Mohammad-Hashem Haghbayan

Energy-Efficient and Reliable Computing in Dark Silicon Era

Turku Centre *for* Computer Science

TUCS Dissertations
No 228, December 2017

Energy-Efficient and Reliable Computing in Dark Silicon Era

Mohammad-Hashem Haghbayan

University of Turku
Department of Future Technologies
20014 TURUN YLIOPISTO

2017

Supervisors

Professor Pasi Liljeberg
Department of Future Technologies, University of Turku
Finland

Adjunct Professor Amir M. Rahmani
Department of Future Technologies, University of Turku, Finland
Marie Curie Global Fellow,
University of California Irvine, USA and TU Wien, Austria

Adjunct Professor Juha Plosila
Department of Future Technologies, University of Turku
Finland

Professor Hannu Tenhunen
Department of Future Technologies, University of Turku
Finland

Reviewers

Professor Jari Nurmi
Department of Electronics and Communications Engineering
Tampere University of Technology
Tampere, Finland

Professor Ian G. Harris
Department of Computer Science
University of California Irvine, CA, USA

Opponent

Professor Peeter Ellervee
Department of Computer Engineering
Tallinn University of Technology, Tallinn, Estonia

ISBN 978-952-12-3646-4
ISSN 1239-1883

The originality of this thesis has been checked in accordance with the University of Turku quality assurance system using the Turnitin Originality Check service.

In memory of my uncles
Ali-Akbar and Mohammad-Hashem Haghbayan
Who died for what they believed in Freedom

Abstract

Dark silicon denotes the phenomenon that, due to thermal and power constraints, the fraction of transistors that can operate at full frequency is decreasing in each technology generation. Moore's law and Dennard scaling had been backed and coupled appropriately for five decades to bring commensurate exponential performance via single core and later muti-core design. However, recalculating Dennard scaling for recent small technology sizes shows that current ongoing multi-core growth is demanding exponential thermal design power to achieve linear performance increase. This process hits a power wall where raises the amount of dark or dim silicon on future multi/many-core chips more and more. Furthermore, from another perspective, by increasing the number of transistors on the area of a single chip and susceptibility to internal defects alongside aging phenomena, which also is exacerbated by high chip thermal density, monitoring and managing the chip reliability before and after its activation is becoming a necessity. The proposed approaches and experimental investigations in this thesis focus on two main tracks: 1) power awareness and 2) reliability awareness in dark silicon era, where later these two tracks will combine together. In the first track, the main goal is to increase the level of returns in terms of main important features in chip design, such as performance and throughput, while maximum power limit is honored. In fact, we show that by managing the power while having dark silicon, all the traditional benefits that could be achieved by proceeding in Moore's law can be also achieved in the dark silicon era, however, with a lower amount. Via the track of reliability awareness in dark silicon era, we show that dark silicon can be considered as an opportunity to be exploited for different instances of benefits, namely life-time increase and online testing. We discuss how dark silicon can be exploited to guarantee the system lifetime to be above a certain target value and, furthermore, how dark silicon can be exploited to apply low cost non-intrusive online testing on the cores. After the demonstration of power and reliability awareness while having dark silicon, two approaches will be discussed as the case study where the power and reliability awareness are combined together. The first approach demonstrates how chip reliability can be used as a supplementary metric for power-reliability management. While the second approach provides a trade-off between workload performance and system reliability by simultaneously honoring the given power budget and target reliability.

Acknowledgements

I would like to express my special appreciation and thanks to my advisers adjunct prof. Amir M. Rahmani, prof. Pasi Liljeberg, adjunct prof. Juha Plosila and prof. Hannu Tenhonen. I want to specially thank Dr. Rahmani for his comprehensive support when I entered the University of Turku. In particular he helped me with transferring his knowledge and experiences to me as much as he could. I want to additionally thank prof. Liljeberg and Dr. Plosila for helping me to cope with the problems I encountered during my PhD.

I want to thank Mr. Mohammad Fattah for generously giving me his helpful Noxim source code which has been the baseline of my research during this four years. I want to thank Dr. Antonio Miele for long-term collaboration and friendship I had with him, for his hospitality during my visit at Politecnic di Milano University, and also for his helpful revision comments on this dissertation. I want to thank Mr. Anil Kanduri to show me the real challenge in the field of research and science. I want to thank Mrs. Sanaz Rahimi Moosavi for her friendly accompaniment and helpful advises during my PhD.

I would also like to thank professor Jari Nurmi and professor Ian G. Harris for accepting to be the reviewer of my thesis, and also for considering my thesis as an honored dissertation. I also thank professor Peeter Ellervee for accepting to be the opponent of my thesis.

I want to thank my family in Iran Zohreh, Fahimeh, Akbar, Tahmoures, my father Bahman, and my mother Tahereh, and my family in Finland Eeva, Sampo, and Arvi(n) for their understanding and support during my PhD and for teaching me how to live in this world, and, finally, I want to dedicate my thesis to my uncles Mohammad-Hashem Haghbayan and Ali-Akbar Haghbayan.

List of original publications

The work discussed in this dissertation is based on the publications listed below:

Paper I **M.H. Haghbayan**, A. Kanduri, A.M. Rahmani, P. Liljeberg, A. Jantsch, H. Tenhunen, "MapPro: Proactive Runtime Mapping for Dynamic Workloads by Quantifying Ripple Effect of Applications on Networks-on-Chip," in IEEE/ACM International Symposium on Networks-on-Chip (NOCS 2015), Canada.

Paper II **M.H. Haghbayan**, A.M. Rahmani, A. Yemane, P. Liljeberg, J. Plosila, A. Jantsch, H. Tenhunen, "Dark Silicon Aware Power Management for Manycore Systems under Dynamic Workloads," in IEEE/ACM The 32nd IEEE/ACM International Conference on Computer Design (ICCD 2014), Korea.

Paper III A.M. Rahmani, **M.H. Haghbayan**, A. Kanduri, A. Yemane, P. Liljeberg, J. Plosila, A. Jantsch, H. Tenhunen, "Dynamic Power Management for Many-Core Platforms in the Dark Silicon Era: A Multi-Objective Control Approach," in IEEE/ACM International Symposium on Low Power Electronics and Design, (ISLPED 2015), Italy.

Paper IV A.M. Rahmani, **M.H. Haghbayan**, A. Miele, P. Liljeberg, A. Jantsch, H. Tenhunen, "Reliability-Aware Runtime Power Management for Many-Core Systems in the Dark Silicon Era," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, (IEEE-TVLSI 2017).

Paper V A. Kanduri, **M.H. Haghbayan**, A.M. Rahmani, P. Liljeberg, A. Jantsch, H. Tenhunen, "Dark Silicon Aware Runtime Mapping for Many-core Systems: A Patterning Approach," in IEEE/ACM International Conference on Computer Design, (ICCD 2015), USA.

Paper VI **M.H. Haghbayan**, A. Miele, A.M. Rahmani, P. Liljeberg, A. Jantsch, C. Bolchini, H. Tenhunen, "Can Dark Silicon Be Exploited to Prolong System Lifetime?," in IEEE Design and Test of Computers (IEEE-D&T), 2017.

Paper VII S. Sami Teräväinen, **M.H. Haghbayan**, A.M. Rahmani, P. Liljeberg, H. Tenhunen, "Software-Based On-Chip Thermal Sensor Calibration for DVFS-enabled Many-core Systems," in IEEE Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT 2015), USA.

Paper VIII **M.H. Haghbayan**, A. Miele, A.M. Rahmani, P. Liljeberg, H. Tenhunen, "A Lifetime-Aware Runtime Mapping Approach for Many-core Systems in the Dark Silicon Era," in IEEE/ACM Design, Automation, and Test in Europe, (DATE 2016), Germany.

Paper IX **M.H. Haghbayan**, A.M. Rahmani, P. Liljeberg, J. Plosila, H. Tenhunen, "Energy-Efficient Concurrent Testing Approach for Many-Core Systems in the Dark Silicon Age," in IEEE Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT 2014), Netherlands.

Paper X **M.H. Haghbayan**, A.M. Rahmani, M. Fattah, P. Liljeberg, J. Plosila, Z. Navabi, H. Tenhunen, "Power-Aware Online Testing of Manycore Systems in the Dark Silicon Era," in IEEE/ACM the Design, Automation, and Test in Europe (DATE 2015), France.

Paper XI **M.H. Haghbayan**, A.M. Rahmani, A. Miele, M. Fattah, J. Plosila, P. Liljeberg, H. Tenhunen, "A Power-Aware Approach for Online Test Scheduling in Many-core Architectures," in IEEE Transactions on Computers, (IEEE-TC 2016).

Paper XII **M.H. Haghbayan**, A. Miele, A.M. Rahmani, P. Liljeberg, H. Tenhunen, "Performance/Reliability-aware Resource Management for Many-Cores in Dark Silicon Era," in IEEE Transactions on Computers, (IEEE-TC 2017).

Contents

Part I

Research Summary

Chapter 1

Introduction

The continuous cramming of transistors onto a chip area which is known as Moore's law has become the feedstock of the exuberant novelty in computer architecture and design. According to Moore's Law [Moore, 1965] every 18 months the number of transistors doubles in a fixed chip area. Such increase in the number of rock bottom elements of computation on a chip led to emergence of hardware redundancies in processor architectures mainly to allow performance increase e.g., instruction level parallelism, branch prediction, and other well known techniques [Hennessy and Patterson, 2012]. By encountering limitations to increase clock frequency more than up to 3-4 GHz [Danowitz et al., 2012], multi-core design emerged allowing chip performance to increase by parallel operation of multiple integrated cores Later, due to the increasing number of cores and restrictions in memory access, i.e., the memory wall [Wulf and McKee, 1995], multiple memory buses with non uniform memory access (NUMA) emerged which increased the bandwidth of data transfer between the cores and memories. NUMA architecture combined with on-chip network based interconnection (NoC) [Rahmani, 2013] outperforms bus-based system in terms of scalability, flexibility, and performance.

Theoretically, it can be proven that scaling down the transistor size and simultaneously decreasing the operating voltage keeps the power density [1] of the chip constant. This is known as *Dennard scaling* [Dennard et al., 1974]. Moore's law and Dennard scaling allowed integration of an increasing number of computing units on a chip without any limitations and resulted in exponential commensurate benefits. However, in the recent technology nodes, the reduction of the supply voltage does not anymore coincide with the technology size reduction, because the voltage scaling is limited by the approaching threshold voltage, i.e. the minimum voltage needed to turn on transistors [Rahmani et al., 2016]. Due to this phenomenon, and also due to the increase in leakage power of transistors, power density is actually increasing when Moore's law continues to hold. This poses a se-

[1] Power consumption per unit of area

rious threat to future-generation chip multiprocessors. High power density causes more heat generation that needs to be dissipated via different cooling methods. Otherwise high temperature may affect the functionality of the chip causing less reliable computation and even burning of the chip. Therefore, to keep the power density at a tolerable level, some parts of the chip need to be kept inactive. Such inactive parts are called *Dark Silicon* [Esmaeilzadeh et al., 2012]. Since the values of the operating voltage and frequency (VF) on the chip are the main contributors of power consumption, dark silicon phenomena is alongside a key technological problem of the targeted multi-core system, known as the *utilization wall*. The *utilization wall* means that with each technology process generation, the percentage of transistors that a chip design can switch at full frequency drops exponentially because of power constraints [Zhang et al., 2013]. Figure 1.1 illustrates the projection of $Dennardian$ and $postDennard$ eras and also drafts the percentage of the dark silicon as the CMOS technology scales. According to a prediction, designers will face more than 90% dark silicon within 6 years if they do not properly attack this phenomenon [Taylor, 2012]. A direct consequence of this is large swaths of a chip's silicon area that must remain mostly passive to stay within the chip's power budget. Currently, only about 1 percent of a modest-sized 32-nm mobile chip can switch at full frequency within a 3-W power budget. At 22-nm, 21% of a fixed-size chip must be powered off, and at 8-nm, this number grows to more than 50%. With each process generation, dark silicon gets exponentially cheaper, whereas the power budget is becoming exponentially more valuable. [Venkatesh et al., 2010].

From another perspective, aggressive advances in integrated circuit manufacturing process and downscaling of CMOS technologies has had a negative effect on the reliability of the devices and aging phenomena. Increased aging mechanisms cause performance degradation and eventual device and system failures. Shrinking the transistor size and dark silicon phenomena have exacerbated such a trend, since they have increased power densities in the device, and consequently the operating temperatures, which are the main cause of aging phenomena. Workload variations and dynamic power management techniques highly contribute to device temperature variations. Aging mechanisms including time dependent dielectric breakdown (TDDB), negative bias temperature instability (NBTI), and electromigration (EM) are among the most increasingly adverse factors that can lead to delay errors and device breakdowns. International Technology Roadmap for Semiconductors (ITRS [Semiconductor-Industry-Association et al., 2011]) recognizes that reliability is becoming a primary design concern in current integrated circuits.

The first naïve idea that might cross the mind to tackle the raised challenges in the dark silicon era is to not to fabricate extra hardware any more. If we cannot use some parts of the chip because of too much power consumption, then why we should fabricate extra hardware? On the opposite, what if we can envision some benefits to fabricate extra hardware in the dark silicon era? To address the latter question following challenges raise:

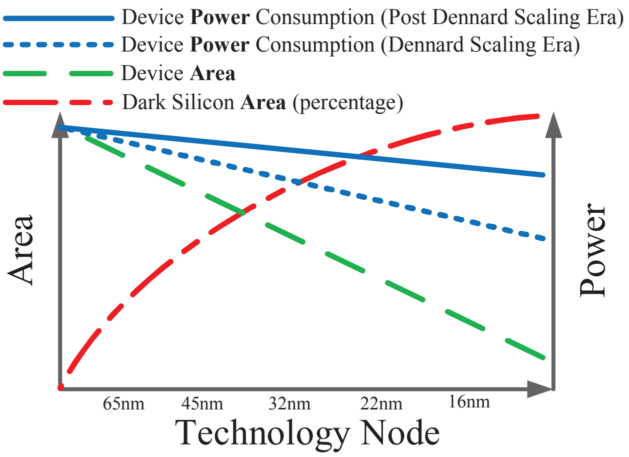

Figure 1.1: Chip area, dark area, and power budget trends with technology scaling

- Is it possible to increase the level of returns in terms of e.g., performance and reliability, while considering chip power consumption with relatively small amount of cost? i.e., *dark silicon aware power management*.
- Is it possible to use dark silicon and drive benefit from it to fulfill other important constraints in hardware design and test such as improving reliability or online testing? i.e., *dark silicon aware reliability management*.

This thesis is an investigation to answer the above questions by considering different existing aspects and constraints in hardware design and computer architecture. Figure 1.2 shows the general overview of the works accomplished in this thesis and cohesion of the publications while dealing with dark silicon phenomena. The highlighted titles in each box are the publications that are fully included in this manuscript and convey the main idea and obtained results in this dissertation. As can be seen, two main tracks to tackle dark silicon phenomena are investigated that are $power$ and $reliability$ management while considering dark silicon.

Generally, our target platform in this thesis is a many-core system which is a special kind of multi-core system for a high degree of parallel processing, containing a large number of independent processor cores. Applications that must be executed on such a system contain several tasks each of which must be executed by one core. The tasks of an application are generally a single function/portion of code requiring specific input data, provided by precedent tasks of the application, and producing specific output data, transmitted to the subsequent tasks. To run an application, tasks have to be dispatched on the grid of processing cores and each task is $mapped$ on a single idle core. In a many-core system with dark silicon, cores that are executing tasks are turned on while other idle cores remain inactive, i.e, dark. It is not essentially important which cores should be active or dark, but the total amount of power consumed by active cores is limited and must not ex-

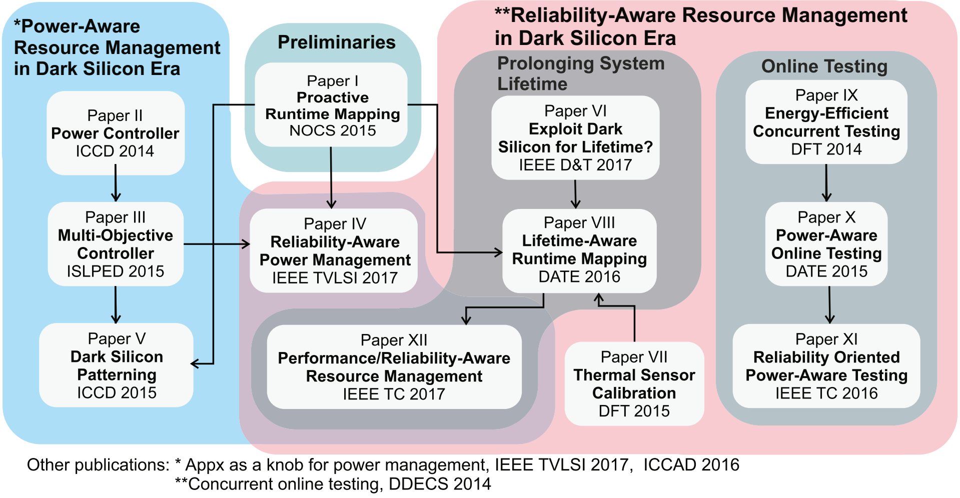
*Power-Aware Resource Management in Dark Silicon Era
Paper II Power Controller ICCD 2014
Paper III Multi-Objective Controller ISLPED 2015
Paper V Dark Silicon Patterning ICCD 2015
Preliminaries
Paper I Proactive Runtime Mapping NOCS 2015
Paper IV Reliability-Aware Power Management IEEE TVLSI 2017
Paper XII Performance/Reliability-Aware Resource Management IEEE TC 2017
**Reliability-Aware Resource Management in Dark Silicon Era
Prolonging System Lifetime
Paper VI Exploit Dark Silicon for Lifetime? IEEE D&T 2017
Paper VIII Lifetime-Aware Runtime Mapping DATE 2016
Paper VII Thermal Sensor Calibration DFT 2015
Online Testing
Paper IX Energy-Efficient Concurrent Testing DFT 2014
Paper X Power-Aware Online Testing DATE 2015
Paper XI Reliability Oriented Power-Aware Testing IEEE TC 2016
Other publications: * Appx as a knob for power management, IEEE TVLSI 2017, ICCAD 2016
**Concurrent online testing, DDECS 2014

Figure 1.2: Illustration of paper cohesion. Each category is labeled with different colors.

ceed an upper bound. This upper bound is called thermal design power (TDP). The first step for considering dark silicon in power and/or reliability management is to design a dark silicon-aware application mapping which copes with application performance requirements while considering dark silicon at each moment. Such a mapping approach for many-core systems, which is discussed in Paper I in the "Preliminaries" box of Figure 1.2, is a substrate to implement our subsequent proposed approaches.

For dark silicon aware power management, mixed operation mode design is used to increase the system's performance as much as possible while system's overall instantaneous power is kept below TDP. To do this, we attempt to decrease the application's execution power via different controlling techniques using dynamic voltage and frequency scaling (DVFS), per-core power gating (PCPG), or applying approximation techniques. However, applying such techniques must be done in an intelligent and careful manner since a blind use of these features might incur negative impact on other system constraints such as core's lifetime or even the overall system performance. Based on this, an agile online monitoring of the system's power and network characteristics is proposed and discussed in Paper II to observe instantaneous system's behaviour and react based on that. Later, in Paper III the power management unit is enhanced to execute in a multi-objective manner, i.e., energy and performance centric manner, while considering wider characteristics of the many-core systems, such as dynamic behaviour of workloads, processing elements utilization, per-core power consumption, and load on network-on-chip. Thermal Design Power (TDP) is a standard design time metric that has been used to determine a safer upper bound on chip's power consumption. Safer operation of a chip is guaranteed as long as power consumption stays within TDP. TDP is a single fixed upper bound that is pessimistically estimated and assuming that all neighbor cores are active and are operating at the worst case of voltage and frequency. While having dark silicon on the chip such assumption is too pessimistic since neighbor cores are not necessarily active and might be mapped sporadically. Hence, alternatively dynamic upper bound for power budget can be used rather than TDP. This more flexible upper bound is called the thermal safe power (TSP). By using TSP rather than TDP, active cores can be patterned alongside inactive cores to evenly distribute temperature and power density on the chip and enhance the utilization. In Paper V, by using TSP as the controller reference accompanying with the patterning strategy in the mapping unit, dark silicon is leveraged to increase the maximum power upper bound and resource utilization.

From another aspect dark silicon can be envisioned as an opportunity to be exploited for enhancing system reliability, particularly lifetime increase and online testing. In a many-core system that is suffering from dark silicon phenomenon, a highly dynamic heterogeneous power distribution is being made while applications enter and leave the system at runtime. This power distribution heterogeneity, including dark/dim cores, is both dimensional and temporal which usually appears in an unplanned manner. By analyzing and directing this power distribution on the chip, there are significant opportunities for the designer to improve system lifetime or to apply non-intrusive online testing. The direction of the power distribution can be performed in different layers from hardware to operating system and via different features such as runtime mapping or power management. Two important case studies that are investigated for dark silicon aware reliability management and are shown in Figure 1.2 are 1) prolonging system lifetime and 2) online testing.

Prolonging system lifetime: extreme downsizing of CMOS technology accompanied by high power density and temperature of recent chips has caused an acceleration in the device aging and wearout process. The main reason for the devices to be more susceptible for aging is the thermal issue. In dark silicon, thermal issues cause only short term problems, but in aging phenomena, long-term temperature experience causes the system to age faster. Our general investigations in Paper VI on recent many-core systems with power management control units show that dark silicon can be considered as an opportunity to be exploited for the sake of prolonging lifetime. To do this, it has been shown that with dark/dim cores on chip, considering the system lifetime in a runtime resource management unit and/or even in a dark silicon management unit can improve the system lifetime without much negative impact on other system constraints like performance. Calculating current lifetime on a chip highly depends on the chip's temperature footprint in its activity period. In recent modern chips, multiple thermal sensors are deployed on the chip area that can be calibrated and used at runtime to provide such a temperature footprint. In Paper VII a low overhead software based thermal sensor calibration for Intel SCC platform, which is one of the few available real platforms for many-core systems, is discussed. Based on lifetime information extracted from the system and by analyzing the thermal sensors' data in long term, a reliability-aware resource management in many-core systems is proposed and discussed in Paper VIII. The approach is based on a hierarchical architecture, composed of a long-term runtime reliability analysis unit based on the temperature footprint extracted from thermal sensors and a short-term runtime mapping unit. The former periodically analyses the aging status of the various processing units with respect to a target value specified by the designer, and performs recovery actions on highly stressed cores.

