Deliverable D3.2 - Evaluation of selected memory and communication technologies and exploitation opportunities in compilation and runtime management

Florent Bruguier, Thibaud Delobelle, Charles Emmanuel Effiong, Abdoulaye Gamatié, Pierre-Yves Péneau, Gilles Sassatelli, Sophiane Senni, Lionel Torres, Erven Rohou

CONTINUUM

Project Ref. Number ANR-15-CE25-0007

D3.2 – Evaluation of selected memory and communication technologies and exploitation opportunities in compilation and runtime management

Version 2.0
(2017)
Final version

Public Distribution

Main contributors:
F. Bruguier, T. Delobelle, C. Effiong, A. Gamatié, P.-Y. Péneau, G. Sassatelli, S. Senni, L. Torres (LIRMM); and E. Rohou (Inria)

Project Partners: Cortus S.A.S, **Inria**, **LIRMM**

Project Partner Contact Information

Cortus S.A.S Michael Chapman 97 Rue de Freyr Le Génésis 34000 Montpellier France Tel: +33 430 967 000 E-mail: michael.chapman@cortus.com	**Inria** Erven Rohou Inria Rennes - Bretagne Atlantique Campus de Beaulieu 35042 Rennes Cedex France Tel: +33 299 847 493 E-mail: erven.rohou@inria.fr
LIRMM Abdoulaye Gamatié Rue Ada 161 34392 Montpellier France Tel: +33 4 674 19828 E-mail: abdoulaye.gamatie@lirmm.fr	

Table of Contents

1 Executive Summary

This deliverable presents a number of experiments where non-volatile memory and 3D interconnect technologies are evaluated. The performance and energy consumption are mainly considered as metrics for the evaluation. The goal is to get insights on their impact within the design of typical heterogeneous multicore architectures as expected in the compute node design studied within the CONTINUUM project. From the resulting observations, we discuss possible opportunities enabling us to better leverage the advantages of such technologies by using compilation techniques and runtime system management approaches.

Please note that the contents of this deliverable is mainly based on the results published in conferences or journals by the consortium members of the CONTINUUM project. More technical details could be found in the corresponding references.

2 Introduction

From the surveyed emerging memory and communication technologies presented in deliverable D3.1 [48], we carry out an evaluation of selected technologies in the present report. The main objective is to assess the impact of their presence in typical heterogeneous multicore system designs on performance and energy consumption. From this assessment, we identify possible opportunities for better leveraging the advantages of such technologies via compilation techniques and runtime system management.

The evaluations presented in the sequel are conducted at a cycle-approximate/accurate level by combining popular tools, such as gem5 [3], McPAT [33] and NVSim [17]. In addition, we adopt a cycle-accurate dedicated to NoC simulation [18]. Choosing adequate simulation supports is central for the current study. Though faster, analytical simulation [11, 1, 2] and transaction-level modeling [40, 30, 31] are not accurate enough for providing us with detaile insights.

The rest of the document is organized around three main sections as follows:

- Section 3 first focuses on the evaluation of performance and power consumption when integrating the STT-RAM memory technology in the cache memory hierarchy. Among the candidate non-volatile memory technologies, STT-RAM currently shows the most promising features (reasonable read/write latency and energy consumption) w.r.t. mainstream volatile technologies such as SRAM. The Parsec benchmark suite is used for evaluating various system configurations. It includes applications and kernels covering both compute-intensive and memory-intensive algorithms. The MAGPIE [15, 16, 45] design framework is used for evaluation. It relies on a transformation flow that leverages some of the aforementioned tools.
- Section 4 is devoted to some investigations on application mapping in three-dimensional (3D) NoC-based multicore systems. A cycle-accurate model is used for performance assessment. Through Silicon Via (TSV) provides shorter vertical interconnect which provides higher bandwidth and enhances performance in 3D integration. However, this integration suffers from process variation induced during the manufacturing. One of such defect is the open-resistive defect caused by impurities and/or defect during manufacturing process. It leads to significant signal propagation delay on the TSV links. The presence of a high number of such defective TSVs significantly degrades performance when data travel across TSVs. In this study, we investigated the impact of process variation on a 3D-NoC based on different architectural parameters and application mapping heuristics. This is carried out with the aim of exploring the system performance under process variation and to determine optimal architectural parameters that can mitigate the effects of such process variation.
- Section 5 discusses some opportunities regarding the exploitation of studied memory and communication technologies, by using techniques in compilation and runtime management. Some of these techniques are already under investigation in the project.
- Finally, Section 6 gives concluding remarks.

3 Evaluation of NVM in Cache Memory Hierarchy

We first introduce the MAGPIE (Manycore Architecture enerGy and Performance evaluatIon Environment) evaluation framework [15, 16, 45] that we devised in order to facilitate system design evaluation. MAGPIE makes it possible to carry out the impact assessment of typical non-volatile memory technologies in the cache memory hierarchy, on performance and energy consumption within a heterogeneous multicore architecture design.

3.1 MAGPIE framework

We give an overview of the MAGPIE design evaluation flow. We describe each involved step and its implementation through the integration of existing tools.

3.1.1 Evaluation workflow

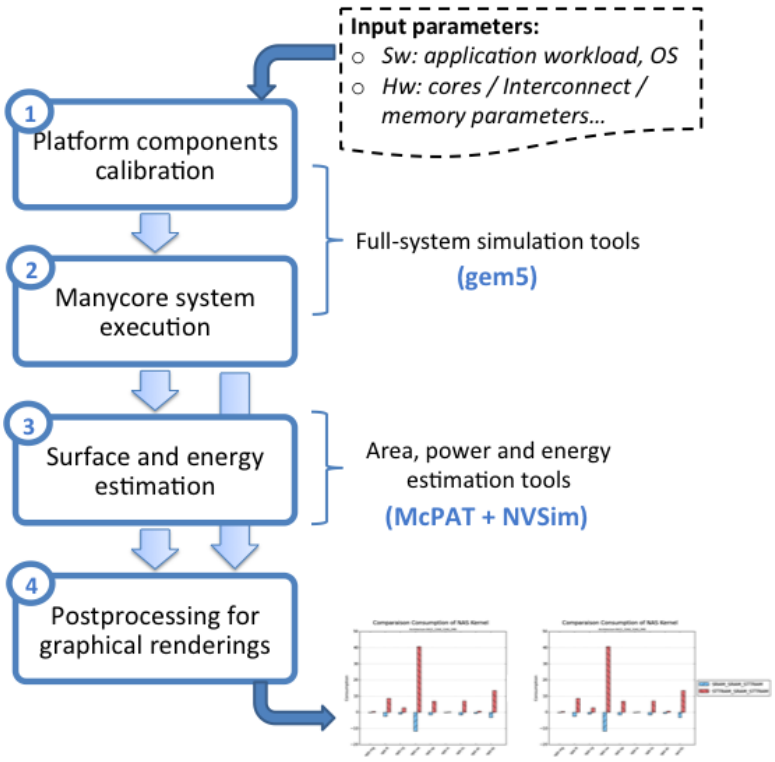


Figure 1: MAGPIE evaluation flow.

MAGPIE[1] framework relies on a generic evaluation flow depicted in Fig. 1. The inputs of the flow comprise information related to the software and hardware parts of the system. The software-related inputs include a gem5 execution script file for each workload/application to be executed, together with

[1]This framework has been developed in collaboration with the GREAT European H2020 project. It is freely distributed via the following address: `http://www.lirmm.fr/continuum-project/pages/magpie.html`

the underlying operating system supported by the considered full-system simulator. From the hardware perspective, a number of parameters of the target manycore architectures are required: types and number of cores, memory hierarchy and its technology-specific properties, and the interconnect type. In general, the candidate full-system simulators for MAGPIE provide an IP library that significantly facilitates the instantiation of target architectures.

Provided the above input information, MAGPIE proceeds through four main steps as follows:

1. Platform components calibration: the basic parameters of all hardware components are set up in the full-system simulator. Typically, the operating frequency of cores, the memory size and access latencies at different levels of the memory hierarchy are defined. For NVMs, their corresponding read/write latencies usually vary according to their characteristics such as their type [35] (e.g., ReRAM, PCRAM, MRAM), technology node (e.g., 65nm, 45nm) and size.

2. Manycore system execution: the full system simulation of the user-customized system is performed in order to obtain detailed execution statistics. Note that since the inputs of the MAGPIE flow can specify at the same time different system design choices, several simulation instances can be launched in parallel. This contributes to accelerate the design space exploration with MAGPIE. Beyond the global performance metrics such as execution time, the activity events related to each hardware device are important information. These events are used for power and energy estimation.

3. Surface and energy estimation: based on detailed execution statistics generated by gem5, the surface, power and energy consumption of the simulated system are estimated. For instance, the dynamic energy of a cache memory is determined based on its collected read/write activity events and the energy consumption of elementary memory access obtained, e.g., with CACTI.

4. Post-processing for graphical renderings: as a major goal of MAGPIE is to assist the user in design space exploration, the final evaluation metrics can be reported in both textual and graphical formats. Then the user can quickly gather insights from each evaluated design.

The next section presents an implementation of the above flow within MAGPIE.

3.1.2 Implementation

We seamlessly combine the gem5 simulator for performance evaluation, and NVSim and McPAT for estimating the energy respectively related to NVMs and the rest of the architecture. These tools are briefly described below.

Considered simulation and estimation tools. The gem5 [3] provides an accurate evaluation of system performance [6] thanks to its high configurability for a fine grained architecture modeling. Its full-system simulation mode runs unmodified operating systems. It includes several predefined architecture component models, e.g., CPU, memory and interconnect. This simulator produces detailed execution statistics (even at micro-architecture level) for power and footprint area estimation.

McPAT [33] is a power, area and timing modeling framework for multi-threaded, multicore, and manycore architectures. It works with a variety of performance and thermal simulators via an XML template-based interface. This interface describes the micro-architecture specification and is used to communicate activity events generated by simulators. McPAT covers three simulation levels for estimation: architectural, circuit and technology (from 90nm to 22nm). NVM technologies are not addressed by McPAT. So, we use NVSim [17], which is a circuit-level estimator for NVM performance, energy and area estimations. It supports different NVM technologies including STT-MRAM, ReRAM, and PCRAM. It uses the same modeling principles as CACTI.

Integration within MAGPIE framework. The MAGPIE framework defines several Python script programs that automates the whole flow illustrated in Fig. 1. The inputs of the flow are first read and used for an automatic calibration of the hardware architecture components in gem5. For NVMs, the NVSim tool is invoked by a script in order to calculate the corresponding read/write latencies based on the desired memory type, memory size, associativity and technology node, as specified in the inputs. Then, gem5 is automatically configured with the computed NVM access latencies. For this purpose, we modified gem5 so as to enable the configuration of memories with asymmetric read and write latencies, such as NVMs. Afterwards, the specified system execution scenarios are run in parallel (according to the number of cores available on the host machine) by automatically triggering the corresponding number of gem5 simulation instances.

