Exploring Design Space of 3D NVM and eDRAM Caches Using DESTINY Tool

Sparsh Mittal*, Matt Poremba¶, Jeffrey S. Vetter*§, Yuan Xie♯
*Oak Ridge National Laboratory
¶Pennsylvania State University
§Georgia Institute of Technology
♯University of California at Santa Barbara
Email: mittals@ornl.gov,mrp5060@psu.edu,vetter@ornl.gov,yuanxie@ece.ucsb.edu

***Abstract*—To enable the design of large sized caches, novel memory technologies (such as non-volatile memory) and novel fabrication approaches (e.g. 3D stacking) have been explored. The existing modeling tools, however, cover only few memory technologies, CMOS technology nodes and fabrication approaches. We present DESTINY, a tool for modeling 3D (and 2D) cache designs using SRAM, embedded DRAM (eDRAM), spin transfer torque RAM (STT-RAM), resistive RAM (ReRAM) and phase change RAM (PCM). 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. DESTINY has been validated against several cache prototypes. We believe that DESTINY will boost studies of next-generation memory architectures used in systems ranging from mobile devices to extreme-scale supercomputers.**

Keywords*—*Cache, SRAM, eDRAM, non-volatile memory (NVM or NVRAM), STT-RAM, ReRAM, PCM, modeling tool, emerging memory technologies, validation.

I. Introduction

Due to recent trends of increasing system core-count and memory bandwidth bottleneck, processor designers are using large size on-chip caches. For example, Intel's Ivytown processor has 37.5MB SRAM LLC [2]. To overcome the limitations of SRAM, such as high leakage power consumption and low density, researchers have explored alternate memory technologies, such as eDRAM, STT-RAM, ReRAM and PCM [3, 4]. These memory technologies enable design of large size caches, for example, Intel's 22nm Haswell processor employs 128MB L4 eDRAM cache [5]. Researchers are also exploring novel fabrication techniques such as 3D integration that enables vertical stacking of multiple layers [6]. 3D stacking offers several benefits such as high density, higher flexibility in routing signals, power and clock and ability to integrate diverse memory technologies for designing hybrid caches.

Lack of open-source, comprehensive and validated modeling tools, however, presents a bottleneck in full study of emerging memory technologies and design approaches. Although a few modeling tools exist, they model only a subset of memory technologies, for example NVSim [7] models only 2D designs of SRAM and NVMs but not eDRAM. As an increasing number of industrial designs utilize 3D stacking [8, 9], research on 3D stacking has become even more important. Existing 3D modeling tools such as CACTI-3DD [10] and 3DCacti [11] do not model NVMs. Further, different tools use different assumptions and modeling frameworks, and hence, comparing the estimates obtained from different tools may be incorrect. Also, tools such as 3DCacti are not capable of modeling of recent technology nodes (e.g. 32nm). It is clear that a single, validated tool which can model both 2D and 3D designs using prominent memory technologies is lacking. In absence of such a tool, several architecture-level studies on 3D caches (e.g. [12]) derive their parameters using a linear extrapolation of 2D parameters which may be sub-optimal or even inaccurate.

A. Contributions

In this paper, we present DESTINY[1], a 3D design-space exploration tool for SRAM, eDRAM and non-volatile memory. DESTINY utilizes the 2D circuit-level modeling framework of NVSim tool for SRAM and NVMs. It also utilizes the coarse- and fine-grained TSV (through silicon via) models from CACTI-3DD tool. Further, DESTINY adds the model of eDRAM (Section III-A) and two additional types of 3D designs (Section III-B). Overall, DESTINY enables modeling of both 2D and 3D designs of five memory technologies (SRAM, eDRAM and three NVMs), which includes both volatile and non-volatile memories. Also, it can model technology nodes ranging from 22nm to 180nm. Finally, by virtue of being an open-source tool, it facilitates reproducible research and easy extension of the tool for many more usage scenarios than that discussed in the paper.

We have compared the results obtained from DESTINY against several commercial prototypes [8, 9, 13–17] to validate 2D design of eDRAM and 3D designs of SRAM, eDRAM and ReRAM in DESTINY (Section IV). The modeling error has been observed to be less than 10% for most cases and less than 20% 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 [7].

Sparsh Mittal and Matt Poremba are co-first authors. This ORNL technical report number ORNL/TM-2014/636 is an extension of our paper [1] accepted in DATE (Design, Automation and Test in Europe) conference, to be held between 9-13 March, 2015. The specific extensions made in this report are listed at the end of Section 1.

[1]The source-code of DESTINY can be cloned from the following `git` repository: https://code.ornl.gov/3d_cache_modeling_tool/destiny.git

DESTINY facilitates exploring a large design space which provides important insights and is also useful for early stage estimation of emerging memory technologies. For example, while it is relatively straightforward to deduce the optimal memory technology for some parameters (e.g. the technology with smallest cell size is likely to have 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, use of a modeling tool such as DESTINY is imperative for full design space exploration and optimization. We believe that DESTINY will be a useful tool in architecture and system-level studies and will assist researchers, designers and technical professionals. We conclude this paper by highlighting the possible extensions to DESTINY (Section VI).

This technical report extends the previous version [1] in several significant ways.

1) 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 V-A), finding the optimal number of layers (e.g. 2D design, 3D design with 1 layer, 3D design with 2 layers etc.) for a given optimization target (Section V-B) etc.
2) We have integrated estimates obtained from DESTINY into a performance simulator and conducted studies on a wide range of benchmarks to show that for different benchmarks, different number of layers are found optimal on considering optimization targets such as EDP and ED^2P (Section V-C).
3) We have discussed the use of DESTINY in gaining insights which are useful for designing architecture-level policies for last level caches fabricated with different memory technologies (Section V-D). These ideas are also useful for designing hybrid caches.
4) We have now discussed the trends in the design of modern processors which mandate use of novel memory technologies (Section II-A) and 3D die-stacking (Section II-B). Further, we have provided device-level data storage mechanism of each memory technology (Section II-C)
5) We have provided qualitative comparison of different modeling tools (Section II-D and Table I) to highlight the features of DESTINY. To motivate the need of a comprehensive modeling tool, we have compared results obtained from CACTI and NVSim which show that different tools may provide different results and in different format (refer Tables II and III).

