AN ENERGY-EFFICIENT HYBRID UNCORE ARCHITECTURE IN FUTURE EMBEDDED CHIP-MULTIPROCESSOR

by

Akram Hadeed
Master of Science, Electronic Engineering , University of Technology, Baghdad, Iraq,, 2000
Bachelor of Science, Electrical Engineering, Baghdad University,1997

A Thesis

presented to Ryerson University

in partial fulfillment of the requirements for the degree of Master of Applied Science in

The program of Electrical and Computer Engineering

Toronto, Ontario, Canada, 2020

AUTHOR’S DECLARATION FOR ELECTRONIC SUBMISSION OF A THESIS

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

I understand that my thesis may be made electronically available to the public.

Abstract

An Energy-Efficient Hybrid Uncore Architecture in Future Embedded Chip-Multiprocessor

©Akram Hadeed, 2020

Master of Applied Science

Electrical and Computer Engineering

Ryerson University

Recently, technology scaling has enabled the placement of an increasing number of cores, in the form of chip-multiprocessors (CMPs) on a chip and continually shrinking transistor sizes to improve performance. In this context, power consumption has become the main constraint in designing CMPs. As a result, uncore components power consumption taking increasing portion from the on-chip power budget; therefore, designing power management techniques, particularly memory and network-on-chip (NoC) systems, has become an important issue to solve. Consequently, a considerable attention has been directed toward power management based on CMPs components, particularly shared caches and uncore interconnected structures, to overcome the challenges of limited chip power budget.

This work targets to design an energy-efficient uncore architecture by using heterogeneity in components (cache cells) and operational parameters (Voltage/Frequency). In order to ensure the minimum impact on the system performance, a run-time approach is investigated to assess the proposed method. An architecture is proposed where the cache layer contains the heterogenous cache banks in all placed in one frequency voltage domain. Average memory access time (AMAT) was selected as a network monitor to monitor the performance on the run-time. The appropriate size and type of the last level cache (LLC) and Voltage/Frequency for the uncore domain is adjusted according to the calculated AMAT which indicates the system demand from the uncore.

The proposed hybrid architecture was implemented, investigated and compared with the a baseline model where only SRAM banks were used in the last level cache. Experimental results on the Princeton Application Repository for Shared-Memory Computers (PARSEC) benchmark suit, show that the proposed architecture yields up to a 40% reduction in overall chip energy-delay product with a marginal performance degradation in average of -1.2% below the baseline one. The best energy saving was 55% and the worse degradation was only 15%.

Acknowledgment

First, I would like to gratefully and sincerely thank my advisor, Dr. F. Mohammadi, for her kindly instructions and help in this research. This work could not have been finished without her guidance, support, and time. Her belief in me throughout the difficult times during my research and my graduate studies is what keeps me motivated to this day.

I would like to thank my friends and all the members of the Ryerson Microsystems research group for their constructive guidance and suggestions as well as their consistent encouragement.

Finally, my deep and sincere gratitude to my family for their continuous and unparalleled love, help and support.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

3D	Three-dimensional
AMAT	Average memory access time
CMOS	Complementary metal oxide semiconductor
CMP	Chip-multiprocessor
CPU	Central processing unit
DPM	Dynamic power management
DRAM	Dynamic random-access memory
DVFS	Dynamic voltage and frequency scaling
EWT	Early write termination
FB	Filter buffer
GPU	Graphics processing unit
HPC	High-performance computing
IC	Integrated circuit
IPC	Instructions per cycle
ITRS	Technology roadmap for semiconductors
LLC	Last level cache
NoC	Network-on-chip
NVM	Non-volatile memory
OAM	Obstruction-aware monitoring

OAP	Obstruction-aware cache management policy
RD	Memory read
RHC	Reconfigurable hybrid cache
SRAM	Static random-access memory
RERAM	Resistive random-access memory
PCRAM	Phase-change memory
STT-RAM	Spin-transfer torque magnetic random-access memory
VFI	Voltage frequency island
WR	Memory write
PARSEC	Princeton Application Repository for Shared-Memo computers

Chapter 1

Introduction

1.1 Motivation

The development of computing systems, data analytics, and tools for big data computing has resulted in an increasing need for low-power computational platforms and high-performance efficiency capable of adjusting the processing capability and storage domains. Furthermore, the high integration density of modern chip-multiprocessors (CMPs), such as shrinking feature sizes and the increasing number of transistors packed into a single chip, resulted in serious design challenges that have surfaced, including high-power densities and problems related to temperature. These problems, in turn, may accelerate chip failure, thereby degrading the performance of the entire system [9]. Moreover, uncore components such as the cache memory contain a large number of transistor elements and consume a significant amount of power. Therefore, multicores, particularly CMPs, allow the reconfiguration of the Last Level Caches (LLCs) on the basis of the target applications [10].

As the number of cores on the chip increases, the demands for on-chip cache dramatically increases especially for LLC. In many-core systems a considerable portion of chip area is allocated to on-chip caches resulting in a major contribution in the overall chip power consumption. Increasing core counts in CMP designs, results from an increase in the shared resources including interconnect and cache hierarchy, these resources of core processing area will be referred as uncore area. The latest process technology brought Dennard scaling to its end in 2005, starting the dark silicon era. This phenomenon states that every new chip generation will have a limited power budget which cannot be exceeded [11]. Thus, the fraction of transistors that can operate at the full frequency is decreasing with each new technology generation. International Technology Roadmap for Semiconductors (ITRS) projected that 21% and 50% of the chip will be off (dark) at the technologies 22nm and 8nm, respectively [12]. Thus, researches and the industry focused on increasing the number of cores per chip to:

- Improve the on-core parallelism,
- Reduce off core communication and consequently enhancing the performance,
- Give better on-chip power and thermal management.

This work targets to design an energy-efficient uncore architecture by using heterogeneity in components (cache cells) and operational parameters (Voltage/Frequency). A run-time approached is investigated to ensure the minimum impact on the system performance. We propose average memory access time (AMAT) as the selected network monitor in our run- time. The appropriate size and type of the last level cache (LLC) and Voltage/Frequency for the uncore domain will be adjusted according to the calculated AMAT.

1.2 Objective

CMPs are widely utilized in a range of applications pertaining to big data analytics and computing systems with high performance. However, the design of CMPs is somewhat curtailed by the power consumption and temperature constraints. In this respect, memory and cache sub-systems have emerged as promising components for the creation of scalable high-performance and energy-efficient platforms. Through the dynamic reconfiguration of memory components, it is possible to improve the power consumption within the given performance constraints. Thus, the main objectives of this work are:

- To design an energy-efficient uncore architecture by using heterogeneity in components (cache cells) and operational parameters (Voltage/Frequency). A run-time approached is investigated to ensure the minimum impact on the system performance.
- To propose average memory access time (AMAT) as the selected network monitor in the runtime. The appropriate size and type of the LLC and Voltage/Frequency for the uncore domain will be adjusted according to the calculated AMAT.

1.3 Contributions

The research work has the following contributions:

1. Optimize and manage the power of the cache memory systems based on iterative process under performance constraints is the main optimization goal of the proposed model at the CMP level.
2. Investigate the efficiency of the proposed hybrid design by using different real-world architectures and comparing it with the baseline model.
3. Evaluate the power consumption and performance of the proposed methods under multi-threaded workloads, as different works use various benchmarks depending on the applications.

1.4 Thesis Organization

The remainder of the thesis is organized as follows:

In chapter 2, the most recent architectures and techniques used in hybrid uncore architecture based on energy consumption were discussed and reviewed.

The energy-efficient CMP with reduced off-chip memory access discussed in chapter 3. In addition, core components of the architecture and the steps required to increase energy efficiency is described.

In chapter 4, the proposed architecture was discussed, implemented and investigated using the Princeton Application Repository for Shared-Memory Computers, or simply the PARSEC benchmark.

The contribution of this thesis is summarized in chapter 5 . In addition, future research topics are suggested. In the end, the references used throughout this thesis are listed.

Chapter 2

Background and Literature Review

2.1 Introduction

The increasing gap between processor speed and main memory lag has driven demand for large on-chip cache memory. For this reason, the cache hierarchy is deeper and with a larger capacity of cache memory is required in the design of today's processors. For example, the Intel i7-3930k processor is equipped with 12MB of SRAM 3rd level cache memory, or the IBM POWER7 processor has the 3rd-level cache of 32MB in size and DRAM Embedded (eDRAM) [3]. However, in modern processors, large on-chip cache stores take up a large fraction of the total chip size, resulting a significant portion from their total energy consumption. The most commonly used cache memory is due to the rapid availability of SRAMs. However, the continuous decrease in the technology size of the transistors results in an increase in leakage power consumption, making SRAM's in-chip memory inefficient to design energy-efficient bottoms. Additionally, limited scalability and soft error sensitivity hampered SRAM technology in the implementation of high-density intrusive cache memory [55].

On the other hand, in order to overcome the increased delay caused by universal connections in two-dimensional technology with the process of reducing the dimensions of transistors, the integration of three layers of cache hierarchy layers on layers processing is used. However, due to the increased power density and temperature in 3D integration, the processing efficiency will be affected. For example, if the chip temperature exceeds the permissible temperature range, then the frequency and voltage of the core or processor cores in the processing layer should be lowered or cut, which also leads to a reduction in the efficiency of the three-dimensional processor system. Therefore, in order to achieve the best solution for increasing the efficiency of multiprocessor systems, it is necessary to improve the efficiency compromise between these two procedures.

The use of the emerging technologies of non-volatile memory due to zero-leakage power consumption and non-volatility, and the high number of cells in this memory class per unit area compared with traditional SRAMs, is a promising solution for solving the high-temperature problem is due to the integration of three-dimensional memory layers on the processing layers in this technology for future generation multiprocessor systems. As already mentioned, leakage power leads to higher temperatures and higher temperatures, leading to an increase in leakage power, and also if the leakage power consumption is not controlled, the temperature parameter will become an inhibitory parameter.

Increasing the density of transistors on the chip and reducing the size of the transistors' good choice to obtain greater density also led to an increase in leakage power. However, the performance is not exponentially scaled because it is restricted by the scaling speed mismatch of power consumption and memory bandwidth. Furthermore, in nowadays computing systems, energy efficiency becomes the primary concern during system design. The traditional scale-out strategy by packing more cores into a single chip is no longer power sustainable.

Therefore, the use of non-volatile memory with the near zero leakage power consumption as layers of LLC memory on processing layers in 3D architecture, in particular due to increased power density and temperatures, will become the biggest challenge in the next-generation technologies, it will be a very promising and rehabilitative solution.

The next sections will discuss the main technologies used to reduced leakage power consumption in CMPs.

2.2 Heterogeneous CPU–GPU System

Increasingly application complexity demands progressively powerful computing systems for high-performance computing (HPC). Such systems will provide multicore couplings such as a central processing unit (CPU) and graphics processing unit (GPUs) integration systems. These integrations provide an opportunity to accelerate many applications with high efficiency in bandwidth throughput. Moreover, owing to the increasing computing capabilities, the connections between the GPUs and the CPUs are carried out by using off-chip interconnects that provide considerable traffic latency and high-power consumption [13-15]. A heterogeneous CMP that integrates CPUs and GPUs on the same chip with the sharing memory can solve this problem, avoid such expensive off-chip data transfers, and lead to improved system performance [16]. Moreover, the interconnection in a single chip should be more efficient and scalable to improve system performance. The 3D integrated circuit (IC) architecture improved communication efficiency.

One possible solution to this challenge, is using a three-dimensional (3D) integrated circuit (IC) architecture for improving the communication efficiency. 3D ICs enhanced the performance of CMPs by focusing on the advantage of high density and low interconnect latency [17-19]. Particularly in terms of two factors: accelerating the throughput and utilizing the power by using the advantages of 3D ICs for heterogeneous CPU–GPU systems [20].

Early researchers have worked on improving the system performance of discrete GPUs by improving their network-on-chip (NoC) design. The GPUs can process many parallel streams of data in their cores independently (i.e., each core has its own task), leading to significant core-to-core communication. The increasing number of computing cores in the GPU system has resulted in an increasingly many-to-few traffic pattern or many cores sending traffic to a few memory controller nodes. This has led to a performance bottleneck. However, the works presented in [21–23] showed that the distribution of these memory controllers at appropriate positions can improve the associated traffic.

In [21], a GPU with a 2D mesh topology was proposed. The controller nodes were placed at the bottom and the top of this configuration. A checkerboard NoC organization was used on the basis of half-routers with limited connectivity to reduce the NoC area while ensuring minimal impact on the performance. While in [22], the researchers proposed the asymmetric virtual channel partitioning of NoCs into two parts for the different traffic routes: one network carries the request packets, and the other network, the reply packets. Therefore, the full monopolization technique is used for either the request or the reply packets to improve performance by providing more resources for each type of traffic. The researchers in [23] presented an asymmetric NoC mesh architecture used for a memory traffic pattern. This pattern provides a connection from L1 to L2 traffic and a second pattern from L2 to L1 traffic for parallel applications.

2.3 Dynamic Power Management for CMP

There are two factors that affect the power consumption of complementary metal-oxide-semiconductor (CMOS) circuits: static and dynamic power consumption. One is the low static power consumption, which is attributed to the leakage current and the dynamic power consumption (i.e. switching power), which in turn depends on the charging and discharging of the capacitor. Dynamic power consumption can be minimized by reducing the activities and scaling the volt-age-frequency level. Furthermore, leakage power consumption can be reduced by utilizing low-power cells or reducing the number of active transistors [24]. In geometries smaller than 65 nm, static power has become the dominant consumer of power consumption. The objective of power management is to maximize the workload performance of the CMP without exceeding the given total power budget for the chip. By applying learning-based power management, optimizing idle periods on the CMP to achieve a better tradeoff between power consumption and performance can be used. As a result, the chip will consume more power, leading to the generation of a large amount of heat, which can affect chip performance. Therefore, power consumption issues are occasionally more important than processor speed. For example, Figure 2.1 shows the AMD Opteron X4 processor action against power consumption. In this figure, the chip still consumes approximately 60% of its maximum power consumption through the idle period, although the CPU utilization approaches 0% [25].

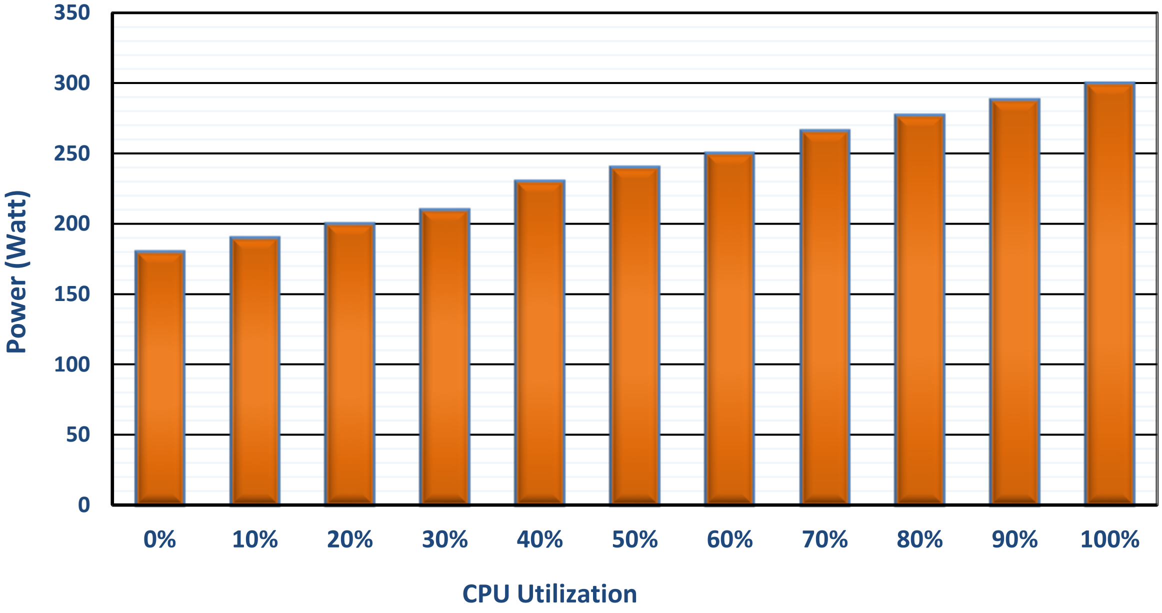


Figure 2.1: Power consumption issues for AMD Opteron X4 processors

In general, dynamic power management (DPM) can minimize the power dissipation by selectively shutting down the inactive system components, although it is very difficult to choose the opportune moment to turn off the inactive components without knowing the actual workloads of the system in advance. The main challenge in DPM optimization methods is the utilization and prediction of the idle times of CMP to achieve high power consumption with less performance degradation. This power consumption occurs when power is used even when the signals do not change, and the system is the idle state. Numerous studies have been published in this domain [26-28]. Several DPM techniques were used to estimate the workloads and adjust the DPM policy. The efficiency of these techniques depends on the accuracy of their workload models, which are not guaranteed during the optimization sequence. Moreover, at the same workloads, they produce significantly different idle states on each core with various task allocation strategies and hence have a considerable impact on the efficiency of the DPM policy.