Each gem5 simulation instance produces the execution statistics file related its design scenario. From these files, all data required by NVSim and McPAT (e.g., execution time of the system, number of read/write transactions for memory blocks) are automatically extracted by another script. As these files can be huge, the script has been defined in such a way that it optimizes the reading of generated gem5 files. Then, it invokes the two estimation tools on the extracted data in order to generate the area, power and energy consumption for each captured design scenario. The results are stored in textual files.

Finally, the above textual files are post-processed by several scripts for generating various user-friendly renderings: CSV files and graphical plots that compare the performance, area, power and energy evaluation of the different scenarios. For instance, the energy breakdown of the main hardware components can be easily plotted for a fine-grained analysis.

Though, MAGPIE does not adopt a metamodeling-based transformations as promoted in existing design frameworks [39, 23, 14], it somehow follows a model-driven engineering approach, where XML is one prominent intermediate representation formalism.

3.2 Evaluation of system configurations

3.2.1 Experimental setup

As a starting point, let us consider the Exynos 5 Octa (5422) chip sketched in Figure 2. It features two quad-core clusters: “big” and “LITTLE” running at 1GHz. Its main parameters are displayed in Figure 2. Each core has its private instruction/data L1 caches (32kB each), while each cluster has

a single shared L2 cache. The L2 caches are connected to the DRAM memory via a 64-bit cache coherent interconnect. The chip incorporates its own system memory in the form of 2GB LPDDR3 RAM integrated in a Package-on-Package (PoP) fashion. This architecture has been modeled in gem5 and models have been calibrated so as to provide sufficient accuracy against this SoC [7, 9].

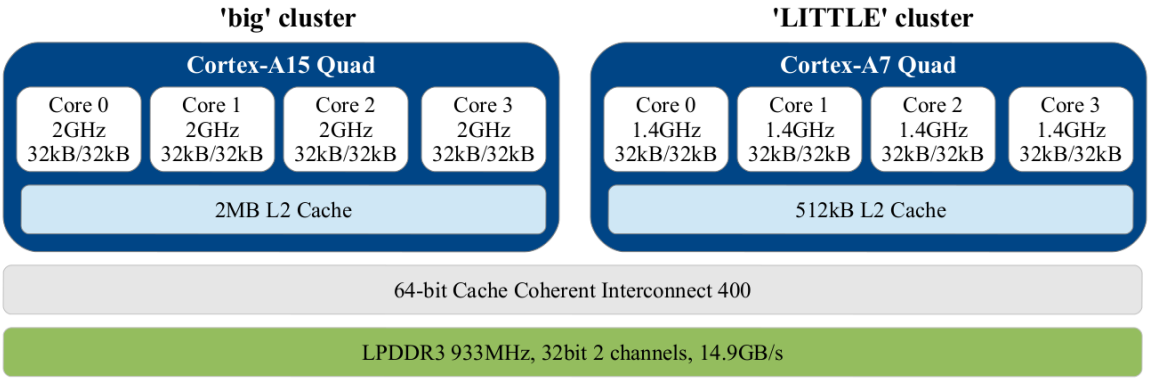


Figure 2: Exynos 5 Octa big.LITTLE architecture

From this initial template, we vary the number of big and LITTLE cores such that the total number of cores is always equal to eight. In the sequel, each architecture configuration composed of X big Cortex-A15 cores is denoted by "XA15". For instance, "0A15" denotes a homogeneous configuration composed of eight LITTLE Cortex-A7 cores while "1A15" refers to a configuration with seven LITTLE Cortex-A7 cores and one big Cortex-A15 core.

The aim is to assess the behavior of such system configurations in presence of MRAM technology. In particular, we consider a 45nm STT-RAM technology, integrated at last-level cache, i.e., L2 cache in the modeled system. In the reported experiments, system designs in which all L2 caches are in SRAM (resp. STT-RAM) are denoted with a suffix indicating the corresponding memory technology, i.e., "-SRAM" (resp. "-MRAM"). Integrating STT-RAM at L1 cache without adequate optimization can be penalizing in case of frequent memory access due to the higher access latency of such a technology compared to SRAM. In [4, 5, 49], we advocated *silent store elimination* as a useful technique for mitigating this issue while considering non-volatile memory in L1 cache.

The Parsec 3.0 benchmark suite is used to evaluate the above architecture design. Figure 3 summarizes the whole Parsec benchmark suite. The considered kernels and applications are compiled for ARMv7 cores. A Linux 3.14 operating system is used. Kernels are executed with their small input sets.

3.2.2 Evaluation results

Figures 4, 5, 6 and 7 respectively report for each Parsec 3.0 kernels or applications the corresponding temporal behavior and on-chip energy-to-solution considering the above scenarios. The estimated energy comprises the following on-chip devices: eight CPUs with their associated L1-instruction and L1-data caches, the two L2 caches, the bus interconnect and main memory controller.

More generally, we observe in those figures that integrating STT-RAM at L2 cache level slightly incurs a performance penalty due to the well-known higher write latency compared to SRAM [42, 43, 44].

Program	Application Domain	Parallelization		Working Set	Data Usage	
		Model	Granularity		Sharing	Exchange
blackscholes	Financial Analysis	data-parallel	coarse	small	low	low
bodytrack	Computer Vision	data-parallel	medium	medium	high	medium
canneal	Engineering	unstructured	fine	unbounded	high	high
dedup	Enterprise Storage	pipeline	medium	unbounded	high	high
facesim	Animation	data-parallel	coarse	large	low	medium
ferret	Similarity Search	pipeline	medium	unbounded	high	high
fluidanimate	Animation	data-parallel	fine	large	low	medium
freqmine	Data Mining	data-parallel	medium	unbounded	high	medium
raytrace	Rendering	data-parallel	medium	unbounded	high	low
streamcluster	Data Mining	data-parallel	medium	medium	low	medium
swaptions	Financial Analysis	data-parallel	coarse	medium	low	low
vips	Media Processing	data-parallel	coarse	medium	low	medium
x264	Media Processing	pipeline	coarse	medium	high	high

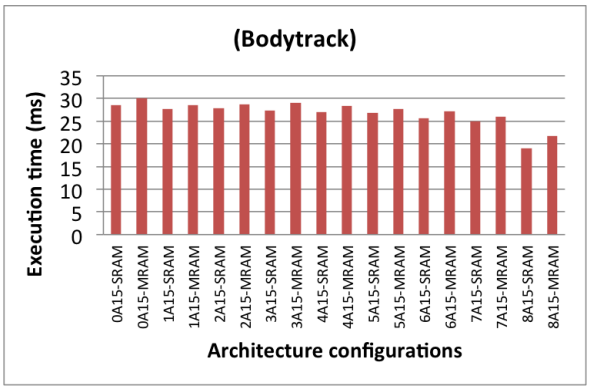
Figure 3: Parsec benchmark suite

The percentage of observed increase in execution time is marginal throughout the different evaluated cases.

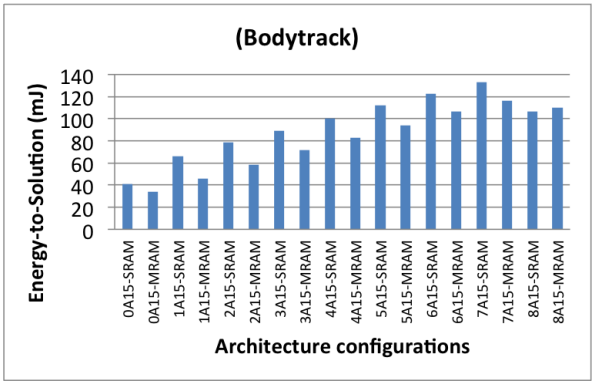
Regarding Energy-to-solution (EtoS), we observe that the major part of configurations integrating STT-RAM in L2 cache shows an improvement of the power consumption compared to SRAM. The energy related to an L2 cache in NVM is determined with NVSim, while McPAT is used to estimate the energy of the other on-chip devices. The obtained energy reduction appears important when the system comprises more LITTLE cores compared to big cores.

While the CONTINUUM project targets around 30% energy reduction in its investigated design solutions, the reported evaluation suggests that this can be reached via a careful integration of NVMs with adequate choice of heterogeneous cores. For instance, the asymmetric heterogeneous eight-core configuration composed of a single big Cortex-A15 core and seven LITLLE Cortex-A7 cores can provide such energy reduction as shown in Figures 4(b) and 4(d). At the same time, Figure 4(f) shows that the same configuration can lead to no gain for some specific application. This suggests that application nature must be also taken into account in order to adapt the system configuration for a better outcome in term of energy. For instance, in Figure 4(f), combining two Cortex-A15 and six Cortex-A7 cores gives better energy reduction. On the other hand, among the whole evaluated scenarios, we observe that the smallest energy reduction is obtained with homogeneous multicore configurations, and in particular with eight big Cortex-A15 cores.

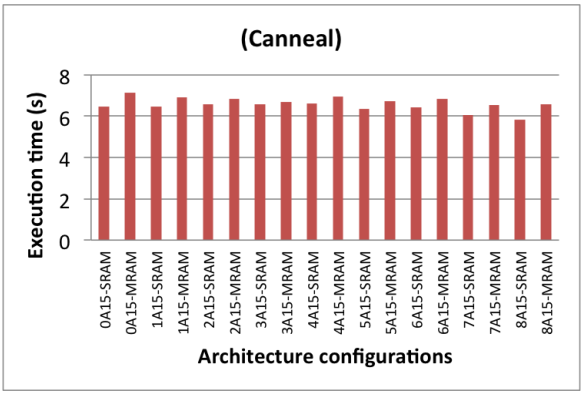
The above evaluations confirm the relevance of NVM technologies such as STT-RAM for energy gain when considered in memory hierarchy. We have been carrying out some complementary studies aiming at leveraging advanced cache replacement techniques [37] in presence of NVMs, for further performance and energy improvements. Finally, some complementary directions, including software optimization techniques, multicore architecture designs and adaptive workload management, are discussed later in Section 5.



