Chapter 2
NVSim: A Circuit-Level Performance, Energy, and Area Model for Emerging Non-volatile Memory

Xiangyu Dong, Cong Xu, Norm Jouppi and Yuan Xie

Abstract Various new non-volatile memory (NVM) technologies have emerged recently. Among all the investigated new NVM candidate technologies, spin-torque transfer memory (STT-RAM, or MRAM), phase change memory (PCRAM), and resistive memory (ReRAM) are regarded as the most promising candidates. As the ultimate goal of this NVM research is to deploy them into multiple levels in the memory hierarchy, it is necessary to explore the wide NVM design space and find the proper implementation at different memory hierarchy levels from highly latency-optimized caches to highly density-optimized secondary storage. While abundant tools are available as SRAM/DRAM design assistants, similar tools for NVM designs are currently missing. Thus, in this work, we develop *NVSim*, a circuit-level model for NVM performance, energy, and area estimation, which supports various NVM technologies including STT-RAM, PCRAM, ReRAM, and legacy NAND flash. *NVSim* is successfully validated against industrial NVM prototypes, and it is expected to help boost architecture-level NVM-related studies.

X. Dong · C. Xu · Y. Xie (✉)
Computer Science and Engineering Department,
Pennsylvania State University, IST Building, University Park, PA 16802, USA
e-mail: xiangyud@qualcomm.com

C. Xu
e-mail: congxu@cse.psu.edu

Y. Xie
e-mail: yuanxie@cse.psu.edu

N. Jouppi
Google, Inc., CA, USA
e-mail: jouppi@acm.org

Y. Xie (ed.), *Emerging Memory Technologies*,
DOI: 10.1007/978-1-4419-9551-3_2,

2.1 Introduction

Universal memory that provides fast random access, high storage density, and non-volatility within one memory technology becomes possible, thanks to the emergence of various new non-volatile memory (NVM) technologies, such as spin-torque transfer random access memory (STT-RAM, or MRAM), phase change random access memory (PCRAM), and resistive random access memory (ReRAM). As the ultimate goal of this NVM research is to devise a universal memory that could work across multiple layers of the memory hierarchy, each of these emerging NVM technologies has to supply a wide design space that covers a spectrum from highly latency-optimized microprocessor caches to highly density-optimized secondary storage. Therefore, specialized peripheral circuitry is required for each optimization target. However, since few of these NVM technologies are mature so far, only a limited number of prototype chips have been demonstrated and just cover a small portion of the entire design space. In order to facilitate the architecture-level NVM research by estimating the NVM performance, energy, and area values under different design specifications before fabricating a real chip, in this work, we build *NVSim*, a circuit-level model for NVM performance, energy, and area estimations, which supports various NVM technologies including STT-RAM, PCRAM, ReRAM, and legacy NAND flash.

The main goals of developing NVSim tool are as follows:

- Estimate the access time, access energy, and silicon area of NVM chips with a given organization and specific design options before the effort of actual fabrications;
- Explore the NVM chip design space to find the optimized chip organization and design options that achieve best performance, energy, or area;
- Find the optimal NVM chip organization and design options that are optimized for one design metric while keeping other metrics under constraints.

We build NVSim by using the same empirical modeling methodology as CACTI [39, 43] but starting from a new framework and adding specific features for NVM technologies. Compared to CACTI, the framework of NVSim includes the following new features:

- It allows to move sense amplifiers from inner memory subarrays to the outer bank level and factor them out to achieve overall area efficiency of the memory module;
- It provides more flexible array organizations and data activation modes by considering any combinations of memory data allocation and address distribution;
- It models various types of data sensing schemes instead of voltage sensing scheme only;
- It allows memory banks to be formed in a bus-like manner rather than the H-tree manner only;
- It provides multiple design options of buffers instead of latency-optimized option that uses logical effort;
- It models the cross-point memory cells rather than MOS-accessed memory cells only;

- It considers the subarray size limit by analyzing the current sneak path;
- It allows advanced target users to redefine memory cell properties by providing a customization interface.

NVSim is validated against several industry prototype chips within the error range of 30 %. In addition, we show how to use this model to facilitate the architecture-level performance, energy, and area analysis for applications that adopt the emerging NVM technologies.

2.2 Background of Non-volatile Memory

In this section, we first review the technology background of four types of NVMs modeled in NVSim, which are STT-RAM, PCRAM, ReRAM, and legacy NAND flash.

2.2.1 NVM Physical Mechanisms and Write Operations

Different NVM technologies have their particular storage mechanisms and corresponding write methods.

2.2.1.1 NAND Flash

The physical mechanism of the flash memory is to store bits in the floating gate and control the gate threshold voltage. The series bit cell string of NAND flash, as shown in Fig. 2.1a, eliminates contacts between the cells and approaches the minimum cell

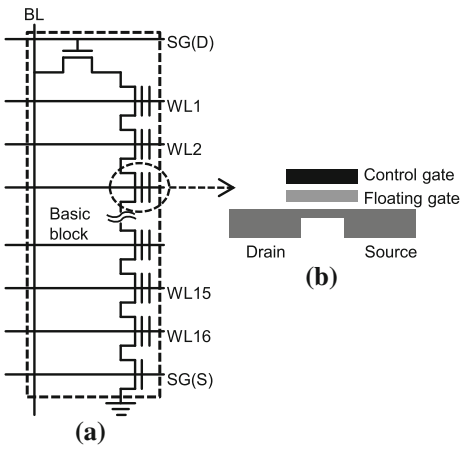

Fig. 2.1 The basic string block of NAND flash and the conceptual view of floating gate flash memory cell (*BL* bit line, *WL* word line, *SG* select gate)

size of $4F^2$ for low-cost manufacturing. The small cell size, low cost, and strong application demands make the NAND flash dominate the traditional non-volatile memory market. Figure 2.1b shows that a flash memory cell consists of a floating gate and a control gate aligned vertically. The flash memory cell modifies its threshold voltage V_T by adding electrons to or subtracting electrons from the isolated floating gate.

NAND flash usually charges or discharges the floating gate by using Fowler–Nordheim (FN) tunneling or hot-carrier injection (HCI). A program operation adds tunneling charges to the floating gate and the threshold voltage becomes negative, while an erase operation subtracts charges and the threshold voltage returns positive.

2.2.1.2 Spin-Torque Transfer RAM

Spin-torque transfer RAM (STT-RAM) uses magnetic tunnel junction (MTJ) as the memory storage and leverages the difference in magnetic directions to represent the memory bit. As shown in Fig. 2.2, MTJ contains two ferromagnetic layers. One ferromagnetic layer has fixed magnetization direction and it is called the reference layer, while the other layer has a free magnetization direction that can be changed by passing a write current and it is called the free layer. The relative magnetization direction of two ferromagnetic layers determines the resistance of MTJ. If two ferromagnetic layers have the same directions, the resistance of MTJ is low, indicating a "1" state; if two layers have different directions, the resistance of MTJ is high, indicating a "0" state.

As shown in Fig. 2.2, when writing "0" state into STT-RAM cells (RESET operation), positive voltage difference is established between SL and BL; when writing "1" state (SET operation), vice versa. The current amplitude required to reverse the direction of the free ferromagnetic layer is determined by the size and aspect ratio of MTJ and the write pulse duration.

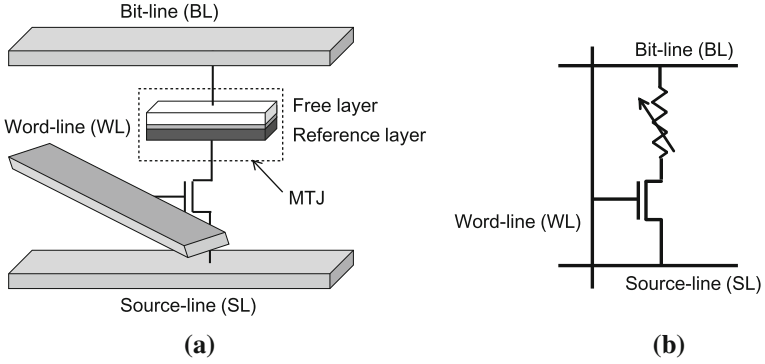

Fig. 2.2 Demonstration of a MRAM cell: **a** structural view; **b** schematic view (*BL* bit line, *WL* word line, *SL* source line)

2.2.1.3 Phase Change RAM

Phase change RAM (PCRAM) uses chalcogenide material (e.g., GST) to store information. The chalcogenide materials can be switched between a crystalline phase (SET state) and an amorphous phase (RESET state) with the application of heat. The crystalline phase shows low resistivity, while the amorphous phase is characterized by high resistivity. Figure 2.3 shows an example of a MOS-accessed PCRAM cell.

