

THESIS

AN INTEGRATED VARIATION-AWARE MAPPING FRAMEWORK FOR FINFET BASED
IRREGULAR 2D MPSOCs IN THE DARK SILICON ERA

Submitted by

Pramit Rajkrishna

Department of Electrical and Computer Engineering

In partial fulfillment of the requirements

For the Degree of Master of Science

Colorado State University

Fort Collins, Colorado

Spring 2016

Master's Committee:

Advisor: Sudeep Pasricha

Anura Jayasumana

Patrick Burns

Copyright by Prमित Rajkrishna 2016

All Rights Reserved

ABSTRACT

AN INTEGRATED VARIATION-AWARE MAPPING FRAMEWORK FOR FINFET BASED IRREGULAR 2D MPSOCs IN THE DARK SILICON ERA

In the deep submicron era, process variations and dark silicon considerations have become prominent focus areas for early stage networks-on-chip (NoC) design synthesis. Additionally, FinFETs have been implemented as promising alternatives to bulk CMOS implementations for 22nm and below technology nodes to mitigate leakage power. While overall system power in a dark silicon paradigm is governed by a limitation on active cores and inter-core communication patterns, it has also become imperative to consider process variations in a holistic context for irregular 2D NoCs. Additionally, manufacturing defects induce link failures, with resultant irregularity in the NoC topology and rendering conventional minimal routing schemes for regular topologies inoperable.

In this thesis, we propose a holistic process variation aware design time synthesis framework (HERMES) that performs computation and communication mapping while minimizing energy consumption and maximizing Power Performance Yield (PPY). The framework targets a 22nm FinFET based homogenous NoC implementation with design time link failures in the NoC fabric, a dark silicon based power constraint and system bandwidth constraints for performance guarantees, while preserving connectivity and deadlock freedom in the NoC fabric. Our experimental results show that HERMES performs $1.32x$ better in energy, $1.29x$ better in simulation execution time and 58.44% better in PPY statistics, over other state-of-the-art proposed mapping techniques for various SPLASH2 and PARSEC parallel benchmarks.

ACKNOWLEDGEMENTS

I would like to thank all the key entities whose encouragement and support has made the completion of this thesis possible. First, I would like to express my sincere gratitude to Prof. Sudeep Pasricha, without whose invaluable guidance this undertaking would not have been possible in the first place. His knowledge in this research area is unparalleled and his expert counsel at key junctures of the research process and incredible patience with my progress has enabled me to bring my thesis to its fruitful conclusion. I can only hope to match the energy with which he juggles a daunting workload of managing research, classes and graduate students with a smile.

Second, I would like to extend my thanks to Prof. Anura Jayasumana and Prof. Patrick Burns for their valuable time to preside in my committee and provide inputs for my thesis. I would like to extend a special note of thanks to Prof. Burns for providing me the opportunity and exposure to university information systems as an analyst and to be a part of the technical team in CSU's ACNS and CASA divisions. I would like to thank my respective supervisors Randy Miotke and Michael Brake and my colleagues for their support during my studies.

Third, I would also like to extend a special note of thanks to Ms. Terry Comerford and my colleagues at the Engineering Success Center for providing me with opportunities to make a positive impact in engaging the engineering student community and being a part of their success. I would also like to thank the talented and wonderful student officers, past and present, of the Engineering College Council (ECC) for making me feel welcome, and for being the amazing folks they are. Third, I would like to thank my fantastic colleagues in the MECS lab, who are extremely

proficient in their research areas and have been very supportive during my entire time spent there. I will miss the banter, camaraderie and group meetings with them.

Fourth, I would also like to thank my colleagues, especially my supervisor Mr. Andy Bourke, Mr. Tony Aug (Vice President and Chief Technology Officer, Arrow Digital Group) and Mr. Matt Anderson (President and Chief Digital Officer, Arrow Digital Group) at Arrow Electronics for providing me the next quantum leap in my career and the necessary impetus to complete my degree.

Finally and most importantly, I would like to extend my thanks to my parents for encouraging me to strive for excellence in my academic and professional pursuits and my sister, who has completed her Masters in Civil Engineering at CSU and was an incredible support to me in the entire time that we spent here. I would also like to thank all my close friends, who have provided me the moral support when I needed it the most.

TABLE OF CONTENTS

ABSTRACT	ii
ACKNOWLEDGEMENTS	iii
TABLE OF CONTENTS.....	v
LIST OF TABLES	vii
LIST OF FIGURES	viii
1 Introduction.....	1
1.1 Motivation	3
1.2 Contributions	8
1.3 Outline.....	10
2 Background	11
2.1 Network Architecture	11
2.1.1 Router Architecture.....	12
2.1.2 FinFET based NoC implementation	14
2.2 Process Variation Model.....	15
2.2.1 Variation Modeling	16
2.2.2 NoC Architecture Variation Modeling	17
2.3 Power Evaluation Model.....	18
2.3.1 Compute Cores	19
2.3.2 NoC Fabric	19
3 Problem Statement.....	22
4 Related Work	24

4.1 Work on Process Variation Modeling, FinFETs	24
4.2 Work on Dark Silicon based Exploration	25
4.2 Work on Routing Techniques for Irregular Topologies	25
4.3 Work on Application Mapping for Various Optimization Objectives	26
5 Methodology	29
5.1 Problem Formulations and Assumptions.....	29
5.1.1 Generation of NoC and Compute Core Power Maps for 100 Dies	31
5.1.2 Tile Selection for Mapping	32
5.1.4 Routing Path Allocation and Selection	38
5.1.5 Best Case System Energy and Overall PPY Evaluation with Constraints	41
6 Results	43
6.1 Simulation Setup	43
6.1.1 Benchmarks, NoC and Evaluation Framework	43
6.1.2 Comparison of HERMES with proposed prior techniques.....	44
6.2 Simulation Results.....	46
6.2.1 Process Variation based Power Maps.....	46
6.2.2 Total Power and System Energy Consumption	47
6.2.3 Power Performance Yield (PPY)	50
6.2.4 Sensitivity Analysis - Execution Time	53
6.2.5 Sensitivity Analysis - System energy	56
6.2.6 Sensitivity Analysis – PPY.....	58
7 Conclusions and Future Work	62
References	64

LIST OF TABLES

Table 1: Mean total power and standard deviation values for NoC components	46
Table 2: Execution times for the PARSEC and SPLASH-2 suites [100 of 144 cores]	48

LIST OF FIGURES

Figure 1: A 4x4 NoC regular mesh topology with cores and routers and links.....	2
Figure 2: Routing Algorithm Classification	3
Figure 3: Illustration of a) tile to tile (<i>T2T</i>) variations, b) die to die (<i>D2D</i>) variations (represented by different colored dies on the wafer) and c) wafer to wafer (<i>W2W</i>) variations (represented by different colored wafers).....	4
Figure 4: Planar CMOS (left) vs FinFET (right) comparison [12].....	5
Figure 5: Intel FinFET Implementation Roadmap [13].....	6
Figure 6: Samsung FinFET Implementation Roadmap [13].....	7
Figure 7: Illustration of mapping and routing of 16 application cores (green circles) to a 5x5 irregular 2D MPSoC with dark silicon constraints and process variations on the tiles. The figure shows both active (red) and inactive (grey) compute cores, with ‘x’ indicating failed links and blue dotted arrows representing bi-directional turn restrictions.	9
Figure 8: Network parameters [24].....	11
Figure 9: Compute core detailed parameters [25].	12
Figure 10: A 4x4 Networks-On-Chip regular mesh topology with detailed router architecture. .	13
Figure 11: Illustration of a 5x5 Networks-On-Chip irregular mesh topology with inactive links and inactive cores due to a dark silicon power budget.	13
Figure 12: Representation of FinFET geometry (left) and three modes [shorted gate (SG), independent gate (IG) and asymmetric-workfunction SG (ASG)] of implementation (right). [24]	14
Figure 13: Detailed FinFET parameters [24] and [25]	15

Figure 14: McPAT-PVT model [25].....	17
Figure 15: FinCANON (ORION-PVT and CACTI-PVT) model [24]	18
Figure 16: Design Flow of the HERMES synthesis framework.....	30
Figure 17: Core selection pseudo code	33
Figure 18: Genetic Algorithm Pseudo code.....	34
Figure 19: Genetic Algorithm Flowchart	35
Figure 20: Generic Two point crossover with illegal mappings.....	36
Figure 21: Illustration of partially mapped crossover.....	36
Figure 22: Illustration of Hash Gene generation [38].....	37
Figure 23. Example of Segment Routing (SR) based segments and routing restrictions (blue arrows) for a 5x5 irregular mesh topology with 10% faults (failed links indicated by red crosses on links), process variations and dark silicon based active (red) and inactive (gray) cores.....	39
Figure 24: The generated power maps for the respective leakage (left) and peak dynamic components (right) of the various compute core and NoC components with respective sigma (standard deviation) values.....	47
Figure 25: Total power consumption across all 100 dies for all eight benchmarks for 100 active cores from a base system size of 144 cores	49
Figure 26: Best case system energy consumption for all eight benchmarks for 100 active cores from a base system size of 144 cores.....	50
Figure 27 : Power Performance Yield (PPY) for the <i>comm_int_low</i> benchmark set with a stringent power constraint of 170 W and a nominal power constraint of 180 W	51
Figure 28: Power Performance Yield (PPY) for the <i>comm_int_high</i> benchmark set with a stringent power constraint of 230 W and a nominal power constraint of 240 W	52

Figure 29: Sensitivity analysis of execution time with varying system sizes having 64, 100, 144 and 256 cores respectively for *blackscholes* and *dedup* with 10% faults..... 54

Figure 30: Sensitivity analysis of execution time with varying fault rates of 10%, 20% and 30% respectively for *blackscholes* and *dedup* with 100 active cores from 144 cores..... 55

Figure 31: Sensitivity analysis of energy consumption with varying system sizes having 64, 100,144 and 256 cores respectively for *blackscholes* and *dedup* with 10% faults..... 56

Figure 32: Sensitivity analysis of energy consumption with varying fault rates of 10%, 20% and 30% respectively for *blackscholes* and *dedup* with 100 active cores from 144 cores..... 57

Figure 33: Sensitivity analysis of PPY with varying system sizes across 64, 100, 144 and 256 cores respectively for *blackscholes* and *dedup* with 10% faults..... 59

Figure 34: Sensitivity analysis of PPY with varying fault rates of 10%, 20% and 30% respectively for *blackscholes* and *dedup* with 100 active cores from 144 cores..... 60

1 Introduction

Multi-processor system-on-chip (MPSoC) based designs are projected to exhibit an incredible trend of increasing size, complexity and heterogeneity over the next decade, with tens of hundreds of cores present on a single die. Network-on-chip (NoC) based fabrics have emerged as a significant design paradigm to establish an efficient communication framework between the cores, as an alternative to bus based designs for MPSoCs, while maintaining low-power consumption and exhibiting fault-tolerance[1] – [3] . Regular tile-based mesh NoC architectures have been proposed as a solution to meet the demanding performance requirements of increasingly complex systems [4] and [5]. A conventional regular mesh NoC architecture consists of a connected matrix of IP cores. Each core contains a compute core connected via a network interface (NI) to an embedded router that performs the function of connecting each core with the neighboring cores via I/O ports. This connection is done via bidirectional links between the corresponding routers, which serve the function of ensuring communication flow between the cores, as shown in Figure 1.

Secondly, the routing algorithm determines the path taken by a packet between a pair of communicating nodes in the network, while ensuring low latency and deadlock freedom. Such NoC routing algorithms can be classified into two categories: (a) deterministic routing schemes or (b) adaptive routing schemes [6]. Deterministic routing schemes use fixed, predefined (usually minimal length) paths between routers, without taking the state of the network into consideration. On the other hand, adaptive schemes are classified into two further subcategories: partially adaptive schemes and fully adaptive schemes. Adaptive schemes introduce path diversity and selection logic between routers, and are generally preferable, as they provide a more suitable

lower-latency routing path that takes the state of the network into account at runtime when making decisions.

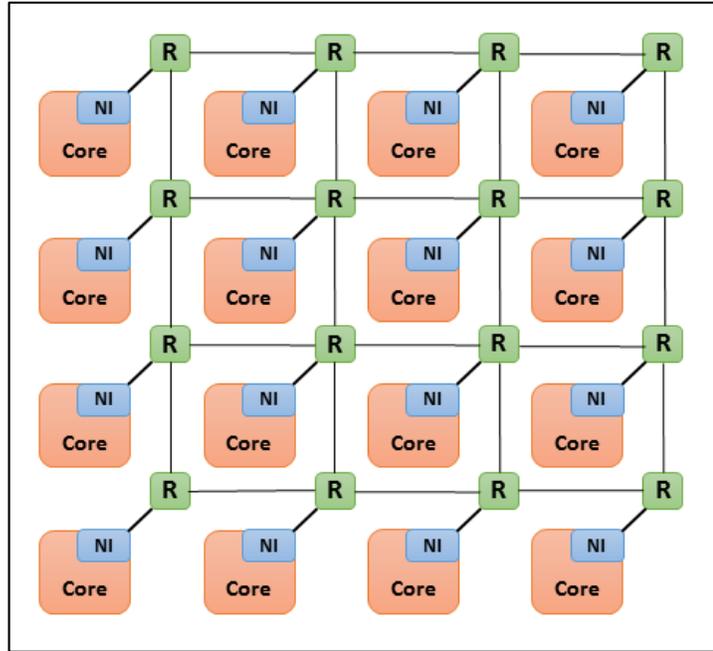


Figure 1: A 4x4 NoC regular mesh topology with cores and routers and links.

Routing schemes can also be categorized as source or distributed, where source routing schemes store the entire path in the packet header while distributed schemes store the final destination in the packet header. This makes distributed routing generally preferable in the context of reduction in packet header size, reduction of storage requirements at the source router and improving adaptivity, while introducing the added logic complexity of the overhead of route computation and selection at each router. Finally, routing algorithms can also be classified into two further categories based on the regularity of topologies to which they are applied. Conventional regular topology based routing schemes like dimension order routing (XY), Odd Even and West First are not applicable to irregular topologies, which have component failures or

unevenly sized components. Topology agnostic routing schemes like UP/DOWN, FX and Segment-based Routing (SR) [33] have been proposed that ensure connectivity in the network in addition to providing deadlock freedom and provisioning routing paths. The various classifications of the routing algorithms are presented in Figure 2.

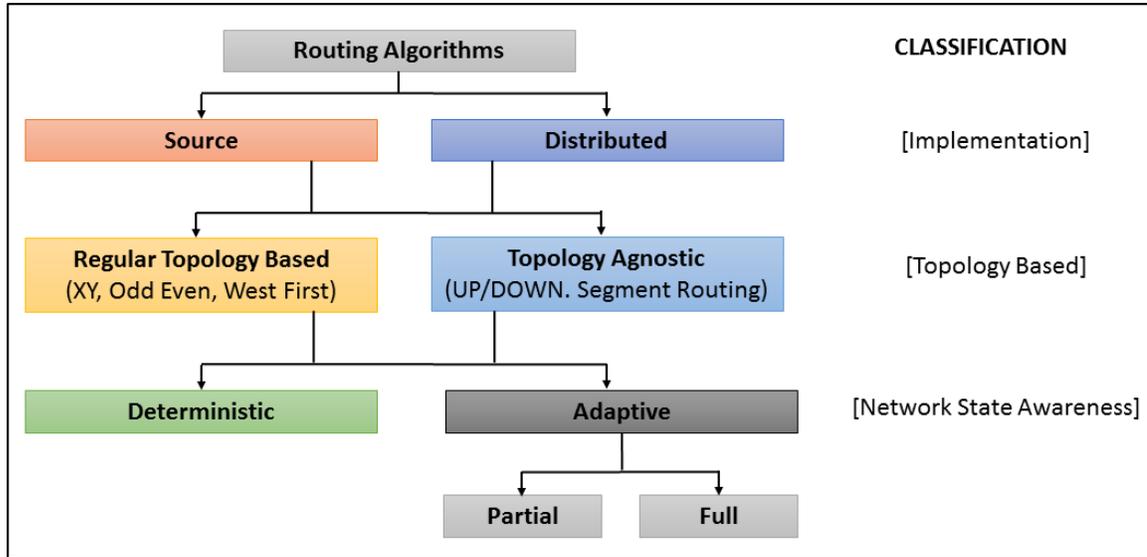


Figure 2: Routing Algorithm Classification

1.1 Motivation

Increasing core counts and decreasing transistor dimensions, especially in the ultra-deep submicron (UDSM) era, have brought into focus three emergent and key problems in MPSoC design: a) process variations that manifest as variations within tiles and between dies, resulting in significant variability in circuit parameters [8] - [10], b) a limit on the total number of tiles that can be switched on simultaneously to meet die power (dark-silicon) constraints [14] - [18], and c) design time component failures, resulting in irregularity of the NoC topology [19]. It has become

imperative to explore these effects simultaneously and in detail, to incorporate their impact in early system level design exploration, especially in the UDSM era.

For MOSFET devices, process variations manifest in three forms (Figure 3): i) spatially correlated *WID* (within-die) variations manifesting as variations between different tiles – constituted by a compute core, network interface and router, manifesting as tile-to-tile (*T2T*) variations on the die, ii) *D2D* (die-to-die) variations that exist as offsets between transistor parameters between two separate dies on the same wafer, and iii) *W2W* (wafer-to-wafer) variations that also exist as offsets between die/transistor parameters between two separate wafers. Process variations are typically manifested as variations in threshold voltage (V_T) and effective channel length (L_{eff}) of transistors, resulting in considerable variability in leakage power and circuit delay [7].

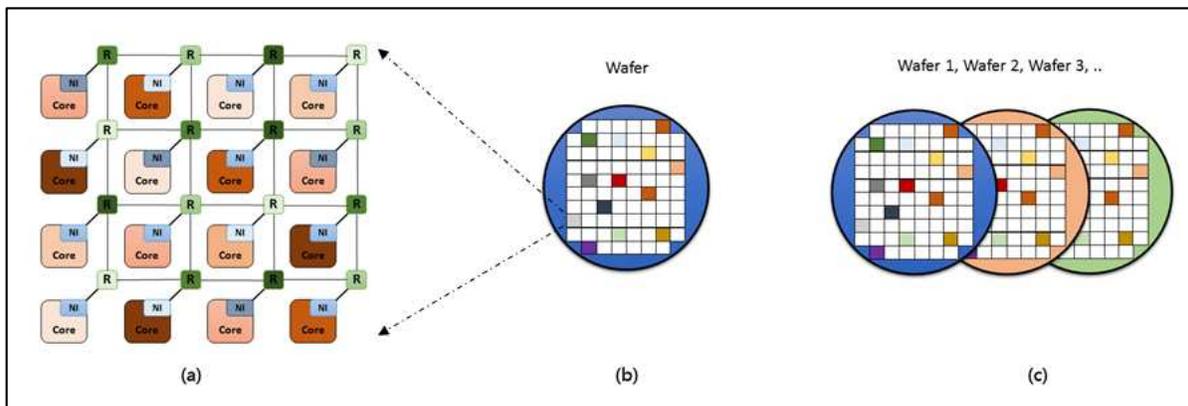


Figure 3: Illustration of a) tile to tile (*T2T*) variations, b) die to die (*D2D*) variations (represented by different colored dies on the wafer) and c) wafer to wafer (*W2W*) variations (represented by different colored wafers)

Process variations at the system level are primarily classified into i) systematic and ii) random variations. Systematic variations are caused due to lithographic constraints and aberrations, while

random variations, such as line edge roughness, are caused by random statistically fluctuating effects and are more pronounced at very low feature sizes. It has also been observed with constantly decreasing transistor dimensions that the random variation component is expected to play a greater role in process variations, than systematic variations [7].

