_{pn}, which bases on diffusion currents through the reverse-biased diodes that form on the interface between substrate and diffusion areas. Common approximations for the introduced currents are given in equation 10 to 12 whereas k_{p1}, k_{p2}, ... are auxiliary parameters, and V_{pn} and I_{diode} denote the voltage and the reverse-biased saturation current across the diode [Nar05, Wes05]. Even though the causes for leakage currents are manifold, three of the most critical parameters should be pointed out, which partly result in an exponential increase of leakage currents. These are the threshold voltage V_{th}, the oxide thickness T_{ox} and the temperature T [Roy03, Rab03, Wes05].

$$I_{sub} = \mu_0 \frac{\varepsilon_{ox}}{T_{ox}} \frac{W}{L} k_{p1} T^2 \cdot e^{\frac{V_{gs} - V_{th}}{k_{p2} T}} \cdot \left[1 - e^{\frac{-V_{ds}}{k_{p3} T}} \right] \tag{10}$$

$$I_{gate} = k_{p4} W L \cdot \left(\frac{V_{gs}}{T_{ox}} \right)^2 \cdot e^{\frac{-k_{p5} T_{ox}}{V_{gs}}} \tag{11}$$

$$I_{pn} = I_{diode} \cdot \left(e^{\frac{k_{p6} V_{pn}}{T}} - 1 \right) \tag{12}$$

Two further constituents add to the total power, as shown in equation 7. The power consumption due to glitches P_{glitch} depends on small-scale timing conditions of internal signals and can hardly be modeled or predicted beforehand. Thus, empirical data is applied as an estimate. Lastly, certain circuit techniques – such as ratioed logic in address decoders – explicitly implement a resistive path to the power rails that results in power dissipation due to static currents P_{static} [Vee00]. As these two constituents can hardly be affected in the design phase or their occurrence is rather rare, they are not further taken into account.

In order to preserve power, it is important to identify the most relevant constituents and the most critical parameters that are to be tackled in the design phase. The smallest impact originates

from short circuit power P_{sc} whereas a fraction of clearly less than ten percent is generally reported for designs with appropriate slope ratios [Ped02, Wes05, Elr97]. However, this percentage will further diminish in the future because of steeper signal slopes and the correlations of supply and threshold voltages under the influence of scaling – consider that $P_{sc} = 0$ for $V_{dd} < 2 \cdot |V_{th}|$. Moreover, the dynamic power P_{dyn} contributes with about 60 % to the total power dissipation although leakage power P_{leak} has also been reported with up to half of the overall power [Yeo04, Sou09, Nar05]. Concluding, this distribution highly depends on the activity factor of the circuit as well as the application domain – for instance, compare high-performance computing and mobile devices. Yet in the future, leakage currents will further gain in importance as the ITRS also predicts an increase of leakage currents by a factor of ten per technology generation [Itr07b].

After all, the most critical parameters to cut down on power cannot always be changed in the design phase or are contradicting with performance optimizations. For example, when technology has been selected, process-relevant parameters are generally fixed – such as charge carrier mobility μ_0, the gate oxide thickness T_{ox} or the gate length L. In addition, a slower clock frequency f, a scaled supply voltage V_{dd} or an increased threshold voltage V_{th} greatly result in low-power integrated circuits. However, these parameters also have significant impact on performance (see equation 4), so that power and performance characteristics have to be traded off carefully. In the end, little load capacitance C_{load} and low temperature T both yield in better system characteristics so that the management of temperature becomes highly beneficial during system operation (see also section 5.3).

The power consumption as introduced is a commonly used design metric. It needs to be used with care though as it is not related to the delay time t_d of a given operation, and thus can be brought down by computing more slowly. Therefore, the **energy per operation** is often expressed in terms of the **Power-Delay-Product** (PDP = $P_{tot} \cdot t_d$), which offers a better comparison of different circuits as long as general requirements are similar. In addition, power is not only crucial in terms of the average amount (as $\overline{P}_{avg}$ or rather P_{tot}) or in relation to the timing (as PDP), but also with respect to the spatial occurrence. This kind of metric is called **power density** $P_{\square}$ and is defined as the power dissipation per unit area A of the die ($P_{\square} = P_{tot} / A$). It is an important parameter as greater power densities directly translate into heat, and thus exacerbate thermal impact on performance, reliability and power in itself (see equation 4, 10 and 12). Moreover, power dissipation also affects further system characteristics. Peak power, IR-drop, thermal stress and electromigration are just some of the related issues that result in design overhead to cope with them or in serious reliability concerns (see section 2.3). If expedient, such additional considerations are introduced and discussed in the appropriate sections below.

2.2.2 Classification of low-power approaches

Herein, the digest of viable approaches to preserve power is aimed at complex integrated systems. To classify such approaches, figure 2-6 depicts different criteria that can be used. Thereby, the **abstraction layer** distinguishes mechanisms based on the level where the underlying implementation is performed – ranging from technology to system level. For example,

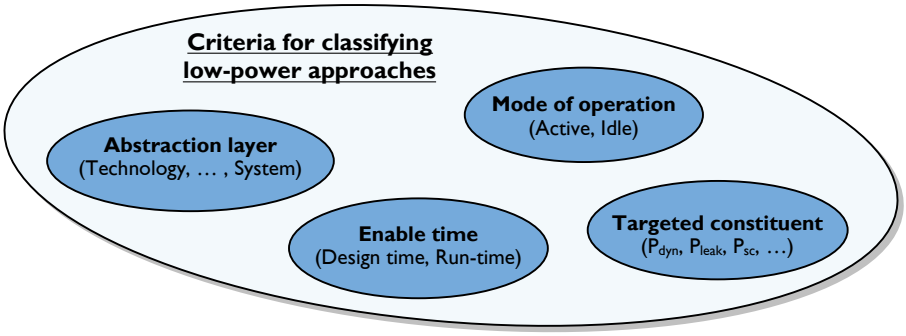

Figure 2-6 : Illustration of different criteria for the classification of viable low-power approaches in complex integrated systems

designers deal with the layout and the detailed physical effects on the technology level, whereas the interaction between complex functional modules (e.g. processors) is taken care of on the system level. Furthermore, the **mode of operation** relates to the state of the system when power savings are achieved. Accordingly, the circuit is said to be in the active mode of operation when digital data is processed. By contrast, no data is processed in the idle mode – albeit the clock network or registers might still be switching. Another criterion is the **enable time** whereas the differentiation is based on the time when the mechanism takes effect. Hence, design time approaches result in constant system parameters. On the other hand, run-time approaches dynamically influence power as well as performance parameters during the operation (e.g. by modifying the supply voltage). Lastly, as not all constituents of power dissipation can be reduced in equal measure, the **targeted constituent** (e.g. P_{dyn} or P_{leak}) is a further criterion.

Because the total power consumption P_{tot} can be dominated by different constituents (e.g. P_{dyn} or P_{leak}) depending on the specific application, it does not make sense to target at a certain constituent of power without appropriate specific knowledge given. Moreover, the criterion of the enable time might be misleading, as all approaches have at least to be implemented at design time. Therefore, a convenient classification was compiled for this thesis (see also table 2-2) that is based on the abstraction layer and the mode of operation. Besides, the stated power savings refer to the total power dissipation P_{tot} as defined in subsection 2.2.1. Finally, the last column in table 2-2 contains additional remarks that are to be considered when applying such approaches. Those diverse low-power proposals are briefly introduced in the following, starting with the technology level at the bottom of the table.

Attempts on the technology level mainly focus on short channel effects in nanotechnology, that is to say, balancing the need for large drive currents and minimized leakage currents [Liu93, Tau98, Roy03]. Both retrograde well and halo implants employ thereto substrate engineering to change the doping profile of the channel region. Thus, the distribution of the electric field and potential contours can purposefully be adapted [Won04, Nar05, Mud06, Roy03]. Another device modification is the use of offset spacers alongside the gate electrode, which can reduce gate leakage and parasitic overlap capacitance. However, transistor performance decreases due to higher resistance in the channel [Nar05].

Table 2-2 : Compiled classification of convenient low-power approaches for the application in complex integrated systems

	Abstraction layer	Mode of operation	Remark
DVS	System	Active / Idle	Decelerated adaptation
DFS	System	Active / Idle	
Body biasing	System	Active / Idle	Low-level wiring
Clock gating	System	Idle	
Parallelization	Architecture	Active	Area overhead
Voltage islands	Architecture	Active / Idle	Voltage level conversion
Input vector control	Module	Idle	
Power gating	Gate / System	Idle	Increased delay
Dual V_{dd}	Gate	Active / Idle	Voltage level conversion and low-level wiring
Dual V_{th}/T_{ox} CMOS	Transistor / Gate	Active / Idle	≥ 2 device types needed
Circuit techniques	Transistor	Active / Idle	Need for tool support
Stack forcing	Transistor	Active / Idle	
Offset spacer	Technology	Active / Idle	Increased delay
Halo implants	Technology	Active / Idle	
Retrograde well	Technology	Active / Idle	

The following layer deals with transistors as abstract models based on differential equations. For instance, stack forcing exploits the fact that leakage currents are significantly smaller in a series connection of transistors [Nar05]. Corresponding series connections can be forced by splitting a single transistor into two devices in series whereas area and performance constraints limit the general application. Moreover, a wide range of varying circuit techniques facilitates power optimizations [Cor07]. Since static CMOS logic is by far the most widespread technique though, other logic styles have a lack of automated design tools to implement complex systems [Cor06c].

Further solutions take advantage of the heterogeneous significance of specific transistors, gates or modules on system characteristics. For example, Dual V_{th} and Dual T_{ox} CMOS identify non-critical paths of a circuit. Subsequently, transistors (or gates) in these paths are exchanged for slower components that feature less leakage currents [Sul04, Sun99, Wei99]. Thereby, circuit performance remains constant because those performance-relevant paths are kept untouched (see also subsection 3.4.2). Dual V_{dd} works similarly whereas here non-critical gates are identified and powered with different supply voltages. While the aforementioned two approaches require at least two device types for nMOS and pMOS at a time, Dual V_{dd} demands voltage level conversion between different domains and an additional supply network for the second supply voltage. Besides, power dissipation of logic gates can largely be circumvented in the idle mode of operation by disconnecting the power rails. This is called power gating and requires additional transistors in series to the logic gates themselves [Ani03]. Therefore, the reduced voltage drop across the logic gate results in derogated performance (see equation 4) as well as in an additional delay to power up the virtual supply rails after an idle mode. That is why the identification of active and idle modes on system level is highly important for the power gating.

Another proposal, the input vector control, provides the inputs of the modules with specific signal patterns during the idle mode [Yua05, Tsa04]. This yields reduced power consumption because the extent of leakage currents bases on the state and the composition of transistors. In addition, different modules in a given architecture can be supplied with miscellaneous supply voltages. Hence, an architecture of such voltage islands can be adapted to the application constraints by powering non-critical modules once again with lower supply voltages. Although power is preserved in the active and in the idle mode of operation, voltage level conversion needs to be performed when crossing module borders. A further approach on the architecture level is to parallelize modules and therewith functionality. The halved clock frequency f of the modules, which is needed to process the same amount of data in parallel, is then exploited to decrease the power consumption during the active mode – e.g. by applying reduced supply voltage V_{dd} (see also equation 8). However, the power savings are accomplished at the price of an area overhead.

Lastly, the system level offers several alternatives for power management, which have in common that they rely somehow on a global and temporal system perspective. This policy facilitates to adapt system characteristics to current application needs. On the one hand, power can be scaled by turning off the clock signal of idle modules. Thereby, such clock gating prevents the unnecessary power loss in the clock network and due to registers. On the other hand, transistor characteristics can be adjusted by changing the potential of the substrate potential – called body biasing [Tsc02]. Therewith, the subthreshold voltage V_{th} is modified, which in turn significantly affects leakage currents and transistor performance. However, to provide an adjustable substrate potential wastes costly die area. Furthermore, both the clock frequency and the supply voltage can dynamically be customized for low-power operation, whereas Dynamic Frequency Scaling (DFS) and Dynamic Voltage Scaling (DVS) are common practice in commercial products – consider such techniques as Intel SpeedStep, AMD Cool 'n' Quiet or VIA PowerSaver [Int04]. Even though the adaptation of DVS is decelerated due to the time constant of the large power network, it allows a quadratic cutback of dynamic power (see equation 8).

More low-power approaches have been published in the past. However, techniques that are industry standard and that have long been integrated in common design tools are already applied for the implementations of this work, but are not enumerated here – as for example transistor sizing, state encoding or retiming [Sch06, Syn09]. Moreover, mechanisms of power reduction in such vast areas as for example analog or memory design are not in line with the objectives of this thesis and would go beyond the scope of the created classification here. The interested reader is referred to corresponding literature [Nar05, Ped02, Vee00].

2.2.3 Interim conclusion for viable low-power solutions

Of primary importance for the evaluation of expedient low-power approaches is the quantity of power that can be preserved. However, the impairment of other design parameters has to be kept at a minimum. In particular, performance is highly critical, as it does not increase as fast as

power dissipation across greater system complexities and smaller technologies – see also Pollack's law [Bor07]. According to that, the Power-Delay-Product (PDP) is a first measure for power efficiency in accordance to performance requirements as long as design and area costs are still kept in mind.

Against this background and the utilization in complex integrated systems, architecture and system level approaches appear highly promising for an implementation (see table 2-2). In doing so, the architectures of complex systems with a large number of modules can effectively be exploited due to the implicit concurrency of computation and communication modules. Beyond that, the different system level approaches can be combined within a global power management to dynamically adapt system characteristics to current requirements. For instance, idle modules can be fully shut down or at least be decelerated when they are not needed for system operation or a specific application, respectively. While DVS and power gating offer the largest savings due to the dependencies on supply voltage, they come at the price of increased delay after an idle phase. By contrast, DFS and clock gating provide less savings but are fairly simple to implement without affecting other system characteristics. The subjacent abstraction layers provide the possibility for further savings whereby their applicability is independent of system size. Here, those approaches modifying non-critical paths are particularly interesting, as they do not affect system performance at all. Among these approaches, Dual V_{th} and Dual T_{ox} CMOS are the preferential choices because they do not necessitate voltage conversion and two device types are de facto standard in current technologies. Finally, transistor and technology level solutions can also offer great improvements. However, circuit designers rely in general on a given technology and a specific library of logic gates. Hence, these solutions are not applicable for the most part.

Concluding, since the dominating constituents of power consumption and further design constraints greatly vary across specific applications and domains, there is no exclusive solution to preserve power in complex integrated systems. However, high-level approaches seem most beneficial as they can consider and exploit the changing requirements. In fact, the miscellaneous causes demand a combined approach of various solutions [Itr07b]. That is the only way to simultaneously exploit the many degrees of freedom across different abstraction layers and, likewise, to target at the diverse constituents of power consumption both during the active and the idle mode of operation.

2.3 Reliability and robustness

Besides requirements for performance and power dissipation, integrated circuits are expected to be robust against temporary disturbances as well as to operate reliably over their entire operational lifetime. As an example of robustness, a user demands from his mobile device to maintain its familiar properties across a wide range of varying ambient temperatures. Beyond that, the device is supposed to work fully reliably in the long term in order to keep malfunctions and

downtimes as rare as possible. Admittedly, to satisfy these requirements is becoming an increasingly serious concern as nanotechnology continues to scale down. For a start, individual tiny transistors comprehend structures of just a few atomic layers and materials with just a sparse number of dopant atoms. Hence, these devices are highly susceptible to material imperfections, particles and deviant physical dimensions as a result of the manufacturing process. Moreover, as the number of transistors on a single die increases, it becomes more likely that some of the billion devices depart considerably from their expected behavior or fail completely. Lastly, circuit behavior is also put at risk during operation in consequence of intermittent intrinsic and extrinsic disturbances – such as capacitive or inductive coupling. This is because noise margins decrease similarly to the related voltage levels of integrated circuits.

Therefore, reliability and robustness issues need to be considered during design and manufacturing of integrated circuits. Such issues are even of increasing crucial concern as smaller technologies further aggravate the impact on engineering and manufacturing costs, product quality as well as time to market. To be able to access these concerns in detail, the following subsection introduces at first the common terminology and analytical definitions. Subsequently, both failure causes and techniques to raise reliability are introduced and classified before an interim conclusion summarizes the findings.

2.3.1 Terminology and analytical definition

The literature on reliability engineering often applies distinctive terms to differentiate between the cause of an abnormal condition and the manifestation in a higher abstraction layer. However, the suggested terms are not consistent and the distinction of causes and their manifestation is ambiguous, because this depends on the considered abstraction layer and the logical state (compare [Kor07], [Joh89], [FedXX], [DinXX] and [IsoXX]). Therefore, in this work the notion of a **failure** is used in the widest sense and in compliance with ISO/CD 10303-226 [IsoXX]. That is, a failure denotes the component's lack of ability to operate its intended function as designed.

Based on this definition, the underlying parameter for reliability analysis is the **failure rate** λ that states how many components fail on average in a determined time interval. For clarification, assume that 16 out of 100 components fail in a given year, hence, that year's failure rate is 0.16 and an individual component fails with a probability of 16 % in that same year, respectively. Strictly speaking, the failure rate represents a conditional probability since it needs to be known how many components survived until the specific time interval of investigation. In addition, the failure rate λ is not only a function of time but also of the chosen technology and external operating parameters – that is $\lambda = \lambda(t, \varepsilon_{ox}, \mu_0, T_{ox}, T, V_{dd}, ...)$. Although an analytical definition can hardly be given due to the complex correlations, the course of the failure rate over time can in general empirically be found and is perceived as the bathtub curve [Joh89, Kor07]. Such an example of the failure rate over time is depicted in figure 2-7 with the three main phases of the overall lifetime highlighted. The first one thereof is called **early life** phase, often also referred to as infant mortality phase. This phase ideally ends when the components are shipped to the

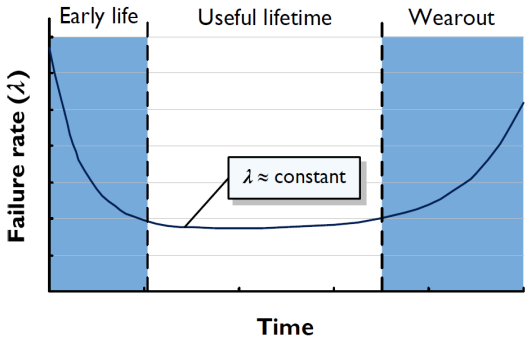

Figure 2-7 : The dependency of the failure rate on system lifetime is usually perceived as the bathtub curve that splits into three phases – whereas the failure rate is mostly considered constant during the useful lifetime

customer and is characterized by a high but decreasing failure rate. The reasons are attributable to manufacturing imperfections and borderline components, which tend to fail early during operational demands. The period of the early life phase is commonly compressed by various techniques of accelerated testing [Cro01], whereas the proportion of functional components after testing is termed yield. Subsequently, the components enter the phase of the **useful lifetime**. Here, individual random failures are the dominating cause for system outages. According to that, the failure rate is rather low and can mostly be supposed as being constant. A rising failure rate, due to aging and wearing, marks the end of this phase. Lastly, within the adjacent **wearout** phase, components suffer from the increased number of failures as they reach the end of their lifetime.

Moreover, the IEEE defines the term reliability as the "ability of a device or system to perform a required function under stated conditions for a specified period of time" [Iee94]. This implies that a component is named unreliable both as it produces a logical failure and as it does not conform to the stated conditions – for instance, an operation takes too long. Hence, the analytical definition of **reliability** $R(t)$ is to be understood as the probability of a component to operate as desired until time t [Joh89]. For example, $R(t_x) = 0.72$ states that there is a 72 % chance that the component is still running at time t_x. To express the entire course of reliability $R(t)$ across the three lifetime phases, the Weibull distribution is often used because it can simply be adapted by the parameter β [Kor07]. Thus, equation 13 represents the reliability $R(t)$ for all three lifetime phases based on different parameters for β:

$$R(t) = e^{-\lambda \cdot t^{\beta}} \quad \text{with} \quad \begin{cases} \beta < 1 & \text{for early life} \\ \beta = 1 & \text{for useful lifetime} \\ \beta > 1 & \text{for wearout} \end{cases} \tag{13}$$

Closely connected to the rather probabilistic definition of reliability is the **Mean Time To Failure** (MTTF), which is the average time a component operates until it fails. Therefore, the MTTF is equal to the expected lifetime of a component in case that the affected component

cannot be recovered. This metric can be derived from integrating the reliability over time (see equation 13). Thereby, most calculations assume a constant failure rate, which corresponds to $\beta = 1$ for the Weibull distribution and an exponential function for the reliability $R(t)$. Accordingly, the MTTF is then given by the reciprocal of the failure rate [Kor07]:

$$\text{MTTF} = \int_0^\infty R(t)\,dt = \int_0^\infty e^{-\lambda \cdot t}\,dt = \frac{1}{\lambda} \qquad \text{for} \quad \beta = 1 \tag{14}$$

Even though the assumption of a constant failure rate is common practice and adequate for most investigations, it should be stressed that other intricate considerations need to be applied for the early life and the wearout phase.

The inaugurated metrics so far reference to outages in the long term. In contrast, **robustness** is the capability to withstand temporary disturbances, which can originate from the operating environment as well as intrinsic and extrinsic noise sources. In the context of integrated circuits, varying ambient temperatures and mechanical stress are examples of environmental disturbances. Moreover, noise stands for unwanted voltage and current variations of any kind inside the integrated circuit – for instance in consequence of inconstant power supplies, capacitive coupling or electromagnetic radiation. After all, the number of disturbance sources that is considered important as well as the extent of each individual disturbance greatly varies across the miscellaneous application domains (e.g. medical, consumer or aerospace). Thus, an acknowledged quantitative measure of robustness does not exist. Instead, most investigations simply state whether an evaluated device does fulfill the application-specific robustness requirements or not.

2.3.2 Categorization of failure causes

Failure causes can be classified according to their chronological appearance and according to their severity for system operation. The latter scheme distinguishes between benign and malicious failures whereas a benign failure does not trigger additional failures at other components. By contrast, malicious failures also derogate other components by producing for instance reasonable looking but incorrect results. However, if an individual failure becomes malicious for system operation often depends on changing conditions of the appropriate applications and is thus ambiguous. Therefore, a convenient categorization of failures was compiled for this thesis that is based on the notion of time (see figure 2-8).

The primal level in figure 2-8 splits failure causes depending on their duration of occurrence. According to that, systems can at best adapt to permanent failures whereas temporary failures make it possible to recover the entire system state without any lasting damage. Moreover, the date of origin allows a further arrangement in the next level. First, **initial failures** do already exist at the time that the component is to enter its useful lifetime – marked as time t_0. Thus, such failures arise from circuit design or the manufacturing process. The listed design examples comprehend failures during the logical implementation and synthesis, in the underlying transistor and gate

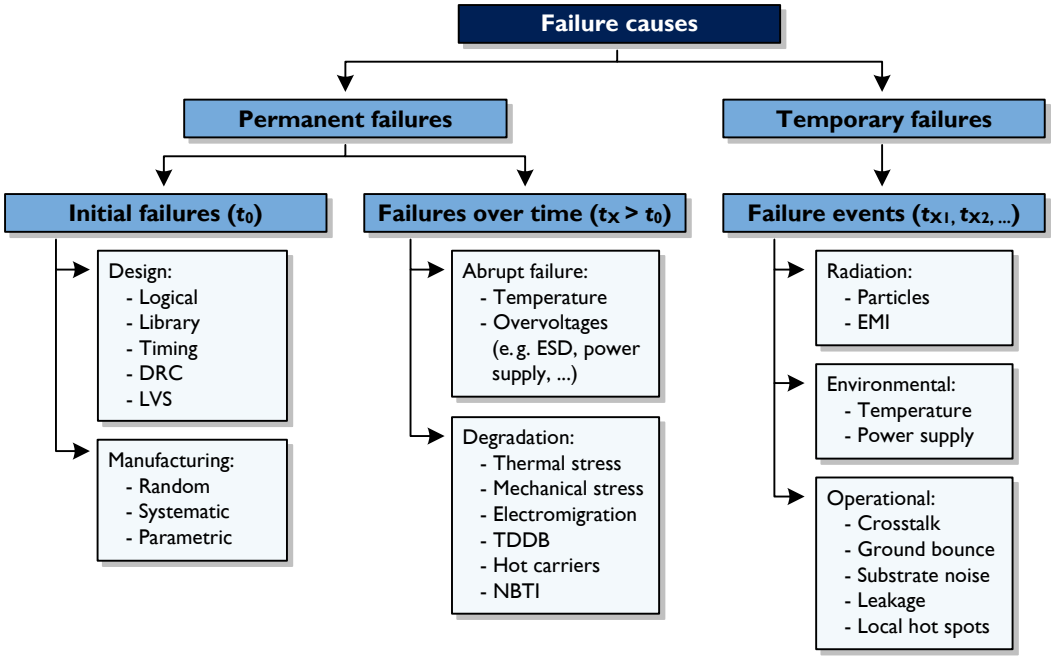

Figure 2-8 : Categorization of different failure causes with respect to their temporal duration (first level) and their date of origin (second level)

libraries as well as during the timing analysis. In addition, design steps that are commonly fully automated by design tools can be the cause for failures too −such as Design Rule Check (DRC) or Layout Versus Schematic (LVS). Concerning the manufacturing, three types of failures can be identified. On the one hand, random failures occur for instance due to undesirable particles on the wafer. On the other hand, systematic failures are the repeatable result of mask patterns, process steps or applied chemical substances. Lastly, manufacturing related parameter variations can cause parametric failures, which describe the fact that integrated circuits work logically correct but do not conform to given design constraints (e.g. on performance or power dissipation) [Shi03, Bor05].

Another group of causes is the permanent **failures over time**. These can be brought forth at any point in time $t_x > t_0$ due to abrupt incidences or due to steady degradation based on wearing and aging. The abrupt occurrence is mostly the result of mishandling −by way of example, exceeding temperatures and voltages, such as Electro-Static Discharge (ESD) [Wes05]. Unlike these fast effects that take place in less than a second, degradation rather happens in the range of years. The most familiar causes are thermal and mechanical stress whereas other effects require a deeper analysis of physical and chemical coherencies. Electromigration for one is the unwanted material transport in conductors because of the gradual movement of ions, which finally leads to open circuits (voids) and shorts (hillocks) [Hu95, Wu02]. Further causes relate to the impairment of transistor characteristics as a result of derogations in the gate oxide. For instance, tunneling currents from the gate inflict irreversible damage on the gate oxide −called Time-Dependent Dielectric Breakdown (TDDB) [Moa90]. Similar damage of the dielectric is also brought on by

high-energy charge carriers (termed hot carriers) that tunnel from the transistor channel into the gate oxide [Hu92]. Lastly, Negative-Bias Temperature Instability (NBTI) designates unintentional charge traps at the interface between gate oxide and channel in pMOS devices due to dangling bonds [Sch03, Pau05]. Even though the root causes are different, the three last named failure effects all result in a critical drift of the threshold voltage over time.

A simple model that is often applied in the first instance to illustrate the coherencies of the diverse permanent failures over time –in particular due to degradation– is the Arrhenius model [Jed02, Cro01, Wan08]:

$$\mathrm{MTTF} \propto k_{\mathrm{fm}} \cdot \mathrm{e}^{\frac{E_{\mathrm{a}}}{k_{\mathrm{Boltz}} \cdot T}} \tag{15}$$

Here, k_{fm} is an empirical constant, k_{Boltz} stands for Boltzmann's constant (8.62·E-5 eV/K), T denotes the absolute temperature and E_{a} is the activation energy of the failure mechanism. Hence, it can be concluded from equation 15 that wearout and aging failures increase exponentially with temperature and system reliability decreases, respectively [Hua04, Sri03, Sri04]. Thus, temperature represents a main challenge in current and future nanotechnology [Itr07a].

Last but not least, **failure events** come about at different times (t_{x1}, t_{x2}, ...) during operation and last just temporary. Hence, such events only involve a change of data, and no harm of the physical circuit itself. Those effects of radiation are generally also referred to as soft-errors –for instance, in consequence of particle strikes (like α-particles or neutrons) and Electro-Magnetic Interference (EMI). Such mechanisms induce charge variations on internal nodes and thus possible failures [Mit05, Haz03, Muk05]. Furthermore, environmental impacts can originate from the ambient temperature and the power supply [Teh10]. Indeed, it should be emphasized that if deviations of these impacts exceed a critical threshold, they could also lead to permanent failures as aforementioned. Finally, the intrinsic operation itself can be the trigger for a great number of diverse failure causes and consequences. These influences range from signal transmission, power supply network and substrate potentials to state preservation on critical nodes and parametric discrepancies –appropriate examples are stated in figure 2-8. Most of these different failure events are reproducible as they rely on certain logical states or transitions. In the end, figure 2-8 can only list an extract of all operational issues due to their large number. The interested reader is referred to appropriate literature for a comprehensive and detailed presentation [Hey03, Hei02, Sri98, Wes05, Rab03, Cat67].

There is no common consensus on what particular failure cause is deemed to be the most critical, as this depends on the application domain and often also on a personal perspective. However, intensified research efforts can be identified in three different fields. Firstly, parameter variations have traditionally rather been considered an issue of manufacturing. As variations of device characteristics –and thus also of system characteristics– have drastically increased though, such parameter variations have to be accounted for during the design stages and system operation as well. Secondly, with the rising appearance of soft-error related outages not long since, models, test equipment and countermeasures have increasingly been published [Shi02, Mit05]. Finally,

permanent lifetime issues have repeatedly been contemplated as a serious concern for future integrated circuits [Itr07c, Sem03]. Thereby, the diverse mechanisms of gate oxide breakdown, electromigration as well as thermal and mechanical stress are mostly stated as those particularly critical issues.

However, all failure causes will gain in importance in the future due to non-ideal scaling of technology, increasing transistor counts and power densities as well as mechanisms of adaptive processing [Sri04]. Thus, appropriate techniques to raise reliability against the background of miscellaneous failure causes have to be closely investigated and characterized.

2.3.3 Classification of techniques to raise reliability

Reliability engineering in general specifies three main approaches that also comprehend the miscellaneous techniques to raise reliability in integrated circuits:

- Physics of failure: Denotes the identification and understanding of the precise physical failure mechanisms in order to avoid such failures in advance.
- Redundancy: Comprises the duplication of components, data and operations, which complies with an additional spatial, informational and temporal overhead.
- Derating: Describes the purposeful operation of a component at less than its given maximum ratings – for instance at a lower operating frequency.

Even though this differentiation is beneficial, it does only insufficiently reflect the detailed correlations of reliability techniques in complex integrated systems. Instead, a novel classification was compiled for this thesis (see figure 2-9) that distinguishes at first structural and behavioral approaches, whereas the former relates to the hardware platform with computation and communication components and the latter relates to the operating system and software applications. Moreover, the kind of failure handling permits a further characterization based on those techniques that avoid or remove failures as well as those that tolerate failures by different means. Lastly, common examples are also listed for each of the various groups, though these examples can only represent an extract. Detailed and extensive descriptions thereof can be found in the basic literature [Cro01, Rab03, Wes05].

The first group in figure 2-9 is **failure avoidance** whereas corresponding techniques try to avoid failures completely in advance. Hence, such techniques concern the selection or adaptation of an applicable technology and manufacturing process as well as the various design steps [Wu02, Chi95, Sas06, Oma07]. As such examples, wire widening aims at avoiding the impact of electromigration, and a stable power net can avoid logic and timing failures by providing consistent power and ground connections. Unlike the try to circumvent failures, mechanisms of **failure removal** aim at removing the failure appearances permanently after they have come into existence. Accordingly, these mechanisms are mostly employed after manufacturing and before the use in the field. Such a common practice is static reconfiguration in nowadays memory production, whereas fuses disconnect defective bit slices, respectively spare bit slices are

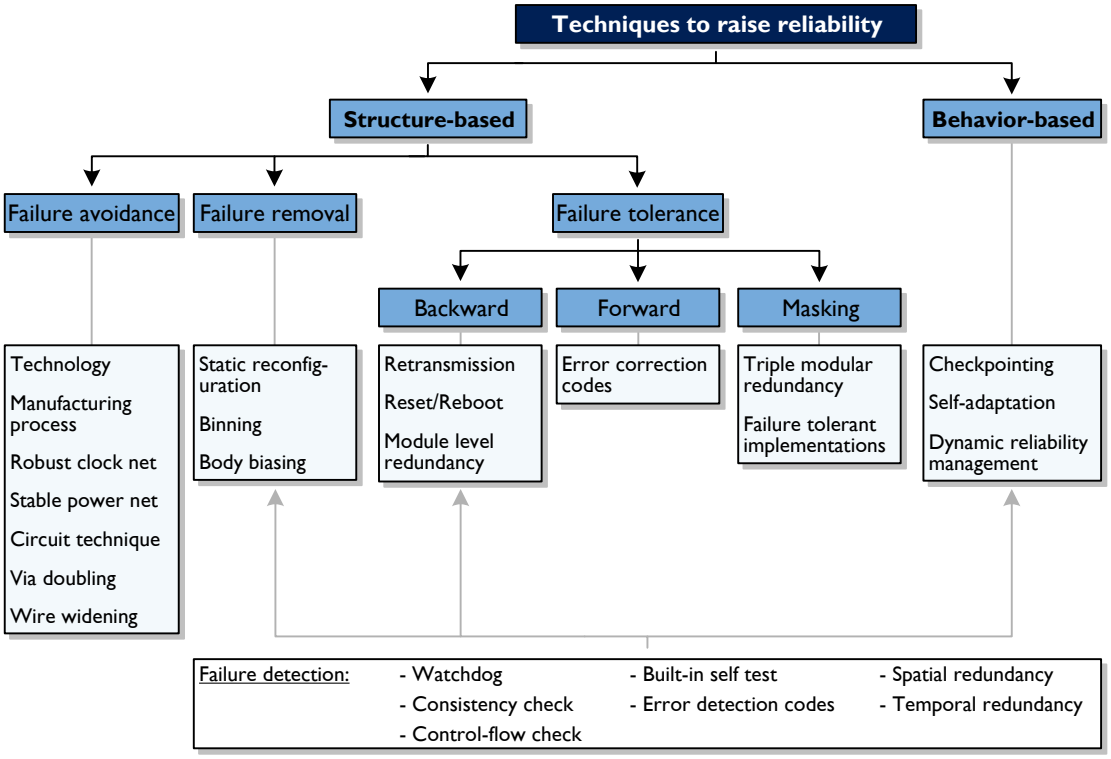

Figure 2-9 : Compiled classification of miscellaneous techniques to raise reliability in integrated systems, primarily related to structure-based types of failure handling

connected by antifuses [Moo86]. Furthermore, manufacturing tests formerly rejected chips out of specification. However, binning sorts these chips according to their measured system characteristics and enables their operation under reduced maximum ratings. Lastly, sundry methods enable to sense process perturbations [Dat06]. With such sensing at hand, body biasing is the most widespread technique that can shift design characteristics into the required ranges [Tsc02].

Another group of techniques to raise reliability is **failure tolerance** whereas failures here are dealt with by different means. The first set of such techniques recovers a backward state in order to do the same operation again. Thereby, the extent of necessary recovery can range from a communication state (retransmission) to an entire system reset. Following, the failing operation is executed again, either on the very same physical component or on another redundant module. By contrast, forward failure tolerance permits to detect and correct failures on the spot. This is achieved by informational redundancy, that is, auxiliary error correction codes are added to the actual amount of data. Thus, these codes allow correcting failures within certain bounds [Spi04]. The third set of failure tolerance uses masking to prevent failures from propagating to other modules and to higher abstraction layers. For instance, triple modular redundancy performs a function three times in parallel. Subsequently, a voting system masks a possible failure of a single unit by forwarding the result of at least two consistent units [Mit01]. Besides, certain implementations can tolerate or even harness failures without sacrificing quality on the application

level [Akg06]. Such convenient candidates exist particularly in the domains of digital signal processing and multimedia as well as in the field of probabilistic algorithms [Geo06, Din02].

Finally, behavior-based approaches can in principal similarly be arranged as those structure-based ones, and, in the strict sense, some introduced techniques can actually be implemented in both hardware and software. However, with regard to the objectives of this thesis, solely three exemplary techniques are stated for clarification. Checkpointing permits tolerating failures by recovering a backward state of the application, which in turn is stored at specific times (called checkpoints [Lin01]). Moreover, failure appearances can be removed when the concerned defective component has been identified. Such self-adaptation [Kar99] utilizes thereby amongst others relocation of tasks and communication streams as well as adjustment of circuit parameters −like body biasing, DVS or DFS (see also subsection 2.2.2). At last, an example of failure avoidance is Dynamic Reliability Management (DRM), which targets preventing or postponing failures by means of microarchitectural awareness of activity and lifetime reliability [Sri03].

It should be pointed out that several of the introduced approaches necessitate additional means of failure detection before the specific techniques can be employed. Therefore, figure 2-9 quotes different examples to take this need into consideration. The listed examples here depict just a digest and range from software-based watchdogs to gate-level redundancy [Mah88, Mit05].

2.3.4 Interim conclusion for reliability approaches

A distinct and comprehensive measure to access the overall reliability of a complex integrated system does not exist by now. This is because application-specific requirements greatly vary and adequate technology-dependent figures as well as fault-models are rarely published. In addition, the different techniques to raise reliability mostly aim at a particular failure cause so that such specific enhancements are no criterion for the system as a whole. Hence, a common measure will prospectively be required as reliability and robustness issues dramatically gain in importance and denote crucial quality metrics. However, for a comparison of different implementations the impact on performance and power consumption will have to be accounted for as well −possibly similar to the PDP that relates power to performance (see subsection 2.2.1). Corresponding considerations are for instance particularly significant for those approaches of redundancy and derating, as they inherently impair performance.

Since the diverse failure causes necessitate different countermeasures, no specific technique to raise reliability can be identified as the most important. Instead, combined approaches are required that target initial failures, failures over time and failure events (see also figure 2-8). Thereby, initial failures are effectively counteracted by techniques of failure avoidance and removal, which are widely found in low-level design stages and manufacturing technology. However, such considerations will also increasingly have to find their way into high-level design to be able to exploit implicit advantages of complex integrated systems −as for example concurrency and redundancy. Albeit all such techniques are beneficial, they do only conditionally tackle lifetime issues −like failures over time and failure events− whereas soft-errors contribute

to the largest fraction of these failures [Mah04, Iye82, Bal69]. Hence, a great number of viable techniques already exist to detect soft-errors and to correct their implications [Mit05]. As soft-errors cannot be prevented, those techniques are found in the group of failure tolerance.

In contrast to failure events that occur temporary and do generally not harm the physical circuit itself, permanent failures over time do not allow a full system recovery and have to be dealt with in the long term. Thus, future complex systems require some means of graceful degradation in order to circumvent a complete system outage due to a single failure. It is striking that appropriate systems −which are capable of performing continuously in the presence of permanent failures− increase manufacturing yield as well as system reliability during the useful lifetime and the wearout phase. Despite these promising chances and the significance for complex systems, only few efforts have been made in this field. Thereby, several convenient techniques to raise reliability are especially suited for exploiting the implicit characteristics of complex integrated systems. For example, the great number of modules allows utilizing potential redundancy of computation and communication modules. Moreover, the modularity enables to contain failures and to adapt individual modules according to determined reliability targets −e.g. by the use of DFS or power gating (see also subsection 2.2.2). Lastly, the operating system can possibly monitor and control on-chip temperatures and distribute computation and communication loads correspondingly.

Concluding, prospective systems will have to acknowledge failures as being inevitable, and will have to provide techniques to deal with these while still operating to an adequate extent. Thereby, costs of verification, manufacturing and test can drastically be reduced when the demands for 100 % correctness are relaxed [Itr07a]. Since no individual technique covers all aspects of reliability, a combined approach of techniques across all design and manufacturing stages is mandatory. However, complex integrated systems offer new chances to effectively tackle permanent failures over time and across the different abstraction layers. It is of utmost priority though that promising solutions can be inserted in an automated fashion as well.

2.4 Architectures for system communication

The development of technology with ever smaller device dimensions has allowed integrating increasingly complex circuits on a single die (see section 2.1). Starting with basic functions, chip designers have made use of the available integration density to develop algorithms and entire Systems-On-Chip (SOC). Thereby, system performance could continuously be increased by two means. On the one hand, by smaller and faster transistors, and on the other hand by improved complex architectures −as for instance, advanced instruction sets, on-chip caches or intense pipelining. This temporal trend of boosting chip complexity is illustrated in figure 2-10 on a logarithmic scale over time [Itr07b]. Unfortunately, the number of gates that can reasonably be designed does not develop as fast as the producible chip complexity (note the design productivity in figure 2-10). The originating design productivity gap translates into diminishing market

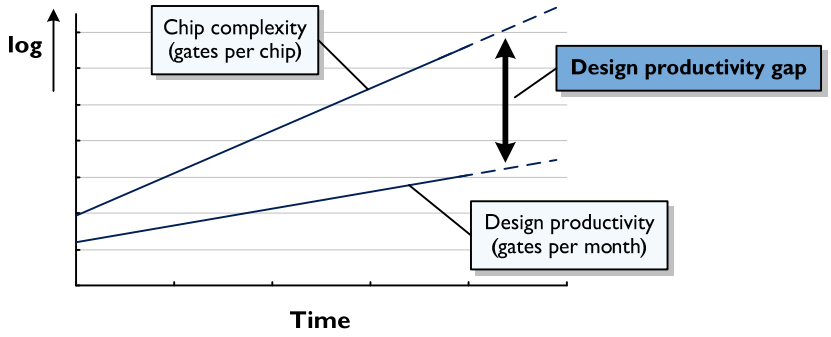

Figure 2-10 : Temporal trend of the growth rates for chip complexity and design productivity resulting in the widening design productivity gap

opportunities due to associated cost, time to market and quality issues [Itr07b, Ras02, Bor07]. In order to avoid that this gap becomes a key limiter for the microelectronics industry, design productivity must scale by more than a factor of two per technology node [All02].

Chip design can contribute to this requirement by the substantial employment of **modularity**. That is to say, the entire chip complexity is divided into smaller functional modules that can be designed independently and more efficiently [Liu04]. This urge for smaller and enclosed modules is further motivated by several technological concerns. For instance, the number of metal layers limits the physical wiring of large modules (see Rent's rule [Lan71, Chr00]) and the dominating wire delays make large synchronous modules prohibitive (see also subsection 3.1.1). Striking examples of such increasingly modular Systems-On-Chip (SOCs) are the microprocessors, which are developing from single-cores to multi- and many-core systems −for instance, note Sun's Niagara series, Rapport's KiloCore and Intel's Tera-scale computing research [Hel06, Hos07]. In fact, there is a wide range of diverse modules −called resources or cores in this context− which can be classified into five fields: general purpose processors, application processors, acceleration logic, memory and input/output interfaces (I/O). Against the background of such modularity, the information flow between the various modules necessitates appropriate architectures for system communication, whereas the architecture is to be understood as both the structural topology and the physical placement.

Even though the mentioned aspects result in a gradual change of design paradigms, they represent a fundamental shift of many design aspects. These very different aspects are commonly exemplified by the movement from local and temporal computation to spatially distributed and concurrent computation. Or in other words, by the change from a computation-centric to a **communication-centric** design perspective [Jan03b, Nur04].

Similarly to the gradual but considerable shift of design paradigms, the most important system characteristics do also change or novel ones come to the fore. Certainly, the introduced parameters of performance, power consumption and system reliability are further on key characteristics. However, performance in communication-centric architectures is expressed by

additional means. Of primary interest there is the amount of data that can be transmitted in a given time. This parameter is called **data rate** and is generally quantified in terms of bits per second (bit/s) or bits per clock cycle. Accordingly, the aggregated data rate denotes the summation of data rates that are attained by all destinations in a specific architecture. Since the conveyed data may contain control information or errors, not the entire amount of data is useful. Therefore, the term **throughput** only considers the useful amount of data that is transmitted in a unit of time, and it can thus never be larger than the corresponding data rate. These two parameters only quote how much information is communicated but not how long it takes. Hence, **latency** finally characterizes the time that elapses from the start of a communication to the reception at the destination. It is generally expressed by means of seconds or clock cycles.

Beyond that, further desired system characteristics gain in importance, though they cannot be evaluated quantitatively. As mentioned afore, modularity is such a key characteristic, which also entails a few more. Here, modularity denotes the encapsulation of functionality –which includes computation as well as communication– with determined interfaces. This enables the arbitrary configuration of heterogeneous modules within a system architecture. Moreover, the independent modules offer a high degree of concurrency, which can be exploited to gain great performance. Lastly, when modules are entirely encapsulated from their surroundings, they can be reused in miscellaneous applications or even across different technologies. Design costs as well as productivity considerably benefit therefrom, which in turn alleviates the design productivity gap. With regard to the continuously growing chip complexity as mentioned at the outset, it is in the end of utmost priority that any communication architecture is scalable. This means that system characteristics rather ameliorate with an increasing number of modules than deteriorate.

Essential market drivers for complex communication architectures with the described characteristics can be found amongst others in the domains of office/home computers, networking and mobile devices [Itr07f]. However, corresponding applications are bound to very different needs –such as computation, memory, I/O or power dissipation. Nonetheless, two main underlying scenarios can be identified [Ben06]. Firstly, homogeneous systems are characterized by many identical resources and different applications operating concurrently (e.g. multi-processor systems). Secondly, heterogeneous systems comprise very different resources whereas mostly just a single or few applications are executed at the same time (e.g. mobile devices). Independent of the various requirements, the following subsections present possible candidates for complex communication architectures, whereas a first qualitative summary is given in table 2-3 at the end of subsection 2.4.3. Finally, an analytical comparison is developed in subsection 2.4.4 that evaluates and contrasts the system characteristics of architectures based on busses and networks-on-chip.

2.4.1 Point-to-point

The primary and most simple manner to join multiple resources of an integrated circuit is the point-to-point topology, which is presented in figure 2-11. Here, diverse and dedicated links

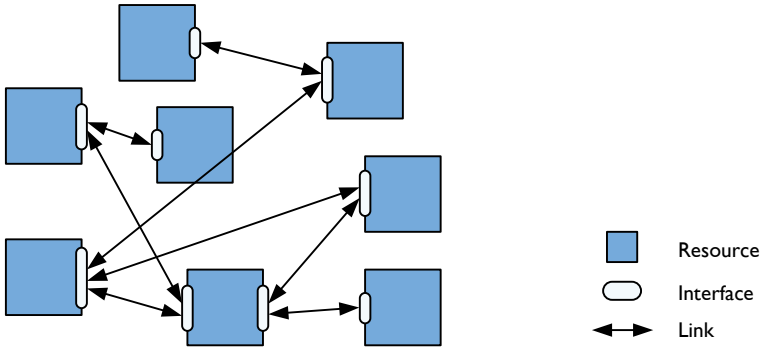

Figure 2-11 : Application-specific topology with dedicated point-to-point links that only connect those resources that require some means of data exchange

connect only those resources that necessitate some means of data exchange. The links themselves are ordinary wires in the simplest case or offer some additional mechanisms to support signal transmission (e.g. repeater insertion or pipelining). It should be noted that resources can generally be of arbitrary size and shape, and that they can contain very different functionality. Therefore, the interfaces match the specific needs of each communication pair belonging together.

The presented type of a point-to-point topology accounts for the following system characteristics −briefly summarized and contrasted with other topologies in table 2-3 below. To begin with, the employed network elements are fairly simple to design. Since all data streams operate on independent dedicated links, no additional mechanisms of arbitration or congestion control are necessary. Moreover, the interfaces do only have to consider the constraints of the concerned communication pairs. For the same reasons, the performance in terms of data rate and latency is good as well as deterministic, as the links can be specifically exhausted to their technological limits.

However, the simple and well-performing point-to-point topology comes at the price of limited scalability and unsuitable reusability. More precisely, performance yet still scales to a certain extent because each additional link adds to the aggregated data rate of the system. This growth is bounded though due to the confined number of available metal layers for the physical layout of the links. Beyond that, the diverse aspects of design costs do not scale at all. By way of example, with each further resource not only one additional link has to be added, but several new and heterogeneous links as well as interfaces. Hence, the costs of area usage, power consumption and engineering increase exorbitantly [Goo02], which means that the connectivity of the resources and the costs have to be traded off against each other. Lastly, the network and all its elements can rarely be reused. This implies the links and the interfaces that are specifically dedicated to the attached resources. But even the resources themselves are affected by the links that cross over them. Thereby, the timing of both links and resources becomes a global issue and the strict modularity is lost [Bry01].

Concluding, point-to-point topologies are suited when the number of resources is small or when communication requirements are low and known at design time. In these cases, the

dedicated links can fully exploit the capabilities of the involved resources and the underlying technology. However, the limitations due to scalability and reuse restrain the reasonable application in complex systems.

2.4.2 Bus-based

The most widespread communication architectures rest upon bus-based topologies [Ald99, Cor99, Win99]. Such a basic example is presented in figure 2-12 a) with diverse resources as in case of the point-to-point topology. However, instead of dedicated links, a common bus interconnects all resources and is shared for the data exchange among each other. Thus, the interfaces are not either specifically adapted to each communication pair, but provide a consistent view of the shared bus and the used transmission protocol. Since the bus is shared, only one sender can drive the bus at a given time, while one, several or all resources can receive the message in parallel. So, the initiator of a message needs to request bus access from a central arbitration module first – not drawn in figure 2-12 to simplify matters. Once bus access is granted the initiator can begin with data transmission. On completion, the bus is released and the next initiator gains bus access. Therefore, the period of time for the arbitration decision and the policy itself significantly impact bus performance.

Even though the network elements are similar across the entire bus topology, they are more complex compared to the point-to-point topology because they need to comply with the global transmission protocol. Additionally, the central bus arbiter represents an essential contribution to the complexity due to its impact on performance. From the shared nature, it also results that the aggregated data rate is limited and divided between all communication participants. Besides, the latency is small once bus access is granted. In case the bus is heavily loaded though, the overall latency becomes high because it is rather dominated by the arbitration time than by the transmission itself.

To evaluate the scalability necessitates a differentiated examination. For a start, performance does not scale for various reasons. On the one hand, more resources mean that the bus access time per participant decreases accordingly. On the other hand, arbiter delay as well as parasitic capacitances grows with every additional resource. As a result, bus timing gets difficult and costly, making the bus the performance bottleneck of the system. Contrary to performance, design costs do scale in principle because each further resource leads to a sole extra interface and a slight increase for the design of the arbiter. However, albeit the bus capacitance scales appropriately with the rising number of resources, it can become prohibitive for power dissipation. Finally, bus-based topologies are suited for the reuse of its elements due to the simple common concept. Merely the central arbiter is specific for a given number of resources and has generally to be customized. The discussed characteristics are briefly summarized in table 2-3 below.

Prevalent bus protocols and implemented topologies today differ significantly from the fundamental introduced functionality – consider IBM's CoreConnect, ARM's AMBA or Silicore's

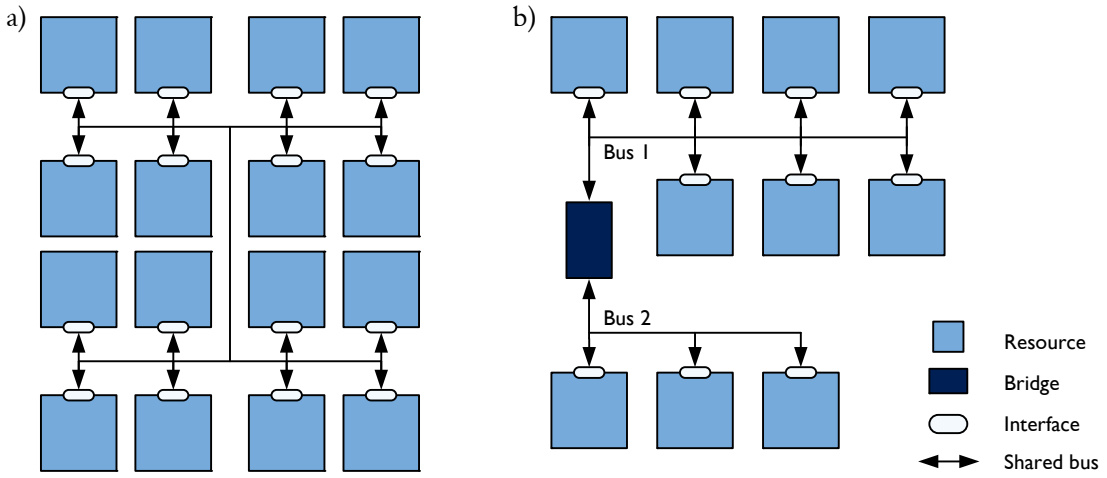

Figure 2-12 : Exemplary topologies of a) a single shared bus and b) a segmented bus system

Wishbone [Sal02, Uss01, Jer05]. Crucial for the evaluation there is that several initiators can transmit data concurrently by means of pipelining or multi-layered and segmented topologies. An example of a segmented bus is given in figure 2-12 b), whereas a bridge interconnects the two bus segments. Thereby, bus 1 and bus 2 can operate alike, or apply different speeds or even diverse protocols. Such advanced topologies are not bus-based anymore in the strict sense, and the delineation to networks-on-chip is blurring (see next subsection). Hence, the mention of bus-based topologies in this work relates to the basic bus as introduced above −and as it is illustrated in figure 2-12 a).

Concluding, the simple and common approach of shared busses makes corresponding topologies a good choice for cost efficient communication architectures. However, the performance is not scalable and suffers from every additional resource that is connected to the bus. For this reason, busses are only conditionally applicable for large numbers of resources and changing communication requirements.

2.4.3 Networks-On-Chip

Networks-on-Chip (NOCs) are a promising option for communication architectures to overcome the challenges of nanotechnology as well as of conventional point-to-point and bus-based topologies [Ben06, Dal01]. Thereby, NOCs are based on the fundamental mechanisms of distributed networks (e.g. the Internet), in particular packet-oriented communication. However, since the cost functions for wiring and memory behave diametrically in distributed and on-chip networks, established approaches cannot simply be taken over.

