Universitat Politècnica de València
Departamento de Informática de Sistemas y Computadores

Addressing Manufacturing Challenges in NoC-based ULSI Designs

A dissertation submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
(Computer Sciences)

Author

Carles Hernández Luz

Advisors

Federico Silla Jiménez
José Francisco Duato Marín

Valencia, May 2012

Agradecimientos

Varios han sido los años de mi vida que he dedicado a la realización de esta tesis. Durante estos años si bien he conseguido compaginar mi vida personal con el desarrollo de la la tesis, ha sido el doctorado el que ha marcado de algún modo u otro los tiempos en mi vida. Por todo esto y por mucho más agradezco a Anna la paciencia que durante todos estos años me ha mostrado. Sin su cariño y comprensión la realización de la tesis hubiera sido sin duda un proceso mucho más tortuoso. También aprovecho para agradecer a mis padres que en todo momento me incentivaran a llegar siempre lo más lejos posible en la formación académica. En numerosas ocasiones a lo largo de la tesis me he preguntado si todo esto realmente vale la pena. Ahora con perspectiva y la tesis bajo el brazo todo se ve distinto.

Por otro lado me gustaría agradecer a Federico Silla, Vicente Santonja y José Duato la oportunidad que me brindaron para realizar el doctorado en Informática en el Grupo de Arquitecturas Paralelas. Quisiera agradecer especialmente a Fede la ayuda que me ha prestado durante estos años y que ha sido fundamental para la realización de la tesis.

Aunque una tesis supone un importante sacrificio personal, no hay que obviar que para la realización de la misma he contado con la inestimable ayuda de muchos compañeros de laboratorio. Durante la etapa inicial de mi doctorado fueron Ricardo, Blas, Crispin, Gaspar, Andres, Paco, Hector, Rafa, David y Noel mis compañeros en el GAP. Con ellos he compartido buenos momentos dentro y fuera del laboratorio. Recuerdo con especial cariño las innumerables ocasiones en las que Hector y yo (al principio con la compañía de David) maldijimos el día en el que decidimos hacer la tesis. Cinco años después esta etapa ha acabado y una nueva empieza en Barcelona. Más tarde

la incorporación de Toni, con el que ya compartí buenos momentos en los trabajos de Microelectrónica de Teleco, ha sido un apoyo fundamental a la hora de afrontar los problemas que desde el lado oscuro, en el cual mi tesis se ha desarrollado, han ido surgiendo. Finalmente nuevos compañeros de laboratorio como Javi, Knut, Mario, Alberto... han hecho posible que el ambiente del laboratorio siga tan bueno como siempre.

Por último no podía olvidarme de agradecer a mis amigos; a los que son heavies y a los que son flamencos, a los que disfrutan de la escalada y a los playeros, a los de la facultad y a los que no han estudiado, que hayan hecho que el tiempo que ha durado la tesis pase tan rápido. Sin duda he disfrutado mucho durante todo este tiempo.

Contents

List of Figures

List of Tables

Abstract

Advances in manufacturing technologies enable the integration of a growing number of transistors on a single chip, devoted mainly to increase the number of cores and cache memory. As the number of cores increases, the communication infrastructure becomes critical. In this context, networks on chip (NoCs) have been chosen as the interconnect choice for current and future Multiprocessor System-on-Chip (MPSoCs) and Chip Multiprocessor (CMPs) systems. The main benefits of NoC-based architectures are higher performance and predictability than previous bus-based designs and larger scalability than crossbar-based designs.

Unfortunately, the same large integration scales that allow to include a high number of components in the same die are also compromising the reliability of designs. In this context process variations arise as one of the main challenges to be faced by designers, and is likely to be the most important source of unreliability for technologies below 45nm. In the context of CMPs and MPSoCs, manufacturing faults may appear in the form of defective cores, memories, links, or switches. For example, due to the small feature size dopant fluctuations or lens aberrations may cause circuits to completely malfunction or to present different behaviour than the expected one.

Our efforts in this dissertation have been first focused on the development of a detailed process, voltage, and temperature (PVT) variability model that improves the current state-of-art modeling of process variations in the computer architecture community. By leveraging the model developed in this thesis the impact of process variations in different NoC-based designs has been characterized. Results of the analysis of different NoC designs reveal that on one hand, systematic variations considerably impact both NoC and processor performance. As a consequence of systematic variations the maximum achievable frequency of CMP components varies considerably across the chip surface. In this sense, systematic variations cause, if ignored, both the reduction of NoC performance and the inefficient utilization of CMP resources. On the other hand, random variations severely impact the performance of NoC links for technologies below 32nm. In addition, higher defect density levels are expected for smaller technologies causing the probability of having faulty wires in NoC links to increase. In this context, efficient variability-aware and fault-tolerant NoC link architectures are required.

In this dissertation we propose to face the impact of systematic variations in the energy and performance of CMPs by means of a variation-aware mapping policy. The proposed mapping policy efficiently schedules applications in CMP systems under process variations. In this proposal, performance is achieved by first considering fast CMP regions and efficiency is provided by choosing regions presenting uniform frequencies. The goal of this mapping strategy is avoiding to map threads to regions where routers and cores present very different speeds, as this would cause both an inefficient utilization of resources and the induction of communication bottlenecks in the NoC.

In order to mitigate the impact of random variations and the presence of defective wires in NoC links, a variable phit-size NoC architecture is proposed in this dissertation. The proposed fault-tolerant NoC architecture addresses the impact of both manufacturing defects and random variations in NoC links by performing a suitable reduction of phit size at network interfaces. By leveraging the variable phit size NoC architecture, the link bandwidth retrieved is maximized.

Finally, in this thesis a novel area-efficient vertical link design is also presented. The proposed fault-tolerant vertical link design is able to noticeably increase the yield of three dimensional Integrated Circuits (3DICs) while keeping performance and minimizing the amount of required resources. Actually, our new proposal significantly reduces the amount of Through-Silicon-Vias (TSVs) per link without reducing performance. In comparison with a conventional N-wide vertical link, our proposal requires $N/2 + m$ TSVs, where m is significantly smaller than $N/2$ (m is usually one or two TSVs). Thus, deploying our proposal reports noticeable area savings.

Resumen

El continuo desarrollo del proceso de fabricación de circuitos integrados permite la integración de un elevado número de transistores en un mismo chip. Este excedente de transistores es generalmente destinado a aumentar el número de procesadores y la capacidad de la memoria cache. Conforme el número de elementos a interconectar dentro de un mismo chip crece la interconexión de los mismos se vuelva crítica. En este contexto, las redes dentro del chip surgen como la solución más eficiente para la interconexión de los distintos elementos dentro del chip. Los principales beneficios de este tipo de arquitecturas son una mayor predictibilidad y escalabilidad que otro tipo de arquitecturas de interconexión basadas en buses o crossbars.

Desafortunadamente, la misma capacidad de integración que permite a los diseñadores incluir un elevado número de componentes en el chip, está comprometiendo a su vez la fiabilidad de los sistemas. En este contexto la variabilidad asociada al proceso de fabricación de los circuitos se está convirtiendo en uno de los principales retos a los que deben hacer frente los diseñadores de circuitos en tecnologías por debajo de los 45nm. En el contexto de los chip multiprocesadores los fallos en la fabricación aparecen en forma de núcleos, encaminadores o enlaces defectuosos.

Los esfuerzos de esta tesis se centran en primer lugar en la elaboración de un modelo detallado y preciso del comportamiento de los circuitos en presencia de variaciones en el voltaje, la fabricación, y la temperatura de los circuitos integrados que mejora las características de los modelos disponibles en el área de arquitectura de computadores. Con el desarrollo de dicho modelo, el impacto de la variabilidad del proceso de fabricación en los circuitos interconectados mediante redes en el chip se ha podido analizar y cuantificar. Los resultados del análisis de diferentes diseños basados en redes en el chip revelan por un lado que la variabilidad sistemática afecta considerablemente a las prestaciones de la red y los procesadores. Como consecuencia de la elevada variabilidad sistemática, la máxima frecuencia de operación de los componentes de los procesadores multi-núcleo varía considerablemente a lo largo de la superficie del chip presentando una elevada correlación espacial. En este sentido, las variaciones sistemáticas causan, si no son tenidas en cuenta, una reducción de las prestaciones de la red y la ineficiente utilización de los recursos de los chip multi-núcleo. Por otro lado se ha observado que las variaciones aleatorias impactan considerablemente las prestaciones de los enlaces de la red para tecnologías por debajo de 32nm. Adicionalmente predicciones de los principales fabricantes de circuitos auguran a su vez aumentos en la densidad de defectos de fabricación a medida que se reduce el tamaño de los transistores. El aumento de la densidad de defectos y del impacto de las variaciones aleatorias en los niveles de dopante en los enlaces de las redes dentro del chip hacen imprescindible la elaboración de diseños de redes dentro del chip capaces de hacer frente tanto a la variabilidad como a la presencia de fallos de fabricación.

En esta tesis proponemos hacer frente al impacto de la variabilidad sistemática en la energía y rendimiento de los procesadores multi-núcleo utilizando una política de mapeo de aplicaciones que tenga en cuenta los efectos de la variabilidad. La política de mapeo propuesta asigna de forman eficiente aplicaciones en el procesador en la presencia de variabilidad. En este algoritmo de mapeo las mejoras de rendimiento se obtienen eligiendo en primer lugar los núcleos más rápidos, mientras que la eficiencia se consigue formando regiones homogéneas. El objetivo de esta estrategia de mapeo es evitar el mapeo de hilos de ejecución en regiones donde los encaminadores y núcleos presentan considerables diferencias en la frecuencia de operación ya que esto causa por una lado la ineficiente utilización de recursos y por el otro lado la aparición de cuellos de botella que deterioran las prestaciones de la comunicación en la red en el chip.

Con el fin de mitigar el impacto de la variabilidad aleatoria y la presencia de cables defectuosos en los enlaces de la red, en esta tesis proponemos el diseño de una arquitectura de red con tamaño de enlace variable. La arquitectura tolerante a fallos propuesta es capaz de hacer frente tanto a los errores de fabricación como a los errores de temporización causados por el aumento de la variabilidad del retardo en los cables. Con el uso de la arquitectura propuesta el tamaño de los paquetes en el nivel físico es ajustado en los interfaces de red. Mediante la utilización de esta arquitectura una gran parte del ancho de banda de los enlaces defectuosos es recuperado.

Finalmente en esta tesis se propone un nuevo y eficiente diseño de enlace vertical tolerante a fallos. El enlace tolerante a fallos propuesto es capaz de reducir en número de chips desechados en los chips tridimensionales. El diseño propuesto reduce el número de conexiones verticales por enlace sin reducir el rendimiento global de la red. En comparación con un enlace vertical de ancho N, nuestro diseño requiere $N/2 + m$ conexiones verticales, donde m es considerablemente menor que $N/2$ (generalmente 1 ó 2). Con esta reducción del número de conexiones verticales nuestra propuesta proporciona tolerancia a fallos al mismo tiempo que reduce considerablemente el área total del chip.

Resum

El desenvolupament del procés de fabricació de circuits integrats permet la integració d'un elevat nombre de transistors en un mateix xip. Aquest excès de transistors es generalment destinat a incrementar el nombre de processadors i la capacitat de les memòries cache. A mesura que el nombre d'elements a interconnectar en un mateix xip creix, la interconnexió dels mateixos es torna critica. En aquest context les xarxes dins del xip sorgeixen com a la solució més eficient per a la interconnexió dels diversos elements dins del xip. Els principals beneficis d'aquest tipus d'arquitectures son l'augment de la predictibilitat i l'escalabilitat front a altres tipus d'arquitectures d'interconnexió basades en busos o crossbars.

Malauradament la mateixa capacitat d'integració que permet als dissenyadors de xips incloure un elevat nombre de components en un mateix xip, esta posant en compromís la fiabilitat dels mateixos sistemes. En aquest context la variabilitat associada al procés de fabricació de circuits integrats afloreix com un dels principals reptes a enfrontar en les tecnologies sota 45nm. En el context dels xip multiprocessadors les fallades en la fabricació apareixen en forma de nuclis, encaminadors, o enllaços defectuosos.

Els esforços d'aquesta tesi s'han centrat en primer lloc en l'elaboració d'un model detallat i precís del comportament dels circuits en presencia de variacions en el voltatge, fabricació i temperatura dels circuits integrats. Aquest model es per si mateix una notòria contribució ja que suposa una millora de les diverses ferramentes disponibles en l'àrea d'arquitectura de computadors. Mitjançant el desenvolupament d'aquest model, l'impacte de la variabilitat del procés de fabricació de circuits integrats basats es xarxes en el xip s'ha pogut analitzar i mesurar. Els resultats de l'anàlisi de diferents dissenys basats en xarxes en el xip mostren per un costat que la variabilitat sistemàtica afecta de forma considerable a les prestacions de la xarxa i els processadors. Com a conseqüència de l'elevada variabilitat sistemàtica, la freqüència d'operació dels diversos components dels processadors multi-nucli varia considerablement al llarg de la superfície del xip presentant una elevada correlació espacial. En aquest sentit les variacions sistemàtiques causen si son ignorades tant la reducció de les prestacions de la xarxa com una utilització ineficient dels recursos del xip multi-nucli. Per altra banda s'ha observat que les variacions aleatòries en els nivells de dopant causen un considerable impacte en les prestacions dels enllaços de la xarxa en el xip sota 32nm. En aquest sentit les prediccions dels principals fabricants de xips auguren un augment en la densitat de defectes associat a la reducció del tamany dels transistors. L'augment de la densitat de defectes i el notori impacte de les fluctuacions en els nivells de dopant en les xarxes en el xip fan imprescindible el desenvolupament de dissenys de xarxes en el xip capaços de tolerar tant la variabilitat del procés de fabricació com la presencia de defectes.

En aquesta tesi proposem fer front al impacte de la variabilitat sistemàtica en l'energia i rendiment dels processadors multi-nucli emprant una política de mapeig d'aplicacions que tinga en compte els efectes de la variabilitat en els diversos components del xip. La política de mapeig proposta assigna de forma eficient aplicacions en el processador en presencia de variabilitat. En aquest algorisme de mapeig les millores de rendiment s'obtenen prioritzant l'ús dels nuclis més ràpids al mateix temps que l'eficiència s'assoleix formant regions homogènies. L'objectiu d'aquesta estratègia es evitar el mapeig de fils d'execució en regions que continguen nuclis i encaminadors amb considerables diferencies en la freqüència d'operació. D'aquesta manera s'evita per un costat la ineficient utilització de recursos i per l'altre l'aparició de colls de botella que deterioren les prestacions de les comunicacions en la xarxa.

Amb l'objectiu de mitigar el impacte de la variabilitat aleatòria i la presencia de cables defectuosos en els enllaços de la xarxa, en aquesta tesi proposem el disseny d'una xarxa amb un tamany d'enllaç variable. L'arquitectura tolerant a fallades proposta es capaç de fer front tant als errors de fabricació com als errors provocats per l'augment de la variabilitat del retard en els cables. D'aquesta manera el nou diseny ajusta de forma adequada el tamany dels paquets de dades en el nivell físic en les interfícies de xarxa. Mitjançant la utilització d'aquesta arquitectura gran part de l'ample de banda dels enllaços defectuosos es recuperat.

Finalment aquesta tesi proposa un nou disseny eficient d'enllaç vertical tolerant a fallades. L'enllaç vertical proposat permet reduir el nombre de xips defectuosos en la fabricació de circuits tridimensionals. El disseny realitzat permet reduir el nombre de connexions verticals per enllaç sense reduir el rendiment global de la xarxa. En comparació amb un enllaç vertical de tamany N, el nostre disseny requereix $N/2+m$ connexions verticals, sent m considerablement menor que $N/2$ (generalment 1 o 2). Amb aquesta reducció del nombre de connexions verticals aconseguim tant assolir un disseny tolerant a fallades com la reducció de l'area total del xip.

Chapter 1

Introduction and Motivation

Advances in manufacturing technologies enable the integration of a growing number of transistors on a single chip, devoted mainly to increase the number of cores and cache memory. As the number of cores increases, the communication infrastructure becomes critical. In this context, networks on chip (NoCs) [25] have been chosen as the interconnect choice for current and future Multiprocessor System-on-Chip (MPSoCs) and Chip Multiprocessor (CMPs) systems. The main benefits of NoC-based architectures are higher performance and predictability than previous bus-based designs and larger scalability than crossbar-based designs. Recent designs from major chip manufacturers include a NoC to interconnect cores and memories [40, 102]. Among the different topologies used in NoCs, the 2D mesh topology is commonly accepted as it perfectly matches the chip surface and suits the tile based design approach, where a tile containing a processing core with its associated L1 and L2 cache levels and the corresponding NoC switch is initially designed and later replicated all over the die.

Unfortunately, the same large integration scales that allow to include a high number of cores in the same die are also compromising the reliability of designs. In this context, process variations arise as one of the main challenges to be faced by designers, and is likely to be the most important source of unreliability for technologies below 45nm [27]. In the context of CMPs and MPSoCs, manufacturing faults may appear in the form of defective cores, memories, links, or switches. For example, due to the small feature size, dopant

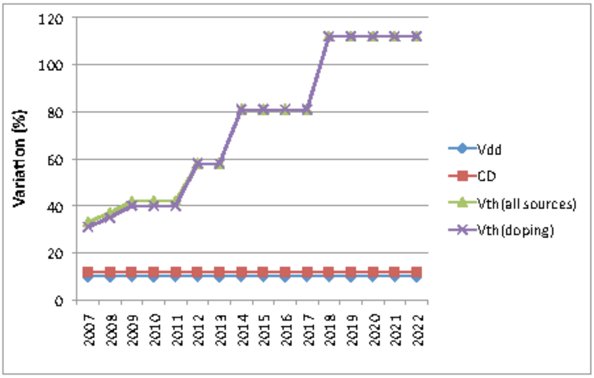


Figure 1.1: Evolution of Parameter Variations

fluctuations or lens aberrations may cause circuits to malfunction completely or deviate from nominal performance/power figures [29]. Figure 1.1 shows the evolution of the main parameter variations as predicted by the ITRS [27]. On one hand, it is possible to see that the critical dimension variation (CD), that is transistor channel length, is already under control and will remain below 12% [1]. The same occurs in the case of the supply voltage (V_{dd}) where variations will be lower than 10%. On the contrary, the overall variation will remain increasing due to the contribution of threshold voltage variations, and more precisely, due to random dopant fluctuations. Note that random dopant fluctuation is an inherent consequence of the reduced average number of dopant atoms in the transistor channel, expected to be lower than 100 for technologies below 32nm [45].

As mentioned before, parameter variation is becoming a major concern that cannot be ignored at any system level, as unpredictable variations affecting silicon devices and metalizations in the bottom layer are directly impacting performance at the upper levels, even influencing application execution time [100]. In between, the system architecture and operating system are also affected by the uncertainty coming from the lowest layer of the system. Moreover, energy efficiency at any of the system levels may also be affected by parameter variations appearing down in the silicon level. In this way, if

[1] Variations in percentage are computed as $3\sigma/\mu$ where σ and μ represent the standard deviation and the mean value of a given parameter, respectively.

ignored, parameter variations may cause the benefits of shifting to smaller technology nodes to be canceled due to an inefficient exploitation of the underlying silicon devices by the upper levels [8]. This inefficient use of silicon resources will likely become more noticeable as process variation is increased when larger integration scales are in use.

Process variations are typically decomposed into die-to-die variations and within-die variations. Die-to-die variations account for variations arising among chips, whereas within-die variations account for variations among devices and interconnects within the same chip [35]. Within-die variability sources can be further divided into front-end and back-end ones. On one hand, the front-end phase of the integrated circuit (IC) fabrication process is related with the steps involved in the creation of devices. Front-end process variation can be further decomposed into systematic and random components. Systematic variation is caused by deviations in the photolitographic process and are generally characterized by a strong spatial correlation, causing differences in neighbour areas to be low. On the contrary, random variation, caused, for example, by dopant level fluctuations, cause different operation characteristics in adjacent areas. On the other hand, the back-end stage comprises steps involved in the wiring definition. In this sense, the main back-end variation sources are capacitance and resistance variations due to imperfections introduced by the chemical metal planarization process. The exact way back-end variation affects a circuit depends on the actual dimensions of the metalizations considered. For example, variations introduced by the chemical metal planarization process in combinational circuits are negligible because wires connecting logic gates are quite short, and its contribution to the total delay is minimum. On the other hand, in the case of NoC links, where metalizations are much longer, the influence of back-end variability may be more noticeable.

As mentioned before, networks-on-chip are today advocated as a scalable interconnect fabric for multi-core processors, overcoming the limitations of shared-bus structures. Therefore, not only fault-tolerance is becoming a critical requirement in designing modern processors, but the on-chip networking scenario is raising new challenges for fault-tolerance and variability-aware design. Because of variation-induced performance asymmetry, the post-silicon NoC architecture will present a non uniform behaviour as the maximum achievable frequencies of its components will differ from the projected one at design time. The first approach to face this uneven distribution of maximum achievable frequency of NoC modules is to slow down the frequency of all modules to that of the slowest element. However, this is not acceptable when delay uncertainty increases. In this regard, to avoid the performance loss caused by keeping synchronism, designers adopted the Globally-Asynchronous-Locally-Synchronous (GALS) philosophy in order to keep performance in such scenarios featuring different frequencies. In the context of NoC-based CMPs, Frequency Islands (FI) have been proposed to face process variations [69] [36]. Actually, FI is a way of implementing the GALS philosophy. By leveraging a FI design approach, different chip domains are able to work at different frequencies, and communication between domains is performed by means of synchronizers. In a FI-based design, variability-awareness is achieved by setting island frequency to the maximum sustainable frequency by the elements in that island. However, as we will show throughout this dissertation, the use of FI does not solve the variability problems, as having components in a NoC working at very different speeds cause a considerably reduction in the overall performance of the network. Additionally, if regions are built mixing components with very different operating characteristics, chip resources will be inefficiently utilized.

Finally, as the number of cores increases, access to memory (in terms of memory bandwidth) is predicted to become a major limiting factor for CMP and MPSoC designs. Therefore, to partially alleviate the problems caused by the pin-out limitations 3DICs have been proposed [81]. The adoption of 3DICs also compromises the reliability and yield of NoC-based chips. In 3DICs, dies are stacked on top of each other and vertical connections are established among them. One of the most promising technologies to enable vertical links between dies is the use of Through-Silicon-Vias (TSVs). However, one of the main drawbacks when deploying TSVs is their high defect rate, specially when compared with traditional 2D wires [55]. Briefly, TSV-based vertical links are exposed to misalignment and random defects. The first kind of failures are introduced in the alignment process of stacked wafers as a consequence of bonding pad shifting [80]. On the other hand, random defects are a consequence of several unpredictable phenomena where most of them are related with the thermal compression process used in the wafer stacking process [55].

1.1 Thesis contributions

This thesis presents several contributions aimed to address manufacturing issues affecting NoC-based designs. In this sense, one of the main contributions of this thesis is a comprehensive detailed model of process, voltage, and temperature (PVT) variations. By leveraging this model the implications of parameter variations in the power and performance of designs can be analyzed. The remaining contributions of this thesis are devoted to face the presence of process variations and manufacturing defects in NoC-based ULSI circuits.

The complete list of main contributions of this thesis are:

- A *framework for injecting PVT variations* into synthesized designs. This framework performs an accurate modeling of process, voltage, and temperature variations. The process variation model presented in this work improves the current state-of-the-art modeling tools available in the computer architecture community. On one hand, instead of using a simplified model of the critical path to inject variations, the whole circuit is taken into account. On the other hand, both devices and wires are considered to accurately analyze the impact of variations in NoC-based designs.

- In this thesis the importance of the impact of parameter variations in NoC-based designs is shown. While some previous studies neglect the impact of variations in the NoC, in this thesis we show the importance of variations in both routers and links and, consequently, in the performance of the whole network. Additionally, we prove the superior robustness of mesochronous clocking schemes against process variation in comparison with traditional synchronous clocking schemes.

- A *variation aware mapping policy* to mitigate the impact of within-die variations of parallel workloads in chip MultiProcessors. The proposed mapping policy provides high efficiency in both energy and performance of applications running on the CMP.

- As technology scales down both variation induced timing errors and manufacturing defects will cause wires of NoC links to malfunction. In this regard, in this thesis a new mechanism able to face the presence of faulty wires in NoC links has been proposed. This proposal (*a variable phit-size NoC architecture*) is based on discarding those wires that are either defective, caused by for example open or shorts, or slower as a consequence of process variations.

- Finally, an *effective fault-tolerant vertical link* has been designed. The vertical link designed is intended for facing reliability problems of vertical link interconnects built with TSVs. The proposed design reduces the area requirements of regular TSV-based vertical link designs while the yield of 3D chips is considerably enhanced.

1.2 Thesis overview

In this dissertation we first propose a detailed process, voltage, and temperature variability model. The proposed modeling of process variations improves the current state-of-art modeling of process variations in the computer architecture community. Thanks to this accurate modeling we are able to quantify the impact of variation in different NoC-based designs. After collecting the measurements of the impact of variations we are in a position to propose different mechanisms to deal with the presence of variations. The first proposed technique is a variation-aware mapping policy intended to face within-die systematic variations affecting the energy and performance of applications running on CMPs. Later, to face variation induced timing failures and permanent faults in NoC links a variable phit size NoC architecture is proposed. The proposed architecture discards slow and faulty wires of links. As a result, link throughput is maximized. Finally, in order to face the presence of defective TSVs in vertical links a novel area-efficient vertical link design has been designed. The proposed vertical link is able to achieve very good yield results at the same time that reduces the area of regular vertical links. All this work is structured into the following dissertation layout:

- Chapter 2 presents the necessary background of both manufacturing issues and chip designs in the ULSI era. Concretely, the basis of the NoC paradigm is presented. Later, the evolution of clocking schemes is discussed. Finally, regarding manufacturing issues, the fundamentals of both parameter variations and 3D stacking technology are presented.

- Chapter 3 introduces the variability model developed in this thesis. In this chapter the different sources of unpredictability considered are carefully detailed. Finally, the model is tested using different circuit benchmarks.

- Chapter 4 shows the impact of process variations in 3 different NoC-based design scenarios. In the first part of this chapter the impact of process variations in NoC links is described. Later, it is measured the impact of process variations in the whole NoC and, finally, the impact of variations in two different NoC approaches is analyzed.

- Chapter 5 presents an application mapping policy that is able to partially overcome the negative impact of within-die systematic variations in the energy and performance of CMPs.

- Chapter 6 proposes a variable phit-size NoC architecture that is able to face the presence of both variation induced timing errors and manufacturing defects affecting NoC links.

- Chapter 7 presents an effective fault-tolerant 3D link design for facing the yield reduction caused by defective TSVs in vertical NoC links.

- Chapter 8 summarizes the main conclusions of this dissertation. In this chapter future directions for continuing the research related to this dissertation are also provided.

Chapter 2

Background

In this chapter the necessary background about current processor design is surveyed. First, some indications about how chip designs have been adapted to the ULSI scenario are given. Later, current and future challenges of chip design associated with the miniaturization of the fabrication process are analyzed.

2.1 Processor Design in the ULSI era

Associated with the adoption of the nanometric design of integrated circuits are two main concerns: the first one is the increase in the complexity of designs and the second one is the reliability of those designs. Both concepts are tightly coupled and their importance is severely exacerbated in the context of Ultra-Large-Scale-Integration (ULSI) designs. On one hand, the higher integration capabilities of current chips allow designers to include a high number of modules in the same die increasing the performance and also the complexity of the resulting designs. The high number of modules working inside the chip makes necessary to explore new communication schemes, as we will see later. On the other hand, the combination of the increase in the relative size of dies and the reverse scaling of delay interconnects makes necessary the use of new clocking schemes in order to control power consumption. In this sense, the Globally-Asynchronous-Locally-Synchronous (GALS) paradigm is presented as the most suitable scheme for very complex system designs.

2.1.1 The Network-on-Chip Paradigm

As the number of cores of MPSoCs and CMPs is considerably increased, Networks-on-chip (NoC) are adopted to be the most suitable interconnection infrastructure for the sake of scalability and performance. In this kind of architectures, communication is performed by means of switches and links. In the interconnection network design several choices like the topology, routing, and flow control are generally imposed by restrictions of resources and determine the performance of the communication. In the particular case of NoCs, most of the current real NoC designs [102] [40] implement 2D-mesh topologies because this topology perfectly matches the spatial features of the chip. However, as stated in [3] the 2D mesh NoC topology does not scale performance very well and tends to concentrate traffic in the center of the network. Therefore, several recent works in the open literature propose optimized NoC topologies while keeping regularity properties as much as possible. In this way, Gilabert et al. propose in [33] to use high-dimensional topologies finding a trade-off between the number of cores per router and the delay of long links. In the same direction, the authors in [3] propose the use of an enhanced concentrated mesh architecture using replicated sub-networks and express channels.

Regarding the switch architecture, the NoC scenario imposes severe area and power constraints. For this reason, for implementing the switch, wormhole switching flow control mechanisms are preferred [18]. High-speed switch designs are achieved by keeping switch radix low, that helps to relax the crossbar and arbitration constraints, and by using pipelined switch architectures. Figure 2.1 shows the different stages of a canonical pipeline implementation [21]. The stages involved in the regular router operation are the initial buffering (BUF), the packet routing (RC) that is just performed on the header flit, the switch allocation (SA), the crossbar traversal (XT), and the link traversal stage (LT). Figure 2.2 shows the arrangement of the required modules for implementing a canonical switch. Note that several different modifications can be applied to this canonical switch for achieving improved performance. For example, to spare the cycle penalization of routing computation, look-ahead routing mechanisms have been proposed [32]. Router buffering is used to store arriving flits that cannot be immediately forwarded onto output links because of contention. Different alternatives exist to perform buffering like input-

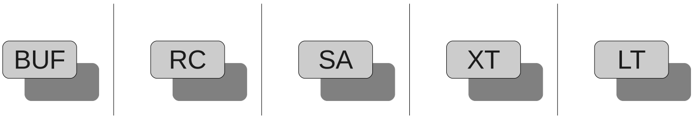


Figure 2.1: Baseline Switch Pipeline

buffered router, output-buffered router, or more sophisticated approaches like distributed shared buffering [96]. In this dissertation different switch architectures are employed for the evaluation of the different proposals. The exact details of the different switch implementations are detailed in the corresponding proposal evaluation section.

Finally, routing mechanisms have also been adapted to meet the particular constraints imposed by the chip. In this sense, chip area restrictions make the use of logic-based routing implementations more appealing than table-based routing approaches [26]. The X-Y routing algorithm is an example of a logic-based routing mechanism that can be employed to route over 2D-meshes. However, the arise of manufacturing defects demands for fault-tolerant routing schemes that help designers to increase chip yield [88]. Recently, logic-based fault-tolerant routing mechanisms have been proposed allowing to route packets in 2D-meshes presenting some faulty links [95] [26].

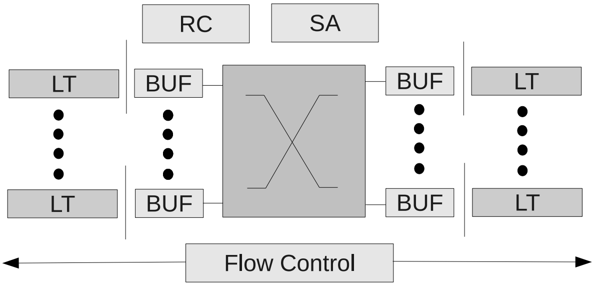
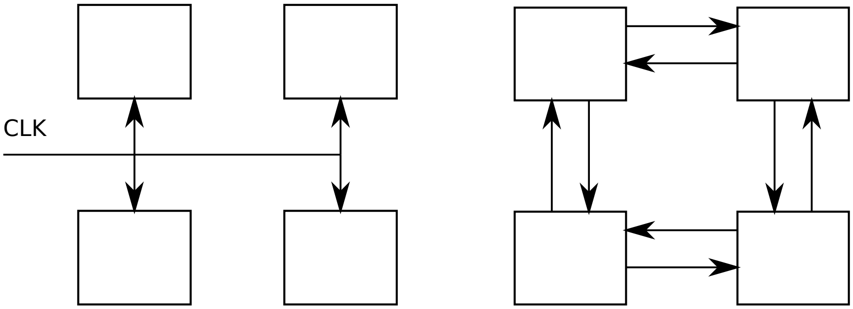


Figure 2.2: Canonical Router Architecture

2.1.2 From GALS to Voltage and Frequency Island Design

As mentioned before, technology feature size becomes smaller with each generation making the traditional synchronous design technique not feasible any more. Notice that the reverse scaling of wire delay increases interconnect de-



(a) Synchronous System

(b) GALS system

Figure 2.3: Comparison of GALS and synchronous systems

lay making the cost of propagating the clock signal across the whole die to become unaffordable. In this context, the GALS paradigm reduces the cost of propagating clock signals as synchronism is only required in local regions. Figure 2.3 shows an schematic of synchronous and GALS systems. As shown in Figure 2.3(a) the synchronous design enables the interaction between modules by means of a unique clock signal that keeps the synchronism in the whole system. On the contrary, as shown in Figure 2.3(b), in the case of GALS systems, communication between regions must be performed explicitly. To do so, synchronizers are required. The complexity of synchronizers depends on the nature of the different clock domains. Different kind of clocking schemes can be used [57], however, its study is out of the scope of this thesis.

The GALS paradigm has recently been adopted by the industry. In this sense, there are already some working examples of Multiclock-chip-Domains (MCD) designs by the main processor manufacturers. For instance, a per-core frequency control was included in AMD's Opteron Quad-Core [20]. Another cost-effective way to provide the appropriate clock signal is the one used in the Montecito CPU by Intel [24]. In this approach only one global PLL (phase loop lock) is required and clock dividers are used to provide the range of supported frequencies. Note that MCD chips are a particular application of the Globally-Asynchronous-Locally-Synchronous philosophy.

Once the GALS paradigm was adopted, designers went one step further and proposed the use of voltage and frequency islands to increase performance and power efficiency. In this kind of architectures, voltage and frequency islands

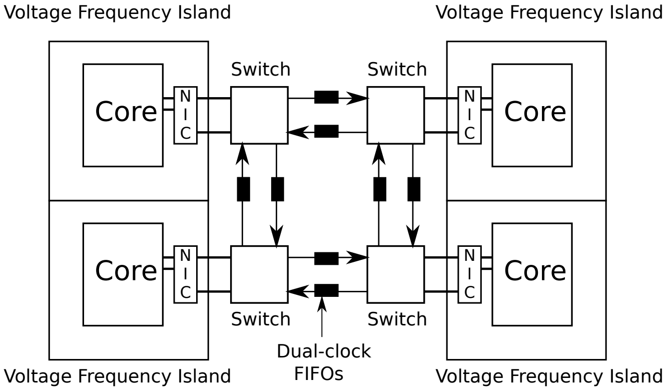


Figure 2.4: Implementing VFI design NoC-based designs

(VFI) or regions are built making possible to dynamically configure different parts of the chip to work at different voltages and speeds. VFI architectures have been proved to increase power and performance efficiency [67]. In this kind of architectures each island is a locally synchronous region operating with its own clock. Dynamic voltage/frequency scaling is performed in each synchronous region using DC-DC voltage regulators and appropriate clock generators. There are examples of already working processor chips provided with the capability of building voltage and frequency islands. For example, the Intel Nehalem architecture [48] is able to perform dynamic voltage and frequency scaling of cores. In the same way the 4-core AMD Opteron [20] is also able adjust the voltage and frequency of individual cores dynamically.

In a FI-based design, variability-awareness is achieved by setting island frequency to the maximum sustainable frequency by the elements in that island. In order to know the clock frequency of each domain, test vectors are applied to chips after the manufacturing process. To do so, the regular test process has to be extended in order to determine the maximum achievable frequency of each domain. Tests have been traditionally performed to bind microprocessors to some frequency level in order to sell them according to their speed. Finally, once the maximum clock domain frequency is found for every frequency island, these data can be recorded into dedicated ROMs. In the context of NoC designs, the use of VFI schemes has been proposed first to achieve energy efficiency [69] and later to face process variations [36]. As in

VFI systems, different chip domains are able to work at different frequencies and different voltage, both synchronizers and voltage converters are required to enable region communication. In the particular case of NoC architectures the most adopted solution to include synchronizers is to replace the regular router input buffers by dual-clock FIFOs [61]. Figure 2.4 shows a VFI NoC-based architecture. However, notice that despite VFI designs behave better than synchronous designs in the presence of process variations they are still unable by themselves to efficiently exploit chip performance as we will shown later in this Thesis.

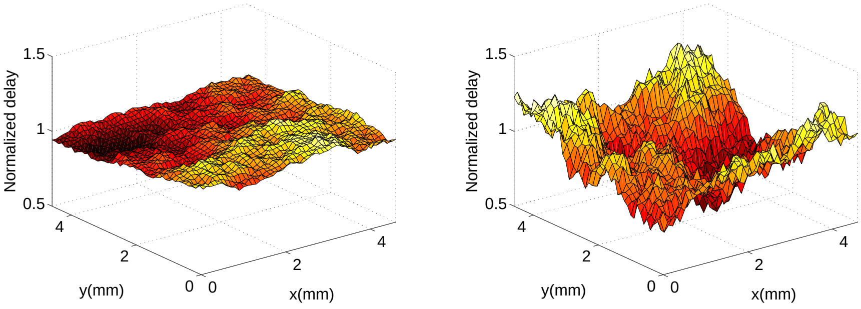
2.2 Manufacturing challenges in the ULSI era

The manufacturing process becomes more challenging with technology scaling. On one hand, as mentioned before, parameter variations considerably impact circuit power and performance for technologies below 45nm [45]. On the other hand, several factors make reliability to be severely compromised in the nanometric era. First, as a consequence of delay variations and power uncertainty of circuits, they could fail. Second, due to the increase of power density, thermal effects may cause circuit operation to fail. Finally, the adoption of new manufacturing scenarios like 3D stacking may involve a considerable yield reduction when fault-tolerant mechanisms are not included in chips [55].

2.2.1 Process Variations

As transistors and wires are reduced in size, it becomes harder to control process variations affecting key physical parameters such as transistor channel length, threshold voltage, and metalization dimensions. The deviations introduced by process variations cause an increased variability in circuit performance and are likely to be more dominant for technologies below 45nm [45]. Parameter variation is not a new concern since its presence is inherent to the manufacturing itself. However, during several decades the impact of variations in processor design has been first controlled and later mitigated as we will see later.

Process variations can be decomposed into Within-Die (WID) and Die-to-Die (D2D) parameter variations. Die-to-die variations cause differences in



(a) Low variation

(b) High variation

Figure 2.5: Normalized delay variation caused by lens aberrations in a 45nm 4.7mm x 4.7mm die

parameter values across nominally identical dies. The typical way to model these variations is by shifting the mean of some parameter value (e.g., wires or transistors dimensions) for all devices in a chip in the same way [5]. For circuit design purposes it is usually assumed that each contribution to die-to-die variation is due to different physical and independent sources and, consequently, these contributions can be grouped together into a single effective die-to-die variation component with a single mean and variance [5] . On the contrary, within-die variation is the deviation of parameters included in the same die. WID variations make transistors and wires within the chip, initially designed to be identical, to present considerable differences in the operating characteristics. In contrast to D2D variation, WID variation contributes to the loss of matched behaviour between structures on the same chip. This makes the behaviour of manufactured chips more difficult to predict. Figure 2.5 shows delay variation of gates located at a given chip location caused by transistor channel length variation. These variations are strongly spatially correlated as transistor channel length variation is caused by lens aberrations. Figures 2.5(a) and 2.5(b) represent scenarios with low and high variation, respectively. Note that in case of low variations gate delay variations change slower. Additionally, Figure 2.5 illustrates why systematic transistor channel length variations are becoming more important as technologies scale down. On one hand, the expected amount of variations affecting physical transistors increases with technology scaling, and this results in an increase of circuit performance variability. On the other hand, as more devices can be printed in the same chip, differences between devices at different chip locations become more noticeably.

Facing Process Variations

Traditionally, to remedy the effects of process variations speed-binning has been used. Speed-binning is usually performed by testing each manufactured chip separately over a range of frequency levels until it fails. As a result of the inherent process variations, the different processors fall into separate speed bins, where they are rated and marketed differently [16]. This process helps the chip manufacturer to create a complete product line from a single design. However, this mechanism that was effective for facing D2D variations results, obviously, inefficient for facing WID variations.

The other common approach to face timing variations is to add worst-case guard-bands to critical paths [6]. However, this comes at the expense of a high reduction in performance when severity of variations is noticeable. As a consequence of parameter variations, like the threshold voltage, transistor channel length, or wire dimensions, the critical path delay of a circuit can be represented as a probability density function (Figure 2.6). In Figure 2.6, two probability density functions corresponding to different manufacturing processes are shown. On one hand, it is possible to see how when variations are low, the narrowest shape, the performance loss incurred by reducing the frequency of the circuit to that of the slowest path (worst-case) is not very significant. However, when delay variation increases, the wider shape, designing for the worst-case results ineffective. In this sense, a more refined technique is using statistical delay calculation tools in order to reduce design margins and improve design speed [68]. This probabilistic framework avoids the pessimistic timing estimation introduced by the classical static timing analysis [72].

In the following chapters of this dissertation we will go deeper into the modeling of process variations and the impact of parameter variations in circuit performance. First, we corroborate that new design approaches are required to face the presence of variations. Later, we propose new techniques to mitigate the negative impact of process variations in both power and performance of circuits.

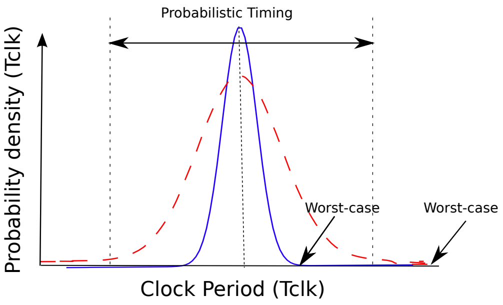


Figure 2.6: Example of several different Gaussian density functions used for choosing the target operating frequency

2.2.2 3D Stacking Technology

There are also other important challenges compromising chip design that are associated with the adoption of new manufacturing scenarios. On one hand, the small size of transistors causes that, due to the increase of power density, thermal effects arise leading to operational malfunctions. Note, however, that despite the modeling framework presented in Chapter 3 also accounts for the delay uncertainty caused by temperature variation, the effects of temperature in circuit behaviour are out of the scope of this dissertation. On the other hand, as the number of components in the chip considerably increases, the pin-out limitation is becoming a critical issue. To solve this, the use of stacked dies have been proposed. Associated with this new manufacturing scenario is the yield reduction of resulting 3D chips. Hereafter, the current state-of-art of 3D chips is surveyed.

