Journal of
Low Power Electronics and Applications

MDPI

Article

DESTINY: A Comprehensive Tool with 3D and Multi-Level Cell Memory Modeling Capability

Sparsh Mittal [1,*,†], Rujia Wang [2,*,†] and Jeffrey Vetter [3]

1 IIT Hyderabad, Telangana 502285, India
2 Department of Electrical and Computer Engineering, University of Pittsburgh, Pittsburgh, PA 15260, USA
3 Oak Ridge National Laboratory, Oak Ridge, TN 37830, USA; vetter@computer.org
* Correspondence: sparsh0mittal@gmail.com (S.M.); rujia.wng@gmail.com or rujia.w@pitt.edu (R.W.)
† These authors contributed equally to this work.

Received: 2 August 2017; Accepted: 4 September 2017; Published: 11 September 2017

Abstract: To enable the design of large capacity memory structures, novel memory technologies such as non-volatile memory (NVM) and novel fabrication approaches, e.g., 3D stacking and multi-level cell (MLC) design have been explored. The existing modeling tools, however, cover only a few memory technologies, technology nodes and fabrication approaches. We present DESTINY, a tool for modeling 2D/3D memories designed using SRAM, resistive RAM (ReRAM), spin transfer torque RAM (STT-RAM), phase change RAM (PCM) and embedded DRAM (eDRAM) and 2D memories designed using spin orbit torque RAM (SOT-RAM), domain wall memory (DWM) and Flash memory. In addition to single-level cell (SLC) designs for all of these memories, DESTINY also supports modeling MLC designs for NVMs. We have extensively validated DESTINY against commercial and research prototypes of these memories. DESTINY is very useful for performing design-space exploration across several dimensions, such as optimizing for a target (e.g., latency, area or energy-delay product) for a given memory technology, choosing the suitable memory technology or fabrication method (i.e., 2D v/s 3D) for a given optimization target, etc. We believe that DESTINY will boost studies of next-generation memory architectures used in systems ranging from mobile devices to extreme-scale supercomputers. The latest source-code of DESTINY is available from the following `git` repository: https://bitbucket.org/sparsh_mittal/destiny_v2.

Keywords: cache; SRAM; eDRAM; non-volatile memory (NVM or NVRAM); STT-RAM; ReRAM; PCM; SOT-RAM; DRAM; DWM; flash; open-source; modeling tool; emerging memory technologies

1. Introduction

As processor core-count rises and key applications become more data-intensive, the memory requirements of modern computing systems are growing tremendously. To cater to these needs, modern processors are using large-sized memory structures, e.g., last level cache, main memory and storage. For example, Intel's 22 nm Haswell processor employs 128 MB eDRAM LLC (last level cache) [1], and the 22-nm Xeon E5-2600 processor has 45 MB SRAM LLC [2]. To address these challenges, researchers are exploring novel memory technologies, fabrication schemes and higher-density cell-designs. For example, eDRAM, STT-RAM, ReRAM, PCM, DWM, SOT-RAM and (NAND) Flash memory have received significant attention in recent years [3] (unless otherwise mentioned, in this paper, Flash refers to NAND Flash). Similarly, 3D integration technology has been explored for achieving higher bandwidth, higher flexibility in routing signals, power, clock and ability to integrate diverse memory technologies for designing hybrid memory designs [4,5]. Finally, since NVMs have a large resistance margin between set and reset states, MLC designs have been studied, which store multiple (e.g., two) bits in each cell to achieve higher density than the SLC designs.

Architectural exploration and system integration of these technologies and design approaches crucially depend on the availability of comprehensive, open-source and validated modeling tools. Existing modeling tools, however, fail to meet these requirements. Tools such as CACTI(3DD) [6,7] and NVSim [8] only model a few memory technologies. Researchers typically use CACTI for modeling SRAM/DRAM and NVSim for modeling NVMs (e.g., [9]); however, these tools use different modeling framework, assumptions and input/output formats. For example, NVSim provides the output in the form of hit/miss/write latency/energy, while CACTI provides the output in the form of access time and random cycle time. Similarly, NVSim provides the ability to find a configuration optimized for a certain target (e.g., area, leakage, etc.), while CACTI does not do so. Each of these input parameters can have marked influence on the output obtained from the tool [8,10,11]. Further, due to differences in modeling frameworks, the outputs of these tools can be different even for the same input configuration. An example of this is shown in Table 1. It is clear that both the output values and output format of each tool are different. This makes it difficult to have one-to-one correspondence between their inputs/outputs. Thus, the lack of comprehensive tools may force researchers to compare estimates from different tools or restrict their choices to only a few memory technologies. These, however, may lead to suboptimal or even incorrect conclusions.

Table 1. CACTI and NVSim results for the same cache configuration: 32-nm, 64 B block, 16-way 4 MB SRAM cache.

CACTI	NVSim
Area: 14.90 mm^2	Area: 6.75 mm^2
Leakage: 0.574 W	Leakage: 0.395 W
Access and random cycle time: 0.634 and 3.119 ns	Hit/miss/write latency: 2.009, 0.314 and 1.079 ns
Read dynamic energy: 0.182 nJ	Hit/miss/write dynamic energy: 0.388, 0.032, 0.363 nJ

Some researchers use in-house modeling tools (e.g., [12,13]); however, experiments conducted with such tools may not be reproducible, and their accuracy may not be verified. Furthermore, in absence of a 3D modeling tool, some studies (e.g., [12]) derive parameters for 3D memories using a linear extrapolation of 2D parameters, which may be inaccurate. Finally, some tools such as 3DCacti [14] have not been updated for recent feature sizes (e.g., sub-45 nm). Clearly, the current state-of-the-art calls for an open-source, comprehensive, validated and up-to-date tool for allowing full design-space exploration of memory technologies and design trends.

Contributions: In this paper, we present DESTINY, a 3D design-space exploration tool for SRAM, eDRAM and non-volatile memory. DESTINY utilizes the 2D SLC circuit-level modeling framework of NVSim tool for SRAM, STTRAM, PCM, ReRAM and Flash. It also utilizes the coarse- and fine-grained TSV (through silicon via) models from the CACTI-3DD tool. Further, DESTINY adds the model of eDRAM (Section 4.1), SOT-RAM, DWM, the capability to model MLC designs (Section 4.4) and two additional types of 3D designs (Section 4.5). Overall, DESTINY enables modeling of both 2D/3D designs of SRAM, eDRAM, STT-RAM, PCM and ReRAM and 2D designs of SOT-RAM, DWM and Flash memory. In addition to SLC models for all memories, DESTINY can also model MLC models of NVMs. Thus, DESTINY can model both volatile and non-volatile memories and both conventional and emerging memories. Furthermore, it models technology nodes ranging from 22 nm to 180 nm. Table 2 summarizes the capabilities of DESTINY and compares them with those of CACTI and NVSim.

Table 2. An overview of the modeling capabilities of some open-source tools.

Tools	SRAM	eDRAM	PCM	STT-RAM	ReRAM	SOT-RAM	Flash	DWM
CACTI(3DD)	2D	2D/3D	✗	✗	✗	✗	✗	✗
NVSim	2D	✗	2D, SLC					✗
DESTINY	2D/3D		2D/3D, SLC/MLC			2D, SLC/MLC		

We have compared the results obtained from DESTINY against several commercial and research prototypes [4,5,11,15–26] to validate the newly-added memories, MLC and 3D models in DESTINY (Section 5). The modeling error has been observed to be less than 10% for most cases and less than 25% for all cases. This can be accepted as reasonable for an academic modeling tool and is also in range with the errors produced by previous tools [8].

DESTINY facilitates exploring a large design space, which provides important insights and is also useful for early stage estimation of emerging memory technologies (Section 6). Further, by virtue of being an open-source tool, it facilitates reproducible research and easy extension of the tool for many more usage scenarios than those discussed in the paper. For example, apart from modeling standard caches, DESTINY can also model assist structures (e.g., victim cache, write-buffer, tag-only caches) and the translational look aside buffer (TLB) [27] designed with different memory technologies. We believe that DESTINY will be a useful tool in architecture and system-level studies and will assist researchers, designers and technical professionals.

This paper extends the previous version [28] in several significant ways:

1. We have presented the motivation behind the development of DESTINY by discussing the design trends in modern processors and the limitations of existing modeling tools (Section 2).
2. We have discussed the device-level data storage mechanism of each the memory technology (Section 3).
3. We have now added support for modeling new memories (DWM, SOT-RAM) and MLC designs for all NVMs (including Flash). We have discussed their modeling framework and validation (Sections 4.2 to 4.4 and 5.1 to 5.7).
4. We have now shown the use of DESTINY in performing design-space exploration, for example finding the optimal memory technology for a given optimization target (Section 6.1), finding the optimal number of 3D layers for a given optimization target (Section 6.2), modeling assist structures (Section 6.3), etc.
5. We have discussed the usefulness of DESTINY in gaining insights for designing management policies for memory structures such as cache, the register file, etc., using different memory technologies (Section 6.4).

2. Motivation and Related Work

2.1. Motivation behind the Design of DESTINY

To meet the challenges of rising core-count and data-intensive applications, modern CPUs and GPUs feature increasingly larger storage structures. For example, IBM's 45-nm Power7 processor had a 32 MB LLC [29]; the 32-nm Power7+ processor had an 80 MB LLC [30]; and the 22-nm Power8 processor had a 96 MB LLC [31]. LLC size in GPUs is also on the rise [32]. Similarly, the total size of the GPU register file has increased from 512 KB on G80 (Tesla) and 2048 KB on GF100 (Fermi) to 7680 KB on GK210 (Kepler) and 14,336 KB on GP100 (Pascal) [33]. It is clear that high-density memory technologies (e.g., NVMs), cell designs (e.g., MLC) and fabrication approaches (e.g., 3D) will be essential to meet the rising memory demands in future computing systems.

Further, given that different memory structures (e.g., register file, shared memory, first and last level caches, main memory) in different processors (CPUs or GPUs) need to be optimized for distinct points in the latency/energy/area spectrum, a comprehensive modeling tool is definitely required that allows complete design-space exploration over memory technologies, design approaches and optimization targets. DESTINY is intended to fulfill this need and also to boost architectural studies of next-generation memory systems.