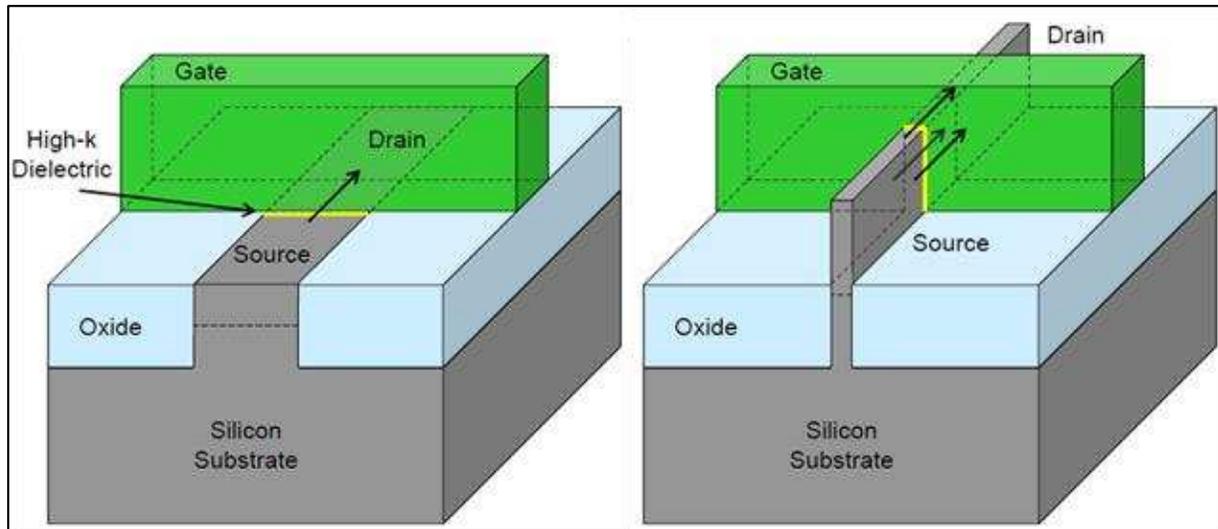


Figure 4: Planar CMOS (left) vs FinFET (right) comparison [12]

In sub 22 nm technology nodes, static (leakage) power consumption has been observed as a key problem area in implementations with traditional planar MOSFETs. To overcome this problem, FinFET based devices have emerged as an effective design paradigm in comparison to bulk CMOS for mitigating leakage power, as shown in Figure 2. FinFETs are double gate based devices with “fins” to increase the overall channel width, enabling better control over the channel current and minimizing leakage [11]. FinFET implementations have also begun replacing traditional bulk CMOS at sub 22 nm nodes, due to better scalability, control of short channel effects, and provisioning for higher transistor density. With the eventual move of the semiconductor industry from planar CMOS to FinFETs, it is crucial to take this new design

paradigm into consideration for any early design space exploration of a system. However, even with the significant advantages presented by FinFET based implementations, process variations still remain a significant problem area and challenge in circuit design with FinFETs. Modeling and mitigating the impact of process variations (both *WID* and *D2D*) for FinFETs is essential for efficient system-level with these devices [10].

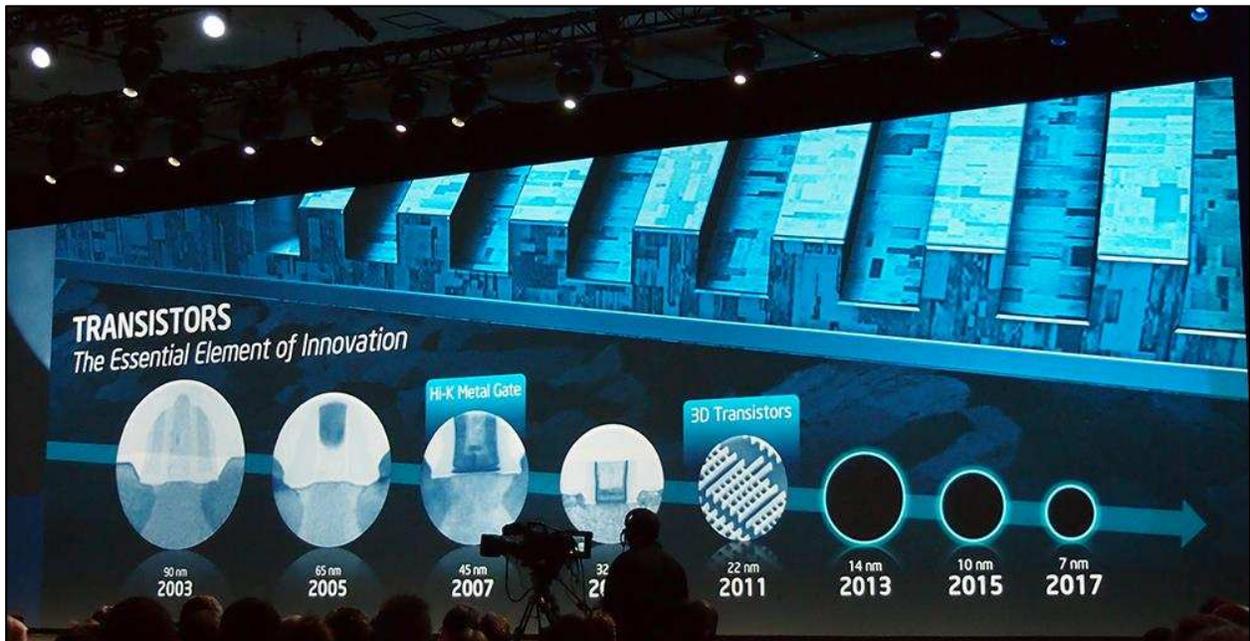


Figure 5: Intel FinFET Implementation Roadmap [13]

Additionally, using ITRS device scaling projections, multicore scaling is projected to hit a “power wall” in which only a specified portion of cores can be powered on simultaneously, based on the system power budget, referred to as the “dark silicon” paradigm [14][17]. With the failure of Dennard scaling, threshold voltage cannot be scaled without exponentially increasing leakage and needs to be held constant, leading to significantly larger sections of the chip to be powered off to meet the chip power budget. It has been projected that at 22 nm, 21% of the chip size must be

powered off to meet power constraints and this figure is projected to increase to 50% at 8 nm [15]. Since future designs will be power limited, it is important to model and examine the effects of process variations on FinFET based implementations in the presence of die power constraints.

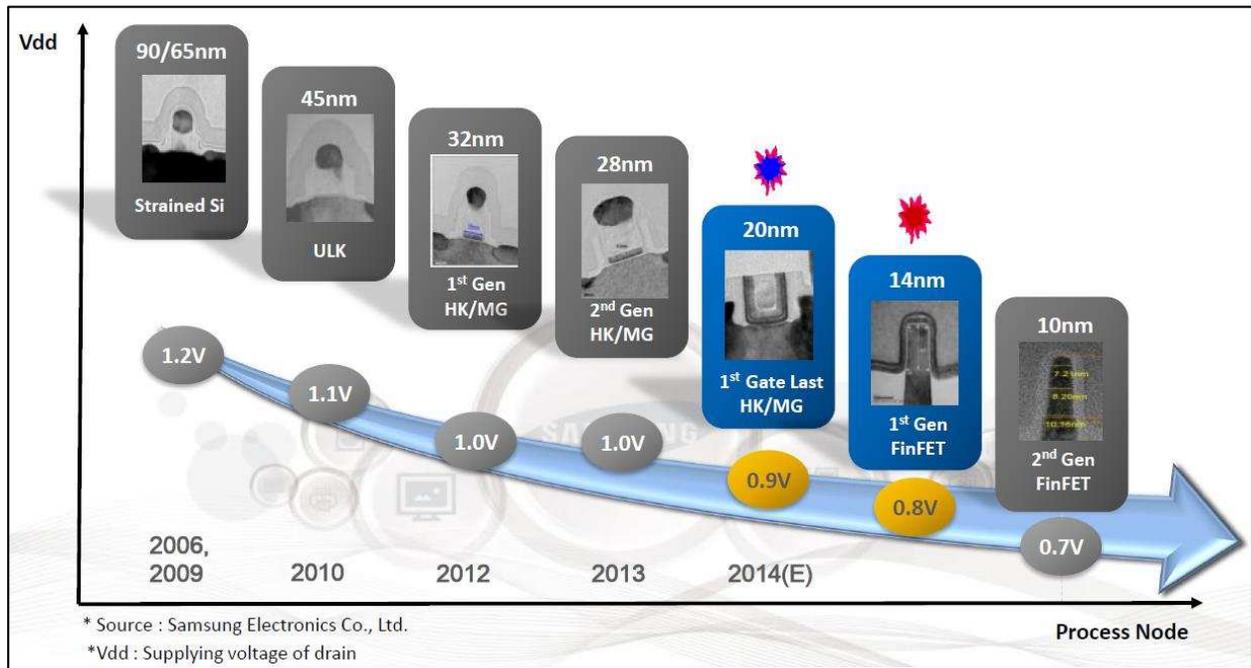


Figure 6: Samsung FinFET Implementation Roadmap [13]

Finally, manufacturing defects in UDSM technology nodes are becoming more and more common, and can induce failed links between cores leading to an irregular NoC topology on the die. This irregularity prevents the adoption of typical NoC routing schemes that are designed for regular topologies (e.g., dimension order XY routing scheme), as they cannot operate viably and in a deadlock-free manner in irregular topologies [30] - [33]. Thus defects necessitate the selection and integration of a suitable topology agnostic NoC routing algorithm in MPSoCs, to provision a suitable routing path that can satisfy design objectives, in the presence of irregularities in the NoC framework. The selection of a topology agnostic routing scheme should also preserve connectivity

in the network, provide deadlock freedom, and offer path diversity for routing. The necessity of path diversity is important in order to explore and select an appropriate minimal routing path between communicating tiles, to balance communication traffic and costs.

All of these considerations necessitate a comprehensive approach as part of early design space exploration for application mapping and subsequent routing on to irregular NoCs with a set of inactive (faulty) links, taking into consideration process variations and dark silicon constraints. Existing application mapping techniques, outlined in [34]-[43], focus on design time mapping and routing on NoCs to optimize various metrics of interest such as energy and latency on regular as well as irregular NoC topologies. Application mapping techniques in [49]-[53] focus on the implications of process variations in the context of application mapping on regular NoC topologies. However, none of these prior works consider the combined effects of dark silicon, FinFET based implementations, process variation implications for application mapping, and routing path selection in the presence of failed links, for irregular NoC topologies. Our work is the first to examine a combined consideration of process variation modeling and dark silicon constraints jointly in early design exploration of parallel applications to be mapped on irregular mesh-based NoC topologies, while also using FinFET devices.

1.2 Contributions

The novel contributions of our framework are summarized as follows:

- We design a holistic design-time synthesis framework (HERMES) to map cores running parallel embedded applications to tiles on a FinFET-based 2D irregular mesh NoC die, while optimizing system energy and maximizing Power Performance Yield (PPY), for a given dark-silicon power budget.

- We design a custom GA implementation with a fast hash function to preserve solution diversity and deliver superior solution quality, along with a min-cost-max-flow (MCF) approach to select the most appropriate bandwidth-constrained minimal path from possible paths for all the pairs of communicating tiles.
- Our framework explores the combined effects of process variations, for both within-die (WID) and die-to-die (D2D) aspects on an irregular mesh NoC, with design time faults at the 22 nm technology node, and perform statistical power modeling of the compute cores and the NoC fabric for evaluation as part of our task and communication mapping strategies.
- We perform a comprehensive comparison of our framework with techniques proposed in prior work using selected parallel application benchmarks (from SPLASH-2 and PARSEC suites), and perform sensitivity analyses for varying dark silicon based core selection scenarios, application sizes, and power budgets.

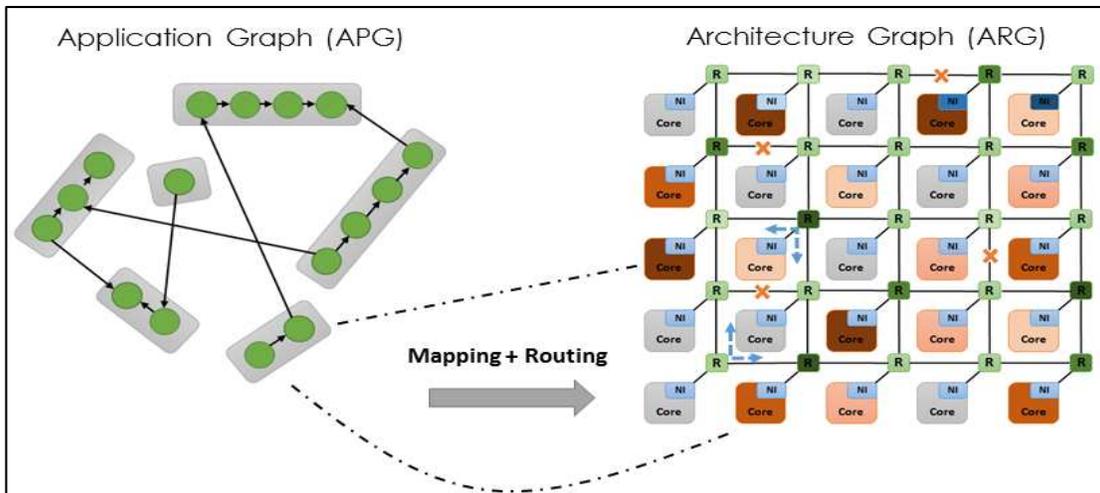


Figure 7: Illustration of mapping and routing of 16 application cores (green circles) to a 5x5 irregular 2D MPSoC with dark silicon constraints and process variations on the tiles. The figure shows both active (red) and inactive (grey) compute cores, with 'x' indicating failed links and blue dotted arrows representing bi-directional turn restrictions.

1.3 Outline

The rest of the thesis is organized as follows:

- Chapter 2 provides a comprehensive overview on process variation modeling, FinFET based processor and NoC models, task mapping methodologies and routing schemes.
- Chapter 3 reviews the existing work done on process variation modeling, FinFETs, the power evaluation model and the topology agnostic routing scheme in consideration.
- Chapter 4 describes the problem statement for the thesis.
- Chapter 5 provides the detailed methodology for evaluation of the framework.
- Chapter 6 presents the simulation setup and results.
- Chapter 7 concludes the thesis with a summary and future work.

2 Background

In this section, we describe the FinFET based network architecture, the process variation model, the power evaluation model for its communication network and the yield metric under consideration for the design space exploration.

2.1 Network Architecture

NETWORK PARAMETERS	
Network parameters	Value
V_{DD} (V)	0.9
Technology	22nm FinFET
Router ports	3~5
Flit size	128 bits
Vcs per port	8
Buffers per port	48 flits (dynamically allocated)
Arbiter model	Round Robin
Crossbar model	Matrix

Figure 8: Network parameters [24].

At the system level, the network architecture under consideration is composed of $m \times n$ tiles, interconnected as a 2-D mesh network. Each tile is composed of a compute core (Core) and a router (R) connected, via a network interface (NI). The router is connected to the four neighboring tiles and its local processing element by bidirectional links via the I/O ports and a matrix crossbar switching fabric is used in the router. The arbiter model for the switch allocator used in the network employs a fair round robin mechanism. The data packets in the network are broken down into “flits” or “flow control bits” and wormhole based flow control is considered for data flow in the

network. The parameters for the network architecture and the compute core are shown in Figure 8 and Figure 9, respectively.

PROCESSOR CONFIGURATION PARAMETERS	
Parameter	Value
Processor design	Alpha
Number of cores	1
Number of ALUs per core	4
Number of FPUs per core	1
L1 instruction cache capacity (<i>KB</i>)	64
L1 data cache capacity (<i>KB</i>)	64
L2 cache capacity (<i>MB</i>)	1.75
Cache block size (<i>bytes</i>)	16
Cache associativity	8
Cache access model	UCA
Coherence protocol	MOESI protocol
Technology	22nm FinFET
Frequency (<i>GHz</i>)	1.2
V_{DD} (<i>V</i>)	0.9
Temperature (<i>K</i>)	358
$3\sigma/\mu$	10%

Figure 9: Compute core detailed parameters [25].

2.1.1 Router Architecture

The network router under consideration is divided into two parts, based on functional units: control path and data path. The control path units comprise of the route computation, virtual channel allocation (VCA) and switch allocation (SA) units. The data path units are the I/O ports, I/O buffers, and the switching fabric. The control path units collectively form the “intelligence” for routing the flits through the network. The route computation logic is the routing table/logic for that tile, which establishes the destination tile based on the source tile for the incoming flits. The

VCA selects the virtual channel, if used by the routing mechanism, to be used by the incoming flits. The SA selects the I/O ports on the switch to be used by the incoming flits. The detailed NoC router architecture is shown in Figure 10.

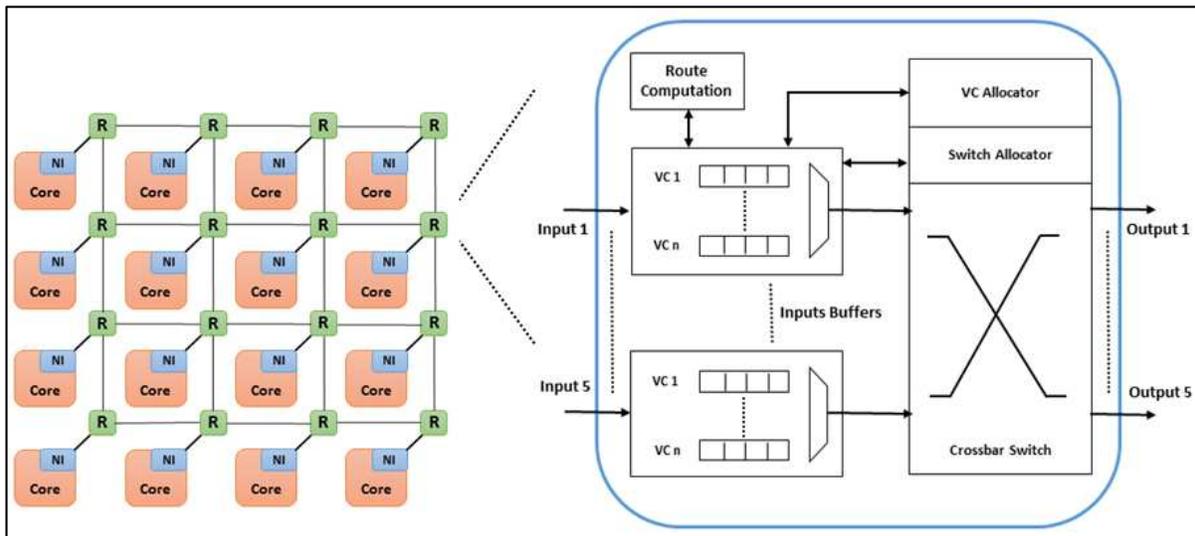


Figure 10: A 4x4 Networks-On-Chip regular mesh topology with detailed router architecture.