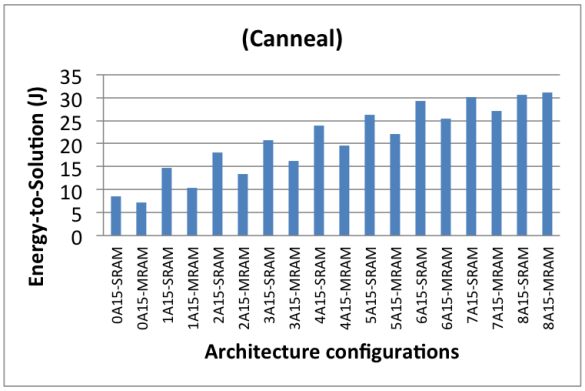
(a) Execution time (bodytrack)



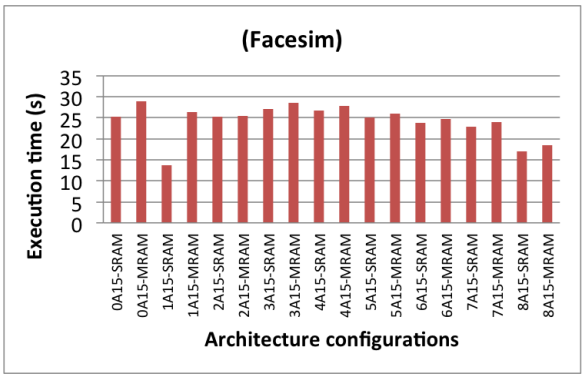
(b) Energy-to-Solution (bodytrack)



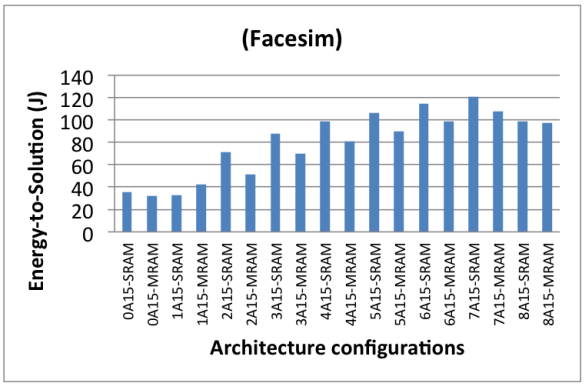
(c) Execution time (canneal)



(d) Energy-to-Solution (canneal)

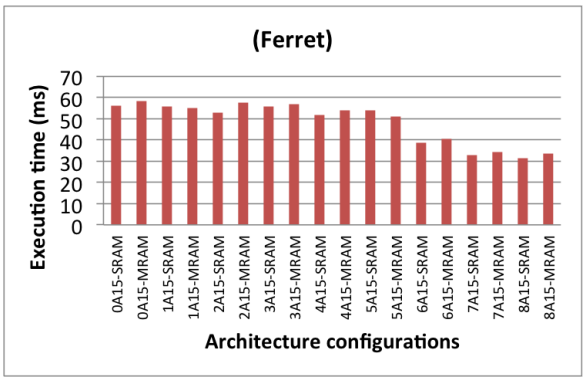


(e) Execution time (facesim)

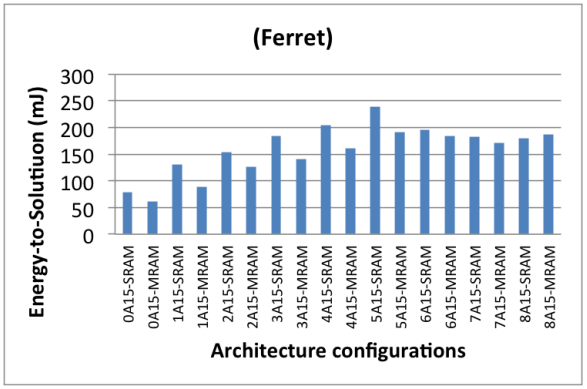


(f) Energy-to-Solution (facesim)

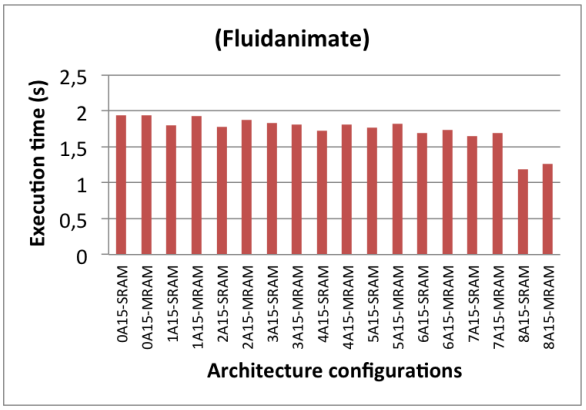
Figure 4: Execution time and energy-to-Solution evaluation for Parsec (part 1)



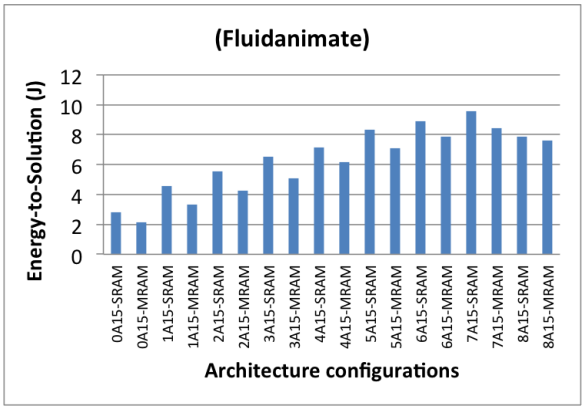
(a) Execution time (ferret)



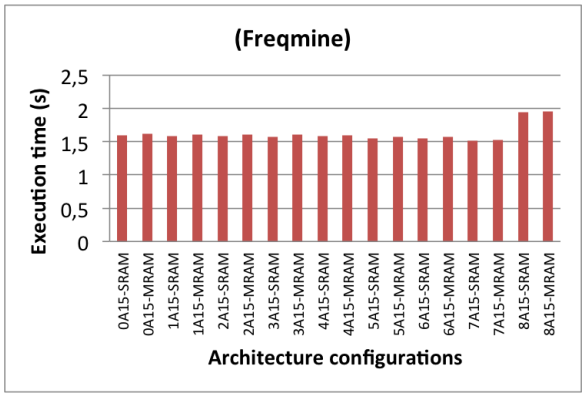
(b) Energy-to-Solution (ferret)



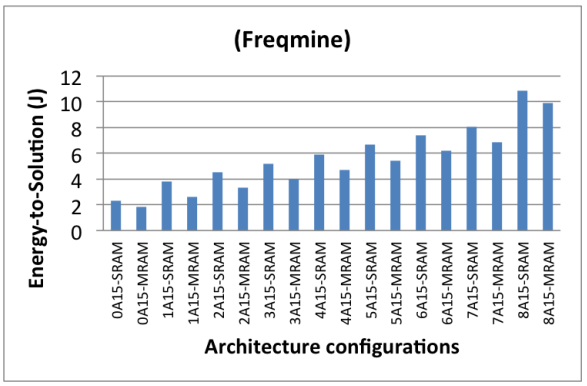
(c) Execution time (fluidanimate



(d) Energy-to-Solution (fluidanimate)

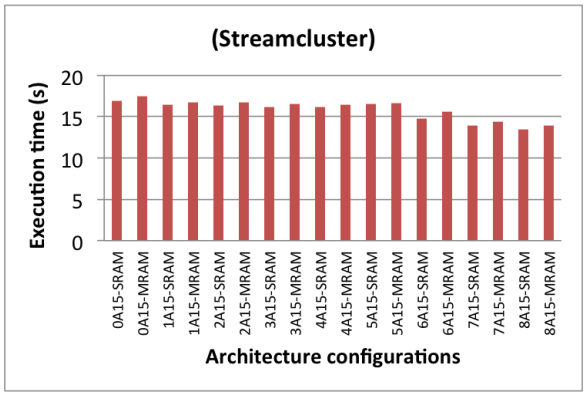


(e) Execution time (freqmine)

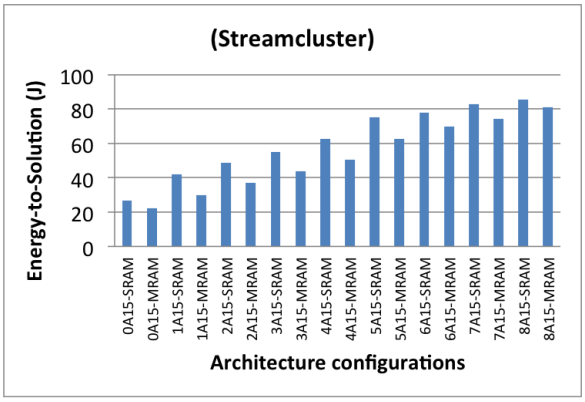


(f) Energy-to-Solution (freqmine)

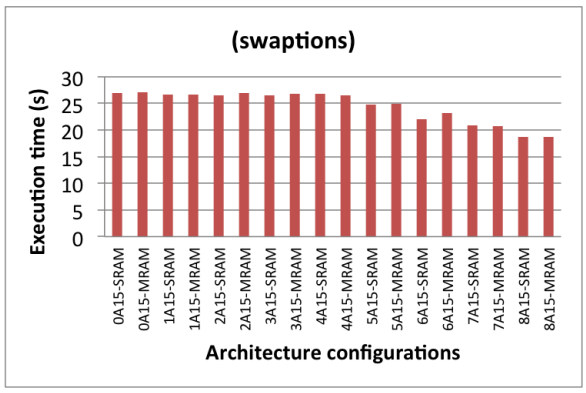
Figure 5: Execution time and energy-to-Solution evaluation for Parsec (part 2)



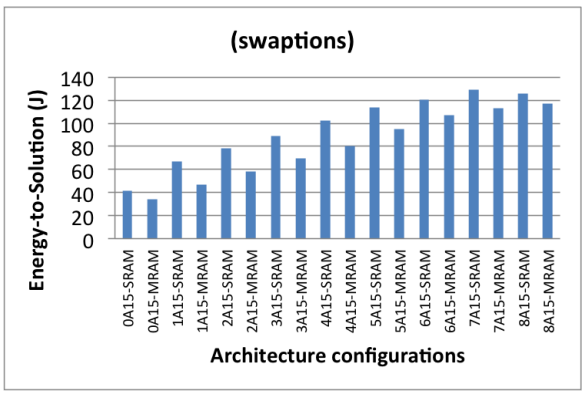
(a) Execution time (streamcluster)



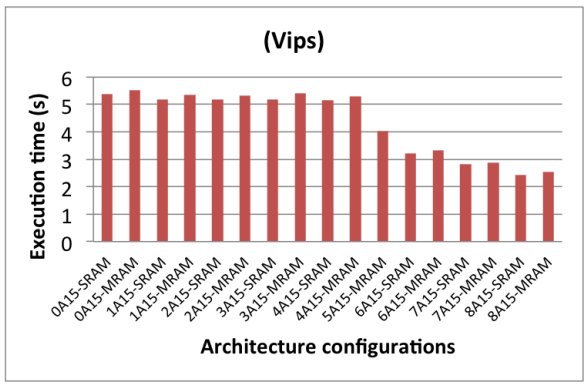
(b) Energy-to-Solution (streamcluster)