Thermal challenges are also becoming an important factor in determining the cost of operating an HPC system. A complexity that arises in the thermal management of large-scale systems is the effect of physical properties (e.g., physical location and access to cooling) on the thermal behavior. Furthermore, modern supercomputers consume an enormous amount of power, where a significant fraction is dedicated to cooling [29]. Tianhe-2, the number 1 system in the top-500 list, consumes up to 17 MW with approximately an additional 7 MW for cooling. As one of the DMP solutions that can be worked against the continually expanding processor count in CMP, the voltage–frequency island (VFI) emerged as an effective power management strategy.

Naturally, because of the ability to dynamically tune the voltage–frequency levels of VFIs to minimize the power consumption with minimal performance degradation, the dynamic voltage and frequency scaling (DVFS) technique can be adopted as an upgraded version of the VFI method. The DVFS technique is widely used as one of the possible solutions for DPM in CMP because it is used to dynamically adjust the voltage-frequency ratio based on the workload. This technique can be executed at different levels of hierarchy: per-core DVFS level [30–32], per-cluster DVFS level [33, 34], and per-chip DVFS level [35, 36]. In [30], researchers presented an offline per-core DVFS algorithm. This per-core DVFS was enabled by on-chip regulators. While in [22], the researchers designed a DVFS algorithm based on the correlation between two parameters: the critical speed and the cache access rate to select suitable values for the frequency-voltage ratio at runtime on the android system. The results showed a slow-down in the execution of the programs because the critical speed-based DVFS algorithm preferred less power consumption to maintain its performance. In [32], the researchers applied an online control techniques based on Lagrange optimization and the calculus of variations to find the required frequency-voltage ratio of both the processing elements (PEs) and the routers in the CMP. For the per-cluster DVFS level, in [33, 34], the researchers proposed a clustered DVFS approach, which selects and implements DVFS in each cluster zone.

On the basis of previous research, major restrictions of using DVFS technique can be summarized as:

- Selecting the DVFS level: at the per-core DVFS level, higher cost and less scalability are obtained when the technique is applied to a processor with a large number of cores. Moreover, at the per-chip DVFS level, selection leads to a neglect of the individual requirements per core. Furthermore, at the cluster level, DVFS obtains uncertain results with an increasing number of clusters.
- A considerable latency of the state transition from one power state to another because the voltage variation takes a while to complete.

Another possible solution to the power optimization challenge is task mapping, where the scheduling ensures that specific cores are selected on the chip for mapping in terms of both the timing behavior and the memory requirements [36]. This technique can be executed in several ways as described in [37-39]. For example, in [37], researchers suggested using instruction per cycle (IPC) to perform the task mapping to cores. Using prediction in the CMP performance parameters to map the task for each core [38]. In [39], the integration of task scheduling and cache partitioning for a single task was discussed. However, all these techniques provide efficient mapping solutions for small-scale systems within an acceptable time and may provide an acceptable result, but in the case of the large-scale and 3D IC architectures, they pose a big challenge to the management of the resources at run-time in a scalable manner.

Moreover, recent researchers have focused on a hybrid multi-level memory cache to replace a single-level cache as a new view for multi-core future processor configuration. For example, the authors of [40] showed that the existing architectures led to a gain of approximately 33% of the total power consumption. This concept has received considerable attention to reduce power consumption and to overcome the high-temperature problems. Other related works have been reported the use of the power of hybrid cache designs. In [4], a hybrid cache with access-aware policies and partition-level policies, a wear-leveling scheme to manage power consumption is proposed. In [41], the researchers focused on the hybrid memory hierarchy, which uses a reconfiguration approach to improve system performance by using a statistical prediction approach. In [42], a hybrid SRAM, STT-RAM, L2 cache is proposed that assigns a writes counter to each block of the cache. Other approaches [31-34] either use a specific threshold to distinguish parts of the cache or require a high hardware overhead to predict the cache behavior.

A 3D cache hierarchy involved with the CMP heterogeneous design is proposed in [43-46] for improving performance and minimizing power consumption. Furthermore, [43] deals with creating a time framework for heterogeneous CMPs. In [44], the researchers proposed an approach to select the best core subset of an application within a specific power capacity for a specific dark silicon area to improve performance. In addition, in [45] and [46], the researchers proposed a synthesized homogeneous core approach to increase the performance engaged with the power constraint for architectural synthesis. These prior studies on the phenomena of dark silicon mainly focused on cores instead of uncore components.

2.4 Use Non-Volatile Memory (NVM)

Non-volatile memories (NVM) is promising feature technology has been investigated as replacement to the power consumer volatile technology memories. The are several NVMs were researched such as phase change random access memory (PCRAM), resistive random access memory (RRAM), magnetic random-access memory (MRAM) and spin-transfer torque random-access memory (STT-RAM). The latest one was the most reasonable candidate for its close characteristics to Static random-access memory SRAM in term of reading interval and energy comparing with other NVMs. STT-RAM technology suffers from long latency and high power consumption during writing operation, for that, the direct replacement of this technology to the SRAM in the memory structure can lead to performance degradation. For this reason, many studies have been done to address these two problems, so that they can benefit from the most important features of NVM without sacrificing system performance. In the following sections description of these methods will be reviewed.

2.4.1. Spin-Transfer Torque Random-Access Memory

One of the most promising ones of this NVM is spin-transfer torque RAM (STT-RAM) for its unique features [6]. It behaves like the SRAM in a read operation for the latency and energy but with near zero leakage power. Table 2.1 [11] shows comparison between the conventional SRAM and STT-RAM at 32 nm in terms of latency, density, power and energy.

Table 2.1. Comparison of different memory technologies at 32nm [11]

Technology	**Area**	**Read Latency**	**Write Latency**	**Leakage Power at 80 ℃**	**Read Energy**	**Write Energy**
1MB SRAM	$3.03\ mm^2$	$0.702\ ns$	$0.702\ ns$	$444.6\ mW$	$0.168\ nJ$	$0.168\ nJ$
4MB STT-RAM	$3.39\ mm^2$	$0.880\ ns$	$10.67\ ns$	$190.5\ mW$	$0.278\ nJ$	$0.765\ nJ$

2.4.2. Reducing the Number of Writing Operations

Reducing the number of writing operations in the hybrid cache bank will reduce the power consumption, this will also effectively enhance the lifespan of memory cells. Normally, in accessing memory, the write operation of all single-row cells is updated, while the identify the bulk of write operations is extra. Reading operations before writing can help identify such extra bits and eliminate write-write operations to save energy and to reduce system performance degradation. At the end, the read operation is again performed to ensure that the stored data is correct [47]. This technique is called Early Write Termination (EWT). In [56], a significantly reduce in energy is achieved for writing operations without losing performance parameters was presented. Evaluations for a two-level cache of 16 MB show that approximately 80% of write energy was reduced, and approximately 33% less in total energy consumption is reduced.

Further reduction in the number of write operations in NVM cells proposed in [48]. When a new data in a block of cache is supposed to be written, the old data is first to read and then the Hamming interval is calculated between the two data. The hamming distance used to compare two strings (binary) of equal length, it represents the number of bit positions in which the two strings are different, more details in [49]. If the Hamming distance is larger than half the size of the memory block, it is logical that the data reverse is stored in order to save fewer bits. To do this, the reversal of the new data is stored along with a status bit with the value 1 indicating that the data is inversely proportional. When reading, the data is either reversed or given to the requestor in the same manner according to the status bit [47].

2.4.3. Hybrid Memory Architecture Based Methods

It used to emerge memory along with the conventional SRAM and DRAM memory through benefits of both groups in: fast as writing and high-density memory, low leakage in at the same time, the software used to solve the problem of NVM of soft memory. In this kind of architecture, instead of building NVM-based entirely hidden caches, a part of the NVM cells can be replaced with SRAM and DRAM elements.

In [50], a structure with the ability to retry for hybrid memory is proposed, as shown in Figure 2.2(a). Zhao et. al. studied diverse memory technologies from the point of view of: (1) latency of reading and writing access, (2) dynamic energy and bandwidth, and (3) energy consumption constraints. They simulated bandwidth-based versus capacity graph for a variety of memory technologies SRAM, STT-RAM and DRAM were used in 1, 2, and 3, respectively, Figure 2.2 (b).

The main goal of their study was to improve the cache memory bandwidth by adapting dynamic memory hierarchy with the bandwidth requirements of various applications. As can be seen, only one memory bandwidth may be offered at any interval of memory capacity. It shows that, there is no memory technology that offers more bandwidth for the entire capacity range. Therefore, this technique uses a combination of memory technologies to provide more bandwidth for each level of hidden memory in a given capacity range. The results obtained from simulations show an improvement of 58% and 14% efficiency for multi-threaded and multi-program applications.

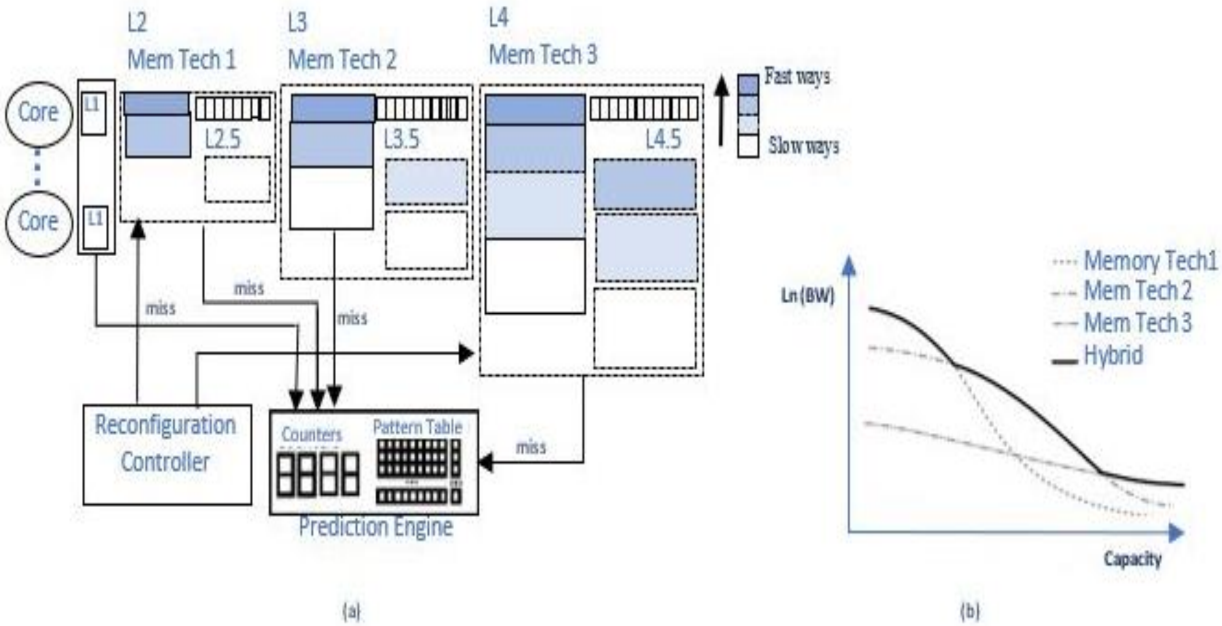


Figure 2.2: Hardware configuration. (a) configuration of reconfigurable hybrid cache hierarchy. (b) overall bandwidth-capacity curve of the hybrid cache hierarchy (3 memory technologies) [50].

Wang et. al. [3] proposed an obstruction-aware cache management policy (OAP) to monitor and detect the LLC-obstruction process periodically. The detection used to manage the cache accesses from different processes. They considered only the last level cache memory (level 3) from STT-RAM and proposed that: The expected access time is reached on shared memory, in which the read/write ratio (P_{Rd}, P_{Miss}) the cache latency rate, T_{Rd}, and T_{Wr} delayed reading and write the LLC cache and T_{Mem} mean the main memory access latency. On the other hand, if the hierarchy of cache memory is removed, the expected access time to memory is according to:

$$T = P_{Rd} \times T_{Rd} + (1 - R_{Rd} \times T_{Wr} + P_{Miss} \times (T_{Mem} + T_{Wr}) \; \acute{T} = T_{Mem} \qquad (2.1)$$

Where P_{Rd} is the read/write ratio, P_{Miss} the cache latency rate, T_{Rd}, T_{Wr} delayed reading and write the hidden cache, and T_{Mem} is the main memory access latency.

If the write-up of a process in the hidden cache does not result in system performance improvement, or $\acute{T} > T_{Mem}$, then the relation 2.2 is established.

$$P_{Miss} > \frac{T_{Mem} - P_{Rd} \times T_{Rd} - (1 - R_{Rd}) \times T_{Wr}}{T_{Mem} + T_{Wr}} \qquad (2.2)$$

When the specification of a process acknowledges the relationship 2.2 at run time, it may lead to bulky write operations in the LLC cache, i.e. increase run-time, and degrade system performance. This kind of process is referred to as the LLC-obstruction processes. In this technique, in order to identify the LLC obstruction, the obstruction-aware monitoring (OAM) hardware is inserted before the LLC memory as shown in Figure 2.3. Each interval, when adjustable in the hardware, consists of two parts:

OAM works on tunable parameter interval which each are divided into two ;First part where cache works under the normal policy and OAMs collects information and all processes are labeled Non-LLC- obstruction these information includes: execution time currentTime, the number of read hits RD, the number of write hits WR, and the number of cache miss rate. Second, at the end of the first part, OAM calculates the following two parameters.

$$MissR > \frac{Miss}{RD + WR} \qquad (2.3)$$

$$OAP_{th} = \frac{T_{Mem} - RD \times T_{Rd} + WR \times T_{Wr}}{T_{Mem} + T_{Wr}} \qquad (2.4)$$

Where the Miss R is the actual Miss rate and OAP_{th} is the threshold value of the OPA. If in a process, the Miss rate is greater than the OAP_{th}, then the OAM will identify this process as LLC-obstruction. If the Miss Rate is smaller than it will be identified as Non-LLC-obstruction, this identifying process is the results of the OAM results. The Obstruction aware policy is illustrated in figure 2.4, where its function can be described as follows;

- L 3 Read Hit: In this case, the data from the third level cache will be transferred to the cache of the corresponding cache level 2.
- L3 Read Miss; Initially, the cache of the L3 requests data from the main memory. When the data is received, if the OAP checks if it is read request is LLC-obstruction, the data is transmitted directly to the L2 cache. Otherwise, date will be stored on L3 normally.
- L3 write Hit: Initially, the OAP controller examines the write request, if it is the LLC-obstruction then L3 invalidates the hit date block otherwise write request would be sent directly to the main memory.
- L3 Write Miss; OAP checks if the request is LLC-obstruction then it this request will be forwarded to the main memory. If it is Non-LLC obstruction, the cache line to be fetched and allocated to write the new date in L3

The results obtained from the above method in a 4-core processor with 8 MB of STT- RAM cache memory level 3 indicate that in addition to a 64% reduction in power consumption, performance is up to 42% and an average of 14% has improved.