2.2. A Comparison of Modeling Tools

Researchers have proposed several tools for modeling and estimating the energy consumption, the performance of processors or their specific components. A few existing tools provide modeling

capability individually for different memory technologies, such as SRAM, DRAM, eDRAM and NVMs. CACTI [7] simulates SRAM caches and has been extended to support eDRAM and DRAM. Furthermore, several improvements have been made to CACTI to improve its modeling capability/accuracy. Mamidipaka et al. [34] proposed eCACTI, which adds a leakage model into CACTI, and Li et al. [35] proposed CACTI-P, which models low-power caches (e.g., cache with sleep transistors). Chen et al. [6] presented CACTI-3DD, which adds a TSV model for DRAM memory; however, this tool is designed for DRAM, and hence, does not allow accurate modeling of 3D SRAM caches. 3DCacti [14] provides the ability to model 3D SRAM; however, this tool has not been updated to support technology nodes below 45 nm. None of these tools model emerging NVMs. NVSim provides 2D modeling of SRAM, ReRAM, STT-RAM, PCM and SLC NAND Flash. NVSim has not been validated for SOT-RAM, and it does not provide a configuration file for modeling it.

None of these tools provide the comprehensive modeling and design space exploration capability as provided by DESTINY. Existing tools also do not provide the capability to model DWM and MLC, etc. As an increasing number of industrial designs utilizes 3D stacking [4,5], research on 3D stacking has become very important. However, existing 3D modeling tools such as CACTI-3DD [6] and 3DCacti [14] do not model NVMs. Another challenge in using multiple tools is that over time, these tools undergo revisions (due to more accurate modeling, bug fixes, etc.) which makes the task of cross-comparison even more difficult. Clearly, DESTINY offers distinct advantages over existing tools, and by virtue of its comprehensive modeling capability, it can be a very useful design space exploration and decision-support tool.

3. A Background on Memory Technologies and MLC Design

3.1. Data Storage Mechanism of Memory Technologies

We now briefly discuss the data storage mechanism of different memory technologies. For more details and discussion of related issues, we refer the reader to previous work [8,15,36–38].

eDRAM: In eDRAM, the data are stored as charge in a capacitor, which is either a deep-trench capacitor or stacked capacitor between metal wire layers on a die. Access is controlled using a single transistor with the capacitor connected to the drain terminal. The gate of the transistor is used to access the device, while the source terminal is used to read or write to the capacitor. Typically, a charged capacitor represents a "1", while a discharged capacitor represents a "0". Over time, eDRAM loses charge, and hence, it requires periodic refresh operations [28].

STT-RAM: STT-RAM utilizes a magnetic tunnel junction (MTJ) as the primary memory storage [36]. An MTJ consists of two ferromagnetic layers. The reference layer has a fixed magnetic polarization, while the free layer has a programmable magnetic polarization. Current passing through the MTJ allows the free layer to change polarization. The MTJ resistance is low when both layers are polarized in the same direction, while polarization in opposite directions yields high resistance. These two resistance values are used to determine the "1" and "0" states, respectively.

SOT-RAM: SOT-RAM is another memory technology that uses MTJ to store data [15]. The major difference between SOT-RAM and STT-RAM is that the SOT-RAM cell has three terminal MTJ to decouple the read and write path. As a result, in SOT-RAM, read and write operations can be performed without disturbance. The read current still flows through the MTJ to sense the resistance as in STT-RAM, but the write current now flows between the source line and the write line, through a hard metal instead of the MTJ [15]. The direction of write current can affect the magnetization of the free layer and change the stored value.

ReRAM: ReRAM uses a metal oxide material between two metal electrodes to store data values [39]. The value depends on the concentration of oxygen vacancies in the metal oxide. Applying current to the two electrodes can move these oxygen vacancies to either form or break down a filament, which allows for high conductance in the metal oxide. A filament formed by oxygen

vacancies has a low resistance state representing a "1". When the filament is broken down, there is a small concentration of oxygen vacancies leading to high resistance state, representing "0".

PCM: PCM uses a chalcogenide material such as GeSbTe (GST) for data storage [18,40,41]. The GST can be changed between crystalline and amorphous phases by heating the material for certain periods of time. A "SET" operation crystallizes the GST by applying a medium temperature (~300 °C) for a relatively long period of time (~150 ns). This allows the material to move and restructure itself into crystalline form. A "RESET" operation switches the material to an amorphous phase by applying high temperature (~800 °C) for a shorter period of time (~100 ns) and quickly removing heat. This causes the material to melt and remain in an amorphous phase when cooled. The crystalline phase shows low resistance corresponding to a "1" bit, while the amorphous phase shows high resistance corresponding to a "0" bit.

DWM: DWM, also known as racetrack memory, stores data in magnetic tapes [11,37]. Each magnetic tape consists of many magnetic domains that can store multiple bits of information. For each magnetic tape, there are multiple read and write ports that can program DWM MTJ just as STT-RAM. By using bidirectional current, the MTJ can be programmed into "1" and "0" states. In additional to read and write operations, DWM also has a shift operation. By injecting the shift current along the DWM tape, the magnetic domains in the tape can be shifted forward and backward to move the desired MTJ under read/write port. Thus, if the desired MTJ is not already aligned to the read/write port, a shift operation is required before a read or write operation.

Flash: Flash memory stores data as charge trapped on a floating gate between the control gate and the channel of a CMOS transistor [40,42]. This is illustrated in Figure 1a. Compared to NOR Flash, NAND Flash has higher density, lower cost and higher write speed, and hence, it is more suitable for data storage. A flash chip can be organized in one or more planes, and each plane has a set of blocks, which contains multiple pages. NAND Flash supports three types of operations, viz., read, program (write) and erase. The read operation can read the entire page by measuring the threshold voltage of the floating gate transistor. A program operation can change the bits from one to zero only, and hence, to change a bit in a page from zero to one, the block that contains the page needs to be erased, which sets all bits in the block to one.

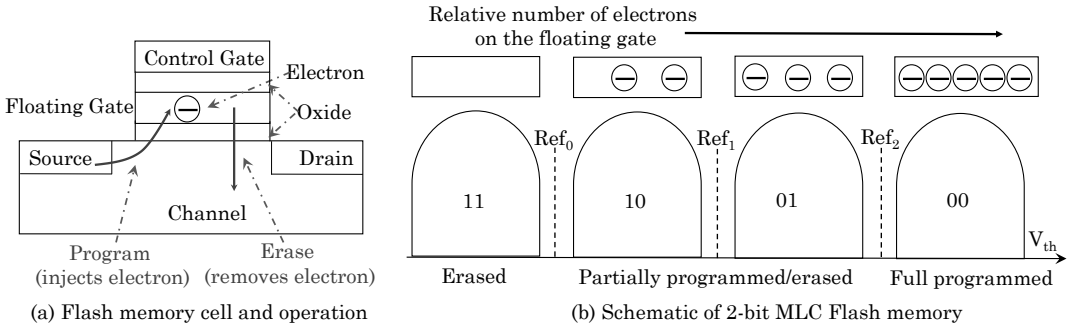

Figure 1. An illustration of Flash memory.

3.2. Multi-Level Cell Memory

MLC design provides the opportunity to increase cell density and therefore enhance memory capacity. Several NVM technologies can support MLC storage and programming. We now briefly discuss the working mechanism of MLC NVMs modeled in DESTINY.

3.2.1. MLC PCM and ReRAM

Both PCM and ReRAM encode data using resistance values. Since the difference between low and high resistance states is very large (e.g., three to four orders of magnitude for PCM [43] and five orders of magnitude for ReRAM [39]), the broad resistance range can be partitioned into several states for storing multiple bits in a cell.

Since both MLC PCM and ReRAM are resistive memories, the programming (writing) approaches for them have similarities. With SET and RESET operations, the cell resistance can be programmed into a lower or higher state. Due to the non-deterministic nature of MLC programming, program-and-verify (P&V) operations are used by both PCM and ReRAM to precisely control the final resistance value [19].

3.2.2. MLC STT-RAM and SOT-RAM

MLC STT-RAM can be designed in two ways [17,44–46]. In the 'series MTJ' design, two MTJs with disparate characteristics are vertically stacked, and in the 'parallel MTJ' design, the free layer is partitioned into soft and hard domains, each of which can store a bit. The parallel MLC has smaller write current and area, but also has a higher error rate than the series MLC design [17].

The two bits in MLC STT-RAM have different switching properties, and hence, they are termed as the hard bit and soft bit, respectively. The hard bit requires larger magnitude current with a longer duration, while soft bit programming is faster and requires a small amount of current.

Recently, Kim et al. [16] proposed an MLC SOT-RAM design that is similar to the MLC STT-RAM design. In the series MLC SOT-RAM design, the lower MTJ can be programmed through the separate write path, and the other MTJ still requires a write current flow through two MTJs. In the parallel design, read and write paths are still decoupled. Programming two MTJs requires different amounts of current flow through the hard metal.

It is clear that there are many similarities between the characteristics and programming strategies of MLC STT-RAM and SOT-RAM, and hence, we categorize them into a single group. They both use two MTJs with disparate characteristics such that switching the hard bit requires a write current with a larger amplitude and longer duration than that required for the soft bit.

3.2.3. MLC Flash

By precisely controlling the amount of charge stored in the floating gate, a flash cell can also store multi-level data. A schematic of a 2 = two-bit MLC Flash is illustrated in Figure 1b. Similar to MLC PCM and ReRAM, the programming of MLC Flash requires longer latency to achieve the narrow threshold voltage distribution by iterative program and verify operations. For writing a two-bit MLC, the threshold voltage of the cell is first programmed into either a temporary state region or an erased state region depending on the least significant bit (LSB) value. In the second step, it is programmed into one of the four regions (11, 10, 01 or 00) based on the LSB and most significant bit (MSB) bits. The read operation requires multiple comparisons with the reference cells to determine the MSB and LSB stored in the cell.

4. DESTINY Modeling Framework

DESTINY is a comprehensive tool able to model multiple memory technologies. Figure 2 presents a high-level diagram of DESTINY. The DESTINY framework is based on the 2D circuit-level model of NVSim, which has been heavily extended to model the 2D eDRAM and 3D design of SRAM, eDRAM, monolithic NVMs and emerging NVMs, like DWM, SOT-RAM and MLC NVMs.

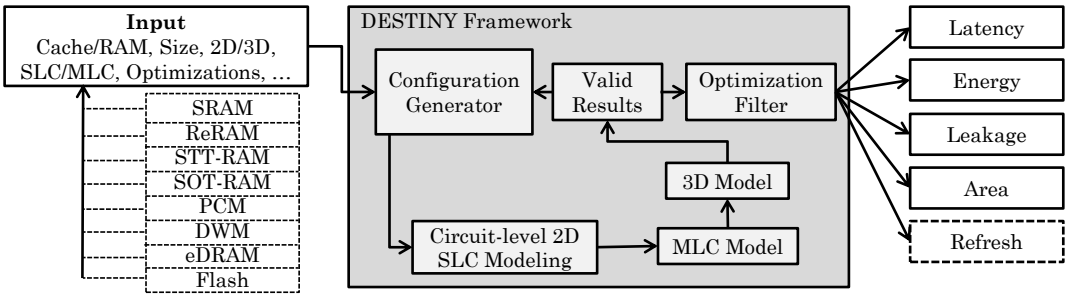

Figure 2. High-level overview of the DESTINY framework.

