An Energy Efficient 3D-Heterogeneous Main Memory Architecture for Mobile Devices

Deepak M. Mathew
Technische Universität Kaiserslautern
Germany
deepak@eit.uni-kl.de

Felipe S. Prado
Technische Universität Kaiserslautern
Germany

Éder F. Zulian
Technische Universität Kaiserslautern
Germany
zulian@eit.uni-kl.de

Christian Weis
Technische Universität Kaiserslautern
Germany
weis@eit.uni-kl.de

Muhammad Mohsin Ghaffar
Technische Universität Kaiserslautern
Germany
ghaffar@eit.uni-kl.de

Norbert Wehn
Technische Universität Kaiserslautern
Germany
wehn@eit.uni-kl.de

Matthias Jung
Fraunhofer Institute for Experimental
Software Engineering (IESE)
Germany
matthias.jung@iese.fraunhofer.de

ABSTRACT

The demand for main memory capacity is ever increasing in mobile devices and embedded systems. Dynamic Random Access Memories (DRAMs) can not keep pace with the required main memory capacities because of the restrictions in improving the cell density due to the slowdown in scaling and the high leakage power consumption. Contrary, emerging Non-Volatile Memories (NVMs), primarily Resistive Random Access Memories (RRAMs), offer a high scaling potential and consume less leakage power than DRAMs. However, they are not suitable to completely replace DRAMs as the main memory, owing to their large read and write access latencies and limited endurance. In this paper, we present the architecture of a novel heterogeneous 3D-stacked on-chip main memory system composed of DRAMs and RRAMs that can fulfill the memory capacity demands of future mobile devices. We evaluate the energy savings of the new architecture for several applications, including some emerging machine learning tasks on mobile devices, by conducting system-level simulations in gem5 using ARM CPU models. We explore and analyze the impacts of different hybrid memory organizations and data allocation policies on reducing the energy and total number of RRAM writes. On average, the new 3D-hybrid architecture consumes 73% lesser energy and 61% lower average power than a 2D-Hybrid memory architecture for applications from the PARSEC benchmark. For a neural network training application, the 3D-hybrid memory saves up to 60% energy in comparison with a DDR4 DRAM-only main memory.

MEMSYS 2020, September 28-October 1, 2020, Washington, DC, USA
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-8899-3/20/09…$15.00
https://doi.org/10.1145/3422575.3422786

CCS CONCEPTS

• **Hardware → Memory and dense storage**; **Non-volatile memory**; • **Computer systems organization → Heterogeneous (hybrid) systems**.

KEYWORDS

RRAM, DRAM, Hybrid Memory, On-device training, Edge AI

ACM Reference Format:
Deepak M. Mathew, Felipe S. Prado, Éder F. Zulian, Christian Weis, Muhammad Mohsin Ghaffar, Norbert Wehn, and Matthias Jung. 2020. An Energy Efficient 3D-Heterogeneous Main Memory Architecture for Mobile Devices. In *The International Symposium on Memory Systems (MEMSYS 2020), September 28-October 1, 2020, Washington, DC, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3422575.3422786

1 INTRODUCTION

The demand for memory capacity in mobile devices and embedded systems is increasing drastically. For example, on average, the DRAM capacity in smartphones has increased by 50 % over the past three years [37]. This growing requirement for memory capacity arises mainly from the increase in mobile applications and their memory footprint. Besides, today's mobile devices keep several applications in the DRAM in a compressed state when they are idle in order to reduce the application relaunch time (an important metric that affects the user experience) [9, 12]. Another factor that contributes to the growing demand for memory capacity is the ongoing paradigm shift from *cloud-centric computing* to *edge computing*, particularly in machine learning. Compared to the conventional cloud-centric machine learning where the data is acquired at the edge and computed in the cloud, computing at the edge has three major benefits: It requires less communication bandwidth, reduces latency, and improves data privacy [11]. Machine learning tasks, such as neural network inference and training, keep the network parameters and the data in DRAM. Therefore, the memory capacity requirements of smartphones and other edge devices will increase significantly in the future due to the widespread adoption of such tasks on mobile devices.

With the increasing memory capacity, primary concerns for such systems are the energy efficiency (since they are battery-driven) and the average power—due to tight Thermal Design Power (TDP). Studies have shown that DRAM can contribute $40 - 80$ % of the total energy consumption while performing machine-learning tasks on mobile devices [4, 46]. DRAM refresh is a significant contributor to the total energy that increases with its capacity [1]. The increased refresh energy at larger memory capacity will decrease the energy efficiency of mobile devices. On top of that, DRAM is facing severe scaling challenges in advanced technology nodes (< 20 nm) [22, 31], which limits the available memory capacity per area (density).

Due to the scaling challenges and the energy overhead of DRAMs, DRAM-based main memory will not be able to meet the requirements of future mobile devices and embedded systems concerning memory capacity and energy efficiency. On the other hand, emerging Nonâ€“Volatile Memories (NVM), such as Phase Change Memory (PCM) and Resistive Random Access Memory (RRAM), are promising alternatives to DRAM due to higher density, scalability, and no refresh energy. However, owing to their larger access latencies and lower endurance (maximum number of writes), they cannot fully replace DRAM in the main memory. Hence heterogeneous / hybrid main memory architectures composed of DRAM and NVM are already in research (and also recently adopted by industry—Intel's Optane DIMM [16], JEDEC's NVDIMM-P [18]). However, prior work on hybrid memories mainly focused on their use in High-Performance Computing (HPC) systems, and not on mobile devices. Besides, from the memory technology point of view, PCM has been extensively studied. Compared to PCM, RRAM has substantially lower write energy, and shorter write access time [53]. Furthermore, the materials used for an RRAM device (e.g., HfO_2 and TaO_2) are compatible with the existing CMOS process and thus enable easier integration. Therefore, in this paper we investigate on RRAM based hybrid main memory for mobile devices.

The requirements of mobile devices differ from that of HPC systems. First, *user experience* is an essential element in evaluating the performance of mobile devices. The quality of user experience is determined by factors such as user perceivable responsiveness (user response time) and smoothness of applications [48]. To ensure good quality user experience, specific tasks, such as image processing and real-time object recognition using neural networks, require main memory with low latency and high bandwidth—i.e., DRAM. Second, mobile devices have strong restrictions in terms of weight, size and form factor when compared to HPC systems. Therefore, current Dual In-line Memory Module (DIMM) based hybrid memories cannot fit into those systems. Third, mobile Multiprocessor System-on-Chip (MPSoC) have limited area and I/O pins. This restricts the off-chip memory bandwidth. Moreover, it necessitates DRAM and NVM sharing the same channel, causing interference with the performance-critical applications running on DRAM. Therefore, existing 2D-hybrid memory architectures are not suitable for mobile devices.

To meet the requirements mentioned above for mobile devices, we present a novel 3D-heterogeneous main memory architecture consisting of DRAM and RRAM. We discuss different use cases of the proposed architecture in mobile devices. The major new contributions of this paper are:

(1) An architecture of a novel 3D-heterogeneous main memory composed of DRAM and RRAM.
(2) Evaluation of the new 3D-architecture against a 2D-hybrid architecture.
(3) A detailed evaluation of the energy benefits and the performance impacts of the new architecture for different hybrid memory system configurations via full-system simulations for two selected machine learning applications on mobile devices.

2 RELATED WORK

In this section, we describe an overview of the prior work on 3D-stacked main memories and hybrid main memory systems.

2.1 3D-stacked DRAM and RRAM

There is plenty of previous research on on-chip main memories using 3D-DRAMs, which are summarized in [43]. They provide more bandwidth and energy efficiency compared to the off-chip DRAMs. Among the few studies on 3D-stacked main memory architectures using RRAM, Yu et al.[54, 55] proposed a monolithic 3D-RRAM composed of different tiers of 1T-1R memory arrays (cf. Section 3) that are fabricated over the processing tier. However, 1T-1R RRAMs, due to the density and scalability problems explained in Section 3, are less likely to replace DRAMs. Moreover, 3D-monolithic stacking of 1T-1R cells is technologically difficult due to the high-temperature processing steps needed for silicon transistors—temperatures > 1000°C required for fabricating transistors in the upper tiers can damage the interconnects in the bottom tier [36]. Therefore, it requires breakthrough technologies such as Carbon Nanotube FET (CNFET) or Molybdenum disulfide (MoS_2) FETs [42]. In a recent work [35], the authors propose N3XT, an energy-efficient abundant data computing system consisting of monolithic integrated compute tiers and CNFET-based RRAM memory tiers. We see this as a promising solution for the future if the technology will be mature.

In contrast to prior research that focuses only on 3D-RRAM, we propose a heterogeneous (hybrid) memory architecture composed of 3D-stacked DRAMs and RRAMs. We focus on crossbar memories, as they provide more capacity than 1T-1R RRAM. To the best of our knowledge, this is the first architecture of a 3D-integrated heterogeneous main memory composed of DRAMs and RRAMs.