The SET operation crystallizes GST by heating it above its crystallization temperature, and the RESET operation melt-quenches GST to make the material amorphous as illustrated in Fig. 2.4. The temperature is controlled by passing a specific electrical current profile and generating the required Joule heat. High-power pulses are required for the RESET operation to heat the memory cell above the GST melting temperature. In contrast, moderate power but longer duration pulses are required for the SET operation to heat the cell above the GST crystallization temperature but below the melting temperature [33].

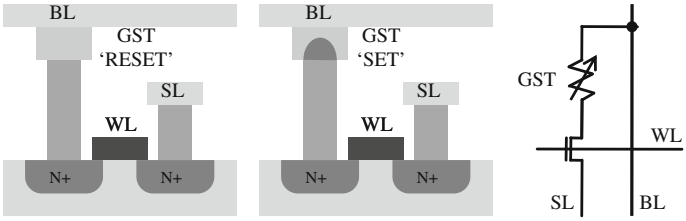

Fig. 2.3 The schematic view of a PCRAM cell with NMOS access transistor (*BL* bit line, *WL* word line, *SL* source line)

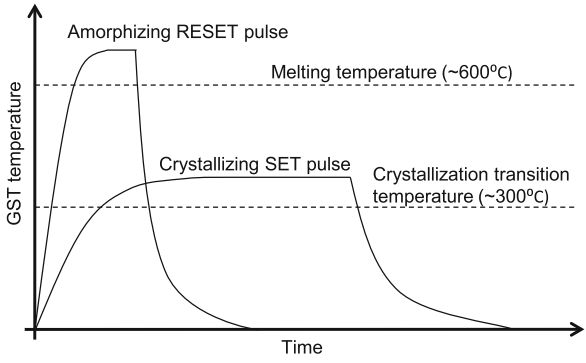

Fig. 2.4 The temperature–time relationship during SET and RESET operations

2.2.1.4 Resistive RAM

Although many non-volatile memory technologies (e.g., aforementioned STT-RAM and PCRAM) are based on electrically induced resistive switching effects, we define resistive RAM (ReRAM) as the one that involves electro- and thermochemical effects in the resistance change of a metal/oxide/metal system. In addition, we confine our definition to bipolar ReRAM. Figure 2.5 illustrates the general concept for the ReRAM working mechanism. An ReRAM cell consists of a metal oxide layer (e.g., Ti [45], Ta [42], and Hf [4]) sandwiched by two metal (e.g., Pt [45]) electrodes. The electronic behavior of metal/oxide interfaces depends on the oxygen vacancy concentration of the metal oxide layer. Typically, the metal/oxide interface shows Ohmic behavior in the case of very high doping and rectifying in the case of low doping [45]. In Fig. 2.5, the TiO_x region is semi-insulating indicating lower oxygen vacancy concentration, while the TiO_{2-x} is conductive indicating higher concentration.

The oxygen vacancy in metal oxide is n-type dopant, whose draft under the electric field can cause the change in doping profiles. Thus, applying electronic current can modulate the IV curve of the ReRAM cell and further switch the cell from one state to the other state. Usually, for bipolar ReRAM, the cell can be switched ON (SET operation) only by applying a negative bias and OFF (RESET operation) only by applying the opposite bias [45]. Several ReRAM prototypes [5, 22, 35] have been demonstrated and show promising properties on fast switching speed and low energy consumption.

2.2.2 Read Operations

The read operations of these NVM technologies are almost the same. Since the NVM memory cell has different resistance in ON and OFF states, the read operation can be accomplished either by applying a small voltage on the bit line and sensing the

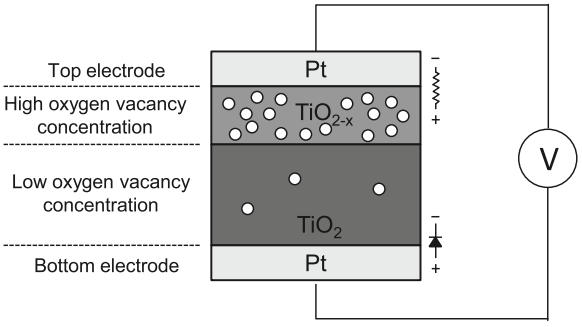

Fig. 2.5 The working mechanism of ReRAM cells

current that passes through the memory cell or by injecting a small current into the bit line and sensing the voltage across the memory cell. Instead of SRAM that generates complement read signals from each cell, NVM usually has a group of dummy cells to generate the reference current or reference voltage. The generated current (or voltage) from the to-be-read cell is then compared to the reference current (or voltage) by using sense amplifiers. Various types of sense amplifiers are modeled in NVSim as we discuss in Sect. 2.5.2.

2.2.3 Write Endurance Issue

Write endurance is the number of times that an NVM cell can be overwritten. Among all the NVM technologies modeled in NVSim, only STT-RAM does not suffer from the write endurance issue. NAND flash, PCRAM, and ReRAM all have limited write endurance, which is the number of times that a memory cell can be overwritten. NAND flash only has write endurance of 10^5–10^6. The PCRAM endurance is now in the range between 10^5 and 10^9 [1, 21, 32]. ReRAM research currently shows endurance numbers in the range between 10^5 and 10^{10} [20, 24]. A projected plan by ITRS for 2024 for emerging NVM, i.e., PCRAM and ReRAM, highlights endurance in the order of 10^{15} or more write cycles [14] . In NVSim, the write endurance limit is not modeled since NVSim is a circuit-level modeling tool.

2.2.4 Retention Time Issue

Retention time is the time that data can be retained in NVM cells. Typically, NVM technologies require retention time of higher than 10 years. However, in some cases, such a high retention time is not necessary. For example, Smullen et al. [36] relaxed the retention time requirement to improve the timing and energy profile of STT-RAMs. Since the trade-off between NVM retention time and other NVM parameters (e.g., the duration and amplitude of write pulses) is on the device level, as a circuit-level tool, NVSim does not model this trade-off directly but instead takes different sets of NVM parameters with various retention time as the device-level input.

2.2.5 MOS-Accessed Structure Versus Cross-Point Structure

Some NVM technologies (for example, PCRAM [18] and ReRAM [3, 18, 20]) have the capability of building cross-point memory arrays without access devices. Conventionally, in the MOS-accessed structure, memory cell arrays are isolated by MOS access devices and the cell size is dominated by the large MOS access device that is necessary to drive enough write current, even though the NVM cell itself is

much smaller. However, taking advantage of the cell nonlinearity, an NVM array can be accessed without any extra access devices. The removal of MOS access devices leads to a memory cell size of only $4F^2$, where F is the process feature size. Unfortunately, the cross-point structure also brings extra peripheral circuitry design challenges and a trade-off between performance, energy, and area is always necessary as discussed in our previous work [44]. NVSim models both the MOS-accessed and the cross-point structures, and the modeling methodology is described in the following sections.

2.3 NVSim Framework

The framework of NVSim is modified from CACTI [38, 39]. We add several new features, such as more flexible data activation modes and alternative bank organizations.

2.3.1 Device Model

NVSim uses device data from ITRS report [14] and the MASTAR tool [15] to obtain the process parameters. NVSim covers the process nodes from 180, 120, 90, 65, 45, 32, to 22 nm and supports 3 transistor types, which are high performance (HP), low operating power (LOP), and low standby power (LSTP).