Online testing: by decreasing the transistor size and increasing the acceptability to internal defects, online testing of the fabricated chips to diagnose permanent faults has become a necessity. With dark silicon on the chip, there is the opportunity to detect permanent faults by performing online testing on the dark cores in runtime and not highly penalizing the overall system performance. With a small amount of dedicated power budget for testing, in Paper IX, an online concurrent test scheduling approach is proposed which enables test routines to be applied on the fraction of chip that cannot be utilized. Later, in Paper X, varieties of scenarios have been investigated where simultaneous occurrence of dark silicon patterns and unused available power budget bring opportunities to perform online testing without any need to dedicate certain amount of power for test. In addition, a test-aware utilization-oriented runtime mapping technique is proposed which considers the utilization of cores and their test criticality during the process of resource allocation for new applications. This technique directs more critical cores to be idle (dark) to be tested in the near future. In Paper XI, fine grained reliability estimation of the cores is employed for tuning the test scheduling and avoiding over testing. Furthermore, since permanent faults differently manifest themselves in different voltage domains, the proposed power-aware online testing is improved to apply test routines in different voltage levels.

In the final part in this thesis power awareness and reliability awareness in dark silicon are combined together that is discussed based on Paper IV and Paper XII. In the first work that is according to Paper IV, the effect of power management on system lifetime has been investigated and power controller has been enhanced to consider core's lifetime while managing the chip's power consumption. It has been shown that the proposed technique not only is effective in honoring the power budget while considerably boosting the system throughput, but also increases the overall system lifetime by means of power consumption balancing among cores. In the second work that is according to Paper XII reliability/performance-aware resource co-management for many-core architectures are discussed. Here, core's aging status is evaluated by comparing against a target reference specified by the designers, and power capping is used in an intelligent manner to cooperate with the resource management unit on behalf of the recovery process of the stressed cores. It should be noted that, unlike the reliability constraint power management that is discussed in Paper IV, in which reliability was a minor constraint in power management, in Paper XII the life time is a target requirement thus leading to a totally different formulation and implementation.

Chapter 2

Preliminaries

2.1 Many-core Model

The target platform in our work is the modern many-core architecture, such as the Intel single-chip cloud computer (SCC) [Howard et al., 2010], the Kalray massively parallel processor array (MPPA) many-core [Kalray, 2017], or the Adapteva Epiphany [Adapteva, 2017]. All these platforms present a similar non-uniform memory access (NUMA) architecture, shown in Figure 2.1, consisting of a 2D mesh of homogeneous processing nodes interconnected via a network-on-chip (NoC) infrastructure. In the specific model we consider, as in [Rahmani et al., 2016, Carvalho et al., 2007, Fattah et al., 2013], that each node (or core) contains a single processor provided with private instruction and data memories and a NoC network interface. The platform is also connected to a host machine, controlling all its activities. For instance, in Intel SCC the Management console personal computer (MCPC) manages the 48-core system via PCI-Express [Howard et al., 2010].

Many-core architectures are generally employed in High Performance Computing (HPC) and embedded HPC scenarios to accelerate computationally inten-

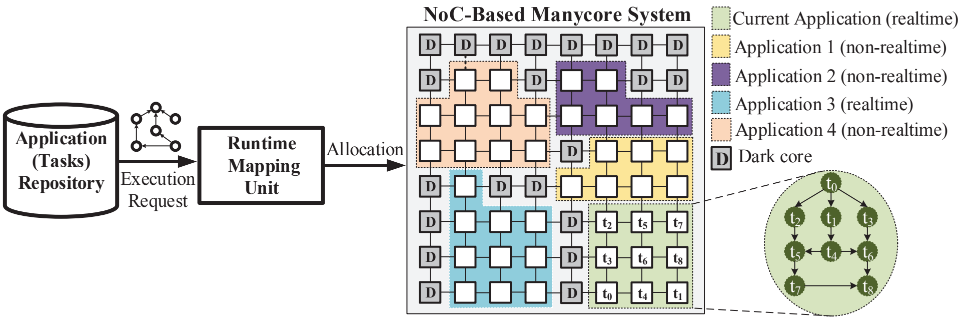
Application (Tasks) Repository
Execution Request
Runtime Mapping Unit
Allocation
NoC-Based Manycore System
Current Application (realtime)
Application 1 (non-realtime)
Application 2 (non-realtime)
Application 3 (realtime)
Application 4 (non-realtime)
D Dark core
D
t0 t1 t2 t3 t4 t5 t6 t7 t8

Figure 2.1: Mesh-based platform with an application mapped onto it (the highlighted region.) where some cores are dark (D)

sive applications such as image, video, or streaming processing. Some interesting use cases discussed in [Kalray, 2017] are the autonomous driving and cryptography acceleration. The commonly-adopted programming model is the dataflow one (as reported [Howard et al., 2010, Adapteva, 2017] for Intel SCC and Adapteva Epiphany) that represents the application through a direct acyclic task graph [Carvalho et al., 2007, Fattah et al., 2013, Haghbayan et al., 2015a], as shown in the bottom-right part of Figure 2.1.

2.2 Application Model

As discussed before, the applications in our platform are modeled as an ensemble of tasks. Each task is a single function/portion of code requiring specific input data, provided by precedent tasks, and producing specific output data, transmitted to the subsequent tasks, as described by the edges in the graph. Each application in the system is represented by a directed graph denoted as a task graph $Ap = TG(T, E)$. Each vertex $t_i \in T$ represents one task of the application, while the edge $e_{i,j} \in E$ stands for a communication between the source task t_i, and the destination task t_j [Fattah et al., 2014]. Task graph of an application extracted using TGG [TGG, 2017] is shown in bottom-right part of Figure 2.1.

An architecture graph $AG(N, L)$ describes the communication infrastructure of the processing elements. We consider a 2D mesh NoC (Figure 2.1) with XY deterministic wormhole routing. The AG graph contains a set of nodes $n_{w,h} \in N$, connected together through unidirectional links $l_k \in L$. Each node is the combination of a PE connected to a router.

We define a non-real-time task as 3-tuple $tnr_i = \langle id_i, ex_i, pr_i \rangle$, and a real-time task as 5-tuple $tr_i = \langle p_i, id_i, ex_i, d_i, pr_i \rangle$, where: id_i stands for the task identification, p_i represents period of task *i*, ex_i represents the task execution time, d_i is its deadline, and pr_i denotes the task priority. We define an abstract time unit, called as tick (e.g. 1 ms) [Fattah et al., 2014].

We define an application as a set of tasks having inter-dependencies. Therefore, application mapping is a one-to-many function. We use a simple mathematical model for representing applications running on the system. Hence, no multitasking is assumed in any node. We denote by Application Matrix (AM) the matrix whose entry *(i, j)* $\in [M] \times [N]$ corresponds to the task's application ID running on the tile located in row *i* and column *j* in a mesh-based NoC topology. For example, the following application matrix shows how four applications with IDs from 1 to 4 are mapped onto a 4×4 mesh-based NoC.

$$AM = \begin{bmatrix} 1 & 1 & 4 & 4 \\ 1 & 1 & 3 & 4 \\ 1 & 2 & 3 & 3 \\ 2 & 2 & 2 & 3 \end{bmatrix} \tag{2.1}$$

2.3 Runtime Mapping

Many-core architectures work in high evolving working scenarios with applications entering and leaving the system with an unknown trend. Nevertheless, applications are highly heterogeneous in terms of size and shape of the task graph and may expose Quality of Service (QoS) requirements, expressed in terms of minimum throughput or latency to be satisfied. For this reason, a *Runtime Mapping unit*, a control routine running on the host machine, receives the execution requests of the various users and decides at runtime which group of resources to reserve for each issued application depending on the available units. In case of unavailability of the minimum amount of processing resources, the request is stored in a ready list to be applied later. Runtime Mapping Layer (RTM, [Rahmani et al., 2016]) is loaded on top of the discussed architecture to handle the variable workload and is shown as Runtime Mapping Unit in Figure 2.1. To be executed, an application has to be dispatched on the grid of processing nodes. Each task is *mapped* on a single idle node, i.e., not executing any other task. Hence, no multi-tasking is assumed at node level. In fact, as stated by Intel in 2011 [Howard et al., 2010], given the abundance of cores, a one-to-one mapping may ease the execution management. For similar reasons, task migration is also not supported. This solution has been later confirmed for the subsequent platforms available on the market. Then, the execution model states that a task is run in a non-preemptive way as soon as all predecessor tasks have been completed and input data received. Communication is performed by means of messages passing based on the specific protocol adopted by the NoC infrastructure.

Application mapping is the phase where a tile for a task is chosen in order to maximize the cores' and network's performance while minimizing latency and power consumption. Given the un-predictable nature and sequence of incoming applications [Bogdan et al., 2010] [Bogdan and Marculescu, 2011], mapping has to be performed dynamically rather than at design time [Chou et al., 2008] [Faruque et al., 2007] [Faruque et al., 2008]. With a wide range of applications entering and leaving such a system, runtime application mapping policies become crucial factor in determining the chip's performance, power consumption and reliability [Carvalho et al., 2007]. We consider run-time mapping as one of the first steps in servicing an incoming application as opposed to reactive steps like task migration. Mapping an entire application consisting of several communicating tasks, satisfying power and performance constraints is a complex process. Assuming that there can be other applications running in parallel on the chip, mapping a new application adds to the complexity and consumes more execution time, degrading the expected performance from a parallel system. Finding a preferable region to map an incoming application with least possible overhead is thus important to ensure high performance of the chip. To dominate the complexity of this phase, the runtime mapping (RTM) unit usually acts in two steps: i) *region selection*, that finds a set of neighboring idle cores to be reserved for the new application, and

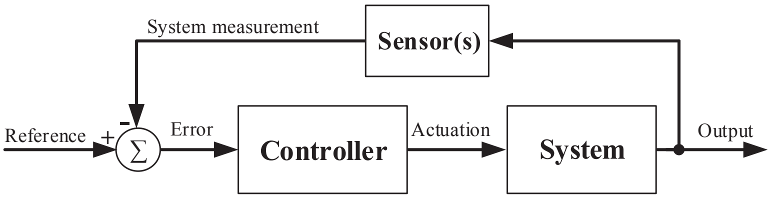

Figure 2.2: structure of the feedback controller

ii) *task mapping*, that dispatches the tasks of a single application onto the selected region. More details about the implementation and results for runtime mapping is discussed in Paper I.

2.4 Feedback Based Controller

In order to implement an efficient dark silicon management approach, an intelligent and stable power administration mechanism using feedback control is required. A general view of a dark silicon aware power management platform is shown in Figure 2.2. As can be seen, a feedback controller using system power measurement is incorporated. Similar to all other control systems, the controller compares the system output with a target value. After comparison, it manipulates the system actuators to minimize the error. The controller policy to tune the actuators strongly depends on the dynamic model of the target system and the system robustness against error disturbance. The dynamic model defines how the system reacts to the inputs including actuations and other inputs. The system robustness is defined as the system stability against overshooting of the output values from the target intended output.

The design of a controller to manage total power consumption highly depends on the system constraints, cost limitation, and available observe/act features. System constraints can be variant system considerations that should be taken into account while manipulating actuators, e.g., network congestion or core stress. Cost limitation determines the complexity level of the controller w.r.t. the requirements of accuracy and speed. For example in case of power violation, how fast and in which accuracy the controller must response as to not damage the chip. Available monitored data and actuators are controller's inputs and outputs respectively and directly shape the controller design. For example in a power controller of many-core system, observation can be from simply tracking total power consumption to monitoring per-tile power, stress, congestion, and injection rate while actuation can range from only PCPG to variant options such as PCPG, DVFS, mapping.

2.5 Reliability Model

According to the definition, the lifetime reliability of a system, $R(t)$, is expressed as the probability that the system has been operational until t. JEDEC Solid State Technology Association [JEDEC Solid State Tech. Association, 2010] expresses the lifetime reliability of a single digital component, such as a processor, by means of the Weibull distribution

$$R(t) = e^{-\left(\frac{t}{\alpha(T)}\right)^{\beta}} \tag{2.2}$$

being t the current instant of time (generally measured in hours), T the constant worst-case processor temperature (Kelvin degrees), β the Weibull slope parameter, and $\alpha(T)$ the scale parameter, or aging rate. The $\alpha(T)$ parameter formulation depends on the considered wear-out mechanisms, that are for instance the electromigration, the hot carrier injection (HCI), or the negative-bias temperature instability (NBTI). If more than one effect is considered, the $\alpha(T)$ formulas are combined according to the sum of failure rate (SOFR) approach. As an example, in the electromigration model, $\alpha(T)$ is modeled according to the Black's equation [Black, 1969]:

$$\alpha(T)_{EM} = \frac{A_0(J - J_{\text{crit}})^{-n} e^{\frac{E_a}{kT}}}{\Gamma\left(1 + \frac{1}{\beta}\right)} \tag{2.3}$$

where A_0 is a process-dependent constant, J is the current density, J_{crit} is the critical current density for the electromigration effect to be triggered (that can be approximated to 0 since $J \gg J_{\text{crit}}$, E_a is the activation energy for electromigration (a constant value), k is the Boltzmann's constant, n is a material-dependent constant, and $\Gamma()$ is the gamma function. Finally, the current density formula is

$$J = \frac{I_{dd}}{WH} \tag{2.4}$$

being W and H width and the thickness of the metalwire, and I_{dd} the current ($I_{dd} = C \cdot V_{dd} \cdot f \cdot p$ where the parameters are the the capacitance of the metalwire, the power voltage, the clock frequency and the switching activity).

Given this lifetime reliability model, the average lifetime of the system is estimated in terms of its Mean Time To Failure (MTTF), defined as the area underlying the reliability function $R(t)$:

$$MTTF = \int_0^\infty R(t)dt. \tag{2.5}$$

MTTF is one of the common approaches for specifying the reliability target for a system: "the system must have at least a MTTF equal to X years". However, in order to compute the MTTF, it is necessary to know the value of the $R(t)$ function for the overall lifetime. This is feasible only when the system presents a predictable aging trend (for instance, when the system has a periodic or fixed activity plan, e.g. [Das et al., 2013, Das et al., 2014]). In the other situations, especially when the system workload is unknown, there is an alternative approach for specifying the reliability target that is to set a given reliability level $R(t_{target})$ the system must have at the end of the envisioned lifetime t_{target}. In other words the reliability target can be specified as "at the end of the working life, estimated in t_{target} years, the system must have at least a reliability of $R(t_{target})$".

For simplicity, Equation 2.2 considers only a constant temperature. This aspect may cause pessimistic non-accurate evaluation of the reliability especially when the focus is on the optimization of the usage of the system to improve its lifetime. Therefore, to consider a varying temperature, Equation 2.2 can be enhanced as following as demonstrated in [Xiang et al., 2010]:

$$R(t) = e^{-\left(\sum_{j=1}^{i} \frac{\tau_j}{\alpha_j(T)}\right)^{\beta}} \tag{2.6}$$

where τ_j represents the duration of each period of time with constant steady-state temperature T_j up to time t (i.e., $t = \sum_{j=1}^{i} \tau_i$).

2.5.1 Analysis of the Reliability Model

The first step in the investigation of the reliability-aware resource management for many-core systems is an accurate analysis on the causes of the aging according to the considered reliability model.

From the considered model, it is possible to state that the temperature is the most relevant parameter in the aging phenomenon. In fact it represents the parameter that varies more sensibly during the time. Moreover, when analyzing the reliability model defined by JEDEC Solid State Technology Association, it is possible to state that the aging rate has an exponential relationship with the temperature. This means that even small changes in the temperature may have large effects on the aging rate, and such effect is propagated on the system lifetime. Actually, the exponential relationship between the temperature and the MTTF can be seen also in the following formula which computes the MTTF for a system working at a constant temperature [JEDEC Solid State Tech. Association, 2010]:

$$MTTF_{EM} = A_0(J - J_{\text{crit}})^{-n} e^{\frac{E_a}{kT}} \tag{2.7}$$

To evaluate quantitatively this relationship, it is possible to analyze the data reported in Figure 2.3 that shows the reliability curve and the MTTF of a single-core system working at a constant worst case temperature. It can be seen that variations of a few degrees in the temperature causes results in very different reliability curves. It is worth noting that this relationship holds also in the scenario of a variable temperature profile.

The second parameter that may vary during the system activity is the current density. However, its effect is more limited, since it has a polynomial relationship

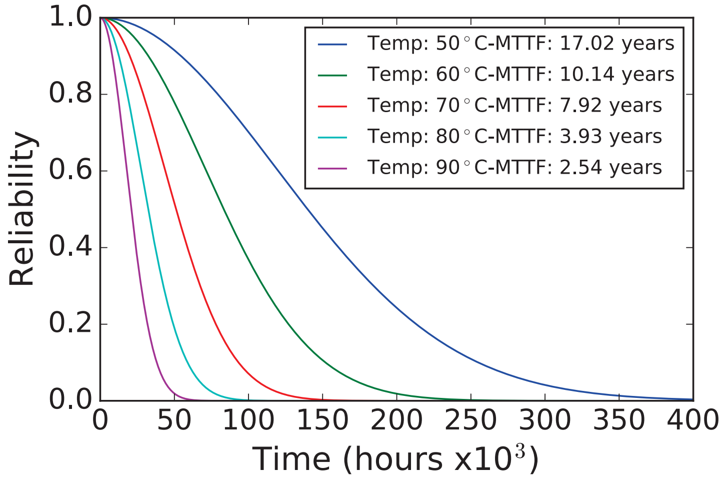

Figure 2.3: Reliability curve of the same system at different constant temperatures.

with the aging rate, and consequently with the MTTF. Moreover, if DVFS is not considered, variations in the current density may be almost neglected w.r.t. the temperature impact.

There is a second more important consideration that is fundamental to take into account when investigating the temperature/reliability relationship: only long-term variations of the temperature trend have a perceptible effect on the reliability curve while instantaneous magnitude of stress causes an almost negligible degradation. As an example, Figure 2.4 shows the reliability curves of a nominal system working at a temperature equal to 60 Celsius degrees constant for the overall lifetime, and three similar systems experiencing a temperature variation to 90 Celsius degrees for a given period of 100, 1000 and 10.000 hours respectively. It is possible to conclude that the effect on the reliability is perceptible when the temperature variation holds for at least 1000 hours.

Finally, if the system has a highly variable temperature profile in time (e.g. variations every few hours or even less), its reliability will be characterized by a long sequence of infinitesimal variations. Indeed, such oscillating curve can be approximated by computing the "average" effect on the reliability of such highly variable temperature profile. To this end, the following formula computes the approximated mean aging rate [Bolchini et al., 2014a]:

$$\alpha = \frac{\sum_{i=0}^{p} \tau_i}{\sum_{i=1}^{p} \frac{\tau_i}{\alpha_i(T)}} \tag{2.8}$$

where the short term period is divided in p steps each one with a steady-state temperature and with a duration equal to τ_i. For instance, if the system has two working

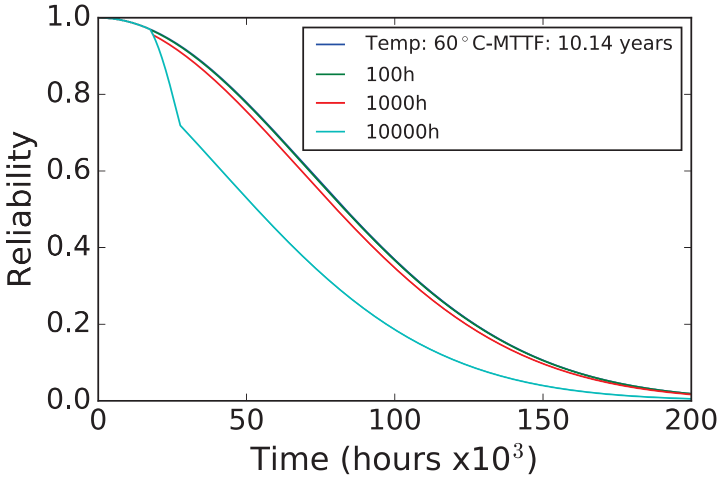

Figure 2.4: Reliability curve of the same system. The nominal temperature is 60 Celsius degrees and at $t = 10.000$ there is a temperature variation to 90 Celsius degrees for a period equal to $100h$, $1000h$, $10000h$ respectively.

points, one at 60 Celsius degrees and the other one at 90 Celsius degrees, and periodically switches from one to the other one with a period of 1 hour, the effect on the reliability of each temperature change will be imperceptible. The average aging rate of such temperature profile will be the same of the scenario in which the system will work at a constant temperature of 79.229 Celsius degrees. A relevant consideration that can be drawn from this example is that if the temperature profile is variable, the resulting average aging rate is not equal to the one considering the mean temperature (as wrongly assumed in [Mercati et al., 2014]). Indeed it will be higher than the mean.

According to the drawn considerations, it is possible to conclude that the reliability is a function that changes very slowly during the lifetime of a system. Moreover, since the distribution of the various applications affects the utilization of the various cores in the device, actually the mapping decisions affect the temperature of the device. As a conclusion, the reliability model is a function of such mapping decisions. However, since applications generally use to arrive and to complete in a short period (from seconds to a few hours), the system has to take very frequently mapping decisions, but, due to the characteristics of the reliability model, such decisions will have an effect on the reliability on the long term. This means that the single mapping decision will have a negligible consequence on the reliability of a system, while a frequent trend in the mapping decisions will affect such reliability. Such considerations have been very useful for an accurate definition of the proposed reliability-aware runtime mapping approach presented in the next section.