2.2 Hybrid Memory System Architecture

Hybrid main memory consists of DRAM and NVM (usually PCM) has been in extensive research during the past decade.There exist majorly *two* hybrid memory organizations: *flat* and *hierarchical.*

In the *flat* memory organization, both the DRAM and NVM are available in the address space. Either the Operating System (OS) [13] or the memory controller [33] manages the allocation and migration of pages between these two memories during run-time. While the optimal data placement on memories is an NP-complete problem [57], a common approach is to monitor the NVM writes and migrate the write-intensive pages (warm pages) to (from) the

DRAM while keeping the cold pages in the NVM. The main drawback of this approach is that the tracking and migration overheads increase with the rise in memory accesses and with the increase in DRAM capacity. For example, when frequently read pages also have to be migrated to the DRAM because the NVM read latency is significantly higher than DRAM, the number of page migrations between DRAM and NVM increases drastically. Also, it increases the number of NVM writes as all the evicted pages from DRAM (even the unmodified) need to be written back to the NVM [50]. Another approach to manage the flat hybrid memory address space is to provide applications direct load / store access to the NVM—a new feature named as *DAX* in Linux and Windows OS [34]—and statically allocate data to the NVM or DRAM. Although this is a promising solution, it needs changes to the application software, requiring programmers aware of the memory allocation, which is abstracted in today's high-level programming languages, such as Python and Java. We evaluate this approach for the proposed architecture.

In the *hierarchical* memory organization, DRAM acts as a hardware managed cache to the NVM, and therefore not in the address space. Tags are stored in the SRAM or in the DRAM (in the same cache line along with the data). In the former case, large cache lines are used ($1 - 4$ KB) to reduce the SRAM capacity needed for tag storage [32]. However, large cache line sizes often lead to unwanted data movements between DRAM and NVM—thus, wasting bandwidth and energy—especially when the application has less spatial locality. On the other hand, when tags are stored in the DRAM, more fine granular cache line sizes (e.g., 64 B) can be used [28, 51, 52]. Therefore, we also evaluate this approach for the proposed architecture.

In contrast to prior work that focuses on 2D-hybrid memory with PCM and DRAM, we investigate the proposed 3D-hybrid memory with RRAM and DRAM. We demonstrate the benefits of the 3D-hybrid compared to the 2D-hybrid. Besides, we also investigate a mixture of the flat and hierarchical organizations—a hardware + software approach—by providing applications direct access to the RRAM and a portion of the DRAM, while using the remaining DRAM as hardware managed cache to the RRAM.

3 BACKGROUND

In this section, we provide the basics of RRAM memory technology and discuss the integration possibilities of a DRAM + RRAM hybrid memory into a Mobile SoC.

3.1 RRAM

A basic metal-oxide RRAM device consists of a Metal-Insulator-Metal (MIM) structure with the insulator layer composed of a binary or ternary transition metal oxide (e.g. HfO_2, TaO_2, $SrTiO_3$) [47]. The resistance state of the RRAM device, either a High-Resistance State (HRS) or a Low-Resistance State (LRS), is used to store logic 0 and logic 1, respectively. When writing a 1 to RRAM, known as the SET operation, it switches from HRS to LRS. When writing a 0 to RRAM, known as the RESET operation, the device switches from LRS to HRS. There exist different bitcell structures and array organizations of RRAM. A simple 1-*Transistor*-1-*Resistor (1T-1R)* RRAM bitcell consists of an RRAM memory device and a MOS transistor to access the memory device. In the memory array, all bitcells in a row are connected to a common wordline similar to the DRAM. This bit cell structure offers densities and read access latencies similar to the DRAM. However, the width of the access transistor does not scale along with the RRAM device scaling since the write current requirements of RRAM devices remain unchanged. Therefore, the bitcell area of this cell structure is limited by the access transistor. Moreover, each memory cell needs to be connected to the silicon substrate, making the vertical stacking of cells infeasible. Due to these drawbacks RRAMs with 1T-1R bitcell structure are not suitable for building high density memories [38].

Another RRAM cell structure is the 1-*Selector*-1-*Resistor (1S-1R)* bitcell. It allows building memory arrays in a crossbar organization [10] with the smallest bitcell area ($4F^2$), thus providing higher memory density than DRAM. Also, multiple crossbar arrays can be stacked vertically to form 3D-crossbar memory, thus further reducing the effective cell area to $< 4F^2$ [10]. Nevertheless, the access latencies of this bitcell structure is higher than that of DRAM, mainly due to the large turn on and turn off delays of the selector [19, 40]. Selector delays ranges from roughly 10 ns to 50 ns based on the selector properties and the switching mechanism [6]. Another RRAM array organization, which is the *3D-Vertical RRAMs (VRRAMs)* [7, 49], provides scalability in the vertical direction similar to NAND flash. However, the read latency of this configuration ($\sim$300ns) is also significantly higher than the typical DRAM read access latencies due to the very small sensing currents (< 100 nA) [25]. 3D-VRRAMs are well suited for replacing NAND flashs since they provide scalability in the vertical direction with less fabrication steps compared to 3D-crossbar memories.

Among the different memory array organizations of RRAM discussed above, crossbar arrays are more suitable for main memories since they provide a good trade-off between access latency and density. Therefore, RRAM crossbar memories are selected for further investigations in this paper. As of today, RRAM has several reliability issues, such as limited write endurance, which depends on the programming conditions of specific cells in the memory array. There are various circuit-level techniques and device-level optimizations to overcome these limitations. In this paper we assume a constant endurance for all cells in the memory array.

3.2 Hybrid Memory

In this section, we examine the various integration possibilities of the hybrid memory (DRAM + RRAM) into a mobile SoC. There are two options for integration: off-chip (2D) or on-chip (3D). In the 2D option, which is typical for HPC systems, either dedicated channels are added to the SoC for hybrid memory, or hybrid memories share the same channel with DRAM. Although this architecture is ideal for HPC systems because of its scalability, there are, however, three drawbacks to this choice concerning mobile devices. First, the pin limitations in mobile SoCs are more stringent because of the chip area constraints. This restricts the number of channels. Second, adding more pins will increase the average power consumption due to the power-hungry I/Os. Third, sharing the same channel with DRAM will cause interference with the performance-critical applications running on DRAM.

Table 1: Specifications of the DRAM and the RRAM Die

Parameter	DRAM Die	RRAM Die
Technology node (nm)	22	28
Capacity (Gb)	8 + 1 (for Cache TAG/ECC)	16
Die Size (mm)	9×11.5	11.7×12.6
Die Thickness (μm)	50	50
Number of Channels	4	4
Number of Banks per Channel	4	4
I/O width	72	4
Interface and Frequency	DDR, 500 MHz	DDR, 500 MHz
Burst Length (BL)	4	16

Hence, we argue for a 3D-hybrid memory architecture where the DRAM and RRAM dies are stacked on top of the SoC and connected via Through Silicon Via (TSV). We chose TSVs for our hybrid main memory architecture because of their wide adoption in the industry.

4 ARCHITECTURE

The 3D-architecture of the proposed main memory system is shown in Figure 1. It consists of two stacked DRAM dice, and four stacked RRAM dice. The bottom layer, which is the System on Chip (SoC) layer, includes 4 Hybrid Memory controllers (MCs). Our exemplary system is composed of multiple DRAM and RRAM dice, which can be stacked on top of each other. However, there are restrictions in freedom of the stacking order. As we can see in Figure 1a, if the DRAM is stacked on top of the RRAM, which would be from the perspective of heat transfer a good solution, the wider I/O data signals, power wires, and control lines of the DRAM have to be routed (via TSVs) through the RRAM die. This will diminish the usable area for the RRAM die. Therefore, we choose the option to stack the RRAM dice on top of the DRAMs as shown in Figure 1b. Furthermore, if the DRAM acts as a cache for the RRAM, more frequently accesses go to the DRAM and not to the RRAM, thus it is also an energy advantage.

Table 1 lists the parameters of the DRAM and the RRAM dice for the architecture shown in Figure 1b. Architectural specifications of the DRAM are based on the WIDE I/O 2 standard [17]. The detailed architecture of the DRAM die is depicted in Figure 2. Each channel has 72 bit wide data I/Os (instead of 64 bit in WIDE I/O 2) to the corresponding MC. The additional 8 bits are used for transferring the cache tag or ECC bits. We implemented a *split-bank* architecture—a bank is divided into two half-banks—to cope with the distributed wide I/O interface, similar to a WIDE I/O [21] or HBM DRAM [8]. Each half-bank can deliver 144 data bits, which matches to the half of the number of I/Os multiplied with the burst length ($36 \times 4 = 144$). Additionally, as we stack the RRAM die on top of the DRAM die, we have to provide areas for pass-through RRAM TSVs as shown in Figure 2.

The architecture of our RRAM die is based on the prototype chip from [23] at 24 nm technology. This prototype chip has a die capacity of 16 Gb per layer and a die area of 130.7 mm^2. In comparison, our RRAM die provides the same capacity (16 Gb) with an estimated area of 147.4 mm^2 (cf. Table 1) at 28 nm technology node. However, the design from [23] had used a NAND-Flash compatible interface. Since our design objective is to use the RRAM for main memory, the internal architecture and organization of the RRAM die resemble a typical DRAM.