II. Background and Related Work

A. Cache size trends in modern processors

With continued CMOS scaling, the number of cores on a chip have grown considerably. To feed data to these cores, the size of last level cache has also grown [18]. For example, IBM's 45nm Power7 processor had a 32MB LLC [19], the 32nm Power7+ processor had an 80MB LLC [20] and the 22nm Power8 processor has a 96MB LLC [21]. It is expected that in near future, CPUs will employ LLCs more than 100MB in size.

Similar trends are also true for GPUs. GPUs have been traditionally used for graphics applications which have limited data-reuse and hence, earlier GPUs had only simple cache hiearchy with small cache size. However, as GPUs become progressively employed in general-purpose applications, the cache sizes in GPUs have been steadily increasing. For example, the GT200 architecture GPUs did not feature an L2 cache, the Fermi GPU has 768KB LLC and the Kepler GPU has 1536KB LLC [22].

B. 3D stacking technology

While large LLCs minimize off-chip accesses by providing higher capacity, they may also lead to large access latencies. To address this, researchers have proposed several approaches, e.g. design of NUCA (non-uniform cache architecture) caches, 3D stacking [8] etc. which aim to bring data closer to the core.

3D stacking has also been shown to improve energy efficiency of GPU caches [23], and since power management is important for GPUs also [22], 3D stacking is considered a promising approach for GPU-based computing systems.

C. Data storage mechanism of memory technologies

We now briefly discuss the data storage mechanism of each memory technology. For more details, we refer the reader to previous work [7, 24].

STT-RAM: STT-RAM utilizes a magnetic tunnel junction (MTJ) as the primary memory storage. An MTJ consists of two ferromagnetic layers. The *reference* layer has a fixed magnetic polarization while a *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.

ReRAM: ReRAM uses a metal oxide material between two metal electrodes to store values. 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 low resistance state representing a "1". When a filament is broken down there is a small concentration of oxygen vacancies and thus high resistance state, representing "0".

PCM: PCM uses a chalcogenide material such as GeSbTe (GST) for data storage. 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 medium temperature (~300 °C) for a relatively long period of time (~150ns). 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 (~100ns) and quickly removing heat. This causes the material to melt and remain in an amorphous phase when cooled. The crystalline phase is low resistance corresponding to a "1" bit, while the amorphous phase is high resistance corresponding to a "0" bit.

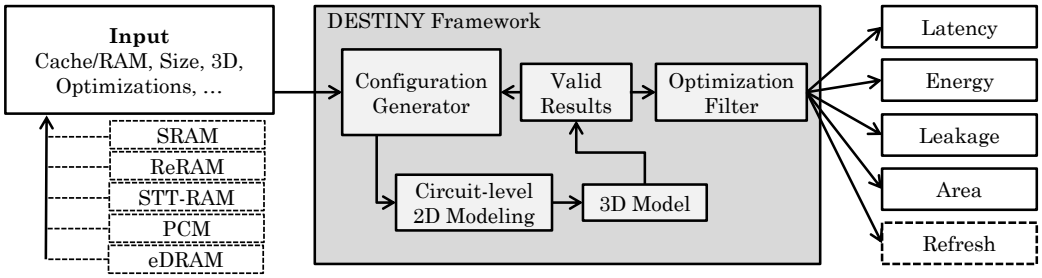

Fig. 1: High-level overview of DESTINY framework. Configurations are generated from extended input model and fed to NVSIM core. Results are fined tuned via 3D model and filtered by optimization to yield result outputs.

eDRAM: In eDRAM, the data are stored as a 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 [25].

D. A comparison of modeling Tools

Researchers have proposed several tools for modeling and estimating the energy consumption, 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 tool [26] simulates SRAM caches and has been extended to support eDRAM and DRAM. Also, several improvements have been made to CACTI to improve its modeling capability/accuracy. Mamidipaka and Dutt propose eCACTI [27] which adds a leakage model into CACTI and Li et al. [28] propose CACTI-P which models low-power caches (e.g., cache with sleep transistors). Chen el al. presented CACTI-3DD [10] 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 [11] provides the ability to model 3D SRAM, however this tool has not been updated to support technology nodes below 45nm. None of these tools model emerging NVMs. NVSim provides 2D modeling of SRAM, ReRAM, STT-RAM, PCM and SLC NAND Flash. However, none of these tools provide the *comprehensive* design space exploration capability as provided by DESTINY.

TABLE I: A comparison of modeling capabilities of different modeling tools

	Memory Technologies			
	eDRAM	NVMs	3D	sub-45nm
CACTI	✓	✗	✗	✓
CACTI-3DD	✓	✗	✓	✓
NVSim	✗	✓	✗	✓
3DCACTI	✗	✗	✓	✗
DESTINY	✓	✓	✓	✓

Several architectural studies use these tools (e.g. [4, 6, 18, 29–31]) and the conclusions derived by those studies depend crucially on these tools. Lack of *v*alidated tools may force the researchers to use tools which do not model the real prototype accurately and produce large errors. Lack of *c*omprehensive tools may force them to consider only few memory technologies which may not be optimal. Extrapolation of 2D tools to get parameters for 3D designs is prone to errors and in absence of 3D modeling tools, the researchers may conduct their studies with 2D designs only, which may not be sufficient for fully exploring the design-space. Some researchers use closed-source/in-house modeling tools (which may be extension of existing tools e.g. [12, 32]), however, experiments conducted with such tools may not be reproducible and their accuracy may not be verified. It is clear that an open-source, comprehensive, validated and up-to-date tool such as DESTINY will be highly useful for the researchers.

Further, each tool uses different inputs and provides the cache parameters in different ways. 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, one input in CACTI is "Type of wire outside mat" with options as semi-global or global. NVSim, however, does not seek an input with the same name. Further, 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 this input parameter can have marked influence on the output obtained from the tool. As an example, as we show in Table II, the output obtained for cache designs optimized for different targets can be vastly different. This makes it difficult for the user to have one-to-one correspondence between the inputs and outputs of two tools. The reason behind this is that both NVSim and CACTI use different modeling framework.