2.6 Thermal Sensors and Thermal Simulation Model

By increasing the number of transistors in a single chip, coupled with breakdown of Dennardian scaling and increasing the on-chip power density, temperature and power management is a necessity in the current and future technologies [Lee et al., 2014]. In addition, different activity rate of functional blocks, non-uniform workload variation, and advanced static and dynamic power management capabilities in recent CMPs result in non-uniform power distribution on the substrate which leads to significant temperature gradient [Ajami et al., 2001]. Large temperature variation across a chip decreases the reliability of the circuits and degrades their performance [Coskun et al., 2008]. Several research studies in the field of dynamic thermal management (DTM) aim at mitigating temperature and power violations at runtime in many-core systems. An efficient DTM technique necessities accurate on-chip thermal sensors in recent technologies to maximize the performance under a restricted chip temperature. Localized sensors can provide critical information regarding the location of hotspots [Lee et al., 2010]. Today's multi-/many-core platforms are often equipped with multiple on-chip thermal sensors to monitor the chip's temperature in a fine-grained manner [Sasaki et al., 2006, Pham et al., 2005, Poirier et al., 2005].

Thermal sensor accuracy is extremely prone to intra-die process variation and aging phenomena, and its report gradually drifts from the nominal value. This can lead to both overestimation as well as underestimation of the real thermal status of the system. For example in [Remarsu and Kundu, 2009a], the authors show that un-calibrated thermal sensors for IBM25PPC750L processors deviate as much as 33C and 48C from their original temperature of 35C and 95C, respectively. Therefore, on-chip thermal sensors need to be calibrated initially before being used. However, the cost of infield calibration is too high which requires infrared camera and additional infrastructures [Remarsu and Kundu, 2009b]. Furthermore, due to device wear out, even though the sensors are well-calibrated before being used, their reports gradually drift away from actual temperature values which demands re-calibration at the time they are being used [Remarsu and Kundu, 2009b]. Thus, many commodity microchips prefer to use un-calibrated thermal sensors to be available for end-users [AMD-Publication, 2006].

This necessitates an efficient technique for sustainable sensor calibration before and while the sensor values are used. In addition, in modern many-core systems which are often enabled with dynamic voltage and frequency scaling (DVFS), thermal sensors located on cores are sensitive to the core's current voltage-frequency (VF) level, meaning that dedicated calibration is needed for each VF level. In Paper VII, a general-purpose software-based auto-calibration strategy for thermal sensors is proposed that operates without using any hardware infrastructures for DVFS-enabled many-core systems. We adopt a 2-point calibration method for calculating the calibration constants of each thermal sensor at each VF level. We demonstrate the efficiency of the proposed calibration strategy on a many-core platform, Intel's Single-chip Cloud Computer (SCC) (that is one of the few real platforms for many-core systems), covering all voltage and frequency combinations on the platform.

Chapter 3

Power Awareness in Dark Silicon Era

The main goal in the power management process is to achieve optimal power-performance efficiency considering thermal design power budget. This necessitates i) monitoring several system characteristics including both communication and computation aspects, ii) categorizing, prioritizing, and processing the information in an intelligent way, iii) and controlling a rich set of actuators. More precisely, a comprehensive Observe-Decide-Act (ODA) loop based multi-objective control approach is needed, which has access to a rich set of sensors and actuators. Dim Silicon concept is a promising approach to increase the overall throughput of chip multiprocessors (CMPs), at the expense of much lower operating frequency [Wang and Skadron, 2013, Kanduri et al., 2017]. It is considered as one of the most effective methods to mitigate the dark silicon phenomenon. The main grounding of our claim can be realized from Figure 3.1. In this figure overall power consumption and speedup for a hypothetical fully parallelized application with arbitrary number of threads (cores) are depicted. In each reported case, the execution time of the

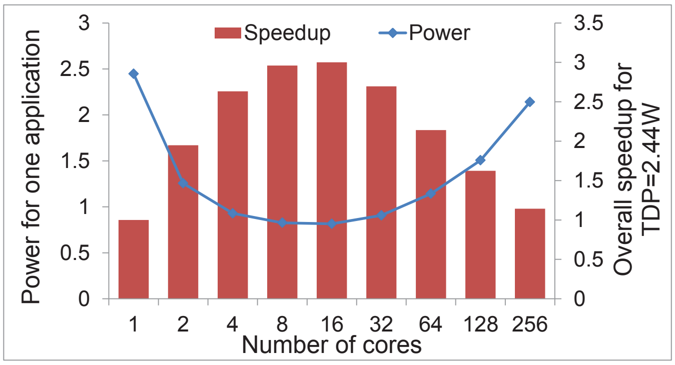

Figure 3.1: Power and speedup while increasing the utilization

application is constant and the differences are the number of threads (cores) employed to execute the application and operating voltage/frequency for the cores on which the threads are running (level of dim silicon). As it can be seen, increasing the number of worker cores causes reduction in overall power consumption until a certain point, i.e., 16 cores. Hence, limiting the amount of power budget to 2.5 Watt and using the surplus power budget by increasing the number of worker cores for further parallel execution, the speedup increases upto 3 times for 16 cores. This example shows that in the dark silicon era where there exists a maximum power cap for the chip, power management combined with execution parallelization results in better performance if manipulation of the voltage/frequency is done appropriately. Of course in the case of no power limit, since the application executions are assumed to be fully in parallel, nominal speedup can be gained that is based on the number of worker cores, e.g., speedup for 16 cores is 16[1].

Implementing an efficient Dim Silicon based approach necessities a comprehensive multi-objective power management mechanism having access to a rich set of on-chip sensors and actuators to utilize several Observe-Decide-Act (ODA) loops (i.e., feedback control) for controlling different aspects of the system. Such a multi-objective power management activity becomes even more challenging when considering near future manycore systems accommodating tens to hundreds of cores interconnected via Network-on-Chip (NoC). On top of that, manycore systems often need to handle an extremely dynamic workload with an unpredictable sequence of different applications entering and leaving the system at a runtime. In addition, due to the need to honor an upper limit on power consumption, i.e. fixed thermal design power (TDP) or dynamic thermal safe power (TSP) [Pagani et al., 2014], in the dark silicon era, a power capping mechanism is required to monitor the instantaneous total system power consumption and manage the power-performance requirements of the system.

The related work on closed-loop dynamic power management for chip multiprocessors can be classified into two main categories:

- **NoC-centric techniques** that utilize different communication related information such as queue length and injection rate as feedback to adjust voltage and frequency of processing elements, routers, or voltage-frequency islands (VFI) accordingly (e.g., [Bogdan et al., 2013] and [David et al., 2011]).

- **Power capping techniques** proposed for bus-based multiprocessor systems which utilize chip/per-core/per-cluster power measurement and per-core performance as sensory data to optimize system power-performance characteristics within a fixed power cap where there is no concern regarding network congestion and saturation (e.g., [Muthukaruppan et al., 2013] and [Ma and Wang, 2012a]).

[1] It should be noted that this example is quite theoretical where/in which there is anything sequential in the application

Even though all the techniques in these categories efficiently control the power consumption for their target platforms, they are not comprehensive enough to consider several factors affecting the performance in manycore systems. Therefore, we first characterise different key parameters which should be taken into consideration to devise a proper power management approach for the dark silicon era. In the following, we list the parameters and discuss their significance:

- **Power Budget:** Due to thermal issues in the dark silicon era, there exists an upper limit on power consumption which is called thermal design power (TDP) if it is a fixed value or thermal safe power (TSP) [Pagani et al., 2014] if it can change dynamically at a runtime depending on the number of active cores in a system. To guarantee the safety of the chip this limit should be strictly honored by the power manager.

- **Application Performance:** In order to monitor the impact of DVFS on application performance, a virtual or physical sensor to measure processors' utilization such as performance counters are needed. The main idea is to monitor how much impact voltage-frequency (VF) upscaling has had in the last epoch to increasing performance, and similarly how much VF downscaling has had negative impact on performance in the previous monitoring time-window.

- **Network-on-Chip Congestion:** Congestion in communication medium can easily lead to a poor efficiency of DVFS process. Assume a pair of producer and consumer processing elements (PEs) where there is a congestion in one or multiple routers in their communication path. VF upscaling of such PEs will result in either zero or marginal performance gain, while a considerable amount of energy can be wasted due to a long waiting time of data transactions. Therefore, utilizing congestion meters in NoC routers can provide to the power manager a beneficial source of information.

- **Application's Network-Intensity:** A source-throttling congestion control mechanism will impact in a limited way performance if it is done only based on network-load [Chang et al., 2012]. Such a mechanism is not application-aware, but rather throttles all applications equally regardless of applications' sensitivity to latency. Different applications impose different injection rates to the network and suffer differently from network congestion. As DVFS on PEs has also affect on application throttling, i.e. VF upscaling (downscaling) of a PE may result in increasing (decreasing) packet injection rate by the PE to the network, applications' characteristics in terms of their network-sensitivity should be also monitored and considered in power management.

- **Applications' Priorities:** There are different types of applications, for instance non-realtime, soft realtime, and hard realtime, where they demand

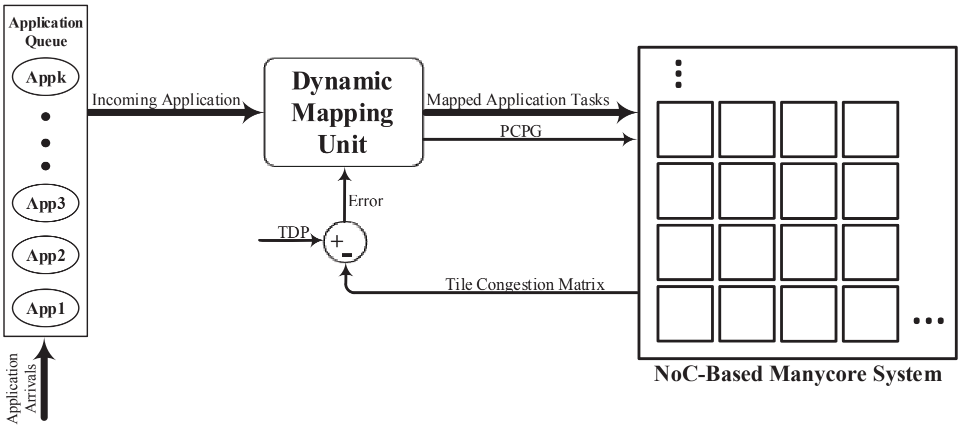

Figure 3.2: Power-aware mapping (PAM)

different quality of service at a runtime. These requirements including the minimum required power-budget for each application need to be considered in the prioritization phase in the controller.

- **Disturbances Caused by Runtime Mapping:** Whenever a new application is mapped onto the system, it is likely to cause a sudden change in overall power consumption that shoots above the TSP/TDP. Such sporadic rises in power consumption should be also considered and proactively managed.

In this part we discuss briefly the proposed power controllers from the naive power aware mapping (PAM) to complex reliability-aware multi-objective (RA-MOC) approach.

3.1 Power-Aware Mapping (PAM)

Figure 3.2 shows system architecure for power aware mapping (PAM) approach that is a simple controller for power management and is discussed in Paper II in more details. In PAM, based on the power feedback from the system and a power estimation of the most recently coming application, mapping the application is postponed until the summation of instantaneous power and estimated power of the new application is below TDP. The only power management actuator (knob) in PAM is PCPG that will be applied to the cores no application running on them, i.e., idle cores. Hence, since this technique simply stops running applications when the power violates, it performs reasonably well and we consider it as a naive baseline model to compare the other proposed methods.

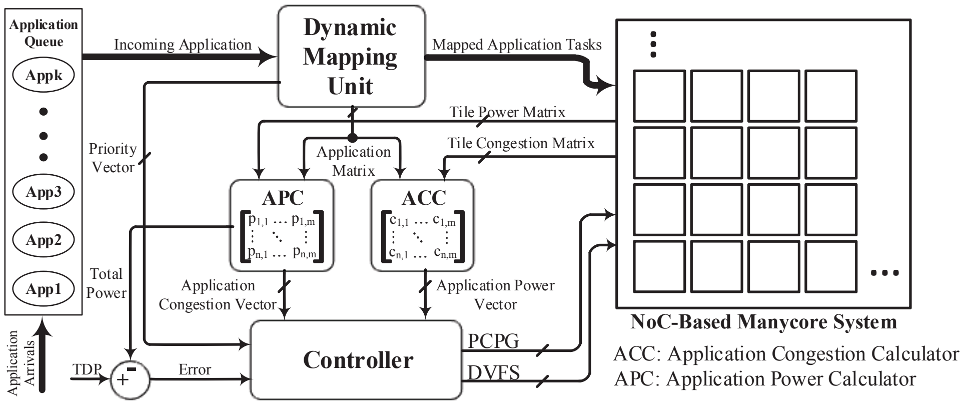

Figure 3.3: Dark silicon aware power management

3.2 Dark Silicon Aware Power Management (DSAPM)

The first attempt to propose an efficient power management technique in dark silicon era is shown in Figure 3.3 that is termed as dark silicon aware power management (DSAPM) and discussed in Paper II. The goal of the controller is to regulate the power using DVFS and PCPG actuators together. As can be seen, a feedback controller using system power measurement is incorporated. Similar to all other control systems, the controller compares the system output with a target value. After comparison, it manipulates the system actuators to minimize the error. The Dynamic Mapping Unit (DMU) tries to allocate system resources connected through the network, to incoming application tasks in an efficient way. It also provides information of the existing application(s) running on the system (RAI) to properly manipulate the actuators (e.g. priority vector and application matrix).

There can be various types of applications with different priorities on the system. The priority of an application determines the level of expected QoS. The power consumption of individual routers also vary dynamically due to primarily the uneven traffic distribution in a network. When a set of system resources are allocated to a task, part of the network become active with packets flowing in different directions. The associated routers regulating the packet flow thus dissipate proportional power in order to manage such traffic. To accurately measure this power dissipation, a power meter is designed within the router microarchitecture [Weldezion et al., 2013].

The power meter reads the rate of packet flow at link level and sends its aggregate value to the central control as a packet. There are four directional links (South, West, East, North) and a local link connecting to a processing element. If there is no packet flow in any of the links, then only the leakage power is consumed. If every link is passing a packet per cycle, then the router is consuming 100% its dynamic power actively. This happens when the network traffic is congested. However, under optimal conditions for unsaturated network, the router level power reading is smaller than 100%.

In our power management platform, the Application Power Calculator (APC) unit calculates the current power consumption of each application based on the Application Matrix provided by the Dynamic Mapping Unit and Tile Power Matrix, measured by the core and router power meters. By masking the Application Matrix on the Tile Power Matrix, the APC block calculates the current power consumption of each application, forms the Application Power Vector, and passes it to the Controller Unit.

It should be noted that if the fine-grained power measurement is not supported by a manycore platform, our power management approach still works fine even in the absence of the APC unit. The only feedback being necessary for our approach is the total chip power consumption. However, the APC unit improves the power allocator's decision by providing extra information regarding the contribution of each application in the total power consumption.

An ideal network configuration is one with a scalable network topology and traffic distribution where every packet is transmitted and received without delay and bandwidth limitation. For example, a network running a highly localized traffic where every node sends packets only to its immediate neighboring node, can be considered as an ideal one because it exploits its maximum performance. There is no traffic congestion in the network and each packet reaches its destination within a predictable latency.

Nevertheless, in practice, network traffic distribution is non-uniform and due to interconnection complexity and intrinsic wire delays such an ideal topology is not feasible. Instead more practical configurations, such as generic 2-D mesh or 3-D cube topologies, are used. However, the scalability of such practical topologies is limited as the capacity of the networks do not grow proportionally to accommodate traffics generated with increasing number of cores [Weldezion et al., 2009].

For each additional core, the network traffic gets more easily congested and the overall throughput per core decreases and hence the total network performance gives a diminishing return due to increased communication distance. This leads to a network performance gap as shown in Figure 3.4 for a network with a Uniform Random Traffic (URT) where every core is not able to send or receive packets in every cycle. In such cases, there is no need for a core to be actively consuming power at high frequencies or voltages. Thus, we find it imperative to take the network performance gap into account when designing dynamic power management manycore systems.

In our platform, each router is equipped with a congestion meter. The congestion meter measures router congestion levels in its recent history. More precisely, it measures the traffic dynamically by calculating the moving average of packet flow in every link of a router as follows:

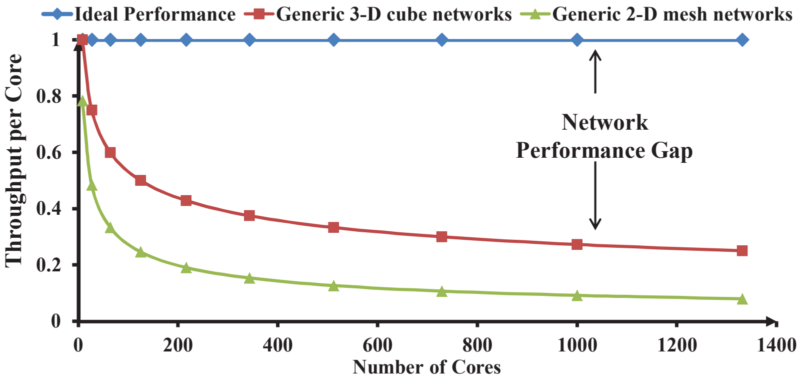

Figure 3.4: The share of network bandwidth per core diminishes with increasing number of cores per chip as shown here for URT.

$$C_{Total} = \frac{1}{W} \sum_{cycle=i}^{i+w} (\theta_{South,i} + \theta_{North,i} + \theta_{East,i} + \theta_{West,i} + \theta_{Local,i}) \qquad (3.1)$$

Where $\theta = \{0, 1\}$ is the presence or absence of a packet in a link at any given cycle, W is the width of the moving window, and C_{Total} is the moving average congestion level.

The congestion level of each router is transferred to the Application Congestion Calculator (ACC). By masking the Application Matrix on the Tile Congestion Matrix provided by the DMU, ACC calculates the average congestion level for each application and sends it to the controller unit.

The DVFS operation considers both normal and near-threshold cases. Voltage-to-frequency scalings are modeled by interpolating empirical results from circuit simulations. Transistor switching speed scales exponentially with the threshold voltage while operating at near-threshold voltage. As a result, near-threshold operation region is highly sensitive to the threshold voltage [Wang and Skadron, 2012]. For instance, the results for 16*nm*, 22*nm*, and 32*nm* technology node scalings are illustrated in Figure 3.5. More details regarding near-threshold frequency and voltage modeling can be found in [Wang and Skadron, 2012].

When bunch of applications are running on the system, the only way to change the overall power is to perform DVFS on the cores. If DVFS cannot be applied, because of reaching the lowest DVFS level or limitation in application QoS requirements, one application should be killed to save the power by PCPG. In the case of power violation, the amount of contribution each application has on overall power consumption and local network traffic are used as the metric to select the candidate application for VF downscaling. Since it is more efficient to apply

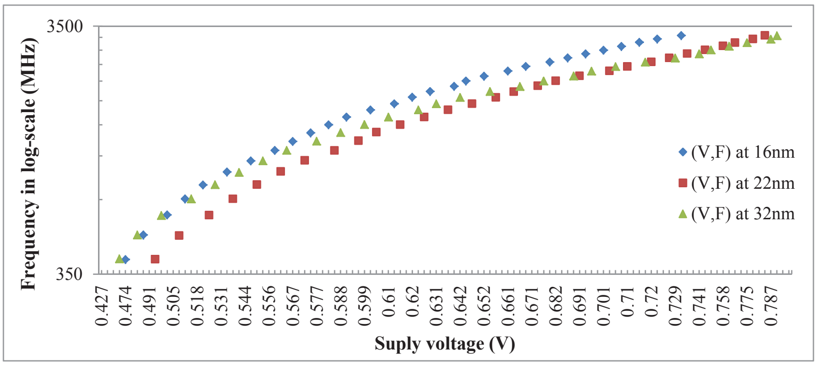

Figure 3.5: Maximum frequency for a given voltage at 16*nm*, 22*nm*, and 32*nm*

DVFS on all the worker cores executing a certain application [2], the more contribution each application has on overall power consumption, the less down scaling in voltage/frequency is needed for power inhibition. Furthermore, since the application operating frequency has a direct relationship with network congestion, the application which produces less local congestion is selected first. More details of the downscaling/upscaling algorithm and obtained result in comparison with the state-of-the-art are discussed in Paper II [ICCD-2014].