In 3DICs, dies are stacked on top of each other and vertical connections are implemented using TSVs. Figure 2.7 shows an schematic of a typical multi-layer 3D chip implementation. The number of required TSVs in a given 3DIC strongly depends on the interconnection architecture. In the case of CMPs, the most suitable interconnection choice is a 3D NoC [25]. In 3D NoC-based designs, stacked dies of a given layer communicate with other layers using vertical links implemented with TSVs. Recently, different 3D NoC architectures have been proposed to perform the inter-layer communication. The first approach, and the more immediate one, is to provide one-hop connectivity between layers. That is, flits from layer i-1 access layer i+1 through the router

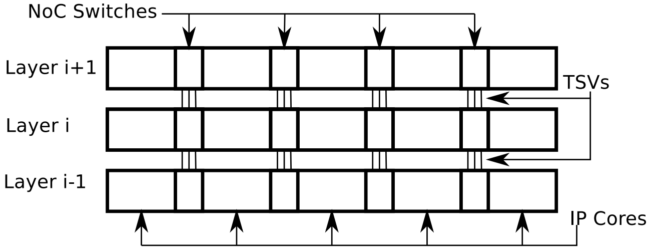


Figure 2.7: Representation of the 3D stacked chip

at layer i. To do so, the radix of the regular 2D router (equal to 5 in a 2D mesh) must be increased to 7 in order to enable the vertical direction. In this approach the higher radix switch can compromise the performance of the design. A second possible approach is the use of a bus to provide connectivity in the vertical direction. This approach takes advantage of the reduced TSV link delay to provide connectivity in one hop to any layer of the stacked dies. However, the use of a shared bus inherently limits the concurrency of communications and may cause traffic contention under high traffic loads. Finally, a more sophisticated approach is the one proposed in [47]. This proposal provides one-hop direct connectivity to go from a given input port of layer *i* to any output port of a router in any other layer. To do so, instead of building a complex 3D crossbar, inter-layer connectivity is provided by means of pass transistors between layers. As a drawback, the increased path diversity of this proposal dramatically increases the complexity of the arbitration because the number of possible requesters that contend for a given output port is considerably increased and, therefore, the possible benefits of an increased connectivity may be reduced. Additionally, this approach does not scale with an increasing amount of stacked dies.

The main benefits of moving to 3D architectures are the increased bandwidth and the reduced average interconnection wire length. However, 3DIC designs also introduce several challenges that have to be faced. On one hand, stacked designs present higher temperatures due to the increase of power density in comparison with traditional 2D architectures [25]. Therefore, 3D chip layouts should be carefully analyzed to face the rising thermal challenges. On the other hand, as the number of cores per layer increases, the number of

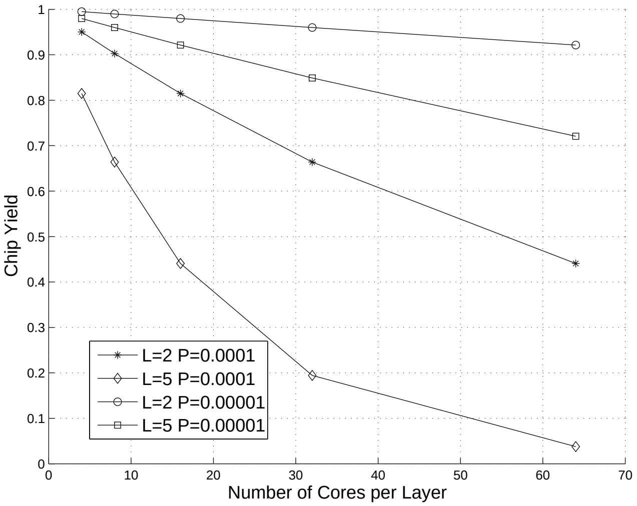


Figure 2.8: Chip yield estimations for different manufacturing scenarios.

required TSVs to enable 3D chip connectivity also increases. The expected huge amount of TSVs present in future 3D chips may lead to two main problems. First, as the number of required vertical links increases, the probability of having faulty TSVs also increases causing a yield reduction of 3D chips. Figure 2.8 shows, for two different TSV rate failure probabilities {P=0.0001, P=0.00001} and for chips containing 2 and 5 layers {L=2, L=5}, the expected yield of 3D chips as a consequence of failures in the vertical interconnects. The failure probabilities have been obtained from [41].

The results in the figure consider 64-bit wide links and a regular 3D-mesh interconnection scheme. As shown in this figure, as the number of cores per layer increases, chip yield is dramatically reduced. For high TSV failure rates (P=0.0001) the achieved yield is very low when the number of cores increases up to 64, being only 3% when the number of stacked dies is 5, and 44% for 2 stacked dies. Specially important is the fact that despite of having a low TSV failure rate (P=0.00001) the yield is also noticeably low when the number of cores increases up to 64, reaching values of 72% and 92% for 5 and 2 stacked dies, respectively. These yield numbers show how important is for chip manufacturers to leverage fault-tolerance mechanisms, as they would not easily make economical profit of their investment otherwise. These results are in concordance with the experimental data collected in [44] where yield of 3D DRAM chips with 4 stacked dies was 15% when no fault tolerance was provided.

The second problem that may present future 3D chips, as a consequence of the expected huge amount of vertical links, is area. Effectively, as the number of required TSVs increases, the footprint area required to build the vertical links cannot be neglected any more [79]. For example, assuming TSV pad dimensions of $8\mu m x 8\mu m$ and a pitch of $16\mu m$ the TSV footprint area for connecting 64 switches to a neighbour layer will be $1.84mm^2$.

As mentioned before, in this dissertation we present a new TSV-based vertical link design that is able to face these two important problems.

2.3 Summary

In this chapter the necessary background about current processor design is surveyed. First, the key insights of the NoC design paradigm has been introduced. In this part of the chapter also the basic details about the most adopted router and network architectures have been presented. Second, the most important challenges compromising the yield and performance of current and future NoC-based designs has been identified. On one hand, traditionally adopted approaches to face process variations have been analyzed. On the other hand, the problems related with the adoption of TSV-based 3D architectures have been stated.

Chapter 3

Modeling PVT Variations in ULSI designs

As integration technologies continue to scale down, parameter variation is becoming a major concern that cannot be ignored at any system level, as unpredictable variations affecting silicon devices and metalizations in the bottom layer are directly impacting performance at the upper levels, even influencing application execution time. In between, the system architecture and operating system are also affected by the uncertainty coming from the lowest layer of the system. Moreover, energy efficiency at any of the system levels may also be affected by parameter variations appearing down in the silicon level. In this way, if ignored, parameter variations may cause the benefits of shifting to smaller technology nodes to be canceled due to an inefficient exploitation of the underlying silicon devices by the upper levels [8]. This inefficient use of silicon resources will likely become more noticeable as process variation is increased when larger integration scales are in use.

Characterizing the impact of the different PVT variation sources in CMPs and MPSoCs is therefore becoming mandatory. This characterization will make it possible to explore architectural and technological solutions that minimize the impact of PVT variations in MPSoC and CMP systems. Additionally, a more accurate modeling of the different uncertainty sources will increase the accuracy of the mechanisms proposed for improving the predictability of manufactured chips. Thus, the availability of such a modeling tool will enhance the time-to-market of chip designs by reducing the differences among the features delivered by the final chip after being manufactured and the behavior and performance expected at initial design time.

In this chapter we present a framework able to accurately introduce and analyze the impact of PVT variations in CMP and MPSoC designs. Briefly, this framework starts from a given chip design implementation, uses a detailed model of the main sources of process, voltage, and temperature variations, and finally computes a given number of circuit instances in order to statistically cover the possible range of PVT variations. In summary, this framework allows to virtually manufacture the chip and get lots of samples to play with. In this way, analyzing the behavior of the virtually manufactured chip in the presence of PVT variations will help designers to better tune mechanisms in the chip before going to the foundry and, therefore, it will allow to increase the predictability of the final product. Moreover, the framework has been devised in such a way that it perfectly fits into the regular chip design flow. In this way, all the features already provided by available commercial CAD tools can be complemented with the additional metrics provided by the proposed framework.

3.1 Related Work

The impact of PVT variations has been thoroughly analyzed during the last years. Concretely, several recent works analyze the impact of process variations in the performance of integrated circuits. One of the first studies that analyzed how parameter variations impact the maximum design frequency was done by Bowman et. al [8]. Later, variability models that characterize variations in microarchitecture have been deployed [92] [42] [97] allowing designers to analyze the consequences of parameter variation in processor architectures. However, these studies focus on the impact of within-die variations in the processor architecture, not considering the implications of variations in the interconnect infrastructure.

Regarding the interconnect, in [66] the effects of process variations in the network on chip are analyzed. Nevertheless, this study only considers variations in routers and neglects the impact of manufacturing deviations in NoC links. Additionally, this study does not consider delay variations between routers and random variability is inaccurately analyzed because it is simplistically modeled as a percentage of the nominal delay, thus not considering other studies that show that random variations strongly depend on the critical path depth and the size of devices [42]. Unfortunately, repeated NoC links also suffer from variability as shown in [63]. In this work, the authors identify the main sources of process variation in NoC links and provide an analytical expression of timing variability from the variation of parameters involved in the interconnect delay. However, this study lacks the influence of spatial features of variability in NoC links. All these inaccuracies are fixed in the model presented in this Thesis, that accurately models process variations in NoC links by considering most of the variability sources.

A more sophisticated modeling of parameter variations is the one done in [36]. In this work, the authors provide a detailed model of PVT variations. In particular, this model considers the impact of device within-die variations, variations in wires, and also the impact of thermal variations in both devices and wires. However, this work still presents some deficiencies. First, the effect of device sizing in the contribution of random threshold voltage variations is not considered. Furthermore, processor, memories, and the network are modeled by using a critical path model which is a simplification of the real design. Instead, the model presented in our work is directly applied to the synthesized designs thus being able to catch the implications of variations at much finer granularity and, therefore, different design implementations can be tested with our framework.

In the context of MPSoCs, PVT variations also cause energy and delay characteristics of different chip components to vary considerably. This makes the behavior of embedded systems more difficult to predict, as mentioned before. In this way, several works have been made considering the impact of PVT variations in the energy and delay of embedded systems. In particular, in [106] a PVT-aware application mapping under hard real-time constraints is proposed. Also, the effect of PVT in the optimal formation of voltage-island designs has been thoroughly investigated. For example, [70] proposes an optimal voltage and frequency island formation considering energy and delay constraints. In the same way the authors of [60] elaborate an algorithm to efficiently adapt the voltage-island formation in the presence of PVT variations.

Finally, PVT variations, and specially process variations, may also cause manufactured chips to fail. These faulty chips reduce yield and therefore make the manufacturing process to become more expensive, reducing the economical benefits for chip makers. In order to avoid this yield reduction, as mentioned before, the straight-forward approach for facing the increasing delay uncertainty is to add higher worst-case guard bands to critical paths [6]. However, this solution makes the overall performance of manufactured chips to decrease. Moreover, as process variations are expected to become more important for smaller technologies, new approaches to face variations are required so that performance is not affected so dramatically. As stated in Chapter 2, the use of statistical delay calculation tools can reduce design margins and improve design speed [68]. This probabilistic framework avoids the pessimistic timing estimation introduced by the classical static timing analysis [72]. On the other hand, variability can be either compensated or tolerated. For example, [75,97] propose to design for the typical case and perform post-silicon compensation at some cost (performance, power). In [22,64] delay failures are tolerated at the cost of performance. These approaches are based in the fact that run-time timing violations are infrequent.

As can be derived from the discussion in this section, there has been a prominent effort to model the effect of PVT variations in chip designs, although all the models reviewed fail at some point. Therefore, it is required an accurate and complete model that allows system designers to asses the performance of their designs prior to manufacturing them, thus helping system developers to tune their designs. In this chapter we move forward and present a framework that accurately models process, voltage, and temperature variations for current deep-submicron designs. In summary, regarding process variations, the proposed framework enhances state-of-the-art tools. On one hand, in the proposed framework the whole interconnect, that is routers and repeated links, is taken into account in order to accurately model the presence of PV in NoC-based designs. Additionally, the modeling of random threshold voltage variations is improved by considering the effect of device size. On the other hand, unlike other proposals that use a model of the critical path, our framework is able to catch the effect of outliers, that is paths that were not critical but become critical as a consequence of PV. This is achieved because the proposed methodology is applied to all design cells individually and the whole design is considered.

3.2 Modeling Within-die Process Variations

The framework presented in this chapter considers three main sources of within-die variations: systematic back-end variations caused by the chemical metal planarization process, front-end systematic variations introduced in the photolitographic process, and front-end random variations caused by the random dopant fluctuations. The first source of variations causes wire dimensions to be modified, whereas the second and the third sources are related to devices. Front-end systematic variations cause deviations in the effective length (L_{eff}) of transistors while random variations affect their threshold voltage.

3.2.1 Modeling Wire Dimension Variation

The main manufacturing deviations affecting wire geometry (systematic back-end variations) are metal thickness (m_t), inter-layer dielectric thickness (T_{ILD}), and metalization width variations. Metal thickness and T_{ILD} variations are produced by imperfections in the metalization and dielectric surface as a consequence of the chemical metal planarization process that causes surface imperfections because of dishing and erosion. Actually, this process may be a very important source of variability according to [63]. On the contrary, metal width variations are caused by lithography defects. However, the effect of wire width variations in circuit delay is negligible. On one hand, the contribution to the total circuit delay of M1 metalizations is minimum and, thus, variations of width cause imperceptible delay variations. On the other hand, as one moves to higher metal layers, lithographic interactions does not cause larger proportional wire width variations as feature sizes are generally much larger [73].

Regardless of the origin of geometry variations, they cause that the equivalent resistance and capacitance of metalizations deviate from its predicted value at design time. Metal thickness variations mainly cause variations in wire resistance whereas T_{ILD} variation causes deviations in interconnect capacitance. However, as according to [63] the effect of variability in the capacitance of global interconnects is negligible, the main source of variation affecting global wire resistance is metal thickness variation.

In order to introduce the impact of metal thickness variations on wire delay we use Equations 3.1 and 3.2. Equation 3.1 represents wire resistance as a function of metalization dimensions. In this equation ρ is metal resistivity, w is wire width, l is metalization length, and t represents metal thickness. As a consequence of metal thickness variations, wire delay will vary. On one hand, as shown in 3.1 metal resistance is inversely proportional to metal thickness. On the other hand, as shown in 3.2 delay is directly proportional to wire resistance. Note that, as capacitance variations are negligible, the effect of variation in wires will not affect the delay of devices[1]. Additionally, for those wires of a link routed in the same layer, the same variation to the metal thickness has been applied in order to satisfy the strong spatial correlation present in the variations introduced by the chemical metal planarization process.

$$R = \frac{\rho \cdot l}{w \cdot t} \tag{3.1}$$

$$d \propto RC \tag{3.2}$$

3.2.2 Modeling L_{eff} Systematic Variation

The systematic component of front-end process variation is strongly related with the photolitographic process. Lens aberrations may lead to an important systematic spatial non-uniformity of L_{eff} over the reticle field [73]. According to [28], 3σ L_{eff} variation[2] can be as high as 12% for 45nm processes. However, it is not enough knowing the maximum percentage of variation in L_{eff}. It is required to know how variations in L_{eff} are spatially distributed in the exposure field as well as how variations in L_{eff} influence variations in devices. This knowledge may influence the proposal of architectural approaches that mitigate L_{eff} variations.

[1] Note that cell delay strongly depends on the output load, which is directly related to wire capacitance.

[2] 3σ is the usual way to express parameter variation where σ stands for the standard deviation.

In order to model the spatial non-uniformity of L_{eff} as well as its correlation we have used Gaussian Random Fields (GRF) [35]. When using GRF, with stationary and isotropic fields [35] [92], the variance (σ_i^2) of the random field $L(x, y)$, representing transistor gate length (L_{eff}) in the (x,y) die position, depends only on the euclidean distance between two given locations. Then, the gate length distribution (L) only depends on a correlation function. The correlation model we have used is the spherical model proposed by [92], which is derived from the measurements of [30]. Equation 3.3 shows the correlation function for this model, where $r = \|l - l'\|$ is the distance between two given locations, l and l', and X_L is a characteristic correlation length depending on the photolitographic process. Basically, $\rho(r)$ states how much the gate length of a transistor located at l is similar to the gate length of a transistor located at l'. If the distance between locations l and l' is larger than X_L then the gate length values of both transistors are totally independent. On the contrary, if both transistors are located closer than X_L then their gate length values are correlated.

$$\rho(r) = \begin{cases} 1 - (3r/2X_L) + (r/X_L)^3/2 & \text{if } (r \leq X_L) \\ 0 & \text{if } (r > X_L) \end{cases} \tag{3.3}$$

X_L depends on the characteristics of the manufacturing process. Understanding the meaning of X_L requires further explanation. Chips are printed using a reticle field of a given size. In this way, spatial patterns of L_{eff} variation are spread over the entire exposure field. This behavior is shown in Figure 3.1. In this figure, the exposure field is enclosed by the thick square, while chip border is denoted by the thin square. Two different exposure field sizes are denoted in the figure. Additionally, it is assumed in this figure that L_{eff} values are correlated for an area that spans half of the reticle field (gray circles). Thus, when a small chip is printed per field (left), spatial features of variability will cover half of the chip. In this case the values of X_L in Equation 3.3 would be 0.5. However, it is possible to print four small chips in a single shot by increasing the exposure field (middle). In this case, X_L must be set to 1 as spatial features of variability are spanned across the entire chip. On the other hand, if we print a chip four times bigger than the initial one (right) by using an enlarged exposure field, then X_L would remain half of the chip size

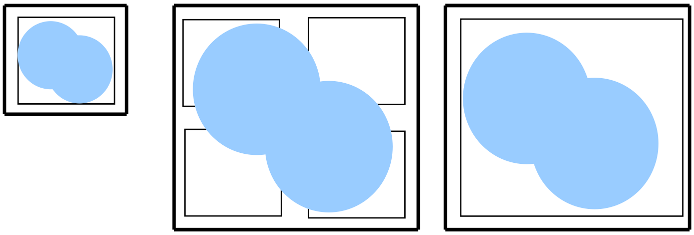


Figure 3.1: Exposure field.

($X_L = 0.5$).

Once L_{eff} values are computed for the entire chip, in order to relate variations in L_{eff} to variations in device delay, we have to compute the systematic component of V_{th}. This will allow to catch the dependence of the threshold voltage with gate length. This dependence is satisfied with the model presented in [11] and shown in Equation 3.4. In that equation V_{th_0} is the threshold voltage for long channel transistors and α_{dibl} is the DIBL coefficient. Then, applying to that formula the parameters for a given technology we get the systematic component of V_{th}. Notice, however, that the computed value for V_{th} cannot be used yet to compute device delay variations, as that delay is also influenced by other components of V_{th}, like the random one explained in the following section. The contribution from all the components must be aggregated in order to get the new device delay.

$$V_{th_{eff}} = V_{th_0} - V_{dd} * exp(-\alpha_{dibl} L_{eff}) \tag{3.4}$$

Finally, in order to translate the explanation above to an algorithm that computes L_{eff} values for the entire chip, an additional concern needs to be addressed. Here the issue is that there are infinite points in the chip surface and, therefore, the chip surface has to be discretized so that an algorithm can deal with it. In our framework, the chip surface has been discretized by using a 1000x1000 square matrix and R [85] has been used to implement the Gaussian Random Fields with the spherical model mentioned before. The resolution selected is the maximum that allows the R framework. Moreover, note that in addition to the spherical model there exist other correlation models available in the literature. For instance, in [42] a different model of systematic device variation is presented. The model proposed uses a simple polynomial function

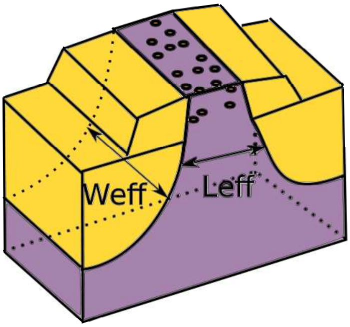


Figure 3.2: Dopant distributed in the transistor channel

fitted with experimental data to represent the spatial correlation caused by the photolitographic process. This is the way to proceed when experimental data of a mature process is available. However, in the absence of these data, the spherical model is still able to catch the spatial correlation caused by lens aberrations.

3.2.3 Modeling Random Threshold Voltage Variation

The main source of front-end random variation is threshold voltage variation due to Gaussian Random Dopant Fluctuations (RDF). Briefly, the concept behind RDF is the following (see Figure 3.2): as technology scales down, the number of dopant atoms that fit into the transistor channel area is getting smaller and smaller. For 45nm and later technologies, this number is in the range of a few tens atoms for 45nm down to around 10 atoms for 16nm, in average. Thus, with these numbers, a few atoms more or less considerably matter. This variation in the number of dopant atoms in the transistor channel is what is known as Random Dopant Fluctuations, which are expected to be the major source of unpredictability affecting future VLSI circuits. According to [45], RDF will increasingly affect deep submicron technologies scaling below 45nm.

We set $3\sigma_{V_{th}}$ variations to 40% according to [28] for a 45nm technology. However, according to Plegrom's law [82] the severity of random V_{th} variations is related with the actual device size. That is, it is not enough to know the percentage of variation but it is required to analyze how each particular device suffers from it depending on its size. This is an improvement over other variability models previously proposed [92] [36] [97]. Therefore, to compute $\sigma_{V_{th}}$ for a given device, according to [2], we use Equation 3.5, and then relate the σ_{Vth_0} value of the minimum size device with the $\sigma_{V_{th}}$ of a device of size h, as shown in Equation 3.6, which clearly shows that $\sigma_{V_{th}}$ can be minimized by increasing the width of devices, represented by h. However, this would increase the area required to implement the circuit as well as the power consumption, which is not a good option.

$$\sigma_{V_{th}} \propto \frac{1}{\sqrt{W_{eff}L_{eff}}} \tag{3.5}$$

$$\sigma_{V_{th}} = \frac{\sigma_{v_{th_0}}}{\sqrt{h}} \tag{3.6}$$

3.3 Modeling Environmental Variations

The sources of manufacturing variability previously analyzed lead to variations in the electrical properties of circuit components [73]. These variations change from one chip to another, but are not modified after manufacturing for a given chip. Therefore, these electrical properties are kept constant after the foundry stage. On the contrary, environmental variations are a consequence of the exact working conditions of the chip and therefore are not constant but vary with time. We distinguish two major components in the environmental variations: supply voltage variations and temperature variations. Temperature variations are due to the thermal conductivity of the silicon substrate and the package of the manufactured IC and depend mainly on the circuit operation. That is, as the circuit is working it will generate more or less heat as a consequence of the amount of computing carried out. The different amount of heat with time and across the die surface causes temperature variations that affect the electrical properties of transistors and wires. On the other hand, supply voltage variations also affect the way a given circuit behaves. In this case the unpredicted behavior is caused because the circuit may be designed out of the exact power supply context that will experience later while working. This may be caused, for example, for a circuit that is replicated, after its design, many times in the die. In this case, any asymmetry in the power supply will make the different replicas not to behave in the same way.

In the following two sections we enter into the details of these environmental variations. Notice that these two variations can be combined with manufacturing process ones in order to assess the final behavior of the chip.

3.3.1 Modeling Supply Voltage Variations

As the complexity of integrated circuits increases, the complexity of the power grid also increases. Actually, in current chip designs a significant amount of wiring resources is devoted to power delivery. In these complex power grid designs, supply voltage variations are present due to differences on the power demand of the different circuit modules connected to the grid [73]. These voltage variations cause differences in the operating conditions of circuit components, as they were designed for a given supply voltage and, therefore, their delay will be different from the initial one set during the design phase. Additionally, in the context of a voltage-island design[3], the problem of variations in the power supply voltage is not diminished at all. In this case, different circuit modules can be supplied at different voltages within a chip. Therefore, for each of the modules, the inaccuracy in the supply voltage is multiplied by the number of different voltages the circuit can be fed. As can be seen, introducing this unpredictability source into the model is mandatory for achieving a robust chip able to work in a stable way.

In order to model the behavior of circuits working at different values of the supply voltage we use the well known alpha power law [89]. According to this law, the dependence of the delay (D) with the supply voltage (V_{dd}) can be represented as:

$$D \propto \frac{V_{dd}}{(V_{dd} - V_{th})^{\alpha}} \tag{3.7}$$

where α is a technology dependent parameter, in our case 1.3[4], and V_{th} is the transistor threshold voltage. The relationship between voltage and delay

[3] The insights of the voltage-island design style are explained in more detail in Chapter 5.

[4] This parameter is equal to 1.3 for all the technologies considered in this study (from 45nm to 16nm).

represented by the alpha power law [89] is specially important for the understanding of the dynamic voltage and frequency scaling mechanisms [58]. Thanks to the behaviour represented by this equation the voltage of a circuit can be lowered when the frequency is reduced, thus saving a considerable amount of energy. This formula will be later used in Section 3.4 to aggregate the effects of most of the sources of variations we are analyzing. In that section it will be shown how to integrate this formula within the framework.

3.3.2 Modeling Thermal Variations

As in the case of voltage variations, different temperatures cause different behaviors in the circuit. Additionally, different parts of the die may experience different amount of heat. Temperature variations across the die are mainly a consequence of differences in power consumption of integrated circuit components that are placed at different chip locations. This is caused in part by the variable workload distribution. This uneven power distribution across the chip surface causes temperature variations that affect in different ways the delay of devices and wires located at different places in the die.

In order to model thermal variations, we must know that, as stated in [73] and [36], temperature has a modest impact in device delay. According to the BSIM model [105] the threshold voltage temperature dependence can be modeled as:

$$V_{th} = K_{V_{th}} * \Delta T + V_{th_{nom}} \tag{3.8}$$

where $K_{V_{th(T)}}$, given in mV/K, is a process dependent parameter that catches the dependence of V_{th} with temperature. Regarding wires, metal resistivity also varies with temperature. Metal resistivity can be expressed as:

$$\rho = \rho_0 + \alpha(T - T_0) \tag{3.9}$$

where ρ_0 is the resistivity at the nominal temperature T_0 and α is the temperature coefficient with units of Ω/K. For example, we know from [36] that the resistivity of copper is reduced by about 30% when temperature is reduced from 100° to 0°. These two formulae will be used in Section 3.4 to combine the thermal variations with the rest of variations.

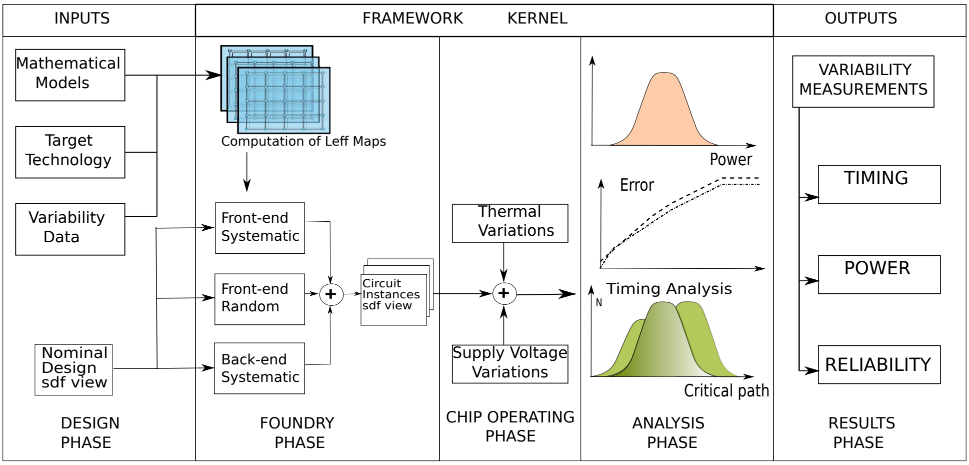


Figure 3.3: Diagram of the Framework for injecting PVT variations.

3.4 A Framework for Injecting PVT Variations into ULSI circuits

The main purpose of this chapter is to present a framework that allows to predict and analyze the behavior of the manufactured chip designs while in the circuit design stage. In this way, more robust designs could be achieved. The framework presented in this chapter is able to introduce process, supply voltage, and temperature variations into a target design. To do so, the tool makes uses of several scripts that parse the required design information and connect with commercial CAD tools to provide a deep analysis of a given target design. After synthesis CAD tools store design information in the *.sdf*, the *.def*, and *.spef* files. Concretely, as we will see later in more detail, the *.sdf* file contains cell or wire delay information, the *.def* file the placement information, and the *.spef* file the capacitance and resistance of all circuit nets and wires.

Figure 3.3 shows a diagram of our proposed framework. On the left side of the flowchart we can find the data available at design phase. These data are basically the synthesized circuit to be analyzed and the characteristics of the technology used to manufacture that circuit. Also, the variability data of the manufacturing process is fed into the framework as well as some mathematical models used later to compute the systematic front-end process variation. All these data are the inputs to our framework, which are summarized in Table 3.1. On one hand, the *.sdf*, the *.def*, and *.spef* files of the target design are required as the primary input of the framework. Additionally, further details of the architecture of the target design can be set. Concretely, we need to know the granularity of the voltage islands if any, the clocking scheme, and the power budget. Regarding the information about the expected parameter variations, in the best possible scenario, experimental data of a given fabrication process is known and these data is the one that would be used to feed the framework. If no experimental data about parameter variations is available, the parameter variation predictions of the ITRS can be used as starting point. Table 3.2 shows the values of the expected variation for metal thickness, threshold voltage, and transistors L_{eff}.

Once the inputs are defined, the main stage of our framework takes them in order to simulate what would happen in the foundry. In this phase, the framework kernel, the different sources of process variations are aggregated and many different instances of the die containing the chip are generated. In summary, what is being done is virtually manufacturing the design so that the produced chips suffer from the variability that would be present in real chips. In order to later collect meaningful statistics and accurately catch the behavior of the chip, several hundred chip instances are produced.

In the next step, once we have virtual chips that can be put into work, we get environmental variation data by applying thermal and supply voltage changes to the manufactured chips. Notice that the foundry phase and the chip operating phase compose the main kernel of our framework.

After generating PVT variation data for as many chip instances as possible, it is required to analyze them. For doing so, our framework also provides a good number of scripts that parse the large outcome of the previous stages and condenses all that variability information into some consolidated results. This is the actual outcome of our framework. Table 3.3 summarizes the output of our framework. In Section 3.4.2 we will further elaborate on them.

In the following subsections we describe how the framework kernel is built from the discussions about variability sources in the previous sections, which stats can our framework provide, and also discuss how to use the framework.

MultiVoltage Design	**Clocking Scheme**	**Parameter Variations**
Granularity max/min V_{dd}	Synchronous vs GALS	$\sigma\{L_{eff}\ m_t\ V_{th}\}$

Table 3.1: Framework Inputs

Technode(nm)	45	32	22	16
$3\sigma V_{th}$	40%	58%	81%	112%
$3\sigma L_{eff}$	12%	12%	12%	12%
$3\sigma m_t$	10%	10%	10%	10%

Table 3.2: V_{th}, L_{eff}, and m_t variation according to ITRS

3.4.1 Framework Kernel

The kernel of the framework consists of a set of perl scripts that, making use of several different design views, statistically generate different chip instances. The generated instances contain different cell and wire delay variations as a consequence of manufacturing process deviations. Later, environmental variations are introduced if desired.

In order to introduce PVT variations into the chip, we leverage the fact that commercial ASIC design flows make use of several views of synthesized designs. Different design views represent different circuit characteristics that are required to finally be able to manufacture a chip. For instance, after the place & route step the *.sdf* and *.def* files are generated representing the accurate delay of each design cell or wire and its placement, respectively. Additionally, capacitance and resistance of all circuit nets and wires can be obtained by analyzing the *.spef* file. Making use of these files, PVT variations can be injected into both devices and wires. To do so, the basic idea is, once parameter variations are modeled, they can be translated into delay variations. In our framework this is performed for devices and wires separately.

In the case of devices we use Equation 3.10 to relate gate delay with the

Timing	**Reliability**	**Power**
Statistical delay	Link error vs frequency	variations (Static, Dynamic)

Table 3.3: Framework Outputs

V_{th} and L_{eff} transistor parameters. Remember, from the discussions in Sections 3.2 and 3.3, that front-end process variations and thermal variations affect L_{eff} and V_{th}. In this way, transistor L_{eff} is computed by taking into account systematic front-end variations whereas the nominal value of transistor V_{th} is modified by adding both front-end random variations (caused by random dopant fluctuations) and systematic front-end variations (caused by lens aberrations). This value of V_{th} is later modified in order to introduce thermal variations. On the contrary, power supply variations are introduced by directly modifying the value of V_{dd} in Equation 3.10.

$$D \propto \frac{L_{eff}^{1.5} * V_{dd}}{(V_{dd} - V_{th})^{\alpha}} \tag{3.10}$$

Notice that each cell in the design is individually considered in order to compute its new delay. The procedure is as follows. From the *.def* file we obtain the exact location of the cell in the die. Once we know the cell position, we leverage the L_{eff} maps obtained in Section 3.2.2 in order to know the new L_{eff} of the cell. As L_{eff} variations are strongly spatially correlated, we can assume that all transistors in the cell will present the same L_{eff} value, which is introduced in Equations 3.4 and 3.10. In order to complete the process, the impact of random V_{th} variations must be computed. For doing so, $\sigma_{V_{th}}$ is computed according to the cell size, as shown in Equation 3.6. Once computed the random contribution for V_{th}, its systematic counterpart is added and, finally, the new delay value is carried out, which is then written into the original *.sdf* file, thus replacing the initial value. In case user of the framework wants to analyze environmental variations then further computations are required. In order to apply thermal variations, the cell delay present in the *.sdf* file should be modified by taking into account the relationship between V_{th} and temperature cell delay as shown in Equation 3.8. In the case of power supply variations Equation 3.7 would be leveraged.

Regarding wire delay, it has been computed after varying metal thickness. As short interconnects have a negligible contribution to the total path delay, we only consider variations affecting global and semi-global interconnects. Note that this assumption helps to considerably speed-up the process of injecting variations into the design interconnects without loss of accuracy. The way back-end variations are introduced in our framework is similar to the process for the front-end ones. In this case, the *.spef* file is used to know the resistance of all the design wires[5]. The new values of resistance are computed after applying thickness variations using Equation 3.1. After this the interconnect delay values in the *.sdf* file are modified according to Equation 3.2. Notice that if thermal variations are to be considered, then wire resistivity should be further modified according to Equation 3.9. Once this procedure is completed for all the cells and wires in the *.sdf*, commercial CAD tools, like Synopsys PrimeTime, can later use this file to perform the timing analysis.

We should remark that this algorithm for generating process variations is repeated multiple times in order to generate multiple instances of the *sdf* design view. These instances are generated by leveraging the multiple L_{eff} maps previously computed. Each of the L_{eff} maps represents a different consequence of the systematic process variation, where random variation is aggregated.

An effect that should be additionally considered is how process variation affects signal slope and how therefore delay is indirectly affected. However, according to [56], the impact of process variation in signal slew is small. Additionally, as in the framework target designs, L_{eff} variation of consecutive cells within a path is negligible, it is possible to consider that signal slews propagate under nominal process conditions in the designs under test.

3.4.2 Framework Metrics

Several different measurements can be performed with our framework. Basically, they can be grouped into timing, reliability, and power metrics.

Timing

The main goal of the proposed framework is to asses how PVT variations impact delay. Therefore, the primary analysis that can be carried out with our framework is the timing analysis. Once the different design instances representing the effect of parameter and voltage variations are generated, timing

[5] Note that in order to speed-up the process of injecting back-end variations only global wires can be selected.

results are obtained by means of a commercial static timing analysis tool. For example, the timing reports of the different *.sdf* files can be collected to generate a statistical representation of the circuit slack or the maximum achievable frequency. Additionally, the percentage of paths that were not initially critical but can become critical in a given number of instances can also be measured.

Reliability

Process variation makes some critical paths of the design to slow down. As a consequence, the resulting maximum operating frequency of the design must be lowered in order to ensure the proper operation of the design. In order to face variation induced timing errors, higher design margins are required causing a considerable performance reduction. However, several recent designs propose to work at higher frequencies being able to tolerate infrequent timing violations [92]. In the context of regular CMPs and MPSoCs, variation induced timing errors can occur in both the processor pipeline and in the network. With the framework presented in this chapter, circuit delay can be represented as a normal distribution of mean μ and standard deviation σ. As the σ of delay is increased as a consequence of process variations, the slack required to ensure the absence of errors must be increased and, therefore, the maximum operating frequency of the design decreases. Following the reasoning given in [63] the probability of failure in a given path is the probability of the delay being more than $D_{nom} + slack$:

$$P(D > D_{nom} + slack) = 1 - P(D < (D_{nom} + slack)) \tag{3.11}$$

$$F_{link} = 1/(D + slack) \tag{3.12}$$

Power

In CMP and MPSoC systems, power estimations are required to ensure that a given power budget is fulfilled. Regular accurate power estimations are time consuming and depend on the circuit switching activity. Additionally, as a consequence of parameter and supply voltage variations, the power consumption of chips can present a huge variation and cause a given power budget to be exceeded. In this framework we provide power variation estimations

Circuit	Gates	Delay (ns)	Description
c3540	1669	2.638	8-bit ALU
c74283	29	0.401	4-bit adder
c7552	3513	2.391	32-bit adder/comparator
c432	160	2.241	27-channel interrupt controller
c74L85	25	0.346	4-bit magnitude comparator
c6288	2416	7.470	16x16 multiplier

Table 3.4: ISCAS benchmark circuits

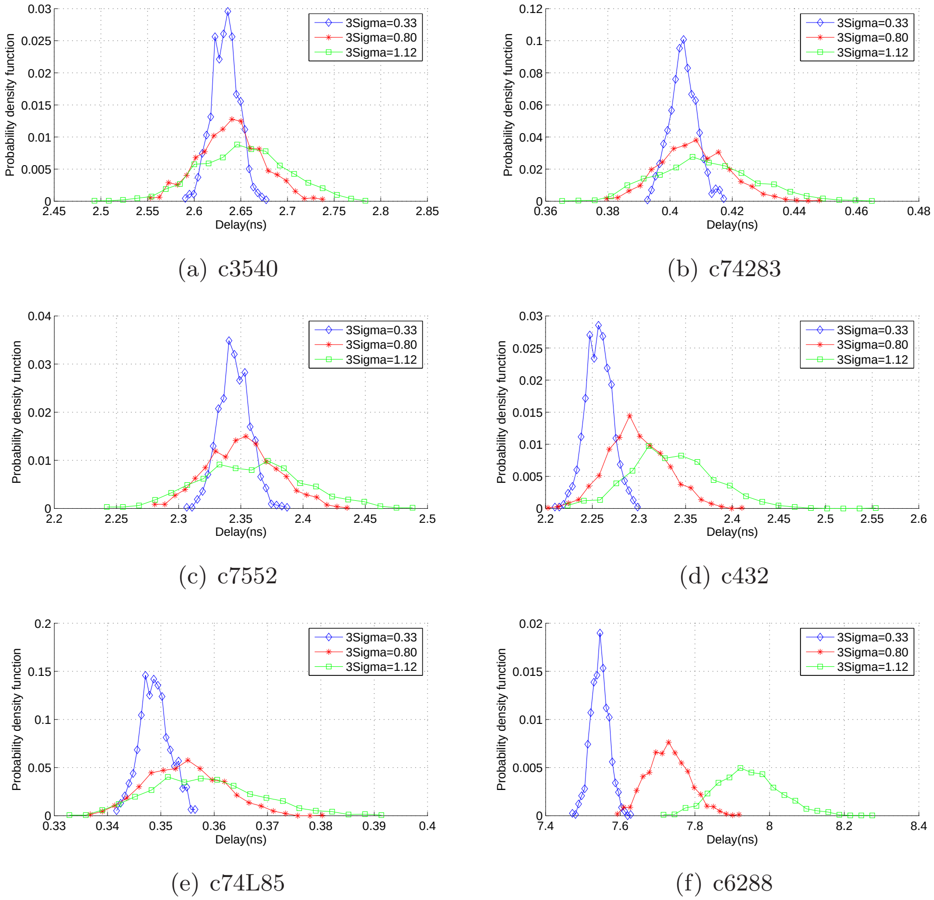
that can be used to tune and adjust the system design to better fit the power requirements in the presence of process variations. The framework provides metrics of both dynamic and static power in the presence of variations and for a given supply voltage.

3.5 Putting the Framework into Work

In this section we show some small examples about different uses of the framework. For this purpose we will leverage representative circuits from the 74X-series [101] as well as circuits included in the ISCAS-85 benchmarks [9]. Table 3.4 shows the description of the circuits considered, the nominal delay of the circuit, and the number of gates of each circuit. We will first show how the framework can be used to estimate process variations for several VLSI technology nodes. Later we will exercise the power supply variations described before. In Section 4.3 we will address the application of the framework to a complete much larger design.

3.5.1 Predicting Process Variability

As told before, random V_{th} variations are expected to become the major source of process variation as technologies scale down. Therefore, it is of special interest to understand how future manufacturing processes will affect circuit behavior. To do so, random V_{th} variations of different severity have been injected into the benchmarks. In particular, 3 different values of $3\sigma V_{th}/\mu$ (0.33 0.88 1.12) have been considered, trying to emulate the severity of variations



(a) c3540

(b) c74283

(c) c7552

(d) c432

(e) c74L85

(f) c6288

Figure 3.4: ISCAS benchmark circuits

expected for 65nm, 32nm, and 16nm manufacturing process, respectively, according to the ITRS analysis.

Figure 3.4 shows the probability density function of circuit delay for the three different manufacturing processes considered and the six benchmarks. Figure 3.4 clearly shows how variations affect different benchmarks in a different way. Table 3.5 summarizes the results of benchmark delay estimations in the presence of parameter variations displayed in Figure 3.4. It can be seen that, on one hand, for simpler circuits, as 74283 and 74L85, variations make the resulting circuit delay very unpredictable. This is because these circuits have short critical paths, and shorter paths are more sensitive to random variations than larger ones, where random variations tend to be canceled by the aggregated effects of the devices in the critical path. In fact, delay uncertainty caused by random variations is proportional to $1/\sqrt{n}$, where n is the critical path depth [38]. On the other hand, more complex circuits as 6288 with higher number of critical paths present lower variation (σ/μ) but, as the number of critical paths is higher, the mean delay is increased. For example, for a 16nm process the mean delay of the 6288 under process variations is increased a 6.25% over the nominal delay.