Figure 3 shows the organization of the RRAM die. The 16 Gb die is divided into four channels. Each channel has 4 banks, and there are 16 banks per die ([23] has also 16 banks). A bank is internally organized as 32 blocks; each block consists of 32 RRAM crossbar arrays (sub-arrays) with the array size of 1 Mb (1024×1024). Various architectural design decisions were taken using the architectural exploration framework from [26]. For instance, the number of sub-arrays per block was limited to 32 as the global wire delays significantly increase beyond that, thereby increasing the read and write access latencies. Similarly, the crossbar array size of 1024×1024 was optimal for both read and write access latency as well as energy.

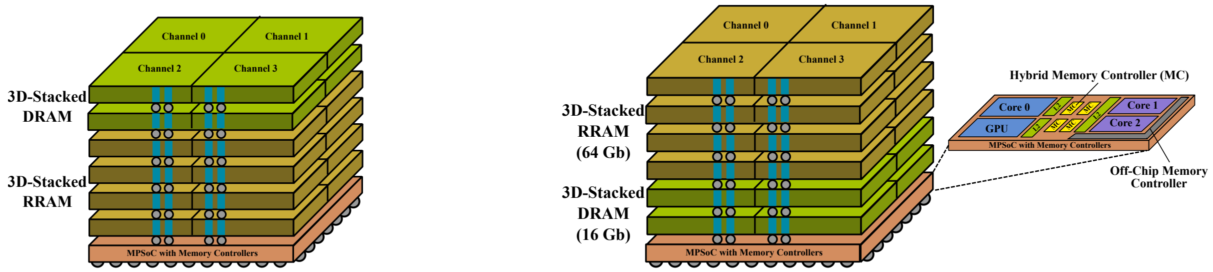
In contrast to the row-buffer (primary sense amplifiers) in DRAMs, we do not employ row-buffer in our RRAM architecture. This is because the limited number of sensing and writing circuits within a bank (area constraints) makes it necessary to perform several serial accesses to the sub-arrays (address looping) in order to fetch the data to / from the row buffer[1]. For instance, considering 64 sense amplifiers in one bank [23], an address looping of 64 is required to fill a row buffer of size 512B. This will drastically increase the latency and energy of a row activate operation.

In our architecture, we access only *one* bit of data from each crossbar array for the read and write operations. This has several benefits: First, it allows to perform a writing 1 (SET operation) and writing 0 (RESET operation) in a single step, thus achieving balanced read and write latencies, which is advantageous for main memory. Second, it allows reducing the write voltage of the crossbar array, as the amount of sneak current and the metal line voltage drop is significantly lower in a single-bit access compared to a multi-bit access. The reduction in write voltage decreases the write energy and improves the reliability of the array periphery (driver transistors). Also, the reduction in sneak currents during reads will improve the read sensing margin, enabling faster and more reliable sensing, especially when the difference between the RRAM resistance states (LRS and HRS) is minimal due to large variations.

Due to the single-bit access scheme, we fetch data from multiple sub-arrays in parallel. Therefore, in order to fetch 64 bits per bank (I/O width × burst length), 64 sub-arrays are activated—distributed over two sub-array blocks. To further increase the data transfer size to the last-level cache-line size (32 Bytes), the same bank spans vertically across different 3D-layers. E.g., bank 0 spans across the four vertical dice, 4 (dice)×4 (data lines) = 16 data lines finally leading to 16×16 (burst length) = 256 bits of data per access to an RRAM vertical channel. Hence this vertical channel is assembled like a vertical DIMM with four RRAM x4 (4 data lines) die parts forming an x16 (16 data lines) I/O channel to the corresponding MC on the MPSoC die (cf. Figure 1b).

The bottom MPSoC die in Figure 1b contains different processing cores (CPUs, GPU, accelerators), Level-1 (L1) and Level-2 (L2) SRAM caches, off-chip memory controller, and MCs. Each MC controls a single vertical DRAM channel and a single vertical RRAM channel. The details of MC and its different operation modes are described in Section 6. The off-chip memory controller provides regular applications the access to LPDDR4 / DDR4 off-chip memory.

[1] RRAM and PCM employ current sense amplifiers that occupy more area than the voltage sense amplifiers typically used in DRAM.



(a) Architecture in which RRAM is Stacked Above the Processing Layer (DRAM on top)

(b) Architecture in which DRAM is Stacked Above the Processing Layer (RRAM on top)

Figure 1: The Proposed 3D-Architectures of the Heterogeneous Memory System.

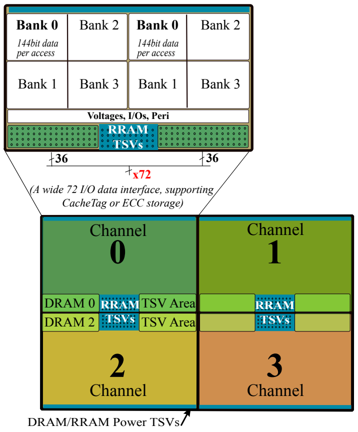


Figure 2: Architecture of the DRAM Die.

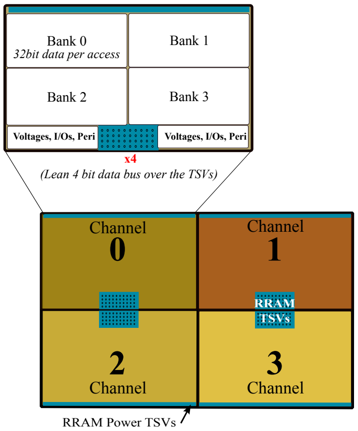


Figure 3: Architecture of the RRAM Die.

Different dice in the 3D-stack are connected using TSVs and microbumps. We assumed for the TSVs an implementation with a diameter of 8 μm and a pitch of 16 μm, which is aligned to the microbump pitch for connecting the different 3D-layers. The lumped capacitance we used in our evaluations was 100 fF for a single TSV itself, and additional 50 fF for the microbump and the I/O circuitries. The resistance for the TSV connection including the microbump was assumed to be 50 mΩ. This technology data was presented in several publications [29, 44]. We tried to minimize the number of TSVs in this design since the primary application target is the cost concerning mobile devices—the reliability of 3D-integrated design decreases with the increase in the number of TSVs and manufacturing costs drastically grows[41, 56].

5 APPLICATIONS

Applications on mobile devices can benefit from this architecture in three different ways: ① Emerging applications in the field of machine learning, such as neural network training, can benefit from the large memory capacity by accessing one or more channels of the on-chip hybrid main memory. ② The on-chip hybrid memory can be used as *zram* [14]: a swapping scheme used in mobile devices (since Android 4.4) in order to reduce memory usage by compressing the cold-pages and storing them in the in-DRAM swap space (instead of using secondary storage as swap). A major drawback of this scheme is the premature termination of processes due to the limited DRAM capacity in these devices, thus degrading the user experience [15]. By employing the on-chip hybrid memory as zram, this problem could be solved as more processes can remain in the main memory without the massive energy overhead of DRAM. Moreover, since the active processes are still in DRAM, the user experience will not be degraded. ③ Performance-critical applications can leverage this architecture by allocating data directly in the on-chip DRAM instead of accessing the slower and power-hungry off-chip DRAM. To limit the scope of this paper, we focus only on the first use case of this architecture in further discussions.

6 SIMULATION FRAMEWORK

This section presents the details of our simulation framework that enables the system-level explorations of RRAM based hybrid main memory architectures. Our framework supports both 2D- and 3D-hybrid memory systems. The major difference between these two systems concerning modeling is the communication protocol between the hybrid memory controller and the host CPU. The 2D-hybrid memory system was modeled based on the NVDIMM-P specifications of JEDEC [18]. In the 2D-hybrid memory system, the hybrid memory controller is situated on the non-volatile DIMM (NVDIMM)—i.e., outside the SoC. NVDIMMs typically share the

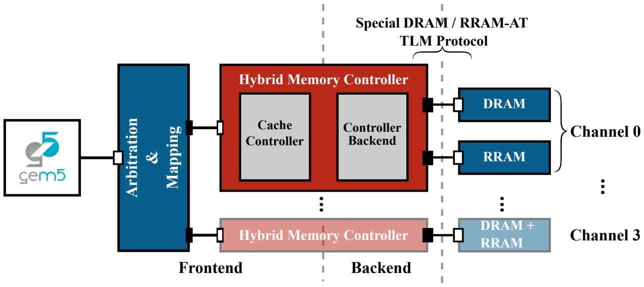


Figure 4: Structure of the Simulation Framework

same physical channel with DRAM DIMMs due to the pin limitations. Hence, a DDR4 / DDR5 compatible protocol is required for DIMM-based 2D-hybrid memory systems. We developed a custom protocol based on recent discussions on the JEDEC's NVDIMM-P specification. On the other hand, in the proposed 3D-hybrid architecture, hybrid memory controllers are placed within the SoC (cf. Figure 1). Hence they are directly connected to the on-chip bus (e.g., AXI).