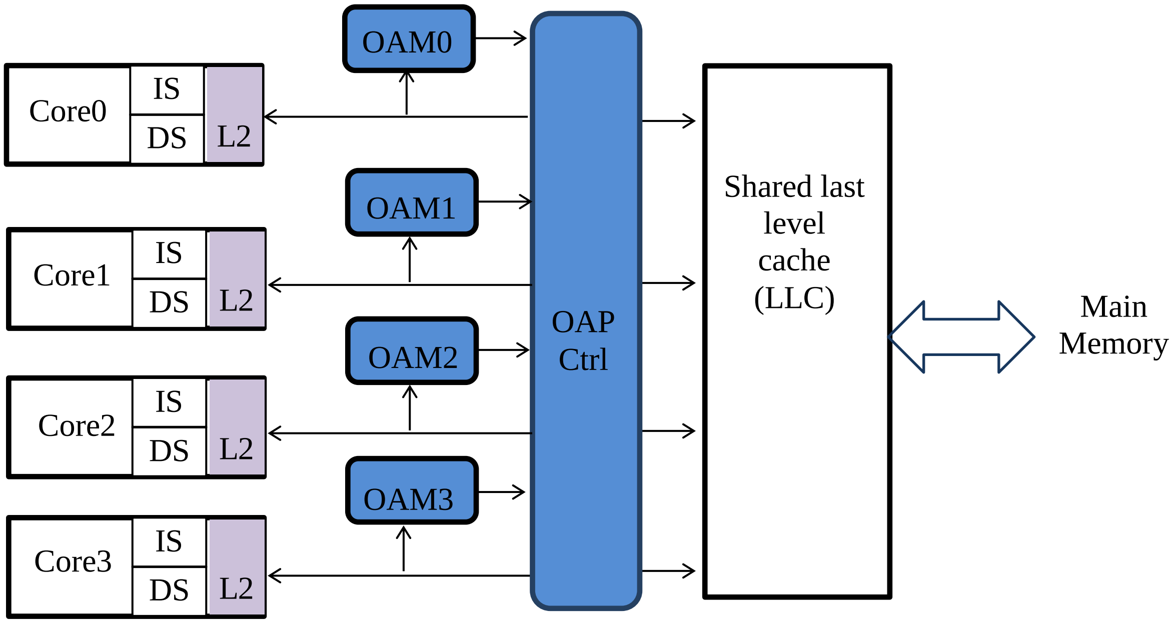


Figure 2.3 A 3-level cache hierarchy enhanced by obstruction-aware cache management policy (OAP) Newly added structures are in blue [3]

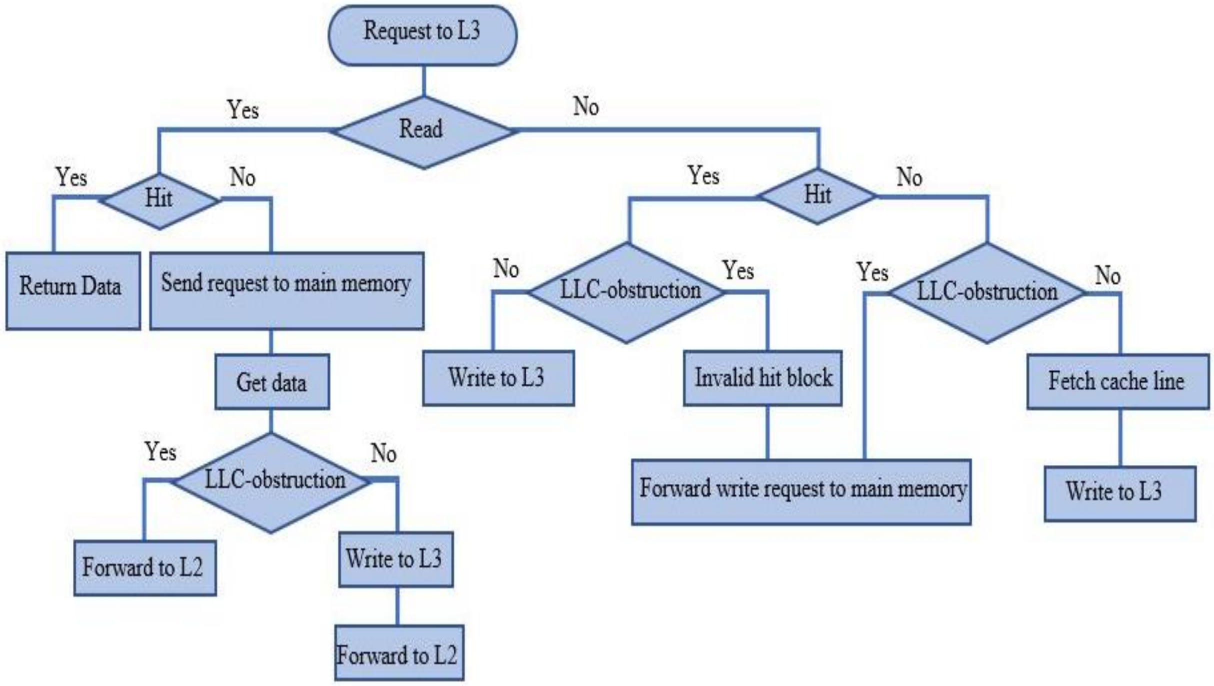


Figure 2.4: OAP controller structure: The OAM-based "latency cache memory" diagnosis involved with each core [3]

In [1], a reconfigurable hybrid cache architecture (RHC) is presented. This architecture reconfigures the size of the SRAM and NVMs by turning off and on the SRAM and NVMs arrays in order to better accommodate the memory required for different workloads. On average, this architecture significantly reduces energy by 64%, 46% and 28% with Maximum of system degradation of 4% with respect to non-reconfigurable cache SRAM-based memory, non-reconfigurable hybrid cache memory and reconfigurable SRAM based cache memory. Figure 2.5 shows the overall structure of the reconfigurable hybrid cache (RHC). The 16-way RHC architecture uses 1MB of 2-waySRAM and 3-megabyte 14-way STT-RAM. In addition, due to the speedy SRAM access, the secret memory entries are fully implemented.

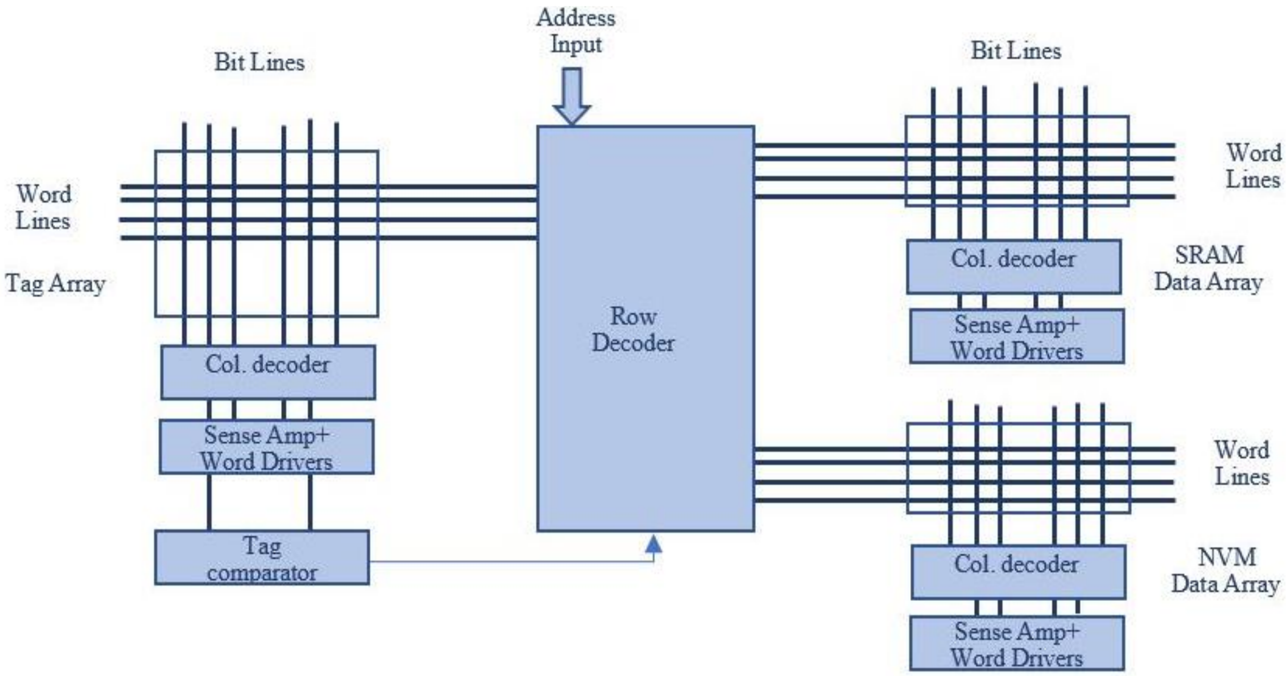


Figure 2.5: Hybrid reconfiguring cache memory design [1]

Figure 2.6 illustrates the design of the power supply gate used in the RHC structure. A centralized management unit has the power to send/receive a signal to any type of SRAM and STT-RAM. The power supply circuitry of anyway from tag and data arrays of SRAM technology uses NMOS transistors to reduce the leakage current. In the RHC structure, SRAM memory cells that are in the same way are connected to a common GND virtual signal. PMOS transistors are used for lateral circuits such as word driver, row and column decoders and amplifiers. However, due to the very low leakage current, NVM memory cells do not utilize the power supply circuitry and turn them off / on by sending a control signal to the side circuits corresponding to each way of the NVM form.

In this technique, two methods are used to reconfigure the RHC structure:

- Way-Based Decay Scheme: A cache decay idea was used to save leakage by powering off the cache block which was not accessed for a long time period (decay interval). This was done by using way-based decay counters, as shown in Figure 2.7. In this method, a two-bit local saturation counter for each block is used. When a global counter increases the number of decay intervals, the counter increases and when the access block is reached, the counter will be zeroed. The counter saves the number of decay blocks in each way, and when it reaches a certain threshold of 90%, the entire path in the cache is silenced. In order to clarify the desired way, the entire tag array remains in place to store the potential hit rate for the desired path. Now, when the potential impact rate is higher than the threshold, it will be cleared in order to reduce the rate of the block in the desired way.
- Hit Counting Scheme: In the previous scheme, due to independent control, a large number of cache

paths may be shut off at the same time. This may degrade the performance though increasing of L2 cache misses with low decay intervals. Also, a single decay interval can’t accurately capture the varied decay intervals of all cache blocks. Contrary to previous methods, this method utilizes a potential hit counter as a decision to shut down. When the counter is lowered from the threshold TH_{off}, the desired way can be turned off. In this way, only after 10 consecutive periods of time, the permission can be turned off. For both SRAM and STT-RAM, the values TH_{on} and TH_{off}, 50 and 0 are considered, respectively.

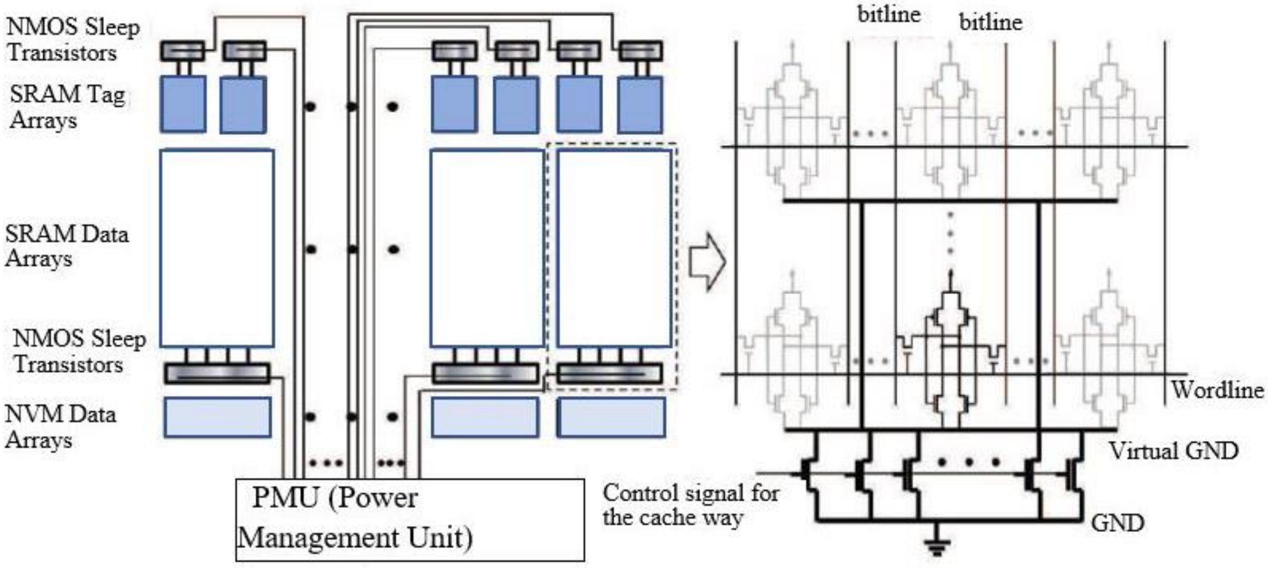


Figure 2.6: Design of the power supply gate for RHC [1]

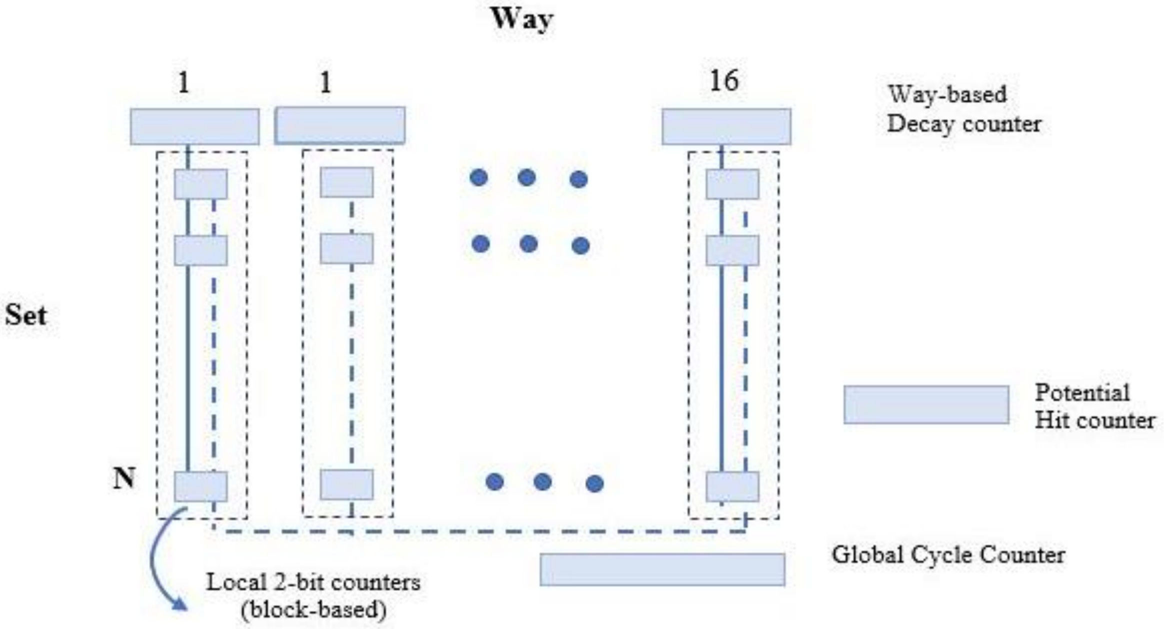


Figure 2.7: Counters for dynamic reconfiguration [1]

Sun et. al. [4] attempted to improve dynamic energy consumption by using NVMS technology along with SRAM one at cache level 1.in addition to the reduction on the power consumption, NVMS reduces the soft errors. In the later works several SRAM buffers were used with STTRAM cache bank to mitigate the long writing latency and high current writing operations in high frequency access like L1. Most of the other researcher are targeting the less frequency access cache levels (L2 and L3). They proposed hybrid SRAM and MRAM cache architecture is shown in Figure 2.8, which has the following two features:

- The added small SRAM buffers can complement the shortfalls of MRAM in writing operations with having the its larger capacity and near zero leakage power, both dynamic and leakage consumption will be reduced.
- Protect cache memory from radiation-induced soft errors, as MRAM is inherently invulnerable to emissive particles.

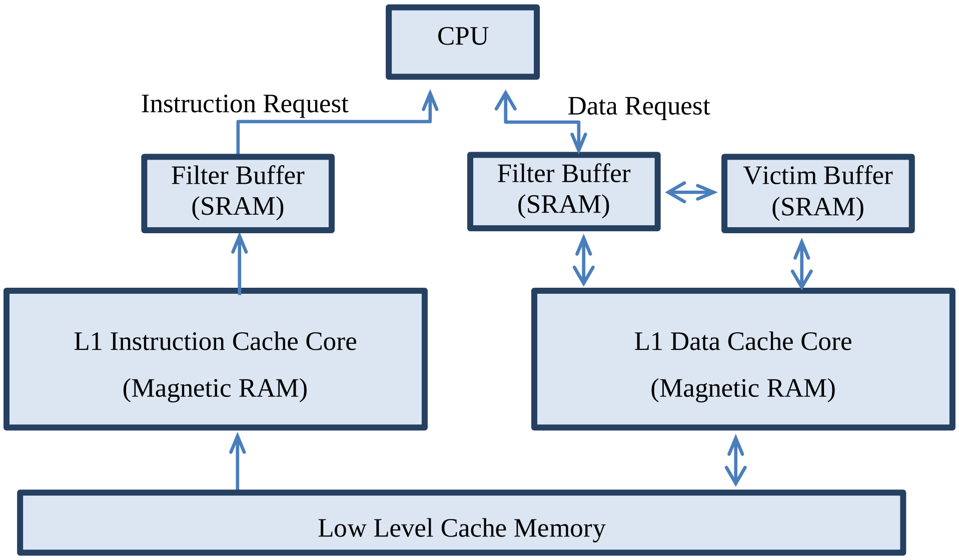


Figure 2.8: SRAM-MRAM hybrid cache [4]

2.5 Summary

As noted in this chapter, memory hierarchy, and especially cache memory, will have to be increased with increasing processing cores and in order to be able to respond to their requests avoiding off chip access. SRAM is the most commonly used technology for cache memory, which has led to a significant portion of leakage power in the subnuclear technology. This chapter introduces the emerging non-volatile memory technology, which benefits from zero-power, nonvolatile leakage, resistance to high-grade, high-density errors. At last, due to the delay and high power in writing operation in this technology, several techniques were reviewed for effective using NVMS in the memory cache architecture. Reconfigurable hybrid cache was the most efficient approach which tries simultaneously to take advantages of both SRAM and STT-RAM technologies.