Results in this section confirm that accurately modeling the critical path of the circuits under design is mandatory. On the contrary, the effect of the number of critical paths and the critical path depth of a given circuit cannot be accurately taken into account . In this regard, the framework presented in this chapter accurately catches the implications of variations as all the paths of the circuit are considered.

3.5.2 Predicting Voltage Supply Variation Effects

In this study we put into work some of the environmental variation sources. More concretely, we are going to introduce power supply variations and also check the accuracy of this part of the framework. For this purpose we compare the synthesis results of the benchmarks in Table 3.4 at different supply voltage conditions with the estimations of our framework for the same voltage conditions. In this way, the benchmark circuits have been synthesized with a 65nm low-power technology library at 1V, 1.1V, and 1.2V. With the *.sdf* file representing circuit delay of the 1V implementation, we feed the framework in order to estimate the *.sdf* files at 1.1V and 1.2V supplies. Once the process is finished, we compare the results of the framework with the actual synthesis results. Table 3.6 shows the framework estimation error at 1.1V and 1.2V. As shown in Table 3.6 the average error for estimating the values at 1.1V is 3.62% whereas for 1.2V the error in Table 3.6 is 5.28%.

Results confirm that the framework has an acceptable accuracy as results of this framework are intended for architecture exploration purposes an thus some small error is allowed. Additionally, results in Table 3.6 also show that, in particular, the model used in our framework for introducing power supply variations is accurate enough and, in general, that the approach embedded into

	$\sigma_{Vth} = 0.33$		
Circuit	μ	shift(%)	σ/μ(%)
c3540	2.6332	0.07	1.0256
c74283	0.4042	0.3	1.8881
c7552	2.3447	0.04	1.0711
c432	2.2570	0.71	1.2215
c74L85	0.3489	0.83	1.3142
c6288	7.5452	0.92	0.6445

	$\sigma_{Vth} = 0.80$		
Circuit	μ	shift(%)	σ/μ(%)
c3540	2.6391	0.3	2.1767
c74283	0.4077	1.17	5.2234
c7552	2.3511	0.32	2.0483
c432	2.2981	2.55	2.8163
c74L85	0.3546	2.49	3.8209
c6288	7.7291	3.39	1.3131

	$\sigma_{Vth} = 1.2$		
Circuit	μ	shift(%)	σ/μ(%)
c3540	2.6525	0.81	3.4080
c74283	0.4112	2.03	7.5390
c7552	2.3585	0.63	3.2446
c432	2.3354	4.21	4.4027
c74L85	0.3583	3.55	5.0859
c6288	7.9431	6.25	2.1923

Table 3.5: Effect of random variations on combinational circuits delay

Circuit	Error(%) for 1.1V	Error(%) for 1.2V
c3540	1.67	4.73
c74283	2.99	4.37
c7552	8.11	8.99
c432	3.11	5.18
c74L85	3.03	3.76
c6288	2.83	4.64
Average	3.62	5.28

Table 3.6: Error introduced by the framework for power supply variations

the framework kernel is able to deliver realistic results. On the other hand, in order to determine the accuracy of the framework for process variations we should compare the results provided by it with results from real manufactured chips. Unfortunately, these data are not available. On one hand, foundries do not disclose these data as this may compromise their business. On the other hand, sometimes, these data does not simply exist. This is the case for fabrication processes that are not currently being used.

3.6 Conclusions

In this chapter we presented a framework able to accurately model and analyze the impact of PVT variations in CMP and MPSoC designs. As shown throughout this chapter, by leveraging the proposed framework the impact of PVT variations can be accurately characterized. The proposed tool is intended to help designers to better tune their designs much before reaching the manufacturing stage. In this way, the predictability of circuit behaviour can be increased.

In order to test the developed framework, process variations have been injected into different ISCAS benchmark circuits. Results of this preliminary study show how the different variability sources impact circuit performance. Concretely, we show that the increase of threshold voltage variations, as we expect for future fabrication processes, causes a non negligible impact in circuit performance. Additionally, in order to test the accuracy of the framework, we compare the results of injecting supply voltage variations to the different benchmarks, with the synthesis results for different supply voltages provided by the technology library. Results of this study confirm that the framework presents an acceptable accuracy.

In summary, we have presented in this chapter a framework, that can be integrated into the regular design flow in order to assess the impact of PVT variations prior to manufacture the chip, thus providing chip designers with a powerful tool that will increase reliability and predictability for submicron technology designs.

Chapter 4

Characterizing NoC-based Designs under Process Variations

In this chapter we apply the framework developed in the previous chapter to measure the impact of process variation on different NoC-based designs. The goal of this chapter is to identify those situations where process variations may cause a considerable degradation of design performance. For that purpose, three different case studies of the impact of variations in NoC-based designs are presented. The first case study analyzed in this chapter, *Characterizing sub-45nm Link Delay Variability*, is aimed to show how the impact of parameter variations in NoC links cannot be neglected. In fact, as we will show later, the impact of random threshold variations will cause a very important delay uncertainty in NoC links when technologies scale below 32nm. In the second case study, *Characterizing Variability in NoCs*, a more global analysis is performed. In this case study the implications of parameter variations in a whole NoC are analyzed. For that purpose, the variability model presented in the previous chapter is applied to routers and links, and the influence of parameter variations in both NoC components is measured. Finally, a more complex application scenario of the variability model is presented. In particular, in the *Assessment of Variability Robustness of MPSOC Architectures* case study, the impact of variation in a whole MPSoC design is studied. First, the impact of process variation in a LEON processor [31] is analyzed. Later, the robustness of two different NoC clocking schemes approaches is compared in the presence of parameter variations.

4.1 Characterizing sub-45nm Link Delay Variability

In this first case study we analyze how process variability affects NoC links. In order to accurately introduce parameter variations in NoC links we use the model presented in Chapter 3. However, as sub-45nm technology libraries are not yet available, it is not possible to perform the synthesis of links for that technologies. Therefore, the regular flow of the model previously presented is slightly modified to be able to work with the Predictive Technology Models (PTM) [104] of technologies below 45nm. Concretely, the more important difference with respect to the regular use of the model is the fact that the links have been directly simulated with SPICE using the PTM models. A simplified scheme of the modified process for injecting variations in NoC links is displayed in Figure 4.1. As can be seen, this methodology has several inputs, some of them related with the NoC design phase and some of them related with the chip manufacturing stage. The first set of inputs is composed of the NoC layout synthesized at design time, the characteristics of the target technology used to implement the chip, and time and power constraints for the links in the network. The second set of inputs includes parameter variation data and several mathematical models. The output of the methodology is the delay for each of the links in the network. More precisely, this methodology provides accurate delay data for each of the wires in every network link.

The methodology is divided into two steps. In the first one, the best link configuration (number of repeaters and their size) is computed according to the design time set of inputs. In the second step, link delay variability data is obtained for all the links in the network according to the link characteristics from the previous step.

For introducing process variations in NoC links we assumed, as it is commonly accepted, that as random and systematic variations are uncorrelated ($\sigma^2 = \sigma^2_{rand} + \sigma^2_{sys}$) [91], the delay in NoC links can be represented by Equa-

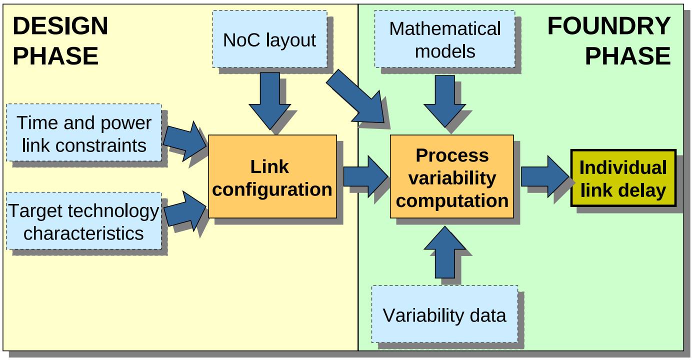


Figure 4.1: Flowchart of the proposed methodology.

tion 4.1. In this equation T_{nom} represents the nominal component of the delay whereas ΔT_{rnd} and ΔT_{sys} represent the timing deviations caused by the random and systematic components of variation, respectively.

$$T_{link} = T_{nom} + \Delta T_{rnd} + \Delta T_{sys} \tag{4.1}$$

4.1.1 NoC link design background

When designing a link several concerns must be taken under consideration for the sake of efficiency. More specifically, power and area must be optimized for a target link delay.

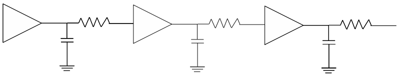


Figure 4.2: Diagram of a repeated link composed of three sections.

Repeater insertion is an efficient method to reduce interconnect delay and signal transition times. Actually, this mechanism allows to minimize link delay by the optimal insertion and sizing of repeaters. However, minimizing delay involves high-sized repeaters and consequently higher power consumption. Consequently, links are designed to reach a given frequency[1] with the minimum possible power dissipation. This is achieved by inserting the proper

[1] Note that in this study NoC operating frequency is set by link delay.

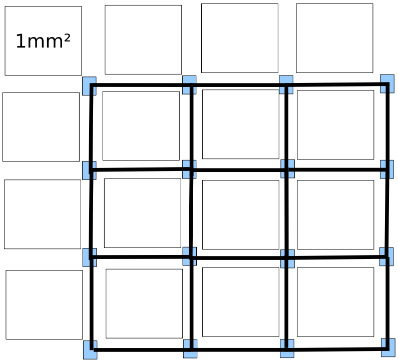


Figure 4.3: Scheme of the network layout and link distribution for a 4x4 multicore chip using a 2D mesh NoC. Small gray squares denote switches.

Technode(nm)	45	32	22	16
Link length(mm)	0.83	0.59	0.41	0.3
Core area (mm^2)	0.48	0.24	0.11	0.06
Vdd(V)	1	0.9	0.8	0.7
Width(um)	140	99.6	68.4	49.78
Spacing(nm)	140	99.6	68.4	49.78
Thickness(nm)	280	199.1	136.9	99.6
Heigth(nm)	290	206.2	141.78	103.11
Dielectric	2.5	2.3	2.1	2.0

Table 4.1: Data for each of the technologies considered

number of minimum sized repeaters [14]. Figure 4.2 shows an schematic of a link wire.

Repeater insertion requires knowing both link length and target link delay. Regarding link length, in this analysis we have used the 65nm real implementation NoC layout [83] as a test-bench where to apply our methodology. According to this implementation, all cores are identical, and their size is 1mm^2. Additionally, the gap between cores is 0.2mm. Therefore, links connecting NoC switches are 1.2mm long. Figure 4.3 shows the location of cores and links in an example die. As we will consider larger integration scales than 65nm, all layout dimensions will be scaled by the factor "$1/s$", where s is the ratio between the smaller technology node and the node size in [83]. Table 4.1

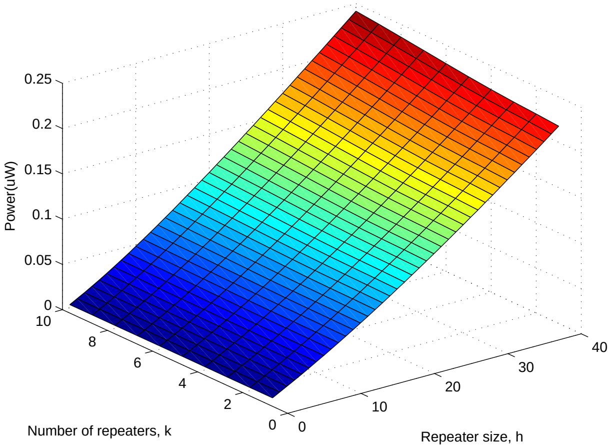
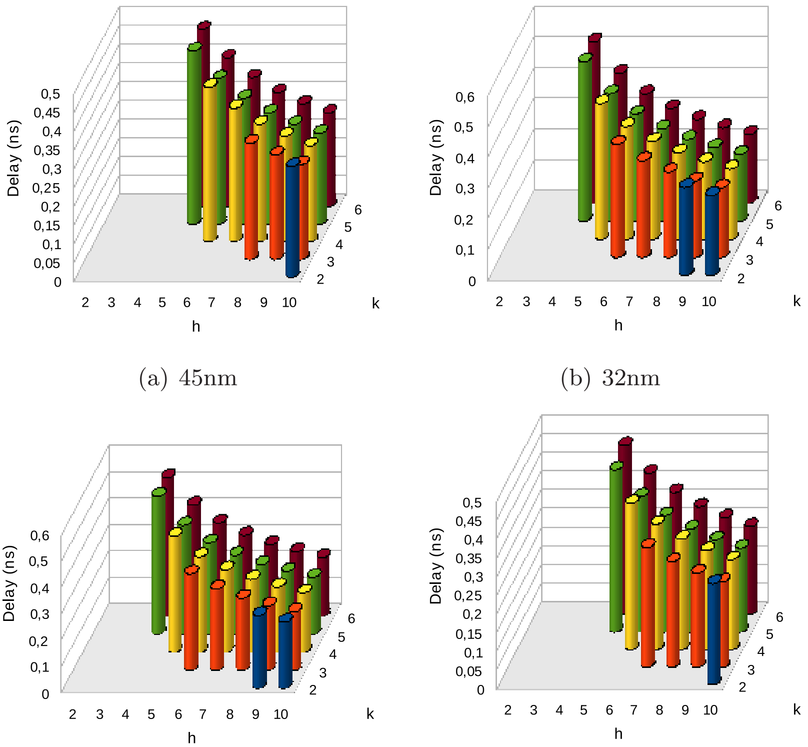


Figure 4.4: Total power dissipation in a repeated interconnect as a function of h and k. Operating frequency is 2 GHz.

shows the size of cores and links for each of the technologies considered in this analysis.

Regarding link delay, it is possible to trade it for power. This is shown in Figure 4.4, which represents how the power consumption of a link wire varies with the number of sections (k) and repeater size (h). Data plotted in this figure has been obtained by simulating a 0.83mm long wire for a 45nm technology. Simulations have been carried out using SPICE and the PTM model for 45nm [104]. The simulated wire is assumed to be placed in a semi-global interconnect metallization layer, whose characteristics are also shown in Table 4.1. Data shown in this table for 45nm technology has been obtained from [52]. Data for the rest of technologies has been scaled from [52]. Wire capacitance that correspond to these features has been calculated according to the expression given in [107]. Wire segments have been modeled by using a 5-pi wire model. As shown in Figure 4.4, on one hand power consumption rapidly increases with repeater size. On the other hand, when the number of sections is increased, power consumption shows a slight increment when the size of repeaters is above 20. Note that high-sized repeaters are considerably faster.



(a) 45nm

(b) 32nm

(c) 22nm

(d) 16nm

Figure 4.5: Link design space for the technologies considered. K is the number of repeaters and h is repeater size.

In this section we will apply the modified methodology to NoC links synthesized using 45nm, 32nm, 22nm, and 16nm technologies. Therefore, it is required to know the link configuration (optimum number of repeaters and their size) for each of these technologies. To do so, we have simulated these links by using SPICE in the same way as described above. Figure 4.5 shows the link design space for the four technology nodes considered. In the plots, bars represent the delay value for a given link configuration, determined by the number of sections (k) and the size of its repeaters (h). As expected, high-sized repeaters provide the minimum delay. However, setting a maximum delay constraint value equal to 0.5ns[2], the optimal configuration is given

[2] Link delay remains constant to fairly compare among technologies the effects of process variation with technology scaling.

Technode(nm)	45	32	22	16
h	5	5	5	5
k	5	4	4	5
Delay(ns)	0.46	0.44	0.44	0.43

Table 4.2: Link configurations for the scaled links

by the minimum repeater size satisfying that constraint. When several possibilities with minimum repeater size exist, we chose the option with fewer repeaters. Table 4.2 shows the link configurations we will use in the rest of the study for the technologies considered. Note that the number of repeaters does not increase with technology scaling because link length is also scaled.

4.1.2 Analyzing the Impact of Back-End Variability

As mentioned in Chapter 3, the Chemical Metal Planarization process is one of the main sources of timing variability. The Chemical Metal Planarization process causes surface imperfections in the wires as a consequence of dishing and erosion. Wide wires, as the ones located in the semi-global layers, are strongly affected by dishing [71] causing considerable changes in the interconnect resistance. In concordance with [62] and [27], we consider a 3σ[3] resistance variation of 15% for all technologies considered in this study. Moreover, as the degree of dishing and erosion strongly depends on the pattern density of the metallization [49] and NoC links are built in a regular layout, it is possible to assume that all wires in the link and all links in the NoC will be affected by the Chemical Metal Planarization process in a similar way.

The resistance variation produced by the planarization process has been introduced in the 5-pi link model previously mentioned. Simulation results show that resistance variation effects on delay are negligible. Delay variation in all technologies remain below 0.1% of the nominal delay. These results seem to be contradictory with the results in [62] where the authors measure a delay variation, as a consequence of dishing, of around 9% of the nominal delay. However, it is necessary to clarify that links in [62] are designed to

[3] 3σ is the usual way to express parameter variation where σ stands for the standard deviation.

Technology node(nm)	45	32	22	16
$3\sigma V_{th}$	40%	58%	81%	112%
$3\sigma L_{eff}$	12%	12%	12%	12%

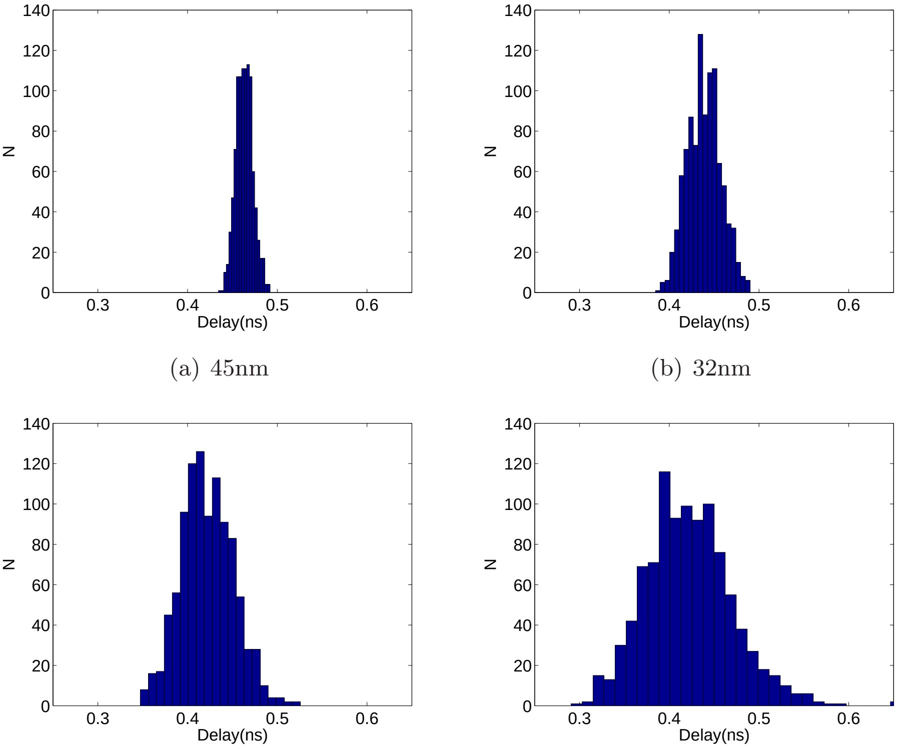
Table 4.3: V_{th} and L_{eff} variation according to ITRS

have minimum delay. More concretely, they consider, for a 65nm technology, a 5mm link with 3 repeaters of size 100. However, interconnects designed to have minimum delay are not the best option for NoC links due to their high power consumption. On the contrary, when links are designed for minimum power operating at a given frequency, delay is dominated by repeater delay and consequently variations in wire resistance have a negligible contribution to the resulting link delay.

4.1.3 Analyzing the Impact of Front-end Random Variation

The main source of random variation in NoC links is threshold voltage variation due to Gaussian random dopant fluctuations (RDF). RDF will increasingly affect deep submicron technologies scaling from 45nm down to 16nm. Table 4.3 shows the values of $\sigma_{V_{th}}$ and $\sigma_{L_{eff}}$ for the technologies considered as provided by the ITRS report [27]. The $\sigma_{V_{th}}$ values shown in Table 4.3 represent the total threshold voltage variation. In [45] it is shown that at least 50% of that variation is due to RDF for a 45nm technology node. Note that these values represent the $\sigma_{V_{th}}$ of minimum size devices. Interconnect repeaters usually have higher gain in order to be able to drive high load values as a consequence of link length. To compute $\sigma_{V_{th}}$ for repeaters, as explained in Chapter 3, we can use Equation 3.5, and then relate the σ_{Vth_0} value of the minimum size device with the $\sigma_{V_{th}}$ of a repeater of size h, as shown in Equation 3.6, which clearly shows that $\sigma_{V_{th}}$ can be minimized by increasing the width of repeaters, represented by h. However, as previously stated, because of power consumption reasons it is necessary to keep the gain of repeaters as low as possible.

Figure 4.6 shows how delay variation associated with the RDF drastically increases with technology scaling. In this figure we consider that 50% of the overall threshold voltage variation is caused by RDF according to [45]. Links



(a) 45nm

(b) 32nm

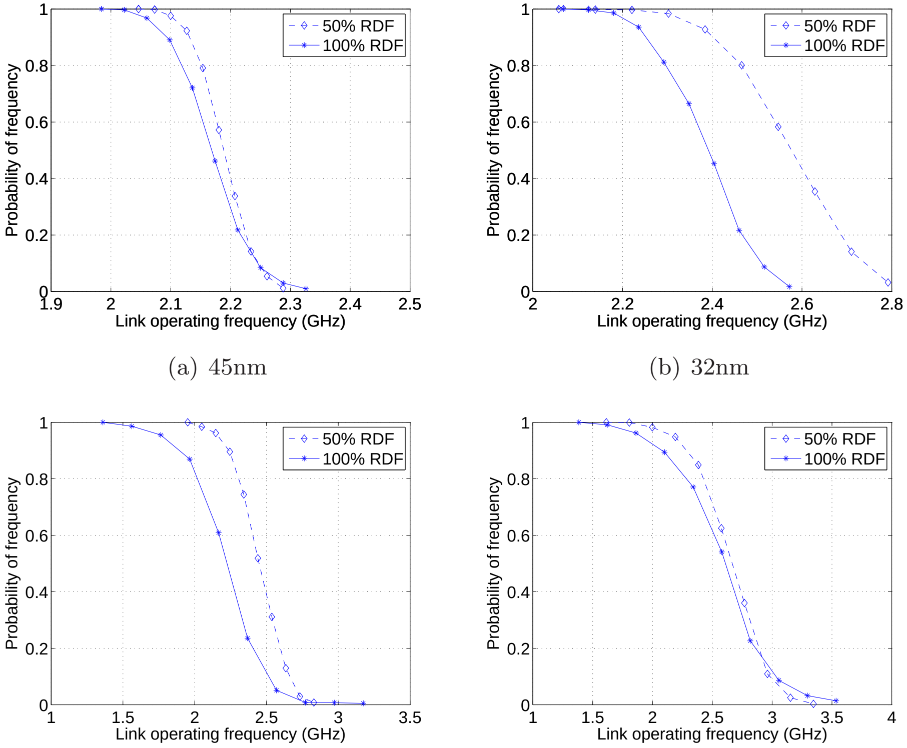
(c) 22nm

(d) 16nm

Figure 4.6: Link delay variation due to front-end random variability.

in this figure are designed as explained in Section 4.1.1. As shown in Figure 4.6, delay uncertainty increases up to a factor of 6 when fabrication processes move from 45nm to 16nm. Note that it is expected that the RDF fraction of the total threshold variation increases for future fabrication process. For this reason, Table 4.4 shows how delay variation increases when the RDF fraction varies between 50% and 100%.

Finally, Figure 4.7 shows the fact that, as a consequence of random variations, not all wires in a given link will be able to operate at the same frequency. For example, for a 45nm technology all link wires will work at 2GHz whereas only 50% of them will operate at 2.15GHz. In this case there is no significant differences among wires for that link. On the contrary, when using a 16nm technology, all wires in the link will work at only 1.5GHz, while 50% of the wires will be able to operate at a frequency higher than 2.5GHz.



(a) 45nm

(b) 32nm

(c) 22nm

(d) 16nm

Figure 4.7: Maximum operation frequency of a link.

4.1.4 Computing delays for each link

Once the L_{eff} value for every repeater is known, delay data for each link can be collected. In order to compute the exact delay value for each link, computed L_{eff} values for each of the repeaters of a given link are introduced in the 5-pi wire model and simulated using SPICE with the PTM models previously mentioned. Note that as we are using SPICE, variations in L_{eff} automatically produce variations in the systematic component of threshold voltage. Moreover, it is worth to point out that the delay computation is independently performed for each of the links in the network.

Table 4.5 shows, for the four technologies considered in this study, the influence on link delay of front-end systematic variation. As can be seen, this component of process variation may contribute to link delay variability

Technology node(nm)	45	32	22	16
ΔT_{rnd} 50% RDF (%)	2	4.23	6.61	11.26
ΔT_{rnd} 100% RDF (%)	2.76	5.32	10.64	16.44

Table 4.4: Range of delay variation as a consequence of RDF

Technology node(nm)	45	32	22	16
ΔT_{sys} (%)	4.31	4.34	6.27	9.31

Table 4.5: Link delay differences due to systematic variation

as much as 9% for the links considered. Data in Table 4.5 are the result of averaging the link delay standard deviation for 100 instances of 8x8 2D mesh NoCs.

4.1.5 Combining all sources of variation

Finally, we have to combine all variability contributions to obtain the link frequency characterization. To do so, all the variability components previously analyzed are combined in the link model presented in Section 4.1.1 and simultaneously simulated in order to collect variability data for each of the links in the network. The right side of Figure 4.8 shows an example of the output provided by our methodology for the 4x4 2D mesh that originated the L_{eff} variation map on the left side of the same figure. Numbers close to each link in the figure are the maximum frequency achievable by that particular link. Note that frequencies in Figure 4.8 are the result of applying all sources of variability previously detailed to links that were initially intended to operate at 2GHz.

On the other hand, the second result provided by our methodology is a distribution of the operating frequency of link wires as shown in Figure 4.7. Note that this frequency distribution is individually provided for every link in the network. Additionally, as it was explained in Section 4.1, link frequency will be influenced by both components of variability, systematic and random. Therefore, the distribution of the operating frequency of link wires will be centered at a different frequency for each link (systematic component) and will have a different shape (random component). Actually, random variability

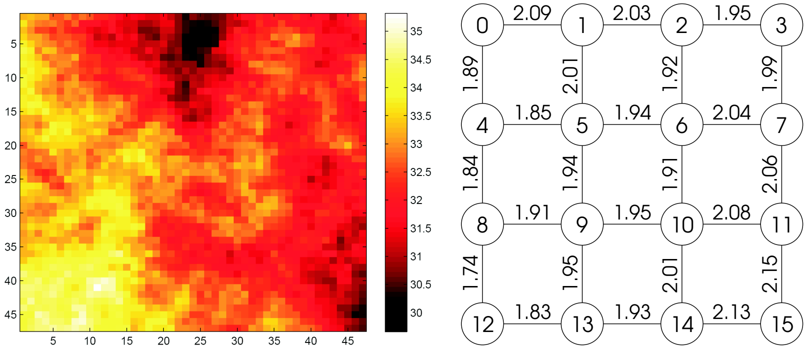


Figure 4.8: Map of L_{eff} variation for a 4x4 chip XL = 1 (left) and a 4x4 NoC showing link frequency in GHz (right).

data will be useful for tuning mechanisms based on using or not single link wires [34].

4.2 Characterizing Variability in NoCs

In this section, a more global study is presented. Concretely, the effects of process variations on the whole network, that is router and links, is analyzed. To do so, we used the framework presented in Chapter 3 to generate 200 instances of an 8x8 mesh NoC test bench. With the generated chip instances, we studied how process variation affects the performance of each of the components of the network. From the 200 NoC instances analyzed, 100 of them were produced using a value equal to 1 for the ρ correlation parameter while the other 100 were produced with $\rho = 0.5$.

4.2.1 Designing a Network-on-Chip

In order to analyze the influence of process variation on NoC performance, we have to design and synthesize the NoC used as a test bench. The first concern to address when designing such a network is which will be the CMP configuration the network will be embedded into. The chip floorplan used for placing network modules is similar to the one presented in Section 4.1.1. As mentioned before, according to this implementation, core area is 1mm^2. Additionally, the gap between cores is 0.2mm. Finally, links connecting NoC

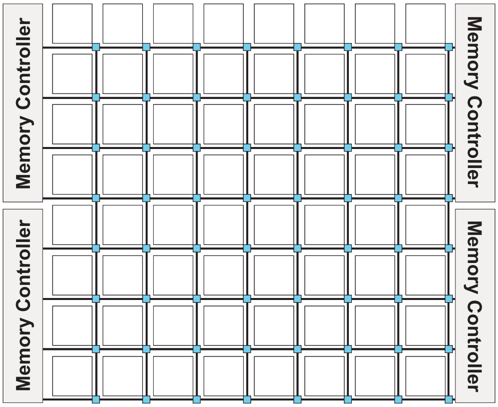


Figure 4.9: Layout and link distribution for the 8x8 CMP test bench. Memory controllers are also shown.

switches are 1.2mm long. Figure 4.9 shows an schematic of the 8x8 CMP that will be used in this analysis as case study. In this study, we are going to synthesize the 8x8 CMP using 45nm instead of 65nm. Therefore, core size and link length must, again, be appropriately reduced according to the feature size of the 45nm technology. Table 4.1 showed the dimensions for cores and links once scaled down to that technology, as well as other physical and electrical parameters of the 45nm technology, as stated in [52]. In the following sections, the router and links designed to be used in such a CMP are described.

Router Architecture

In this section we describe the router design used in this case study. Figure 4.10 shows the main components of the router. The router is a pipelined input buffered wormhole router with five stages: input buffer (IB), routing (RT), switch allocator (SW), crossbar (XB), and link traversal (LT). We have designed a simple router with no virtual channels and five input and output ports. Thus, four ports are intended to provide connectivity with the neighboring routers in the 2D mesh and the fifth port connects to the local computing core. Link width is set to 4 bytes. Flit size is also set to 4 bytes. Input buffers can store four flits. A Stop&Go flow control protocol has been deployed in order to control the advance of flits between adjacent routers. Additionally, the routing stage has been implemented to support the XY routing algorithm.

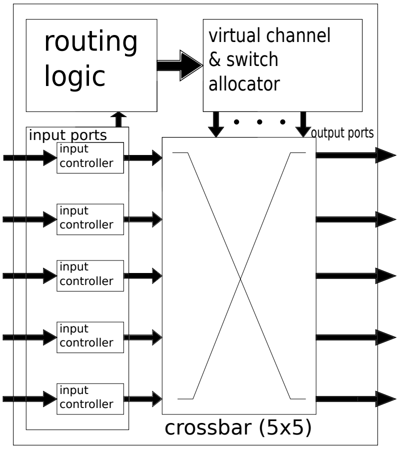


Figure 4.10: Router schematic.

Moreover, routing is performed for each input port individually. Similarly, a global SW module performs the switch allocation for each output port. Finally, the SW module has been designed using a round-robin arbiter according to [94].

area / Freq	Prelayout	Postlayout
area (um^2)	17651	19779
freq (GHz)	1.75	1.33

Table 4.6: router area and frequency

module	area (um^2)	critical path (ns)	gates	critical path depth
IB	3113.45	0.55	177	6
RT	124.26	0.32	72	8
SW	337.88	0.52	35	12
XB	1975.6	0.75	519	8

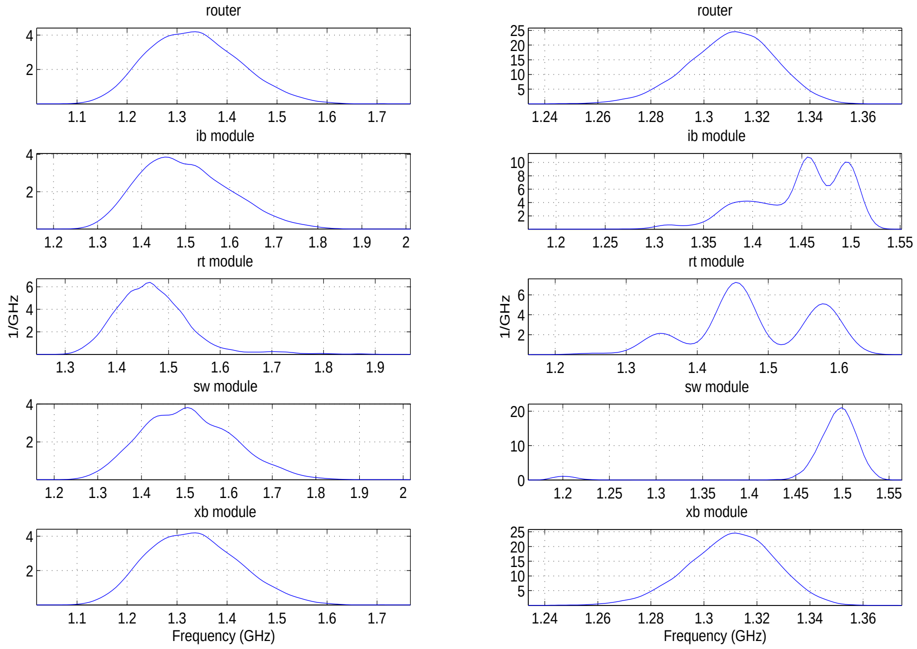
Table 4.7: Area, delay, and number of gates for the router modules

The router has been implemented using the 45nm technology open source Nangate [52] with Synopsys DC. We have used M1-M3 metalization layers to perform the Place&Route with Cadence Encounter. Table 4.6 summarizes the frequency and area results of the router implementation. As shown in this table the nominal router frequency is 1.33GHz (postlayout).

Table 4.7 summarizes the delay, area, and number of gates for each of the modules of the router. Note that the area numbers in that table are for a single instance of each module, but some of them are replicated in the designed router. This has to be taken into account if area numbers in Tables 4.6 and 4.7 are compared. Additionally, the area data in Table 4.7 have been obtained by independently synthesizing each module to work at its maximum frequency. When, on the contrary, the whole router is synthesized at once, those numbers slightly change. On the other hand, it is noteworthy to mention that, although the number of gates in the critical path in the XB stage is not the highest one, the gates present in it are slower than the gates present in other stages due to the large load the gates in the XB stage support, thus causing the XB stage to become the bottleneck in our router.

Link Design

In the same way that in the previous case study, for a proper link design several concerns must be taken into consideration. More specifically, power and area must be optimized for a target link delay. For example, as links are usually long interconnects, they will present a considerable capacitance and resistance. To deal with them, repeaters are used, as showed in Figure 4.2. In our case, in order to minimize power consumption [14], we have chosen the proper number of minimum sized repeaters that satisfies the delay constraint imposed by the frequency of the router presented in the previous section. As shown before, the post-layout router delay is equal to 0.75ns. Therefore, our link has been designed to present a delay similar to the delay of the router in order to save power. To satisfy those premises we chose a supply voltage equal to 0.9V and a link consisting of 5 repeaters of size 2. Additionally, links are placed in metalization layers M4 and M5, whose physical dimensions were shown in Table 4.1. With this configuration we obtained a nominal link delay of 0.67ns by using SPICE and the PTM model for 45nm [104] for the link simulations.



(a) Only systematic L_{eff} variations ($\rho = 1$).

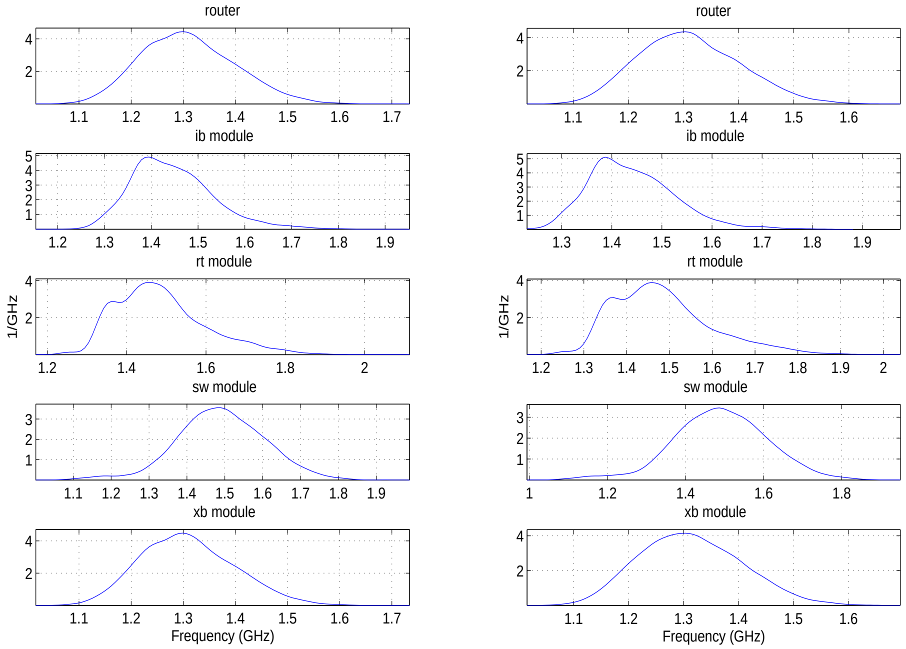
(b) Only random V_{th} variations.

Figure 4.11: Frequency variation in the router pipeline.

4.2.2 Variability in the Router

We analyzed the influence in the router of random and systematic variations. It is important to remind the reader that the nominal operating frequency without variability is 1.333 GHz. Figures 4.11(a) and 4.11(b) show the probability distribution function (*pdf*) of the operating frequency of the router and each of its stages in two scenarios: when only systematic variation with correlation 1 is considered (4.11(a)) and when only random variation is taken into account (4.11(b)). Figures 4.11(a) and 4.11(b) show that systematic variation has a larger influence in the operating frequency of the router than random variability.

Additionally, as it is shown in Figure 4.11(a), frequency variations in a router when only systematic variability exists present similar *pdf* for all the modules. This is due to the fact that systematic variability is highly correlated.



(a) Systematic correlation of variability set to 1.

(b) Systematic correlation of variability set to 0.5.

Figure 4.12: Frequency variation in the router pipeline as a consequence of both systematic and random variations.

Thus, as the router is small, all the components of the router are affected by variability in a similar way. This means that if the frequency of one of the stages is reduced because of systematic variability, then the frequency of the others stages will probably be also reduced. Therefore, it can be seen as a biased variability that causes that the critical path does not change among the routers, thus making the XB stage to keep being the bottleneck in almost all instances of the routers.

On the other hand, frequency variation in a router when only random variability is considered does not present similar *pdf* for all the stages. This is due to the nature of random variability which differently affects two adjacent components. Thus, random variability may even be reduced or canceled as the number of gates in a chain of logic increases [38]. Therefore, the critical

stage	sys($\rho = 1$)	sys($\rho = 0.5$)	rnd	full($\rho = 1$)	full($\rho = 0.5$)
IB	0.9935	0.8941	0.1493	0.9659	0.8318
RT	0.9968	0.7617	0.2867	0.9685	0.7651
SW	0.9945	0.9857	0.0938	0.8896	0.8833
XB	1.0000	0.9997	0.9998	0.9754	0.9876

Table 4.8: Correlation between stage delay and router delay

Parameters	sys($\rho = 1$)	sys($\rho = 0.5$)	rnd	full($\rho = 1$)	full($\rho = 0.5$)
Nom. Freq.	1.3333	1.3333	1.3333	1.3333	1.3333
Max. Freq.	1.7153	1.6639	1.3661	1.6835	1.6420
Mean Freq.	1.3339	1.3344	1.3101	1.3099	1.3112
Min. Freq.	1.0707	1.0811	1.2422	1.0672	1.0672
σ/μ	0.0672	0.0684	0.0124	0.0681	0.0692

Table 4.9: Nominal, maximum, mean, and minimum frequencies and frequency variation of routers

path of a stage may change depending on the variability map of each router. Thus, the peaks of the *pdf* in Figure 4.11(b) represent a different critical path inside the corresponding stage. This non-biased variability makes that the bottleneck of the router will not always be the XB stage but other stages may constraint the maximum operating frequency of the router. This can be seen in Table 4.8, which shows the correlation between the operating frequency of the stages and the operating frequency of their router. Note that the correlation when only systematic (sys) variability is considered is higher than when only random (rnd) variability is taken into account. Additionally the correlation of the XB stage is the highest one in all cases, as explained before.

Figures 4.12(a) and 4.12(b) show the probability distribution function of the operating frequency of the router and each of its stages when systematic and random sources of variation are simultaneously considered. Figure 4.12 and Table 4.8 show that there exist small differences in frequency between high and low correlation.

Table 4.9 shows the main parameters of each configuration described above. It shows the nominal, maximum, mean, and minimum frequencies and the

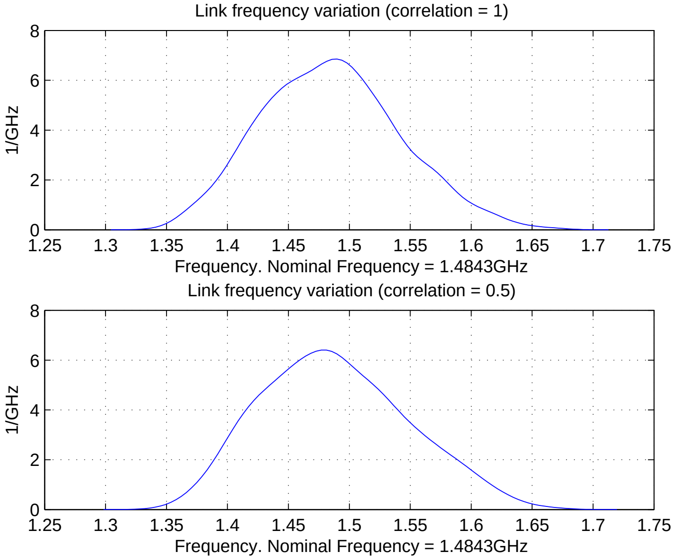


Figure 4.13: Link frequency variation as a consequence of systematic L_{eff} variations.

frequency variation of each *pdf*. Frequency variation is computed as (σ/μ) where σ is the standard deviation and μ is the mean of the *pdf*. Data in Table 4.9 confirm that the exact value of the ρ parameter generates very small differences. Moreover, random variability moves the mean frequency more than the systematic one. As mentioned before, this is due to the fact that random variability makes the critical path to change from one instance of the router to another more often than systematic variability.