Figure 4 shows the structure of our simulation framework. The framework consists of several SystemC TLM 2.0 models of the hybrid memory (DRAM and RRAM) and memory controllers that are coupled to the gem5 simulator, which serves as a realistic stimuli generator. The hybrid memory controller comprises a configurable frontend and a backend. The frontend consists of a single arbiter and cache controllers for each channel. It can be configured to three different operation modes representing the hybrid memory organizations discussed in Section 2. Those are: *cached* (hierarchical), *non-cached* (flat), and *semi-cached* (flat + hierarchical). In the cached operation mode, referred to as *3D-hybrid (3DH)*, the complete DRAM in each channel is used as write-back cache for the RRAM. Tags are stored in the DRAM along with the ECC bits similar to the approach employed in Intel's Knights Landing architecture [39] for the HBM+DRAM hybrid memory. The non-cached operation mode, referred to as *3DH-Direct Access (3DH-DA)*, provides a flat memory address space consists of both RRAM and DRAM. The cache is bypassed in this mode. In the semi-cached mode, referred to as *3DH-DA with Cache (3DH-DAC)*, half of the DRAM in each channel is configured as cache (similar to 3DH), whereas, the remaining half is available in the address space. We modified the physical page allocator of gem5 [3] so that the programmer can specify the preferred address allocation region (i.e., RRAM / DRAM) directly from the application, in both 3DH-DA and 3DH-DAC configurations. The arbitration and mapping block of the controller frontend coordinates with gem5 in forwarding the memory transactions to the specified address regions. In addition to the operation modes mentioned above, for the evaluations with RRAM only main memory, the controller supports masking the DRAM; we call this as *RRAM only mode (RRAM)*.

The backend consists of models of DRAM[20] and RRAM controllers. Each controller communicates with their respective memory model using an extended TLM 2.0 protocol with memory-specific phases. The framework is integrated into the gem5 full-system simulator to enable closed-loop simulations with the CPU.

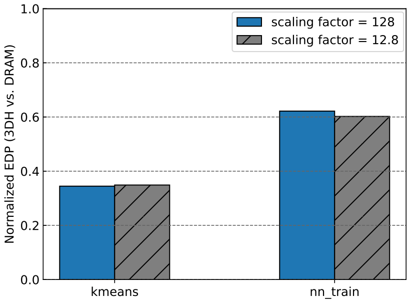


Figure 5: Accuracy Comparison of Two Scaled Systems.

Table 2: Specifications of the Simulated System.

Heterogeneous Memory	2GB RRAM: $t_{RL} - t_{CCDR} - t_{WL} - t_{CCDW} = 36 - 16 - 14 - 36\ ns$; $eRD - eWR = 903.6 - 2765.1\ pJ; P_{leak} = 2.7\ mW$. 512MB 3D-DRAM: $t_{RL} - t_{RCD} - t_{RP} - t_{RAS} = 12 - 14 - 14 - 24\ ns$; $I_{DD0} - I_{DD4R} - I_{DD4W} - I_{DD5} = 51 - 271 - 271 - 241\ mA$.
DDR4 DRAM	2GB DRAM: $t_{RL} - t_{RCD} - t_{RP} - t_{RAS} = 13.3 - 13.3 - 13.3 - 32.5\ ns$; $I_{DD0} - I_{DD4R} - I_{DD4W} - I_{DD5} = 243 - 738 - 675 - 472\ mA$.
Processing Core	ARM V8A HPI CPU, 2 cores, 64-bit, 2.0 GHz, L1 local cache (32 KB), Shared L2 cache (64 KB), 32-Byte Cache-line

7 EXPERIMENTAL METHODOLOGY

To improve the accuracy, all simulations were performed by executing applications on the detailed CPU model from ARM in the gem5 System call Emulation (SE) mode. However, improving the simulation accuracy will significantly increase the simulation time, particularly when simulating applications with a large memory footprint. Besides, when evaluating the hybrid memory configurations in which DRAM acts as a cache to the RRAM, it is essential to keep the memory footprint larger than the cache size. We kept the memory footprint 3X the DRAM cache size to simulate the worst-case scenario. A single channel of our hybrid memory consists of 2 GB RRAM and 512 MB DRAM. Since it was impossible to perform full-system simulations with such a large memory footprint in a reasonable time (< 1 week for each exploration), we simulated a scaled-down memory system with 4 MB DRAM and 16 MB RRAM (scaling factor of 128). Also, the processor cache sizes were reduced to the minimum size (cf. Table 2). It should be noted that only the memory capacities (i.e., the address space) were scaled, whereas, the timing and current specifications remained unchanged so that the accuracy of the experimental results are not affected. Figure 5 illustrates the accuracy comparison of two scaled hybrid systems for the evaluated machine learning applications. It shows that there is negligible variation in the accuracy of results from scaling the memory system in the way we described. Table 2 lists the major parameters of the simulated systems. The timings and energy specifications of DRAM and RRAM dice were generated using the architectural exploration frameworks [45] and [27], based on the architectural specifications in Table 1.

To compare the 2D- and 3D-hybrid systems (cf. Section 8.1), we executed several applications from the PARSEC CPU benchmark suite [2] and two machine learning applications—representing supervised (neural network training) and unsupervised (kmeans) learning. Since our goal is to investigate the 3D-hybrid memory architecture for machine learning on mobile devices, we focus on the machine learning applications for the comprehensive analysis of different hybrid memory controller configurations in Section 8.2. For evaluating our memory allocation policy in the hybrid memory controller configurations with flat address space (3DH-DA and 3DH-DAC), it was necessary to choose a C implementation of both applications since the C allows memory allocation through the *malloc* and *mmap* functions[2]. These functions are not directly accessible to the programmer in high-level programming languages, such as Python. This restrained us from selecting the standard benchmarks (which are written in Python or C++) for our evaluations. Therefore, we chose the C implementation of a neural network training application from a publically available git repository [5]. This application performs training on the MNIST data set for recognizing handwritten digits. It uses Stochastic Gradient Descent (SGD) as optimization function. Since we focus on transfer learning, our goal is not to train the network from scratch, instead, to improve the accuracy of a pre-trained network. Therefore, all simulations start with a pre-trained network with 75% accuracy and finish when the network achieves an accuracy of 82%. Although the selected network can achieve up to 92% accuracy in further training, we limit the training accuracy to reduce the simulation time. Likewise, for the kmeans application, we selected the C implementation from [30]. The goal of this specific application is to classify 8000 objects, each with 196 co-ordinates, into 3 clusters. We limited the number of iterations to 3 in order to complete the simulations within a reasonable time.

8 RESULTS

This section presents the experimental evaluation results of the new 3D-hybrid memory architecture and demonstrates its benefits over the 2D-hybrid memory. For evaluations of the 2D-hybrid memory, we used the PARSEC benchmark suite. Although this is a CPU benchmark not only focusing on embedded applications, we used this benchmark to compare the 2D architecture, which is the state-of-the-art for HPC, with the 3D architecture. For detailed assessments of the 3D-hybrid memory architecture in mobile devices, we used the emerging machine learning applications.