III. Modeling Framework

DESTINY is a comprehensive tool able to model multiple memory technologies. Figure 1 presents a high-level diagram of DESTINY. DESTINY framework utilizes the 2D circuit-level model of NVSim, which has been extended to model 2D eDRAM and 3D design of SRAM, eDRAM and monolithic NVMs. For a given memory technology, the device-level parameters (e.g. cell size, set-voltage, reset voltage) are provided as input to DESTINY. Then, possible configurations are generated which are passed to the circuit-level modeling code. For 3D designs, 3D modeling is also done and the

TABLE II: Results obtained from NVSim for three different optimization targets (Lat. = latency, En. = energy). Cache parameters: SRAM 4MB 16-way cache, 64B block size, 32nm. Clearly, for different optimization targets, the outputs can be vastly different.

	Area (mm^2)	Hit Lat. (ns)	Miss Lat. (ns)	Write Lat. (ns)	Hit En. (nJ)	Miss En. (nJ)	Write En. (nJ)	Leakage Power (W)
Area	5.375	34.03	8.16	25.88	0.698	0.042	0.086	0.325
Read Latency	6.747	2.01	0.31	1.08	0.388	0.032	0.363	0.395
Read Dynamic Energy	10.156	109.37	16.79	92.5	0.108	0.004	0.088	0.464

TABLE III: Comparison of CACTI and NVSim results for same cache configurations shown in Table II (dynE = dynamic energy). Clearly, both the output values and output format for each tool are different

CACTI	NVSim
Area: 14.90 mm^2	Area: 6.75mm^2
Leakage: 0.574W	Leakage: 0.395W
Access time: 3.119ns Random cycle time: 0.634ns	Hit latency: 2.009ns Miss latency: 0.314ns Write latency: 1.079ns
Read dynE: 0.182nJ	Hit dynE: 0.388nJ Miss dynE: 0.032nJ Write dynE: 0.363nJ

generated configurations can have different number of 3D layers (e.g. 1, 2, 4, 8, 16 etc.). Those designs which 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 V-A, DESTINY also provides 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 left of Figure 1). 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 V-B).

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

A. eDRAM Model

NVSim provides an incomplete model of eDRAMs 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 40μs – 100μs [8, 9] for temperature in the range of 380K. We implemented a refresh model based on Kirihata et al. [33], 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 which 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 [30], 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.

B. 3D Model

Several types of 3D stacking have been explored in the literature, such as face-to-face, face-to-back, and monolithic stacking [34, 35]. In all 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 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 it has adverse effect on alignment.

The most common 3D stacking is known as face-to-back bonding. In this form, through silicon vias (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 must carefully consider placement and avoid transistors when being formed through the bulk silicon while face-to-face does not. Therefore, face-to-face offers the potential for higher via density. The downside of face-to-face 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 which 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 [10]. 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 [17]. Second, the monolithic stacking model for 3D horizontal ReRAM is provided [15]. Face-to-back bonding using TSVs is utilized in other designs [13, 16].

A few previous works (e.g. [10]) 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 which are modeled in CACTI-3DD, but is unacceptable in caches which 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 which do not require extra buffering. In this work, we provide a TSV model which may act as a buffered or unbuffered TSV as well as vias used in face-to-face bonding.

With 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 the layers. In the coarse-grained model, TSVs are used to broadcast *undecoded* row and column select signals to all 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 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) [15] uses a concept similar to cross-point designs (refer Figure 2). 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.

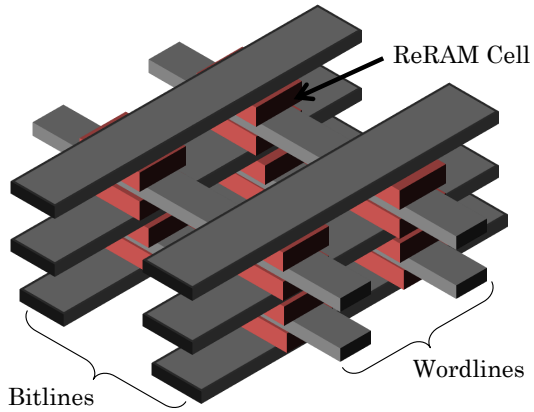

Fig. 2: 4-layer monolithically stacked ReRAM. Storage elements are sandwiched directly between layers of wordlines (south-east orientation) and bitlines (south-west orientation).

An example of HReRAM is shown in Figure 2 with 4 layers monolithically stacked. This monolithic stacking implies that 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 which are considered optimal for a 2D design. Typically this effect can be slightly mitigated using diode accessed cells as in [15] and hence, we provide the option to model diodes or no access device.

IV. Validation Results

To evaluate the accuracy of DESTINY, we validate it against several industrial prototypes. We obtain the cache/macro configuration from the corresponding prototype papers and use them to set the device-level input parameters for DESTINY. Finally, we compare the results from actual value and 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 20% in all cases.

A. 3D SRAM Validation

We validate the 3D SRAM model of DESTINY against two previous works [16, 17] which utilize hSpice models to simulate latency and energy of 3D-stacked SRAM caches. Hsu and Wu [16] sweep over various cache sizes ranging from 1MB to 16MB. Their work assumes stacking at the bank level, that is, a 2D planar cache containing N banks can be stacked up to N-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[2]. 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 IV shows the validation results of DESTINY against the 3D SRAM design in [16]. Notice that the errors are consistently less than 4%.

Puttaswamy and Loh [17] explore the design space of 3D SRAM for 65nm technology node. Their work considers a range of cache sizes from 16KB to 4MB. 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 IV. Clearly, the errors are always less than 15% which shows that DESTINY is reasonably accurate in modeling 3D SRAM caches.

B. 2D and 3D eDRAM Validation

As stated before, the eDRAM model in NVSim is incomplete and has not been validated against any prototype. Hence,

[2]The bank layouts are specified as mat×mat, and subarray layouts are specified as wordline×bitline.

TABLE IV: Validation for 3D SRAM model.