4.2.3 Variability in Links

Figures 4.13, 4.14, and 4.15 show link operating frequency variation as a consequence of systematic L_{eff} variation, random V_{th} variations, and both random and systematic variations, respectively. Figure 4.13 shows that when considering only systematic variations, link operating frequency varies between 1.35GHz and 1.7GHz, for both values of ρ, despite that the nominal frequency was 1.48GHz. The exact value of correlation does not introduce significant differences, as shown by the σ parameter ($\sigma = 7.2$ and $\sigma = 7.0$ for $\rho = 1$ and

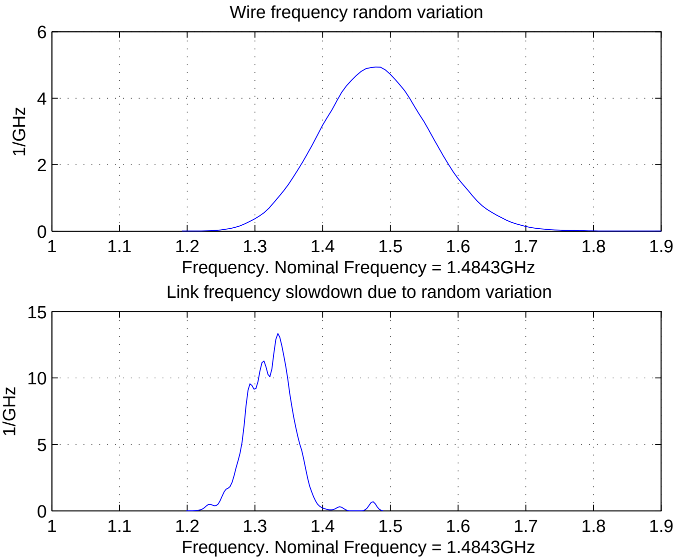


Figure 4.14: Link frequency variation as a consequence of random V_{th} variations.

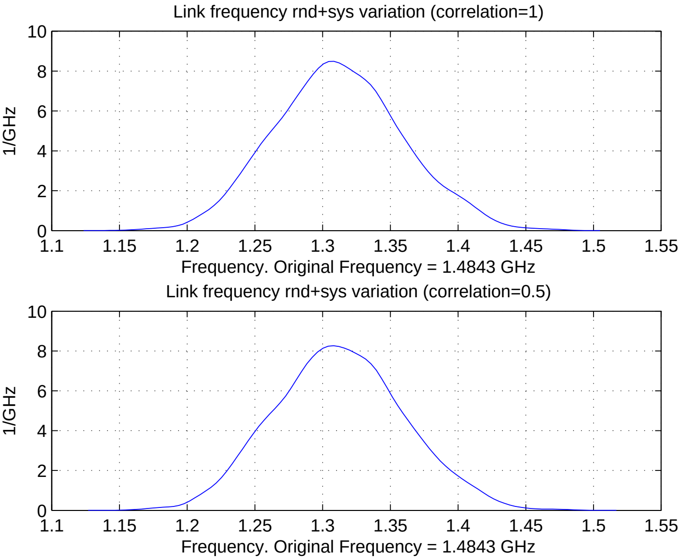
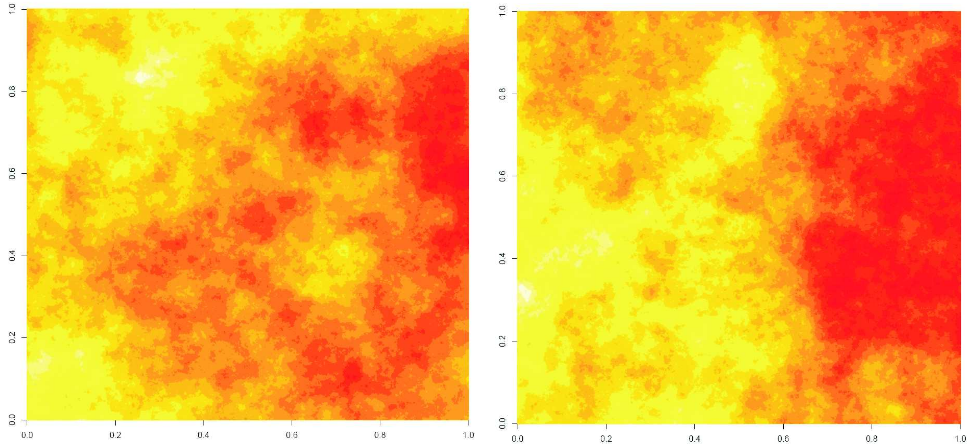


Figure 4.15: Link frequency variation as a consequence of both systematic and random variations.

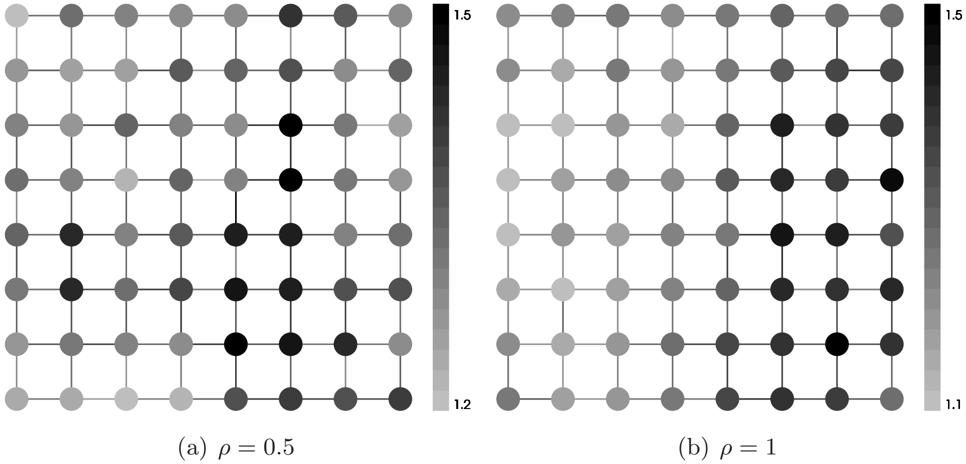


(a) $\rho = 0.5$

(b) $\rho = 1$

Figure 4.16: L_{eff} maps for chips with $\rho = 0.5$ and $\rho = 1$

$\rho = 0.5$, respectively). When only random variations are analyzed, variations in wires move practically in the same range than systematic variations. The top plot of Figure 4.14 shows the maximum achievable operating frequency of all wires of each link in the network. However, as all wires of a given link have to work at the same frequency, the slowest wire will cause a considerable operating frequency slowdown. This is shown in the bottom plot of Figure 4.14, where the mean frequency is reduced to 1.34GHz and the frequency variation is reduced to 5.58%. This effect is similar to the behavior analyzed by Bowman et. al in [7]. In that work, it was shown that when the number of critical paths increases, the mean delay increases and the standard deviation decreases, respectively. In the case of links, a higher number of wires per link will cause a higher frequency slowdown but also a reduction in the standard deviation of the link operating frequency. Finally, Figure 4.15 shows that when random and systematic variations are simultaneously considered, the average link operating frequency is reduced as a consequence of random variations. The mean values of link operating frequency are 1.31GHz and 1.32GHz ($\rho = 1$ and $\rho = 0.5$). However, frequency variation of links is almost the same than when considering only systematic variations, $\sigma = 7.2$ and $\sigma = 7.5$ for $\rho = 1$ and $\rho = 0.5$, respectively.



(a) $\rho = 0.5$ (b) $\rho = 1$

Figure 4.17: Operating frequency distribution in a NoC in the presence of process variation.

4.2.4 Variability in the NoC

Once variability for routers and links has been independently analyzed, we can aggregate those results in order to provide comprehensive data for the entire NoC. According to the results in previous sections, the traditional synchronous design technique is not feasible anymore because NoC clock frequency should be lowered to match the frequency of the slowest component in the network, noticeably reducing network performance.

This can be seen in Figures 4.16 and 4.17. Figure 4.16 shows two different L_{eff} maps, one for each value of correlation. It can be seen in these maps that L_{eff} values smoothly change over the chip surface. On the other hand, Figure 4.17 shows the resulting operating frequency for routers and links after being affected by variability. As can be seen, neighboring routers and/or links present similar operating frequencies. Additionally, Table 4.10 shows, for a 45nm technology node, how frequency variation (σ/μ) increases considerably for bigger regions as a consequence of the non-uniform frequency distribution of routers across the chip surface.

regions	sys($\rho = 1$)	sys($\rho = 0.5$)	rnd	full($\rho = 1$)	full($\rho = 0.5$)
2x2	0.0342	0.0382	0.0109	0.0365	0.0403
3x3	0.0441	0.0491	0.0118	0.0460	0.0506
4x4	0.0511	0.0561	0.0121	0.0526	0.0574
5x5	0.0563	0.0609	0.0123	0.0576	0.0619
6x6	0.0604	0.0637	0.0124	0.0615	0.0647
7x7	0.0637	0.0661	0.0124	0.0647	0.0669
8x8	0.0666	0.0680	0.0124	0.0675	0.0686

Table 4.10: nocs σ/μ from regions

4.3 Assessment of Variability Robustness of MPSOC Architectures

A larger example of the use of the framework introduced in Chapter 3 is presented in this section in order to show the potentials of that framework. More concretely, the impact of parameter variations in performance and robustness of a NoC-based MPSoC is analyzed. In this example, the impact of parameter variations in the computing cores and in the network architecture is measured, thus demonstrating the generality of our framework.

4.3.1 System under Test

The system under test used as an example is a 16-core NoC-based MPSoC. A schematic of the design floorplan is shown in Figure 4.18. As shown in this figure the MPSoC used is a regular multiprocessor design where computing cores and memories are interconnected by means of a 4x4 2D mesh NoC architecture. Note that the connection between cores and the network is performed by using a network interface (NIC) that allows the processor to inject and eject packets properly. Additionally, in order to be able to decouple the frequency of cores from the NoC frequency, the usual FIFO queues present in the NIC have been replaced by dual-clock FIFOs.

In this example, the LEON processor has been chosen as the computing core to use. The LEON3 processor [31] is an open source 32-bit CPU microprocessor based on the SPARC-V8 RISC architecture. The LEON has been

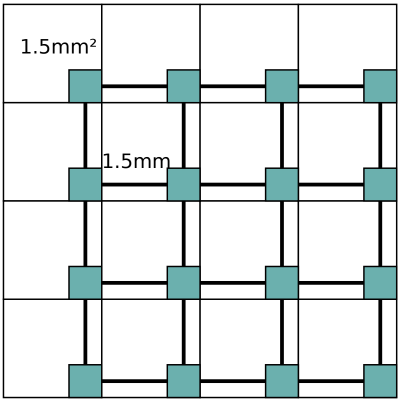
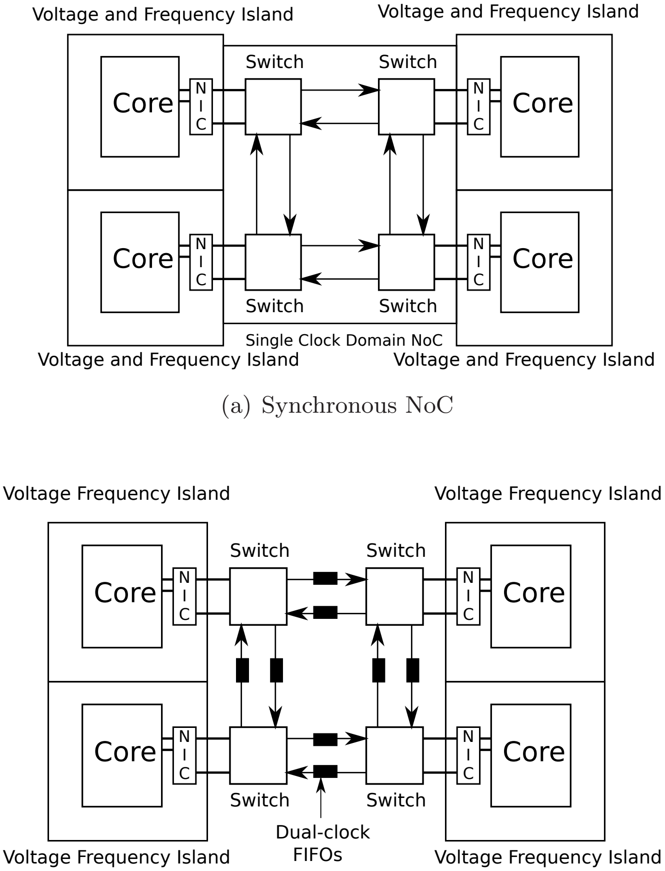


Figure 4.18: Schematic of MPSoC floorplan.

synthesized using a 65nm low-power technology library from ST microelectronics [99], achieving a maximum operating frequency of 440.9MHz. Once the LEON processor is synthesized it is imported as a macro block during the rest of the MPSoC design process.

For the communication infrastructure a 16-node mesh NoC will be leveraged. The reference topology of our experiment is a 4x4 mesh network where each switch is connected to a core and its associated memory using links of 1.5mm. Two different NoC approaches have been designed. The first design, the one shown in Figure 4.19(a), uses only one clock domain for the entire network while the second design, shown in Figure 4.19(b), is a network inferred as a collection of mesochronous domains, instead of a global synchronous domain, yet retaining a globally synchronous perspective of the network itself. For both NoC implementations the xpipesLite NoC architecture [98] is used as baseline experimental setting to implement both platforms. The flow control protocol used by xpipesLite is stall/go: a forward signal, synchronous with data, flags data availability (valid), while a backward signal flags a destination buffer full (stall) or empty (go) condition.

Both the synchronous and the mesochronous platforms have been designed to be seamlessly integrated into an industrial design flow using commercial tools for physical synthesis. Only standard cells are used and no full custom component. As far as the physical synthesis is concerned, the same bottom-up methodology has been utilized for both platforms. Specifically, each network



(a) Synchronous NoC

(b) Mesochronous NoC

Figure 4.19: Schematic of the different NoC scenarios

switch has been placed and routed in isolation with a target frequency of 500MHz. The clock tree of each switch has been synthesized with a tight skew constraint of 5% of the target clock period. Once the local clock tree is characterized with its input delay, skew, and input capacitance, a macromodel is built in order to be used in the next design step. Furthermore, in order to implement a hierarchical clock tree synthesis, a buffer has been inserted into the input clock pin of each switch block. Once the switches have been placed and routed, they are imported as macro blocks in the main network design along with their libraries detailing both timing and physical characteristics. The next step consists of performing a top-level clock tree synthesis by leveraging the switch macromodels previously extracted. In fact, this model can be used to characterize the bottom clock tree given that these local clock trees will not be modified by the place&route tool. Please notice that the hierarchical clock tree synthesis has been used both for the synchronous and the mesochronous platforms, since this is a standard methodology for parallel hardware platforms. The only difference is the skew constraint in the top level clock tree, which can be loosened for the mesochronous design while should be tightly enforced for the synchronous one. The final step of our hierarchical methodology consists of routing the switch-to-switch links and performing parasitics extraction for accurate static-timing analysis and power estimation. Timing closure for both the synchronous and mesochronous NoC has been achieved at 500 MHz by performing exactly the same physical synthesis steps.

4.3.2 Injecting Variations into the LEON Processor

In order to analyze the impact of process variations in the LEON processor, systematic front-end variations, random front-end variations, and back-end variations have been introduced in the processor design. The variability sources considered are: transistors channel length ($\sigma_{L_{eff}} = 12\%$), threshold voltage ($3\sigma_{V_{th}} = 33\%$) and the metal thickness ($3\sigma_t = 10\%$). These values reflect expected variations for a 45nm technology node [28].

Figure 4.20 shows the impact of process variations in the LEON processor. In the top plot the probability density function of the maximum achievable frequency is depicted. According to the results shown in this plot the maximum operating frequency of cores within a chip vary around +-15% of the nominal frequency as a consequence of process variation. Concretely, the measured standard deviation of the maximum achievable frequency is 9.44%. This value shows a significant impact of variations in the maximum operating frequency. Note that the measured variations can compromise either the performance or the reliability of the LEON processor. In fact, there exists a trade-off between the reliability and the performance as shown in the bottom plot. In this plot the timing error probability is represented as a function of the operating frequency. Note that this plot shows that the use of a frequency-island design scheme is mandatory as the use of this architectural feature allows each processor to be clocked independently. On the contrary, if a synchronous clocking

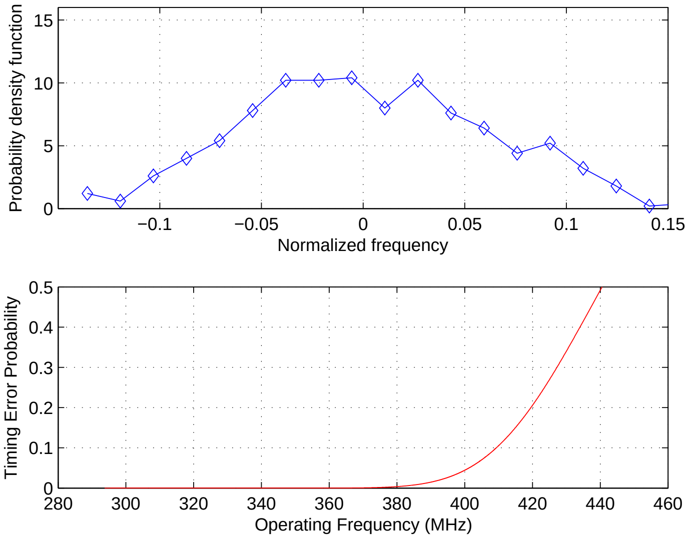


Figure 4.20: Impact of variations in the Leon processor.

scheme is used, processor frequency should have to be lowered below 370MHz to ensure the absence of errors while achieving a reasonable yield.

We have also measured how important is the impact of random variations in the LEON processor. The results obtained confirm that the contribution of systematic variations is considerably more important than the contribution of random variations. This can be explained by the fact that the critical path of the LEON processor synthesized is very deep and this makes random variations to be compensated. This result is specially important in order to chose the way process variations are compensated. On one hand, the use of processor frequency islands can partially face the impact of process variations in processor frequency, as the frequency of processors can individually be adjusted. On the other hand, operating frequency can be kept high if timing violations can be tolerated on run-time. This can be done by using special circuitry at some performance cost [92]. In our case, as systematic variations heavily impact processor frequency, individually adjusting the frequency of processors is the best choice to retrieve most of the performance of the MPSoC design. However, scenarios where random variations severely impact processor performance may require the combination of both approaches to efficiently face delay variations.

4.3.3 Synchronous vs Mesochronous NoC

In order to show how variations affect synchronous NoC performance, variations into the synchronous NoC design have been injected. Note that the severity of variations injected is equal to the variations expected for both 65nm and 45nm technologies (Table 3.2). Figure 4.21 shows the probability density function of the maximum achievable frequency of the design in the presence of parameter variations. As shown, the mean of the probability density function (pdf) of the maximum achievable frequency is lowered with respect to the nominal design frequency in both cases. Note that, as expected, for a 45nm technology the frequency reduction caused by variations is more noticeable. Moreover, variations make design frequencies to vary between -10% to 4%, and between -12% and 6%, of the nominal frequency, for 65nm and 45nm technologies, respectively. Note that measured variations will cause a non-negligible impact on circuit performance even for a 65nm technology where variations are expected to be low. Concretely, we should lower the frequency of the manufactured design from 571.2MHz to 514.1MHz in the case of 65nm, and from 571.2MHz to 503.7 in the case of 45nm to avoid timing violations. Note that despite the target frequency was set to be 500MHz, the synchronous design can operate at higher frequencies thanks to available slack. The effect of manufacturing variations in the NoC design also causes in last instance the reduction of application predictability. Moreover, the percentage impact of such delay variability on the NoC operating speed is of course expected to grow as NoCs are synthesized for higher speeds than considered in this example.

In order to face the reduction of performance, reliability, and predictability of the NoC architectures, designers might think of a mesochronous NoC architecture (see Figure 4.19(b)). We will now demonstrate the ability of our tool to quantify potential benefits of a specific mesochronous synchronization infrastructure ahead of actual silicon implementation. The robustness of both platforms is hereafter compared. To do so, variations are injected into the clock tree of both designs. The motivation of this lies on the fact that the

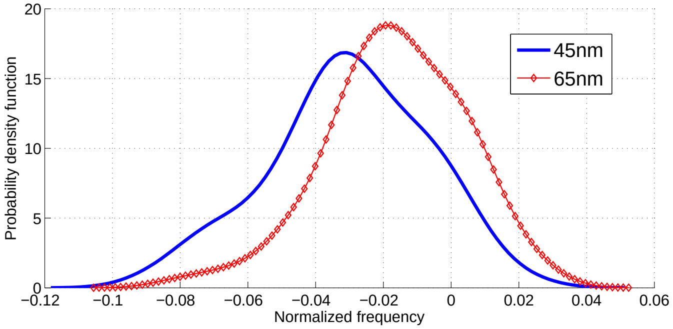


Figure 4.21: Measured variations for the synchronous NoC as expected for 45nm and 65nm design.

two architectures differ only in the switch interfaces, therefore variability effects affecting internal switch gates and/or nets are likely to have the same impact on the designs under test. Figure 4.22 shows the probability density function (pdf) of the maximum achievable frequency of both synchronous and mesochronous designs when variations are injected to the nominal design. The variations injected are both systematic and random. Concretely, the variability sources considered are: transistor channel length ($\sigma_{L_{eff}} = 12\%$), threshold voltage ($3\sigma_{V_{th}} = 33\%, 58\%$)[4], and the metal thickness ($3\sigma_t = 10\%$). These values reflect expected variations for a 45nm technology node [28].

Results confirm that for the synchronous design the variability injected into the clock tree has a considerable impact on the maximum achievable frequency. Concretely, we have measured a standard deviation of the maximum frequency equal to 6.8% and 6.2% for the cases of low and high random variations, respectively. Note that differences in the pdf for high and low random variation are minimum and that the probability density function of maximum achievable frequency is centered. This can be explained by the fact that random variations do not have a significant impact in the top level clock tree, thanks in part to the large size of clock buffers.

The mesochronous design is clearly able to absorb the variations introduced in the clock tree, since it is able to preserve the nominal frequency in all cases. Obviously, since the clock skew directly impacts the critical path of

[4] Those values of $3\sigma_{V_{th}}$ represent the low and high variation scenarios, respectively.

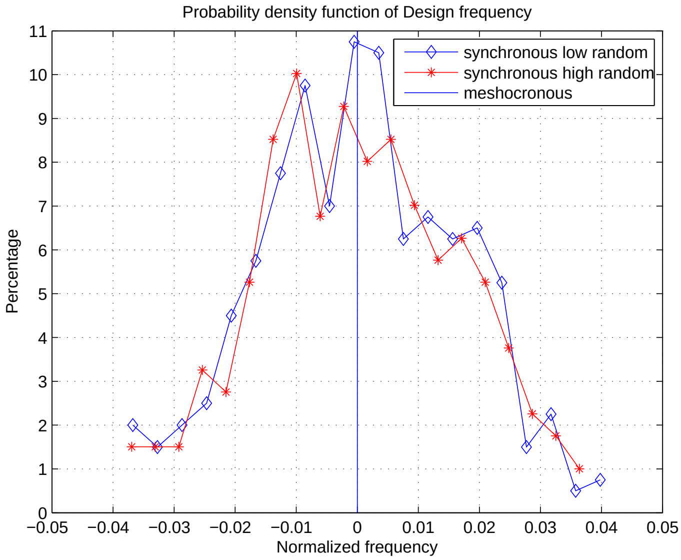


Figure 4.22: Variability robustness comparison.

the synchronous design, it is sometimes possible that this latter works at a higher speed than the mesochronous one (depending on the sign of the skew). However, this argument is not able to counter the conclusion: the mesochronous NoC proves more robust to process variations in the top-level clock tree.

Regardless of the outstanding results provided by the mesochronous NoC design, we would like to remark the fact that our framework makes possible to assess the benefits of a given design (the mesochronous in this case) much prior to its real implementation, thus reducing time-to-market at the same time that yield is increased.

4.4 Conclusions

In this chapter, three different ways of applying the variability model presented in Chapter 3 have been shown. In the first case study the impact of process variation in link delay for NoCs has been presented. Results of this study confirm that link variation cannot be neglected anymore. In this sense variability models for NoC-based CMP architectures should also consider link variation in addition to variation in cores and switches. Other important results of this first study are the ones related with the impact of the different variability sources on delay variation. First, it is specially important the fact that resistance variation has a negligible impact in NoC links when links are designed with power and timing constraints. Second, we have measured an increasing importance of random variation in delay uncertainty with technology scaling. This is the confirmation that random variation will play a very important role in future technologies. Consequently, new approaches able to face the presence of random variation will be required to better exploit link wire characteristics.

In the second application scenario we have applied the variation model to an 8x8 mesh NoC implemented in 45nm technology, collecting data on how variability affects routers and links and also the individual stages of routers.

Finally, in order to show the potentials of the developed framework, process variations have been injected into a complete MPSoC design. First, 45nm variations were introduced in a synthesized LEON processor. Results from this first analysis show how maximum operating processor frequency is noticeably degraded because of process variations. Additionally, the impact of parameter variation in the NoC architecture interconnecting the processors has been also characterized. One one hand, we have shown that smaller technologies increasingly impact the performance of synchronous network-on-chip designs. On the other hand, we have shown how using more complex clocking schemes, like a mesochronous one, NoC design variations can be absorbed making possible to preserve the nominal frequency of the NoC.

Chapter 5

Addressing Within-die Systematic Variations in NoC-based Designs

In the context of CMPs, parameter variations may cause a decrease both in performance and in energy efficiency if the presence of process variations is neglected. As a consequence of within-die variations, CMP cores and memories will present different maximum achievable frequencies and different values of leakage power. Additionally, the NoC used to support inter-core communications is also particularly sensitive to process variations as the presence of components in a network working at different frequencies can cause a considerably reduction in the overall performance of the network [37].

Large CMPs pose OSs new burdens for the efficient execution of applications. For example, as different cores are located at different distances from the on-chip memory controllers, the performance of a given application may be influenced by the exact location of the cores that application is assigned to. The same is true regarding the distance among the cores executing a given application. In order to reduce that distance and therefore allow for faster inter-thread communications, cores assigned to the application should be selected from a contiguous region in the die. Moreover, the shape of that region will determine the maximum diameter of the subnetwork containing those cores, thus directly affecting communication performance and therefore application execution time.

As can be seen, current and future CMPs demand additional enhancements in the OS schedulers in order to efficiently take advantage of the tremendous amount of available resources. However, in order to completely leverage those resources in an efficient way, variability data must also be considered in the mapping process. If such data are not taken into account, an application may be assigned to a slower part of the chip while other faster cores remain idle. Also, if variability data are not considered in the mapping process, a given application demanding a relatively large number of resources may be mapped to a region composed of slow and fast devices, thus partially wasting resources as the faster cores and routers in that region will end up working at the speed of the slowest ones.

In this chapter a mapping policy that makes use of variability information is proposed. This policy efficiently schedules applications in CMP systems under process variations. In this proposal, performance is achieved by first considering fast CMP regions and efficiency is provided by choosing regions presenting uniform frequencies. The goal of this mapping strategy is avoiding to map threads to regions where routers and cores present very different speeds, as this would cause both an inefficient utilization of resources and the induction of communication bottlenecks in the NoC. There also exist some possible improvements over this basic approach that make possible to perform an efficient management of the chip by increasing core utilization and minimizing the fragmentation introduced when a variability-aware mapping policy is chosen. These enhancements are also considered in this work.

5.1 Related Work

As shown in the previous chapter, process variation causes that different tiles and links present different maximum achievable frequencies. Therefore, either the entire chip frequency is reduced to that of the slowest one or different tiles are clocked at different frequencies. In this way, some initial previous work has been carried out with multiclock chips in order to face variability effects [69]. Nevertheless, although multiclock systems can be initially used to mitigate the frequency slowdown suffered by a synchronous CMP as a consequence of process variations, these systems are not able by themselves to minimize the impact of variability in energy and performance. In this sense, [36] states that performance improvements of per-core frequency islands in CMPs with respect to a synchronous design are outweighted by the performance loss introduced due to the synchronization mechanisms required when crossing different clock domains.

Process variation does not only affect the way computing cores and NoC components are designed in order to minimize its impact on them. Variability also influences other higher-level layers of the system hierarchy, like the operating system. Effectively, several recent works [100] [39] [19] have shown that mapping strategies used by the operating system to map processes to cores must take into consideration not only the availability of idle cores but also new technology-dependent constraints, such as process variation. Considering this kind of technology related features is becoming mandatory in order to design mapping policies that improve the performance of the system by reducing application execution time and/or consumed power. For example, in [100] different mapping strategies are presented to face core-to-core process variations. Similarly, in [39] is proposed a dynamic variation-aware thread mapping algorithm. Unfortunately, both studies focus on crossbar based CMP architectures where all cores access a shared one-hop away L2 cache by means of that crossbar. Thus, the usefulness of these studies is limited to CMPs featuring a small number of cores, where this kind of interconnect is feasible. For larger CMP configurations, a NoC replacing the crossbar is required, as argued before. In this new context, distances are larger than one hope and, therefore, additional issues must be taken into account when performing the mapping. This is exactly what [15] [43] do, where an analysis of static and dynamic mapping strategies, respectively, is carried out in the context of NoC-based CMPs. However, although these studies consider the effect of using a NoC to interconnect cores and caches, they do not consider the effects of process variation at all.

Up to our knowledge, only the work carried out by Intel [19] analyzes the implications of different mapping strategies in a tile-based CMP that uses a NoC, achieving very high improvements both in application execution time and energy/flop. However, this work only analyzes the possible benefits of variation-aware mapping strategies when applications are mapped in isolation and, additionally, a 3D stacking memory architecture is assumed by default. This is in fact the best application scenario of a variation-aware mapping approach.

5.2 Towards a Variation-Aware Application Mapping in NoC based CMPs

In this section we analyze the main design issues for performing a variation-aware application mapping. For that purpose, in first instance the implications of frequency islands designs are analyzed. This analysis shows how the GALS paradigm is unable by itself to efficiently face the presence of process variations. Finally, we characterize the network performance of regions with different shapes.

5.2.1 Architecting Frequency Islands in GALS-based NoCs

According to the results in Chapter 4, where the impact of parameter variations in a NoC was shown, the traditional synchronous design technique is not feasible any more because NoC clock frequency should be lowered to match the frequency of the slowest component in the network, noticeably reducing network performance. This fact is widely known by NoC architects. In fact, NoC designers propose to use the GALS philosophy in order to keep performance in such scenarios featuring different frequencies. Frequency Islands (FI) is a way of implementing GALS systems. Actually, FI schemes have been proposed to face process variation in NoC-based CMPs [69] [36]. In these systems, different chip domains are able to work at different frequencies, and communication between domains is performed by means of synchronizers. Such systems are commonly referred to as MultiClock Domain (MCD) chips. In a MCD architecture, each FI requires the generation of an appropriate local clock signal.

As mentioned before, there are already some working examples of MCD chips by the main processor manufacturers. For instance, a per-core frequency control is already included in AMD's Opteron Quad-Core [20]. Another cost-effective way to provide the appropriate clock signal is the one used in the

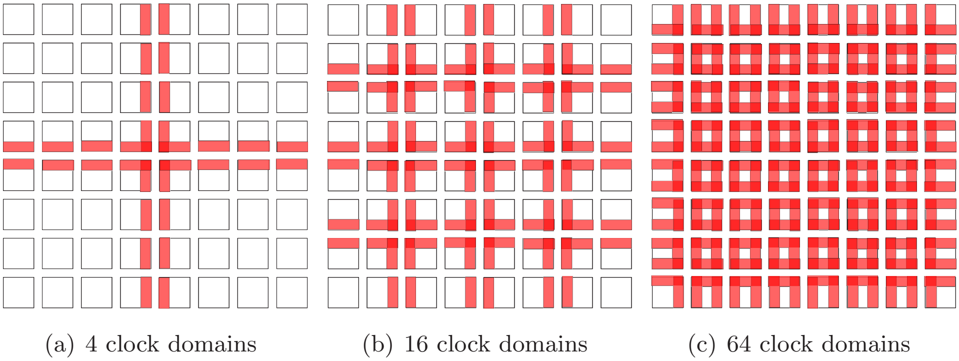

(a) 4 clock domains (b) 16 clock domains (c) 64 clock domains

Figure 5.1: Synchronization granularity.

Montecito CPU by Intel [24]. In this approach only one global PLL (phase loop lock) is required and clock dividers are used to provide the range of supported frequencies.

In a FI-based design, variability-awareness is achieved by setting island frequency to the maximum sustainable frequency by the elements in that island. In order to know the clock frequency of each domain, test vectors are applied to chips after the manufacturing process. To do so, the regular test process has to be extended in order to determine the maximum achievable frequency of each domain. Note that tests have been traditionally performed to bind microprocessors to some frequency level in order to sell them according to their speed. Finally, once the maximum clock domain frequency is found for every frequency island, these data can be recorded into dedicated ROMs.

It is also possible to deploy frequency islands in a NoC-based CMP. In this case, a tile-based frequency island partitioning scheme could be leveraged, where the entire tile, including both the processing element and the router, works at the same frequency. In a tile-based multiclock domain approach the granularity of the synchronization may range from one individual clock per tile to one clock per die. Between both ends we may find MCD configurations that group several tiles into one clock domain. For example, Figure 5.1 shows several ways of achieving a multiclock domain configuration in a 64-tile NoC-based CMP, presenting the synchronizer location to support 4, 16, and 64 clock domains. In each domain, clock frequency is set by the slowest component of the region. Notice, however, that the granularity of clock domains will represent an overhead in both the number of synchronizers required and the clock generation circuitry.

Deploying multiple clock domains in a NoC-based CMP requires some kind of synchronization mechanism to be included at the NoC level. In this way, synchronization between domains may be performed by means of a dual-clock FIFO. There exist several recent implementations [13] [61] allowing cost-effective synchronization by means of a dual-clock FIFO. From the architectural point of view, the modification of the GALS-based NoC is minimum as we only have to replace regular input buffers by dual-clock FIFOs at the routers in the border of the clock domain in order to allow communication between different clock domains. However, synchronization still incurs in area and latency penalty. Actually, this penalty, and the associated performance, will be analyzed in next section.

5.2.2 Consequences of Variability in GALS based NoCs

In this section we analyze the consequences of variability in GALS-based NoCs. For that purpose, we have simulated the 8x8 CMP test bench network presented in Section 4.2.1. Remember that routers in that network present a maximum achievable frequency of 1.33GHz. The performance metrics we have collected are mainly link and buffer utilization, message latency, network throughput, and application execution time. To do so, we have run simulations both with synthetic traffic and with application traces. For the synthetic traffic we considered that average message generation rate is constant and the same for all the nodes. The inter arrival time is generated using a uniform distribution. Once the network has reached a steady state, the flit generation rate is equal to the flit reception rate (traffic). We have evaluated the full range of traffic from low load to saturation. With respect to message destination, we have considered that it is randomly chosen among all the cores in the network. Message length is constant and equal to 5 flits. Regarding application-driven simulations, we have chosen 4 different applications from the PARSEC benchmark suite [4], as representative parallel application workloads. More specifically, we have made use of the freqmine, swaptions, canneal, and blackhole PARSEC kernels. These four benchmarks are built with Virtutech Simics [59] to run on 4, 8, 9, and 16 cores.

Clock Domains	1		4		16		64	
Synchronizers	0		32		96		224	
correlation (ρ)	0.5	1	0.5	1	0.5	1	0.5	1
Th	0.85	0.85	0.88	0.89	0.92	0.94	0.98	0.98

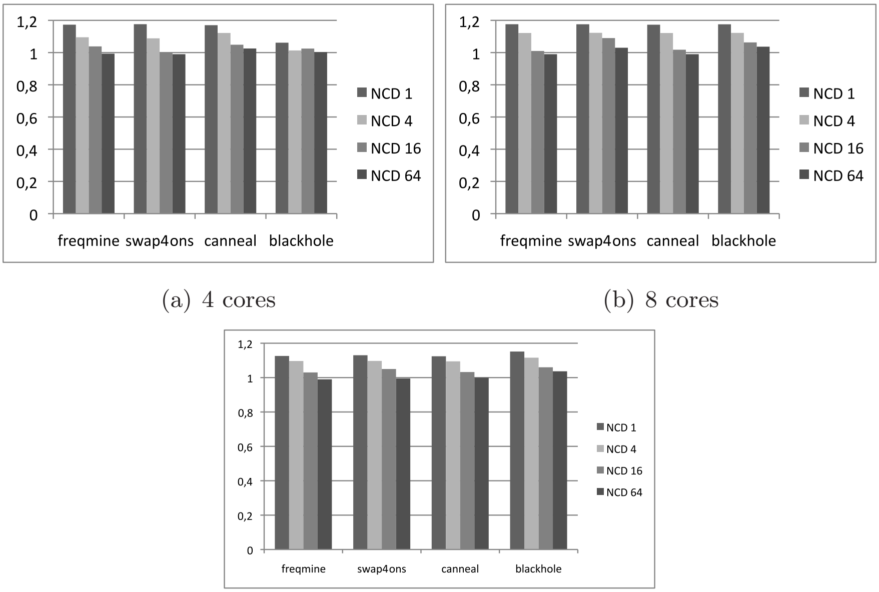
Table 5.1: Raw performance of a 64-tile CMP depending on the number of clock domains

Implications of the Granularity of Frequency Islands

As mentioned before, a fine-grain frequency island granularity causes more synchronizers to be required, thus incurring in larger area and latency penalties. However, a larger granularity also provides a better adaptation of the network to the diversity of frequencies of the underlying devices, thus probably providing better overall performance despite of the extra latency caused by synchronizers. Therefore, it is necessary to investigate this trade-off.

Table 5.1 shows the number of dual-clock FIFOs required to support the different number of clock domains shown in Figure 5.1, and the raw performance associated to each configuration. Note that in the absence of process variations all CMP tiles work at the same frequency and therefore total throughput Th is equal to NTh_i, where Th_i represents the individual tile throughput and N is the number of tiles in the chip. On the other hand, for a CMP affected by process variations, it is possible, by increasing the number of synchronizers, to retrieve most of the throughput loss caused by frequency variations, as shown in this table in the row labeled as " Th". That row shows the average throughput for all the 200 chip instances "manufactured" in Section 4.2 (100 instances for correlation 0.5 and 100 instances for correlation 1). Throughput is normalized to the one achieved by a variability-free chip. For the chips considered in this study, throughput loss ranges from 15% for a synchronous design (1 clock domain) to only 2% when we have 64 clock domains.

Figure 5.2 shows how application execution time varies with the granularity of clock domains. More concretely, we have simulated the simultaneous execution of 4 identical instances of a given application running on 4, 8, and 16 cores, respectively. We have carried out these experiments for the 4 appli-



(a) 4 cores

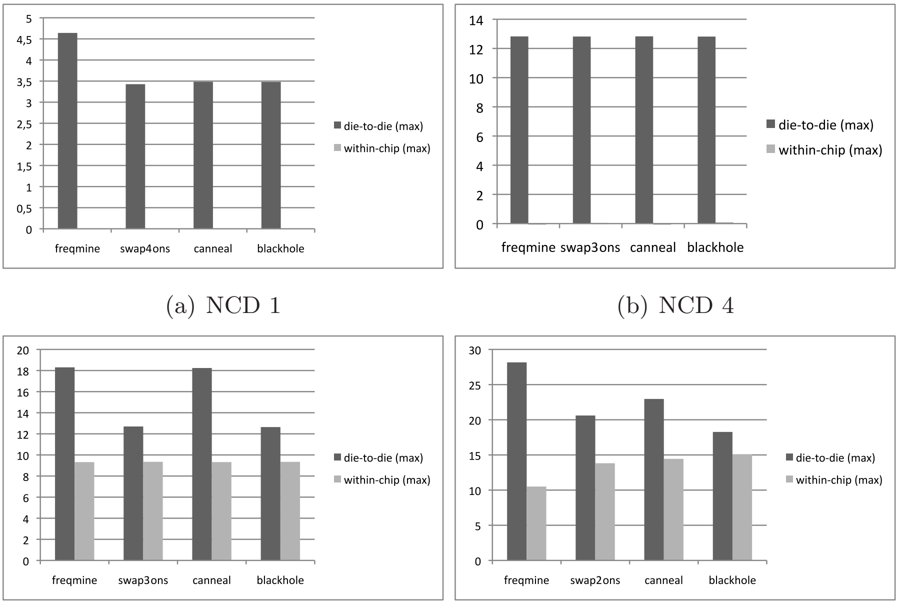
(b) 8 cores

(c) 16 cores

Figure 5.2: Application execution time relative to the variation-free CMP.

cations mentioned above and for all the chips produced in Section 4.2. In the experiments, we have used the 4 different MCD chip configurations shown in Table 5.1. In this way, label “NCD 1” in Figure 5.2 refers to a chip containing a single clock domain, where frequency has been set to the slowest element in the network. In the same way, labels “NCD 4”, “NCD 16”, and “NCD 64” refer to MCD chips divided into 4, 16, and 64 clock domains, respectively, as shown in Figure 5.1. Moreover, notice that results in Figure 5.2 are normalized to the average execution time of the four instances for the variability-free chip, where all tiles run at 1.33 GHz. That is, NCDi results for freqmine, for example, are normalized to the average execution time of the four instances of freqmine in the absence of process variations. Note that the execution time of the four instances in the absence of variability will not be exactly the same, because of the intricacies of cache coherency and network traffic and accesses to memory controllers. Finally, results in Figure 5.2 show the execution time for the replica of the application that took longer to finish in any of the chips.

As can be seen in Figure 5.2, as the number of frequency islands is in-



(a) NCD 1

(b) NCD 4

(c) NCD 16

(d) NCD 64

Figure 5.3: Across-chip and within-chip variation of the application execution time.