3.3 Multi-Objective Controller (MOC) for Power Management

The design of the controller can be more complex especially when different strategies to manipulate the actuators result in significant difference in performance and power consumption. In general the effect of changing VF, as one of the important power management actuators, can be investigated from two aspects namely its contribution on overall power consumption and its effect on the system performance. The final goal of the controller design must be to regulate the power consumption with the least negative effect on the system performance while considering maximum power upper bound. In the previous section, we suggested two metrics to select the target application to perform VF down/up-scaling namely tile power matrix and congestion matrix. More investigation on the characteristics of applications in their contribution on chip power consumption and network traffic shows that applications behave differently while applying VF scaling in terms of changing the overall power consumption and network traffic. For example, VF down-scaling on applications that are consuming more power, because of higher task activity or etc, results in higher power reduction comparing to VF down-scaling on applications with less power consumption. Moreover, VF down-scaling on applications with

[2] More details and supporting discussions can be found in Papers II, Paper III, and Paper IV

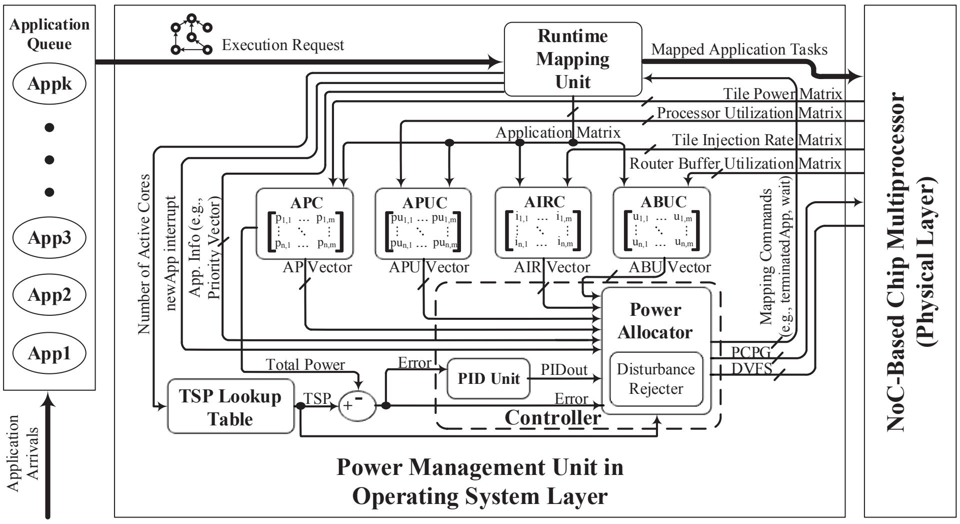

Figure 3.6: Overview of the multi-objective dark silicon aware power management system (AIRC: Application Injection Rate Calculator, ABUC: Application Buffer Utilization Calculator, APUC: Application Processor Utilization Calculator, APC: Application Power Calculator)

higher packet injection rate decreases network traffic more than VF down-scaling on applications with lower packet injection rate. To consider such application characteristics in power management alongside other observations such as power consumption and network congestion, an efficient multi-objective control approach is proposed and discussed in Paper III which considers workload characteristics, per-core power and performance measurements, network-load, disturbances caused by runtime mapping, and total chip power measurement all together.

Details of the multi-objective controller (MOC) are presented in Figure 3.6. The framework represents a general controlling strategy for manycore systems enabled with run-time mapping that can easily be applied to any NoC topologies such as 3D architectures. Run-time Mapping Unit (RMU) allocates cores in the NoC-based system to tasks of applications commenced for execution. Some information regarding the mapped applications is also provided by this unit to be passed to other controlling units. This is what we call Runtime Application Information (RAI). The priority of an application is proportional to the amount of expected QoS for that application. On a system, there might be different types of applications running with different priorities. For example soft realtime and non-realtime application might have different levels of priority in such systems.

Likewise the previous controller, an efficient observe, decide, and act (ODA) management strategy in this context requires several observation units to monitor different system characteristics at runtime. Later we show that such observation can be enhanced by including other system characteristics such as thermal profile,

Application Types

Non-Congested (*NC*) **Non-Intensive (*NI*)**	**Non-Congested (NC)** **Intensive (I)**
Congested (C) **Non-Intensive (NI)**	**Congested (C)** **Intensive (I)**

Figure 3.7: Four possible applications types classified by multi-objective power management algorithm

aging profile, etc.

To better consider applications' behaviour in power management strategy, two general metrics have been used namely application injection rate (AIR) and application performance-power ($D_{prf-pwr}$). Based on applications' injection rate (IR) information obtained from Application Injection Rate Vector (AIRV), all the applications are classified into two categories, intensive (I_{set}) and non-intensive (NI_{set}), the applications are also classified into two categories, congested (C_{set}) and non-congested (NC_{set}) based on their buffer utilization (BU) information obtained from Application Buffer Utilization Vector (ABUV). An application is considered to be congested if its corresponding routers' buffer utilization value is larger than a predefined threshold, for example 75%. Figure 3.7 shows four possible application types after classification based on injection rate and congestion. Then, every application is tagged at runtime with a 2-bit label which can get one of these values: *NI_NC* (non-intensive, non-congested), *NI_C* (non-intensive, congested), *I_NC* (intensive, non-congested), and *I_C* (intensive, congested). These tags are variable and updated in every iteration. This provides appropriate target set of applications that can be upscaled or downscaled to maximize network throughput.

Performance-power ratio (i.e., $D_{prf-pwr}$) is another metric for an application to be selected for VF upscaling or downscaling. In [Ma and Wang, 2012a], product of core utilization ($Util$) and aggregated frequency ($Freq$) is used as a high-level computational capacity metric. In this metric, the frequency is weighted to deduct the idling cycles. We extend this metric by aggregating core utilization in an application ($appUtil$), provided by $APUC$, to calculate the performance of an application as:

$$Perf_{current} = appUtil \times Freq_{current} \tag{3.2}$$

Then, the performance-power ratio is calculated as the following:

$$D_{prf-pwr} = \frac{Perf_{next} - Perf_{current}}{Power_{next} - Power_{current}} \tag{3.3}$$

$Power_{current}$ is the power consumption of the current application provided by the APC unit. $Power_{next}$ and $Perf_{next}$ are the estimated power consumption and performance of the application after the DVFS process. The next level of voltage and frequency (V_{dd_next} and $Freq_{next}$) are estimated for the candidate applications based on the magnitude of PID_{out} and application size. The $Perf_{next}$ and $Power_{next}$ are calculated as follows:

$$Perf_{next} = Perf_{current} \times \frac{Freq_{next}}{Freq_{current}} \tag{3.4}$$

$$Power_{next} = Power_{current} \times \frac{Freq_{next}}{Freq_{current}} \times (\frac{V_{dd\_next}}{V_{dd\_current}})^2 \tag{3.5}$$

After calculating $D_{prf-pwr}$ for all the applications in *appSet*, a simple $quicksearch$ algorithm is performed to find the application with the lowest and highest $D_{prf-pwr}$ value as the target application for DVFS, respectively.

Power regulation is performed by PID controller through applying VF scaling on target application selected by the power management algorithm. However the PID controller cannot immediately regulate a drastic power overshoot which might happen during the commence of a new application. To tackle such unwanted sporadic event, a specified unit is designed inside the controller unit, shown in Figure 3.6 as $DisturbanceRejecter$, to pro-actively scale down a selected set of applications to collect the power budget required by the new application before it starts.

3.4 Dark Silicon Patterning

Conventional mapping strategies schedule applications and tasks within an application in a contiguous and tightly packed manner in order to i) reduce inter-task communication latency and ii) maintain a regular geometrical structure to avoid dispersion of incoming applications. Despite the lower communication penalty, contiguous mappings accumulate temperature faster due to densely mapped active cores which mutually affect each others' temperature. This leads to on chip temperatures approaching critical temperature by consuming a relatively lower power budget, limiting resource utilization. Potential hot spots and thermal violations are likely to be handled by dynamic thermal management techniques, which can subsequently reduce the performance further. This can be addressed with an adaptive mapping policy that schedules applications in a sparse manner, as opposed to the conventional dense mappings. Applications and tasks within an application are mapped in a spread out manner in order to balance the heat distribution across the chip evenly. With hot active cores being neighboured by cool inactive cores, temperature accumulation is slowed down and the cores can consume relatively higher power before reaching critical temperature. This increases the utilizable power budget under safe thermal limits, improving performance. We refer to this technique as dark silicon patterning/aligning inevitable dark cores (inactive) alongside active cores to reduce mutual heat effect among them. Although spread out map-

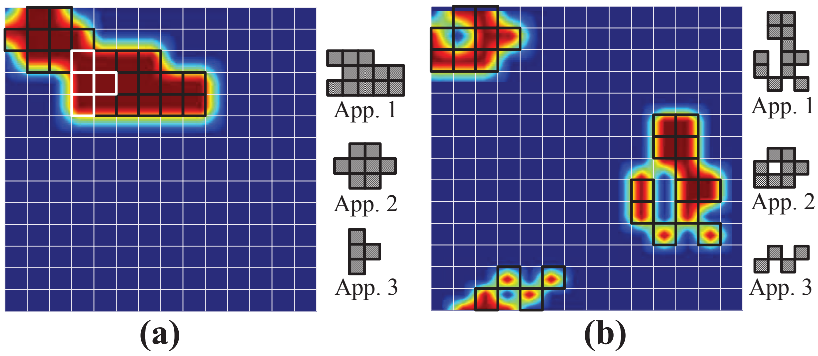

Figure 3.8: Thermal profiles of contiguous and spatially distributed mappings

pings induce communication penalty, the power budget gains would be significant, making it a better trade-off.

The impact of spatial alignment of active cores on power budget is explained through a motivational example, presented in Figure 3.8. Three applications App1, App2 and App3 with 9, 12 and 7 tasks respectively are assumed to be running on the system. Conventional mapping approaches offer lower inter-task communication latency by greedily mapping all the applications contiguously. Mapping of the 3 applications contiguously on a NoC-based many-core system with 144 cores is shown in Figure 3.8(a). The power budget (TSP) of this system as computed by TSP library is 66W. A non-contiguous and spread-out mapping of the same applications (as well tasks) is shown in Figure 3.8(b). This mapping provides a power budget (TSP) of 74.6W, as calculated by TSP library. An improvement of 8.6W in power budget can be observed for the spread-out, patterned mapping as opposed to tightly packed and contiguous mapping. Contiguous mapping avoids dispersion, but it leads to poor thermal profile of the chip due to prorogation of heat among neighboring applications and tasks and thus resulting in lower power budget. Contrastingly, spatially distributed mapping of applications offers higher power budget as effect of heat among different applications is negligible. Also, active cores are patterned along with inactive cores such that heat effects of neighboring cores running the same application are minimized.

The top level abstraction of the system implemented in the patterning approach is shown in Figure 3.9. Runtime mapping unit (RMU) estimates the power of incoming application and checks if the chip currently has enough power budget to run the new application, more details of this technique are discussed in Section 3.1. The application is forwarded onto the system if there is available budget. In case of un-availability, the application waits until the system can allocate enough budget, perhaps with currently running application(s) leaving the system after finishing their execution. TSP Calculator receives current mapping configuration of the system as input and computes the realistic upper bound on power budget, the

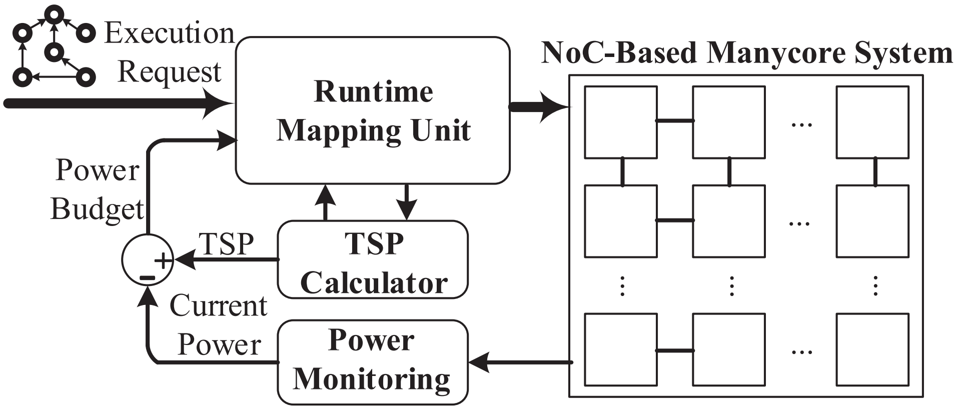

Figure 3.9: System architecture of the proposed patterning approach

thermal safe power (TSP). The RMU feeds this new budget value to the chip and updates the maximum power budget of the chip to the TSP provided by the TSP Calculator. The complementary elaboration of the proposed idea and motivational examples can be found in Paper V.

Chapter 4

Reliability Awareness in Dark Silicon Era: Prolonging System Lifetime

In the previous chapter, it was shown that how dark silicon can be mitigated to increase the overall performance while honoring the maximum power budget, i.e., TDP or TSP. In this chapter, dark silicon is considered from another perspective: to opportunistically exploit dark silicon to satisfy other design constraints (e.g., obtaining higher reliability in our study). Recently, reliability has emerged as a key metric in the field of chip design and it has become one of the main constraints and limiting factor alongside performance and power. Past studies [Xiang et al., 2010] have shown that many types of failure are exponentially dependent on temperature, and a $10 - 15°$C difference in operating temperature may result in a $2\times$ difference in the overall lifespan of a chip. In fact, the adoption of a TDP (or TSP) only partially solves the issues related to the increased power densities, and the resulting high operating temperatures within the device. Even if TDP avoids excessive temperature peaks, the overall temperature profiles that characterize modern devices are considerably higher than that of in the past. As discussed in the ITRS reports in 2011 [Semiconductor-Industry-Association et al., 2011], such high temperatures, combined with the extreme downscaling of CMOS technologies, have caused an acceleration in device aging and wear-out processes. As a matter of fact, modern circuits are more susceptible to phenomena such as electromigration or time dependent dielectric breakdown, that lead to circuit degradation causing delay errors and, eventually, device breakdowns. Thus, we are currently experiencing a dramatic decrease of lifetime in modern digital systems that can be considered while dealing with dark silicon phenomena.

In the past years, researchers (e.g. [Ma and Wang, 2012b, Chantem et al., 2013, Gnad et al., 2015, Sun et al., 2014, Huang and Xu, 2010]) have proposed system-level strategies for slowing down the aging process. In these works, the main idea is to balance the utilization of processing cores and to use dynamic voltage and frequency scaling (DVFS) to keep the operating temperature and the accumulated stress under control over the system's service time. However, as mentioned before, management of many-core systems is a complex problem where several matters need to be considered, e.g., dynamicity of running workloads and power consumption. For these reasons, many-core systems are generally provided with an advanced runtime resource management layer orchestrating applications' mapping and power distribution across the system. In this scenario, the straightforward integration of existing approaches is not effective, since in this scenario only a part of the complex problem is considered, and often, such scenarios have partially contradicting objectives with the resource management policies viz., to enhance power-performance characteristics vs. to enhance balanced allocation.

In Paper VI, through presenting empirical evidence derived from an extensive set of experiments we show how reliability management can be considered as another objective to reinforce the resource management. Moreover, we elaborate on the challenges related to the definition of reliability-aware runtime resource management strategies for the considered architecture under the dark silicon scenario. As resource management should control several knobs such as task allocation, voltage/frequency manipulation, etc, this might lead to an objective overlap among different control units and requires further coordination toward obtaining final actuation in these units. Thus, despite in dark silicon management where core-wide power saving techniques have been employed, in this chapter our goal is to exploit dark silicon through runtime application mapping to minimize the overlap with the power management. Later we provide more comprehensive study in which dark silicon aware power and reliability management are merged together as a multi-objective co-management of system resources.

Runtime application mapping policy is one of the key factors that determines the performance and energy efficiency of many-core systems [de Souza Carvalho et al., 2010] running a dynamic workload. Most of the existing runtime application mapping algorithms focus on communication minimization and contiguity of the applications in the system. With dark silicon, the number of cores that can be active is dynamic and subjective to activity of other worker cores. In fact, the abundance of cores and the infeasibility of using all the cores at the same time provide a unique opportunity for the runtime management unit to balance the utilization stress among the processing units to prolong the system lifetime. In the following, we propose a runtime reliability-aware resource management technique for many-core systems to evenly distribute the workload stress across the system.

4.1 Reliability-Aware Runtime Mapping

The proposed reliability-aware runtime mapping technique is composed of two phases (i.e. nested feed-back controllers): i) runtime reliability analysis (i.e., the long-term management), and ii) reliability-aware runtime mapping (i.e., the short-term management). In the long term phase, a reliability analysis unit monitors fine grained thermal profile and computes the aging status of each subsystem by means of a state-of-the-art statistical reliability model. In the short term, based on the information obtained from the long term analysis, online resource management is adopted not only to fulfill the target reliability requirements which is specified in design time, but also to balance the lifetime of the system by excluding the more stressed cores from the mapping selection pool.

The overall architecture of the proposed lifetime-aware runtime mapping approach is shown in Figure 4.1. Similar to dark silicon management techniques, as shown in Figure 4.1, the approach uses a centralized controller, implementing a feedback control loop, which are organized into two main units, 1) being in charge of application mapping and the other 2) dealing with lifetime reliability. Indeed, such partitioning of the activities is motivated since two different time horizons are considered here: mapping activities are performed at a lower frequency, since applications can be issued every minute and they last for a period ranging from some seconds to a few hours, while reliability can be managed with long-term decisions, as the aging is a slow phenomenon and has perceptible effects over epochs lasting for days or weeks.

The **Reliability Analysis Unit** is the *long-term controller* responsible for monitoring the aging status of the cores. The unit computes an aging reference, that is a target reliability curve defined to reach the specified reliability requirement $R(t_{target})$ at the end of the lifetime t_{target}. This aging reference suggests the controller on how fast each core should age to fulfill the given reliability target. Then, at predefined *long-term epochs* the unit analyzes the current reliability value of each core w.r.t. the target aging reference to compute a specific reliability metric describing the aging trend. The unit gathers the cores' aging status by a utility module, called **Reliability Monitor**. This unit continuously reads the cores' temperatures (every few seconds), and accordingly updates the $R(t)$ values by using the model discussed in [Bolchini et al., 2014b], and in particular the following equation:

$$R(t) = e^{-\left(\sum_{j=1}^{i} \frac{\tau_j}{\alpha_j(T)}\right)^{\beta}} \tag{4.1}$$

where τ_j represents the interval between the measure T_j andT_{j-1} up to time t (i.e., $t = \sum_{j=1}^{i} \tau_i$).

The **Runtime Mapping Unit** is the *short-term controller* that dispatches the arrived applications on the grid of processing cores. It is activated at application arrival events and takes decisions according to 1) the profiled characteristics of the arrived application (e.g. the required dynamic power consumption of each task),

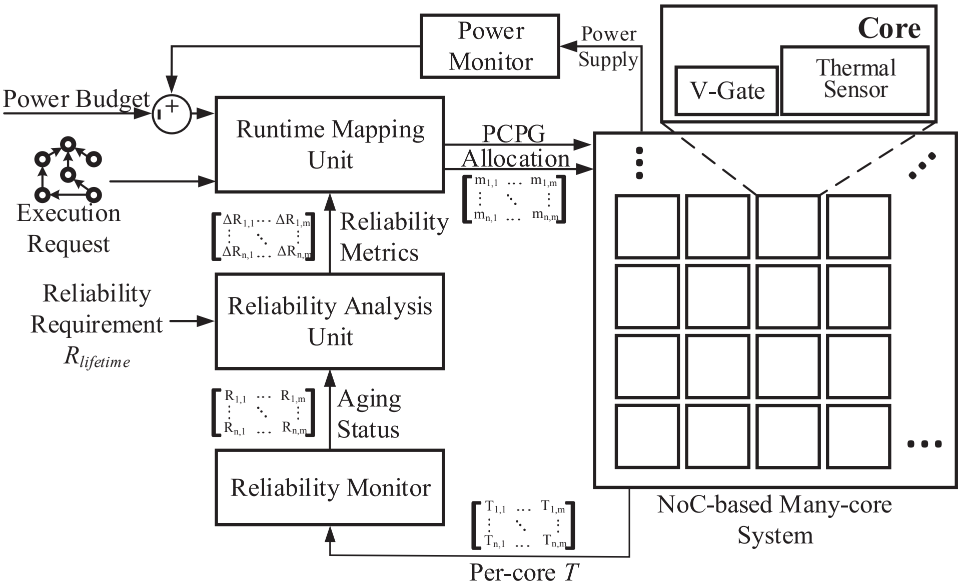

Figure 4.1: The overall architecture of the proposed runtime lifetime-aware mapping controller.

2) the current power consumption received by the **Power Monitor**, and 3) the information received by the Reliability Analysis Unit. In particular, the reliability metrics of the various cores are used as weights in the mapping decisions. The unit is also provided with a waiting queue, where applications are temporarily stored if the system is not able to immediately admit them due to unavailability of idle cores or a violation of the power budget. Finally, the actuation phase is executed by the Runtime Mapping Unit which dispatches the applications. The details of the various modules briefly discussed here are presented and formalized in Paper VIII.

Chapter 5

Reliability Awareness in Dark Silicon Era: Online Testing

As discussed before, increasing density of logic gates in silicon chips and susceptibility to internal defects have led to increase in permanent fault manifestation in nanometer technology devices. One viable solution to handle such reliability quest is to detect and manage permanent failures in operational components via concurrent error detection and online testing. Traditionally, error detection and/or correction is generally implemented by redundancy-based techniques [Sieworek and Swarz, 1982], such as duplication with comparison (DWC) or triple modular redundancy (TMR) for concurrent online detection and correction of errors and built-in-self-test (BIST) or test access mechanism (TAM) for online/offline fault detection, which present a high cost due to area occupation. Another strategy for online testing is Software Based Self-Test (SBST, [Foutris et al., 2010, Kaliorakis et al., 2014]), which is implemented via periodic execution of specific testing routines devoted to the functional solicitation of the circuitry for the detection of permanent failures. Since such strategy does not require any additional circuitry, it represents the most promising solution for consumer electronic devices. Indeed, an example of its large scale deployment is in the automotive on-board computing systems [Bernardi et al., 2011] such as [Haghbayan et al., 2015b, Kaliorakis et al., 2014].

Many-core systems fall under this umbrella of the digital devices which can significantly benefit from SBST [Skitsas et al., 2013, Khodabandeloo et al., 2011, Haghbayan et al., 2010]. In fact, such systems commonly do not often feature any integrated hardware for online testing and, are subject to a considerable stress caused by intensive data-processing workload. The deployment of SBST in many-core systems offers at the same time new opportunities and challenges. Due to the importance of the power consumption, in the dark silicon era, there is a quest for a *power-aware online test scheduling approach* to detect faults occurring in many-core architectures with minimum performance degradation. Another challenge of test scheduling is the high dynamicity and heterogeneity of the executed workload. This makes the amount of dark area on the chip (i.e., total chip utilization) highly variable. Furthermore, due to the emergence of new concepts, such as dim silicon [Wang and Skadron, 2012] as a way to minimize dark areas and increase the number of active cores, the system might reach up to 100% core utilization (if the majority of running application are not performance-demanding) by making use of power management features through Dynamic Voltage and Frequency Scaling (DVFS) [Rahmani et al., 2015]. This makes the behavior of such systems to be highly related to the characteristics of the workload. At different moments of time it is possible to have considerable dark areas with small resource utilization due to the fact that some other group of cores are set on a high voltage-frequency level thus reserving the majority of the overall power budget. On the other hand it is also possible to have small dark areas with large resource utilization by globally setting a very low voltage-frequency level. Therefore, if suitable scenarios are opportunistically identified (when there is enough power slack), such temporary dark areas can be favorable targets for online testing in order to improve the system reliability [Shafique et al., 2014, Haghbayan et al., 2014]. Nevertheless, DVFS knobs also introduce other issues over the testing process [Kavousianos and Chakrabarty, 2013, Haghbayan et al., 2012]. As faults are manifested in different ways in different configurations, systems should be tested at multiple voltage-frequency settings. Therefore, the test scheduling needs to take into account the fact that SBST routines should be executed on various cores at different voltage-frequency levels.