For a given memory technology, the device-level parameters (e.g., cell size, read and write voltage) are provided as input to DESTINY. Then, possible memory organization configurations are generated, which are passed to the circuit-level modeling code. For MLC designs, MLC modeling is done with specified MLC parameters in configuration (refer to Section 4.4). Similarly, 3D modeling is done for 3D designs, and the generated configurations can have different numbers of 3D layers, e.g., 1, 2, 4, 8, 16, etc (refer to Section 4.5). Those designs that are physically infeasible are considered as invalid and are therefore discarded. As an example, if the refresh period of an eDRAM cache design is greater than its retention period, it is considered invalid. This reduces the number of possible options to be explored. The remaining configurations are passed through an optimization filter, which selects the optimal configuration based on the given target such as smallest read latency or least area, etc.

As we show in Section 6.1, DESTINY also provides the capability to do design space exploration across multiple memory technologies, for example finding an optimal technology for a given metric/target. For such cases, the device-level parameters for multiple memory technologies are fed as input to DESTINY (shown in the left of Figure 2). Using these, the best results for each technology are obtained, which are further compared to find the optimal memory technology. A similar approach is also used for finding the optimal layer count for a given optimization target (Section 6.2). DESTINY can model both cache (which has both tag and data array) and RAM (with has only data array). For RAM structures, only one bank is modeled, and for multi-bank modeling, the parameters obtained from DESTINY for one bank can be integrated into architecture-level main memory simulators.

In what follows, we discuss the specific extensions made in the DESTINY modeling framework.

4.1. eDRAM Model

NVSim provides an incomplete model of eDRAM, which has also not been validated against any prototype. To enable modeling of eDRAM, we separate the peripheral and device logic to simulate multiple types of technologies. eDRAM requires refresh for maintaining data integrity and typical retention periods range from 40 µs to 100 µs [4,5] for temperature in the range of 380 K. We implemented a refresh model based on Kirihata et al. [47], in which all subarrays are refreshed in parallel, row-by-row. The benefit of this approach is that the refresh operations do not significantly reduce the availability of banks to service requests. It is also easy to extend DESTINY to model other refreshing schemes such as refreshing the mats in parallel.

From the perspective of performance and feasibility, eDRAM cache designs that provide bank availability (i.e., the percentage of time where the bank is not refreshing) below a threshold are not desired, and hence, they are discarded by DESTINY. The retention period of eDRAM varies exponentially with the temperature [28], and hence, DESTINY scales the retention period accordingly to model the effect of the temperature. DESTINY provides the refresh latency, energy and power as the output of the tool.

4.2. DWM Model

Several DWM models have been proposed in the research literature [11,37]. The DWM model used in DESTINY is based on that proposed by Zhang et al. [11], and it is straightforward to extend it to model the DWM designs proposed by other researchers. Zhang et al. argue that due to the tape-like architecture of DWM, organizing DWM cells as traditional array-like memory does not lead to an area-efficient implementation. Hence, they propose laying out multiple tapes on the access transistors and placing shift transistors above and below the tapes. This model is illustrated in Figure 3.

There are three important parameters that determine the latency, energy and architecture of the DWM macro unit. First, the length of DWM tape determines how many tapes are needed for a given cache capacity. If we have a long DWM tape, the number of tapes required is reduced. In addition, the shift latency/energy is proportional to the length of DWM tape. A longer tape requires more effort to shift it. Second, the number of read and write ports in each tape determine the maximum domains that need to be shifted on each read/write operation and the overhead region at the top/bottom of

the tape reserved for shift-out bits. Third, the number of tapes that are vertically grouped into one macro unit also need to be defined to achieve different cell area efficiency. For example, in Figure 3, four tapes are grouped together to form a macro unit. The read/write access port is placed every four domains, and thus, the maximum shift distance is four.

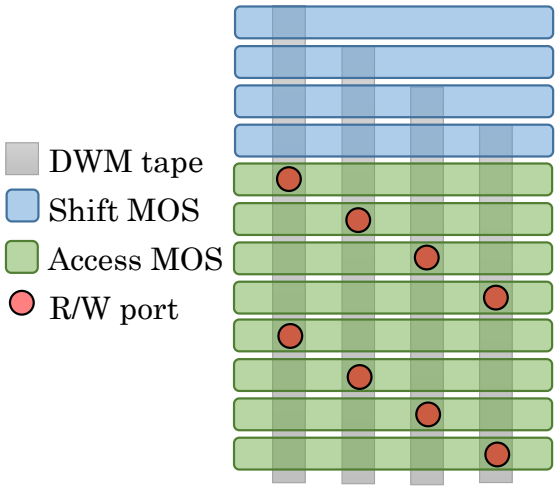

Figure 3. DWM layout and model.

4.3. SOT-RAM Model

In our model, SOT-RAM and STT-RAM share most of the peripheral circuit. The read path and reading strategy for both technologies are the same. Compared with STT-RAM, SOT-RAM has much smaller cell write latency and current, because in SOT-RAM, the write current does not pass through the MTJ, but through a hard metal, as shown in Figure 4. The write operation is also bidirectional, and it magnetizes the free layer above the hard metal with spin orbit torque. However, due to the added write line, SOT-RAM has a higher cell area compared to STT-RAM. In our framework, we do not model the write line separately, but reflect it as an increased cell area in the cell configuration file.

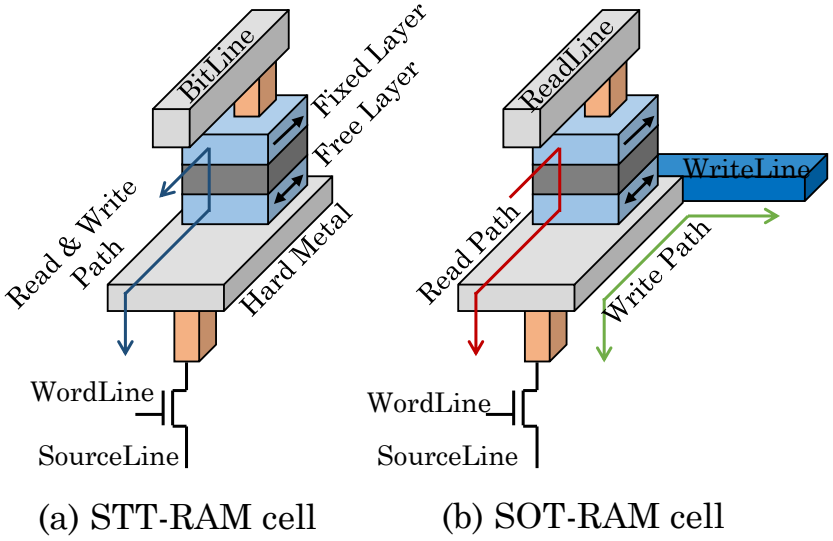

Figure 4. Comparison of the STT-RAM and SOT-RAM cell model.

4.4. MLC Model

To model both the SLC and MLC design, DESTINY allows specifying whether the cell is 1-bit, 2-bit, 3-bit, etc. We now discuss the modeling of the MLC read operation and the MLC write operation in DESTINY.

4.4.1. Read Operation Modeling

Since a multi-level cell stores more than one bit of data, a multi-step comparison is required for reading the stored data. We use the binary-search read-out model in our framework [48], and the number of readout steps is determined by the number of stored bits. For example, with two-bit/cell and three-bit/cell MLCs, two- and three- read steps (respectively) are required.

Figure 5 illustrates the two-step read procedure. The readout voltage (or current) is first compared with a threshold λ_1, and if greater, the MSB (most significant bit) is taken as zero. A second comparison with λ_2 determines the LSB (least significant bit). Similarly, if the voltage were less than λ_1, the MSB would be one, and a second comparison with λ_0 determines the LSB.

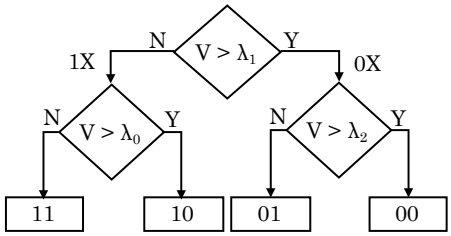

Figure 5. MLC read model. Note that $\lambda_0 < \lambda_1 < \lambda_2$.

The latency and energy model for MLC read is extended by adding extra sense amplifier overhead. We assume that for each step of sensing, the latency that comes from the sense amplifier is fixed. Therefore, the MLC reading latency depends on the sense amplifier latency and the number of comparison steps.

4.4.2. Write Operation Modeling for MLC PCM, ReRAM and Flash

As discussed before, the write operation to MLC PCM and ReRAM is performed using the iterative P&V strategy. There are two P&V strategies, viz., 'reset-before-set' and 'set-before-reset'. 'Reset before set' first gives the memory cell a reset pulse to change the cell to a high resistance state. Afterwards, multiple set and verify operations are performed to change the cell resistance to the desired range. Similarly, 'set before reset" initially writes the cell to a low resistance state with a set pulse and then changes it to higher resistance states gradually with multiple reset and verify operations. Figure 6a,b illustrates these programming strategies. DESTINY supports using both of these strategies.

Similarly, programming MLC Flash uses iterative P&V strategy. Before programming a page, the whole block is erased. The erase operation is similar to the initialization reset operation in 'reset-before-set'. Then, by iteratively changing the programming gate voltage V_{GP}, the desired V_T is achieved for representing multi-bit per cell. Figure 6c shows the 'program and verify' scheme used for MLC Flash.

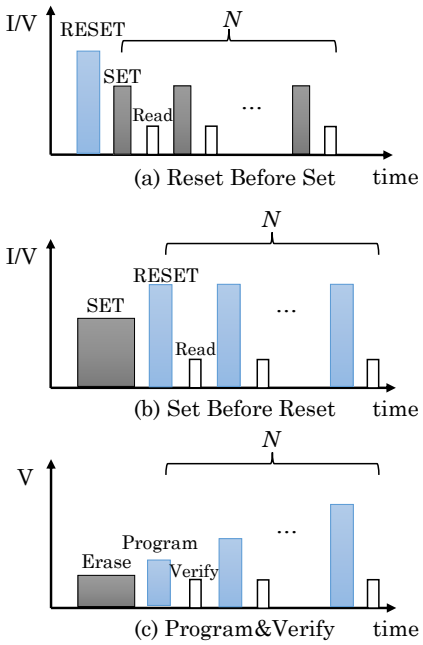

Figure 6. MLC write model ((**a**,**b**) apply to MLC PCM and MLC ReRAM, whereas (**c**) applies to MLC Flash) [18].