Two common examples of topologies for networks-on-chip are illustrated in figure 2-13 a) and b). The former, the mesh-based topology, is the most widespread type due to its regularity [Sal07a, Sal08], and the latter, the star topology, has shown to be convenient in area-

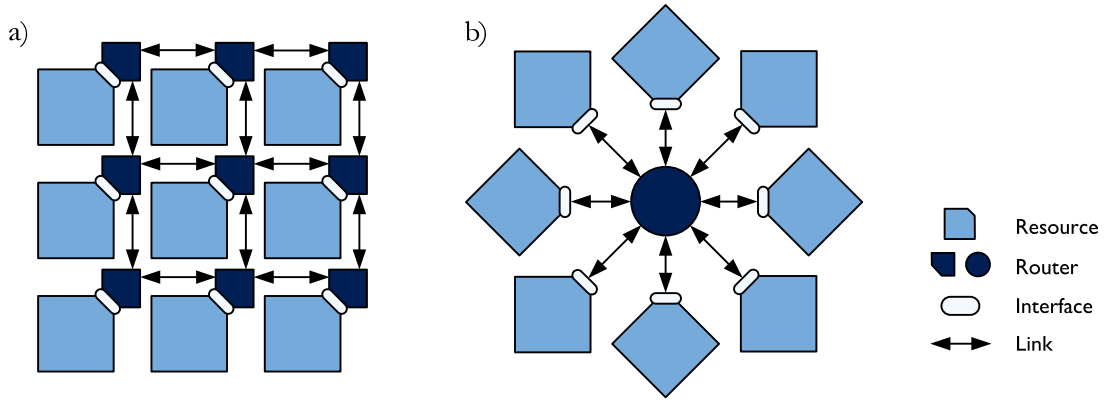

Figure 2-13 : Two examples for common topologies of networks-on-chip: a) Mesh-based topology of a 3x3 network b) Star topology

constrained implementations [Lee04, Lee05]. In any case, similar to the bus-based topology, the interfaces there provide consistent gateways to the miscellaneous, heterogeneous resources. However, instead of a bus that directly connects the communication participants altogether, messages have to make use of several links and intermediate routers to reach their destinations. Hence, the communication network of links and routers is entirely encapsulated from the resources and provides multiple concurrent data paths. This comes at a price though, since routing and arbitration decisions as well as congestion control have to be carried out in each router.

Both the distributed nature of communication control and the competing concurrent data streams result in complex network elements. The concurrency also has to be considered when evaluating the performance of the NOCs. On the one hand, the data rate is high as each link adds to the aggregated data rate of the system. On the other hand, data rate as well as latency is also highly affected by the type of network traffic. Particularly, latency is a function of the network load and the distance from the sender to the destination. Furthermore, networks-on-chip are highly scalable as it concerns performance and cost metrics. First and foremost, each additional resource implicates at least a further link, respectively a router, which translates into an increased data rate. And as long as communication distances are kept small, latency does not suffer therefrom. With respect to the design costs, not only the resources but also the routers and links are highly modular. Thus, to join more resources is straight forward and with no considerable implications for the existing architecture. Moreover, the strict modularity also allows reusing all network elements in very different applications, or to specifically adapt different links and routers independently – for instance, by means of pipelining [Bje06]. The aforementioned characteristics can be revisited in the brief summary of table 2-3.

Concluding, the advantageous aspects of scalability and reusability favor networks-on-chip for the usage in complex communication architectures. However, even though the performance also benefits from such topologies, the complexity by reason of the network elements can be significant. Hence, this overhead of networks-on-chip can make their implementation prohibitive, especially if communication requirements are low.

Table 2-3 : Brief summary of the discussed system characteristics for the different types of introduced communication architectures

	Point-to-point	Bus-based	Network-On-Chip
Network elements	Simple	Medium	Complex
Data rate	High and guaranteed	Limited because shared	High
Latency	Low and deterministic	Small after bus access is granted	Varying (function of distance and congestion)
Scalability	Limited	Limited	Fully scalable
of performance	✓	×	✓
of design costs	×	✓	✓
Network reuse	Not suited	Partly suited	Suited

The presented system characteristics of all three communication architectures are summarized in table 2-3 and allow a first assessment. In a nutshell, point-to-point topologies are solely applicable for very small architectures, which benefit from the simplicity of the network elements and the good performance. Bus-based topologies are suited for cost-efficient implementations with rather low communication requirements. However, both topologies do not qualify for the use in complex communication architectures due to the constraints of scalability and reusability. Lastly, networks-on-chip are well suited for large network sizes and sophisticated communication demands since they are fully scalable. For this reason, such topologies can help to overcome or at least mitigate the design productivity gap [Jan03a].

2.4.4 Analytical comparison: Bus vs. NOC

Essential system characteristics of bus-based topologies and Networks-On-Chip (NOCs) are analytically derived and compared in the following. Thereby, the underlying architectures rest upon those structures as illustrated in figure 2-12 a) and figure 2-13 a). Point-to-point topologies are excluded from this comparison though, as they represent no substantial alternative for the communication requirements of complex systems [Bol04a, Lee06]. Anyway, all given results in the diagrams below were computed based on fundamental parameters found in the according literature [Itr07a, Itr07b, Itr07e, Ben06, Jer05].

For decades, system performance has primarily been related to the computational parameters of a single processing unit. In the course of microarchitectural advances, it was empirically observed that performance increases roughly proportional to the square root of the increase in complexity −referred to as Pollack's law [Bor07]. For instance, doubling the complexity of a microprocessor yields only a 40 % growth in performance. However, this development is unacceptable in the long term, since the power consumption is linearly proportional to the complexity (see section 2.2.1). On this account, current architectures try to exploit the potentials of parallel computing. In the ideal case, running an application on two parallel resources cuts the

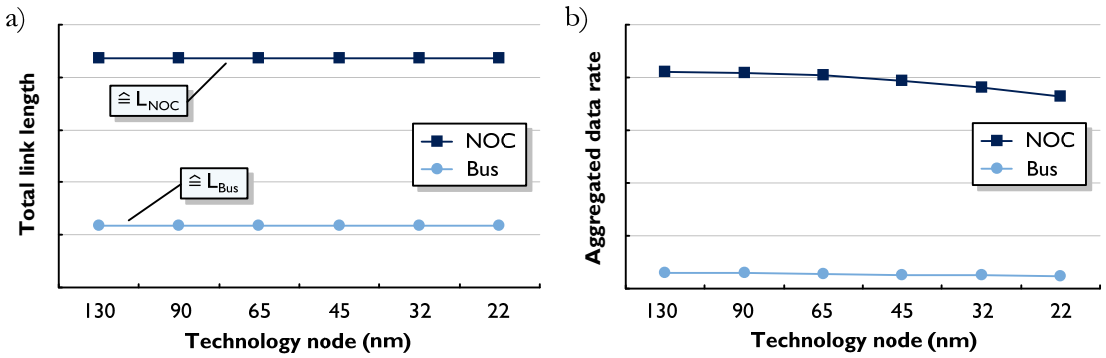

Figure 2-14 : Comparison of a) the total link length and b) the aggregated data rate across different technology nodes for an NOC and a shared bus (for fixed chip and network size)

execution time in half. This assumes that the application can be split into two equally large portions, which is rarely the case in practice. Generally speaking, the performance speedup S_{Amdahl} is a function of the number of available concurrent resources N_{res} and of the application's portion that can be executed in parallel k_{par}. This correlation is expressed by Amdahl's law [Amd67]:

$$S_{\text{Amdahl}} = \frac{1}{\left(1-k_{\text{par}}\right) + \frac{k_{\text{par}}}{N_{\text{res}}}} \tag{16}$$

The equation states that the performance speedup S_{Amdahl} is bounded by the portion of the application that has to be executed in series – this amounts to $1/(1-k_{\text{par}})$ for $N_{\text{res}} \to \infty$. Therefore, both the individual resources themselves and the degree of parallelism impact the computational performance significantly [Hil08]. However, this examination does not consider the communication among the resources, and thus distinguishes NOC and bus-based topologies by no means at all. For this reason, the results of the comparison herein and the following chapters solely relate to the communication characteristics of the architectures if not stated any different.

The subsequent comparison centers on those key aspects in line with the objectives of this thesis: performance, power consumption and reliability. It must be born in mind though that such a conceptual approach originates indeed revealing insights for both topologies considered, but not in terms of absolute figures since certain assumptions need to be made. Anyway, the two diagrams in figure 2-14 a) and b) plot the total link length and the aggregated data rate of NOC and bus-based topologies for progressing technology nodes. These results refer to constant chip size and a reasonable network size n of eight [Bor07, Lei06], which conforms to 64 resources in a quadratic arrangement (i.e. $N_{\text{res}} = n^2 = 64$). On this account, the total link length of the bus L_{Bus} and the NOC L_{NOC} are constant too, as they are dependent on the network size n and the width of the resources d_{res} – assuming that the width equals the height [Bol04a]:

$$L_{\text{Bus}} \propto 0.5 \cdot d_{\text{res}} \left(n^2 - 4\right) \tag{17}$$

$$L_{\mathrm{NOC}} \propto 2 \cdot d_{\mathrm{res}} \left(n^2 - n\right) \tag{18}$$

The total link length of the NOC is considerably larger, which is for a start an indication for higher area costs – observe figure 2-14 a). However, this alleged disadvantage translates into greater performance because it results from the multitude of concurrent links. Furthermore, the maximum frequency with which the links (i.e. also the bus) can be run is not a function of the total link length, but of the longest length of any particular link. With this in mind, the correlation of the frequency f can be expressed as:

$$f \propto \frac{1}{L_{\mathrm{WC}} \cdot t_{\mathrm{signal}} + t_{\mathrm{FF}} + t_{\mathrm{skew}}} \qquad \text{with} \quad t_{\mathrm{skew}} = t_{\mathrm{skew}}(L_{\mathrm{WC}}) \tag{19}$$

Here, L_{WC} is the worst case link length in the topology, t_{signal} denotes the signal delay per unit length, and t_{FF} and t_{skew} are the delays because of the associated flip-flops and the clock skew, respectively. Since the bus spans the entire chip in contrast to the short links of an NOC, L_{WC} is substantially larger in case of the bus. NOC-based topologies benefit from this fact in two ways because both the signal delay ($L_{\mathrm{WC}} \cdot t_{\mathrm{signal}}$) and the delay due to clock skew t_{skew} depend on L_{WC}. By means of the frequency, the relations of the **aggregated data rate** DR_{agg} can be described as follows:

$$DR_{\mathrm{agg}} \propto \frac{\alpha_{\mathrm{sat}} \cdot M_{\mathrm{act}} \cdot f \cdot W_{\mathrm{data}}}{\overline{d}_{\mathrm{avg}}} \tag{20}$$

Thereby, α_{sat} terms the saturation point of the communication architecture, W_{data} is the data width of the links (respectively the bus) and M_{act} represents the number of active participants that can send a message in parallel. Lastly, $\overline{d}_{\mathrm{avg}}$ relates to the average distance from the sender to the receiver in terms of links (also called hops) that are used. Hence, $\overline{d}_{\mathrm{avg}}$ equals one in case of the bus and $\overline{d}_{\mathrm{avg}} = n \cdot 2/3$ in case of the mesh-based NOC [Dal04, Str01]. As figure 2-14 b) shows, NOCs offer drastically higher aggregated data rate across the various technology nodes. The causes for this immense difference are intricate. On the one hand, the high saturation point α_{sat} and the average distance $\overline{d}_{\mathrm{avg}} = 1$ favor busses. On the other hand though, these two parameters cannot compensate for the disadvantages of the busses due to the single active sender M_{act} and the low frequency f – whereas the data width W_{data} is considered the same for both architectures. Finally, the data rates exhibit a declining course towards smaller technologies, which originates from a decreasing maximum frequency – in particular due to t_{signal} and t_{skew}. However, this is only true for the simplified assumptions taken here, such as constant data width W_{data}, chip and network size n. When such parameters are adjusted according to the specific technology nodes, it generally also results in an increase of the aggregated data rate DR_{agg} for smaller technologies – consider equation 20 and figure 2-14 b).

As mentioned above, adequate communication architectures also have to be scalable with regard to an increasing number of resources. Thus, the diagrams in figure 2-15 a) and b) relate the power consumption and the aggregated data rate against the network size. For this comparison of

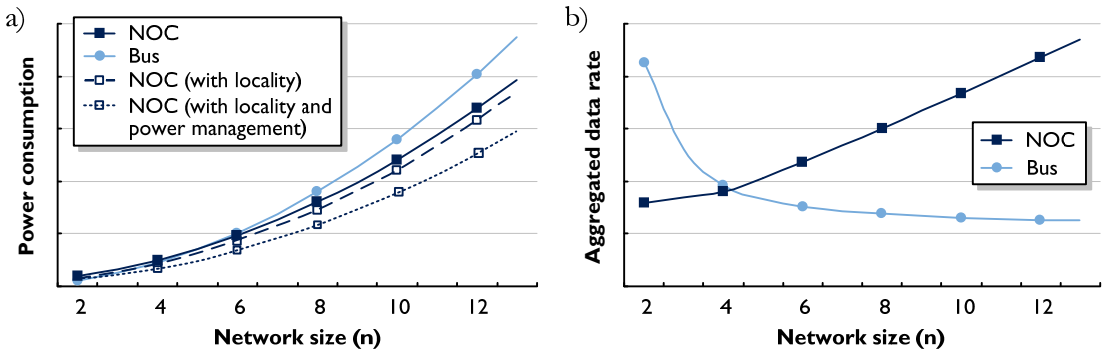

Figure 2-15 : Development of a) the total power consumption and b) the aggregated data rate for an increasing network size n, i.e. the number of resources $N_{res} = n^2$ (for 65 nm technology and equal frequencies for each network size)

the NOC and the bus-based topology, a 65 nm technology and constant chip size are assumed to simplify matters. Moreover, unlike the previous comparison, the frequency of the NOC is set to the maximum frequency of the bus for each network size. This facilitates a more reasonable evaluation of the power consumption, whereas solely the communication structure is accounted for. Accordingly, the total power consumption P_{tot} is the summation of all links as well as all modules that exclusively contribute to the communication among the resources – such as the routers or the central arbitration [Lee07]:

$$P_{tot} = \sum_{i=1}^{N_{module}} P_{module_i} + \sum_{i=1}^{N_{link}} P_{link_i} \tag{21}$$

$$\Rightarrow \quad P_{tot} = P_{tot}(N_{module}, N_{link}, L_{WC}, W_{data}, f, \alpha_{util}, ...)$$

Here, P_{module_i} and P_{link_i} denote the power dissipation of the i-th module and link, respectively. Correspondingly, N_{module} and N_{link} name the number of modules and links. From equation 21 it follows that P_{tot} depends on the topology (e.g. N_{module}, N_{link}), the physical placement (e.g. L_{WC}) as well as on the implementation of the various network elements (e.g. W_{data}). Lastly, P_{tot} is also affected by the way the components are operated (e.g. f, α_{util}), whereas α_{util} indicates the degree of utilization. A further breakdown of the parameters finally ends in equation 7.

The derived results for the total power consumption P_{tot} of the NOC and bus-based topologies are plotted in figure 2-15 a). As a start, it should be stressed that the quadratic rise there corresponds in fact to a linear increase of P_{tot} with the number of resources N_{res} – since $n^2 = N_{res} \propto P_{tot}$ – so that the scalability is not restrained. To be precise, busses are slightly superior to NOCs only for very small network sizes because of the lower complexity in the network elements. However, as the link capacitance of the busses has to be switched as a whole, this contributor starts to dominate the overall power consumption with increasing network size. By contrast, NOCs only have to load those links that are required to reach the destination. This fact can be exploited to achieve power savings, which is additionally drawn in figure 2-15 a) by means

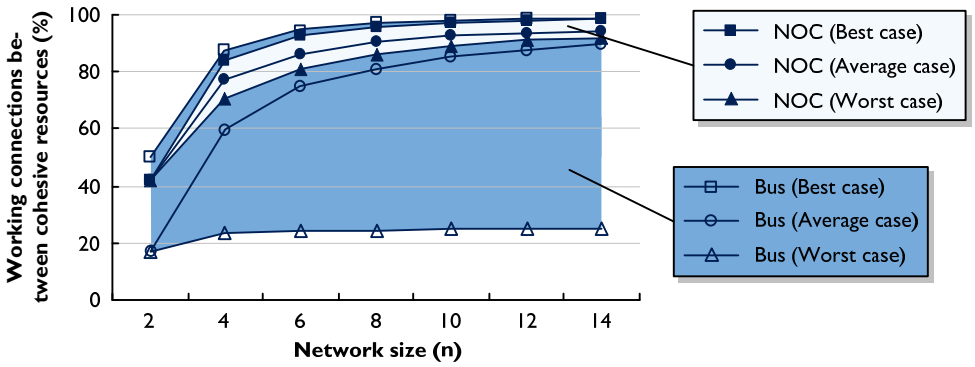

Figure 2-16 : Remaining working connections between cohesive system resources in case of a single, permanent and benign failure for an NOC and a bus-based topology

of two examples. First, locality describes the approach to mainly communicate with resources nearby, which reduces the average distance of a transmission $\overline{d}_{\text{avg}}$ (studied in subsection 4.4.1). Second, network elements that do not actively take part in the communication can be shut down. Such a power management can for instance be based on clock or power gating (investigated in subsection 3.4.1).

As mentioned in subsection 2.2.1, power figures themselves are only of limited value when they are not related to performance. Therefore, the aggregated data rate is drawn for the exact same scenario in figure 2-15 b). Here again, the large link capacitance of the bus primarily causes the decline of the data rate for larger networks. In case of the NOC though, the aggregated data rate rises continuously owing to the concurrent links that add with each further resource. This means that NOCs clearly outperform busses as soon as the network size is sufficiently large because they offer significantly higher data rates while consuming less power. Moreover, it is important to notice that all the results here are derived for the maximum frequencies of the bus. In effect, NOCs can operate at much higher frequencies whereby the appropriate graphs in figure 2-15 would additionally be shifted upward. Contrariwise, the power advantage of NOCs for large networks would further enlarge, if the aggregated data rates were scaled down to the bus – suitable leverages are for instance the frequency f, the data width W_{data} or the supply voltage V_{dd}.

Finally, the reliability of NOCs and busses was derived for this thesis and an analytical comparison. For that purpose, it was assumed that a single, permanent and benign failure occurs in the communication architecture, which circumvents the transfer of a message along the concerned element. Hence, resources that necessitate the defective element to communicate with others cannot establish one particular or several connections. Accordingly, figure 2-16 relates the percentage of working connections between all cohesive resources against the network size. Since the location of the failure in the topology significantly affects the number of working connections, the average case as well as the best and worst case is plotted for each study. The small spread of best and worst cases of the NOC results from the diverse and independent paths between the resources. On the contrary, a single failure can in the worst case cut the bus topology

in half, which leaves a cohesive system with half the number of resources behind. However, the probability of this worst case to happen decreases for larger network sizes just as the number of diverse paths in NOCs increases. Therefore, the percentage of working connections ameliorates in both cases, making larger systems more reliable. Lastly, it shall be noted that the results for the NOC are based upon a simple static routing scheme – namely XY-routing. Thus, the independent, concurrent links of the NOC permit to enhance the results by applying adaptive routing schemes that bypass the defective elements (investigated in section 4.1).

2.5 Resulting objectives for this work

The preceding sections addressed the main challenges of complex integrated systems. Therewith, the compiled classifications and purposeful findings form the basis to derive the resulting objectives for this work now. To begin with, it was shown that performance has originally been a prime characteristic for the scaling of technology and for the design of integrated circuits. In the meantime, serious concerns have additionally brought the power consumption to the attention of the designer. And recently, the importance of reliability and robustness as further characteristics is being stressed repeatedly [Itr07a]. However, the correlations of those diverse issues among one another have not yet been addressed thoroughly (consider sections 2.1 to 2.3). Therefore, one objective for this work is to demonstrate that performance, power consumption and reliability are closely intertwined. Accordingly, they have to be traded off against each other and can not be treated separately. Figure 2-17 illustrates this coherence whereby the change of a particular parameter will also affect the other two.

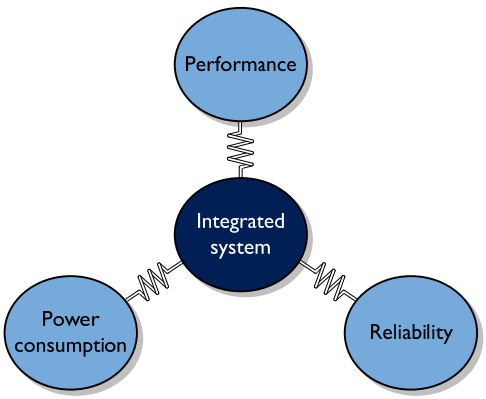

Figure 2-17 : Prospective design of complex integrated systems will have to trade off performance, power consumption and reliability

Furthermore, the technological advances have enabled to manufacture ever more complex integrated systems on a single substrate. This trend though has come along with growing difficulties for the design productivity, which has led to modular and communication-centric architectures. It was demonstrated in section 2.4 that Networks-On-Chip (NOCs) represent the

most applicable approach to cope with the increasing number of modules and their demanding communication requirements in complex integrated systems. However, to this day many of the challenges to design and to operate networks-on-chip still persist −in particular with respect to lifetime reliability. Thus, another objective of this work is to develop approaches for an improved implementation of components and architectures based on networks-on-chip.

In order to achieve the resulting objectives, the system design herein pursues a bottom-up approach −as illustrated in figure 2-18. That is, an underlying technology and a suited design style (e.g. a circuit technique) are chosen first. Subsequently, the modules for communication and computation are implemented. That comprehends all aspects from physical signaling to end-to-end connections as well as applications that manage or make use of the available features. Lastly, the design of such complex systems necessitates the use and the adaptation of an automated design flow before considerations for the assembly complete the entire chip design. Although such a layered representation (as the one in figure 2-18) is descriptive and commonly used, it pretends that the diverse layers are self-contained. However, this misconception masks the interlocking dependencies of the different layers among one another. For instance, design decisions within an individual layer generally also affect other layers. Hence, it is a further objective of this work to demonstrate as well as to exploit the fact that the diverse layers are actually intertwined and can rarely be regarded separately. Lastly, since the aspects of technology, design style, design flow and assembly are not solely related to networks-on-chip, they are only considered where due reference is necessary, so that the following implementations center on the communication and computation.

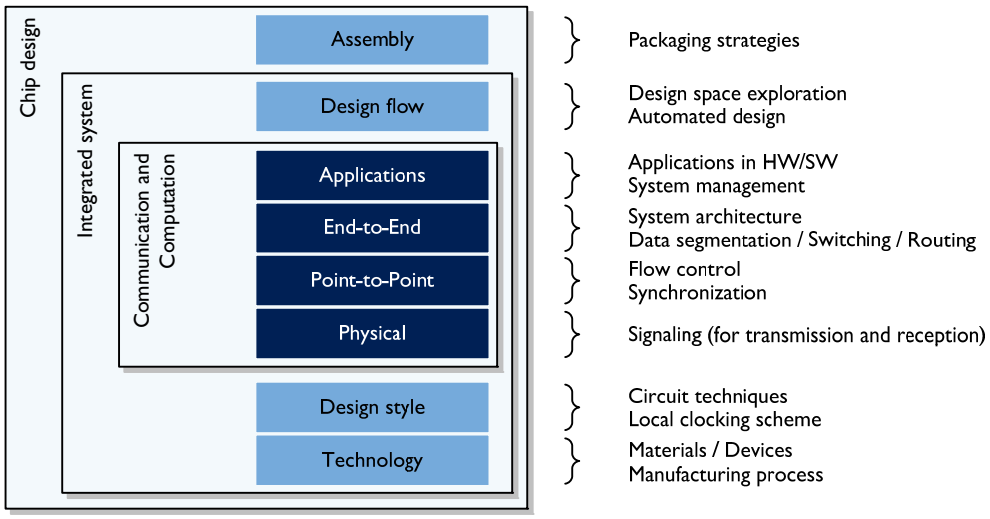

Figure 2-18 : The layered representation of chip design is descriptive but masks the interlocking dependencies of the individual layers among one another

Chapter 3

Components in on-chip networks

This chapter describes the implementation of fundamental components of Networks-On-Chip (NOCs), which serve as the foundation for the architectural investigations in the following chapters. Thereby, diverse alternatives to implement the components are evaluated, and several options for improvements are presented. According to the targeted bottom-up approach, the links connecting neighboring modules are covered first in sections 3.1 and 3.2. While the links operate on abstract signals, the routers are concerned with packets and their transmission between distant communication participants. Thus, the routers are dealt with in the subsequent sections 3.3 and 3.4. Lastly, in case that the communication protocols of the resources and the network are the same, the interfaces are just simple wires and do not need to be considered. However, their extent can be significant when unequal interfaces necessitate adaptations or when additional services are performed by the interface [Rad05, Wik03]. Therefore, it would go beyond the scope of this work to examine all possible implications of the interfaces, so that they are only mentioned where their negligence would distort the findings.

The physical limits of technology scaling have brought forth promising new proposals such as multi-gate transistors or 3D-integration (see also subsection 2.1.2). Even though one will indeed undergo a shift towards new technologies, a fundamental change is not foreseeable in the near future [Itr07c]. This means that device structures and manufacturing will be further refined, which enables to integrate billions of transistors on a single die of up to several square centimeters in size [Itr07a]. To accommodate these developments and to have a general foundation as a starting point, this work relates to monolithic integrated circuits and employs a current 65 nm technology from STMicroelectronics for the implementations [Cir09]. The corresponding technology offers design rules, transistor models and diverse gate libraries so that the network components can be implemented from the scratch or in a cell-based approach. On the basis of the associated gate libraries, static CMOS logic is universally used as the circuit technique of

choice – in particular with Standard Threshold Voltage (Standard-V_{th}) and 1.2 V supply voltage, whereas VHDL is used for the Register Transfer Level (RTL) design. Accordingly, such standard design tools from Cadence, Synopsys and Mentor Graphics are applied to ease and automate the design process.

3.1 Signal transmission across links

In the early days of microelectronics the transistors were very slow in proportion to the wires that interconnected them. Hence, the wires could be assumed to be ideal without evident curtailing of the accuracy of circuit modeling. In the course of technology scaling, this assumption does not hold true anymore, and yields unrealistic and by far too optimistic results [Ho01]. This conclusion becomes apparent against the background that wires in complex integrated systems have to span distances of several centimeters while operating in the gigahertz range [Itr07a]. Moreover, the wires total to an overall length of a number of kilometers [Itr07e]. Therefore, the physical impact of the wires and their real behavior need to be understood in order to model system characteristics accurately as well as to design chip components best possible.

The greatest importance of wires originates from the **links** in complex networks-on-chip because these connect the communication and computation modules across the entire chip. Figure 3-1 depicts such an example of a link between two adjacent routers. The distance between the connected components is a crucial factor, though it depends on chip size, topology and physical placement. Furthermore, the link width denotes how many signals (respectively wires) have to be transferred in parallel, whereas the link width subsumes both data and control signals. In any case, the essential task of a link is the transmission of signals from one end to the other. This corresponds to plain wires in the simplest case. However, diverse mechanisms are usually applied in order to improve signal transmission according to given requirements – such as differential signaling, signal amplification or signaling in the current and time domain [Rab03,

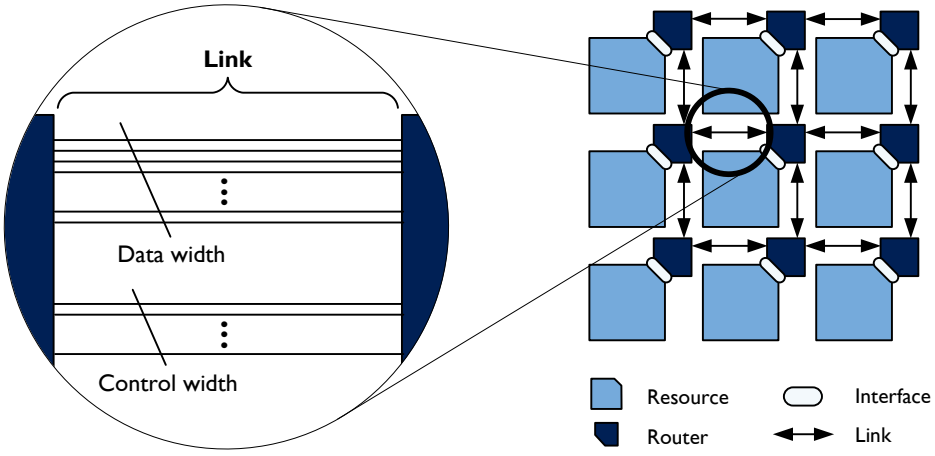

Figure 3-1 : Links connect the various modules and facilitate signal transmission between adjacent routers (whereas the link width subsumes data and control width)

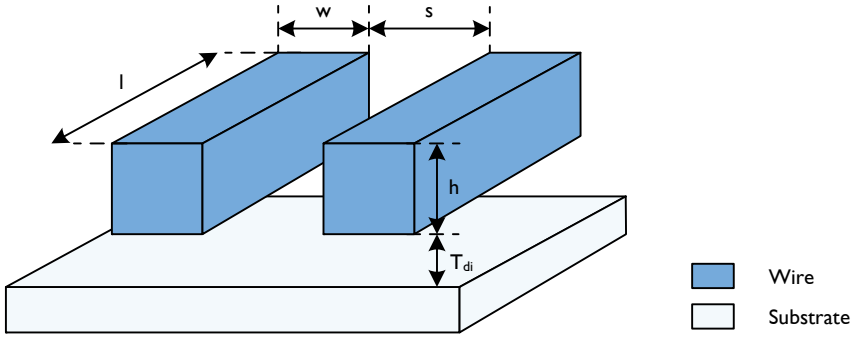

Figure 3-2 : Schematic illustration of the physical arrangement of on-chip wires with their geometrical identifiers

Syl00]. Beyond that, further complex approaches can also be part of the links, as for instance pipelining, synchronization or the coding for power and reliability targets [Pan08, Ber05].

3.1.1 Fundamentals of wires

The consideration of wire capacitance C_{wire} in the equations for performance and power – see equation 4 and 8 with $C_{load} = C_{mos} + C_{wire}$ – has been a first hint for the importance of wires for the entire system. In order to extract the different relevant parameters of wires and to estimate their influence, the basic correlations are introduced in the following. First of all, the physical arrangement of wires in integrated circuits is schematically illustrated in figure 3-2. Thereby, the geometrical measures of an individual wire are given by its width w, its height h and its length l. Moreover, the location of the wire in relation to the wires aside is denoted by the spacing s as well as the vertical distance to the adjacent layers is defined by the thickness of the dielectric T_{di} – in figure 3-2, the adjacent layer underneath is the substrate.

The **wire resistance** R_{wire} is a measure of the obstruction that is opposed to an electric current. According to that, the resistance directly affects performance, which is also reflected by the time constant τ that equals the product of resistance and capacitance. The value of the wire resistance R_{wire} can be derived from the geometrical measures and is defined as:

$$R_{wire} = R_{\square} \cdot \frac{l}{w} \qquad \text{with} \quad R_{\square} = \frac{\varrho}{h} \quad \text{and} \quad R_{wire} = R_{wire}(T) \tag{22}$$

Here, ϱ is the electrical resistivity and relates to the material of the wire. Since the resistivity ϱ and the wire height h are specific to the technology and cannot be set by the circuit designer, they are often united to form the sheet resistance $R_{\square}$. Besides, albeit the resistance R_{wire} might seem to be constant, it is a function of the temperature and increases proportionally with approximately 0.4 % per degree of temperature [Wes05]. With respect to scaling, wire resistance for a given length will increase significantly with smaller technologies, because the resistance is inversely proportional to the cross section of the wire – i.e. $R_{wire} \propto 1/(h \cdot w)$. In order to mitigate the boost of resistance, materials with lower resistivity are being integrated and the wire height h is not as

aggressively scaled as the width [Ho01]. Lastly, the examination of further factors –such as skin effect, contact resistance or scattering– is only of marginal importance here, and can be found in the literature [Syl00, Ho01, Wes05, Rab03].

The second parasitic parameter is the **wire capacitance** C_{wire}, which represents the amount of charge that must be added or removed in order to alter the electrostatic potential of a wire. In principle, wire capacitance C_{wire} can be deduced in a first approach from the parallel-plate capacitor and is divided into three constituents [Bal06a, Ho01]:

$$\begin{aligned} C_{wire} &\approx 2 \cdot \frac{\varepsilon_{di}}{T_{di}} w \cdot l + 2 \cdot \frac{\varepsilon_{di}}{s} h \cdot l + C_{fringe} \qquad \text{with} \quad C_{fringe} \propto l \\ &= 2 \cdot C_{ver} + 2 \cdot C_{hor} + C_{fringe} \end{aligned} \tag{23}$$

The first constituent C_{ver} stems from the two layers that are above and below the wire under investigation. The capacitance there is proportional to the area (i.e. $C_{ver} \propto w \cdot l$) and the permittivity of the dielectric ε_{di} between the layers. In contrast, the capacitance is inversely proportional to the distance of the adjacent layers, which is the thickness of the dielectric T_{di} in this case. The correlations also apply analogously for the second constituent C_{hor}, although this constituent relates to the adjacent wires on the very same layer. Correspondingly, the area here is derived from the sidewalls of the wire (i.e. $h \cdot l$) and the distance is given by the spacing s. It should be stressed though that the permittivity of the dielectric ε_{di} is not necessarily uniform and that the distances to the surrounding wires may also be different. At last, the constituent C_{fringe} refers to the fringing fields that cannot be covered by the model of the parallel-plate capacitor. Albeit various approximations have been proposed [Yua82, Bar88], C_{fringe} is most precisely determined by numerical field solvers [Dav03, Wes05].

The development of wire capacitance per unit length for decreasing technologies is not easy to forecast, since the changes of the different parameters somehow even out –since $w/T_{di} \approx$ constant and $h/s \approx$ constant. As aforementioned, the wire height h is in the strict sense not so aggressively scaled in order not to degrade the wire resistance R_{wire}. This in turn though is compensated by the introduction of low-$\varkappa$ dielectrics [Itr07e, Wes05]. In any case, wire capacitance C_{wire} per unit length is projected to remain roughly constant [Ho01]. Similar to wire resistance R_{wire}, wire capacitance C_{wire} also affects performance in terms of the time constant τ, respectively the RC-delay. In addition, the capacitance is also a main factor for the dynamic power dissipation P_{dyn}, as it sums to the load capacitance –see equation 8 with $C_{load} = C_{mos} + C_{wire}$– and has to be charged and discharged accordingly. For this reason, power consumption as a result of wires is consistently reported to account for up to 50 % of the total power dissipation [Mag04].

Finally, the question if it is necessary to consider inductance is controversially answered and often depends on the use case. In either case, the extraction of inductance is a time and data intensive undertaking, and rather impractical from a physical layout. Furthermore, simulation

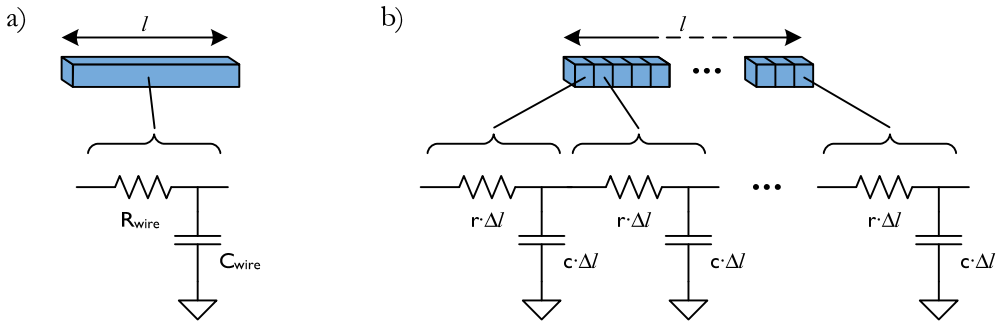

Figure 3-3 : Representation of basic wire models for wire length l: a) Lumped-RC model b) Distributed-RC model (whereas $R_{wire} = r\,l$ and $C_{wire} = c\,l$)

models also greatly expand, as inductance affects not only adjacent wires but also wires further away. To circumvent associated issues, most digital design technologies dictate design rules that make inductance negligible. Against this background, it is commonly agreed that inductance is only a serious issue for such fields as power networks, analog circuits or packaging [Itr07d, Wes05, Rab03, Ho01]. Therefore, inductance is also disregarded for the investigations herein.

So far, the parasitic parameters have been introduced separately, though the references to the time constant τ have already indicated their joint interrelation with performance. Consequently, the **delay of a wire** t_{wire} can be written as:

$$t_{wire} = k_{vr} \cdot \tau = k_{vr} \cdot R_{wire} C_{wire} = k_{vr} \cdot r\,c\,l^2 \qquad \text{with } R_{wire} = r\,l \text{ and } C_{wire} = c\,l \tag{24}$$

Whereas k_{vr} is a factor related to the voltage range of the delay time, r terms the resistance per unit length and c states the capacitance per unit length, which reveals the proportionality of both the wire resistance R_{wire} and the capacitance C_{wire} to the length l of the wire. According to this, the delay of a wire is even a function of the length squared (i.e. $t_{wire} \propto l^2$) so that the significance for system performance is additionally exacerbated.

3.1.2 Models for wires and complex links

The previous subsection highlighted the relevance of wire parasitics, and how they can be extracted from the physical geometry. This subsection introduces the model that was chosen in order to investigate the impact of wires, in particular in the context of complex links. The simplest models only consider a sole parameter – for instance the resistance or the capacitance, but not both altogether. However, on the basis of the findings in the previous subsection, both resistance and capacitance need to be contemplated jointly. Hence, the simplest approach considering this demand is the **lumped-RC model**, which is depicted in figure 3-3 a). It consists of one resistor and one capacitor in each case that represent the lumped parameters of the entire wire length l. Although the lumped-RC model is beneficially simple, it is inaccurate and produces pessimistic results for the performance. With respect to accuracy, the **distributed-RC model** is a

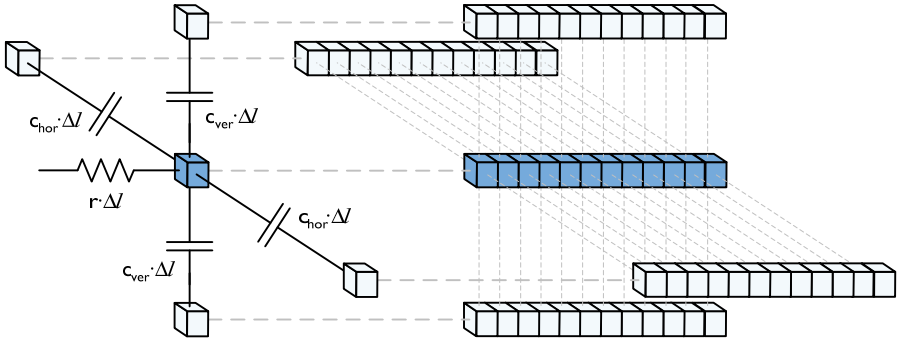

Figure 3-4 : Chosen distributed model for complex links with various wires running all around the wire under investigation – i.e. aside (in the same metal layer), above and below (in different layers)

better choice because it superiorly reflects the distributed nature of wire parasitics. A corresponding model is shown in figure 3-3 b) for the same scenario as for the lumped model. It is composed of $l/\Delta l$ sections, each with resistance $r\cdot\Delta l$ and capacitance $c\cdot\Delta l$. Thereby, accuracy and complexity can be traded off against each other by selecting an appropriate granularity.

The distributed-RC model is certainly suited to simulate the behavior of a single wire, though it is inappropriate for modeling complex links where various wires additionally interact with each other. Especially the static connection to ground does not reproduce the influences due to dynamically changing potentials on surrounding wires, which has significant effect on reliability and performance [Cap05, Ber05]. However, a further approach that can cope with the mentioned requirements for modeling complex links is illustrated in figure 3-4. It is basically an extended distributed model with four capacitors connected to the surrounding wires – instead of the one capacitor connected to ground. Thus, performance and reliability issues can be investigated thoroughly so that this wire model is used in the following, whereas the granularity is adapted to the corresponding needs. It should be noted though that the two horizontal capacitances $c_{hor}\cdot\Delta l$ as well as the vertical ones $c_{ver}\cdot\Delta l$ are identically termed in the figure to simplify matters. In effect, these can all have different values.

3.2 Approaches to improve signal transmission

The links that connect the distributed modules in on-chip systems are one of the fundamental components. For this reason, their characterization is essential in order to evaluate implementations and design decisions of system architectures. Consequently, the following subsections investigate the implications of repeater insertion and further complex solutions, which are applied to improve signal transmission. Thereby, the results of the implementations are based on the introduced distributed model for complex links (see figure 3-4).

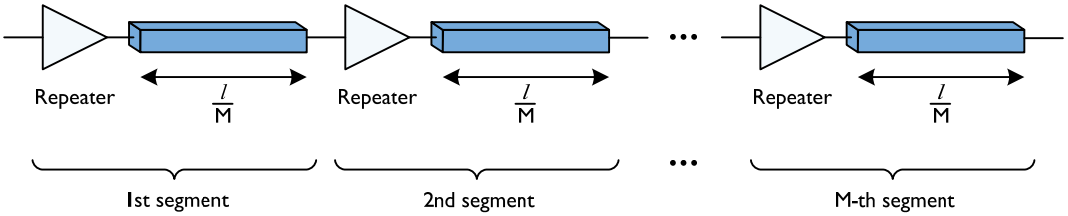

Figure 3-5 : Repeaters split the total wire length l in M segments of length l/M, whereas each repeater drives the incoming signal to the next segment

The measure of performance here is the **link delay** that describes to what extent a signal is delayed when passing along a link. More precisely, it is the time between the change of a signal at the input and the corresponding output of a link −whereas such points in time are defined as the moments when the signals cross 50 % of the supply voltage (i.e. $V_{dd}/2$). However, since signal delays are generally different for rising and falling slopes (denoted as $t_{0\to1}$ and $t_{1\to0}$) the link delay t_{link} is defined as the average of the two transition types [Rab03]:

$$t_{link} = \frac{t_{0\to1} + t_{1\to0}}{2} \qquad \text{with } t_{0\to1} \approx t_{1\to0} \tag{25}$$

Note that the link delay t_{link} matches the wire delay t_{wire} (as given in equation 24) when neighboring modules are simply connected by wires. In any case, all implemented designs below aim at roughly equal delays for rising and falling slopes (i.e. $t_{0\to1} \approx t_{1\to0}$). Thereby, the presented results refer to a link in the lowest metal layer (i.e. metal1) with 0.09 µm wire width and three times minimum spacing between the neighboring wires. A detailed description of the accomplished comprehensive investigations concerning wire width, spacing, layer selection, parameter variation and further aspects is omitted here, but can be found in the supervised theses [Säm07] and [Bra09].

3.2.1 Repeater insertion

Since wire delay is a function of the length squared (see equation 24 whereby $t_{wire} \propto l^2$), signal transmission poses a major concern for complex systems. Therefore, various approaches have been suggested to improve signal delay across long wires, whereas repeater insertion is the most prevalent technique [Bak85, Nal00, Wes05, Rab03]. Its fundamental idea is to split the total wire length l into M segments of length l/M each [Gla85]. This approach is illustrated in figure 3-5, where all segments are supplemented by a repeater that actively drives the incoming signal. Such repeaters are mostly inverters (or inverter pairs) and entail a certain delay for each segment. Consequently, the delay of a segment t_{seg} consists of the repeater delay t_{rep} plus the wire delay for length l/M (see equation 26). The determination of the number of segments M becomes thus crucial for the link delay with repeaters $t_{link,rep}$. If there are too many segments (large M), link delay

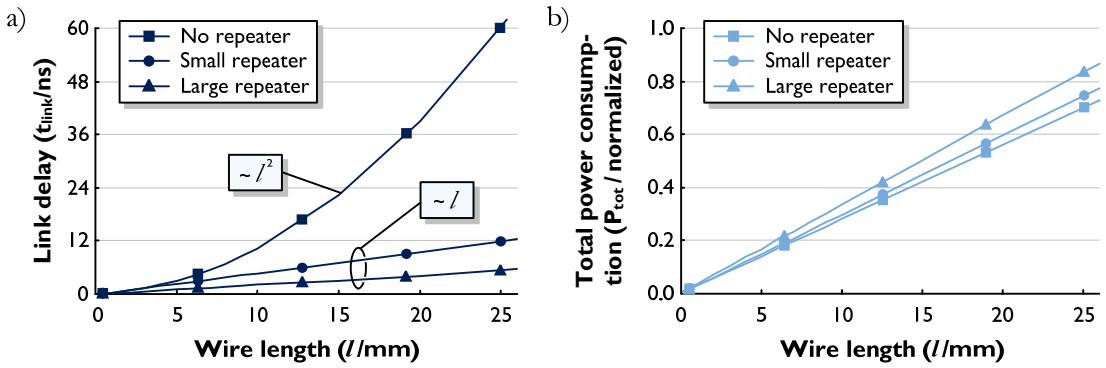

Figure 3-6 : Extract of the simulation results for repeater insertion in dependence on the wire length l, respectively the link length: a) Link delay t_{link} b) Total power consumption P_{tot}

is dominated by the repeater delay t_{rep}. Contrariwise, wire delay dominates if there are too few segments (small M).

$$t_{link,rep} = M \cdot \left[t_{rep} + rc \cdot \left(\frac{l}{M} \right)^2 \right] \qquad \text{with} \quad t_{seg} = t_{rep} + rc \cdot \left(\frac{l}{M} \right)^2 \tag{26}$$

In any case, when the number of segments M is chosen proportional to the wire length l, the link delay with repeaters $t_{link,rep}$ increases only linearly with the length l. Beyond that, two important aspects should be noted. First, simple repeaters impose unidirectional signal transmission, which favors their use rather in networks-on-chip than in bus-based architectures. Second, repeaters can cause a significant amount of short circuit current. This is because of the input signals of the repeaters that slowly change and thus turn both transistors of the applied inverters on for a significant time [Wes05].

However, repeater insertion is common practice in nowadays integrated systems. Therefore, various links of networks-on-chip were implemented with repeaters in order to determine their performance and power consumption. Such characterization establishes the foundation for the investigations of the complete architectures in chapters 4 and 5. In any case, inverters were used as repeaters herein to achieve best performance [Wes05]. Initially, the number of segments M and an appropriate size for the repeaters were analytically estimated based on the Elmore delay [Bak85, Liu04, Nal00, Bak90]. Although the estimate served as a useful starting point, extensive simulations were carried out for this work to closely determine the correlations of design parameters and to find the best setup for repeater insertion.

A meaningful extract of the entire results is presented in figure 3-6, where link delay t_{link} and total power consumption P_{tot} are plotted against the wire length l. Note that wire length corresponds to the link length in reference to the investigations of NOC-based architectures. At any rate, all link delays in figure 3-6 a) increase for longer wire length. However, the three charted cases differ substantially in the kind of their increase. On the one hand, link delay is a quadratic function of wire length when no repeaters are inserted into the link (i.e. $t_{wire} \propto l^2$). On the other

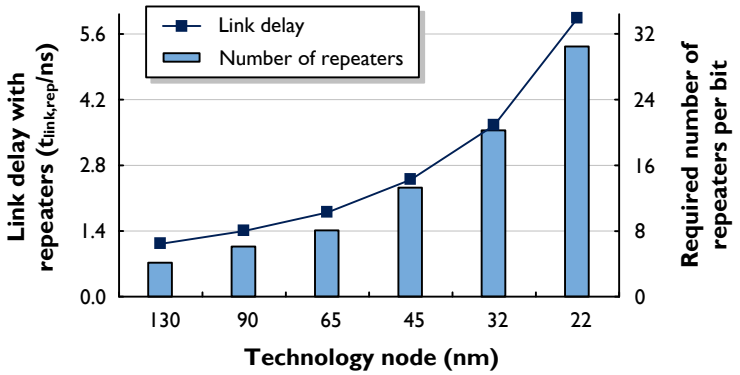

Figure 3-7 : Link delay and the required number of repeaters increase substantially across decreasing technology nodes (figures based on Elmore delay for a 10 mm link)

hand, link delay exhibits a linear dependency when repeaters are applied (i.e. $t_{wire} \propto l$). According to that, the results confirm the analytical description of equations 24 and 26. It is also interesting to notice that the size of the repeaters enables to adapt the link delay to application needs. Simply stated, larger (but fewer) repeaters perform better than smaller repeaters if reasonable limits are not exceeded. After all, it can be stated as a rule of thumb that it necessitates one repeater per millimeter of wire length for the used 65 nm technology.

The diagram in figure 3-6 b) shows the total power consumption P_{tot} for the same scenario as described before. Since the corresponding link delays vary considerably though, all power values are normalized to the same delay in order to gain a fair comparison (recall subsection 2.2.1). In contrast to the link delays in figure 3-6 a), repeater insertion only slightly impacts the power consumption. This is because the overall capacitance of the links is dominated by the wires and not by the additional repeaters. Besides, leakage and short circuit power contribute only a relatively small portion to the total power consumption. In a nutshell, while larger repeaters are highly beneficial in terms of performance (i.e. link delay), they worsen the power consumption. Hence, the size (and the number) of repeaters has to be chosen with care so as to trade performance off for power consumption.

Since both on-chip wires and transistors are subject to great changes due to the continuous scaling [Itr07e, Itr07a], it is interesting to examine how repeater insertion is affected by different technologies (see also subsections 3.1.1 and 2.1.1). Therefore, an analytical study was compiled for this work, whereas the results are depicted in figure 3-7 across decreasing technology nodes. Here, the computations are based on fundamental parameters found in the technology road-maps [Itr07a, Itr07b, Itr07e]. The results relate to the Elmore delay for a 10 mm link [Bak85, Nal00], whereas the link delay with repeaters and the required number of repeaters per bit are shown on the two ordinates of the diagram. Thus, figure 3-7 illustrates that link delay increases drastically for smaller technologies, which is attributed to the deteriorated wire parasitics and transistor properties. However, the required number of repeaters changes similar to the link delay. That implies that link performance and the effort to apply repeater insertion degrade to the same

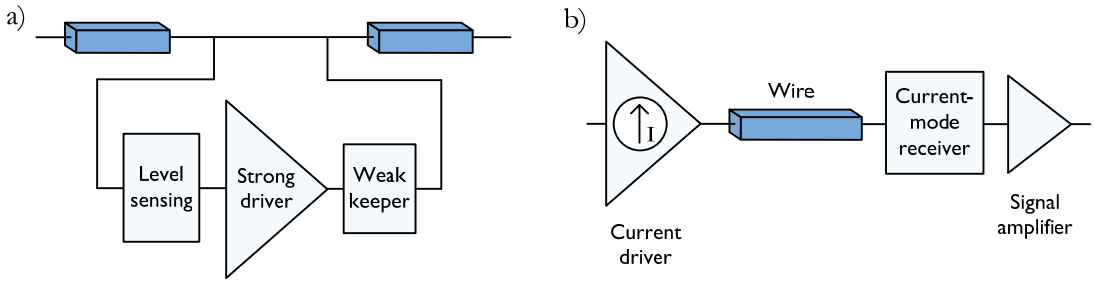

Figure 3-8 : Illustration of two techniques to improve signal transmission: a) Boosters sense when the wire is switching and aid the signal change b) Current sensing transmits data based on the strength of current (in contrast to the commonly used voltage level)

rigorous extent for smaller technologies. Moreover, since signal transmission across longer distances is crucial for system performance, on-chip links are generally designed to operate at their physical limits. This makes the links additionally susceptible for the growing number of failure causes in future technologies –as for instance, timing failures due to temperature or parameter variations [Bra09].

Concluding, repeater insertion is a capable approach to improve signal transmission and is supported by common automated design tools. The scaling of technology though largely impairs the link characteristics and the costs for repeater insertion. Hence, it necessitates other advanced techniques to overcome this threatening trend. Nonetheless, the results also underline the significance of wires and abstract communication in complex integrated systems [Syl00, Ho01].

3.2.2 Further solutions

Because of the challenges for repeater insertion in smaller technologies, a wide range of further solutions has been proposed to improve signal transmission. From these solutions, two selected approaches are presented in the following, which have been implemented and thoroughly investigated for this work.

One of them is the application of **boosters** [Nal02]. Figure 3-8 a) illustrates the basic design of a booster that is placed in parallel to the associated wire. It is important to note that this is in contrast to repeaters that are placed in series with the wire (see also figure 3-5). However, the first stage of the booster is the level sensing that detects signal changes on the wire. In case of a transition, the strong driver is turned on and supports the rising or falling slope –whereas the operation is based on the principles of hysteresis and positive feedback [Wes05]. After the transition is completed, the strong driver is turned off and the weak keeper maintains the signal level on the wire. In a nutshell, since the level sensing requires a certain delay, boosters are only beneficial for slow signal slopes (i.e. for long wires).

The second presented solution is **current sensing** [Mah01, Kat02]. The main idea is to transmit data based on the strength of a current, rather than based on the voltage level as it is

normally the case. Such an approach is shown in figure 3-8 b), where the current driver transfers the incoming signal into a determined level of current. The wire itself is terminated in the current-mode receiver in such a way that the current can flow along the wire. Hence, the transmitted signal can be converted back to the common voltage-mode, whereas the signal amplifier helps to restore the full voltage swing. In summary, current sensing performs well but implicates a certain design complexity. A drawback though is that the described setup draws static current. Thus, current sensing is primarily useful for long wires with high signal activity (i.e. rather high data rates).

There is a whole series of further interesting solutions, each with its own specific advantages and disadvantages. Such techniques range from elaborate design of physical links themselves (e.g. shielding, staggered layout) to various modifications of the fundamental principles of signal transmission −such as wave-pipelining, differential signaling or serial interconnects [Xu03, Lee06, Dob05, Bal06c, Cha03]. Besides, another set of approaches applies encoding to on-chip signaling in order to improve link characteristics, whereas mainly power consumption and reliability are aimed at [Pan06, Lee06]. It would go beyond the scope of this thesis though to introduce all of those published ideas. Nonetheless, one important aspect that is associated with signal transmission in networks-on-chip shall finally be mentioned: the synchronization. Since on-chip links connect distant components in complex integrated systems, they also have to deal with heterogeneous clock domains or parameter variations across the chip −which compromise the timing, respectively clocking [Krs07]. Hence, transmitted data has to be synchronized to the local clocking scheme of the receiving component. However, to simplify matters synchronization and issues of distributed timing are not further considered in this thesis without loss of generality. The interested reader is referred to corresponding literature [Itr07b, Rab03, Krs07, Tee07].

Concluding, albeit individual features of the miscellaneous solutions outperform repeater insertion, the sum of all important aspects is not convincing for their use in networks-on-chip. One general concern is the lack of design tools that support the automated implementation in complex systems. For these reasons, repeater insertion remains the prevalent approach to improve signal transmission. It is thus also used for the investigations of NOC-based architectures in this work.