8.1 2D vs 3D-hybrid Memory

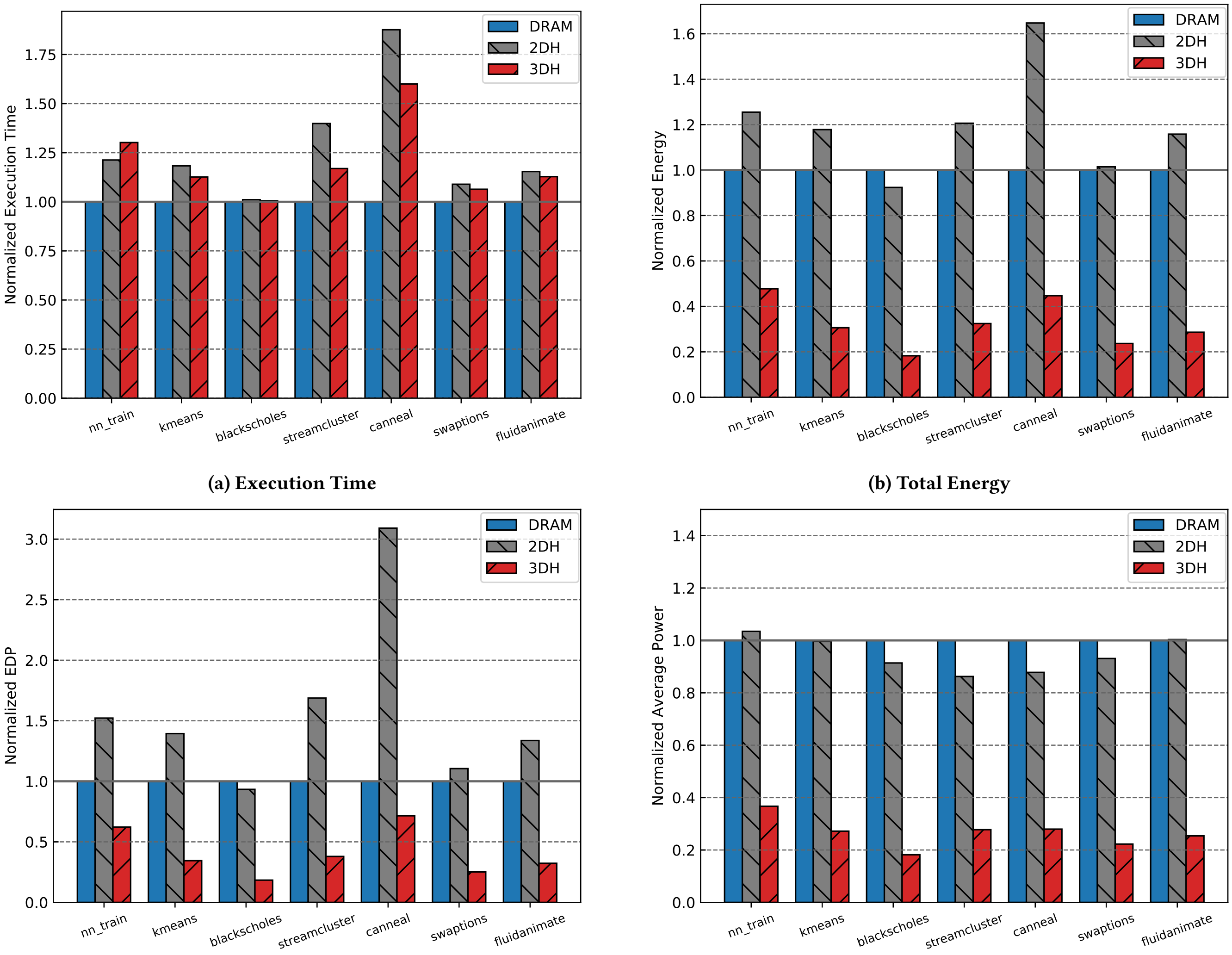
We first evaluate the benefits of the proposed 3D-hybrid main memory architecture against the 2D-hybrid memory and DDR4 DRAM. The 2D- and 3D-hybrid memory controllers operate in the cached mode—i.e., 2DH and 3DH, as described in Section 6. Figure 6 plots the experimental results for the applications described in Section 7 normalized to the DRAM baseline—i.e., main memory consists of only DDR4 DRAM. The results show that the 3DH system significantly reduces the energy and average power compared to the 2DH and the baseline. On average (across all applications), the 3DH consumes 73% and 68% lower energy compared to the 2DH and DDR4 DRAM, respectively. Similarly, the average power for 3DH is 61% and 74% lower than the 2DH and DDR4 DRAM. However, the performance degradation (cf. Figure 6a) is severe for some applications: e.g., the execution time of canneal is roughly 60% higher in 3DH compared to the baseline. The increase in execution time is due to the low cache hit ratio (50%). Nevertheless, the Energy Delay Product (EDP) for canneal is still 25% lower in 3DH compared to the baseline (cf. Figure 6c). This proves that the proposed 3D-hybrid architecture is better compared to the other two architectures. On the other hand, the 2DH system shows EDP higher than the baseline for most applications. This is due to the higher energy consumption of 2DH compared to the baseline (cf. Figure 6b), emanating from the memory accesses to the DDR4 DRAM cache. In the cache implementation, tags are stored in the DRAM instead of SRAM. Therefore, every access to the hybrid memory requires one DRAM access for the tag look-up even for a cache miss, thus increasing the energy overhead. Note here that there is no additional energy overhead for a hit, as the data comes along with the tag. Hence, the energy overhead of 2DH compared to the baseline is significantly higher for applications with low cache hit rate, e.g., in canneal. In contrast, accesses to the 3D-DRAM in 3DH configuration consume significantly lower power and energy compared to DDR4 DRAM. Hence this configuration has substantial power and energy savings compared to the 2DH and the baseline.

Nevertheless, it is worth mentioning that the benefits of 2DH will be significant when the memory system has capacity misses leading to page faults. Due to the limitations of our experimental set-up, we are unable to simulate this scenario. Therefore, we emulated an example scenario for the *stream* benchmark, on a Linux virtual machine with 4 GB DRAM and SSD based secondary storage. The application memory footprint was set to 3 GB. We executed the application first with 4 GB DRAM, and then with 3 GB DRAM (by reducing the virtual machine's memory capacity). In the former case, there were negligible capacity misses (swap space usage was 20 MB), while in the latter case, the OS starts using the swap space (increased to 800 MB) due to the lack of main memory capacity, resulting in frequent page faults. As shown in Figure 7, the average execution time of all stream kernels has increased by at least one order of magnitude when the application has to access the swap space (NAND Flash SSD) due to page faults. A hybrid main memory (both 2DH and 3DH) will drastically reduce the performance degradation in this scenario by leveraging the larger capacity (compared to DRAM) and fast access latency (compared to NAND Flash) of RRAM.

8.2 Evaluation of 3D-hybrid Memory Architecture

In this section, we conduct a detailed investigation of the proposed 3D-hybrid memory architecture for different memory controller configurations described in Section 6. Aforementioned in the previous section, the benefits of 3DH (reduction in energy and average power consumption) come with a marginal performance degradation. This might not be acceptable for the mobile applications that

[2] Although these functions access the logical address, we modified gem5 to make the physical memory allocator aware of the two address regions: DRAM and RRAM, based on the flags passed through these functions.



(a) Execution Time

(b) Total Energy

(c) Energy-Delay Product

(d) Average Power

Figure 6: Comparison of 2D- and 3D-Hybrid Memory Architectures with the DDR4 DRAM for various Applications

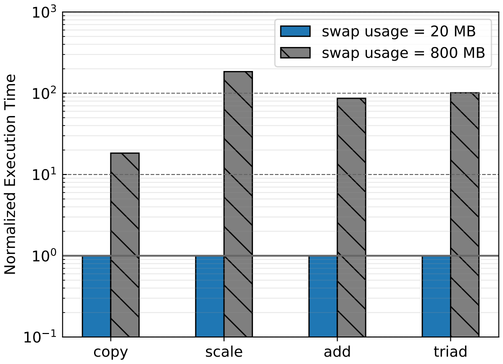
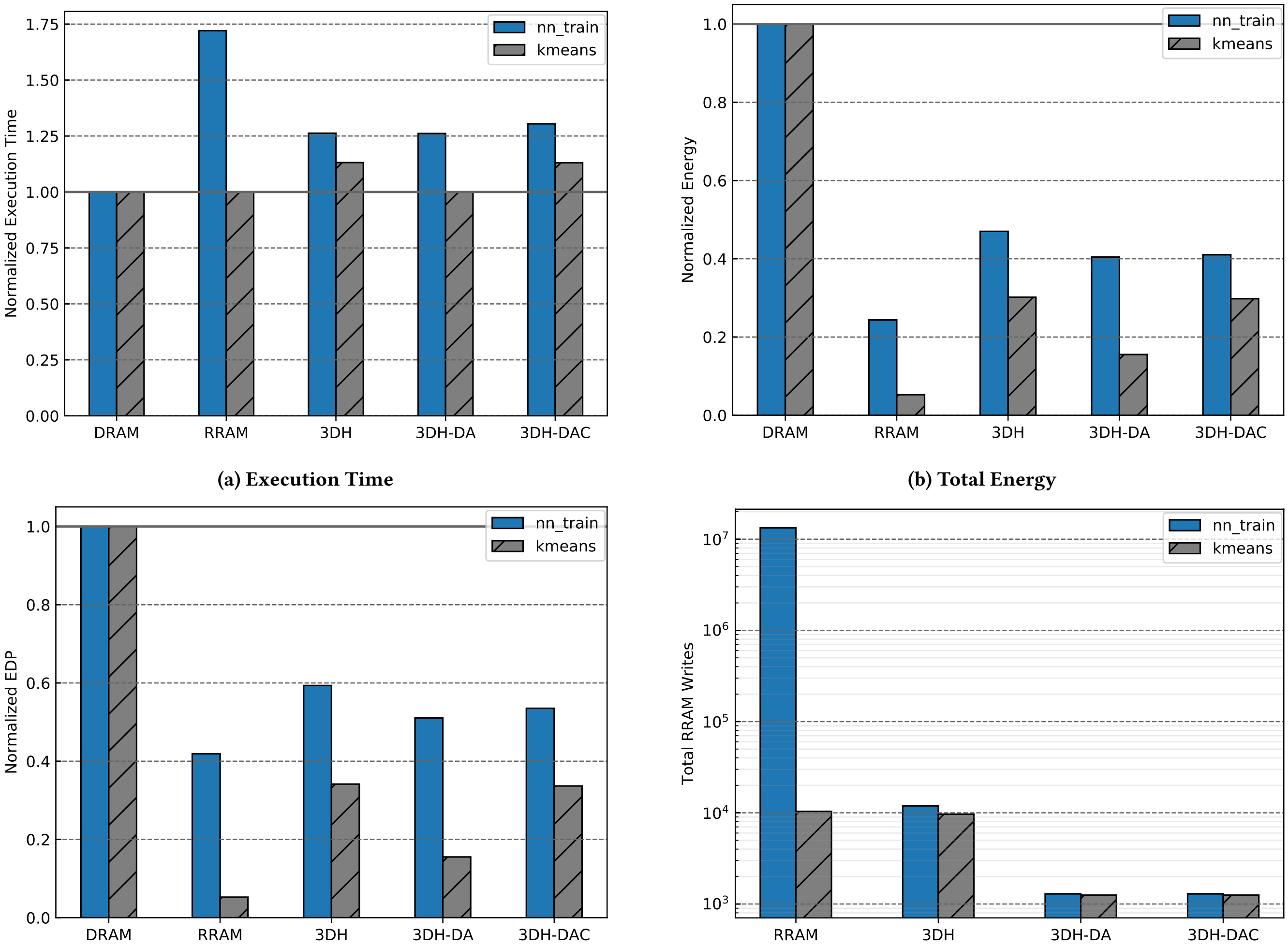