Chapter 3

Proposed Methods

3.1 Introduction

As mentioned in chapter 2, the hidden hierarchy of significance is significant from the point of view of the operational capability to deal with the memory wall and reduce the processor access time to memory. Considering the significant allocation of transistors to this part of the chip, and due to the increased power consumption of conventional SRAM memory leakage power in the sub-micron technology, a significant portion of the chip's power consumption is utilized for the last level cache. Therefore, the proposed methods which is meant to cope up with these challenges, will be explained in more details in this chapter.

3.2 Proposed Architecture: Motivation

Today, due to the fact that the dark silicon problem and the lack of essential parts of the chip are justified, the area is of little importance and it is trying to spend energy efficiently [51]. To this end, the integration of heterogeneous processor cores with different processing power and power consumption, or the same architecture with a different manufacturing process, is positioned in a chip as a solution for achieving efficient energy CMPs. Then, depending on the system requirements, it tries to switch between the kernels. Therefore, even when the system has a bit of workload, it does not require high processing power and can save significant power consumption by switching to a weaker processor

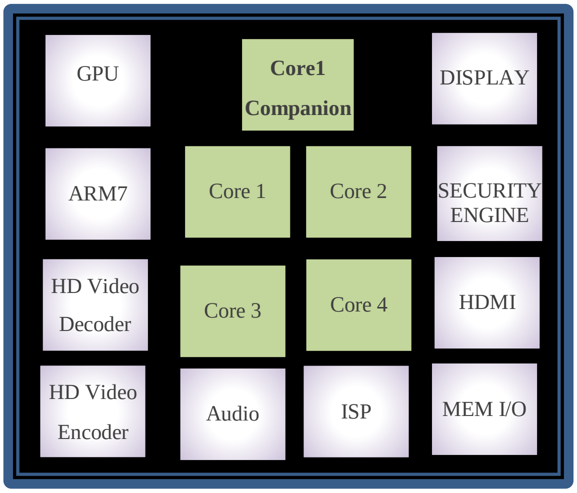


Figure 3.1: Companion processor core with less power consumption than other processing cores [5]

For example, in [5], NVIDIA has introduced a new plan for integrating heterogeneous processing cores to reduce power consumption and improve mobile phone performance. In addition to the four operating fast cores that are integrated with the standard manufacturing process, this design uses, with these cores, the same architecture but with the power optimized core. This core is named Companion core as can be seen in Figure 3.1. Companion core is built using low power process technology and yield a weaker performance than other cores.

So, when there is no need for high processing power, the four main processing cores are off to maintain the battery's optimum battery life and the Companion processor is replaced. Otherwise, based on system load, each of the four cores or some of them is used simultaneously. This is illustrated in Figure 3.2.

The above technique is also used to overcome the problem of dark silicon in [52]. In our work, as in the above method, we have tried to integrate along with the CMOS technology several cores with lower tunnel field-effect transistors (TFETs) technology.

In this regard, in the proposed method we use heterogeneous integration in hidden memory to overcome dark silicon, contrary to the above methods. In this work, the LLC memory of the two types are SRAM and STT-RAM, and its architecture is reconfigured to fit the workload of the system.

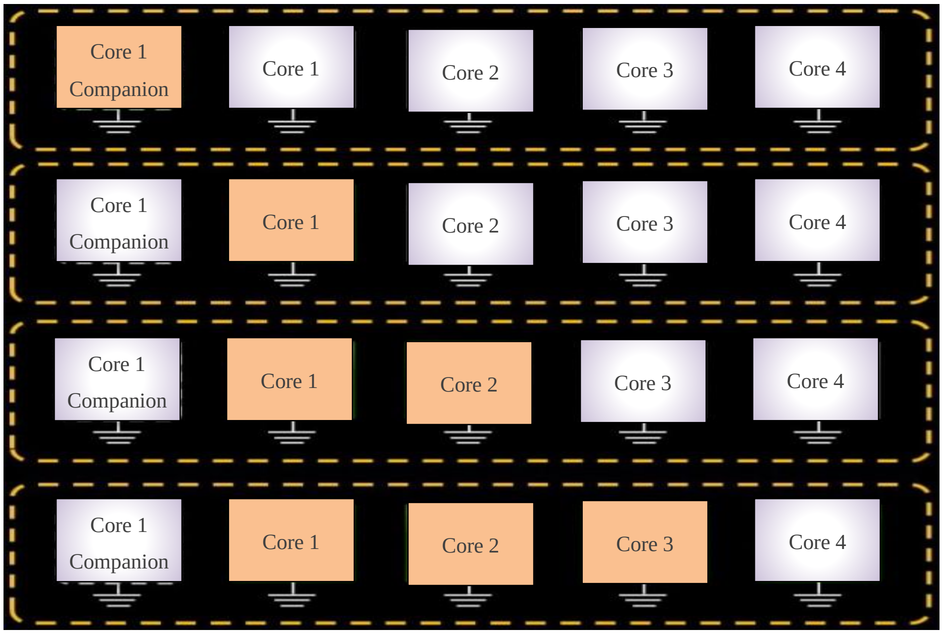


Figure 3.2: Managing cores based on system workload [5]

3.3 Hybrid STT-RAM and SRAM cache system

Usually, STT-RAM cells have a longer life span than PRAM $(4 \times 10)^{12}$ versus 10^9 writing cycles, see Table 3.1. In addition, the lifetime of the STT-RAM and PRAM memory, based on the number of write cycles, has been computed in a single-chip combination cache including 1MB SRAM and 3MB NVM in the paper [1]. Table 3.1 shows the results for 3 benchmark programs with the predominant writing operation of the medical imaging and PARSEC series.

For PRAM-based combination cache memory, the life span is limited to 4.7 to 196.12 days, while STT-RAM-based cache memory can last more than 10 years. Therefore, STT- RAM seems more suitable for onchip last level cache. In this thesis, the proposed methods of non-volatile memory STT-RAM is used for the longer lifetime and closer performance to the SRAM technology.

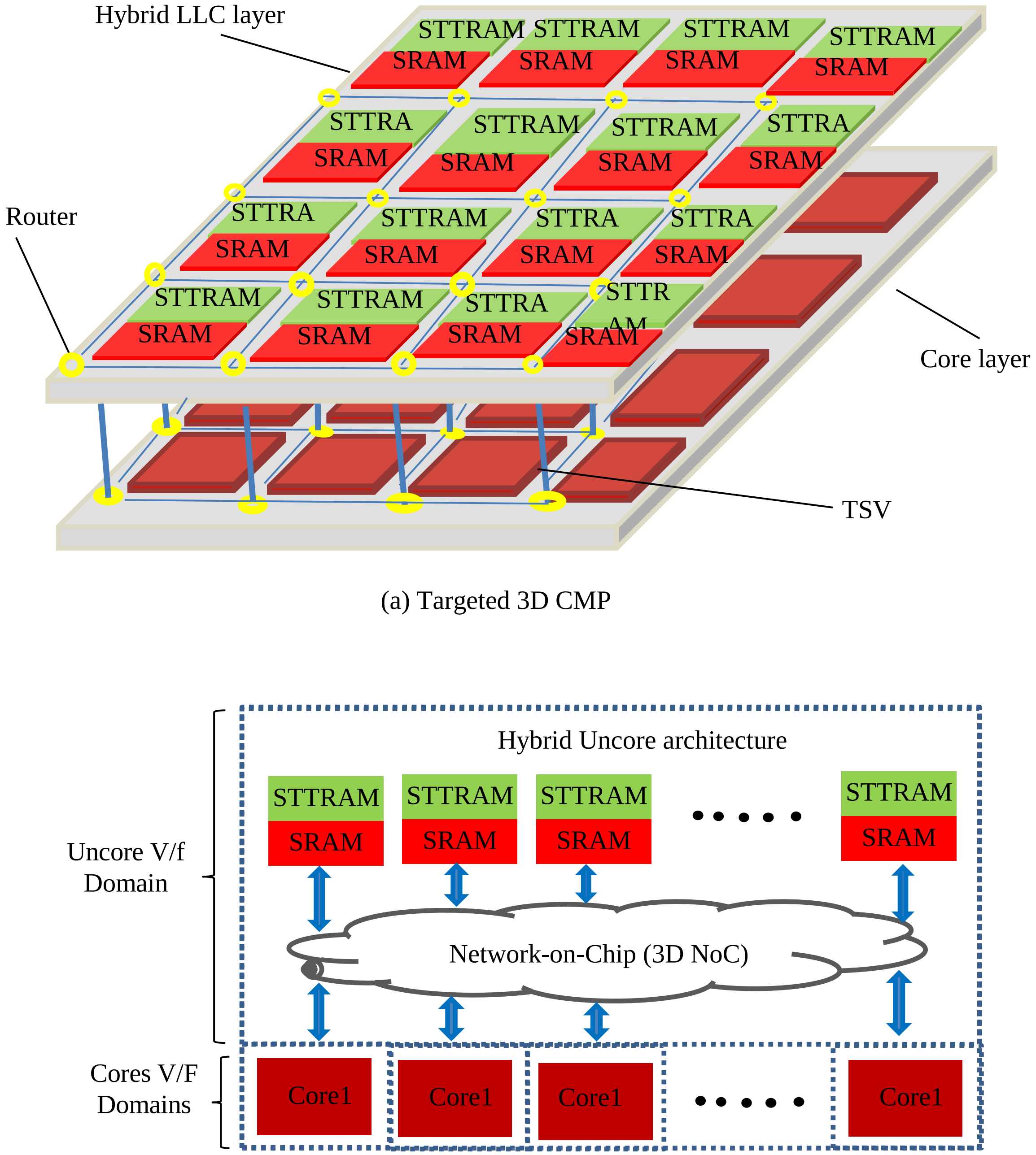
Table 3.1: Life span of STT-RAM and PRAM memory for 3 benchmark programs [1]

Workloads	Registration	Segmentation	Fluid-animate
PRAM (days)	4.7	196.12	39.33
STT-RAM (years)	12.88	537.32	107.6

Using STT-RAM and SRAM memory cells together in hybrid cache architecture targeting more energy efficient CMPs, requires close monitor to the system performance. Due to the high energy and delay of writing operations in STT-RAM technology, its complete replacement with SRAM technology can not only offset the benefits of its use, but it can also degrade the system performance. So, in this way, both technologies simultaneously to be used in hybrid system.

3.3.1 Proposed Method Architecture

In the is work, a 3D architecture CMP is proposed which consists of two layers, Cache layer of 16 cache banks is connected with processing core layer of 16 cores. These two layers are connected by thick Through Silicon Vias (TSV)s as shown in Figure3.3. In this model, each LLC memory bank includes an SRAM bank and an identical STT- RAM bank, connected through a network-selective system on the chip. The STT-RAM memory is denser than SRAM memory and it is four times as large in the same area. Therefore, the total latency of the proposed latency is 1MB of SRAM and 4MB of STT-RAM. Based on the network's performance, the type of each of the memory banks varies with the switching on and off from one of STT-RAM into SRAM banks or vice versa. So, at any moment, only one of these two technologies are used in each bank. Meanwhile, with every change in bank type it will be empty and starts to work again when it is selected.



(a) Targeted 3D CMP

(b) Separate Voltage / Frequency domain and hybrid uncore structure

Figure 3.3: Overall view of the proposed architecture method

The difference between last level cache memory banks results in two occurrences:

1. By changing the type of the banks, the volume of memory of the LLC will increase (from the 1 megabyte to 4 megabytes). This will reduce the LLC Miss rate and may improve the memory performance of the system.
2. Changing the LLC banks from SRAM to STT-RAM reduces uncore power consumption as the STT-RAM has lower leakage power which has become more critical in submicron technology.

System performance may be improved by decreasing the Miss Rate, but this improvement has plenty of dependencies on the nature of benchmark applications and the number of writing operations in it plays an important role. For example, if the benchmark application has a small working set, so that 1MB of SRAM memory is enough for it, the increase in the last level cache memory associated with the depletion of transmitted banks will only lead to an increase in the Miss of rate.

Furthermore, if the number of writing operations in the bank is high, the high latency of write intervals in STT-RAM technology will block future access and can have a reciprocal impact on the system's performance. The later can offset the enhancement of the Miss rate reduction, based on the nature of the benchmark application the performance of the system may improve or degrade.

3.3.2 Proposed Method Algorithm

Hybrid reconfigurable last level cache memory with dynamic voltage frequency scaling for the uncore area are applied in run-time approach based on the status of the system network, to ensure the minimal impact on the system performance. For this reason, accurate performance monitor is very critical.

Meanwhile, the average memory access time (AMAT) is known as one of the best metrics for system performance monitors, which, in addition to monitoring the current state of congestion in the network, also demonstrates performance well. When there is a miss of a private cache (L1 and L2), the network is used to fetch the target block in the LLC memory. If this criterion is low, it indicates that less demand is required from the uncore area, and slower uncore can be set. When AMAT is high it indicates that system network is demanding faster LLC to enhance the performance.

Therefore, in this technique, the mean of memory access time is used as a system monitor for the instantaneous estimation of system performance. The general relationship of this criterion is calculated as follows:

$$Average\ Memory\ Access\ Time = L_1 Hittime + L_1 Missrate \times L_1 Misspenalty \qquad (3.1)$$

For a system with two levels of hidden memory:

$$L_1 Misspenalty = L_2 Hittime + L_2 Missrate \times L_2 Misspenalty \qquad (3.2)$$

$$L_2Misspenalty = Memory\ Latency \tag{3.3}$$

In proposed technique, at each computation intervals of 1 millisecond, AMAT is calculated and compared with the reference value that calculated from training session. If this the calculated AMAT is less than reference, this means that the system performance is improved from the point of view of the average memory access time. In this case, power saving measures can be applied.

Algorithm 1 represents the proposed algorithm for our method and Equations 3.4 and 3.4 use to update the AMAT.

$$AMAT = AMAT + (1 - L_HR) \times L_2HR \times L_2Sram + (1 - L_1HR) \times (1 - L_2HR) \times External \tag{3.4}$$

$$AMAT = AMAT + (1 - L_HR) \times L_2RHR \times L_2STTR + L_2WHR \times L2STTW + (1 - L1HR) \times (1 - L2HR) \times External \tag{3.5}$$

Energy saving measures consist of two techniques, first the type of one of the LLC memory banks that has the highest reading Rate to be changed from the SRAM to the STT-RAM so that according to lack of energy and high writing activity have the least negative impact on power consumption and system latency. At the same time, their type will be changed from the SRAM to the STT-RAM in the next period to reduce the leakage power consumption. Second technique is to dynamically scale the voltage frequency of the uncore area.

Now, if the AMAT is greater than the reference value, the system's instantaneous performance is lower than the reference state and in order to improve it the type of one of the cache banks that has the highest write access rates to be changed from STT-RAM to SRAM. Therefore, the performance of the system is significant and attempts to reduce the power consumption of the chip by the least performance degradation. damage to it. The complete algorithm of the proposed method is observed in Algorithm 1.

Algorithm 1. Dynamic Voltage/Frequency scaling and LLC configuration

```
L1HR is average L1 hit rate of all cores
L2HR(i) is hit rate of L2 bank i
L2RHR(i) is read hit rate of L2 bank i
L2WHR(i) is write hit rate of L2 bank i
L2RN(i) is number of read access to L2 bank i
L2WN(i) is number of write access to L2 bank i
L2_STTR is STT-RAM read latency
L2_STTW is STT-RAM write latency
Minw is cache tile with Minimum number of write access
Maxw is cache tile with Maximum number of write access
$\boldsymbol{f}$ is uncore domain frequency
$\boldsymbol{v}$ is uncore domain voltage
$\boldsymbol{\eta}$ is constant for frequency and AMAT relation
$\boldsymbol{\beta}$ is constant for frequency and AMAT relation
AMAT = L1HR×L1_Sram_latency
for (int i=0; i < NumberOfBank; i++ ) do
  if (Type(i) = SRAM)
    AMAT += (1-L1HR) ×(L2HR(i)×L2_Sram_latency +(1-  L2HR(i))×External);
  else \\STT-RAM
    AMAT += (1-L1HR)× (L2RHR(i)×L2_STTR+L2WHR(i)×  L2_STTW+(1-L2HR(i))×External);
  end if;
end for;
Minw = 0;
Maxw = 0;
if (AMAT <AMAT_ref)
 for ( int i=0; i<NumberOfBank ; i++ ) do
   if (L2RN(i) / (L2WN(i) + 1) > L2RN(Minw) / (L2WN(Minw)+1) and Type(i)==SRAM ){
       Minw = i;
       $\boldsymbol{f} = \boldsymbol{\eta}/(AMAT - \boldsymbol{\beta})$
       $v = f^2$
  end if;}
 end for;
 Type(Minw) = STT-RAM;
else
 for ( int i=0; i< Number_Of_Bank; i++ ) do
    if ( L2WN(i) > L2WN(Maxw) and Type(i)==STT-RAM )
       Maxw = i;
    end if;
 end for;
 Type (Maxw) = SRAM;
end if
```

3.3.3 Uncore Voltage / Frequency Scaling

Scale V/F of the uncore domain to a lower or higher level depending on the estimated AMAT for the same analogy of the previous section. For each interval, AMAT is being calculated and compared to the AMATref and accordingly scale the uncore V/F. When the estimated AMAT is smaller than AMATref , a lower V/F level is set for the uncore domain shown in Figure 3.4. The following equation translates the calculated AMAT into frequency [10]

$$f = \frac{\eta}{AMAT - \beta} \tag{3.6}$$

where η and β are, constants connect the inverse relation between the frequency and AMAT in equation (4). These constants are required to ensure a stable system to avoid any oscillation when the calculated AMAT leads to lower V/f which in turns results AMAT requires going back to the first V/F. Voltage is related to the value of the frequency and the voltage as shown in Algorithm1

3.3.4 Collecting Data at the Cores

AMAT is selected as the network monitor to observe the system performance on run-time approach, all the required information to calculate AMAT that shown in its formula in previous section are to be collected from all the cores. This information includes; the specified values of access time to the private cache memory levels 1, 2 and the main memory, the values of the variable loss rate of cache memory level 1, 2 are also required.