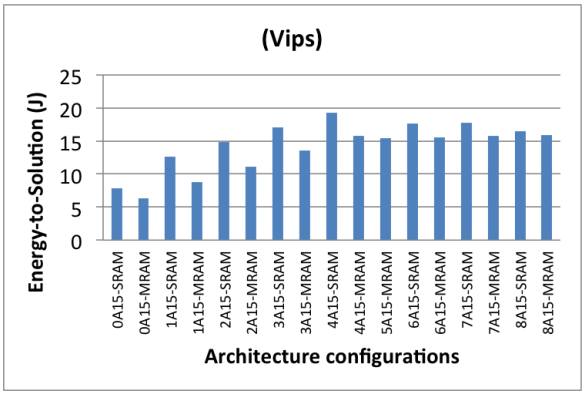
(c) Execution time (swaptions)



(d) Energy-to-Solution (swaptions)

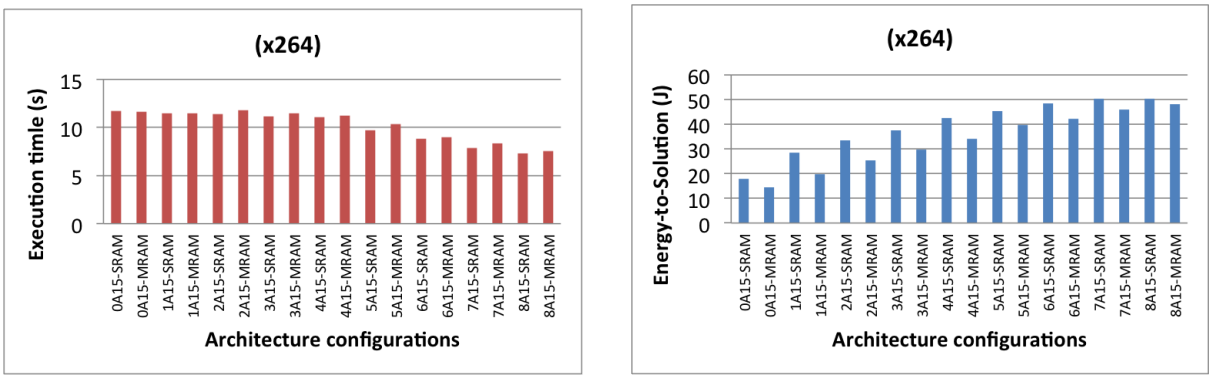


(e) Execution time (vips)



(f) Energy-to-Solution (vips)

Figure 6: Execution time and energy-to-Solution evaluation for Parsec (part 3)



(a) Execution time (x264)

(b) Energy-to-Solution (x264)

Figure 7: Execution time and energy-to-Solution evaluation for Parsec (part 4)

4 Evaluation of a 3D NoC Model

The CONTINUUM project initially aims at using NoCs as candidate interconnects in the considered multicore heterogeneous compute node. In particular, 2D NoCs were the main target, while some investigations were also planed on 3D NoCs in order to identify potential avenues for energy-efficiency improvement when such advanced technologies will become mainstream in manufactured SoCs.

However, our ongoing work shows that the Cortus technology, which has been adopted for implementing our compute node design, rather accommodates crossbar instead of NoC. Indeed, there is a trade-off between complexity (die area required, power consumption), transfer speed, latency and throughput. Very simple systems composed of a few cores can use a bus. As the complexity increases, a crossbar becomes more attractive, allowing multiple accesses between cores via high-speed paths. Since the number of potential paths between cores increases the complexity of the crossbar increases to a point that a large portion of the die is reserved for the crossbar and timing closure becomes increasingly difficult. At this point a NoC becomes better. The cost in terms of die area and power consumption of the network and the increase in latency are comparatively lower. Nevertheless, the point at which a network-on-chip becomes necessary can be postponed by using a multi-level crossbar system proposed by Cortus, where the number of communicating cores per crossbar is reduced. This is therefore the solution adopted in the CONTINUUM project.

The rest of this section is devoted to another study about 3D interconnect as introduced before so as to identify possible exploitation opportunities in the future, beyond the current project.

4.1 3D NoC design challenge

Three-dimensional (3D) integration provides improved performance, increased package density, noise reduction, smaller footprint and reduced power consumption than conventional two dimensional 2D process [22, 27]. Unlike the 2D process, 3D process exploits the Z-direction vertical to enhance system performance. This is achieved by stacking multiple dies in the Z-direction and interconnecting them using Through silicon via (TSV). TSV provide shorter wire length which corresponds to reduced

power and increase interconnection density, thereby enhancing the overall system functionality and performance [28, 52].

On the other hand, 3D manufacturing process can lead to process variation caused by additional processing and stacking steps [18]. Process variation reduces manufacturing yield and leads to significant performance degradation. As stated in [50], TSV delays can vary significantly due to defects and/or impurities that are introduced during the manufacturing process. Such defect is known as *open-resistive defect*. TSV links with open resistive defect maintain *weak* electrical connection between dies, which leads to significant signal propagation delays [34, 29].

Fig. 8 shows a partially connected three-layer 3D-NoC with non-defective and open-resistive TSVs links between the layers. Such heterogeneous configuration leads to unbalanced data propagation delay, where a data travel across defective TSV is slower due to defect caused by the manufacturing process. The overall system performance is impaired because of accumulating delays incurred when traversing many of such defective TSV links. Therefore, from a performance evaluation point of view, it is imperative to investigate the performance of communication architectures under process variation, with a view to finding out the best and worse case system performance even under process variation.

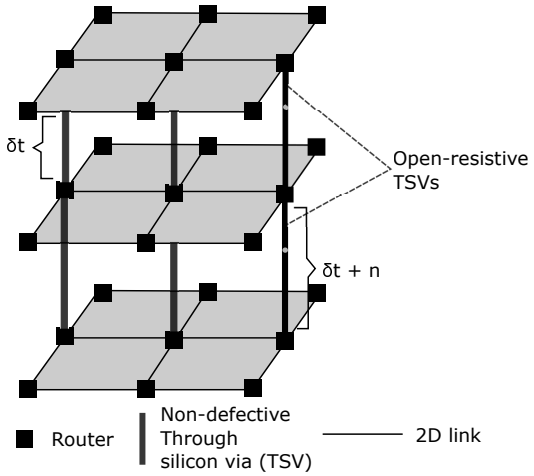


Figure 8: Partially connected 3D-NoC with open resistive TSVs. δt represents delay to traverse a non-defective TSV, while $\delta t + n$ represent n additional delay caused by open-resistive defect

One possible approach to detecting defective TSVs is to do a post silicon validation test. This involves prototyping on actual silicon but before product release. This approach incurs cost and design time overhead. The use of redundant TSVs to replace failed TSVs have been proposed [26, 53]. This method increases silicon area, cost and design complexity. A recent approach suggested the use of asynchronous delay-insensitive logic for inter-tier links [13]. This method makes it possible to exploit TSV links regardless of their delays. This approach however leads to an asymmetric NoC communication performance [18].

In the current study, we follow a different approach to address the problem by exploring the extent to which the communication architecture performance is affected by process variation. First, we carry out extensive analysis on the impact of process variation on communication performance depending on state-of-the-art mapping heuristics of real world applications and different architectural parameters

such as TSV size, number of TSVs between each pair of communicating nodes, etc. Finally, we explore how the different architectural parameters can be combined to alleviate the detrimental effects of defective TSVs.

4.2 Experimental implementation of a 3D NoC architecture

The considered architecture is Globally Asynchronous Locally Synchronous (GALS) based, where synchronous router communicates asynchronously. Two different schemes were used to provide asynchronous communication between the routers. These are: i) bi-synchronous FIFOs for intra-layer communication between two communicating routers, ii) fully asynchronous serialized vertical links for inter-layer communication. Bi-synchronous FIFOs provide a low cost, scalable and area efficient interface for routers with different clock frequencies and phase [36]. The inter-layer communication scheme between two routers residing one hop away from each other at two consecutive layers is given in Figure 9.

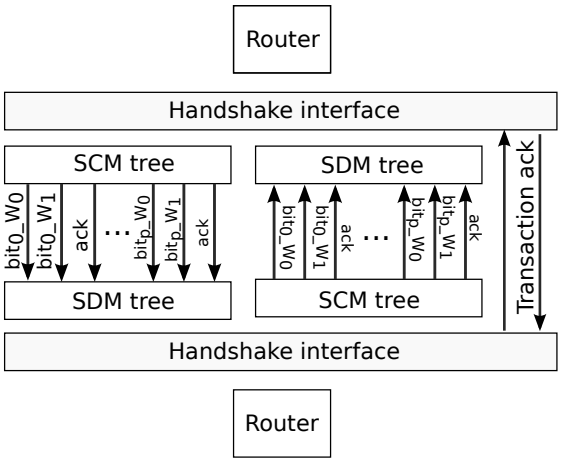


Figure 9: Interlayer communication

The vertical links were serialized using fully asynchronous Quasi Delay Insensitive (QDI) asynchronous logic presented in [13]. Data transfer on the vertical link is encoded using four-phase dual-rail asynchronous protocol. This protocol uses two wires to encode one bit of information to be transmitted. An additional wire is needed to send back an acknowledgement from the receiver to the sender. Therefore, a total of three TSVs is needed to transmit one data bit on the vertical link. This protocol is delay-insensitive, ensuring that the vertical links behave correctly and reliably regardless of link delay caused by voltage, temperature, or manufacturing process variations.

Figure 9 shows the up and down vertical links between two communicating routers, where each arrow represents a TSV. The serial channel consists of trees of autonomous multiplexers used for serializing, i.e., self-controlled multiplexers (SCM) and de-serializing, i.e., self-controlled-demultiplexers (SDM) [13] data traversing the vertical links. The number of TSVs between each pair of communicating router varies according to the serialization rate. For a 3D-NoC with N-bits communication data, 3N TSVs are need to transmit the data through a vertical link, i.e., no serialization. This is because, 3 TSVs (i.e., bit_Woi, bit_W1i and acki) are needed to transmit 1 parallel bit data through the link as

shown in Fig. 9. Serialization is used to reduce the number of TSV links in the network. This is achieved by multiplexing bits in time according to the serialization rate (i.e. the number of parts a flit must be divided into).