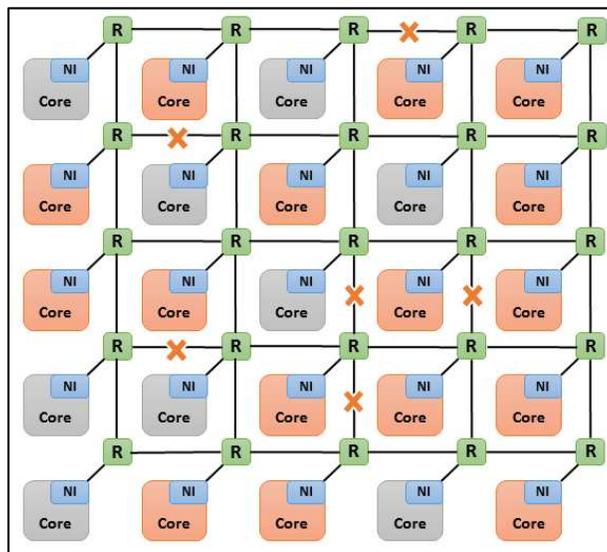


Figure 11: Illustration of a 5x5 Networks-On-Chip irregular mesh topology with inactive links and inactive cores due to a dark silicon power budget.

2.1.2 FinFET based NoC implementation

At the device level, the entire NoC architecture is assumed to be implemented with FinFET based technology. FinFET devices employ a thin fin as the channel body, where the gate length (L_G) is equal to the fin length, with front and back gates. The fin thickness (T_{SI}) is small in comparison to the gate length, ensuring effective control by the gate over the channel, as shown in Figure 12(a). FinFETs can be implemented in three different inverter modes: Shorted-gate (SG), independent-gate (IG) and asymmetric-workfunction shorted gate (ASG), as shown in Figure 12(b). In the SG mode, the front and back gates are shorted together. This leads to both higher on state and leakage current values, while being less than planar CMOS implementations. In the IG mode, the back gates are subject to a reverse bias low voltage instead of being shorted, leading to much lower leakage current than IG mode. This comes at the cost of a higher implementation area, due to the extra contacts required for the back gates.

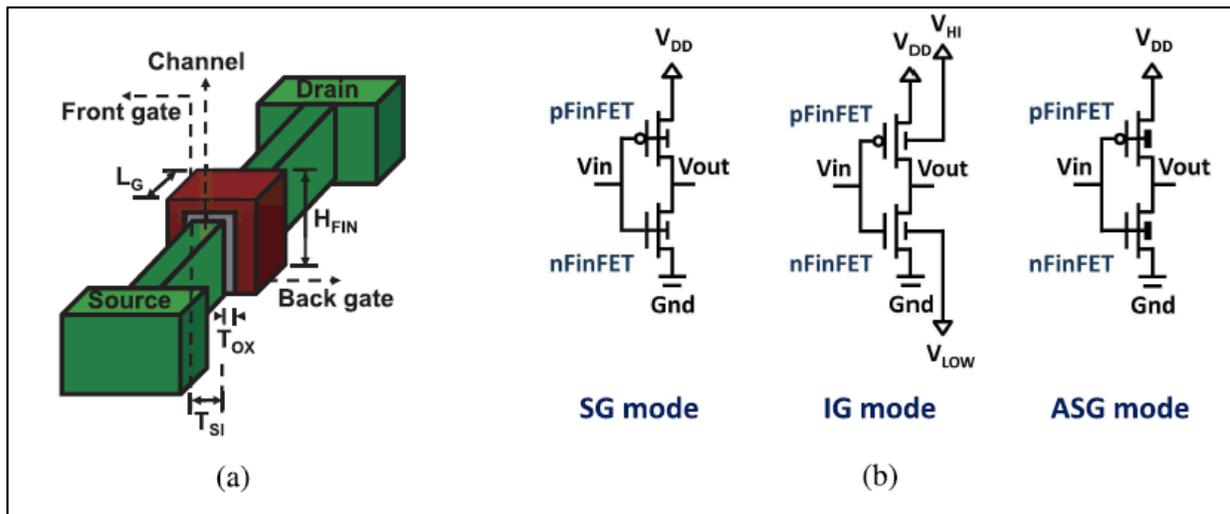


Figure 12: Representation of FinFET geometry (left) and three modes [shorted gate (SG), independent gate (IG) and asymmetric-workfunction SG (ASG)] of implementation (right). [24]

In the ASG mode, the front and back gates are tied together like SG, but have different work functions. The work function of a metal gate is defined as the minimum energy required to move an electron from the gate to a point in the vacuum, just outside the gate. The ASG mode provides the best tradeoff between the low leakage current of the IG mode and comparably high on-currents of the SG mode, and is the implementation of choice for our network architecture. The advantages offered by FinFETs over conventional planar CMOS implementations are shorter delay, higher on-state vs off-state current ratio, and significantly lower leakage. The transistor design parameters for the FinFET considered for the implementation are presented in Figure 13.

22-nm FinFET DESIGN PARAMETERS	
FinFET parameters	Value
Fin height, H_{FIN} (nm)	40
Gate length, L_G (nm)	20
Fin thickness, T_{SI} (nm)	10
Oxide thickness, T_{OX} (nm)	1
Body doping, N_{BODY} (cm^{-3})	10^{16}
Source/drain doping, N_{SD} (cm^{-3})	10^{20}
Front/back gate thickness, H_{GF}, H_{GB} (nm)	34
Underlap near source/drain, L_{UN} (nm)	5
Fin pitch, F_P (nm)	60
Supply voltage, V_{DD} (V)	0.9
Workfunctions (SG & IG mode), Φ_{GN}/Φ_{GP} (eV)	n/p-type: 4.4/4.8
Workfunctions (ASG mode), Φ_{GF}/Φ_{GB} (eV)	n-type: 4.4/4.8; p-type: 4.8/4.4

Figure 13: Detailed FinFET parameters [24] and [25]

2.2 Process Variation Model

With decreasing transistor dimensions in the deep sub-micron era, especially in the 32 nm technology nodes and below, factoring in process variations have become important considerations

in circuit design. Process variations are a combination of two components: a) systematic variations due to lithographic constraints and b) random variations due to effects such as dopant density fluctuations. With the semiconductor industry exhibiting a clear trend of moving to FinFET based designs, it has become imperative to characterize the effect of process variations on network architectures implemented using these devices, for a more comprehensive early design space exploration.

2.2.1 Variation Modeling

In our work, process variations are characterized by two key components: a) WID (within-die) variations and b) D2D (die-to-die) variations, via an existing model called VARIUS, built using the R statistical modeling tool [21]. The mathematical representation of the variation of a given circuit parameter P, is given by:

$$\Delta P = \Delta P_{WID} + \Delta P_{D2D} = (\Delta P_{sys} + \Delta P_{rand}) + \Delta P_{D2D} \dots (2.1)$$

In Equation 1, the variation in circuit parameter P is represented as the total of the within-die variation components ΔP_{WID} (due to systematic variations ΔP_{sys} and random variations ΔP_{rand}) and the die to die variation component ΔP_{D2D} . WID variations for circuit parameters are modeled as a combination of systematic and random components, while the D2D variations are modeled as an offset value. According to empirical data and ITRS projections for variability, the impact of the systematic and random components are considered to be equal at 32 nm, and has been assumed to be the same for the 22 nm technology node considered in this paper. For the WID variation model, transistor parameters are modeled to be spherically correlated with a distance coefficient (ϕ). The

systematic and random components are considered to be normally distributed and independent of each other, resulting in the total WID component to also be considered as distributed normally.

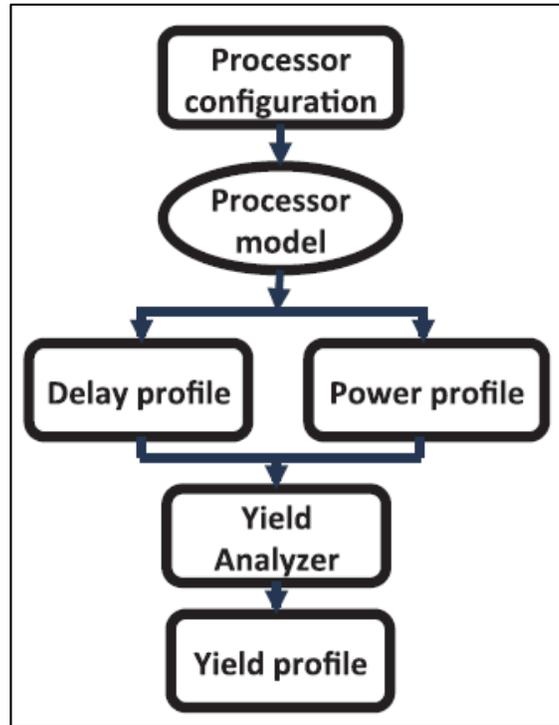


Figure 14: McPAT-PVT model [25]

2.2.2 NoC Architecture Variation Modeling

The various components of the network architecture are modeled using mean power and standard deviation values from two FinFET based compute core, cache and NoC modeling simulation frameworks [24] and [25]. The compute cores are modeled from extensions to the McPAT tool [25] and the NoC architecture is modeled using extensions to ORION [23] and CACTI [27] modeling tools, incorporating the effects of process variations on the various functional units of the processor, NoC components and cache respectively. The generated power profiles have mean peak dynamic and leakage components, with their separate standard deviation

values for statistical modeling. The compute and NoC simulation frameworks, named McPAT-PVT and FinCANON are illustrated in Figure 14 and Figure 15, respectively. The power statistics for the various NoC architecture components are generated as part of the design space framework, by a modified VARIUS implementation that accepts the mean power values and their respective standard deviations to model the variations in the power values, for both the WID and D2D components.

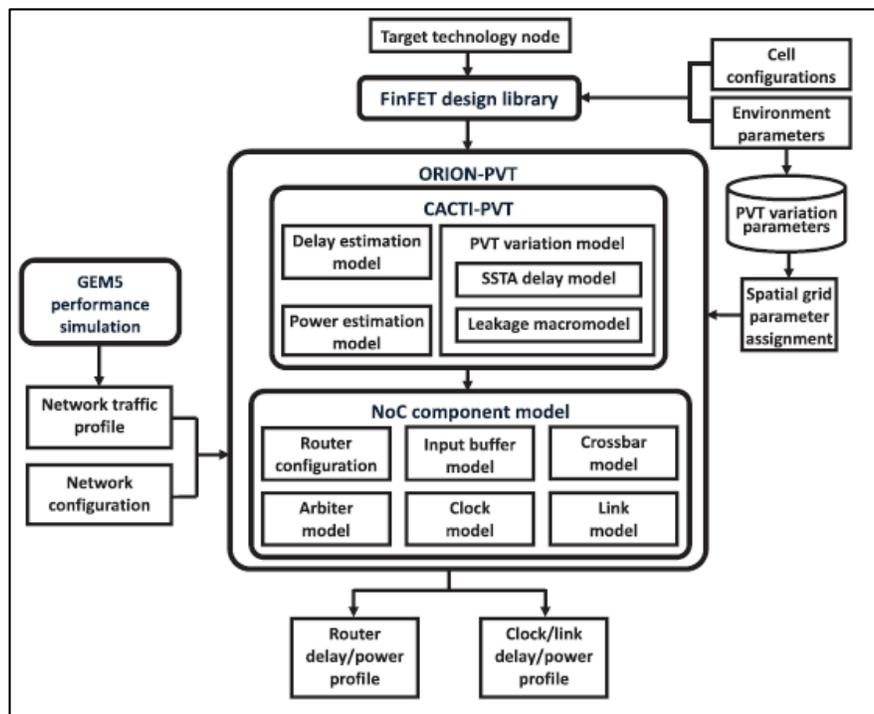


Figure 15: FinCANON (ORION-PVT and CACTI-PVT) model [24]

2.3 Power Evaluation Model

The power evaluation mechanism for the network architecture is modeled in two parts: a) power consumption of the compute cores and b) power consumption of the network fabric. The

combined power consumption of the system is given by the sum of the compute and NoC elements as,

$$P_{Combined} = P_{Total}^{Compute} + P_{Total}^{NoC} \quad \dots (2.2)$$

2.3.1 Compute Cores

The total power consumption P_{Total} for a single core on the die is modeled as the summation of its leakage power P_L and peak dynamic power P_D values, generated from McPAT-PVT [15]. The peak dynamic power component is chosen (to model the worst case dynamic power consumption scenario). The total peak dynamic power figure will change from core to core, due to variation induced deviations from the mean of the peak dynamic power values for the leakage and dynamic components. Mathematically,

$$P_{Tile}^{Compute} = P_L + P_D \quad \dots (2.3)$$

For a total of C cores selected out of N total cores on a die due to power budget constraints, the total power is computed as the summation of the individual total power consumption values of the selected cores on the die. The equation can be represented as,

$$P_{Total}^{Compute} = \sum_{i=1}^C P_{Li} + P_{Di} \quad \dots (2.4)$$

2.3.2 NoC Fabric

A model for energy consumption of network routers was proposed in [27]. Using this model, the power consumed by a bit of data moving through the network can be determined. A bit of data

being transported through a router has power consumption based on classification of router functional units into two categories: a) control path and b) data path. As the data path components significantly dominate over control path components in the power consumption statistics by an order of magnitude [27], the control path components have been ignored for this exploration. The data path components are primarily comprised of the switch, I/O buffers, and internal interconnection wires inside the switching fabric and their bit power consumption statistics are designated as P_{Switch} , P_{Buffer} and P_{Wire} , respectively. A bit of data being transported between routers also incurs a power cost on the link connecting the routers and is designated as P_{Link} . The average power consumed in sending one bit of data between routers on neighboring tiles, with a single hop distance between them, is established as:

$$P_{bit} = P_{Switch} + P_{Buffer} + P_{Wire} + P_{Link} \quad \dots (2.5)$$

Again, as the NoC link power consumption is orders of magnitude more than the power consumed by the internal wires in the switching fabric and the buffers [27], the equation reduces to

$$P_{bit} = P_{Switch} + P_{Buffer} + P_{Link} \quad \dots (2.6)$$

For one bit data to be transferred between two tiles t_a to t_b with n hops between them,

$$P_{bit}(between\ t_a\ and\ t_b) = n * (P_{Switch} + P_{Buffer}) + (n - 1) * P_{Link} \quad \dots (2.7)$$

For B bits of data to be transferred in a single communication between two tiles t_a to t_b with n hops between them,

$$P_{comm}(\text{between } t_a \text{ and } t_b) = B * [n * (P_{Switch} + P_{Buffer}) + (n - 1) * P_{Link}] \dots (2.8)$$

For M total communications on the NoC fabric, the total power consumed in the entire NoC fabric can be represented as,

$$P_{Total}^{NoC} = \sum_{j=1}^M P_{comm_j} \dots (2.9)$$

2.4 Power Performance Yield

In order to evaluate the number of dies meeting the performance and power constraints, a proposed design metric called *PPY* (Power-Performance Yield) [29] is used to evaluate the number of dies out of a set number of test dies that meet the design constraints. *PPY* is calculated as the percentage of test dies out of the total number of dies that meet the power and performance constraints. This metric is extremely effective for estimating the most accurate yield of test dies due to simultaneous consideration of power and performance constraints.

For X dies that meet the power performance constraint out of Y total test dies, the *PPY* is defined as,

$$PPY = \left(\frac{X}{Y} * 100\right) \dots (2.10)$$

3 Problem Statement

The previous chapters established the context of several key issues for the design time synthesis in the context of application mapping and routing on NoC subject to process variations and manufacturing issues, affecting the normal operating and regularity of the NoC fabric respectively. First, the increasing effect of process variations on the operating parameters of the chip, especially in the deep submicron nodes, cause of significant concern in modeling modern NoC architectures for early design space exploration. Second, modeling NoC architectures for novel device architectures has become imperative as more system implementation trends move towards FinFET based designs. Third, with a limit on the total chip area that can be powered on due to dark silicon considerations, design time exploration strategies need to incorporate power constraints in the design process. These concerns merit a holistic early design space exploration strategy that incorporates all of these considerations in the design flow.

For early design space exploration, application mapping to the tiles on the network architecture is performed to generate the core to tile mapping that corresponds to the optimization objective of the system. The mapping strategies are primarily classified into a) numerical techniques like Integer Linear Programming (ILP) based techniques and b) evolutionary techniques like Particle Swarm Optimization (PSO) and Simulated Annealing (SA). With increasing system sizes trending towards integrating hundreds of cores on chip, the choice of numerical techniques as a mapping strategy would be infeasible in terms of time consumption, as application mapping is a NP hard problem and the algorithm time increases exponentially with system size [25]. However, while evolutionary techniques may not produce the exact mapping solution for a given optimization

objective, they are very scalable with system size in comparison with other numerical techniques and provide a good solution with comparable solution quality.

Again, with induced irregularities in the network fabric of the NoC, it has also become imperative to select an appropriate topology agnostic routing scheme to maintain connectivity and deadlock freedom in the network fabric, while ensuring optimized communication flows between the mapped tiles and meeting design constraints. Also, with the presence of multiple paths between tiles, appropriate path selection strategies also need to be explored to select the most appropriate path for mapping the communication flows. These combined considerations merit the combined integration of an effective routing scheme and path selection strategy in the exploration framework that performs comparably with regular routing schemes.

The goal of this thesis and the objective of our framework is to **generate a one-to-one core to tile mapping**, and **map the communication flows between cores to the tiles** on a 2-D irregular mesh NoC, statistically modeled with process variations in a 22nm FinFET implementation, for a given set of parallel applications, **such that the system energy is minimized and PPY is maximized**, while the network connectivity and deadlock freedom is guaranteed, and the link bandwidth and dark silicon based power (computation and communication) constraints are met. We also statistically model the distributions of the power elements for the various network architecture components for a 22 nm FinFET implementation and use that in the various stages of the design time framework.

4 Related Work

A significant body of work has been done for process variation characterization, application mapping and topology agnostic routing algorithms in recent years. The exploration areas can be detailed as (1) Work on process variation modeling and FinFETs, (2) Work on dark silicon based exploration, (3) Work on routing techniques for irregular topologies, and (4) Work on application mapping for various optimization objectives.

4.1 Work on Process Variation Modeling, FinFETs

There has been a substantial amount of exploration in the areas the mathematical modeling and characterization of process variations for planar CMOS and FinFET based-designs. The impact of process variations and power/performance implications on multicore architectures based on planar CMOS was first studied in [7]. The usage of spherical and normal distribution models to characterize WID and D2D process variations for planar CMOS implementations was proposed and modeled using the R statistical package in [21]. The leakage power implications for FinFET based devices over conventional bulk CMOS devices were explored in [11]. The effect of WID process variations on FinFET based architectures was explored and characterized in detail in [8]. Again, in [24] and [25], detailed WID process variation modeling of a FinFET based processor, cache and various NoC components was performed from to generate power and delay values with respective deviations from their mean and peak dynamic values. However, none of these efforts have modeled process variations for FinFET based devices incorporating both WID and D2D variations. For the first time, as part of our proposed HERMES framework, we perform variation modeling of FinFET based NoC implementations that considers both WID and D2D variations.