Given these motivations, this chapter presents a power-aware online testing approach for the dark silicon era to exploit dark silicon for a transparent power-aware online test scheduling in many-core systems. The sections are chronologically ordered based on development process of the proposed approach. The proposed approaches mainly benefits from the high probability of finding dark cores in large many-core systems and occasionally available power slacks for dynamically scheduling SBST routines. This process is performed by the mapping unit, on the idle cores that have experienced a high stress in the recent past. In Section 5.1 we first present online concurrent test scheduling under dedicated power budget that is based on Paper IX. In Section 5.2 we extend foregoing approach by proposing a non-intrusive power-aware online testing to functionally test the cores in their idle times that is based on Paper X. In particular, the approach exploits a criticality metric, computed based on core utilization, to select the units for test. Then, a test and scheduling approach selects the actual cores among the candidates based on two conditions, 1) the cores should be idle (i.e., not currently involved in the execution of an application) and 2) there should be some power slack available to be used for the execution of the running applications. In Section 5.3 the power-aware online testing approach is reinforced by also reconsidering the online reliability status of idle cores. Further, the test scheduling approach selects also the optimal possible voltage-frequency settings to execute the SBST routine by considering the system's power budget.

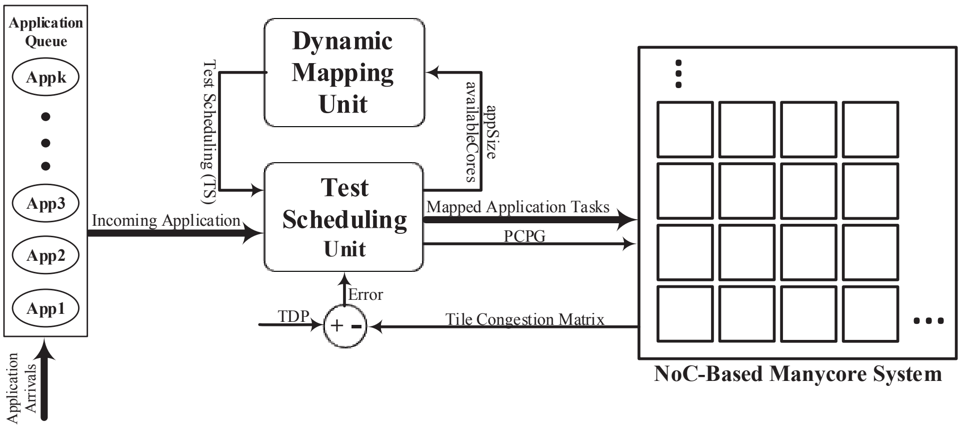

Figure 5.1: System architecture of energy efficient online testing approach

5.1 Energy-Efficient Concurrent Testing

Figure 5.1 shows the system architecture of the energy efficient concurrent testing approach. The mapping algorithm uses a predefined power threshold (TDP), the current power consumption of the system, and an estimation of the power consumption of the new application waiting to be mapped. Likewise PAM method (Section 3.1), if instantanous total system power consumption plus the estimated power consumption of the new application, exceeds a threshold, TDP, the central mapper will wait for releasing some cores. Then, it allocates the available cores in the system to the new application.

The Test Scheduling Unit shown in Figure 5.1 is responsible for allocating test applications to the cores. The test application consists of an off-line generated software which is mapped in the same way as ordinary applications, according to the mapping algorithm. The Test Scheduling Unit chooses the best idle or dark core in the system to be the core under test. This process continues repeatedly at runtime and when a faulty core is detected the mapping unit is informed. Our test scheduling algorithm aims to maximize the test speed by increasing the parallelism of the test procedure while honoring the dedicated power bound for the test purpose. To this end, we apply dynamic voltage and frequency scaling (DVFS) down to near-threshold operation to the cores under test to minimize their power consumption while increasing the number of parallel cores under test. Reducing the frequency and voltage of a core under test increases the duration of the test procedure. However, experimental results show that parallel nature of the approach decreases the total duration of the test process.

5.2 Power-Aware Online Testing

Many-core systems with highly dynamic workloads are generally subject to highly varying workloads, i.e., different types of applications arrive with an unknown trend and are characterized by different performance requirements, variable amount of data to be elaborated, different demand of processing resources just to mention a few. Therefore, in each instant of time, the phasic behaviour of workloads and their distribution on many-core system affect instantaneous power availability on the system. Such variable workloads accompanying the variable change in overall power consumption brings the promising opportunity to perform online testing in many-core systems. Actually, the highly variable and evolving status of the many-core system due to the dynamic workload presents periods with a high resource and power utilizations and period with a low utilization. Therefore, an opportunistic online test scheduling method can take advantage of the second kind of situations in order to test the dark cores as long as there is enough room in the remaining power budget. We found that without dedicating certain amount of power budget for testing and only by monitoring different scenarios, low overhead online test application can be performed. The proposed framework for dark silicon aware online testing is presented in Figure 5.2. It is an extension of the classical runtime power management framework discussed in Section 3.2, with some additional components devoted to the execution of the test-related activities.

The goal of the proposed approach is to transparently run SBST routines during the system activities without affecting the execution of the nominal workload. Thus, the aim is to guarantee that processing cores are not affected by permanent failures and, at the same time, to maintain the required level of performance for the running workload. The basic idea is to test each core with a rate proportional to the stress it has been affected due to its utilization. If a core is frequently used for execution of applications, it is highly stressed and therefore needs frequent tests. On the other side, if the core has been rarely allocated, it does not require urgent testing in the near future. The benefit of this approach is to guarantee the necessary test frequency without performing cores' over-testing that would have a negative effect on the execution of the nominal workload in terms of larger power consumption and unnecessary resources occupation, or cores' under-testing that would reduce the reliability of the system.

Test Scheduling Unit (TSU) is devoted to select the cores that need to be tested according to the experienced stress and the scheduling of the testing task on those cores. The experienced stress is estimated by means of a criticality metric. It is computed according to a specific hardware component integrated within each core counting the number of executed instructions. The TSU works in a tightly-coupled way with RTM and DPM units to define a proper test scheduling. In particular, the RTM unit has been slightly modified in order to take into account the fact that if a core is candidate for the test procedure, it should not be considered for mapping purposes. In Paper X, the various activities of TSU are discussed in

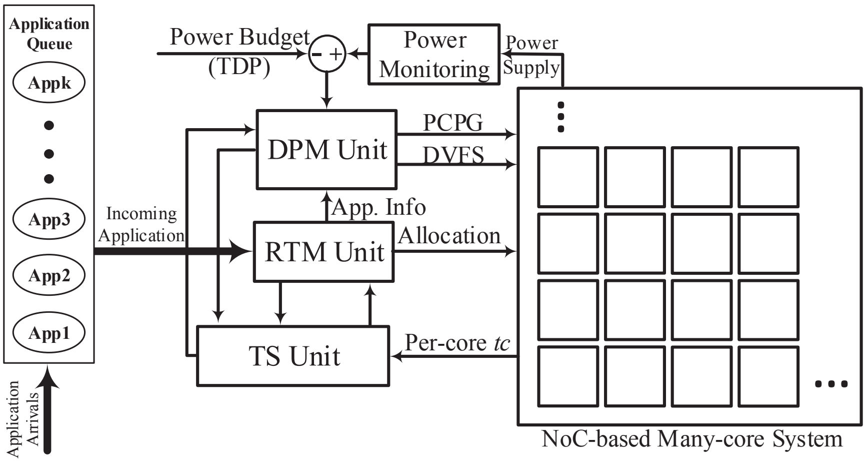

Figure 5.2: The system architecture including the online testing framework

details together with the internal modifications to RTM Unit necessary to handle the test information received by TSU.

5.3 Aging-Aware Tuning of Test Scheduling

At the beginning of the system lifetime, when a core is new, it is presumable that it will have a low failure probability, and, therefore, it is not strictly necessary to test the core very often. However, as the core ages, this necessity of online testing increases proportionally to the failure probability. Therefore, we extend our online testing approach to also consider the aging status of the chip. To develop this idea we propose an approach to dynamically tune the testing period based on the core reliability profile of the system.

The enhanced version of the proposed dark silicon aware online testing approach is presented in Figure 5.3, in which certain additional components are devoted to track the aging status of the system. As shown in Figure 5.3, the system contains a Reliability Monitor (RM), which is in charge of computing the aging status of each core. The RM can be implemented in two different ways: in hardware, by means of wear-out sensors, or in software, by using a statistical lifetime reliability model relying on the existence of per-core thermal sensors within the platform. The RM implements the standard lifetime reliability model based on a Weibull distribution [JEDEC Solid State Tech. Association, 2010] that is explained in Chapter 3.

Another feature which is added to the proposed aging-aware online testing is to perform test scheduling for each core at different voltage/frequency (VF) levels. Based on the recent studies, some specific faults manifest themselves in a particular VF settings [Ali et al., 2006]. These studies have concluded that multi-/many-core

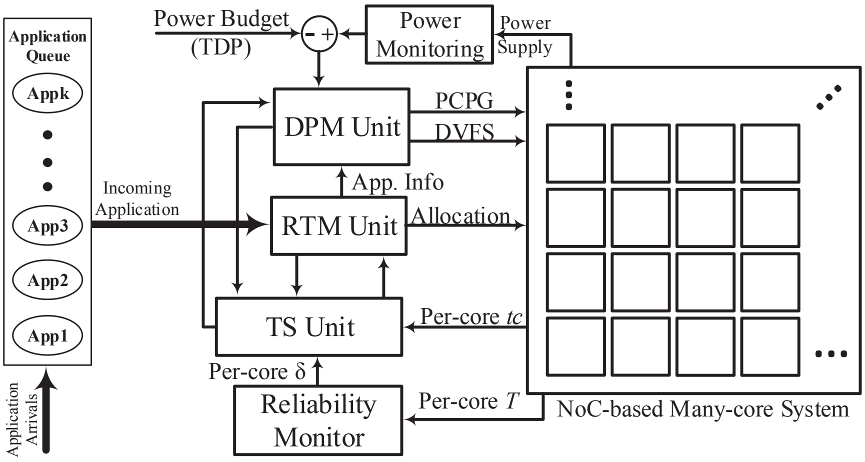

Figure 5.3: Proposed system architecture for reliability oriented power aware online testing

systems equipped with DVFS should be tested at multiple voltage levels to ensure that cores can operate reliably at different conditions. Testing at multiple voltage levels is more challenging compared to that of single voltage level testing, because in each voltage level a separate SBST routine execution is needed and the maximum possible operating frequency is limited [Kavousianos et al., 2012]. Test scheduling and repetitively running test process for every voltage level drastically increases the overall Test Application Time (TAT) that has a direct impact on the overall system performance. At low voltage levels, test process becomes slower as the frequency is lower, resulting in a longer TAT. To apply online testing on cores running at different voltage levels, it is essential to use a test scheduling policy with the minimum negative impact on system performance. To this end, allocated core(s) need to be detected and enough power budget need to be available for the test purpose so that the upper power consumption bound will not be violated. However, as test power consumption at different voltage levels considerably varies, the suitable frequency level at each voltage level should be properly determined at runtime. This approach is demonstrated and elaborated in Paper XI.

Chapter 6

Power-Reliability Awareness in Dark Silicon Era

So far in this thesis we discuss how to manage reliability and power in dark silicon era. For power management a multi-objective feed-back based controller is proposed to pursue performance optimization while fulfilling the power budget. The actuators were PCPG and DVFS and the observations were based on large set of monitoring parameters such as workload characteristics, network congestion, and power-performance characteristics of cores. On the other hand, for reliability management in dark silicon era two approaches of lifetime-aware runtime mapping and power-aware online testing are proposed. In this chapter, two approaches are presented in which power and reliability management are performed together in a coordinated manner. In the first approach, we extend the multi-objective dynamic power management technique presented in Chapter 3 to monitor the reliability of the cores as the feedback, alongside current power consumption and other network characteristics, while utilizing fine-grained voltage and frequency scaling and per-core power gating as the actuators. The second approach provides a trade-off between system performance and reliability by simultaneously honoring the given power budget and a target reliability metric. A novel runtime reliability analysis unit is introduced to estimate the aging status of each core and computing a set of metrics showing the per-core reliability trend over the system lifetime. Moreover, the application mapping and resource management scheme are extended to co-manage performance and reliability in order to meet the required target reliability while having a minimal negative impact on system performance.

6.1 Reliability-Aware Multi-objective Power Controller

In this section multi-objective controller discussed in Chapter 3 is being extended by considering reliability in power management alongside network characteristics and power consumption. To do this, the effect of power management on system

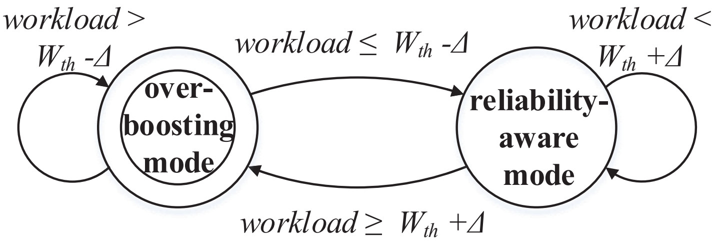

Figure 6.1: State machine diagram of the algorithm within the Operating Mode Selector

lifetime has been investigated and power controller has been enhanced to consider core's lifetime while managing the power. Fine-grained change of power consumption can affect the core stress and lifetime distribution on the chip. Therefore, alongside performance fulfillment, the multi-objective controlling mechanism must also consider the current reliability state in its actuator manipulation to enhance the overall system lifetime in the long term.

Since cores are stressed in long-term, to design a reliability-aware multi objective controller ($RA - MOC$) two operation modes are considered, *over-boosting mode* and *reliability-aware mode*. The *over-boosting mode* is characterized by a highly intensive workload. For this reason, the system needs to work at full speed, and, consequently, reliability issues are ignored. Whereas, in *reliability-aware mode*, metrics from the reliability monitoring are taken into account to avoid thermal hotspots while trying to obtain an acceptable performance as well. Therefore, the ultimate goal of these two operating modes is to obtain in the short term optimal system's performance while mitigating unnecessary stress and wear-out in the system in the long term. Therefore, the Power Controller Unit is provided with an Operating Mode Selector which monitors the overall amount of workload the system is experiencing in the current period of time and consequently decides the proper operating mode. The Operating Mode Selector internally behaves as a Finite State Machine, as shown in Figure 6.1, that on the basis of a given threshold switches between the over-boosting and the reliability-aware mode. A tolerance guardband can be used around the threshold value to avoid excessive oscillations between the two different operating modes.

Figure 6.2 shows the reliability-aware power controller. As can be seen, the framework is provided with a monitor devoted to the observation of the aging status of the various tiles, i.e., Reliability Analysis Unit. This unit periodically samples the current temperature of each tile by means of the available sensors and updates the current chip reliability using the current temperature and previous reliability measurement. The controller switches between *over-boosting mode* and *reliability-aware mode* based on observation from workload in Policy Making Unit.

We observed that, there might still be possibility to further balance the reliability by deviating from current determined VF level with a negligible performance

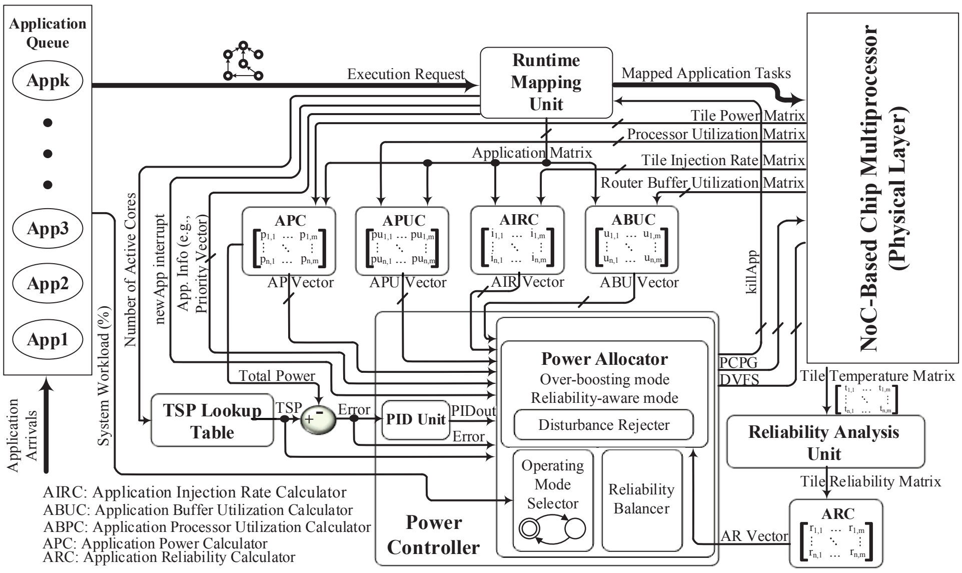

Figure 6.2: Overview of reliability-aware multi-objective controller

penalty, even though when the controller is in *reliability-aware mode*. In this way, we can achieve a more efficient reliability balancing in long term. To this end, a reliability balancer module is added to Power Controller that tries to adjust the distribution of power on chip by changes in VF levels of the tiles to reshape the unbalanced reliability distribution. More details about the implementation and algorithms are discussed in Paper IV.

6.2 Reliability-Aware Resource Management

In this section, a performance-reliability aware resource management for many-core systems is proposed to deal with the trade-off between system performance and reliability while honoring the given power budget. Figure 6.3 shows the framework of the proposed approach. It is an extension of the runtime resource management layer unit in operating system level to handle aging issues concurrently to the nominal application mapping and power management.

The proposed resource management layer is organized into two main units being in charge of the power and reliability management. These activities follow two very different time horizons: application mapping and power management activities are performed with a short-term frequency, since applications can be issued every moment and they last for a period ranging from some seconds to a few hours, while reliability can be managed with long-term decisions, since the aging of a system is a relatively slow phenomenon and has perceptible effects over epochs lasting for days, weeks or even months.

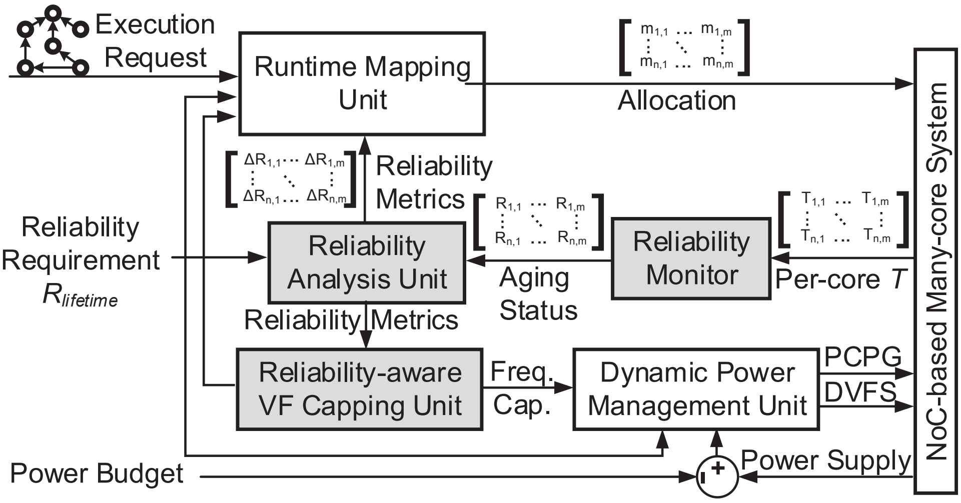
Execution Request
Runtime Mapping Unit
$m_{1,1} \dots m_{1,m}$; $m_{n,1} \dots m_{n,m}$
Allocation
NoC-based Many-core System
$\Delta R_{1,1} \dots \Delta R_{1,m}$; $\Delta R_{n,1} \dots \Delta R_{n,m}$
Reliability Metrics
$R_{1,1} \dots R_{1,m}$; $R_{n,1} \dots R_{n,m}$
$T_{1,1} \dots T_{1,m}$; $T_{n,1} \dots T_{n,m}$
Reliability Requirement $R_{lifetime}$
Reliability Analysis Unit
Aging Status
Reliability Monitor
Per-core T
Reliability Metrics
Reliability-aware VF Capping Unit
Freq. Cap.
Dynamic Power Management Unit
PCPG
DVFS
Power Budget
Power Supply
+
-

Figure 6.3: The proposed reliability-aware RRM layer.

The central part of the framework in Figure 6.3 (filled in gray color) represents the *long-term controller*, which performs the reliability management and contains the **Reliability Monitor**, **Reliability Analysis Unit** and **Reliability-aware VF Capping Unit**. The former is an utility unit that computes that aging status of each node within the architecture by continuously reading the temperature from the per-core sensors and applying the adopted reliability model. Then, the Reliability Analysis Unit monitors aging status of the various nodes according to the information gathered by the Reliability Monitor. In particular, according to the reliability requirement $R(t_{target})$ at the end of the lifetime t_{target}, provided at the beginning of the service life by the system architect, it computes the target reliability curve. This curve represents an aging reference showing how fast each node should age in order to fulfill the given reliability requirement. Then, the unit periodically analyzes the current reliability value of each node w.r.t. the target aging reference to compute specific reliability metrics describing the aging trend to be used in the mapping decisions. Finally, the Reliability-aware VF Capping Unit takes additional recovery actions to unstress nodes that have already consumed the available "reliability budget", i.e., their reliability is considerably below the reference curve. Its main strategy is to cap maximum voltage/frequency levels of selected nodes to reduce temperature peaks, and, consequently, slow down the aging trend.