The additional TSV link referred to as `Transaction ack` in Figure 9, is used by the handshake interface to connect the asynchronous channel with the router and to inform the router when a new transaction can be initiated. For this signal to be valid, the receiving router must be able to receive a data and data corresponding to the previous transaction must have been sampled.

Since the architecture uses purely asynchronous logic design, the serialization subsystem performance solely depends on propagation delays of both gates and TSVs. The latency of each TSV is therefore accounted for in the description of each asynchronous serializer of the simulation model. This approach makes it possible to accurately analyze various metrics such as bandwidth and communication latencies in the NoC.

4.3 Application mapping on NoC-based multicore architecture

Application mapping is crucial in multi/many core systems because of vast application requirements. Applications are normally decomposed into a set of tasks which can be executed in parallel on different cores. Mapping heuristics determines how application tasks are mapped on the cores. The choice of mapping heuristics for a given application determines if the application requirements will be met or not. Mapping application tasks on multi/many core systems is carried out with a view to optimizing criteria such as compute performance and energy consumption [46]. In order to carry out our performance exploration, we consider some of the well-known application mapping heuristics and investigate the performance of the network depending on the mappings. In this work, we consider only static application mapping for simplicity. We also assume that only one task is mapped on a node. The considered mappings are briefly explained in the subsequent subsections.

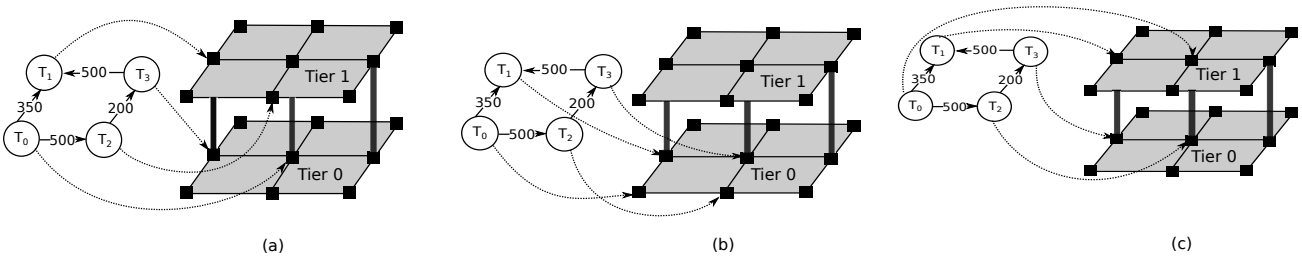


Figure 10: Application mapping heuristic (a)3D-Mincomm mapping (b) 2D-Mincomm mapping (c) Critical path mapping

3D Minimum communication mapping In 3D minimum communication mapping (3D-MinComm), tasks with the most communication are mapped close to each other so that they can communicate using the vertical link. The goal of this mapping is to exploit the high bandwidth TSVs in order to optimize performance. Fig. 10(a) shows a 3D-MinComm mapping of an application with four tasks onto a two-tier communication architecture. In the application task graph, the nodes represent tasks, while the number between two nodes represents the communication

volume. Each node on the communication architecture has a corresponding *xyz* address. If two tasks exchange large communication volume, 3D-MinComm attempts to map the first task on a node with address x_n, y_n, z_n and the second task on another node with address x_n, y_n, z_{n+1} or x_n, y_n, z_{n-1}. This is shown in Fig. 10(a) where Task T_0 is mapped on a node with address x_1, y_1, z_0 and Task T_1 is mapped on a node with address x_1, y_1, z_1.

2D Minimum communication mapping Unlike 3D-MinComm mapping, 2D Minimum communication mapping (2D-MinComm) allocates application tasks with large communication close to each other mainly on the same tier. 2D-MinComm exploits the horizontal links to optimize performance by reducing the number of hops a packet takes from its source to destination node. Fig. 10(b) shows a possible 2D-MinComm mapping of application tasks onto 3D-NoC where the communicating tasks mainly utilize the horizontal links.

Critical path mapping In *critical path* mapping (CP), the tasks on the longest path of the application task graph are mapped first before the other tasks. The goal of this mapping is to reduce network contention and packet latency on those paths while exploiting TSVs as much as possible. As shown in Fig. 10(c) tasks T_0, T_1, and T_2, which are on the critical path are mapped beginning from bottom right of the first tier onward.

Least-Comm mapping In *Least-comm-middle* (LCM) mapping, tasks with the lowest communication activity are mapped at the middle tier(s). This is done with a view to balancing the load on the network since more packets tend to traverse the middle tier(s).

4.4 Performance evaluation

In order to assess the performance of the NoC under process variation, we inject traces from real-world application benchmarks into the network. The applications include: video conference encoder (VCE), Wifi baseband receiver (WIFI), multimedia system (MMS) and E3S consumer benchmarks [51]. The applications characteristics are given in Table 1.

Table 1: Application benchmarks characteristics

Application	Number of tasks	Communication volume (packets)
VCE	24	52060
MMS	25	644098
E3S	12	131
WIFI	25	2160798

We carry out some evaluations by considering a 3x3x3 mesh network topology. This network uses *ZXY* dimension ordered routing and a credit based flow control without virtual channels. In order to investigate the network performance under variability, we simulated the network while taking into account different design parameters, i.e., serialization rates, TSV sizes and application mappings.

Impact of application mapping on 3D-NoC

The goal of this experiment is to investigate how the mapping heuristics described previously impact the network performance particularly in terms of network link utilization. Links with extremely high utilization create hotspots in terms of temperature and power consumption.

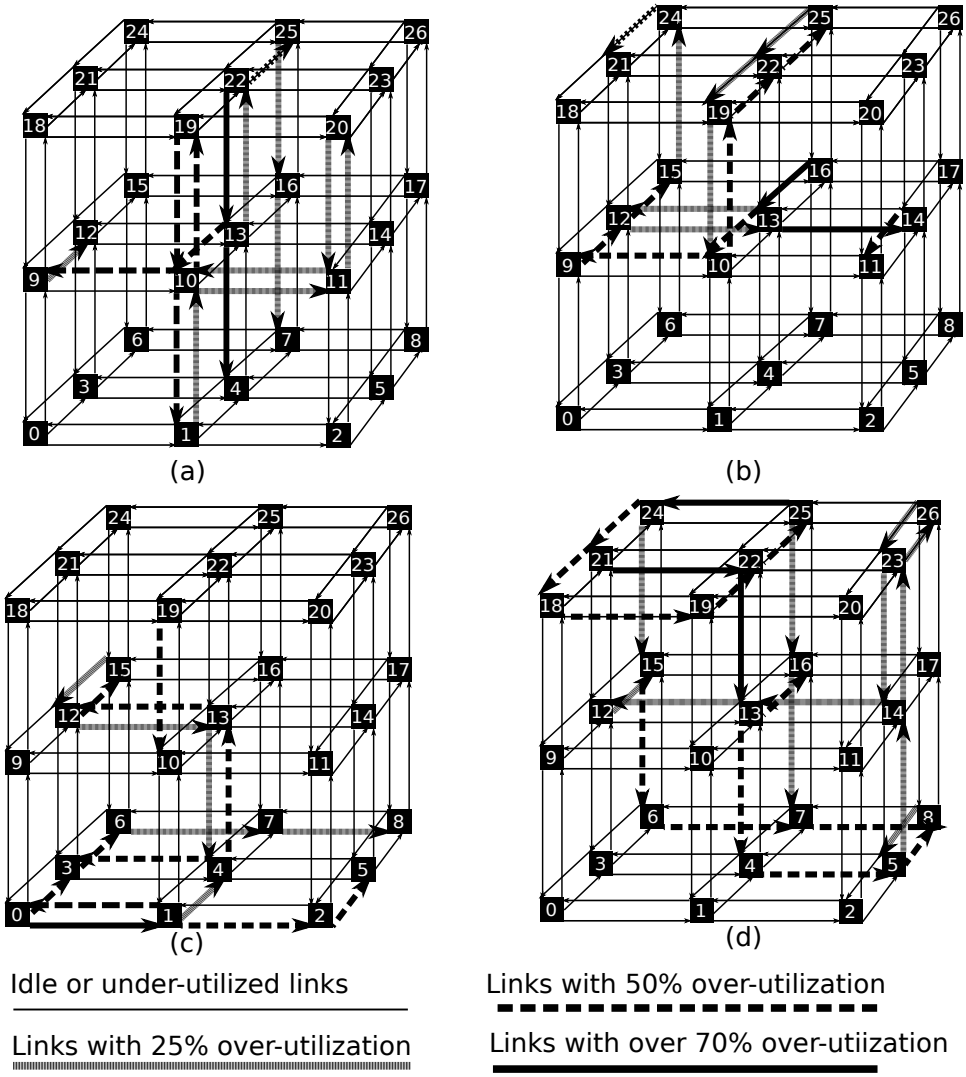


Figure 11: Network links utilization of VCE application based on mappings: (a) 3D-MinComm, (b) 2D-MinComm, (c) CP, (d) LCM.

Fig. 11 shows the link utilization of the mappings for VCE application. We have considered links with the load greater than a threshold to be *over-loaded*. It can be observed that for 3D-MinComm mapping, most of the TSVs links are over-loaded, while most 2D-links are either idle or under-utilized. The reason being that this mapping attempts to exploit the high bandwidth TSVs. Similar argument can be made for 2D-MinComm mapping except that in this case the 2D-links are overloaded. Also, from Fig. 11(b) the middle and top layers are overload, while the bottom layer remains under-utilized. This creates unbalanced network load corresponding to hotspots in the over-utilized layers. In similar

manner, CP mapping creates hotspots in the first two layers while, the topmost layer is not over-utilized as seen in Fig. 11(c). Hotspots appear on each layer of the network for LCM mapping as seen in Fig. 11(d). Compared to the other mappings, a greater percentage of the network links are over-utilize for LCM mapping.

Impact of architectural parameters on 3D-NoC performance

In order to investigate the impact of architectural parameters on 3D-NoC performance, we mapped the applications introduced previously on 3D-NoC with three configurations : a) architecture without serialization and process variation [*Arch_only*], b) architecture with serialization but without process variation [*Arch_SER*], c) architecture with serialization and process variation [*Arch_SER_PV*]. We considered serialization rate of 2 for the second and third configurations. Serialization rate of 2 implies that 16 TSVs are used between each pair of communicating router in the vertical direction. For the third configuration, we considered small TSVs (*SM*) out of which 15% are defective i.e. they have large-open resistive defect. We considered 400 ps delay for the defective TSVs.