2.3.2 Array Organization

Figure 2.6 shows the array organization. There are 3 hierarchy levels in such organization, which are *bank*, *mat*, and *subarray*. Basically, the descriptions of these levels are as follows:

- *Bank* is the top-level structure modeled in NVSim. One non-volatile memory chip can have multiple banks. The bank is a fully functional memory unit, and it can be operated independently. In each bank, multiple mats are connected together in either H-tree or bus-like manner
- *Mat* is the building block of bank. Multiple mats in a bank operate simultaneously to fulfill a memory operation. Each mat consists of multiple subarrays and one predecoder block
- *Subarray* is the elementary structure modeled in NVSim. Every subarray contains peripheral circuitry including row decoders, column multiplexers, and output drivers.

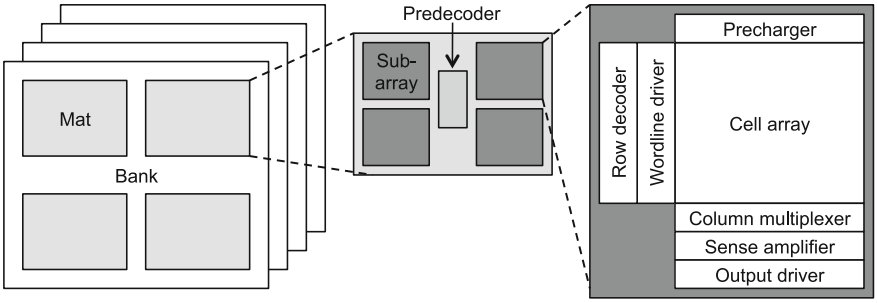

Fig. 2.6 The memory array organization modeled in NVSim: A hierarchical memory organization includes banks, mats, and subarrays with decoders, multiplexers, sense amplifiers, and output drivers

Conventionally, sense amplifiers are integrated on the subarray level as modeled in CACTI [38, 39]. However, in NVSim model, sense amplifiers can be placed either on the subarray level or on the mat level.

2.3.3 Memory Bank Type

For practical memory designs, memory cells are grouped together to form memory modules of different types. For instance,

- The main memory is a typical random access memory (RAM), which takes the address of data as input and returns the content of data;
- The set-associative cache contains two separate RAMs (data array and tag array) and can return the data if there is a cache hit by the given set address and tag;
- The fully associative cache usually contains a content-addressable memory (CAM).

To cover all the possible memory designs, we model 5 types of memory banks in NVSim: one for RAM, one for CAM, and three for set-associate caches with different access manners. The functionalities of these 5 types of memory banks are listed as follows:

1. *RAM*: Output the data content at the I/O interface given the data address
2. *CAM*: Output the data address at the I/O interface given the data content if there is a hit
3. *Cache with normal access*: Start to access the cache data array and tag array at the same time; the data content is temporarily buffered in each mat; if there is a hit, the cache hit signal generated from the tag array is routed to the proper mats, and the content of the desired cache line is output to the I/O interface
4. *Cache with sequential access*: Access the cache tag array at first; if there is a hit, then access the cache data array with the set address and the tag hit information and finally output the desired cache line to the I/O interface

5. *Cache with fast access*: Access the cache data array and tag array simultaneously; read the entire set content from the mats to the I/O interface; selectively output the desired cache line if there is a cache hit signal generated from the tag array.

2.3.4 Activation Mode

We model the array organization and the data activation modes using eight parameters, which are

- N_{MR}: number of rows of mat arrays in each bank;
- N_{MC}: number of columns of mat arrays in each bank;
- N_{AMR}: number of active rows of mat arrays during data accessing;
- N_{AMC}: number of active columns of mat arrays during data accessing;
- N_{SR}: number of rows of subarrays in each mat;
- N_{SC}: number of columns of subarrays in each mat;
- N_{ASR}: number of active rows of subarrays during data accessing;
- and N_{ASC}: number of active columns of subarrays during data accessing.

The values of these parameters are all constrained to be power of two. N_{MR} and N_{MC} define the number of mats in a bank, and N_{SR} and N_{SC} define the number of subarrays in a mat. N_{AMR}, N_{AMC}, N_{ASR}, and N_{ASC} define the activation patterns, and they can take any values smaller than N_{MR}, N_{MC}, N_{SR}, and N_{SC}, respectively. On the contrary, the limitation of array organization and data activation pattern in CACTI is caused by several constraints on these parameters such as $N_{AMR} = 1$, $N_{AMC} = N_{MC}$, and $N_{SR} = N_{SC} = N_{ASR} = N_{ASC} = 2$.

NVSim has these flexible activation patterns and is able to model sophisticated memory accessing techniques, such as single subarray activation [41].

2.3.5 Routing to Mats

In order to first route the data and address signals from the I/O port to the edge of memory mats and from mat to the edges of memory subarrays, we divided all the interconnect wires into three categories: *Address Wires*, *Broadcast Data Wires*, and *Distributed Data Wires*. Depending on the memory module types and the activation modes, the initial number of wires in each group is assigned according to the rules listed in Table 2.1. We use the terminology block to refer to the memory words in RAM and CAM designs and the cache lines in cache designs. In Table 2.1, N_{block} is the number of blocks, W_{block} is the block size, and A is the associativity in cache designs. The number of *Broadcast Data Wires* is always kept unchanged, the number of *Distributed Data Wires* is cut by half at each routing point where data are merged, and the number of *Address Wires* is subtracted by one at each routing point where data are multiplexed.

Table 2.1 The initial number of wires in each routing group

			N_{AW}	N_{BW}	N_{DW}
Random access memory (RAM)			$\log_2 N_{block}$	0	W_{block}
Content-addressable memory (CAM)				W_{block}	0
Cache	Normal access	Data array	$\log_2 (N_{block}/A)$	$\log_2 A$	W_{block}
		Tag array	$\log_2 (N_{block}/A)$	W_{block}	A
	Sequential access	Data array	$\log_2 N_{block}$	0	W_{block}
		Tag array	$\log_2 (N_{block}/A)$	W_{block}	A
	Fast access	Data array	$\log_2 (N_{block}/A)$	0	$W_{block} A$
		Tag array	$\log_2 (N_{block}/A)$	W_{block}	A

N_{AW} Number of address wires
N_{BW} Number of broadcast data wires
N_{DW} Number of distributed data wires

We use the case of the cache bank with normal access to demonstrate how the wires are routed from the I/O port to the edges of the mats. For simplicity, we suppose the data array and the tag array are two separate modules. While the data and tag arrays usually have different mat organizations in practice, we use the same 4×4 mat organization for the demonstration purpose as shown in Figs. 2.7 and 2.8. The total 16 mats are positioned in a 4×4 formation and connected by a 4-level H-tree. Therefore, N_{MR} and N_{MC} are 4. As an example, we use the activation mode in which two rows and two columns of the mat array are activated for each data access, and the activation groups are Mat $\{0, 2, 8, 10\}$, Mat $\{1, 3, 9, 11\}$, Mat $\{4, 6, 12, 14\}$, and Mat $\{5, 7, 13, 15\}$. Thereby, N_{AMR} and N_{AMC} are 2. In addition, we set the cache line size (block size) to 64 B, the cache associativity to $A = 8$, and the cache bank capacity to 1 MB, so that the number of cache lines (blocks) is $N_{block} = 8M/512 = 16{,}384$, the block size in the data array is $W_{block,data} = 512$, and the block size in the tag array is $W_{block,tag} = 16$ (assuming 32-bit addressing and labeling dirty block with one bit).

According to Table 2.1, the initial number of address wires (N_{AW}) is $\log_2 N_{block}/A = 11$ for both data and tag arrays. For data array, the initial number of broadcast data wires ($N_{BW,data}$) is $\log_2 A = 3$, which is used to transit the tag hit signals from the tag array to the corresponding mats in the data array; the initial number of distributed data wires ($N_{DW,data}$) is $W_{block,data} = 512$, which is used to output the desired cache line from the mats to the I/O port. For tag array, the broadcast data wire ($N_{BW,tag}$) is $W_{block,tag} = 16$, which is sent from the I/O port to each of the mat in the tag array; the initial number of distributed data wires ($N_{DW,tag}$) is $A = 8$, which is used to collect the tag hit signals from each mat to the I/O port and then send to the data array after a 8-to-3 encoding process.