Figure 7: Performance Degradation due to Capacity Misses.

directly affect the user experience (e.g., image streaming, neural network inference), as mentioned in Section 1. Hence, the applications that demand large memory capacity and not directly influencing the user experience will be the preferred choice for the proposed 3D-hybrid memory on mobile devices. Therefore, we selected two emerging big data applications on mobile devices that satisfy these criteria. First, the neural network training (nn_train), which has much significance in mobile devices due to the recent advancements in Artificial Intelligence (AI) towards on-device learning—e.g., to customize features, such as predictive keyboard and natural language processing, to the individual mobile user. Second, the conventional clustering algorithm, kmeans, an unsupervised learning technique that could be used to derive meaningful information from the raw data on mobile devices (e.g., user activity and health



(a) Execution Time

(b) Total Energy

(c) Energy-Delay Product

(d) Total Number of Writes

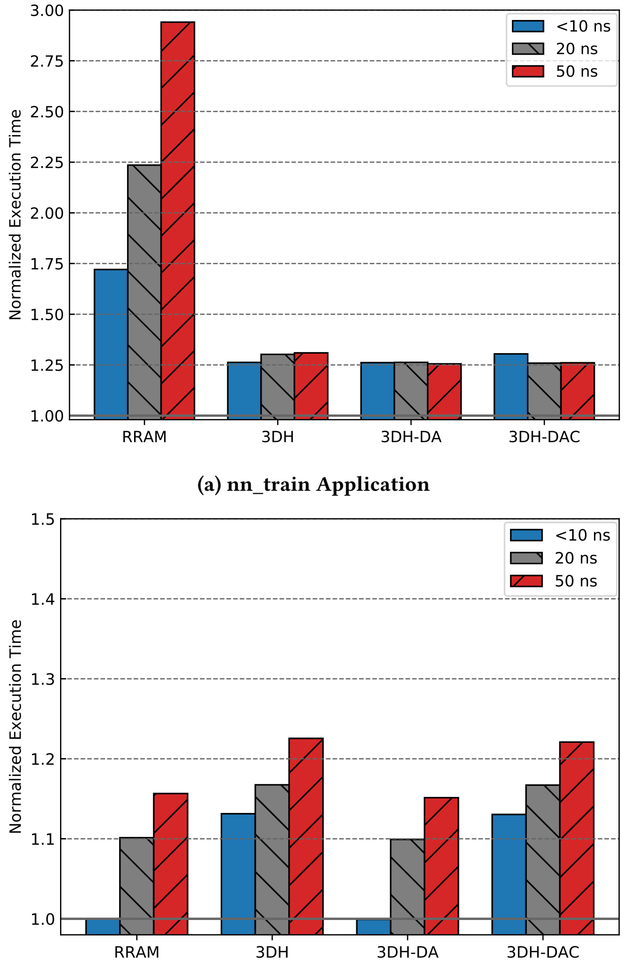
Figure 8: Comparison of 3D-hybrid Memory Architectures for Neural Network Training and Kmeans Clustering.

data from the wearables, images from the camera). Details of these applications are discussed in Section 7.

Allocating data to the DRAM and the RRAM address space is still an open challenge for the hybrid memory with flat hierarchy—i.e., 3DH-DA and 3DH-DAC configurations. We use a simple, yet effective, workload-specific static allocation policy: the network (i.e., weights and activations) is stored in the DRAM since the training involves frequent accesses and updates of the weights, the training data is allocated in the RRAM since it involves mainly reads. We follow a similar approach for kmeans: the cluster centroids and distances are stored in the DRAM, unclassified data is stored in the RRAM.

The experimental results of both applications are plotted in Figure 8 normalized to the baseline (DRAM). Similar to the previous results in Section 8.1, all 3D systems demonstrate significant energy reduction compared to the baseline. Remarkably, the RRAM only main memory offers the highest energy savings for both applications. However, it has also the largest performance degradation (∼ 75%) for the nn_train application (cf. Figure 8a). Compared to the RRAM only system, the performance reduction of nn_train is not severe in 3DH (∼ 25%) due to the high DRAM cache hit rate (90%). In contrast, the low hit rate of kmeans (20%) worsens the execution time and energy of 3DH compared to the other configurations. It must be noted that the 3DH-DA with our simple allocation policy performs equally or better than the 3DH for both applications. Although the 3DH-DAC has higher execution time and energy compared to the 3DH-DA, it provides more flexibility to the programmer in memory management since it is infeasible to track each allocation and de-allocation in complex programs.

When comparing the EDP in Figure 8c, RRAM only configuration is the best choice for both applications despite its more substantial execution time for the nn_train. However, since RRAM has limited endurance, frequent writes will damage the memory cells. Therefore, the number of writes to the RRAM should be minimized. As



(a) nn_train Application

(b) kmeans Application

Figure 9: Influence of the RRAM Selector Delay.

shown in Figure 8d, the nn_train has approximately *three* orders of magnitude larger writes compared to the kmeans, in the *RRAM* configuration. However, the number of writes for nn_train has significantly reduced in the 3DH configuration (high cache hit rate), thus improving the RRAM reliability. The reduction in writes for kmeans is only marginal due to its low cache hit rate. Notably, our allocation policy in 3DH-DA and 3DH-DAC significantly reduces the number of RRAM writes compared to 3DH (approximately 10 X reduction), in both applications. The 3DH-DAC enables the programmer to allocate data structures even with marginal–moderate writes into the hybrid address space since the DRAM cache filters the direct writes to the RRAM. Therefore, we expect this configuration will perform even better in reducing the RRAM writes for complex programs. Further research is needed in this direction.

8.3 Influence of Selector Delay

In this section, we evaluate how the increase in RRAM access latencies due to selector delays (cf. Section 3) influence the performance of the two machine learning applications for the different hybrid memory configurations. Based on the reported numbers in [6], we considered three selector delays: 10 ns (default for all evaluations), 20 ns, and 50 ns.

Figure 9 shows the normalized execution time for nn_train and kmeans applications compared to the DRAM baseline. The execution time of nn_train significantly increases (up to 3 X) with the rise in selector delay for the *RRAM only* configuration. In contrast, there is an only marginal increase in the execution time for kmeans (up to 15%) for the same system configuration. Notably, all three hybrid configurations (3DH, 3DH-DA, and 3DH-DAC) are unaffected by the selector delays for the nn_train application. This is due to the large cache hit rate in 3DH, and due to the efficacy of our allocation policy in the other two configurations. Conversely, there is a significant increase in the execution time for kmeans with the rise in selector delays in all hybrid configurations. Apparently, the higher sensitivity of execution time to the selector delay in the 3DH system is due to the lower cache hit ratio. However, the increased selector delay sensitivity in the 3DH-DA configuration shows that there are still a few frequent accesses to the RRAM address space of the hybrid memory, which could be allocated to the DRAM address space by the programmer. In other words, this shows the potential for further improving the allocation strategy. Notable here is the lower sensitivity of the 3DH-DAC configuration to the selector delay, although the execution time is yet higher than the 3DH-DA due to the overhead of managing the DRAM cache. However, these overheads can be reduced by improving the cache, for instance, by introducing a miss-map structure to avoid the DRAM cache tag look-up in the case of a miss [24]. Consequently, we expect that the 3DH-DAC configuration—i.e., the hardware-software approach—will perform better even if the programmer fails to allocate some pages in the DRAM address space that are frequently accessed.

9 CONCLUSION

In this paper, we presented the architecture of a novel heterogeneous 3D-main memory composed of RRAMs and DRAMs. This new architecture exploits the capabilities of 3D-integration to combine the benefits of RRAMs (high density, scalability, and low energy) and DRAMs (fast read/write accesses, and high endurance). Thus, it offers large on-chip main memory capacities for emerging big data applications on mobile devices. Our evaluations showed that, on average, the proposed 3D-hybrid architecture consumes 73% lower energy and 61% lower average power than a 2D-Hybrid memory architecture. Besides, we evaluated the new architecture for various hybrid memory organizations using different machine learning tasks on mobile devices. For a neural network training application, the 3D-hybrid memory provides up to 52% and 60% energy savings for the hierarchical and flat memory organizations, respectively, compared to the main memory with DDR4 DRAM. An RRAM only main memory offers the maximum energy savings (75%) for this application. However, it suffers from 3–4 orders of magnitude larger number of RRAM writes than the hybrid memory, thus causing endurance issues in the RRAM. Finally, we analyzed the influence of RRAM memory cell selector delays on the performance of these applications.

ACKNOWLEDGMENTS

This work was funded by the EU OPRECOMP project under grant agreement No. 732631 (http://oprecomp.eu), the Carl-Zeiss Stiftung under the Nachwuchsförderprogram 2015, and the DFG grant No. WE 2442/14-1.