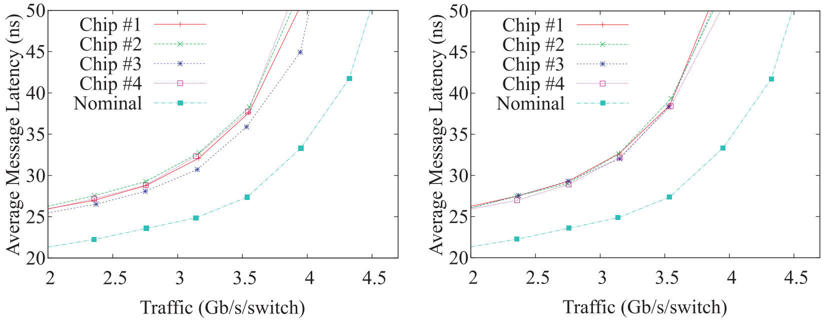
creased, performance also increases, despite of the additional latency introduced by the synchronizers. The reason is obvious. As the chip is split into more frequency islands, the better capabilities of the faster elements in the chip are leveraged, what is not possible for a smaller number of frequency islands. Moreover, it can also be seen that most of the performance is retrieved by dividing the 64-core CMP into 16 clock domains, although increasing the amount of clock domains to 64 still delivers additional improvements. Finally, in the case for 64 clock domains, some of the applications are slightly faster when running in a variability-affected CMP than when running in a variability-free chip. The reason for this is that variability may increase the frequency for some components of the chip. Therefore, under the right combination of circumstances, lower execution times could be achieved. Nevertheless, this is a probabilistic issue, and it may also happen that longer execution times are experienced. In general, variability causes that application execution time becomes uncertain, as shown in Figure 12.

Figure 5.3 shows the maximum application execution time variation due to process variability from one chip to another (across chips variation) and also for the different application instances running on the same chip (within-die variation). Across-chip variations are computed as the maximum difference between applications running in the same location for all the chips considered. In the case for within-die variations, data in Figure 5.3 show in percentage the maximum difference in execution time for the different instances run in a given chip. Moreover, Figure 5.3 shows that differences in execution time are noticeable, specially as the number of clock domains increases. These results clearly show that processor manufacturers will have trouble assessing in advance the performance of their future products in advance, as variability will increase uncertainty. On the other hand, regarding within-die differences, they are negligible in the case for one clock domain, as the four instances of a given application are executed in cores running at the same frequency. In the case for four clock domains, differences are almost zero because the four instances have been executed in cores belonging to the same clock domain. However, when cores devoted to the applications run at different frequencies, within-die differences arise, as can be seen in the cases for 16 and 64 clock domains.

Characterizing Communication Bottlenecks in MCD Chips

In this section we dig into the communication troubles caused by asynchronisms in multiclock domain chips. To do so, we use uniform traffic to characterize the traffic behavior of a GALS-based NoC. In this section only results for the case of 64 clock domains are shown. However, the conclusions drawn in this section also apply for other MCD chip configurations.

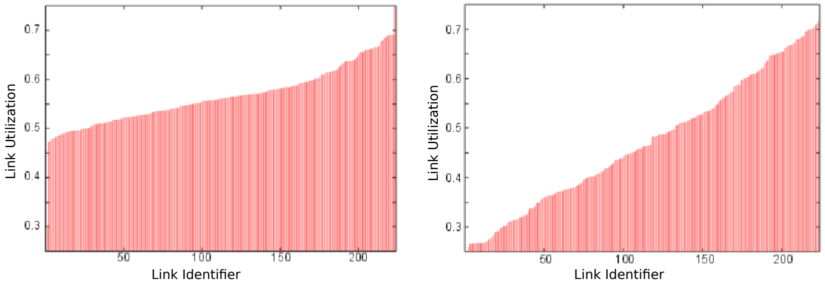
Figure 5.4 shows the average message latency versus received traffic for several of the chips analyzed. Low and high correlation values are considered ($\rho = 0.5$ and $\rho = 1.0$). As can be seen, in the presence of variability (curves labeled “Chip #n”) the network is able to manage almost 20% less traffic than in the absence of process variation. Moreover, average message latency is increased by 23% even for low traffic loads. On the other hand, it is interesting to notice that the overall performance of the network in the presence of variability is almost independent of the exact characteristics of that vari-



(a) $\rho = 0.5$

(b) $\rho = 1$

Figure 5.4: Network performance in the presence of process variation.



(a) Nominal network

(b) Network with variability

Figure 5.5: Link utilization in the presence of process variation.

ability. This is shown by all the "Chip #n" curves being almost overlapped. Nevertheless, when the correlation of the manufacturing process is lower, we can see more differences in network performance.

One of the reasons for the differences in performance shown in Figure 5.4 is the lower average network bandwidth caused by random variation in links (note that systematic variation does not cause a reduction in aggregated network bandwidth). However, this reduction in network bandwidth does not completely explain the plots in Figure 5.4. An important contribution to that performance reduction is shown in Figure 5.5, that displays link utilization for all the links in the network when it is close to saturation both with and without process variation. As can be seen, process variation causes larger differences in link utilization, as shown in Figure 5.5(b), where a few links present a much larger utilization (on the right end) than others, or a much

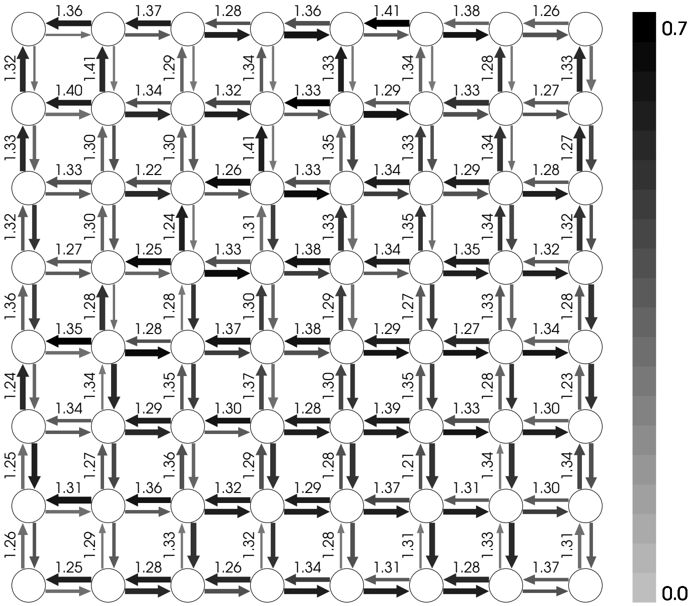


Figure 5.6: 64-core CMP bottlenecks as a consequence of link delay variation.

lower utilization (left end). This uneven distribution of link utilization helps to explain the performance loss in Figure 5.4. Finally, these utilization numbers are graphically displayed in Figure 5.6. Numbers next to a link represent the frequency for that particular link.

Similarly to link utilization, buffer utilization is also affected by process variations. Figure 5.7 shows the variation of buffer utilization with the flit injection rate. Plots labeled as nominal represent the network working at the nominal frequency whereas plots labeled as "corr=1" represent the average results of all chips with high correlation. Results for low correlation ($\rho = 0.5$) are not shown as they are very similar to high correlation results. This figure shows how buffer utilization decreases in a NoC with several clock domains. The ideal network presents a buffer utilization around 50% while in a network with process variations the average buffer utilization is around 40%. Additionally, it is possible to see how in a GALS-based NoC in the presence of process variations the standard deviation of buffer utilization is much greater than in a fully synchronous network where buffer utilization is more uniformly distributed. This behavior is better shown in Figure 5.8 where the spatial

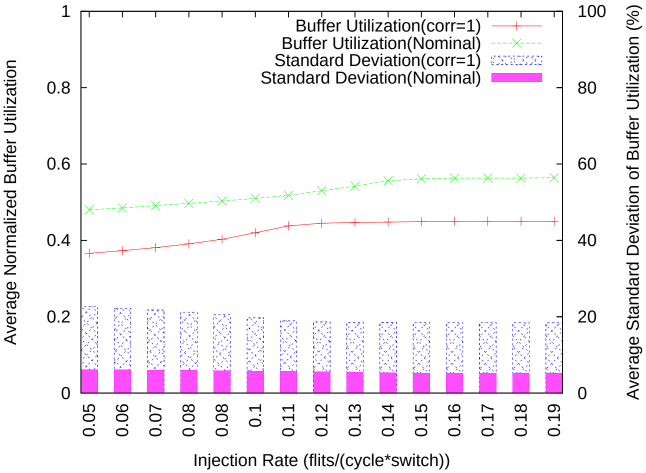
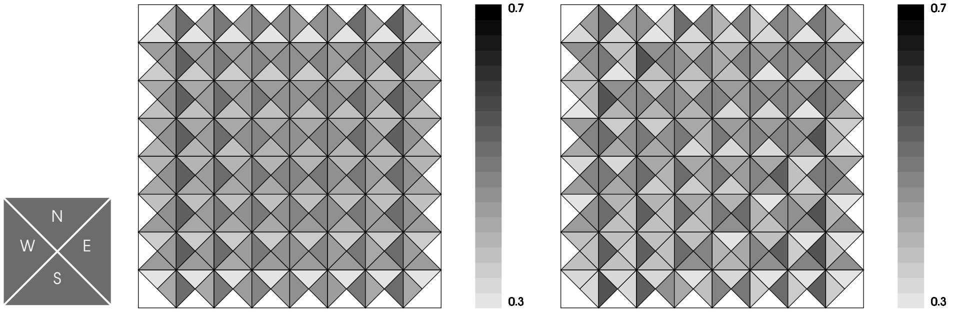


Figure 5.7: Buffer utilization in a 64-clock domain CMP.

distribution of buffer utilization is represented for an injection rate of 0.11 flits/cycle. The left side of the figure shows an enlarged representation of a router (big square) and its input buffers (triangles). The rest of the figure represents the utilization of input buffers in a nominal network and a chip presenting variation with high correlation ($\rho = 1$). Note that frequency distribution of this chip was shown in Figure 4.17(b). As shown in this figure, in a GALS-based NoC in the presence of variations, there will exist, on one hand, hot spots as a consequence of the non-uniformity of the operating frequency distribution and, on the other hand, some underutilized buffers. This behavior causes in first instance a decrease in the average buffer utilization, as previously shown, and in last instance a reduction in network performance as it was shown in Figure 5.4.

5.2.3 Implications of Region Shape on Network Performance

In order to assign to a given application several cores of the chip, one of the concerns that quickly arise is which should be the shape of the region those cores make up. On one hand, allowing any shape would allow to better adapt the underlying resources to the needs of the application. For example, if an application requires 8 cores and the region in the chip presenting the highest frequency uniformity is L-shaped, then that region would be assigned to the



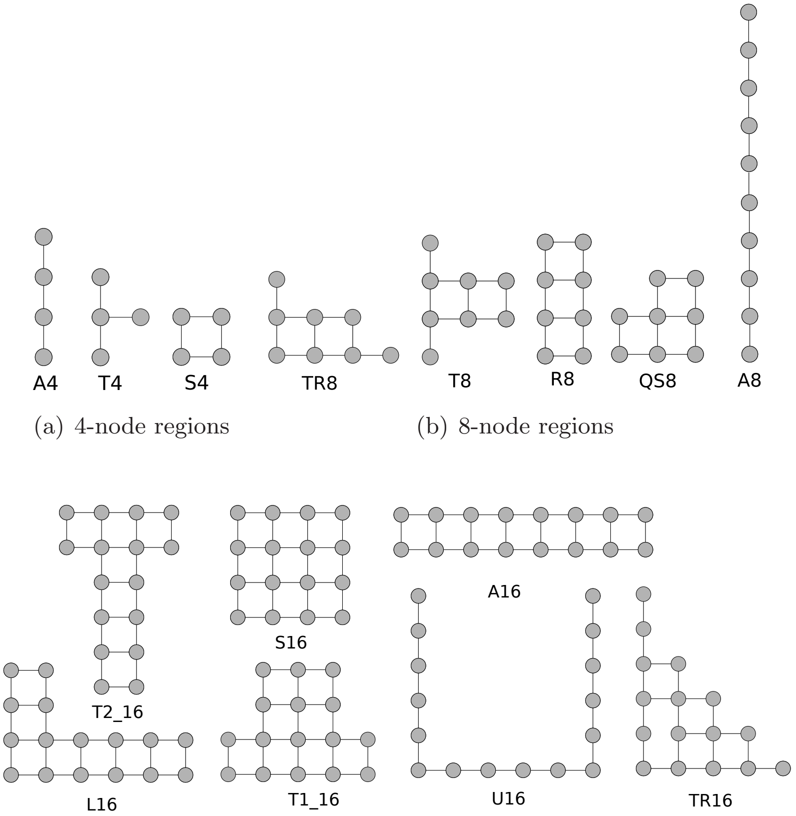
(a) Router (b) Nominal network (c) Network with variability ($\rho = 1$)

Figure 5.8: Buffer utilization in the presence of process variations.

application. However, some shapes may provide lower performance due to a lower bisection bandwidth. For example, the L-shape region assigned to the previous application may present lower performance than a square-like region, which is expected to report better communication capabilities. Therefore, in order to analyze which are the relative benefits of using one shape or another, in this section we analyze the performance of several regions having different shapes. We carry out this analysis in the absence of process variation, and with synthetic traffic (uniform distribution of destinations) in order to isolate the effect of the shape from other additional concerns that could mask this analysis.

Figure 5.9 shows the different main shapes for regions containing 4, 8, and 16 cores. As can be seen, some of them provide noticeable shorter paths and larger connectivity than others. It is expected that these better characteristics translate into better performance. This is shown in Figure 5.10, that plots, for each of the shapes in Figure 5.9, the performance achieved under uniform traffic. As can be seen in Figure 5.10, in the case for 4-core regions, the best performance is achieved by the square shape, as expected. Similar results are provided for 8 and 16-core regions. Note that square-like shapes present better performance than other shapes with larger paths, as expected.

In the following, we are going to focus the analysis in next section on square regions, given that they are the ones presenting much larger performance.



(a) 4-node regions

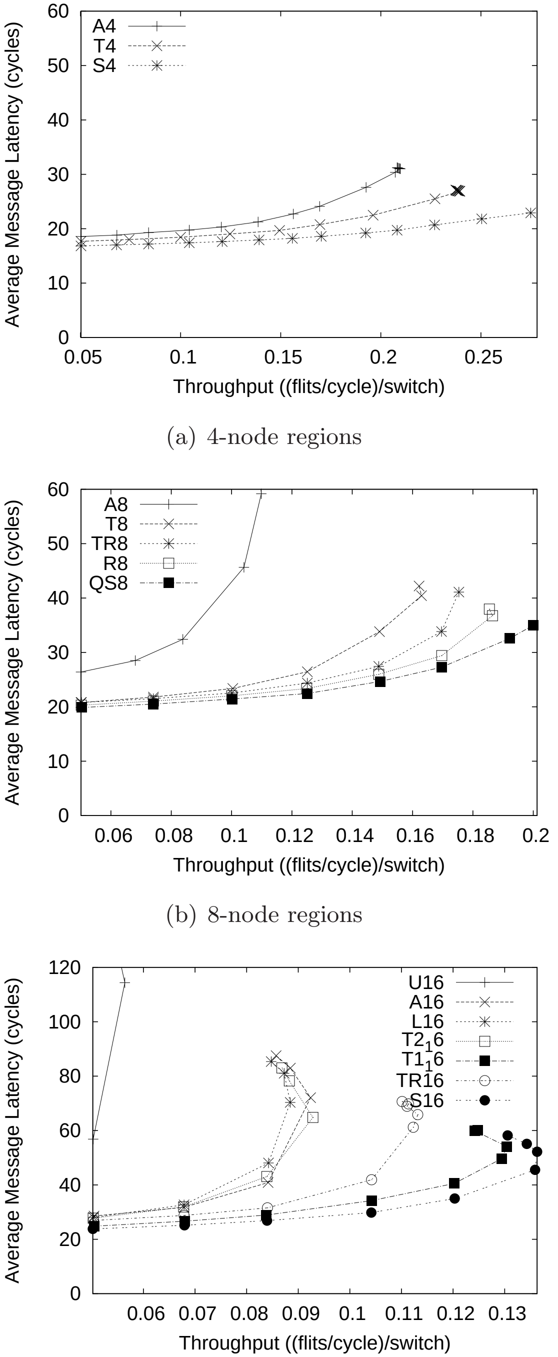
(b) 8-node regions

(c) 16-node regions

Figure 5.9: Main shapes for 4, 8 and 16-core regions

5.3 Variation-Aware Application Mapping in NoC-based CMPs

Once the implications of the frequency islands granularity and the shape of regions are analyzed we are in a position to propose a variation-aware mapping algorithm in a suitable CMP scenario. Application mapping is critical for NoC based platforms as the performance of applications will strongly depend on the resulting mapping. There exist several choices to perform the mapping and the selection of the optimal strategy will considerably benefit the overall performance of the chip in terms of application performance and power consumption. When the mapping is carried out in a static way, the



(a) 4-node regions

(b) 8-node regions

(c) 16-node regions

Figure 5.10: Performance for each region shape

compiler determines the suitable mapping for threads and, once the threads are mapped, they remain in the same core until execution finishes. On the contrary, it is possible to choose a dynamic mapping strategy that is able to move threads from one core to another in order to take advantage of changes with time in data access patterns.

In this section we evaluate the possible benefits of variation-aware mapping strategies where the mapping decisions are taken at run time and take into account variability data. As a first approach, we do not allow to move threads once the application is mapped. Furthermore, we consider that the mapping algorithm must provide a set of neighbor cores forming a region, that is, a set of adjacent cores. Therefore, all threads from a given application are forced to be mapped into a single region. In other words, despite of having enough idle cores spread across the CMP, if the application threads cannot be mapped into a single region, then the mapping is not performed until other applications complete their execution and there are enough free cores for the new application being mapped. Note that providing applications sets of cores grouped in a single contiguous region supports later virtualization mechanisms that may require traffic isolation. Additionally, only regular regions are considered in this study as routing is perform leveraging the XY routing algorithm. However, as there exist recent efforts that provide efficient routing mechanisms in non-rectangular regions [26], it will be interesting as future work to explore the benefits of mapping application threads to irregular regions.

As told before, process variation will cause differences in frequency and leakage across CMP tiles. Frequency variation can cause an inefficient utilization of the CMP. This is a consequence of basically two reasons. On one hand, in a NoC-based CMP having tiles working at different frequencies, a reduction in network performance arises due to bottlenecks appearing when slow and fast regions are mixed. This was shown in Figure 5.8, where it is represented the average buffer utilization of NoC routers when all tiles work at the same frequency (Figure 5.8(b)), and the buffer utilization of routers when, as a consequence of process variations, frequency of tiles is different (Figure 5.8(c)). Results of Figure 5.8(c) were generated using a uniform traffic message distribution, and the values of average buffer utilization were taken when the injected traffic in both scenarios was exactly the same. Figure 5.8 showed

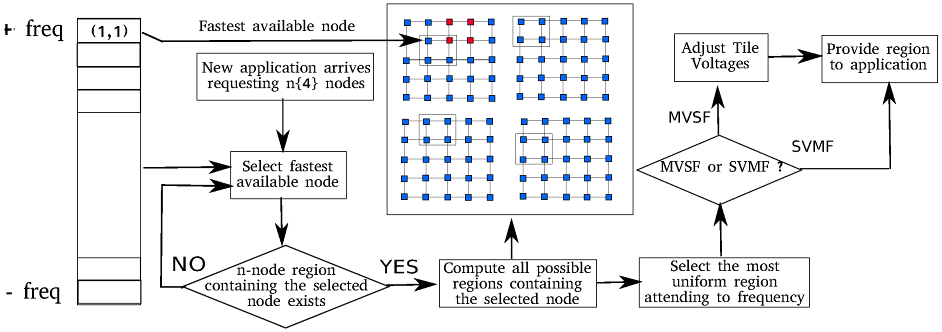

+ freq
(1,1)
- freq
Fastest available node
New application arrives requesting n{4} nodes
Select fastest available node
NO
n-node region containing the selected node exists
YES
Compute all possible regions containing the selected node
Select the most uniform region attending to frequency
MVSF or SVMF ?
MVSF
SVMF
Adjust Tile Voltages
Provide region to application

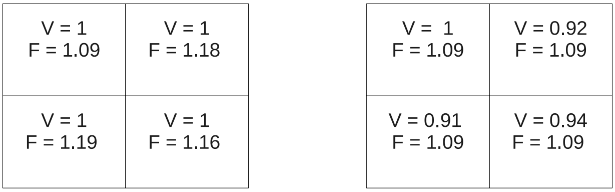
Figure 5.11: Schematic of the proposed algorithm.

two main effects. The first one is that the average buffer utilization is higher in a NoC affected by process variation for the same amount of traffic injected in the network, as shown in Figure 5.8(c), where buffer utilization values range from 0.3 to 0.7. The second effect is that despite the traffic simulated was uniform, the usual bottlenecks of the 2D mesh NoC using XY routing (shown in Figure 5.8(b) are masked by a more chaotic behavior produced by the effect of systematic variation correlation, as can be seen in Figure 5.8(c).

On the other hand, when parallel workloads are running in the CMP, process variation can cause an inefficient utilization of CMP cores. The reason is that despite of having an initial balanced distribution workload among cores, faster cores will finish earlier than the slower ones and this will cause that actual utilization of CMP faster cores decreases. In this context, we propose to map threads taking into account variability information of cores in order to simultaneously be able of reducing application execution time and of achieving an energy-efficient mapping approach.

Variation-Aware Mapping Algorithm

Our variation-aware (*VA*) mapping proposal tries to map application threads to faster cores. Figure 5.11 represents the different steps performed by the proposed algorithm. Although Figure 5.11 shows the general algorithm, an example for an application requesting 4 cores in a 5x5 CMP is additionally depicted. To efficiently assign threads to cores according to its frequency, the operating system keeps a list with the frequency of all the nodes in the



(a) MFSV region (b) SFMV region

Figure 5.12: Example of 4-tile regions using the two possible configurations.

chip that are currently available. Cores in that list are ordered according to their frequency in order to perform a faster search. This helps to considerably reduce the complexity of the *VA* algorithm[1] Note that frequency information of cores is provided by chip manufacturers after chip test or, alternatively, it could be found out by using an appropriate test at boot up time. Whenever a new incoming application requires to be mapped to a given number of cores, the list is accessed to get its first element (the available core with the highest frequency). Once the fastest available core is selected, the next step is making sure that there exists a regular region for the required amount of cores that additionally contains the selected core. If not, then the next core in the list would be selected. When finally a core that allows to create a region of the required size is selected, then all the possible regions, if there is more than one region available, containing that core are computed in order to select the best one, which will be the region presenting the lowest frequency variation across the cores in the region. Figure 5.11 shows, in the central part, the four different regions, in the 5x5 CMP, that exist containing the fastest available core. Once the best region is selected, it is provided to the application, so that it begins its execution. Finally, the cores assigned to the application are removed from the list of available cores.

On the other hand, as current CMP architectures allow to perform a per-core voltage and frequency selection, several frequency/voltage configurations can be applied to the cores in the region selected by the *VA* mapping algorithm. In this work we consider two kinds of configurations to be the most suitable configurations to achieve performance and energy-efficiency:

[1] The complexity of the mapping algorithm could be masked if idle cores perform an speculative computation of next optimal mappings.

- Multiple Frequency Single Voltage (*MFSV*) Region

  In this configuration, cores and routers of the region work at their maximum achievable speed. The resulting region will be composed by a set of tiles of slightly different frequency using the same voltage. Figure 5.12(a) shows a possible region using this configuration.

- Single Frequency Multiple Voltage (*SFMV*) Region

  This configuration adjusts the frequency of all the tiles to the frequency of the slowest one. This makes possible to reduce the voltage of the tiles that were initially able to work at higher frequencies. The resulting region will be composed of a set of tiles working at the same frequency but using different voltages. Figure 5.12(b) shows a region using such a configuration, where faster tiles, in addition to reducing their frequency, reduce their voltage in order to be more energy-efficient.

In the rest of the chapter we will intensively use the terms *VA*, *MFSV*, and *SVMF*. *MFSV* and *SVMF* will refer to the configurations depicted in Figure 5.12. However, when the details of the actual frequency/voltage configuration of the cores in the region selected by the VA algorithm are not important, we will just use the term *VA*.

5.3.1 Evaluation Methodology

For the evaluation of our application mapping proposal, in order to find a more realistic scenario, we use a NoC-based CMP configuration different to the one presented in Section 4.2. The main differences between the new and the old configuration are tile dimensions and router design. The concrete details of the new configuration are explained here after.

We decided that our CMP chip should include 64 cores interconnected by an 8x8 bi-dimensional mesh, which is in line with current proposals from industry [40] [86] [102] and academia [17] [110]. Additionally, the CMP should follow the tiled approach [109], which is a widely accepted design option [40] [110] [86] [102]. Thus, the basic building block will include a general purpose CPU with its associated private L1 instruction and data cache banks, and a fragment of a distributed L2 shared cache. The tile will also include part of the coherency protocol directory, which is distributed among them. Finally, in order to provide connectivity among the different tiles, they will also include a router connecting them to the NoC.

Parameter	**Values**
Core	1GHz, in-order, single thread
L1 inst cache	32KB, 4-way
L1 data cache	32KB, 4-way
L2 cache	128KB, 8-way

Table 5.2: Tile configuration and parameters

Table 5.2 summarizes the characteristics of the tile used in this study. In order to include the 64 tiles while keeping die size below the area budget for current designs [54] [110], the CPU block should be relatively small. Thus, an in-order single-threaded 1GHz CPU was chosen, which would additionally help in not consuming too much power. On the other hand, each tile includes a 32KB L1 instruction cache and a 32KB L1 data cache. These sizes were selected according to similar studies [110] [109] and commercial products [102]. Finally, L2-cache size was chosen to be 8MB, which was the maximum size that kept the die area inside the area budget for current designs. Thus, each of the tiles included 128KB of L2 cache.

In order to interconnect the 64 tiles in the CMP chip, we used an 8x8 2D-mesh NoC design that was synthesized in a 45nm technology. The router implemented in that NoC uses wormhole switching and has five virtual channels. That number of virtual channels was selected so that support for the underlying coherency protocol is provided. When designing the network, link length needs to be known so that some of the electronical parameters of the network are properly set. This length depends on the exact size of the tiles, as shown in Figure 4.9. Thus, to avoid designing the whole tile, which is out of the scope of this chapter, we used the McPAT tool [51] to find out the area required by the tile when synthesized using a 45nm technology node. Table 5.3 shows a summary of the results provided by this tool. As can be seen, as the tile requires 5.77mm^2, links will be 2.4mm long if square tiles are assumed. On the other hand, total die size is slightly larger than 400mm^2, assuring that it is manufacturable [54].

Component	Area (mm^2)
Die	406.732
Tile	5.77
Core	3.2971
L2 per tile	1.5140
Directory per tile	0.9585

Table 5.3: Area requirements for the main components of the CMP

Building SFMV and MFSV regions

In order to implement the MFSV and SFMV regions described in the previous section, the CMP design must be able to individually set frequency and/or voltage for each core. Delay and supply voltage are directly related with the well known alpha power law [90]. As maximum operating frequency is inversely proportional to delay, the dependence of the operating frequency with the supply voltage (V) can be represented as:

$$f \propto \frac{(V - V_{th})^{\alpha}}{V} \tag{5.1}$$

where α is a technology dependent parameter, in our case 1.3, and V_{th} is the transistor threshold voltage.

Simulation Infrastructure

In order to characterize the performance of our core assignment algorithm, we have simulated the execution of the applications from the PARSEC benchmark suite [4] in our 8x8 CMP test bench. The benchmarks were built to run with 1, 2, 4, 8, and 16 threads. These benchmarks run on a home built simulator. On the top of the simulator a coherence layer models the cache coherence protocol. The protocol chosen to keep memory coherence is the one used in the SGI Origin Processor [50]. Note that the simulator has been enhanced with the capability of working with several frequencies. To do this, it is necessary to establish a minimum step to the temporal resolution. This is in fact a trade-off between simulation time and a frequency/time resolution. The chosen resolution was 1 MHz meaning that two tiles working at frequencies differing less than 1MHz were set to work at the same frequency. However,

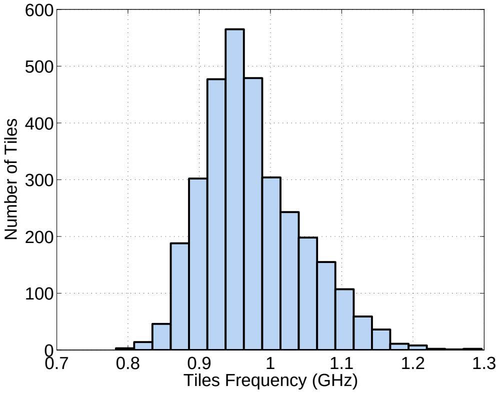


Figure 5.13: Distribution of tile frequency as a consequence of process variations.

this resolution is enough as the effects on power and performance of variations lower than 1MHz are imperceptible.

Additionally, to characterize the impact of process variations, 50 different instances of the 8x8 CMP test bench were considered for each execution. The difference from one instance to another is the different effect of process variation, as explained in next section.

Parameter Variations

Parameter variation has been accurately modeled following the model presented in Chapter 3. In particular, variability information has been obtained from the predictions of the ITRS [27]. More concretely, as our NoC design was implemented with a 45nm technology node, we have set V_{th}, L_{eff}, and m_t variations to $3\sigma_{V_{th}}/\mu = 40\%$, $3\sigma_{L_{eff}}/\mu = 12\%$, and $3\sigma_{m_t}/\mu = 10\%$, respectively. Additionally, according to the size of the designed CMP, spatial correlations have been generated with $\rho = 0.5$, meaning that L_{eff} values are correlated in a region equal to half of the chip size. Using these values of parameter variation with the model explained in [37], the 50 chips mentioned above were generated. Figure 5.13 shows the distribution of tile frequency for the 50 generated chips. As shown in the figure, as a consequence of process variations, tile frequency ranges from 0.78GHz to 1.3GHz.

Power and Energy Metrics

Tile power is estimated using McPAT [51]. Concretely, dynamic power is estimated with McPAT based on the statistics generated by our simulator. To estimate leakage power we start from the measurements of McPAT and, in order to introduce leakage power variations, we use the model proposed in [10]. Starting from this model, leakage power can be obtained at different values of V_{th}. Finally, energy measurements are directly derived from power measurements.

5.3.2 Evaluation

For the evaluation of our proposal, the performance and energy of applications is analyzed. Results of the proposed thread assignment policy are compared with a mapping strategy where the region that threads are mapped to is randomly assigned inside the CMP.

Applications Executed in Isolation

In this section, our *VA* mapping strategy proposal is first compared with a mapping strategy where the required region is randomly selected inside the CMP when applications are mapped in isolation, that is, the CMP is running one application at a time. This would represent a slightly loaded system. Note that in both mapping strategies, *VA* and *Random*, square-like shapes, if possible, are used to avoid the performance loss introduced by shapes with lower bisection bandwidth. More specifically, 2x2 and 4x4 regions are selected for 4-thread and 16-thread applications, while a 2x4 (or 4x2) region is selected for 8-thread applications. The proposed mapping strategy is evaluated using both multiple frequency single voltage regions (*MFSV*), and single frequency multiple voltage regions (*SFMV*).

Figure 5.14 shows application execution time results for 4-core applications. In this figure, the average speed-up (AVR), the best case speed-up (BC), and the worst case speed-up (WC) are represented. The *MFSV* mapping achieves an average execution time reduction of 10%. Additionally, *MFSV* presents an average speed-up of 11% and some applications present, in certain chips, reductions of the execution time around 23%. On the other hand, the *SFMV*

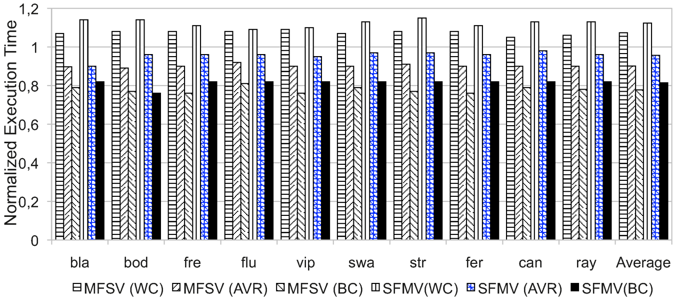


Figure 5.14: 4-core applications execution time relative to *Random*.

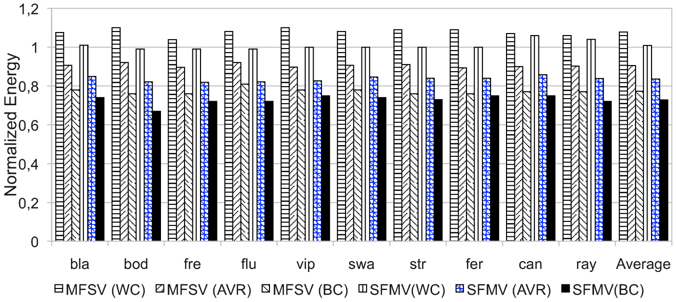


Figure 5.15: 4-core applications average energy relative to *Random*.

mechanism presents a 4% reduction of the average application execution time.

Application energy requirements results are shown in Figure 5.15. In this case, the *SFMV* mechanism achieves the best results. Concretely, *SFMV* reduces the average energy requirements of applications in a 16%. Additionally, the maximum energy reduction (BC) of this approach is 33% when running the *bodytrack* (bod) PARSEC benchmark. Note that although our proposals achieve considerably better results than the *Random* mapping, there exist certain unlikely situations where the *Random* mechanism is able to overcome the *VA* mapping. This is the result of a combination of factors that come into play, like the proximity of threads to memory controllers, differences in on-chip and off-chip memory bandwidth requirements [4], and differences in the thread-core assignment distribution inside the region. These factors, combined with the existence of regions providing similar raw throughput, make that *Random* mapping provides, in some cases, better results.

Figures 5.16 and 5.17 show the application execution time and the energy

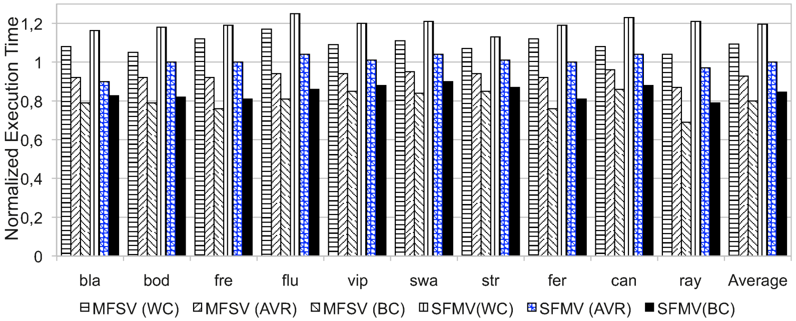


Figure 5.16: 8-core applications execution time relative to *Random*.

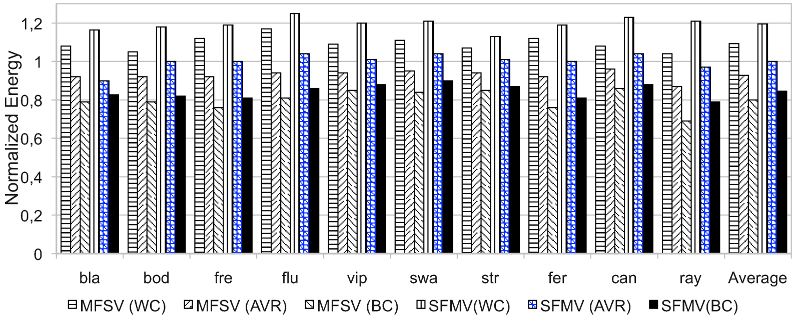
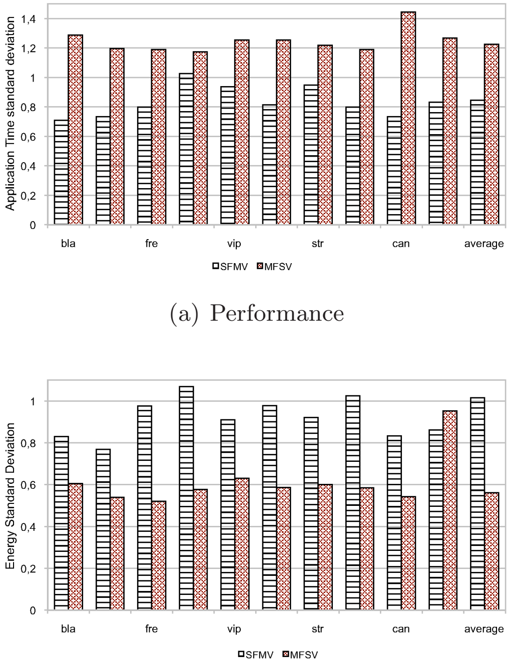


Figure 5.17: 8-core applications average energy relative to *Random*.

requirements for 8-core applications. As expected, the average benefits of the *VA* mapping are lower for applications running with more threads. Concretely, *MFSV* achieves an average reduction of applications execution time of 7%. This happens because bigger regions present lower frequency homogeneity. However, there still exist some chips presenting a reduction of the execution time of 24%. On the other hand, the *SFMV* mechanism achieves an average application execution time similar to the *Random* mapping. This result points out that when there are several components of the CMP working at different frequencies, all of them end running, in practice, at the frequency of the slowest one. In the case of the *MFSV* mapping, as frequencies are selected in a homogeneous way, the frequency of the slowest one is, in average, 7% higher than when the *Random* mapping is leveraged. Finally, the *SFMV* mechanism presents an average reduction of application energy of 13%, as a consequence of reducing the frequency of some of the tiles.



(a) Performance

(b) Energy

Figure 5.18: Standard deviation of applications execution time and energy relative to *Random*.

Increasing Performance and Energy predictability

Process variation causes tiles in a CMP to present different frequency and power values. These different tile features make the behavior of applications running on the CMP more difficult to predict. On one hand, the different operating frequencies presented by the NoC, cores, and memories among tiles cause the application execution time being difficult to estimate. This unpredictability may become critical for systems under real time constraints. On the other hand, the different power features of tiles may cause that a CMP exceeds a given power budget. Additionally, in systems making use of batteries an accurate estimation of energy requirements is mandatory. Note that unpredictability is caused by tile differences inside a particular chip and also across chips.

Figure 5.18 shows the results of the standard deviation of both, performance (Figure 5.18(a)) and power (Figure 5.18(b)), of the *MFSV* and *SFMV* mechanisms, when applications run on 4 cores. These results are relative to *Random*. Note, that low values of standard deviation mean high predictability. On the contrary, high standard deviation values correspond to systems where metrics are difficult to predict. As shown in Figure 5.18(a), the *MFSV* mechanism increases the predictability of the application execution time regardless the chosen application. Concretely, the *MFSV* achieves a 15% average improvement in the application execution time. However, the *SFMV* mechanism introduces a more unpredictable performance estimation as lowering the voltage of faster tiles is by itself another source of performance unpredictability. On the other hand, the *SFMV* mechanism is able to increase energy predictability a 44% in average. The *MFSV* presents only a 2% worse energy requirements predictability than *Random*. This is an important result as it shows that the best application execution time and time predictability can be achieved guaranteeing a given power or energy budget.

Impact of Memory Controller Location

In order to characterize the impact of the location of memory controllers in the performance of the variation-aware mapping, a 3D memory stacking scenario has been considered. In this scenario, all NoC routers are directly connected to memory controllers and, thus, the effect of different distances to memory controllers is canceled. Simulations of applications running on a 3D stacking architecture confirm that the location of memory controllers does not affect the average reduction of application execution time achieved by the *VA* mechanism, as shown in Figure 5.19.

Figure 5.19 shows a comparison of the best and worst results of the *MFSV* mechanism when using or not a 3D stacking architecture for 8-core applications. Results for 4-core and 16-core applications are similar. These results show that the 3D stacking architecture slightly reduces unpredictability of application execution time. Note that this is, in fact, an expected behavior as the proximity to memory controllers is, by itself, a source of unpredictability affecting to both the *Random* and the *VA* mapping strategies.

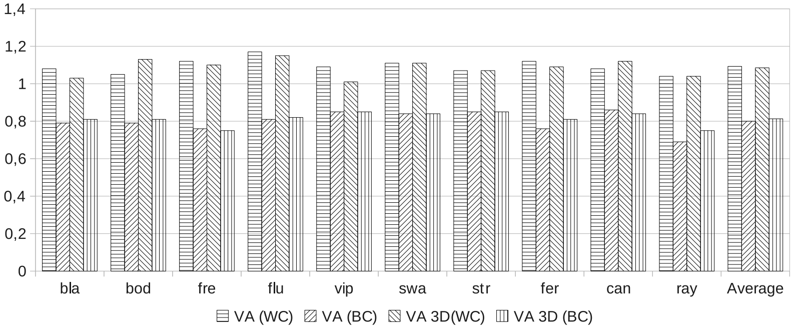


Figure 5.19: 8-core applications execution time relative to *Random* with 3D.