Therefore, at run-time, it is necessary to calculate the values of these variables that are related to the workload of the system. Given that the LLC is composed of 16 banks comprise each database that stores the number of Hit Rate of reading and writing operations and, by aggregating these values, the Miss Rate of this level of cache memory is obtained. Of course, it should be noted that in the algorithm used to change the type of each bank at the end computation interval, it is required to distinguish between the number of Hit and Miss rate of reading and writing access. Therefore, the total number of Hit and Miss of read and write operations for each bank is to be stored at any time interval. A centrally located pre specified core is required to execute the AMAT calculation and finally compare it with the pre-specified reference value. The purpose of central location is to ensure the minimal distance to the rest of the cores which consequently reduce task latency of collecting these information from the processor cores and the LLC memory, figure 3.4 illustrates the central location of the power control unit (PCU) which is core 10. The gathered information is written into control flits as shown in Figure 3.5 assuming 128bit wide links, each control flits has 128 bits to transmit the following information ; L1 Hit rate represents L1 rate successful read access at the end of the time interval, L2 number of Read Hit (L2) represents the number of successfully access memory caches of Level 2, number of Read Miss (L2) represents the number of unsuccessful attempts to read cache memory level 2, number of Write Hit (L2) represents the number of successful accesses for the write access of Level 2 and number of Write Miss (L2) represents the number of unsuccessful attempts to write at level 2 cache.

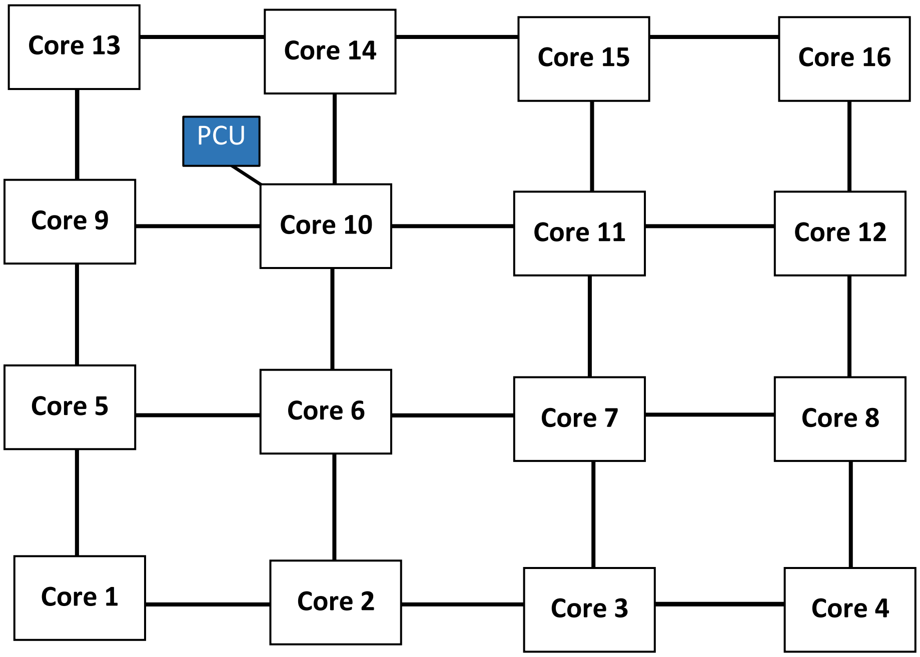


Figure 3.4: Central location for the power control unit [57]

FT	VC	Route info	MSG	L1 Hit Rate	No. of Read Hit (L2)	No. of Read Miss (L2)	No. of Write Hits (L2)	No. of Write Miss (L2)

Figure 3.5: Bit fields of a control flit [57]

3.4 Summary

In this chapter, a reconfigurable hybrid cache architecture for the last level cache memory was introduced for the modern CMPs. These methods, target more energy efficient uncore CMPs by adjusting the type/size of the LLC and dynamically scale the operational parameter of the uncore area (Voltage/Frequency). Several techniques were presented to reduce the overhead calculation and minimizing the shortfalls of the NVMs and slowing the Voltage/Frequency of the uncore area under the performance constraints. In order to ensure the minimal impact on the system performance, energy saving techniques are to be applied on the system based on the system network status which indicate the system demand from the uncore area. AMAT was selected to observe the system network. The proposed architecture will be compared with baseline one which has SRAM LLC banks only without DVFS in the uncore area.

Chapter 4

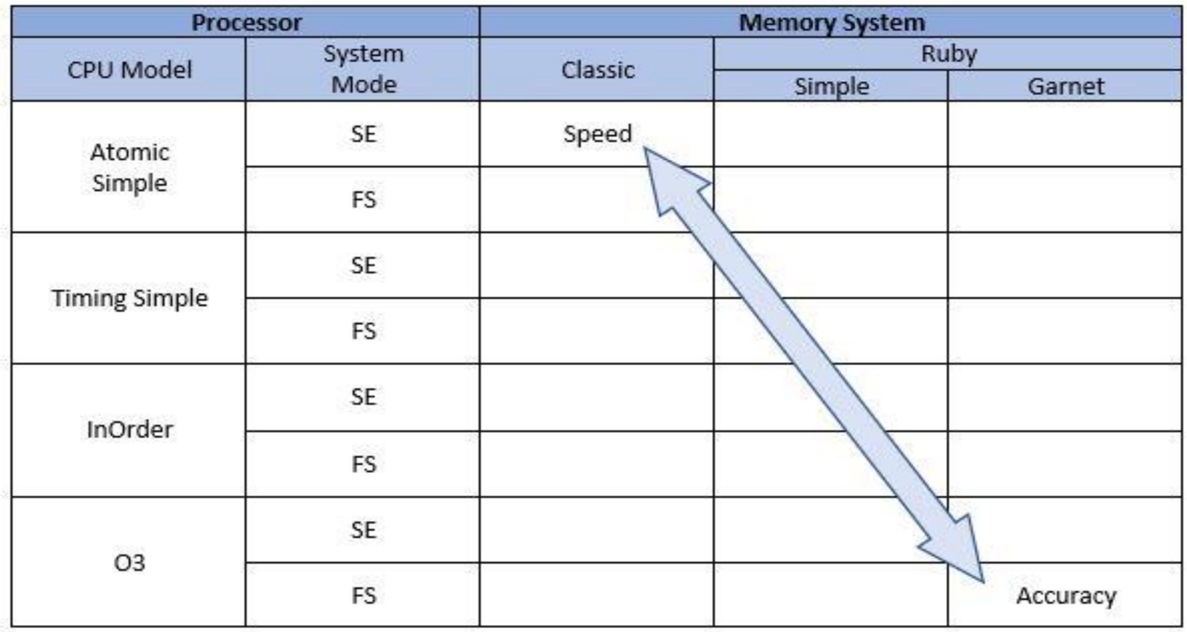
Simulation and Experimental Results

4.1 Introduction

The simulation platform, selected benchmark and its applications is described in this chapter. The experimental results and discussion are also presented.

4.2 Platform Setup

The Gem5 system simulator was designed to simulate a system behavior with 16 processor cores, along with the McPAT, the CACTI memory simulator for SRAM and DRAM technology, and The NVsim memory simulator is used for emerging non-volatile memory technologies. In the process, PARSEC benchmarking is also used to validate the proposed method. The following briefly describes the used simulators to implement and compare the baseline and proposed architectures.



Processor		Memory System		
CPU Model	System Mode	Classic	Ruby	
			Simple	Garnet
Atomic Simple	SE	Speed		
	FS			
Timing Simple	SE			
	FS			
InOrder	SE			
	FS			
O3	SE			
	FS			Accuracy

Figure 4.1: Gem5 simulator covers a wide range of speed and accuracy [6]

4.2.1 Gem5 Simulator

The Gem5 simulator is a full system simulator which integrates the best parts of the two M5 and GEMS simulators. The Gem5 simulator is a highly configurable simulation framework which can evaluate a large range of systems with different CPU models, system execution modes, and memory models. The Gem5 simulator provides the flexibility of easily simulating systems with ability for simulating a set of instructions for various architectures and modeling that combine accurate and flexible memory structure such as hierarchy coherence protocols [6]. Figure 4.1 shows the wide range speed and accuracy tradeoffs. Key features of this simulator are described below:

The Gem5 simulator provides four different processor models with a distinct point in the range of speed and accuracy: AtomicSimple, TimingSimple, In-Order, and O3. Atomic-Simple and TimingSimple are models without a pipeline that execute and execute only one command per clock. The AtomicSimple processor is the fastest model with at least 1 command per clock, with all memory accesses complete quickly. While TimingSimple simulates memory access scheduling by allowing only one per request at a time. The InOrder pipeline model and runtime commands are simulated. At the same time, the number of pipeline levels, width, and the number of hardware threads on the processor can also be changed. Finally, in the O3 processor, in addition to the pipeline, off-the-clock execution commands are simulated along with the data dependencies between commands, computational sections, memory access, and pipeline levels.

Each processor model can be launched in one of two SE or FS modes. In SE mode, with the simulation of hardware, most system-level services are prevented from the operating system and device modeling. In contrast, in FS mode, both user and kernel-level commands are executed and a complete system with all devices and systems.

The Gem5 simulator includes two different models of the Classic and Ruby memory system. The Classic model provides a fast, easy-to-set memory system, while the Ruby model inherited from the GEMS simulator has a flexible structure with precision simulation capabilities along with a wide range of protocols.

The Gem5 simulator also can handle workloads on several processor architectures. Currently, the simulator supports ARM, ALPHA, MIPS, Power, SPARC, and X86. Meanwhile, this group simulator includes AMD, ARM, HP, MIPS and universities in Princeton, MIT, Michigan, Texas and Wisconsin [6].

4.2.2 McPAT Simulator

McPAT is the first integrated framework for modeling power consumption, area and time for multi-core / multi-core processors developed by HP Labs. This simulator is designed to work with a variety of performance simulators, power and ...etc., and with manufacturing technology in the range of 90 to 22 nm. At the micro-architecture level, McPAT includes models for the core components of the chip processor, including sequential and unplanned processing cores, network chips, shared cache memory, main memory controllers, and several clock domains. Also, on the circuit level, this simulator supports time-critical modeling, area modeling and dynamic power modeling, short circuit and leakage support [7].

McPAT uses a flexible XML interface to simplify the use of multiple performance simulators. This interface includes fixed architecture configuration data and variable system activity statistics, which are the same as the output files config.ini and the stats.txt Gem5 simulator. To do this, in this simulation, a script in Python is used in the Linux environment to extract data from the XML interface from these two Gem5 outputs, and then to get the energy and area data, look like McPAT builder. In Figure 4.2, the overall framework of the McPAT simulator is fully observed.

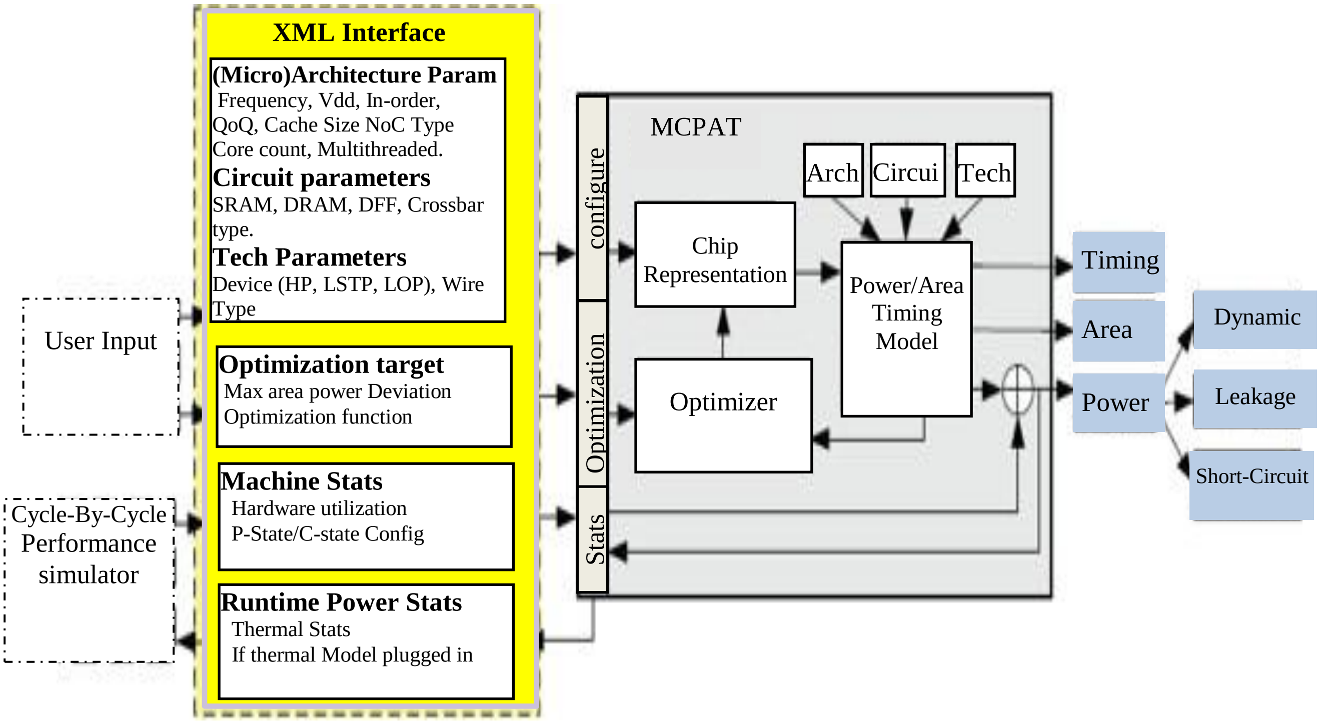


Figure 4.2: Overview of the McPAT simulator framework [7]

4.2.3 CACTI Simulator

CACTI is a hidden and original memory simulator designed by HP to evaluate and measure the performance of the hierarchy of memory. As manufacturing technology shrinks, the instability of the manufacturing process between the transistors and the delay in wires increases. Therefore, the structure of future cache memory is significantly related to the characteristics of interconnection networks whose various components are interconnected. 3D CACTI tool is the latest version of this simulator that focuses on the design of 3D cache access time and power estimation [8].

In Figure 4.3(a), the logical structure of a hidden memory is shown. The requested address to the cache is initially logged in to a decoder. After that, the word line link for the data array and the tag is activated and the content of a full line is routed through the network of wires to the sense amplifier. The routed output of the tag array is compared to the input address to ensure that the desired data is secured in one of the desired set paths. Eventually, if there is data in the cache memory, the comparator feeds the multifunction control input to send the data to the request's processor. In Figure 4.3(b), we can also observe the physical organization of the data array [8].

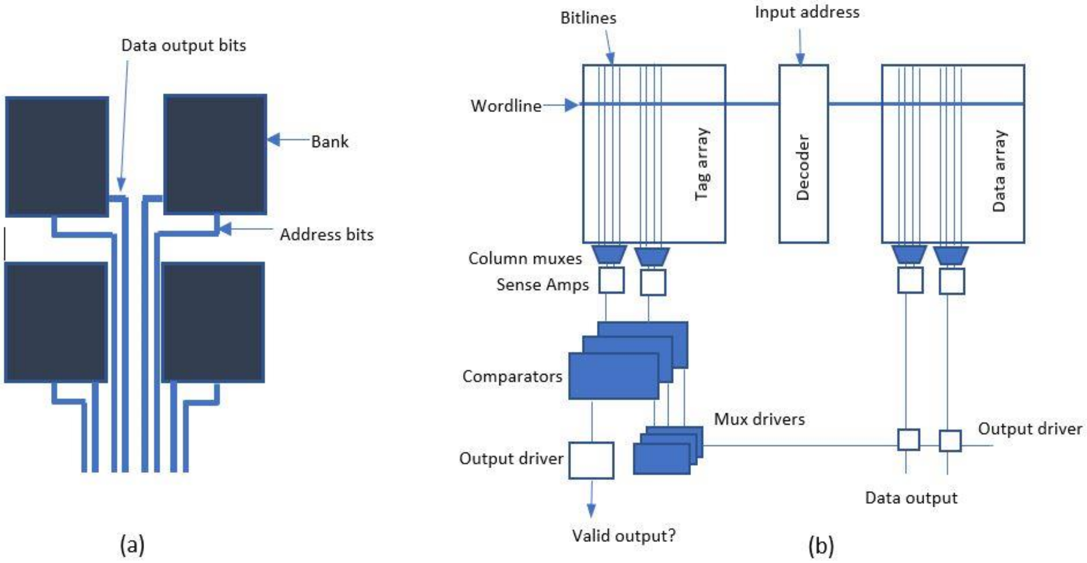


Figure 4.3: (a) Logical structure of the hidden memory; (b) Physical organization of the data array [8]

The CACTI simulator has major parameters such as cache memory, cache memory size, the number of hidden cache paths with associative mapping, manufacturing technology. The number of ports and the number of independent banks is received as an input. After the simulation, information such as delay with power consumption and area as output is generated.