The rest of Figure 6.3 represents the *short-term controller*, containing the set of units devoted to the management of the nominal activities of the system. These units have been specifically enhanced to take also into account the reliability metrics provided by the long-term controller in the decision process. In particular, the reliability metrics are used as weight in the mapping decisions in order to prioritize younger nodes, while power management takes into account the reliability-driven maximum VF configuration. More details are discussed in Paper XII.

Chapter 7

Discussion and Conclusion

The need to utilize controlling mechanisms in management of dark silicon is becoming more evident particularly when number of cores in a chip increases. In this dissertation, we discussed and introduced a comprehensive multi-objective feedback based controller approach to manage dark silicon while protecting many-core systems against power consumption violation from a certain limit while maximizing system utilization and throughput. The target system architecture across the dissertation was a Network-on-Chip-based multiprocessor system using dynamic application mapping where applications enter and leave the system at runtime. We utilized a closed loop feedback system with comprehensive cross layer sensor data such as processing elements' power-performance measurements, application workloads, and network congestion to monitor the system. Several methods for scaling the voltage and frequency of the cores, and a proactively avoid power consumption violations were discussed. It was shown that the proposed techniques can efficiently mitigate the dark silicon and manipulate power management actuators. The obtained results show improvements in system throughput as well as reductions in power violations for the proposed platform when compared to state-of-the-art power management policies.

From another perspective, this dissertation shows how dark silicon phenomenon can be exploited to increase the system reliability in many-core systems by introducing reliability awareness in 1) runtime resource management and 2) online testing process. Our results on reliability awareness are promising and pave the way for more optimized and reliable resource management policies. Our proof-of-concept approach achieves up to 20% lifetime improvement, where we anticipate future enhancements can be pursued if more advanced techniques are invoked. The proposed power-aware online testing approach consists of a non-intrusive online test scheduling algorithm using software-based self test techniques to test idle cores in the system while respecting the system's power budget. Moreover, a criticality metric is proposed to identify and rank in terms of their reliability status. The goal of the approach is to guarantee prompt detection of permanent faults, while minimizing the performance overhead and satisfying the limited available power budget. Experimental results show that the proposed power-aware online testing approach can 1) efficiently utilize temporarily unused cores and available power budget for the testing purposes, within less than 1% penalty on system throughput and only using 2% of the actual consumed power 2) evenly distribute the stress of the cores and 3) cover all voltage-frequency levels throughout the test procedure.

Furthermore, as the final approach, an enhanced reliability-performance co-management shows how reliability and power management can be amalgamated together with the aid of novel reliability analysis unit and reliability-aware runtime mapper. Moreover, a coupled customized reliability-aware power capping unit performs core-level voltage/frequency scaling to provide excessively stressed cores with a recovery period. Our experimental results demonstrate the effectiveness of the strategy to fulfill the target reliability in long term with negligible performance penalty and power budget violation compared with the state-of-the-art.

Chapter 8

Overview of Original Publications

Articles published including the results and analysis from the thesis are summarized below.

8.1 Paper I: MapPro: Proactive Runtime Mapping for Dynamic Workloads by Quantifying Ripple Effect of Applications on Networks-on-Chip

In this paper, a proactive region selection strategy for application mapping is proposed, prioritizing nodes that offer lower congestion and dispersion. The proposed strategy, MapPro, quantitatively represents the propagated impact of spatial availability and dispersion on the network with every new mapped application. This allows us to identify a suitable region to accommodate an incoming application that results in minimal congestion and dispersion. The network is clustered into squares of different radii to suit applications of different sizes and proactively select a suitable square for a new application, eliminating the overhead caused with typical reactive mapping approaches. We evaluated our proposed strategy over different traffic patterns and observed gains of up to 41% in energy efficiency, 28% in congestion and 21% dispersion when compared to the state-of-the-art region selection methods.

The main contribution of this paper is listed as follows:

- Quantification of spatial availability in runtime mapping strategy, internal congestion and dispersion into a unified metric

- Modeling of the ripple effect of a newly mapped application on the remaining un-occupied nodes.

- Proactive first node selection for a generic mesh-type NoC running dynamic workloads.

Author's contribution: The author is responsible for experimental setup, evaluation of the algorithm over synthetic workloads, comparison against the state-of-the-art and drawing conclusions on advantages of MapPro.

8.2 Paper II: Dark Silicon Aware Power Management for Manycore Systems under Dynamic Workloads

In this paper a PID (Proportional Integral Derivative) controller based dynamic power management method is proposed that considers an upper bound on power consumption, the Thermal Design Power (TDP). To avoid violation of the TDP constraint for manycore systems running highly dynamic workloads, it provides fine-grained DVFS (Dynamic Voltage and Frequency Scaling) including near-threshold operation. In addition, the method distinguishes applications with hard real-time, soft real-time and no real-time constraints to treat them with appropriate priorities. In simulations with dynamic workloads and mixed-critical application profiles, we show that the method is effective in honoring the TDP bound and it can boost system throughput by upto 43% compared to a naive TDP scheduling policy.

The key contributions of this work are as follows:

- Dynamic power management with explicit consideration on TDP constraints.
- A feedback controller providing fine-grained (per core) dynamic voltage and frequency scaling (DVFS) including near-threshold operation.
- Congestion-aware power management designed for NoC-based manycore systems, considering the network performance gap.
- To demonstrate the efficiency of the proposed approach in the dark silicon era, we model a manycore system using current and future technology nodes down to 16*nm* for different die area budgets.

Author's contribution: The author proposed and implemented the main controller by largely extending Noxim, a SystemC based manycore system simulator. The author added support for voltage-frequency scaling to the Noxim platform and implemented the PID controller for runtime power management in different technology nodes. Furthermore, the author integrated power and technology scaling models of Niagara-like cores from MCPAT and Lumos [Wang and Skadron, 2012].

8.3 Paper III: Dynamic Power Management for Many-Core Platforms in the Dark Silicon Era: A Multi-Objective Control Approach

In this paper, a multi-objective dynamic power management method is proposed that simultaneously considers upper limit on total power consumption, dynamic behaviour of workloads, processing elements utilization, per-core power consumption, and load on network-on-chip. Fine-grained voltage and frequency scaling, including near-threshold operation, and per-core power gating are utilized to optimize the performance. In addition, a disturbance rejecter is designed that scales down activity in running applications when a new application commences execution, to prevent sharp rise in power consumption that might lead to power budget violations. Simulations of dynamic workloads and mixed time-critical application profiles show that our method is effective in honoring the power budget while considerably boosting the system throughput and reducing power budget violation, compared to the state-of-the-art power management policies.

The key contributions of this work are as follows:

- Providing a comprehensive dark silicon aware power management platform for NoC-based manycore systems under limited power budget (both TDP and TSP) and running dynamic workloads (i.e., supporting runtime mapping)

- Design of a multi-objective feedback-based controller providing per-core power gating (PCPG) and per-core DVFS considering workload characteristics, network congestion, and power-performance characteristics of processing elements (PEs)

- Integrate a proactive runtime mapping technique to reject the disturbance (i.e., to avoid the high overshoot) which happens when a new application is mapped onto the system in runtime

- Integrating dynamic TSP [Pagani et al., 2014] as reference at runtime, enabling developers to choose dynamic TSP or fixed TDP as power budget

Author's contribution: The author implemented the multi-objective power management approach in message-passing based manycore systems. Furthermore, the author integrated the dynamic TSP calculation unit to the platform that became one of the the references for PID controller. The author implemented the features to extract network congestion, application injection rate and performance characteristics from the system. The author contributed with the write-up and presentation.

8.4 Paper IV: Reliability-Aware Runtime Power Management for Many-Core Systems in the Dark Silicon Era

In this paper, we propose a multi-objective dynamic power management technique that uses power consumption, network characteristics, and reliability of the cores as the feedback to actuate fine-grained voltage and frequency scaling and per-core power gating. In addition, disturbance rejecter and reliability balancer are designed and added to smoothen power consumption in the short-term and reliability in the long-term, respectively. Simulations of dynamic workloads and mixed criticality application profiles show that our method is effective in honoring the power budget, boosting the system throughput, and increases the overall system lifetime by minimizing aging effects by means of power consumption balancing.

A preliminary version of the approach has been proposed in Paper III. This work is extended to also consider lifetime reliability issues together with power and performance optimization, as follows:

- Adding the reliability analysis unit to calculate fine grained reliability based on the temperature profile feedback from the system.
- Developing novel decision policies targeted for two different operating modes: Over-boosting Mode, when the system is experiencing an intensive workload, and Reliability-aware Mode, when the non-intensive workload offers the controller the opportunity to prolong the system lifetime.
- Extending the metrics for VF scaling decisions considering reliability of the system.
- Adding an additional reliability balancing module running at coarse time intervals.
- Evaluating the efficiency of our approach to provide high performance while prolonging the system's lifetime and fulfilling the given power budget.

Author's contribution: The author implemented the reliability-analysis unit and integrated it with Noxim platform. The author implemented the reliability-aware power management policy and added it to the manycore system. The author implemented the detection policy of workload and the corresponding state machine and added them to the manycore platform to have two different modes of operation, i.e., normal mode and reliability-aware mode. The author contributed with the write-up and presentation.

8.5 Paper V: Dark Silicon Aware Runtime Mapping for Many-core Systems: A Patterning Approach

In this paper, a dark silicon aware runtime application mapping approach is proposed that patterns active cores alongside the inactive cores in order to evenly distribute power density across the chip. This approach leverages dark silicon to balance the temperature of active cores to provide higher power budget and better resource utilization, within a safe peak operating temperature. In contrast with exhaustive search based mapping approach, our agile heuristic approach has a negligible runtime overhead. Our patterning strategy yields a surplus power budget of up to 17% along with an improved throughput of up to 21% in comparison with other state-of-the-art run-time mapping strategies, while the surplus budget is as high as 40% compared to worst case scenarios.

The key contributions of this work are as follows:

- A dark silicon aware runtime application mapping approach that aligns active cores with dark cores to offer higher power budget.

- A closed-loop power budgeting platform that keeps the maximum power consumption under safe operational power (i.e., TSP) which varies at runtime.

Author's contribution: The author contributed in implementing the power management for manycore system upon which the patterning approach is performed. Furthermore the author contributed in proposing the idea and implementation of the algorithm of first node selection in the patterned based mapping. The author contributed with a part of write-up and presentation of the paper.

8.6 Paper VI: Can Dark Silicon Be Exploited to Prolong System Lifetime?

In this paper, we claim that dark silicon can be exploited for reliability purposes by efficiently managing system resources (both cores and power) in order to prolong the system lifetime while achieving the same level of performance. Moreover, the opportunities given by dark silicon for lifetime improvement in many-core systems is discussed by presenting empirical evidence derived from an extensive set of experiments. Moreover, we elaborate on the challenges related to the definition of reliability-aware runtime resource management strategies for the considered architecture under dark silicon scenario. Our experiments demonstrate that a reliability-aware runtime resource management approach can improve the lifetime of the system up to 39% w.r.t. its nominal counterpart. It should be noted that further improvements can also be achieved by using more advanced techniques.

Author's contribution: The author implemented different techniques of runtime mapping and power management on many-core system and investigated and extracted the effect of those techniques on the system's lifetime. The author contributed with the write-up and presentation.

8.7 Paper VII: Software-Based On-Chip Thermal Sensor Calibration for DVFS-enabled Many-core Systems

In this paper, a general-purpose software-based auto-calibration strategy for thermal sensors is proposed without using any hardware infrastructures for DVFS-enabled many-core systems. We adopt a 2-point calibration method for calculating the calibration constants of each thermal sensor at each VF level. We demonstrate the efficiency of the proposed calibration strategy on a many-core platform, Intel's Single-chip Cloud Computer (SCC), covering all voltage and frequency combinations on the platform.

Author's contribution: The author contributed in proposing idea for and implementation of software-based sensor calibration algorithm. Co-author Sami Teräväinen also contributed in implementing the main thermal sensor calibration algorithm on SCC platform which is a real platform of manycore system with 48 cores. The author contributed with the write-up and presentation.

8.8 Paper VIII: A Lifetime-Aware Runtime Mapping Approach for Many-core Systems in the Dark Silicon Era

In this paper, we propose a novel lifetime reliability-aware resource management approach for many-core architectures. The approach is based on hierarchical architecture, composed of a long-term runtime reliability analysis unit and a short-term runtime mapping unit. The former periodically analyses the aging status of the various processing units with respect to a target value specified by the designer, and performs recovery actions on highly stressed cores. The calculated reliability metrics are utilized in runtime mapping of the newly arrived applications to maximize the performance of the system while fulfilling reliability requirements and the available power budget. Our extensive experimental results reveal that the proposed reliability-aware approach can efficiently select the processing cores to be used over time in order to enhance the reliability at the end of the operational life (up to 62%) while offering comparable level of performance against state-of-the-art runtime mapping approaches.

The key contributions of this work are as follows:

- Proposing a lifetime reliability aware runtime mapping to fulfill systems' target reliability requirements while considering performance and limited power budget in many-core systems.
- Exploiting dark silicon to maximize the overall system lifetime by choosing the less stressed resources and providing long term recovery period for highly stressed cores.
- Utilizing fine-grained temperature feedback to dynamically analyze the reliability and develop design time target reliability analysis to be used as a

metric in the runtime mapping algorithm.

Author's contribution: The author implemented the reliability-aware mapping algorithm on Noxim platform which is SystemC based message-passing manycore system. Furthermore, the author integrated the reliability model provided by Dr. Antonio Miele from Politecnico di Milano into the platform. The author integrated Hotspot to the manycore system to calculate the temperature profile required for modeling the reliability. The author contributed with the write-up and presentation.

8.9 Paper IX: Energy-Efficient Concurrent Testing Approach for Many-Core Systems in the Dark Silicon Age

In this paper an online concurrent test scheduling approach is proposed for the fraction of chip that cannot be utilized due to the restricted utilization wall. Dynamic voltage and frequency scaling including near-threshold operation is utilized in order to maximize the concurrency of the online testing process under the constant power. As the dark area of the system is dynamic and reshapes at a runtime, our approach dynamically tests unused cores of runtime to provide tested cores for incoming applications to enhance system reliability. Empirical results show that our proposed concurrent testing approach using dynamic voltage and frequency scaling (DVFS) improves the overall test throughput by 250% compared to the state-of-the-art dark silicon aware online testing approaches under the same power budget.

The key contributions of this paper are listed as follows:

- Dynamic voltage and frequency scaling for testing cores in order to parallelize the testing process with constant power allocated for test

- Test scheduling algorithm to select cores to be tested among free or dark cores with lowest throughput penalty

- Power aware test scheduling according to the number of available resources and current power of the system

Author's contribution: The author implemented the test scheduling unit to apply test routines on dark cores, honoring upper limit on power consumption. Furthermore, the author implemented the dynamic voltage and frequency scaling for test routines. The author also extracted the test routine model for testing the Niagara-like in-order cores from hardware description language (HDL) code of SPARC-like processor. The author contributed with the write-up and presentation.

8.10 Paper X: Power-Aware Online Testing of Manycore Systems in the Dark Silicon Era

In this paper, a power-aware online testing method for many-core systems is proposed. The proposed power-aware method uses non-intrusive online test scheduling strategy to functionally test the cores in their idle period. In addition, we propose a test-aware utilization-oriented runtime mapping technique that considers core utilization and their test criticality in the mapping process. Our extensive experimental results reveal that the proposed power-aware online testing approach can efficiently utilize temporarily free resources and available power budget for the testing purposes, within less than 1% penalty on system throughput for the 16nm technology.

The key contributions of this work are as follows:

- Power-aware online test scheduling method with explicit consideration of limited power budgets in many-core systems using runtime application mapping.
- Test-aware runtime mapping algorithm that considers cores with high test criticality in the mapping process.
- Detecting suitable scenarios in many-core systems when online testing methods can be applied in a minimally intrusive (often non-intrusive) way.
- Feedback controller based power management mechanism considering power consumption of cores in the normal operation and test modes.
- Modeling a many-core system using current and future technology nodes down to 16*nm* for different die area budgets to demonstrate the efficiency of the proposed approach in the dark silicon era.

Author's contribution: The author implemented the test criticality analysis unit, test scheduling unit, and test-aware mapping algorithm. Furthermore, the author extracted the test routine model of Niagara like in-order cores from hardware description language (HDL) code of an SPARC-like processor to be applied on system level model. The author contributed with the write-up and presentation.

8.11 Paper XI: A Power-Aware Approach for Online Test Scheduling in Many-core Architectures

This paper proposes a power-aware non-intrusive online testing approach for many-core systems. The approach schedules software based self-test routines on the various cores during their idle periods, while honoring the power budget and limiting delays in the workload execution. A test criticality metric, based on a device aging model, is used to select cores to be tested at a time. Moreover, power and reliability issues related to the testing at different voltage and frequency levels are also handled. Extensive experimental results reveal that the proposed approach can i) efficiently test the cores within the available power budget causing a negligible performance penalty, ii) adapt the test frequency to the current cores' aging status, and iii) cover available voltage and frequency levels during the testing.

The main contributions of this paper, which is major extension of Paper X, are:

- An enhanced power-aware online test scheduling method with explicit consideration of limited power budgets in many-core systems using runtime application mapping.
- An efficient test scheduling method for testing the cores in different voltage-frequency settings.
- Extending the test criticality metric [Haghbayan et al., 2015b] with a lifetime reliability estimation to drive fault occurrence probability as a priority to test the cores, and to balance the regularity of the test according to the aging status of the cores.
- Modeling and evaluation of a many-core system using various current and future technology nodes (32*nm*, 22*nm*, and 16*nm*) for different die area budgets to demonstrate the efficiency of the proposed approach in the dark silicon era.

Author's contribution: The author implemented the extended version of the power-aware online test scheduling for manycore systems. Co-author Dr. Antonio Miele from Politecnic di Milano contributed with providing life-time characteristics of the Niagara like in-order cores. The author integrated these lifetime characteristics of the cores into the manycore platform and extracted the results of lifetime for maycore system in long term. The author contributed with the write-up and presentation.

8.12 Paper XII: Performance/Reliability-aware Resource Management for Many-Cores in Dark Silicon Era

In this paper we propose a novel lifetime reliability/performance-aware resource co-management approach for many-core architectures in the dark silicon era. The approach is based on a two-layered architecture, composed of a long-term runtime reliability controller and a short-term runtime mapping and resource management unit. The former evaluates the cores' aging status w.r.t. a target reference specified by the designer, and performs recovery actions on highly stressed cores by means of power capping. The aging status is utilized in runtime application mapping to maximize system performance while fulfilling reliability requirements and honoring the power budget. Experimental evaluation demonstrates the effectiveness of the proposed strategy, showing that it outperforms recent state-of-the-art techniques.

The key contributions of a mature version of the framework that we propose here are the following:

- Proposing a two-step application mapping approach which considers reliability metrics w.r.t. a lifetime target and the current VF map of the architecture to balance the performance/reliability trade-off while fulfilling the power budget.
- Defining a maximum VF capping strategy compliant with state-of-the-art reliability-agnostic power management approaches to unstress specific areas of the device that have been aged faster than the prevision.
- Presenting a more advanced reliability analysis unit with a detailed discussion on the reliability monitor.
- Presenting an extensive experimental evaluation revealing that the proposed approach can carefully guarantee the required lifetime of the chip for different power management strategies in long-term with a negligible performance penalty.

Author's contribution: The author implemented the reliability-aware application mapping algorithm. Furthermore the author implemented the reliability-aware voltage/ferequency capping algorithm and added it to the manycore simulator. The author contributed with the write-up and presentation.

Bibliography

[TGG, 2017] (2017). TGG: Task Graph Generator. `http://sourceforge.net/projects/taskgraphgen/`. Last update: 2015-08-02.

[Adapteva, 2017] Adapteva (2017). Adapteva Epiphany. http://www.adapteva.com/.

[Ajami et al., 2001] Ajami, A., Banerjee, K., and Pedram, M. (2001). Analysis of substrate thermal gradient effects on optimal buffer insertion. In *IEEE/ACM International Conference on Computer Aided Design (ICCAD)*, pages 44–48.

[Ali et al., 2006] Ali, N., Zwolinski, M., Al-Hashimi, B., and Harrod, P. (2006). Dynamic voltage scaling aware delay fault testing. In *Proc. European Test Symp. (ETS)*, pages 15–20.

[AMD-Publication, 2006] AMD-Publication (2006). "revision guide for amd npt family 0fh processor". In *AMD Publication #33610*, page 37.

[Bernardi et al., 2011] Bernardi, P., Grosso, M., Sanchez, E., and Ballan, O. (2011). Fault grading of software-based self-test procedures for dependable automotive applications. In *Proc. Design, Automation & Test in Europe (DATE)*, pages 1–2.

[Black, 1969] Black, J. R. (1969). Electromigration 8212;a brief survey and some recent results. *IEEE Transactions on Electron Devices*, 16(4):338–347.

[Bogdan et al., 2010] Bogdan, P., Kas, M., Marculescu, R., and Mutlu, O. (2010). Quale: A quantum-leap inspired model for non-stationary analysis of noc traffic in chip multi-processors. In *Proceedings of the 2010 Fourth ACM/IEEE International Symposium on Networks-on-Chip*, pages 241–248. IEEE Computer Society.

[Bogdan and Marculescu, 2011] Bogdan, P. and Marculescu, R. (2011). Non-stationary traffic analysis and its implications on multicore platform design. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 30(4):508–519.

[Bogdan et al., 2013] Bogdan, P., Marculescu, R., and Jain, S. (2013). Dynamic Power Management for Multidomain System-on-chip Platforms: An Optimal Control Approach. *ACM Trans. Design Automation for Electronic Systems*, 18(4):46:1–46:20.