We adopt the model from Zhao et al. [18] to compute the MLC programming latency and energy. For example, in the "reset before set" programming style, the cell level latency and energy can be calculated by adding the latency/energy in each operation [18]:

$$Latency = L_{Reset} + N \times (L_{Set} + L_{Read} + 2 \times \delta) \quad (1)$$

$$Energy = E_{Reset} + N \times (E_{Set} + E_{Read}) \quad (2)$$

where N shows the number of iterations required to write a cell, and δ is the interval time between two operations. L_X and E_X refer to the latency and energy values for the X operation, respectively. The latency and energy models above only apply to a single cell write. By adding peripheral circuits parts, DESTINY can model the latency and energy for a single cache block access, as well.

4.4.3. Write Operation Modeling for MLC STT-RAM and SOT-RAM

MLC for STT-RAM and SOT-RAM can be designed in two types: 'series MTJs' and 'parallel MTJs' [16,17]. DESTINY can model both these types of MLC designs. In MLC MRAM, the bits requiring relatively higher and lower switching current are referred to as 'hard bit' and 'soft bit', respectively. In both serial and parallel MLC designs, writing the hard bit also flips the soft bit. Hence, if the final state is 00 or 11 (i.e., both soft and hard bits are the same), the write operation can be completed in a single step; however, if the final state is 01 or 10 (i.e., both bits are different), a second write operation is also required [17]. Since the information about the data written is not available in a circuit-level tool (such as DESTINY), we model a simple writing scheme in DESTINY, where a write operation needs to switch both the hard and soft bit, in that order. Thus, the write latency and energy is calculated by adding the two MTJ write pulses and write energy values together, as shown in the following formula:

$$Latency = L_{HardBit} + L_{SoftBit} \quad (3)$$

$$Energy = E_{HardBit} + E_{SoftBit} \quad (4)$$

Note that based on the modeling of this simple writing scheme, it is straightforward to extend DESTINY for modeling sophisticated (e.g., data-dependent) writing schemes [17].

4.5. 3D Model

Several types of 3D stacking have been explored in the literature, such as face-to-face, face-to-back and monolithic stacking [49,50]. In all of these approaches (except in monolithic stacking), dies are bonded together using various techniques (e.g., wafer-to-wafer, die-to-wafer or die-to-die bonding). The difference in these approaches lies in terms of their effect on die testing and yield. Wafer-to-wafer may reduce the yield by bonding a dysfunctional die anywhere in the stack. Die-to-wafer and die-to-die can minimize this by testing individual dies, although this has an adverse effect on alignment.

The most common 3D stacking is known as face-to-back bonding. In this form, TSVs are used to penetrate the bulk silicon and connect the first metal layer (the back) to the top metal layer (the face) of a second die. In face-to-face bonding, the top metal layer of one die is directly fused to the top metal layer of a second die. Monolithic stacking does not utilize TSVs at all. Instead, monolithically stacked dies build devices on higher metal layers connected using normal metal layer vias wherever necessary.

Each approach offers its own advantages and disadvantages. Face-to-back bonding must carefully consider placement and avoid transistors when being formed through the bulk silicon, while face-to-face bonding does not. Therefore, face-to-face bonding offers the potential for higher via density. The downside of face-to-face bonding is that only two layers can be formed in this manner. Monolithic stacking provides the benefit of highest via density; however, this technique cannot be applied in a design that requires transistors to be formed on higher layers, since this can destroy the previously formed transistors.

The 3D model of DESTINY facilitates all of the aforementioned flavors of 3D stacking. Apart from this, the granularity at which TSVs are placed can be either coarse- or fine-grained, similar to the approach in CACTI-3DD [6]. This granularity defines how many TSVs are placed and what portions of a cache (e.g., peripheral circuits or memory cells) reside on different dies. We utilize these models in our validation. First, a model for direct die-to-die stacking with face-to-face bonding is provided [24]. Second, the monolithic stacking model for 3D horizontal ReRAM is provided [22]. Face-to-back bonding using TSVs is utilized in other designs [20,23].

A few previous works (e.g., [6]) assume that TSVs in face-to-back are buffered, which may lead to redundant buffering in some designs and also increases the latency and energy overhead of the TSVs. This overhead may be acceptable in large-sized DRAMs that are modeled in CACTI-3DD, but is unacceptable in caches that are relatively smaller in size and becomes increasingly obvious with smaller memory macro designs. Further, several memory peripheral components already provide full-swing logic signals that do not require extra buffering. In this work, we provide a TSV model that may act as a buffered or unbuffered TSV, as well as vias used in face-to-face bonding.

With the rising number of layers, the number of memory mats in each layer is reduced, and hence, we need to select a scheme for folding of the memory banks. Our coarse- and fine-grained models assume simplistic folding scheme, where the mats are equally divided in all of the layers. In the coarse-grained model, TSVs are used to broadcast undecoded row and column select signals to all of the layers at once. One logic layer is assumed to provide output in this model over a shared TSV bus spanning all layers. The fine-grained model differs by broadcasting decoded row and column signals to all of the layers. It is assumed that a dedicated logic layer is used for all pre-decoder units. The resulting design uses more TSVs, but its area and latency may be reduced.

Monolithically stacked horizontal ReRAM (HReRAM) [22] uses a concept similar to cross-point designs. The limitations of the cross-point designs, however, are the increased sneak current and voltage drop associated with increasing subarray size. In 3D design, this limitation becomes even more prominent, and it further limits the subarray size of 3D-stacked ReRAM as the sneak current can potentially flow into upper layers, as well. To alleviate this limitation, we extended the cross-point model in NVSim to account for the increased number of wordlines and bitlines in the third dimension.

An example of HReRAM is shown in Figure 7 with four layers monolithically stacked. In this design, there are no TSVs between memory layers. Instead, the memory cells are sandwiched between wordlines and bitlines, and additional layers are added similar to adding more metal layers. Our 3D ReRAM model considers current flow between all inactive bitlines and wordlines when a single cell is read. This model dramatically reduces the number of valid designs when considering ReRAM, so we cannot simply stack cross-point arrays that are considered optimal for a 2D design. Typically, this effect can be slightly mitigated using diode accessed cells as in [22], and hence, we provide the option to model diodes or a no access device.

As with the choice of the number of layers, DESTINY allows modeling both odd and even numbers of layers, as long as cache capacity divided by the number of layers is power-of-two, since without it, different numbers of mats will be placed in each layer, which will reduce the area efficiency and also complicate the circuit-design. For example, (32 MB, four-layer), (24 MB, three-layer), (20 MB, five-layer) designs, etc., can be easily modeled in DESTINY, whereas (32 MB, six-layer), (18 MB, three-layer) designs, etc., cannot be modeled.

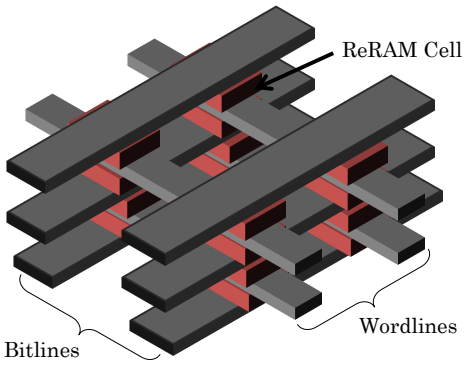

Figure 7. Four-layer monolithically stacked ReRAM. Storage elements are sandwiched directly between layers of wordlines (south-east orientation) and bitlines (south-west orientation).

5. Validation Results

To evaluate the accuracy of DESTINY, we have validated it against several industrial prototypes. Due to the emerging nature of many memory technologies, industrial prototypes could not be found and hence, we have also used research papers to find validation targets. We obtain the cache/macro configuration from the corresponding papers and use them to set the device-level input parameters for DESTINY. Finally, we compare the value from the prototype and the projected value from DESTINY and obtain the percentage modeling error. As shown below, the modeling error is less than 10% in most cases and less than 25% in all cases.

5.1. DWM Validation

We validate our DWM model against that shown by Zhang et al. [11]. We choose the MU-64-32-4 configuration [11], which means that each tape has 64 bits, the total access ports in a macro unit is 32 and four tapes are grouped together. Therefore, the read/write ports are placed every eight bits per tape. The process node is 65 nm. A 32-MB data array is designed under the area optimization target [11]. The design includes two mats, and each mat includes 2 $\times$ 2 subarrays (note that the bank layouts are specified as mat $\times$ mat, and subarray layouts are specified as wordline $\times$ bitline). The results are shown in Table 3. Clearly, six out of eight parameters show less than 11% modeling error.

Table 3. Validation of DWM [11].

Metric	Actual	DESTINY	Error
Area (mm^2)	6.89	7.954	15.44%
Read Latency (ns)	5.83	4.424	−24.12%
Write Latency (ns)	12.49	12.635	1.16%
Shift Latency (ns)	5.31	4.878	−8.14%
Read Energy (pJ)	236.63	257.582	8.85%
Write Energy (pJ)	1032	1145	10.95%
Shift Energy (pJ)	214.61	230	7.17%
Leakage Power (mW)	163.72	147	−10.21%

5.2. SLC SOT-RAM Validation

We validate the SLC SOT-RAM model against that of Guillaume et al. [15]. Guillaume et al. designed a 16-MB L3 cache using SOT-RAM. We adopt an 8 × 1 mats organization for modeling this. The results are shown in Table 4. It can be observed that errors in both read and write latencies are less than 10%.

Table 4. SLC SOT-RAM validation [15].

Metric	Actual	DESTINY	Error
Read Latency (ns)	3.8	3.87	1.84%
Write Latency (ns)	2.8	2.562	−8.50%

5.3. MLC SOT-RAM Validation

For validating MLC SOT-RAM, we use the series-MLC (S-MLC) cell configuration from the work of Kim et al. [16], which is the only MLC SOT-RAM paper that we could find. We only validate cell write energy, since this is the only parameter out of area/latency/energy that is shown in the paper of Kim et al. The write energy is calculated by multiplying write power with the write pulse duration; however, the write pulse duration is not shown in [16]. Since MLC SOT-RAM shares many similarities with MLC STT-RAM, we assume the write pulse duration based on STT-RAM. Specifically, the write pulse for the hard bit is assumed to be 10 ns [51]. For the write pulse for the soft bit, we experiment with two different values, viz., 2 ns and 10 ns, based on the values for SLC SOT-RAM and MLC STT-RAM [15,51]. For these values, the write energy found by DESTINY is 2.29 pJ and 3.25 pJ, respectively, as shown in Table 5. Kim et al. [16] show the write energy to fall within 1.8 pJ and 3.8 pJ, depending on the incoming and originally stored data. Clearly, the results provided by DESTINY are within the range of values shown by Kim et al.