3.3 Packet transmission across routers

On-chip links are essentially concerned with the physical definition of abstract signals and their transmission between adjacent modules. By contrast, **routers** manage data traffic between distant communication participants based on compound data units. The decomposition of such compound data units is shown in figure 3-9 together with the terminology. The smallest unit there is the physical unit (**phit**), which denotes all data signals that can be transferred in a single cycle across a link −i.e. the number of all these signals equals the data width W_{data}. When too many phits arrive at a module, it can cause the buffers to overflow. In order to prevent such an event, flow control schemes are applied that operate on flow control units (**flits**), whereas a flit

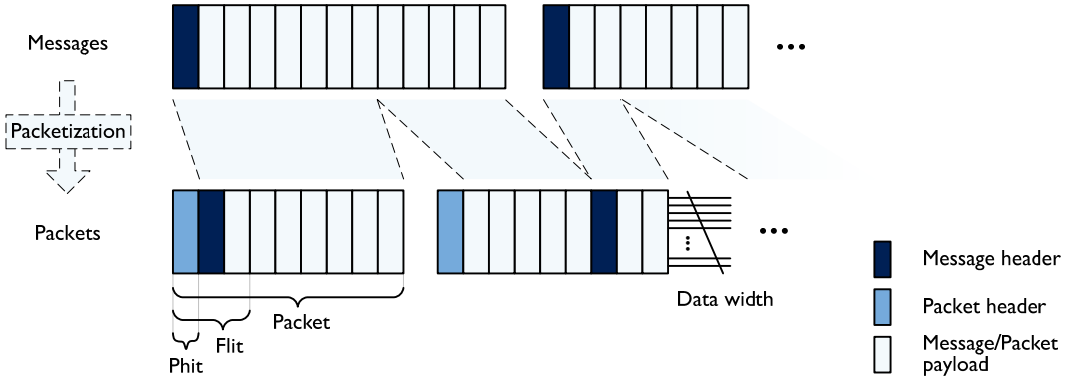

Figure 3-9 : Decomposition of application messages into data units that are handled in the different layers of an on-chip network, namely packets, flits and phits

constitutes at least one or several phits dependent on the application. Furthermore, several flits make up **packets** and these in turn represent **messages**, which are exchanged between the applications. The reason for the definition of packets is that the message size can exceed the constraints of the network – for instance, due to limited buffering or blocked network resources. Hence, packetization is an important task of the network interfaces and can also greatly impact communication characteristics [Ben06, Ber04]. Lastly, the headers of both messages and packets contain control information – at least the destination address – whereas the actual data contents are divided into the phits of the payloads. Note that flow control in this work is based on a single phit so that the notion of a flit is equivalent to a phit in the following.

As aforementioned, the primary task of the routers is to connect distant modules. This at first glance simple task actually requires a set of diverse protocols to accomplish flow control, routing and arbitration. The detailed meanings and correlations of these protocols are introduced in the following subsections together with investigations on their impact on communication characteristics. Independent of the different protocols and their functionality though, routers have to circumvent three fundamental network issues [Dal04, Ben06, Dua03]:

- Deadlock: Packets block each other at intermediate modules in the attempt to gain access to network resources (e.g. buffers or links), whereby this situation is static.
- Livelock: Packets move continuously within the network without ever reaching their destinations – for instance on faulty cyclic paths.
- Starvation: Packets stall because they never get access to a particular network resource, although the resource is dynamically granted to others.

The following subsections present the implementation of on-chip routers. However, it cannot be the target of this work to design the best possible router as this depends on application demands. Instead, different design options are investigated that demonstrate the interlocking characteristics of parameters as well as design layers. Furthermore, the results form the basis to

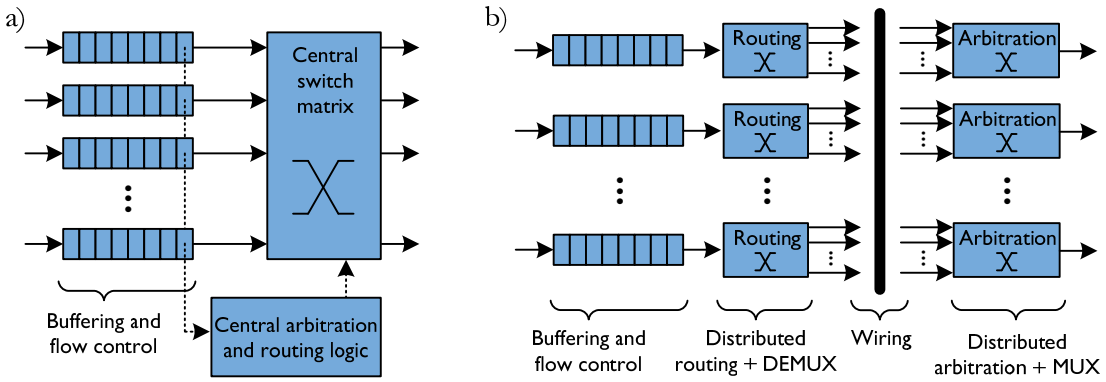

Figure 3-10 : Schematic illustration of a router architecture in a a) centralized and a b) distributed manner that supports fine-grained modularity

derive essential improvements for the routers and for the entire communication architecture (see section 3.4 and chapter 4).

3.3.1 Router architecture and general functionality

The general functionality of a router is described based on the different modules with their specific tasks. These modules are reflected in the most simple and straight forward architecture that is illustrated in figure 3-10 a) [Rij01, Kav04, Mel05, Bje06, Dal04]. For a start, the drawn architecture features a given number of incoming links that are buffered in independent storages [Kar87, Hlu88]. Hereinafter called FIFOs as these storage elements mostly work on the principle of First In First Out (FIFO) −i.e. the first stored element is also the first to leave the storage. Since the FIFOs are of finite size, some type of flow control logic is associated with them in order to obviate buffer overflow and data loss (see also subsection 3.3.2). Once a packet header is stored in a FIFO, the control information of the header is extracted and provided to the central arbitration and routing module. Based on the destination address, the routing logic determines which outgoing link is appropriate to reach the recipient. Subsequently, the arbitration logic has to grant access to the chosen link, and has to set the central switch matrix accordingly. Thus, as several data streams may simultaneously compete for the same output link, the arbitration has to fairly grant access to the links and can also contribute to resolve congestions.

The introduced centralized architecture is the most often cited one in common publications [Mil04a, Goo05, Kim08, Fra07, Rij01, Kav04, Bje06, Dal04, Mel05]. However, this type of architecture suffers from two main aspects. First, the centralized nature implies that the modules are rather large and complex to design, because they have to consider all incoming data streams concurrently. Second, a fine-grained modularity is not given so that changes of a function (e.g. the routing scheme) or the specialization of a certain link may affect the entire module. To overcome these concerns, this thesis suggests using a completely distributed router architecture as

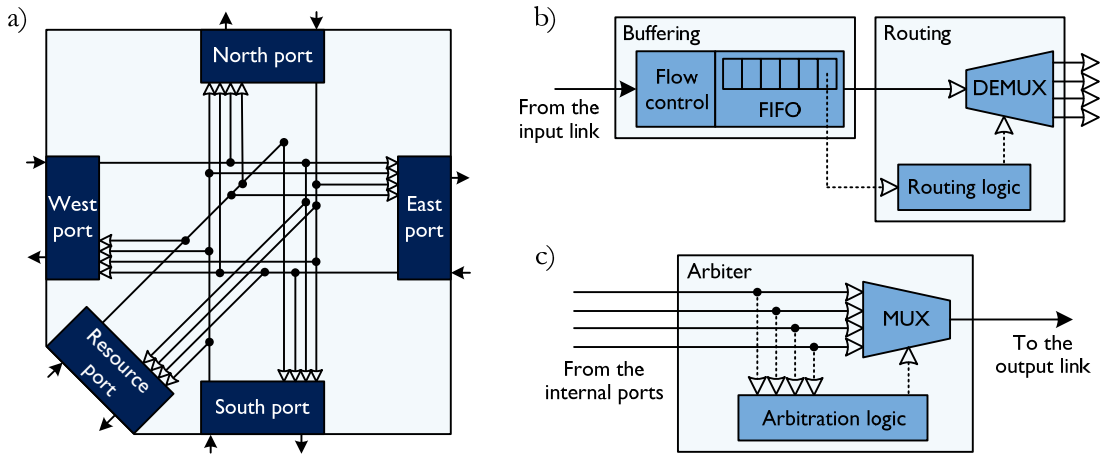

Figure 3-11 : Highly modular router architecture with a) five distributed ports whereas each port has independent modules for b) the incoming packets and c) the outgoing packets

shown in figure 3-10 b). Thereby, each incoming data stream features buffering and flow control just as in the centralized router too. The routing logic though is distributed and specifically assigned to each buffer. Analogously, the arbitration logic is also distributed but associated with the outgoing links in this case. Finally, the switch matrix is divided over those different routing and arbitration modules, whereby the functionality breaks down to a demultiplexer (DEMUX) and a multiplexer (MUX), respectively. This means that the wiring between DEMUX and MUX is such that every routing logic only forwards packets to the applicable arbitration logic.

When comparing these two approaches, they perform roughly similar in terms of performance, area and power consumption. However, the great advantage of the distributed architecture arises from its fine-grained modularity, which simplifies both general design changes and individual specializations. A few existing architectures [Zef04, Kim06, Mul04] partially utilize distributed approaches, though they do not fully exploit the advantages as regards power savings and reliability enhancements – as for instance clock gating based on link activity (see also subsection 3.4.1). Therewith, this is a first example how the design decision of the architectural layer is intertwined with design options in the other layers.

According to the preceding findings, the distributed architecture is chosen for the implementations herein and is illustrated in further detail in figure 3-11. This architecture serves as a starting point and as a reference, and it is thus described in the following together with the basic functional protocols. The example of the figure refers to a router with five independent **ports**, whereupon the number of ports is also termed **router degree**. In this case the router degree is five, as it can be found in mesh-based topologies – note figure 2-13 a). With reference to such two-dimensional topologies the ports are named according to the local resource and the points of the compass (i.e. west, north, ...). Furthermore, each port connects to its external neighbor by one incoming as well as one outgoing link, and comprises the corresponding control logic (see figure 3-11 b) and c), respectively). Hence, incoming data enters the buffering module that contains the FIFO itself and the flow control, which is a simple Request/Acknowledge

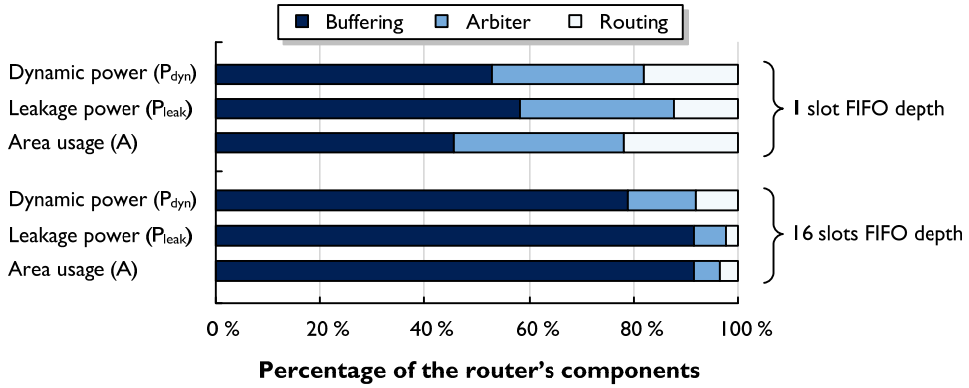

Figure 3-12 : Breakdown of power consumption (P_{dyn} and P_{leak}) and area usage by the modules of the distributed router architecture (for a data width of 64 bit)

scheme (Req/Ack). Subsequently, the routing module computes the appropriate output port and sets the DEMUX according to a dimension-ordered routing algorithm (namely XY-routing). Finally, the arbiter module of the selected output port grants access on the basis of a round-robin scheme, and multiplexes the selected data stream to the outgoing link.

The introduced router architecture employs admittedly basic protocols, nevertheless it can already indicate those modules that are of increased interest for improvements in respect of power dissipation and area usage. Therefore, the percentage of the different components of a router is depicted in figure 3-12 for the dynamic and the leakage power as well as for the area usage. The figures there relate to the three modules as shown in figure 3-11 b) and c) and are quoted for a FIFO depth of both one and sixteen slots −one slot conforms to one phit with a data width of 64 bit. According to that, the buffering module dominates the power and area metrics already for a FIFO depth of just one slot, which in fact boils down to a single pipeline stage. Hence, the FIFO should be targeted foremost to improve the given cost metrics of the router. Two things should be noted though. On the one hand, the implementation is based on standard-cells to construct the FIFOs and the switching elements, and in many cases customized implementations offer improvements [Mul06a, Wan03]. On the other hand, these figures rely on very basic protocols for flow control, routing and arbitration so that the proportions slightly alter when more complex protocols are implemented −such as virtual channels, adaptive routing or prioritized arbitration [Ben06]. Some of these existing design alternatives are introduced and evaluated in the following subsections, whereby several design improvements are derived as well.

3.3.2 Switching scheme and flow control

The **switching scheme** mostly refers to the type of connection and the granularity of data units that are transferred −note circuit-switched vs. packet-switched circuits. On the other hand, **flow control** for the most part rather references to the type of handshaking between two linked modules. However, these terms are very closely connected and are also used synonymously in the basic literature [Mar09, Ben06, Dal04, Dua03]. As a matter of fact, both terms describe the

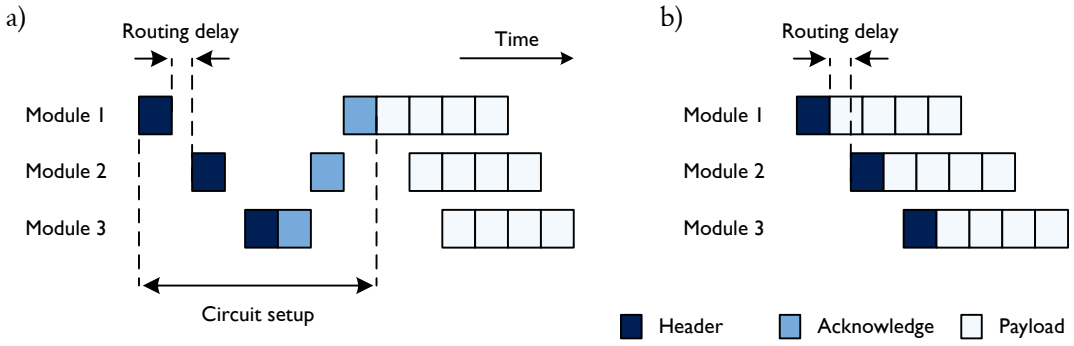

Figure 3-13 : Data transfer across three routers with a) circuit-switching and b) packet-switching, namely wormhole (the routing delay subsumes router, link and contention delays)

method in which messages are sent through the network, strictly speaking, if and when parts of the data are forwarded, buffered or dropped.

An often contemplated first distinction contrasts circuit-switching with packet-switching (see figure 3-13). In the former case, a circuit is set up prior to any payload entering the network. This is generally done by a header flit that traverses the network in order to reserve a circuit and the associated network resources. When the header has reached the destination, an acknowledge is returned to the sender whereby the actual payload can subsequently be transmitted without any further delay along the reserved circuit. By contrast, in case of packet-switching the payload immediately follows the header and can also be buffered within the network while awaiting access to network resources. Furthermore, different packets of the same message may also make their way through the network on diverse paths with different delays. Apart from that, several schemes of packet-switching are generally further distinguished based on their granularity of data units for flow control and the need for buffering resources – note store-and-forward or virtual-cut-through [Ben06]. However, as buffering highly affects the router costs (as shown in figure 3-12), wormhole switching represents the predominant technique for on-chip networks due to minimum buffer requirements of just a single flit [Bje06, Ben06]. Therefore, the following investigations solely relate to wormhole switching as representative of packet-switching – as it is also illustrated in figure 3-13 b).

For the sake of simplicity, all techniques of the compiled comparison herein are referred to as flow control schemes independent of a possible, more detailed classification. This being said, the selected techniques for the investigation are circuit-switching, dropping, Req/Ack, credit-based and On/Off. The first two candidates necessitate the least buffering of at most the header. While circuit-switching operates as explained before, dropping inspects the header and forwards the header – and the successive parts of the payload – instantaneously to the next router. In case of necessary, but blocked network resources the entire packet is dropped and discarded from the network. Hence, dropping requires additional protocols in the network interfaces in order to retransmit dropped and thus lost packets. The further three flow control schemes are conventional techniques of wormhole switching that mainly differ in the way of their

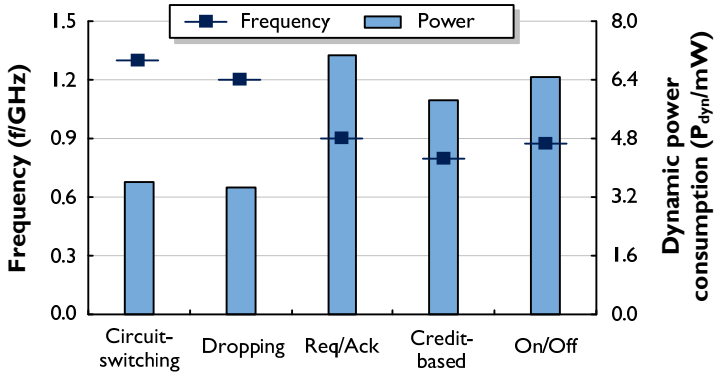

Figure 3-14 : Comparison of frequency and dynamic power consumption for different schemes of flow control (FIFO depth = 4 slots for Req/ack, credit-based and On/Off)

handshaking. Req/Ack sends a request together with each flit indicating the consecutive buffer to take over the flit [Dua03]. Once the flit is buffered, an acknowledge signal is returned to the initiating router so that it takes two clock cycles to transfer a flit. Credit-based and On/Off techniques reduce this delay to a single cycle by keeping track of the adjacent buffer with a counter and by signaling only when the buffer is full, respectively [Ben06, Dal04].

The router characteristics of the synthesized implementations in terms of the frequency f and the dynamic power consumption P_{dyn} are depicted in figure 3-14 for the different schemes of flow control and a data width of 32 bit. Striking are the positive properties of circuit-switching and dropping, which both feature high operating frequencies while dissipating rather sparse power. The reason is the small buffer usage in contrast to the remaining schemes that are equipped with FIFOs of four slots depth. Hence, their area requirements are substantially larger, which translates into higher power consumption and a lower operating frequency. Thereby, power dissipation and frequency are dominated by the buffering module, mainly by the FIFO.

However, the operating frequency f is no measure for the communication performance itself. Therefore, packet delays and the achieved data rates are contrasted in figure 3-15 a) for the different flow control schemes under investigation. **Packet delay** is defined as the time that it takes for a packet to reach its destination. This time starts when the packet header is about to enter the network and ends when the last flit has left the network. Thus, packet delay is a function of the distance between sender and receiver, the packet length and the physical delays of the modules −namely the router and link delays including the waiting times due to contention for network resources. For a start, figure 3-15 a) displays the ideal packet delay (i.e. without contentions) in respect of clock cycles for packets with 10 flits and a communication distance of 6 hops, which is synonymous with crossing seven routers. The ideal packet delay is the highest for circuit-switching due to the circuit setup at the beginning. The second largest packet delay is exhibited by the Req/Ack scheme. Even though, this scheme does not suffer from a longsome circuit setup, it is seriously affected by the delay of two cycles for the transfer of a single flit. On that account, the remaining flow control schemes offer the same and the lowermost ideal packet delays.

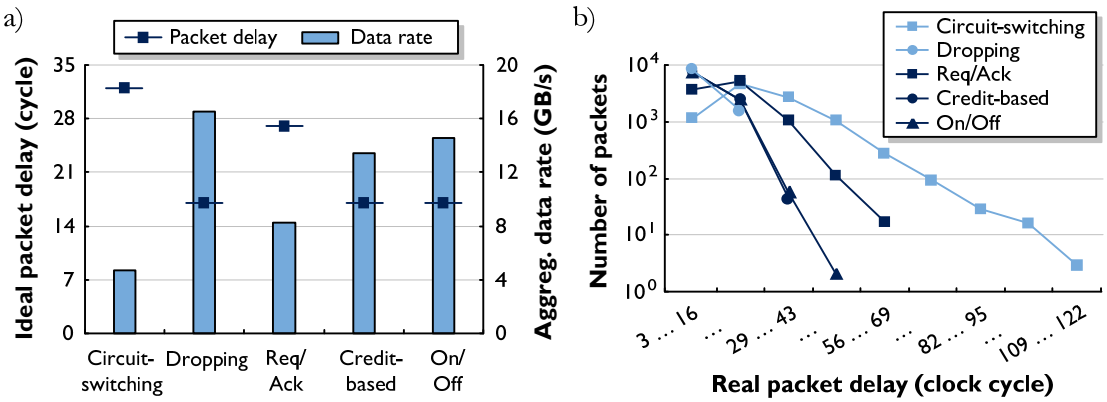

Figure 3-15 : a) Ideal packet delay is roughly inversely proportional to the achieved aggregated data rate b) Distribution of real packet delays for different schemes of flow control

The simulated aggregated data rates in a network with 81 resources –i.e. network size $n = 9$ and average distance $\overline{d}_{avg} = 6$ hops– are depicted on the second ordinate of figure 3-15 a). The aggregated data rate behaves about contrary to the ideal packet delay. More precisely, a high ideal packet delay entails small data rates and vice versa. That implies that the high operating frequency of, for instance, circuit-switching cannot compensate for the delays induced by the packets and their congestions in the network. By way of example, a circuit-switched packet occupies and blocks network resources substantially longer along its path so that it also forces other packets to wait longer for the blocked resources, which eventually hurts the data rate.

This behavior is further exemplified by the histogram of 10 000 packet delays for the different flow control schemes in figure 3-15 b) –note the logarithmic scale of the ordinate. The evaluation there closely matches the order of aggregated data rates in figure 3-15 a) whereby circuit-switching ranks at the lower end with significantly larger packet delays (up to 122 clock cycles) and a rather small data rate. This is followed by the Req/Ack scheme and both the credit-based and the On/Off schemes at the top end. Finally, the packet delays of dropping seem to be the best (all smaller than 29 clock cycles) as they do not suffer from any contentions. However, these figures only relate to those packets that successfully reached their destinations. In fact, a large number of packets were dropped (in this case 2173 packets), which requires additional retransmissions and potentially packet reordering at the receiving module. Such requirements only displace complexity into other abstraction layers and make dropping unbearable from a cost perspective, in particular when the communication load is high and the loss rate of packets blows up. Admittedly, different communication loads as well as other packet lengths (i.e. the number of flits per packet) also impact the remaining flow control schemes, but not in such a devastating manner as for dropping.

Concluding, credit-based and On/Off flow control offer the best communication characteristics, though they suffer from higher power dissipation. However, power consumption can be traded off for data rate by modifying the data width or FIFO depth (see also next subsection 3.3.3). Thus, these schemes are best suited for the application in networks-on-chip.

In any case, credit-based and On/Off flow control do not offer any means of Quality of Service (QoS) [Rij03]. To overcome that issue, this thesis suggests the following solution. QoS in on-chip networks is largely referred to as guaranteed throughput with reference to streaming and real-time applications [Rij03, Mur05a]. Circuit-switching inherently offers such guarantees as soon as a circuit is set up between sender and receiver. The same condition though holds also true for packet-switched flow control when a packet spans from the sender to the receiver. The important change in the implementation now is to avoid that the network resources are released after the first packet is processed. This can simply be achieved when an extra control signal is associated with each packet that indicates the end of a communication. By way of example, if this control signal ends the communication with the last flit of the first packet, the behavior is exactly as with conventional packet-switching. Several packets can be transferred along the same path though without competing anew for the network resources when the control signal only ends the communication with the last flit of the last packet. The control of such an additional signal needs to be handled in the interfaces, which produces a slight overhead. However, with this kind of implementation the advantages of both circuit and packet-switched schemes can be united and exploited according to the current application needs.

A couple of techniques have also been published that approach QoS by more elaborate methods. These techniques offer indeed gradual levels of QoS, but at the price of increased complexity both in the network and in the interfaces – such as hybrid switching, slot allocation or looped containers [Kim08, Geb09, Rad05, Rij03, Mil04b]. The suitability of both these complex techniques and the aforementioned, suggested approach ultimately depends on application-specific requirements. Hence a closing evaluation can only be given when these constraints are known to the circuit designer.

3.3.3 Data width and FIFO depth

In contrast to the previous subsections, changing the data width or the FIFO depth neither changes the architecture nor the kind of functionality. Nonetheless, these parameters highly impact cost and communication metrics, and thus are worthwhile to take a closer look at. Therefore, figure 3-16 presents the synthesis results of frequency f and dynamic power consumption P_{dyn} for miscellaneous data widths and FIFO depths, whereas the horizontal axes are logarithmically scaled. When considering the frequency f in figure 3-16 a) first, it becomes apparent that f drops for an increasing data width as well as FIFO depth. Thereby, the frequencies rather vigorously decrease for smaller parameters and gradually saturate for larger data widths and FIFO depths. This tendency applies to both parameters, but with a stronger sloping characteristic for the FIFO depth.

A similar course also appears for the dynamic power consumption P_{dyn} in figure 3-16 b), albeit for reversed trends of the parameters. More precisely, the power dissipation increases both for larger data widths and FIFO depths. It is interesting to notice there that the falling frequency f cannot countervail the growth in complexity (i.e. in capacitance C_{load}), so that the power growth actually approaches a linear dependency on data width as well as FIFO depth. Consider that the

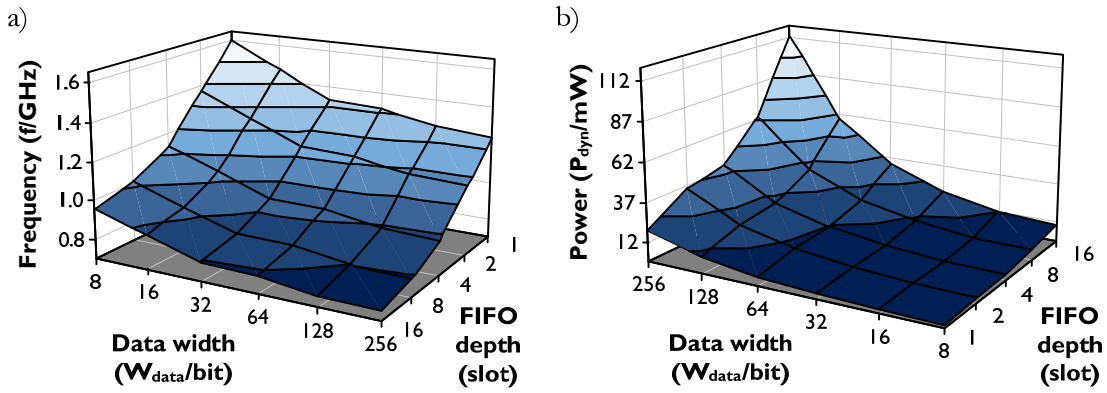

Figure 3-16 : Impact of varying data width and FIFO depth on the router's a) frequency and b) dynamic power consumption (power measured at maximum frequency)

proportionalities of P_{dyn} (as in equation 8) simplify to $P_{dyn} \propto C_{load}$, since f saturates towards constant values for large parameters. Finally, the leakage power P_{leak} – which is not accounted for in the figures – is in the range of a few microwatts and thus more than three orders of magnitude smaller than the dynamic power P_{dyn}.

As it has been shown before, frequency and power consumption are not fully meaningful if they are not related to their corresponding operation. That is why data rate and energy are depicted in figure 3-17 for diverse data widths and FIFO depths. The convention of the **energy** here attributes to the power P_{tot} and time for the average transfer of one kilobyte t_{kB} – i.e. energy equals the PDP = $P_{tot} \cdot t_{kB}$ as introduced in subsection 2.2.1. In case of the data width in figure 3-17 a), the data rate increases about linearly with larger data widths – consider the logarithmic scale in the diagrams. This course results from the fact that the growth of the data width clearly exceeds the diminishing frequency – note equation 20 whereby the data rate is proportional to $f \cdot W_{data}$. Accordingly, the energy rather sharply drops at the beginning, but approximates a nearly constant level for large data widths due to the data rate that compensates for the power rise. However, the communication metrics do not develop likewise advantageously for the FIFO depth, which is shown in figure 3-17 b). Thereby, both data rate and energy exhibit negative trends for larger FIFO depths, because a large FIFO hurts the frequency as well as the power consumption without directly increasing the data rate. It should be noted though that larger FIFOs are still useful to prevent network congestions. Thus, the individual data rate of a router in fact performs better with larger FIFOs when the overall network is heavily loaded or when the traffic is bursty – this behavior is also charted in figure 3-17 b) by the dashed curves. Besides, packet delays significantly benefit from larger FIFOs because local congestions are more quickly resolved when entire packets can temporary be buffered [Sun02]. This in turn also benefits the executed applications so that system performance and power consumption improve.

Concluding, increasing the data width benefits the data rate, but is largely constrained by area and power requirements. Furthermore, the FIFO depth is a costly parameter that nonetheless can

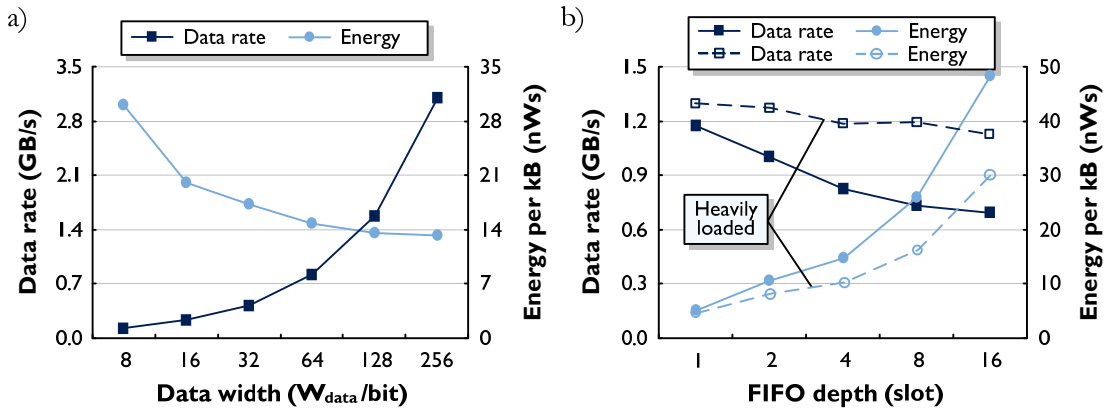

Figure 3-17 : Communication performance of an individual router in terms of the data rate and the energy per transfer of one kilobyte for different a) data widths (with 4 slots FIFO depth) and b) FIFO depths (with 64 bit data width) – note the logarithmic axes in both diagrams

still pay off under certain conditions, such as heavy or bursty traffic. Commonly, published implementations trade off power and performance resulting in moderate data widths and FIFO depths [Sal08]. Hence, a data width of 64 bit and a FIFO depth of 8 slots will be chosen for the reference architecture in chapter 4.

3.4 Approaches to enhance router characteristics

This section introduces several approaches to enhance router characteristics that can be applied to routers independent of their architecture, functionality or selected parameters – as described in section 3.3. Thereby, the implemented techniques are motivated through the development of Networks-On-Chip (NOCs) and their components. However, they are not necessarily restricted to NOCs and can partly also be transferred to other integrated circuits in order to improve system characteristics.

3.4.1 Clock and power gating to preserve power

Both techniques of clock and power gating aim at reducing the total power consumption while the component is idle. Hence, they are worthwhile to be considered for the implementation in NOCs since the individual components of the communication network exhibit very different behavior as regards their active operation time. First of all, clock gating is examined, whereas published results of gated routers relate to quite different levels of granularity. On the one hand, a coarse approach is described in [Mul06a], whereby the clock signal of the entire router is gated according to its activity. On the other hand, a fine-grained design is pictured in [Kim08] where only those modules obtain an active clock signal that are involved in processing a certain flit.

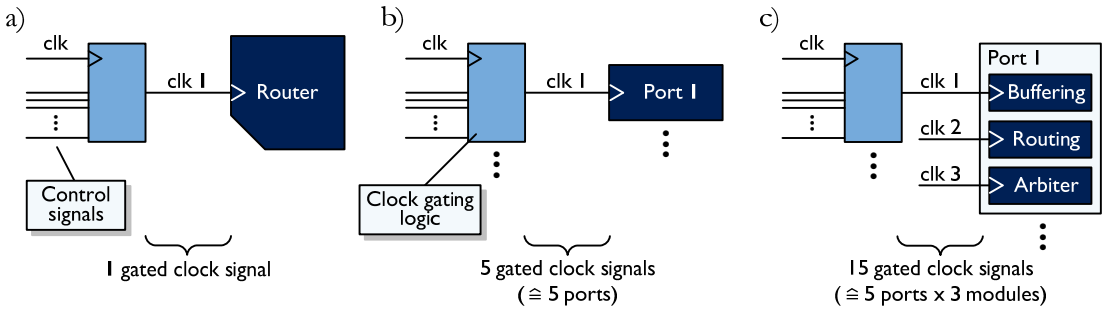

Figure 3-18 : Granularity of clock gating approaches: a) Gating the entire router b) Gating of individual ports c) Gating of individual modules, thus distinguishing input and output streams

In this thesis, the introduced reference router of subsection 3.3.1 is exploited by means of clock gating so as to compare its appropriateness for different levels of granularity – thus, spanning from a coarse to a fine-grained level of clock gating [Bhu07]. The fundamental prerequisite for such an investigation was already established by the distributed and highly modular architecture that enables to gate the various modules of the router individually. According to that, the first version gates the router as a whole, hence requiring only one clock gating logic that provides the gated clock signal – see the router-level version in figure 3-18 a). Furthermore, a port-level clock gating is illustrated in figure 3-18 b) whereas each of the five ports is provided with its own adjusted clock signal. And lastly, the fine-grained module-level version of figure 3-18 c) is proposed, which represents the smallest possible partitioning for the clock gating of a router. This version requires fifteen gated clock signals for the three modules within each port of the router.

The impact of clock gating in terms of the dynamic power consumption P_{dyn} is shown in figure 3-19 for the different versions of clock gating and under several traffic loads. These results were derived for a test setup of 10 000 packets with an average length of 10 flits that were randomly injected to the five ports – each port features 4 slots FIFO depth and 32 bit data width. Three discoveries can be observed from the achieved results. First, clock gating reduces the dynamic power by roughly 90 %, which originates from the savings of the large number of sequential elements and the clock network itself. Second, the finer the level of granularity of clock gating the larger are the attainable reductions. This follows from the fact that fewer modules are unnecessarily clocked. For instance, in case of a single flit crossing the router, only three modules are clocked in case of the router with module-level clock gating – namely input buffering and routing as well as output arbitration. By contrast, all fifteen modules are clocked in case of the router-level gating, and six modules in case of the port-level gating. In concrete terms, the module-level clock gating based on the proposed router architecture in this work improves the power savings by 52 % on average compared to the router-level design and by 42 % compared to the port-level design. Third, power consumption is a function of the traffic load. This in turn also favors the module-level approach, because here the modules are clocked the rarest on average.

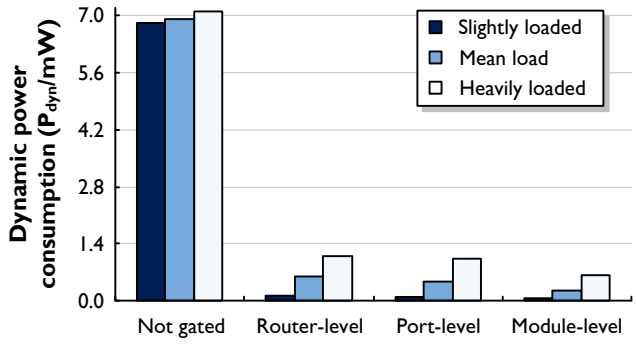

Figure 3-19 : Dynamic power consumption of a router for different granularities of clock gating and three exemplary traffic loads (while leakage power P_{leak} is about 2.8 µW)

Concluding, the module-level clock gating offers the best improvements since the overhead of clock gating logic is negligible in proportion to the rest of the router −notice that all versions occupy around 26 000 µm² and exhibit about 2.8 µW of leakage power P_{leak}. Finally, it should be considered that the delay of the clock gating logic can possibly impact the router performance, albeit the impact is generally marginal [Mul06a].

Unlike clock gating, power gating additionally targets leakage power P_{leak} by means of disconnected power rails in times of inactivity −i.e. during the idle mode of operation. For that purpose, extra transistors are interconnected between the power rails and the actual logic transistors, whereby the entire logic can be shut down. Although power dissipation is eliminated as far as possible, power gating comes along with several severe concerns. At first, power gating signifies considerable design overhead due to the extra transistors themselves and further necessary logic to restore the state of the component after an idle phase. Furthermore, power gating also impacts system performance since the power down implicates a wakeup time after the idle phase, and the extra transistors reduce the effective supply voltage of the logic −which again increases the delay (see equation 4). These contemplated issues restrain the general application in networks-on-chip. Nonetheless, power gating can still be beneficial when the active and idle phases of operation are known in advance or when a convenient system management is at hand (see also sections 4.5 and 5.3).

3.4.2 Application of different threshold voltages

Current technologies feature transistor and gate types with various threshold voltages. Therewith, predetermined designs can be differently synthesized in order to trade off performance and power dissipation. For instance, a design that is synthesized with a gate library of Low Threshold Voltage (Low-V_{th}) yields a fast design with both high dynamic and leakage power. Contrariwise, a design with gates of type High Threshold Voltage (High-V_{th}) leads to a rather slow design, though with little dynamic and leakage power consumption. For clarification,

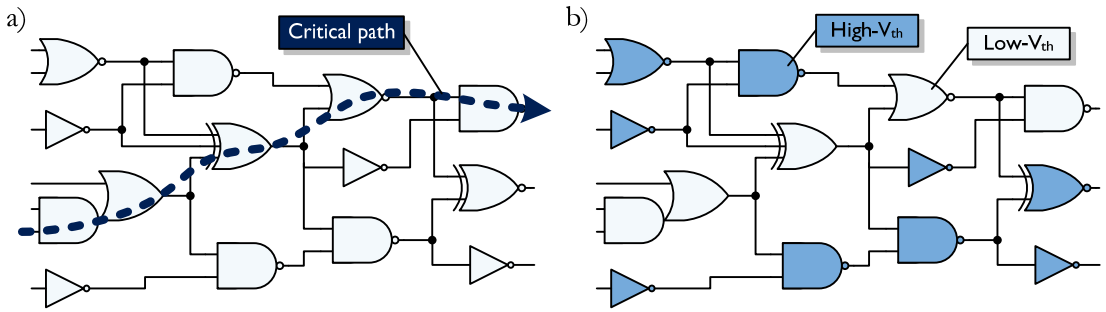

Figure 3-20 : a) Initial netlist with the critical path highlighted b) A Dual-V_{th} design: gates that do not prolong a critical path are replaced by gates with high threshold voltage (High-V_{th})

the reference router (with 4 slots FIFO depth and 32 bit data width) was synthesized with the three diverse gate types that are available within the selected technology. It should be stressed that the results in table 3-1 correspond to the same frequency target of 530 MHz so as to gain a fair comparison. The dynamic power P_{dyn} in this example decreases slightly for smaller threshold voltages, which is owing to the same frequency target and the smaller area, respectively. However, the leakage power varies in the order of magnitudes between the different gate types and can thus be crucial in applications with long idle times [Yeo04, Sou09, Nar05].

Table 3-1 : Results for the synthesis of a router with different gate types in terms of the threshold voltage for the same frequency target of 530 MHz

	High-V_{th}	Standard-V_{th}	Low-V_{th}
Dynamic power (P_{dyn}/mW)	3.01	2.79	2.71
Leakage power (P_{leak}/µW)	0.26	1.65	18.08
Area (A/µm²)	23 446	22 832	21 712

Instead of synthesizing the entire design with one type of threshold voltage, several techniques strive for combining the performance advantage of Low-V_{th} devices with the leakage benefits of High-V_{th} devices [Wan98, Gao05, Sul04, Sun99, Wei99]. Such techniques –in summary called Dual-V_{th}– work as follows: A design is at first synthesized with solely low threshold devices, so as to reach the highest maximum frequency. Subsequently, the devices of the critical paths are identified –see figure 3-20 a) with a simple example. Those critical gates there are the ones that determine the longest delay within the circuit. Thereupon, the remaining gates are gradually substituted for gates with high threshold voltage as long as the new gates do not prolong any critical path –note the modifications in figure 3-20 b). Thus, the timing margin (called slack) of the non-critical gates is taken advantage of in order to apply slower gates that feature reduced leakage currents. That is to say, only the signal propagation within the non-critical paths is delayed, while the overall performance remains constant.

The Dual-V_{th} technique –as depicted in figure 3-20 b)– replaces the logic gates as a whole. Certainly, better results can be achieved when the exchange is based on individual transistors

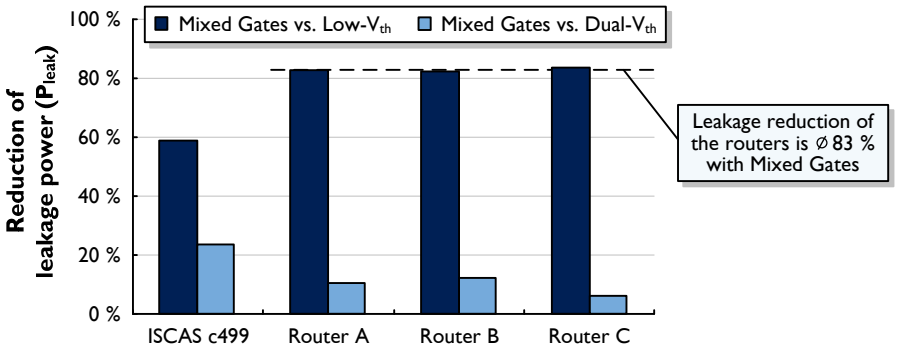

Figure 3-21 : Reduction of leakage power for diverse routers and a reference design (i.e. ISCAS'85) due to the application of different gate types −whereas performance is constant within the individual designs

instead of entire gates. However, such an approach is not in line with standard design tools [Sil07]. Therefore, the **Mixed Gates** strategy was developed that offers a fine-grained approach without sacrificing the applicability of gate-based design tools [Sil06, Sil07]. While Dual-V_{th} utilizes gates with either High-V_{th} or Low-V_{th} (see figure 3-20), the Mixed Gates technique also applies logic gates where different threshold voltages are mixed within a single gate. Therewith, the slacks of the gates can more comprehensively be exploited so that more gates can be substituted for.

In order to evaluate the impact of the aforementioned techniques, various designs were synthesized correspondingly. The originated results for the reduction of leakage power are displayed in figure 3-21 for the Mixed Gates approach in comparison to the common Low-V_{th} and the Dual-V_{th} designs. Thereby, the designs constitute three representative routers that greatly vary in complexity, and a reference from the ISCAS'85 benchmark suite [Han99]. First of all, it is important to mention that within the different designs the maximum frequencies are identical for all three techniques of the investigation. Moreover, both the area usage and the dynamic power also diverge only marginally. However, leakage power is drastically reduced when different threshold voltages are applied. More precisely, Mixed Gates achieve an average leakage reduction of 83 % for the routers compared to the conventional Low-V_{th} designs. This stands for an additional 10 % cutback in relation to the Dual-V_{th} designs. Such improvements are significantly higher than for the reference design (i.e. ISCAS c499) because the routers consist of two unbalanced pipeline stages. That implies that all paths in the short pipeline stage are not critical for the router performance. Hence, all these gates can be replaced.

Concluding, the application of different threshold voltages significantly decreases leakage power without a performance penalty. However, these leakage savings are potentially traded off against reliability because such approaches increase the number of critical paths, which leads to more probable timing errors and reduced yield. On the other hand, it should be investigated in future works to what extent the gate exchange can also be used to increase system reliability against specific causes. For instance, the described replacement by High-V_{th} gates commonly

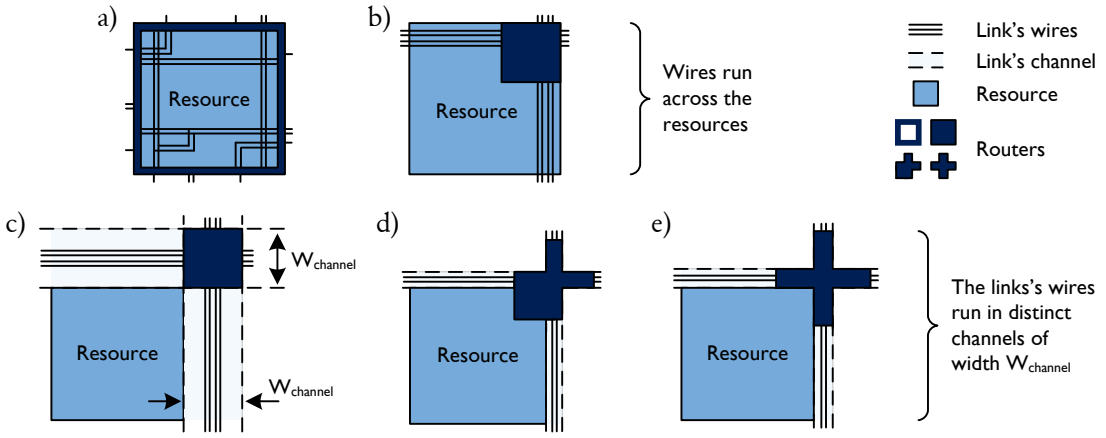

Figure 3-22 : Illustration of different approaches for laying out a router: a) Thin router b) Integrated router c) Square router d) Cross router e) Plus router

hardens the system against gate oxide breakdown [Cor08]. Lastly, against the background of complex systems both the allocation algorithms and the timing analysis of such multi-threshold designs will have to be further investigated, since they necessitate extensive and costly computational efforts [Sal07b, Sil07].

3.4.3 Router layout for reduced area costs

The introduced techniques of the two preceding subsections can improve a given design by modifying its gate netlist. Beyond that, system characteristics can also be ameliorated for a fixed gate netlist by the type of layout. Referring to networks-on-chip, it is the links and the routers that largely impact the communication costs as regards the area usage. Since the router area is determined during synthesis –respectively by the gate netlist– it is the geometry of the layout that remains to reduce the area costs.

Early publications often describe the router geometry as a thin switch –hereinafter called thin router to avoid misunderstanding [Dal01, Bje06]. Figure 3-22 a) illustrates such a scenario where the router is distributed around the associated resource. Instead, those few complex prototypes that have been released rather integrate the router cohesively as part of the resource, which facilitates a higher operating frequency of the router –shown in figure 3-22 b) [Mul06b, Van07]. However, in case of the thin as well as the integrated router, wires of the communication network run across the resources. Accordingly, the strict modularity is forfeited and undesirable layout issues emerge –such as signal integrity, repeater placement and wiring congestions. For these reasons, such two layout approaches are only applicable for homogeneous systems due to associated productivity concerns (see also section 2.4). As a result, current works depict on-chip networks with square routers and dedicated channels for the links, as pictured in figure 3-22 c) [Jan03a, Bje06, Mil04a]. Since the resulting physical channel width $W_{channel}$ –i.e. the

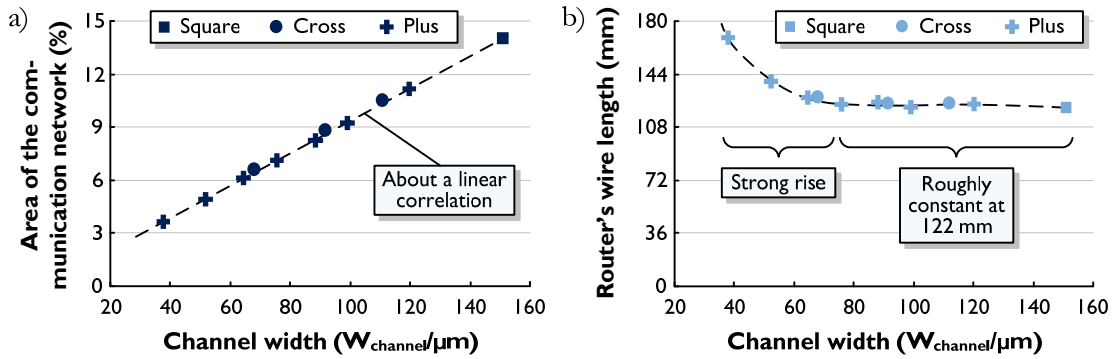

Figure 3-23 : a) Percentage for the area of the communication network (i.e. links and routers) b) The internal wire length of a router increases strongly for narrow channel widths – results are based on post-layout data

distance between adjacent resources – is derived from the router area though, the links contribute decisively to the required area for the communication network.

To overcome the concerns of lost modularity and wasted chip area that common router layouts exhibit, this thesis suggests two enhanced structures as depicted in figure 3-22 d) and e). Thereby, the channel width $W_{channel}$ is effectively reduced by relocating part of the router into the actual channels of the links. The first version in figure 3-22 d) is called **cross router** and retains an extension at the lower left side that mainly comprises the port of the resource. As opposed to the cross router that only places the four remaining ports (i.e. west, north, ...) inside the channels of the links, the second version in figure 3-22 e) positions all five ports within the channels. The distinctive form of the arising layout gives this router its name: **plus router**.

The three versions that do not forfeit the modularity of the resource were practically layouted in order to evaluate the router characteristics based on back-annotated results. Starting point was a given gate netlist and equal core utilization so that the router area was constant across all versions and parameters. More precisely, each router accounts for an area of 24 500 µm² and the area usage of a resource is assumed to be 3.72 mm² [Van07].

Therefrom, figure 3-23 a) was derived that depicts the percentage for the area of the communication network against the channel width $W_{channel}$ for the square, the cross and the plus router. The area there is defined as the sum of the router and link area. According to that, the area of the communication network with cross or plus router decreases approximately proportional to the channel width due to shrinking area requirements of the links. This discovery clearly favors the proposed cross and plus routers whereas the channel width should be as small as possible from this perspective. However, it is not only the layout of the link itself that restricts the least suitable width of the channel, but also other characteristics of the router that are affected. To clarify this impact, figure 3-23 b) relates the router's internal wire length against the channel width $W_{channel}$. The results are based on post-layout data and relate to the sum of the

lengths for the entire physical wiring of a router. Thus, the router's wire length is a function of the geometric arrangement of the logic gates and is roughly constant across a wide range of channel widths – here from $W_{channel}$ = 155 µm down to 75 µm. For small channel widths though (< 75 µm), the cross and plus routers extend far into the channels and the router's wire length exhibits a strong rise, which deteriorates performance and power figures. By way of example, the initial routers with about 122 mm wire length operate at 385 MHz and dissipate 2.96 mW of power. In contrast, for a channel width of 38 µm the plus router comprises 169 mm wire length and the maximum frequency drops down to 341 MHz with a power consumption of 2.66 mW. At any rate, the area of the communication network can be reduced here by about 50 % without impairing other router characteristics.

Concluding, the geometric layout in form of the cross and the plus router is greatly beneficial for complex networks-on-chips, whereas the cross version is to be favored when the port of the resource includes additional functionality, respectively necessitates larger area. As a rule of thumb, the channel width that balances most parameters best arises when the flow control and the buffers are located within the actual channels. Lastly, the integrated router signifies admittedly the least area overhead but trades communication area off against strict modularity.

Chapter 4

Architectures and algorithms of networks-on-chip

The previous chapter introduced and investigated the fundamental components of on-chip networks. However, it is not only the properties of those individual links and routers that account for an efficient integrated system, but it is two more domains that additionally impact the entire system characteristics. The first is the architecture that determines both the structural type of configuration and the physical arrangement of the components. The other domain is the algorithms that are decisive in order to operate complex systems and to exhaust the potential of the underlying architectures. Therefore, this chapter covers such corresponding aspects and deduces several approaches for enhancements.

Before several quantitative parameters are introduced, a few qualitative remarks on communication architectures are considered. The development of complex architectures based on Networks-On-Chip (NOCs) favors rather domain-specific platforms than application-specific solutions. The cause for this is the immense non-recurring costs and thus the need for high volume production in order to make profits [Soi03, Lat08]. On these accounts, information about the on-chip traffic patterns is not available for the most part so that refinements can hardly rely on application-specific knowledge. Instead, complex architectures have to satisfy various purposes, which necessitates a more general implementation. However, generalization often produces underutilization so that system characteristics and costs have to be traded off carefully. Independent of the application though, all on-chip communication architectures should comply with the following objectives:

- Data integrity: Data should arrive unchanged at the destination node, thus it should not be corrupted by any means.

- Lossless transmission: Neither packets nor any kind of data is to be lost during a transfer.
- In-order delivery: The sequence of data (e.g. packets or flits) should remain the same so that data does not have to be costly resorted.

The types of published communication topologies span a very wide range, whereas many of these suggestions were originally motivated by distributed networks [Dal04, Dua03, Tan02]. Against the background of networks-on-chip though, only few topologies promise to be appropriate for the practical use. For example, star topologies or specifically tailored, heterogeneous topologies are implemented in particular for small and application-specific systems [Lee04, Bor03, And03]. Furthermore, domain-specific platforms necessitate a higher degree of flexibility and generality as opposed to application-specific systems. Hence, corresponding topologies are rather based upon regular structures that are complemented with domain-specific customizations – such as in multimedia, data retrieval or telecommunication domains [Lat08, Lat07, Soi03]. Lastly, general purpose processors benefit from regular topologies since most of the resources are of homogeneous nature. Thus, the planar mesh topology is the most prevalent representative in this field [Van07, Mul06b, Sal07a].

When it comes down to the network size, a similar spread of approaches has been reported as for the topologies. Such network sizes range from a very few to very many resources and depend on the needs of the applications and the acceptable overhead. At the lower end, a couple of prototypes have been manufactured with less than 25 resources [Lee06, Lat08, Lee04, Mul06b, Bor03, Geb09]. Beyond that, a great number of scientific works investigated network sizes of up to 256 resources based on simulations, abstract models or the predictions of technology roadmaps [Pen06, Jan03b, Art05, Mul06b]. Up to now, Intel's TeraFLOPS processor and Tilera's TILE-Gx processor family are by far the largest practical implementations of NOCs with up to 100 homogeneous resources [Hos07, Til08]. In order to select an appropriate network size for this thesis, two things need to be considered that constrain the choice. On the one hand, on-chip networks only become fully beneficial for larger system sizes (see subsection 2.4.4). On the other hand, as chip size is confined by manufacturing technology, there is a practical upper limit for the number and size of resources as well. Concluding, based on the contemplated aspects of the various topologies and network sizes, the following examinations emanate from a system with 81 resources arranged in a planar mesh topology.

Before routing algorithms are evaluated in the next section, a couple of definitions and correlations are introduced in the following, which come into play when investigating complex communication architectures. First of all, a given topology is described by inherent parameters that relate to the structure itself. For example, the **communication distance** $d(n_s,n_d)$ denotes the distance from the sending node n_s to the destination node n_d in terms of hops. Hence, the **diameter** d_{max} of a topology (as given in equation 27) results from the longest distance between any communication pair from the set of all nodes N [Dal04]. Lastly, also based on the communication distance $d(n_s,n_d)$, the **average distance** $\overline{d}_{avg}$ is calculated from the arithmetic mean of all connections between different nodes (see equation 28).