[Bolchini et al., 2014a] Bolchini, C., Carminati, M., Gribaudo, M., and Miele, A. (2014a). A lightweight and open-source framework for the lifetime estimation of multicore systems. In *Proc. Int. Conf. on Computer Design (ICCD)*, pages 166–172.

[Bolchini et al., 2014b] Bolchini, C. et al. (2014b). A lightweight and open-source framework for the lifetime estimation of multicore systems. In *ICCD*.

[Carvalho et al., 2007] Carvalho, E., Calazans, N., and Moraes, F. (2007). Heuristics for Dynamic Task Mapping in NoC-based Heterogeneous MPSoCs. In *Rapid System Prototyping, 2007. RSP 2007. 18th IEEE/IFIP Int. Workshop on*, pages 34–40.

[Chang et al., 2012] Chang, K., Ausavarungnirun, R., Fallin, C., and Mutlu, O. (2012). HAT: Heterogeneous Adaptive Throttling for On-Chip Networks. In *Proc. Int. Symp. on Computer Architecture and High Performance Computing (SBAC-PAD)*, pages 9–18.

[Chantem et al., 2013] Chantem, T., Xiang, Y., Hu, X., and Dick, R. (2013). Enhancing multicore reliability through wear compensation in online assignment and scheduling. In *Proc. Conf. on Design, Automation & Test in Europe (DATE)*, pages 1373–1378.

[Chou et al., 2008] Chou, C.-L., Ogras, U., and Marculescu, R. (2008). Energy- and Performance-Aware Incremental Mapping for Networks on Chip With Multiple Voltage Levels. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 27(10):1866–1879.

[Coskun et al., 2008] Coskun, A., Rosing, T., Whisnant, K., and Gross, K. (2008). Static and dynamic temperature-aware scheduling for multiprocessor socs. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, (9):1127–1140.

[Danowitz et al., 2012] Danowitz, A., Kelley, K., Mao, J., Stevenson, J. P., and Horowitz, M. (2012). Cpu db: Recording microprocessor history. *Queue*, 10(4):10:10–10:27.

[Das et al., 2013] Das, A., Kumar, A., and Veeravalli, B. (2013). Reliability-driven task mapping for lifetime extension of networks-on-chip based multiprocessor systems. In *Proc. Conf. on Design, Automation & Test in Europe (DATE)*, pages 689–694.

[Das et al., 2014] Das, A., Kumar, A., Veeravalli, B., Bolchini, C., and Miele, A. (2014). Combined DVFS and Mapping Exploration for Lifetime and Soft-error Susceptibility Improvement in MPSoCs. In *Proc. Conf. on Design, Automation & Test in Europe (DATE)*, pages 61:1–61:6.

[David et al., 2011] David, R., Bogdan, P., Marculescu, R., and Ogras, U. (2011). Dynamic Power Management of Voltage-Frequency Island Partitioned Networks-on-Chip Using Intel Sing-Chip Cloud Computer. In *Proc. Int. Symp. on Networks-on-Chip (NOCS)*, pages 257–258.

[de Souza Carvalho et al., 2010] de Souza Carvalho, E. L., Calazans, N. L. V., and Moraes, F. G. (2010). Dynamic task mapping for MPSoCs. *IEEE Design & Test of Computers*, 27(5):26–35.

[Dennard et al., 1974] Dennard, R., Gaensslen, F., Rideout, V., Bassous, E., and LeBlanc, A. (1974). Design of ion-implanted MOSFET's with very small physical dimensions. *IEEE Journal of Solid-State Circuits*, 9(5):256–268.

[Esmaeilzadeh et al., 2012] Esmaeilzadeh, H., Blem, E., St. Amant, R., Sankaralingam, K., and Burger, D. (2012). Dark Silicon and the End of Multicore Scaling. *IEEE Micro*, 32(3):122–134.

[Faruque et al., 2007] Faruque, A., Abdullah, M., Ebi, T., and Henkel, J. (2007). Run-time adaptive on-chip communication scheme. In *Computer-Aided Design, 2007. ICCAD 2007. IEEE/ACM International Conference on*, pages 26–31. IEEE.

[Faruque et al., 2008] Faruque, A., Abdullah, M., Krist, R., and Henkel, J. (2008). Adam: run-time agent-based distributed application mapping for on-chip communication. In *Proceedings of the 45th annual Design Automation Conference*, pages 760–765. ACM.

[Fattah et al., 2013] Fattah, M., Daneshtalab, M., Liljeberg, P., and Plosila, J. (2013). Smart hill climbing for agile dynamic mapping in many-core systems. In *Proc. Design Automation Conf. (DAC)*, pages 1–6.

[Fattah et al., 2014] Fattah, M., Rahmani, A.-M., Xu, T., Kanduri, A., Liljeberg, P., Plosila, J., and Tenhunen, H. (2014). Mixed-Criticality Run-Time Task Mapping for NoC-Based Many-Core Systems. In *International Conference onParallel, Distributed and Network-Based Processing*, pages 458–465.

[Foutris et al., 2010] Foutris, N., Psarakis, M., Gizopoulos, D., Apostolakis, A., Vera, X., and Gonzalez, A. (2010). MT-SBST: Self-test optimization in multithreaded multicore architectures. In *Proc. Int. Test Conf. (ITC)*, pages 1–10.

[Gnad et al., 2015] Gnad, D., Shafique, M., Kriebel, F., Rehman, S., Sun, D., and Henkel, J. (2015). Hayat: Harnessing Dark Silicon and variability for aging deceleration and balancing. In *Proc. of Design Automation Conf. (DAC)*, pages 1–6.

[Haghbayan et al., 2010] Haghbayan, M., Karamati, S., Javaheri, F., and Navabi, Z. (2010). Test Pattern Selection and Compaction for Sequential Circuits in an HDL Environment. In *Asian Test Symp. (ATS)*, pages 53–56.

[Haghbayan et al., 2014] Haghbayan, M., Rahmani, A., Liljeberg, P., Plosila, J., and Tenhunen, H. (2014). Online Testing of Many-Core Systems in the Dark Silicon Era. In *Proc. Int. Symp. on Design and Diagnostics of Electronic Circuits Systems (DDECS)*, pages 141–146.

[Haghbayan et al., 2015a] Haghbayan, M.-H., Kanduri, A., Rahmani, A.-M., Liljeberg, P., Jantsch, A., and Tenhunen, H. (2015a). MapPro: Proactive Runtime Mapping for Dynamic Workloads by Quantifying Ripple Effect of Applications on Networks-on-Chip. In *Int. Symp. on Networks-on-Chip (NOCS)*, pages 1–8.

[Haghbayan et al., 2015b] Haghbayan, M.-H., Rahmani, A.-M., Fattah, M., Liljeberg, P., Plosila, J., Navabi, Z., and Tenhunen, H. (2015b). Power-aware online testing of manycore systems in the dark silicon era. In *Proc. Int. Conf. on Design, Automation & Test in Europe (DATE)*, pages 435–440.

[Haghbayan et al., 2012] Haghbayan, M. H., Safari, S., and Navabi, Z. (2012). Power constraint testing for multi-clock domain SoCs using concurrent hybrid BIST. In *Proc. Int. Symp. on Design and Diagnostics of Electronic Circuits Systems (DDECS)*, pages 42–45.

[Hennessy and Patterson, 2012] Hennessy, J. and Patterson, D. (2012). Computer architecture, fifth edition: A quantitative approach (the morgan kaufmann series in computer architecture and design). *Elsevier*.

[Howard et al., 2010] Howard, J. et al. (2010). A 48-Core IA-32 message-passing processor with DVFS in 45nm CMOS. In *Proc. Int. Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, pages 108–109.

[Huang and Xu, 2010] Huang, L. and Xu, Q. (2010). Characterizing the lifetime reliability of manycore processors with core-level redundancy. In *Proc. Int. Conf. on Computer-Aided Design (ICCAD)*, pages 680–685.

[JEDEC Solid State Tech. Association, 2010] JEDEC Solid State Tech. Association (2010). Failure mechanisms and models for semiconductor devices. JEP122G.

[Kaliorakis et al., 2014] Kaliorakis, M., Psarakis, M., Foutris, N., and Gizopoulos, D. (2014). Accelerated online error detection in many-core microprocessor architectures. In *Proc. VLSI Test Symp. (VTS)*, pages 1–6.

[Kalray, 2017] Kalray (2017). Kalray MPPA Manycore. http://www.kalrayinc.com/.

[Kanduri et al., 2017] Kanduri, A., Haghbayan, M. H., Rahmani, A. M., Liljeberg, P., Jantsch, A., Tenhunen, H., and Dutt, N. (2017). Accuracy-aware power management for many-core systems running error-resilient applications. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 25(10):2749–2762.

[Kavousianos and Chakrabarty, 2013] Kavousianos, X. and Chakrabarty, K. (2013). Testing for SoCs with advanced static and dynamic power-management capabilities. In *Proc. Conf. on Design, Automation & Test in Europe (DATE)*, pages 737–742.

[Kavousianos et al., 2012] Kavousianos, X., Chakrabarty, K., Jain, A., and Parekhji, R. (2012). Test Schedule Optimization for Multicore SoCs: Handling Dynamic Voltage Scaling and Multiple Voltage Islands. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 31(11):1754–1766.

[Khodabandeloo et al., 2011] Khodabandeloo, B., Hoseini, S., Taheri, S., Haghbayan, M. H., Babaei, M. R., and Navabi, Z. (2011). Online Test Macro Scheduling and Assignment in MPSoC Design. In *Proc. Asian Test Symposium (ATS)*, pages 148–153.

[Lee et al., 2010] Lee, J., Skadron, K., and Chung, S. (2010). Predictive temperature-aware dvfs. *IEEE Transactions on Computers*, 59(1):127–133.

[Lee et al., 2014] Lee, W., Wang, Y., and Pedram, M. (2014). Vrcon: Dynamic reconfiguration of voltage regulators in a multicore platform. In *Design, Automation and Test in Europe Conference and Exhibition (DATE)*, pages 1–6.

[Ma and Wang, 2012a] Ma, K. and Wang, X. (2012a). PGCapping: Exploiting Power Gating for Power Capping and Core Lifetime Balancing in CMPs. In *Proc. Int. Conf. on Parallel Architectures and Compilation Techniques (PACT)*, pages 13–22.

[Ma and Wang, 2012b] Ma, K. and Wang, X. (2012b). PGCapping: Exploiting Power Gating for Power Capping and Core Lifetime Balancing in CMPs. In *Proc. Int. Conf. on Parallel Architectures and Compilation Techniques (PACT)*, pages 13–22.

[Mercati et al., 2014] Mercati, P., Bartolini, A., Paterna, F., Rosing, T., and Benini, L. (2014). A Linux-governor based Dynamic Reliability Manager for android mobile devices. In *Proc. Conf. on Design, Automation & Test in Europe (DATE)*, pages 1–4.

[Moore, 1965] Moore, G. (1965). ramming more components onto integrated circuits. *Electronics, 38(8).*

[Muthukaruppan et al., 2013] Muthukaruppan, T., Pricopi, M., Venkataramani, V., Mitra, T., and Vishin, S. (2013). Hierarchical power management for asymmetric multi-core in dark silicon era. In *Proc. Design Automation Conf. (DAC)*, pages 1–9.

[Pagani et al., 2014] Pagani, S., Khdr, H., Munawar, W., Chen, J.-J., Shafique, M., Li, M., and Henkel, J. (2014). TSP: Thermal Safe Power: Efficient Power Budgeting for Many-core Systems in Dark Silicon. In *Proc. Int. Conf. on Hardware/Software Codesign and System Synthesis (CODES)*, pages 10:1–10:10.

[Pham et al., 2005] Pham, D., Asano, S., Bolliger, M., Day, M., Hofstee, H., Johns, C., Kahle, J., Kameyama, A., Keaty, J., Masubuchi, Y., Riley, M., Shippy, D., Stasiak, D., Suzuoki, M., Wang, M., Warnock, J., Weitzel, S., Wendel, D., Yamazaki, T., and Yazawa, K. (2005). The design and implementation of a first-generation cell processor. In *IEEE International Solid-State Circuits Conference (ISSCC)*, pages 184–592 Vol. 1.

[Poirier et al., 2005] Poirier, C., McGowen, R., Bostak, C., and Naffziger, S. (2005). "power and temperature control on a 90nm itanium reg-family processor". In *IEEE International Solid-State Circuits Conference (ISSCC)*, pages 304–305 Vol. 1.

[Rahmani, 2013] Rahmani, A. (2013). Exploration and design of power-efficient networked many-core systems. *PhD thesis*.

[Rahmani et al., 2016] Rahmani, A., Liljeberg, P., Hemani, A., Jantsch, A., and Tenhunen, H. (2016). *The Dark Side of Silicon*. Springer, Switzerland, 1st edition edition.

[Rahmani et al., 2015] Rahmani, A.-M., Haghbayan, M.-H., Kanduri, A., Weldezion, A., Liljeberg, P., Plosila, J., Jantsch, A., and Tenhunen, H. (2015). Dynamic Power Management for Many-Core Platforms in the Dark Silicon Era: A Multi-Objective Control Approach. In *Proc. Int. Symp. on Low Power Electronics and Design (ISLPED)*, pages 1–6.

[Remarsu and Kundu, 2009a] Remarsu, S. and Kundu, S. (2009a). On process variation tolerant low cost thermal sensor design in 32nm cmos technology. In *ACM Great Lakes Symp*, page 487âĂŞ492.

[Remarsu and Kundu, 2009b] Remarsu, S. and Kundu, S. (2009b). On process variation tolerant low cost thermal sensor design in 32nm cmos technology. In *Proceedings of the 19th ACM Great Lakes Symposium on VLSI (GLSVLSI)*, pages 487–492.

[Sasaki et al., 2006] Sasaki, M., Ikeda, M., and Asada, K. (2006). -1/+0.8 deg;c error, accurate temperature sensor using 90nm 1v cmos for on-line thermal monitoring of vlsi circuits. In *IEEE International Conference on Microelectronic Test Structures*, pages 9–12.

[Semiconductor-Industry-Association et al., 2011] Semiconductor-Industry-Association et al. (2011). Int. technology roadmap for semiconductors (ITRS), 2011 edition.

[Shafique et al., 2014] Shafique, M., Garg, S., Henkel, J., and Marculescu, D. (2014). The EDA Challenges in the Dark Silicon Era: Temperature, Reliability, and Variability Perspectives. In *Proc. Design Automation Conf. (DAC)*, pages 185:1–185:6.

[Sieworek and Swarz, 1982] Sieworek, D. P. and Swarz, R. S. (1982). *The Theory and Practice of Reliable System Design*. Digital Press.

[Skitsas et al., 2013] Skitsas, M., Nicopoulos, C., and Michael, M. (2013). Daemonguard: O/s-assisted selective software-based self-testing for multi-core systems. In *Procedeeings of the IEEE Int. Symp. on Defect and Fault Tolerance in VLSI and Nanotechnology Systems*, pages 45–51.

[Sun et al., 2014] Sun, J., Lysecky, R., Shankar, K., Kodi, A., Louri, A., and Roveda, J. (2014). Workload Assignment Considering NBTI Degradation in Multicore Systems. *Journal Emerg. Technol. Comput. Syst.*, 10(1):4:1–4:22.

[Taylor, 2012] Taylor, M. (2012). Is dark silicon useful? harnessing the four horsemen of the coming dark silicon apocalypse. In *Proc. Design Automation Conference (DAC)*, pages 1131–1136.

[Venkatesh et al., 2010] Venkatesh, G., Sampson, J., Goulding, N., Garcia, S., Bryksin, V., Lugo-Martinez, J., Swanson, S., and Taylor, M. B. (2010). Conservation cores: Reducing the energy of mature computations. In *Proceedings of the Fifteenth Edition of ASPLOS on Architectural Support for Programming Languages and Operating Systems*, ASPLOS XV, pages 205–218, New York, NY, USA. ACM.

[Wang and Skadron, 2012] Wang, L. and Skadron, K. (2012). Dark vs. Dim Silicon and Near-Threshold Computing Extended Results. In *University of Virginia Department of Computer Science Technical Report TR-2013-01*.

[Wang and Skadron, 2013] Wang, L. and Skadron, K. (2013). Implications of the Power Wall: Dim Cores and Reconfigurable Logic. *IEEE Micro*, 33(5):40–48.

[Weldezion et al., 2013] Weldezion, A., Grange, M., Pamunuwa, D., Jantsch, A., and Tenhunen, H. (2013). A scalable multi-dimensional NoC simulation model for diverse spatio-temporal traffic patterns. In *Procedding of the IEEE International 3D Systems Integration Conference*, pages 1–5.

[Weldezion et al., 2009] Weldezion, A., Grange, M., Pamunuwa, D., Lu, Z., Jantsch, A., Weerasekera, R., and Tenhunen, H. (2009). Scalability of network-on-chip communication architecture for 3-D meshes. In *Proc. Int. Symp. on Networks-on-Chip (NOCS)*, pages 114–123.

[Wulf and McKee, 1995] Wulf, W. A. and McKee, S. A. (1995). Hitting the memory wall: Implications of the obvious. *SIGARCH Comput. Archit. News*, 23(1):20–24.

[Xiang et al., 2010] Xiang, Y., Chantem, T., Dick, R., Hu, X., and Shang, L. (2010). System-level reliability modeling for MPSoCs. In *Proc. Conf. on Hardware/Software Codesign and System Synthesis (CODES)*, pages 297–306.

[Zhang et al., 2013] Zhang, Y., Peng, L., Fu, X., and Hu, Y. (2013). Lighting the dark silicon by exploiting heterogeneity on future processors. In *Proceedings of the 50th Annual Design Automation Conference*, DAC '13, pages 82:1–82:7, New York, NY, USA. ACM.

Part II

Original Publications

Paper I

MapPro: Proactive Runtime Mapping for Dynamic Workloads by Quantifying Ripple Effect of Applications on Networks-on-Chip

M.H. Haghbayan, A. Kanduri, A.M. Rahmani, P. Liljeberg, A. Jantsch, H. Tenhunen

Published in IEEE/ACM International Symposium on Networks-on-Chip (NOCS 2015), Canada.

Paper II

Dark Silicon Aware Power Management for Manycore Systems under Dynamic Workloads

M.H. Haghbayan, A.M. Rahmani, A. Yemane, P. Liljeberg, J. Plosila, A. Jantsch, H. Tenhunen

Published in IEEE/ACM The 32nd IEEE/ACM International Conference on Computer Design (ICCD 2014), Korea.

Paper III

Dynamic Power Management for Many-Core Platforms in the Dark Silicon Era: A Multi-Objective Control Approach

A.M. Rahmani, M.H. Haghbayan, A. Kanduri, A. Yemane, P. Liljeberg, J. Plosila, A. Jantsch, H. Tenhunen

Published in IEEE/ACM International Symposium on Low Power Electronics and Design, (ISLPED 2015), Italy.

Paper IV

Reliability-Aware Runtime Power Management for Many-Core Systems in the Dark Silicon Era

A.M. Rahmani, M.H. Haghbayan, A. Miele, P. Liljeberg, A. Jantsch, H. Tenhunen

Published in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, (IEEE-TVLSI 2017).

Paper V

Dark Silicon Aware Runtime Mapping for Many-core Systems: A Patterning Approach

A. Kanduri, M.H. Haghbayan, A.M. Rahmani, P. Liljeberg, A. Jantsch, H. Tenhunen

Published in IEEE/ACM International Conference on Computer Design, (ICCD 2015), USA.

Paper VI

Can Dark Silicon Be Exploited to Prolong System Lifetime?

M.H. Haghbayan, A. Miele, A.M. Rahmani, P. Liljeberg, A. Jantsch, C. Bolchini, H. Tenhunen

Published in IEEE Design and Test of Computers (IEEE-D&T), 2017.

Paper VII

Software-Based On-Chip Thermal Sensor Calibration for DVFS-enabled Many-core Systems

S. Sami Teräväinen, M.H. Haghbayan, A.M. Rahmani, P. Liljeberg, H. Tenhunen

Published in IEEE Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT 2015), USA.

Paper VIII

A Lifetime-Aware Runtime Mapping Approach for Many-core Systems in the Dark Silicon Era

M.H. Haghbayan, A. Miele, A.M. Rahmani, P. Liljeberg, H. Tenhunen

Published in IEEE/ACM Design, Automation, and Test in Europe, (DATE 2016), Germany.

Paper IX

Energy-Efficient Concurrent Testing Approach for Many-Core Systems in the Dark Silicon Age

M.H. Haghbayan, A.M. Rahmani, P. Liljeberg, J. Plosila, H. Tenhunen

Published in IEEE Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT 2014), Netherlands.

Paper X

Power-Aware Online Testing of Manycore Systems in the Dark Silicon Era

M.H. Haghbayan, A.M. Rahmani, M. Fattah, P. Liljeberg, J. Plosila, Z. Navabi, H. Tenhunen

Published in IEEE/ACM the Design, Automation, and Test in Europe (DATE 2015), France.

Paper XI

A Power-Aware Approach for Online Test Scheduling in Many-core Architectures

M.H. Haghbayan, A.M. Rahmani, A. Miele, M. Fattah, J. Plosila, P. Liljeberg, H. Tenhunen

Published in IEEE Transactions on Computers, (IEEE-TC 2016).

Paper XII

Performance/Reliability-aware Resource Management for Many-Cores in Dark Silicon Era

M.H. Haghbayan, A. Miele, A.M. Rahmani, P. Liljeberg, H. Tenhunen

Published in IEEE Transactions on Computers, (IEEE-TC 2017).