Design	Metric	Actual	Projected (DESTINY)	Error
1MB [16] 2 layers	Latency	1.85 ns	1.91 ns	+3.54%
	Energy	5.10 nJ	5.05 nJ	-0.98%
1MB [16] 4 layers	Latency	1.75 ns	1.80 ns	+2.68%
	Energy	4.5 nJ	4.51 nJ	+0.18%
4MB [17] 2 layers	Latency	7.85 ns	7.23 ns	-7.91%
	Energy	0.13 nJ	0.13 nJ	-2.59%
4MB [17] 4 layers	Latency	6.10 ns	6.95 ns	+14.03%
	Energy	0.12 nJ	0.13 nJ	+4.75%
2MB [17] 2 layers	Latency	5.77 ns	5.78 ns	+0.05%
	Energy	0.12 nJ	0.13 nJ	+2.74%
2MB [17] 4 layers	Latency	4.88 ns	5.53 ns	+13.5%
	Energy	0.12 nJ	0.13 nJ	+8.46%
1MB [17] 2 layers	Latency	3.95 ns	3.90 ns	-1.11%
	Energy	0.11 nJ	0.11 nJ	-0.13%
1MB [17] 4 layers	Latency	3.07 ns	3.04 ns	-0.85%
	Energy	0.11 nJ	0.11 nJ	-0.89%

we validate both 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, ECC logic, etc.) and are closest to the modeling assumptions of DESTINY and hence we compare against a macro. For fair comparison, we remove ECC, spare, and parity wordlines and bitlines as these are not modeled in DESTINY.

TABLE V: Validation of 2D and 3D eDRAM.

Design	Metric	Actual	Projected (DESTINY)	Error
2D 2Mb 65nm [9]	Latency	<2 ns	1.46 ns	—
	Area	0.665 mm^2	0.701 mm^2	+5.42%
2D 1Mb 45nm [8]	Latency	1.7 ns	1.73 ns	+1.74%
	Area	0.239 mm^2	0.234 mm^2	-2.34%
2D 2.25Mb 45nm [14]	Latency	1.8 ns	1.75 ns	-2.86%
	Area	0.420 mm^2	0.442 mm^2	+5.31%
3D 1Mb 2-layers[13]	Latency	<1.5 ns	1.42 ns	—
	Area	0.139 mm^2	0.149 mm^2	+9.32%

Barth and Reohr et al. [9] present a 65nm 2D eDRAM prototype. To validate against it, we use the 2Mb macro layout with total 8 subarrays and thus, use the organization of a 1024×2048 bank layout. Klim et al. [14] and Barth and Plass et al. [8] present 45nm 2D eDRAM designs. We validate against them using a subarray layout of 256×1024 as used by them. From Table V, it is clear that the modeling errors in 2D eDRAM validation for all cases is less than 6%.

Golz et al. present a 3D eDRAM prototype with 2 layers in 32nm technology [13]. We use the 1Mb array as our validation target. Based on the 16Kb μArray size of 32×512 and 1Mb layout, we assume two 1024×512 subarrays. From Table V, the modeling error in area is less than 10% and thus, DESTINY can be accepted as reasonably accurate.

C. 3D ReRAM Validation

Our final validation target is a monolithically stacked ReRAM memory [15], also known as 3D *horizontal* ReRAM [3]. 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 a RAM 8Mb in size. The design consists of 4 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 VII.

TABLE VII: Validation of 3D ReRAM.

Metric	Actual [15]	Projected (DESTINY)	Error
Read Latency	25 ns	24.16 ns	-3.37%
Write Latency	17.20 ns	20.13 ns	+17.05%

It is clear that the read latency projection of DESTINY is very close to the value reported in [15] while the error in write latency is higher. This can be attributed to the fact that Kawahara et al. [15] 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 8.2ns – 55ns. Our prediction falls in the lower end of the plot which is closer to the 8.2ns write pulse for a total of 17.2ns write time at the bank level.

V. Design Space Exploration using DESTINY

With increased number of options for memory technologies and fabrication techniques, the number of possible design options increase exponentially. Hence, a designer must make a right choice to optimize for a given target. DESTINY can be a very useful design-space-exploration and decision-support tool for such scenarios. In this section, we present features of DESTINY to demonstrate this.

A. 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 five options viz. ReRAM, STT-RAM, PCM, eDRAM, and SRAM. Each cache has same configuration, viz. 1-layer 32MB 16-way cache designed with 32nm node. Table VI shows the optimal memory technology found by DESTINY for each of the seven different optimization targets.

The results can be understood as follows. The ReRAM is designed as a cross-point style memory which is the most area efficient way to design ReRAM, resulting in low area usage. STT-RAM has the lowest read latency, however the energy required is higher than that of eDRAM resulting in eDRAM design having the lowest read energy and read EDP. NVMs are known to have high write latency/energy and hence,

[3]Note that vertical ReRAM design has also been proposed which reduces the fabrication cost, however we are not aware of any prototypes.

TABLE VI: Design space exploration results of determining the optimal memory technology for a desired optimization target (refer Section V-A). The table shows results on all parameters for comparison purposes.

Optimization Target	Optimal Technology	Area	Read Latency	Write Latency	Read Energy	Write Energy	Read EDP	Write EDP
Area	ReRAM	**0.95 μm^2**	46.66 ns	58.57 ns	0.24 nJ	0.14 nJ	11.11	8.00
Read Latency	STT-RAM	6.61 μm^2	**2.78 ns**	5.40 ns	1.12 nJ	0.74 nJ	3.10	4.01
Write Latency	eDRAM	16.95 μm^2	3.56 ns	**1.76 ns**	0.52 nJ	0.50 nJ	1.85	0.88
Read Energy	eDRAM	11.83 μm^2	78.74 ns	40.21 ns	**0.17 nJ**	0.40 nJ	13.13	15.92
Write Energy	SRAM	47.08 μm^2	126.84 ns	86.05 ns	2.40 nJ	**0.02 nJ**	304.02	1.67
Read EDP	eDRAM	13.20 μm^2	3.92 ns	2.74 ns	0.29 nJ	0.35 nJ	**1.15**	0.95
Write EDP	eDRAM	14.15 μm^2	3.28 ns	1.82 ns	0.38 nJ	0.35 nJ	1.24	**0.64**

TABLE VIII: Design space exploration results of determining optimal number of 3D-stacked layers for various optimization targets for STT-RAM (refer Section V-B). The table shows results on all parameters for comparison purposes.