4.2 Work on Dark Silicon based Exploration

The impact of “dark silicon” on the active area of a die was explored in detail across various benchmarks and architectures by Burger et al. in [14]. Taylor [15] explores various CMOS design strategies for adapting to dark silicon based considerations and outlines a set of dark silicon based design principles to offset the effects caused due to the failure of Dennard scaling. Process variation aware mapping problems have been explored in detail in [16]-[18]. In [49] and [50], Kapadia et al. characterize process variations by modeling V_T variations using the statistical model proposed in [21] and formation of voltage islands to maximize power performance yields statistics for various benchmarks. In [51], Mazjoub et al. propose a VI-aware (voltage-island-aware) and core-placement approach to achieve a balance between limits of spatial extent of the WID variations across cores, and communication patterns between VIs, by varying the shape of VIs to minimize total power. In [53], Wang et al. propose a mechanism to optimize performance yield by utilizing multiple test-chips representative of the spatially correlated systematic WID variations in addition to random variations. *However, there is no existing work that explores the concept of dark silicon with process variations, in the context of FinFET based devices. In our framework, we model FinFET based irregular NoCs, in addition to considerations of design time faults, process variations and dark silicon power constraints, for a more comprehensive and evolutionary framework.*

4.2 Work on Routing Techniques for Irregular Topologies

There is also a substantial body of work undertaken to address the problem of routing packets for a NoC topology with an irregular configuration. In [30], Schroder et al. proposed a deadlock free technique for constructing routing tables for an irregular NoC topology using a breadth first search (BFS) algorithm to maintain connectivity in an irregular topology by creating a breath first

spanning (BFS) tree and then placing bidirectional routing restrictions to ensure deadlock freedom. However, this technique causes significant traffic imbalance in the network by concentrating all traffic around the root node of the BFS tree. In order to mitigate this problem, a new technique to preserve connectivity was proposed in [31] and [32], designated Segment-based Routing (SR). SR starts with a root node and performs a random search for segments to visit all the network nodes and links. After the search process is completed and the segments are computed, bidirectional routing restrictions are placed in every computed segment by the search process. The placement of the bi-directional turn restrictions in a segment are independent of turn restrictions placement in the other computed segments. This maintains deadlock freedom while enabling greater freedom in the placement of routing restrictions in the computed segments to maintain better traffic balance in the NoC. A comprehensive study on the performance of various topology agnostic routing schemes was studied in [33], which established that Segment-based Routing offered better overall performance, among all the evaluated topology agnostic routing algorithms. *In our framework, we extend the concept of SR by implementing a min cost max flow (MCF) based path selection mechanism as an additional layer to select the best bandwidth constrained path to perform routing between communicating tile pairs.*

4.3 Work on Application Mapping for Various Optimization Objectives

Finally, there has also been extensive exploration into a diverse range of application mapping techniques with different optimization objectives. Mapping techniques can be primarily classified into three key categories: (1) techniques like ILP (Integer Linear Programming) and UNISM [36] that produce an exact solution to the problem with a given optimization objective, (2) Evolutionary techniques like GA (Genetic Algorithm), PSO (Particle Swarm Optimization) and SA (Simulated Annealing) that produce a “good” enough, if not exact, optimal solutions that meet the

optimization objectives while maintaining constraints, and (3) techniques like CART [41] that use algorithms to perform iterative improvement of solutions, with defined termination logic to converge on a possible “close to optimal” solution.

With trends indicating increasing system sizes into tens of hundreds of cores, techniques producing an exact solution are increasingly becoming infeasible due to extremely large simulation timings to compute the solution to the optimization problem. This has resulted in evolutionary and specialized techniques being increasingly used as a preferred option to explore the solution space to generate a suitable solution for the optimization problem with reasonable simulation timings. In [34]-[43] and [48], various linear and evolutionary techniques such as ILP, GA, PSO and SA are used to explore the solution space for various constrained single objective and multi objective optimization problems to optimize metrics like energy, delay and power. In [34], Hu et al. propose a branch and bound technique to perform energy and bandwidth aware mapping for regular mesh networks to optimize communication energy while maintaining performance by link bandwidth restrictions, which forms the basis of our exploration. This is compared to a conventional simulated annealing approach that is also used to find the bandwidth constrained, energy optimal one to one core to tile mapping solution, for a regular homogenous NoC. In [41] and [42], a local search technique is employed to perform bandwidth constrained mapping to minimize NoC power consumption, with a single pair of cores being swapped at a time for each iteration of the algorithm, similar to the SA approach, until the power value settles and no further improvements are observed. A set of random seeds are used to perturb the core tile mapping a set number of iterations, every time the value settles to give enough flexibility to explore the entire solution space. In [48], a modified PSO technique is proposed using the concept of designating velocity values as core swaps between tiles, for updating the particle’s position for every epoch to maintain the ordered

nature of a mapping and to avoid illegal mappings (same core mapped to different tiles). *While the above techniques cover process variations related application scheduling and mapping, our work is the first to perform holistic integration of process variations in the context of FinFET devices and dark silicon regime based power budgets in a framework for early design space exploration on irregular mesh NoCs with design time faults, due to manufacturing defects.*

5 Methodology

In this chapter, we describe the flow of our HERMES framework. We first describe the problem formulation, followed by the variation – aware generation of the power maps for dies, power-aware core selection, custom Genetic Algorithm (GA) for core-to-tile mapping, and finally the topology agnostic routing scheme with min cost max flow (MCF) based routing path selection.

Our proposed framework introduces novelty in the form of an integrated tile selection, task mapping, communication routing, and path selection strategy in the context of process variation modeling on FinFET based NoC architectures with a dark silicon power constraint. Our framework emphasizes system energy minimization along with maximizing power performance yield (*PPY*).

5.1 Problem Formulations and Assumptions

We assume the following inputs to our problem:

- A 2D die with an irregular 2D mesh NoC, defined as an architecture graph (ARG) $G(T,L)$, with dimensions (m, n) and number of tiles $T = m \times n$, with each tile composed of a compute core connected to a NoC router via a network interface (NI) and L total links connecting the tiles. A percentage of the L links is considered to be inactive due to manufacturing defects, inducing irregularity in the NoC fabric by the loss of some connectivity across the network. A fixed and rated core frequency f , is considered for the all compute cores in the set of tiles T , the routers, and the links for a given die, and subsequently for all the 100 dies under consideration.
- A power map [*P-Map*] set (peak dynamic $\{P_D\}$, leakage $\{P_L\}$) is defined for components for an individual die, constituting the overall distribution modeling WID (within-die)

process variations. A set of power maps are then generated for the 100 dies under consideration, with an offset value to model D2D variations.

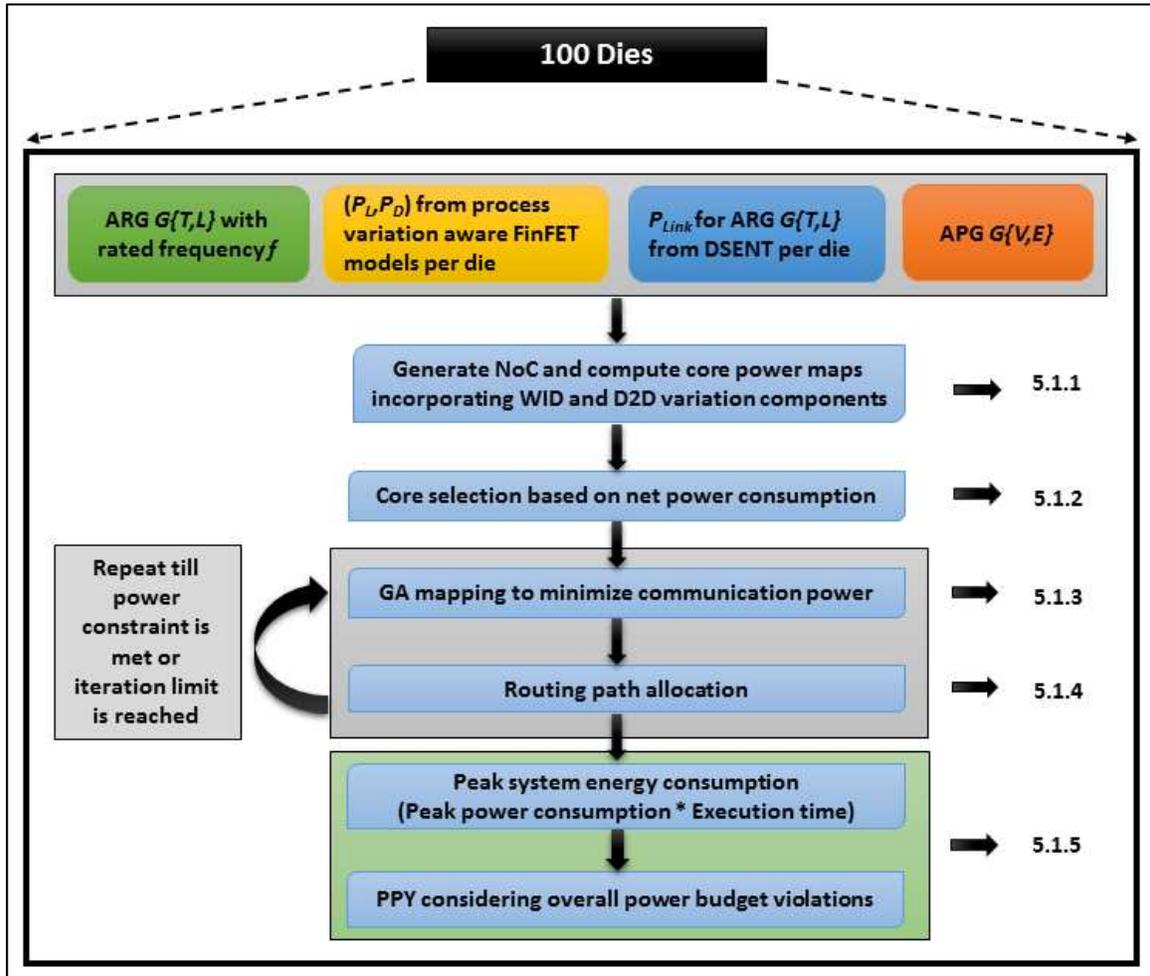


Figure 16: Design Flow of the HERMES synthesis framework

- A set of link bandwidth constraints BW , defined as $\{BW_1, BW_2, \dots, BW_L\}$ for the given operational subset of L links. The constraint values will be used to evaluate bandwidth violations on the links by the communicating cores on the NoC framework and are

dependent on the routing configuration applied to the network. The link power statistics P_{Link} are generated for varying link lengths with NoC size.

- An application graph (APG) $G(V,E)$ with a set of V vertices representing homogenous application cores on which tasks have been mapped, and a set of edges E representing communication volumes between communicating cores. The set of application graphs under consideration are assumed to be independent sets of parallel tasks to model concurrent communications.

Objective: Given the above inputs, the objective of the framework is to generate an one-to-one core to tile mapping, and map the communication flows between cores to the tiles on a 2-D irregular mesh NoC, statistically modeled with process variations in a 22nm FinFET implementation, for a given set of parallel applications, such that the system energy is minimized and PPY is maximized, while the network connectivity and deadlock freedom is guaranteed, and the link bandwidth and dark silicon constraints are satisfied.

5.1.1 Generation of NoC and Compute Core Power Maps for 100 Dies

The power maps for the compute core and NoC components are modeled using the VARIUS [21], FinCANON [24], and McPAT-PVT [25] tools, for all the 100 dies that are considered as inputs to our framework. VARIUS is built using the R statistical package using the *geoR* and *RandomFields* packages to model the spherical distribution of normal distributions of parameters. It incorporates both the WID (within-die) and D2D (die-to-die) variations and models both systematic and random variations and generates a V_T map based on the WID standard deviation (σ_{WID}), D2D standard deviation (σ_{D2D}) and spherical correlation coefficient (ϕ) values. McPAT-PVT and FinCANON are extensions to the original tools, taking into consideration process variations on FinFET based processor, cache and NoC implementations.

McPAT-PVT and FinCANON generate a set of mean peak dynamic and leakage power statistics with their respective standard deviation values (σ_{WID}) for the processor and the various NoC components, respectively. We have modified VARIUS to generate power maps directly for the various die components and for multiple dies from the mean power and standard deviation values from McPAT-PVT and FinCANON.

5.1.2 Tile Selection for Mapping

The algorithm to perform selection of tiles for mapping applications is outlined in Figure 17. Once the power values are assigned, the compute cores are then sorted on the basis of their overall power (*dynamic + leakage*) consumption values (**step 1 and 2**). The overall power values for the individual compute cores in a die will be different due to process variations. As core to tile mapping is performed on a one-to-one basis, the total necessary number and location of tiles are selected from the sorted set $\{(P_{DI}, P_{LI}) \dots (P_{DT}, P_{LT})\}$, on the basis of the total number of cores in the application core graph (**step 3**). The number of selected tiles for mapping are the ones with the least total power values from the ranked set. This results in the selection of the compute cores that minimize overall compute power (**step 4**). The NoC fabric, including the attached routers to the unused compute cores is assumed to be active, to preserve connectivity and reachability in the entire die.

5.1.3 GA-based Mapping Algorithm

In this step, we perform communication power minimization between the selected tiles for mapping, by using a custom Genetic Algorithm (GA) heuristic. The various steps for the GA are outlined in Figure 18 and are as follows:

- **Initial Population Generation** – The GA takes an initial set of randomly assigned ordered core to tile mappings, as the input population to the algorithm. The tiles in the mappings

are the selected tiles from the tile selection portion of the framework, with the least total compute power consumption. The initial set of application core-tile mappings is generated using the inbuilt C++ rand function, to generate the desired number of mappings in the population. The initial crossover and mutation probabilities, P_c and P_m are set to chosen values and the prime number p is selected for generating the hash values for removing duplicates in the solution (**step 1**).

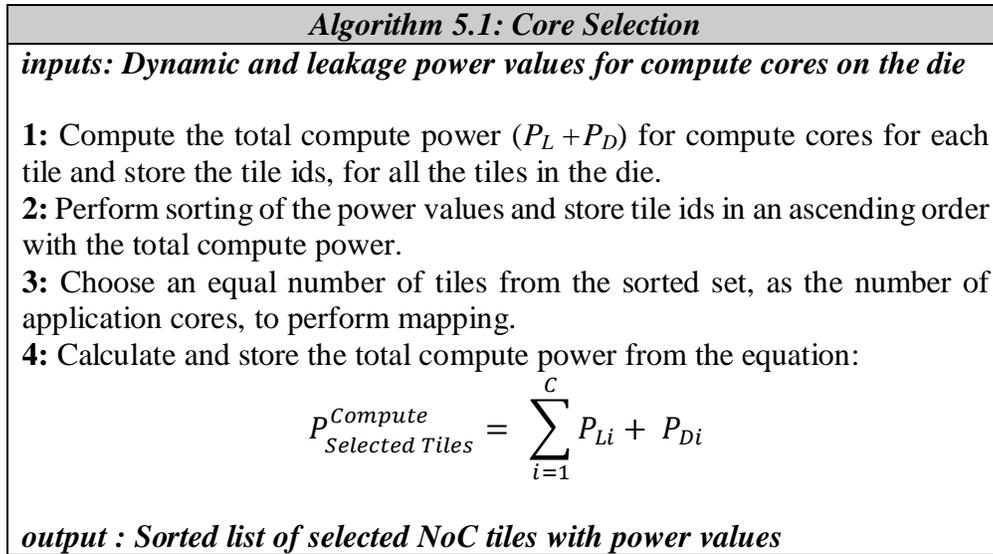


Figure 17: Core selection pseudo code

- **Cost Evaluation** – Each mapping is then evaluated for the total communication power and the cost is stored with the mapping. The cost of each mapping is evaluated for each set of communicating pairs over the NoC data path components with the min cost max flow based multi commodity flow (MCF) technique [36] (Section 4.5), for a suitable bandwidth constrained path. In the event a suitable path is not found, the mapping is discarded and a new mapping is evaluated for a solution. A ranking function is then used to sort the

mappings based on their costs and the best mapping with the cost is stored to be compared with the best mapping of subsequent iterations of the algorithm (**step 2 and 8**). The mapping with the best communication cost is stored for every iteration of the algorithm (**step 11**).

<i>Algorithm 5.2: Custom Genetic Algorithm</i>
<p><i>inputs: set of core tile mappings with communication power values</i></p> <p>1: Generate initial random population P of individuals of size N, set crossover and mutation probabilities P_c and P_m and an initial prime number p for hash generation.</p> <p>2: Evaluate the cost of the individuals for every communicating core-tile pair and store the initial best mapping.</p> <p>3: for number of generations $< max_generations$</p> <p>4: while size of new population $P' < N$</p> <p>5: Select two members from P using tournament selection.</p> <p>6: if ($P_{Xover} > P_c$), perform two point crossover for the members.</p> <p>7: if ($P_{Mutation} > P_m$), perform random swap on the new members.</p> <p>8: Generate hash values using p for new members and add to P', if there are no duplicates and update the best mapping.</p> <p>9: end while</p> <p>10: end for</p> <p>11: Evaluate and return the best mapping.</p> <p><i>output : core tile mapping with minimum communication power</i></p>

Figure 18: Genetic Algorithm Pseudo code

- **Candidate Selection** – The selection of candidates for crossover and mutation is done by binary tournament selection (without replacement) [38]. Tournament selection is performed by running tournaments between several randomly chosen mappings in the population and selecting the ones with the best fitness cost for crossover. Tournament selection is a very efficient way to perform candidate selection among the GA population set, with the selection pressure being the probability of individuals being selected. The

selection pressure can be adjusted by simply changing the tournament size to expand the search space and allow for weaker individuals to be selected. In our GA framework, we perform binary tournament selection without replacement, by choosing the best from two selected mappings and then taking the chosen mapping out of contention, for the remaining tournaments to prevent the mapping being selected more than once for the next generation (step 5).

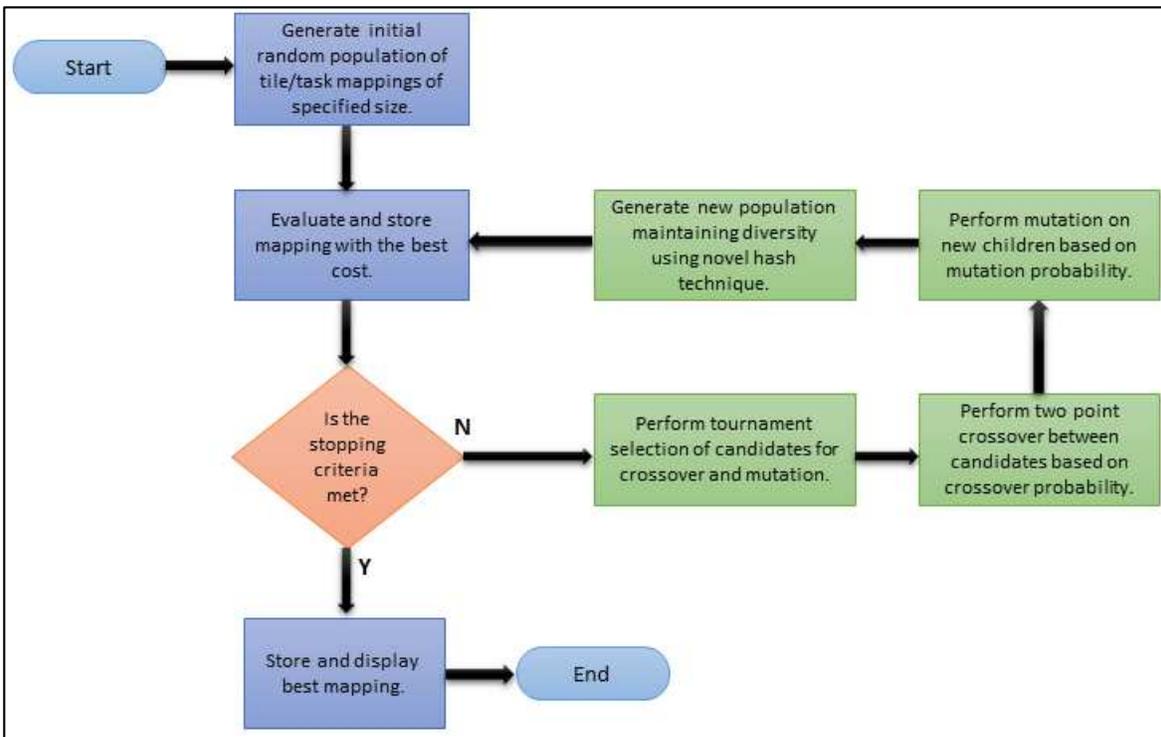


Figure 19: Genetic Algorithm Flowchart

- **Partially Matched Crossover (PMX)** – The crossover mechanism considered for this scheme is partially matched crossover (PMX), as the population is an ordered set and conventional single-point and two-point crossovers would result in illegal mappings, as the same application core could be mapped to different tiles, as shown in Figure 20. Partially

mapped crossover is an ordered crossover technique that preserves the ordered set, avoiding illegal mappings. It has been illustrated in [46] that PMX offers superior solution quality in comparison to other ordered crossover techniques like cyclic crossover and ordered crossover. To perform cross over and mutation, two candidates are randomly selected from the mapping pool and crossover probability (P_{Xover}) and ($P_{Mutation}$) are randomly generated, every time the selection occurs. When P_{Xover} exceeds P_c , PMX crossover is performed between the selected mappings (**step 6**). Again, when $P_{Mutation}$ exceeds P_c for two new child mappings, mutation is performed on the new mapping, by selecting two random indices to swap based on the Linear Congruential Generator algorithm [34] (**step 7**). The partially matched crossover process is illustrated in Figure 21.