Turku Centre for Computer Science
TUCS Dissertations

1. **Marjo Lipponen**, On Primitive Solutions of the Post Correspondence Problem
2. **Timo Käkölä**, Dual Information Systems in Hyperknowledge Organizations
3. **Ville Leppänen**, Studies on the Realization of PRAM
4. **Cunsheng Ding**, Cryptographic Counter Generators
5. **Sami Viitanen**, Some New Global Optimization Algorithms
6. **Tapio Salakoski**, Representative Classification of Protein Structures
7. **Thomas Långbacka**, An Interactive Environment Supporting the Development of Formally Correct Programs
8. **Thomas Finne**, A Decision Support System for Improving Information Security
9. **Valeria Mihalache**, Cooperation, Communication, Control. Investigations on Grammar Systems.
10. **Marina Waldén**, Formal Reasoning About Distributed Algorithms
11. **Tero Laihonen**, Estimates on the Covering Radius When the Dual Distance is Known
12. **Lucian Ilie**, Decision Problems on Orders of Words
13. **Jukkapekka Hekanaho**, An Evolutionary Approach to Concept Learning
14. **Jouni Järvinen**, Knowledge Representation and Rough Sets
15. **Tomi Pasanen**, In-Place Algorithms for Sorting Problems
16. **Mika Johnsson**, Operational and Tactical Level Optimization in Printed Circuit Board Assembly
17. **Mats Aspnäs**, Multiprocessor Architecture and Programming: The Hathi-2 System
18. **Anna Mikhajlova**, Ensuring Correctness of Object and Component Systems
19. **Vesa Torvinen**, Construction and Evaluation of the Labour Game Method
20. **Jorma Boberg**, Cluster Analysis. A Mathematical Approach with Applications to Protein Structures
21. **Leonid Mikhajlov**, Software Reuse Mechanisms and Techniques: Safety Versus Flexibility
22. **Timo Kaukoranta**, Iterative and Hierarchical Methods for Codebook Generation in Vector Quantization
23. **Gábor Magyar**, On Solution Approaches for Some Industrially Motivated Combinatorial Optimization Problems
24. **Linas Laibinis**, Mechanised Formal Reasoning About Modular Programs
25. **Shuhua Liu**, Improving Executive Support in Strategic Scanning with Software Agent Systems
26. **Jaakko Järvi**, New Techniques in Generic Programming – C++ is more Intentional than Intended
27. **Jan-Christian Lehtinen**, Reproducing Kernel Splines in the Analysis of Medical Data
28. **Martin Büchi**, Safe Language Mechanisms for Modularization and Concurrency
29. **Elena Troubitsyna**, Stepwise Development of Dependable Systems
30. **Janne Näppi**, Computer-Assisted Diagnosis of Breast Calcifications
31. **Jianming Liang**, Dynamic Chest Images Analysis
32. **Tiberiu Seceleanu**, Systematic Design of Synchronous Digital Circuits
33. **Tero Aittokallio**, Characterization and Modelling of the Cardiorespiratory System in Sleep-Disordered Breathing
34. **Ivan Porres**, Modeling and Analyzing Software Behavior in UML
35. **Mauno Rönkkö**, Stepwise Development of Hybrid Systems
36. **Jouni Smed**, Production Planning in Printed Circuit Board Assembly
37. **Vesa Halava**, The Post Correspondence Problem for Market Morphisms
38. **Ion Petre**, Commutation Problems on Sets of Words and Formal Power Series
39. **Vladimir Kvassov**, Information Technology and the Productivity of Managerial Work
40. **Frank Tétard**, Managers, Fragmentation of Working Time, and Information Systems

41. **Jan Manuch**, Defect Theorems and Infinite Words
42. **Kalle Ranto**, Z_4-Goethals Codes, Decoding and Designs
43. **Arto Lepistö**, On Relations Between Local and Global Periodicity
44. **Mika Hirvensalo**, Studies on Boolean Functions Related to Quantum Computing
45. **Pentti Virtanen**, Measuring and Improving Component-Based Software Development
46. **Adekunle Okunoye**, Knowledge Management and Global Diversity – A Framework to Support Organisations in Developing Countries
47. **Antonina Kloptchenko**, Text Mining Based on the Prototype Matching Method
48. **Juha Kivijärvi**, Optimization Methods for Clustering
49. **Rimvydas Rukšėnas**, Formal Development of Concurrent Components
50. **Dirk Nowotka**, Periodicity and Unbordered Factors of Words
51. **Attila Gyenesei**, Discovering Frequent Fuzzy Patterns in Relations of Quantitative Attributes
52. **Petteri Kaitovaara**, Packaging of IT Services – Conceptual and Empirical Studies
53. **Petri Rosendahl**, Niho Type Cross-Correlation Functions and Related Equations
54. **Péter Majlender**, A Normative Approach to Possibility Theory and Soft Decision Support
55. **Seppo Virtanen**, A Framework for Rapid Design and Evaluation of Protocol Processors
56. **Tomas Eklund**, The Self-Organizing Map in Financial Benchmarking
57. **Mikael Collan**, Giga-Investments: Modelling the Valuation of Very Large Industrial Real Investments
58. **Dag Björklund**, A Kernel Language for Unified Code Synthesis
59. **Shengnan Han**, Understanding User Adoption of Mobile Technology: Focusing on Physicians in Finland
60. **Irina Georgescu**, Rational Choice and Revealed Preference: A Fuzzy Approach
61. **Ping Yan**, Limit Cycles for Generalized Liénard-Type and Lotka-Volterra Systems
62. **Joonas Lehtinen**, Coding of Wavelet-Transformed Images
63. **Tommi Meskanen,** On the NTRU Cryptosystem
64. **Saeed Salehi**, Varieties of Tree Languages
65. **Jukka Arvo**, Efficient Algorithms for Hardware-Accelerated Shadow Computation
66. **Mika Hirvikorpi**, On the Tactical Level Production Planning in Flexible Manufacturing Systems
67. **Adrian Costea**, Computational Intelligence Methods for Quantitative Data Mining
68. **Cristina Seceleanu**, A Methodology for Constructing Correct Reactive Systems
69. **Luigia Petre**, Modeling with Action Systems
70. **Lu Yan**, Systematic Design of Ubiquitous Systems
71. **Mehran Gomari**, On the Generalization Ability of Bayesian Neural Networks
72. **Ville Harkke**, Knowledge Freedom for Medical Professionals – An Evaluation Study of a Mobile Information System for Physicians in Finland
73. **Marius Cosmin Codrea**, Pattern Analysis of Chlorophyll Fluorescence Signals
74. **Aiying Rong**, Cogeneration Planning Under the Deregulated Power Market and Emissions Trading Scheme
75. **Chihab BenMoussa**, Supporting the Sales Force through Mobile Information and Communication Technologies: Focusing on the Pharmaceutical Sales Force
76. **Jussi Salmi**, Improving Data Analysis in Proteomics
77. **Orieta Celiku**, Mechanized Reasoning for Dually-Nondeterministic and Probabilistic Programs
78. **Kaj-Mikael Björk**, Supply Chain Efficiency with Some Forest Industry Improvements
79. **Viorel Preoteasa**, Program Variables – The Core of Mechanical Reasoning about Imperative Programs
80. **Jonne Poikonen**, Absolute Value Extraction and Order Statistic Filtering for a Mixed-Mode Array Image Processor
81. **Luka Milovanov**, Agile Software Development in an Academic Environment
82. **Francisco Augusto Alcaraz Garcia**, Real Options, Default Risk and Soft Applications
83. **Kai K. Kimppa**, Problems with the Justification of Intellectual Property Rights in Relation to Software and Other Digitally Distributable Media
84. **Dragoş Truşcan**, Model Driven Development of Programmable Architectures
85. **Eugen Czeizler**, The Inverse Neighborhood Problem and Applications of Welch Sets in Automata Theory

86. **Sanna Ranto**, Identifying and Locating-Dominating Codes in Binary Hamming Spaces
87. **Tuomas Hakkarainen**, On the Computation of the Class Numbers of Real Abelian Fields
88. **Elena Czeizler**, Intricacies of Word Equations
89. **Marcus Alanen**, A Metamodeling Framework for Software Engineering
90. **Filip Ginter**, Towards Information Extraction in the Biomedical Domain: Methods and Resources
91. **Jarkko Paavola**, Signature Ensembles and Receiver Structures for Oversaturated Synchronous DS-CDMA Systems
92. **Arho Virkki**, The Human Respiratory System: Modelling, Analysis and Control
93. **Olli Luoma**, Efficient Methods for Storing and Querying XML Data with Relational Databases
94. **Dubravka Ilić**, Formal Reasoning about Dependability in Model-Driven Development
95. **Kim Solin**, Abstract Algebra of Program Refinement
96. **Tomi Westerlund**, Time Aware Modelling and Analysis of Systems-on-Chip
97. **Kalle Saari**, On the Frequency and Periodicity of Infinite Words
98. **Tomi Kärki**, Similarity Relations on Words: Relational Codes and Periods
99. **Markus M. Mäkelä**, Essays on Software Product Development: A Strategic Management Viewpoint
100. **Roope Vehkalahti**, Class Field Theoretic Methods in the Design of Lattice Signal Constellations
101. **Anne-Maria Ernvall-Hytönen**, On Short Exponential Sums Involving Fourier Coefficients of Holomorphic Cusp Forms
102. **Chang Li**, Parallelism and Complexity in Gene Assembly
103. **Tapio Pahikkala**, New Kernel Functions and Learning Methods for Text and Data Mining
104. **Denis Shestakov**, Search Interfaces on the Web: Querying and Characterizing
105. **Sampo Pyysalo**, A Dependency Parsing Approach to Biomedical Text Mining
106. **Anna Sell**, Mobile Digital Calendars in Knowledge Work
107. **Dorina Marghescu**, Evaluating Multidimensional Visualization Techniques in Data Mining Tasks
108. **Tero Säntti**, A Co-Processor Approach for Efficient Java Execution in Embedded Systems
109. **Kari Salonen**, Setup Optimization in High-Mix Surface Mount PCB Assembly
110. **Pontus Boström**, Formal Design and Verification of Systems Using Domain-Specific Languages
111. **Camilla J. Hollanti**, Order-Theoretic Mehtods for Space-Time Coding: Symmetric and Asymmetric Designs
112. **Heidi Himmanen**, On Transmission System Design for Wireless Broadcasting
113. **Sébastien Lafond**, Simulation of Embedded Systems for Energy Consumption Estimation
114. **Evgeni Tsivtsivadze**, Learning Preferences with Kernel-Based Methods
115. **Petri Salmela**, On Commutation and Conjugacy of Rational Languages and the Fixed Point Method
116. **Siamak Taati**, Conservation Laws in Cellular Automata
117. **Vladimir Rogojin**, Gene Assembly in Stichotrichous Ciliates: Elementary Operations, Parallelism and Computation
118. **Alexey Dudkov**, Chip and Signature Interleaving in DS CDMA Systems
119. **Janne Savela**, Role of Selected Spectral Attributes in the Perception of Synthetic Vowels
120. **Kristian Nybom**, Low-Density Parity-Check Codes for Wireless Datacast Networks
121. **Johanna Tuominen**, Formal Power Analysis of Systems-on-Chip
122. **Teijo Lehtonen**, On Fault Tolerance Methods for Networks-on-Chip
123. **Eeva Suvitie**, On Inner Products Involving Holomorphic Cusp Forms and Maass Forms
124. **Linda Mannila**, Teaching Mathematics and Programming – New Approaches with Empirical Evaluation
125. **Hanna Suominen**, Machine Learning and Clinical Text: Supporting Health Information Flow
126. **Tuomo Saarni**, Segmental Durations of Speech
127. **Johannes Eriksson**, Tool-Supported Invariant-Based Programming

128. **Tero Jokela**, Design and Analysis of Forward Error Control Coding and Signaling for Guaranteeing QoS in Wireless Broadcast Systems
129. **Ville Lukkarila**, On Undecidable Dynamical Properties of Reversible One-Dimensional Cellular Automata
130. **Qaisar Ahmad Malik**, Combining Model-Based Testing and Stepwise Formal Development
131. **Mikko-Jussi Laakso**, Promoting Programming Learning: Engagement, Automatic Assessment with Immediate Feedback in Visualizations
132. **Riikka Vuokko**, A Practice Perspective on Organizational Implementation of Information Technology
133. **Jeanette Heidenberg**, Towards Increased Productivity and Quality in Software Development Using Agile, Lean and Collaborative Approaches
134. **Yong Liu**, Solving the Puzzle of Mobile Learning Adoption
135. **Stina Ojala**, Towards an Integrative Information Society: Studies on Individuality in Speech and Sign
136. **Matteo Brunelli**, Some Advances in Mathematical Models for Preference Relations
137. **Ville Junnila**, On Identifying and Locating-Dominating Codes
138. **Andrzej Mizera**, Methods for Construction and Analysis of Computational Models in Systems Biology. Applications to the Modelling of the Heat Shock Response and the Self-Assembly of Intermediate Filaments.
139. **Csaba Ráduly-Baka**, Algorithmic Solutions for Combinatorial Problems in Resource Management of Manufacturing Environments
140. **Jari Kyngäs**, Solving Challenging Real-World Scheduling Problems
141. **Arho Suominen**, Notes on Emerging Technologies
142. **József Mezei**, A Quantitative View on Fuzzy Numbers
143. **Marta Olszewska**, On the Impact of Rigorous Approaches on the Quality of Development
144. **Antti Airola**, Kernel-Based Ranking: Methods for Learning and Performace Estimation
145. **Aleksi Saarela**, Word Equations and Related Topics: Independence, Decidability and Characterizations
146. **Lasse Bergroth**, Kahden merkkijonon pisimmän yhteisen alijonon ongelma ja sen ratkaiseminen
147. **Thomas Canhao Xu**, Hardware/Software Co-Design for Multicore Architectures
148. **Tuomas Mäkilä**, Software Development Process Modeling – Developers Perspective to Contemporary Modeling Techniques
149. **Shahrokh Nikou**, Opening the Black-Box of IT Artifacts: Looking into Mobile Service Characteristics and Individual Perception
150. **Alessandro Buoni**, Fraud Detection in the Banking Sector: A Multi-Agent Approach
151. **Mats Neovius**, Trustworthy Context Dependency in Ubiquitous Systems
152. **Fredrik Degerlund**, Scheduling of Guarded Command Based Models
153. **Amir-Mohammad Rahmani-Sane**, Exploration and Design of Power-Efficient Networked Many-Core Systems
154. **Ville Rantala**, On Dynamic Monitoring Methods for Networks-on-Chip
155. **Mikko Pelto**, On Identifying and Locating-Dominating Codes in the Infinite King Grid
156. **Anton Tarasyuk**, Formal Development and Quantitative Verification of Dependable Systems
157. **Muhammad Mohsin Saleemi**, Towards Combining Interactive Mobile TV and Smart Spaces: Architectures, Tools and Application Development
158. **Tommi J. M. Lehtinen**, Numbers and Languages
159. **Peter Sarlin**, Mapping Financial Stability
160. **Alexander Wei Yin**, On Energy Efficient Computing Platforms
161. **Mikołaj Olszewski**, Scaling Up Stepwise Feature Introduction to Construction of Large Software Systems
162. **Maryam Kamali**, Reusable Formal Architectures for Networked Systems
163. **Zhiyuan Yao**, Visual Customer Segmentation and Behavior Analysis – A SOM-Based Approach
164. **Timo Jolivet**, Combinatorics of Pisot Substitutions
165. **Rajeev Kumar Kanth**, Analysis and Life Cycle Assessment of Printed Antennas for Sustainable Wireless Systems
166. **Khalid Latif**, Design Space Exploration for MPSoC Architectures

167. **Bo Yang**, Towards Optimal Application Mapping for Energy-Efficient Many-Core Platforms
168. **Ali Hanzala Khan**, Consistency of UML Based Designs Using Ontology Reasoners
169. **Sonja Leskinen**, m-Equine: IS Support for the Horse Industry
170. **Fareed Ahmed Jokhio**, Video Transcoding in a Distributed Cloud Computing Environment
171. **Moazzam Fareed Niazi**, A Model-Based Development and Verification Framework for Distributed System-on-Chip Architecture
172. **Mari Huova**, Combinatorics on Words: New Aspects on Avoidability, Defect Effect, Equations and Palindromes
173. **Ville Timonen**, Scalable Algorithms for Height Field Illumination
174. **Henri Korvela**, Virtual Communities – A Virtual Treasure Trove for End-User Developers
175. **Kameswar Rao Vaddina**, Thermal-Aware Networked Many-Core Systems
176. **Janne Lahtiranta**, New and Emerging Challenges of the ICT-Mediated Health and Well-Being Services
177. **Irum Rauf**, Design and Validation of Stateful Composite RESTful Web Services
178. **Jari Björne**, Biomedical Event Extraction with Machine Learning
179. **Katri Haverinen**, Natural Language Processing Resources for Finnish: Corpus Development in the General and Clinical Domains
180. **Ville Salo**, Subshifts with Simple Cellular Automata
181. **Johan Ersfolk**, Scheduling Dynamic Dataflow Graphs
182. **Hongyan Liu**, On Advancing Business Intelligence in the Electricity Retail Market
183. **Adnan Ashraf**, Cost-Efficient Virtual Machine Management: Provisioning, Admission Control, and Consolidation
184. **Muhammad Nazrul Islam**, Design and Evaluation of Web Interface Signs to Improve Web Usability: A Semiotic Framework
185. **Johannes Tuikkala**, Algorithmic Techniques in Gene Expression Processing: From Imputation to Visualization
186. **Natalia Díaz Rodríguez**, Semantic and Fuzzy Modelling for Human Behaviour Recognition in Smart Spaces. A Case Study on Ambient Assisted Living
187. **Mikko Pänkäälä**, Potential and Challenges of Analog Reconfigurable Computation in Modern and Future CMOS
188. **Sami Hyrynsalmi**, Letters from the War of Ecosystems – An Analysis of Independent Software Vendors in Mobile Application Marketplaces
189. **Seppo Pulkkinen**, Efficient Optimization Algorithms for Nonlinear Data Analysis
190. **Sami Pyöttiälä**, Optimization and Measuring Techniques for Collect-and-Place Machines in Printed Circuit Board Industry
191. **Syed Mohammad Asad Hassan Jafri**, Virtual Runtime Application Partitions for Resource Management in Massively Parallel Architectures
192. **Toni Ernvall**, On Distributed Storage Codes
193. **Yuliya Prokhorova**, Rigorous Development of Safety-Critical Systems
194. **Olli Lahdenoja**, Local Binary Patterns in Focal-Plane Processing – Analysis and Applications
195. **Annika H. Holmbom**, Visual Analytics for Behavioral and Niche Market Segmentation
196. **Sergey Ostroumov**, Agent-Based Management System for Many-Core Platforms: Rigorous Design and Efficient Implementation
197. **Espen Suenson**, How Computer Programmers Work – Understanding Software Development in Practise
198. **Tuomas Poikela**, Readout Architectures for Hybrid Pixel Detector Readout Chips
199. **Bogdan Iancu**, Quantitative Refinement of Reaction-Based Biomodels
200. **Ilkka Törmä**, Structural and Computational Existence Results for Multidimensional Subshifts
201. **Sebastian Okser**, Scalable Feature Selection Applications for Genome-Wide Association Studies of Complex Diseases
202. **Fredrik Abbors**, Model-Based Testing of Software Systems: Functionality and Performance
203. **Inna Pereverzeva**, Formal Development of Resilient Distributed Systems
204. **Mikhail Barash**, Defining Contexts in Context-Free Grammars
205. **Sepinoud Azimi**, Computational Models for and from Biology: Simple Gene Assembly and Reaction Systems
206. **Petter Sandvik**, Formal Modelling for Digital Media Distribution

207. **Jongyun Moon**, Hydrogen Sensor Application of Anodic Titanium Oxide Nanostructures
208. **Simon Holmbacka**, Energy Aware Software for Many-Core Systems
209. **Charalampos Zinoviadis**, Hierarchy and Expansiveness in Two-Dimensional Subshifts of Finite Type
210. **Mika Murtojärvi**, Efficient Algorithms for Coastal Geographic Problems
211. **Sami Mäkelä**, Cohesion Metrics for Improving Software Quality
212. **Eyal Eshet**, Examining Human-Centered Design Practice in the Mobile Apps Era
213. **Jetro Vesti**, Rich Words and Balanced Words
214. **Jarkko Peltomäki**, Privileged Words and Sturmian Words
215. **Fahimeh Farahnakian**, Energy and Performance Management of Virtual Machines: Provisioning, Placement and Consolidation
216. **Diana-Elena Gratie**, Refinement of Biomodels Using Petri Nets
217. **Harri Merisaari**, Algorithmic Analysis Techniques for Molecular Imaging
218. **Stefan Grönroos**, Efficient and Low-Cost Software Defined Radio on Commodity Hardware
219. **Noora Nieminen**, Garbling Schemes and Applications
220. **Ville Taajamaa**, O-CDIO: Engineering Education Framework with Embedded Design Thinking Methods
221. **Johannes Holvitie**, Technical Debt in Software Development – Examining Premises and Overcoming Implementation for Efficient Management
222. **Tewodros Deneke**, Proactive Management of Video Transcoding Services
223. **Kashif Javed**, Model-Driven Development and Verification of Fault Tolerant Systems
224. **Pekka Naula**, Sparse Predictive Modeling – A Cost-Effective Perspective
225. **Antti Hakkala**, On Security and Privacy for Networked Information Society – Observations and Solutions for Security Engineering and Trust Building in Advanced Societal Processes
226. **Anne-Maarit Majanoja**, Selective Outsourcing in Global IT Services – Operational Level Challenges and Opportunities
227. **Samuel Rönnqvist**, Knowledge-Lean Text Mining
228. **Mohammad-Hashem Hahgbayan**, Energy-Efficient and Reliable Computing in Dark Silicon Era

Turku Centre *for* Computer Science

http://www. tucs.fi

tucs@abo.fi

University of Turku
Faculty of Mathematics and Natural Sciences

- Department of Information Technology
- Department of Mathematics and Statistics

Turku School of Economics

- Institute of Information Systems Science

Åbo Akademi University
Faculty of Science and Engineering

- Computer Engineering
- Computer Science

Faculty of Social Sciences, Business and Economics

- Information Systems

ISBN 978-952-12-3646-4
ISSN 1239-1883

Mohammad-Hashem Haghbayan

Energy-Efficient and Reliable Computing in Dark Silicon Era