REFERENCES

[1] I. Bhati, M. T. Chang, Z. Chishti, S. L. Lu, and B. Jacob. 2016. DRAM Refresh Mechanisms, Penalties, and Trade-Offs. *IEEE Trans. Comput.* 65, 1 (Jan 2016), 108–121. https://doi.org/10.1109/TC.2015.2417540

[2] Christian Bienia. 2011. *Benchmarking Modern Multiprocessors.* Ph.D. Dissertation. Princeton University.

[3] Nathan Binkert, Bradford Beckmann, Gabriel Black, Steven K. Reinhardt, Ali Saidi, Arkaprava Basu, Joel Hestness, Derek R. Hower, Tushar Krishna, Somayeh Sardashti, Rathijit Sen, Korey Sewell, Muhammad Shoaib, Nilay Vaish, Mark D. Hill, and David A. Wood. 2011. The Gem5 Simulator. *SIGARCH Comput. Archit. News* 39, 2 (Aug. 2011), 1–7. https://doi.org/10.1145/2024716.2024718

[4] Ian Bratt. 2018. Arm's First-Generation Machine Learning Processor. https://www.hotchips.org/hc30/2conf/2.07_ARM_ML_Processor_HC30_ARM_2018_08_17.pdf

[5] Andrew Carter. [n.d.]. MNIST Neural Network in C. https://github.com/AndrewCarterUK/mnist-neural-network-plain-c

[6] An Chen. 2017. Memory selector devices and crossbar array design: a modeling-based assessment. *Journal of Computational Electronics* 16, 4 (01 Dec 2017), 1186–1200. https://doi.org/10.1007/s10825-017-1059-7

[7] H. Chen, S. Yu, B. Gao, P. Huang, J. Kang, and H. . P. Wong. 2012. HfOx based vertical resistive random access memory for cost-effective 3D cross-point architecture without cell selector. In *2012 International Electron Devices Meeting.* 20.7.1–20.7.4. https://doi.org/10.1109/IEDM.2012.6479083

[8] J. H. Cho, J. Kim, W. Y. Lee, D. U. Lee, T. K. Kim, H. B. Park, C. Jeong, M. Park, S. G. Baek, S. Choi, B. K. Yoon, Y. J. Choi, K. Y. Lee, D. Shim, J. Oh, J. Kim, and S. Lee. 2018. A 1.2V 64Gb 341GB/S HBM2 stacked DRAM with spiral point-to-point TSV structure and improved bank group data control. In *2018 IEEE International Solid - State Circuits Conference - (ISSCC).* 208–210. https://doi.org/10.1109/ISSCC.2018.8310257

[9] E. Confalonieri, P. Amato, D. Balluchi, D. Caraccio, and M. Dallabora. 2015. Mobile Memory Systems. In *2015 Mobile Systems Technologies Workshop (MST).* 1–7.

[10] Inc. Crossbar. 2017. *Crossbar ReRAM Technology.* https://www.crossbar-inc.com/assets/resources/white-papers/Crossbar-ReRAM-Technology.pdf

[11] Shuiguang Deng, Hailiang Zhao, Weijia Fang, Jianwei Yin, Schahram Dustdar, and Albert Zomaya. 2019. Edge Intelligence: The Confluence of Edge Computing and Artificial Intelligence.

[12] Android developer guide. 2019. Memory allocation among processes. https://developer.android.com/topic/performance/memory-management

[13] G. Dhiman, R. Ayoub, and T. Rosing. 2009. PDRAM: A hybrid PRAM and DRAM main memory system. In *2009 46th ACM/IEEE Design Automation Conference.* 664–669. https://doi.org/10.1145/1629911.1630086

[14] Linux Documentation. 2019. *zram: Compressed RAM-based block devices.* https://www.kernel.org/doc/Documentation/blockdev/zram.txt

[15] J. Han, S. Kim, S. Lee, J. Lee, and S. J. Kim. 2018. A Hybrid Swapping Scheme Based On Per-Process Reclaim for Performance Improvement of Android Smartphones (August 2018). *IEEE Access* 6 (2018), 56099–56108.

[16] Intel. 2019. Product Brief: Intel Optane DC Persistent Memory. https://www.intel.com/content/dam/www/public/us/en/documents/product-briefs/optane-dc-persistent-memory-brief.pdf

[17] JEDEC. 2014. Wide I/O 2 standard (JESD229-2). (2014).

[18] JEDEC. 2017. *JEDEC DDR5 & NVDIMM-P Standards Under Development.* https://www.jedec.org/news/pressreleases/jedec-ddr5-nvdimm-p-standards-under-development

[19] S. H. Jo, T. Kumar, C. Zitlaw, and H. Nazarian. 2015. Self-limited RRAM with ON/OFF resistance ratio amplification. In *2015 Symposium on VLSI Technology (VLSI Technology).* T128–T129. https://doi.org/10.1109/VLSIT.2015.7223715

[20] Matthias Jung, Christian Weis, and Norbert Wehn. 2015. DRAMSys: A flexible DRAM Subsystem Design Space Exploration Framework. *IPSJ Transactions on System LSI Design Methodology (T-SLDM)* (August 2015).

[21] Jung-Sik Kim, Chi Sung Oh, Hocheol Lee, Donghyuk Lee, Hyong-Ryol Hwang, Sooman Hwang, Byongwook Na, Joungwook Moon, Jin-Guk Kim, Hanna Park, Jang-Woo Ryu, Kiwon Park, Sang-Kyu Kang, So-Young Kim, Hoyoung Kim, Jong-Min Bang, Hyunyoon Cho, Minsoo Jang, Cheolmin Han, Jung-Bae Lee, Kyehyun Kyung, Joo-Sun Choi, and Young-Hyun Jun. 2011. A 1.2V 12.8GB/s 2Gb mobile Wide-I/O DRAM with 4x128 I/Os using TSV-based stacking. In *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2011 IEEE International.* 496–498. https://doi.org/10.1109/ISSCC.2011.5746413

[22] Mark Lapedus. 2019. DRAM Scaling Challenges Grow. https://semiengineering.com/dram-scaling-challenges-grow/

[23] T. Liu, T. H. Yan, R. Scheuerlein, Y. Chen, J. K. Lee, G. Balakrishnan, G. Yee, H. Zhang, A. Yap, J. Ouyang, T. Sasaki, S. Addepalli, A. Al-Shamma, C. Chen, M. Gupta, G. Hilton, S. Joshi, A. Kathuria, V. Lai, D. Masiwal, M. Matsumoto, A. Nigam, A. Pai, J. Pakhale, C. H. Siau, X. Wu, R. Yin, L. Peng, J. Y. Kang, S. Huynh, H. Wang, N. Nagel, Y. Tanaka, M. Higashitani, T. Minvielle, C. Gorla, T. Tsukamoto, T. Yamaguchi, M. Okajima, T. Okamura, S. Takase, T. Hara, H. Inoue, L. Fasoli, M. Mofidi, R. Shrivastava, and K. Quader. 2013. A 130.7mm lt;sup gt;2 lt;/sup gt; 2-layer 32Gb ReRAM memory device in 24nm technology. In *2013 IEEE International Solid-State Circuits Conference Digest of Technical Papers.* 210–211. https://doi.org/10.1109/ISSCC.2013.6487703

[24] Gabriel H. Loh and Mark D. Hill. 2011. Efficiently Enabling Conventional Block Sizes for Very Large Die-stacked DRAM Caches. In *Proceedings of the 44th Annual IEEE/ACM International Symposium on Microarchitecture* (Porto Alegre, Brazil) *(MICRO-44).* ACM, New York, NY, USA, 454–464. https://doi.org/10.1145/2155620.2155673

[25] Q. Luo, X. Xu, T. Gong, H. Lv, D. Dong, H. Ma, P. Yuan, J. Gao, J. Liu, Z. Yu, J. Li, S. Long, Q. Liu, and M. Liu. 2017. 8-Layers 3D vertical RRAM with excellent scalability towards storage class memory applications. In *2017 IEEE International Electron Devices Meeting (IEDM).* 2.7.1–2.7.4. https://doi.org/10.1109/IEDM.2017.8268315

[26] Deepak Mathew, André Lucas Chinazzo, Christian Weis, Matthias Jung, Bastien Giraud, Pascal Vivet, Alexandre Levisse, and Norbert Wehn. 2019. RRAMSpec: A Design Space Exploration Framework for High Density Resistive RAM. In *2019 International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation (SAMOS).*

[27] Deepak M. Mathew, André Lucas Chinazzo, Christian Weis, Matthias Jung, Bastien Giraud, Pascal Vivet, Alexandre Levisse, and Norbert Wehn. 2019. RRAM-Spec: A Design Space Exploration Framework for High Density Resistive RAM. In *Embedded Computer Systems: Architectures, Modeling, and Simulation,* Dionisios N. Pnevmatikatos, Maxime Pelcat, and Matthias Jung (Eds.). Springer International Publishing, Cham, 34–47.