4.2.4 NVsim Simulator

Over the past few years, a series of new non-volatile memory technologies have emerged. Among all these technologies, STT-RAM, PCRAM and ReRAM can be referred to as potential alternatives for DRAM, FLASH and SRAM in registers, last level cache memory, and main memory. To evaluate the performance of this type of emerging memory technology, we need a tool like CACTI. NVsim is a circuit-level simulator for estimating the performance, energy, and space of non-volatile memory. NVsim also has the ability to simulate SRAM and DRAM technologies. Meanwhile, this simulator has been validated by comparing it with the original non-volatile memory industrial prototype [53].

4.2.5 Noxim Simulator

The Noxim simulator is a chip network simulator developed by Maurizio Palesi, Davide Patti and Fabrizio Fazzino at the University of Catania in Italy. The Noxim simulator is designed using System C, a language for describing a system based on C++ language, and its main platform is a Linux operating system. Noxim uses the command-line interface to describe network parameters on the chip. The modified version of this simulator, in addition to the two-dimensional net, has the ability to simulate a three-dimensional grid. It also comes equipped with HotSpot software and can report chips power consumption with other parameters. The user can determine the network size, router buffer size, packet size, routing algorithm, packet type and packet injection rate, and type of traffic on the network. The simulator measures power performance, delay, and power consumption in the network and results in it in the output. This information is calculated in both the form of mean and results in each relationship. Also users are allowed to evaluate different parameters such as the total number of received packets/flits, average global power, minimum/maximum delay, total power consumption, latency/power/ energy per connection, . . . etc. also collect [54].

4.2.6 PARSEC Benchmark suite

PARSEC stands for Princeton Application Repository for Shared-Memory Computers. It is a benchmark set of applications for studies of Chip-Multiprocessors (CMPs). Earlier available benchmarks for multiprocessors have focused on high-performance computing applications and used a limited number of synchronization methods. PARSEC includes emerging applications in recognition, mining and synthesis as well as systems applications that mimic large-scale multi-threaded commercial programs. Table 4.1 shows that the benchmark applications which vary in the working set (small, medium and large), locality, data sharing and synchronization. The benchmark suite has been made available to the public. Experimental results will be presented as a comparison between the baseline and the proposed architectures for 12 applications from the PARSEC benchmark suits [2].

Table 4.1: Characteristics of the PARSEC benchmark set [2]

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
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

4.3 Simulation Results

The simulation process used to evaluate the method is presented in Figure 4.4 with the system simulation parameters as detailed in Table 4.2. The simulation was carried out on the baseline architecture when the last level cache has only SRAM bank and it was repeated with a hybrid reconfigurable LLC system as proposed earlier. Simulation results include different parameters to investigate the energy-saving and performance of the hybrid architecture with respect to the baseline one.

Table 4.2: Parameters of the simulated system [2]

Parameter	Value
Technology	32 nm
No. of Cores	16 Alpha Cores
Configuration	1GHz, Eight-way issue out-of-order
Private L1 Cache	SRAM, 32KB data/32KB, instruction,4-way set associative, MESI directory-based coherency, 64B line size, LRU policy, writer-back policy
Shared L2 Cache	Hybrid reconfigurable cache / Baseline:1MB SRAM, 4 way set associative Hybrid: 1MB SRAM, 4 way set associative + 4MB STT-RAM, 16 way set associative 64B line size, LRU policy, write-back policy
Main Memory	2GB DRAM
Network Router	2-stage wormhole switched, virtual channel flow control, 2 VCs per port, a buffer with a depth of 4 flits per each VC, 5 flits buffer depth, 8 flits per Data Packet, 1 flit per address packet, each flit is set to be 16-byte long

XY routing is a popular deterministic routing algorithm, it is a dimension order routing which routes packets first in x- or horizontal direction to the correct column and then in y- or vertical direction to the receiver [58]. X-Y routing algorithm works with a high-performance in low traffic situation, while in high traffic situation other adaptive algorithms such as odd-even and NoC have better performance [59]. We used X-Y routing in the proposed 3D architecture of CMP. In this routing method, when a packet is routed from the core layer to the cache layer, it is first routed using X-Y routing to a particular router in the core layer, followed by a TSV traversal in the vertical direction to a router in the cache region. Finally, the packet follows X-Y routing again in the cache layer to the destination cache bank. No such path restriction is imposed when communicating between the cache layer to the core layer i.e. all 64 TSVs can be used in an architecture with 64 cores. Wormhole switching decides when the packet moves forward from a router instead of defining the route to the destination. It uses virtual channel flow control, 2 channels per port which facilitate multiplexing multiple packets through same physical channels. Each channel serves 4 flits and the buffers have 5 flits depth as shown in table 4.2

Figure 4.5, shows the hit and miss Rate for the read and write in the LLC memory for each of the PARSEC benchmark application for the baseline and the proposed architectures. This chart shows reductions on the read miss for hybrid over several applications due to larger capacity comparing with baseline architecture. For example, fluidanimate read miss rate of the proposed model was improved by 30% below that of the baseline one. Improvements on the read miss rate can be seen also for canneal and fraqmine. On the other hand, miss rate increased for blackscholes which indicates that backscholes application has small working set and it was satisfied with the 1 MB SRAM LLC (the baseline model). The increase in the miss rate in backscholes caused cold-start miss due to change the bank type from SRAM to STTRAM.

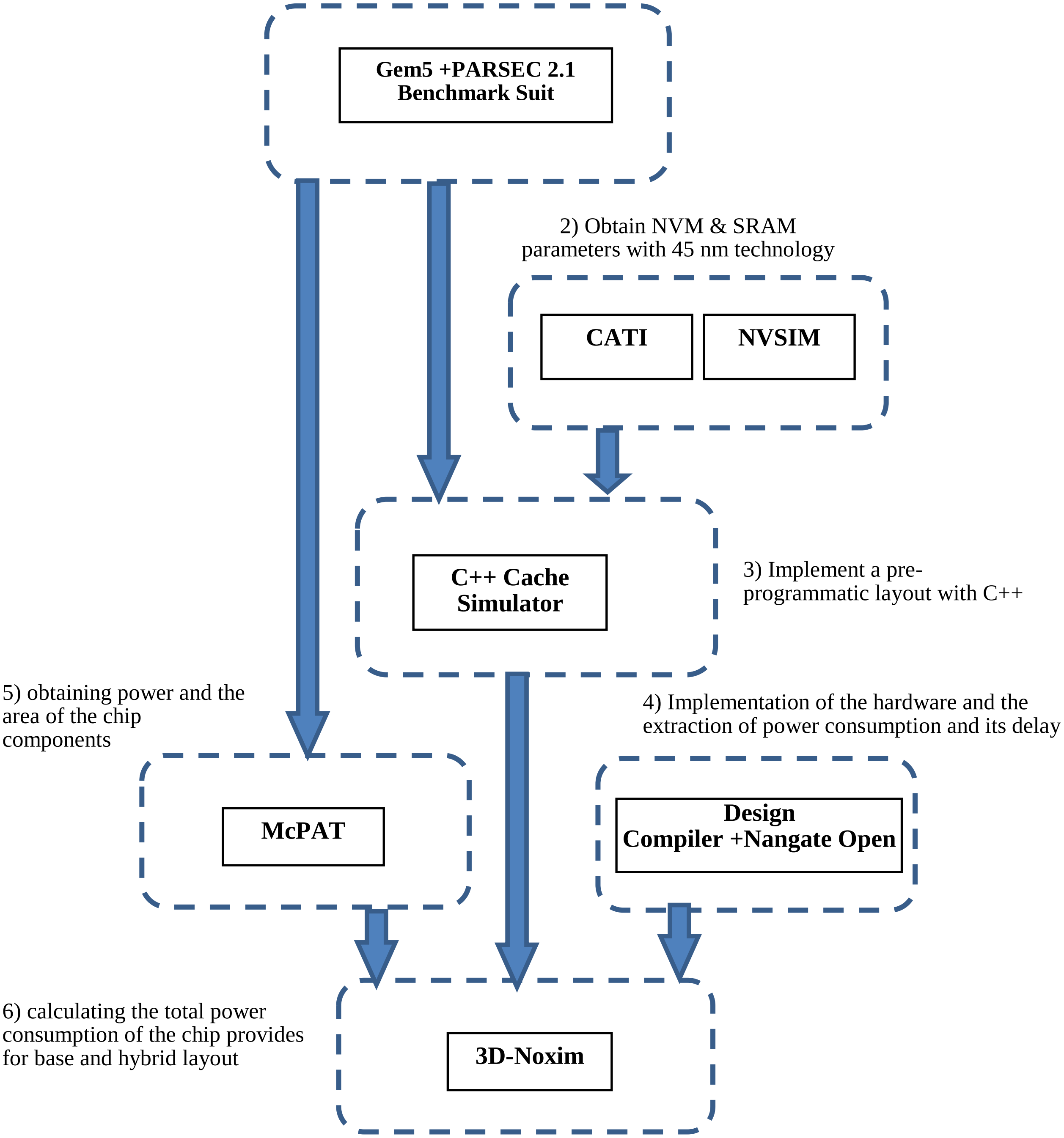

Gem5 +PARSEC 2.1 Benchmark Suit
2) Obtain NVM & SRAM parameters with 45 nm technology
CATI
NVSIM
C++ Cache Simulator
3) Implement a pre-programmatic layout with C++
5) obtaining power and the area of the chip components
4) Implementation of the hardware and the extraction of power consumption and its delay
McPAT
Design Compiler +Nangate Open
6) calculating the total power consumption of the chip provides for base and hybrid layout
3D-Noxim

Figure 4.4: Simulation process to evaluate the proposed method

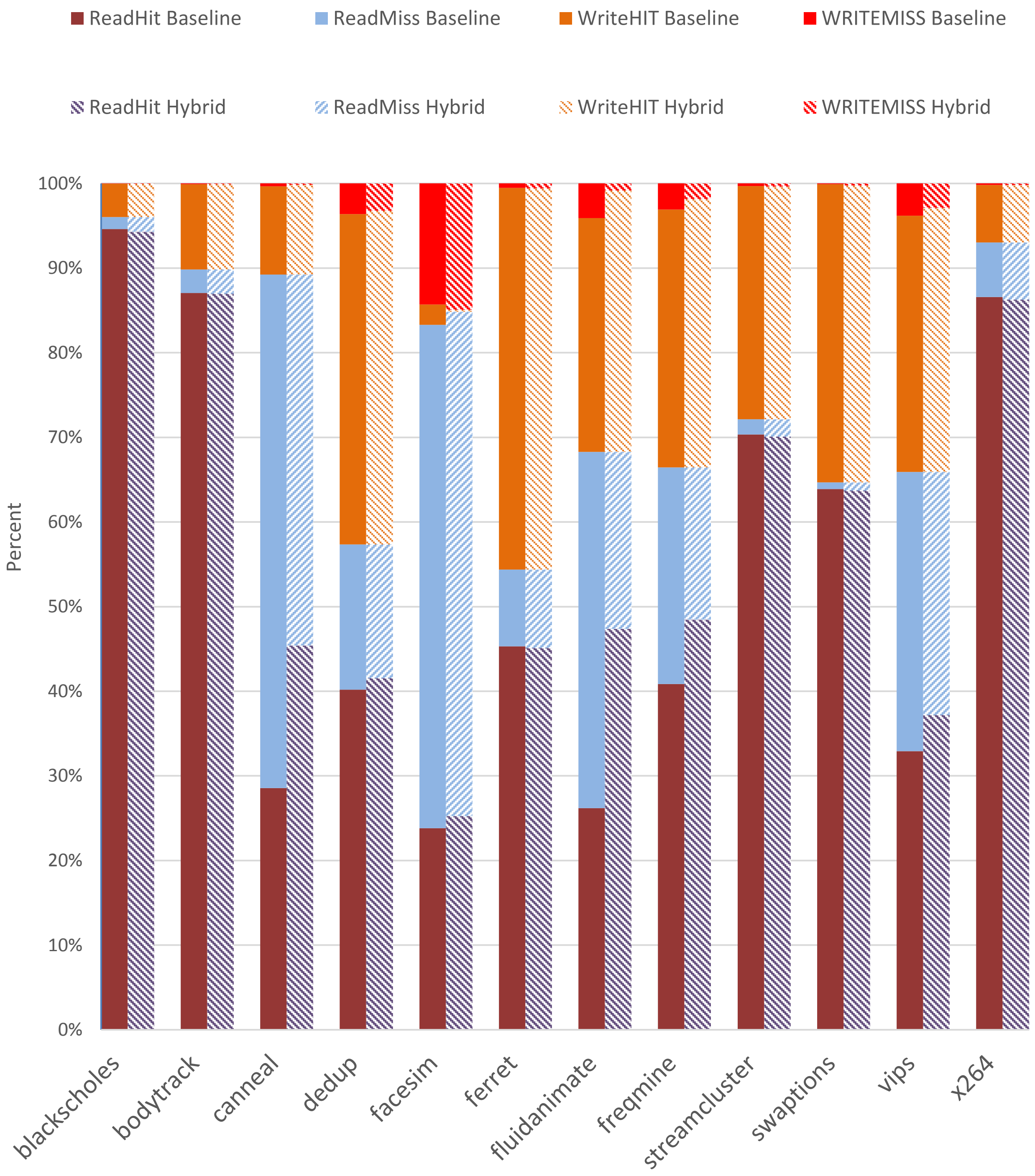


Figure 4.5: Percentage of Hit/Miss Write and Read access in the LLC for each of the PARSEC application for the baseline and hybrid architectures

Figure 4.6 illustrates participation percentages of each SRAM and STT-RAM technology in the proposed LLC memory structure. As STT-RAM technology contributes more to the LLC memory structure, it is expected that the miss rate of this level of memory will be reduced by increasing its capacity. However, this is not always true, because of the high percentage of writing accesses on some applications and changing the bank type to STTRAM, this improvement will be reduced and may even lead to system degradation of the hybrid structure. For example, cannael application has 89.2% reading hits and the STTRAM participation in LLC is nearly 80% from the LLC banks but its read miss was improved by 27 % below the baseline one as shown in the figure 4.8. Highest improvement of the reading miss was recorded for fluidanimate the large working set application. On the other hand, the small working set application backscholes experience increase in the miss rate as explained before.