Table 5. MLC SOT-RAM validation [16].

Metric	Actual	DESTINY	Error
Write Energy (pJ)	1.8–3.8	2.29–3.25	—

5.4. MLC STT-RAM Validation

We validated our MLC STT-RAM model with that of Hong et al. [17]. They modeled a 16-MB LLC designed with MLC STT-RAM at 45 nm with a cache block size of 64 B. A 16-MB MLC cache is organized with 64 × 2 mats, and each mat has 2 × 2 subarrays. The results with DESTINY are shown in Table 6. It is clear that the results provided by DESTINY show less than 10% error.

Table 6. MLC STT-RAM validation [17].

Metric	Actual	DESTINY	Error%
Area (mm^2)	3.42	3.68	7.69%
Read Latency (ns)	4.8	4.83	0.67%
Write Latency (ns)	21.2	20.46	−3.48%
Read Energy (pJ)	181.01	192.00	6.07%
Write Energy (pJ)	349.64	378.00	8.11%
Leakage Power (mW)	324.54	324.50	−0.01%

5.5. MLC PCM Validation

Table 7 shows our validation results against the PCM MLC model in Zhao et al. [18] with the 'reset before set' strategy and writing the value 11. The duration of reset pulse is 75 ns, whereas that of set pulse is 15 ns, and 23 (N) iterations are performed to write the value, following [18]. The interval between each operation is 1 ns (δ).

Table 7. MLC PCM cell level validation [18].

Metric	Actual	DESTINY	Error
Write Latency (ns)	696	696.96	0.14%
Write Energy (pJ)	122.92	122.92	0%

Note that the numbers reported here are for cell level latency/energy values and not chip-level values. Due to the high PCM cell write latency, the latency of the peripheral circuitry of the PCM bank is considered as negligible, and hence, the cell latency dominates the overall latency. For this reason, several previous works also show cell-level latencies only [8]. Thus, our cell-level results show less than 1% error since we use the exact cell model as that described by Zhao et al. [18].

5.6. MLC ReRAM Validation

We validate the MLC ReRAM model against a 4-Mb ReRAM prototype [19]. We configure the memory as 4 × 8 mats, and each mat contains one subarray. The prototype chip can be programmed either as SLC or MLC [19]. We choose the number of set iterations as 16, and the duration of the set and reset pulses is 5 ns according to Xu et al. [39], since these parameters are not reported by Sheu et al. [19]. In MLC mode, the ReRAM prototype has a write latency of 160 ns [19]. Using 'reset before set' scheme in DESTINY, we observe a value of 157.8 ns, as shown in Table 8. Thus, our result shows less than 2% error.

Table 8. MLC ReRAM validation [19].

Metric	Actual	DESTINY	Error
Write Latency (ns)	160	157.823	−1.36%

5.7. MLC Flash Validation

We validate DESTINY against two MLC Flash prototypes:

1. a 159 mm^2 32-nm 32 Gb MLC Flash prototype [25] and
2. a 182 mm^2 56-nm 16 Gb MLC Flash prototype [26]

We configure the chip as having two mats and a single subarray, following both [25,26]. We validate the area for both prototypes (since they do not report latency/energy values) and present the results in Table 9. Clearly, the magnitude of error in DESTINY's output is less than 7% and 1%, respectively.

Table 9. MLC Flash validation [25,26].

Metric	Actual	DESTINY	Error (%)
32 nm 32 Gb MLC Flash prototype [25]			
Area (mm^2)	159	148.12	−6.84%
56 nm 16 Gb MLC Flash [26]			
Area (mm^2)	182	183.54	0.84%

5.8. 3D SRAM Validation

We validate the 3D SRAM model of DESTINY against two previous works [23,24] that utilized hSpice models to simulate the latency and energy of 3D-stacked SRAM caches. Hsu and Wu [23] sweep over various cache sizes ranging from 1–16 MB. Their work assumes stacking at the bank level, that is a 2D planar cache containing M banks can be stacked up to M-layers. Since NVSim does not model banks, we only compare against the smallest cache size. Our proposed design assumes shared vertical bitlines, which corresponds to the fine-grained model in DESTINY. Analogous to their bank folding method, we assume a fixed configuration for two layers and fold along a single dimension in the bank layout to estimate four-layer latency and energy. Our two-layer design assumes a 4×32 bank layout. Based on the aspect ratio of our SRAM cells and the size of the subarray, this design attempts to keep the area square, which is likely the configuration of an hSpice model. The four layer design folds along the number of mats per column assuming a 4×16 bank layout. Table 10 shows the validation results of DESTINY against the 3D SRAM design in [23]. Notice that the errors are consistently less than 4%.

Table 10. Validation for the 3D SRAM model.

Design	Metric	Actual	DESTINY	Error
1 MB [23]	Latency	1.85 ns	1.91 ns	3.54%
2 layers	Energy	5.10 nJ	5.05 nJ	−0.98%
1 MB [23]	Latency	1.75 ns	1.80 ns	2.68%
4 layers	Energy	4.5 nJ	4.51 nJ	0.18%
4 MB [24]	Latency	7.85 ns	7.23 ns	−7.91%
2 layers	Energy	0.13 nJ	0.13 nJ	−2.59%
4 MB [24]	Latency	6.10 ns	6.95 ns	14.03%
4 layers	Energy	0.12 nJ	0.13 nJ	4.75%
2 MB [24]	Latency	5.77 ns	5.78 ns	0.05%
2 layers	Energy	0.12 nJ	0.13 nJ	2.74%
2 MB [24]	Latency	4.88 ns	5.53 ns	13.5%
4 layers	Energy	0.12 nJ	0.13 nJ	8.46%
1 MB [24]	Latency	3.95 ns	3.90 ns	−1.11%
2 layers	Energy	0.11 nJ	0.11 nJ	−0.13%
1 MB [24]	Latency	3.07 ns	3.04 ns	−0.85%
4 layers	Energy	0.11 nJ	0.11 nJ	−0.89%

Puttaswamy and Loh [24] explored the design space of 3D SRAM for the 65-nm technology node. Their work considers a range of cache sizes from 16 KB to 4 MB. As explained above, we assume a fixed cache dimension for each cache size and fold the four-layer design in half to measure the results. These validation results are also shown in Table 10. Clearly, the errors are always less than 15%, which shows that DESTINY is reasonably accurate in modeling 3D SRAM caches.

5.9. 2D and 3D eDRAM Validation

As stated before, the eDRAM model in NVSim is incomplete and has not been validated against any prototype. Hence, we validate both the 2D and 3D model of eDRAM. The prototype works referenced below typically provide information at the macro level, rather than a full bank. Macros are well suited for verification since they are a memory-dense unit (i.e., there is no test circuitry, error-correcting code, logic, etc.) and are closest to the modeling assumptions of DESTINY; hence, we compare against a macro. The choice of macro as the granularity of validation also has the benefit of excluding components such as error-correcting code (ECC), spare and parity wordlines and bitlines, as well as allowing a fair comparison since these components are also not modeled in DESTINY.

Barth et al. [5] present a 65-nm 2D eDRAM prototype. To validate against it, we use the 2-Mb macro layout with a total of eight subarrays and, thus, use the organization of a 1024 × 2048 bank layout. Klim et al. [21] and Barth et al. [4] present 45-nm 2D eDRAM designs. We validate against them using a subarray layout of 256 × 1024 as used by them. From Table 11, it is clear that the modeling errors in 2D eDRAM validation for all cases is less than 6%.

Table 11. Validation of 2D and 3D eDRAM.

Design	Metric	Actual	DESTINY	Error
2D 2 Mb	Latency	<2 ns	1.46 ns	—
65 nm [5]	Area	0.665 mm^2	0.701 mm^2	5.42%
2D 1 Mb	Latency	1.7 ns	1.73 ns	1.74%
45 nm [4]	Area	0.239 mm^2	0.234 mm^2	−2.34%
2D 2.25 Mb	Latency	1.8 ns	1.75 ns	−2.86%
45 nm [21]	Area	0.420 mm^2	0.442 mm^2	5.31%
3D 1 Mb	Latency	<1.5 ns	1.42 ns	—
2-layers[20]	Area	0.139 mm^2	0.149 mm^2	9.32%

Golz et al. [20] present a 3D eDRAM prototype with two layers in 32-nm technology. We use the 1-Mb array as our validation target. Based on the 16 Kb μarray size of 32 × 512 and 1 Mb layout, we assume two 1024 × 512 subarrays. From Table 11, the modeling error in area is less than 10%, and thus, DESTINY can be accepted as reasonably accurate. Given the similarities between eDRAM and DRAM, we believe that DESTINY can also model DRAM technology.

5.10. 3D ReRAM Validation

As for 3D ReRAM, we validate against a monolithically stacked ReRAM memory [22], also known as 3D horizontal ReRAM. In monolithically stacked designs, additional wordlines and bitlines are stacked directly by fabricating extra metal layers with NVM cells used in place of vias. This type of design does not use TSV or flip-chip style bonding. Our validation therefore considers our more detailed model of cross-point architecture spanning multiple layers.

We design the simulated memory as an 8-Mb RAM. The design consists of four subarrays each accessed in parallel with a 64-bit input bus. We again remove the ECC logic and specify two monolithically stacked layers per die with one die total. The results of validation are shown in Table 12.

Table 12. Validation of 3D ReRAM [22].

Metric	Actual	DESTINY	Error (%)
Read latency (ns)	25	24.16	3.36
Read bandwidth (MB/s)	305	315.786	−3.54
Write latency (MB/s)	17.2	20.13	−17.03
Write bandwidth (MB/s)	443.56	379	14.55

It is clear that the read latency projection of DESTINY is very close to the value reported in [22], while the error in write latency is higher. This can be attributed to the fact that Kawahara et al. [22] use a write optimization to reduce sneak current, which is not modeled in DESTINY. Furthermore, the range of acceptable write pulse times according to their shmoo plot is very wide, ranging from 8.2 ns to 55 ns. Our prediction falls in the lower end of the plot, which is closer to the 8.2-ns write pulse for a total of 17.2 ns of write time at the bank level.

6. Design Space Exploration Using DESTINY

With the increased number of options for memory technologies and fabrication techniques, the number of possible design options increases exponentially. Since no single memory technology is superior for all parameters and optimization targets, a designer must make the right choice for each design scenario. While it is relatively straightforward to deduce the optimal memory technology for some parameters (e.g., the technology with the smallest cell size is likely to have the lowest area), this is not easy for other parameters such as read/write EDP (energy delay product), since they depend on the interaction of multiple factors. Clearly, the use of a modeling tool such as DESTINY is imperative for full design space exploration and optimization. In this section, we present the features of DESTINY to demonstrate this. Note that the design exploration can be done by providing multiple cell files in the configuration file, all of which are automatically simulated by DESTINY. By setting different optimization target, DESTINY can generate the best configuration for each target.