From the I/O port to the edges of the mats, the numbers of wires in the three categories are changed as follows and demonstrated in Figs. 2.7 and 2.8, respectively.

1. At node A, the activated mats are distributed in both the upper and the bottom parts, so node A is a *merging* node. As per the routing rule, the address wires and broadcast data wires remain the same, but the distributed data wires are

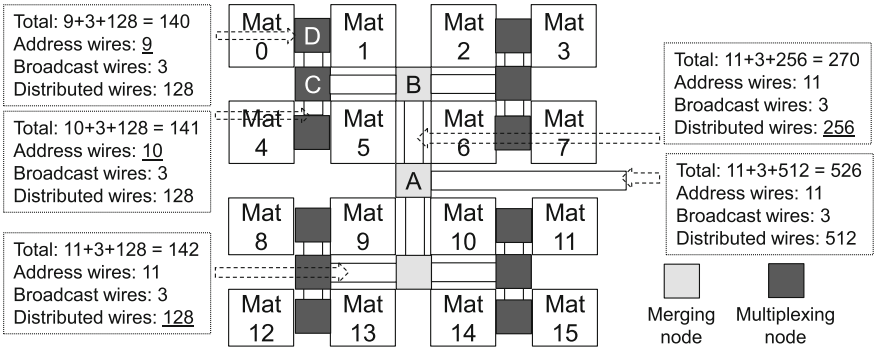

Fig. 2.7 The example of the wire routing in a 4 × 4 mat organization for the *data array* of a 8-way 1 MB cache with 64 B cache lines

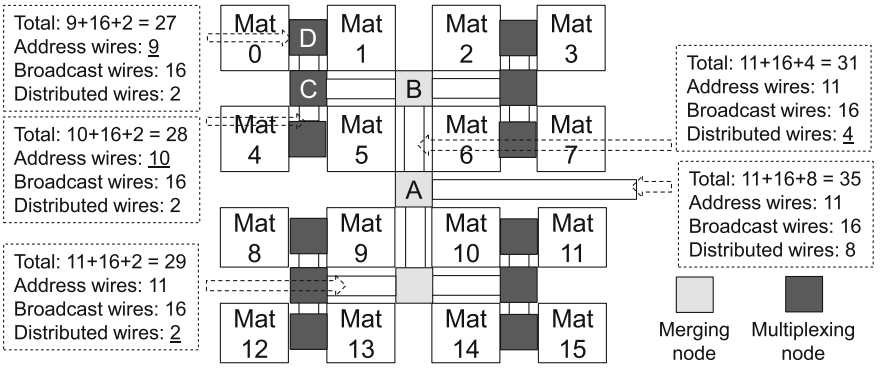

Fig. 2.8 The example of the wire routing in a 4 × 4 mat organization for the *tag array* of a 8-way 1 MB cache with 64 B cache lines

cut into half. Thus, the wire segment between node A and B have $N_{\mathrm{AW}} = 11$, $N_{\mathrm{BW,data}} = 3$, $N_{\mathrm{DW,data}} = 256$, $N_{\mathrm{BW,tag}} = 16$, and $N_{\mathrm{DW,tag}} = 4$

2. Node B is again a *merging* node. Thus, the wire segment between node B and C have $N_{\mathrm{AW}} = 11$, $N_{\mathrm{BW,data}} = 3$, $N_{\mathrm{DW,data}} = 128$, $N_{\mathrm{BW,tag}} = 16$, and $N_{\mathrm{DW,tag}} = 2$
3. At node C, the activated mats are allocated only in one side, either from Mat 0/1 or from Mat 4/5, so Node C is a *multiplexing* node. As per the routing rule, the distributed data wires and broadcast data wires remain the same, but the address wires are decremented by 1. Thus, the wire segment between node C and node D have $N_{\mathrm{AW}} = 10$, $N_{\mathrm{BW,data}} = 3$, $N_{\mathrm{DW,data}} = 128$, $N_{\mathrm{BW,tag}} = 16$, and $N_{\mathrm{DW,tag}} = 2$
4. Finally, node D is another *multiplexing* node. Thus, the wire segments at the mat edges have $N_{\mathrm{AW}} = 9$, $N_{\mathrm{BW,data}} = 3$, $N_{\mathrm{DW,data}} = 128$, $N_{\mathrm{BW,tag}} = 16$, and $N_{\mathrm{DW,tag}} = 2$.

Thereby, each mat in the data array takes the input of a 9-bit set address and a 3-bit tag hit signals (which can be treated as the block address in a 8-way associative set),

and it generates the output of a 128-bit data. A group of 4 data mats provide the desired output of a 512-bit (64 B) cache line, and four such groups cover the entire 11-bit set address space. On the other hand, each mat in the tag array takes the input of a 9-bit set address and a 16-bit tag, and it generates a 2-bit hit signals (01 or 10 for hit and 00 for miss). A group of 4 tag mats concatenate their hit signals and provide the information whether a 16-bit tag hits in a 8-way associated cache with a 9-bit address space, and four such groups extend the address space from 9 bit to the desired 11 bit.

Other configurations in Table 2.1 can be explained in the similar manner.

2.3.6 Routing to Subarrays

The interconnect wires from mat to the edges of memory subarrays are routed using the same H-tree organization as shown in Fig. 2.9, and its routing strategy is the same wire partitioning rule described in Sect. 2.3.5. However, NVSim provides an option of building mat using a bus-like routing organization as illustrated in Fig. 2.10. The wire partitioning rule described in Sect. 2.3.5 can also be applied to the bus-like organization with a few extensions. For example, a *multiplexing* node with a fanout of N decrements the number of address wires by $\log_2 N$ instead of 1; a *merging* node with a fanout of N divides the number of distributed data wires by N instead of 2.

Furthermore, the default setting of including sense amplifiers in each subarray can cause a dominant portion of the total array area. As a result, for high-density memory

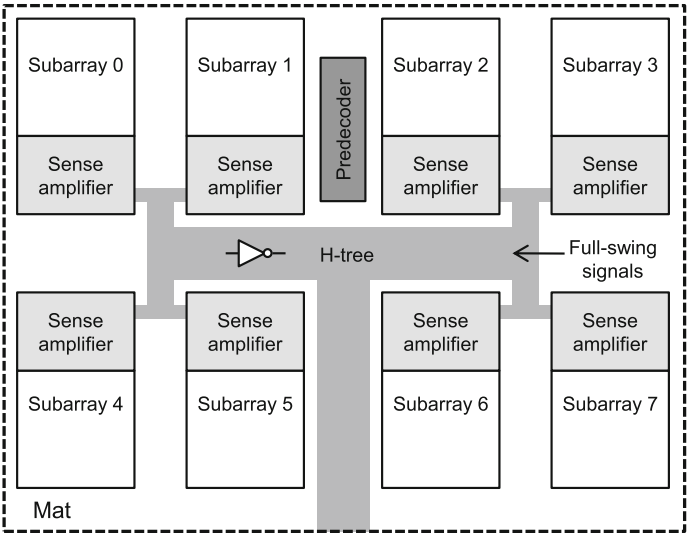

Fig. 2.9 An example of mat using internal sensing and H-tree routing

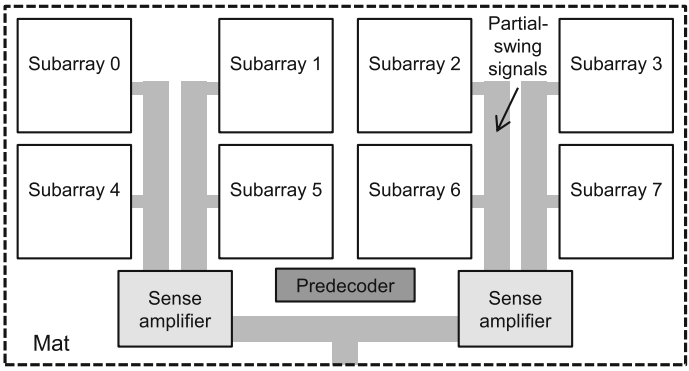

Fig. 2.10 An example of mat using external sensing and bus-like routing

module designs, NVSim provides an option of moving the sense amplifiers out of the subarray and using external sensing. In addition, a bus-like routing organization is designed to associate with the external sensing scheme.