Figure 20: Generic Two point crossover with illegal mappings

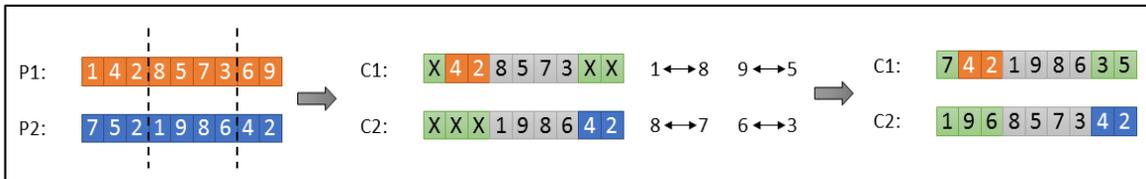


Figure 21: Illustration of partially mapped crossover

- **Hash Generation** – The removal of duplicates in the generated GA population has been shown to improve the solution quality and convergence speed of the algorithm. However,

with an increase in system and population size, one to one comparison becomes very time consuming as every single generated individual needs to be compared with other individuals present in the new population. A new technique was proposed by Ronald [47], with the use of unique hash values assigned to each individual in the population and comparison of the hash values, in order to add a newly generated individual to the new population. The algorithm to create hash values for a given mapping is shown in Figure 22. First, a prime number c is chosen, typically 10x more the population size (**step 1**). We choose a mapping from the population and set both the start index i and the initial hash value r as 0 (**step 2**).

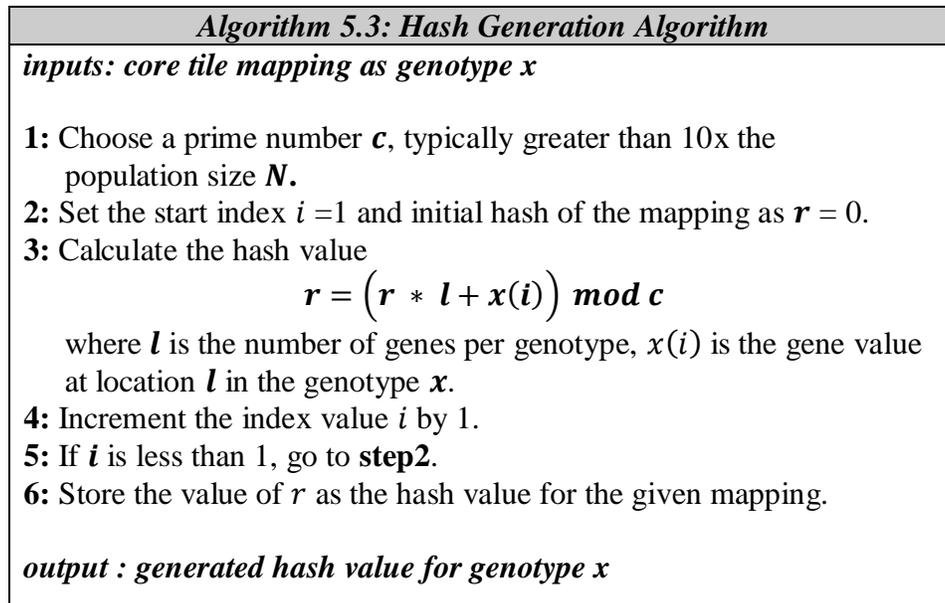


Figure 22: Illustration of Hash Gene generation [38]

We then update the hash value iteratively for the entire mapping, using the given formula and store it with the mapping (**steps 3 to 6**). We then compare the hash value of the newly generated child mappings with the hash values of the existing mappings in the new population P' . The

procedure is repeated till the new population P' reaches the population limit of N (**Figure 18, step 8**).

5.1.4 Routing Path Allocation and Selection

In this section, we describe the routing implications for irregular network topologies and the choice of a suitable routing algorithm. The routing logic for the algorithm could be deterministic or adaptive in nature, with deadlock freedom for packets ensured using unidirectional or bidirectional turn restrictions in the network to break cycles. However, due to the irregular nature of the network under consideration due to failed links, conventional dimension order routing techniques designed for regular networks are not applicable for packet routing in the irregular scheme. There are several proposed topology agnostic deterministic and partially adaptive routing algorithms like UP/DOWN [30], FX [36] and Segment-based routing [31], which have been used for routing packets in irregular networks. Performance comparisons between the various topology agnostic routing schemes have indicated that a properly chosen segment-based routing configuration performs significantly better than the other schemes for various optimization objectives [33].

In the NoC fabric, we consider each router to contain a routing table populated with routing entries using Segment-based routing (SR) logic. Segment-based routing (SR) is a topology agnostic deterministic routing technique that is based on the placement of bidirectional routing restrictions based on a segmentation technique performed on the topology [31], [32]. The process of computing segments for an irregular topology starts with the choice of an initial router and the neighboring routers and links are visited in a recursive manner, till the initial router is encountered again and the visited routers and links are designated as visited and a segment is designated. New

initial routers are selected from the visited set and subsequent segments are discovered recursively till all routers and links are marked as visited.

This preserves connectivity in the network, as well as identifies segments to place bidirectional restrictions independent of other segments. The process of searching for segments can be repeated for various root nodes resulting in different sets of possible routing configurations. However, the number of possible routing configurations rises significantly with increasing system size, causing exhaustive exploration of routing configurations to be a potential challenge with future systems. A segment routing configuration on a 5 x 5 irregular mesh topology with 10% link faults and bidirectional turn restrictions is illustrated in Figure 23.

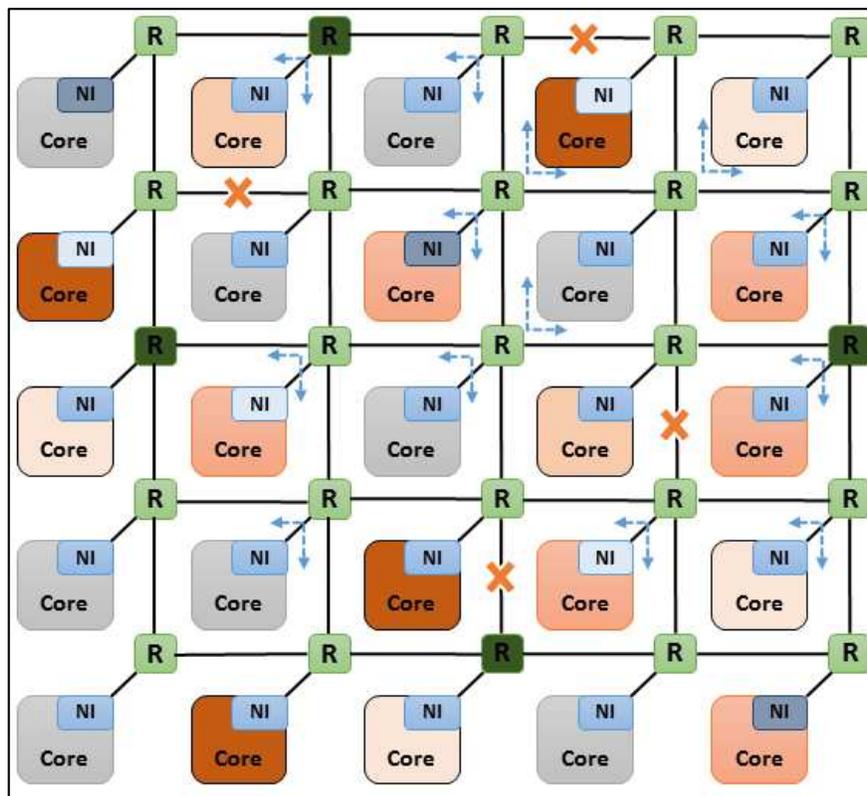


Figure 23. Example of Segment Routing (SR) based segments and routing restrictions (blue arrows) for a 5x5 irregular mesh topology with 10% faults (failed links indicated by red crosses on links), process variations and dark silicon based active (red) and inactive (gray) cores.

For a given application core to tile mapping selected from the GA, the routing paths need to be selected and analyzed for communication between the cores, before and after each iteration of the GA. Since there could be multiple valid paths between two cores, minimization of communication power is achieved by selection of the routing path with minimum hops that meets the link bandwidth constraints without any violations. This is modeled as a minimum cost maximum flow (MCF) problem and solved using a linear programming (LP) solver implementation, *lp_solve* [54], for all communicating tile pairs. Given the network $G(T, L)$, edge $l \in L$ having capacity $c(t_i, t_j)$. For each tile pair $t_i, t_j \in T$, there exist K candidate paths with attributes $K_i = \{t_i, t_j, r_i\}$, where h is the required link bandwidth. The communication flow between the tiles is represented by $f_m(l_i)$.

In Equations (4.1 – 4.3), we define the various constraints for the MCF problem formulation. Equation 4.1 formulates the capacity constraints for all the NoC links by setting the sum of all communication flows $f_m(l_i)$ through a link i , to be less than or equal to the bandwidth constraint for that link between tile i and tile j .

MCF Formulation:

Capacity Constraints:

$$\sum_{m=1}^K f_m(l_i) \leq c(t_i, t_j) \dots (4.1)$$

Flow conservation:

$$\sum_{n \in T} f_m(l_i) = 0 \quad \dots (4.2)$$

$$\text{when } u \neq t_i, t_j \forall u, v, f_m(l_i) = -f_m(l_i)$$

Demand satisfaction:

$$\sum_{w \in T} f_m(t_i, w) = \sum_{w \in T} f_m(w, t_j) = h_i \quad \dots (4.3)$$

*Minimize hop count * communication volume:*

$$\sum_{(l_i) \in E} (h(l_i) * \sum_{m=1}^K f_m(l_i)) \quad \dots (4.4)$$

Equation 4.2 formulates the communication flow integrity check by setting the total sum of flows between the tiles as zero. Equation 4.3 formulates the required link bandwidth check to check if the communication flows meet the required link bandwidth h_i , which is also less than $c(t_i, t_j)$. Equation 4.4 establishes the optimization objective as a product of the total link bandwidths of the total number links involved in the communication flows $h(l_i)$ and the m communication flows $f_m(l_i)$ through the links.

5.1.5 Best Case System Energy and Overall PPY Evaluation with Constraints

The overall power consumption is evaluated as the sum of the computation and communication power of the selected and mapped tiles from the entire NoC, per die. The computation power is evaluated as the sum of the dynamic and leakage powers of the mapped compute cores. The communication power is estimated from the bit model proposed in [27], for the data dependent NoC components.

The total power is evaluated for all the 100 dies and compared against the power budget to evaluate the percentage of dies that meet the constraint and calculate the PPY based on Equation 2.10. The best case system energy consumption E_{system} for the mapping with the lowest total

power consumption P_{Total} among all the 100 dies and corresponding simulation execution time T_{Sim} , is defined as:

$$E_{system} = (P_{Total} * T_{Sim}) \quad \dots (4.5)$$

6 Results

In this chapter, we present the setup and results of our simulation framework. In section 6.1, we briefly discuss the various components of the simulation framework and the setup of the frameworks under comparison. In section 6.2, we present process variation maps for the various NoC components under consideration. In section 6.3, we present the overall power statistics and system energy consumption for all the frameworks under consideration, across all benchmarks. In section 6.4, we perform sensitivity analyses for system energy and PPY statistics across varying system sizes and faults. Finally in section 6.5, we conclude by presenting execution time overhead analysis results for all the frameworks under consideration.

6.1 Simulation Setup

6.1.1 Benchmarks, NoC and Evaluation Framework

Our simulations were performed using eight parallel benchmarks from the SPLASH-2 (*cholesky*, *fft*, *lu* and *ocean*) and PARSEC (*blackscholes*, *dedup*, *canneal* and *fluidanimate*) benchmark suites [55] - [57]. The compute cores attached to the NoC are modeled after the ALPHA [22] processor, using a 22 nm low leakage asymmetric gate function (ASG) FinFET model, with a core frequency of 1.2 GHz. The single core design has 4 ALUs and 1 FPU, and a superscalar and out-of-order execution based design. The L1 instruction and data cache capacities are 64 bits with the L2 cache capacity set to 1.75 MB. The cache is configured with a block size of 16 bytes, associativity as 8, access model as UCA and MOESI coherence protocol. The router buffers are assumed to have a capacity of up to 48 flits, with each flit size being 64 bits. The link power statistics for the NoC are generated using DSENT [26] for the 22 nm technology node for a frequency of 1.2 GHz, with data width of 64 bits and link lengths of 1 mm for the 12×12 NoC

and 0.5 mm for the 14×14 NoC. We use a 144 tile NoC topology with 10% failed links as the base case, in a 12×12 square mesh configuration for evaluation, of which up to 100 tiles can be selected at a given time due to dark silicon constraints.

For evaluation, the power budgets for the system of 100 tiles (selected from the 144 total tiles due to dark silicon scenario), are set to 170W (stringent) and 180W (nominal) for less communication intensive benchmarks like *cholesky*, *ocean*, *blackscholes* and *fluidanimate*, designated as the *comm_int_low* benchmark set and 230W (stringent) and 240W (nominal) for more communication intensive benchmarks like *dedup*, *canneal*, *lu* and *fft*, designated as the *comm_int_high* benchmark set. The link bandwidth constraints are empirically set at 47 MBps for *cholesky_100* and *ocean_100*, 12.5 MBps for *blackscholes_100* and *fluidanimate_100*, 160 MBps for *dedup_100*, 134 MBps for *lu_100*, 155 MBps for *canneal_100* and 136 MBps for *fft_100* respectively.

6.1.2 Comparison of HERMES with proposed prior techniques

The HERMES framework is compared with proposed frameworks using Simulated Annealing (SA) [34], Local Search technique (CART) [41] and [42] Particle Swarm Optimization (PSO) based approaches [48]. As the optimization techniques in the proposed frameworks do not employ FinFETs, dark silicon considerations and process variation aware mapping, we have modeled them to the best of our knowledge in terms of our framework for a fair comparison between the different techniques. Additionally, we have modified the implementations in [34] and [48] to model an irregular topology configuration with varying fault rate with the respective routing tables created using the Segment Routing (SR) methodology, as the original implementations are based on regular topologies with conventional dimension order routing. Finally, we also apply the MCF

based path selection model to [34], [41] and [48] to ensure a fair comparison between the techniques for path selection on irregular mesh topology configurations.

Due to the computationally intensive nature of the cost function, we run the Genetic Algorithm (GA) in our simulation framework for a population range of 50 to 100, with a limit on the number of generations till 300. The crossover probability is set to 0.6 and the mutation probability is set to 0.02. We set the prime number as 50021 for the hash generation technique in [47] to maintain population diversity in the GA.

As for the compared techniques, we use a start temperature = 100 for the SA, with a total limit on the number of iterations as 200. For the PSO, we use a swarm size of 50 to 100, with a limit on the number of epochs till 300, similar to the GA. The local probability variable is set at 0.3 and the global probability variable is set at 0.1. For the local search technique (CART), we run the algorithm for an iteration count limit of 100 for 3 random seeds performing random perturbations on a mapping to explore a new solution space.

We evaluate and compare the energy consumption, simulation execution time and the power performance yield (PPY) for all 4 techniques and 8 benchmarks. Additionally, for a sensitivity analysis, we conduct the fault rate variation scenario on a 100 core system with fault rates varying from 10% to 30% of failed links in the network topology. We also conduct the system size variation scenario with varying compute core selection sizes of 64, 100, 144 and 256 compute cores, with a fixed fault rate of 10% failed links. We select 64, 100 and 144 cores to perform the analysis from the base case system of 144 cores. We also perform additional analysis of a system of 256 cores with 10% faults to observe the scalability in performance of HERMES compared to the other proposed frameworks.

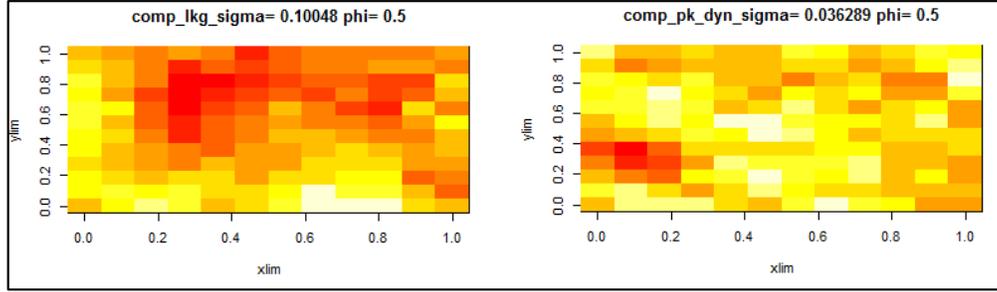
6.2 Simulation Results

6.2.1 Process Variation based Power Maps

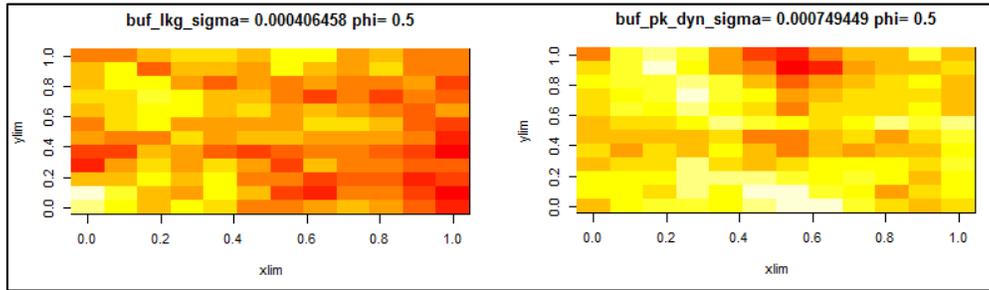
The process variation maps are generated from a modified VARIUS implementation using the *geoR* and *RandomFields* packages, using power values for the compute core, NoC switch and crossbar from McPAT-PVT for a 22nm process node for a 12x12 mesh NoC topology. The leakage and dynamic process variation maps are generated with a fixed spherical correlation coefficient (ϕ) = 0.5 for WID variations and an offset of 6% for D2D variations. The mean leakage and mean peak dynamic power values and their respective standard deviations for the significant NoC components are shown in Table 1. In Fig 24 (a), we plot the process variation maps for peak dynamic power (mean = 1.11777 W, std. deviation = 0.036289 W) and leakage power (mean = 0.338900 W, std. deviation = 0.10048 W) for the compute core on the tile. In Fig 24 (b), we plot the process variation maps for peak dynamic power (mean = 0.00246344 W, std. deviation = 0.000749449 W) and leakage power (mean = 0.0172786 W, std. deviation = 0.000406458 W) for the NoC buffer in the router. In Fig 24 (c), we plot the process variation maps for peak dynamic power (mean = 0.000202701 W, std. deviation = 3.50856e-05 W) and leakage power (mean = 0.00177863 W, std. deviation = 5.99292e-05 W) for the NoC crossbar in the router. We also observe that the variation maps for peak dynamic power and leakage power exhibit differences due to different standard deviation values, indicating no correlation between modeling of peak dynamic and leakage power for the models in [24] and [25].