Optimization Target	Optimal Layer Count	Area	Read Latency	Write Latency	Read Energy	Write Energy	Read EDP	Write EDP
Area	16	**2.07** μm^2	51.14 ns	45.30 ns	2.87 nJ	2.63 nJ	146.68	119.27
Read Latency	16	2.90 μm^2	**1.77 ns**	5.38 ns	3.15 nJ	2.90 nJ	5.58	15.60
Write Latency	16	3.15 μm^2	1.82 ns	**5.36 ns**	3.17 nJ	2.92 nJ	5.77	15.63
Read Energy	1	19.21 μm^2	148.11 ns	113.43 ns	**0.36 nJ**	0.14 nJ	53.13	15.48
Write Energy	1	19.78 μm^2	148.11 ns	113.43 ns	0.77 nJ	**0.13 nJ**	113.89	15.28
Read EDP	4	7.59 μm^2	2.94 ns	6.23 ns	0.91 nJ	0.69 nJ	**2.67**	4.28
Write EDP	2	12.31 μm^2	5.63 ns	6.93 ns	0.70 nJ	0.48 nJ	3.95	**3.31**

they are not optimal for write latency/energy/EDP. The write energy of SRAM is lower than eDRAM, however the write latency for SRAM is much higher than that of eDRAM when optimized for write energy, resulting in eDRAM having the lowest write EDP. The major reason why eDRAM is optimal over SRAM in terms of latency is due to the size of the H-tree based interconnect in our design. Since our 32MB cache is designed as a single bank, H-tree latency dominates in both designs. However, the SRAM cell size is approximately four times larger than that of eDRAM (146 F^2 vs. 39 F^2), which makes the size of the SRAM cache nearly four times larger. This increase in overall size has a direct impact on the total latency.

It is noteworthy that PCM was not found optimal for any optimization target used by us. PCM typically has very high read and write energy and latency compared to the other technologies. For area optimization, PCM comes close to ReRAM (0.97 μm^2 for PCM compared to 0.95 μm^2 for ReRAM), however due to the relatively small cache size (32MB) the peripheral circuitry required for PCM makes its area larger than that of ReRAM. 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 also noteworthy that the results obtained here hold for the particular cell-level parameters used as input for each technology and other parameters/configurations may yield different technologies as optimal for each target.

B. Finding the optimal layer count in 3D stacking

We now show the capability of DESTINY to find optimal number of 3D die layers for a given optimization target. In this case, DESTINY explores both a 2D design and different number 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 32nm, 32MB, 16 ways (same as above). Table VIII shows the results. It is clear that for different optimization targets, different number of layers are found as optimal.

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 4MB, thus the design with maximum number of layers is not selected as the optimal result.

The energy is minimized by avoiding the overhead of TSVs and hence, the design with 1 layer, i.e. a 2D design consumes the least amount of energy. The optimization of EDP presents an interesting case, since, as shown above, the trends of variation in energy and latency values are opposite. For this reason, it is expected that an intermediate value of layer count will be optimal for EDP. With increasing layer count, the write latency decreases at a slower rate than the read latency. For this reason, the optimal value of write EDP is obtained at 2 layers while that of read EDP is obtained at 4 layers.

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.

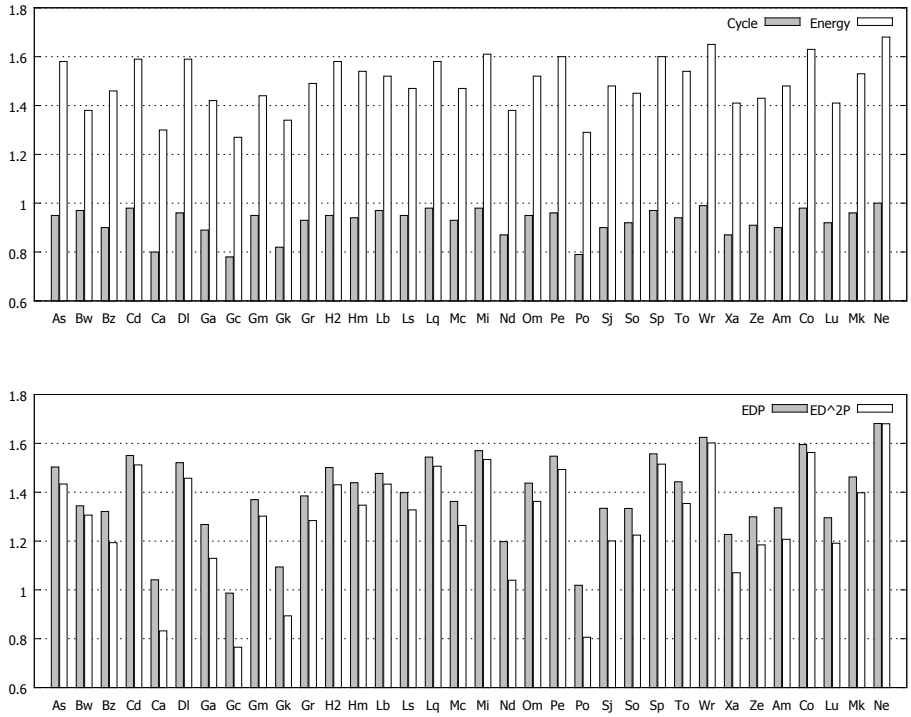

Fig. 3: Ratio of simulation-cycles, energy, EDP and ED^{2}P for 2layer design over 1layer design for different benchmarks obtained from architectural simulation

C. Design of workload-specific caches

We use interval-core model in Sniper x86-64 simulator for performing the simulations. We perform single-core simulations. We use all 29 benchmarks from SPEC CPU2006 suite with *ref* inputs and 5 benchmarks from HPC (high-performance computing) field (shown in italics in Table IX).