Figure 2.9 shows a common mat using H-tree organization to connect all the sense amplifier-included subarrays together. In contrast, the new external sensing scheme is illustrated in Fig. 2.10. In this external sensing scheme, all the sense amplifiers are located at the mat level and the output signals from each sense amplifier-free subarray are partial swing. It is obvious that the external sensing scheme has much higher area efficiency compared to its internal sensing counterpart. However, as a penalty, sophisticated global interconnect technologies, such as repeater inserting, cannot be used in the external sensing scheme since all the global signals are partial swing before passing through the sense amplifiers.

2.3.7 Subarray Size Limit

The subarray size is a critical parameter to design a memory module. Basically, smaller subarrays are preferred for latency-optimized designs since they reduce local bit line and word line latencies and leave the global interconnects to be handled by the sophisticated H-tree solution. In contrast, larger subarrays are preferred for area-optimized designs since they can greatly amortize the peripheral circuitry area. However, the subarray size has its upper limit in practice.

For MOS-accessed subarray, the leakage current paths from unselected word lines are the main constraint to the bit line length. For cross-point subarray, the leakage current path issue is much more severe as there is no MOSFET in such subarray that can isolate selected and unselected cells [23]. The half-select cells in cross-point subarrays serve as current dividers in the selected row and columns, preventing the array size from growing unbounded since the available driving current is limited.

The minimum current that a column write driver should provide is determined by

$$I_{\text{driver}} = I_{\text{write}} + (N_r - 1) \times I(V_{\text{write}}/2) \tag{2.1}$$

where I_{write} and V_{write} are the current and voltage of either RESET or SET operation. Nonlinearity of memory cells is reflected by the fact that the current through cross-point memory cells is not directly proportional to the voltage applied on it, which means non-constant resistance of the memory cell. In NVSim, we define a nonlinearity coefficient, K_r, to quantify the current divider effect of the half-selected memory cells as follows:

$$K_r = \frac{R(V_{\text{write}}/2)}{R(V_{\text{write}})} \tag{2.2}$$

where $R(V_{\text{write}}/2)$ and $R(V_{\text{write}})$ are equivalent static resistance of cross-point memory cells biased at $V_{\text{write}}/2$ and V_{write}, respectively. Then, we derive the upper limit in a cross-point subarray size by

$$N_r = \left(\frac{I_{\text{driver}}}{I_{\text{write}}} - 1\right) \times 2 \times K_r + 1 \tag{2.3}$$

$$N_c = \left(\frac{I_{\text{driver}}}{I_{\text{write}}} - N_{\text{sc}}\right) \times 2 \times K_r + N_{\text{sc}} \tag{2.4}$$

where I_{driver} is the maximum driving current that the write driver attached to the selected row/column can provide and N_{sc} is the number of selected columns per row. Thus, N_r and N_c are the maximum numbers of rows and columns in a cross-point subarray.

As shown in Fig. 2.11, the maximum cross-point subarray size increases with larger current driving capability or larger nonlinearity coefficient.

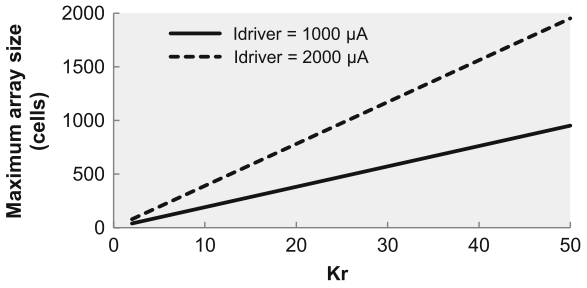

Fig. 2.11 Maximum subarray size versus nonlinearity and driving current

2.3.8 Two-Step Write in Cross-Point Subarrays

In cross-point structure, SET and RESET operations cannot be performed simultaneously. Thus, two steps of write operations are required in the cross-point structure when multiple cells are selected in a row.

In NVSim, we model two write methods for cross-point subarrays. The first one separates SET and RESET operations as Fig. 2.12 shows, and it is called SET-before-RESET. The second one erases all the cells in the selected row before the selective RESET operation as Fig. 2.13 shows, and it is called ERASE-before-RESET (EbR). Supposing the 4-bit word to write is "0101," we first write "x1x1" ("x" here means bias row and column of the corresponding cells at the same voltage to keep their original states) and then write "0x0x" in SET-before-RESET (SbR) method, or we first SET all the four cells and then write "0x0x" in ERASE-before-RESET method. The first method has smaller write latency since the erase operation can be performed before the arrival of the column selector signal, but it needs more write energy due to the redundant SET on the cells that are RESET back in the second step. Here, ERASE-before-RESET is chosen rather than ERASE-before-SET because SET operation usually consumes less energy than RESET operation does.

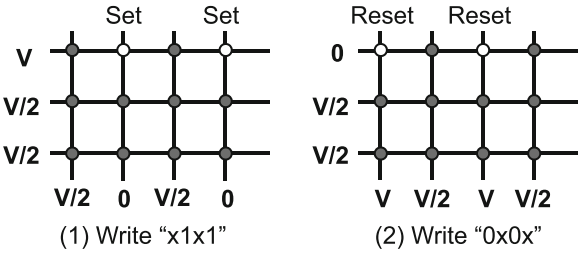

Fig. 2.12 Sequential write method: SET-before-RESET

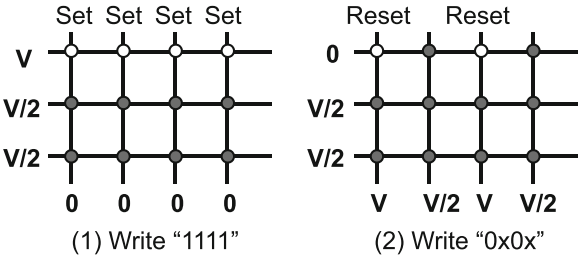

Fig. 2.13 Sequential write method: ERASE-before-RESET

2.4 Area Model

Since NVSim estimates the performance, energy, and area of non-volatile memory modules, the area model is an essential component of NVSim, especially given the facts that interconnect wires contribute a large portion of total access latency and access energy and the geometry of the module becomes highly important. In this section, we describe the NVSim area model from the memory cell level to the bank level in details.

2.4.1 Cell Area Estimation

Three types of memory cells are modeled in NVSim: MOS-accessed, cross-point, and NAND string.

2.4.1.1 MOS-Accessed Cell

MOS-accessed cell corresponds to the typical 1T1R (1-transistor-1-resistor) structure used by many NVM chips [1, 11, 13, 17, 19, 30, 40], in which an NMOS access device is connected in series with the non-volatile storage element (i.e., MTJ in STT-RAM, GST in PCRAM, and metal oxide in ReRAM) as shown in Fig. 2.14. Such an NMOS device turns on/off the access path to the storage element by tuning the voltage applied to its gate. The MOS-accessed cell usually has the best isolation among neighboring cells due to the property of MOSFET.

In MOS-accessed cells, the size of NMOS is bounded by the current needed by the write operation. The size of NMOS in each MOS-accessed cell needs to be sufficiently large so that the NMOS has the capability of driving enough write current.