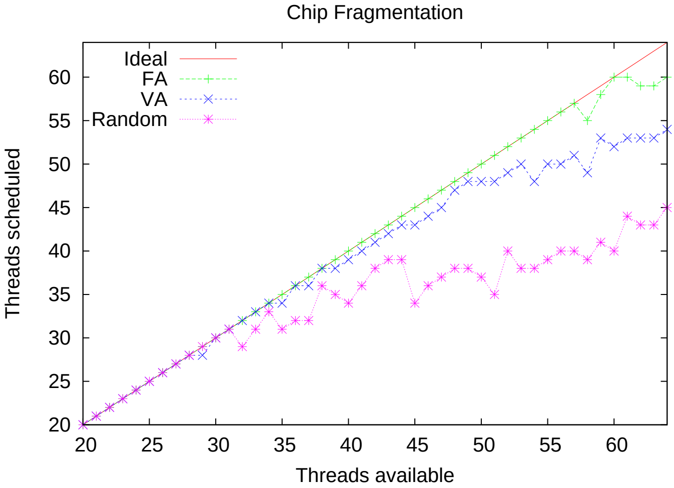
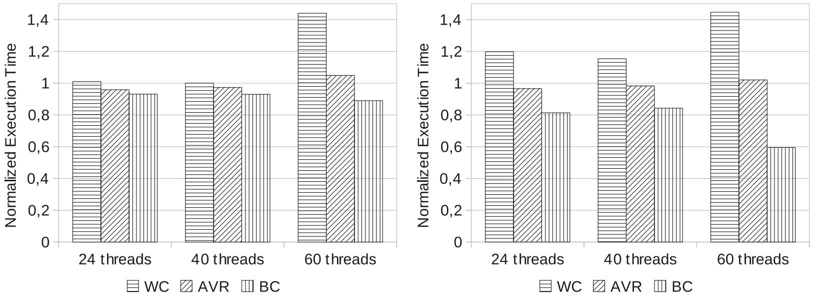


Figure 5.20: Chip utilization achieved with different mapping policies

Several Applications Concurrently Executed

When several applications run concurrently in a CMP, the use of a variation-aware mapping strategy may cause the fragmentation of the CMP. Additionally, note that mapping threads to regions with higher performance can make the mapping of incoming applications to other regions impossible. These two problems are not present in a simpler mapping algorithm that maps regions next to each other in order to maximize chip occupancy. Figure 5.20 shows the ability of different mapping strategies to map threads. The set of workloads used to characterize chip utilization are a combination of different PARSEC applications running with 1, 2, 4, 8, and 16 threads. The workloads have



(a) Entire workload

(b) Applications

Figure 5.21: Results of the *MFSV* mechanism relative to *FA*.

been built varying the total number of threads to be scheduled from 1 to 64. Applications were randomly chosen assuming that in a CMP running parallel workloads 30% of applications have 1, 2, or 4 threads, 60% of applications have 8 threads, and the remaining 10% of applications have 16 threads. Figure 5.20 shows a comparison of 3 different mapping strategies. The line labeled as "*VA*" represents the variation-aware mapping, the line labeled as "*Random*" represents the random mapping, and finally the line labeled as "*FA*" refers to a mapping strategy that tries to avoid the chip fragmentation. This fragmentation-aware mapping algorithm maps incoming applications next to the previously scheduled applications. Additionally, the line labeled as "*ideal*" represents the case of an ideal mapping algorithm that is able to map all incoming threads while there exist idle cores in the CMP[2]. As shown in this figure, the *Random* mapping is unable to efficiently map several applications in the chip for more than 35 threads. On the contrary, the "*VA*" mapping presents an ideal behavior below 48 threads, being able to map a maximum of 54 threads. Finally, the "*FA*" algorithm presents an ideal behavior for a number of threads lower than 55. These results show how variation-aware mapping does not introduce chip fragmentation for workloads below 48 threads. Furthermore, even for the *FA* mechanism, fragmentation cannot be avoided beyond 55 threads.

Figure 5.21 shows a performance comparison between the *VA* and the *FA* mechanisms when different sets of workloads are executed. More concretely,

[2] Note that in case that splitted regions are allowed, all algorithms are able to map threads until no idle cores exist in the CMP.

the figure shows application execution time for 24, 40, and 60-thread workloads, for the worst-case workload (WC), the best-case workload (BC), and the average results (AVR). Figure 5.21(a) shows the time required by the *MFSV* to execute the whole workload relative to the *FA* mechanism. As shown for the 24 and 40-threads workloads, the *MFSV* mechanism presents an average reduction of the whole workload execution time of 3% and 4%, respectively. Note that this improvement is not very important as the workload execution time is set by the last scheduled application which is assigned to a suboptimal region. On the contrary, for 60-thread workloads the workload execution time of *MFSV* is in average a 5% slower than the *FA* mechanism. This is caused by the fact that in some chips, due to the spatial features of process variation, all workload applications cannot be mapped concurrently in the CMP and then the average workload execution time is penalized by the applications waiting to be scheduled. On the other hand, Figure 5.21(b) shows the average execution time of individual applications across workloads. Note that the average execution time of individual applications is just increased by 1%. This result is specially important as it shows how the variation aware mapping presents a negligible penalization in the average application performance in the worst case scenario (a high number of threads need to be allocated).

5.4 Conclusions

In this chapter we have presented a variability-aware mapping algorithm that is intended to minimize the negative effects of process variation in CMPs at the same time that total energy used by applications is reduced. The proposal is based on assigning application idle cores in the CMP that additionally form a uniform region from the frequency point of view. Moreover, that region is composed of the fastest available idle cores at the time of the mapping. This variability-aware mapping has been presented in two different flavors: Multiple Frequency Single Voltage Region *MFSV* and Single Frequency Multiple Voltage Region *SFMV*. The difference among both options is the way cores in the selected region are set to the same frequency voltage whereas their operating frequency is the maximum achievable one after being affected by variability. In the second option, all the cores in the region are set to the same frequency, and therefore the voltage of the fastest ones can be reduced to adjust them to the lower frequencies, thus achieving energy efficiency.

Simulation results show the benefits of using our proposal. In the case of the *MFSV* mapping strategy, average execution time is reduced down-to 10%, while for some applications this reduction increases up to 23%. These improvements in execution additionally cause the reduction in total energy of up to 24%. In the case of *SFMV* approach, speed ups are smaller while energy savings increase up to a 33%, as this policy is focused on energy more than performance.

Chapter 6

Facing Permanent and Variation-Induced Timing Failures in NoC Links

Process variability makes silicon devices to become increasingly less predictable, forcing chip designers to use techniques to avoid losing performance and keeping yield. As mentioned before, the immediate technique used to guarantee the proper operation of the chip against variation-induced timing failures is reducing the clock frequency so that all the parts of the chip can properly work. Unfortunately, this low-cost technique is not useful as variability increases because of the large performance penalty. Additionally, according to the predictions of the ITRS [27], defect density levels also increase with technology scaling. Notice that NoC links are specially prone to manufacturing defects, as they are usually routed in the upper metalization layers and require a high number of vias to reach active devices located at the silicon surface [34]. Actually, the probability of having faulty links in a NoC might considerably increase in future CMP systems, expected to be implemented with 22nm technology by 2015 [74] due to the great variability caused by the much smaller transistor size, the increase of defect density levels, and the huge number of links present in the network.

For the reasons previously mentioned, new fault-tolerant link designs are required to deal with the presence of failures. In this chapter we propose a new technique to overcome the presence of failures in NoC links. The proposed mechanism, a variable phit size NoC architecture, is intended to face both manufacturing defects and variation-induced timing errors in regular 2D links. More precisely, we present a new mechanism that adapts link operation to the real conditions of the manufactured chip and therefore it is able to keep links working in the presence of variations. The benefits of such a technique are twofold. On one hand, such a variability-aware mechanism avoids chip performance to be significantly decreased. On the other hand, yield is maintained. Otherwise, an important fraction of the manufactured chips should be discarded.

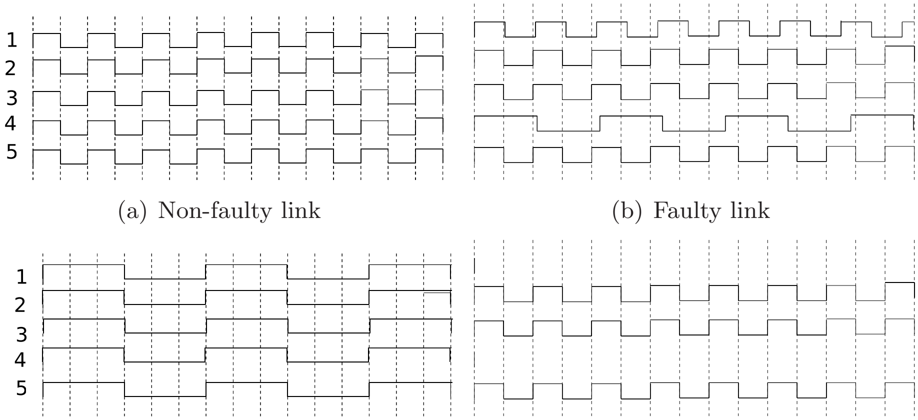
6.1 Related work

Recently, variability-aware design has arisen as one of the hot topics in computer architecture and high speed digital design. Consequently, the impact of variations in circuit performance has been thoroughly analyzed. Several works in the literature pointed out the importance of random process variations in NoC interconnects. In this sense, the implications of variations in NoC link interconnect are analyzed in [38] [103]. Concretely, in [38] different measures of delay uncertainty for technologies from 45nm to 16nm are provided. Similarly, the authors of [103] analyze delay variation of next technology nodes. Both studies remark the importance of process variation in communication links when technologies scale down.

There are several proposals that are able to tolerate the presence of faulty wires. Some of them are based on tolerating infrequent run-time timing violations, where delay failures are tolerated at the cost of performance because errors involve the information re-transmission [23] [65]. On the contrary, other kind of proposals focus on increasing interconnect yield by using redundant hardware. In this sense, the authors of [34] propose the use of spare wires to tolerate a bounded number of faults without decreasing communication performance.

In this chapter we propose a new fault-tolerant link design. The proposed design is based on a variable phit[1] size NoC architecture that is suitable for facing both variation-induced timing errors and defective wires in regular 2D

[1] The phit term is used in the interconnection networks area to define link width.



(a) Non-faulty link

(b) Faulty link

(c) After reducing frequency

(d) After discarding wires

Figure 6.1: Example of a link under variation-induced timing failures and two ways of keeping it working

NoC links. To do so, the phit size is efficiently adapted by using a variable phit-size network interface built with an omega network to perform the flit-bit shifting.

6.2 Problem Statement

As mentioned before, our goal is to deal with links where not all wires are fault-free. Wire failures can be caused by delay variations or by the presence of manufacturing defects. Delay uncertainty makes wires in a link to present a maximum achievable frequency lower than design frequency. An example of such a link is shown in Figure 6.1. Figure 6.1(a) shows a link composed of five wires working at the frequency targeted at design time, f_{clk}. This is a non-faulty link. Figure 6.1(b) shows a faulty link. In this link, wires 1 and 4 are not able to switch at the original f_{clk} frequency. Wire 1 is slightly slower than the design frequency (90% of f_{clk}, approximately) while wire 4 switches at less than half of the initial frequency (40% of f_{clk}).

The link in Figure 6.1(b) would usually be labeled as a faulty link at system initialization so that its use is precluded. If this link interconnects two switches of the NoC, a fault-tolerant routing algorithm [1] would be required in order to keep the network working. However, some bandwidth could still be retrieved from this faulty link if it is not discarded. In fact, if the original link shown in Figure 6.1(a) has an aggregated bandwidth $B_a = 5bw$ bps, where bw is the targeted bandwidth of each wire, the faulty link in Figure 6.1(b) is still able to deliver approximately $B_b = 3bw + 0.4bw + 0.9bw = 4.3bw$ bps representing 85% of the initial bandwidth[2]. In real NoC links composed of 128 or 256 wires, for example, having a few slower wires would mean that almost 100% of the bandwidth is still available. Thus, discarding the entire link means wasting bandwidth.

There are multiple possibilities to retrieve bandwidth from faulty links. When failures are caused by process variations the initial solution is basically reducing the frequency of the link to operate at the frequency of the slowest wire [108]. This option allows the link to be operational at the expense of a considerable reduction in performance. In fact, as the frequency of the whole link is reduced to match that of the slowest wire, the reduction in performance could be noticeable. In the case of the 5-wire link shown in Figure 6.1(c), the available bandwidth retrieved with this technique would be $B_c = 0.33(5bw) = 1.65bw$ bps. Note that this important reduction in bandwidth would also be present in real and wider links, because the whole link reduces its frequency independently of the number of wires[3] .

On the contrary, when link malfunction is caused by manufacturing defects, like shorts and opens, reducing the frequency of the link is not enough to keep the link working. In those cases, the immediate approach to tolerate failures is by precluding the use of faulty links. For that purpose, fault-tolerant routing mechanisms are required [1]. However, avoiding the use of some faulty links may considerably reduce bisection bandwidth causing an important reduction of network performance.

In this chapter we propose a different approach to retrieve the bandwidth still available in faulty 2D links. This technique is based on discarding the wires that are faulty regardless of the origin of failures. This idea is shown in

[2] $4.3bw$ bps is computed as follows: 3 non-faulty wires provide $3bw$ bps. Additionally, one of the faulty wires provides $0.9bw$ bps and the other faulty wire contributes with $0.4bw$ bps. Therefore, the aggregated bandwidth is $4.3bw$ bps.

[3] Actually, all the wires in the link, shown in Figure 6.1(c) should work at the slowest frequency, that is, at 0.4x, and therefore, the aggregated available bandwidth would be 0.4 $(5bw)= 2.0bw$ bps. However, in order to switch wires synchronously with the clock, the operational frequency should be reduced to one third.

Figure 6.1(d), where wires 1 and 4 are not used. In this case, link bandwidth is reduced to $B_d = 3bw$ bps, that corresponds to 60% of the original bandwidth. When real links with 128 or 256 wires are considered, between 90% and 95% of the initial bandwidth is still available. Obviously, in order to transmit information properly, flits must be suitably sliced and their bits sent across the non-faulty wires. Therefore, additional hardware is required at the transmitter to slice flits according to the available wires and to allow the transmission of bits belonging to two consecutive flits during the same clock cycle, if necessary. Also, additional hardware is required at the receiver to retrieve the original flits. This technique is called *Phit Reduction*. We will refer to it as *PR*.

6.3 Variable Phit Size NoC Architecture

As mentioned before, the wires of a link can be faulty mainly as a consequence of two independent reasons: *permanent faults*, caused by defects affecting wires or vias, and *timing violations* as a consequence of the increasing process variation. In this section a NoC architecture with variable phit size is presented. This NoC architecture is able to tolerate the presence of faulty wires by performing a variable phit size flit injection. In the variable phit size NoC architecture proposed, link failures can be tolerated for an unbounded number of faulty wires.

In order to implement this mechanism, two possibilities arise. The first one, hereafter named *Local Phit Reduction* (LPR), is oriented to fabrication processes with very high variability or a high density of manufacturing defects, because, in that case the differences in the number of faulty wires among different links are high. This approach requires the inclusion of complex modules in every router port. The included port modules that enable the use of the LPR approach must be able to slice flits and to transmit them across the non-faulty wires, and later reconstruct them. Thus, this mechanism is costly in hardware but provides good performance because transmission across each link is performed using the maximum available link bandwidth.

The other way of implementing this technique is named *Global Phit Reduction* (GPR). When differences in the expected number of faulty wires across the links are low, for example when the expected defect density is low, the

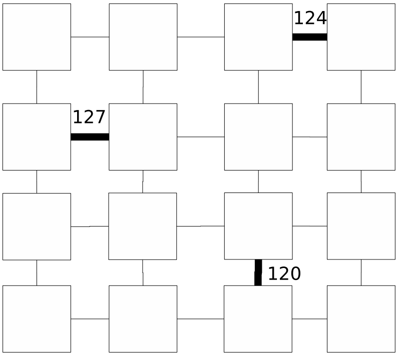
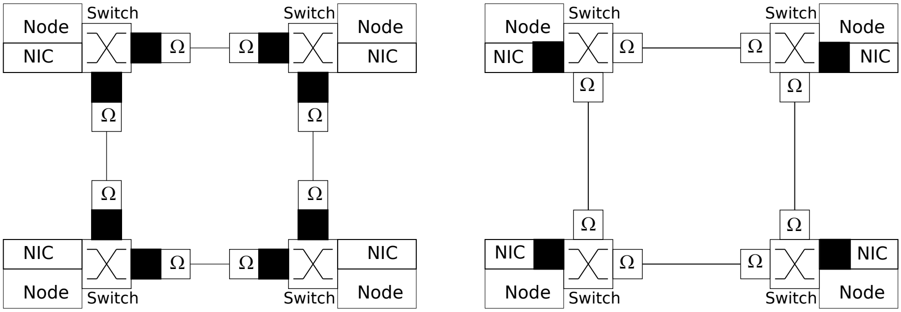


Figure 6.2: Mesh with faulty links. Numbers next to the links show the amount of non-faulty wires they have. Non-labeled links have no faulty wires

inclusion of a hardware module to slice and rebuild flits at every port of the network wastes, unnecessarily, silicon area. In this case, reducing the phit size for the whole network is a better solution. In order to perform this reduction, it is necessary to identify the link with more faulty wires. To do so, an initialization algorithm is run across the network to find such a link. Once this link is found, it will bound the phit size for the whole network to the number of its non-faulty wires. This is accomplished by configuring all the network interfaces so that they adjust transmission to that phit size. Consequently, the number of hardware modules needed is considerably reduced as they are only required at the elements able of injecting and extracting traffic to and from the network. Figure 6.2 shows an example of a 4x4 128-bit wide mesh network with 3 faulty links. In this case, the initialization mechanism would fix the phit size to 120. Thereafter, all transmissions are performed in a 120-bit phit basis. Only network interfaces sending packets have to slice flits, and only end receivers have to rebuild flits, considerably saving silicon resources with respect to the LPR approach because the hardware modules located at the external links are quite simple. These modules just have to allow flit bits to use non-faulty wires. In the following sections the concrete details of the fault-tolerant NoC architecture are presented.

The variable phit size NoC architecture proposed in this chapter consists on the use of receiver and transmitter modules to deal with flits with a variable phit size. As mentioned in the previous section, this approach is presented in



(a) *LPR*

(b) *GPR*

Figure 6.3: Diagram of the proposed variable phit size NoC architecture

two different flavours: *LPR* when a different phit size is used in each link individually, and *GPR* when the phit of the whole network or a given region is reduced to the same value. In this last approach phit size is set by the link presenting more faulty wires. When the *LPR* approach is applied, receiver and transmitter modules are required at both link ends. On the contrary, when using the *GPR* approach, these modules are located only at the network interfaces. In both cases, the use of a variable phit size module allows a transparent communication in the network with regular cores and memories. Figure 6.3 shows the arrangement of the required modules for applying both the LPR (Fig. 6.3(a)) and the GPR (Fig. 6.3(b)) mechanisms in a 4-node network. In Figure 6.3, black squares represent both transmitter and receiver modules able to inject flits with a variable phit size. Thus including the logic required to slice and rebuild flits. On the other hand, boxes labeled as Ω represent NxN 1-bit wide omega crossbars only. Figure 6.3(a) shows that when using the *LPR* approach, the complex hardware modules are required at every external output/input port. On the contrary, when using the *GPR* approach (Fig 6.3(b)) the complex hardware transmitter and receiver modules are required only at network interfaces (NIC) as all transmissions are based on a given phit size used in the whole network (or region)[4]. Finally, to select a subset of non-faulty wires among the total wires of a link, Ω link-crossbars are placed in between the router data path and the physical link in both output

[4] Note that regions with different phit size can be built using the proposed architecture in order to improve the behaviour of the design in the presence of highly spatially correlated variations.

and input links. Further details of the NxN 1-bit wide omega crossbar required by both the LPR and the GPR mechanisms are given in Section 6.3.4.

In the following subsections the implementation details of the modules required to build the proposed NoC architecture, the omega crossbars, the transmitter, and the receiver, are shown. Regardless of the way of applying the PR mechanism, modules able to perform a variable phit size flit injection are required. The only difference is that in the case of the LPR approach these modules are required in every router link, whereas when using the GPR approach the hardware modules to adapt the phit size are only required at network interfaces.

6.3.1 Omega Crossbar Design

As told before, for an efficient implementation of the PR mechanism we use the properties of the omega network. Concretely, we take advantage of two features of the omega network. The first one is the reconfiguration capability of this network. This feature is used in the PR mechanism to efficiently split faulty and non-faulty wires. The latter property of the omega network we used is the bit-shifting capability. This property will be used to efficiently perform a variable phit size flit injection.

The omega (Ω) and the inverse omega (Ω^{-1}) networks are examples of the classic multistage networks that use the perfect shuffle interconnection between the switching stages. Figure 6.4 shows an Ω^{-1} network connecting 8 inputs to 8 outputs (N=8). For N input/outputs, an Inverse Omega network contains N/2 (2x2) switches at each stage, and log_2N stages. Figure 6.5 shows an schematic of a 2x2 switch using transmission gates as cross-points[5]

Configuring the Omega Network

For the configuration of the omega network we use its routing properties. An omega network can be successfully routed following the XOR-routing mechanism [21]. This mechanism is explained in Equation 6.1. In this equation $Source_i$ and $Destination_i$ represent the i-bit of the binary representation of

[5] In order to avoid the signal degradation caused by simple pass transistors transmission gates are employed. For a high number of cascaded transistors the use of signal repeaters may be required.

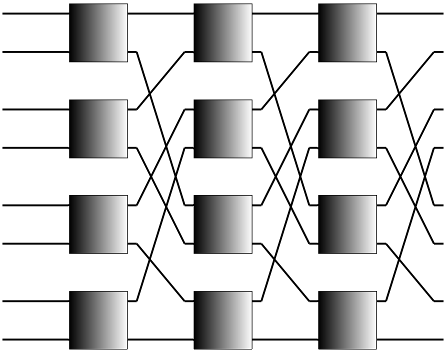


Figure 6.4: 8x8 Inverse omega network

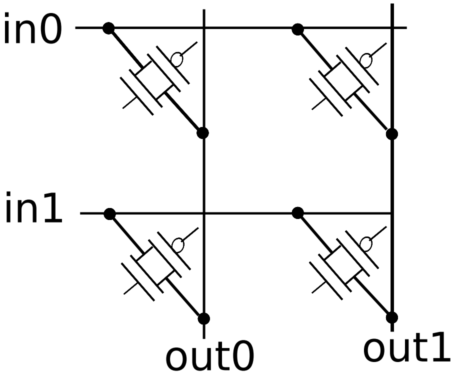


Figure 6.5: 2x2 matrix with pass transistors

source and destination, whereas the C_i value is the i-bit of the configuration matrix which is set to pass or cross (1 sets the switch to cross and 0 to pass). The values of the 2x2 switches of the Ω^{-1} are computed in the same way but the bits are read in the opposite direction. For example in a reverse 8x8 omega network, like the one shown in Figure 6.4, if a connection between the second input (001) and the third output (010) needs to be established, the switches that traverse the network are set to 011 (cross-cross-pass). This is the result of the expression $(001) \oplus (010) = 011$.

$$C_i = Source_i \oplus Destination_i \tag{6.1}$$

Non-blocking permutations with the Omega Network

Thanks to the reconfiguration capabilities of the Ω network, the fault-tolerant NoC design covers all the possible NoC link failure patterns and for an unbounded number of faulty wires. However, when a high number of wires in a link are faulty, performance is severely degraded. Nevertheless, previous proposals where not able to keep the link working in these circumstances while our proposal does [34]. To understand how all the failure patterns can be tolerated with our proposal we should analyze the non-blocking permutations of the Ω^{-1} network. Note that properties that apply to the Ω^{-1} are also satisfied by the Ω network in the opposite direction. According to [53], a permutation π is passable by the Ω^{-1} network if and only if for all pairs

$$x \rightarrow \pi(x), y \rightarrow \pi(y) \tag{6.2}$$

in the permutation,

$$M(x, y) + L(\pi(x), \pi(y)) < n \tag{6.3}$$

where $n = log_2 N$. Note that functions $M(x, y)$ and $L(x, y)$ are defined as the number of most significant bits and least significant bits, respectively, which agree in the binary expansions of x, y. The non-blocking properties of the Ω^{-1} are thoroughly analyzed in [12]. In this study, it is demonstrated that Ω^{-1} is non-blocking for a set of monotonic inputs with concentrated output destinations. Note that this property matches the sorting requirements the phit reduction mechanism.

6.3.2 Variable Phit Size Flit Injection

Links presenting some faulty wires have an effective phit size lower than the nominal one[6]. In this regard, to enable the injection of flits across faulty links, we have designed a modified injection queue (FIFO) where originally sized flits (Nbits) are suitably sliced and injected into links with a given phit size. Figure 6.6 shows a simplified scheme of the variable phit size transmitter module. The proposed circuitry makes use of a 2N-bit wide omega network to

[6] In this study in concordance with the major part of NoC designs proposed, phit size and flit size are equal.

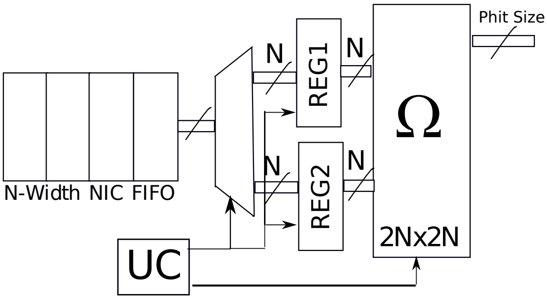


Figure 6.6: Schematic of the modified NIC injection queue

perform the cyclic shifting of flit bits. Additional hardware resources required by the variable flits size NIC injector are a 2-to-1 N-bit wide mux, the unit control (UC), the flow control mechanism[7], and the configuration information of the 2Nx2N omega network. The unit control (UC) is composed of a 2N counter and the combinational logic to configure the 2x2 switches of the Ω network. The counter is increased every cycle by m, being m the value of the new phit size (the number of non-faulty wires). The value of this counter defines the bit-shifting required to properly forward the bits stored in the register to the output.

Figure 6.7 shows how the proposed flit slicing mechanism works. The example in Figure 6.7 shows a 4-bit width link where due to the presence of failures, the flit size has been re-adjusted to 3. In this figure it is represented the stored flit bits in the registers directly connected with the 2N omega network (REG1 and REG2 shown in Figure 6.6), the required cyclic shift and the transmitted bit flits. In the example of the figure, when the transmission starts (t=0), an entire flit is stored in REG1. In that case, the omega network does not shift the bits and the 3 first bits of the flit are directly forwarded to the output. In the next cycle (t=1) the following bits to be transmitted are the 4^{th} of the previous flit and the following 2 bits of the second flit stored in REG2. Those bits are forwarded to the output by performing a 3-bit shift with the omega network. This is how the transmitter works. However, when enough flits are accumulated in the registers, a flow-control signal stalls forwarding a new flit into the REG1 or REG2 registers in order to avoid the

[7] A flow control mechanism is required as the original flit size is equal or larger than the number of non-faulty link wires.

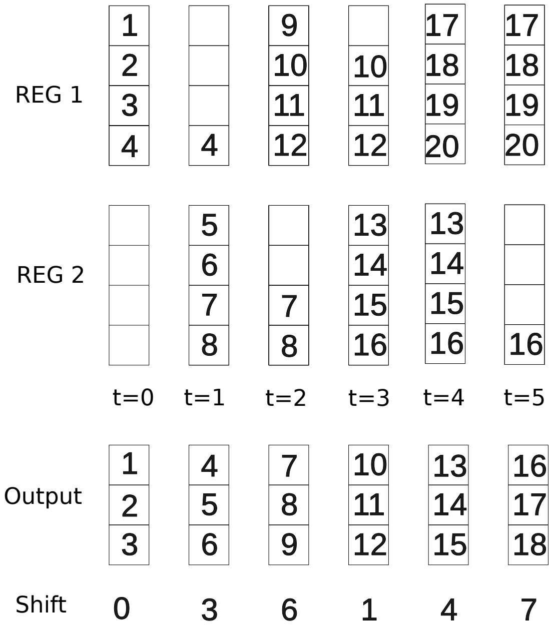


Figure 6.7: Example of splitted flit transmission

overflow of these registers REG1 and REG2. This is shown in Figure 6.6 at t=4 where a stall signal avoids the injection of a new flit in the next cycle.

The proposed variable phit size flit injection link design can be applied to any NoC architecture in any of the two approaches of the mechanism, and allows full compatibility with the rest of network modules. In this regard, different IP modules of previously designed non fault-tolerant NoC designs can be re-used. For applying the LPR approach the only requirement, in order to reduce the overheads of extra buffering resources, is the use of an output queuing switch architecture. In that case, we only need to replace the regular output buffering resources by the proposed variable flit size NIC injector. Thanks to that, the area overhead of the proposal is minimized. In the case of using a switch without output buffering, output buffering costs should be taken into account to compute the area overhead. On the other hand, using the GPR approach variable phit size flit injectors are required only at network interfaces. In that case the the regular NIC queue is replaced by the modified variable phit size NIC injection queue.

6.3.3 Variable Phit Size Flit Ejection

In the same way that a transmitter module is required to perform a variable phit size flit injection, a receiver module is required for receiving flits of a variable phit size. In the proposed variable phit size NoC architecture flits arrive at input ports or network interfaces with a size equal or smaller than the at-design phase flit size. The receiver module is required for reconstructing flits in its original form. The circuitry required for that purpose is analogous to the one shown in Figure 6.6 that represented the transmitter module.

6.3.4 Link Crossbar Design

As told before, in the PR mechanism, for both the LPR and the GPR approach, a crossbar is required in order to group the faulty wires of a link and isolate them. The link crossbar design is in fact a NxN 1-bit wide omega network. This crossbar is placed in between the router data path and the physical link in both output and input links. For performing the faulty wires grouping, faulty wires are identified at the testing stage. Once the proper configuration information of link-crossbars is elaborated, it is stored in ROM memories. Those wires that were identified as faulty on initialization time are discarded. For an efficient implementation of the previously mentioned link crossbar, we use an omega network.

Once the faulty wires of a link are identified the omega network placed in between the router and the link has to be configured. This configuration is performed once at initialization time. Note, that for configuring the link crossbar we use the routing properties of the omega network as explained in Section 6.3.1. The initialization algorithm computes the values of all 2x2 switches (pass or cross). As transistors are permanently set to pass or cross, the delay introduced by the omega network is minimum and is only due to the parasitic on resistance of pass transistors.

Figure 6.8 shows an example of the grouping capabilities of the omega network. In the example of the figure, a 4-bit width link with one faulty wire is shown. The omega network located in between the link and the router data path groups faulty and non-faulty wires, in order to be able to isolate faulty wires. The manner in which these switches are set determines the way faulty

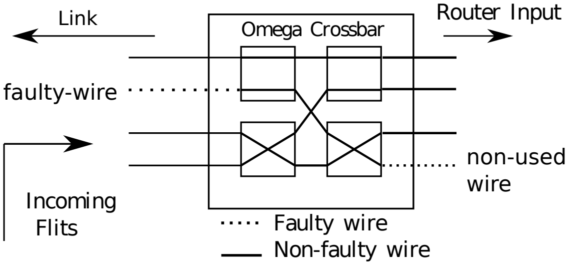


Figure 6.8: Example of how non-faulty wires are grouped by the omega network

and non-faulty wires are grouped.

6.4 Evaluation

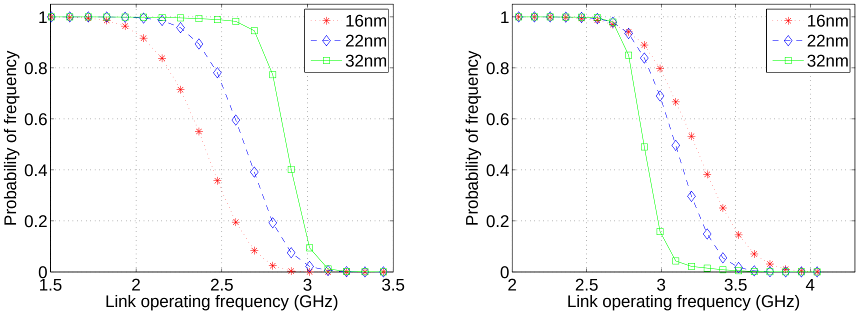
In this section the variable phit-size NoC architecture is evaluated. First, we focus on the details of 2D NoC link designs. Later, performance and area overhead of the proposal are analyzed.

6.4.1 NoC Link Model

Repeater insertion is an efficient method to reduce interconnect delay and signal transition times. However, as told in Chapter 4, in order to design links with the minimum possible power dissipation a different approach is required. For that purpose, we follow the reasoning given in [14]. Concretely, links are designed using the minimum device size that allows to reach a given target frequency. In this study we consider two different link designs. The first link, the High-Speed link (HP), is designed for working at 3GHz. The second link, the Low-Power link (LP), is designed for achieve a working frequency of 2GHz. In both cases we consider a link length of 2.4mm, according to the CMP floorplan designed in Chapter 5. As this floorplan refers to a 45nm CMP implementation, link length has been scaled for the rest of technologies considered in this study. Table 6.1 summarizes link configuration for both the HP and the LP scenario. Remember that as explained in Section 4.1.1, a given link configuration is determined by the number of sections (k) and the size of its repeaters (h).

Technology	Length(mm)	High Speed			Low Speed		
		delay(ns)	k	h	delay(ns)	k	h
32nm	1.71	0.306	5	18	0.48	3	10
22nm	1.17	0.328	5	16	0.45	3	10
16nm	0.85	0.33	5	16	0.44	3	10

Table 6.1: Link configurations used in the study



(a) Low-Power (b) High-Performance

Figure 6.9: Impact of variation in wire delay for both high-performance and low-power links.

6.4.2 Injecting Process Variations into NoC Links

Before presenting a performance comparison of our proposal it is necessary to present how process variation affects links. In order to measure the performance of the proposed PR mechanism against process variations, we have injected parameter variations in all link wires of 100 chips instances using the framework presented in Chapter 3. For the placement of links we used the 64-core floorplan presented in Section 5.3.1. Links considered for this study are 128-bit wide. The variability sources injected are transistor effective channel length variation ($\sigma_{L_{eff}}$), random threshold voltage variations ($\sigma_{V_{th}}$), and metal thickness variation (σ_{m_t}). It is important to remark that the severity of variations applied to the designed links is the one provided in the ITRS report as expected for the technologies considered in this study. The values of the different variability sources can be found in Table 3.2.

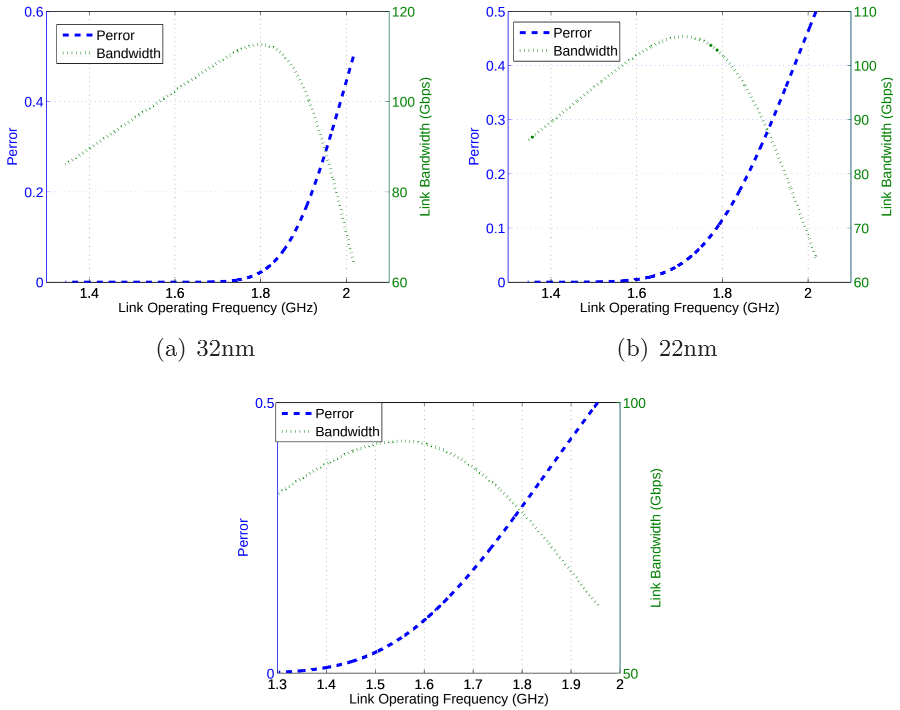
Figure 6.9 shows the impact of parameter variations in both the high-performance and low-power NoC links. Concretely, this figure shows the cumulative distribution function of the maximum achievable link frequency for 32nm, 22nm, and 16nm. As shown in the figure, as technologies scale down the impact of parameter variations is more noticeable. Note that lower slope means higher variability. The reason behind this is that, as stated in Chapter 3, random dopant fluctuations are dominating over the other components of variations for technologies below 32nm, and its contribution considerably increases as the size of transistors is reduced. Figure 6.9(a) shows that in the *LP* link as a consequence of variations, maximum achievable wire frequency ranges from 1.5GHz to 3.1GHz (for a 16nm technology). In the case of HP links (Figure 6.9(b)), maximum wire achievable frequency varies between 2.5GHz and 4.1GHz (for a 16nm technology). The measured variation (σ/μ) for a 16nm technology of the maximum achievable frequency is 16.4% and 11.1%, for LP and HP links respectively. LP links suffer from higher variability as the required device size is lower. The effect of device size in delay variation was analyzed in Chapter 3.

6.4.3 Performance Evaluation

In this section we compare the performance of the phit reduction (*PR*) approach with the traditional frequency reduction (*FR*) approach[8]. Note, that in order to keep all the link wires working, in a variable frequency NoC link design (*FR*), the frequency of the link must be lowered in order to switch at the frequency of the slowest wire. On the contrary, when our proposal (*PR*) is leveraged, higher frequencies can be used. For that purpose we have to tolerate the presence of faulty wires. In this sense, the variable phit size NoC architecture allows the use of higher frequencies at the expense of discarding some slower wires. As a result the bandwidth of the link can be maximized by finding the optimal trade-off between link operating frequency and the number of faulty wires.

Figures 6.10 and 6.11 elaborate on the mentioned trade-off. These figures show the link error probability for 32nm, 22nm, and 16nm technologies, for both the HP and LP links, respectively. In these figures link error probability is computed using Equation 3.11. Additionally, Figures 6.10 and 6.11 also

[8] Note that as we are evaluating link performance the *PR* mechanism applies to both the *LPR* and the *GPR* approach.



(a) 32nm

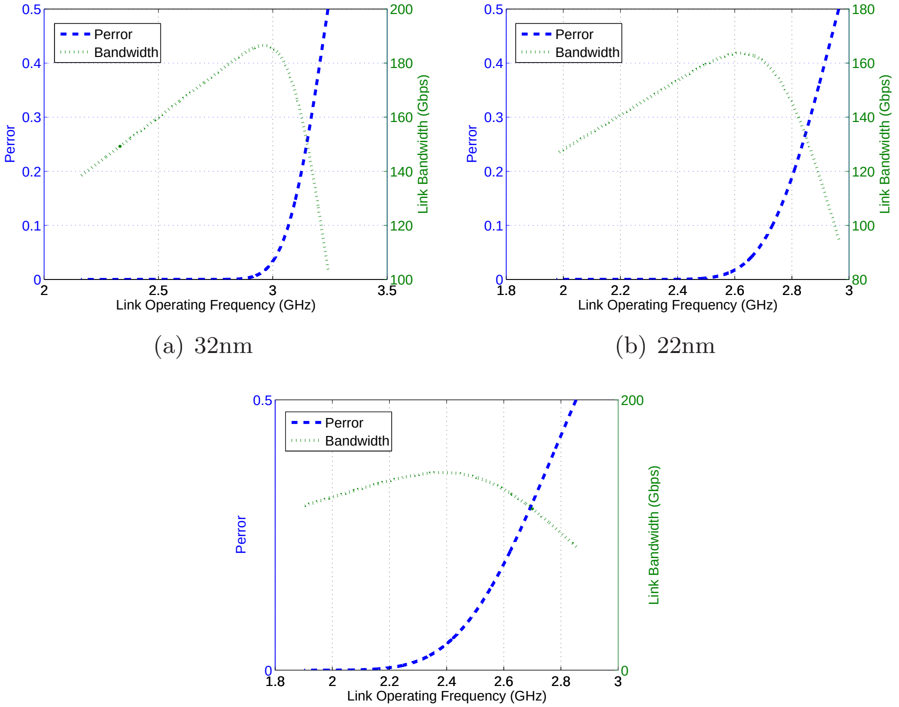
(b) 22nm

(c) 16nm

Figure 6.10: Link error probability and maximum achievable bandwidth for Low-Power links.

show link bandwidth when applying the PR mechanism at different frequencies. For a 32nm technology, the LP link achieves a maximum bandwidth of 112.7Gbps (Figure 6.10(a)), while the bandwidth of the HP is 186.6 Gbps (Figure 6.11(a)). On the contrary, for a 32nm technology the FR mechanism achieves is 86.0333 Gbps for the LP link, and 138.3 Gbps for the HP link. Notice that the nominal bandwidth, the bandwidth of a link where all wires work at nominal frequency, is 143.8 Gbps and 209.2Gbps, for the LP and HP designs, respectively.

Tables 6.3 and 6.2 show link bandwidth degradation caused by delay variation for the LP and HP links, respectively. Concretely, these tables compare the performance of PR and FR mechanisms in the presence of process variations. As shown in this table, using the PR mechanism, the impact of process variation in NoC links is reduced. Concretely, in the worst case scenario, 16nm



(a) 32nm

(b) 22nm

(c) 16nm

Figure 6.11: Link error probability and maximum achievable bandwidth for High-Performance links.

technology and the LP link, using the PR approach the bandwidth is still 92.9 Gbps, a 65% of the nominal bandwidth. This bandwidth is an 11.1% higher than the one achieved by the FR approach.

Results confirm that a higher link bandwidth can be retrieved from links in the presence of variation using the PR approach. This mechanism allows to work at higher frequencies by tolerating the presence of faulty wires. Furthermore, using the PR mechanism the presence of manufacturing defects is also tolerated, and thus, the NoC yield is improved.

6.4.4 Area Results

In this section the area overhead of the proposed variable phit-size NoC architecture is computed. The area results shown in this section correspond to the implementation of the variable flit NoC architecture in a 45nm technology.

	32nm		22nm		16nm	
	PR	FR	PR	FR	PR	FR
BW(Gbps)	186.5609	138.3107	163.6772	126.4592	146.1798	121.7358
Freq(GHz)	2.9618	2.1611	2.6213	1.9759	2.3864	1.9021

Table 6.2: Bandwidth and operating frequency for the HP link using the PR and FR mechanisms.

	32nm		22nm		16nm	
	PR	FR	PR	FR	PR	FR
BW(Gbps)	112.6759	86.0333	105.3603	86.1243	92.8926	83.3893
Freq(GHz)	1.7997	1.3443	1.7112	1.3457	1.5526	1.3030

Table 6.3: Bandwidth and operating frequency for the LP link using the PR and FR mechanisms.

The area overheads are due to the inclusion of the variable phit size flit injector and ejectors modules required, by both the LPR and GPR approaches, and by the required omega crossbars at every input and output external port of the NoC switches. Note that for an efficient omega network implementation a full custom design is required. Therefore, area costs of the omega network has been computed according to its transistor count which is $8 * \frac{N}{2} * \log_2(N)$. Table 6.4 shows the area overheads of the fault-tolerant architecture proposed for a 64-bit NoC implementation. In this table the area required by a 64-bit five port router architecture is also shown. The details of this router architecture are given in Appendix A. The LPR mechanism requires an area that is 18% the area of a regular five port NoC switch. However, when using the GPR approach the area overhead of the proposal is reduced. Concretely, this latter approach increases switch area only by 9%.