TABLE IX: Workloads used in the experiments

As(astar), Bw(bwaves), Bz(bzip2), Cd(cactusADM)
Ca(calculix), Dl(dealII), Ga(gamess), Gc(gcc)
Gm(gemsFDTD), Gk(gobmk), Gr(gromacs), H2(h264ref)
Hm(hmmer), Lb(lbm), Ls(leslie3d), Lq(libquantum)
Mc(mcf), Mi(milc), Nd(namd), Om(omnetpp)
Pe(perlbench), Po(povray), Sj(sjeng), So(soplex)
Sp(sphinx), To(tonto), Wr(wrf), Xa(xalancbmk)
Ze(zeusmp), *Co(CoMD), Lu(lulesh), Mk(mcck)*
Ne(nekbone), Am(amg2013)

The processor frequency is 2GHz. Both L1-I and L1-D caches are 4-way 16KB caches and have a latency of 2 cycles. The L2 cache is designed using ReRAM. We compute the energy consumption of only L2 cache and not other components of the processor. The L2 cache size and associativity are fixed to 4MB and 16-way, respectively. We test with two design options, viz. a 2D design i.e. single layer (called "1layer") and a 3D design with 2 layers (called "2layer"). Table X shows the parameters obtained for these two designs using DESTINY.

The simulations were performed for 300M instructions. We evaluate four metrics, viz. simulation cycles, energy consumption, EDP (energy-delay product) and ED^{2}P (energy delay-squared product). For each of these metrics, we show the ratio of their value for 2layer design over that for 1layer design. The results are shown in Figure 3. For example, in the figure, the gray values marked "Cycle" show the following quantity for different benchmarks:

$$\frac{\text{SimulationCycle}_{\text{2layer}}}{\text{SimulationCycle}_{\text{1layer}}}$$

Similar point also applies for energy, EDP and ED^{2}P. Since for each of these four metrics, smaller value is better, a ratio of less than 1 indicates that the 2layer design is better and a ratio of greater than 1 indicates that the 1layer design is better.

From Figure 3, it is clear that for all benchmarks, the 2layer design achieves lower simulation time and hence, provides better performance. This can be inferred from Table X, as the

TABLE X: Parameters obtained using DESTINY for 1layer and 2layer cache designs

	1layer	2layer
Hit Latency (ns)	11.435	3.989
Miss Latency (ns)	1.702	1.697
Write Latency (ns)	26.978	22.744
Hit Energy (nJ)	0.335	0.363
Miss Energy (nJ)	0.335	0.363
Write Energy (nJ)	0.068	0.094
Leakage Power (W)	0.091	0.155

1layer cache has higher cache access latency. As for energy, the 1layer design leads to lower energy consumption for all benchmarks and hence, it is better. This is due to the fact that the 2layer cache dissipates large amount of leakage power. Since more than 90% of energy consumed by the last level cache is in the form of leakage energy [36], the total energy consumed by 2layer configuration becomes high.

On EDP metric, the 1layer cache design is still better on nearly all benchmarks. This is because, its relative energy advantage is higher than its relative disadvantage of larger L2 access latency. A small increase in LLC latency is typically hidden due to write-buffer, out-of-order execution etc. [36]. Since EDP gives equal weightage to both energy and performance, the 1layer design is found to be better.

On ED^2P metric, the 2layer design is found to better for calculix, gcc, gobmk and povray. For remaining benchmarks, the 1layer design is better. The use of ED^2P metric indicates giving higher priority to the performance. For the above four benchmarks, the performance advantage of 2layer design is high and hence, for these benchmarks, the 2layer design is found to be higher.

It is clear from this architectural study that out of a 2D or 3D cache design, the choice of best design depends on both the metric used and the workload. Further, DESTINY can be a very useful tool for early-stage design exploration.

D. Gaining insights for designing architectural techniques

The latency, area and energy parameters provided by DESTINY can help the architects in finding the strength and weaknesses of each memory technology. This can be useful from two perspectives. First, it can help in selecting the best optimization target (e.g. write latency, area, write EDP etc.) during the cache design stage. Second, it can help in designing a suitable architectural/runtime/compiler technique for managing the cache designed with that technology at a particular level in cache hiearchy. Third, it can also guide design of a hybrid cache composed of multiple memory technologies. In what follows, we

It is clear that no single memory technology is superior on all parameters.

Techniques for SRAM: SRAM has low access latency, however, its density is low and its leakage power consumption is also high. Hence, SRAM is suitable for designing L1 caches which are optimized for low access latency. The last level cache designed with SRAM may consume large leakage energy and hence, suitable techniques for reducing their leakage power need to be applied [18, 36].

Techniques for eDRAM: Due to its low leakage and higher density (than SRAM), eDRAM is suitable for last level caches. However, due to its low retention period, eDRAM requires frequent refresh operations and hence, effective techniques for reducing the refresh energy of eDRAM caches are required [4, 30]. Also, the retention period of eDRAM reduces exponentially with increasing temperature and hence, effective thermal management techniques are also required. Since refresh operations interfere with cache access, eDRAM is not suitable for L1 caches.

Techniques for NVMs: Due to their high density and near-zero leakage, NVMs are very suitable for LLCs (note that, their limited write endurance need to overcome for achieving reasonable cache lifetimes [24, 29, 37]). However, their write latency and energy are high which make them unsuitable for designing L1 caches (Note that, some researchers have proposed techniques to trade-off the retention time of STT-RAM to improve its performance [32], which can enable STT-RAM to be used for designing L1 caches). Also, at architecture level, techniques need to be designed to manage dynamic energy, reduce effective write latency and number of write operations etc. [24, 29, 38]. Among three NVMs, PCM has the largest write latency, due to which, PCM is suitable for L4 cache [6] or main memory [38].

Further, it is well-known that read and write requests have different criticalities. A read miss in cache stalls the processor (unless there are independent instructions to be executed), however, most write misses do not lie on critical path since cache writes can be buffered. For this reason, it is important to treat read and write requests differently. From the output of DESTINY, it is clear that write latency of NVMs is much larger than their read latency [29]. This fact makes it even more important to distinguish between read and write requests in NVMs, since a long write request may lead to blocking of cache ports which can lead to stalling of processor. This observation is useful for designing cache management policies which give higher priority to serving read requests.

Techniques for SRAM-NVM hybrid caches: We show the issues involved in designing an SRAM-NVM hybrid cache and similar ideas also apply to other hybrid cache designs. The hybrid cache can be designed across the sets (where different sets are designed using SRAM and NVM, respectively) or across the ways (where different ways are designed using SRAM and NVM, respectively). Of these, the latter design is more common [6, 24, 29, 37]. The reason for this is that way-based hybrid cache design allows easy migration of data between SRAM and NVM ways, which allows leveraging the properties of both memory technologies.