Table 1: Mean total power and standard deviation values for NoC components

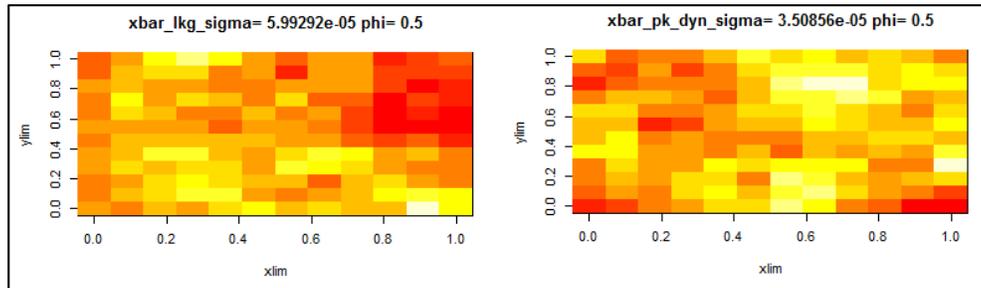
	Leakage Power (W)		Peak Dynamic Power (W)	
	Mean	Std. Deviation	Mean	Std. Deviation
Compute Core	0.338908	0.10048	1.11777	0.036289
NoC Buffer	0.0172786	0.000406458	0.00246344	0.000749449
NoC Crossbar	0.00177863	5.99292e-05	0.000202701	3.50856e-05



(a) Compute Core



(b) NoC Crossbar



(c) NoC Buffer

Figure 24: The generated power maps for the respective leakage (left) and peak dynamic components (right) of the various compute core and NoC components with respective sigma (standard deviation) values.

6.2.2 Total Power and System Energy Consumption

In Figure 25, we present the results of the total power, with the range of power values represented by error bars, across all the 100 dies for the *comm_int_low* and *comm_int_high* benchmark sets. In Figure 26, we present the results for the best case system energy consumption

across all the 100 dies (from the die with the lowest total power), for the *comm_int_low* and *comm_int_high* benchmark sets. For execution time figures to evaluate the system energy consumption, from Equation 4.5, we considered the best case die for the execution time values across the various algorithms, we tabulate the results in Table 2.

Table 2: Execution times for the PARSEC and SPLASH-2 suites [100 of 144 cores]

Run Time (seconds)	SA	HERMES	PSO	CART
<i>blackscholes_100</i> [PARSEC]	8665	6700	7123	8523
<i>dedup_100</i> [PARSEC]	8723	6965	7333	8632
<i>fluidanimate_100</i> [PARSEC]	8821	6854	7435	8535
<i>canneal_100</i> [PARSEC]	8735	6745	7395	8594
<i>cholesky_100</i> [SPLASH-2]	8698	6731	7112	8412
<i>fft_100</i> [SPLASH-2]	8780	6774	6992	8292
<i>ocean_100</i> [SPLASH-2]	8432	6814	7213	8213
<i>lu_100</i> [SPLASH-2]	8541	6790	7011	8311

The SA, while providing a lowest total power value close to that provided by the GA as observed in Fig 25, is observed to perform poorly in searching a large solution space for a suitable solution at larger system sizes as observed in Table 2, due to swapping random core tile pairs and repeated evaluation of an expensive cost function to evaluate the communication power figure. The CART technique also provides poor power and run times statistics, similar to the SA, due to the randomized nature of the local search technique resulting in a lot of variance in the final power statistics. The performance gain of our framework over the PSO based mapping technique is marginally lower than the GA because we observe that, while the swap based particle update mechanism is very effective as is PMX crossover technique in searching a large solution space, it

still loses out on the solution quality and convergence speed offered by the fast hash function and generates slightly higher power statistics and execution times compared to the GA.

Again, using the power statistics in Fig 25 and tabulated execution time figures in Table 2 in Equation 4.5, we observe that our HERMES framework offers excellent improvement in best case system energy consumption over the SA and CART techniques for both the *comm_int_low* and *comm_int_high* benchmark sets, due to the much faster execution times of the GA used in the framework as observed in Table 2, owing to the effective PMX crossover technique [46] and the fast hash algorithm [47] that offers faster convergence with excellent solution quality.

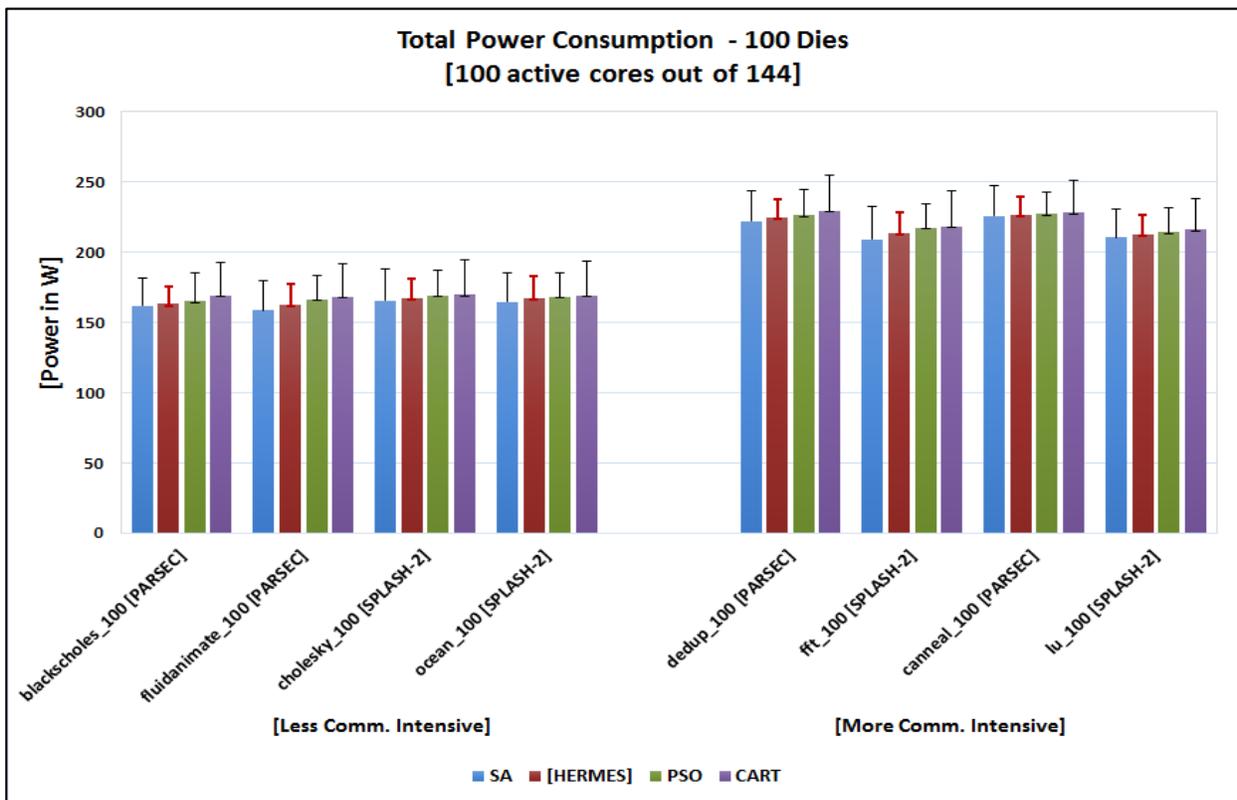


Figure 25: Total power consumption across all 100 dies for all eight benchmarks for 100 active cores from a base system size of 144 cores

In Fig 26, for the *comm_int_low* benchmark set *blackscholes_100*, *fluidanimate_100*, *cholesky_100* and *ocean_100*, we observe improvements across all benchmarks, up to 1.28x for the SA, 1.11x for the PSO and 1.32x for CART. Again, for the *comm_int_high* benchmark set *dedup_100*, *canneal_100*, *lu_100* and *fft_100*, we observe improvements across all benchmarks, up to 1.29x for the SA, 1.10x for the PSO and 1.28x for CART. Overall, our HERMES framework provides maximal improvements of at least 1.29x, 1.11x and 1.32x in energy consumption for the *comm_int_low* and *comm_int_high* based benchmarks, compared to the SA [34], PSO [48] and CART [41] and [42] techniques, respectively.

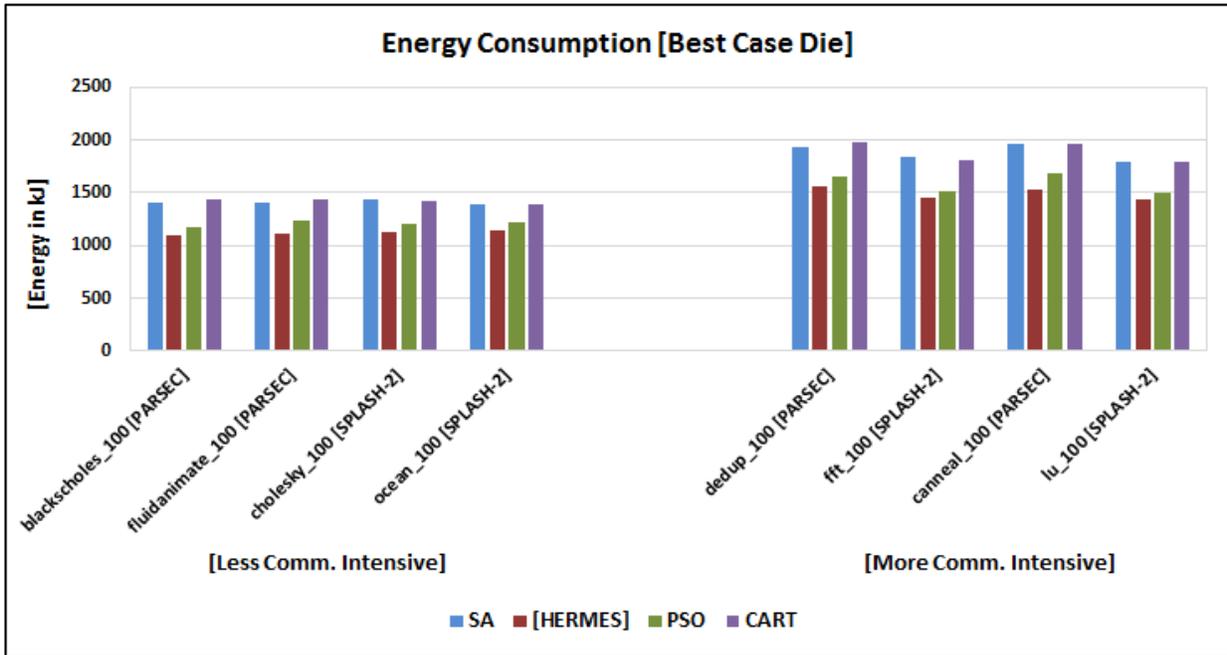


Figure 26: Best case system energy consumption for all eight benchmarks for 100 active cores from a base system size of 144 cores

6.2.3 Power Performance Yield (PPY)

In Figure 27, we present the results for the Power Performance Yield (PPY) statistics among all the dies for the *comm_int_low* benchmark set with a stringent power constraint of 170 W and

a nominal power constraint of 180 W. Quantitatively expressing the PPY statistics for the *comm_int_low* benchmark set with a power constraint of 170 W, we observe improvements up to 30% over the SA, 53% over CART and up to 36% over the PSO implementation respectively.

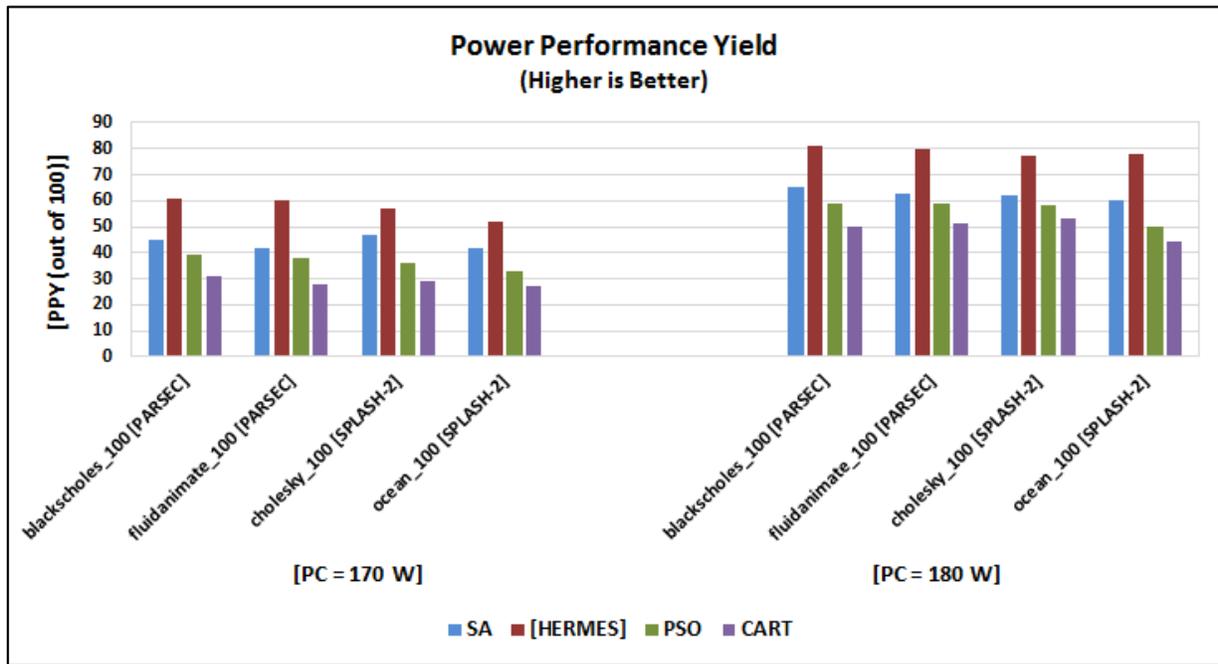


Figure 27 : Power Performance Yield (PPY) for the *comm_int_low* benchmark set with a stringent power constraint of 170 W and a nominal power constraint of 180 W

Again, for the *comm_int_low* benchmark set with a power constraint of 180 W, we observe an improvements of up to 23%, 35.9% and 43.5% for a nominal power constraint of 180 W, for the SA, PSO, and CART based implementations, respectively. We observed maximum overall improvements of up to 30%, 36%, and 53.33% in energy consumption for the *comm_int_low* benchmark set with a stringent power constraint of 170 W; of up to 23%, 35% and 43% improvement in energy consumption for the *comm_int_low* benchmark set with a nominal power constraint of 180 W compared to the SA [34], PSO [48] and CART [41] and [42] techniques,

respectively. We observed a higher PPY for all solutions in this case, because a relaxed power constraint allows the inclusion of more dies that meet the relaxed power constraint. In Figure 28, we present the results for the Power Performance Yield (PPY) statistics among all the dies for the `comm_int_high` benchmark set with a stringent power constraint of 230 W and a nominal power constraint of 240 W. Here we also observe that HERMES performs consistently with improvements up to 16.92%, 36.92%, and 50.77% across the board with a 230 W power constraint and up to 19.23%, 39.74%, and 58.44% for a nominal power constraint of 240 W, over the SA, PSO, and CART implementations, respectively. Again, we observed a higher PPY for all algorithms with a relaxed power constraint, as it allows for the inclusion of more dies.

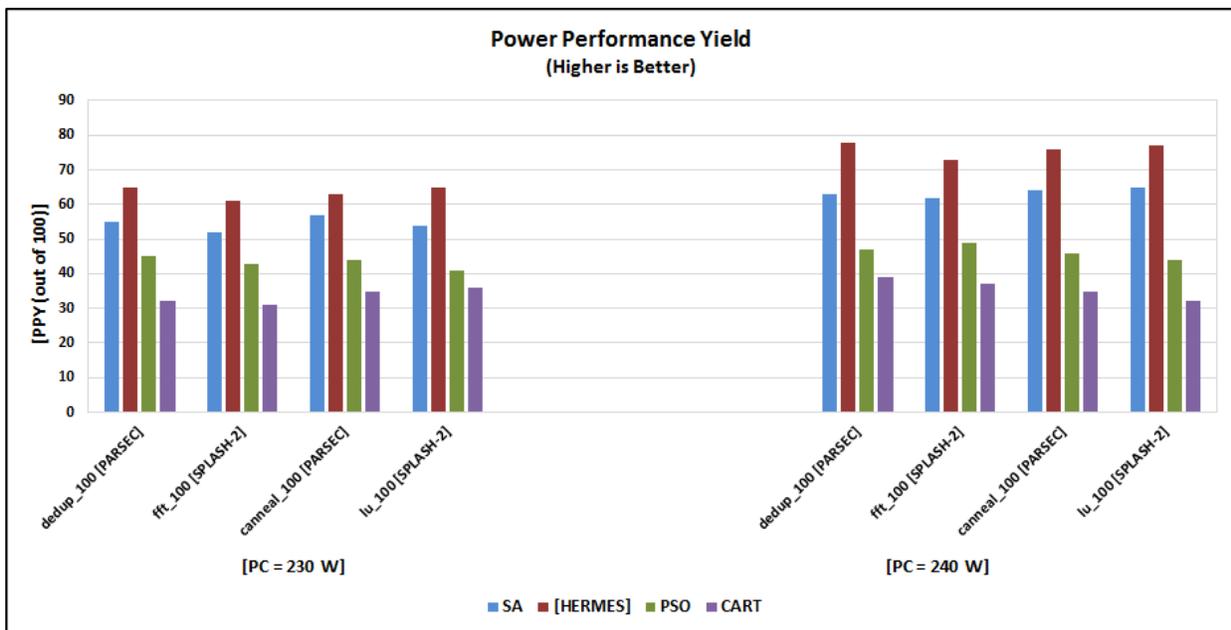


Figure 28: Power Performance Yield (PPY) for the `comm_int_high` benchmark set with a stringent power constraint of 230 W and a nominal power constraint of 240 W

In summary, for PPY evaluation scenarios for the benchmark sets, we clearly observe that HERMES offers excellent gains over the PSO, SA and CART implementations owing to the

superior solution quality exhibited by use of the hash algorithm, resulting in very competitive power statistics in both the stringent and the relaxed scenarios. This results in more dies that meet the power constraints and exhibiting far superior PPY statistics compared to the other algorithms. The PSO while offering competitive execution times loses out on solution quality due to the lack of a diversity maintenance function, while the SA does not consistently provide the best power results for all the dies. CART suffers from the randomized nature of the algorithm and offers the least in terms of PPY statistics.