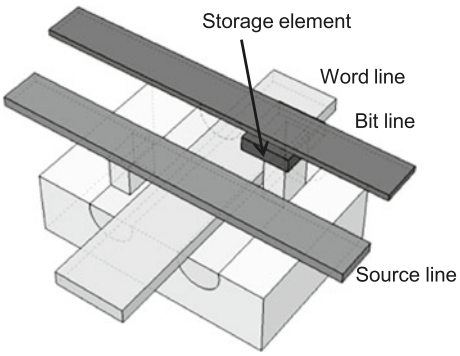

Fig. 2.14 Conceptual view of a MOS-accessed cell (1T1R) and its connected word line, bit line, and source line

The driving current of NMOS, I_{DS}, can be first-order estimated as follows[1],

$$I_{DS} = K\frac{W}{L}\left[\left(V_{GS} - V_{TH}\right)V_{DS} - \frac{V_{DS}^2}{2}\right] \tag{2.5}$$

if NMOS is working at the linear region or calculated by

$$I_{DS} = \frac{K}{2}\frac{W}{L}\left(V_{GS} - V_{TH}\right)^2\left(1 + \lambda V_{DS}\right) \tag{2.6}$$

if NMOS is working at the saturation region. Hence, no matter in which region NMOS is working, the current driving ability of NMOS is proportional to its width-to-length (W/L) ratio,[2] which determines the NMOS size. To achieve high cell density, we model the MOS-accessed cell area by referring to DRAM design rules [9]. As a result, the cell size of a MOS-accessed cell in NVSim is calculated as follows:

$$\text{Area}_{\text{cell,MOS-accessed}} = 3\left(W/L + 1\right)\left(F^2\right) \tag{2.7}$$

in which the width-to-length ratio (W/L) is determined by Eq. 2.5 or 2.6 and the required write current is configured as one of the input values of NVSim. In NVSim, we also allow advanced users to override this cell size calculation by directly importing the user-defined cell size.

2.4.1.2 Cross-Point Cell

Cross-point cell corresponds to the 1D1R (1-diode-1-resistor) [21, 22, 31, 46, 47] or 0T1R (0-transistor-1-resistor) [3, 18, 20] structures used by several high-density NVM chips recently. Figure 2.15 shows a cross-point array without diodes (i.e., 0T1R structure). For 1D1R structure, a diode is inserted between the word line and the storage element. Such cells either rely on the one-way connectivity of diode (i.e., 1D1R) or leverage materials' nonlinearity (i.e., 0T1R) to control the memory access path. As illustrated in Fig. 2.15, the widths of word lines and bit lines can be the minimal value of 1F and the spacing in each direction is also 1F; thus, the cell size of each cross-point cell is

$$\text{Area}_{\text{cell,cross-point}} = 4\left(F^2\right) \tag{2.8}$$

[1] Equations 2.5 and 2.6 are for long-channel drift/diffusion devices, and the equations are subjected to change depending on the technology, though the proportional relationship between the current and W/L still holds for very advanced technologies.

[2] Usually, the transitor length (L) is fixed as the minimal feature size, and the transistor width (W) is adjustable.

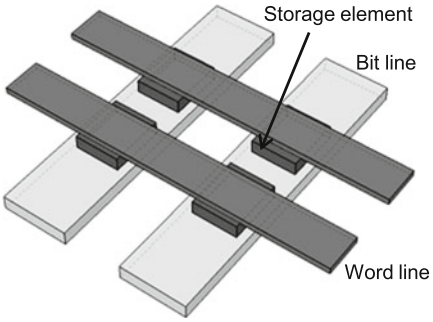

Fig. 2.15 Conceptual view of a cross-point cell array without diode (0T1R) and its connected word lines and bit lines

Compared to MOS-accessed cells, cross-point cells have worse cell isolation but provide a way of building high-density memory chip because they have much smaller cell sizes. In some cases, the cross-point cell size is constrained by the diode due to limited current density, and NVSim allows the user to override the default $4F^2$ setting.

2.4.1.3 NAND String Cell

NAND string cells are particularly modeled for NAND flash. In NAND string cells, a group of floating gates are connected in series and two ordinary gates with contacts are added at the string ends as shown in Fig. 2.16. Since the area of the floating gates can be minimized to 2 × 2F, the total area of a NAND string cell is

$$\text{Area}_{\text{cell,NANDstring}} = 2\,(2N + 5)\,(F^2) \tag{2.9}$$

where N is the number of floating gates in a string and we assume that the addition of two gates and two contacts causes 5F in the total string length.

2.4.2 Peripheral Circuitry Area Estimation

Besides the area occupied by memory cells, there is a large portion of memory chip area that is contributed to the peripheral circuitry. In NVSim, we have peripheral circuitry components such as row decoders, prechargers, and column multiplexers on the subarray level, predecoders on the mat level, and sense amplifiers and write drivers on either the subarray level or mat level, depending on whether internal or external data sensing scheme is used. In addition, on every level, interconnect wires might occupy extra silicon area if the wires are relayed using repeaters.

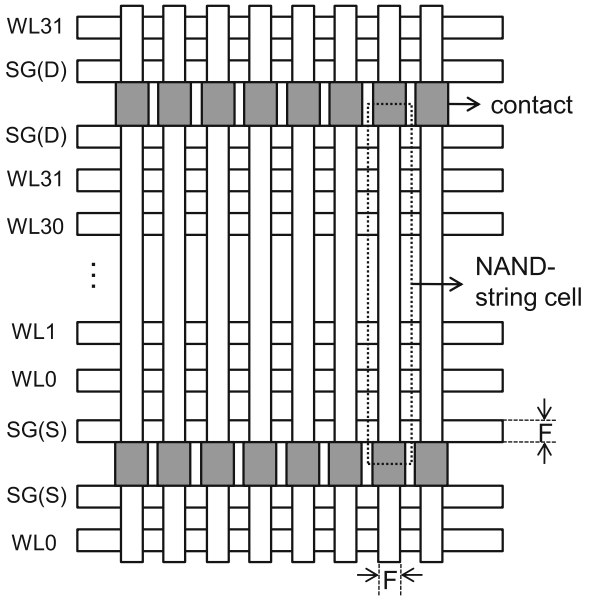

Fig. 2.16 The layout of the NAND string cell modeled in NVSim

In order to estimate the area of each peripheral circuitry component, we delve into the actual gate-level logic design as similar to CACTI [39]. However, in NVSim, we size transistors in a more generalized way than CACTI does.

The sizing philosophy of CACTI is to use logical effort [37] to size the circuits for minimum delay. NVSim's goal is to estimate the properties of NVM chips of a broad range, and these chips might be optimized for density or energy consumption instead of minimum delay; thus, we provide optional sizing methods rather than only applying logical effort. In addition, for some peripheral circuitry in NVM chips, the size of some transistors is determined by their required driving current instead of their capacitive load, and this violates the basic rules of using logical effort.

Therefore, we offer three transistor sizing choices in the area model of NVSim: one optimizing latency, one optimizing area, while another balancing latency and area. An example is illustrated in Fig. 2.17, demonstrating the different sizing methods when an output buffer with 4,096 times the capacitance of a minimum-sized inverter is to be designed. In a latency-optimized buffer design, the number of stages and all of the inverter sizing in the inverter chain are calculated by logical effort to achieve minimum delay (30 units) while paying a huge area penalty (1,365 units). In an area-optimized buffer design, there are only two stages of inverters, and the size of the last stage is determined by the minimum driving current requirement. This type of buffer has the minimum area (65 units), but is much slower than the latency-optimized buffer. The balanced option determines the size of last-stage inverter by its driving current requirement and calculates the size of the other inverters by logical effort. This results in a balanced delay and area metric.

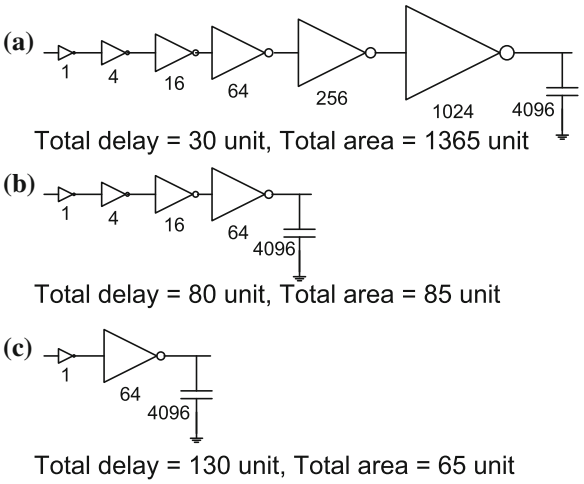

Fig. 2.17 Buffer designs with different transistor sizing: **a** latency-optimized; **b** balanced; **c** area-optimized

2.5 Timing and Power Models

As an analytical modeling tool, *NVSim* uses RC analysis for timing and power. In this section, we describe how resistances and capacitances are estimated in NVSim and how they are combined to calculate the delay and power consumption.