6.1. Finding the Optimal Memory Technology

We first show the capability of DESTINY to find the best memory technology for a given optimization target. We consider the SLC design of six memory technologies (ReRAM, STT-RAM, SOT-RAM, PCM, eDRAM and SRAM) and the MLC design of four memory technologies (ReRAM, STT-RAM, SOT-RAM and PCM). Each cache has the same configuration, viz., one-layer 16-MB 16-way cache designed with the 32-nm node and 64-B block size. Furthermore, the device roadmap is LOP (low operating power), and the sequential cache access mode is used. We did not include DWM and Flash in this comparison because of the different memory organization of DWM and very high latency of Flash (refer to Section 4.2). Table 13 shows the optimal memory technology found by DESTINY for each of the eight different optimization targets.

Table 13. Design space exploration results of determining the optimal memory technology for a desired optimization target (refer to Section 6.1). The table shows the results on all parameters for comparison purposes (Lat. = latency, En. = energy, Pw. = power). EDP, energy delay product.

Optimization Target	Optimal Technology	Area (mm^2)	Read Lat. (ns)	Write Lat. (ns)	Read En. (nJ)	Write En. (nJ)	Read EDP	Write EDP	Leakage Pw. (mW)
Area	**MLC PCM**	**0.376968**	231.679	1806.28	0.956917	427.855	221.697	772,826	3.39841
Read Latency	**SOT-RAM**	5.41092	**2.18991**	1.55141	0.270523	0.368301	0.592421	0.571385	223.238
Write Latency	**SOT-RAM**	4.48621	2.5709	**1.43888**	0.300592	0.384013	0.772791	0.552548	234.453
Read Energy	**eDRAM**	4.96351	15.9016	15.8517	**0.0621384**	1.83554	0.988099	29.0964	12.0347
Write Energy	**SRAM**	22.528	275.116	274.083	0.681917	**0.032005**	187.606	8.77202	1303.36
Read EDP	**SOT-RAM**	3.73355	2.44682	1.58979	0.214578	0.310745	**0.525035**	0.494018	169.274
Write EDP	**SOT-RAM**	3.73355	2.44682	1.58979	0.214578	0.310745	0.525035	**0.494018**	169.274
Leakage	**MLC RRAM**	0.581464	124.354	529.061	0.608305	11.9998	75.6454	6348.61	**3.51614**

The results can be understood as follows. The MLC PCM is designed as a cross-point style memory, which is the most area efficient way to design PCM, resulting in the lowest area usage. For MLC ReRAM, we use the MOS-accessed structure, which avoids the sneak current; therefore, it has the lowest leakage power. Note that when ReRAM uses the non-access transistor crossbar (0T1R) design, it has a sneak path that makes its leakage power high due to sneak current. The write latency

and energy of ReRAM are very high due to the iterative write programming. The main benefits of ReRAM are its area efficiency and low leakage power.

Most NVMs, e.g., PCM, STT-RAM, are known to have high write latency, and hence, they are not optimal for write latency. However, SOT-RAM is an exception to this. Since SOT-RAM has decoupled read and write paths, its read and write operations can be individually optimized, and hence, it presents as the best candidate when optimized under read and write latency. However, the cell size of SOT-RAM is larger than that of STT-RAM, which results in increased cache area.

The write energy of SRAM is the lowest; however, the write latency of SRAM is much higher than that of SOT-RAM when optimized for write energy, resulting in SOT-RAM having the lowest write EDP. The major reason behind the lower latency of SOT-RAM compared to SRAM is the size of the H-tree-based interconnect in our design. Since our 16-MB cache is designed as a single bank, H-tree latency dominates in both designs. However, the SRAM cell size is approximately ten-times larger than that of SOT-RAM (146 F^2 vs. 15 F^2 in our cell configuration), which makes the size of the SRAM cache nearly six-times larger. This increase in overall size has a direct impact on the total latency. Similarly, eDRAM is found to have the lowest read energy. However, its read latency is much higher than that of SOT-RAM; therefore, SOT-RAM also has the lowest read EDP.

STT-RAM is outperformed by SOT-RAM on latency and energy values and by ReRAM in area and leakage, and hence, STT-RAM also does not appear as optimal for any target. For area optimization, MLC ReRAM comes close to MLC PCM (0. 377 mm^2 for PCM compared to 0.384 mm^2 for ReRAM); however, the peripheral circuitry required for ReRAM makes its area a little bit larger than that of PCM.

The results of this study show that different optimization targets can potentially yield different memory technologies as the optimal cache design and DESTINY can be a convenient tool for finding the best technology for each target. It is noteworthy that the results obtained here hold for the particular cell-level parameters used as input for each technology, and other parameters/configurations may provide different technologies as optimal for each target. Furthermore, note that DESTINY provides latency/area/energy values, and in addition to these, other factors such as reliability, (e.g., write-endurance, soft-errors [38]) commercial manufacturability, cost, etc., may also impact the choice of optimal memory technology.

6.2. Finding the Optimal Layer Count in 3D Stacking

We now show the capability of DESTINY to find the optimal number of 3D die layers for a given optimization target. In this case, DESTINY explores both a 2D design and different numbers of layers in 3D design. In other words, it explores designs with 1, 2, 4, 8 and 16 layers (the maximum layer count is fixed to 16). We use an STT-RAM cache with 32-nm technology node, 32 MB with 16 ways. Table 14 shows the results. It is clear that for different optimization targets, different numbers of layers are found as optimal.

Table 14. Design space exploration results of determining the optimal number of 3D-stacked layers for various optimization targets for STT-RAM (refer to Section 6.2). The table shows results on all parameters for comparison purposes (D = layer count).

Optimization Target	Optimal D	Area (mm^2)	Read Lat. (ns)	Write Lat. (ns)	Read En. (nJ)	Write En. (nJ)	Read EDP	Write EDP	Leakage Pw. (mW)
Area	**16**	**2.6614**	119.2630	125.6090	2.9371	3.1065	350.2820	390.2090	34.2075
Read Latency	**16**	3.6750	**3.2444**	10.7870	3.1518	3.2280	10.2257	34.8209	1281.6100
Write Latency	**16**	3.9662	3.3027	**10.7563**	3.3282	3.3734	10.9921	36.2853	2477.7100
Read Energy	**2**	9.0916	127.2620	131.9070	**0.2395**	0.4362	30.4805	57.5427	30.0845
Write Energy	**2**	13.7392	503.1190	509.8560	0.4479	**0.3271**	225.3390	166.7570	57.0693
Read EDP	**4**	7.5184	4.4265	11.6705	0.5698	0.6816	**2.5222**	7.9541	787.0820
Write EDP	**4**	8.0937	4.7674	11.8760	0.5439	0.6348	2.5929	**7.5393**	431.0840
Leakage Pw.	**1**	16.8930	1813.0100	1810.2800	0.6430	0.6009	1165.6900	1087.8700	**7.8901**

The results can be understood as follows. The area is computed as the maximum size of any die in the stack, and hence, it is minimized when the layer count is set to the largest value. Latency is also optimized for the maximum number of layers since 3D stacking enables shorter global interconnect as the subarray sizes become much smaller. However, this is not always true in general. To confirm this, we checked with progressively reduced cache sizes and found that the TSV latency begins to dominate the overall latency at a size around 4 MB; thus the design with the maximum number of layers is not selected as the optimal result.

As for dynamic energy, we note that with the rising number of layers, the cache circuit gets divided into an increasing number of layers, which lowers the H-tree energy due to decreasing routing cost. However, with the increasing number of layers, the dynamic energy of the TSV array also increases, and hence, the total TSV dynamic energy increases in a quadratic manner since it depends on the product of number of layers and dynamic energy of TSV array. Furthermore, note that in this case study, the number of layers is increasing in a geometric manner (1, 2, 4, 8, 16). Thus, total dynamic energy, which depends on H-tree energy, TSV energy and mat energy, is determined by the interaction of these factors. Now, read dynamic energy is lowest for the two-layer design, and the optimal design had a bank configuration of 1×1 in each layer. This configuration does not incur any H-tree energy, and since TSV energy is relatively small for two-layer, the total read dynamic energy of the two-layer design (0.2395 nJ) was lower than that of the one-layer design (0.2499 nJ) and other 3D designs. Write dynamic energy is also lowest for the two-layer design, and the optimal design had a bank configuration of 8×1 in each layer. On going from one layer to two layers, the decrease in mat and H-tree energy is larger than the additional TSV energy (for reasons explained above). Hence, the write dynamic energy of the two-layer design (0.3271 nJ) was lower than that of the one-layer design (0.3361 nJ) and other 3D designs. TSV energy rapidly increases with four and more layers, and hence, they were not found optimal for read or write dynamic energy.

The optimization of EDP presents an interesting case, since, as shown above, the trends of variation in energy and latency values are generally opposite. For this reason, it is expected that an intermediate value of layer count will be optimal for EDP. As explained above, with the increase in the number of layers, TSV dynamic energy increases at a much faster rate than the decrease in latency, and hence, the four-layer design was found optimal for read and write EDP. Leakage power increases linearly with the number of layers, and hence, it was found to be lowest for the one-layer design. Clearly, the choice of the number of layers can have a profound effect on the optimal value of the different parameters, and a tool such as DESTINY is vital for performing design optimization.

In a technical report [10], we have also shown the integration of DESTINY results in a multi-core performance simulator to illustrate the use of DESTINY in guiding the design of workload-specific caches.

6.3. Modeling Assist Structures

DESTINY is useful to model not only the key memory structures (e.g., cache and register file), but 'assist structures' as well. For example, some works use victim cache, write buffer or shadow tags for keeping evicted blocks, consolidating writes or collecting profiling data, etc. These structures are used as part of a broader architectural techniques, and to comprehensively account for their overhead, it is important to account for the overhead of these assist structures. These structures can be easily modeled in DESTINY by specifying their parameters as such or modifying the DESTINY code as required. Furthermore, DESTINY provides separate parameters for tag and data arrays, which is especially useful for seeing the fractional contribution of both arrays and modeling different assist structures, which may store only data or only tag or both.

6.4. Gaining Insights for Designing Architectural Techniques

The latency, area and energy parameters provided by DESTINY can help the architects with finding the strengths and weaknesses of each memory technology. This can be useful for (1) finding

the best memory technology and even a combination of them for designing a processor component and (2) designing architectural techniques for managing processor components designed with each technology. We now illustrate this with some examples.