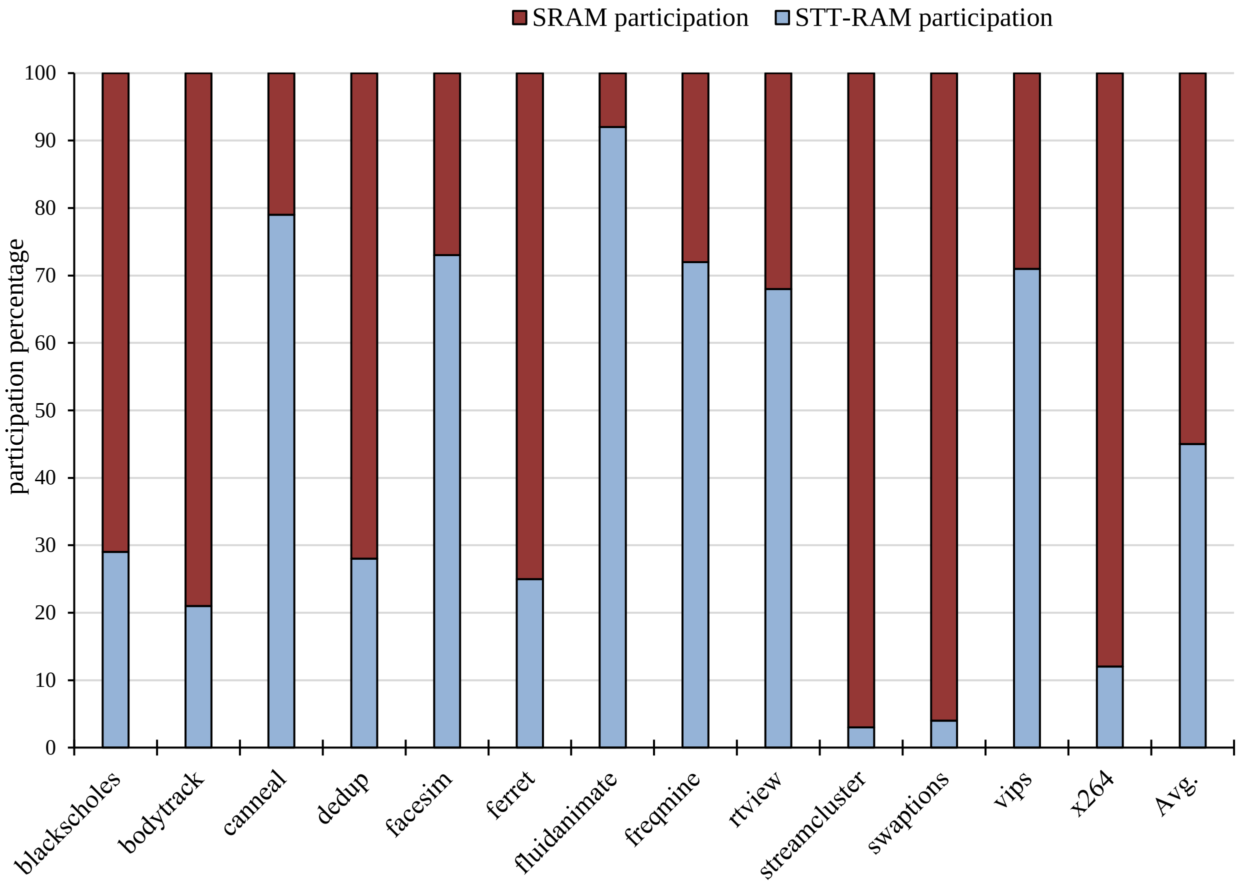


Figure 4.6: Participation percentages of SRAM and STT-RAM LLC in the hybrid architecture

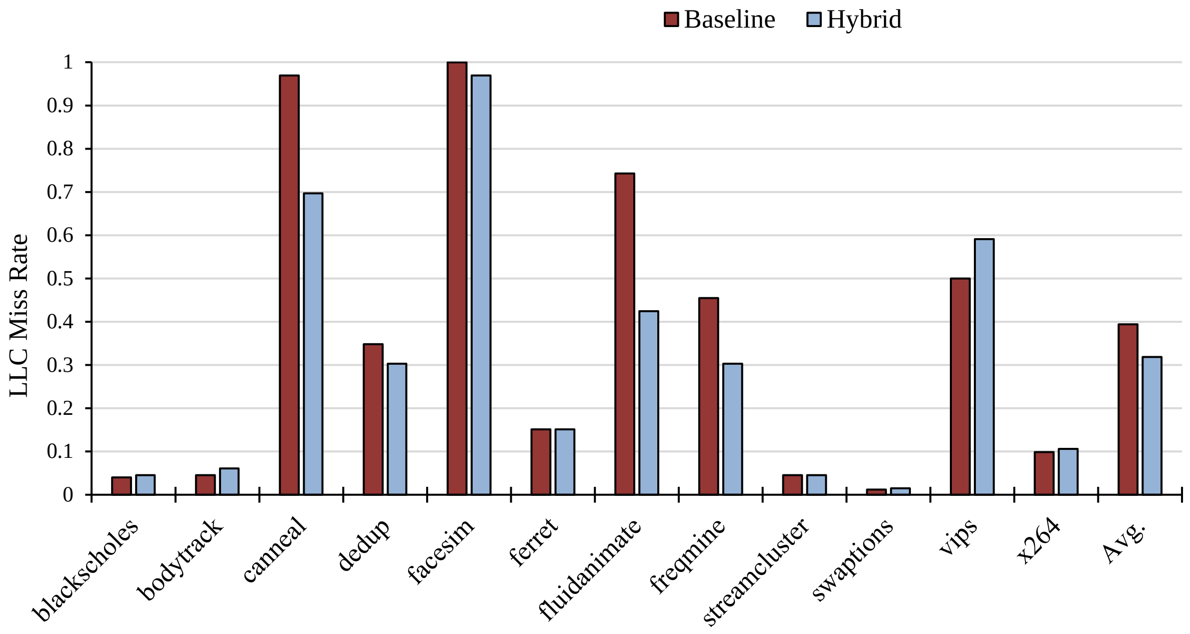


Figure 4.7: Comparison of the normalized Miss Rate in the LLC between the proposed and baseline architectures

Figure 4.8 shows improvement on average of 18 % in the miss rate in the proposed hybrid architecture below the baseline one. The change in the type of LLC bank results in the invalidation of the bank's content and the LLC memory filling stage in that bank when it is restored. Consequently, the increase in LLC capacity, leads to an increase in the miss rate for some applications. In benchmarks applications such as streamcluster and swaptions, where the participation percentage of STT-RAM banks is very low, the time was not sufficient to offset the increase in the rate of miss rate due to the change in the type of the banks.

However, the reduction of the read miss rate does not mean improving system performance, since the negative impact of long writing latency of STT-RAM banks can affect this improvement by adding more time to the average memory access time and consequently, the system performance. For an instant, debug benchmark application in figure 4.7, the proposed model improved about 9 % of less miss rate compared to baseline, whereas, due to the nature of writing access in this benchmark application, the average memory access time has not improved, it degrades down to 4.3% longer than the baseline one.

Average memory access time (AMAT) for each of the benchmark applications of both baseline the proposed hybrid model is illustrated in figure 4.8, where the results were normalized to the canneal application which has the best enhancement over all the rest of the benchmark applications. According to this chart, despite an 18% improvement in the Miss Rate, only 4.4% improvement is recorded in AMAT as explained before due to cost of the overhead computation, long writing latency of the STTRAM, cold-state Miss rate with the DVFS of the uncore area all these factors contributed to reducing the improvements of the AMAT to that level which will have a modest impact on applications time and consequently on their performance.

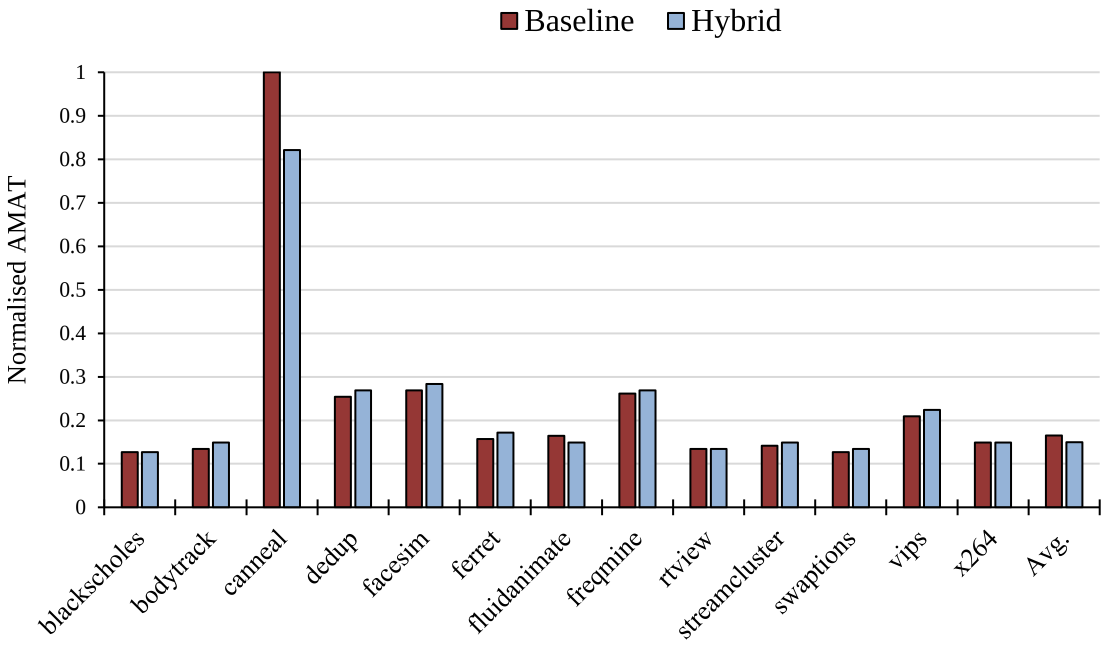


Figure 4.8: Comparison of the normalized AMAT between the proposed and baseline architectures

The change in the type of LLC banks from SRAM to STT-RAM will lead to a reduction in leakage power in the system, this reduction depends on the participation percentage of STT-RAM technology from the LLC. The high power of writing interval in STT-RAM technology and the nature of high-level writing access in benchmark applications is another factor to differentiate the power saving in each of these applications. For example, according to Figure 4.6, the canneal benchmark application has low write access rates of about 10%, which, along with the high percentage STT-RAM technology of about 80%, could significantly reduce power consumption by 56%. On the other hand, with more than 90% of STT-RAM's participation in the fluidanimate benchmark, due to the high rate of write access, it has a lower power consumption than the canneal benchmark of about 33%. This is evident in Figure 4.9. In addition, the proposed method reduced overall power consumption by an average of 43.1%.

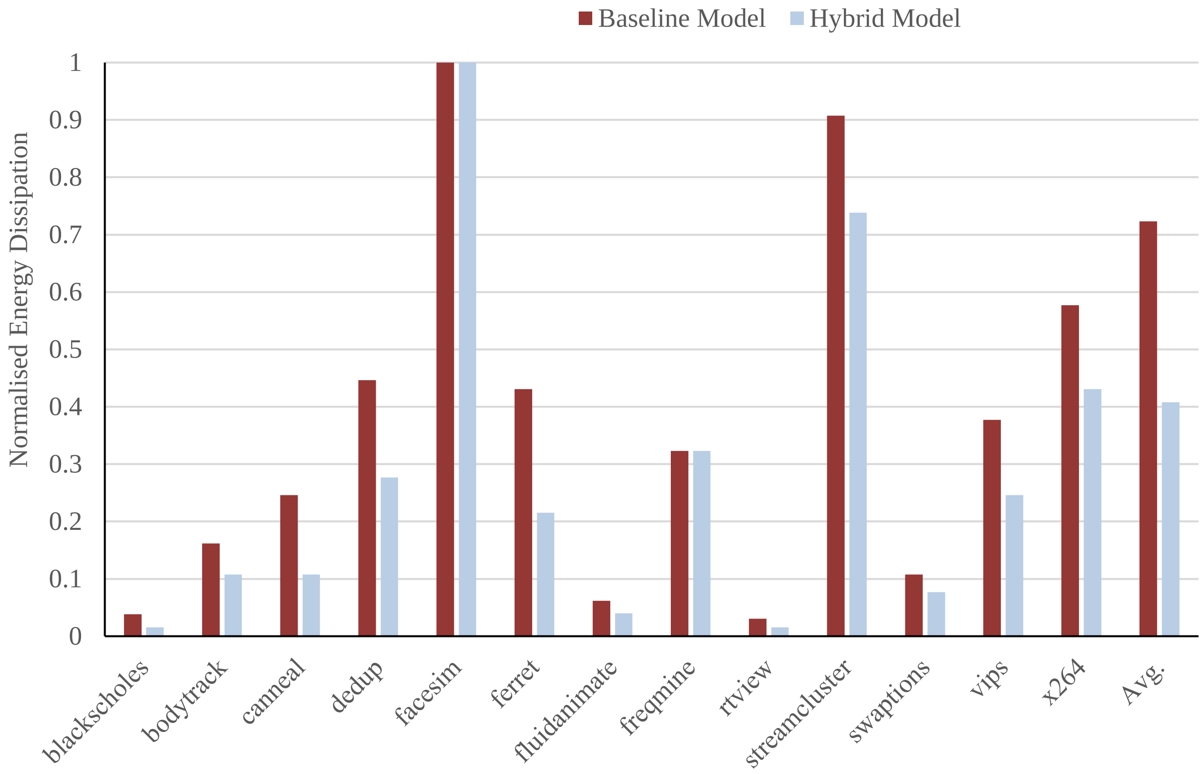


Figure 4.9 Comparison of the normalized total energy consumption between the proposed and baseline architectures

Finally, in order to ensure the optimal design of the proposal from the point of view of power consumption and system performance, the energy-delay product (EDP) parameter has been compared for both architectures. This parameter states that reducing system power consumption can be worth the degradation of its performance. EDP of the hybrid proposed model was improved in an average of 40% below that of the baseline model as shown in figure 4.10. the highest improvement was recorded for the fluidanimate at 54.5% below the baseline architecture and the least EDP improvement as expected was for blackscholes at 15%.

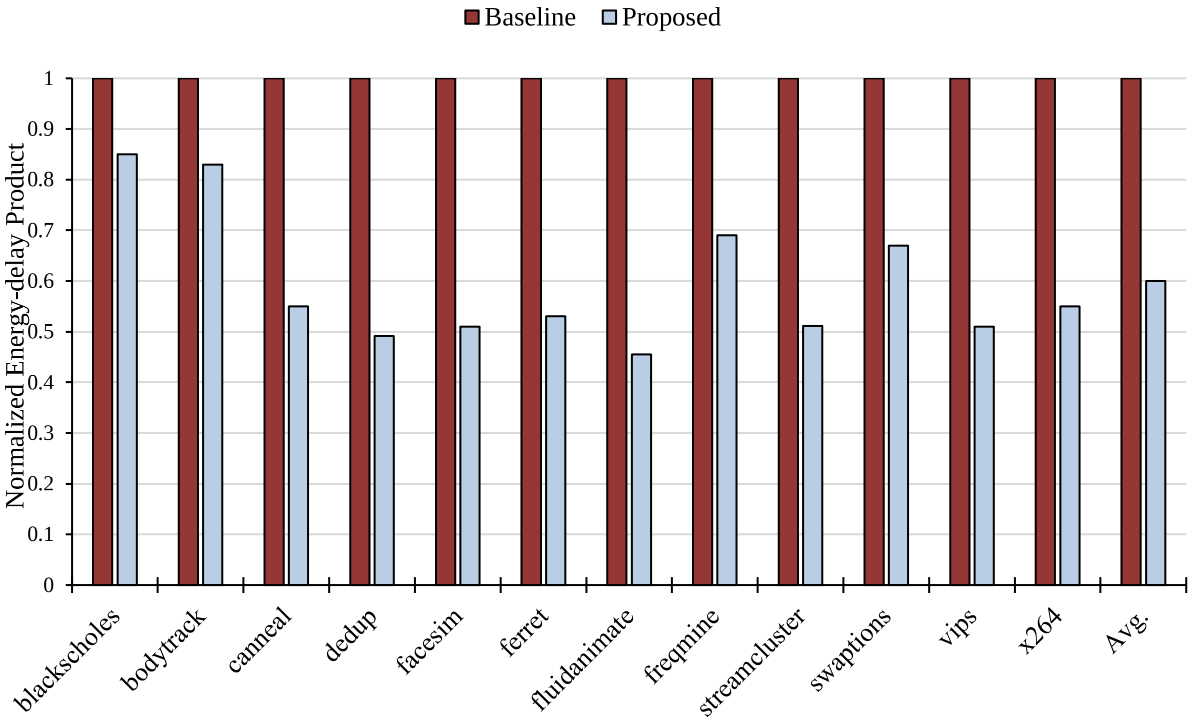


Figure 4.10: Comparison of the normalized energy-delay product between the proposed and baseline architectures

4.4 Summary

In this chapter, the proposed method which integrates the STT-RAM and SRAM technologies to construct hybrid reconfigurable LLC memory of the 3D CMP was investigated for several parameters. The simulation of the baseline model that uses only SRAM LLC was compared with the proposed hybrid LLC by adding DVFS on the uncore area. Based on these simulations, the proposed architecture improved the energy-delay product on average of 40 % below the baseline one. Performance measured in instruction per cycle, of the proposed model was affected marginally with only -1.2% on average below the baseline one, with worse degradation of 15%. Interestingly, two applications show performance enhancements due to the larger LLC capacity which reduced the Miss Rate. The variation of the results among the used benchmark application indicates the dependency of the results on the nature of the applications.

Chapter 5

Conclusions and Suggested Future Works

Improve power consumption in multi-core processors with minimal degradation of system performance is the main goal of this thesis. The objective of the research work was to reduced power consumption at the uncore area of the chip of multiprocessors. This reduction was aimed at integrating hybrid cache bank of STTRAM with the conventional SRAM ones and scaling the Voltage/Frequency based on the network status. The purpose of the hybrid cache bank is to combine the benefits of near-zero leakage power, higher density and powerful scalability with fast read speed of the STTRAM technology with fast writing cycle and low dynamic power consumption of the SRAM. DVFS was applied whenever the uncore has less demand from the network using the AMAT as our selected network monitor.