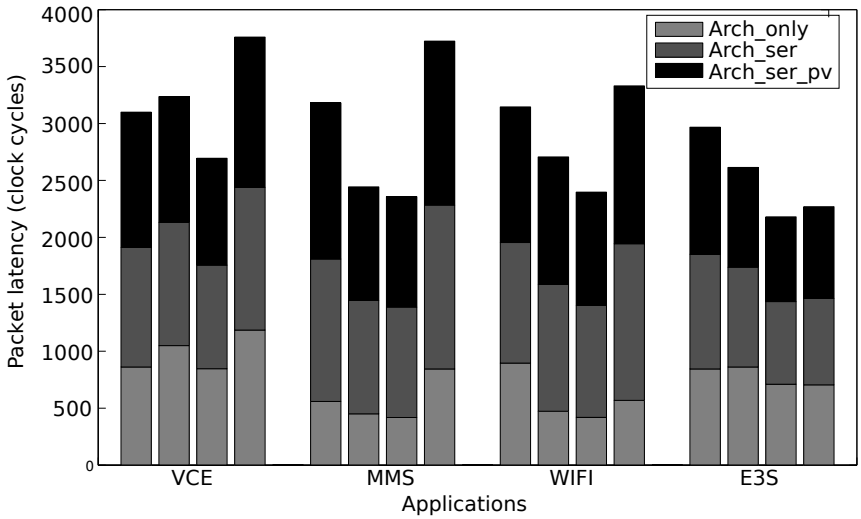


Figure 12: Impact of architectural parameters on 3D-NoC performance

Fig. 12 shows the performance of the network configurations based on the application mappings. It can be observed that for all the applications, the network packet latency increases significantly for the *Arch_SER* when compared to *Arch_only*. To illustrate this point, let us consider the MMS application. We observe an increase of over 123% in packet latency for *Arch_SER* compared to *Arch_only* configuration for *3D-MinComm* mapping. The reason being that *Arch_only* has twice as many vertical links as *Arch_SER* configuration. Therefore, more flits can travel in parallel in the *Arch_SER* than in *Arch_SER*. This demonstrates that vertical link serialization can significantly impair network performance especially when more communications utilize the vertical links. The network packet latency is further degraded for *Arch_SER_PV* when compared to *Arch_only* configuration. For the MMS application, we observed that the packet latency increases by 145% for *Arch_SER_PV* when compared to *Arch_only 3D-MinComm* mapping. Similar argument can be made for the other applications with *3D-MinComm* mapping. This indicates that open-resistive TSVs lead to network

performance degradation. However, for the other mappings, TSV defects do not significantly degrade the network performance for *Arch_SER_PV* when compared to *Arch_SER* configuration. The reason being that these mappings do not mainly utilize the TSV links as opposed to *3D-MinComm* mapping that seeks to exploit the TSV links, but suffer significant performance loss when the TSV links are defective.

Further analyses of the results reveal that the serialization rate and process variation do not significantly impair the network performance for VCE and E3S applications using the other mappings (i.e. CP, LCM, and 2D-MinComm). One possible reason for this is that for these applications and mappings, only few of the total number of available vertical links are utilized. Therefore, the network performance is not significantly impaired since serialization and process variation concern only the vertical links. As an example, the E3S application has only 131 tasks therefore, the application can be mapped using CP, LCM, and 2D-MinComm in such as way that mainly the 2D-links are utilized.

Impact of serialization on 3D-NoC performance

In order to obtain a general optimal serialization rate for a 3D-NoC, we have considered the configurations shown in Table 2 for our simulations. In this table, *Def-TSVs* refers to defective TSVs.

Table 2: Serialization rates network configuration

App	Mapping	SR	% of Def-TSVs	TSV size
VCE	3D-MinComm	2	-	SM
			15	
		4	-	
			15	

Fig. 13 shows the resulting network packet latency for each configuration. It can be observed that the network packet latency increases by 16% for serialization rate of 4 when compared to a serialization rate of 2 for a network without defective TSVs and by 29% for a network with defective TSVs. This demonstrates that although serialization rate of 4 can reduce the network area overhead by 75% when compared to a network without serialization, significant performance degradation occur. Therefore, the trade-off between performance and area cost.

Impact of TSV size on 3D-NoC performance

In order to obtain a general optimal TSV size for a 3D-NoC, we have considered the configurations shown in Table 3 for our simulations. Here, *D-TSVs* represents defective TSVs. We have considered a delay of 48 ps, 190 ps, 400 ps for defective large (LG), medium (MD) and small (SM) TSVs respectively.

Fig. 14 shows the performance of the network with the different TSVs. It can be observed that network packet latency is increased by 4%, 13% for MD and SM TSVs respectively when compared to LG TSVs. The network packet latency is not significantly impaired for MD TSVs when compared to LG TSVs. The reason is that the vertical link scheme utilizes quasi delay-insensitive asynchronous logic

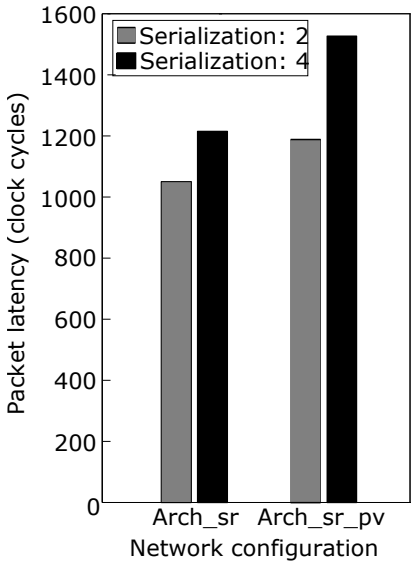


Figure 13: Performance of 3D-NoC with serialization rates

Table 3: Network configurations for TSV size

App	Mapping	SR	% D-TSVs	TSV size
VCE	3D-MinComm	2	15	LG
				MD
				SM

which ensures that bandwidth of TSV links with resistive defect can still be exploited regardless of their propagation delay.

Impact of process variation on 3D-NoC performance

In order to explore the extent to which process variation affects a 3D-NoC performance, we have considered the configurations shown in Table 4 for our simulations. We compare a network without defective TSVs to networks with defective TSVs.

Table 4: Network configuration for process variation impact

App	Mapping	SR	TSV size	% of Def-TSVs
VCE	3D-MinComm	2	15	-
				5
				10
				15
				20

Fig. 15 shows the performance of the network with process variation. It can be observed that network packet latency increases as the number of defective TSVs. The network latency is increased by 4%, 9%, 13%, 15% for 5%, 10% and 15% and 20% defective TSVs when compared a network without any

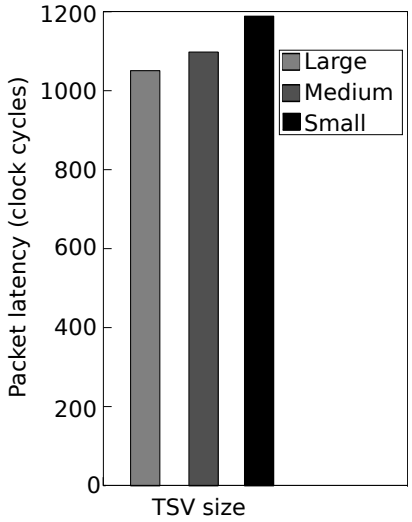


Figure 14: Performance of 3D-NoC configurations with TSV sizes

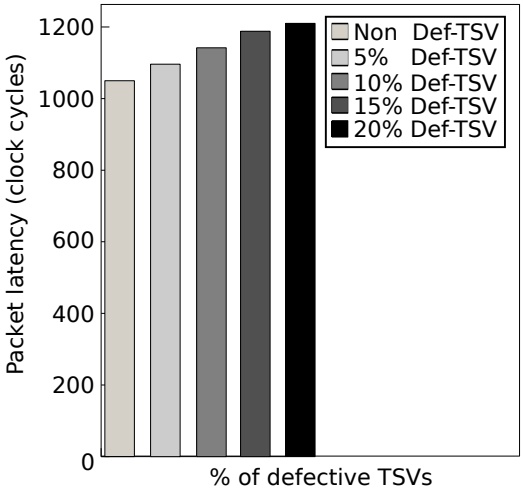


Figure 15: Performance of 3D-NoC configurations with defective TSVs

defective TSVs. This result shows that the presence of a large number of defective TSVs in a 3D-NoC leads to performance degradation and yield loss.

5 Opportunities in Compilation and Runtime Management

From the experiments carried out in previous sections, a number of insights are identified regarding potential opportunities for leveraging both compilation techniques and resource allocation techniques via suitable runtime management.

The main directions we already identified are summarized through the following items:

Energy-efficiency improvement by addressing the asymmetric nature of NVM memory accesses. The evaluation of multicore systems presented in Section 3, while integrating non-volatile memory (NVM) technology in the cache hierarchy showed that energy saving is possible thanks to the characteristics of such memory technologies, i.e., a negligible leakage current that favors a significant decrease in static power consumption compared to concurrent SRAM-based solutions. At the same time, we observed that the high latency (and energy consumption) of write operations on NVMs can be detrimental to system performance due to the resulting longer execution time.

Therefore, an opportunity that is currently under consideration in the CONTINUUM project is to exploit adequate code transformations (or optimization) so as to mitigate the negative effect of writes on performance when using NVMs [38]. After some preliminary studies, we have been investigating the so-called *silent store elimination* in programs [4, 5, 49]. Intuitively, a store in a program is said to be silent if it writes a value to a memory address where the same value is already stored. Then, our approach consists in transforming a given source code such that silent stores can be avoided. This enables to reduce the total number of writes. Our silent store code optimization has been implemented in LLVM compiler.

Energy-efficiency optimization via retention time relaxation. Another opportunity that is worth mentioning is to exploit further characteristics of NVMs, particularly the trade-off between their power consumption requirement and their non volatility capacity [4]. Indeed, it is well-known that relaxing the retention time of NVMs through a reduction of the planar area of their cell, contributes to decreasing their write current [47]. Consequently, this reduces their high dynamic energy and write latencies.

Now, let us consider that the multicore system under design includes multiple NVM memory banks, with various non volatility capacity. The intuition here is to allocate data across those memory banks, e.g., according to the liveness of the corresponding variables in a program. For instance, a variable that is alive for a short period during program execution could be mapped to memory banks with low retention time, i.e., requiring a low dynamic energy and shorter write latency. On the opposite, a variable that must be alive for a longer period would be mapped to memory banks with high retention capability such that it could be accessed whenever needed during execution.

Leveraging liveness analysis of variables in programs according to compilation techniques is therefore a relevant opportunity to improve the energy-efficiency in presence of relaxed NVM retention time.

Compiler-assisted Adaptive Code Placement in Heterogeneous Systems. On the other hand, the evaluations reported in Section 3 showed the impact of the heterogeneous nature of cores on

both performance and energy consumption. Typically, big cores such as Cortex-A15 provide high performance at the expense of more dissipated power. LITTLE cores on the other hand are counter-parts of big cores. This trade-off calls for adequate core selection for a given program in order to reach a good compromise in terms of performance and power consumption.