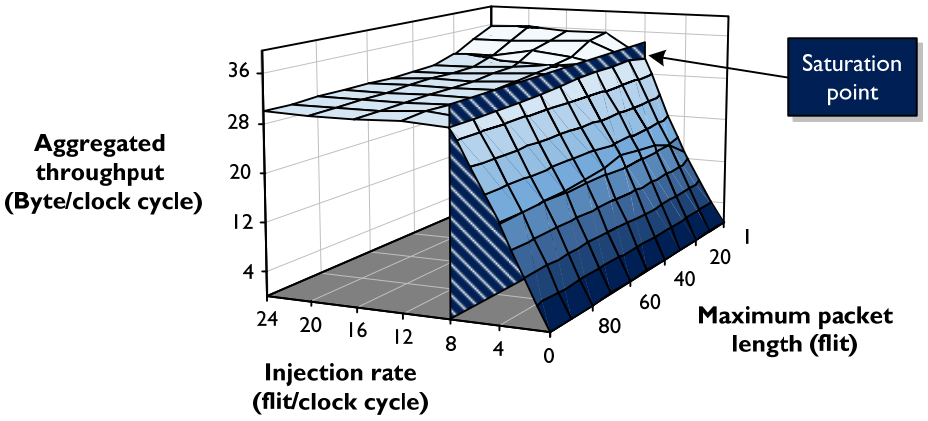

Figure 4-1 : The parameters of packet length and injection rate vary during the operation of an NOC and essentially affect the communication characteristics

$$d_{\max} = \max_{n_s, n_d \in N} d(n_s, n_d) \tag{27}$$

$$\bar{d}_{\text{avg}} = \frac{1}{|N| \cdot (|N| - 1)} \cdot \sum_{n_s, n_d \in N} d(n_s, n_d) \qquad \text{with} \begin{cases} n_s \neq n_d \\ |N| = N_{\text{res}} \end{cases} \tag{28}$$

In addition, two more parameters are introduced that vary during the operation of a communication architecture, and which largely impact system characteristics. These are the **packet length** and the injection rate. The former amounts to the number of flits that belongs to a packet. And the latter, the **injection rate**, specifies the amount of data that is supposed to be injected into the network −strictly speaking, how many flits per clock cycle are offered to all network interfaces. For clarification, figure 4-1 presents the aggregated throughput of a 9x9 mesh network (i.e. with 81 resources) for miscellaneous injection rates and maximum packet lengths, whereas the individual packet lengths are distributed uniformly from one to the determined maximum. Thereby, the throughput only slightly drops for larger packets because two factors counterbalance one another. Namely, the throughput of small packets suffers from the relatively large overhead of control data (e.g. the destination address) in comparison to the useful amount of data in the payload. Then again, large packets span across the entire network and thus block and delay other packets more likely in the attempt to grant access to network resources. In contrast to the packet length, the injection rate affects the throughput significantly, whereas two sections can be distinguished. For small injection rates the throughput progresses approximately proportional, and for large injection rates the throughput is about constant. The transition there from one section to the other is called the saturation point α_{sat} (as sketched in figure 4-1).

Even though the throughput is constant above the saturation point, the network should be operated below this point nonetheless, because the packet delays increase dramatically for large injection rates. For a start, since all flits of a packet are emitted sequentially, the **packet delay** t_{packet} is a function of the serialization delay t_{serial}, or rather the packet length (see

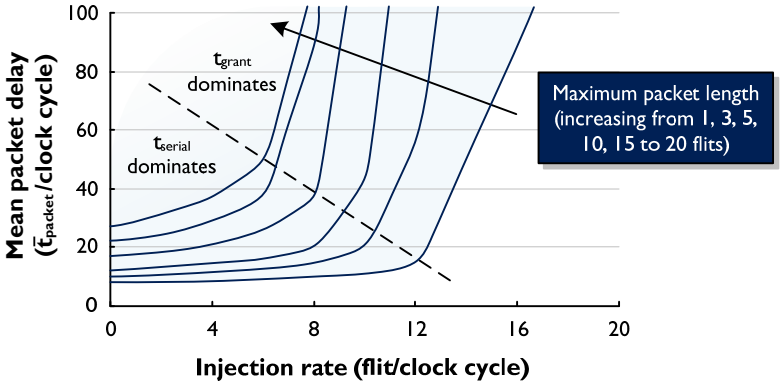

Figure 4-2 : In communication architectures with wormhole switching, large packets span across the entire network, and thus more likely block and delay other packets – which results in the earlier boost of the mean packet delay

equation 29). Besides, the distance in terms of traversed routers d_{router} determines how many router delays t_{router} and contention delays t_{grant} arise – i.e. the delay for processing the packet header, respectively the delay while awaiting grant access to blocked network resources. In the strict sense, link delay actually has to be added to the packet delay as well. However, because of the design decisions used here, this is neglected without loss of generality.

$$t_{packet} = d_{router} \cdot \left(t_{router} + t_{grant} \right) + t_{serial} \qquad \text{with} \begin{cases} t_{serial} \propto \text{Packet length} \\ d_{router} = 1 + d(n_s, n_d) \end{cases} \tag{29}$$

In order to demonstrate its dependencies, the mean packet delay $\overline{t}_{packet}$ is plotted in figure 4-2 for different packet lengths over the injection rate. According to that, the packet delay t_{packet} is dominated by the serialization delay t_{serial} for lower injection rates (i.e. below the dashed line) and is thus closely proportional to the packet length (see equation 29). Above the dashed line though, the packet delay is dominated by the waiting times t_{grant} for blocked network resources. As mentioned before, large packets span across the entire network. Therefore, larger packets exhibit the drastic increase of the packet delay t_{packet} already for lower injection rates, because their waiting times t_{grant} increase disproportionately. In summary, bearing in mind both packet delays and aggregated throughput of the system, rather moderate packet lengths are expedient for the use in unknown or random traffic scenarios. In doing so, such packets should be injected at a rate below the saturation point due to the negative impact on the packet delays.

As a result of the findings so far, an underlying test setup can be determined. The definitions and default values are gathered in table 4-1 and are used persistently in the subsequent investigations. In the event of a deviation from the given setup, such adapted definitions are mentioned in the corresponding sections. Since the parameters have been introduced and motivated in the previous sections, solely a few hints on the simulation are dropped here. Every

Table 4-1 : Collection of default values and definitions as a starting point for the following simulations and investigations

	Parameter	Definition
Architecture	Topology	Planar mesh
	Network size	9x9 = 81 resources
	Addressing	Geographical (lower left = 0,0)
	Address size	8 bit (4 bit per dimension)
Router	Router degree	5 ports
	Flow control	Wormhole with Req/Ack
	Routing	Dimension-ordered (XY)
	Arbitration	Round-robin
	FIFO depth	8 slots
	Data width	64 bit
Link	Type	With repeaters
Simulation	Packet length	1 – 10 flit
	Traffic pattern	Uniform random
	Settling phase	1000 packets
	Number of test cases	20 000 packets
	Injection rate	4.05 flit/clock cycle (≙10 %)

simulation starts with a settling phase of 1000 packets in order to create a realistic initial state of the communication architecture – such as busy links or loaded buffering resources. Subsequently, the examination starts and lasts for 20 000 packets, whereas only these packets are monitored and regarded for the analysis. Concerning the traffic patterns that are used to inject packets, there is a lively discussion in the scientific community [Gre07a, Sal05, Sal07a]. It has been shown that diverse traffic patterns reveal different communication characteristics – like peak performance or worst case throughput [Dal04]. Likewise, traffic patterns also advantage architectures to different degrees, thus affecting design decisions – for instance about the topology or the routing algorithm [Bop93, Pif94, Seo05, Hu04a, Gre07b]. However, neither recognized traffic patterns nor established benchmark suites exist for networks-on-chip to this day [Sal05, Gre07a]. Therefore, a uniform random traffic pattern is primarily used in this work. Such traffic pattern is the most widely used one and an acceptable compromise to demonstrate parameter dependencies [Sal07a, Dal04]. Furthermore, the random scheme is supplemented by other schemes where appropriate reference is necessary (see particularly subsection 4.4.1 and chapter 5).

4.1 Evaluation of routing algorithms

Within a given architecture, routing algorithms compute a path that packets travel on from source node n_s to destination node n_d. Such a routing path R_{path} is an ordered set of links as shown in equation 30 [Dal04]. Hence, a packet from source node n_s enters link l_1 first, traverses

the further links successively (i.e. l_2, l_3, ... , l_{m-1}) and reaches the destination node n_d through link l_m. Consequently, the cardinality of the routing path R_{path} equals the distance in terms of links (or hops) between the two communicating nodes.

$$R_{path} = \{l_1, l_2, l_3, \ldots, l_m\} \qquad \text{with } |R_{path}| = d(n_s, n_d) \quad n_s, n_d \in N \tag{30}$$

Since various paths may exist between a single communication pair (n_s,n_d), a routing algorithm actually performs two different tasks in order to select a unique path. First, a set of feasible paths is identified. Second, from the found set the most appropriate path is chosen and applied to the packet. In doing so, the former task deals with the topology and deadlock avoidance, whereas the latter addresses issues of adaptivity and link utilization. Therefore, routing algorithms are essential in communication architectures, as they determine to what extent the potentials of a given topology can be exhausted. Independent of the topology and the requirements of the systems though, all routing algorithms target the following characteristics:

- Connectivity: At least one routing path R_{path} can be found for any pair of communication participants.
- Adaptivity: Alternative paths are dynamically selected depending on, for example, congestions, link utilization or component failure.
- Reliability: Data communication remains possible in case of faulty components or corrupted packets.
- Communication freedom: Routing paths are selected such that deadlocks and livelocks do not occur in the network.

In order to evaluate different routing algorithms, the following subsection introduces at first a taxonomy of routing schemes. Building on this, selected algorithms are investigated concerning their communication performance in subsection 4.1.2 and their functionality in the presence of failures in subsection 4.1.3, respectively.

4.1.1 Taxonomy of routing schemes

There is a vast number of routing schemes [Ni93, Dua03, Dal04, Ben06] so that a classification helps to distinguish the fundamental properties of such algorithms. Figure 4-3 illustrates a convenient taxonomy of routing algorithms with respect to the location of the routing decision, the type of adaptivity and the communication distance. The meanings of the three main categories and their subgroups are explained in the following.

The category of the **routing decision** names three possible methods where the routing decision for individual packets can actually be made. In distributed schemes, each packet carries the destination address which is extracted by the routers. Therewith, the appropriate output links (respectively ports) are locally determined in the routers by an algorithmic function or by referring to a lookup table [Bol05]. By contrast, source-based techniques do not include the

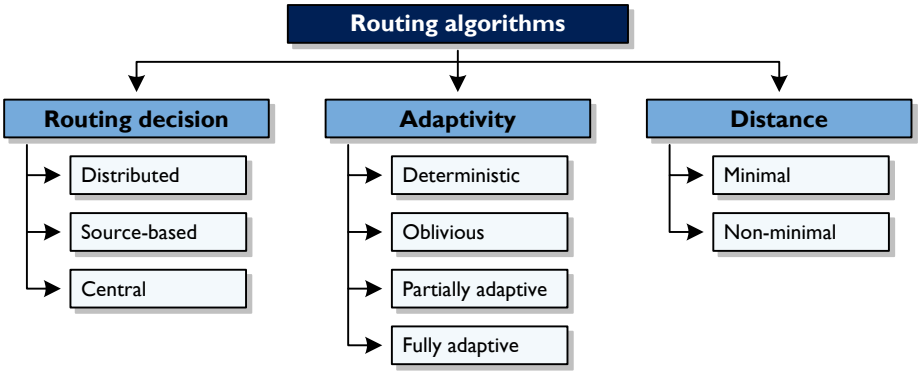

Figure 4-3 : Taxonomy of routing algorithms as regards the location of the routing decision, the type of adaptivity and the communication distance

destination address in the packet, and they calculate the routing path only once as a whole at the source node [Tan02, Bod95]. The packet is then given a sequence of switching directives, whereby each router just extracts the current directive and sets the switch matrix accordingly [Goo05]. Lastly, the definition of central routing decisions is closely connected with the previous two, but with the important difference that all routing paths are known centrally and can thus be adapted with the aid of a global perspective. However, modified routing paths have to be passed to the routers or network interfaces −e.g. in order to update lookup tables− whereupon the transmission itself appears like distributed or source-based routing.

The second category, the **adaptivity** of a routing algorithm, refers to the extent that the network state is considered for selecting among different available routing paths [Dua03]. Those conditions that lead to changing routing paths over time −for the same communication pairs− are generally traffic congestions [Dal04]. In line with the objectives of this work though, component failures are another crucial aspect that routing paths need to be adapted to. As a start, deterministic and oblivious routing schemes do not consider the network state at all. The distinction between these two approaches results from the fact that deterministic schemes always return the same unique routing path for a communication pair [Dal87]. By contrast, oblivious algorithms return different paths that may be randomly picked −such as toggle-XY or Valiant's algorithm [Seo05, Val82]. Beyond that, partially adaptive routing schemes take the network state into account, but partly restricts the potential routing paths −whereas the restrictions may relate to the lengths of the paths or to the allowed changes of direction [Yan89, Gla92a]. Finally, fully adaptive algorithms are not constrained anymore and can thus adapt best to dynamically changing network conditions.

In the end, the **distance** of a path can also be used to group routing algorithms. According to that, minimal routing schemes only allow paths that are no longer than the shortest possible path between the corresponding source and destination nodes. Otherwise, with non-minimal routing, packets may meanwhile depart from their destination as well, which enlarges the distance −i.e. the number of required hops. While this approach expands the alternatives to adjust to the

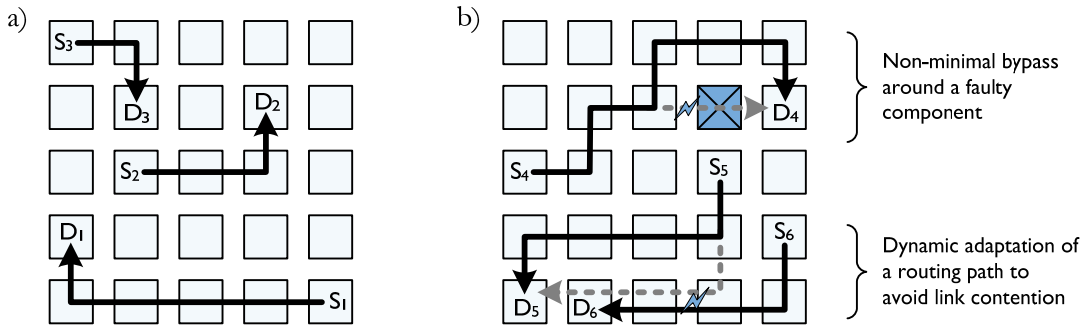

Figure 4-4 : Illustration of two routing examples representing a) a deterministic and minimal routing, namely XY-routing, as well as b) an adaptive and non-minimal routing (Legend: S_x Source, D_x Destination)

current network state, it is more costly from a power perspective for the most part. It should be pointed out that the basic literature on distributed networks often states a couple more categories in order to classify routing algorithms –like the type of progress, the number of receivers or the type of implementation [Dua03, Dal04, Aga09]. However, essentially due to opposite cost containments of on-chip and distributed networks, such additional categories are generally neglected against the background of networks-on-chip –consider for instance the different significance of wires, memory and logic complexity, or topology and component fluctuation.

Finally, two examples are given to clarify the introduced classification. The first one is shown in figure 4-4 a) as an instance of a deterministic and minimal routing algorithm. Hence, the associated routing relation R_{route} always returns the same routing path R_{path} for a specific communication pair from the set of all nodes N, so that no alternative paths have to be evaluated [Dal04]:

$$R_{route} : N \times N \mapsto R_{path} \tag{31}$$

More precisely, figure 4-4 a) displays three different routing paths from the sources S_x to the destinations D_x for the most prevalent routing algorithm in NOCs [Geb09]. This is the dimension-ordered scheme in the form of the distributed XY-routing [Seo05, Sal08]. Thereby, a packet is routed first in the horizontal X-dimension until the column of the destination is reached. Subsequently, the packet is routed in the Y-dimension until the packet is consumed at the destination node D_x. Figure 4-4 b) illustrates the second example, which is a fully adaptive and non-minimal routing algorithm. The appropriate routing relation R_{route} for a distributed implementation results in a set of potential output links (l_a, l_b, ...) for the current router:

$$R_{route} : N \times N \mapsto \{l_a, l_b, \ldots\} \tag{32}$$

Based on the network conditions at present, the most appropriate link is chosen, whereas two essential situations for such a decision are shown in figure 4-4 b). On the one hand, a packet

bypasses a faulty component so that the destination D_4 can be reached. On the other hand, the routing path for the communication pair (S_5, D_5) is adapted in order to avoid link contention with the routing path of (S_6, D_6).

4.1.2 Communication performance

The multitude of published routing algorithms for on-chip networks makes it impossible to compile a complete comparison of their communication performance. Therefore, selected representatives are chosen to demonstrate their impact in principle on frequency, power consumption, area usage and data rate as a first insight. The range of routing algorithms here is geared to the turn-model from Glass and Ni [Gla92a].

Accordingly, the most restricted version that still offers full connectivity is dimension-ordered routing, which only allows four of the eight potential turns in a mesh network. While the general mode of operation was already introduced in the previous subsection and figure 4-4 a), XY-routing is considered both as an algorithmic and a table-based version −denoted XY-Algorithmic and XY-Table [Dal04, Bol04b]. In addition, three versions of partially adaptive routing with minimal distance are also included in the comparison. These three routing algorithms employ six of the eight possible turns in such a way that deadlock freedom is guaranteed [Gla92a]. For instance, in west-first routing no packet may turn into the west direction, which accords to the two prohibited turns. Thus, when a destination is west of the source, the corresponding packet always has to be routed west first, hence the name of such routing. In any other case, the most suited link may be chosen dynamically so that west-first routing can adjust to the network state in parts. Analogously, north-last and negative-first routing algorithms perform in the same manner, but utilize another set of six from the eight turns [Dal87, Yan89, Gla92a].

In the end, two routing schemes are considered that do not constrain any turn −i.e. greedy and source-based routing. However, as all eight turns can be used, deadlocks as well as livelocks may occur due to cyclic dependencies. In order to deal with these events, two things were minded during the implementation. Firstly, both routing schemes were implemented for minimal distance, which prevents livelocks. Secondly, it was assumed that the switching directives for source-based routing are chosen such that deadlock freedom is achieved. In case of greedy routing though, deadlocks cannot completely be avoided so that an additional counter was implemented that drops the entire packet when no progress can be made for a specific time −which is an indication for a deadlock [Lan10, Sof07, Pri08]. Apart from that, greedy routing monitors the links that are attached to the router and forwards packets according to the local link utilizations. Source-based routing is also fully adaptive, but it can only adjust the routing path at the source node, thus before the switching directives are embedded in the packet header (see subsection 4.1.1). Besides, in the domain of distributed networking, there exist many more algorithms with supplemental functionality and flexibility −such as backtracking or spanning tree [Dua03, Tan02]. For the operation in networks-on-chip though, this extra complexity presents an unnecessary expense and does thus not pay off [Ben06].

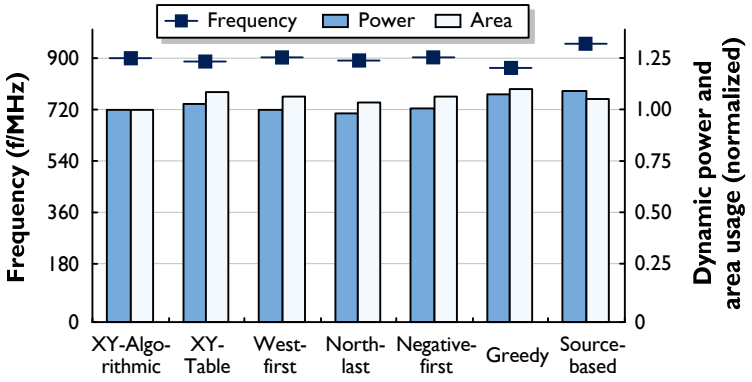

Figure 4-5 : Router characteristics in reference to frequency, dynamic power and area for different routing algorithms – with power and area normalized to 6.89 mW, respectively 26 593 µm²

As a start, the contemplated router implementations were synthesized for a first comparison. Thereby, a data width of 32 bit and a FIFO depth of 4 slots were chosen so as to better reveal the influence of the routing algorithms on the router characteristics. The corresponding results for operating frequency, dynamic power consumption and area usage are depicted in figure 4-5 for the various routing schemes. It stands out that all values are fairly similar, which indicates that the buffering module still plays the decisive role, and not the routing module (see also subsection 3.3.1). However, a few things are still noteworthy. In case of the frequency, solely greedy and source-based routing differ worth mentioning from the reference router (i.e. XY-Algorithmic with 900 MHz). While greedy routing suffers from the additional logic complexity for the comprehensive adaptivity, the frequency of the source-based scheme benefits since no local routing decision needs to be computed. In case of dynamic power and area, the results paint a somewhat diffuse picture. Simply stated, increased complexity yields larger area, and, as long as the frequency is not cut down, higher power consumption – because $P_{\mathrm{dyn}} \propto f \cdot C_{\mathrm{load}}$. By way of example, the area usage of XY-Algorithmic is lower than for all other routing versions, whereas XY-Table and source-based routing are not affected by the logic complexity itself but by the lookup table, respectively by the multiplexers which are required to shift the switching directives in the packet header. The same chain of argument holds also true for the dynamic power consumption, whereas source-based routing is additionally affected by the higher maximum frequency.

As promising synthesis results do not necessarily translate into the best network characteristics, the communication properties of different routing algorithms were also evaluated in simulations of the network architecture. Since the results of similar routing schemes are nearly identical (e.g. west-first, north-last and negative-first), only four distinctive algorithms are cited in the further diagrams. Correspondingly, the aggregated data rate of those selected algorithms is plotted in figure 4-6 a) against the injection rate. Thereby, the course of the graphs increases linearly for all four routing schemes as long as the injection rate is low. The saturation points though discriminate between the further courses. On the one hand, the data rates of XY-Algo-

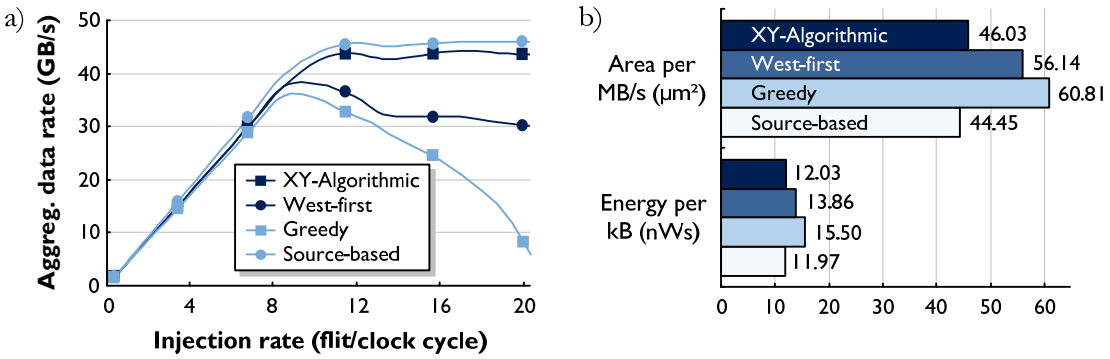

Figure 4-6 : Communication characteristics of selected minimal routing algorithms in an NOC under uniform random traffic: a) Aggregated data rate against the injection rate b) Metrics relating power and area to the available data rate

rithmic and source-based routing saturate later and at a higher level (i.e. at >42 GB/s). It should be noted that a dimension-ordered scheme was employed to set the switching directives of the source-based routing, whereby the course equals that of XY-Algorithmic with a slight advantage due to the higher maximum frequency. On the other hand, greedy and west-first routing not only saturate earlier −here at about 8.5 flits per clock cycle− but also exhibit a diminishing data rate above the saturation points. This is because the local decision making of both routing schemes does not accord to the globally best results [Gla92a, Dal04]. Furthermore, the rising network congestion produces an increasing number of packets to be dropped in case of greedy routing due to the counter used for deadlock avoidance.

In order to relate the communication performance to the power and area costs, figure 4-6 b) presents such figures with reference to the highest individual data rates −i.e. at about the saturation points. Thereby, the area and power data only account for the routers of the simulated architecture. Thus, XY-Algorithmic and source-based routing require the least area for transferring 1 MB/s, whereas west-first and particularly greedy routing are afflicted with the low data rate and the additional area due to the local mechanisms of adaptivity. The same picture emerges for the energy as well, which is also given in figure 4-6 b) for the transmission of one kilobyte. That is to say, greedy routing ranges at the lower end with 28 % energy overhead compared to source-based routing at the top end.

Before the findings on communication performance are summarized, attention should be called to a few more things. The adaptive routing schemes are great examples of how the abstraction layers are intertwined and that they can rarely be improved separately. For instance, when only observing the source-based routers themselves, the generation or storage of the switching directives is not taken into account. Furthermore, since packets may be dropped in case of greedy routing here, additional functionality is necessary in order to make such routing scheme feasible. Corresponding logic needs to handle amongst others the retransmission of dropped packets or the reordering of the packet sequence. Hence, the sole evaluation and refinement of the functionality within an abstraction layer can lead to deceptive results and erroneous design

decisions. Moreover, the given figures rely on a uniform random traffic scheme, which reflects a fairly balanced load distribution in the network. This favors deterministic routing algorithms, since those adaptive schemes can hardly exploit any imbalance of router activities or link utilizations. In fact, most adaptive routing mechanisms outperform deterministic schemes in heterogeneous and unbalanced traffic scenarios [Bop93, Pif94, Seo05, Hu04a, Gre07b].

Concluding, to give a general recommendation for selecting a qualified routing algorithm is difficult, since it depends on a large variety of assumptions. Nonetheless, the applicability of deterministic and adaptive routing schemes can be formulated as a rule of thumb in a first step. Thus, deterministic routing is well suited for the use in small or application-specific systems, because it is simple, efficient and the routing paths can be customized at design time. On the contrary, adaptive routing is more appropriate in order to deal with unknown and heterogeneous traffic patterns that occur in rather large architectures [Ben06]. However, the vast majority of published evaluations on routing in networks-on-chip focus exclusively on communication characteristics. The effects of failures on system functionality though are entirely neglected for the most part. Therefore, the next subsection considers the presence of failures and states necessary precautions.

4.1.3 System functionality in the presence of failures

The prior observations on routing assumed a fully functional and faultless communication architecture. However, this premise cannot be maintained against the background of increasing system complexities and continuously smaller technologies (see also section 2.3). Even though yield, timing issues or soft-errors are acknowledged problems, reliability and robustness is seldom considered in conjunction with routing algorithms.

An analytical model is proposed in [Dal07] for identifying those parts of a router that are most susceptible to temporary failures. Thereby, the paper only targets the model itself and does not name any solution statements. [Fra07] explores the impact of soft-errors and crosstalk, and compares various mechanisms for hardening a router against such issues. Similar studies are also presented in [Kim05a, Ali07] whereas the investigations consider both the links and the routers. Here, reliability improvements against soft-errors are achieved by the application of different error detection/correction schemes. However, permanent failures are not accounted for in all these investigations. Indeed, some proposals also focus on permanent failures and demonstrate improved system functionality in the presence of such failures [Gre07b, Sch07, Kim07, Ala07, Dua03]. This is accomplished by enhanced router designs and fault-tolerant routing algorithms, whereas the impact on communication performance and especially on power consumption is not mentioned. Particularly the increased power dissipation represents a significant drawback for those mechanisms that apply a significant amount of additional logic and messaging – such as in [Kim07] and [Dua03]. Lastly, other methods to simply detect on-chip network failures are motivated by common test procedures, as for instance scan chains or Built-In Self-Test (BIST) [Wan08, Gre06, Akt02, Jan03a]. Their target is generally the highest possible failure

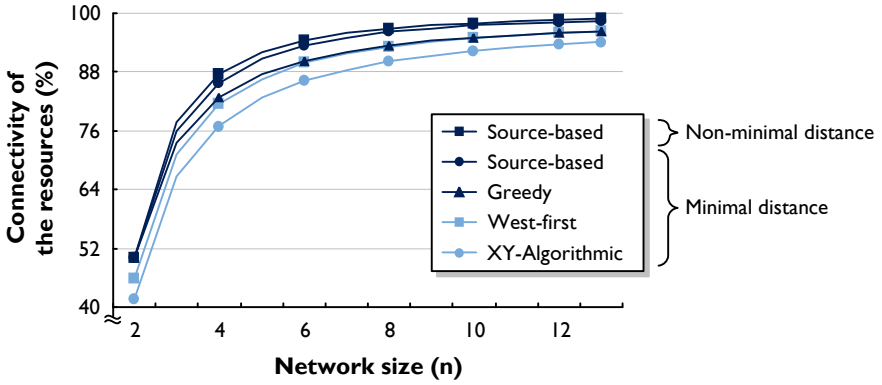

Figure 4-7 : Influence of different routing algorithms on the accessibility among the resources in the presence of a single faulty router

coverage, though evaluations of underlying routing algorithms or the impact on communication performance are only a secondary aspect. As a result, permanent failures and their impact on system functionality are investigated in the following for different routing algorithms. The objective is to deliver new insights into the trade-off between communication performance, power consumption and reliability.

For that purpose, a single router within the previously implemented architecture was defined as being struck by a benign failure. Based on this setup, system functionality was surveyed for different routing algorithms and network sizes as well as for uniformly distributed locations of the failure. The obtained results for the average system functionality are plotted in figure 4-7 against the network size for those selected algorithms of the prior subsection. Additionally, source-based routing was also implemented in a non-minimal way. System functionality in figure 4-7 is referred to in the form of the accessibility among the resources. For example, a connectivity of 80 % denotes that 192 connections of those 240 possible communication pairs in a 4x4 network architecture are on average still intact in the presence of one faulty router. According to that, all algorithms exhibit a similar course. More precisely, the percentage of working connections increases for larger network sizes and eventually approaches full connectivity (i.e. 100 %). This is because of the boosting number of routers and connections (i.e. $N_{res} \cdot (N_{res} - 1)$ with $N_{res} = n^2$) in contrast to the constant single failure. Thereby, the ranking reflects the degree of adaptivity with deterministic XY-routing at the lower end followed by west-first and greedy routing as examples of partial and full adaptivity. However, since source-based routing can also incorporate global network knowledge, both the minimal and the non-minimal schemes outperform the others. The advances of the non-minimal scheme are bought by a distance penalty though.

Since the probability of failures will increase, it is likely that several permanent failures compromise system functionality. Such a scenario comprehends initial failures as well as permanent failures over time and impacts miscellaneous aspects – as for instance reliability, yield or graceful degradation (see also subsection 2.3.2). Therefore, figure 4-8 displays the findings of

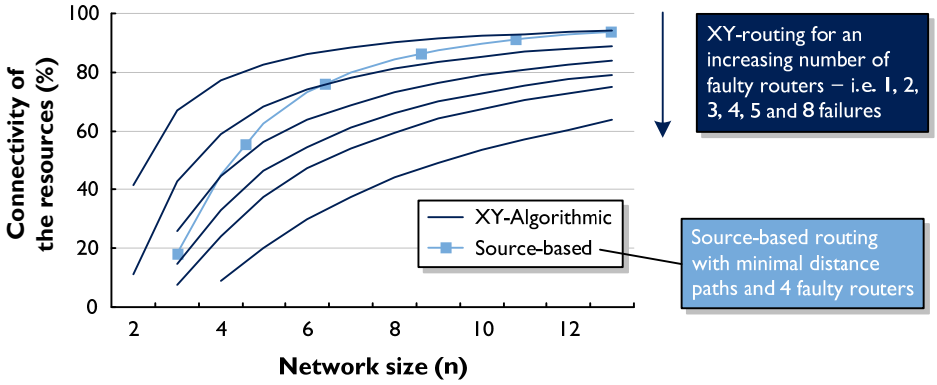

Figure 4-8 : Presentation of deterministic XY-routing and adaptive source-based routing by means of the connectivity of the resources for different numbers of faulty routers

the same scenario as before but with diverse numbers of randomly distributed permanent failures. For the purpose of clarity, the selection of routing algorithms is reduced to XY-Algorithmic and source-based routing, both implemented with minimal distance. As a start, the higher number of failures reduces the connectivity significantly, which is equivalent with decreased reliability and system functionality, respectively. By way of example, the connectivity of a 9x9 network architecture with XY-routing boils down from over 91 % in the presence of one failure to 49 % in the presence of eight failures. A similar dependency can also be observed for other routing schemes, but with a less severe impact. In order to determine this difference, source-based routing with four failures is also plotted in figure 4-8 exemplarily. From the perspective of connectivity, the usable adaptivity makes source-based routing superior to deterministic XY-routing, even when suffering from more failures. For instance, source-based routing with four failures outperforms XY-routing with just two failures already for a network size larger than 36 resources (i.e. network size $n = 6$).

A final evaluation of routing algorithms still necessitates the consideration of further aspects that may even reside in other abstraction layers. Indeed, the communication network itself can be made more robust in order to prevent or at least delay the occurrence of certain failures – for instance, by conservative physical layout, hardened registers or robust state machines. However, since it is unrealistic to assume that complex communication architectures are entirely faultless in the long run, both failure detection and resolution mechanisms have to be added in any case. Consider for example that a message dependency [Lan10] or a corrupted packet header may lead to deadlock regardless of whether the used routing algorithm is actually deadlock-free. Similarly, a faulty network component may jam a specific data stream and consequently stall the entire network communication.

For such reasons, a reliable on-chip network requires in the strict sense four additional mechanisms: detection, resolution, recovery and adaptation [Mur09, Lan10]. The simplest

Table 4-2 : Qualitative summarization of the evaluations in this section for different routing schemes (Legend: + Good, ○ Neutral, − Bad)

	Data rate	Dynamic power	Area usage	Reliability	Additional overhead
XY-routing (deterministic)	+	+	+	−	+
West-First (partially adaptive)	○	+	○	○	+
Greedy (fully adaptive)	−	○	○	○	−
Source-based (fully adaptive)	+	○	○	+	−

solutions for failure detection are counters, which can be used to detect deadlocks and livelocks −by means of counting the waiting time of a packet, respectively the number of undergone hops [Ala07]. Other malicious failures are more difficult to detect and demand continuous consistency and self-tests [Akt02, Gre06, Wan08]. In any case, once a failure is detected, the deficient network condition needs to be resolved in order to be able to transfer data again. Appropriate approaches range from simply dropping specific packets to redundant networks, floating buffers and resetting network components, [Ni93, Anj95, Kon91, Nic06]. Albeit the network is functional after the resolution of a failure, the communication state has to be recovered, which can be accomplished by such mechanisms as retransmitting lost packets or the rollback of a communication protocol. Lastly, depending on the type of failure, the network has to adapt itself so as to avoid that the same faulty condition happens again. For example, changing routing paths to permanently bypass faulty routers. A promising approach that takes on the four just stated mechanisms to obtain reliable NOCs is the novel, multistage approach of the Error Resiliency Layer (ERL), which is presented in subsection 4.4.2.

Concluding, adaptive routing algorithms pay off in respect of reliable communication, and rank according to their extent of flexibility to adapt to faulty conditions in the network. However, adaptivity as well as reliability is commonly traded off against performance and power consumption, respectively area usage −see table 4-2 with a brief comparison. Furthermore, routing schemes partially require additional design overhead outside the actual network (e.g. the storage of switching directives). This fact is also qualitatively reflected in the last column of table 4-2. Finally, this thesis suggests using source-based routing because the communication network itself is not stressed with additional logic and unavoidable failures in complex systems can be accounted for. Certainly, reliable communication architectures necessitate in fact further functionality to cope with failures, which is often considered an issue of higher abstraction layers or entirely disregarded. Hence, it requires a combined and comprehensive approach across all abstraction layers to facilitate both a reliable and efficient communication architecture.

4.2 Heterogeneous distribution of packet FIFOs

The previous section investigated amongst others different routing algorithms and their extent of adaptivity to adjust communication characteristics at run-time. One of the resulting advantages is that routing paths can dynamically be customized during system operation in order to balance workloads and to avoid congestions as well as performance losses. Another approach at run-time within the on-chip network is the dynamic allocation of buffering resources within routers and links. In the former case, floating buffers within the routers are allocated to those ports that require additional storage for avoiding or diminishing congestions of data streams [Anj95, Kon91, Nic06]. In the latter case, the basic operation of links can adaptively be enhanced with dedicated link buffers. Thus, such designed links can operate in a pipelined fashion at peak times [Kod07]. However, besides their potential advantages, the aforementioned approaches require additional on-chip overhead for the monitoring of network characteristics and the dynamic adaptation.

Hence, several other published works aim at static adaptations during the design time, which obviates the on-chip overhead, although at the risk of increased engineering efforts. Corresponding approaches adapt the design to the demands of the system and its applications that are known in advance –as for instance to stressed traffic paths or to the occurrence of data rate and delay requirements. Consequently, NOC architectures can be customized for a particular application scenario so as to trade off system characteristics [Ben02, Bol04b]. As an initial example, xPipesCompiler and Sunfloor are two academic design tools that instantiate application-specific NOC architectures on the basis of parametrizable fundamental building blocks (e.g. links and routers) [Jal04, Mur09]. However, application-specific enhancements can be carried out in all other abstraction layers as well. For instance, it was shown that the application mapping onto a given architecture can yield improvements when the local proximity of communication graphs is exploited [Hu03a, Mur04]. By contrast, three further design techniques alter the network itself depending on the utilization of individual network components. On the one hand, long range links are used to supplement NOC architectures with additional links in order to relieve congested areas of the network [Ogr05, Ogr06b, Ogr06a]. On the other hand, both links and routers can also be customized to better serve the actual traffic loads [Guz06, Hu04b]. Consider that corresponding changes –e.g. of the data widths, the frequencies or of the amount of buffering– leave the topology unchanged.

However, on the basis of the orientation of this work towards complex integrated systems, application-specific knowledge is hardly available for network improvements. The published papers discussed above are thus only of very limited feasibility for the use in both domain-specific and general purpose platforms. Therefore, the following subsections derive promising improvements for the heterogeneous distribution of packet FIFOs that are independent of particular applications as far as possible.

4.2.1 FIFO depth based on utilization

There are generally three starting points for optimizations within a communication architecture that seem promising without application-specific knowledge. These are first and foremost the topology as well as the physical placement of resources, which together define the foundation of the architecture. While the impact of the topology is investigated in section 4.3, the physical placement of resources is not a focal point of this work −note the introduction of the reference scenario at the beginning of chapter 4. The third option is the routing algorithms, which are connected to the underlying architecture and which significantly impact communication properties (see section 4.1). Accordingly, routing algorithms within a mesh-based architecture are analyzed in the following in order to exploit their characteristics for improvements during the design time −in particular, for the distribution of FIFOs.

The preceding simulations on routing algorithms have indicated that concentrated traffic arises primarily in the center of the network. The common reason is the try to set up minimal paths between sender and receiver of a message −in order to keep communication distances small, and thus to improve power and performance. Therefore, most connections lead somehow through the center of the network. This is the cause for the concentrated traffic, which is also a critical trigger for the origination of packet congestions [Nil03]. Figure 4-9 a) illustrates the distribution of traffic based on the normalized **router activity** across the introduced network architecture (consider table 4-1). The parameter of activity here is to be understood as the number of times that a flit is dealt with. For example, ten flits traversing a router denote an activity of ten for that specific router. According to that, the traffic distribution pictures a hemispherical shape with its maximum in the center −i.e. at the pair of coordinates (x,y) = (4,4). From there, the router activity drops down in all directions and reaches its minima in the corners of the network at about 30 % of the highest activity. Even though the results in figure 4-9 a) relate to the most common routing algorithm (i.e. XY-routing), this observation applies similarly to most other routing algorithms as well [Sof07].

Since the implemented traffic pattern has significant influence on communication characteristics, several other patterns than the uniform random case were also contemplated. One such result is plotted in figure 4-9 c) that is based on the same architecture but applies a local traffic scheme. More precisely, the local traffic scheme is based on a Gaussian distribution function so that neighboring resources of a sender are more likely to receive packets than in the uniform case. Hence, the average communication distance $\overline{d}_{avg}$ decreases drastically from 6 hops for the uniform case down to 2.6 hops for the local pattern. However, the hemispherical shape of the router activity still persists, although with a shallower course towards the borders of the network. Generally speaking, concentrated traffic in the center of the network originates for most traffic patterns when the resources take about evenly part in the communication. Further details and results on local traffic patterns are considered in subsection 4.4.1.

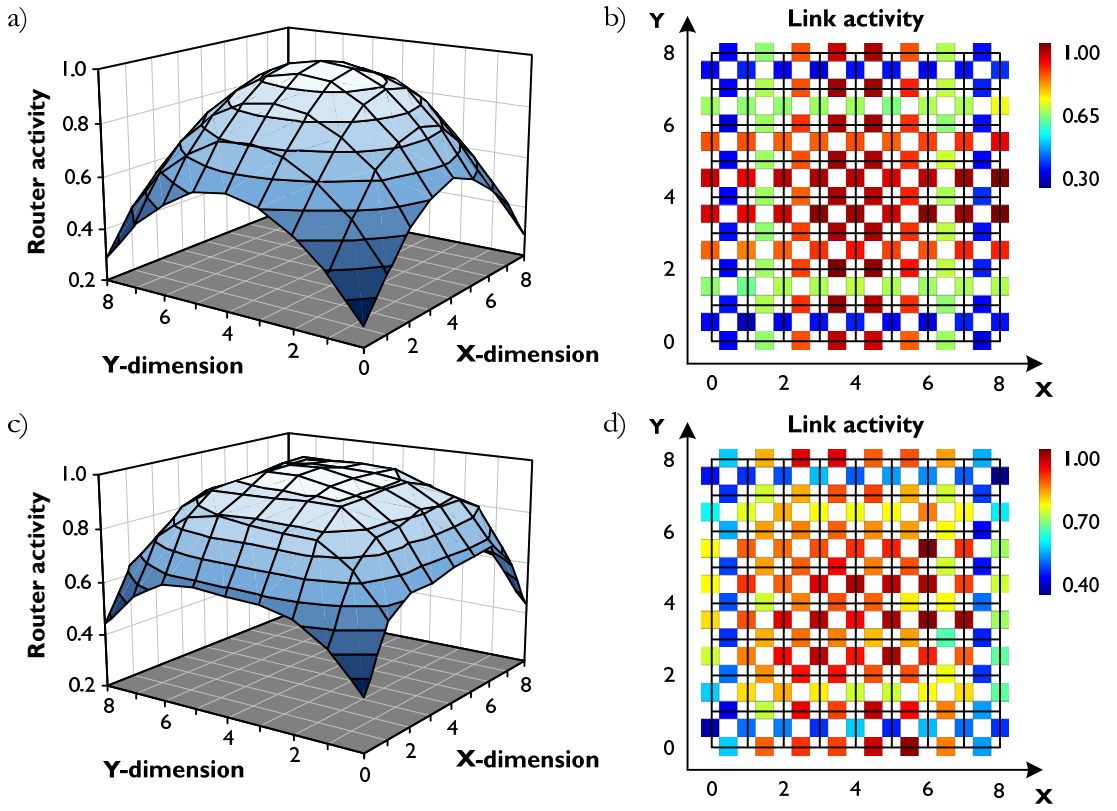

Figure 4-9 : Depiction of the normalized network activity with XY-routing: a) Router and b) link activity under uniform traffic as well as c) router and b) link activity under local traffic (with a Gaussian distribution)

An analog presentation of the normalized activity is shown in figure 4-9 b) and d) for the links of the communication network. As the diagrams comprehend both horizontal and vertical links, auxiliary lines help to allocate the results to the different links. Hence, vertical links are pictured along the vertical lines and horizontal links on the horizontal lines. Accordingly, the routers connecting all links would be located on the intersections of the auxiliary lines. Even though the results of the link activities exhibit a more unequal course compared to the router activities, they also reflect the concentrated traffic in the center of the network – for both cases of uniform and local traffic, see figure 4-9 b) and d). In summary, the activity of routers and links varies across the network and is highest in the center. This observation holds true for most routing algorithms and sundry traffic scenarios.

The observation of the centered traffic implicates that congestions also most likely occur in the center of the network. Besides, subsection 3.3.3 revealed that larger FIFOs in the routers can help to resolve such network congestions. Therefore, this section strives for a new **heterogeneous distribution** of the packet FIFOs of the routers so that areas with heavy traffic loads benefit from larger FIFO depths. By contrast, areas with little network activity are equipped with smaller FIFOs in order to reduce the total number of FIFO slots, and thus the overall

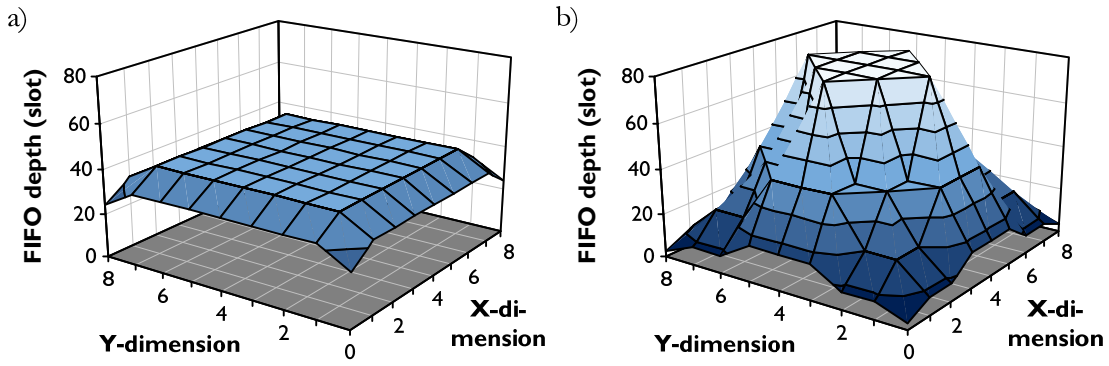

Figure 4-10 : Two examples of FIFO distributions across the network routers: a) Uniform FIFO depth b) Heterogeneous FIFO depth based on the activity of each router (granularity 1, 2, 4, 8 and 16 slots)

power consumption. Starting point for the heterogeneous FIFO distributions in the following is the default architecture, which comprehends uniform ports, each with a FIFO depth of eight slots. This uniform FIFO distribution across the network routers is displayed in figure 4-10 a), whereas the scale of the FIFO depth relates to all five ports of a particular router −hence, 5 ports with 8 slots each equals 40 slots per router. Consider that the decline at the border of the network is because such routers consist of less than 5 ports.

As opposed to the uniform distribution, figure 4-10 b) depicts a heterogeneous FIFO distribution that is based on the router activity as given in figure 4-9 a) −this FIFO distribution is referred to as 'Router activity' in the subsequent discussion. Here, the used FIFO sizes range from 1, 2, 4 and 8 slots to 16 slots at most. By way of example, the routers in the corners carry out the least activity and thus have FIFOs with just 1 slot, whereas the routers in the center have FIFOs with 16 slots in each port. Thereby, a distribution of FIFO depths evolves that represents the router activities of figure 4-9 a). However, since the distribution of link activities differs from the distribution of router activities, another distribution of FIFO depths was implemented that correlates with the link activities in figure 4-9 b). More precisely, in this case all FIFOs of a single router have the same depth that is calculated from the average activity of the adjacent links −referred to as 'Avg. link activity' in the following. Admittedly, a precise examination of the links in figure 4-9 b) reveals that the activities of the links connected to a certain router can vary significantly. Therefore, two further distributions were realized where the FIFO depth of a specific port only depends on the activity of the attached link −referred to as 'Link activity' and 'Min. resource'. Therewith, a single router can have FIFOs different in size, which is in contrast to the preceding distributions where all ports of a specific router are equally large. The distinction of the last two distributions is based on unequal threshold values for the FIFO selection and a different calculation of FIFO depths for the resource port of the routers. In case of the 'Link activity', the FIFO depth of the resource port is computed from the average FIFO depth of the remaining router ports. By contrast, all resource ports of 'Min. resource' feature FIFOs with just one slot.

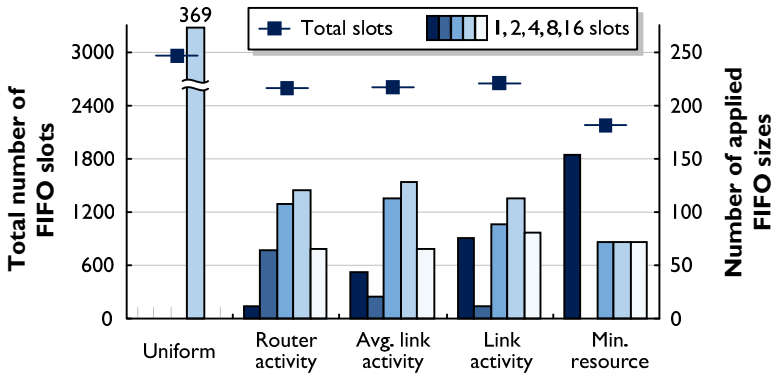

Figure 4-11 : Illustration of the total number of FIFO slots across the network for the different scenarios of FIFO distributions as well as the breakdown by their individual sizes

4.2.2 System characteristics for different FIFO distributions

The previous subsection described the motivation and the different implementations of heterogeneous FIFO distributions. It follows the evaluation of system characteristics when those various distributions are applied. As a start, the static impact on the communication architecture is discussed, which primarily concerns the area and implementation costs. Thus, the total number of FIFO slots in the network is depicted in figure 4-11 on the primary ordinate across the proposed distributions. It should be noted that the total number of slots not only depends on the type of distribution but also on the specified thresholds for selecting the diverse FIFO sizes. The threshold values here were chosen such that the overall number of slots is reduced – which aims at reducing the overall power consumption accordingly. In statistical terms, the heterogeneous FIFO distributions comprehend at least 11 % less FIFO slots in the communication network compared to the conventional uniform case – whereby 100 % conforms to the 2952 FIFO slots of the uniform distribution (i.e. 369 ports times 8 slots). Because of the minimum FIFO depth at the resource ports, 'Min. Resource' even decreases the total number of slots by almost 27 %.

Moreover, the second ordinate of figure 4-11 states the allocation of FIFO sizes for each implemented scenario. Even though the underlying scales of activity (e.g. the router activity) for selecting the FIFOs were divided equidistantly, the resulting numbers of applied FIFO sizes are non-uniform within the individual scenarios. For instance, 'Router activity' applies 108 and 120 FIFOs with 4 slots, respectively 8 slots. In addition, about 65 FIFOs with 2 and 16 slots each are used as well as 12 single-buffered FIFOs. In the strict sense, the variety of applied FIFO sizes implies a certain overhead in the design library. However, FIFOs are standard components in almost all design libraries and are thus available in a great variety of sizes and design styles. Hence, the design overhead for the application in heterogeneous FIFO distributions is considered insignificant.

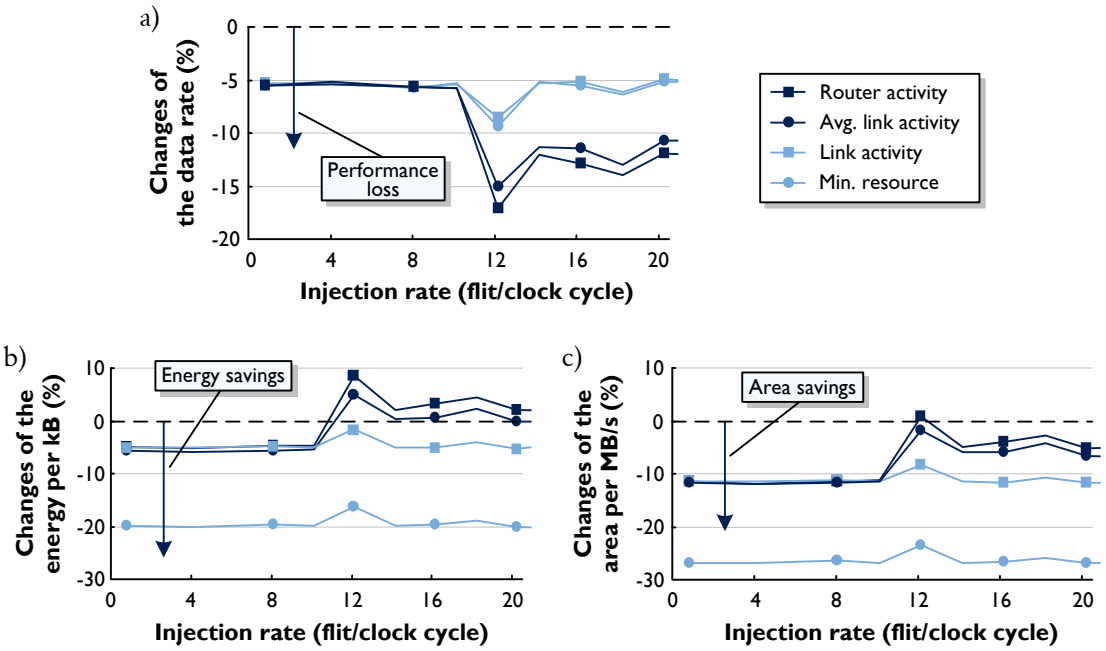

Figure 4-12 : Plot of the changes due to the heterogeneous FIFO distributions in relation to the conventional uniform implementation: a) Data rate b) Energy per kB and c) Area per MB/s

Since the architectural properties are beneficial due to the reduced number of FIFO slots, it is worthwhile to take a closer look at the communication characteristics of the heterogeneous FIFO distributions. The results thereto are presented in figure 4-12 a) to c) in terms of the data rate, the energy per transfer of one kilobyte and the area requirements needed to achieve 1 MB/s. These data are plotted in relation to the conventional uniform FIFO distribution and against the injection rate. It is important to consider that negative changes of the data rate in figure 4-12 a) stand for an undesirable performance loss. By contrast, negative changes of both energy and area metric denote advantageous savings while the performance is taken into account as well.

When considering the data rate in figure 4-12 a) first, a roughly constant performance loss of 5.4 % can be recognized for all heterogeneous distributions up to an injection rate of approximately 10 flits per clock cycle. This is because the heterogeneous distributions apply routers with FIFOs of up to 16 slots depth, which are slower compared to the routers with just 8 slots FIFO depth as used in the uniform reference scenario. To be precise, a maximum operating frequency of 749 MHz for the 8 slot routers faces a maximum frequency of 709 MHz for the 16 slot routers (see also subsection 3.3.3). Apart from that, all distributions suffer significantly around the saturation point at 12 flit/clock cycle, which is an indication of the reduced overall buffering capacity in the network that is missing in order to compensate for temporal congestions. However, the diverse FIFO distributions differ from one another by the degree of their performance slump and by their capability to recover. On the one hand, the data rates of 'Router activity' and 'Avg. link activity' suffer the most and recover only slightly to a

performance loss of about 12 % above the saturation point. Thereby, these two distributions are characterized by the fact that all ports within a single router have the same FIFO depth. On the other hand, both 'Link activity' and 'Min. Resource' feature different FIFO depths even within a single router. In effect, this fact reproduces the actual network activity better, and leads to a favorable progress of the data rate. Accordingly, the decline around the saturation point is rather small (i.e. roughly 9 %) and the data rate is actually fully restored to the initial level of circa 5 % – which is solely due to the frequency penalty.