As shown, the variable phit size NoC architecture proposed requires non-

	Area (um^2)	Overhead
Switch	28356.72	-
LPR	5107	18%
GPR	2655.72	9.37%

Table 6.4: Area overhead of the fault-tolerant architecture for a 64-bit flit NoC.

Interconnect width	Spare(99%)	Spare(99.9%)	Omega Crossbar
32	288	708	640
64	548	2180	1536
128	5312	10128	3584

Table 6.5: Hardware cost comparison

negligible area resources, however, similar proposals that tolerate manufacturing faults are also costly in terms of area resources. In this sense, Table 6.5 shows a complexity comparison of our proposal and the use of spare wires [34]. This table shows the required cross-points to implement both approaches. Note that in this table the overhead of the variable flit size NoC architecture in NICs is not computed. The results for the spare crossbar represent the cross-point requirements for achieving a link yield of 99% and 99.9% given a wire error probability of 0.01. On the contrary, the cross-points required for implementing the PR approach are fixed regardless the number of faulty wires, and, therefore, achieving a 100% link yield. Results of this table show how for wide links the use of an omega link crossbar requires less cross-points than the use of an spare crossbar [34]. Consequently, for a 128-bit wide link the spare crossbar requires 1.48× and 2.83× the number of cross-points of an omega link crossbar for achieving only 99% and 99.9% yield, respectively.

6.5 Conclusions

In this chapter a new fault-tolerant NoC architecture is presented. The proposed architecture is based on variable phit-size injection and ejection of flits that allows the use of links presenting some faulty-wires. This variable phit-size NoC architecture is presented in two flavours: LPR and GPR. The LPR approach causes an important overhead of NoC area and is oriented for such manufacturing scenarios with high defect density levels and dominated by a strong systematic L_{eff} variation. On the contrary, the GPR reduces considerably the area overhead and is suitable for scenarios with low defect density and high random threshold variations.

The main advantages of the proposed variable phit size NoC architecture are the ability to tolerate both manufacturing defects and variation induced timing errors. On one hand, this approach achieves better performance than other variation tolerant link design approaches based on reducing the link frequency. Results shown in the evaluation section confirm that for 32nm, 22nm, and 16nm technologies reducing the phit size retrieves more bandwidth from faulty links than reducing the frequency. Concretely, we achieve a 31% and 34.4% bandwidth improvement ratio over frequency reduction techniques for LP and HP links, respectively.

On the other hand, the PR mechanism also outperforms other proposals oriented to face the presence of faulty wires in NoC links, as is able to maximize yield at a reasonable area costs and regardless the amount and the origin of failures. Results shown that for 128-wide links our proposal requires less area than the use of spare wires, while the yield achieved is also superior.

Chapter 7

Facing Defective TSVs in 3D NoC Links

As mentioned before, as the number of cores increases, memory bandwidth is predicted to become a major limiting factor for CMP and MPSoC designs. In this context, three dimensional Integrated Circuits (3DICs) arise as a promising technological solution to partially alleviate the problems caused by the pin-out limitations [81]. In 3DICs, dies are stacked on top of each other and vertical connections are established among them. The most adopted solution technology to enable vertical links between dies is the use of Through Silicon Vias (TSVs). However, TSV-based links present two main drawbacks. One is the high defect rate of TSVs. The other one is TSV footprint area. In this context, 3D designs demand for efficient mechanisms able to overcome both yield and area overheads of regular vertical link designs

The high defect rate of TSVs demands for efficient mechanisms and designs in order to overcome the yield reduction caused by TSV manufacturing problems in 3DICs. Currently, most of the previous work on fault tolerant 3D architectures rely on the use of redundant TSVs [55, 78], where a few extra TSVs are added to each vertical link. However, minimizing the number of extra TSVs (while keeping system performance and yield high) is specially appealing and needed because, as the number of nodes (cores or memories) continues increasing, the impact of redundant TSVs on area footprint becomes impossible to neglect [79] [46] [78]. In particular, in [79], TSV footprint area of future 3DICs is predicted to be similar to the area of one computing core.

In this chapter we propose a new fault-tolerant vertical TSV-based link design able to noticeably increase the yield of 3DIC chips while keeping performance and minimizing the amount of required resources. Actually, our new proposal significantly reduces the amount of TSVs per link without reducing performance. In comparison with a conventional N-wide vertical link, our proposal requires $N/2+m$ TSVs, where m is significantly smaller than $N/2$ (m is usually one or two TSVs). Thus, deploying our proposal reports noticeable area savings.

To reach high values of yield and to reduce the number of TSVs required, our proposal exploits the speed of vertical TSV links. Our design is based on adding an Omega network at both ends of the vertical link (at the switch boundaries). This network allows a programmed configuration for efficient use of fault-free TSVs. In our proposal, for each vertical link of width $N/2+m$, a subset of $N/2$ functional and fault-free TSVs is identified. These $N/2$ fault-free TSVs are used at double the clock rate to transmit one flit per cycle in a fault-free manner by means of a proper configuration of the Omega network. In our proposal, the operating frequency of the rest of the design is not affected, because both edges of the clock signals are used in the TSV in accordance to the switch design. In this way, the higher speed of vertical TSV links is exploited to first reduce its width and, thus, area footprint, and second to provide fault-tolerance at minimum cost – the number of TSVs is reduced from N down to $N/2+m$. Thus, the net result is a fault-tolerant TSV-based link design that dramatically improves yield (near 100%) while reducing the overall area requirements (45% savings for 64-bit links, including the overhead of the two Omega networks) and keeping the same system performance. The maximum frequency achieved for the fault-tolerant link is 3.47 GHz when using a 45nm technology. For those cases where this frequency is not enough, we additionally propose a co-design of the switch and TSV-based link that enables to increase the TSV-based link frequency up to 4.72 GHz, which should be enough high for most designs.

7.1 Related Work

Current and future fabrication processes enable the use of smaller feature sizes and new manufacturing technologies, as 3D stacking. Unfortunately, this poses an increase in reliability uncertainty in many different areas, being one of them related to network links. In this regard, several recent works in the literature have pointed out some of the problems affecting chip interconnects. For example, the implications of variability in NoC link interconnect are analyzed in [38] [103]. In [38] different measurements of delay uncertainty are provided for 45nm technologies down to 16nm. Similarly, delay variations of next technology nodes are analyzed in [103]. Both studies remark the importance of process variation in 2D communication links as technology scales down, concluding that the presence of the increasing delay variability and the increase of defect densities in manufactured chips significantly increase the probability of having a faulty wire in any of the links in the network.

To cope with the presence of faults in links several proposals have been made that are able to tolerate faulty wires. Some of them tolerate infrequent run-time timing violations, where delay failures are tolerated at the cost of performance [23,65]. Other proposals focus on increasing interconnect yield by using redundant hardware. The first work that proposed a compact hardware implementation to use spare wires in NoC links was [34]. Spare wires are added to tolerate a limited number of faults without decreasing communication performance. To do so, a crossbar is used to choose a set of non-faulty wires to perform the transmissions. The same idea has been applied to vertical links in [55]. However, as stated in [79] and [78], the use of a high number of TSV wires leads to physical implementation challenges and overheads due to the large area requirements of TSVs. Moreover, the proposal in [55] is unable to tolerate most of the failures caused by misalignments because fault tolerance is limited to non-consecutive failures. On the other hand, to minimize the overhead, in [78] a combination of limited spare TSVs with the use of error correction codes is used. In the same way, serialization schemes have been proposed to reduce TSV footprint [79] [77]. However, although serializing transmissions effectively reduces the required silicon area, its noticeable increase in delay makes this technique much less appealing.

In this chapter we propose a new TSV-based fault-tolerant vertical link design that minimizes the number of redundant TSVs per link. In fact, the new proposal allows to significantly reduce the total amount of TSVs per link, thus reducing the overall TSV area footprint. Additionally, fault tolerance is provided regardless of the failure pattern (thus, covering misalignment and random defects), while system performance is kept.

7.2 An Effective Fault-Tolerant Vertical Link Design

In this section we present a completely different approach to fault-tolerance in 3D chips. We introduce a fault-tolerant vertical link design that tolerates the presence of faulty TSVs minimizing the need for extra ones. Actually, the proposed design requires less TSVs than a regular vertical link design. In this section we describe how we will reduce TSV link width from the regular N-bit link down to $N/2 + m$, being m much smaller than N, thus saving area in practice.

7.2.1 Fault-Tolerant $\frac{N}{2} + m$ 3D Link Architecture

The proposed fault-tolerant vertical link is based on the fact that vertical links implemented with TSVs are much faster than traditional 2D links. If both, vertical and horizontal links, use the same clock frequency, then the lower delay of the TSV-based link allows to transmit a flit in two halves within a single clock cycle. Splitting a flit into two halves allows to reduce the number of effective TSVs from N down to $\frac{N}{2}$. However, in order to ensure fault-tolerance, some redundant TSVs are required. In our proposal, the amount of redundant TSVs will be denoted by m. We will later analyze the exact value for m.

This change in the number of TSVs needs, however, the corresponding change in the transmission scheme. In our proposal, fault-free transmission is provided by adding a hardware reconfiguration logic at both ends of the vertical link (see Figure 7.1) that allows to select $\frac{N}{2}$ TSVs from the set of $N/2 + m$ TSVs of the link.

The added hardware is an Omega network (Ω), which is used at both ends

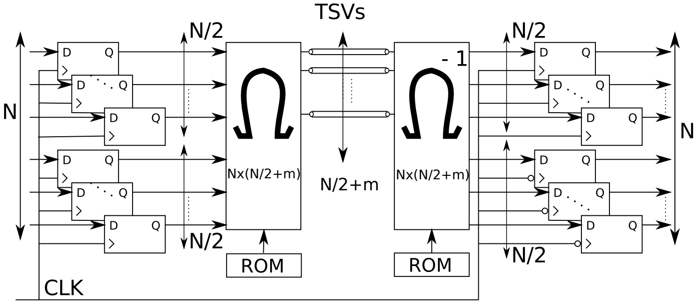


Figure 7.1: Diagram of the proposed fault-tolerant vertical link. D flip-flops are within the corresponding switch.

of the vertical link in order to allow link reconfiguration at low cost. An Ω network is made of $log_2 N$ stages with $\frac{N}{2}$ 2 × 2 switches at each stage. The Ω network has the concentration property, as it can be configured to concentrate any set of unrelated input signals (not necessarily consecutive) into a set of consecutive output signals and vice versa. In order to avoid faulty TSVs we will use such property. Indeed, during the test stage, faulty TSVs of the link are identified and the proper configuration of the Ω network is computed and stored in a ROM memory associated with the Ω network. This configuration will be used later to reconfigure the Ω network for each of the splitted flit transmissions. In this way, at the beginning of the clock cycle, the Ω network will be properly configured in order to forward the first half of the flit to the selected fault-free $\frac{N}{2}$ TSVs. Then, at the middle of the clock cycle the Ω network will be reconfigured again so that the remaining half flit is forwarded through the same $\frac{N}{2}$ TSVs. For that purpose, half of the input registers of the switch need to be replaced by flip-flops triggered by the negative flank of the clock signal. Figure 7.2 shows a timing diagram for this procedure: the first half of the flit ($\frac{N}{2}$ bits) is transmitted during the first half of the cycle whereas the second half is transmitted during the second half of the clock cycle. As can be deduced, what is changed from one configuration to the other configuration of the Omega network is the set of input signals coming from the switch that will be forwarded through the link. Also, in the receiving side, what is changed is the set of outputs form the Omega network that deliver information to the input registers. In this way, notice that the Ω network at the input of the vertical link is used to de-concentrate half of the flit into the right set of $\frac{N}{2}$

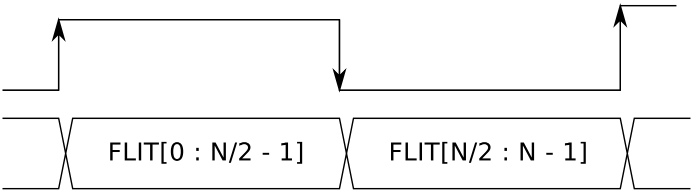


Figure 7.2: Timing of a transmission of a flit through a vertical link.

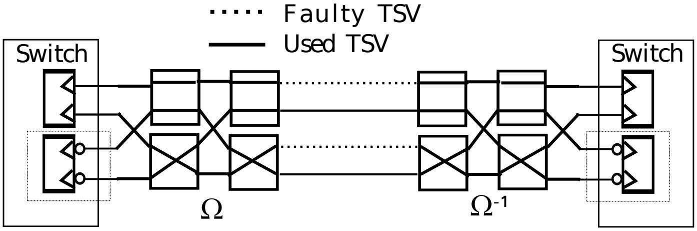


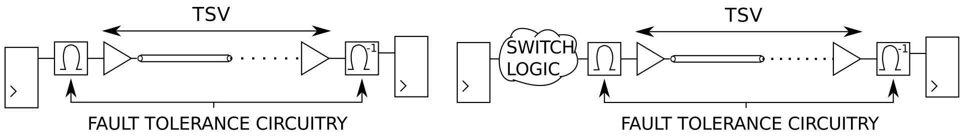
Figure 7.3: Example of a 4-bit vertical link while transmitting the second half of the flit.

TSVs, whereas the inverse Ω^{-1} network at the other end of the link is used to concentrate the transmitted data back to the corresponding half of the flit. Figure 7.3 shows an example of how the de-concentration and concentration operations are performed by the Ω and Ω^{-1} networks in a 4-bit vertical link.

On the other hand, the new fault-tolerant link design relies on the assumption that the probability of having a number of faulty TSVs in a link larger than m is negligible. Notice that the actual number of extra TSVs required (m) to achieve high yield results depends on the TSV failure probability and the width of the link (N). In the evaluation section we compute the proper value of m.

In summary, we are proposing a fault-tolerant vertical link design where the number of required TSVs is noticeably lower than in a regular vertical link. This reduction in the number of TSVs requires using two Ω networks per link. However, the area required by both Ω networks (and their ROMs) is lower than the savings provided by a reduced amount of TSVs. Thus, the overall result is a reduction in area, as will be shown in the next section.

Notice that another choice for implementing our N/2+m proposal could be using a combination of multiplexers at the front and back of the TSV link, what would sound much less costly and therefore a more desirable solution.



(a) Output registered switch

(b) Output non-registered switch

Figure 7.4: Schematic of the link for two switch architecture scenarios

However, using Ω networks provides more flexible reconfiguration capabilities that a set of multiplexers, better scalability with link width, while the fan-out seen by the output register is kept much lower [41].

7.2.2 Integrating the Vertical Link into the Switch Architecture

The proposed TSV-based vertical link design needs to be properly integrated in a switch design. Indeed, two different scenarios can be identified. The first one is a switch architecture where the last pipeline stage has its outputs registered. This scheme in shown in Figure 7.4(a). In this switch architecture, the vertical link is directly connected from buffer (output buffer) to buffer (input buffer). The buffer in the sender switch represents the buffer of the last pipeline stage of the switch and the buffer in the receiver switch represents the buffer of the initial pipeline stage of the switch.

Another possible scenario occurs when the switch outputs are non-registered. This later configuration is shown in Figure 7.4(b). In that case the delay of the last switch pipeline stage is computed as the sum of the combinational logic delay of the last switch stage plus the link delay. From now on, in the rest of the chapter a switch architecture with its outputs registered is considered by default. The reason is that the switch architecture in Figure 7.4(a) is the most suitable one for CMPs [25] [76] as it provides higher operating frequencies. However, the fault-tolerant technique presented in this chapter can be easily extended for the case of a switch with non-registered outputs.

7.2.3 Timing Analysis

The fault-tolerant circuitry introduced in the vertical link (Ω network and ROM) adds some delays to the transmission path, thus being necessary to analyze the new network performance. In order to analyze the resulting operating frequency of the network, we should differentiate between switches and links. The slowest path (critical path) amongst all paths of a switch and links will set the operating frequency of the network. Thus, the critical path of the network can be computed as[1]:

$$T = \max\{\text{switch stage delay}, \text{2D-link delay}, \text{TSV-link delay}\} \tag{7.1}$$

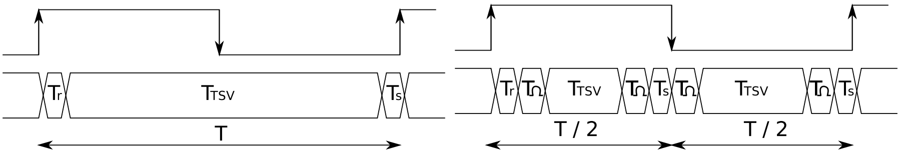
where the switch stage delay is the delay of the slowest stage in the pipeline of the switch. The 2D-link delay is the delay that a flit suffers when crossing a metalization wire in the 2D plane, which includes the wire delay, repeaters delay, the setup time, and the delay time of the registers that are connected to each link wire. The TSV-link delay is the delay that involves the TSV with its repeaters plus the setup and delay time of the registers connected to that via.

In the context of NoCs, the critical path in the formula above will usually be set by the critical path of the switch as the 2D-link delay can be highly reduced by introducing an optimal number of high-sized repeaters [14]. However, this comes at the expense of a considerable increase in link power consumption. On the other hand, TSVs present a delay that is significantly lower than the rest of the components of the network.

In order to transmit both halves of the flit within the same clock cycle, while keeping the complexity of the circuit low, the negative edge flank of the clock is sampled by half of the registers at the receiving end of the link (see Figure 7.1). Note that all of the registers in that design work at the same clock frequency, and hence, there is no need to increase the complexity of the clock signal, that remains untouched. In this way, as clock frequency is not increased power consumption of the proposal is not increased except for that of the Ω network. Nevertheless, their contribution to the total power is negligible.

Using both flank edges of the clock induces, however, tighter timing constraints to the link design. Figure 7.5(a) shows the delay of each component

[1] Note that the following equation is for a switch where the last pipeline stage has its outputs registered. In the case of non-registered outputs the time of the combinational logic in the last stage of the pipelined switch should be added to the link delay.



(a) Standard TSV-based link (b) Fault-tolerant TSV-based link

Figure 7.5: Timing constraints of a standard TSV-based link design and the proposed fault tolerant vertical link design

for a TSV link with no fault tolerance support. The delay constraint for that link design is

$$T_\mathrm{r} + T_\mathrm{S} + T_\mathrm{TSV} \leq T \tag{7.2}$$

where T_S and T_r are the setup and delay time of the registers, respectively. T_TSV is the TSV delay (link delay). Figure 7.5(b) shows the time diagram of a fault-tolerant vertical link design. In this case all the input registers at the transmitting end load at the same positive flank of the clock signal, whereas half of the registers at the receiving end load at the positive flank and the rest at the negative flank. As can be seen in Figure 7.5(b), the first half of the clock cycle is the critical one, setting the critical path of the fault-tolerant vertical link. Then, the latency of the fault-tolerant link is:

$$T_\mathrm{S} + T_\mathrm{r} + 2 * T_\Omega + T_\mathrm{TSV} \leq \frac{T}{2} \tag{7.3}$$

where T_Ω is the delay of an Ω network. Now the timing condition is set to $T/2$, penalizing the maximum achievable frequency of the fault tolerant link design with respect to the conventional one. However, as TSV wires in the vertical link have an order of magnitude higher transmission rates than horizontal links [14], the speed of messages traveling along horizontal links and then using vertical links will be bounded by the horizontal link speed, thus not achieving the potentials of the vertical TSV speed. This fact relaxes the timing constraints of our link design in practice.

Finally, it is noteworthy to mention that a register must hold its data signal an amount of time after the clock event (hold time). In our circuit, the hold time timing constraint of a register is not critical, being the setup time timing constraint the one that sets the minimum delay of the fault tolerant vertical link circuit as described above. In the evaluation section we will obtain the maximum frequency achievable in our link design.

7.2.4 Omega Network Properties for Fault Tolerance Support

The reconfiguration capabilities of the Ω network enables the vertical link design to cover all the failure patterns (assuming at most m failed TSV wires). This can be explained by analyzing the non-blocking permutations of the Ω^{-1} network. The permutations supported for both the Ω and the Ω^{-1} networks were shown in Section 6.3.4.

7.3 Evaluation

In the previous section we have presented our fault-tolerant proposal. Also an initial discussion on yield and timing has been introduced. In this section results of area, timing, and yield are thoroughly analyzed.

7.3.1 Yield

As shown in Chapter 2, chip yield can be dramatically reduced when no fault-tolerance is provided to vertical links. In our proposal, where at least N/2 TSVs per link are required to be non-faulty, the minimum number of additional TSVs needed to ensure a fault-free transmission depends on the TSV failure rate P_{TSV} and the width of the link. Assuming a probability P_{TSV} of having a faulty TSV, the probability of having a vertical link with a number of faulty TSVs lower than m is given by Equation 7.4. In this equation C(N,i) represents the number of possible combination of i wires from a total of $\frac{N}{2} + m$ wires.

$$LinkYield \leq \sum_{i=0}^{m} C(\frac{N}{2} + m, i)(P_{TSV})^{i}(1 - P_{TSV})^{\frac{N}{2}+m-i} \tag{7.4}$$

Additionally, in order to analyze chip yield, the number of TSV links in the chip needs to be considered as well. Note that if a 3D mesh topology is assumed, the number of vertical links in the network will directly depend on the number of nodes per layer and the number of stacked chips. Taking all these data into consideration, Table 7.1 shows the expected values of chip yield for different scenarios when an increasing amount of extra TSVs (m) is considered.

Layers	2			
Nodes	16		64	
P	LOW	HIGH	LOW	HIGH
m=0	98.9812	90.2663	95.9867	66.3902
m=1	99.9998	99.9841	99.9993	99.9366
m=2	99.9999	99.9999	99.9999	99.9999

Layers	5			
Nodes	16		64	
P	LOW	HIGH	LOW	HIGH
m=0	95.9867	66.3902	84.8877	19.4274
m=1	99.9993	99.9366	99.9974	99.7468
m=2	99.9999	99.9999	99.9999	99.9997

Table 7.1: Chip Yield for 64-bit links implemented by using $N/2 + m$ TSVs.

Layers	2			
Nodes	16		64	
P	LOW	HIGH	LOW	HIGH
m=0	97.9728	81.4802	92.1345	44.0766
m=1	99.9994	99.9358	99.9974	99.7433
m=2	99.9999	99.9999	99.9999	99.9995

Layers	5			
Nodes	16		64	
P	LOW	HIGH	LOW	HIGH
m=0	92.1345	44.0766	72.0592	3.7743
m=1	99.9974	99.7433	99.9897	98.9773
m=2	99.9999	99.9995	99.9999	99.9979

Table 7.2: Chip Yield for 128-bit links implemented by using $N/2 + m$ TSVs.

64-bit wide vertical links are implemented with $N/2 + m$ TSVs. Results in Table 7.1 provide yield data for chips containing either 2 or 5 stacked layers, each of them containing either 16 or 64 cores. Additionally, low (0.00001) and high (0.0001) TSV failure probabilities have been considered, as well as adding zero, one, or two extra TSVs per link (m). Note that values of chip yield assume that stacked dies are fault free and the possible failures affecting yield are only due to the failure rate of TSV bonding.

Results in Table 7.1 confirm that chip yield can be considerably penalized when no extra TSVs are considered. For example, even for the smallest chip configuration considered (2 dies with 16 cores each), 10% of the chips should be discarded. For the largest configuration, this percentage increases up to 80%. Results in Table 7.1 also show how with only 1 or 2 extra TSVs per link, chip yield can be noticeably enhanced. Concretely, in the case of having 2 stacked 16-node dies with a TSV failure probability P=0.0001, chip yield increases from 66.3902% when no extra TSV is added, up to 99.9366%, and 99.9999% with 1 or 2 extra TSVs, respectively.

The main conclusion from the data in Table 7.1 is that by using our proposal only 2 extra TSVs per link are enough in order to provide good yield results (a 64-bit vertical link implemented with 34 TSVs). Table 7.2, that shows data for 128-bit wide links, provides similar conclusions.

7.3.2 Area Overhead

In this section we analyze the cost of introducing the fault-tolerant vertical link design. We will compare that cost with the area used by a 3D-switch. To do so, we perform the synthesis of a 3D modular switch using the 45nm technology open source Nangate [52] with Synopsys DC. The details of the 3D modular switch architecture are given in Appendix A.2. We have used M1-M3 metalization layers to perform the Place&Route with Cadence Encounter. The total link width is 64 bits (implemented with 32+2 TSVs). Links (from buffer to buffer) are modeled using Virtuoso Analog Design Environment by Cadence. Table 7.3 shows the area occupied by a 3D switch, and the extra area occupied by the link design inserted in the up and down input/output ports. As can be seen, the extra area occupied by the link is only 6.69% of the area of the whole switch.

	Area (um^2)
Baseline Switch	39699.41
Vertical Link Design	2655.74

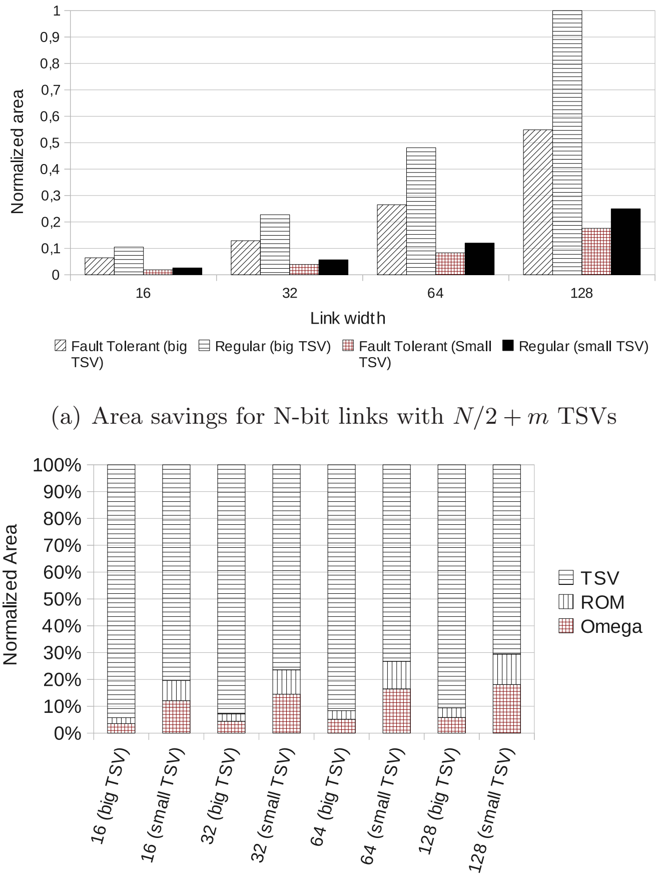
Table 7.3: Increment in area for the enhanced TSV vertical link.

The area of the fault-tolerant vertical link is the sum of the footprint area of TSVs and the area overhead of the fault-tolerant circuitry (the transistors needed to implement the different Ω networks and the ROM memories used to store the link configuration). The Ω networks and the ROMs have been designed using Virtuoso from Cadence. To achieve high performance, each 2x2 switch of an Ω network has been implemented with 8 CMOS transmission gates. Thus, the transistor count of an N-bit Ω network is $8 * \frac{N}{2} * \log_2(N)$. Similarly, the number of transistors used by a ROM is $5 * \frac{N}{2} * \log_2(N)$. The remaining area of the vertical link is the footprint area occupied by the $\frac{N}{2} + m$ TSVs.

Figure 7.6(a) shows an area comparison of a regular vertical link composed of N TSVs and the fault-tolerant vertical link with $\frac{N}{2} + m$ TSVs including the fault-tolerant circuitry. For this comparison, 2 extra TSVs per link (m) have been considered[2]. Results are normalized to the 128-bit non-fault-tolerant design. Two TSV designs are studied: big TSVs with $8\mu m$ diameter and $16\mu m$ pitch, and small TSVs with $4\mu m$ diameter and $8\mu m$ pitch. Those dimensions have been extracted from the ITRS report [27] representing current and future TSV dimensions, respectively. As shown in the figure, the fault-tolerant link design requires less area than the regular one. This is a consequence of the large reduction in the number of TSVs. This reduction is larger than the overhead of the fault-tolerant circuitry. Note that area savings improve with link width, specially for big TSVs. This is because the circuitry area becomes less significant for high link width values.

Figure 7.6(b) shows a decomposition of the fault-tolerant vertical link area. As shown in this figure, the TSV area is the major contribution to the whole link area. It is noteworthy to mention that as the width of the link increases the contribution of the circuitry becomes more important due to the use of larger

[2] Note that, according to Tables 7.1 and 7.2, using the same amount of extra TSVs for different link widths provides different yield values. Nevertheless, the focus in Figure 7.6 is area overhead.



(a) Area savings for N-bit links with $N/2 + m$ TSVs

(b) Fault Tolerant Link area decomposition

Figure 7.6: Area breakdown.

Ω networks. However, even for a 128-bit link, the contribution of TSVs is still a 70% of the total area (or 90% if big TSVs are used). Additionally, in order to increase the scalability of our approach, serialization schemes can be used in combination with the proposed vertical link design. Also, error correction code (ECC) could be used in addition to our proposal. Both methods, serialization and ECC, are orthogonal to it.

	Delay (ps)
Omega network	33
TSV	31
Register delay	38
Setup Time	9

Table 7.4: Delays for different components of the fault-tolerant vertical link.

7.3.3 Fault-Tolerant Vertical Link Timing Analysis

The fault tolerant circuit increases the delay of the TSV-based link, as pointed out in Section 7.2.3. Table 7.4 shows the worst-case delays for the different components of the new TSV-based link design. As in the previous section, we modeled the circuit using the Analog Design Environment Virtuoso from Cadence and the 45nm technology open source Nangate [52]. TSVs have been modeled using an RC model, where the R and C parameters have been obtained from [93].

As can be seen in Table 7.4, the Ω network presents a delay similar to the TSV delay. Therefore, according to equations in Section 7.2, a decrease in the maximum transmission frequency is expected. This is shown in Table 7.5 which shows the delay and the maximum achievable frequency for both, the non-fault tolerant and the fault-tolerant vertical links. The fault-tolerant circuitry highly reduces the maximum vertical link frequency, from 12.82 GHz down to 3.47 GHz. This is due to the fact that the delay of a simple TSV is extremely small and, therefore, just by introducing a few gates makes the frequency to be highly reduced. Nevertheless, as mentioned in Section 7.2, in a mesochronous NoC design where all the NoC components work at the same frequency, the original TSV frequency is much higher than frequencies in current chip designs. Thus, a large slack can be taken for granted. Indeed, the reduction in the vertical link frequency does not imply a reduction in performance as long as the NoC is not designed to operate at frequencies higher than 3.47GHz.

Table 7.5 also shows the vertical link delay measured in FO4 units, that is, we compare the minimum critical path of our proposal with the minimum delay that a technology library can provide (FO4). We have measured the FO4 for the 45nm technology open source Nangate [52] using the Analog Design

Vertical Link	Delay (ps)	Max. Frequency (GHz)	Delay (FO4)
Non-fault tolerant	78	12.82	2.23
Fault tolerant	288	3.47	8.23

Table 7.5: Operating frequency comparison vertical link designs.

Environment Virtuoso by Cadence. We have obtained FO4 to be approximately set to 35ps. Thus, the delay of the fault-tolerant link is only 8.23 FO4.

7.4 Reaching Higher Frequencies with the Fault-Tolerant Vertical Link Design

The fault-tolerant vertical link solves TSV failures by transmitting a flit in two halves during a single clock cycle. The main benefit is that the number of required TSVs noticeably decreases without requiring the rest of the network to be modified. However, it increases the vertical link delay, thus imposing a higher limit in the maximum achievable frequency (as shown in the previous section). This frequency limit is quite high when compared to current NoC deployments. For example, the Tile-Gx100 chip by Tilera works at a maximum frequency of 1.5GHz and the *Single-chip Cloud Computing* prototype chip by Intel [86] including 24 dual-core tiles based on the x86 works at maximum frequency of 1.84GHz. Nevertheless, in some cases higher frequencies may be required as in the case of the Polaris Intel prototype [40] that reaches frequencies close to 5GHz.

In this section we enhance our design in order to achieve higher frequencies. Thus, the enhanced fault-tolerant mechanism presented in this section will be suitable for those aggressive scenarios where the timing constraints are not fulfilled by the baseline fault-tolerant mechanism presented in Section 7.2. This is achieved by modifying the switch design.

Figure 7.7 shows an schematic of the enhanced fault-tolerant mechanism. The only difference with the initial mechanism (proposed in Section 7.2) is that half of the input registers are modified in order to work also with the negative edge of the clock signal. This simple modification aims at reducing the delay of the vertical link but forces the rest of the components of the

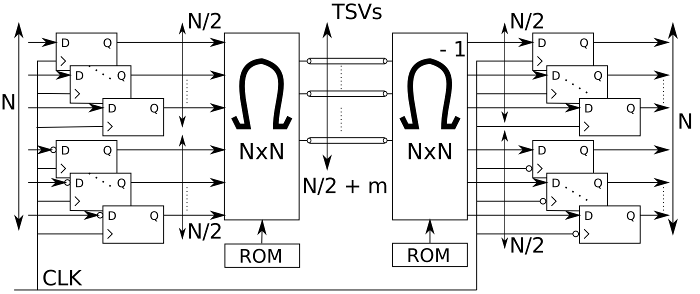
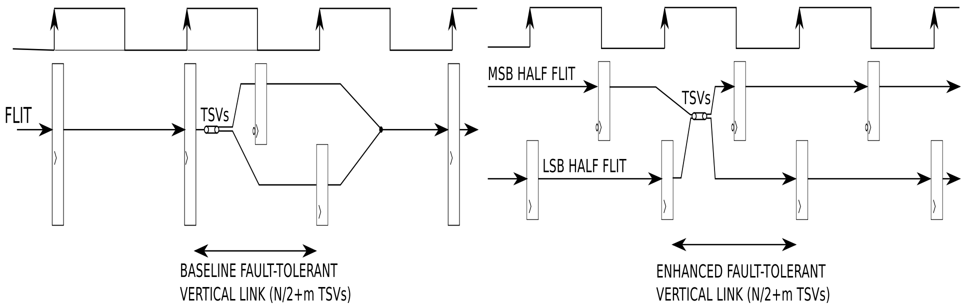


Figure 7.7: Diagram of the enhanced fault tolerant vertical link.

network to work at both edges of the clock. Figure 7.8 depicts the pipeline schematic for the previous design and for the enhanced proposal. As it can be seen, in the original design registers clocked at negative edge are used only in the TSV stage. In contrast, the enhanced fault-tolerant mechanism forces also the switch to toggle at the negative edge of the clock. A flit that crosses a switch is splitted into two parts: the first half of the flit crosses the network traversing registers that toggle with the positive edge of the clock cycle, while the second half of the flit crosses the network traversing buffers that work with the negative edge of the clock signal (see Figure 7.8(b)). Both halves only share the $N/2 + m$ TSV that compound the vertical link – as in the case of the baseline fault-tolerant circuit. Note that the same philosophy that is applied to the vertical link could be applied to any 2D link, that is, the number of wires of a 2D link can be reduced to $N/2 + m$. However, this solution is not suggested because as the timing constraint increases, the power consumption of a 2D wire highly increases, becoming unaffordable.

It is important to remark that the whole new design works at the same operating frequency as before. Thus, no increment in the complexity of the clock signal generation is required. Notice also that no loss in performance is incurred. Indeed, the critical path of the enhanced fault-tolerant proposal does not change with respect to the initial design. In fact, the critical path of the switch is set by the control path. As it can be seen in Figure 7.8, the control path remains identical in both designs.

Redesigning the entire switch to toggle at both edges of the clock signal has an important benefit, as it relaxes vertical link timing constraints. Figure 7.9 shows the new time diagram. Each half of the flit is treated independently and



(a) Baseline fault-tolerant pipeline schematic.

(b) Enhanced fault-tolerant pipeline schematic.

Figure 7.8: Pipeline schematic of the baseline and the enhanced fault-tolerant designs.

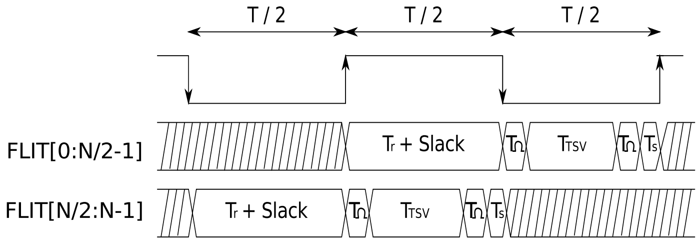


Figure 7.9: Timing diagram of the enhanced fault-tolerant vertical link.

sequentially. That is, while the second half is being loaded into the registers, the first half is crossing the TSV link.

As the timing constraint is relaxed, the delay required to traverse the vertical link is reduced. The setup time of the registers still sets the timing constraint. Thus, as the TSV link stage tasks are divided now into two halves (loading into the registers and crossing the link), the minimum critical path is now computed as the maximum of these minimum critical paths. The critical path of the enhanced proposal is the maximum of:

$$T_{\mathrm{r}} \leq \frac{T}{2} \tag{7.5}$$

$$T_{\mathrm{S}} + 2 * T_{\Omega} + T_{\mathrm{TSV}} \leq \frac{T}{2} \tag{7.6}$$

where the first equation represents the timing constraint of loading a register, and the second equation represents the timing constraint of traversing

Vertical Link	Delay (ps)	Maximum Frequency (GHz)	Delay (FO4)
Non-fault tolerant	78	12.82	2.23
Fault tolerant design	288	3.47	8.23
Enhanced	212	4.72	6.06

Table 7.6: Operating frequency comparison.

the TSV link. As it will be shown later, the second equation sets the critical path. Note that the delay of the enhanced fault-tolerant proposal shown in Equation 7.6 is lower than the delay of the initial fault-tolerant vertical link shown in Equation 7.3.

7.4.1 Evaluation

In this section we perform the timing analysis for the enhanced mechanism. This mechanism presents the very same properties than the previous proposal regarding yield and area as the only difference among them is the edge used for some flip-flops in the switch. As a case study of this enhanced mechanism, we have applied it to the switch presented in the appendix. Results of Table 7.6 compare the data for the initial proposal (already presented in Table 7.5) with the features of the enhanced mechanism. As can be seen, the enhanced fault tolerant proposal relaxes the timing constraint (see Equation 7.3 and 7.6), and hence, the maximum achievable frequency is increased up to 36% with respect to the initial fault-tolerant vertical link design (from 3.47 GHz to 4.72 GHz). This increment in the maximum achievable frequency comes at the expense of increasing the complexity of the switch. However, performance of the entire NoC is not affected when inserting a fault-tolerant vertical link mechanism if the switch stage delay or the 2D link delay (see Equation 7.1) are higher than 288 ps for the initial fault-tolerant vertical link, or 212 ps when inserting the enhanced proposal. If those numbers are compared in terms of F04 delays, the enhanced proposal provides a critical path that is only 6.06 FO4. Note that the critical path of the 3D switch presented in the appendix at the end of the chapter is 530 ps, that is, 15.14 FO4. With this configuration, the TSV link design is plausible.

7.5 Conclusions

In this chapter a new fault-tolerant TSV link design is proposed to deal with the high failure rate of TSVs in 3DIC designs. The new proposal exploits the slack available in vertical links that use TSVs in order to perform a splitted transmission of flits using the positive and negative edges of the clock. An Ω network is placed between the switch and the vertical link allowing to transmit flits in two halves by selecting a subset of $\frac{N}{2}$ fault-free TSVs. Results confirm that the additional hardware cost of this proposal is affordable as the the circuitry overhead of this proposal os just 6.69% of a 3D switch.

One of the main advantages of the new proposal is noticeably reducing the total number of TSVs in the chip, as TSV-link width is reduced from N bits down $\frac{N}{2} + m$ bits. Actually, this reduction, and the associated area reduction, will be more noticeable in the future as technology scales down to larger integration scales, due to the fact that TSVs do not linearly scale to smaller sizes as other components, like devices or horizontal metalizations do.

On the other hand, results show that the ability of the new proposal to tolerate faulty TSVs is much superior to other proposals. More concretely, for the expected TSV failure rates, the yield of our approach tends to 100% regardless of the exact pattern of failures. This feature of the new proposal may be even more interesting than its area savings as TSV reliability is an important concern, perhaps being a larger road-block to their commercial adoption than their area overhead.

Chapter 8

Conclusions and Future Directions

In this chapter the main conclusions of this dissertation are summarized. Additionally, future directions for continuing the research of this dissertation are given. Finally, in the last section of this chapter the main results of this dissertation, in terms of scientific publications, are listed.

8.1 Conclusions

As integration technologies continue to scale down, parameter variation is becoming a major concern that cannot be ignored at any system level, as unpredictable variations affecting silicon devices and metalizations in the bottom layer are directly impacting performance at the upper levels. In this context, the modeling of process, voltage, and temperature variations is becoming mandatory as technologies scale down. By modeling the effects of parameter deviations, designers may increase design predictability and better exploit design resources by performing a better tuning of their designs. In this way, the main effort of this thesis has been focused on a detailed model of process, voltage, and temperature variations. The model presented in Chapter 3 characterizes the main variability sources affecting to both CMPs and MPSoCs designs, and takes into account all crossing dependencies existing among these sources of variability. This model provides improved accuracy over other variability models as variations are injected directly to synthesized designs. Other improvements of the model are the detailed characterization of chip interconnects and the accurate modeling of random dopant fluctuations. Additionally, the framework developed for modeling variations can be integrated into the regular design flow in order to assess the impact of PVT variations prior to manufacture the chip, thus providing chip designers with a powerful tool that will increase reliability and predictability for submicron technology designs.

Once the model is developed, the impact of process variation in MPSoC and CMP designs can be characterized. In this sense, in Chapter 4 the impact of variation in NoC-based designs is assessed. First, we showed that the impact of parameter variations in NoC links is not negligible, as stated in previous works. We also showed that NoC link delay is specially sensitive to random dopant fluctuations, and is expected to become the major source of variation for technologies below 32nm. Later, in a more global study we showed how parameter variations severely impact the maximum achievable frequency of NoC components. Additionally, the impact of systematic L_{eff} variations in the maximum achievable frequency of NoC routers was shown. Measurements of this study detected strong spatial correlation among router frequencies in the chip. Additionally, we also showed that differences on the maximum achievable frequency of routers and links in the NoC may induce communication bottlenecks that severely impact NoC performance. Finally, a different application scenario was presented. This scenario was intended to check the robustness of MPSoC architectures against process variations. We injected variations into a whole MPSoC design, that is, both processors and the NoC. For that purpose we perform the synthesis of an open source processor and a custom designed NoC. In this example, different clocking schemes for the NoC interconnect have been characterized to prove its variability robustness. Results of this study confirm that, on one hand, Voltage Islands are required to connect the processor to the NoC in order to mitigate the impact of variations in the maximum achievable frequency of processors. On the other hand, it was stated that mesochronous clocking schemes behave more robustly against process variations than traditional synchronous clocking schemes.