2.5.1 Generic Timing and Power Estimation

In NVSim, we consider the wire resistance and wire capacitance from interconnects, turn-on resistance, switching resistance, gate and drain capacitances from transistors, and equivalent resistance and capacitance from memory storage elements (e.g., MTJ in STT-RAM and GST in PCRAM). The methods of estimating wire and parasitic resistances and capacitances are modified from the previous versions of CACTI [39, 43] by several enhancements. The enhancements include updating the transistor models by latest ITRS report [14], considering the thermal impact on wire resistance calculation, adding drain-to-channel capacitance in the drain capacitance calculation, and so on. We build a lookup table to model the equivalent resistance and capacitance of memory storage elements since they are the properties of certain non-volatile memory technology. Considering NVSim is a system-level estimation tool, we only model the static behavior of the storage elements and record the equivalent

resistances and capacitances of RESET and SET states (i.e., R_{RESET}, R_{SET}, C_{RESET}, C_{SET}).[3]

After calculating the resistances and capacitances of nodes, the delay of each logic component is calculated by using a simplified version of Horowitz's timing model [12] as follows:

$$\text{Delay} = \tau \sqrt{\left(\ln \frac{1}{2}\right)^2 + \alpha\beta} \tag{2.10}$$

where α is the slope of the input, $\beta = 1/g_m R$ is the normalized input transconductance by the output resistance, and $\tau = \text{RC}$ is the RC time constant.

The dynamic energy and leakage power consumptions can be modeled as

$$\text{Energy}_{\text{dynamic}} = C V_{\text{DD}}^2 \tag{2.11}$$

$$\text{Power}_{\text{leakage}} = V_{\text{DD}} I_{\text{leak}} \tag{2.12}$$

where we model both gate leakage and subthreshold leakage currents in I_{leak}.

The overall memory access latency and energy consumption are estimated by combining all the timing and power values of circuit components together. NVSim follows the same methodology that CACTI [39] uses with minor modifications.

2.5.2 Data Sensing Models

Unlike other peripheral circuitries, the sense amplifier is an analog design instead of a logic design. Thus, in NVSim, we develop a separate timing model for the data sensing schemes. Different sensing schemes have their impacts on the trade-off between performance, energy, and area. In NVSim, we consider three types of sensing schemes: current sensing (CS), current-in voltage sensing (CIVS), and voltage divider sensing (VDS).

In the current sensing scheme as shown in Fig. 2.18, the state of memory cell (STT-RAM, PCRAM, or ReRAM) is read out by measuring the resulting current through the selected memory cell when a read voltage is applied: The current on the bit line is compared to the reference current generated by reference cells, the current difference is amplified by current-mode sense amplifiers, and they are eventually converted to voltage signals.

Figure 2.19 demonstrates an alternative sensing method by applying a current source on the selected memory cell and sensing the voltage via the voltage-mode sense amplifier.

The voltage divider sensing scheme is presented by introducing a resistor (R_x) in series with the memory cell as illustrated in Fig. 2.20. The resistance value is selected to achieve the maximum read sensing margin, and it is calculated as follows:

[3] One of the exceptions is that NVSim records the detailed IV curves for cross-point ReRAM cells without diode because we need to leverage the nonlinearity of the storage element.

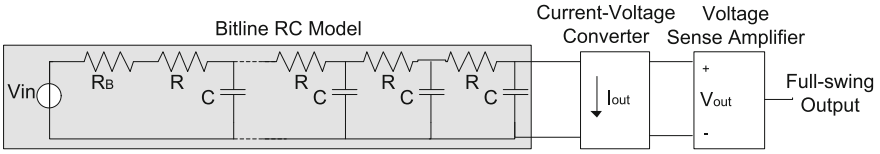

Fig. 2.18 Analysis model for current sensing scheme

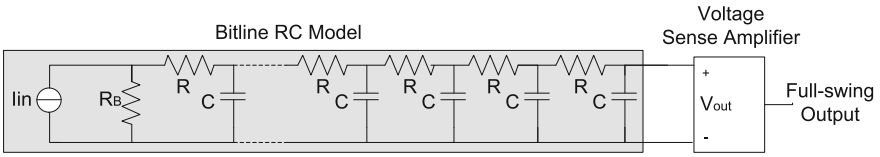

Fig. 2.19 Analysis model for current-in voltage sensing scheme

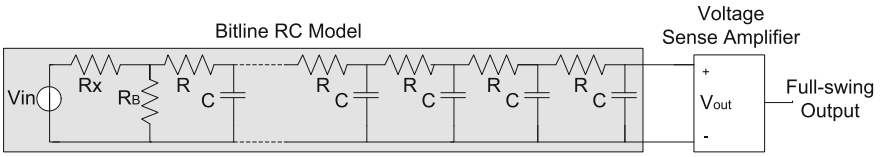

Fig. 2.20 Analysis model for voltage divider sensing scheme

$$R_x = \sqrt{R_{\text{on}} \times R_{\text{off}}} \tag{2.13}$$

where R_{on} and R_{off} are the equivalent resistance values of the memory cell in LRS and HRS, respectively.

2.5.2.1 Bitline RC Model

We model the bit line RC delay analytically for each sensing scheme. The most significant difference between the current-mode sensing and voltage-mode sensing is that the input resistance of ideal current-mode sensing is zero, while that of ideal voltage-mode sensing is infinite. And, the most significant difference between current-in voltage sensing and voltage divider sensing is that the internal resistance of an ideal current source is infinite, while the resistor R_x serving as a voltage divider can be treated as the internal resistance of a voltage source. Delays of current-in voltage sensing, voltage divider sensing, and current sensing are given by the following equations using Seevinck's delay expression [34]:

$$\delta t_v = \frac{R_T C_T}{2} \times \left(1 + \frac{2R_B}{R_T}\right) \tag{2.14}$$

$$\delta t_{vd} = \frac{R_T C_T}{2} \times \left(1 + \frac{2(R_B || R_x)}{R_T}\right) \tag{2.15}$$

$$\delta t_i = \frac{R_T C_T}{2} \times \left(\frac{R_B + \frac{R_T}{3}}{R_B + R_T}\right) \tag{2.16}$$

where R_T and C_T are the total line resistance and capacitance, R_B is the equivalent resistance of the memory cell, and R_x is the resistance of voltage divider. In these equations, t_v, t_{vd}, and t_i are the RC delays of current-in voltage sensing, voltage divider sensing, and current sensing schemes, respectively. $R_x||R_B$, instead of R_B, is used as the new effective pull-down resistance in Eq. 2.15 according to the transformation from a Thevenin equivalent to a Norton equivalent.

Equations 2.14 and 2.15 show that voltage divider sensing is faster than current-in voltage sensing with the extra cost of fabricating a large resistor. Comparing Eq. 2.16 with 2.14 and 2.15, we can see the current sensing is much faster than current-in voltage sensing and voltage divider sensing since the former delay is less than the intrinsic line delay $R_T C_T/2$, while the latter delays are larger than $R_T C_T/2$. The bit line delay analytical models are verified by comparing them with the HSPICE simulation results. As shown in Fig. 2.21, the RC delays derived by our analytical RC models are consistent with the HSPICE simulation results.

2.5.2.2 Current–Voltage Converter Model

As shown in Fig. 2.18, the current–voltage converter in our current-mode sensing scheme is actually the first-level sense amplifier, and the CACTI-modeled voltage sense amplifier is still kept in the bit line model as the final stage of the sensing scheme. The current–voltage converter senses the current difference $I_1 - I_2$, and then, it is converted into a voltage difference $V_1 - V_2$. The required voltage difference produced by current–voltage converter is set by default to 80 mV. Although this value is the minimum sensible voltage difference of the CACTI-modeled voltage sense amplifier, advanced user can override it for specific sense amplifier design. We refer to a previous current–voltage converter design [34], and the circuit schematic is shown in Fig. 2.22. This sensing scheme is similar to the hybrid I/O approach [28], which can achieve high-speed, robust sensing, and low power operation.