Since the data rate alone is no explicit measure of communication characteristics, power consumption and area requirements need to be considered likewise. Correspondingly, changes of the energy in relation to the uniform scenario are plotted in figure 4-12 b). It appears that all heterogeneous FIFO distributions offer significant power savings that resemble the reduced numbers of total FIFO slots in the network – recall the cutback in buffering capacity as stated in figure 4-11. However, the energy metric, as also given in the figure, incorporates the data rate as well. Therefore, only 'Link activity' and 'Min. resource' can preserve energy savings across the entire range of injection rates, whereas 'Min. resource' obtains a considerable decrease of about 20 %. In contrast, both 'Router activity' and 'Avg. link activity' exhibit a performance as well as an energy penalty above the saturation point.

Furthermore, figure 4-12 c) reveals a similar characteristic for the area requirements – which are needed to transfer 1 MB/s – as just seen for the energy. The main difference is the initial area demands of the different FIFO distributions. In fact, the physical synthesis yields even higher area savings than the total numbers of FIFO slots suggest. For this reason, the results of all heterogeneous FIFO distributions in figure 4-12 c) show area savings across the whole range of injection rates – neglecting the single outlier –, although the performance penalty is already accounted for in the presented area metric. Here, it is 'Min. resource' again that stands out with savings of up to 27 %. To a great extent, the striking improvements of 'Min. resource' are based upon the small FIFOs of the resource ports. Such an implementation does not limit the general applicability, especially when common assumptions are valid. Firstly, the resources are capable to process data faster than what the network can deliver [Mil07]. And secondly, local memory of the resources can additionally be used to buffer packets in case of delays and congestions [Hu04b].

Concluding, the proposed heterogeneous FIFO distributions are another example for essential design parameters that are interlocked – i.e. performance, power and area at this point. However, the changes here occur with a better ratio for power and area so that the achieved savings actually improve the stated energy and area metrics considerably. Moreover, such architectural changes are also an example that the abstraction layers are intertwined too. For instance, if architecture and network state are known to the system management, applications can be mapped such that their communication favors those routers with small FIFOs. This in turn enables a power reduction without sacrificing performance. Lastly, the introduced idea of a heterogeneous FIFO distribution was demonstrated by means of a mesh topology and XY-routing. The general approach though can be applied to all kinds of topologies and routing schemes, and can also aim at either power (as done here) or performance improvements. Prospectively, the application of

more fine-grained FIFO sizes and the distinction of up and downstream of a single link seem further options for advancements because the fine-grained FIFO distributions ('Link activity' and 'Min. resource') offer the best system characteristics. Lastly, based on the beneficial results, it also seems interesting to apply the proposed approach for a similar customization of the links.

4.3 Clustered topologies for cost savings

The preceding investigations were all referring to the two-dimensional mesh, because it is the most widespread topology [Sal07a, Geb09]. However, since the communication network has major impact on the design costs of the overall system, it is expedient to consider other topological alternatives as well [Hos07, Lee04, Geb09]. From a design perspective, performance, power consumption and area usage are the primary parameters of interest in such an investigation. Whereas the performance of a communication network is mostly expressed in terms of data rate and packet delay, but also in terms of communication distance and utilization. Even though most published works on NOC topologies allow for the mentioned parameters, they generally assume faultless operation and neglect the consequences of permanent failures. Therefore, common design parameters as well as reliability are considered in the following when advanced clustered topologies are presented.

Analytical descriptions of topologies in the literature generally employ additional parameters that are helpful to evaluate and classify diverse topologies. Hence, corresponding metrics are introduced for a start. Such a first-order model is the **Core-to-Router Ratio** (CRR), which serves as a simple indication of power and area requirements:

$$\mathrm{CRR} = \frac{N_{\mathrm{res}}}{N_{\mathrm{router}}} \qquad \text{with} \quad N_{\mathrm{res}} = |N| \tag{33}$$

The CRR is calculated from the ratio of the number of resources N_{res} (i.e. the cores) divided by the number of routers N_{router}. In case of the standard mesh, the CRR equals one because every resource is connected to its own router. Generally, a large CRR is desirable from a power and area perspective since this means that many computational resources are interconnected by a small number of routers. The complexity of the individual routers though is not accounted for.

To compare diverse topologies as regards communication performance necessitates a more sophisticated point of view. Thus, the set of all communication nodes N is partitioned into two disjoint sets N_1 and N_2. The **cut of the network** $C(N_1,N_2)$ determines this type of partitioning because it represents the set of all links that start within N_1 and end within N_2, or vice versa. Whereby, the cardinality $|C(N_1,N_2)|$ conforms to the number of links in the cut −i.e. $|C(N_1,N_2)|$ equals m in equation 34.

$$C(N_1, N_2) = \{l_1, l_2, \ldots, l_{\mathrm{m}}\} \qquad \text{with} \quad N_1 \cap N_2 = \varnothing \text{ and } N_1 \cup N_2 = N \tag{34}$$

Based on the definition of a cut, the **bisection bandwidth** B_B can be calculated, which provides a basic measure of communication performance (see equation 35). Here, a bisection is a special kind of cut that partitions the network into two equally large segments – strictly speaking $|N_2| \leq |N_1| \leq |N_2| + 1$. The bandwidth of such a bisection is given by the sum of all link bandwidths within the cut, whereas B_{L_i} denotes the bandwidth of the i-th link and relates to a bidirectional link. Accordingly, the bisection bandwidth B_B is the minimum bandwidth over all bisections of the network (see equation 35). In case that the link bandwidths are equal, the calculation of the bisection bandwidth B_B simplifies to the product of the link bandwidth B_L and the bisection link count B_c. Strictly speaking, the bisection link count B_c in equation 36 corresponds to the number of links $|C(N_1,N_2)|$ in the smallest bisection.

$$B_B = \min_{\text{bisections}} \sum_{l \in C(N_1,N_2)} B_{L_i} \quad \text{with} \quad \begin{cases} B_L = 2 \cdot f \cdot W_{\text{data}} \\ |N_2| \leq |N_1| \leq |N_2| + 1 \end{cases} \tag{35}$$

$$\Rightarrow \quad B_B = B_L \cdot B_c \quad \text{with} \quad \begin{cases} B_L = B_{L_1} = B_{L_2} = \ldots = \text{constant} \\ B_c = \min_{\text{bisections}} |C(N_1,N_2)| \end{cases} \tag{36}$$

Since equation 35 and 36 refer to idealized assumptions, practical results are generally smaller [Dal04, Dua03]. Nonetheless, the bisection bandwidth B_B marks a useful measure for the comparison of miscellaneous topologies.

4.3.1 Setup of advanced topologies

Since not only the number of routers is decisive in a network but also their complexity, individual routers are investigated before different topologies are presented. Mesh-based topologies and their variants commonly base on the same type of router within an architecture – for instance, flow control and routing scheme are consistent within the network. The routers though still differ in terms of one essential parameter, the router degree. Consider for example the routers in the corners of a mesh topology (with only three ports) in contrast to the routers in the center (with five ports).

Therefore, figure 4-13 relates frequency, power consumption and area usage against the router degree – whereas the figures in this section refer to 32 bit data width and 4 slots FIFO depth. Put simply, all three parameters change about linearly with the number of router ports, which is because of the dominating FIFOs and the amount of required control logic (see subsection 3.3.1). While power and area demands rise with each additional port though, the frequency decreases significantly. In fact, smaller operating frequencies attenuate the growth in power dissipation (see equation 8). However, they also directly entail degraded communication performance – such as reduced aggregated data rates or increased packet delays (see also equation 20 and 29). Besides, smaller router degrees are also reported to be better suited for fault-tolerant topologies [Leh07]. Accordingly, higher router degrees are critical for communication characteristics and should be avoided [Ferr08].

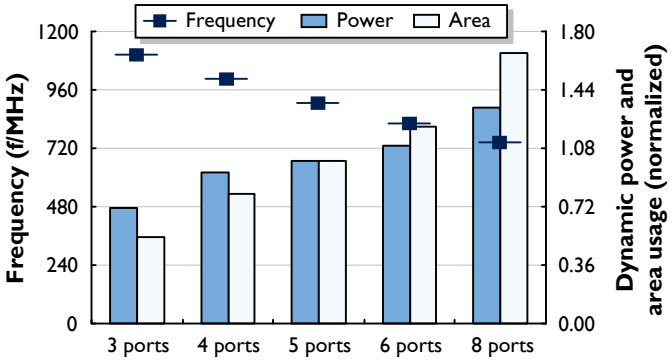

Figure 4-13 : Impact of the router degree (i.e. the number of ports) on the design properties after synthesis −with power and area normalized to the 5 port version with 6.89 mW, respectively 26 593 µm²

With those introductory remarks on the router degree being said, present publications on topological modifications −that emanate from conventional mesh-based networks− are discussed in the following. Express channels are such a modification. They extend regular mesh networks with additional links that connect distant routers, instead of only neighboring ones [Dal91]. The same idea is basically also gone after in such works on the insertion of long range links [Ogr05, Ogr06a, Ogr06b]. The difference is that express channels describe a general modified topology and long range links are applied for application-specific customization. However, both approaches suffer from topological irregularity, which in turn results in timing concerns and the need for higher router degrees.

In contrast to the previous two techniques that keep the number of routers unchanged (CRR = 1), another group of publications aims at reducing network overhead in consequence of fewer routers (i.e. CRR > 1). For instance, concentrated meshes connect not only one but several resources to a single router. Hence, the CRR is larger than or equal to two (CRR ≥ 2) while the slightly more complex addressing scheme is still negligible [Bal06b, Gil08, Kum09]. Reducing the number of routers further, finally results in hierarchical or common star topologies −whereas in the latter case only one router connects all resources −i.e. CRR = N_{res} / 1 [Lee06]. In fact, all these approaches yield smaller communication distances and less area needs, though at the price of larger router degrees. Therefore, [Ferr08] proposed to insert split interfaces at the resources in order to keep the routers simple −i.e. with no more than five ports. Thus, several resources can be connected to a single router port. However, this approach only shifts complexity from the routers into the network interfaces and necessitates elaborate sharing and arbitration of the resource needs in the split interfaces.

A set of works also investigated the impact of topologies with more routers than resources (i.e. CRR < 1), or rather, more ports than in the conventional mesh. Here, [Par06] and [Leh07] both assume that the resources possess more than one network interface to send and receive packets. The difference in their approaches is that [Par06] employs the emerging redundancy to achieve better performance, whereas [Leh07] focuses on reliability improvements. Moreover,

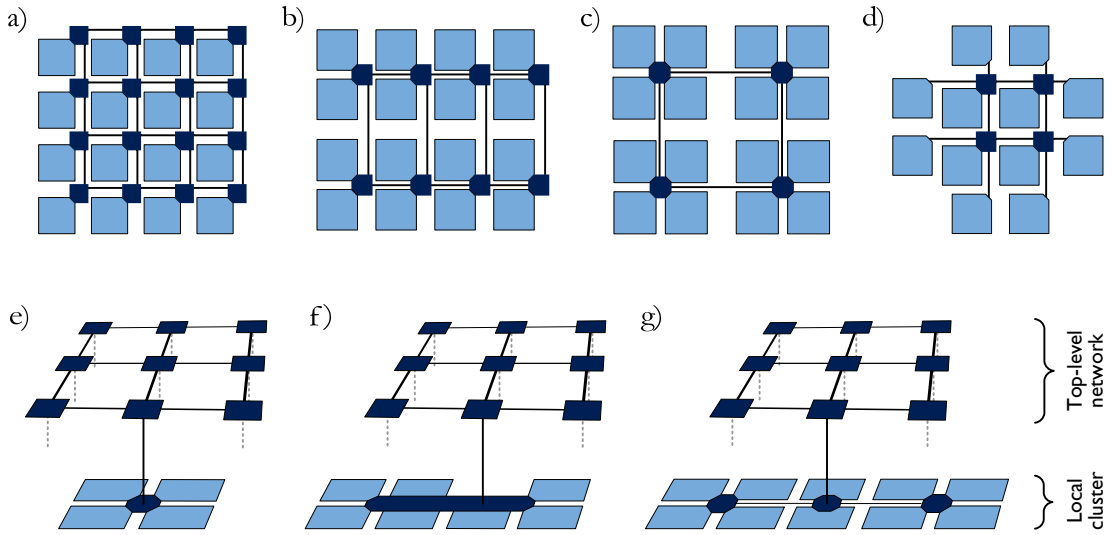

Figure 4-14 : Setup of different topologies: a)-c) Mesh (with CRR = 1, 2 and 4) d) BEAM e)-g) various advanced, clustered topologies referred to as Cluster (1-5), Cluster (1-8) and Cluster (3-5) (Legend: ■ Resources, ■ Routers)

partially and fully redundant networks were also suggested, whereas the double-Y net is one of the better known examples [Sey98, Dua03, Gla92b]. Summarizing the last set of works, the redundancy of the diverse approaches improves performance and reliability, but is traded for undesirable power and area costs. Finally, several published works apply hierarchical concepts as known from distributed networking. Such concepts are presented in [Put07] and [Bou07] as hybrid topologies. That means that a conventional mesh is combined with bus-based communication, respectively with a ring interconnect. A generalization of the same approach is described in [Lan09], where the types of communication schemes in the hierarchy are selected in dependence on the application. Thereby, communication-centric resources are prioritized during the setup of the topology in order to improve system characteristics.

Concluding, none of the mentioned publications considers design costs, communication characteristics as well as reliability jointly, and fully exploits the implicit characteristics of networks-on-chip. Therefore, the proposed topologies in this section aim at balancing all parameters appropriately to obtain eligible communication properties and reduced hardware overhead. Starting point is the idea of clustered topologies that interconnect the resources and communication modules in a hierarchical manner. Besides, it is aimed at keeping the complexity of the network components and protocols as small as possible. Hence, the following clustered topologies emanate from simple mesh-based routers with an XY-routing scheme.

Several existing topologies as well as the proposed clustered topologies are introduced in the following together with their addressing schemes. The standard mesh serves as reference here and is termed Mesh (CRR=1) to keep the topologies apart. The Mesh (CRR=1) is illustrated in figure 4-14 a) as a 4x4 network and necessitates only a simple destination tag in the packet header to indicate the receiving resource – as shown in figure 4-15 a). In concrete terms, this means that

the pair of coordinates (x,y) describes the resources in the two-dimensional mesh uniquely. Furthermore, figure 4-14 b) and c) picture two types of concentrated meshes [Bal06b, Gil08]. Namely, the Mesh (CRR=2) with two resources connected to each router and the Mesh (CRR=4) with four resources per router, respectively. The associated addressing scheme is shown in figure 4-15 b), whereas the destination tag only uniquely identifies the targeted router. Hence, an additional resource tag distinguishes the final destination resource. The fourth implementation is the Border-Enhanced Mesh (BEAM), depicted in figure 4-14 d) [Cor10, Kub09]. BEAM rests upon a standard mesh, which is a 2x2 network in the figure, though it adds further resources to the borders of the network. Consequently, the border routers link not only one but two resources to the network −or even three in the corners. Thus, the four routers in figure 4-14 d) accommodate twelve resources. Even though the simple addressing scheme of figure 4-15 a) is sufficient to uniquely identify the resources, the XY-routing algorithm requires a few adjustments.

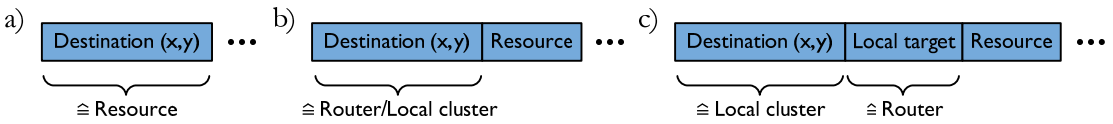

Figure 4-15 : Addressing schemes of the various introduced topologies: a) For Mesh (CRR=1) and BEAM b) For Mesh (CRR=2, 4) and Cluster (1-5, 1-8) c) For Cluster (3-5)

The proposed **clustered topologies** herein are based on a hierarchical communication approach, which is explained with reference to figure 4-14 e). Thereby, four resources form a local cluster, whereas the data exchange among each other is performed by the local router. This local router also connects the cluster to the top-level network so that communication among the clusters takes place across the mesh-based top-level network. Note that similar clusters are connected to every router of the top-level network, these are merely omitted for a better presentation. Since the clustered topology in figure 4-14 e) comprises 1 local router with 5 ports per cluster, it is referred to as Cluster (1-5). Accordingly, the topology in figure 4-14 f) is termed Cluster (1-8) because it employs 1 local router with 8 ports in each cluster. Thus, the local router connects seven resources instead of four as in the former case. Besides the different router degrees of the local routers, both Cluster (1-5) and Cluster (1-8) use the addressing scheme as shown in figure 4-15 b). In these two cases, the destination tag labels the local clusters, or rather the local routers of the clusters. The last routing decision within the clusters is then based on the additional resource tag again. The last clustered topology presented here is illustrated in figure 4-14 g). It comprehends 3 local routers per cluster with 5 ports each and is termed Cluster (3-5) correspondingly. The associated addressing scheme is depicted in figure 4-15 c) and consists of three tags. Hence, the destination tag guides the packets to the destined clusters whereupon the local router is selected by the local target tag. Lastly, the resource tag identifies the final destination for the transmitted packet.

Concerning the implementation of the miscellaneous routers, the reference router can be applied without significant modifications for all topologies. Solely the allocation of the ports as

well as the consideration of the different addressing schemes has to be taken into account for the diverse topologies. Even though the addressing schemes in figure 4-15 appear to be uneven in complexity – ranging from one to three tags – their extent in terms of required bits is very much the same. By way of example, the Mesh (CRR=1) requires an 8 bit destination tag to address 100 resources, strictly speaking 4 bit for each dimension in a 10x10 network. By comparison, the Cluster (1-5) applies 6 bit to distinguish 25 clusters, which attributes to two times 3 bit for the 5x5 top-level network. In addition, the resource tag contains another 2 bit to address the 4 resources within each local cluster. Thus, both Mesh (CRR=1) and Cluster (1-5) necessitate 8 bit to connect 100 resources. Therefore, the design parameters (such as frequency or power) of the differing routers are identical in principle, as long as the router degree is the same.

4.3.2 Network properties and simulation setup

A first benchmark of comparison is the diameter d_{max} of a topology, which expresses the longest distance between any communication pair – refer to equation 27 with the definition [Dal04]. The analytical expressions of the diameters for the topologies under investigation are given in equation 37 to 39. For the flat topologies, the longest communication distance results when sending a packet from one corner of the network to the other. Thus, the diameter d_{max} is calculated by the network size n in the horizontal and vertical dimension – note that the equations here refer to a quadratic network and state the number of hops. In case of the clustered topologies, those hops to and from the local clusters have to be incorporated as well (see equation 38). Furthermore, Cluster (3-5) contributes another two hops since there are several local routers per cluster (see equation 39).

$$d_{max} = 2 \cdot (n-1) \qquad \text{for Mesh (CRR = 1, 2 and 4) and BEAM} \qquad (37)$$

$$d_{max} = 2 \cdot (n-1) + 2 \qquad \text{for Cluster (1-5) and Cluster (1-8)} \qquad (38)$$

$$d_{max} = 2 \cdot (n-1) + 4 \qquad \text{for Cluster (3-5)} \qquad (39)$$

Even though these equations state the diameter d_{max} for a given network size n, they conceal the fact that the network size n of the various topologies is not constant for the same number of resources. Additionally, the diameter d_{max} only reflects the longest communication distance and neglects the distribution of all other connections. Therefore, figure 4-16 a) relates the mean distance in terms of traversed routers $\overline{d}_{router}$ for the different topologies against the number of resources N_{res}. Here, a small mean distance $\overline{d}_{router}$ is desirable from the perspective of performance (see equation 20 and 29). Generally, the mean distance $\overline{d}_{router}$ increases disproportionately slow for all topologies compared to the number of resources N_{res}. Thereby, the ranking closely reflects the network size n that is needed to accommodate the resources – respectively the size of the top-level network in case of the clustered topologies. However, the functions of the clustered topologies exhibit an additional upward offset due to the extra hops from and to the cluster routers (consider equation 38 and 39).

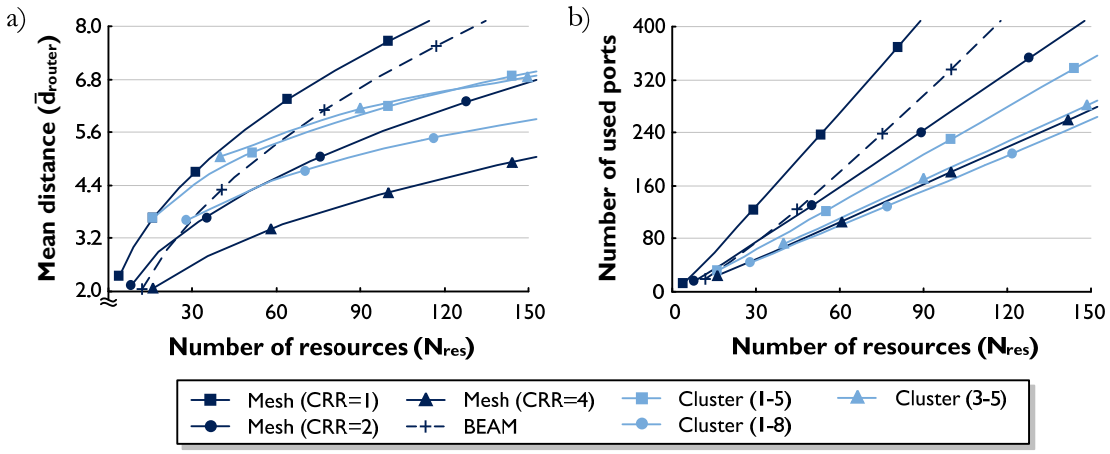

Figure 4-16 : Network characteristics of different topologies against the number of resources: a) Mean distance in terms of traversed routers b) Total number of required router ports

Furthermore, the number of used ports is plotted in figure 4-16 b) for the diverse topologies and an increasing number of resources N_{res}. Compared to the number of used routers, the number of used ports here constitutes a better indication for area and power needs since the router degrees vary significantly. All topologies in the diagram exhibit an approximately linear increase of the required ports in order to connect a rising amount of resources. In contrast to the mean distance $\overline{d}_{router}$ in figure 4-16 a) though, the clustered topologies do not suffer from the extra hop to the cluster routers in this case. Hence, they rank best together with the Mesh (CRR=4).

For the detailed simulations of communication characteristics, all introduced topologies were implemented with about hundred resources. The resulting analytical characteristics of the topologies are contrasted in table 4-3, which allows an initial evaluation. First of all, the number of resources is presented that states how many resources are linked within the implemented designs. Because of their specific compositions, some of the topologies depart from the targeted number of hundred resources. The next column states the number of required routers whereas the network dimensions are additionally specified in the brackets. In case of the clustered

Table 4-3 : Collection of analytical properties for the implemented topologies and a targeted network size of approximately hundred resources

	Number of resources	Number of routers	CRR	Router degree	PCR	Bisection link count
Mesh (CRR=1)	100	100 (10x10)	1.0	≤ 5	4.60	10
Mesh (CRR=2)	98	49 (7x7)	2.0	≤ 6	2.71	8
Mesh (CRR=4)	100	25 (5x5)	4.0	≤ 8	1.80	6
BEAM	96	64 (8x8)	1.5	5	3.33	8
Cluster (1-5)	100	50 (5x5)	2.0	≤ 5	2.30	6
Cluster (1-8)	112	32 (4x4)	3.5	≤ 8	1.71	4
Cluster (3-5)	90	36 (3x3)	2.5	≤ 5	1.87	4

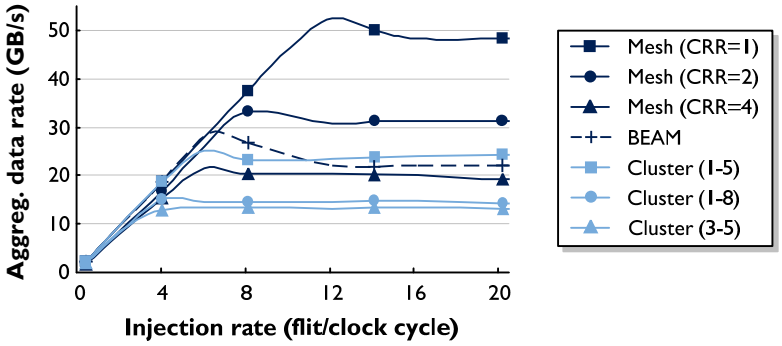

Figure 4-17 : Illustration of the aggregated data rate against the injection rate for the implemented topologies –with approximately hundred resources and uniform random traffic pattern

topologies, the network dimensions relate to the size of the top-level network. For instance, the Cluster (1-5) has 5x5 = 25 routers in the top-level network plus the respective 25 local cluster router. Accordingly, the numbers of routers point out considerable differences between the topologies, which are correspondingly reflected in the CRR of the subsequent column. However, the CRR does not allow for the varying complexities of the routers that can be expressed here by the router degree (see third last column). This matter of fact yields a better suited metric, which is newly proposed in this work, the **Port-to-Core Ratio** (PCR):

$$\mathrm{PCR} = \frac{N_{\mathrm{port}}}{N_{\mathrm{res}}} \tag{40}$$

Comparable to the CRR, the PCR is calculated from the ratio of the number of ports N_{port} divided by the number of resources N_{res} (i.e. cores). Since the number of resources N_{res} is in the denominator though a small PCR is preferable –in contrast to the CRR. According to that, the topologies with larger networks translate into poor PCRs, which is an indication for undesirable power and area needs. Finally, the last column contains the number of links in the smallest bisection. In this case, those topologies with poor PCR benefit now from their large networks with respect to good prerequisites for high communication performance (see equation 36).

4.3.3 Communication characteristics and design costs

The implemented topologies were extensively simulated in order to evaluate their communication characteristics in the face of the design costs. First, figure 4-17 plots the aggregated data rates of the diverse topologies over the injection rate, whereas all functions exhibit in principle a similar course. That is to say, the data rates rise linearly for smaller injection rates. Besides, above the saturation points the data rates settle down at rather constant levels. However, the particular injection rates where the networks saturate differ significantly, and thus the maximum data rates also vary substantially. By way of example, the Mesh (CRR=1) saturates

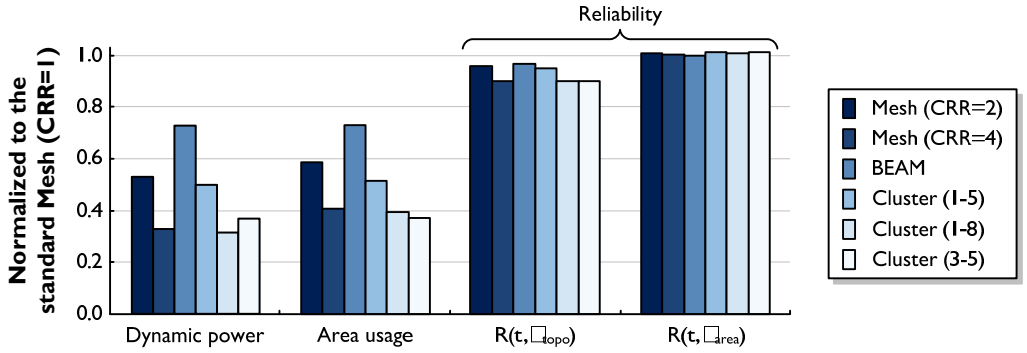

Figure 4-18 : Figures of dynamic power, area and reliability for the various topologies in relation to a standard Mesh (CRR=1) −whereas reliability is stated as a function of the topology's failure rate $R(t, \lambda_{topo})$ and area $R(t, \lambda_{area})$

above an injection rate of 12 flit/clock cycle and reaches an aggregated data rate of approximately 50 GB/s at most. In contrast, the Cluster (3-5) reaches already at 4 flit/clock cycle its maximum of about 13 GB/s. These findings can analytically be explained taking equation 36 with the definition of the bisection bandwidth B_B into account. Since every second packet crosses the bisection in case of uniform traffic, the aggregated data rate cannot exceed twice the bisection bandwidth B_B. This matter of fact is considered in equation 41 and determines the ideal aggregated data rate DR_{ideal} of a topology.

$$DR_{ideal} = 2 \cdot B_L \cdot B_c = 2 \cdot B_B \qquad \text{for uniform traffic} \tag{41}$$

$$\Rightarrow \quad DR_{res} \leq \frac{DR_{ideal}}{N_{res}} \tag{42}$$

Furthermore, on the basis of the ideal aggregated data rate DR_{ideal} an upper bound for the data rate of a resource DR_{res} can also be deduced. This correlation is given in equation 42 and denotes the highest reasonable injection rate, or in other words, the upper limit for the saturation point. Therefore, the topologies with small-sized networks are constrained due to lower bisection link counts B_c, which results in smaller aggregated data rates and earlier network saturation −consider the measures in table 4-3 and figure 4-17. In fact, simulated data rates can be substantially lower than the upper bounds due to load imbalances as well as imperfect flow control and routing schemes [Dal04, Dua03].

As the measures in table 4-3 have already pointed out, the diverse topologies trade performance for power and area overhead. Hence, the prior results of communication performance are supplemented with corresponding investigations now, whereas the figures solely correspond to the network routers. Figure 4-18 presents the parameters of power, area and reliability in relation to a standard Mesh (CRR=1) as the reference. For a start, all topologies necessitate considerably less area compared to the Mesh (CRR=1), whereas the ranking closely matches the classification based on the PCR (see table 4-3). By way of example, BEAM requires

27 % and Cluster (1-5) even 63 % less area than the reference topology. Moreover, the dynamic power consumption P_{dyn} is also quoted in the figure – recall equation 8 whereby $P_{dyn} \propto f \cdot C_{load}$. Accordingly, since area usage is closely coupled with dynamic power dissipation, the rating of the power values in figure 4-18 is very similar to the area values. However, those topologies that require larger router degrees have to operate the networks with smaller frequencies – consider table 4-3 and figure 4-13. This fact turns into further power savings for the corresponding topologies, whereby Mesh (CRR= 4) and Cluster (1-8) rank best from a power perspective.

The last parameter to be observed here is the reliability of the various topologies. At first, reliability is specified in dependence of the same failure rate for all topologies λ_{topo} – named $R(t, \lambda_{topo})$ in the figure. The definition of reliability is to be understood as the percentage of connections that are still functional in case the architecture is affected by a failure. Given this backdrop, the standard Mesh (CRR=1) is more reliable than all other topologies because it features the largest network size. Simply stated, the amount of connections that are affected by a certain failure is smaller the larger the network is – this is also reflected in the standings of the other topologies. Even though several published works argue based on such an examination, they neglect the impact of system size. A larger architecture though is in general more likely to be affected by a failure. In order to express the reliability of the topologies with respect to their area usage, the communication architectures are thought of as series connections of their components [Leh07]. According to that, the system reliability of such a series connection $R_S(t)$ can be expressed by the product of the particular component reliabilities $R_i(t)$ [Joh89, Kor07]. This relation is given in equation 43 whereas λ_S and λ_i denote the failure rates of the system and of the components, respectively.

$$R_S(t) = \prod_{i=1} R_i(t) = \prod_{i=1} e^{-\lambda_i \cdot t} = e^{-\lambda_S \cdot t} \qquad \text{with} \quad \lambda_S = \sum_{i=1} \lambda_i \tag{43}$$

$$\Rightarrow \quad R_{S2}(t) = \left[R_{S1}(t)\right]^{\frac{A_2}{A_1}} \qquad \text{with} \quad A_1 \propto \lambda_{S1} \text{ and } A_2 \propto \lambda_{S2} \tag{44}$$

With reference to the exponential function, equation 44 can be derived that sets the reliabilities of two systems – i.e. $R_{S1}(t)$ and $R_{S2}(t)$ – in relation to their areas A_1 and A_2. Based on this equation, the last category of figure 4-18 states the reliabilities considering both the composition and the area of the diverse topologies – termed $R(t, \lambda_{area})$. Since the failure rate λ_{area} relates to the area there, the small-sized topologies can compensate for their reduced network redundancy because they are simply less often affected by a failure in absolute figures. As a result, all topologies are about even in terms of their communication reliability.

The miscellaneous results are finally contrasted in table 4-4 to allow a comprehensive comparison of the different studied topologies. Thereby, those best measures of each column are emphasized by a check mark. The first four data columns present the essential characteristics from the previous investigations normalized to the Mesh (CRR=1), namely the maximum data rate, power consumption, area usage and reliability. To sum up, there is no evident best topology

Table 4-4 : Summarization of network characteristics for the different implemented topologies (figures are normalized to 52.61 GB/s, 634.1 mW and 2.45 mm²)

	Maximum data rate	Dynamic power	Area usage	Reliability	Energy per kB	Area per MB/s
Mesh (CRR=1)	✓ 1.00	1.00	1.00	1.00	1.00	1.00
Mesh (CRR=2)	0.63	0.53	0.58	1.01	0.84	✓ 0.93
Mesh (CRR=4)	0.41	0.33	0.41	1.00	✓ 0.80	1.00
BEAM	0.54	0.73	0.73	1.00	1.35	1.35
Cluster (1-5)	0.48	0.50	0.52	1.01	1.05	1.08
Cluster (1-8)	0.29	✓ 0.31	0.40	1.01	1.09	1.37
Cluster (3-5)	0.25	0.37	✓ 0.37	1.01	1.45	1.48

under all circumstances, but rather different reasonable solutions depending on the parameter of primary interest. Put simply, the network size −which is required to accommodate a given number of resources in a specific topology− represents the basic lever to trade communication performance for power consumption and area needs, respectively. For instance, the large network of the Mesh (CRR=1) features the best data rate but also the highest power dissipation and area needs. Contrariwise, the small network of the Cluster (1-8) suffers from low data rate while requiring the least power and little area.

While the results for reliability do not apparently distinguish between the topologies here, two further metrics are given in table 4-4 that combine different design aspects. These two metrics −the energy per kB and the area per MB/s− set the communication performance in proportion to power consumption and area needs, respectively. When considering these aspects, it is two other topologies again that flag the best measures, namely, the Mesh (CRR=2) and the Mesh (CRR=4). Apart from that, there is no perceived measure to this day that accounts for reliability with respect to performance, power or area. Therefore, this thesis proposes evaluating prospective integrated circuits based on the newly defined **Energy-Reliability Ratio** (ERR), which is defined as the division of energy (i.e. the power-delay-product) by reliability:

$$\mathrm{ERR} = \frac{P_{\mathrm{tot}} \cdot t_{\mathrm{d}}}{\mathrm{Reliability}} = \frac{\mathrm{Energy}}{\mathrm{Reliability}} \tag{45}$$

Against the background of future, faulty technologies, such a measure allows a better comparison of different design approaches without disguising a key parameter. Nonetheless, a few crucial questions relating to the definition of reliability itself still remain. For instance, how to quantify reliability, which physical causes have to be considered and how to characterize their appearance −in particular, concerning the frequency of occurrence or the extent of impact. Albeit a few measures for accessing reliability in NOCs have been suggested in [Gre07a], a comprehensive metric is neither acknowledged nor available at all. Moreover, appropriate characterizations of failure causes and their occurence have to be provided by the manufacturers.

This is a mandatory requirement in order to integrate meaningful failure models into the automated design process of integrated circuits. Consequently, extended benchmark suites will have to be developed that determine faulty conditions and allow assessing both communication and computation properties of complex systems. Given that a great variety of diverse application scenarios exists, such benchmarks will have to be platform-dependent and domain-specific.

Concluding, the proposed clustered topologies are beneficial alternatives for communication architectures that are power or area-constrained. Furthermore, the local clusters are also advantageous in an architecture of parameter islands since the clusters are independent and encapsulated – except for the connection to the top-level network. Thus, different clusters can be operated with diverse frequencies, voltages or even varying protocols in order to efficiently adapt the system to actual needs [Put07]. However, the favorable attributes of power and area are traded for communication performance so that, depending on the application, very different topologies can be most qualified. Lastly, the results here were related to uniform traffic. It is promising though to investigate how system characteristics change when most of the traffic occurs within the clusters themselves. Hence, the impact of such local traffic is presented in subsection 4.4.1.

4.4 Exploiting architectural characteristics

Networks-On-Chip (NOCs) are characterized by their modular composition and inherent redundancy of the communication architecture. These properties distinguish NOCs from most conventional approaches and necessitate the development of refined and novel design solutions. On the other hand, such architectural characteristics can also be exploited in order to improve complex integrated systems. Correspondingly, two approaches are investigated in the following subsections that refer to the special attributes of on-chip networks.

4.4.1 Local traffic

In bus-based architectures, the entire capacitance of the shared medium has to be charged and discharged independent of the locations of communication participants (see subsection 2.4.2 and 2.4.4). By contrast, in NOC-based architectures only those dedicated wires have to be switched that are required to connect sender and receiver. Hence, performance and power consumption depend on the communication distance in this case. This architectural characteristic of NOCs can be taken advantage of when the average communication distance is reduced – in consequence of local application mapping.

In order to examine the impact of communication distance, three common probability distributions are used here to select the destination resources in the network [Dal04, Par00]. The first one is the discrete uniform distribution. Its Probability Mass Function (PMF) is given in equation 46, whereby each event k from a set of m elements is equally probable to occur. Referring to the reference NOC with 81 resources, the probability of each resource to be the receiver of a specific packet is $1/m = 1/81 \approx 1.23$ % in this uniform case.

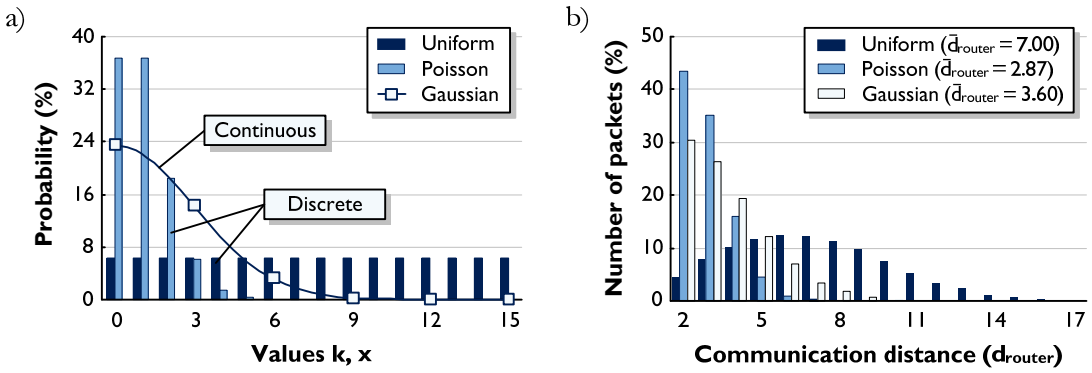

Figure 4-19 : a) Plot of the original probability distributions and their b) effect on the resulting communication distances in terms of traversed routers d_{router} (for the reference NOC with N_{res} = 81 and CRR=1)

$$f(k) = \begin{cases} \frac{1}{m} & \text{for } k = k_i \text{ and } i = 1, 2, \ldots, m \\ 0 & \text{otherwise} \end{cases} \tag{46}$$

The second implemented function is based on the Poisson distribution, whereas λ_P conforms to both mean and variance of the probability mass function in equation 47. For simulations of local traffic, the Poisson distribution can be used such that nearby resources of a sender are more probable to become the receivers of packets [Pan05]. Lastly, the third function in equation 48 is applied in the same way but relates to the Gaussian distribution. Note that this is a continuous probability distribution −unlike the previous two distributions− that is thus represented by a Probability Density Function (PDF). Here, x is a continuous random variable, μ depicts the mean and σ stands for the standard deviation of the distribution.

$$f(k; \lambda_P) = \frac{(\lambda_P)^k \cdot e^{-\lambda_P}}{k!} \quad \text{with } k \in \mathbb{N}_0 \text{ and } \lambda_P \in \mathbb{R}_{>0} \tag{47}$$

$$f(x; \mu, \sigma) = \frac{1}{\sigma \cdot \sqrt{2\pi}} \cdot e^{-\frac{1}{2} \cdot \left(\frac{x-\mu}{\sigma}\right)^2} \quad \text{with } x, \mu \in \mathbb{R} \text{ and } \sigma \in \mathbb{R}_{\geq 0} \tag{48}$$

Figure 4-19 presents the theoretical peculiarities of the introduced distributions and their effect on communication distances in a standard mesh topology. As a start, figure 4-19 a) plots the probabilities for the values x and k in the range from zero to fifteen −whereas m = 16, λ_P = 1, μ = 0 and σ = 3. While the probabilities are constant in the uniform case, they are unevenly distributed in the other two cases. More specifically, the probabilities of the Poisson and Gaussian distribution are high for lower values and drop considerably for larger values of k (respectively x). Transferred to the practical implementation, this means that the result of a uniform random number generator directly determines the destination resource of a communication. Thereby, m = 81 and sending a packet to oneself is excluded (i.e. $n_s \neq n_d$). In

contrast to the uniform case, the result of a Poisson random number generator cannot directly determine the destination resource. Instead, the result is regarded as the communication distance between sender and receiver in terms of traversed routers. That entails two consequences for the implementation. At first, the outcome of a Poisson number generator has to be increased by two since at least two routers have to be traversed to reach another resource. Subsequently, the obtained communication distance is randomly distributed to the horizontal and vertical directions of the network to determine the destination resource – originating from the sender of the packet. Results from the Gaussian distribution are handled likewise whereas only non-negative integers are considered in this case.

The resulting communication distances for the implemented probability distributions are shown in figure 4-19 b) for the reference topology with 81 resources – i.e. $d_{\text{router}} \in [2, 17]$. The ordinate there states the relative fractions of packets for specific communication distances in terms of traversed routers d_{router}. The courses of the simulation results in figure 4-19 b) do not exactly match the theoretical distributions as presented in figure 4-19 a). That is because of the two-dimensional topology and the limiting borders of the network. For instance, consider that solely the four resources in the network corners can if at all communicate over the longest distance of 16 hops, respectively 17 routers. However, the chosen approach reproduces local traffic effectively whereby most of the packets reach nearby resources. In quantitative terms, compared to the 7 routers of the uniform case, the mean communication distance $\overline{d}_{\text{router}}$ is reduced to 2.87 routers in case of the Poisson and 3.6 routers in case of the Gaussian implementation. The mean distances of all contemplated scenarios are summarized together with further metrics in table 4-5 at the end of this subsection.

Since the communication distance can only serve as an indication for communication characteristics, a closer look is taken at the packet delays and data rates. Therefore, the standard Mesh (CRR=1) and the Cluster (3-5) from the previous section 4.3 are comprehensively investigated. Thereby, a data width of 64 bit is employed here and the mesh topology comprises 81 resources. Figure 4-20 presents the histograms of packet delays for the two topologies and the different traffic distributions – for an injection rate of 2.03 flit/clock cycle. When considering the results in figure 4-20 a) for the Mesh (CRR=1) first, it can be observed that the distributions of packet delays diverge from the ones for the communication distances in figure 4-19 b). There are essentially two reasons for this. On the one hand, not only the distance but also the varying packet lengths affect the packet delays (note equation 29). On the other hand, although the injection rate is rather low, some packets are additionally delayed due to congestions in the network [Bal06b]. The emerging mean packet delays $\overline{t}_{\text{packet}}$ are additionally stated in the diagrams and in table 4-5. According to that, the substantial decrease of communication distances yields moderate improvements for the packet delays of the Mesh (CRR=1) because the delays are influenced by communication distance, packet length and network congestion.

Besides, the packet delays from Cluster (3-5) are shown in figure 4-20 b). To begin with, the packet delays of the uniform traffic spread over a wide range compared to the mesh topology.

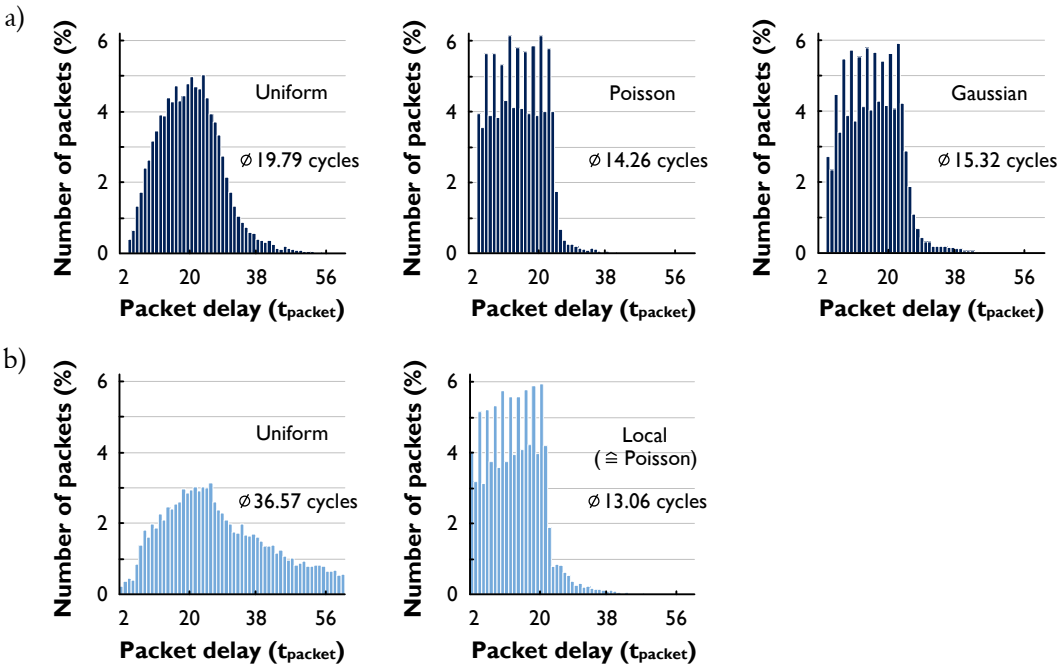

Figure 4-20 : Histograms of packet delays (t_{packet} in clock cycles) for the a) Mesh (CRR=1) and the b) Cluster (3-5) depending on the distribution functions to determine the destination resources (for an injection rate of 2.03 flit/clock cycle)

This is a sign of network saturation and is thus also expressed in the high mean packet delay of 36.57 clock cycles. The second traffic pattern though relates to a local scheme and reveals a very different distribution of packet delays. The local traffic here is identical to the Poisson traffic of the Mesh (CRR=1) in the sense that the same neighboring resources are determined as communication destinations. However, the mean distance $\overline{d}_{router}$ as well as the mean packet delay $\overline{t}_{packet}$ is smaller within the Cluster (3-5) because of a better CRR −in other words, because there are less routers in the network (see table 4-3 and table 4-5). Thus, in case of the Cluster (3-5) the improvements due to local traffic are significant since the network is relieved from saturation −as it is occurring with the uniform pattern.

As local traffic has shown to be beneficial, communication characteristics of all five introduced scenarios are contemplated across varying injection rates. Hence, figure 4-21 relates the mean packet delay and aggregated data rate against the injection rate for the two topologies and their diverse traffic patterns. Put simply, all results exhibit the same characteristic courses as explained at the beginning of chapter 4. That is, both mean packet delay and aggregated data rate behave adequately for lower injection rates. Above the saturation point though, packet delays boost rapidly whereas the data rates settle at constant levels. When examining the results of the mean packet delay in figure 4-21 a) for low injection rates first, the improvements due to local traffic appear again for the Mesh (CRR=1) as well as the Cluster (3-5) as found in the histograms of figure 4-20. While the communication distances are considerably smaller −e.g. $\overline{d}_{router}$ scales

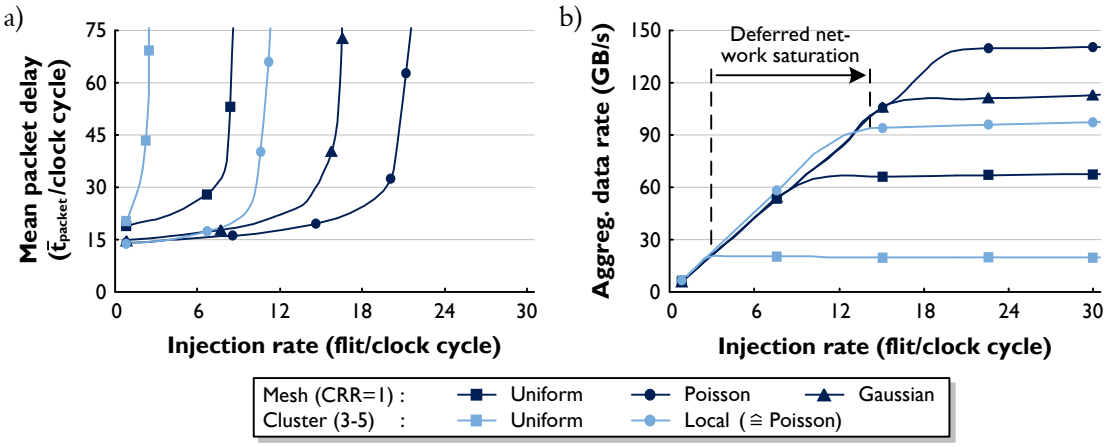

Figure 4-21 : Impact of different traffic patterns and injection rates on the a) mean packet delay and the b) aggregated data rate for the Mesh (CRR=1) and the Cluster (3-5)

from 7 routers down to 2.87 routers– the improvements of the mean packet delays are comparatively small. This is because, for low injection rates, the packet delays are dominated by the serialization delay t_{serial} (i.e. by the packet length) and not by the communication distance d_{router} (note equation 29). By contrast, for high injection rates, it is the contention delay t_{grant} that dominates, whereby small communication distances become highly beneficial. This correlation is also reflected figure 4-21 a) by the fact that network saturation arises much later.

Since the aggregated data rate increases linearly with the injection rate up to the saturation point, deferred network saturation also increases the maximum data rate of a topology. Such deferred network saturation is illustrated exemplarily for the Cluster (3-5) in figure 4-21 b). Hence, the maximum aggregated data rate of the Cluster (3-5) is five times higher under local traffic –from initially 20.45 GB/s to 100.13 GB/s– compared to uniform traffic (see figure 4-21 b) and table 4-5). The maximum data rate of the Mesh (CRR=1) also ameliorates under local traffic (i.e. Poisson and Gaussian), but only by a factor of two. The relative improvements of the Cluster (3-5) are considerably larger because most of the packets are sent within the clusters, whereby the better CRR of the Cluster (3-5) can be exploited and the small top-level network does not constrain the traffic. After all, the discussed communication parameters are summarized in table 4-5.

Concluding, local traffic leads to performance increase in on-chip networks because of smaller communication distances. Contrariwise, when the data rate remains unchanged such reduced distances can also be applied to save power –e.g. in consequence of enhanced application mapping. In any case, clustered topologies benefit more from local traffic since they can exploit their concentration of resources within the clusters (i.e. the higher CRR). Lastly, the impact on reliability cannot explicitly be determined since some failure causes depend on the logical state and not on switching activity (see subsection 2.3.2). Besides, unbalanced local traffic can increase

Table 4-5 : Change in communication characteristics when local traffic is enforced (modeled by Poisson and Gaussian distributions; mean distance given in routers and mean packet delay in clock cycles)

		Mean distance ($\bar{d}_{router}$)	Mean packet delay ($\bar{t}_{packet}$)	Maximum data rate (GB/s)
Mesh (CRR=1)	Uniform	7.00	19.79	69.31
	Poisson	2.87	14.26	144.32
	Gaussian	3.60	15.32	116.02
Cluster (3-5)	Uniform	6.02	36.57	20.45
	Local	2.16	13.06	100.13

absolute temperature and temperature gradients, which both can deteriorate reliability as well. In summary, an appropriate system management has to balance the aspects altogether according to the needs of the architecture and its applications (see also sections 4.5 and 5.3).

4.4.2 Distributed monitoring and control

The modular nature of networks-on-chip allows designing and operating parts of the architecture under heterogeneous conditions – for instance, with diverse supply voltages, frequencies or even unequal technologies [Kim08]. Furthermore, even in homogeneous architectures, intra-die parameter variations cause considerable differences in performance and power dissipation across the chip [Bon00, Aga03]. And lastly, running applications lead to unbalanced workloads concerning computation as well as communication, which influences amongst others performance and wearout. In a nutshell, those various aspects impact the performance, power and reliability of the system. However, they are not known in advance and thus need to be monitored and controlled during system operation [De09].

Monitoring of computation attracted early attention by researchers and the industry so that several approaches were published and are in fact in practice [Arm07, Ver01, Oke04]. Recently, there is an urge to also monitor physical parameters during system operation due to the gaining importance of parameter variations, adaptive system operation (e.g. DVS and DFS), reconfiguration and long-term degradation [De09, Rid10, Kim05b]. The parameters to be measured include first and foremost temperature, voltage, current and aging [Ham07]. Thus, those previous works focus on computation in coarse-grained architectures or consider temperature for dynamic power management [Ska04]. However, they do not reflect the characteristics of highly modular networks-on-chip with their distributed communication system. Hence, several other works applied traffic counters and communication statistics to exploit this kind of information for performance improvements [Nol04, Geb09]. A comprehensive description of such monitoring and its impact for system design was given in [Cio05] and [Cio06b], though reliability is not considered. Therefore, this section presents an advanced

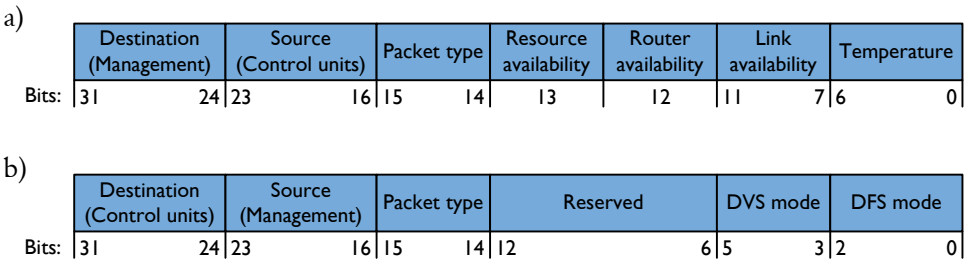

Figure 4-22 : Configuration of the two control packets for distributed monitoring and control in NOCs: a) Packet as sent and b) packet as received by the control units

general concept to implement distributed monitoring and control in networks-on-chip. The approach not only aims at monitoring the system, but also accounts for active, distributed control mechanisms. Beyond this, the seamless interaction between the hardware architecture and a flexible system management (i.e. in software) is another goal.