As shown in Chapter 4 communication bottlenecks arising as a consequence of process variations noticeably impact the performance of the NoC. In the context of CMPs, variations also cause cores an memories to present an unpredictable behaviour causing the degradation of performance and predictability of applications running on the CMP. To overcome this, a variation-aware mapping policy that faces systematic variations in GALS-based CMPs is presented in Chapter 5. In this chapter, we showed that in order to efficiently exploit chip multi-core resources, variation-aware mapping mechanisms are required. In this way, by taking into account the post-manufacturing information of the different chip components, both performance and predictability of applications running on the CMP can be noticeably improved. The proposal presented is based on assigning application idle cores in the CMP that additionally form a uniform region from the frequency point of view. Moreover, that region is composed of the fastest available idle cores at the time of the mapping. This variability-aware mapping has been presented in two different flavors: Multiple Frequency Single Voltage Region *MFSV* and Single Frequency Multiple Voltage Region *SFMV*. Results showed the benefits of using our proposal. In the case of the *MFSV* mapping strategy, average execution time is reduced down-to 10%, while for some applications this reduction increases up to 23%. These improvements in execution additionally cause the reduction in total energy of up to 24%. In the case of *SFMV* approach, speed ups are smaller while energy savings increase up to a 33%, as this policy is focused on energy more than performance.

As stated in Chapter 4, random variations will severely impact the performance of NoC for technologies below 32nm. In that chapter we showed that reducing link frequency reduces considerably link performance when delay uncertainty increases. Additionally, as defect density levels increase with technology scaling, and NoC links are specially prone to manufacturing defects, reducing the frequency is not enough and, therefore, fault-tolerant NoC link designs are required. In this way, to solve some of the performance and yield issues affecting NoC links, a variable phit-size NoC architecture was presented in Chapter 6. In this NoC architecture, wire failures are tolerated by enabling a variable phit-size injection and ejection of flits at the network interfaces. Note that the proposed mechanism is able to tolerate the presence of faulty wires regardless the origin of failures and for an unbounded number of faulty wires. Results confirm that the proposed variable architecture retrieves more bandwidth than traditional variable frequency links in the presence of high variations. Concretely, the proposed link design is able to achieve for a 32nm technology 31% and 34.4% speed-up over a variable-frequency link, for LP and HP links, respectively.

As mentioned before, memory bandwidth is predicted to become a major limiting factor for CMP and MPSoC designs in the many-core era. In this context, three dimensional Integrated Circuits (3DICs) arise as a promising technological solution to partially alleviate the problems caused by the pin-out limitations. In 3DICs the stacked dies communicate through vertical links. The most adopted solution to implement vertical links is by using Through Silicon Vias (TSVs). However, TSV-based links present two main drawbacks. One is the high defect rate of TSV. The other one is TSV footprint area. In this context, 3D designs demand for efficient mechanisms able to overcome both yield and area overheads of regular vertical link designs. In this regard, in Chapter 7 a new fault-tolerant vertical link design is proposed. The new design exploits the slack available in vertical links that use TSVs in order to perform a splitted transmission of flits using the positive and negative edges of the clock. Results showed that the additional hardware cost of this proposal is affordable as the circuitry overhead of this proposal is just 6.69% of a 3D switch. Additionally, one of the main advantages of the new proposal is noticeably reducing the total number of TSVs in the chip, as TSV-link width is reduced from N bits down $\frac{N}{2} + m$ bits. Actually, this reduction, and the associated area reduction, will be more noticeable in the future as technology scales down to larger integration scales, due to the fact that TSVs do not linearly scale to smaller sizes as other components, like devices or horizontal metalizations do. On the other hand, results show that the ability of the new proposal to tolerate faulty TSVs is much superior to other proposals. More concretely, for the expected TSV failure rates, the yield of our approach tends to 100% regardless of the exact pattern of failures.

In summary this dissertation presented a detailed characterization of the impact of process variation in both MPSoCs and CMPs. This has been carried out by the development of a powerful framework able to inject variations in synthesized designs. Once the framework was developed and the behaviour of different NoC-based designs under process variations was characterized, we were in a position to propose several mechanism able to face the impact of process variation. In particular, we presented three mechanisms. The first mechanism, a variation-aware mapping policy, was intended to face the impact of systematic variations in NoC-based CMPs. The second one, a variable phit-size NoC architecture was intended to face the presence of random variations and permanent faults in NoC links. Finally, by using some of the concepts of the fault-tolerant 2D, a new TSV-based vertical link NoC design has been proposed. All three proposed mechanism have been proved to achieve good results at a reasonable hardware cost when comparing with other current state-of-art proposals.

8.2 Future Directions

Future research directions to continue the work presented in this dissertation are given below.

- As stated in this dissertation, random dopant fluctuations are expected to be the major source of variations for technologies below 32nm. Additionally, it was also shown that predicting the number of dopant atoms in the transistor channel is specially difficult for low sized devices. As a consequence of this, the contribution of random threshold voltage variations would be specially noticeable in minimum-sized devices. In the context of CMPs, cache memories will be the components most impacted by random dopant fluctuations as they are usually built with minimum size devices. Therefore, an immediate extension of this work is to quantify the impact of random variations in memory access time with the model proposed in this dissertation.

- In order to improve the benefits of the proposed variation-aware mapping algorithm new features have to be incorporated to the baseline mechanism. On one hand, building irregular regions is the easiest way to reduce chip fragmentation and consequently increase chip utilization. For that purpose fault-tolerant routing mechanisms have to be included the CMP architecture. On the other hand, enabling to migrate threads dynamically may increase mapping performance due to two reasons. First, regions with better features may be more intensively used. Second, chip fragmentation may be minimized as the reordering of regions would be possible. Moreover, as thread migration may cause an unbalanced utilization of chip resources, and probably the resources presenting better features would be the more utilized, thermal variations should be also taken into account. Note that workload execution causes temperature variations in the CMP involving the variation of the operational characteristics of wires and devices. In this context, if the effect of temperature is neglected in the mapping algorithm, the possible benefits of increasing the use of the best possible regions may be canceled by the performance penalization caused by temperature increase.

- In future manufacturing scenarios the unreliability caused by process variations will dominate over other sources of unreliability. In this sense, as variation induced timing errors are the most frequent errors, the global phit size reduction (GFR) is the better way of applying the variable phit-size NoC architecture. However, a detailed modeling of failure patterns may help to find such situations where the local phit size reduction (LPR) mechanism can be applied.
- In order to increase the effectiveness of the proposed vertical link design, serialization schemes and error correction codes may be incorporated in the proposed vertical link design. In this way, a better area-performance-yield trade-off may be found.

8.3 Publications related to this thesis

The following section enumerates the publications related with this dissertation. Note that publications shown in this list have been either published or submitted for publication. The first list of publications correspond to international journal papers.

- Hernández, C., Silla, F., Duato, J, Ludovici, D., and Bertozzi, D., ”A Framework for Exploring the MPSOC Design Robustness in the Presence of PVT Variations”, submitted to ACM Transactions on Embedded Computing Systems.

- Hernández, C., Roca, A., Silla, F., Flich, J. and Duato, J, ”A Novel Low-Area Fault-Tolerant Vertical Link Design for Eective 3D Stacking”, submitted to the Special Issue on 3D VLSI System Integration of The Computer Journal.

- Hernández, C., Roca, A., Silla, F., Flich, J. and Duato, J, ”Fault-Tolerant Vertical Link Design for Effective 3D Stacking” IEEE Computer Architecture Letters, 29 Jun. 2011. IEEE computer Society Digital Library.

- Hernández, C., Roca, A., Flich, J., Silla, F. and Duato, J. On the Impact of Process Variation in GALS-based NoC Performance, accepted for publication in the Transactions on Computer-Aided Design of Integrated Circuit and Systems.

- Hernández, C., Roca, A., Flich, J., Silla, F. and Duato, J. (2011). Characterizing the impact of process variation on 45 nm NoC-based CMPs. Journal of Parallel and Distributed Computing, 71(5), 651 - 663.

The following list include publications in international conferences.

- Hernández, C., Silla, F. and Duato, J (2011). Energy- and Performance-Efficient Thread Mapping in NoC-based CMPs under Process Variations. Proceedings of the 2011 IEEE International Conference on Parallel Processing, Taipei.

- Hernández, C., Silla, F. and Duato, J (2010). A Methodology for the Characterization of Process Variation in NoC Links. In 2010 Design, Automation & Test in Europe Conference & Exhibition (DATE 2010), pages 685-690. Dresden, Germany : EDDA.

- Hernández, C., Roca, A., Silla, F., Flich, J. and Duato, J (2010). Improving the Performance of GALS-Based NoCs in the Presence of Process Variation. In 2010 ACM/IEEE International Symposium on Networks-on-Chip (NOCS), pages 35 - 42. Grenoble, France : ACM.

- Hernández, C., Silla, F., Santonja, V. and Duato, J (2009). A new mechanism to deal with process variability in NoC links. In IPDPS 2009

- Proceedings of the 2009 IEEE International Parallel and Distributed Processing Symposium, pages IEEE Computer Societ. Rome, Italy.

The following publications have been presented in non-peer-reviewed conferences, poster sessions, and workshops without proceedings.

- Hernández, C., Silla, F. and Duato, J (2010) Measuring the Impact of Process Variation in NoC Links. XXI Jornadas de Paralelismo, Valencia.
- Hernández, C., Silla, F., Santonja, V. and Duato, J (2009). Phit reduction to deal with process variation. 5th International Summer School on Advanced Computer Architecture and Compilation for Embedded Systems, 2009.
- Hernández, C., Silla, F., Santonja, V. and Duato, J (2008). Variability tolerance versus fault-tolerance for NoC links. Fourth International Summer School on Advanced Computer Architecture and Compilation for Embedded Systems, 2008.
- Hernández, C., Silla, F., Santonja, V. and Duato, J (2008). Recuperando ancho de banda de los enlaces fallidos en NoCs. XIX Jornadas de Paralelismo.
- Hernández, C., Silla, F. and Duato, J (2009). Characterizing random variations in NoC links
- Hernández, C., Silla, F., Santonja, V. and Duato, J (2008) Dealing with variability in NoC links. 2nd Workshop on Diagnostic Services in Network-on-Chips 2008.

Finally, we present a list including journal and international conference publications where the variation model proposed in this thesis has been employed.

- Strano, A., Hernández, C., Silla, F. and Bertozzi, D (2010). Self-Calibrating Source Synchronous Communication for Delay Variation Tolerant GALS Network-on-Chip Design. accepted for publication in International Journal of Embedded and Real-Time Communication Systems.

- Strano, A., Hernández, C., Silla, F. and Bertozzi, D (2010). Process variation and layout mismatch tolerant design of source synchronous links for GALS networks-on-chip. In System on Chip (SoC), 2010 International Symposium on, pages 43 -48.

- Rodrigo, S., Hernández, C., Flich, J., Silla, F., Duato, J., Medardoni, S. et al (2009). Yield-oriented evaluation methodology of network-on-chip routing implementations. In System-on-Chip, 2009. SOC 2009. International Symposium on, pages 100 -105

Appendix A

Switch Architectures

In this appendix the features of two different switch architectures used in this dissertation are summarized. Note that throughout this dissertation the canonical switch has been also employed. However, the details of this well known architecture were given in Chapter 2. First, we show the details of a switch architecture oriented for MPSoC systems. Second, we describe a modular switch architecture. This modular switch design in the one used to enable the use of a vertical dimension in 3D NoC designs.

A.1 MPSoC Switch architecture

MPSoCs and CMPs are systems oriented to different purposes. Usually, CMPs are general-purpose systems oriented to high performance computing where high speed designs are required. On the contrary, in general, MPSoCs are devoted to an specific application and, therefore, its components are specifically designed to perform a given task. Moreover, the typical MPSoC scenario imposes severe power constraints. In this sense, MPSoC designs would usually target a lower frequency, when compared with regular CMP designs, in order to control power dissipation. As timing constraints of MPSoC designs are relaxed for reducing power dissipation, deeply pipelined designs are not required. In the particular context of NoC switch architectures, MPSoC switch designs present in general among the different features one cycle and low buffering capabilities.

In this dissertation, we use the baseline switch architecture proposed in [57]

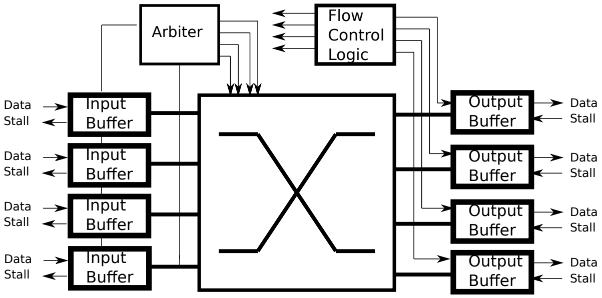


Figure A.1: Schematic of the MPSoC switch architecture

for MPSoC designs. This switch architecture is implemented with the Xpipes library [98]. Figure A.1 shows a simplified schematic of this architecture. As shown in this figure, this switch architecture presents buffers at both inputs and outputs. Flow control is performed by two dedicated wires as explained in [84]. Finally, round-robin arbitration is performed at each port individually.

In order to build a GALS system the baseline switch architecture shown in Figure A.1 has to be modified. Concretely, switch input buffers are replaced by synchronizers. Those synchronizers allow the use of different clocking schemes as the mesochronous NoC implemented in Chapter 4.

A.2 A 3D Modular Switch Architecture

In [87] a modular switch architecture was proposed. The main property of the switch is its modularity. A basic module performs the arbitration, buffering, and multiplexing tasks (see Figure A.2) in such a way that the composition of the different modules carry out the tasks of a switch in a coordinated way. Each stage of the switch is equivalent to an AC module. In parallel with the AC module, we implement an RC module that performs routing tasks.

The modular switch is the baseline architecture used for implementing a 3D switch architecture. The 3D-switch is a pipelined buffered wormhole switch with seven input/output ports. Four input/output ports are used to connect the switch to its neighbouring switches in the 2D plane. Two ports are used to connect switches in the vertical direction (up/down ports). Finally, the last port is used to connect the switch to its own processor. The switch pipeline is set to three stages. Therefore, an incoming packet needs three cycles to be

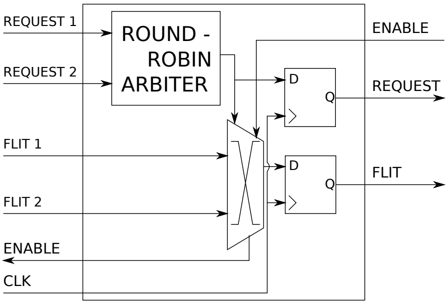


Figure A.2: Main module of the 3D switch

routed and forwarded through the switch. Additionally, link delay is set to one cycle in any direction.

The 3D switch described presents two main properties. First, the buffer is spread across the different pipeline stages, that is, every stage in the switch has its own outputs registered. Second, each output port is managed independently. In this way, each output port has its own circuitry (output port controller) which is not connected to the circuitry of the rest of output ports thus working independently from each other. Each input port can reach any output port direction except the output port that goes in the same direction (U turns are not allowed). Also, the local port connecting the switch to the core can be reached from any input port. Thus, each output port works as a 6-to-1 switch. Each output port is able to buffer, route, and forward those packets that request that output port. The independence of the output port circuitry minimizes the impact of the increased number of input/output ports [25]. In fact, this 3D-switch is a 3D-extension of a previous 2D-switch. To allow the up/down directions present in a 3D-mesh, we just added two extra input/output ports to the initial 2D-switch design and changed the RC module accordingly.

Bibliography

[1] A. Mejia, J. Flich, S. Reinemo, T. Skeie, and J. Duato. Segment-Based Routing: An Efficient Fault-Tolerant Routing Algorithm for Meshes and Tori. In *2006 International Parallel and Distributed Processing Symposium (IPDPS 2006)*, 2006.

[2] A. Asenov, S. Kaya, and J. H. Davies. Intrinsic threshold voltage fluctuations in decanano mosfets due to local oxide thickness variations. *IEEE Transactions on Electron Devices*, 49:112–119, 2002.

[3] J. Balfour and W. J. Dally. Design tradeoffs for tiled CMP on-chip networks. In *Proceedings of the 20th Annual Int. Conf. on Supercomputing*, pages 187–198, 2006.

[4] C. Bienia, S. Kumar, J. P. Singh, and K. Li. The parsec benchmark suite: characterization and architectural implications. In *Proceedings of the 17th International Conference on Parallel Architectures and Compilation Techniques*, pages 72–81, New York, NY, USA, 2008. ACM.

[5] Duane S. Boning and Sani Nassif. Models of process variations in device and interconnect. In *Design of High Performance Microprocessor Circuits, chapter 6*. IEEE Press, 1999.

[6] Shekhar Borkar, Tanay Karnik, and Vivek De. Design and reliability challenges in nanometer technologies. In *Proceedings of the 41st annual Design Automation Conference*, DAC '04, pages 75–75, New York, NY, USA, 2004. ACM.

[7] K. A. Bowman, S. G. Duvall, and J. D. Meindl. Impact of die-to-die and within-die parameter fluctuations on the maximum clock frequency dis-

tribution for gigascale integration. *IEEE Journal of Solid-State Circuits*, 37(2):183 –190, February 2002.

[8] K.A. Bowman, S.G. Duvall, and J.D. Meindl. Impact of die-to-die and within-die parameter fluctuations on the maximum clock frequency distribution for gigascale integration. *Solid-State Circuits, IEEE Journal of*, 37(2):183 –190, feb 2002.

[9] F. Brglez and H. Fujiwara. A neutral netlist of 10 combinational benchmark circuits and a target translator in fortran. In *IEEE International Symposium on Circuits and Systems*, 1989.

[10] J.A. Butts and G.S. Sohi. A static power model for architects. In *Microarchitecture, 2000. MICRO-33. Proceedings. 33rd Annual IEEE/ACM International Symposium on*, pages 191 –201, 2000.

[11] Yu Cao and L.T. Clark. Mapping statistical process variations toward circuit performance variability: an analytical modeling approach. In *Design Automation Conference, 2005. Proceedings. 42nd*, pages 658 – 663, june 2005.

[12] V. Chandramouli and C.S. Raghavendra. Nonblocking properties of interconnection switching networks. *Communications, IEEE Transactions on*, 43(234):1793 –1799, feb/mar/apr 1995.

[13] T. Chelcea and S.M. Nowick. Robust interfaces for mixed-timing systems with application to latency-insensitive protocols. In *Design Automation Conference, 2001. Proceedings*, pages 21 – 26, 2001.

[14] G. Chen and E. G. Friedman. Low-power repeaters driving RC and RLC interconnects with delay and bandwidth constraints. *IEEE Transactions on Very Large Scale Integration Systems*, 14(2):161 – 172, February 2006.

[15] G. Chen, F. Li, S. W. Son, and M. Kandemir. Application mapping for chip multiprocessors. In *Proceedings of the Design Automation Conference*, pages 620 –625, June 2008.

[16] Abhishek Das. Microarchitectural approaches for optimizing power and profitability in multi-core processors, 2010.

[17] R. Das, O. Mutlu, T. Moscibroda, and C. R. Das. Application-aware prioritization mechanisms for on-chip networks. In *Proceedings of the Annual IEEE/ACM International Symposium on Microarchitecture*, pages 280 –291, December 2009.

[18] G. De Micheli and L. Benini. *Networks on Chips: Technology and Tools (Systems on Silicon).* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2006.

[19] S. Dighe, S. Vangal, P. Aseron, S. Kumar, T. Jacob, K. Bowman, J. Howard, J. Tschanz, V. Erraguntla, N. Borkar, V. De, and S. Borkar. Within-die variation-aware dynamic-voltage-frequency scaling core mapping and thread hopping for an 80-core processor. In *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2010 IEEE International*, pages 174 –175, feb. 2010.

[20] J. Dorsey, S. Searles, M. Ciraula, S. Johnson, N. Bujanos, D. Wu, M. Braganza, S. Meyers, E. Fang, and R. Kumar. An integrated quad-core opteron processor. In *Solid-State Circuits Conference, 2007. ISSCC 2007. Digest of Technical Papers. IEEE International*, pages 102 –103, feb. 2007.

[21] Jose Duato, Sudhakar Yalamanchili, and Ni Lionel. *Interconnection Networks: An Engineering Approach.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2002.

[22] D. Ernst, Nam Sung Kim, S. Das, S. Pant, R. Rao, Toan Pham, C. Ziesler, D. Blaauw, T. Austin, K. Flautner, and T. Mudge. Razor: a low-power pipeline based on circuit-level timing speculation. In *Microarchitecture, 2003. MICRO-36. Proceedings. 36th Annual IEEE/ACM International Symposium on*, pages 7 – 18, dec. 2003.

[23] D. Ernst, Nam Sung Kim, S. Das, S. Pant, R. Rao, Toan Pham, C. Ziesler, D. Blaauw, T. Austin, K. Flautner, and T. Mudge. Razor: a low-power pipeline based on circuit-level timing speculation. In *Microarchitecture, 2003. MICRO-36. Proceedings. 36th Annual IEEE/ACM International Symposium on*, pages 7 – 18, dec. 2003.

[24] T. Fischer, J. Desai, B. Doyle, S. Naffziger, and B. Patella. A 90-nm variable frequency clock system for a power-managed itanium architecture processor. *Solid-State Circuits, IEEE Journal of*, 41(1):218 – 228, jan. 2006.

[25] Jose Flich and Davide Bertozzi. *Designing Network On-Chip Architectures in the Nanoscale Era.* CRC Press, December 2010.

[26] Jose Flich and Jose Duato. Logic-based distributed routing for nocs. *IEEE Comput. Archit. Lett.*, 7:13–16, January 2008.

[27] International Technology Roadmap for Semiconductors. ITRS 2007 Online Edition. Available at `http://www.itrs.net/Links/2007ITRS/Home2007.htm`.

[28] The International Technology Roadmap for Semiconductors. 2009 edition. Technical report, 2009.

[29] P. Friedberg, Y. Cao, J. Cain, R. Wang, J. Rabaey, and C. Spanos. Modeling within-die spatial correlation effects for process-design co-optimization. In *Proceedings of the Sixth International Symposium on Quality of Electronic Design*, pages 516 – 521, March 2005.

[30] P. Friedberg, Y. Cao, J. Cain, R. Wang, J. Rabaey, and C. Spanos. Modeling within-die spatial correlation effects for process-design co-optimization. In *Quality of Electronic Design, 2005. ISQED 2005. Sixth International Symposium on*, pages 516 – 521, march 2005.

[31] Research Gaisler. *The LEON processor user's manual*, 2001.

[32] M. Galles. Spider: a high-speed network interconnect. *Micro, IEEE*, 17(1):34 –39, jan/feb 1997.

[33] Francisco Gilabert, Simone Medardoni, Davide Bertozzi, Luca Benini, María Engracia Gomez, Pedro Lopez, and José Duato. Exploring high-dimensional topologies for noc design through an integrated analysis and synthesis framework. In *Proceedings of the Second ACM/IEEE International Symposium on Networks-on-Chip*, NOCS '08, pages 107–116, Washington, DC, USA, 2008. IEEE Computer Society.

[34] C. Grecu, A. Ivanov, R. Saleh, and P.P. Pande. Noc interconnect yield improvement using crosspoint redundancy. In *Defect and Fault Tolerance in VLSI Systems, 2006. DFT '06. 21st IEEE International Symposium on*, pages 457 –465, oct. 2006.

[35] Brendan Hargreaves, Henrik Hult, and Sherief Reda. Within-die process variations: how accurately can they be statistically modeled? In *Proceedings of the 2008 Asia and South Pacific Design Automation Conference*, ASP-DAC '08, pages 524–530, Los Alamitos, CA, USA, 2008. IEEE Computer Society Press.

[36] Sebastian Herbert and Diana Marculescu. Mitigating the impact of variability on chip-multiprocessor power and performance. *IEEE Trans. Very Large Scale Integr. Syst.*, 17:1520–1533, October 2009.

[37] C. Hernández, A. Roca, F. Silla, J. Flich, and J. Duato. Improving the performance of GALS-based NoCs in the presence of process variation. In *Proceedings of 4th International Symposium on Networks-on-Chip*, May 2010.

[38] C. Hernández, F. Silla, and J. Duato. A methodology for the characterization of process variation in noc links. In *Proceedings of the Design, Automation and Test in Europe Conference*, pages 685–690, March 2010.

[39] S. Hong, S.H.K. Narayanan, M. Kandemir, and O. Ozturk. Process variation aware thread mapping for chip multiprocessors. In *Proceedings of the Design, Automation and Test in Europe Conference Exhibition*, pages 821 –826, April 2009.

[40] Y. Hoskote, S. Vangal, A. Singh, N. Borkar, and S. Borkar. A 5-ghz mesh interconnect for a teraflops processor. *Micro, IEEE*, 27(5):51 –61, sept.-oct. 2007.

[41] Ang-Chih Hsieh, TingTing Hwang, Ming-Tung Chang, Min-Hsiu Tsai, Chih-Mou Tseng, and H.-C. Li. Tsv redundancy: Architecture and design issues in 3d ic. In *Design, Automation Test in Europe Conference Exhibition (DATE), 2010*, pages 166 –171, march 2010.

[42] E. Humenay, D. Tarjan, and K. Skadron. Impact of process variations on multicore performance symmetry. In *Design, Automation Test in Europe Conference Exhibition, 2007. DATE '07*, pages 1 –6, april 2007.

[43] M. Kandemir, O. Ozturk, and S.P. Muralidhara. Dynamic thread and data mapping for NoC based CMPs. In *Proceedings of the Design Automation Conference*, pages 852 –857, July 2009.

[44] Uksong Kang, Hoe-Ju Chung, Seongmoo Heo, Duk-Ha Park, Hoon Lee, Jin Ho Kim, Soon-Hong Ahn, Soo-Ho Cha, Jaesung Ahn, DukMin Kwon, Jae-Wook Lee, Han-Sung Joo, Woo-Seop Kim, Dong Hyeon Jang, Nam Seog Kim, Jung-Hwan Choi, Tae-Gyeong Chung, Jei-Hwan Yoo, Joo Sun Choi, Changhyun Kim, and Young-Hyun Jun. 8 gb 3-d ddr3 dram using through-silicon-via technology. *Solid-State Circuits, IEEE Journal of*, 45(1):111 –119, jan. 2010.

[45] C. Kenyon, A. Kornfeld, K. Kuhn, M. Liu, A. Maheshwari, W. Shih, S. Sivakumar, G. Taylor, P. VanDerVoorn, and K. Zawadzki. Managing process variation in intel's 45nm CMOS technology. *Intel Technology Journal. http://www.intel.com/technology/itj/2008/v12i2/3-managing/1-abstract.htm*, June 2008.

[46] Dae Hyun Kim and Sung Kyu Lim. Through-silicon-via-aware delay and power prediction model for buffered interconnects in 3d ics. In *Proceedings of the 12th ACM/IEEE international workshop on System level interconnect prediction*, SLIP '10, pages 25–32, New York, NY, USA, 2010. ACM.

[47] Jongman Kim, Chrysostomos Nicopoulos, Dongkook Park, Reetuparna Das, Yuan Xie, Vijaykrishnan Narayanan, Mazin S. Yousif, and Chita R. Das. A novel dimensionally-decomposed router for on-chip communication in 3d architectures. *SIGARCH Comput. Archit. News*, 35:138–149, June 2007.

[48] S. Kottapalli and Jeff Baxter. Nehalem-ex cpu architecture. *Hot chips 21.*

[49] S. Lakshminarayanan, P. J. Wright, and J. Pallinti. Electrical characterization of the copper CMP process and derivation of metal layout rules. *IEEE Transactions on Semiconductor Manufacturing*, 16(4):668 – 676, November 2003.

[50] J. Laudon and D. Lenoski. The sgi origin: A ccnuma highly scalable server. In *Computer Architecture, 1997. Conference Proceedings. The 24th Annual International Symposium on*, pages 241 –251, jun 1997.

[51] S. Li, J. H. Ahn, R. D. Strong, J. B. Brockman, D. M. Tullsen, and N. P. Jouppi. McPAT: An integrated power, area, and timing modeling framework for multicore and manycore architectures. In *Proceedings of the International Symposium on Microarchitecture*, pages 469 –480, December 2009.

[52] The Nangate Open Cell Library. 45nm FreePDK. Available at `https://www.si2.org/openeda.si2.org/projects/nangatelib/`.

[53] N. Linial and M. Tarsi. Interpolation between bases and the shuffle exchange network. *Eur. J. Comb.*, 10:29–39, February 1989.

[54] C. Liu, A. Sivasubramaniam, and M. Kandemir. Organizing the last line of defense before hitting the memory wall for CMPs. In *Proceedgins of the 10th International Symposium on High Performance Computer Architecture*, pages 176 – 185, February 2004.

[55] I. Loi, S. Mitra, T.H. Lee, S. Fujita, and L. Benini. A low-overhead fault tolerance scheme for tsv-based 3d network on chip links. In *Computer-Aided Design, 2008. ICCAD 2008. IEEE/ACM International Conference on*, pages 598 –602, nov. 2008.

[56] Xiang Lu, Zhuo Li, Wangqi Qiu, D.M.H. Walker, and Welping Shi. Parade: parametric delay evaluation under process variation [ic modeling]. In *Quality Electronic Design, 2004. Proceedings. 5th International Symposium on*, pages 276 – 280, 2004.

[57] D. Ludovici. Technology aware network-on-chip connectivity and synchronization design, 2011.

[58] Grigorios Magklis, Greg Semeraro, David H. Albonesi, Steven G. Dropsho, Sandhya Dwarkadas, and Michael L. Scott. Dynamic frequency and voltage scaling for a multiple-clock-domain microprocessor. *IEEE Micro*, 23:62–68, 2003.

[59] P.S. Magnusson, M. Christensson, J. Eskilson, D. Forsgren, G. Hallberg, J. Hogberg, F. Larsson, A. Moestedt, and B. Werner. Simics: A full system simulation platform. *Computer*, 35(2):50 –58, feb 2002.

[60] S.S. Majzoub, R.A. Saleh, S.J.E. Wilton, and R.K. Ward. Energy optimization for many-core platforms: Communication and pvt aware voltage-island formation and voltage selection algorithm. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 29(5):816 –829, may 2010.

[61] I. Miro Panades and A. Greiner. Bi-synchronous FIFO for synchronous circuit communication well suited for network-on-chip in GALS architectures. In *Proceedings of the First International Symposium on Networks-on-Chip*, pages 83 –94, May 2007.

[62] M. Mondal, T. Ragheb, X. Wu, A. Aziz, and Y. Massoud. Provisioning on-chip networks under buffered RC interconnect delay variations. In *Proceedings of the 8th International Symposium on Quality Electronic Design*, pages 873 –878, March 2007.

[63] M. Mondal, T. Ragheb, Xiang Wu, A. Aziz, and Y. Massoud. Provisioning on-chip networks under buffered rc interconnect delay variations. In *Quality Electronic Design, 2007. ISQED '07. 8th International Symposium on*, pages 873 –878, march 2007.

[64] S. Murali, R. Tamhankar, F. Angiolini, A. Pulling, D. Atienza, L. Benini, and G. De Micheli. Comparison of a timing-error tolerant scheme with a traditional re-transmission mechanism for networks on chips. In *System-on-Chip, 2006. International Symposium on*, pages 1 –4, nov. 2006.

[65] S. Murali, R. Tamhankar, F. Angiolini, A. Pulling, D. Atienza, L. Benini, and G. De Micheli. Comparison of a timing-error tolerant scheme with a

traditional re-transmission mechanism for networks on chips. In *System-on-Chip, 2006. International Symposium on*, pages 1 –4, nov. 2006.

[66] C. A. Nicopoulos, S. Srinivasan, A. Yanamandra, D. Park, V. Narayanan, C. Das, and M. Irwin. On the effects of process variation in network-on-chip architectures. *IEEE Transactions on Dependable and Secure Computing*, PP(99):1 –1, 2010.

[67] Koushik Niyogi and Diana Marculescu. Speed and voltage selection for gals systems based on voltage/frequency islands. In *Proceedings of the 2005 Asia and South Pacific Design Automation Conference*, ASP-DAC '05, pages 292–297, New York, NY, USA, 2005. ACM.

[68] Synopsys PrimeTime VX Application Note. Implementation methodology with variation-aware timing analysis. Technical report, 2007.

[69] U. Y. Ogras, R. Marculescu, P. Choudhary, and D. Marculescu. Voltage-frequency island partitioning for GALS-based networks-on-chip. In *Proceedings of the Design Automation Conference*, pages 110 –115, June 2007.

[70] Umit Y. Ogras, Radu Marculescu, Puru Choudhary, and Diana Marculescu. Voltage-frequency island partitioning for gals-based networks-on-chip. In *Proceedings of the 44th annual Design Automation Conference*, DAC '07, pages 110–115, New York, NY, USA, 2007. ACM.

[71] M. Orshansky, S. Nassif, and D. Boning. *Design for Manufacturability and Statistical Design: A Comprehensive Approach.* Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[72] Michael Orshansky and Kurt Keutzer. A general probabilistic framework for worst case timing analysis. In *Proceedings of the 39th annual Design Automation Conference*, DAC '02, pages 556–561, New York, NY, USA, 2002. ACM.

[73] Michael Orshansky, Sani Nassif, and Duane Boning. *Design for Manufacturability and Statistical Design: A Comprehensive Approach.* Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[74] J.D. Owens, W.J. Dally, R. Ho, D.N. Jayasimha, S. W. Keckler, and L. Peh. Research challenges for on-chip interconnection networks. *IEEE Micro*, 27(5):96–108, 2007.

[75] G. Paci, D. Bertozzi, and L. Benini. Effectiveness of adaptive supply voltage and body bias as post-silicon variability compensation techniques for full-swing and low-swing on-chip communication channels. In *Design, Automation Test in Europe Conference Exhibition, 2009. DATE '09.*, pages 1404 –1409, april 2009.

[76] Dongkook Park, S. Eachempati, R. Das, A.K. Mishra, Yuan Xie, N. Vijaykrishnan, and C.R. Das. Mira: A multi-layered on-chip interconnect router architecture. In *Computer Architecture, 2008. ISCA '08. 35th International Symposium on*, pages 251 –261, june 2008.

[77] V. Pasca, L. Anghel, and M. Benabdenbi. Fault tolerant communication in 3d integrated systems. In *Dependable Systems and Networks Workshops (DSN-W), 2010 International Conference on*, pages 131 –135, 28 2010-july 1 2010.

[78] V. Pasca, L. Anghel, C. Rusu, R. Locatelli, and M. Coppola. Error resilience of intra-die and inter-die communication with 3d spidergon stnoc. In *Design, Automation Test in Europe Conference Exhibition (DATE), 2010*, pages 275 –278, march 2010.

[79] S. Pasricha. Exploring serial vertical interconnects for 3d ics. In *Design Automation Conference, 2009. DAC '09. 46th ACM/IEEE*, pages 581 –586, july 2009.

[80] Robert Patti. Impact of wafer-level 3d stacking on the yield of ics, 2007.

[81] Vasilis F. Pavlidis and Eby G. Friedman. *Three-dimensional Integrated Circuit Design.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2008.

[82] Marcel J. M. Pelgrom, Aad C. J. Duinmaijer, and Andanton P. G. Welbers. Matching properties of mos transistors. *IEEE J. Solid-State Circuits*, 24:1433–1440, 1989.

[83] A. Pullini, F. Angiolini, S. Murali, D. Atienza, G. De Micheli, and L. Benini. Bringing nocs to 65 nm. *Micro, IEEE*, 27(5):75 –85, sept.-oct. 2007.

[84] Antonio Pullini, Federico Angiolini, Davide Bertozzi, and Luca Benini. Fault tolerance overhead in network-on-chip flow control schemes. In *Proceedings of the 18th annual symposium on Integrated circuits and system design*, SBCCI '05, pages 224–229, New York, NY, USA, 2005. ACM.

[85] R Development Core Team. *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria, 2006. ISBN 3-900051-07-0.

[86] J. Rattner. Single-chip cloud computer: An experimental many-core processor from intel labs. Available online at http://download.intel.com/pressroom/pdf/rockcreek/SCC_Announcement_JustinRattner.pdf.

[87] Antoni Roca, Jos Flich, Federico Silla, and Jos Duato. A low-latency modular switch for cmp systems. *Microprocessors and Microsystems*, 35(8):742 – 754, 2011. ¡ce:title¿Design and Verification of Complex Digital Systems¡/ce:title¿.

[88] S. Rodrigo, C. Hernández, J. Flich, F. Silla, J. Duato, S. Medardoni, D. Bertozzi, A. Mejía, and D. Dai. Yield-oriented evaluation methodology of network-on-chip routing implementations. In *Proceedings of the 11th international conference on System-on-chip*, SOC'09, pages 100–105, Piscataway, NJ, USA, 2009. IEEE Press.

[89] T. Sakurai and A. R. Newton. Alpha-power law mosfet model and its applications to cmos inverter delay and other formulas. *IEEE Journal of Solid-state Circuits*, 25:584–594, 1990.

[90] T. Sakurai and A.R. Newton. Alpha-power law mosfet model and its applications to cmos inverter delay and other formulas. *Solid-State Circuits, IEEE Journal of*, 25(2):584 –594, apr 1990.

[91] S. R. Sarangi, B. Greskamp, R. Teodorescu, J. Nakano, A. Tiwari, and J. Torrellas. VARIUS: A model of process variation and resulting timing errors for microarchitects. *IEEE Transactions on Semiconductor Manufacturing*, 21(1):3 –13, February 2008.

[92] Smruti R. Sarangi, Brian Greskamp, Radu Teodorescu, Jun Nakano, Abhishek Tiwari, and Josep Torrellas. Varius: A model of process variation and resulting timing errors for microarchitects. In *in IEEE Transactions on Semiconductor Manufacturing*, 2008.

[93] Ioannis Savidis, Syed M. Alam, Ankur Jain, Scott Pozder, Robert E. Jones, and Ritwik Chatterjee. Electrical modeling and characterization of through-silicon vias (tsvs) for 3-d integrated circuits. *Microelectron. J.*, 41:9–16, January 2010.

[94] Eung S. Shin, Vincent J. Mooney, III, and George F. Riley. Round-robin arbiter design and generation. In *Proceedings of the 15th international symposium on System Synthesis*, ISSS '02, pages 243–248, New York, NY, USA, 2002. ACM.

[95] Tor Skeie, Frank Olaf Sem-Jacobsen, Samuel Rodrigo, José Flich, Davide Bertozzi, and Simone Medardoni. Flexible dor routing for virtualization of multicore chips. In *Proceedings of the 11th international conference on System-on-chip*, SOC'09, pages 73–76, Piscataway, NJ, USA, 2009. IEEE Press.

[96] Vassos Soteriou, Rohit Sunkam Ramanujam, Bill Lin, and Li-Shiuan Peh. A high-throughput distributed shared-buffer noc router. *IEEE Comput. Archit. Lett.*, 8:21–24, January 2009.

[97] Bonesi Stefano, Davide Bertozzi, Luca Benini, and Enrico Macii. Process variation tolerant pipeline design through a placement-aware multiple voltage island design style. *Design, Automation and Test in Europe Conference and Exhibition*, 0:967–972, 2008.

[98] Stergios Stergiou, Federico Angiolini, Salvatore Carta, Luigi Raffo, Davide Bertozzi, and Giovanni De Micheli. Xpipes lite: A synthesis oriented

design library for networks on chips. In *In DATE*, pages 1188–1193. IEEE, 2005.

[99] STMicroelectronics. *CMP: Circuits Multiprojets*.

[100] R. Teodorescu and J. Torrellas. Variation-aware application scheduling and power management for chip multiprocessors. In *Proceedings of the 35th International Symposium on Computer Architecture*, pages 363 – 374, June 2008.

[101] TexasInstruments. The ttl data book for design engineers. 1981.

[102] TILERA. TILE-Gx processors family. Available at `http://www.tilera.com/products/TILE-Gx.php`.

[103] Faiz ul Hassan, B. Cheng, W. Vanderbauwhede, and F. Rodriguez. Impact of device variability in the communication structures for future synchronous soc designs. In *System-on-Chip, 2009. SOC 2009. International Symposium on*, pages 068 –072, oct. 2009.

[104] Arizona State University. Predictive technology model. Available at `http://ptm.asu.edu`.

[105] Berkeley University. Bsim4 6.4 mosfet manual. Available at `http://www-device.eecs.berkeley.edu/\~bsim3/BSIM4//BSIM464/BSIM464\_Manual1.pdf`.

[106] Feng Wang, C. Nicopoulos, Xiaoxia Wu, Yuan Xie, and N. Vijaykrishnan. Variation-aware task allocation and scheduling for mpsoc. In *Proceedings of the 2007 IEEE/ACM international conference on Computer-aided design*, ICCAD '07, pages 598–603, Piscataway, NJ, USA, 2007. IEEE Press.

[107] S. Wong, T. G. Lee, D. Ma, and C. Chao. An empirical three-dimensional crossover capacitance model for multilevel interconnect VLSI circuits. *IEEE Transactions on Semiconductor Manufacturing*, 13(2):219 –227, May 2000.

[108] F. Worm, P. Ienne, P. Thiran, and G. De Micheli. A robust self-calibrating transmission scheme for on-chip networks. *IEEE Trans. Very Large Scale Integr. Syst.*, 13(1):126–139, 2005.

[109] M. Zhang and K. Asanovic. Victim replication: maximizing capacity while hiding wire delay in tiled chip multiprocessors. In *Proceedings of the 32nd International Symposium on Computer Architecture*, pages 336–345, June 2005.

[110] L. Zhao, R. Iyer, S. Makineni, J. Moses, R. Illikkal, and D. Newell. Performance, area and bandwidth implications on large-scale cmp cache design. In *Proceedings of the Workshop on Chip Multiprocessor Memory Systems and Interconnects*, 2007.