Since SRAM consumes large leakage power and has low density, only few ways are designed using SRAM, e.g. in a 16-way cache, one to four ways are commonly designed with SRAM. Since NVMs are write-agnostic, the write-intensive blocks can be directed to SRAM and the remaining blocks are directed to NVM [6, 12, 24, 37]. Also, the hot data items can be stored in SRAM. This helps in achieving the density of NVM at the access speed of SRAM.

A related but different approach is to use SRAM as a buffer for capturing the hot data or as a write-buffer for minimizing the writes to NVM cache [12]. An SRAM buffer of size 32 or 64 can capture a large fraction of hot blocks.

VI. Future Work and Conclusion

Due to the emerging nature of these memory technologies, only a limited number of prototypes have 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 also. We plan to perform these validations as these prototypes become available. Further, we

plan to extend DESTINY to model MLC (multi level cell) support for NVMs and also model other emerging memory technologies such as race track memory [24]. Furthermore, we plan to fully integrate DESTINY in a performance simulator to enable architecture/system-level study of these technologies at different levels in cache hierarchy and find the optimal memory technology for a given workload. Since DESTINY has been written in C++, it can be easily integrated with existing simulators.

The tools such as DESTINY or CACTI provide energy estimates for core memory array and decoder etc., however, the other components of the cache such as replacement logic, write-buffer etc. also consume a sizable fraction of energy [39]. DESTINY can be extended to also model the energy of these components as shown in previous works (e.g. [39]). We also plan to improve the speed of DESTINY by using techniques such as multithreading.

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

Acknowledgment

This manuscript has been authored by UT-Battelle, LLC under Contract No. DE-AC05-00OR22725 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan). This work was performed when Matthew Poremba was working as a graduate student intern at Oak Ridge National Laboratory.

References

[1] M. Poremba, S. Mittal, D. Li, J. S. Vetter, and Y. Xie, "DESTINY: A Tool for Modeling Emerging 3D NVM and eDRAM caches," in *Design Automation and Test in Europe (DATE)*, 2015.

[2] S. Rusu, H. Muljono, D. Ayers, S. Tam, W. Chen, A. Martin, S. Li, S. Vora, R. Varada, and E. Wang, "Ivytown: A 22nm 15-core enterprise Xeon® processor family," in *IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, 2014, pp. 102–103.

[3] J. S. Vetter and S. Mittal, "Opportunities for nonvolatile memory systems in extreme-scale high performance computing," *Computing in Science and Engineering*, 2015.

[4] S. Mittal, "A Cache Reconfiguration Approach for Saving Leakage and Refresh Energy in Embedded DRAM Caches," Iowa State University, USA, Tech. Rep., 2013.

[5] N. Kurd *et al.*, "Haswell: A family of IA 22nm processors," in *IEEE ISSCC*, 2014, pp. 112–113.

[6] X. Wu, J. Li, L. Zhang, E. Speight, R. Rajamony, and Y. Xie, "Hybrid cache architecture with disparate memory technologies," in *ACM SIGARCH Computer Architecture News*, vol. 37, no. 3, 2009, pp. 34–45.

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

[8] J. Barth, D. Plass, E. Nelson, C. Hwang, G. Fredeman, M. Sperling, A. Mathews, T. Kirihata, W. Reohr, K. Nair, and N. Caon, "A 45 nm SOI Embedded DRAM Macro for the POWER Processor 32 MByte On-Chip L3 Cache," *Solid-State Circuits, IEEE Journal of*, vol. 46, no. 1, pp. 64–75, Jan 2011.

[9] J. Barth, W. Reohr, P. Parries, G. Fredeman, J. Golz, S. Schuster, R. E. Matick, H. Hunter, C. Tanner, J. Harig, H. Kim, B. Khan, J. Griesemer, R. Havreluk, K. Yanagisawa, T. Kirihata, and S. Iyer, "A 500 MHz Random Cycle, 1.5 ns Latency, SOI Embedded DRAM Macro Featuring a Three-Transistor Micro Sense Amplifier," *Solid-State Circuits, IEEE Journal of*, vol. 43, no. 1, pp. 86–95, Jan 2008.

[10] K. Chen, S. Li, N. Muralimanohar, J. H. Ahn, J. B. Brockman, and N. P. Jouppi, "CACTI-3DD: Architecture-level modeling for 3D die-stacked DRAM main memory," in *Conference on Design, Automation and Test in Europe*, 2012, pp. 33–38.

[11] Y.-F. Tsai, Y. Xie, N. Vijaykrishnan, and M. J. Irwin, "Three-dimensional cache design exploration using 3DCacti," in *International Conference on Computer Design (ICCD)*, 2005, pp. 519–524.

[12] G. Sun, X. Dong, Y. Xie, J. Li, and Y. Chen, "A novel architecture of the 3D stacked MRAM L2 cache for CMPs," in *IEEE 15th International Symposium on High Performance Computer Architecture (HPCA)*, 2009, pp. 239–249.

[13] J. Golz, J. Safran, B. He, D. Leu, M. Yin, T. Weaver, A. Vehabovic, Y. Sun, A. Cestero, B. Himmel, G. Maier, C. Kothandaraman, D. Fainstein, J. Barth, N. Robson, T. Kirihata, K. Rim, and S. Iyer, "3D stackable 32nm High-K/Metal Gate SOI embedded DRAM prototype," in *Symposium on VLSI Circuits (VLSIC)*, 2011, pp. 228–229.

[14] P. Klim, J. Barth, W. Reohr, D. Dick, G. Fredeman, G. Koch, H. Le, A. Khargonekar, P. Wilcox, J. Golz, J. B. Kuang, A. Mathews, T. Luong, H. Ngo, R. Freese, H. Hunter, E. Nelson, P. Parries, T. Kirihata, and S. Iyer, "A one MB cache subsystem prototype with 2GHz embedded DRAMs in 45nm SOI CMOS," in *VLSI Circuits, 2008 IEEE Symposium on*, June 2008, pp. 206–207.