The fundamental approach is to implement distributed modules within the network that monitor as well as control system characteristics. To simplify matters, such modules are called **control units** and they can comprehend the following tasks. On the one hand, monitoring includes the observation of physical parameters (e.g. temperature, voltage) as well as of computation and communication characteristics (e.g. statistics, availability or failures). Therewith, system behavior and changes can be sensed but not be affected. On the other hand, control describes the active influence on system behavior [Kim08]. This can be achieved, amongst others, by the control of frequencies and voltages (i.e. DFS and DVS) or by the adaptation of computation and communication – such as application remapping or the relocation of routing paths.

The pictured concept was implemented and comprehensively investigated concerning its impact for the system design as a whole. Due to the extent of gained results, those findings concerning the design of the control unit itself are only briefly depicted here. A detailed description of the miscellaneous design considerations can be found in the compiled and supervised works [Weg09] and [Weg10].

- Control units can both send and receive packets. Thus, monitored data can be sent and control commands can be received from other modules – for instance, from the system management that comprises global awareness (see also section 4.5).
- The trigger of activity (e.g. generating and sending a packet) is event-based.
- Control packets are identified by the 'packet type' field of the header flit.
- Monitored parameters are the availabilities of resources, routers and links as well as the local temperature. Whereas the detection of these parameters is assumed to be given based on functional, physical or virtual sensors [Nol04, Rid10, Han06].
- Control commands can be received that allow setting the frequency and the supply voltage of the associated resource.

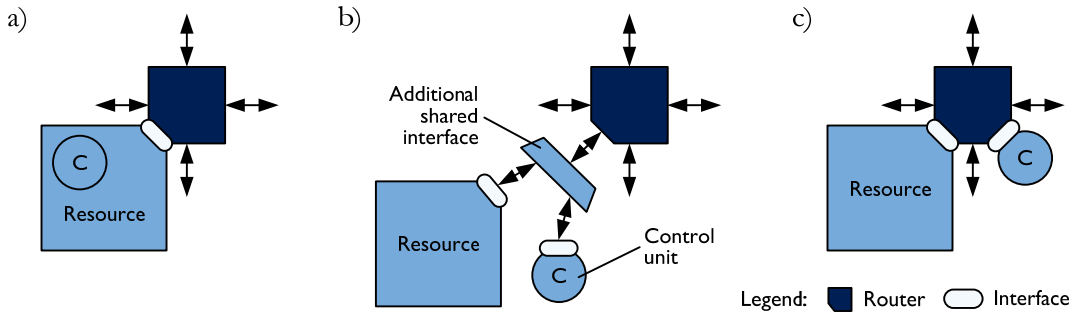

Figure 4-23 : Illustration of different alternatives to connect the control units to the on-chip network: a) Integrated in resource b) With an additional interface c) With an extra port

On the basis of the preceding design decisions for the control unit itself, implications for the use in a NOC-based architecture are discussed now. For this purpose, the reference architecture builds the starting point with a data width of 32 bit and a FIFO depth of 4 slots. Hence, figure 4-22 a) and b) depict the configurations of the applied control packets for sending and receiving. Thereby, the packet in figure 4-22 a) is sent from the control unit (i.e. the source) to the system management (i.e. the destination) whereas the packet type labels the control packet. After those fundamental fields of a common packet header, it follows the different monitored parameters, whereby one flit is sufficiently large to contain the selected data. Thus, the temperature value occupies seven bits and the availabilities of the modules are represented by single bits, which demands five bits for the attached links of a router. Figure 4-22 b) shows the packet configuration that is sent in the opposite direction, meaning that in this case the source corresponds to the system management and the destination is the control unit. Accordingly, this kind of control packet includes two 3 bit wide fields to set supply voltage and frequency of the associated resource (see fields DVS and DFS mode). Note that the source field here can also be exploited to indicate when the location of the system management changes – for instance, due to an outage or an intentional relocation.

Now that the individual control units are determined, a closer look is taken at the different alternatives of how to connect the control units to the on-chip communication network. Thereby, the option of a redundant control network is disregarded because of the associated area and power overhead [Cio06c, Geb09]. This being said, figure 4-23 a) illustrates the first approach whereas the control unit is fully integrated within the resource. Consequently, the control unit shares the functionality of the existent interface. The second approach in figure 4-23 b) places the control unit outside the resource while both modules still share the same router port. Hence, an additional shared interface switches data and arbitrates communication demands between the control unit and the resource. Lastly, the third design shown in figure 4-23 c) employs an extra router port whereby the communication of the resource is independent from the control unit.

Table 4-6 summarizes the results of the different implementations, whereas a conventional router without any control capabilities serves as the reference. To begin with, all designs exhibit an area overhead due to the extended functionality. Thereby, the approach with the control unit

Table 4-6 : Comparison of the different design alternatives to connect the distributed control units to the on-chip network (normalized to a reference without any control capabilities)

	Frequency	Area usage	Remark
Reference (no control unit)	1.000	1.000	
Integrated in resource	1.000	1.075	Modularity forfeited
Additional interface	1.000	1.112	Additional latency
Extra port	0.922	1.193	Data and control packets do not interfere

integrated in the resource suffers from the least area increase because the existing resource interface is shared. For the same reason, packet delivery can be delayed when both control unit and resource necessitate the interface concurrently. The most severe constraint though is that modularity is forfeited or the integration might not even be possible at all – for example in case that the resource represents a given third-party Intellectual Property (IP). The approach with the additional interface occupies more area due to the increased complexity, although the data rate of the port still has to be shared among one another. The extra area though also facilitates that the control unit works in case of a faulty or powered down resource. While the operating frequency remains the same, communication performance is nonetheless detracted because the additional interface implicates at least a further clock cycle of latency. Finally, the extra port of figure 4-23 c) facilitates that data and control packets do not interfere when accessing the network. However, this comes at a price since the extra port increases the complexity of all router ports – for instance, consider arbitration and the switch matrices (see also figure 4-13). Therefore, not only the area usage is the highest but also the operating frequency is cut down so that communication performance deteriorates accordingly. In summary, the design of the control unit with the additional interface is most applicable since it offers the best system characteristics without forfeiting the modularity of the architecture.

Concluding, the integration of distributed monitoring and control within networks-on-chip facilitates to effectively adapt system characteristics to current conditions, whereas such additional functionality comes at the price of slightly increased design costs. Thereby, the introduced control units cover miscellaneous aspects ranging from physical parameters to resource availabilities, and feature a seamless interface to the system management. Therefore, an appropriate system management can exploit the combination of its global system awareness and the knowledge of low-level parameters.

It should be noted that the presented implementation demonstrates the basic component of a novel, general concept to approach a cross-layered system management. This is of particular interest to address reliability, which is referred to in the following brief description. Figure 4-24 illustrates the new concept whereas the advanced functionality is situated in the **Error Resiliency Layer** (ERL) that joins software and hardware components. Consequently, reliability is tackled in a multistage approach:

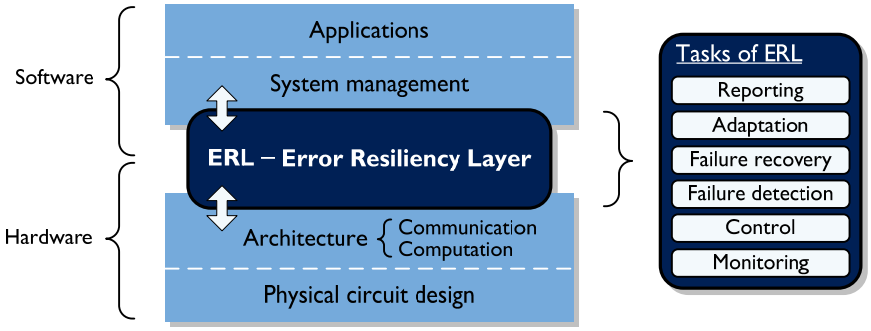

Figure 4-24 : General concept to address reliability in complex integrated systems by means of an Error Resiliency Layer (ERL) that abstracts complexity and offers an effective interface between hardware and software

1. Simple failures are dealt with in the architecture itself. By way of example, corresponding approaches are error correction or the repeated execution of an operation −i.e. exploiting temporal or spatial redundancy.
2. Those reliability concerns that cannot be dealt with in the architecture are processed by ERL. The most basic tasks are monitoring and the execution of control commands, which reflects the functionality of the aforementioned control units. Beyond that, modules are recovered after failures have been detected (e.g. by resolving a deadlock and resetting state machines). Furthermore, elementary adaptations are also carried out in ERL (e.g. throttling frequency). Lastly, in order to keep the design costs for ERL as small as possible, global awareness is not advisable at this stage. Hence, statistics and severe failures are reported to the system management.
3. The reporting of ERL provides a global perspective of the architectural characteristics to the system management. Therefore, further measures can be taken to handle failures or to avoid that failures compromise software applications. Moreover, elaborate algorithms can also be applied to prevent or delay failures (i.e. proactive operation) and to adapt the entire system based on the global and hardware-dependent knowledge.

In summary, the newly proposed Error Resiliency Layer (ERL) offers a multistage approach to increase reliability. Since different tasks are already executed in ERL, the system management is less burdened with the underlying complexity. Thus, ERL represents an effective means to interface the hardware platform and the software system so that flexible, global improvements can be performed based on the state of the architecture.

4.5 System management

The management of computing systems is generally associated with the Operating System (OS). The primary task of such kind of software is to coordinate concurrent applications and to schedule shared resources −like disk space or main memory. Thus, the OS abstracts

sophisticated functionality related to the hardware architecture and provides the functionality through programming interfaces to the applications [Sil08, Tan01]. However, due to the distributed, communication-centric nature of Networks-On-Chip (NOCs) and the continuous scaling of technology, a series of new concerns and tasks has to be accounted for. Hence, to simply adapt common operating systems for the use in NOCs is not appropriate. In order to express the difference to common operating systems, corresponding software is referred to as system management in conjunction with on-chip networks.

The multifaceted tasks of the system management comprehend amongst others the partitioning and mapping of software applications onto the distributed resources as well as scheduling, test, diagnostics and further administrative tasks. Thereby, those diverse tasks have to be performed such that requirements of the applications are satisfied – for instance, with regard to computation, communication or quality of service. While this is already a demanding challenge, physical parameters have to be considered additionally (e.g. temperature) since they affect performance, power consumption and reliability. It is important to note that the system management has to cope with all decisive factors at run-time and that these factors can change for various reasons. On the one hand, varying applications and data sets are executed over time, which results in altering system characteristics – such as work load or temperature. On the other hand, even the hardware architecture cannot be considered steady since reliability concerns can cause modules to fail entirely or to operate at lower ratings. Beyond that, reconfiguration or IP licensing will prospectively exacerbate this circumstance [Hec06, Ull04, Cou06].

This implies that system management for NOC needs to administrate both abstract functionality and hardware conditions at run-time. Only in due consideration of these two diverse aspects in combination, efficient system operation can be accomplished that balances the demands and constraints of performance, power dissipation and reliability. Therefore, this section introduces an advanced concept that is suited to account for the particular characteristics of modular, distributed on-chip networks.

4.5.1 Existing approaches

To begin with, several published works deal with system properties at design time. In [Sri04], reliability is estimated on a microarchitectural level in advance to motivate design decisions. Furthermore, against the background of known applications and their specific behavior, miscellaneous algorithms have been proposed to map those applications onto the distributed resources of an NOC [Asc05, Hu03b, Mur04]. However, dynamic variations during system operation cannot be reflected by these attempts.

There is also a variety of implementations that tackle system tasks at run-time. Thereby, a central coordinating instance with global awareness is commonly assumed [Kav04]. A hierarchical approach is presented in [Ran07], whereas fault-tolerance is increased by means of reconfiguration and a high degree of redundancy. A similar, hierarchical design has been proposed that

exploits DVS and DFS [Gua09] since the dynamic adjustment of voltage and frequency has significant impact on both power and performance [Bro01]. The type of algorithmic to control the system is further expanded in [Hua00] and [Lu05a] to allow for temperature as well. The underlying architecture though is not specified whereby the specific characteristics of networks-on-chip are not regarded. While these approaches modify the operating conditions of a running application, the location of the application itself can also be adapted. Such adaptive mapping is for example described in [Hun04] and [Kum06] whereas the adaptation is based on temperature estimates. Even though, all introduced schemes assume a central coordinating instance, they can principally be implemented in a distributed approach as well [Ben02, Pus08].

The aforementioned proposals refer to the system level and abstract the underlying architecture, whereby the influence of the hardware and the interface between hardware and software is broadly neglected. By contrast, detailed descriptions of the connection between system management and hardware architecture are given in [Nol04] and [Hec06]. The former publication manages application mapping and flow control by polling the resources for packet statistics. The second paper applies Remote Method Invocation (RMI) to provide reconfigurable resources and to abstract functional details −which resembles distributed systems [Pud06]. Since both descriptions are tailored for specific implementations, they lack the allowance for other decisive factors.

Concluding, while specific solutions are at hand, an approach that covers the required aspects of networks-on-chip as a whole is not available. Those required aspects include the trade-off between performance, power and reliability, an effective interface between hardware and software as well as the consideration of dynamic changes at run-time. Therefore, a concept based on service-oriented architectures is proposed in the following, which can incorporate and exploit the characteristics of networks-on-chip.

4.5.2 Service-oriented architectures and their use in integrated systems

A **Service-Oriented Architecture** (SOA) is an abstract design concept that aims at structuring and using diverse functionality in distributed systems [Oas06, Erl05]. While the origination of SOA is associated with business processes, there exist several noteworthy implementations in practice today −such as JINI, UPnP, Web service protocols or DPWS [Sun01, Upn08, W3c04, Oas09]. Because of the heterogeneous application domains, SOA-related terminology is often ambiguous. Hence, the used terms herein refer to the context of integrated systems as close as possible.

In the context of SOA, a **service** represents any kind of functionality that is provided by hard or software. For example, the decryption of a video is a service, which in turn might be composed of multiple smaller services (e.g. quantization and discrete cosine transform). A generic course of events of services is illustrated in figure 4-25, which involves the following stages. At the start, addressing makes sure that every service can be uniquely identified in the network.

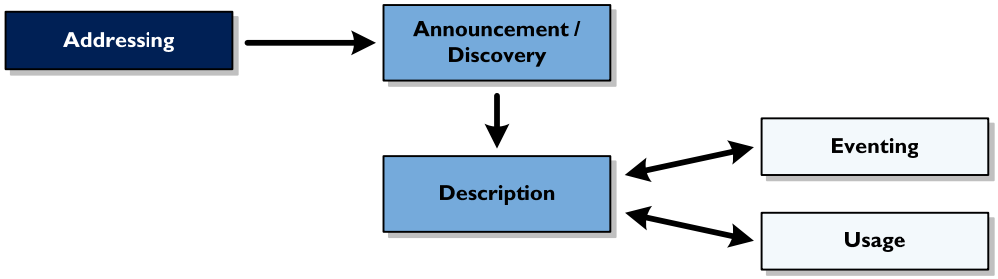

Figure 4-25 : Course of events of services that represent any kind of functionality, whereas the services are provided by hardware or software within a distributed system

Subsequently, there are two mechanisms to become aware of the available services in a network. First, those modules that provide services to others announce their existence. Contrariwise, modules that want to use a service start a discovery. After a service provider and a service user have found each other, a description is exchanged that details the offered functionality and its usage. Finally, based on the prior description, the service can be utilized, controlled or manipulated (in summary called usage). Additionally, a module can also subscribe to a service in order to be informed about state changes (i.e. eventing) – for example, if the service is busy or idle. In a nutshell, modular functionality and determined communication are applied to deal with finding, using and managing distributed services in SOAs. Since such issues have to be likewise addressed by the system management of networks-on-chip, it is worthwhile to delve deeper into the application of SOAs.

The main characteristics of SOAs and NOCs are listed here to exemplify their commonalities [Mor09, Cor06a]. Note that a service conforms to complex functionality as well as to simple tasks in NOC.

- Distributed architecture: Both concepts feature a number of distributed modules in a communication network that host diverse services.
- Encapsulation/Abstraction: The detailed implementation of a service is hidden from the user and the access is made possible via determined interfaces.
- Reuse/Composability: Complex functionality is divided into smaller tasks with the objective to facilitate reuse and to compose other applications.
- Portability: Productivity concerns motivate portability across various platforms and application domains in order to exploit the benefits of reuse.

Besides those commonalities, a crucial difference is that NOCs are highly constrained, for instance due to power consumption, rather low data rate or thermal distribution. By contrast, SOAs are commonly based on heterogeneous networks with few constraints (e.g. consider memory capacity) and very different communication stacks. Hence, to directly take over present implementations of SOAs is inappropriate. However, the underlying concept is still beneficial when the constraints of NOCs are accounted for.

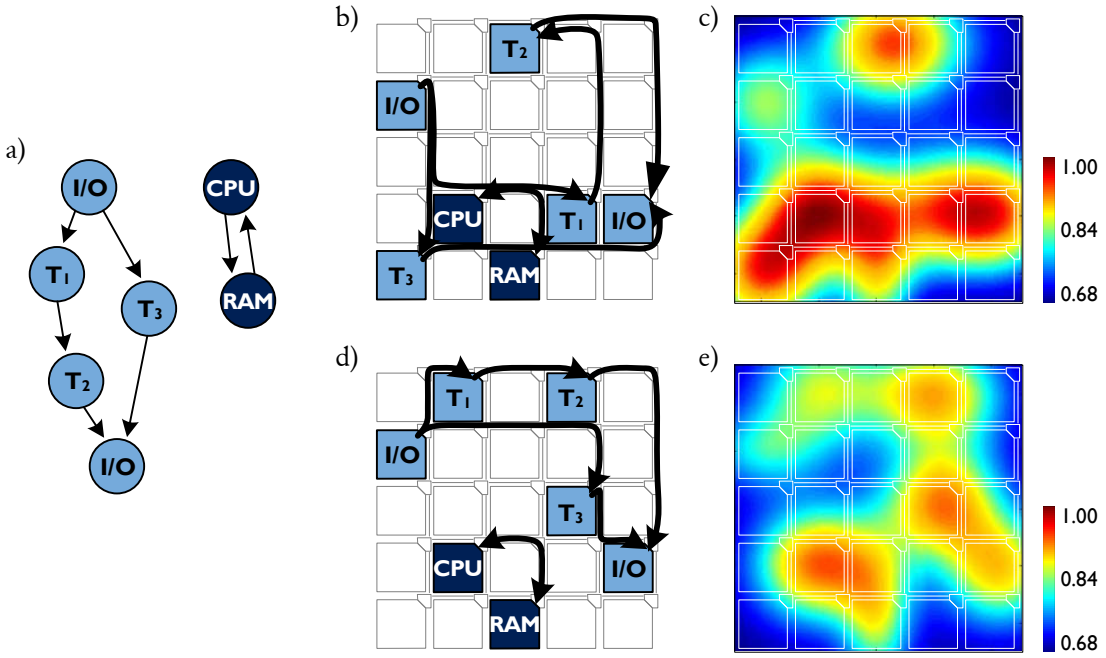

Figure 4-26 : Motivational example of an extended SOA concept for the system management of NOCs: a) Two application graphs b)+d) Mappings within the architecture and their resulting communication paths c)+e) Temperature distributions of the running applications

Therefore, this thesis proposes to submit those NOC-specific characteristics along with the service announcement, respectively service description and eventing. This can be achieved by extending the conventional **metadata** of a service with those parameters that are required to effectively trade off performance, power and reliability in NOCs [Mor09]. That includes amongst others such parameters as:

- the estimated duration for service completion,
- the number of present service users (or the frequency of average usage),
- a metric that denotes the power consumption for the service execution,
- metrics that refer to the current physical condition (e.g. temperature, wearout) and
- the location in the network as this determines the communication costs.

The exact number and specification of the parameters depends on the particular implementation with its system management. Nonetheless, a simple, illustrative example is shown in figure 4-26 that demonstrates how the consideration of architecture-related parameters impacts system operation −a detailed description of the used design tools and simulators is given in section 5.1. The two application graphs shown in figure 4-26 a) constitute the starting point [Han05, Mar09, Mur04]. The former is a generic streaming application with five tasks −i.e. twice I/O and tasks T_1, T_2, T_3− which runs in two concurrent data streams (e.g. a video decoder). The second graph refers to a simple communication scheme between memory (i.e. the

RAM) and Central Processing Unit (CPU). Accordingly, a first approach to map the two applications onto a 5x5 mesh network is shown in figure 4-26 b) together with the resulting temperature distribution in figure 4-26 c). The diagram shows that the temperature is higher at those modules that execute the individual tasks and perform the communication. A second mapping is shown in figure 4-26 d) and e), though the system characteristics are considered here for the selection of resources and routing paths.

Although the two scenarios in figure 4-26 execute the same applications, they result in quite different system properties. The first evident metric of comparison is the communication distance, which is approximately one third smaller in the lower scenario. More precisely, the sum of all communication paths is 26 hops in the upper scenario compared to just 18 hops in the lower scenario. Hence, even if all resources are assumed to operate equally fast, the lower scenario features better performance due to the impact of communication. Furthermore, the better performance and the smaller communication distance also translate into lower power consumption, because of longer idle times of the resources and reduced network activity, respectively – while assuming the same throughput in both scenarios. In addition, the power dissipation of the active resources is distributed over a larger area in the second scenario so that the power densities are also relieved. Hence, the performance advantage in combination with the spatial arrangement of the involved resources finally results in a reduced maximum temperature – compare figure 4-26 c) and e).

Concluding, the consideration of diverse system characteristics is crucial for an efficient system management, in particular for the mapping of computation and communication. The novel, proposed extension of the service metadata facilitates such a hardware-related management in an effective manner. Moreover, the eventing of SOA can also be exploited to dynamically adapt the system to changing conditions.

4.5.3 Implementation considerations for SOA

The previous section has shown that the concept of Service-Oriented Architectures (SOAs) is beneficial for the system management in NOCs. The fundamental question for an implementation though is whether the system is managed in a centralized or in a distributed manner. The **centralized approach** assumes that a sole central instance initiates and controls all activities and system issues, whereby such a design entails a global system perspective. In this case, all resources (i.e. service providers) announce their services to the central management that generates a repository of all on-chip services and their states – refer to figure 4-27 a). Thereby, the states are dynamically updated by the eventing. Based on the central repository, the system management can execute applications by delegating tasks to the distributed resources. The description is therefore submitted to the resources (i.e. service users), which enables them to directly use the provided services. The centralized approach is thus characterized by a complex central instance with large memory needs for the repository. By contrast, there is very little effort required in the

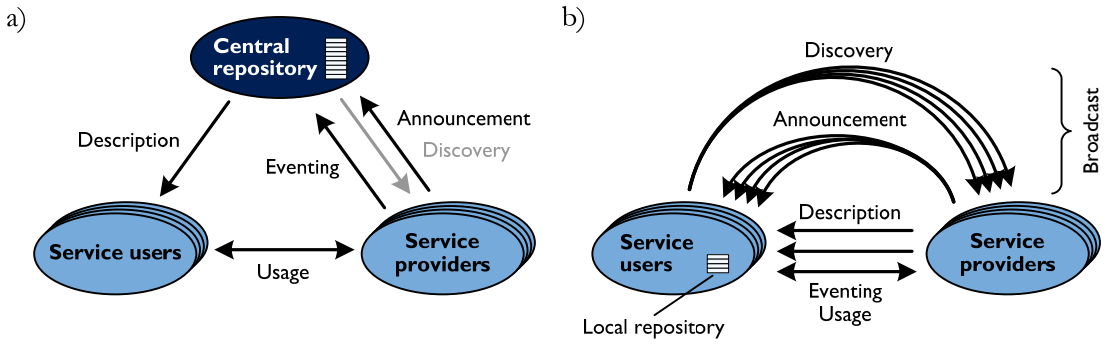

Figure 4-27 : Illustration of two possible implementations of the SOA concept: a) Centralized approach b) Distributed approach (which requires multi and broadcasts)

distributed resources. However, a failure or the entire outage of the central coordinating instance can be fatal. Hence, measures for relocating the central instance have to be considered.

In contrast to the centralized approach, the system management and the service repository are spread across the network in the **distributed approach**. While the complex central instance is avoided, the global system perspective is thus forfeited in this case. Therefore, all resources need to find each other independently, which requires to broadcast the service announcement, respectively the service discovery −as illustrated in figure 4-27 b). Thereupon, a small local repository is generated in each resource that only contains those services of interest. By means of the local repository, resources subscribe to remote services in order to receive the description and to be notified about state changes (i.e. eventing). Lastly, service usage takes place directly between the service users and providers. In summary, the distributed approach represents a modular and flexible design because of its allocation of functionality and memory across the network. Since a central instance is not in charge though, this approach forfeits global awareness and can implicate a certain amount of redundancy −such as repeated storage of services in the repositories.

Both design approaches of SOA were simulated so as to compare their traffic schemes in a network-on-chip. Figure 4-28 shows the normalized results based on the reference architecture with 81 resources, whereas the charted router activities neglect the data packets and solely refer to the control packets of SOA. Moreover, an average number of five subscribers for each service were assumed. Hence, since all control packets lead to or emanate from the central instance, there is a significant accumulation of activity in the network center of the centralized approach −observe figure 4-28 a) where the central instance is located at address (4,4). This emergence represents a critical bottleneck and can lead to congestions as well as increased packet delays. Therefore, the centralized approach becomes inefficient with rising network size because both the number and the average distance of control packets also rise.

The router activities of the distributed approach in figure 4-28 b) though, reveal a very different outcome. First of all, there is no such outstanding peak as in the centralized approach. Instead, the router activities resemble a hemispherical shape, which originates from the

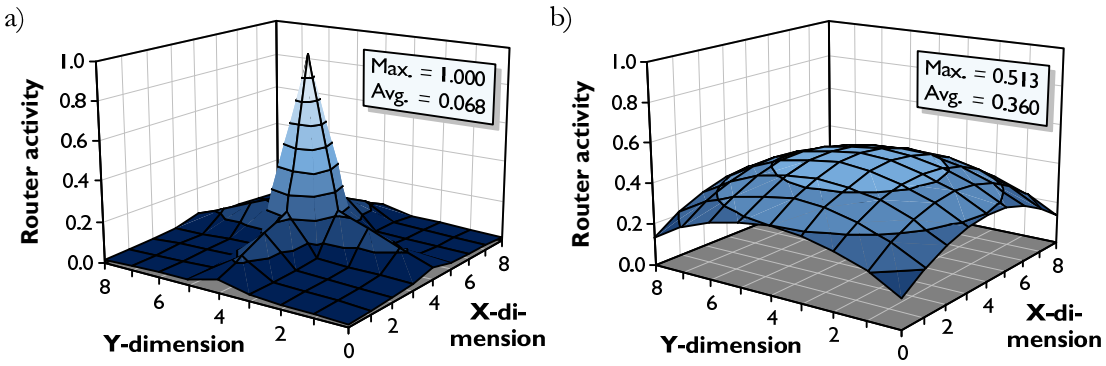

Figure 4-28 : Distribution of the normalized router activities as regards the control packets of SOA in a network-on-chip based on a) a centralized and b) a distributed design approach

dimension-ordered routing (see also figure 4-9). Even though the central peak (i.e. the maximum) is circumvented, the average activity is considerably higher –in statistical terms, 0.36 for the distributed compared to 0.068 for the centralized approach. The difference mainly depends on the number of resources that subscribe to a service. For example, each eventing requires sending one packet to the central instance in the centralized approach. By contrast, the eventing of the distributed approach implicates sending a multicast, which corresponds to one packet to every subscriber of the service. Hence, for the assumed five subscribers –as for the results in figure 4-28 b)– about five times more control packets arise for the same use case. In fact, a detailed analysis of the difference in the number of packets (or rather router activities) necessitates the combined consideration of the system setup, the type of service usages and the communication distances as well.

Furthermore, the distributed approach of a SOA dissipates more power. This is because of the additional packets, a longer average communication distance and the redundancy of local repositories. Thus, one attempt to reduce the power consumption of the distributed approach is to obviate the local repositories by starting a discovery every time a service is about to be used. However, this is impractical as it adds further delay and results in a drastic increase of control packets because each discovery is a broadcast. Another attempt to improve the implementation of SOA is a mixture of the centralized and the distributed approach. Thereby, the system management remains in the local resources, which corresponds to the distributed approach, but the services are listed in a central repository. As a result, both local repositories and broadcasts are avoided. In this case, the router activities resemble the distribution of the centralized approach in figure 4-28 a), though with roughly twice the number of packets –due to the request to and the reply from the central repository. Finally, clustered topologies, as introduced in section 4.3, offer another possibility to exploit the benefits of the centralized and the distributed approach of a SOA. Figure 4-29 illustrates such a topology whereas each local cluster is managed in the form of the centralized approach. Moreover, system management among the clusters operates over the top-level network as in the distributed approach.

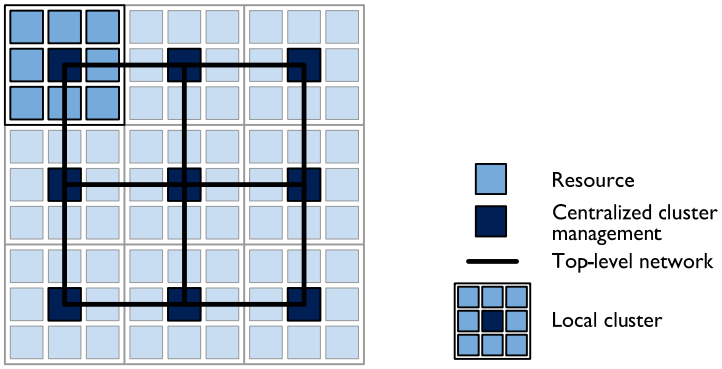

Figure 4-29 : A clustered topology supports the combined implementation of a centralized and a distributed approach of a Service-Oriented Architecture (SOA)

Concluding, service-oriented architectures can be implemented in quite different ways. Thereby, the centralized and the distributed approach as well as their design variants exhibit diverse advantages and drawbacks. Due to the complexity of a final implementation and the correlations across the various abstraction layers, a definite evaluation cannot be given here. However, it can be stated that service-oriented architectures are a qualified candidate for the operation in networks-on-chip since they establish those decisive conditions for a hardware-related system management.

Chapter 5

Case studies of complex systems

Integrated systems are characterized by their considerable complexity. Hence, architectural explorations during the design time have to account for the large design space that is spanned by the topology, the types of resources, routers and links as well as by the various functional protocols. In fact, a similar range of possibilities arises additionally from software design [Soi03]. In order to cope with the immense overall design complexity, it necessitates elaborate tools that support an automated design flow [Itr07a, Itr07b]. Therefore, section 5.1 initially names the requirements for efficient system design and describes how they were established for the case studies herein. Subsequently, two case studies are presented, whereas the first one in section 5.2 aims at demonstrating the intertwined impacts of different design decisions on performance, power consumption and reliability. The second implementation in section 5.3 exemplifies how the consideration of temperature affects system design and operation across the various abstraction layers.

5.1 Requirements for efficient system design

The design of an integrated system has not only to fulfill specified functionality, but also to account for associated rigid demands on, for instance, performance and power dissipation. Thus, it necessitates verifying that an implemented system complies with all given requirements. The most definite approach to do so is to manufacture the implemented system, though this is prohibitive from a monetary and temporal point of view. Another approach is to emulate a system on a flexible computing platform −which is mostly done on Field Programmable Gate Arrays (FPGAs) [Wol07, Gen05]. While both mentioned attempts facilitate to run a large set of test cases in a short period of time, they are restricted to available technology. Hence, prospective aspects can not at all and many important concerns can only hardly be evaluated (such as failure

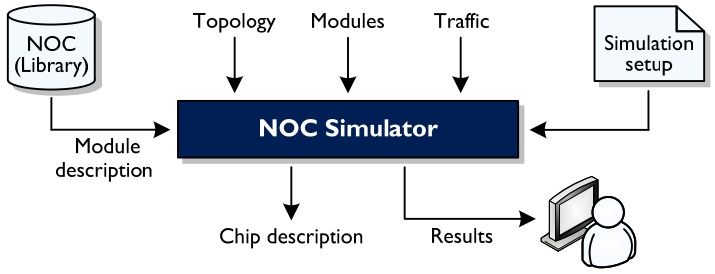

Figure 5-1 : Illustration of the implemented NOC simulator with its corresponding in- and outputs

causes). For this reason, simulation is essential to develop and to verify complex systems, although it requires significant computational efforts. Accordingly, various approaches have been published that rely on diverse hardware description languages [Pen06, Xi06, Wik04, Jal04, Sig02], whereas simulation time is roughly similar in all simulators when the level of detail is the same. Functional abstraction though can drastically decrease simulation time and enables to handle the enormous complexity of integrated systems [Neu09, Ghe05]. However, a drawback of abstraction is that the use of black-box models tends to conceal physical dependencies and global concerns – for example, note transistor wearout, power distribution and temperature profiles. Particularly the consideration of reliability is not regarded across the different layers of abstraction to date.

The specifically developed NOC simulator for this thesis is implemented in VHDL and allows evaluating the characteristic results of a determined chip description [Cor06b]. Figure 5-1 illustrates the NOC simulator with its in- and outputs. To begin with, the NOC architecture is described by the topology as well as by the types of modules (e.g. resources, routers) and their locations. Thereby, those network modules are provided by the NOC library as RTL-level descriptions, which ease the further use during synthesis and layout. By contrast, the resource behavior is highly abstracted and solely modeled by its traffic behavior. For this purpose, basic synthetic traffic patterns are applied that are able to reflect miscellaneous scenarios and applications [Dua03, Lu05b]. Figure 5-2 presents a compiled classification of such traffic patterns and lists the most familiar representatives. Thus, traffic can be characterized based on three attributes. On the one hand, spatial distributions destine the relation between sender and receiver. While such statistical alternatives were investigated in subsection 4.4.1, heterogeneous patterns are often applied to reveal explicit network characteristics or to recreate certain task behavior [Dal04]. On the other hand, temporal and size distributions determine the probability of packet generation and the size of a packet, respectively [Wan02]. Corresponding parameters in the previous analyses are the injection rate and the packet length (see figure 4-1 and figure 4-2).

Since temperature has significant influence on performance, power and reliability, it was searched for a possibility to calculate the temperature distribution during system operation. The most suitable approach is to take advantage of the duality between the electrical and thermal conductivity [Kre00]. Table 5-1 contrasts this duality whereas heat flow conforms to the electric current. Hence, heat flow through a thermal resistance results in a temperature difference, which

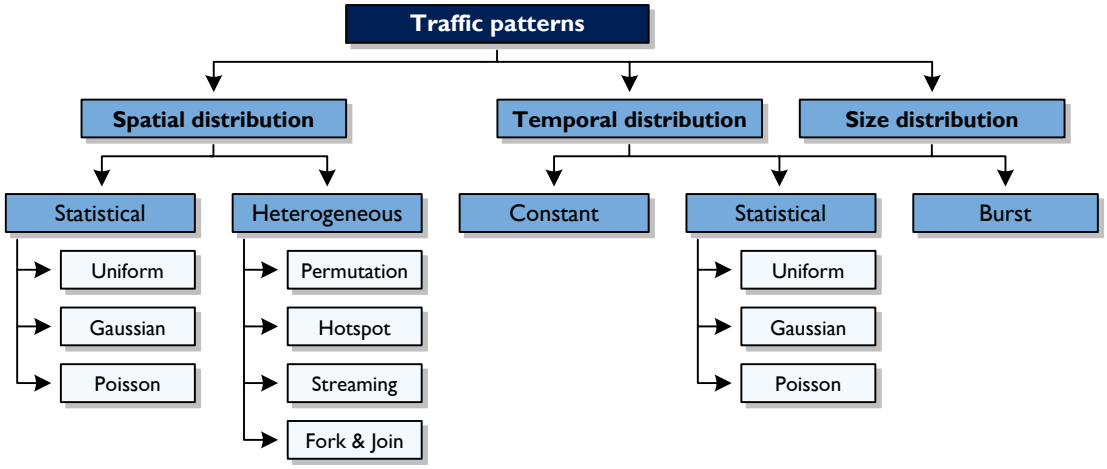

Figure 5-2 : Classification of synthetic traffic patterns that facilitate to reproduce the behavior of resources in diverse scenarios and applications

matches the correlation of the electrical resistance and voltage. The additional consideration of thermal capacitance finally allows modeling the dynamic properties of temperature over time. By this means, simple resistors and capacitors can be used to model temperature distributions with common analog circuit simulators (e.g. SPICE). Several publications exploit the introduced duality to model temperature in integrated circuits [Bro07, Dho00, Ska02, Yan06]. However, they are concerned with conventional processor architectures or suffer from coarse granularity −which is suited to estimate peak temperature but not dynamic spatial distributions [Ska04].

Table 5-1 : Duality of the electrical and the thermal model, which is applied to simulate temperature distributions during system operation

Electrical model		Thermal model
Current flow (I in A)		Heat flow (P in W)
Voltage (V in V)		Temperature difference (T in K)
Electrical resistance (R_{el} in V/A)	⇔	Thermal resistance (R_{th} in K/W)
Electrical capacitance (C_{el} in A·s/V)		Thermal capacitance (C_{th} in W·s/K)
Time constant ($\tau = R_{el} \cdot C_{el}$ in s)		Time constant ($\tau = R_{th} \cdot C_{th}$ in s)

Therefore, an enhanced, fine-grained approach was newly developed for this thesis, which is depicted in figure 5-3 with reference to a 3x3 NOC architecture. First of all, the chip area is segmented into small tiles, whereas the size of a tile determines the granularity, or rather the spatial accuracy of temperatures. Thus, the tile size can be used to trade accuracy off against computational effort. Each tile is composed of several resistors and a capacitor that are set according to the material and the size of the associated chip area. Thereby, the four resistors on the plane of the tile model heat spreading across the area of the chip. By contrast, the two orthogonal resistors constitute the connections to the actual source of power dissipation and the overlying heat sink, respectively. According to the introduced duality, the lower resistor connects

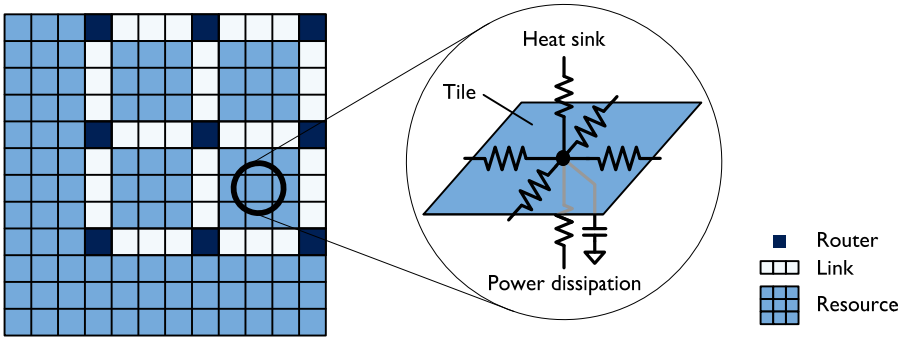

Figure 5-3 : The chip area of the NOC-based architecture is segmented into small tiles with several RC-elements each, which facilitates to model temperature distributions

in fact to a current source, which represents the origin of heat flow (see table 5-1). Here, the strength of current and the chronological sequence are derived from activity simulations of the chip architecture. Hence, the resulting RC-circuit can be simulated and then be evaluated as regards temperature. Further features of the model and the appropriate work steps are omitted here to simplify matters. The detailed description though can be found in those compiled and supervised works from [Toc10] and [Poe09].

Besides the need for simulation, efficient system design requires a large set of other tools to transfer an abstract application specification into a manufacturable layout. Due to the immense system complexity, it is of utmost importance to automate the design process as far as possible in order to increase productivity [Itr07b]. Therefore, this thesis suggests a possible approach in the following, whereas the explanation briefly describes how the specific requirements of NOC-based architectures can be incorporated into a standard design flow.

The adjusted design flow is presented in figure 5-4, whereas common and detailed work steps (e.g. repeated verification) are omitted for clarification. Starting point is a precise application specification that states the given conditions and intended demands. Based on this, the different tasks of the application are characterized in terms of their particular requisites for, amongst others, computation, data rate or memory. This allows partitioning the tasks into hardware and software components. Thereby, the partitioning relies on models from the NOC library, which describe the available on-chip modules. Once all modules are selected, the NOC architecture is configured (third step in figure 5-4). Hence, topology, resource mapping and specific parameters of NOC modules (e.g. FIFO depth) are determined at this point. Subsequently, simulations are carried out in order to compare the effective system characteristics with the demands of the application specification. In case the achieved results are not satisfying, the NOC-based architecture has to be improved so as to fulfill the specification. Such improvement necessitates iterating both configuration and simulation, which can thus stand for a decisive cost factor. However, since the simulations are based on a certain level of abstraction, the final NOC architecture has to be transferred to a manufacturable layout. Hence, the last design step is the NOC generation that uses the system description from the previous configuration and the implemented modules from the NOC library. Normally, the implemented modules are available in the form of a hardware

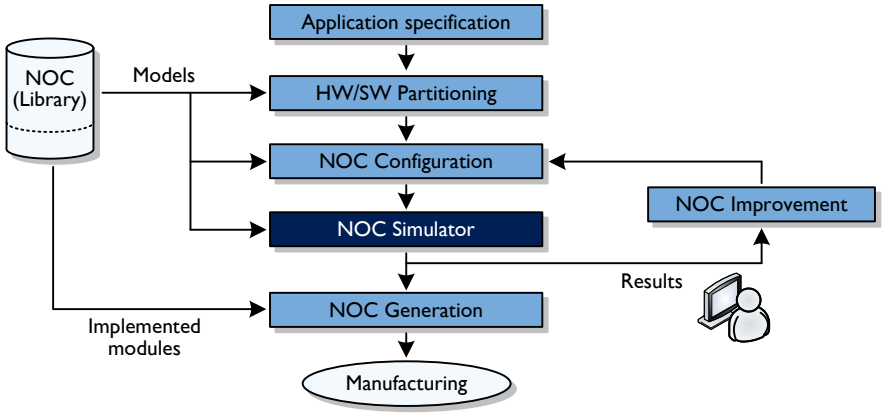

Figure 5-4 : Simplified representation of a design flow for Networks-On-Chip (NOC)

description language, a synthesized gate netlist or a final layout. Even though the NOC generation is sketched as a single step in figure 5-4, it commonly comprises a variety of elaborate tasks before the system can finally be manufactured – for instance, synthesis, floorplanning, place-and-route or RC extraction.

It should be pointed out that those late design steps can significantly impact system characteristics and design decisions across all abstraction layers. By way of example, the use of redundant vias or the physical wiring of the communication links can influence performance, power and reliability to a great extent [Lee10]. Concluding, abstraction indeed alleviates the design of complex systems, though it has to be based on accurate and comprehensive models to avoid design overhead or deficient system operation in the field.

5.2 Study I: Broadband packet processing

The best known example of distributed networking is the Internet, which stands out due to the tremendous amount of traffic that is constantly transmitted. As a result, an enormous quantity of packets has to be processed by various means to meet the requirements of both customers and providers. Moreover, data rates in the Internet continue to further increase because of demanding applications (e.g. IP-TV), a growing number of Internet users as well as new devices and technologies (e.g. 6LoWPAN) [Cof01, Hui08, Cio06a]. That implies that capable integrated systems are needed in order to handle the high amount of data, whereas Network-On-Chip (NOC) represents a suitable candidate for the design of such architectures. Therefore, this section describes the design of a packet processing chip that can be found in a broadband access network – in simple terms, a network between end customers and the Internet. The detailed correlations of the implementation are extracted from a longtime industrial cooperation and can be found in [Kub06], [Kub09], [Wid06a] and [Wid06b]. However, the primary objective is to demonstrate how the different design parameters are intertwined and that they have to be traded off against each other. This particularly concerns performance, power consumption and reliability.

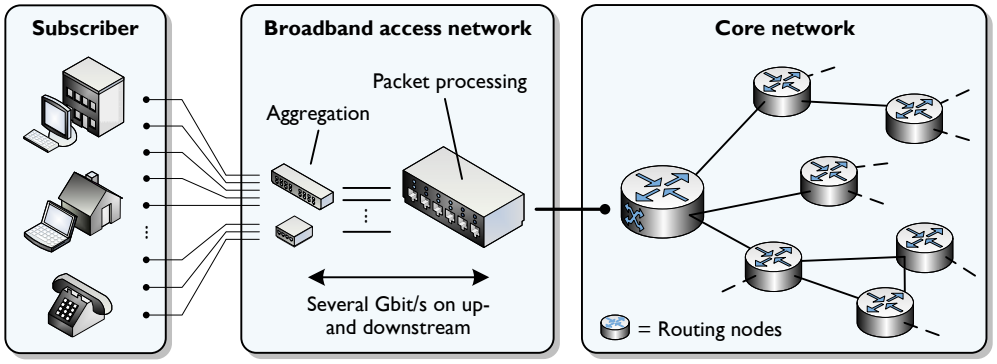

Figure 5-5 : Illustration of the application scenario, where the broadband access network connects a large number of customers (called subscribers) to the core network

5.2.1 Application scenario and system design

To begin with, the application scenario is described before the actual system design is explained in the second part of this subsection. Figure 5-5 illustrates the underlying application scenario whereas the **Broadband Access Network** (BAN) connects a multitude of customers (called subscribers) to the core network of a provider – and thus to the Internet. A subscriber can be any kind of customer who pays a fee in order to have access to the Internet and its associated services. Hence, very different types of data and interfaces originate from private and business subscribers. On that account, the first stage of the broadband access network aggregates the diverse connections to a common communication technology. This is basically a multiplexing of individual data streams onto a high-capacity transmission protocol – such as gigabit Ethernet. The second stage though inspects the data and provides different functionality of **packet processing**. Such functionality can comprehend amongst others authorization, authentication and accounting, security or quality of service. Due to the large number of connected subscribers, the packet processing has to be capable of handling several gigabit per second (Gbit/s) on up- and downstream. The BAN is located in relatively close proximity to the subscriber and enables to connect to the core network, which performs the actual routing of data over longer distances. Since performance plays a decisive role in the core network, more and more functionality is shifted into the BAN in order to relieve the core network from complexity.

In a nutshell, increasing data rates and functional complexity necessitate high-performance solutions for packet processing in the broadband access network. Furthermore, reliability and availability are additionally crucial because the packet processing can potentially mark a single point of failure for the attached subscribers.

The mentioned challenges of packet processing were addressed within an industrial cooperation. Thus, different functional components were already characterized in detail and available for the system design of this thesis [Kub09, Wid08, Kub06, Wid06a, Wid06b]. Thereby, the existing implementation (called MATMUNI) is based on a point-to-point connected data path and offers three essential functions:

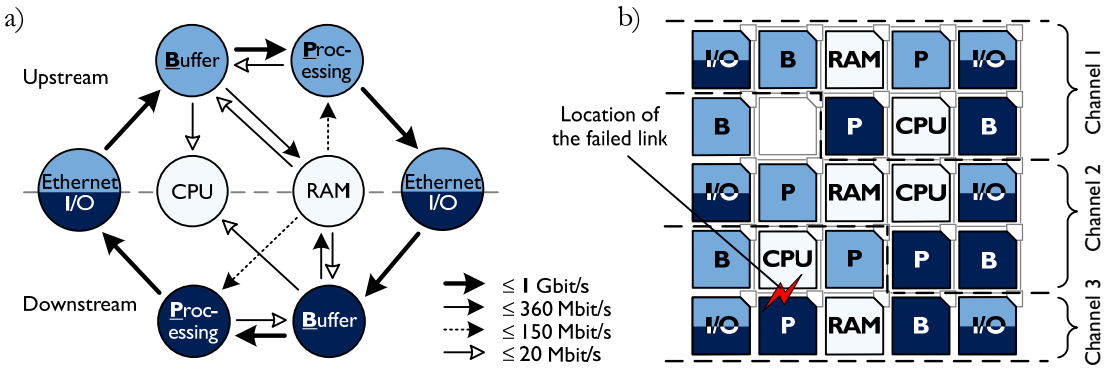

Figure 5-6 : a) Application graph for one bidirectional channel of packet processing with up- and downstream based on gigabit Ethernet b) Application mapping onto a 5x5 NOC architecture with three independent channels (with the faulty link for the simulation scenario)

- MAC address translation: The miscellaneous MAC addresses of the attached subscribers are translated to a distinct MAC address of the provider. This reduces address administration in the core network and attends to security aspects [Kub06].
- Traffic Management: The volume of data is metered for each subscriber. With this information at hand, arbitration policies engage to satisfy the diverse subscribers and their requirements [Wid06b, Kub06].
- Multi-Protocol Label Switching (MPLS): Data packets from the subscribers are encapsulated and supplemented with MPLS labels. While the labels are originally intended to forward packets, they are also used to transmit additional information −for instance, location information [Wid06a].

The functional components of packet processing (i.e. MATMUNI) are shown in figure 5-6 a) in terms of an application graph with the maximum data rates. The illustrated bidirectional channel consists of two data paths for up- and downstream, which are differently colored in the figure but work analogously. Besides, the stated data rates relate to Ethernet packets and reference to gigabit Ethernet interfaces. Based on this, the data flow is briefly explained in the following by means of the upstream. The Ethernet interface receives packets from the distributed off-chip network (i.e. from the subscribers) whereupon the packets are synchronized and segmented according to the needs of the on-chip communication architecture. Then the packets are sent to the Buffer (B) whereas the key information of the packets is extracted and forwarded to the memory (i.e. the RAM) in order to find an applicable rule for the packet processing. In case that no rule exists in the memory for a conveyed key information, the concerned packet is sent to the CPU where elaborate mechanisms can be executed in software −e.g. reconfiguring the existing rule set. However, normally both packet and rule are transmitted to the resource that executes the actual packet Processing (P) −refer to the stated functionality of the above enumeration. Finally, the processed packet is forwarded to the other Ethernet interface and then to the external network, which is the core network in case of the upstream (see also figure 5-5).

After the previous description of the application scenario, the actual system design is explained now. Thus, the application mapping of the introduced packet processing onto an architecture based on network-on-chip is depicted in figure 5-6 b). Starting point for the system design there is a 5x5 mesh network that accommodates 25 resources. Correspondingly, three independent channels can be implemented, which occupy a total of 24 of the available resources. Those three channels are indicated by the labels and the dashed auxiliary lines. According to figure 5-6 a), packets enter the system at the borders – on the left for the upstream, on the right for the downstream – and cross the architecture horizontally. While the resources operate completely independent and concurrently, some of the communication paths overlap as a matter of fact. Hence, packets on these paths have to compete for the router and link capacities.

Since the primary objective of this case study is to demonstrate how performance, power consumption and reliability are intertwined, three different versions of the architecture were implemented. All versions emanate from the pictured application mapping and the reference architecture (see also table 4-1), though with 32 bit data width and 4 slots FIFO depths. According to that, the first system design is called XY-32 as it is based on XY-routing and a data width of 32 bit. The second design only differs in the applied data width of 16 bit, and is thus termed XY-16. Finally, the third design employs a data width of 32 bit again, but together with source-based routing – it is named SB-32. In addition to these three designs, two different scenarios were simulated in order to evaluate the reliability. Hence, the first scenario assumes that the architecture is fully functional and faultless. By contrast, the second simulation scenario determines one link to suffer from a benign and permanent failure. Thereby, the faulty link is defined as the horizontal one that connects the resources at addresses (0,0) and (1,0) – see figure 5-6 b) where the location of the faulty link is plotted.

5.2.2 Comparison of system characteristics

The presented system characteristics in this subsection relate to post-layout data, whereas all systems were designed for a comparable operating frequency of 500 MHz. And the applied work loads picture different traffic traces from MATMUNI. As a start, the comparison addresses the faultless simulation scenario. For this purpose, the associated results are given in figure 5-7 as filled markers within the diagram. Thereby, the total power consumption of the on-chip network is plotted against the overall data rate for each design. Additionally, all results of the comparison herein are quantitatively and qualitatively summarized in table 5-2 at the end of this subsection. It should be noted that both the data rate in the diagram and the notion of performance in the table relate to the processed Ethernet traffic. By way of example, XY-32 achieves a data rate of 10.55 Gbit/s. This means that Ethernet packets with a total volume of 10.55 Gbit can be processed per second in this case, whereas the traffic is almost evenly split between the three channels of the architecture. More precisely, each channel carries about 1.76 Gbit/s on both up- and downstream – i.e. 3 channels with 2 streams and 1.76 Gbit/s each equals 10.55 Gbit/s. In fact, such performance exceeds the capacity of standard gigabit Ethernet interfaces, though this is neglected to compare the maximum ratings of the designs. While the stated data rates relate to

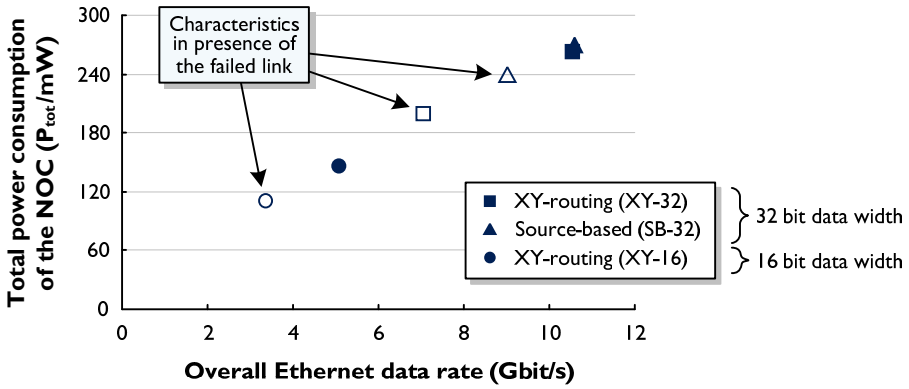

Figure 5-7 : Comparison of system characteristics for the application of NOC-based architectures in broadband packet processing (based on post-layout data)

the processing of Ethernet packets, the aggregated data rates inside the networks-on-chip are nearly four times higher. In case of XY-32, an aggregated on-chip data rate of 39.14 Gbit/s is required between the resources to achieve the 10.55 Gbit/s over the Ethernet channels.

The ranking of the three implemented designs in terms of the processed data rate is first and foremost contingent on the data width. This is because the operating frequencies, the packet lengths as well as the latencies for communication and computation are principally equal. Accordingly, XY-32 and SB-32 provide the best performance, whereas the source-based routing (i.e. SB-32) cannot supplementary benefit from its adaptive nature in such a static scenario. Furthermore, XY-16 reaches strictly speaking less than 50 % of the data rate compared to the two 32 bit versions. Although the data width is simply cut in half, the data rate of XY-16 is also hurt due to an increased relative overhead for the control information in each packet header – since the packet lengths in terms of flits are similar for all designs.