[28] J. Meza, J. Chang, H. Yoon, O. Mutlu, and P. Ranganathan. 2012. Enabling Efficient and Scalable Hybrid Memories Using Fine-Granularity DRAM Cache Management. *IEEE Computer Architecture Letters* 11, 2 (2012), 61–64.

[29] D. Milojevic, H. Oprins, J. Ryckaert, P. Marchal, and G. Van der Plas. 2011. DRAM-on-logic Stack – Calibrated thermal and mechanical models integrated into PathFinding flow. In *2011 IEEE Custom Integrated Circuits Conference (CICC).* 1–4. https://doi.org/10.1109/CICC.2011.6055357

[30] OPRECOMP. 2018. OPRECOMP Micro-Benchmarks. https://github.com/oprecomp/micro-benchmarks

[31] S. K. Park. 2015. Technology Scaling Challenge and Future Prospects of DRAM and NAND Flash Memory. In *2015 IEEE International Memory Workshop (IMW).*

[32] Moinuddin K. Qureshi, Vijayalakshmi Srinivasan, and Jude A. Rivers. 2009. Scalable High Performance Main Memory System Using Phase-change Memory Technology. In *Proceedings of the 36th Annual International Symposium on Computer Architecture* (Austin, TX, USA) *(ISCA '09).* ACM, New York, NY, USA, 24–33. https://doi.org/10.1145/1555754.1555760

[33] Luiz E. Ramos, Eugene Gorbatov, and Ricardo Bianchini. 2011. Page Placement in Hybrid Memory Systems. In *Proceedings of the International Conference on Supercomputing* (Tucson, Arizona, USA) *(ICS '11).* Association for Computing Machinery, New York, NY, USA, 85–95. https://doi.org/10.1145/1995896.1995911

[34] Andy Rudoff. 2017. *Persistent Memory Programming.* https://www.usenix.org/system/files/login/articles/login_summer17_07_rudoff.pdf

[35] M. M. Sabry Aly, T. F. Wu, A. Bartolo, Y. H. Malviya, W. Hwang, G. Hills, I. Markov, M. Wootters, M. M. Shulaker, H. . Philip Wong, and S. Mitra. 2019. The N3XT Approach to Energy-Efficient Abundant-Data Computing. *Proc. IEEE* 107, 1 (Jan 2019), 19–48. https://doi.org/10.1109/JPROC.2018.2882603

[36] M. M. Shulaker, T. F. Wu, M. M. Sabry, H. Wei, H. . P. Wong, and S. Mitra. 2015. Monolithic 3D integration: A path from concept to reality. In *2015 Design, Automation Test in Europe Conference Exhibition (DATE).* 1197–1202. https://doi.org/10.7873/DATE.2015.1111

[37] Gary Sims. [n.d.]. *How much RAM does your phone need in 2019?* https://www.androidauthority.com/how-much-ram-do-you-need-in-smartphone-2019-944920/

[38] Stefan Slesazeck and Thomas Mikolajick. 2019. Nanoscale resistive switching memory devices: a review. *Nanotechnology* 30, 35 (jun 2019), 352003. https://doi.org/10.1088/1361-6528/ab2084

[39] A. Sodani, R. Gramunt, J. Corbal, H. Kim, K. Vinod, S. Chinthamani, S. Hutsell, R. Agarwal, and Y. Liu. 2016. Knights Landing: Second-Generation Intel Xeon Phi Product. *IEEE Micro* 36, 2 (2016), 34–46.

[40] Sung Hyun Jo, T. Kumar, S. Narayanan, W. D. Lu, and H. Nazarian. 2014. 3D-stackable crossbar resistive memory based on Field Assisted Superlinear Threshold (FAST) selector. In *2014 IEEE International Electron Devices Meeting.* 6.7.1–6.7.4. https://doi.org/10.1109/IEDM.2014.7046999

[41] D. Velenis, M. Stucchi, E. J. Marinissen, B. Swinnen, and E. Beyne. 2009. Impact of 3D design choices on manufacturing cost. In *2009 IEEE International Conference on 3D System Integration.* 1–5.

[42] C. Wang, C. McClellan, Y. Shi, X. Zheng, V. Chen, M. Lanza, E. Pop, and H. . Philip Wong. 2018. 3D Monolithic Stacked 1T1R cells using Monolayer MoS2 FET and hBN RRAM Fabricated at Low (150°C) Temperature. In *2018 IEEE International Electron Devices Meeting (IEDM)*. 22.5.1–22.5.4. https://doi.org/10.1109/IEDM.2018.8614495

[43] Christian Weis, Matthias Jung, and Norbert Wehn. 2016. *3D Memories (Book chapter in the Handbook of 3D Integration)*. Vol. 4. Wiley-VCH.

[44] Christian Weis, Igor Loi, Luca Benini, and Norbert Wehn. 2013. Exploration and Optimization of 3-D Integrated DRAM Subsystems. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 32, 4 (April 2013), 597–610.

[45] Christian Weis, Abdul Mutaal, Omar Naji, Matthias Jung, Andreas Hansson, and Norbert Wehn. 2017. DRAMSpec: A High-Level DRAM Timing, Power and Area Exploration Tool. *International Journal of Parallel Programming* 45, 6 (01 Dec 2017), 1566–1591. https://doi.org/10.1007/s10766-016-0473-y

[46] Paul N. Whatmough, Chuteng Zhou, Patrick Hansen, Shreyas K. Venkataramanaiah, Jae-sun Seo, and Matthew Mattina. 2019. FixyNN: Efficient Hardware for Mobile Computer Vision via Transfer Learning. *CoRR* abs/1902.11128 (2019). arXiv:1902.11128 http://arxiv.org/abs/1902.11128

[47] H. . P. Wong, H. Lee, S. Yu, Y. Chen, Y. Wu, P. Chen, B. Lee, F. T. Chen, and M. Tsai. 2012. Metal–Oxide RRAM. *Proc. IEEE* 100, 6 (June 2012), 1951–1970. https://doi.org/10.1109/JPROC.2012.2190369

[48] Jackie Wu Kerry Jiang Bing Wei Liu Xiao-Feng Li, Yong Wang. 2012. Mobile OS Architecture Trends. *Intel Technology Journal, Volume 16, Issue 4* (2012).

[49] Xiaoxin Xu, Q. Luo, Tiancheng Gong, Hangbing Lv, Shibing Long, Qi Liu, S. S. Chung, Jing Li, and Ming Liu. 2016. Fully CMOS compatible 3D vertical RRAM with self-aligned self-selective cell enabling sub-5nm scaling. In *2016 IEEE Symposium on VLSI Technology*. 1–2. https://doi.org/10.1109/VLSIT.2016.7573388

[50] Vinson Young, Zeshan Chishti, and Moinuddin K. Qureshi. 2019. TicToc: Enabling Bandwidth-Efficient DRAM Caching for both Hits and Misses in Hybrid Memory Systems. *CoRR* abs/1907.02184 (2019). arXiv:1907.02184 http://arxiv.org/abs/1907.02184

[51] V. Young, Z. A. Chishti, and M. K. Qureshi. 2019. TicToc: Enabling Bandwidth-Efficient DRAM Caching for Both Hits and Misses in Hybrid Memory Systems. In *2019 IEEE 37th International Conference on Computer Design (ICCD)*. 341–349.

[52] V. Young, C. Chou, A. Jaleel, and M. Qureshi. 2018. ACCORD: Enabling Associativity for Gigascale DRAM Caches by Coordinating Way-Install and Way-Prediction. In *2018 ACM/IEEE 45th Annual International Symposium on Computer Architecture (ISCA)*. 328–339. https://doi.org/10.1109/ISCA.2018.00036

[53] S. Yu and P. Chen. 2016. Emerging Memory Technologies: Recent Trends and Prospects. *IEEE Solid-State Circuits Magazine* 8, 2 (Spring 2016), 43–56. https://doi.org/10.1109/MSSC.2016.2546199

[54] Ye Yu and N.K. Jha. 2018. A Monolithic 3D Hybrid Architecture for Energy-Efficient Computation. *IEEE Transactions on Multi-Scale Computing Systems* PP (11 2018), 1–1. https://doi.org/10.1109/TMSCS.2018.2882433

[55] Y. Yu and N. K. Jha. 2018. Energy-Efficient Monolithic Three-Dimensional On-Chip Memory Architectures. *IEEE Transactions on Nanotechnology* 17, 4 (July 2018), 620–633. https://doi.org/10.1109/TNANO.2017.2731871

[56] Yuang Zhang, Li Li, Zhonghai Lu, Axel Jantsch, Minglun Gao, Hongbing Pan, and Feng Han. 2014. A survey of memory architecture for 3D chip multi-processors. *Microprocessors and Microsystems* 38, 5 (2014), 415 – 430. https://doi.org/10.1016/j.micpro.2014.03.007

[57] Pin Zhou, Vivek Pandey, Jagadeesan Sundaresan, Anand Raghuraman, Yuanyuan Zhou, and Sanjeev Kumar. 2004. Dynamic Tracking of Page Miss Ratio Curve for Memory Management. *SIGARCH Comput. Archit. News* 32, 5 (Oct. 2004), 177–188. https://doi.org/10.1145/1037947.1024415