6.2.4 Sensitivity Analysis - Execution Time

In Figure 29, we plot the variation of execution time versus the change in system size for all four techniques, as described in Section 6.1.2, for the *blackscholes* benchmark from the *comm_int_low* set and *dedup* benchmark from the *comm_int_high* set, for varying system sizes ranging from 64, 100 and 144 compute cores selected from a 144 core system as well as a 256 core system. We also observe here that HERMES scales very well with system size, with the combination of the PMX crossover and hash algorithm providing extremely fast convergence, compared to other techniques. Quantitatively expressing the trends for the *blackscholes* benchmark for a 64 core system HERMES exhibits a performance advantage of 1.12x over the SA, 1.17x over CART, and 1.10x over the PSO, respectively. Comparing this to the base case of a 256 cores system with all 256 active cores, we observe that HERMES provides excellent execution time gains of 3.84x over the SA and 3.92x over the PSO, and maintains a steady advantage of 1.10x over the PSO, respectively. This highlights the ability of HERMES to be able to efficiently search a much larger solution space compared to the other techniques. Similarly for the *dedup* benchmark, we also observe a trend with HERMES scaling very well compared to the other techniques. We observe that HERMES offers a performance improvement of 1.15x over the SA, 1.14x over CART,

and $1.07x$ over the PSO respectively, for a 64 core selection scenario from a 144 core system. These statistics improve up to $3.80x$ over the SA, $3.92x$ over CART and $1.30x$ over the PSO technique respectively, for a 256 core system with all active cores. Again in Figure 30, we plot the variation of execution time versus the change in fault rates for all four techniques with a low fault rate of 10% failed links, up to a high fault rate of up to 30% failed links in the NoC topology for a 144 core system with 100 cores selected, for the *blackscholes* benchmark from the *comm_int_low* benchmark set and for the *dedup* benchmark from the *comm_int_high* set. We observe a similar trend with HERMES being the most efficient with an effective combination of an efficient crossover (PMX) technique and fast hash function resulting in substantially lower execution times than the techniques proposed in [34], [41], [42] and [48].

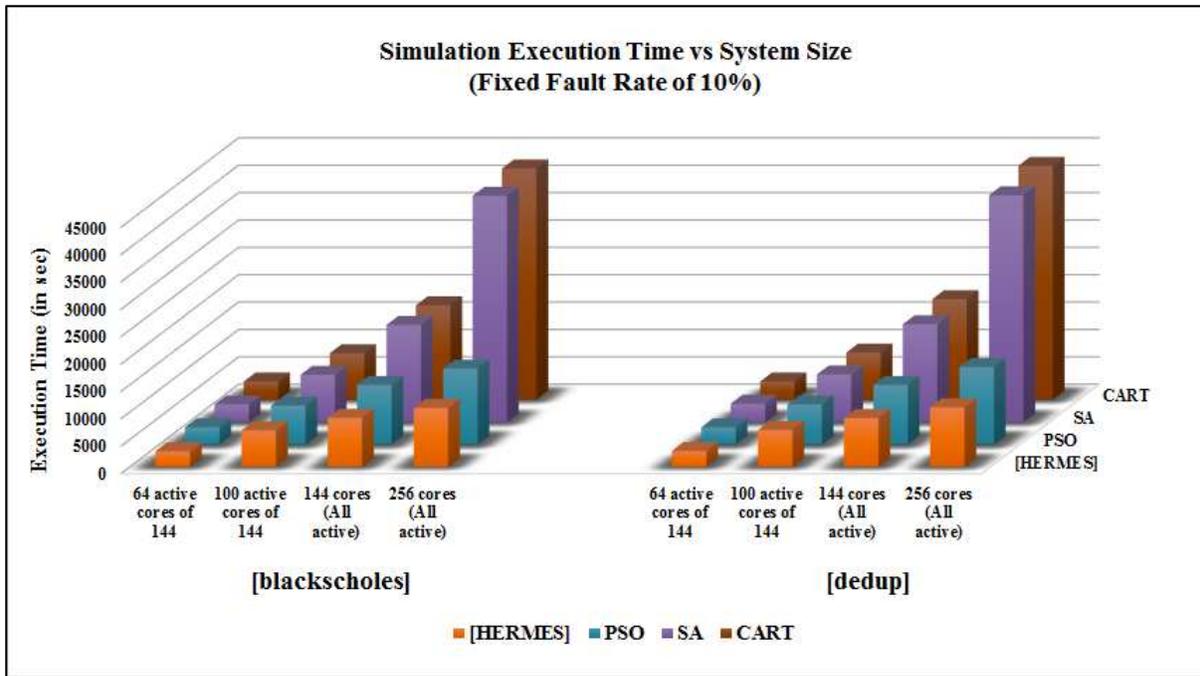


Figure 29: Sensitivity analysis of execution time with varying system sizes having 64, 100, 144 and 256 cores respectively for *blackscholes* and *dedup* with 10% faults

Quantitatively expressing the trends for the *blackscholes* benchmark, for a low fault rate of 10%, HERMES exhibits a performance advantage of 1.29x over the SA, 1.27x over CART and 1.06x over the PSO, respectively. Comparing this to a high fault rate scenario of 30%, we observe that HERMES maintains the execution time gains of 1.28x over the SA and 1.32x over the PSO and 1.14 x over the PSO, respectively.

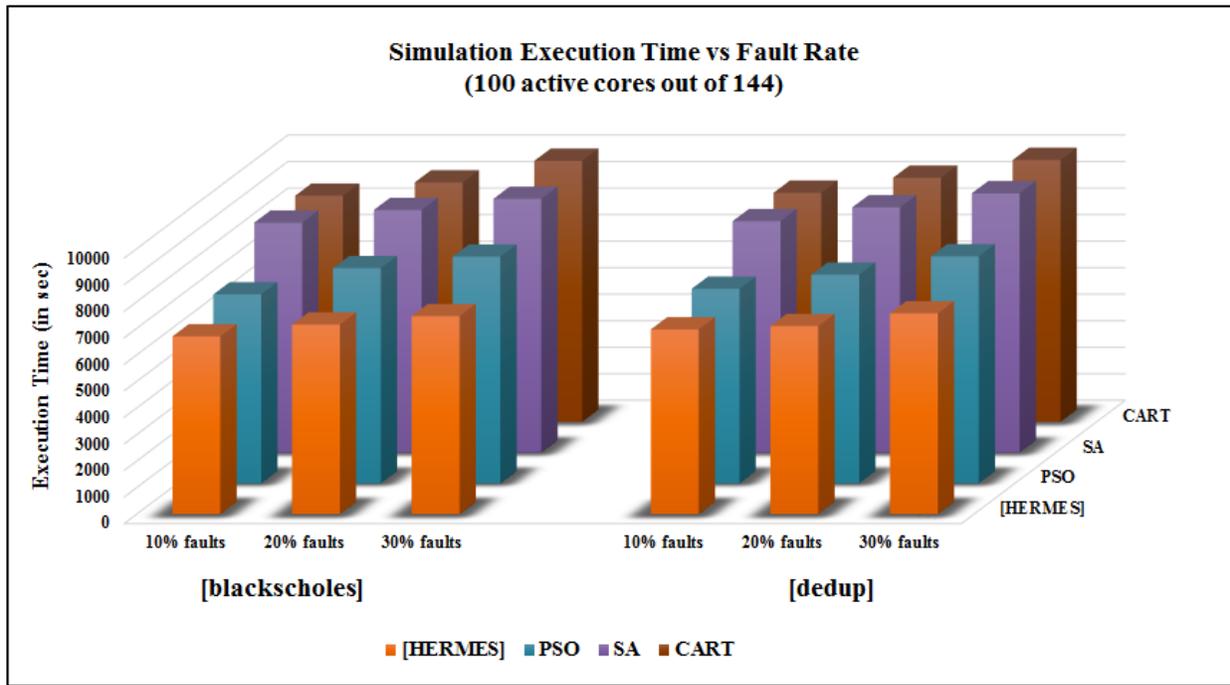


Figure 30: Sensitivity analysis of execution time with varying fault rates of 10%, 20% and 30% respectively for *blackscholes* and *dedup* with 100 active cores from 144 cores

Expressing the trends for the *dedup* benchmark, for a low fault rate of 10%, HERMES exhibits a performance advantage of 1.25x over the SA, 1.24x over CART and 1.05x over the PSO, respectively. For high fault rate scenario of 30%, we observe that HERMES maintains the execution time gains of 1.29x over the SA and 1.30x over the PSO and 1.13x over the PSO, respectively.

6.2.5 Sensitivity Analysis - System energy

In order to perform sensitivity analyses for system energy consumption, we consider a set of single benchmarks, each from the *comm_int_low* benchmark set (*blackscholes*) and *comm_int_high* benchmark set (*dedup*) to analyze the impact of varying fault rates and system sizes on the simulation execution time of the various algorithms, and we conduct performance comparison in terms of energy and execution time. In Figure 31, we plot the variation of energy consumption versus the change in system size for all four techniques with fixed a 10% fault rate for the *blackscholes* benchmark from the *comm_int_low* set and *dedup* benchmark from the *comm_int_high* set, for varying system sizes, as outlined in Section 6.1.2.

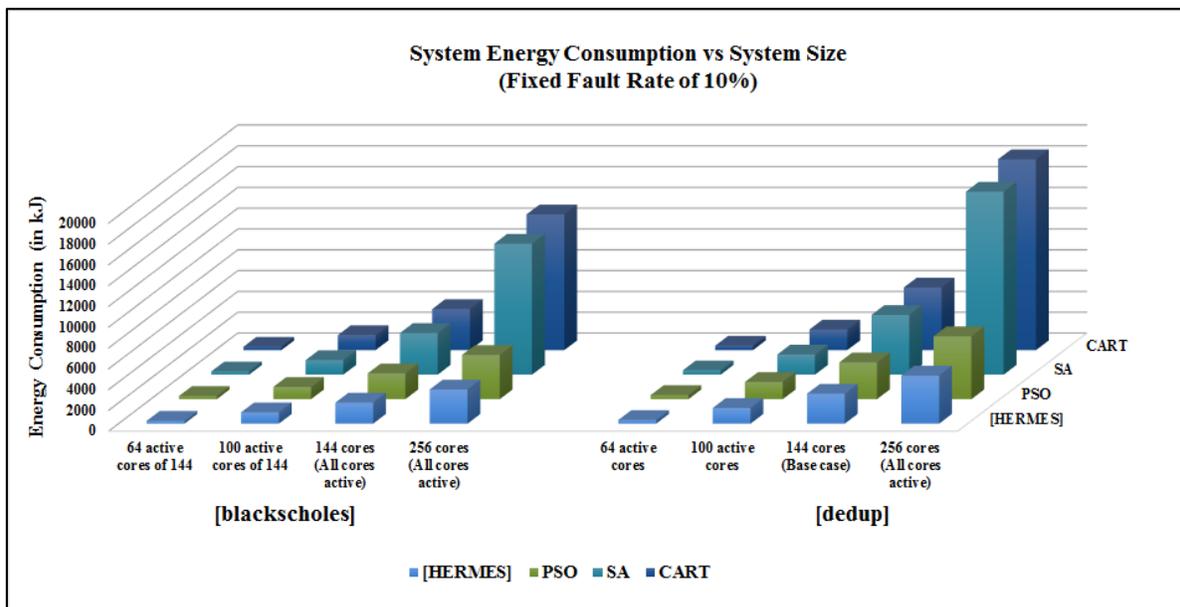


Figure 31: Sensitivity analysis of energy consumption with varying system sizes having 64, 100,144 and 256 cores respectively for *blackscholes* and *dedup* with 10% faults

Here we observe a similar scaling trend with execution time, as the system energy consumption computation is directly proportional to the execution time. Quantitatively expressing the system

energy consumption statistics, for the *blackscholes* benchmark at 64 cores, we observe a gain of 1.12x over the SA, 1.19x over CART, and 1.11x over the PSO, respectively. Increasing the system size to the base case of all active cores for a 256 core system, results in excellent gains with 3.83x over the SA, 3.97x over CART and a steady gain of 1.29x over the PSO. Similarly, for the *dedup* benchmark at 64 cores, we observe a gain of 1.15x over the SA, 1.18x over CART and 1.09x over the PSO respectively. This significantly improves with system size to 3.80x over the SA, 3.96x over the PSO and 1.31x over the PSO, at the case of all active cores, for a 256 core system with all active cores.

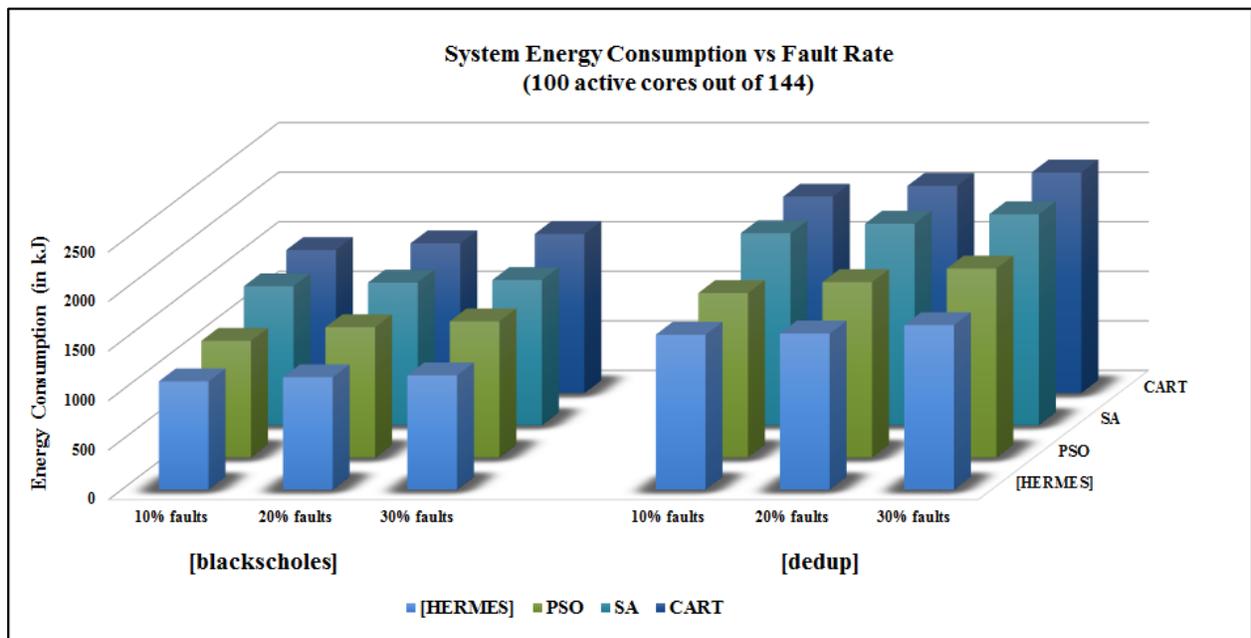


Figure 32: Sensitivity analysis of energy consumption with varying fault rates of 10%, 20% and 30% respectively for *blackscholes* and *dedup* with 100 active cores from 144 cores

In Figure 32, we plot the variation of energy consumption versus the change in fault rates for all four techniques with a low fault rate of 10% failed links, up to a high fault rate of up to 30% failed links in the NoC topology, for the *blackscholes* benchmark from the *comm_int_low*

benchmark set and for the *dedup* benchmark from the *comm_int_high* set. In these varying fault scenarios, the MCF path selection mechanism in our framework exhibits a steady scaling trend in terms of execution time with an increasing fault rate. Since we have included the MCF-based mechanism in all of the algorithms for a fair comparison, we observe steady scaling in the execution time for all the algorithms with increasing fault rate, highlighting the effectiveness of the path selection mechanism in selecting appropriate bandwidth constrained paths between all communicating core tile pairs with a marginal impact on execution time. Expressing the results quantitatively for the *blackscholes* benchmark, we observe a steady trend for all the techniques with increasing fault rates from 10% to 30%, with HERMES maintaining a steady performance gain of up to 1.28x over the SA, 1.39x over CART, and a marginal gain of 1.19x over the PSO technique. Similarly for the *dedup* benchmark, we also observe a steady trend for all the techniques with increasing fault rates from 10% to 30%, with HERMES maintaining a steady performance gain of up to 1.29x over the SA, 1.34x over CART, and a marginal gain of 1.15x over the PSO technique.

6.2.6 Sensitivity Analysis – PPY

In Figure 33, we plot the variation of PPY time versus the change in system size for all four techniques, as described in Section 6.1.2, for the *blackscholes* benchmark from the *comm_int_low* set and *dedup* benchmark from the *comm_int_high* set, for varying system sizes ranging from 64, 100 and 144 compute cores selected from a 144 core system as well as a 256 core system. Here, we again observe that HERMES scales very well with system size and offers very competitive power figures across all the 100 dies, with the combination of the PMX crossover and hash algorithm providing extremely fast convergence and improved power values, compared to the other techniques in [34], [41] and [48]. Quantitatively expressing the trends for the

blackscholes benchmark for a 64 core system, HERMES exhibits a performance advantage of 26.67% over the SA, 46.67% over CART, and 35% over the PSO, respectively. Comparing this to the case of a 256 cores system with all 256 active cores, we observe that HERMES maintains steady gains of 29.03% over the SA and 51.61% over the PSO, and a steady advantage of 38.71% over the PSO, respectively. This highlights the ability of HERMES to be able to provide quality power values and resulting in more dies meeting the power constraints, compared to the other techniques. Similarly for the *dedup* benchmark, we also observe a trend with HERMES scaling very well in comparison to the other techniques. We observe that HERMES offers a performance improvement of 20% over the SA, 49.33% over CART and 40% over the PSO respectively, for a 64 core selection scenario from a 144 core system. These statistics improve up to 20.27% over the SA, 48.65% times over CART and 40.54% over the PSO technique respectively, for a 256 core system with all active cores.

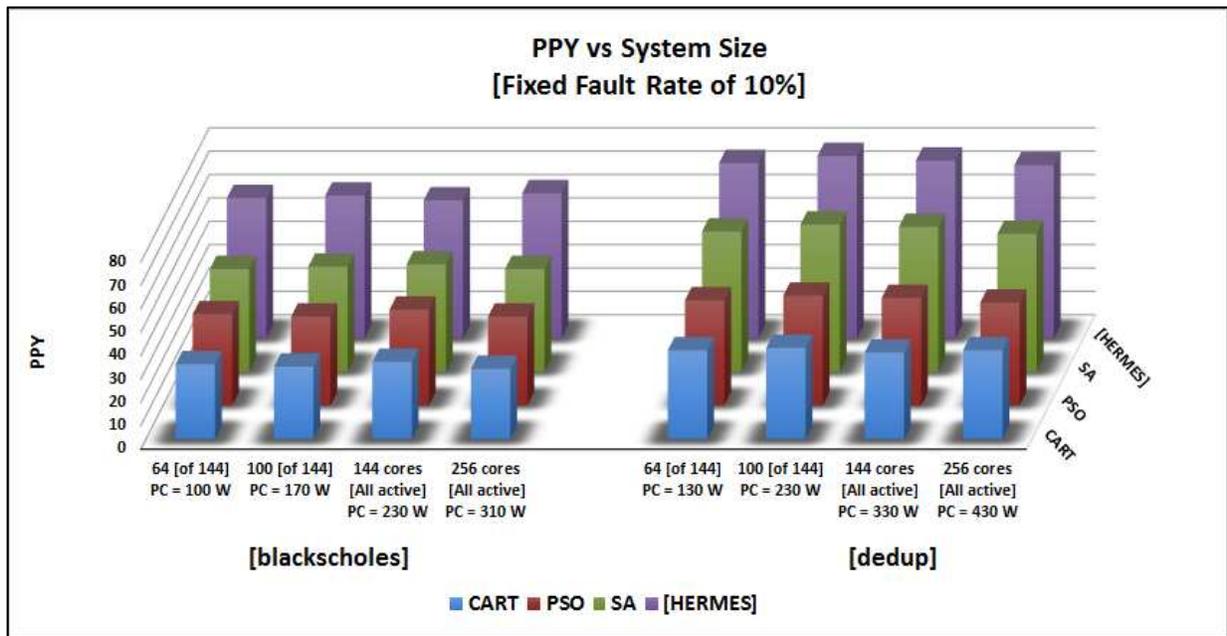


Figure 33: Sensitivity analysis of PPY with varying system sizes across 64, 100, 144 and 256 cores respectively for *blackscholes* and *dedup* with 10% faults

Finally in Figure 34, we plot the variation of PPY versus the change in fault rates for all four techniques with a low fault rate of 10% failed links, up to a high fault rate of up to 30% failed links in the NoC topology for a 144 core system with 100 cores selected, for the blackscholes benchmark from the comm_int_low benchmark set and for the dedup benchmark from the comm_int_high set. We also observe a similar trend with HERMES being the most efficient with an effective combination of an efficient crossover (PMX) technique and fast hash function resulting in more dies meeting the power constraints, than the techniques proposed in [34], [41] and [48]. Quantitatively expressing the trends for the blackscholes benchmark, for a range of faults rate of 10% to 30%, HERMES exhibits a performance advantage of up to 35.42% over the SA, 54.17% over CART, and 36.07 % over the PSO, respectively. Expressing the trends for the dedup benchmark, for a range of faults rate of 10% to 30%, HERMES exhibits a performance advantage of 27.69% over the SA, 55.28% over CART, and 39.74% over the PSO, respectively.