Here, both runtime management and compilation techniques can play an important role. Thanks to the former, one can dynamically assign threads or application in order to reach the expected compromise [8, 10, 25, 12]. Beyond useful system information monitored at runtime (e.g., CPU usage, instruction per cycle, cache miss rate and power consumption), further interesting information resulting from compiler-based static code analysis can be also helpful. For instance, compute-intensive code fragments include many arithmetic operations will be expected to run on big cores while code fragments consisting mainly of read/write of some inputs/outputs will be rather executed on LITTLE cores. On the other hand, the difference between the features of cores (e.g., cache hierarchy, presence of FPU or not...) could be reflected through the applied compilation options according to the target core. Multiversioning is one relevant technique under consideration within the CONTINUUM project. Finally, machine learning techniques could be foreseen in the workload scheduling and mapping loop for efficient execution in heterogeneous systems [24].

Workload allocation for optimized interconnect traffic. Finally, as illustrated in Section 4 on the evaluation of 3D-NoC multicore architecture design, application mapping can benefit from the aforementioned runtime management decisions in order to optimize the communication traffic. Note that the same observation generally holds for 2D NoCs, such as typical mesh networks [41, 54, 31, 20, 19]. Beyond the workload allocation issue itself, it can be interesting to consider interconnects that favors an adaptive data routing inside the network [21] for high throughput. Finally, code optimization techniques could be beneficial to the interconnect activity. Typically, Lepak et al. [32] previously showed that eliminating silent stores helps to reduce multiprocessor bus traffic. We can therefore take advantage of this feature while applying this technique.

6 Concluding remarks

In this deliverable, we described some evaluations where non volatile memory and 3D interconnect technologies are evaluated. On the one hand, energy saving opportunity was shown thanks to the low leakage current of non volatile memories. On the other hand, 3D interconnects provide an alternative design for improving system performance by accelerating the communication traffic. Nevertheless, both technologies present challenging aspects that can significantly reduce these benefits. For instance, the higher cost of write operations in non volatile memories and the process variability issue can affect TSV links in 3D interconnects.

Therefore, in order to mitigate these potential limitations, a number of opportunities have been discussed by advocating both compilation and runtime system management techniques. Some of these techniques are already under investigation within the CONTINUUM project. The preliminary results [49] show promising improvements of energy-efficiency.

References

[1] An, X., Boumedien, S., Gamatié, A., and Rutten, É. CLASSY: a clock analysis system for rapid prototyping of embedded applications on mpsocs. In Corporaal, H. and Stuijk, S., editors, *Workshop on Software and Compilers for Embedded Systems, Map2MPSoC/SCOPES 2012, Sankt Goar, Germany, May 15-16, 2012*, pages 3–12. ACM, 2012. doi: 10.1145/2236576.2236577. URL https://doi.org/10.1145/2236576.2236577.

[2] An, X., Gamatié, A., and Rutten, É. High-level design space exploration for adaptive applications on multiprocessor systems-on-chip. *J. Syst. Archit.*, 61(3-4):172–184, 2015. doi: 10.1016/j.sysarc.2015.02.002. URL https://doi.org/10.1016/j.sysarc.2015.02.002.

[3] Binkert, N., Beckmann, B., Black, G., Reinhardt, S. K., Saidi, A., Basu, A., Hestness, J., Hower, D. R., Krishna, T., Sardashti, S., Sen, R., Sewell, K., Shoaib, M., Vaish, N., Hill, M. D., and Wood, D. A. The gem5 simulator. *SIGARCH Comput. Archit. News*, 39(2):1–7, August 2011. ISSN 0163-5964. doi: 10.1145/2024716.2024718. URL https://doi.org/10.1145/2024716.2024718.

[4] Bouziane, R., Rohou, E., and Gamatié, A. How could compile-time program analysis help leveraging emerging nvm features? In *2017 First International Conference on Embedded Distributed Systems (EDiS)*, pages 1–6, 2017. doi: 10.1109/EDIS.2017.8284031.

[5] Bouziane, R., Rohou, E., and Gamatié, A. LLVM-based silent stores optimization to reduce energy consumption on STT-RAM cache memory. In *European LLVM Developers Meeting (EuroLLVM'17), Saarbrucken, Germany*, 2017.

[6] Butko, A., Garibotti, R., Ost, L., and Sassatelli, G. Accuracy evaluation of gem5 simulator system. In *7th Int'l Workshop on Reconfigurable and Communication-Centric Systems-on-Chip (ReCoSoC)*, pages 1–7, 2012. ISBN 978-1-4673-2570-7. doi: 10.1109/ReCoSoC.2012.6322869.

[7] Butko, A., Gamatié, A., Sassatelli, G., Torres, L., and Robert, M. Design exploration for next generation high-performance manycore on-chip systems: Application to big.little architectures. In *2015 IEEE Computer Society Annual Symposium on VLSI, ISVLSI 2015, Montpellier, France, July 8-10, 2015*, pages 551–556. IEEE Computer Society, 2015. doi: 10.1109/ISVLSI.2015.28. URL https://doi.org/10.1109/ISVLSI.2015.28.

[8] Butko, A., Bessad, L., Novo, D., Bruguier, F., Gamatié, A., Sassatelli, G., Torres, L., and Robert, M. Position Paper: OpenMP scheduling on ARM big.LITTLE architecture. In *MULTIPROG: Programmability and Architectures for Heterogeneous Multicores*, Prague, Czech Republic, January 2016. URL https://hal-lirmm.ccsd.cnrs.fr/lirmm-01377630.

[9] Butko, A., Bruguier, F., Gamatié, A., Sassatelli, G., Novo, D., Torres, L., and Robert, M. Full-system simulation of big.little multicore architecture for performance and energy exploration. In *10th IEEE International Symposium on Embedded Multicore/Many-core Systems-on-Chip, MCSOC 2016, Lyon, France, September 21-23, 2016*, pages 201–208. IEEE Computer Society, 2016. doi: 10.1109/MCSoC.2016.20. URL https://doi.org/10.1109/MCSoC.2016.20.

[10] Butko, A., Bruguier, F., Gamatié, A., and Sassatelli, G. Efficient Programming for Multicore Processor Heterogeneity: OpenMP versus OmpSs. In *OpenSuCo*, Frankfurt, Germany, June 2017. URL https://hal-lirmm.ccsd.cnrs.fr/lirmm-01723762. Held in conjunction with the 2017 ISC High Performance Computing Conference.

[11] Caliri, G. V. Introduction to analytical modeling. In *26th International Computer Measurement Group Conference, December 10-15, 2000, Orlando, FL, USA, Proceedings*, pages 31–36. Computer Measurement Group, 2000. URL http://www.cmg.org/?s2member_file_download=/proceedings/2000/0004.pdf.

[12] Cohen, A. and Rohou, E. Processor Virtualization and Split Compilation for Heterogeneous Multicore Embedded Systems. In *47th Annual Design Automation Conference*, Anaheim, CA, United States, June 2010. URL https://hal.inria.fr/inria-00472274.

[13] Darve, F., Sheibanyrad, A., Vivet, P., and Petrot, F. Physical implementation of an asynchronous 3d-noc router using serial vertical links. In *2011 IEEE Computer Society Annual Symposium on VLSI*, pages 25–30, July 2011.

[14] Dekeyser, J.-L., Gamatié, A., Etien, A., Ben Atitallah, R., and Boulet, P. Using the UML Profile for MARTE to MPSoC Co-Design. In *First International Conference on Embedded Systems & Critical Applications (ICESCA'08)*, Tunis, Tunisia, May 2008. URL https://hal.inria.fr/inria-00524363.

[15] Delobelle, T., Péneau, P., Gamatié, A., Bruguier, F., Senni, S., Sassatelli, G., and Torres, L. MAGPIE: System-level Evaluation of Manycore Systems with Emerging Memory Technologies. In *Workshop on Emerging Memory Solutions - Technology, Manufacturing, Architectures, Design and Test at Design Automation and Test in Europe - DATE'2017, Lausanne, Switzerland*, 2017.

[16] Delobelle, T., Péneau, P.-Y., Senni, S., Bruguier, F., Gamatié, A., Sassatelli, G., and Torres, L. Flot automatique d'évaluation pour l'exploration d'architectures à base de mémoires non volatiles. In *Conférence d'informatique en Parallélisme, Architecture et Système, Compas'16, Lorient, France*, 2016.

[17] Dong, X., Xu, C., Xie, Y., and Jouppi, N. P. Nvsim: A circuit-level performance, energy, and area model for emerging nonvolatile memory. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 31(7):994–1007, 2012. doi: 10.1109/TCAD.2012.2185930.

[18] Effiong, C., Lapotre, V., Gamatié, A., Sassatelli, G., Todri-Sanial, A., and Latif, K. On the performance exploration of 3d nocs with resistive-open tsvs. In *2015 IEEE Computer Society Annual Symposium on VLSI*, pages 579–584, July 2015.

[19] Effiong, C., Sassatelli, G., and Gamatié, A. Scalable and power-efficient implementation of an asynchronous router with buffer sharing. In Kubátová, H., Novotný, M., and Skavhaug, A., editors, *Euromicro Conference on Digital System Design, DSD 2017, Vienna, Austria, August 30 - Sept. 1, 2017*, pages 171–178. IEEE Computer Society, 2017. doi: 10.1109/DSD.2017.55. URL https://doi.org/10.1109/DSD.2017.55.

[20] Effiong, C., Sassatelli, G., and Gamatié, A. Roundabout: A network-on-chip router with adaptive buffer sharing. In *15th IEEE International New Circuits and Systems Conference, NEWCAS 2017, Strasbourg, France, June 25-28, 2017*, pages 65–68. IEEE, 2017. doi: 10.1109/NEWCAS.2017.8010106. URL https://doi.org/10.1109/NEWCAS.2017.8010106.

[21] Effiong, C., Sassatelli, G., and Gamatié, A. Distributed and dynamic shared-buffer router for high-performance interconnect. In Jantsch, A., Matsutani, H., Lu, Z., and Ogras, Ü. Y., editors, *Proceedings of the Eleventh IEEE/ACM International Symposium on Networks-on-Chip, NOCS 2017, Seoul, Republic of Korea, October 19 - 20, 2017*, pages 2:1–2:8. ACM, 2017. doi: 10.1145/3130218.3130223. URL https://doi.org/10.1145/3130218.3130223.

[22] Feero, B. S. and Pande, P. P. Networks-on-chip in a three-dimensional environment: A performance evaluation. *IEEE Transactions on Computers*, 58(1):32–45, Jan 2009.