6.4.1. Finding the Best Memory Technology for Processor Components

Since the access latencies of register file and L1 cache have a crucial impact on processor frequency and performance, they must be designed using low-latency memories, e.g., SRAM and SOT-RAM. However, the GPU register file is significantly larger than the CPU register file [33,52], and hence, fast and high-density technologies, e.g., eDRAM, one-bit DWM, STT-RAM, etc., may be also useful for designing the GPU register file, in conjunction with architectural management techniques.

LLC (e.g., L3) needs to have high capacity to avoid off-chip accesses and low leakage to save power. Hence, eDRAM and NVMs (e.g., SOT-RAM, STT-RAM, DWM) are suitable for designing LLC. Due to their high latency, PCM and ReRAM are not suitable for LLC; however, due to their high-density, they are suitable for designing main memory. Different from other array-style RAMs (e.g., ReRAM or STT-RAM), which provide random access to data, DWM is a tape-style racetrack memory where an access requires shifting operations. Thus, DWM is suited for memory structures that inherently perform serial access, e.g., FIFO (first-in first-out) buffers. Due to its low leakage power, DWM is also suitable as cache; however, the benefit of DWM reduces for random memory access patterns due to increased shifting overhead. For LLC and main memory, memory density can be further improved by using MLC and 3D designs. 3D stacking is also useful for reducing the access latency of memory technologies.

6.4.2. Designing Architectural Management Techniques for Memory Technologies

Techniques for SRAM and eDRAM: From the DESTINY results, we note that SRAM has low-density and large leakage power. Hence, leakage-reduction techniques are required for managing SRAM-based LLC [32]. Due to its low retention period, eDRAM requires frequent refresh operations, and hence, refresh management techniques are required for eDRAM-based caches and GPU RF (register file) [33]. Since the retention period of eDRAM reduces with increasing temperature, effective thermal management techniques are also required.

Techniques for NVMs: From the DESTINY output, it is clear that the latency/energy of the NVM write operations is higher than that of read operations. Hence, architectural techniques are required for reducing their effective write latency/energy and the number of write operations, e.g., cache bypassing and data compression [12,51,53–55]. This is even more important for MLC NVMs since their latency values are higher than those of SLC NVM.

Techniques for hybrid caches and memories: From the DESTINY results (e.g., Table 13), we conclude that no technology is optimal on all of the targets, and hence, to bring the best of different memory technologies together, SRAM-NVM hybrid caches and DRAM-NVM hybrid main memory can be designed. In these designs, write-intensive blocks can be migrated to SRAM/DRAM, and the remaining blocks can be migrated to NVM [9,12,56] to achieve the density of NVM at the access speed of SRAM/DRAM. Similarly, by using SRAM/DRAM as the cache or write-buffer for NVM, write accesses to NVM can be reduced.

7. Future Work and Conclusions

Due to the emerging nature of these memory technologies, only a limited number of prototypes has been demonstrated. Due to the lack of prototypes, we could not validate 3D STT-RAM and 3D PCM, although based on our validation results with 3D ReRAM, we expect that DESTINY will be accurate in modeling them as well. Furthermore, in absence of commercial prototypes, some validations have been performed using research prototypes. We plan to perform these validations as their (commercial) prototypes become available. We also plan to improve the speed of DESTINY by using techniques such as multithreading.

Currently, DESTINY models 22-nm–180-nm feature sizes, and we will extend DESTINY to model feature sizes below 22 nm. Since process variation becomes increasingly severe at small feature sizes [57], we will also integrate the process variation model in DESTINY. Similarly, several reliability issues (e.g., read-disturbance in STT-RAM and write-disturbance in PCM) become especially severe at small feature sizes [58], so we will also integrate code for modeling of these errors in DESTINY. Further, we plan to use DESTINY for studies on approximate computing whereby reducing the number of write-iterations in MLC NVM, a trade-off between accuracy and write latency/energy can be achieved [59].

In this paper, we presented DESTINY, a comprehensive, validated tool for modeling both 2D/3D and SLC/MLC designs of prominent conventional and emerging memory technologies. We described the modeling framework of DESTINY and also performed validations against a large number of prototypes. We demonstrated the capability of DESTINY to perform design space exploration over memory technologies and 3D layer counts and provide insights for designing architectural management policies for different memory technologies. We believe that DESTINY will be useful for architects, CAD designers and researchers.

Acknowledgments: Support for this work was provided by Science and Engineering Research Board (SERB), India, award number ECR/2017/000622 and Office of Advanced Scientific Computing Research in the U.S. Department of Energy. The authors thank Matt Poremba for his help in earlier stages of this work.

Author Contributions: All of the authors discussed the idea. S.M. and R.W. searched relevant literature and performed the experiments, and J.V. supervised the project. All authors discussed the results. S.M. and R.W. developed the manuscript, and all three authors approved it.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kurd, N.; Chowdhury, M.; Burton, E.; Thomas, T.P.; Mozak, C.; Boswell, B.; Mosalikanti, P.; Neidengard, M.; Deval, A.; Khanna, A. Haswell: A family of IA 22nm processors. *IEEE J. Solid-State Circuits* **2015**, *50*, 49–58.
2. Bowhill, B.; Stackhouse, B.; Nassif, N.; Yang, Z.; Raghavan, A.; Morganti, C.; Houghton, C.; Krueger, D.; Franza, O.; Desai, J.; et al. The Xeon® processor E5-2600 v3: A 22 nm 18-core product family. In Proceedings of the IEEE International Solid-State Circuits Conference (ISSCC), San Francisco, CA, USA, 22–26 February 2015; pp. 1–3.
3. Vetter, J.S.; Mittal, S. Opportunities for Nonvolatile Memory Systems in Extreme-Scale High Performance Computing. *Comput. Sci. Eng.* **2015**, *17*, 73–82.
4. Barth, J.; Plass, D.; Nelson, E.; Hwang, C.; Fredeman, G.; Sperling, M.; Mathews, A.; Kirihata, T.; Reohr, W.; Nair, K.; et al. A 45 nm SOI Embedded DRAM Macro for the POWER Processor 32 MByte On-Chip L3 Cache. *IEEE J. Solid-State Circuits* **2011**, *46*, 64–75.
5. Barth, J.; Reohr, W.; Parries, P.; Fredeman, G.; Golz, J.; Schuster, S.; Matick, R.E.; Hunter, H.; Tanner, C.; Harig, J.; et al. A 500 MHz Random Cycle, 1.5 ns Latency, SOI Embedded DRAM Macro Featuring a Three-Transistor Micro Sense Amplifier. *IEEE J. Solid-State Circuits* **2008**, *43*, 86–95.
6. Chen, K.; Li, S.; Muralimanohar, N.; Ahn, J.H.; Brockman, J.B.; Jouppi, N.P. CACTI-3DD: Architecture-level modeling for 3D die-stacked DRAM main memory. In Proceedings of the Design, Automation & Test in Europe Conference & Exhibition (DATE), Dresden, Germany, 12–16 March 2012; pp. 33–38.
7. Wilton, S.J.E.; Jouppi, N. CACTI: An enhanced cache access and cycle time model. *IEEE J. Solid-State Circuits* **1996**, *31*, 677–688.
8. Dong, X.; Xu, C.; Jouppi, N.; Xie, Y. NVSim: A circuit-level performance, energy, and area model for emerging nonvolatile memory. *IEEE Counc. Electron. Des. Autom.* **2012**, *31*, 994–1007.
9. Syu, S.M.; Shao, Y.-H.; Lin, I.-C. High-endurance hybrid cache design in CMP architecture with cache partitioning and access-aware policy. In Proceedings of the 23rd ACM international conference on Great lakes symposium on VLSI, Paris, France, 2–3 May 2013.

10. Mittal, S.; Poremba, M.; Vetter, J.; Xie, Y. *Exploring Design Space of 3D NVM and eDRAM Caches Using DESTINY Tool*; Technical Report ORNL/TM-2014/636; Oak Ridge National Laborator: Oak Ridge, TN, USA, 2014.
11. Zhang, C.; Sun, G.; Zhang, W.; Mi, F.; Li, H.; Zhao, W. Quantitative modeling of racetrack memory, a tradeoff among area, performance, and power. In Proceedings of the 20th Asia and South Pacific Design Automation Conference (ASP-DAC), Chiba, Japan, 19–22 January 2015; pp. 100–105.
12. Sun, G.; Dong, X.; Xie, Y.; Li, J.; Chen, Y. A novel architecture of the 3D stacked MRAM L2 cache for CMPs. In Proceedings of the IEEE 15th International Symposium on High Performance Computer Architecture, Raleigh, NC, USA, 14–18 February 2009; pp. 239–249.
13. Sun, Z.; Bi, X.; Wu, W.; Yoo, S.; Li, H.H. Array organization and data management exploration in racetrack memory. *IEEE Trans. Comput.* **2016**, *65*, 1041–1054.
14. Tsai, Y.F.; Xie, Y.; Vijaykrishnan, N.; Irwin, M.J. Three-Dimensional Cache Design Exploration Using 3DCacti. In Proceedings of the International Conference on Computer Design (ICCD), San Jose, CA, USA, 2–5 October 2005; pp. 519–524.
15. Prenat, G.; Jabeur, K.; Vanhauwaert, P.; Pendina, G.; Oboril, F.; Bishnoi, R.; Garello, K.; Gambardella, P.; Tahoori, M.; Gaudin, G. Ultra-Fast and High-Reliability SOT-MRAM: From Cache Replacement to Normally-off Computing. *IEEE Trans. Multi-Scale Comput. Syst.* **2016**, *2*, 49–60.
16. Kim, Y.; Fong, X.; Kwon, K.W.; Chen, M.C.; Roy, K. Multilevel Spin-Orbit Torque MRAMs. *IEEE Trans. Electron Devices* **2015**, *62*, 561–568.
17. Hong, S.; Lee, J.; Kim, S. Ternary cache: Three-valued MLC STT-RAM caches. In Proceedings of the International Conference on Computer Design (ICCD), Seoul, Korea, 19–22 October 2014; pp. 83–89.
18. Zhao, M.; Xue, Y.; Hu, J.; Yang, C.; Liu, T.; Jia, Z.; Xue, C.J. State Asymmetry Driven State Remapping in Phase Change Memory. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* **2017**, *36*, 27–40.
19. Sheu, S.S.; Chang, M.F.; Lin, K.F.; Wu, C.W.; Chen, Y.S.; Chiu, P.F.; Kuo, C.C.; Yang, Y.S.; Chiang, P.C.; Lin, W.P.; et al. A 4 Mb embedded SLC resistive-RAM macro with 7.2 ns read-write random-access time and 160 ns MLC-access capability. In Proceedings of the IEEE International Solid-State Circuits Conference, San Francisco, CA, USA, 20–24 February 2011; pp. 200–202.
20. Golz, J.; Safran, J.; He, B.; Leu, D.; Yin, M.; Weaver, T.; Vehabovic, A.; Sun, Y.; Cestero, A.; Himmel, B.; et al. 3D stackable 32 nm High-K/Metal Gate SOI embedded DRAM prototype. In Proceedings of the Symposium on VLSI Circuits (VLSIC), Kyoto, Japan, 15–17 June 2011; pp. 228–229.
21. Klim, P.; Barth, J.; Reohr, W.; Dick, D.; Fredeman, G.; Koch, G.; Le, H.; Khargonekar, A.; Wilcox, P.; Golz, J.; et al. A one MB cache subsystem prototype with 2 GHz embedded DRAMs in 45 nm SOI CMOS. In Proceedings of the 2008 IEEE Symposium on VLSI Circuits, Honolulu, HI, USA, 18–20 June 2008; pp. 206–207.
22. Kawahara, A.; Azuma, R.; Ikeda, Y.; Kawai, K.; Katoh, Y.; Tanabe, K.; Nakamura, T.; Sumimoto, Y.; Yamada, N.; Nakai, N.; et al. An 8 Mb multi-layered cross-point ReRAM macro with 443 MB/s write throughput. *IEEE J. Solid-State Circuits* **2013**, *48*, 178–185.
23. Hsu, C.L.; Wu, C.F. High-performance 3D-SRAM architecture design. In Proceedings of the IEEE Asia Pacific Conference on Circuits and Systems (APCCAS), Kuala Lumpur, Malaysia, 6–9 December 2010; pp. 907–910.
24. Puttaswamy, K.; Loh, G. 3D-Integrated SRAM Components for High-Performance Microprocessors. *Comput. IEEE Trans.* **2009**, *58*, 1369–1381.
25. Kim, H.; Park, J.H.; Park, K.T.; Kwak, P.; Kwon, O.; Kim, C.; Lee, Y.; Park, S.; Kim, K.; Cho, D.; et al. A 159 mm^2 32 nm 32 Gb MLC NAND-flash memory with 200 MB/s asynchronous DDR interface. In Proceedings of the IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC), San Fransico, CA, USA, 7–11 February 2010; pp. 442–443.
26. Cernea, R.; Pham, L.; Moogat, F.; Chan, S.; Le, B.; Li, Y.; Tsao, S.; Tseng, T.Y.; Nguyen, K.; Li, J.; et al. A 34 MB/s-program-throughput 16Gb MLC NAND with all-bitline architecture in 56 nm. In Proceedings of the IEEE International Solid-State Circuits Conference, San Francisco, CA, USA, 3–7 February 2008; pp. 420–624.
27. Mittal, S. A Survey of Techniques for Architecting TLBs. *Concurr. Comput.: Pract. Exper.* **2017**, *29*, 10.