[15] A. Kawahara, R. Azuma, Y. Ikeda, K. Kawai, Y. Katoh, K. Tanabe, T. Nakamura, Y. Sumimoto, N. Yamada, N. Nakai, S. Sakamoto, Y. Hayakawa, K. Tsuji, S. Yoneda, A. Himeno, K. Origasa, K. Shimakawa, T. Takagi, T. Mikawa, and K. Aono, "An 8Mb multi-layered cross-point ReRAM macro with 443MB/s write throughput," in *IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, 2012, pp. 432–434.

[16] C.-L. Hsu and C.-F. Wu, "High-performance 3D-SRAM architecture design," in *IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)*, 2010, pp. 907–910.

[17] K. Puttaswamy and G. Loh, "3D-Integrated SRAM Components for High-Performance Microprocessors," *Computers, IEEE Transactions on*, vol. 58, no. 10, pp. 1369–1381, Oct 2009.

[18] S. Mittal, "A survey of architectural techniques for improving cache power efficiency," *Elsevier Sustainable Computing: Informatics and Systems*, vol. 4, no. 1, pp. 33–43, 2014.

[19] R. Kalla, B. Sinharoy, W. J. Starke, and M. Floyd, "Power7: IBM's next-generation server processor," *Micro, IEEE*, vol. 30, no. 2, pp. 7–15, 2010.

[20] V. Zyuban, S. Taylor, B. Christensen, A. Hall, C. Gonzalez, J. Friedrich, F. Clougherty, J. Tetzloff, and R. Rao, "IBM POWER7+ design for higher frequency at fixed power," *IBM Journal of Research and Development*, vol. 57, no. 6, pp. 1–1, 2013.

[21] E. J. Fluhr, J. Friedrich, D. Dreps, V. Zyuban, G. Still, C. Gonzalez, A. Hall, D. Hogenmiller, F. Malgioglio, R. Nett *et al.*, "POWER8: A 12-core server-class processor in 22nm SOI with 7.6 Tb/s off-chip bandwidth," in *IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, 2014, pp. 96–97.

[22] S. Mittal, "A Survey Of Techniques for Managing and Leveraging Caches in GPUs," *Journal of Circuits, Systems, and Computers (JCSC)*, vol. 23, no. 8, 2014.

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

[24] S. Mittal, J. S. Vetter, and D. Li, "A Survey Of Architectural Approaches for Managing Embedded DRAM and Non-volatile On-chip Caches," *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, 2014.

[25] S. Mittal, J. S. Vetter, and D. Li, "Improving energy efficiency of Embedded DRAM Caches for High-end Computing Systems," in *23rd International ACM Symposium on High Performance Parallel and Distributing Computing (HPDC)*, 2014, pp. 99–110.

[26] S. J. Wilton and N. P. Jouppi, "Cacti: An enhanced cache access and cycle time model," *Solid-State Circuits, IEEE Journal of*, vol. 31, no. 5, pp. 677–688, 1996.

[27] M. Mamidipaka and N. Dutt, "eCACTI: An enhanced power estimation model for on-chip caches," TR-04-28, CECS, UCI, Tech. Rep., 2004.

[28] S. Li, K. Chen, J. H. Ahn, J. B. Brockman, and N. P. Jouppi, "CACTI-P: Architecture-level Modeling for SRAM-based Structures with Advanced Leakage Reduction Techniques," in *Proceedings of the International Conference on Computer-Aided Design*, ser. ICCAD '11, 2011, pp. 694–701.

[29] S. Mittal and J. S. Vetter, "Addressing inter-set write-variation for improving lifetime of non-volatile caches," in *5th Annual Non-Volatile Memories Workshop*, 2014.

[30] A. Agrawal, A. Ansari, and J. Torrellas, "Mosaic: Exploiting the Spatial Locality of Process Variation to Reduce Refresh Energy in On-Chip eDRAM Modules," in *HPCA*, 2014.

[31] S. Mittal and J. S. Vetter, "EqualWrites: Reducing Intra-set Write Variations for Enhancing Lifetime of Non-volatile Caches ," *IEEE Transactions on VLSI Systems*, 2015.

[32] C. W. Smullen, V. Mohan, A. Nigam, S. Gurumurthi, and M. R. Stan, "Relaxing non-volatility for fast and energy-efficient STT-RAM caches," in *17th International Symposium on High Performance Computer Architecture (HPCA)*. IEEE, 2011, pp. 50–61.

[33] T. Kirihata, P. Parries, D. Hanson, H. Kim, J. Golz, G. Fredeman, R. Rajeevakumar, J. Griesemer, N. Robson, A. Cestero, M. Wordeman, and S. Iyer, "An 800MHz embedded DRAM with a concurrent refresh mode," in *Solid-State Circuits Conference, 2004. Digest of Technical Papers. ISSCC. 2004 IEEE International*, Feb 2004, pp. 206–523.

[34] B. Black, D. Nelson, C. Webb, and N. Samra, "3D processing technology and its impact on IA32 microprocessors," in *IEEE International Conference on Computer Design: VLSI in Computers and Processors, 2004. ICCD 2004.*, Oct 2004, pp. 316–318.

[35] R. Patti, "Three-dimensional integrated circuits and the future of system-on-chip designs," *Proceedings of the IEEE*, vol. 94, no. 6, pp. 1214–1224, June 2006.

[36] S. Mittal and Z. Zhang, "EnCache: A Dynamic Profiling Based Reconfiguration Technique for Improving Cache Energy Efficiency," *Journal of Circuits, Systems, and Computers*, vol. 23, no. 10, 2014.

[37] S. Mittal and J. S. Vetter, "AYUSH: A Technique for Extending Lifetime of SRAM-NVM Hybrid Caches," *IEEE Computer Architecture Letters (CAL)*, 2014.

[38] S. Mittal, "A survey of power management techniques for phase change memory," *International Journal of Computer Aided Engineering and Technology (IJCAET)*, 2015.

[39] P. R. Panda, S. Roy, S. Chandrasekaran, N. Sharma, J. Kaur, S. K. Kandalam *et al.*, "High level energy modeling of controller logic in data caches," in *ACM Great Lakes symposium on VLSI (GLSVLSI)*, 2014, pp. 45–50.