However, the lower data rate is traded off for lower power consumption. Consequently, XY-16 ranks best as regards power, whereas the total power consumption in figure 5-7 subsumes all links and routers of the communication networks. In concrete terms, this means that XY-16 dissipates 145.9 mW compared to approximately 265 mW for both 32 bit versions. Thus, the power savings equal merely 45 % since the power consumption is not directly proportional to the data width (see also subsection 3.3.3). In conclusion, against the background of performance and power, XY-32 and SB-32 exhibit eventually the better energy metrics. It should be pointed out though that source-based routing requires in effect additional overhead outside the communication network. For instance, the generation or storage of the switching directives is located in the resources and interfaces, respectively (see also section 4.1). Hence, it is not accounted for in the power figures here that solely refer to the routers and links. For this reason, SB-32 is qualitatively rated worse than XY-32 in table 5-2.

The results of the second simulation scenario relate to the setup with the faulty link and are given in figure 5-7 as framed markers. In this case, XY-32 and XY-16 are not capable to adapt to the failure since they are based on the deterministic XY-routing scheme. Therefore, the entire

Table 5-2 : Summary of the diverse results: Performance in terms of the overall Ethernet data rate, total power consumption of routers and links as well as reliability with respect to the remaining data rate in case of the given failure (Legend: + Good, ○ Neutral, − Bad)

		Performance (Gbit/s)	Power consumption (mW)	Reliability
32 bit	XY-routing	+ 10.55	○ 262.4	− 67 %
	Source-based	+ 10.57	− 269.4	○ 85 %
16 bit	XY-routing	− 5.07	+ 145.9	− 67 %

Ethernet traffic across channel 3 stalls because both up- and downstream rely on the faulty link. As the other two channels (i.e. channel 1 and 2) are not affected, the overall Ethernet data rate drops down to about 67 % of its original maximum – e.g. XY-32 decreases from 10.55 Gbit/s down to 7.06 Gbit/s. The stated relative proportion of the maximum data rate is also referred to in table 5-2 as a basic quantitative measure for the reliability of the different designs. In contrast to XY-32 and XY-16, the source-based routing of the design SB-32 facilitates to adapt to the failure. The affected routing paths are thereto relocated in order to bypass the faulty link. Thereupon, all three Ethernet channels can still be operated. Nonetheless, the overall data rate of SB-32 is reduced as well, even though only to 85 % of its maximum data rate (see figure 5-7 and table 5-2). The reason for the reduction is that the relocated paths of channel 3 interfere now with the routing paths of channel 2. Hence, both channels compete for the same network resources, which cannot satisfy the required combined data rates. That is why both channels evenly suffer from a loss of data rate.

The measure of reliability in table 5-2 hides one important aspect against the background of the application scenario. That is the distribution of data rates between the different channels of the designs. For example, in case of XY-32 and XY-16 all subscribers attached to channel 3 loose their access to the core network, which can entail serious consequences. By contrast, in case of SB-32 the subscribers attached to channel 2 and 3 exhibit a minor degradation of quality in the worst case. Lastly, since the total power consumption is closely connected to the network activity, it decreases similar to the overall data rate of the faulty scenario (see figure 5-7).

Concluding, this case study of broadband packet processing shows that performance and power consumption are intertwined. However, since nanotechnology is increasingly susceptible to failures, reliability is a major concern and crucial for both product lifetime and availability. Hence, reliability has to be thoroughly considered in conjunction with performance and power, whereas better reliability is mostly traded off against deteriorated performance or power dissipation. Therefore, the selection of the best suited design depends after all on the weighting of the individual characteristics and on the specific constraints of the application. In the future, novel measures will have to be introduced that express the quality of an integrated system with respect to the closely intertwined parameters of performance, power and reliability altogether.

5.3 Study II: Adaptive system management

While the previous section demonstrated how system characteristics are intertwined, this case study aims at exemplifying how design decisions and system operation impact various abstraction layers. Starting point is the approach of adaptive system management as it is becoming common practice in nowadays integrated circuits in order to preserve power – such as Intel SpeedStep or VIA PowerSaver [Int04]. However, such solutions do not consider reliability and the special demands of complex Networks-On-Chip (NOC). Those additional demands comprehend for instance the spatial distribution of computation and communication, the available implicit redundancy or diverse types of failure causes. Therefore, the following subsection introduces the setup of a complex system with adaptive system management for NOC running a multimedia application. The subsequent evaluation in subsection 5.3.2 places particular emphasis on the temperature distribution during operation. Based on these results, it is demonstrated how temperature impacts design decisions and system operation across various abstraction layers.

5.3.1 System setup

The underlying application of this case study is a simple example from the multimedia domain, which is often referenced to in published NOC implementations [Var04, Ber04, Ber07, Sal09, Lu05b, Lee07]. Together with the system management it is, strictly speaking, two applications that are illustrated with their application graphs in figure 5-8 a). Since flexibility is a crucial factor for the system management, it is most advantageous to implement it in software. Hence, the management is represented by the simple communication scheme of the CPU and the memory (i.e. the RAM). The actual multimedia application here is a Motion JPEG (M-JPEG) encoder that compresses a video stream by processing each frame of the video separately [Iso03]. Thereby, the exact details for the design were derived from published implementations [Oka97, Sun03, Lu05b]. Apart from those intricacies, the data flow of a video stream can be described by three main steps: Discrete Cosine Transform (DCT), Quantization (Q) and Encoding (E). In fact, the given application graph in figure 5-8 a) comprises two streams of such work steps that process diverse parts of the video concurrently − the streams are differently colored in the figure.

However, the original, raw video is received by the I/O interface whereupon color space conversion and downsampling can optionally be performed. Thereafter, blocks consisting of 8x8 pixels from a single video frame are sent to the DCT. The DCT transforms the given set of pixels into a block of 8x8 DCT coefficients, whereas the calculation represents the performance bottleneck of the M-JPEG encoder. Hence, it sets an upper limit for the throughput of the video stream. The following Quantization (Q) divides each DCT coefficient by a specific constant. This conforms to a lossy compression, which exploits the peculiarities of human visual perception [Sch09]. The last step is the Encoding (E) that converts blocks of 8x8 values into data sets of varying but reduced size. To be precise, the encoding itself involves three successive tasks to compress the incoming data: the zig-zag operation as well as run-length and Huffman encod-

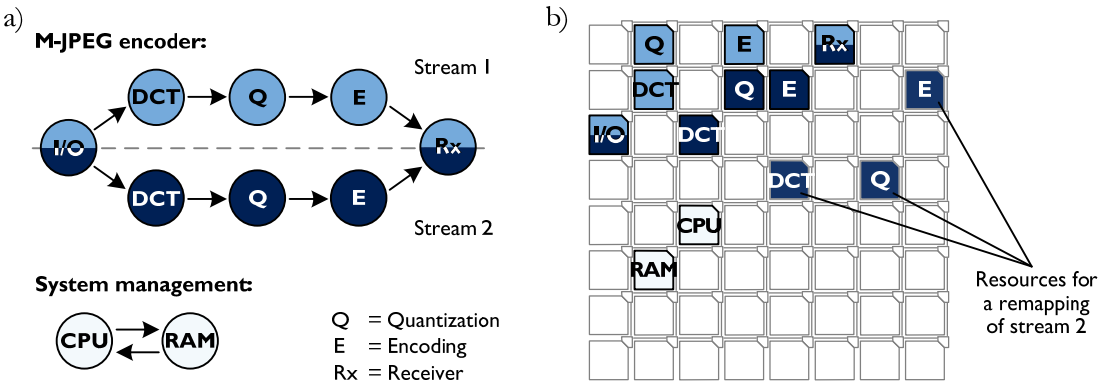

Figure 5-8 : a) Underlying application graphs of the system management and the M-JPEG encoder with two parallel data streams b) Application mapping onto an 8x8 NOC architecture (including the indicated resources for a later remapping of stream 2)

ing [Iso03]. Finally, the Receiver (Rx) merges the two streams and provides the compressed video for further use – such as storage or off-chip transmission.

For the actual implementation of this case study, the M-JPEG encoder was assumed to be a contingent application within a general purpose processor. Therefore, the design herein emanates from the reference NOC architecture – note table 4-1 and figure 5-8 b). The network size though comprehends 8x8 resources and the routers apply a source-based routing scheme with 32 bit data width and 4 slots FIFO depth. Furthermore, each router is supplemented with a control unit, which is connected via an additional interface as introduced in subsection 4.4.2. Whereas the control units of the investigation here are capable of measuring temperatures and setting the frequencies of the resources (as part of DFS). Lastly, the system management is modeled in order to operate the architecture and its applications. This adaptive system management is in principle based on the suggested concepts of the Error Resiliency Layer (ERL) and the Service-Oriented Architecture (SOA) – the corresponding details can be found in subsection 4.4.2 and section 4.5, respectively figure 4-24. Note that the system management is only emulated to execute the essential tasks, thus it is not fully implemented due to the immense complexity. For the same reason, the resources are not implemented but their traffic behavior is reproduced by synthetic traffic patterns (see also figure 5-2). Correspondingly, the size and power dissipation of the resources is gathered from Intel's 80-core chip as the input for the applied temperature model – see also figure 5-3 [Hos07, Van07]. Despite the abstractions, the communication network itself still comprehends a complexity of nearly 400 000 gates.

The 8x8 NOC architecture is illustrated in figure 5-8 b) with the used application mapping for the system management and the M-JPEG encoder. Thereby, it is assumed that all tasks of figure 5-8 a) can individually be mapped onto different resources of the NOC-based architecture. However, this does not mean that the diverse tasks can be mapped onto every resource. For instance, consider that the I/O interface or the RAM require physically determined off-chip connections (i.e. pins). Hence, such resources (respectively tasks) are fixed to specific locations

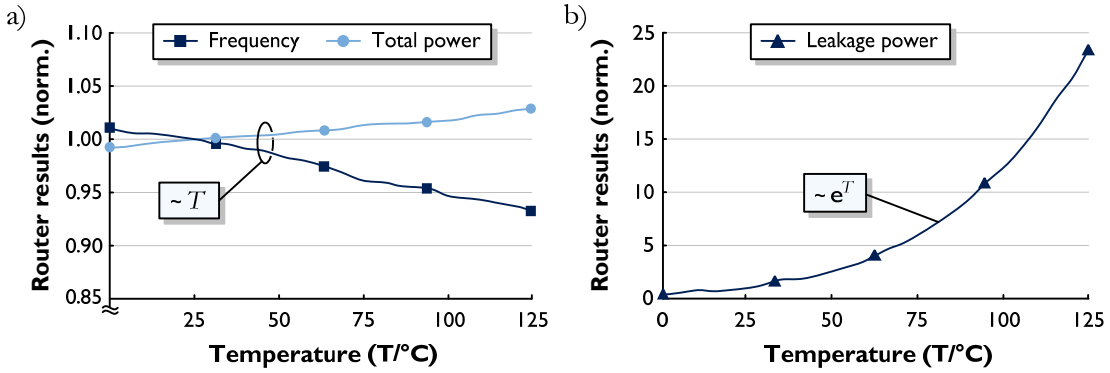

Figure 5-9 : Post-layout data of a single router from the above system setup against the operating temperature −results are normalized to those values at 25 °C, to be precise f = 500 MHz, P_{tot} = 4327 µW and P_{leak} = 2334 nW

within the architecture. Accordingly, the system management with its CPU and RAM is mapped onto the resources at addresses (2,3) and (1,2). Similarly, the tasks of the M-JPEG encoder are mapped between the I/O interface and the receiver (Rx) in the upper left corner of the architecture. Whereas both data streams are located in close proximity to each other, which facilitates small communication distances.

5.3.2 Distribution of temperature during operation

Sections 2.1 to 2.3 described that temperature can critically affect performance, power consumption as well as reliability. In fact, the further scaling of technology even exacerbates temperature-related concerns. This is because appropriate heat removal remains roughly constant in the foreseeable future while the increasing transistor count still raises the power density [Itr07g, Itr07a]. Therefore, adaptive management plays a prospective key role in order to control on-chip temperature and its impact on system characteristics [Itr07a]. This crucial requirement is addressed in the following by examining the dynamic distribution of temperature during system operation.

However, before investigating the implemented NOC architecture with the adaptive system management, a closer look is taken at fundamental parameters in order to verify and quantify the influence of temperature. For this purpose, different parameters of a router are plotted in figure 5-9 against the temperature −whereas the results are normalized to those values at 25 °C. All results are based on post-layout data of a single router from the introduced setup in subsection 5.3.1. For a start, the router frequency is sketched in figure 5-9 a) as a measure of performance since the frequency vitally determines the data rate and latency of the router (see equation 20). In any case, the frequency is linearly dependent on the temperature T and differs with about 8 % across the plotted range, whereas the frequency decreases for higher temperature. Even though this deviation seems small, it entails one of two significant consequences. Either the router has to be run at the lowest potential frequency to account for the worst case condition of

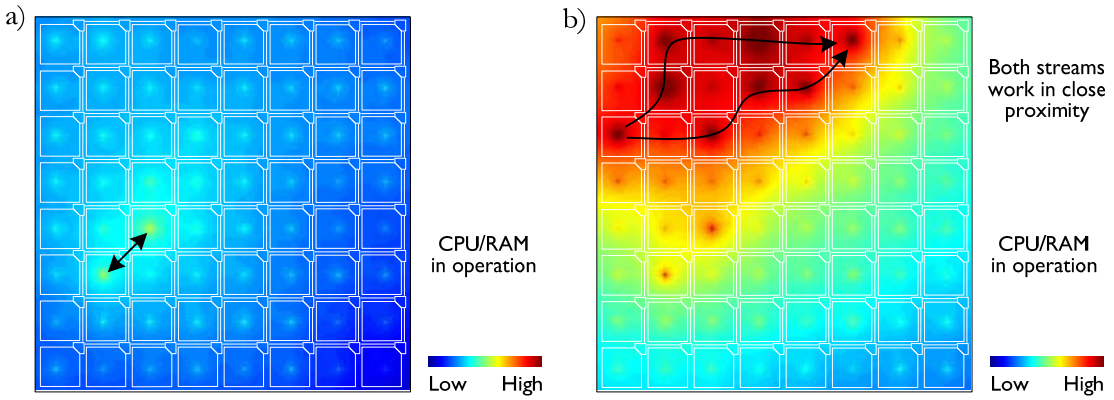

Figure 5-10 : Illustration of the temperature distribution during operation: a) Solely the system management is executed b) System management and M-JPEG encoder run in parallel, whereas both streams of the encoder work in close proximity in the upper left area

the temperature, which equals a loss in performance. Or an increasing number of timing failures have to be dealt with, which primarily aggravates the power dissipation to deal with them.

Similar to the frequency, the total power consumption in figure 5-9 a) is affected by temperature as well, although the power increases with temperature in this case. The rise of the total power though does in general only slightly impact system operation as long as given design limits are not exceeded – e.g. with regard to the external power supply or the on-chip supply network. Figure 5-9 b) additionally depicts the leakage power of the router. Since the router was simulated with heavy traffic (i.e. high activity), the leakage here only represents a small portion of the total power consumption. However, the leakage power exhibits an exponential increase with temperature, which can be particularly crucial for components with long idle times. Lastly, the reliability of the router is not pictured as a function of temperature because the underlying technology does not provide an appropriate measure. Based on the Arrhenius model and equation 15, one could say though that reliability worsens with increasing temperature [Wan08, Cro01]. In summary, higher and unbalanced on-chip temperatures impair performance, power consumption as well as reliability. Hence, an adaptive system management that monitors and controls temperature can facilitate to avoid thermal hot spots and other related concerns (e.g. temperature cycling and mechanical stress).

This being said, the distribution of temperature during operation of the introduced system is presented now, which was obtained by running a functional simulation and by using the developed temperature model that exploits the duality of the electrical and thermal model (see figure 5-3). Accordingly, figure 5-10 and figure 5-11 show selective points in time for the temperature of the architecture. Thereby, the blue end of the scale denotes low temperatures (>62 °C) and the red end marks high temperatures (up to 103 °C). At the beginning of this case study, the system is in an idle state and solely the system management with CPU and RAM is in operation – illustrated in figure 5-10 a). Thus, the on-chip temperature is rather low and about evenly distributed across the architecture. A small elevation can merely be recognized for the

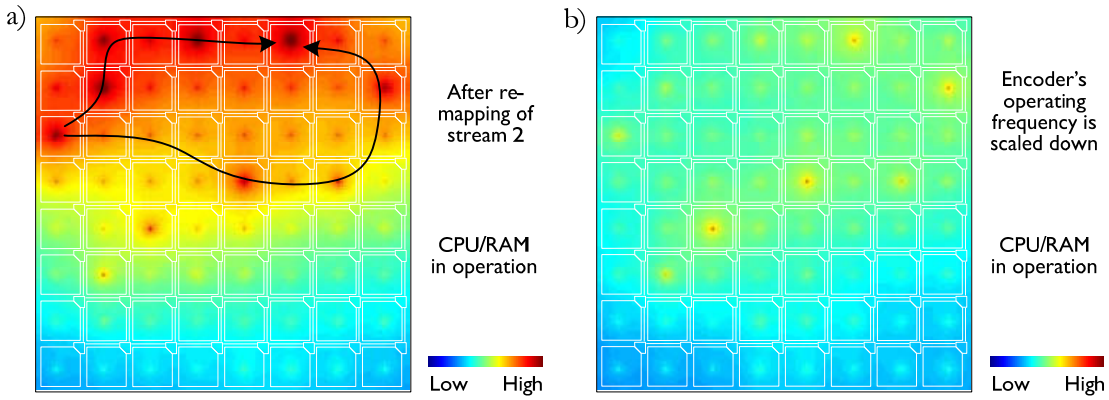

Figure 5-11 : Second illustration of the temperature distribution during operation: a) Stream 2 of the M-JPEG encoder is remapped to balance thermal impact b) Scaling down of the operating frequency of the encoder to restrict thermal concerns as a whole (whereas $f = 0.6 \cdot f_{max}$)

resources of the system management. However, once the M-JPEG encoder starts to operate, two effects become apparent in figure 5-10 b). On the one hand, a significant increase in temperature develops in the upper left corner of the chip area where both streams of the encoder work in close proximity to each other. On the other hand, particular hot spots arise at those resources that contribute to the encoding of the video – compare that with the mapping in figure 5-8 b).

While the close proximity of the running resources keeps the communication distances small, it also entails elevated power densities in the concerned areas, and thus aggravated thermal hot spots. As a result, several local temperatures exceed a determined upper threshold. This is observed by the distributed control units, which in turn send control packets to the adaptive system management to report the condition – consider subsection 4.4.2 and the related multistage approach of ERL. Thereupon, the management exploits its global system awareness and the available redundancy of the NOC-based architecture to react appropriately. In the present case, the lower stream (i.e. stream 2) is remapped onto idle resources a little further away – as sketched in figure 5-8 b). In fact, the remapping increases the communication distances among the resources, but this is negligible since the M-JPEG encoder is focused on throughput and not on latency. However, the separated data streams of the encoder lead to a more evenly distributed temperature and reduced thermal hot spots – examine the differences of figure 5-10 b) and figure 5-11 a). More precisely, both the power densities and the maximum on-chip temperature are noticeably decreased.

In the event that certain temperatures still exceed the upper threshold or that lower encoder throughput also serves the application requirements, the adaptive management can additionally scale down the operating frequencies of the involved resources (i.e. DFS). The impact of such scaling is shown in figure 5-11 b) whereas the clock frequency f of the M-JPEG encoder is reduced to 60 % of the maximum frequency f_{max}. Thus, the temperature profile is similar to figure 5-11 a), but with a greatly reduced maximum temperature and relatively small temperature variations across the architecture. This advancement comes at a price though because in contrast

to the preceding remapping, the scaling of operating frequencies impairs the performance of the encoder as regards data throughput. To be precise, the throughput of encoded video frames is cut down by 40 % – due to the clock frequency $f = 0.6 \cdot f_{max}$. Hence, application requirements and temperature control have to be carefully traded off against each other.

Even though the temperature values here are not fully adjusted, the case study nonetheless reveals the dynamic behavior of temperature and the significance of an adaptive system management. It can be strikingly deduced thereof that design decisions and system operation affect various abstraction layers, so that individual layers can rarely be regarded separately [Itr07a]. A couple of examples are emphasized now to prove the previous statement. First of all, temperature affects wire and transistor delay (see equations 4 and 24 or figure 5-9). Therefore, such low-level delay variations have to be dealt with on gate or module level [Mit05]. Against the background of complex networks-on-chip, these effects actually concern the system level too, since the overall operation has to be synchronized and adapted between the distributed resources. Furthermore, the selected type of dealing with temperature-related failures on lower abstraction layers (e.g. on technology or gate level) also influences design decisions as regards the communication architecture. For instance, the number of residual failures determines if error correction is better implemented on a hop-to-hop or on an end-to-end basis [Mur05b, Jan05].

Lastly, several examples specifically relate to the design and operation of the implemented system setup. The monitoring of the distributed control units is meaningless if the triggered control packets (i.e. the status reports) are not evaluated in another abstraction layer. The other way round, the control units have to serve as actuators for commands from the high-level management in order to influence system behavior and physical parameters (see also subsection 4.4.2). Consequently, global issues – such as temperature, power consumption or wearout – can effectively be tackled when high-level management and low-level approaches exploit their mutual advantages. A final example here for the intertwined abstraction layers is the source-based routing scheme. While its use appears beneficial when solely routers and links are contemplated, additional efforts are required on other abstraction layers. By way of example, on system level where the generation or storage of the switching directives is carried out (see also section 4.1).

Concluding, an adaptive system management can effectively control on-chip temperatures. It is suited though to accomplish other global assignments to the same extent – as for example application mapping, load balancing or reliability management. However, a high-level system management can only be capable when it is aware of as well as exploits the properties of the underlying abstraction layers. Similar to the management during operation, individual decisions at design time also influence diverse abstraction layers (e.g. recall the routing scheme). Therefore, particular abstraction layers can rarely be regarded separately because they are actually closely intertwined. In the future, great efforts will be required to develop mechanisms and models that provide appropriate levels of abstraction without neglecting physical details or the correlations among diverse abstraction layers.

Chapter 6

Conclusion and Outlook

Over the last decades, continuous scaling of technology has been the key to success in order to meet the increasing consumer expectations on performance and flexibility. However, current nanotechnology of integrated circuits is about to approach definite physical limits. This trend is associated with a growing number of serious issues that derogate performance, power consumption as well as reliability. Furthermore, the boosting complexity of integrated systems also constrains the design productivity. Against this background, Network-On-Chip (NOC) is the emerging design paradigm to overcome the present challenges of technology and conventional system architectures. Hence, this thesis makes a significant contribution to the improved development of complex integrated systems based on NOC in order to tackle existing concerns. Thereby, the essential approach is to perceive the integrated system as a whole. This facilitates to address as well as to exploit the intertwined design parameters and abstraction layers, whereas particular interest is aimed at the trade-off between performance, power and reliability.

As a start, the main challenges of integrated systems were introduced, which establish the basis for the subsequent, elaborate investigations. For this purpose, several convenient and novel classifications were compiled that group both the causes and potential concepts to resolve the associated issues. Although the parameters of performance, power and reliability were described in separate sections, their correlations among each other were especially emphasized. As one of the findings, temperature was identified as a crucial factor because it affects the mentioned parameters likewise, and because it exhibits a decisive dynamic, spatial and temporal behavior. Lastly, different architectural options for system communication were presented, whereas the specifically developed analytical comparison proves the significance of networks-on-chip for the design of complex integrated systems.

The actual investigations were carried out on the basis of a current 65 nm technology from STMicroelectronics and pursued a gradual implementation of the entire system. Thus, the basic

components of on-chip networks were designed first. In the process, the importance of wires and their influence on link characteristics was worked out. Moreover, miscellaneous routers were implemented, whereupon the benefits of diverse design alternatives were evaluated (e.g. of flow control or FIFO depth). Based on these results, several improvements were derived. For instance, the fine-grained approach to gate the clock of the router reduces the dynamic power consumption by more than 90 %. Besides, the enhanced router layout decreases the area costs of the communication network by about 50 % without impairing other communication characteristics. In any case, the large variety of originated components enables to assemble on-chip networks with very different demands.

In line with the objectives of this thesis, NOC-based systems as a whole were thereupon investigated, which comprises the architecture itself and the necessary algorithms to operate it. The achieved improvements primarily exploit the mutual advantages of the diverse and intertwined abstraction layers. By way of example, the application of local traffic increases the performance of the proposed clustered topologies substantially – in case of the Cluster (3-5) topology from 20.45 GB/s to 100.13 GB/s. This makes clustered topologies a highly beneficial approach, especially for energy and area-constrained designs. Another major outcome is the demonstrated, distinguished combination of low-level functionality and high-level management. Thereby, the distributed control units, the multistage concept of the Error Resiliency Layer (ERL) as well as the service-oriented system management represent considerable contributions for prospective advancements.

Finally, two exemplary case studies of complex systems clarify the relevance of the compiled objectives. In order to accomplish the implementation of those complex systems, necessary requirements for the efficient design and simulation were established first. This involved the description of the developed functional simulator and its integration into a possible design flow. In this context, an enhanced model was also developed that allows simulating the dynamic distribution of temperature during operation of a NOC-based system. With these tools at hand, the first case study picks up an application scenario for broadband packet processing. The obtained results show that the examination of sole design parameters can lead to deceptive and erroneous design decisions. In addition, reliability cannot be further neglected when evaluating the quality of complex integrated systems. Therefore, the parameters of performance, power consumption and reliability have to be carefully traded off against each other. The second case study relates to the need for adaptive system management. Based on the dynamic distribution of temperature during operation, it is evidenced that decisions at design and run-time essentially impact several abstraction layers. Accordingly, it is not only required to account for the correlations between the layers, but it is also highly beneficial to exploit their mutual advantages.

Besides, the acquired results and concepts constitute the prerequisites for further research and prospective advancements. By way of example, it seems promising to investigate to what extent the developed temperature model can improve the design process of NOC-based architectures because temperature-related issues can early be considered. Similarly, an adaptive system management can purposefully be developed that handles the miscellaneous challenges as regards

temperature. Furthermore, since nanotechnology is becoming increasingly susceptible to failures during manufacturing and operation, a growing number of aggravated and novel concerns have to be dealt with. Thereby, the multistage concept of the Error Resiliency Layer (ERL) serves as a capable approach for implementations that are aimed at the interaction between efficient hardware solutions and a flexible system management in software.

In the end, this thesis also points out unresolved questions, whereas two of the most important are recapitulated here. First, the results within the academic community can hardly be compared due to the immense space of influencing parameters. Therefore, it necessitates accepted benchmarking methods to fairly access proposed solutions for networks-on-chip. Such benchmarks will have to be geared towards different application domains and have to include aspects of reliability. Second, albeit the relevance of failures is increasingly acknowledged, there is no convenient measure to express the quality of a design taking reliability into account. With the definition of the Energy-Reliability Ratio (ERR), an initial proposal was made in this thesis. However, it will require great efforts to refine and to enforce methods that determine reliability in miscellaneous complex integrated systems.

List of figures

List of tables

Abbreviations and symbols

Abbreviation		Explanation
6LoWPAN	–	IPv6 over Low-power Wireless Personal Area Network
B	–	Resource that buffers Ethernet frames
BAN	–	Broadband Access Network
BIST	–	Built-In Self-Test
CMOS	–	Complementary Metal Oxide Semiconductor
CPU	–	Central Processing Unit
CRR	–	Core-to-Router Ratio
DCT	–	Discrete Cosine Transform
DEMUX	–	Demultiplexer
DFM	–	Design For Manufacturing
DFS	–	Dynamic Frequency Scaling
DPWS	–	Devices Profile for Web Services
DRC	–	Design Rule Check
DRM	–	Dynamic Reliability Management
DVS	–	Dynamic Voltage Scaling
E	–	Resource that encodes parts of video frames
EMI	–	Electro-Magnetic Interference
ERL	–	Error Resiliency Layer
ERR	–	Energy-Reliability Ratio
ESD	–	Electro-Static Discharge
FIFO	–	First In First Out (refers to storage elements)
Flit	–	Flow control unit
FPGA	–	Field Programmable Gate Array
High-V_{th}	–	High Threshold Voltage (refers to transistor and gate types)
I/O	–	Input/Output interface
IC	–	Integrated Circuit
ICT	–	Information and Communication Technologies
IEEE	–	Institute of Electrical and Electronics Engineers
IP	–	Intellectual Property
IP-TV	–	Internet Protocol Television

IPv6	–	Internet Protocol version 6
ITRS	–	International Technology Roadmap for Semiconductors
Low-V_{th}	–	Low Threshold Voltage (refers to transistor and gate types)
LVS	–	Layout Versus Schematic
MAC	–	Media Access Control address
M-JPEG	–	Motion Joint Photographic Experts Group
MOS	–	Short form for MOSFET
MOSFET	–	Metal Oxide Semiconductor Field Effect Transistor
MPLS	–	Multi-Protocol Label Switching
MTTF	–	Mean Time To Failure
MUX	–	Multiplexer
NBTI	–	Negative-Bias Temperature Instability
nMOS	–	n-type transistor (MOSFET)
NOC	–	Network-On-Chip
On/Off	–	On-off flow control scheme
OS	–	Operating System
P	–	Resource that executes packet processing of Ethernet frames
PCR	–	Port-to-Core Ratio
PDF	–	Probability Density Function
PDP	–	Power-Delay-Product
Phit	–	Physical unit
PMF	–	Probability Mass Function
pMOS	–	p-type transistor (MOSFET)
Q	–	Resource that quantizes parts of video frames
QoS	–	Quality of Service
RAM	–	Random Access Memory
Req/Ack	–	Request-acknowledge flow control scheme
RF	–	Radio Frequency
RMI	–	Remote Method Invocation
RTL	–	Register Transfer Level
Rx	–	Resource that receives a compressed video stream
SOA	–	Service-Oriented Architecture
SOC	–	System-On-Chip
SPICE	–	Simulation Program with Integrated Circuit Emphasis
Standard-V_{th}	–	Standard Threshold Voltage (refers to transistor and gate types)
TDDB	–	Time-Dependent Dielectric Breakdown
UPnP	–	Universal Plug and Play
VHDL	–	Very high speed integrated circuit Hardware Description Language

Symbol		Explanation
α	–	Activity factor
α_{sat}	–	Saturation point of a topology (in percent of the theoretical maximum)
α_{util}	–	Degree of utilization
β	–	Parameter of the Weibull distribution
ε_{di}	–	Permittivity of the dielectric adjacent to a wire
ε_{ox}	–	Permittivity of the gate oxide
λ	–	Failure rate
λ_i	–	Failure rate of the i-th component
λ_P	–	Mean and variance of the Poisson distribution
λ_S	–	Failure rate of a system with all components in series (equivalent λ_{S1}, λ_{S2}, ...)
λ_{topo}	–	Failure rate of a topology
μ	–	Mean of the Gaussian distribution
μ_0	–	Mobility of charge carriers
ϱ	–	Electrical resistivity
σ	–	Standard deviation of the Gaussian distribution
τ	–	Time constant
A	–	Area (equivalent A_1, A_2, ...)
B_B	–	Bisection bandwidth
B_c	–	Link count in the smallest bisection of a topology
B_L	–	Bandwidth of a bidirectional link (equivalent B_{L_1}, B_{L_2}, ...)
c	–	Capacitance per unit length
$C(N_1,N_2)$	–	Cut of the network
C_{el}	–	Electrical capacitance
C_{fringe}	–	Wire capacitance due to fringing fields
C_{hor}	–	Wire capacitance between two horizontally separated wires
c_{hor}	–	Wire capacitance per unit length between two horizontally separated wires
C_{load}	–	Load capacitance
C_{mos}	–	Combined capacitances of MOS transistors (e.g. diffusion, gate capacitance)
C_{th}	–	Thermal capacitance
C_{ver}	–	Wire capacitance related to two vertically separated layers
c_{ver}	–	Wire capacitance per unit length related to two vertically separated layers
C_{wire}	–	Wire capacitance
$d(n_s,n_d)$	–	Communication distance from n_s to n_d in terms of links/hops

$\overline{d}_{avg}$ – Average distance from sender to receiver in terms of links/hops
d_{max} – Diameter of a topology in terms of links/hops
DR_{agg} – Aggregated data rate
d_{res} – Width of the resources (respectively the height for a quadratic shape)
DR_{ideal} – Ideal aggregated data rate
d_{router} – Communication distance in terms of traversed routers
$\overline{d}_{router}$ – Mean communication distance in terms of traversed routers
DR_{res} – Data rate of a resource
D_x – Destination node of a communication (equivalent D_1, D_2, D_3, ...)
E_a – Activation energy of a failure mechanism
f – Clock frequency
f_{max} – Maximum frequency
h – Wire height
I – Electric current
$I(t)$ – Supply current at time t
I_{diode} – Reverse-biased saturation current of a diode
I_{ds} – Drain-source current
I_{gate} – Gate oxide current
I_{pn} – Junction current (between p-type and n-type areas)
I_{sc} – Short circuit current
I_{sub} – Subthreshold current
k – Event of a discrete random variable
k_{Boltz} – Boltzmann's constant (8.62·E-5 eV/K)
k_{fm} – Empirical constant of the Arrhenius failure model
k_p – Auxiliary parameter (equivalent k_{p1}, k_{p2}, k_{p3}, ...)
k_{par} – Portion of the application that can be executed in parallel
k_{vr} – Constant based on the voltage range within the RC model
L – Gate length
l – Wire length
L_{Bus} – Total link length of a bus
L_{NOC} – Total link length of a network-on-chip
L_{WC} – Worst case link length in a topology
l_x – Link in the communication network (equivalent l_1, l_2, l_3, ... and l_a, l_b, ...)
M – Number of segments for repeater insertion
M_{act} – Number of active participants that can send a message in parallel
n – Network size (i.e. $N_{res} = n^2$ in a quadratic arrangement of a mesh)
N – Set of all communication nodes (generally the resources)
n_d – Destination node (generally a resource)
N_{link} – Number of links
N_{module} – Number of modules
N_{port} – Number of ports

N_{res} – Number of resources (generally equals $|N|$)
N_{router} – Number of routers
n_s – Sending node (generally a resource)
N_x – Subset of the communication nodes N (equivalent N_1, N_2, ...)
P – Dissipated power of a heat source
$P(t)$ – Instantaneous power at time t
$P_{\square}$ – Power density per unit area
$\overline{P}_{avg}$ – Average power over a given time interval
P_{dyn} – Dynamic power consumption
P_{glitch} – Power consumption due to glitches
P_{leak} – Leakage power
P_{link_i} – Power consumption of the i-th link
P_{module_i} – Power consumption of the i-th module
P_{sc} – Short circuit power
P_{static} – Power consumption due to static current
P_{tot} – Total power consumption
Q – Charge
r – Resistance per unit length
$R(t)$ – Probabilistic reliability at time t
$R_{\square}$ – Sheet resistance
R_{el} – Electrical resistance
$R_i(t)$ – Reliability of the i-th component at time t
R_{on} – On-state resistance of a transistor
R_{path} – Routing path as an ordered set of links
R_{route} – Routing relation
$R_S(t)$ – Reliability of a system with all components in series (equivalent $R_{S1}(t)$, ...)
R_{th} – Thermal resistance
R_{wire} – Wire resistance
S – Scaling factor for physical dimensions
s – Spacing of wires in the same metal layer
S_{Amdahl} – Speedup of performance as expressed by Amdahl's law
S_x – Source node of a communication (equivalent S_1, S_2, S_3, ...)
T – Temperature
$t_{0\rightarrow1}$ – Signal delay across a link for a rising slope at the output
$t_{1\rightarrow0}$ – Signal delay across a link for a falling slope at the output
t_0 – Start time of the useful lifetime of a component
t_d – Delay time
T_{di} – Thickness of the dielectric between vertically separated layers
t_{FF} – Delay induced by flip-flops
t_{grant} – Contention delay because of missing grant to blocked network resources
t_{kB} – Time for the average transfer of one kilobyte

t_{link}	–	Average delay when passing along a link
$t_{link,rep}$	–	Link delay with repeaters
t_{MOS}	–	Delay time of a transistor
T_{ox}	–	Gate oxide thickness
t_{packet}	–	Packet delay
$\overline{t}_{packet}$	–	Mean packet delay
t_{rep}	–	Delay of a repeater
t_{router}	–	Router delay (i.e. the delay for processing the packet header)
t_{sc}	–	Period of time in which a short circuit current flows
t_{seg}	–	Delay of a segment (subsumes repeater and wire delay)
t_{serial}	–	Serialization delay due to the sequential injection of flits into the network
t_{signal}	–	Signal delay per unit length
t_{skew}	–	Delay due to clock skew
t_{wire}	–	Delay of a wire
t_x	–	Point in time (equivalent t_{x1}, t_{x2}, t_{x3}, ...)
U	–	Scaling factor for voltages
V	–	Voltage
V_{dd}	–	Supply voltage
V_{ds}	–	Drain-source voltage
V_{gs}	–	Gate-source voltage
V_{pn}	–	Voltage across a diode
V_{th}	–	Threshold voltage
W	–	Gate width
w	–	Wire width
$W_{channel}$	–	Channel width
W_{data}	–	Data width of a link (respectively a bus)
x	–	Continuous random variable of the Gaussian distribution
x,y	–	Pair of coordinates in the two-dimensional mesh network

References

[Aga03] A. Agarwal, D. Blaauw, and V. Zolotov, "Statistical timing analysis for intra-die process variations with spatial correlations," *International Conference on Computer-Aided Design (ICCAD)*, 2003, pp. 900-907.

[Aga09] A. Agarwal, C. Iskander, and R. Shankar, "Survey of Network on Chip (NoC) architectures & contributions," *Journal of Engineering, Computing and Architecture*, vol. 3, 2009, pp. 15-29.

[Akg06] B. Akgul, L. Chakrapani, P. Korkmaz, and K. Palem, "Probabilistic CMOS technology: A survey and future directions," *Intern. Conf. on Very Large Scale Integration and System-On-Chip (VLSI-SOC)*, 2006, pp. 1-6.

[Akt02] C. Aktouf, "A complete strategy for testing an on-chip multiprocessor architecture," *IEEE Design & Test of Computers*, vol. 19, 2002, pp. 18-28.

[Ala07] A. Alaghi, N. Karimi, M. Sedghi, and Z. Navabi, "Online NoC switch dault detection and diagnosis using a high level fault model," *IEEE Intern. Symp. on Defect and Fault-Tolerance in VLSI Systems (DFT)*, 2007, pp. 21-29.

[Ald99] P. Aldworth, "System-on-a-chip bus architecture for embedded applications," *International Conference on Computer Design (ICCD)*, 1999, pp. 297-298.

[Ali07] M. Ali, M. Welzl, S. Hessler, and S. Hellebrand, "An end-to-end reliability protocol to address transient faults in network on chips," *Design, Automation and Test in Europe (DATE), Workshop on Diagnostic Services in Network-on-Chips*, 2007, pp. 141-145.

[All02] A. Allan, D. Edenfeld, W. Joyner, A. Kahng, M. Rodgers, and Y. Zorian, "2001 Technology roadmap for semiconductors," *Computer*, vol. 35, 2002, pp. 42-53.

[Amd05] AMD, "Taking it to the next level: The AMD Geode LX800@0.9W processor," *From AMD (White paper)*, 2005.

[Amd67] G. Amdahl, "Validity of the single processor approach to achieving large scale computing capabilities," *Proc. of the joint computer conference (AFIPS)*, 1967, pp. 483-485.

[And03] A. Andriahantenaina and A. Greiner, "Micro-network for SoC: Implementation of a 32-port SPIN network," *Design,Automation and Test in Europe (DATE)*, 2003, pp. 3-4.

[Ani03] M. Anis and M. Elmasry, *Multi-threshold CMOS digital circuits: Managing leakage power*, Springer, 2003.

[Anj95] K. Anjan and T. Pinkston, "An efficient, fully adaptive deadlock recovery scheme: DISHA," *International Symposium on Computer Architecture (ISCA)*, 1995, pp. 201-210.

[Arm07] ARM, "Embedded trace macrocell − Architecture specification," 2007. Available: http://www.arm.com/

[Art05] Arteris, "A comparison of network-on-chip and busses," *Arteris report (White paper)*, 2005. Available: http://www.arteris.com/

[Asc05] G. Ascia, V. Catania, and M. Palesi, "Mapping cores on network-on-chip," *Intern. Journal of Computational Intelligence Research*, vol. 1, 2005, pp. 109-126.

[Bac82] G. Baccarani, M. Wordeman, and R. Dennard, "Generalized scaling theory and its application to a 1/4 micron MOSFET design," *IEEE Trans. on Electron Devices*, vol. 29, 1982, pp. 1660-1661.

[Bak85] H. Bakoglu and J. Meindl, "Optimal interconnection circuits for VLSI," *IEEE Trans. on Electron Devices*, vol. 32, 1985, pp. 903-909.

[Bak90] H. Bakoglu, *Circuits, interconnections, and packaging for VLSI*, Addison-Wesley, 1990.

[Bal06a] R. Balasubramonian, N. Muralimanohar, K. Ramani, L. Cheng, and J. Carter, "Leveraging wire properties at the microarchitecture level," *IEEE Micro*, vol. 26, 2006, pp. 40-52.

[Bal06b] J. Balfour and W. Dally, "Design tradeoffs for tiled CMP on-chip networks," *International Conference on Supercomputing (ICS)*, 2006, pp. 187-198.

[Bal06c] J. Balachandran, et al., "Efficient link architecture for on-chip serial links and networks," *Intern. Symp. on System-On-Chip (SOC)*, 2006, pp. 1-4.

[Bal69] M. Ball and F. Hardie, "Effects and detection of intermittent failures in digital systems," *Proc. of the joint Computer Conference (AFIPS)*, 1969, pp. 329-335.

[Bar88] E. Barke, "Line-to-ground capacitance calculation for VLSI: A comparison," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 7, 1988, pp. 295-298.

[Ben02] L. Benini and G. de Micheli, "Networks on chip: A new paradigm for systems on chip design," *Design, Automation and Test in Europe (DATE)*, 2002, pp. 418-419.

[Ben06] L. Benini and G. De Micheli, *Networks on chips: Technology and tools*, Morgan Kaufmann, 2006.

[Ber04] D. Bertozzi and L. Benini, "Xpipes: A network-on-chip architecture for gigascale systems-on-chip," *IEEE Circuits and Systems*, vol. 4, 2004, pp. 18-31.

[Ber05] D. Bertozzi, L. Benini, and G. De Micheli, "Error control schemes for on-chip communication links: The energy-reliability tradeoff," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, 2005, pp. 818-831.

[Ber07] D. Bertozzi, S. Kumar, M. Palesi (Eds.), *Networks-on-Chip*, Hindawi Publishing, 2007.

[Bhu07] R. Bhutada and Y. Manoli, "Complex clock gating with integrated clock gating logic cell," *Intern. Conf. on Design & Technology of Integrated Systems (DTIS)*, 2007, pp. 164-169.

[Bje06] T. Bjerregaard and S. Mahadevan, "A survey of research and practices of network-on-chip," *ACM Computing Surveys*, vol. 38, 2006, Article 1.

[Bjö81] K. Björkqvist and T. Arnborg, "Short channel effects in MOS-transistors," *Physica Scripta*, vol. 24, 1981, pp. 418-421.

[Bmw08] BMWi, "Stromverbrauch von Informations- und Kommunikationstechnik in Deutschland," Technical report, 2008.

[Bod95] N. Boden, et al., "Myrinet: A gigabit-per-second local area network," *IEEE Micro*, vol. 15, 1995, pp. 29-36.

[Bol04a] E. Bolotin, I. Cidon, R. Ginosar, and A. Kolodny, "Cost considerations in network on chip," *Integration, the VLSI Journal*, vol. 38, 2004, pp. 19-42.

[Bol04b] E. Bolotin, "QNoC: QoS architecture and design process for network on chip," *Journal of Systems Architecture*, vol. 50, 2004, pp. 105-128.

[Bol05] E. Bolotin, I. Cidon, R. Ginosar, and A. Kolodny, "Efficient routing in irregular topology NoCs," *CCIT Report No. 554, Technion-Israel Institute of Technology*, 2005.

[Bon00] D. Boning and S. Nassif, "Models of process variations in device and interconnect," In: A. Chandrakasan, W. Bowhill, and F. Fox (Eds.), "Design of high-performance microprocessor circuits," Wiley, 2000, pp. 98-115.

[Bop93] R. Boppana and S. Chalasani, "A comparison of adaptive wormhole routing algorithms," *International Symposium on Computer Architecture (ISCA)*, 1993, pp. 351-360.

[Bor03] M. Borgatti, et al., "A multi-context 6.4Gb/s/channel on-chip communication network using 0.18µm Flash-EEPROM switches and elastic interconnects," *IEEE International Solid-State Circuits Conference (ISSCC)*, 2003, pp. 466-501.

[Bor05] S. Borkar, "Designing reliable systems from unreliable components: The challenges of transistor variability and degradation,"" *IEEE Micro*, vol. 25, 2005, pp. 10-16.

[Bor07] S. Borkar, "Thousand core chips − A technology perspective," *Design Automation Conference (DAC)*, 2007, pp. 746-749.

[Bor99] S. Borkar, "Design challenges of technology scaling," *IEEE Micro*, vol. 19, 1999, pp. 23-29.

[Bou07] S. Bourduas and Z. Zilic, "A hybrid ring/mesh interconnect for network-on-chip using hierarchical rings for global routing," *Intern. Symp. on Networks-on-Chip (NOCS)*, 2007, pp. 195-204.

[Bra09] K. Brackebusch, "Analyse von Ursachen und Lösungsansätzen zur Zuverlässigkeit der Signalübertragung in Nanometer-Technologien," Student research thesis, Rostock University, Germany, 2009.

[Bro01] D. Brooks and M. Martonosi, "Dynamic thermal management for high-performance microprocessors," *Intern. Symp. on High-performance Computer Architecture*, 2001, pp. 171-182.

[Bro07] D. Brooks, R. Dick, R. Joseph, and L. Shang, "Power, thermal, and reliability modeling in nanometer-scale microprocessors," *IEEE Micro*, vol. 27, 2007, pp. 49-62.

[Bry01] R. Bryant, et al., "Limitations and challenges of computer-aided design technology for CMOS VLSI," *Proc. of the IEEE*, vol. 89, 2001, pp. 341-365.

[Cap05] P. Caputa, R. Källsten, and C. Svensson, "Capacitive crosstalk effects on on-chip interconnect latencies and data-rates," *NORCHIP Conference*, 2005, pp. 281-284.

[Cat67] I. Catt, "Crosstalk (noise) in digital systems," *IEEE Trans. on Electronic Computers*, vol. EC-16, 1967, pp. 743-763.

[Cha03] R. Chang, N. Talwalkar, C. Yue, and S. Wong, "Near speed-of-light signaling over on-chip electrical interconnects," *IEEE Journal of Solid-State Circuits*, vol. 38, 2003, pp. 834-838.

[Che06] T. Chen, "Where CMOS is going: Trendy hype vs. real technology," *IEEE International Solid-State Circuits Conference (ISSCC)*, 2006, pp. 1-18.

[Chi95] V. Chiluvuri and I. Koren, "Layout-synthesis techniques for yield enhancement," *IEEE Trans. on Semiconductor Manufacturing*, vol. 8, 1995, pp. 178-187.

[Cho02] C. Choi, "Modelling of nanoscale MOSFETS," Ph.D. dissertation, Stanford University, USA, 2002.

[Chr00] P. Christie and D. Stroobandt, "The interpretation and application of Rent's rule," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 8, 2000, pp. 639-648.

[Cio05] C. Ciordas, T. Basten, A. Radulescu, K. Goossens, and J. Meerbergen, "An event-based monitoring service for networks on chip," *ACM Trans. on Design Automation of Electronic Systems*, vol. 10, 2005, pp. 702-723.

[Cio06a] J. Cioffi, et al., "Vectored DSLs with DSM: The road to ubiquitous gigabit DSLs," *World Telecommunications Congress (WTC)*, 2006.

[Cio06b] C. Ciordas, A. Hansson, K. Goossens, and T. Basten, "A monitoring-aware network-on-chip design flow," *Conf. on Digital System Design (DSD)*, 2006, pp. 97-106.

[Cio06c] C. Ciordas, K. Goossens, A. Radulescu, and T. Basten, "NoC monitoring: Impact on the design flow," *IEEE International Symposium on Circuits And Systems (ISCAS)*, 2006, pp. 1981-1984.

[Cir09] Circuits Multi-Projets (CMP), "CMP Annual Report," 2009.

[Cof01] K. Coffman and A. Odlyzko, "Internet growth: Is there a 'Moore's Law' for data traffic?", In: J. Abello, P. Pardalos, and M. Resende (Eds.), "Handbook of massive data sets," Kluwer, 2001, pp. 47-93.

[Cor06a] C. Cornelius, H. Bohn, and D. Timmermann, "Service-oriented approaches for the operation of large on-chip networks," *NORCHIP Conference*, 2006, pp. 183-186.

[Cor06b] C. Cornelius and D. Timmermann, "Development and operation of networks-on-chip," *IFIP Intern. Conf. on Very Large Scale Integration (VLSI-SoC)*, 2006, pp. 19-23.

[Cor06c] C. Cornelius, S. Köppe, and D. Timmermann, "Dynamic circuit techniques in deep submicron technologies: Domino logic reconsidered," *IEEE International Conference on IC Design and Technology (ICICDT)*, 2006, pp. 53-56.

[Cor07] C. Cornelius, F. Grassert, S. Köppe, and D. Timmermann, "Deep submicron technology: Opportunity or dead end for dynamic circuit techniques," *Intern. Conf. on VLSI Design*, 2007, pp. 330-335.

[Cor08] C. Cornelius, et al., "Encountering gate oxide breakdown with shadow transistors to increase reliability," *Symposium on Integrated circuits and system design (SBCCI)*, 2008, pp. 111-116.

[Cor10] C. Cornelius, P. Gorski, S. Kubisch, and D. Timmermann, "Trading hardware overhead for communication performance in mesh-type topologies," *Conf. on Digital System Design (DSD)*, 2010. (in press)

[Cor99] B. Cordan, "An efficient bus architecture for system-on-chip design," *Custom Integrated Circuits Conference*, 1999, pp. 623-626.

[Cou06] N. Couture and K. Kent, "Periodic licensing of FPGA based intellectual property," *IEEE Intern. Conf. on Field Programmable Technology (FPT)*, 2006, pp. 357-360.

[Cro01] D. Crow and A. Feinberg (Eds.), *Design for reliability*, CRC Press, 2001.

[Dal01] W. Dally and B. Towles, "Route packets, not wires: On-chip interconnection networks," *Design Automation Conference (DAC)*, 2001, pp. 684-689.

[Dal04] W. Dally and B. Towles, *Principles and practices of interconnection networks*, Morgan Kaufmann, 2004.

[Dal07] A. Dalirsani, M. Hosseinabady, and Z. Navabi, "An analytical model for reliability evaluation of NoC architectures," *IEEE International On-Line Testing Symposium (IOLTS)*, 2007, pp. 49-56.

[Dal87] W. Dally and C. Seitz, "Deadlock-free message routing in multiprocessor interconnection networks," *IEEE Trans. on Computers*, vol. C-36, 1987, pp. 547-553.

[Dal91] W. Dally, "Express cubes: Improving the performance of k-ary n-cube interconnection networks," *IEEE Trans. on Computers*, vol. 40, 1991, pp. 1016-1023.

[Dat06] R. Datta, J. Abraham, A. Diril, A. Chatterjee, and K. Nowka, "Adaptive design for performance-optimized robustness," *Intern. Symp. on Defect and Fault Tolerance in VLSI Systems (DFT)*, 2006, pp. 3-11.

[Dav03] J. Davis and J. Meindl, *Interconnect technology and design for gigascale integration*, Norwell, Kluwer, 2003.

[De09] V. De, "Energy efficient designs with wide dynamic range," *Symp. on Energy Efficient Electronic Systems*, 2009.

[Den74] R. Dennard, F. Gaensslen, H. Yu, V. Rideout, E. Bassous, and A. Leblanc, "Design of ion-implanted MOSFET's with very small physical dimensions," *IEEE Journal of Solid-State Circuits*, vol. 9, 1974, pp. 256-268.

[Dho00] A. Dhodapkar, C. Lim, G. Cai, and W. Daasch, "TEM$_2$P$_2$EST: A thermal enabled multi-model power/performance estimator," *Intern. Workshop on Power-Aware Computer Systems*, 2000, pp. 112-125.

[Din02] Y. Ding and M. Rabin, "Hyper-encryption and everlasting security," *Symp. on Theoretical Aspects of Computer Science (STACS)*, 2002, pp. 1-26.

[DinXX] DIN EN ISO 8402, "Quality management and quality assurance, Version 1995-08, 2.10."

[Dob05] R. Dobkin, I. Cidon, R. Ginosar, A. Kolodny, and A. Morgenshtein, "Fast asynchronous bit-serial interconnects for network-on-chip," *CCIT Report No. 529, Technion-Israel Institute of Technology*, 2005.

[Doy06] B. Doyle, et al., "Transistor elements for 30nm physical gate length and beyond," *Intel Technology Journal*, vol. 6, 2006, pp. 42-54.

[Dua03] J. Duato, S. Yalamanchili, and L. Ni, *Interconnection networks: An engineering approach*, Morgan Kaufmann, 2003.

[Elr97] M. Elrabaa, I. Abu-Khater, and M. Elmasry, *Advanced low-power digital circuit techniques*, Springer, 1997.

[Erl05] T. Erl, *Service-oriented architecture: Concepts, technology, and design*, Prentice Hall, 2005.

[FedXX] Federal Standard 1037C, "Telecommunications: Glossary of telecommunication terms."

[Ferr08] A. Ferrante, S. Medardoni, and D. Bertozzi, "Network interface sharing techniques for area optimized NoC architectures," *Conf. on Digital System Design (DSD)*, 2008, pp. 10-17.

[Fra07] A. Frantz, M. Cassel, F. Kastensmidt, E. Cota, and L. Carro, "Crosstalk- and SEU-aware networks on chips," *IEEE Design & Test of Computers*, vol. 24, 2007, pp. 340-350.

[Gao05] F. Gao and J. Hayes, "Total power reduction in CMOS circuits via gate sizing and multiple threshold voltages," *Design Automation Conference (DAC)*, 2005, pp. 31-36.

[Geb09] F. Gebali, H. Elmiligi, and M. El-Kharashi (Eds.), *Networks-on-chips: Theory and practice*, CRC Press, 2009.

[Gen05] N. Genko, et al., ""A complete network-on-chip emulation framework," *Design, Automation and Test in Europe (DATE)*, 2005, pp. 246-251.

[Geo06] J. George, B. Marr, B. Akgul, and K. Palem, "Probabilistic arithmetic and energy efficient embedded signal processing," *Intern. Conf. on Compilers, Architecture and Synthesis for Smbedded Systems (CASES)*, 2006, pp. 158-168.

[Ghe05] F. Ghenassia (Ed.), *Transaction-level modeling in SystemC − TLM concepts and applications for embedded systems*, Springer, 2005.