28. Poremba, M.; Mittal, S.; Li, D.; Vetter, J.S.; Xie, Y. DESTINY: A Tool for Modeling Emerging 3D NVM and eDRAM caches. In Proceedings of the Design Automation and Test in Europe (DATE), Grenoble, France, 9–13 March 2015.
29. Kalla, R.; Sinharoy, B.; Starke, W.J.; Floyd, M. Power7: IBM's next-generation server processor. *IEEE Micro* **2010**, *30*, 7–15.
30. Zyuban, V.; Taylor, S.; Christensen, B.; Hall, A.; Gonzalez, C.; Friedrich, J.; Clougherty, F.; Tetzloff, J.; Rao, R. IBM POWER7+ design for higher frequency at fixed power. *IBM J. Res. Dev.* **2013**, *57*, 1.
31. Fluhr, E.J.; Friedrich, J.; Dreps, D.; Zyuban, V.; Still, G.; Gonzalez, C.; Hall, A.; Hogenmiller, D.; Malgioglio, F.; Nett, R.; et al. POWER8: A 12-core server-class processor in 22 nm SOI with 7.6 Tb/s off-chip bandwidth. In Proceedings of the IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC), San Francisco, CA, USA, 9–13 February 2014; pp. 96–97.
32. Mittal, S. A Survey of Architectural Techniques For Improving Cache Power Efficiency. *Sustain. Comput. Inform. Syst.* **2014**, *4*, 33–43.
33. Mittal, S. A Survey of Techniques for Architecting and Managing GPU Register File. *IEEE Trans. Parallel Distrib. Syst.* **2016**, *28*, 16–28.
34. Mamidipaka, M.; Dutt, N. *eCACTI: An Enhanced Power Estimation Model for On-Chip Caches*; Technical Report; TR-04-28, UC Irvine: Irvine, CA, USA, 2004.
35. Li, S.; Chen, K.; Ahn, J.H.; Brockman, J.B.; Jouppi, N.P. CACTI-P: Architecture-level Modeling for SRAM-based Structures with Advanced Leakage Reduction Techniques. In Proceedings of the International Conference on Computer-Aided Design, San Jose, CA, USA, 7–10 November 2011; pp. 694–701.
36. Mittal, S.; Vetter, J.S.; Li, D. A Survey Of Architectural Approaches for Managing Embedded DRAM and Non-volatile On-chip Caches. *IEEE Trans. Parallel Distrib. Syst.* **2015**, *26*, 1524–1537.
37. Mittal, S. A Survey of Techniques for Architecting Processor Components using Domain Wall Memory. *ACM J. Emerg. Technol. Comput. Syst.* **2016**, *13*, doi:10.1145/2994550.
38. Mittal, S.; Vetter, J. Reliability Tradeoffs in Design of Volatile and Non-volatile Caches. *J. Circuits Syst. Comput.* **2016**, *25*, 1650139.
39. Xu, C.; Niu, D.; Muralimanohar, N.; Jouppi, N.P.; Xie, Y. Understanding the trade-offs in multi-level cell ReRAM memory design. In Proceedings of the Design Automation Conference (DAC), Austin, TX, USA, 29 May–7 June 2013; pp. 1–6.
40. Mittal, S.; Vetter, J.S. A Survey of Software Techniques for Using Non-Volatile Memories for Storage and Main Memory Systems. *IEEE Trans. Parallel Distrib. Syst.* **2016**, *27*, 1537–1550.
41. Mittal, S. A Survey of Power Management Techniques for Phase Change Memory. *Int. J. Comput. Aided Eng. Technol.* **2016**, *8*, 424–444.
42. Byeon, D.S.; Lee, S.S.; Lim, Y.H.; Kang, D.; Han, W.k.; Kim, D.H.; Suh, K.D. A comparison between 63 nm 8 Gb and 90 nm 4 Gb multi-level cell NAND flash memory for mass storage application. In Proceedings of the IEEE Asian Solid-State Circuits Conference, Hsinchu, Taiwan, 1–3 November 2005; pp. 13–16.
43. Seong, N.H.; Yeo, S.; Lee, H.H.S. Tri-level-cell phase change memory: Toward an efficient and reliable memory system. In Proceedings of the International Symposium on Computer Architecture, Tel-Aviv, Israel, 23–27 June 2013; pp. 440–451.
44. Ishigaki, T.; Kawahara, T.; Takemura, R.; Ono, K.; Ito, K.; Matsuoka, H.; Ohno, H. A multi-level-cell spin-transfer torque memory with series-stacked magnetotunnel junctions. In Proceedings of the Symposium on VLSI Technology, Honolulu, HI, USA, 15–17 June 2010.
45. Wen, W.; Zhang, Y.; Mao, M.; Chen, Y. State-restrict MLC STT-RAM designs for high-reliable high-performance memory system. In Proceedings of the Design Automation Conference (DAC), San Francisco, CA, USA, 1–5 June 2014; pp. 1–6.
46. Mittal, S. A Survey of Soft-Error Mitigation Techniques for Non-Volatile Memories. *Computers* **2017**, *6*, 8.
47. Kirihata, T.; Parries, P.; Hanson, D.; Kim, H.; Golz, J.; Fredeman, G.; Rajeevakumar, R.; Griesemer, J.; Robson, N.; Cestero, A.; et al. An 800MHz embedded DRAM with a concurrent refresh mode. *IEEE Int. Solid-State Circuits* **2005**, *40*, 1377–1387.
48. Jiang, L.; Zhao, B.; Zhang, Y.; Yang, J. Constructing large and fast multi-level cell STT-MRAM based cache for embedded processors. In Proceedings of the Design Automation Conference (DAC), San Francisco, CA, USA, 3–7 June 2012; pp. 907–912.

49. Black, B.; Nelson, D.; Webb, C.; Samra, N. 3D processing technology and its impact on iA32 microprocessors. In Proceedings of the IEEE International Conference on Computer Design, San Jose, CA, USA, 11–13 October 2004; pp. 316–318.
50. Patti, R. Three-Dimensional Integrated Circuits and the Future of System-on-Chip Designs. *Proc. IEEE* **2006**, *94*, 1214–1224.
51. Chen, Y.; Wong, W.F.; Li, H.; Koh, C.K. Processor caches built using multi-level spin-transfer torque RAM cells. In Proceedings of the Low Power Electronics and Design (ISLPED), Fukuoka, Japan, 1–3 August 2011; pp. 73–78.
52. Mittal, S. A Survey of Techniques for Designing and Managing CPU Register File. *Concurr. Comput.: Pract. Exper.* **2016**, *29*, e3906.
53. Mittal, S. A Survey Of Cache Bypassing Techniques. *J. Low Power Electron. Appl.* **2016**, *6*, 5.
54. Mittal, S.; Vetter, J. A Survey Of Architectural Approaches for Data Compression in Cache and Main Memory Systems. *IEEE Trans. Parallel Distrib. Syst.* **2016**, *27*, 1524–1536.
55. Mittal, S.; Vetter, J. A Technique For Improving Lifetime of Non-volatile Caches using Write-minimization. *J. Low Power Electron. Appl.* **2016**, *6*, 1.
56. Mittal, S.; Vetter, J.S. AYUSH: A Technique for Extending Lifetime of SRAM-NVM Hybrid Caches. *IEEE Comput. Arch. Lett.* **2015**, *14*, 115–118.
57. Mittal, S. A Survey Of Architectural Techniques for Managing Process Variation. *ACM Comput. Surv.* **2016**, *48*, 54.
58. Mittal, S. A Survey Of Architectural Techniques for Near-Threshold Computing. *ACM J. Emerg. Tech. Comput. Syst.* **2015**, *12*, 46
59. Mittal, S. A Survey Of Techniques for Approximate Computing. *ACM Comput. Surv.* **2016**, *48*, 62.