5.1 Conclusions

Through this research, the dynamic and leakage power consumption shows a critical issue concern in the CMPs. Our results of the proposed hybrid architecture yield an overall reduction in energy-delay product over all the benchmark applications with an average of 40% below the baseline architecture. The results of the performance show a marginal reduction on average of -1.2% degradation and the worse degradation was 15%. Nature of application was the critical element which changed the response of the proposed architecture, with more memory intensive application hybrid system enhances the performance with larger capacity last level cache and the performance increased compensating the slower cache, cold state Miss rate and the cost of overhead computation time.

5.2 Suggested Future Works

With the increase in the number of integrated processors on the chip, demand for the hierarchy of cache memory will increase, which in turn will lead to an increase in the amount of memory latency and power consumption. Thus, uncore area power consumption particularly the last level cache memory will become an important element in terms of CMP power consumption. There are several suggestions to extend this work toward more energy-efficient uncore at CMP level:

- Applying Machine Learning (ML) methods to reduce the overhead computation which uses a training session to find the optimal hybrid architecture memory with the minimum power consumption to perform the accurate LLC configuration and voltage frequency scaling at runtime. This can be performed by using modern ML techniques as Neural Network at different levels of hierarchy (per-uncore domain or per-uncore clusters if uncore is divided into different clusters.
- The other possible improvements that can be achieved by changing some of the utilized techniques in this work, for example, instead of changing one SRAM bank to STT-RAM each computation cycle, two banks can be carried out. Another alternative could be the algorithm criterion to assign the LLC bank number to be changed as a function of the difference between the calculated AMAT from the reference AMAT for an instant, the lower difference only one bank to be reconfigured and two banks for the higher difference this will reduce the impact on the system when it's close to the reference status and speed up the change whenever its possible.
- As mentioned in the literature review, SRAM buffers were used to enable the STTRAM utilization in the upper cache memory (core private cache L1), the same technique could be used in the last level cache STT-RAM. This is a costly design approach may be sound reasonable when larger capacity is designed specifically to handle the high rate of write access.
- The last suggestion for future work is to take on the consideration of the routing algorithm. In this work it was assumed that routing is deterministic, and the latency related to the routing has no effect on the calculated AMAT as it has the same effect on the proposed and baseline architectures for that it will not add a difference on the comparison between their results. With adaptive routing, packets are traveling in a different route and the latency will be a function for the network status and consequently the AMAT will be affected due to its dependence on the latency of the packets traveling throughout the network.

Bibliography

[1] Y.-T. Chen, J. Cong, H. Huang, B. Liu, C. Liu, M. Potkonjak, and G. Reinman, “Dynamically reconfigurable hybrid cache: An energy-efficient last-level cache design,” *in Proceedings of the Conference on Design, Automation and Test in Europe, pp. 45–50, EDA Consortium,* 2012.

[2] C. Bienia, S. Kumar, J. P. Singh, and K. Li, “The parsec benchmark suite: Characterization and architectural implications,” *in Proceedings of the 17th international conference on Parallel architectures and compilation techniques, pp. 72–81, ACM,* 2008.

[3] J. Wang, X. Dong, and Y. Xie, “OAP: An obstruction-aware cache management policy for stt-ram last-level caches,” *in Proceedings of the Conference on Design, Automation and Test in Europe, pp. 847–852, EDA Consortium,* 2013

[4] H. Sun, C. Liu, W. Xu, J. Zhao, N. Zheng, and T. Zhang, “Using magnetic RAM to build low-power and soft error-resilient L_1 cache,” *IEEE Transactions on Very Large-Scale Integration (VLSI) Systems, vol. 20, no. 1, pp. 19–28,* 2010.

[5] S. M. P. Variable, "Variable smp - a multi-core CPU architecture for low power and high performance*”,https://www.nvidia.com/content/PDF/tegra_white_papers/tegra-whitepaper-0911b.pdf"* 2011.

[6] N. Binkert, B. Beckmann, G. Black, S. K. Reinhardt, A. Saidi, A. Basu, J. Hestness, D. R. Hower, T. Krishna, S. Sardashti, et al., “*The gem5 simulator,” ACM SIGARCH Computer Architecture News, vol. 39, no. 2, pp. 1–7,* 2011.

[7] S. Li, J. H. Ahn, J. B. Brockman, and N. P. Jouppi, “Mcpat 1.0: An integrated power, area, and timing modeling framework for multicore architectures,” *HP Laboratories, Technical Report HPL-2009-206,* 2009.

[8] N. Muralimanohar, R. Balasubramonian, and N. P. Jouppi, “*Cacti 6.0: A tool to under- stand large caches,” the University of Utah and Hewlett Packard Laboratories, Tech. Rep, vol. 147,* 2009.

[9] S. Pagani, P. D. S. Manoj, A. Jantsch and J. Henkel, "Machine Learning for Power, Energy, and Thermal Management on Multicore Processors: A Survey," *in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 39, no. 1, pp. 101-116,* 2018.

[10] A. Dorostkar, A. Asad, M. Fathy, M. R. Jahed-Motlagh, and F. Mohammadi, “Low- power heterogeneous uncore architecture for future 3D chip-multiprocessors,” *ETRI Journal, vol. 40, no. 6, pp. 759–773,* 2018.

[11] S. Niknam, A. Asad, M. Fathy, and A.-M. Rahmani, “Energy-efficient 3d hybrid processor-memory architecture for the dark silicon age,” *in 2015 10th International Symposium on Reconfigurable Communication-centric Systems-on-Chip (ReCoSoC), pp. 1– 8, IEEE,* 2015.

[12] B. Hoefflinger, “ITRS: The international technology roadmap for semiconductors,” *pp. 161–174, Springer, 2011.*

[13] J. Power, A. Basu, J. Gu, S. Puthoor, B. M. Beckmann, M. D. Hill, S. K. Reinhardt, and D. A. Wood, “Heterogeneous system coherence for integrated CPU-GPU systems,” *in Proceedings of the 46th Annual IEEE/ACM International Symposium on Microarchitecture, pp. 457–467,,* 2013.

[14] M. J. Schulte, M. Ignatowski, G. H. Loh, B. M. Beckmann, W. C. Brantley, S. Gurumurthi, N. Jayasena, I. Paul, S. K. Reinhardt, and G. Rodgers, “Achieving exascale capabilities through heterogeneous computing*,” IEEE Micro, vol. 35, no. 4, pp. 26–36,* 2015.

[15] J. Hestness, S. W. Keckler, and D. A. Wood, “GPU computing pipeline inefficiencies and optimization opportunities in heterogeneous CPU-GPU processors”, *in 2015 IEEE International Symposium on Workload Characterization, pp. 87–97, IEEE,* 2015.

[16] B. K. Joardar, R. G. Kim, J. R. Doppa, P. P. Pande, D. Marculescu, and R. Marculescu, “Learning-based application-agnostic 3D NOC design for heterogeneous manycore systems,” *IEEE Transactions on Computers, vol. 68, no. 6, pp. 852–866,* 2018.

[17] W. R. Davis, J. Wilson, S. Mick, J. Xu, H. Hua, C. Mineo, A. M. Sule, M. Steer, and P. D. Franzon, “Demystifying 3D ICS: The pros and cons of going vertical,” *IEEE Design & Test of Computers, vol. 22, no. 6, pp. 498–510,* 2005.

[18] A. Al Maashri, G. Sun, X. Dong, V. Narayanan, and Y. Xie, “3D GPU architecture using cache stacking: Performance, cost, power and thermal analysis,” *in 2009 IEEE International Conference on Computer Design, pp. 254–259,* 2009.

[19] C. C. Liu, I. Ganusov, M. Burtscher, and S. Tiwari, “Bridging the processor-memory performance gap with 3d ic technology,” *IEEE Design & Test of Computers, vol. 22, no. 6, pp. 556–*

564, 2005.

[20] S. Das, J. R. Doppa, P. P. Pande, and K. Chakrabarty, "Design-space exploration and optimization of an energy-efficient and reliable 3-d small-world network-on-chip", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 36, no. 5, pp. 719–732,* 2016.

[21] A. Bakhoda, J. Kim, and T. M. Aamodt, "Throughput-effective on-chip networks for manycore accelerators," *in Proceedings of the 2010 43rd annual IEEE/ACM international symposium on microarchitecture, pp. 421–432, IEEE Computer Society,* 2010.

[22] H. Jang, J. Kim, P. Gratz, K. H. Yum, and E. J. Kim, "Bandwidth-efficient on-chip interconnect designs for gpgpus", *52nd ACM/EDAC/IEEE Design Automation Conference (DAC),* pages 1-6, 2015.

[23] A. K. Ziabari, J. L. Abell´an, Y. Ma, A. Joshi, and D. Kaeli, "Asymmetric NOC architectures for GPU systems," *in Proceedings of the 9th International Symposium on Networks-on-Chip, Article No.: 25 pages. 1-8,* 2015.

[24] K. Kaur and A. Noor, "Power estimation analysis for CMOS cell structures," *International Journal of Advances in Engineering & Technology, vol. 3, no. 2, p. 293,* 2012.

[25] H. Shen, "Adaptive Power Management for Computers and Mobile Devices*". PhD thesis, Syracuse University,* 2014.

[26] K. Ma, X. Wang, and Y. Wang, "Dppc: dynamic power partitioning and control for improved chip multiprocessor performance*," IEEE Transactions on Computers, vol. 63, no. 7, pp. 1736–1750,* 2013.

[27] J. Lin, H. Zheng, Z. Zhu, H. David, and Z. Zhang, "Thermal modeling and management of DRAM memory systems", *ACM SIGARCH Computer Architecture News [0163-5964] vol. 35,iss:2 page 312,* 2007.

[28] T. Simunic, L. Beniani, and G. De Micheli, "Event-driven power management of portable systems," *in Proceedings 12th International Symposium on System Synthesis, pp. 18–23,* 1999.

[29] C. Top500, "Tianhe-2 supercomputer takes no. 1 ranking on 41st top500 list.",

https://www.top500.org/news/lists/2013/06/press-release/

[30] W. Kim, M. S. Gupta, G.-Y. Wei, and D. Brooks, “System-level analysis of fast, per-core DVFS using on-chip switching regulators,” *in 2008 IEEE 14th International Symposium on High-Performance Computer Architecture, pp. 123–134,* 2008.

[31] W.-Y. Liang and P.-T. Lai, “Design and implementation of a critical speed-based dvfs mechanism for the android operating system,” *in 2010 5th International Conference on Embedded and Multimedia Computing, pp. 1–6,* 2010.

[32] P. Bogdan, R. Marculescu, S. Jain, and R. T. Gavila, “An optimal control approach to power management for multi-voltage and frequency islands multiprocessor platforms under highly variable workloads,” *in 2012 IEEE/ACM Sixth International Symposium on Networks-on-Chip, pp. 35–42,* 2012.

[33] T. Kolpe, A. Zhai, and S. S. Sapatnekar, “Enabling improved power management in multicore processors through clustered dvfs,” in 2011 Design, Automation & Test in Europe, pp.1-6, 2011.

[34] C. M. Kamga, “CPU frequency emulation based on DVFS,” *ACM SIGOPS Operating Systems Review, vol. 47, no. 3, pp. 34–41,* 2013.

[35] M.-F. Chang and W.-Y. Liang, “Learning-directed dynamic voltage and frequency scaling for computation time prediction*,” in 2011 IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications, pp. 1023–102*9, 2011.

[36] M. Otoom, P. Trancoso, H. Almasaeid, and M. Alzubaidi, “Scalable and dynamic global power management for multicore chips*,” in Proceedings of the 6th Workshop on Parallel Programming and Run-Time Management Techniques for Many-core Architectures, pp. 25–30, ACM,* 2015.

[37] A. K. Singh, M. Shafique, A. Kumar, and J. Henkel, “Mapping on multi/many-core systems: a survey of current and emerging trends,” *in 2013 50th ACM/EDAC/IEEE Design Automation Conference (DAC), pp. 1–10,* 2013.

[38] M. Becchi and P. Crowley, “Dynamic thread assignment on heterogeneous multiprocessor architectures,” *in Proceedings of the 3rd conference on Computing frontiers, pp. 29– 40, ACM,* 2006.

[39] J. Wang, Z. Chen, J. Guo, Y. Li, and Z. Lu, “Aco-based thermal-aware thread-to-core

mapping for dark-silicon-constrained cmps," *IEEE Transactions on Electron Devices, vol. 64, no. 3, pp. 930–937,* 2017.

[40] A. Radhamani and E. Baburaj, "Research on power optimization techniques for multi core architectures*," in International Conference on Advances in Computing and Communications, pp. 172–181, Springer,* 2011.

[41] J. Zhao, C. Xu, and Y. Xie, "Bandwidth-aware reconfigurable cache design with hybrid memory technologies," *in Proceedings of the International Conference on Computer-Aided Design, pp. 48–55, IEEE Press,* 2011.

[42] A. Jadidi, M. Arjomand, and H. Sarbazi-Azad, "High-endurance and performance- efficient design of hybrid cache architectures through adaptive line replacement," *in Proceedings of the 17th IEEE/ACM international symposium on Low-power electronics and design, pp. 79–84, IEEE Press,* 2011.

[43] Y. Turakhia, B. Raghunathan, S. Garg, and D. Marculescu, "Hades: Architectural synthesis for heterogeneous dark silicon chip multi-processors*," in Proceedings of the 50th Annual Design Automation Conference, p. 173, ACM,* 2013.

[44] B. Raghunathan, Y. Turakhia, S. Garg, and D. Marculescu, "Cherry-picking: exploiting process variations in dark-silicon homogeneous chip multi-processors*," in 2013 Design, Automation & Test in Europe Conference & Exhibition (DATE), pp. 39–44, IEEE,* 2013.

[45] J. Allred, S. Roy, and K. Chakraborty, "Designing for dark silicon: a methodological perspective on energy-efficient systems," *in Proceedings of the 2012 ACM/IEEE international symposium on Low power electronics and design, pp. 255–260, ACM,* 2012.

[46] J. M. Allred, S. Roy, and K. Chakraborty, "Dark silicon aware multicore systems: Employing design automation with architectural insight," *IEEE Transactions on Very Large-Scale Integration (VLSI) Systems, vol. 22, no. 5, pp. 1192–1196,* 2013.

[47] Y. Xie, "Modeling, architecture, and applications for emerging memory technologies," *IEEE Design & test of computers, vol. 28, no. 1, pp. 44–51,* 2011.

[48] M. K. Qureshi, V. Srinivasan, and J. A. Rivers, "Scalable high-performance main memory system using phase-change memory technology," *in ACM SIGARCH Computer Architecture News, vol. 37, pp. 24–33,* 2009.

[49] A.J. Chenginimattom, D. P John "Methods for Reducing the Activity Switching Factor*" in International Journal of Engineering Research and Development, Volume 11, Issue 03, pages.17-25,* 2015.

[50] J. Zhao, C. Xu, and Y. Xie, "Bandwidth-aware reconfigurable cache design with hybrid memory technologies," *in Proceedings of the International Conference on Computer-Aided Design, pp. 48–55, IEEE Press,* 2011.

[51] M. B. Taylor, "Is dark silicon useful? harnessing the four horsemen of the coming dark silicon apocalypse*," in DAC Design Automation Conference 2012, pp. 1131–1136, IEEE,* 2012.

[52] K. Swaminathan, E. Kultursay, V. Saripalli, V. Narayanan, M. T. Kandemir, and S. Datta, "Steep-slope devices: From dark to dim silicon," *IEEE Micro, vol. 33, no. 5, pp. 50–59*, 2013.

[53] X. Dong, C. Xu, Y. Xie, and N. P. Jouppi, "Nvsim: A circuit-level performance, energy, and area model for emerging nonvolatile memory," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 31, no. 7, pp. 994–1007*, 2012.

[54] M. Palesi, D. Patti, and F. Fazzino, "Noxim," Network-on-Chip Simulator," *http://sourceforge.net/projects/noxim/,* 2008.

[55] Jung, Jinwook, Yohei Nakata, Masahiko Yoshimoto, and Hiroshi Kawaguchi. "Energy-efficient Spin-Transfer Torque RAM cache exploiting additional all-zero-data flags*." In Quality Electronic Design (ISQED), 2013 14th International Symposium on, pp. 216-222. IEEE,* 2013.

[56] Zhou, Ping and Zhao, Bo and Yang, Jun and Zhang, Youtao "Energy reduction for STT-RAM using early write termination," *Proceedings of the 2009 International Conference on Computer-Aided Design,ACM, pp. 264-268,* 2009.

[57] A. Hadeed and F. Mohammadi "Energy Efficient Hybrid LLC in Future 3D CMP" *IEEE 9th Latin American Symposium on Circuits & Systems (LASCAS),* 2018.

[58] J. Duato. "A new theory of deadlock-free adaptive routing in wormhole networks", *IEEE Transactions on Parallel and Distributed Systems., 4(12):1320 –1331, Dec.* 1993

[59] G. Ascia, V. Catania, M. Palesi, and D. Patti. "Neighbors-on-path: A new selection strategy for on-chip networks." *In 2006 IEEE/ACM/IFIP Workshop on Embedded Systems for Real Time Multimedia, pp. 79-84. IEEE,* 2006.