[23] Gamatié, A., Beux, S. L., Piel, É., Atitallah, R. B., Etien, A., Marquet, P., and Dekeyser, J. A model-driven design framework for massively parallel embedded systems. *ACM Trans. Embed. Comput. Syst.*, 10(4):39:1–39:36, 2011. doi: 10.1145/2043662.2043663. URL https://doi.org/10.1145/2043662.2043663.

[24] Gamatié, A., Ursu, R., Selva, M., and Sassatelli, G. Performance prediction of application mapping in manycore systems with artificial neural networks. In *10th IEEE International Symposium on Embedded Multicore/Many-core Systems-on-Chip, MCSOC 2016, Lyon, France, September 21-23, 2016*, pages 185–192. IEEE Computer Society, 2016. doi: 10.1109/MCSoC.2016.17. URL https://doi.org/10.1109/MCSoC.2016.17.

[25] Garibotti, R., Butko, A., Ost, L., Gamatié, A., Sassatelli, G., and Adeniyi-Jones, C. Efficient embedded software migration towards clusterized distributed-memory architectures. *IEEE Transactions on Computers*, 65(8):2645–2651, 2016. doi: 10.1109/TC.2015.2485202.

[26] Hsieh, A. C. and Hwang, T. Tsv redundancy: Architecture and design issues in 3-d ic. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 20(4):711–722, April 2012. ISSN 1063-8210.

[27] Kim, D. H., Athikulwongse, K., and Lim, S. K. A study of through-silicon-via impact on the 3d stacked ic layout. In *2009 IEEE/ACM International Conference on Computer-Aided Design - Digest of Technical Papers*, pages 674–680, Nov 2009.

[28] Kim, D. H., Mukhopadhyay, S., and Lim, S. K. Through-silicon-via aware interconnect prediction and optimization for 3d stacked ics. In *Proceedings of the 11th International Workshop on System Level Interconnect Prediction*, SLIP '09, pages 85–92, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-576-5.

[29] Kologeski, A., Kastensmidt, F. L., Lapotre, V., Gamatié, A., Sassatelli, G., and Todri-Sanial, A. Performance exploration of partially connected 3d nocs under manufacturing variability. In *2014 IEEE 12th International New Circuits and Systems Conference (NEWCAS)*, pages 61–64, 2014. doi: 10.1109/NEWCAS.2014.6933985.

[30] Latif, K., Effiong, C. E., Gamatié, A., Sassatelli, G., Zordan, L. B., Ost, L., Dziurzanski, P., and Soares Indrusiak, L. An Integrated Framework for Model-Based Design and Analysis of Automotive Multi-Core Systems. In *FDL: Forum on specification & Design Languages*, Work-in-Progress Session, Barcelona, Spain, September 2015. URL https://hal-lirmm.ccsd.cnrs.fr/lirmm-01418748.

[31] Latif, K., Selva, M., Effiong, C., Ursu, R., Gamatié, A., Sassatelli, G., Zordan, L., Ost, L., Dziurzanski, P., and Indrusiak, L. S. Design space exploration for complex automotive applications: an engine control system case study. In *Proceedings of the 2016 Workshop on Rapid Simulation and Performance Evaluation - Methods and Tools, RAPIDO@HiPEAC 2016, Prague, Czech Republic, January 18, 2016*, pages 2:1–2:7. ACM, 2016. doi: 10.1145/2852339.2852341. URL https://doi.org/10.1145/2852339.2852341.

[32] Lepak, K. M., Bell, G. B., and Lipasti, M. H. Silent stores and store value locality. *IEEE Transactions on Computers*, 50(11):1174–1190, 2001.

[33] Li, S., Ahn, J. H., Strong, R. D., Brockman, J. B., Tullsen, D. M., and Jouppi, N. P. Mcpat: An integrated power, area, and timing modeling framework for multicore and manycore architectures. In *Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture*, MICRO 42, page 469–480, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605587981. doi: 10.1145/1669112.1669172. URL https://doi.org/10.1145/1669112.1669172.

[34] Metzler, C., Todri, A., Bosio, A., Dilillo, L., Girard, P., and Virazel, A. Through-silicon-via resistive-open defect analysis. In *2012 17th IEEE European Test Symposium (ETS)*, pages 1–1, May 2012.

[35] Mittal, S., Vetter, J. S., and Li, D. A survey of architectural approaches for managing embedded dram and non-volatile on-chip caches. *IEEE TPDS*, 26(6):1524 – 1537, June 2015.

[36] Panades, I. M. and Greiner, A. Bi-synchronous fifo for synchronous circuit communication well suited for network-on-chip in gals architectures. In *First International Symposium on Networks-on-Chip (NOCS'07)*, pages 83–94, May 2007.

[37] Péneau, P., Novo, D., Bruguier, F., Sassatelli, G., and Gamatié, A. Performance and energy assessment of last-level cache replacement policies. In *2017 First International Conference on Embedded Distributed Systems (EDiS)*, pages 1–6, 2017. doi: 10.1109/EDIS.2017.8284032.

[38] Péneau, P., Bouziane, R., Gamatié, A., Rohou, E., Bruguier, F., Sassatelli, G., Torres, L., and Senni, S. Loop optimization in presence of STT-MRAM caches: A study of performance-energy tradeoffs. In *26th International Workshop on Power and Timing Modeling, Optimization and Simulation, PATMOS 2016, Bremen, Germany, September 21-23, 2016*, pages 162–169. IEEE, 2016. doi: 10.1109/PATMOS.2016.7833682. URL https://doi.org/10.1109/PATMOS.2016.7833682.

[39] Quadri, I. R., Gamatié, A., Boulet, P., and Dekeyser, J.-L. Modeling of Configurations for Embedded System Implementations in MARTE. In *1st workshop on Model Based Engineering*

for Embedded Systems Design - Design, Automation and Test in Europe (DATE 2010), Dresden, Germany, March 2010. URL https://hal.inria.fr/inria-00486845.

[40] Schirner, G. and Dömer, R. Quantitative analysis of the speed/accuracy trade-off in transaction level modeling. *ACM Trans. Embed. Comput. Syst.*, 8(1), January 2009. ISSN 1539-9087. doi: 10.1145/1457246.1457250. URL https://doi.org/10.1145/1457246.1457250.

[41] Schonwald, T., Zimmermann, J., Bringmann, O., and Rosenstiel, W. Fully adaptive fault-tolerant routing algorithm for network-on-chip architectures. In *10th Euromicro Conference on Digital System Design Architectures, Methods and Tools (DSD 2007)*, pages 527–534, 2007. doi: 10.1109/DSD.2007.4341518.

[42] Senni, S., Brum, R. M., Torres, L., Sassatelli, G., Gamatié, A., and Mussard, B. Potential applications based on nvm emerging technologies. In *2015 Design, Automation Test in Europe Conference Exhibition (DATE)*, pages 1012–1017, 2015. doi: 10.7873/DATE.2015.1120.

[43] Senni, S., Torres, L., Sassatelli, G., Gamatié, A., and Mussard, B. Emerging non-volatile memory technologies exploration flow for processor architecture. In *2015 IEEE Computer Society Annual Symposium on VLSI*, pages 460–460, 2015. doi: 10.1109/ISVLSI.2015.126.

[44] Senni, S., Torres, L., Sassatelli, G., Gamatié, A., and Mussard, B. Exploring MRAM technologies for energy efficient Systems-on-Chip. *IEEE JETCAS*, 6(3):279–292, 2016.

[45] Senni, S., Delobelle, T., Coi, O., Péneau, P., Torres, L., Gamatié, A., Benoit, P., and Sassatelli, G. Embedded systems to high performance computing using STT-MRAM. In Atienza, D. and Natale, G. D., editors, *Design, Automation & Test in Europe Conference & Exhibition, DATE 2017, Lausanne, Switzerland, March 27-31, 2017*, pages 536–541. IEEE, 2017. doi: 10.23919/DATE.2017.7927046. URL https://doi.org/10.23919/DATE.2017.7927046.

[46] Singh, A. K., Shafique, M., Kumar, A., and Henkel, J. Mapping on multi/many-core systems: Survey of current and emerging trends. In *Proceedings of the 50th Annual Design Automation Conference*, DAC '13, pages 1:1–1:10. ACM, 2013. ISBN 978-1-4503-2071-9.

[47] Smullen, C. W., Mohan, V., Nigam, A., Gurumurthi, S., and Stan, M. R. Relaxing non-volatility for fast and energy-efficient stt-ram caches. In *Proceedings of the 2011 IEEE 17th International Symposium on High Performance Computer Architecture*, HPCA '11, pages 50–61, Washington, DC, USA, 2011. IEEE Computer Society. ISBN 978-1-4244-9432-3. URL http://dl.acm.org/citation.cfm?id=2014698.2014895.

[48] The CONTINUUM Project Consortium. Survey on emerging memory and communication technologies. Technical Report Deliverable D3.1, June 2016.

[49] The CONTINUUM Project Consortium. Description of specific optimizations for low-power. Technical Report Deliverable D2.2, April 2017.

[50] Topol, A. W., Tulipe, D. C. L., Shi, L., Frank, D. J., Bernstein, K., Steen, S. E., Kumar, A., Singco, G. U., Young, A. M., Guarini, K. W., and Ieong, M. Three-dimensional integrated circuits. *IBM Journal of Research and Development*, 50(4.5):491–506, July 2006.

[51] Tran, A. T. and Baas, B. Noctweak: a highly parameterizable simulator for early exploration of performance and energy of networks on-chip. Technical Report ECE-VCL-2012-2, VLSI Computation Lab, ECE Department, University of California, Davis, July 2012. http://www.ece.ucdavis.edu/vcl/pubs/2012.07.techreport.noctweak/.

[52] Umemoto, M., Tanida, K., Nemoto, Y., Hoshino, M., Kojima, K., Shirai, Y., and Takahashi, K. High-performance vertical interconnection for high-density 3d chip stacking package. In *Electronic Components and Technology Conference, 2004. Proceedings. 54th*, volume 1, pages 616–623 Vol.1, June 2004.

[53] Ye, F. and Chakrabarty, K. Tsv open defects in 3d integrated circuits: Characterization, test, and optimal spare allocation. In *Proceedings of the 49th Annual Design Automation Conference*, DAC '12, pages 1024–1030. ACM, 2012. ISBN 978-1-4503-1199-1.

[54] Yongfeng Xu, Jianyang Zhou, and Shunkui Liu. Research and analysis of routing algorithms for noc. In *2011 3rd International Conference on Computer Research and Development*, volume 2, pages 98–102, 2011. doi: 10.1109/ICCRD.2011.5764092.