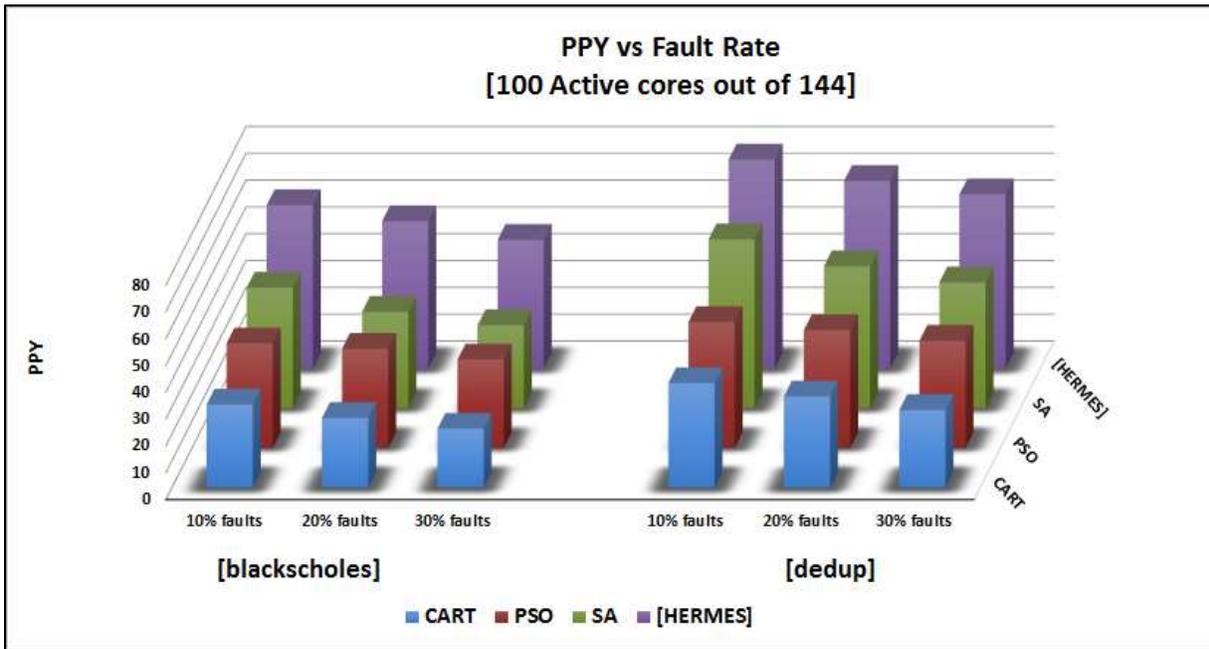


Figure 34: Sensitivity analysis of PPY with varying fault rates of 10%, 20% and 30% respectively for *blackscholes* and *dedup* with 100 active cores from 144 cores

In summary, we observe that our HERMES framework gives us very competitive energy, simulation time and PPY statistics compared to the SA, GA and PSO techniques for parallel benchmarks of varying communication loads from the SPLASH-2 and PARSEC benchmark suite. It also provides excellent scalability for execution time, system energy and PPY with system size and varying fault rates for both less and more communication intensive benchmarks. It provides us the perfect balance of solution quality and speed compared to other techniques while including process variations, dark silicon considerations, and routing path selection mechanisms in a holistic framework.

7 Conclusions and Future Work

Due to the increasing impact of process variations in deep submicron technology nodes, especially in FinFET based designs, it has become increasingly important to incorporate process variation based models in the application mapping and routing synthesis framework. Conventional design flows where power evaluation is done without process variation awareness, design time fault consideration and dark silicon based power budgets tend to generate suboptimal solutions with budget overruns and higher power dissipation. For the first time, we have incorporated process variation aware FinFET based design methodology into a custom mapping and routing framework for irregular mesh based NoCs to generate an optimal power figure with a given system power budget. Our framework also incorporates a custom GA-based application mapping mechanism with a topology agnostic routing scheme to generate a holistic solution that performs up to 1.32x better in energy, 1.29x better in simulation execution time, and 50% better in PPY statistics respectively, than other proposed schemes for a base system size of 100 active cores out of 144 total cores. Our scheme also maintains steady scaling for energy, execution time and PPY values with up to up to 3.97x better in energy, 3.92x better in simulation execution time, and 55.38 % better in PPY statistics, respectively, for varying system sizes of 64 to 256 cores and fault rates of 10% to 30%.

There are several key areas that merit attention as possible extensions to our work, which can be outlined as follows:

- Near-threshold computing (NTC) has been proposed as the next evolution in energy efficient computing with dark silicon power budgets. NTC based methodologies have threshold voltages closer to the supply voltage compared to conventional super-threshold

computing (STC) and lower frequencies, which enables lower power dissipation but results in more leakage. The inclusion of this consideration with process variations in our design time mapping framework would be a very relevant extension for further research.

- With decreasing transistor dimensions below 22 nm, random variations are predicted to play a greater role in process variation, necessitating more detailed exploration, especially for FinFET based implementations. It has become imperative to include more detailed models for random variations to explore process variations in more accurate detail for deep submicron process geometries.
- We have considered a simple model to show faulty links in a NoC topology with two possible states for links, as either active or disabled. However, incorporation of a process variation model for the active links would add an extra dimension to the exploration methodology.
- We have considered mapping of a single application to a homogeneous NoC fabric in our framework. Extension of our framework to include multi-application mapping to a heterogeneous NoC fabric with multiple types of FinFET based processors would add an interesting perspective to the problem.

The inclusion of these considerations in the system level design synthesis is left as a very promising frontier for future work and would be very interesting extensions to the framework.

References

- [1] W. Dally and B. Towles, "Route packets, not wires: on-chip interconnection networks," in *IEEE Proceedings Design Automation Conference*, pp. 684-689, Jun. 2001.
- [2] S. Pasricha, and N. Dutt. "On-Chip Communication Architectures," Morgan Kaufman, ISBN 978-0-12-373892-9, Apr 2008.
- [3] W. Dally and B. Towles, "Principles and practices of interconnection networks," Elsevier, 2004.
- [4] L. Benini and G. Micheli, "Networks on chips: a new SoC paradigm," *Computer* 35.1, pp. 70-78, 2002.
- [5] S. Pasricha, and N. Dutt, "Trends in emerging on-chip interconnect technologies," *IPSIJ Transactions on System LSI Design Methodology*, 2008.
- [6] V. Rantala, T. Lehtonen and J. Plosila, "Network on Chip Routing Algorithms," *Turku Centre for Computer Science Technical Report*, August 2006
- [7] E. Humenay, D. Tarjan, K. Skadron, "Impact of process variations on multicore performance symmetry," *IEEE Proceedings Design, Automation & Test in Europe*, pp. 1653-1658, Apr. 2007.
- [8] S. Bhardwaj et al, "Modeling of intra-die process variations for accurate analysis and optimization of nano-scale circuits," *Proceedings of the 43rd annual Design Automation Conference*. ACM, 2006.
- [9] J. Aarestad, C. Lamech, J. Plusquellic, D. Acharyya, and Kanak Agarwal, "Characterizing within-die and die-to-die delay variations introduced by process variations and SOI history effect," *Proceedings of the 48th Design Automation Conference*, pp. 534-539. ACM, 2011.

- [10] D. Boning, and S. Nassif, “Models of process variations in device and interconnect,” *Design of high performance microprocessor circuits*, 2001.
- [11] B. Swahn and S. Hassoun, “Gate sizing: FinFET versus. 32nm bulk MOSFETs,” *IEEE Proceedings Design, Automation & Test in Europe*, pp. 528–531, Jul. 2006.
- [12] <https://www.semiwiki.com/forum/content/1908-finfet-process-modeling-extraction-16-nm-below.html> - Referenced on 29th April 2015.
- [13] <http://www.dailytech.com/TSMC+Were+Far+Superior+To+Intel+and+Samsung+as+a+Partner+Fab/article34148.htm> - Referenced on 29th April 2015.
- [14] H. Esmailzadehy, E. Blemz, R. St. Amantx, K. Sankaralingamz, D. Burger, "Dark Silicon and the End of Multicore Scaling", pp. 365 – 376, *International Symposium on Computer Architecture*, Jun. 2011.
- [15] M. B. Taylor, “A Landscape of the New Dark Silicon Design Regime”, *IEEE Micro*, pp. 8–19, Oct. 2013.
- [16] Raghunathan, Bharathwaj, et al, “Cherry-picking: exploiting process variations in dark-silicon homogeneous chip multi-processors,” *Proceedings of the Conference on Design, Automation and Test in Europe*, EDA Consortium, 2013.
- [17] H. Esmailzadeh et al, “Power limitations and dark silicon challenge the future of multicore,” *ACM Transactions on Computer Systems (TOCS)* 30.3 (2012): 11.
- [18] Shafique, Muhammad, et al. "The EDA challenges in the dark silicon era," *Design Automation Conference (DAC)*, 2014 51st ACM/EDAC/IEEE. IEEE, 2014.
- [19] Rodrigo, Samuel, et al, “Addressing manufacturing challenges with cost-efficient fault tolerant routing,” *Fourth ACM/IEEE International Symposium on Networks-on-Chip (NOCS)*, 2010. IEEE, 2010.

- [20] P. Mishra, A. N. Bhoj, and N. K. Jha, “Die-level leakage power analysis of FinFET circuits considering process variations,” in *International Symposium on Quality Electronic Design*, pp. 347–355, Mar. 2010.
- [21] S. R. Sarangi, B. Greskamp, R. Teodorescu, J. Nakano, A. Tiwari, J. Torrellas, “VARIUS: A Model of Process Variation and Resulting Timing Errors for Microarchitects”, *IEEE Transactions on Semiconductor Manufacturing*, pp. 3-13 ,Feb 2008.
- [22] A. Jain, W. Anderson, T. Benninghoff, D. Bertucci, M. Braganza, J. Burnette, T. Chang, J. Eble, R. Faber, D. Gowda, J. Grodstein, G. Hess, J. Kowaleski, A. Kumar, B. Miller, R. Mueller, P. Paul, J. Pickholtz, S. Russell, M. Shen, T. Truex, A. Vardharajan , D. Xanthopoulos, T. Zou, “A 1.2 GHz Alpha microprocessor with 44.8 GB/s chip pin bandwidth,” in *IEEE Proceedings International Solid-State Circuits Conference*, pp. 240–241, , Feb. 2001.
- [23] A. Kahng, B. Li, L. Peh, K. Samadi, “ORION 2.0: A Power-Area Simulator for Interconnection Networks,” *IEEE Transactions on Very Large Scale Integration*, pp 191-196, Dec. 2011.
- [24] C. Lee, N. Jha, “FinCANON: A PVT-Aware Integrated Delay and Power Modeling Framework for FinFET-Based Caches and On-Chip Networks”, *IEEE Transactions on Very Large Scale Integration Systems*, pp 1150 – 1163, Apr. 2014.
- [25] A. Tang, Y. Yang, C. Lee, N. Jha, “McPAT-PVT: Delay and Power Modeling Framework for FinFET Processor Architectures Under PVT Variations”, *IEEE Transactions on Very Large Scale Integration Systems*, Sept. 2014.
- [26] C. Sun, C. Chen, G. Kurian, L. Wei, J. Miller, A. Agarwal, L. Peh and V. Stojanovic, "DSENT – A Tool Connecting Emerging Photonics with Electronics for Opto-Electronic

- Networks-on-Chip Modeling", *IEEE/ACM International Symposium on Networks on Chip (NoCs)*, pp 201-210, May2012.
- [27] S. J. E. Wilton and N. P. Jouppi, "CACTI: An enhanced cache access and cycle time model," *IEEE Journal on Solid-State Circuits*, pp. 677–688, May 1996.
- [28] T. Ye, L. Benini, and G. De Micheli, "Analysis of power consumption on switch fabrics in network routers," *IEEE Proceedings Design Automation Conference*, pp. 524–529, Jun. 2002.
- [29] K. Bhardwaj, S. Roy, and K. Chakraborty, "Power-Performance Yield Optimization for MPSoCs Using MILP," *International Symposium on Quality Electronic Design*, pp 764-771, Mar. 2012.
- [30] M. D. Schroeder, A. D. Birrell, M. Burrows, H. Murray, R. M. Needham, T. L. Rodeheffer, E. H. Satterthwaite, C. P. Thacke, "Autonet: a high-speed, self-configuring local area network using point-to-point links," *IEEE Journal on Selected Areas in Communications*, pp. 1318 – 1335 ,Oct. 1991.
- [31] A. Mejia, J. Flich, J. Duato, S. Reinemo, T. Skeie, "Segment-based routing: an efficient fault tolerant routing algorithm for meshes and tori," *IEEE International Parallel and Distributed Processing Symposium*, Apr. 2006.
- [32] A. Mejia, J. Filch, J. Duato "On the Potentials of Segment-Based Routing for NoCs," *IEEE International Conference on Parallel Processing*, pp 594-603, Sept. 2008.
- [33] J. Flich, "A Survey and Evaluation of Topology-Agnostic Deterministic Routing Algorithms", *IEEE Transactions on Parallel & Distributed Systems*, pp. 405-425, Mar. 2012.

- [34] J. Hu, R. Marculescu, "Energy and Performance-Aware Mapping for Regular NoC Architectures," *IEEE Design, Automation, and Test in Europe Conference*, pp. 551 – 562, Jun. 2003.
- [35] G. Ascia, V. Catania, M. Palesi, "Mapping cores on network-on-chip." *International Journal of Computational Intelligence Research 1(1-2)*, pp 109–126, 2005.
- [36] O. He, S. Dong, W. Jang, J. Bian and D. Pan, "UNISM: Unified Scheduling and Mapping for general Networks on Chip," *IEEE Transactions on Very Large Scale Integration Systems*, Aug. 2012.
- [37] G. Ascia, V. Catania, M. Palesi, "Multi-objective Mapping for Mesh-based NoC Architectures," *International Conference on Hardware/Software Codesign and System Synthesis*, pp 182-187, Sept. 2004.
- [38] T. Lei, S. Kumar, "A Two-step Genetic Algorithm for Mapping Task Graphs to a Network on Chip Architecture," *IEEE Proceedings of the Euromicro Symposium on Digital System Design*, 2003.
- [39] C. Rhee, H. Jeong, S. Ha, "Many-to-Many Core-Switch Mapping in 2-D Mesh NoC Architectures," *Proceedings of the IEEE International Conference on Computer Design*, 2004
- [40] R. Tornero, J. Orduna, M. Palesi, J. Duato, "A Communication-Aware Topological Mapping technique for NoCs," *Euro-Par 2008 – Parallel Processing, Lecture Notes in Computer Science Volume 5168*, pp 910-919, 2008.
- [41] R. Tornero, J. Ordua, A. Mejia, J. Flich, J. Duato, "CART: Communication-Aware Routing Technique for Application-Specific NoCs," *IEEE EUROMICRO Conference on Digital System Design Architectures, Methods and Tools*, pp 26-31, Sept. 2008,

- [42] D. Shin, J. Kim, "Power-Aware Communication Optimization for Networks-On-Chip with Voltage Scalable Links," *International Conference on Hardware/Software Codesign and System Synthesis*, pp 170-175, Sept. 2004.
- [43] R. Tornero, S. Kumar, S. Mubeen, J.M. Orduna, "Distance Constrained Mapping to Support NoC Platforms Based on Source Routing," *Euro-Par Workshops, LNCS 6048*, pp 16-25, 2010.
- [44] R. Eberhart, Y. Shi, "Particle Swarm Optimization: Developments, Applications and Resources," *Proceedings of the 2001 Congress on Evolutionary Computation*, 2001.
- [45] K. Wang, L. Huang, C. Zhou, W. Pang, "Particle Swarm Optimization for Traveling Salesman Problem", *IEEE Proceedings of the Second International Conference on Machine Learning and Cybernetics*, Nov 2003.
- [46] N. Kumar, Karambir, R. Kumar, "A Comparative Analysis of PMX, CX and OX Crossover operators for solving Travelling Salesman Problem", *International Journal of Latest Research in Science and Technology, Vol.1, Issue 2*, pp.98-101, Aug 2012.
- [47] S. Ronald, "Preventing Diversity Loss in a Routing Genetic Algorithm with Hash Tagging," *Complex Systems: Mechanism of Adaptation*, pp 133-140, 1994.
- [48] P.K. Sahu, P. Venkatesh, S. Gollapalli, S. Chattopadhyay, "Application Mapping onto Mesh Structured Network-on-Chip using Particle Swarm Optimization", *IEEE Computer Society Annual Symposium on VLSI*, pp 335-336, Jul. 2011.
- [49] N. Kapadia, S. Pasricha, "Process Variation Aware Synthesis of Application-Specific MPSoCs to Maximize Yield", *International Conference on Embedded Systems and International Conference on Embedded Systems*, pp 270-275, Jan. 2014.

- [50] N. Kapadia and S. Pasricha, "VERVE: A Framework for Variation-Aware Energy Efficient Synthesis of NoC-based MPSoCs with Voltage islands," *International Symposium on Quality Electronic Design*, pp. 603-610, Mar. 2013.
- [51] S. Majzoub, R. Saleh, S. Wilton, and R. Ward, "Energy Optimization for Many-Core Platforms: Communication and PVT Aware Voltage-Island Formation and Voltage Selection Algorithm", *IEEE Transactions on Computer-Aided Design*, pp. 816-829, May 2010.
- [52] D. Mirzoyan, B. Akesson, and K. Goossens, "Process-variation aware mapping of real-time streaming applications to MPSoCs for improved yield," *International Symposium on Quality Electronic Design*, pp. 41-48, 2012.
- [53] F. Wang, Y. Chen, C. Nicopoulos, X. Wu, Y. Xie, N. Vijaykrishnan, "Variation-Aware Task and Communication Mapping for MPSoC Architecture," *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, pp. 295 – 307, Feb. 2011.
- [54] <http://lpsolve.sourceforge.net/5.5/>
- [55] C. Bienia, S. Kumar, J.P. Singh, K. Li, "The PARSEC benchmark suite: characterization and architectural implications," *Princeton University Technical Report*, Jan 2008.
- [56] S.V. Woo, M. Ohara, E. Torrie, J.P. Singh, and A. Gupta, "The SPLASH-2 programs: characterization and methodological characterization," *IEEE Proceedings International Symposium on Computer Architecture*, pp. 24-36, May 1995.
- [57] N.B. Williams, C. Fensch, S. Moore, "A communication characterisation of Splash-2 and Parsec", *IEEE International Symposium on Workload Characterization*, pp. 86-97, Oct. 2009.