[Gil08] F. Gilabert, et al., "Exploring high-dimensional topologies for NoC design through an integrated analysis and synthesis framework," *Intern. Symp. on Networks-on-Chip (NOCS)*, 2008, pp. 107-116.

[Gla85] L. Glasser and D. Dobberpuhl, *The design and analysis of VLSI circuits*, Addison-Wesley, 1985.

[Gla92a] C. Glass and L. Ni, "The turn model for adaptive routing," *Intern. Symp. on Computer Architecture*, 1992, pp. 278-287.

[Gla92b] C. Glass and L. Ni, "Maximally fully adaptive routing in 2D meshes," *Intern. Conf. on Parallel Processing*, 1992, pp. 101-104.

[Goo02] K. Goossens, J. van Meerbergen, A. Peeters, and P. Wielage, "Networks on silicon: Combining best-effort and guaranteed services," *Design, Automation and Test in Europe (DATE)*, 2002, pp. 423-425.

[Goo05] K. Goossens, J. Dielissen, and A. Radulescu, "Æthereal network on chip: Concepts, architectures, and implementations," *IEEE Design & Test of Computers*, vol. 22, 2005, pp. 414-421.

[Gre06] C. Grecu, P. Pande, A. Ivanov, and R. Saleh, "BIST for network-on-chip interconnect infrastructures," *IEEE VLSI Test Symposium (VTS)*, 2006, pp. 30-35.

[Gre07a] C. Grecu, et al., "Towards open network-on-chip benchmarks," *Intern. Symp. on Networks-on-Chip (NOCS)*, 2007, pp. 205-212.

[Gre07b] D. Greenfield, A. Banerjee, J. Lee, and S. Moore, "Implications of Rent's rule for NoC design and its fault-tolerance," *Intern. Symp. on Networks-on-Chip (NOCS)*, 2007, pp. 283-294.

[Gua09] L. Guang, et al., "Hierarchical power monitoring for on-chip networks," *Intern. Conf. on Parallel, Distributed, and Network-Based Processing*, 2009.

[Gut01] E. Gutiérrez, J. Deen, and C. Claeys (Eds.), *Low temperature electronics: Physics, devices, circuits, and applications*, Academic Press, 2001.

[Guz06] Z. Guz, et al., "Efficient link capacity and QoS design for network-on-chip," *Design, Automation and Test in Europe (DATE)*, 2006, pp. 1-6.

[Ham07] H. Hamann, et al., "Temperature-limited microprocessors: Measurements and design implications," *Intern. Conf. on VLSI Design*, 2007, pp. 427-432.

[Han05] A. Hansson, K. Goossens, and A. Radulescu, "A unified approach to constrained mapping and routing on network-on-chip architectures," *IEEE/ACM Intern. Conf. on Hardware/Software Codesign and System Synthesis (CODES+ISSS)*, 2005, pp. 75-80.

[Han06] Y. Han, I. Koren, and C. Krishna, "Temptor: A lightweight runtime temperature monitoring tool using performance counters," *International Symposium on Computer Architecture (ISCA), Workshop on Temperature-Aware Computer Systems*, 2006.

[Han07] A. Hansson, M. Coenen, and K. Goossens, "Undisrupted quality-of-service during reconfiguration of multiple applications in networks on chip," *Design, Automation and Test in Europe (DATE)*, 2007, pp. 1-6.

[Han99] M. Hansen, H. Yalcin, and J. Hayes, "Unveiling the ISCAS-85 benchmarks: A case study in reverse engineering," *IEEE Design & Test of Computers*, vol. 16, 1999, pp. 72-80.

[Haz03] P. Hazucha, et al. "Neutron soft error rate measurements in a 90-nm CMOS process and scaling trends in SRAM from 0.25-µm to 90-nm generation," *IEEE Intern. Electron Devices Meeting*, 2003, pp. 21.5.1-21.5.4.

[Hec06] R. Hecht, S. Kubisch, H. Michelsen, E. Zeeb, and D. Timmermann, "A distributed object system approach for dynamic reconfiguration," *IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, 2006, pp. 196-203.

[Hei02] M. van Heijningen, M. Badaroglu, S. Donnay, G. Gielen, and H. De Man, "Substrate noise generation in complex digital systems: Efficient modeling and simulation methodology and experimental verification," *IEEE Journal of Solid-State Circuits*, vol. 37, 2002, pp. 1065-1072.

[Hel06] J. Held, J. Bautista, and S. Koehl, "From a few cores to many: A tera-scale computing research overview," *Research at Intel (White paper)*, 2006.

[Hey03] P. Heydari and M. Pedram, "Ground bounce in digital VLSI circuits," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 11, 2003, pp. 180-193.

[Hil08] M. Hill and M. Marty, "Amdahl's law in the multicore era," *Computer*, vol. 41, 2008, pp. 33-38.

[Hlu88] M. Hluchyj and M. Karol, "Queueing in high-performance packet switching," *IEEE Journal on Selected Areas in Communications*, vol. 6, 1988, pp. 1587-1597.

[Ho01] R. Ho, K. Mai, and M. Horowitz, "The future of wires," *Proc. of the IEEE*, vol. 89, 2001, pp. 490-504.

[Hos07] Y. Hoskote, S. Vangal, A. Singh, N. Borkar, and S. Borkar, "A 5-GHz mesh interconnect for a teraflops processor," *IEEE Micro*, vol. 27, 2007, pp. 51-61.

[Hu03a] J. Hu and R. Marculescu, "Exploiting the routing flexibility for energy/performance aware mapping of regular NoC architectures," *Design, Automation and Test in Europe (DATE)*, 2003, pp. 688-693.

[Hu03b] J. Hu and R. Marculescu, "Energy-aware mapping for tile-based NoC architectures under performance constraints," *Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2003, pp. 233-239.

[Hu04a] J. Hu and R. Marculescu, "DyAD − Smart routing for networks-on-chip," *Design Automation Conference (DAC)*, 2004, pp. 260-263.

[Hu04b] J. Hu and R. Marculescu, "Application-specific buffer space allocation for networks-on-chip router design," *International Conference on Computer-Aided Design (ICCAD)*, 2004, pp. 354-361.

[Hu92] C. Hu, "IC reliability simulation," *IEEE Journal of Solid-State Circuits*, vol. 27, 1992, pp. 241-246.

[Hu95] C. Hu, K. Rodbell, T. Sullivan, K. Lee, and D. Bouldin, "Electromigration and stress-induced voiding in fine Al and Al-alloy thin-film lines," *IBM Journal of Research and Development*, vol. 39, 1995, pp. 465-497.

[Hua00] M. Huang, J. Renau, S. Yoo, and J. Torrellas, "A framework for dynamic energy efficiency and temperature management," *IEEE/ACM Intern. Symp. on Microarchitecture (MICRO)*, 2000, pp. 202-213.

[Hua04] W. Huang, M. Stan, K. Skadron, K. Sankaranarayanan, S. Ghosh, and S. Velusam, "Compact thermal modeling for temperature-aware design," *Design Automation Conference (DAC)*, 2004, pp. 878-883.

[Hui08] J. Hui and D. Culler, "IP is dead, long live IP for wireless sensor networks," *ACM Conference on Embedded Network Sensor Systems*, 2008, pp. 15-28.

[Hun04] W. Hung, et al., "Thermal-aware IP virtualization and placement for networks-on-chip architecture," *IEEE International Conference on Computer Design (ICCD)*, 2004, pp. 430-437.

[Hun05] W. Hung, G. Link, N. Vijaykrishnan, N. Dhanwadaf, and J. Conner, "Temperature-aware voltage islands architecting in system-on-chip design," *Intern. Conf. on Computer Design*, 2005, pp. 689-694.

[Iee94] IEEE. Std 896.9-1994, "IEEE Standard for fault tolerant extensions," 1994.

[Int04] Intel, "Enhanced Intel SpeedStep technology for the Intel Pentium M processor," *From Intel (White paper)*, 2004.

[Iso03] ISO/IEC 15444, "Information technology − JPEG 2000 image coding system: Core coding system," 2003.

[IsoXX] ISO/CD 10303-226, "Industrial automation systems and integration."

[Itr07a] ITRS, "International technology roadmap for semiconductors," *In chapter: Executive summary*, 2007. Available: http://www.itrs.net/

[Itr07b] ITRS, "International technology roadmap for semiconductors," *In chapter: Design*, 2007. Available: http://www.itrs.net/

[Itr07c] ITRS, "International technology roadmap for semiconductors," *In chapter: Process integration, devices, and structures*, 2007. Available: http://www.itrs.net/

[Itr07d] ITRS, "International technology roadmap for semiconductors," *In chapter: Modeling and simulation*, 2007. Available: http://www.itrs.net/

[Itr07e] ITRS, "International technology roadmap for semiconductors," *In chapter: Interconnect*, 2007. Available: http://www.itrs.net/

[Itr07f] ITRS, "International technology roadmap for semiconductors," *In chapter: System Drivers*, 2007. Available: http://www.itrs.net/

[Itr07g] ITRS, "International technology roadmap for semiconductors," *In chapter: Assembly and Packaging*, 2007.

[Iye82] R. Iyer and D. Rossetti, "A statistical load dependency of CPU errors at SLAC," *Fault-Tolerant Computing Symposium (FTCS)*, 1982, pp. 373-372.

[Jal04] A. Jalabert, S. Murali, L. Benini, and G. de Micheli, "xPipesCompiler: A tool for instantiating application specific networks on chip," *Design, Automation and Test in Europe (DATE)*, 2004, pp. 884-889.

[Jan03a] A. Jantsch and H. Tenhunen (Eds.), *Networks on chip*, Kluwer, 2003.

[Jan03b] A. Jantsch, "NoCs: A new contract between hardware and software," *Conf. on Digital System Design (DSD)*, 2003, pp. 10-16.

[Jan05] A. Jantsch, R. Lauter, and A. Vitkowski, "Power analysis of link level and end-to-end data protection in networks on chip," *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2005, pp. 1770-1773.

[Jed02] JEDEC, "Failure mechanisms and models for semiconductor devices," *In JEDEC Publication JEP122AA*, 2002.

[Jer05] A. Jerraya and W. Wolf (Eds.), *Multiprocessor systems-on-chips*, Morgan Kaufmann, 2005.

[Joh89] B. Johnson, *Design and analysis of fault-tolerant digital systems*, Addison-Wesley, 1989.

[Kar87] M. Karol, M. Hluchyj, and S. Morgan, "Input versus output queuing on a space-division packet switch," *IEEE Trans. on Communications*, vol. 35, 1987, pp. 1347-1356.

[Kar99] G. Karsai and J. Sztipanovits, "A model-based approach to self-adaptive software," *IEEE Intelligent Systems and Their Applications*, vol. 14, 1999, pp. 46-53.

[Kat02] A. Katoch, E. Seevinck, and H. Veendrick, "Fast signal propagation for point to point on-chip long interconnects using current sensing," *European Solid-State Circuits Conference (ESSCIRC)*, 2002, pp. 195-198.

[Kav04] N. Kavaldjiev, G. Smit, and P. Jansen, "Two architectures for on-chip virtual channel Router," *Proc. of Progress, Symposium on Embedded Systems*, 2004, pp. 96-102.

[Kim05a] J. Kim, D. Park, C. Nicopolous, N. Vijaykrishnan, and C. Das, "Design and analysis of an NoC architecture from performance, reliability and energy perspective," *Symp. on Architecture for Networking and Communications Systems (ANCS)*, 2005, pp. 173-182.

[Kim05b] C. Kim, K. Roy, S. Hsu, R. Krishnamurthy, and S. Borkar, "An on-die CMOS leakage current sensor for measuring process variation in sub-90nm generations," *International Conference on Integrated Circuit Design and Technology (ICICDT)*, 2005, pp. 221-222.

[Kim06] J. Kim, C. Nicopoulos, D. Park, V. Narayanan, M. Yousif, and C. Das, "A gracefully degrading and energy-efficient modular router architecture for on-chip networks," *International Symposium on Computer Architecture (ISCA)*, 2006, pp. 4-15.

[Kim07] Y. Kim and Y. Kim, "Fault tolerant source routing for network-on-chip," *IEEE Intern. Symp. on Defect and Fault-Tolerance in VLSI Systems (DFT)*, 2007, pp. 12-20.

[Kim08] K. Kim, J. Kim, S. Lee, M. Kim, and H. Yoo, "A 76.8 GB/s 46 mW low-latency network-on-chip for real-time object recognition processor," *IEEE Asian Solid-State Circuits Conference*, 2008, pp. 189-192.

[Kod07] A. Kodi, A. Sarathy, and A. Louri, "Design of adaptive communication channel buffers for low-power area-efficient network-on-chip architecture," *Symp. on Architecture for Networking and Communications Systems (ANCS)*, 2007, pp. 47-56.

[Kon91] S. Konstantinidou and L. Snyder, "Chaos router: Architecture and performance," *Intern. Symp. on Computer Architecture*, 1991, pp. 212-221.

[Kor07] I. Koren and C. Krishna, *Fault-tolerant systems*, Morgan Kaufmann, 2007.

[Kre00] F. Kreith (Ed.), *CRC Handbook of thermal engineering*, CRC Press, 2000.

[Krs07] M. Krstic, E. Grass, F. Gürkaynak, and P. Vivet, "Globally asynchronous, locally synchronous circuits: Overview and outlook," *IEEE Design & Test of Computers*, vol. 24, 2007, pp. 430-441.

[Kub06] S. Kubisch, H. Widiger, D. Duchow, D. Timmermann, and T. Bahls, "Wirespeed MAC address translation and traffic management in access networks," *World Telecommunications Congress (WTC)*, 2006.

[Kub09] S. Kubisch, "Architekturen für Ethernet-basierte Teilnehmerzugangsnetzwerke und deren Umsetzung in Hardware," Ph.D. dissertation, Rostock University, Germany, 2009.

[Kum06] A. Kumar, L. Shang, L. Peh, and N. Jha, "HybDTM: A coordinated hardware-software approach for dynamic thermal management," *Design Automation Conference (DAC)*, 2006, pp. 548-553.

[Kum09] P. Kumar, Y. Pan, J. Kim, G. Memik, and A. Choudhary, "Exploring concentration and channel slicing in on-chip network router," *Intern. Symp. on Networks-on-Chip (NOCS)*, 2009, pp. 276-285.

[Lan09] A. Lankes, T. Wild, and A. Herkersdorf, "Hierarchical NoCs for optimized access to shared memory and IO resources," *Conf. on Digital System Design (DSD)*, 2009, pp. 255-262.

[Lan10] A. Lankes, T. Wild, A. Herkersdorf, S. Sonntag, and H. Reinig, "Comparison of deadlock recovery and avoidance mechanisms to approach message dependent deadlocks in on-chip networks," *Intern. Symp. on Networks-on-Chip (NOCS)*, 2010, pp. 17-24.

[Lan71] B. Landman and R. Russo, "On a pin versus block relationship for partitions of logic graphs," *IEEE Trans. on Computers*, vol. C-20, 1971, pp. 1469-1479.

[Lat07] D. Lattard, et al., "A telecom baseband circuit based on an asynchronous network-on-chip," *IEEE International Solid-State Circuits Conference (ISSCC)*, 2007, pp. 258-261.

[Lat08] D. Lattard, E. Beigne, F. Clermidy, Y. Durand, and R. Lemaire, "A reconfigurable baseband platform based on an asynchronous network-on-chip," *IEEE Journal of Solid-State Circuits*, vol. 43, 2008, pp. 223-235.

[Lee04] K. Lee, et al., "A 51mW 1.6GHz on-chip network for low-power heterogeneous SoC platform," *IEEE International Solid-State Circuits Conference (ISSCC)*, 2004, pp. 152-161.

[Lee05] K. Lee, et al., "Networks-on-chip and networks-in-package for high-performance SoC platforms," *Asian Solid-State Circuits Conference (ASSCC)*, 2005, pp. 485-488.

[Lee06] K. Lee, S. Lee, and H. Yoo, "Low-power network-on-chip for high-performance SoC design," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 14, 2006, pp. 148-160.

[Lee07] H. Lee, N. Chang, U. Ogras, and R. Marculescu, "On-chip communication architecture exploration: A quantitative evaluation of point-to-point, bus, and network-on-chip approaches," *ACM Trans. on Design Automation of Electronic Systems*, vol. 12, 2007, Article 23.

[Lee10] K. Lee, S. Lin, and T. Wang, "Enhanced double via insertion using wire bending," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 29, 2010, pp. 171-184.

[Leh07] T. Lehtonen, P. Liljeberg, and J. Plosila, "Fault tolerance analysis of NoC architectures," *IEEE International Symposium on Circuits And Systems (ISCAS)*, 2007, pp. 361-364.

[Lei06] S. Leibson, "The future of nanometer SOC design," *Intern. Symp. on System-on-Chip (SoC)*, 2006, pp. 1-6.

[Lia08] Y. Liao, G. Mehta, M. Liu, Y. Su, and N. Raman, "Fully automated physical implementation methodology for Tolapai − The first IA based SoC," *International Conference on Solid-state and Integrated Circuit Technology (ICSICT)*, 2008, pp. 1815-1818.

[Lin01] Y. Ling, J. Mi, and X. Lin, "A variational calculus approach to optimal checkpoint placement," *IEEE Trans. on Computers*, vol. 50, 2001, pp. 699-708.

[Liu04] J. Liu, L. Zheng, and H. Tenhunen, "Interconnect intellectual property for Network-on-Chip (NoC)," *Journal of Systems Architecture*, vol. 50, 2004, pp. 65-79.

[Liu93] Z. Liu, et al., "Threshold voltage model for deep-submicrometer MOSFETs," *IEEE Trans. on Electron Devices*, vol. 40, 1993, pp. 86-95.

[Lu05a] Z. Lu, J. Lach, M. Stan, and K. Skadron, "Improved thermal management with reliability banking," *IEEE Micro*, vol. 25, 2005, pp. 40-49.

[Lu05b] Z. Lu and A. Jantsch, "Traffic configuration for evaluating networks on chips," *International Workshop on System-on-Chip for Real-Time Applications (IWSOC)*, 2005, pp. 535-540.

[Mag04] N. Magen, A. Kolodny, U. Weiser, and N. Shamir, "Interconnect-power dissipation in a microprocessor," *Intern. Work. on System-level Interconnect Prediction*, 2004, pp. 7-13.

[Mah01] A. Maheshwari and W. Burleson, "Current-sensing for global interconnects in Very Deep SubMicron (VDSM) CMOS," *Workshop on VLSI*, 2001, pp. 66-70.

[Mah04] A. Maheshwari, W. Burleson, and R. Tessier, "Trading off transient fault tolerance and power consumption in deep submicron (DSM) VLSI circuits," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 12, 2004, pp. 299-311.

[Mah88] A. Mahmood and E. McCluskey, "Concurrent error detection using watchdog processors − A survey," *IEEE Trans. on Computers*, vol. 37, 1988, pp. 160-174.

[Mar09] R. Marculescu, U. Ogras, L. Peh, N. Jerger, and Y. Hoskote, "Outstanding research problems in NoC Design: System, microarchitecture, and circuit perspectives," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 28, 2009, pp. 3-21.

[Mel05] A. Mello, L. Tedesco, N. Calazans, and F. Moraes, "Virtual channels in networks on chip: Implementation and evaluation on Hermes NoC," *Symposium on Integrated Circuits and Systems Design (SBCCI)*, 2005, pp. 178-183.

[Mil04a] M. Millberg, E. Nilsson, R. Thid, S. Kumar, and A. Jantsch, "The Nostrum backbone − A communication protocol stack for networks on chip," *Intern. Conf. on VLSI Design*, 2004, pp. 693-696.

[Mil04b] M. Millberg, E. Nilsson, R. Thid, and A. Jantsch, "Guaranteed bandwidth using looped containers in temporally disjoint networks within the Nostrum network on chip," *Design, Automation and Test in Europe (DATE)*, 2004, pp. 890-895.

[Mil07] M. Millberg and A. Jantsch, "Increasing NoC performance and utilisation using a dual packet exit strategy," *Conf. on Digital System Design (DSD)*, 2007, pp. 511-518.

[Mit01] S. Mitra and E. McCluskey, "Design of redundant systems protected against common-mode failures," *VLSI Test Symposium (VTS)*, 2001, pp. 190-195.

[Mit05] S. Mitra, N. Seifert, M. Zhang, Q. Shi, and K. Kim, "Robust system design with built-in soft-error resilience," *Computer*, vol. 38, 2005, pp. 43-52.

[Moa90] R. Moazzami and C. Hu, "Projecting gate oxide reliability and optimizing reliability screens," *IEEE Trans. on Electron Devices*, vol. 37, 1990, pp. 1643-1650.

[Moo65] G. Moore, "Cramming more components onto integrated circuits," *Electronics*, vol. 38, 1965, pp. 114-117.

[Moo86] W. Moore, "A review of fault-tolerant techniques for the enhancement of integrated circuit yield," *Proc. of the IEEE*, vol. 74, 1986, pp. 684-698.

[Mor09] G. Moritz, C. Cornelius, F. Golatowski, D. Timmermann, and R. Stoll, "Differences and commonalities of service-oriented device architectures, wireless sensor networks and networks-on-chip," *IEEE Intern. Conf. on Advanced Information Networking and Applications (AINA), Workshop SOCNE*, 2009, pp. 482-487.

[Mud06] S. Mudanai, R. Rios, W. Shih, P. Packan, and S. Lee, "Halo doping: Physical effects and compact modeling," *Workshop on Compact Modeling*, 2006, pp. 644-647.

[Muk05] S. Mukherjee, J. Emer, and S. Reinhardt, "The soft error problem: An architectural perspective," *Symp. on High-Performance Computer Architecture*, 2005, pp. 243-247.

[Mul04] R. Mullins, A. West, and S. Moore, "Low-latency virtual-channel routers for on-chip networks," *International Symposium on Computer Architecture (ISCA)*, 2004, pp. 188-197.

[Mul06a] R. Mullins, "Minimising dynamic power consumption in on-chip networks," *Intern. Symp. on System-on-Chip (SoC)*, 2006, pp. 1-4.

[Mul06b] R. Mullins, A. West, and S. Moore, "The design and implementation of a low-latency on-chip network," *Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2006, pp. 164-169.

[Mur04] S. Murali and G. De Micheli, "Bandwidth-constrained mapping of cores onto NoC architectures," *Design, Automation and Test in Europe (DATE)*, 2004, pp. 896-901.

[Mur05a] S. Murali, L. Benini, and G. De Micheli, "Mapping and physical planning of networks-on-chip architectures with quality-of-service guarantees," *Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2005, pp. 27-32.

[Mur05b] S. Murali, et al., "Analysis of error recovery schemes for networks on chips," *IEEE Design & Test of Computers*, vol. 22, 2005, pp. 434-442.

[Mur09] S. Murali, *Designing reliable and efficient networks on chips*, Springer, 2009.

[Nal00] A. Nalamalpu and W. Burleson, "Repeater insertion in deep sub-micron CMOS: Ramp-based analytical model and placement sensitivity analysis," *IEEE Intern. Symp. on Circuits and Systems*, 2000, pp. 766-769.

[Nal02] A. Nalamalpu, S. Srinivasan, and W. Burleson, "Boosters for driving long onchip interconnects − Design issues, interconnect synthesis, and comparison with repeaters," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 21, 2002, pp. 50-62.

[Nar05] S. Narendra and A. Chandrakasan (Eds.), *Leakage in nanometer CMOS technologies*, Springer, 2005.

[Neu09] M. Neuenhahn, J. Schleifer, H. Blume, and T. Noll, "Quantitative comparison of performance analysis techniques for modular and generic network-on-chip," *Journal of Advances in Radio Science*, vol. 7, 2009, pp. 107-112.

[Ni93] L. Ni and P. McKinley, "A survey of wormhole routing techniques in direct networks," *Computer*, vol. 2, 1993, pp. 62-76.

[Nic06] C. Nicopoulos, et al., "ViChaR: A dynamic virtual channel regulator for network-on-chip routers," *Intern. Symp. on Microarchitecture (MICRO)*, 2006, pp. 333-346.

[Nil03] E. Nilsson, M. Millberg, J. Oberg, and A. Jantsch, "Load distribution with the proximity congestion awareness in a network on chip," *Design, Automation and Test in Europe (DATE)*, 2003, pp. 1126-1127.

[Nol04] V. Nollet, T. Marescaux, and D. Verkest, "Operating-system controlled network on chip," *Design Automation Conference (DAC)*, 2004, pp. 256-259.

[Nur04] J. Nurmi, H. Tenhunen, J. Isoaho, and A. Jantsch (Eds.), *Interconnect-centric design for advanced SoC and NoC*, Kluwer, 2004.

[Oas06] OASIS, "Reference model for service oriented architecture," 2006.

[Oas09] OASIS, "Devices profile for web services − Version 1.1," 2009.

[Ogr05] U. Ogras and R. Marculescu, "Application-specific network-on-chip architecture customization via long-range link insertion," *International Conference on Computer-Aided Design (ICCAD)*, 2005, pp. 246-253.

[Ogr06a] U. Ogras and R. Marculescu, ""It's a small world after all": NoC performance optimization via long-range link insertion," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 14, 2006, pp. 693-706.

[Ogr06b] U. Ogras, R. Marculescu, H. Lee, and N. Chang, "Communication architecture optimization: Making the shortest path shorter in regular networks-on-chip," *Design, Automation and Test in Europe (DATE)*, 2006, pp. 6-11.

[Oka97] S. Okada, Y. Matsuda, T. Watanabe, and K. Kondo, "A single chip motion JPEG codec LSI," *IEEE Trans. on Consumer Electronics*, vol. 43, 1997, pp. 418-422.

[Oke04] H. O'Keeffe, "The Nexus 5001 forum standard providing the gateway to the embedded systems of the future," IEEE-ISTO 5001, 2004.

[Oma07] M. Omana, D. Rossi, and C. Metra, "Latch susceptibility to transient faults and new hardening approach," *IEEE Trans. on Computers*, vol. 56, 2007, pp. 1255-1268.

[Pan05] P. Pande, C. Grecu, M. Jones, A. Ivanov, and R. Saleh, "Performance evaluation and design trade-offs for network-on-chip interconnect architectures," *IEEE Trans. on Computers*, vol. 54, 2005, pp. 1025-1040.

[Pan06] P. Pande, A. Ganguly, and C. Grecu, "Energy reduction through crosstalk avoidance coding in NoC paradigm," *Conf. on Digital System Design (DSD)*, 2006, pp. 689-695.

[Pan08] P. Pande, A. Ganguly, H. Zhu, and C. Grecu, "Energy reduction through crosstalk avoidance coding in networks on chip," *Journal of Systems Architecture*, vol. 54, 2008, pp. 441-451.

[Pap04] G. Papadopoulos, In San Jose Mercury News, Dated February 7, 2004.

[Par00] K. Park and W. Willinger (Eds.), *Self-similar network traffic and performance evaluation*, Wiley, 2000.

[Par06] D. Park, C. Nicopoulos, J. Kim, N. Vijaykrishnan, and C. Das, "A distributed multi-point network interface for low-latency, deadlock-free on-chip interconnects," *Intern. Conf. on Nano-Networks and Workshops*, 2006, pp. 1-6.

[Pau05] B. Paul, H. Kufluoglu, M. Alam, and K. Roy, "Impact of NBTI on the temporal performance degradation of digital circuits," *IEEE Electron Device Letters*, vol. 26, 2005, pp. 560-562.

[Ped02] M. Pedram and J. Rabaey (Eds.), *Power aware design methodologies*, Kluwer, 2002.

[Pen06] S. Penolazzi and A. Jantsch, "A high level power model for the Nostrum NoC," *Conf. on Digital System Design (DSD)*, 2006, pp. 6-9.

[Pif94] G. Pifarre, L. Gravano, S. Felperin, and J. Sanz, "Fully adaptive minimal deadlock-free packet routing in hypercubes, meshes, and other networks: Algorithms and simulations," *IEEE Trans. on Parallel and Distributed Systems*, 1994, pp. 247-263.

[Pig06] C. Piguet, *Low-power CMOS circuits: Technology, logic design and CAD tools*, CRC Press, 2006.

[Poe09] S. Poeggel, "Modellierung der Temperaturverteilung in on-chip Netzwerken," Student research thesis, Rostock University, Germany, 2009.

[Pri08] F. Pribbernow, "Realisierung eines Routers für on-chip Netzwerke mit dem Ziel der Optimierung von Verlustleistung und Zuverlässigkeit," Diploma thesis, Rostock University, Germany, 2008.

[Pud06] A. Puder, K. Römer, and F. Pilhofer, *Distributed systems architecture – A middleware approach*, Morgan Kaufmann, 2006.

[Pus08] D. Puschini, F. Clermidy, P. Benoit, G. Sassatelli, and L. Torres, "Temperature-aware distributed run-time optimization on MP-SoC using game theory," *IEEE Symp. on VLSI*, 2008, pp. 375-380.

[Put07] C. Puttmann, J. Niemann, M. Porrmann, and U. Ruckert, "GigaNoC – A hierarchical network-on-chip for scalable chip-multiprocessors," *Conf. on Digital System Design Architectures (DSD)*, 2007, pp. 495-502.

[Rab03] J. Rabaey, A. Chandrakasan, and B. Nikolić (Eds.), *Digital integrated circuits: A design perspective*, Prentice hall, 2003.

[Rad05] A. Radulescu, et al., "An efficient on-chip NI offering guaranteed services, shared-memory abstraction, and flexible network configuration," *IEEE Trans. on Computer-aided design of Integrated Circuits and Systems*, vol. 24, 2005, pp. 4-17.

[Ran07] P. Rantala, J. Isoaho, and H. Tenhunen, "Novel agent-based management for fault-tolerance in network-on-chip," *Conf. on Digital System Design (DSD)*, 2007, pp. 551-555.

[Ras02] P. Rashinkar, P. Paterson, and L. Singh, *System-on-a-chip verification: Methodology and techniques*, Kluwer, 2002.

[Rid10] Ridgetop Group, "Sentinel Silicon – Application guide," 2010. Available: http://www.ridgetop-group.com/

[Rij01] E. Rijpkema, K. Goossens, and P. Wielage, "A router Architecture for Networks on Silicon," *Proc. of Progress, Workshop on Embedded Systems*, 2001.

[Rij03] E. Rijpkema, et al., "Trade-offs in the design of a router with both guaranteed and best-effort services for networks on chip," *Design, Automation, and Test in Europe (DATE)*, 2003, pp. 125-139.

[Roy03] K. Roy, S. Mukhopadhyay, and H. Mahmoodi-Meimand, "Leakage current mechanisms and leakage reduction techniques in deep-submicrometer CMOS circuits," *Proc. of the IEEE*, vol. 91, 2003, pp. 305-327.

[Sak90] T. Sakurai and A. Newton, "Alpha-power law MOSFET model and its application to CMOS inverter delay and other formulas," *IEEE Journal of Solid-State Circuits*, vol. 25, 1990, pp. 584-594.

[Sal02] E. Salminen, V. Lahtinen, K. Kuusilinna, and T. Hämäläinen, "Overview of bus-based system-on-chip interconnections," *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2002, pp. 372-375.

[Sal05] E. Salminen, T. Kangas, J. Riihimaki, and T. Hämäläinen, "Requirements for network-on-chip benchmarking," *NORCHIP Conference*, 2005, pp. 82-85.

[Sal07a] E. Salminen, A. Kulmala, and T. Hämäläinen, "On network-on-chip comparison," *Conf. on Digital System Design (DSD)*, 2007, pp. 503-510.

[Sal07b] J. Salzmann, F. Sill, and D. Timmermann, "Algorithm for fast statistical timing analysis," *Intern. Symp. on System-on-Chip (SoC)*, 2007, pp. 1-4.

[Sal08] E. Salminen, A. Kulmala, and T. Hämäläinen, "Survey of network-on-chip Proposals," *Report from OCP-IP (White paper)*, 2008, pp. 503-510. Available: http://www.ocpip.org/

[Sal09] E. Salminen, C. Grecu, T. Hämäläinen, and A. Ivanov, "Application modelling and hardware description for network-on-chip benchmarking," *IET Computers & Digital Techniques*, vol. 3, 2009, pp. 539-550.

[Säm07] H. Sämrow, "Untersuchung des Einflusses von Technologie und Layout auf die Signal-Übertragung in Nanometer-Technologien," Diploma thesis, Rostock University, Germany, 2007.

[Sas06] Y. Sasaki, K. Namba, and H. Ito, "Soft error masking circuit and latch using Schmitt trigger circuit," *IEEE Intern. Symp. on Defect and Fault Tolerance in VLSI Systems (DFT)*, 2006, pp. 327-335.

[Sch03] D. Schroder and J. Babcock, "Negative bias temperature instability: Road to cross in deep submicron silicon semiconductor manufacturing," *Journal of Applied Physics*, vol. 94, 2003, pp. 1-17.

[Sch06] L. Scheffer, L. Lavagno, and G. Martin (Eds.), *EDA for IC implementation, circuit design, and process technology*, CRC Press, 2006.

[Sch07] T. Schönwald, J. Zimmermann, O. Bringmann, and W. Rosenstiel, "Fully adaptive fault-tolerant routing algorithm for network-on-chip architectures," *Conf. on Digital System Design (DSD)*, 2007, pp. 527-534.

[Sch09] P. Schelkens, A. Skodras, and T. Ebrahimi, *The JPEG 2000 suite*, Wiley, 2009.

[Sem03] SEMATECH, "Critical reliability challenges for the International Technology Roadmap for Semiconductors (ITRS)," *In Technology Transfer 03024377 A-TR*, 2003.

[Sem09] Semiconductor Industry Association (SIA), "Global semiconductor sales," 2009. Available: http://www.sia-online.org

[Seo05] D. Seo, A. Ali, W. Lim, N. Rafique, and M. Thottethodi, "Near-optimal worst-case throughput routing for two-dimensional mesh networks," *International Symposium on Computer Architecture (ISCA)*, 2005, pp. 432-443.

[Sey98] A. Seydim, "Wormhole Routing in Parallel Computers," Technical report, Southern Methodist University, 1998.

[Shi02] P. Shivakumar, M. Kistler, S. Keckler, D. Burger, and L. Alvisi, "Modeling the effect of technology trends on the soft error rate of combinational logic," *Intern. Conf. on Dependable Systems and Networks (DSN)*, 2002, pp. 389-398.

[Shi03] P. Shivakumar, S. Keckler, C. Moore, and D. Burger, "Exploiting microarchitectural redundancy for defect tolerance," *International Conference on Computer Design (ICCD)*, 2003, pp. 481-488.

[Sho50] W. Shockley, *Electrons and holes in semiconductors, with applications to transistor electronics*, D. Van Nostrand, 1950.

[Sig02] D. Siguenza-Tortosa and J. Nurmi, "VHDL-based simulation environment for Proteo NoC," *IEEE Intern. High-level Design Validation and Test Workshop*, 2002, pp. 1-6.

[Sil06] F. Sill, C. Cornelius, S. Kubisch, and D. Timmermann, "Mixed gates: Leakage reduction techniques applied to switches for on-chip networks," *Intern. Workshop on Reconfigurable Communication-centric SoCs (ReCoSoC)*, 2006, pp. 76-82.

[Sil07] F. Sill, "Untersuchung und Reduzierung des Leckstroms integrierter Schaltungen in Nanometer-Technologien bei konstanten Performanceanforderungen," Ph.D. dissertation, Rostock University, Germany, 2007.

[Sil08] A. Silberschatz, P. Galvin, and G. Gagne, *Operating system concepts*, Wiley, 2008.

[Ska02] K. Skadron, T. Abdelzaher, and M. Stan, "Control-theoretic techniques and thermal-RC modeling for accurate and localized dynamic thermal management," *Intern. Symp. on High-performance Computer Architecture*, 2002, pp. 17-28.

[Ska04] K. Skadron, et al., "Temperature-aware microarchitecture: Modeling and implementation," *ACM Trans. on Architecture and Code Optimization*, vol. 1, 2004, pp. 94-125.

[Sof07] S. Sofke, "Implementierung und Vergleich verschiedener Router-Konzepte für on-chip Netzwerke," Student research thesis, Rostock University, Germany, 2007.

[Soi03] J. Soininen, et al., "Extending platform-based design to network on chip systems," *Intern. Conf. on VLSI Design*, 2003, pp. 401-408.

[Sou09] D. Soudris, C. Piguet, and C. Goutis, *Designing CMOS circuits for low power*, Springer, 2009.

[Spi04] M. Spica and T. Mak, "Do we need anything more than single bit error correction (ECC)?," *Intern. Workshop on Memory Technology, Design and Testing*, 2004, pp. 111-116.

[Sri03] J. Srinivasan, S. Adve, P. Bose, J. Rivers, and C. Hu, "RAMP: A model for reliability aware microprocessor design," *IBM Research Report RC23048*, vol. 23048, 2003.

[Sri04] J. Srinivasan, S. Adve, P. Bose, and J. Rivers, "The impact of technology scaling on lifetime reliability," *Intern. Conf. on Dependable Systems and Networks (DSN)*, 2004, pp. 161-170.

[Sri98] P. Srivastava, A. Pua, and L. Welch, "Issues in the design of domino logic circuits," *Great Lakes Symposium on VLSI (GLSVLSI)*, Lafayette, USA, 1998, pp. 108-112.

[Str01] D. Stroobandt, *A priori wire length estimates for digital design*, Kluwer, 2001.

[Sul04] A. Sultania, D. Sylvester, and S. Sapatnekar, "Transistor and pin reordering for gate oxide leakage reduction in dual Tox circuits," *IEEE International Conference on Computer Design (ICCD)*, 2004, pp. 228-233.

[Sun01] Sun Microsystems, "Jini architecture specification Version 1.2," 2001.

[Sun02] Y. Sun, S. Kumar, and A. Jantsch, "Simulation and evaluation for a network on chip architecture using NS-2," *NORCHIP Conference*, 2002.

[Sun03] S. Sun and S. Lee, "A JPEG chip for image compression and decompression," *Journal of VLSI Signal Processing*, vol. 35, 2003, pp. 43-60.

[Sun99] V. Sundararajan and K. Parhi, "Low power synthesis of dual threshold voltage CMOS VLSI circuits," *Intern. Symp. on Low Power Electronics and Design*, 1999, pp. 139-144.

[Syl00] D. Sylvester and K. Keutzer, "A global wiring paradigm for deep submicron design," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 19, 2000, pp. 242-252.

[Syn09] Synopsys, "Design compiler optimization reference manual, Version C-2009.06," 2009.

[Tan01] A. Tanenbaum, *Modern operating systems*, Prentice Hall, 2001.

[Tan02] A. Tanenbaum, *Computer networks*, Prentice Hall, 2002.

[Tau98] Y. Taur and T. Ning, *Fundamentals of modern VLSI devices*, Cambridge University Press, 1998.

[Tee07] P. Teehan, M. Greenstreet, and G. Lemieux, "A survey and taxonomy of GALS design styles," *IEEE Design & Test of Computers*, vol. 24, 2007, pp. 418-428.

[Teh10] M. Tehranipoor and K. Butler, "Power supply noise: A survey on effects and research," *IEEE Design & Test of Computers*, vol. 27, 2010, pp. 51-67.

[Til08] Tilera Corporation, "Tile Processor Architecture − Technology Brief," 2008. Available: http://www.tilera.com/

[Toc10] A. Tockhorn, C. Cornelius, H. Saemrow, and D. Timmermann, "Modeling temperature distribution in networks-on-chip using RC-circuits," *IEEE Intern. Symp. on Design and Diagnostics of Electronic Circuits and Systems (DDECS)*, 2010, pp. 229-232.

[Tsa00] C. Tsai and S. Kang, "Cell-level placement for improving substrate thermal distribution," *IEEE Trans. on Computer-aided Design of Integrated Circuits and Systems*, vol. 19, 2000, pp. 253-266.

[Tsa04] Y. Tsai, D. Duarte, N. Vijaykrishnan, and M. Irwin, "Characterization and modeling of run-time techniques for leakage power reduction," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 12, 2004, pp. 1221-1233.

[Tsc02] J. Tschanz, et al., "Adaptive body bias for reducing impacts of die-to-die and within-die parameter variations on microprocessor frequency and leakage," *IEEE Journal of Solid-State Circuits*, vol. 37, 2002, pp. 1396-1402.

[Ull04] M. Ullmann, M. Hübner, B. Grimm, and J. Becker, "An FPGA run-time system for dynamical on-demand reconfiguration," *IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, 2004, pp. 135-142.

[Upn08] UPnP Forum, "UPnP device architecture 1.0," 2008.

[Uss01] R. Usselmann, "OpenCores SoC bus review, Rev. 1.0," *From OpenCores.org (White paper)*, 2001. Available: http://www.opencores.org/

[Val82] L. Valiant, "A Scheme for Fast Parallel Communication," *SIAM Journal on Computing*, vol. 11, 1982, pp. 350-361.

[Van07] S. Vangal, et al., "An 80-tile 1.28TFLOPS network-on-chip in 65nm CMOS," *IEEE International Solid-State Circuits Conference (ISSCC)*, 2007.

[Var04] G. Varatkar and R. Marculescu, "On-chip traffic modeling and synthesis for MPEG-2 video applications," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 12, 2004, pp. 108-119.

[Vee00] H. Veendrick, *Deep-submicron CMOS ICs: From basics to ASICs*, Springer, 2000.

[Ver01] B. Vermeulen, S. Oostdijk, and F. Bouwman, "Test and debug strategy of the PNX8525 Nexperia digital video platform system chip," *International Test Conference (ITC)*, 2001, pp. 121-130.

[W3c04] W3C Working Group, "Web Services Architecture," 2004.

[Wan02] M. Wang, T. Madhyastha, N. Chan, S. Papadimitriou, and C. Faloutsos, "Data mining meets performance evaluation: fast algorithms for modeling bursty traffic," *International Conference on Data Engineering (ICDE)*, 2002, pp. 507-516.

[Wan03] H. Wang, L. Peh, and S. Malik, "Power-driven design of router microarchitectures in on-chip networks," *IEEE/ACM Intern. Symp. on Microarchitecture*, 2003, pp. 105-116.

[Wan08] L. Wang, C. Stroud, and N. Touba (Eds.), *System-on-chip test architectures: Nanometer design for testability*, Morgan Kaufmann, 2008.

[Wan98] Q. Wang and S. Vrudhula, "Static power optimization of deep submicron CMOS circuits for dual VT technology," *International Conference on Computer-Aided Design (ICCAD)*, 1998, pp. 490-496.

[Weg09] T. Wegner, "Integration von Konzepten zur Erhöhung der Zuverlässigkeit in Networks-on-Chip," Master thesis, Rostock University, Germany, 2009.

[Weg10] T. Wegner, C. Cornelius, A. Tockhorn, and D. Timmermann, "A reliability-aware approach for monitoring and control of temperature in networks-on-chip," *IEEE International Conference on Computer Design (ICCD)*, 2010. (submitted)

[Wei99] L. Wei, Z. Chen, K. Roy, Y. Ye, and V. De, "Mixed-Vth (MVT) CMOS circuit design methodology for low power applications," *Design Automation Conference (DAC)*, 1999, pp. 430-435.

[Wes05] N. Weste and D. Harris, *CMOS VLSI design: A circuits and systems perspective*, Addison-Wesley, 2005.

[Wid06a] H. Widiger, S. Kubisch, D. Duchow, T. Bahls, and D. Timmermann, "A simplified, cost-effective MPLS labeling architecture for access networks," *World Telecommunications Congress (WTC)*, 2006.

[Wid06b] H. Widiger, S. Kubisch, and D. Timmermann, "An integrated hardware solution for MAC address translation, MPLS, and traffic management in access networks," *IEEE Conf. on Local Computer Networks (LCN)*, 2006, pp. 272-279.

[Wid08] H. Widiger, "Paketverarbeitende Systeme − Algorithmen und Architekturen für hohe Verarbeitungsgeschwindigkeiten," Ph.D. dissertation, Rostock University, Germany, 2008.

[Wik03] D. Wiklund and D. Liu, "SoCBUS: Switched network on chip for hard real time embedded systems," *IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, 2003, pp. 8-15.

[Wik04] D. Wiklund, S. Sathe, and D. Liu, "Network on chip simulations for benchmarking," *IEEE Intern. Workshop on System-on-Chip for Real-Time Applications*, 2004, pp. 269-274.

[Win99] S. Winegarden, "A bus architecture centric configurable processor system," *Custom Integrated Circuits Conference*, 1999, pp. 627-630.

[Wol07] P. Wolkotte, P. Hölzenspies, and G. Smit, "Fast, accurate and detailed NoC simulations," *Intern. Symp. on Networks-on-Chip (NOCS)*, 2007, pp. 323-332.

[Won04] B. Wong, A. Mittal, and Y. Cao, *Nano-CMOS circuit and physical design*, John Wiley, 2004.

[Wu02] E. Wu, et al., "Interplay of voltage and temperature acceleration of oxide breakdown for ultra-thin gate oxides," *Journal of Solid-State Electronics*, vol. 46, 2002, pp. 1787-1798.

[Xi06] J. Xi and P. Zhong, "A system-level network-on-chip simulation framework integrated with low-level analytical models," *International Conference on Computer Design (ICCD)*, 2006, pp. 383-388.

[Xu03] J. Xu and W. Wolf, "A wave-pipelined on-chip interconnect structure for networks-on-chips," *Symp. on High Performance Interconnects*, 2003, pp. 10-14.

[Yan06] Y. Yang, Z. Gu, C. Zhu, L. Shang, and R. Dick, "Adaptive chip-package thermal analysis for synthesis and design," *Design, Automation and Test in Europe (DATE)*, 2006, pp. 1-6.

[Yan89] J. Yantchev and C. Jesshope, "Adaptive, low latency, deadlock-free packet routing for networks of processors," *IEE Proc. of Computers and Digital Techniques*, vol. 136, 1989, pp. 178-186.

[Yeo04] K. Yeo and K. Roy, *Low voltage, low power VLSI subsystems*, McGraw-Hill, 2004.

[Yua05] L. Yuan and G. Qu, "Enhanced leakage reduction technique by gate replacement," *Design Automation Conference (DAC)*, 2005, pp. 47-50.

[Yua82] C. Yuan and T. Trick, "A simple formula for the estimation of the capacitance of two-dimensional interconnects in VLSI circuits," *IEEE Electron Device Letters*, vol. 3, 1982, pp. 391-393.

[Zef04] C. Zeferino, M. Kreutz, and A. Susin, "RASoC: A router soft-core for networks-on-chip," *Design, Automation and Test in Europe (DATE)*, 2004, pp. 198-203.

Theses

1. Continuous scaling of technology has been the key to success for the semiconductor industry. By now, current technology is about to reach determined physical limits, which impairs the further development. Therefore, it necessitates novel design paradigms to overcome or at least to mitigate the present challenges of nanotechnology.
2. Yield and reliability have traditionally been considered an issue of manufacturing. Even though temporary failures are slowly being accepted as a common concern, most designs still assume a faultless operation. However, to further neglect lifetime reliability and permanent failures in particular, will lead to insufficient system design and operation.
3. The crucial parameters of performance, power consumption and reliability are closely intertwined. Hence, it is important to assess integrated systems with respect to these parameters altogether. A comprehensive and accepted metric though does not exist to date.
4. Wires considerably impact on-chip signal transmission, specifically across longer distances. In the future, their influence in relation to gate delay will further exacerbate and wires will mark a decisive property for the design of complex integrated systems.
5. Repeater insertion is the prevalent approach to improve signal transmission. However, the change of wire characteristics in smaller technologies results in a drastic increase of required repeaters. Although this trend evolves into an extensive cost factor, other feasible alternatives are not yet at hand.
6. Integrated circuits develop into ever more complex systems. Conventional point-to-point and bus-based architectures though are not scalable, and thus not capable to meet the boosting demands of such systems – e.g. in terms of communication performance.
7. Network-On-Chip (NOC) is the emerging architecture for the design of complex integrated systems because it benefits from its modularity and its concurrency in regards to computation as well as communication.
8. The abstraction of functionality into independent layers facilitates the understanding of NOC in principle, but it masks the intertwined correlations of the various abstraction layers among each other. This can entail deficient design decisions and prevents the exploitation of the mutual advantages of the layers.
9. A fine-grained router architecture is the desirable starting point for sundry improvements.

By way of example, clock and power gating achieve remarkable power savings when they are applied to such router architectures.

10. Elaborate router layout reduces the area costs of networks-on-chip to a great extent without impairing communication characteristics.

11. Complex NOC-based architectures provide multi-purpose, redundant modules and execute varying applications. Hence, system characteristics are not fully known at design time, and design improvements can hardly rely on application-specific knowledge.

12. Dimension-ordered routing is the most widespread routing scheme for on-chip networks. Since such routing is deterministic and not able to adapt to changing run-time conditions or faulty components, its use is inappropriate in prospective complex systems.

13. Topology and routing scheme constitute fundamental characteristics of an NOC. These characteristics can effectively be exploited by means of a heterogeneous distribution of packet FIFOs in order to improve performance and power dissipation.

14. Mesh-based topologies stand out due to good communication performance, but high area costs. Clustered topologies can considerably relieve such costly area efforts −whereas it is exceedingly beneficial when local traffic is enforced as well.

15. Distributed control units monitor and control system behavior at a low level of abstraction. However, as part of the multistage concept of the Error Resiliency Layer (ERL), this facilitates to advantageously interconnect efficient hardware solutions and flexible software implementations with global awareness.

16. System management plays a key role in complex integrated systems to cope with the multitude of associated tasks as well as with changing temporal and spatial conditions. For this purpose, the adaptation of working principles from Service-Oriented Architectures (SOA) presents a capable approach to tackle the present, extensive challenges.

17. Scaling of technology leads to higher power consumption per unit area. Since power density translates into heat, thermal impact on performance, reliability and power itself deteriorates. Against this background, a fine-grained temperature model −which takes advantage of the electrical and thermal duality− allows considering temperature during the design of an architecture and its system management.

18. Up to now, there is no qualified method to compare different networks-on-chip. Hence, it urgently requires appropriate benchmark suites to assess such complex systems. Because of the great variety of existing application scenarios, such benchmarks have to be platform-dependent, domain-specific and have to account for diverse types of failures.

19. The definition of the Energy-Reliability Ratio (ERR) is a first attempt to express the quality of a design in due consideration of reliability. However, it will require great efforts to refine and to enforce acknowledged methods that determine reliability and robustness in complex integrated systems.

Abstract

The continuous scaling of technology is associated with an escalating number of concerns due to both the tiny dimensions and the enormous complexity of today's integrated systems. Against this background, Network-On-Chip (NOC) is the emerging design paradigm to cope with the diverse issues of nanotechnology. Therefore, this thesis contributes to an advanced understanding and improved implementation of NOC.

As a start, the main challenges of integrated systems are introduced, whereas several compiled classifications ease the understanding of the subsequent, elaborate investigations. The actual implementations are accomplished based on a 65 nm technology from STMicroelectronics, and begin with a study of the basic components of on-chip networks. This comprises on the one hand the characterization of links and corresponding techniques to improve signal transmission. On the other hand, miscellaneous routers are implemented in order to evaluate the characteristics of diverse design alternatives −whereas several enhancements are derived based on the obtained findings. NOC-based systems are thereupon investigated, which comprehends the architecture itself and the required algorithms to operate it. Thereby, the sundry, achieved improvements exploit the peculiarities of the diverse but intertwined abstraction layers. A tangible example is the proposed concept of the Error Resiliency Layer (ERL) that successfully combines efficient, low-level functionality with a flexible, high-level system management.

Lastly, two selected case studies prove the relevance of NOC and the pursued objectives of this thesis. To this end, requirements for efficient system design and simulation are established first. In this context, a novel fine-grained temperature model is presented that allows simulating the dynamic distribution of temperature in NOC-based architectures. By all means, the case studies evidently demonstrate that design parameters as well as abstraction layers are closely intertwined. Therefore, it is crucial to consider networks-on-chip as a whole so that performance, power consumption and reliability are purposefully traded off against each other. Finally, a conclusion summarizes the acquired findings and identifies interesting areas for further advancements based on results of this thesis.

Abstrakt

Die kontinuierliche Skalierung der Technologie ist mit einer dramatisch zunehmenden Anzahl von Problemen verbunden. Dies resultiert sowohl aus den winzigen Abmessungen als auch aus der enormen Komplexität heutiger integrierter Systeme. Vor diesem Hintergrund ist Network-On-Chip (NOC) der entscheidende Designansatz um mit den verschiedenen Problemen der Nanotechnologie fertig zu werden. Deshalb trägt diese Arbeit zu einem erweiterten Verständnis und einer besseren Implementierung von NOC bei.

Zunächst werden die wesentlichen Herausforderungen integrierter Systeme eingeführt. Hierbei erleichtern mehrere, erarbeitete Klassifizierungen das Verständnis der folgenden, umfangreichen Untersuchungen. Die eigentlichen Implementierungen basieren auf einer 65 nm Technologie von STMicroelectronics und beginnen mit der Studie der grundlegenden Komponenten von on-chip Netzwerken. Dies umfasst zum einen die Charakterisierung der Links und entsprechender Techniken um die Signalübertragung zu verbessern. Zum anderen werden zahlreiche Router implementiert um die Eigenschaften einzelner Designalternativen zu bewerten −wobei aus den erhaltenen Ergebnissen mehrere Verbesserungen abgeleitet werden. Anschließend werden NOC-basierte Systeme untersucht, was die Architektur und die notwendigen Algorithmen für den Betrieb beinhaltet. Die diversen, erzielten Verbesserungen nutzen dabei die besonderen Merkmale der unterschiedlichen, aber voneinander abhängigen Abstraktionsschichten. Ein konkretes Beispiel dafür ist das Konzept des Error Resiliency Layer (ERL), das effiziente, low-level Funktionalität mit flexiblem, high-level Management kombiniert.

Schließlich belegen zwei ausgesuchte Fallstudien die Bedeutung von NOC und der verfolgten Zielstellung dieser Arbeit. Dazu werden zuerst die Anforderungen für einen effizienten Systementwurf und die funktionale Simulation geschaffen. In diesem Zusammenhang wird ein neues Temperaturmodell präsentiert, dass die Simulation der dynamischen Verteilung von Temperatur in NOC-basierten Architekturen erlaubt. Die Fallstudien veranschaulichen eindrucksvoll, dass sowohl Designparameter als auch Abstraktionsschichten eng miteinander verbunden sind. Deshalb ist es entscheidend Networks-On-Chip als Ganzes zu betrachten damit funktionale Leistung, Stromverbrauch und Zuverlässigkeit zielgerichtet gegeneinander abgewogen werden können. Am Ende werden die erlangten Ergebnisse zusammengefasst und interessante Bereiche benannt, die sich aus dieser Arbeit ergeben und weitere Verbesserungen versprechen.