To avoid unnecessary calculation, the current–voltage converter is modeled by directly using the HSPICE-simulated values and building a look-up table of delay, dynamic energy, and leakage power (Table 2.2).

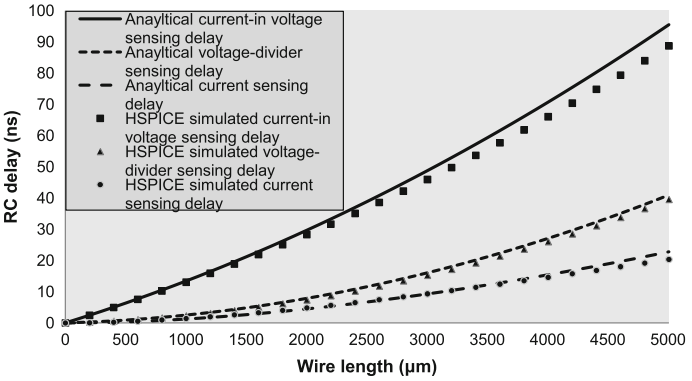

Fig. 2.21 Delay model verification of three sensing schemes comparing to HSPICE simulations

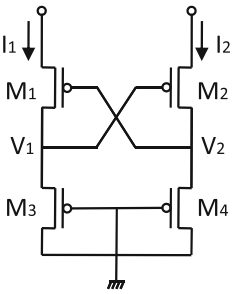

Fig. 2.22 The current–voltage converter modeled in NVSim

2.5.3 Cell Switching Model

Different NVM technologies have their specific switching mechanism. Usually, the switching phenomenon involves magnetoresistive, phase change, thermochemical, and electrochemical effects, and it cannot be estimated by RC analysis. Hence, the cell switching model in NVSim largely relies on the NVM cell definition. The predefined NVM cell switching properties include the SET/RESET pulse duration (i.e., t_{SET} and t_{RESET}) and SET/RESET current (i.e., I_{SET} and I_{RESET}) or voltage. NVSim does not model the dynamic behavior during the switching of the cell state, the switching latency (i.e., cell write latency) is directly the pulse duration and the switching energy (i.e., cell write energy) is estimated using Joule's first law that is

$$\begin{aligned} \mathrm{Energy}_{\mathrm{SET}} &= I_{\mathrm{SET}}^2 R t_{\mathrm{SET}} \\ \mathrm{Energy}_{\mathrm{RESET}} &= I_{\mathrm{RESET}}^2 R t_{\mathrm{RESET}} \end{aligned} \tag{2.17}$$

in which the resistance value R can be the equivalent resistance of the corresponding SET or RESET state (i.e., R_{SET} or R_{RESET}). However, for NVM technologies that have threshold switching phenomenon (e.g., PCRAM and ReRAM), the resistance

Table 2.2 The delay and power lookup table of current–voltage converter

Process node (nm)	130	90	65	45	32
Delay (ns)	0.49	0.53	0.62	0.80	1.07
Dynamic energy (fJ)	85.2	87.2	90.0	102.6	125.6
Leakage power (nW)	14.0	18.7	25.7	44.1	125.4

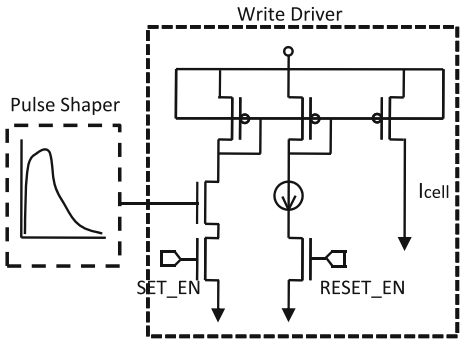

Fig. 2.23 The circuit schematic of the slow-quench pulse shaper used in [21]

value R always equals to the resistance of the low-resistance state. This is because when a voltage above a particular threshold is applied to these NVM cells in the high-resistance state, the resulting large electrical fields greatly increase the electrical conductivity [2].

2.6 Miscellaneous Circuitry

Some specialized circuitry is required for certain types of NVMs. For instance, some PCRAM chips need pulse shaper to reform accurate SET and RESET pulses, and NAND flash and some PCRAM chips need charge pump to generate the high-voltage power plane that is necessary for write operations.

2.6.1 Pulse Shaper

Some PCRAM needs specialized circuits to handle its RESET and SET operations. Specific pulse shapes are required to heat up the GST quickly and to cool it down gradually, especially for SET operations. This pulse shaping requirement is achieved by using a slow-quench pulse shaper. As shown in Fig. 2.23, the slow-quench pulse shaper is composed of an arbitrary slow-quench waveform generator and a write driver.

In NVSim, the delay impacts of the slow-quench shaper are neglected because they are already included in the RESET/SET calculation of the timing model. The energy impacts of the shaper are modeled by adding an energy efficiency during the RESET/SET operation, which we set the default value to 35 % [11], and it can be overridden by advanced user. The area of slow-quench shapers is modeled by measuring the die photographs [11, 21].

2.6.2 Charge Pump

The write operations of NAND flash and some PCRAM chips require voltage higher than the chip supply voltage. Therefore, a charge pump that uses capacitors as energy storage elements to create a higher voltage is necessary in a NAND flash chip design. In NVSim, we neglect the silicon area occupied by charge pump since the charge pump area can vary a lot depending on its underlying circuit design techniques, and the charge pump area is relatively small compared to the cell array area in a large-capacity NAND chip. However, we model the energy dissipated by charge pumps during the program and erase operations in NVSim because they contribute a considerable portion of the total energy consumption. The energy consumed by charge pumps is referred from an actual NAND flash chip design [16], which specifies that a conventional charge pump consumes 0.25 μJ at 1.8 V supply voltage. We use this value as the default in NVSim.

2.7 Validation Result

NVSim is built on the basis of generic assumptions of memory cell layouts, circuit design rules, and CMOS fabrication parameters, whereas the performance, energy, and area of a real non-volatile memory design depend on the specific choices of all these. However, as described in previous sections, we provide a set of knobs in NVSim to adjust the design parameters such as memory organization, wire type, transistor type, data sensing schemes, and others. Therefore, NVSim is capable of emulating a real memory chip, and comparing the NVSim estimation result to the actual memory chip parameters can show the accuracy of NVSim.

Hence, we validate NVSim against NAND flash chips and several industrial prototype designs of STT-RAM [40], PCRAM [1, 17], and ReRAM [35] in terms of area, latency, and energy. We first extract the information from real chip design specifications to set the input parameters required by NVSim, such as capacity, line size, technology node, and array organization. Then, we compare the performance, energy, and area estimation numbers generated from NVSim to the actual reported numbers in those chip designs. The validation results are listed in this section. Note that all the simulation results are for nominal cases since process variations are not supported in current version of NVSim.

Table 2.3 NVSim's NAND flash model validation with respect to a 50 nm 2 Gb NAND flash chip (B-SLC2) [10]

Metric	Actual	Projected	Error (%)
Area	23.85 mm^2	22.61 mm^2	−5.20
Read latency	21 μs	25.2 μs	+20.0
Program latency	200 μs	200.1 μs	+0.1
Erase latency	1.25 ms	1.25 ms	+0.0
Read energy	1.56 μJ	1.85 μJ	+18.6
Program energy	3.92 μJ	4.24 μJ	+8.2
Erase energy	34.5 μJ	36.0 μJ	+4.3

2.7.1 NAND Flash Validation

It is challenging to validate the NAND flash model in NVSim since the public manufacturer datasheets do not disclose sufficient data on the operation latency and power consumption for validation purpose. Instead, Grupp et al. [10] report both latency and power consumption measurements of several commercial NAND flash chips from different vendors. Grupp's report does not include the NAND flash silicon area; hence, we set the actual NAND flash chip area by assuming an area efficiency of 90 %. The comparison between the measurement [10] and the estimations given by NVSim is listed in Table 2.3. The estimation error is within 20 %.

2.7.2 STT-RAM Validation

We validate the STT-RAM model against a 65 nm prototype chip [40]. We let 1 bank $= 32 \times 8$ mats and 1 mat $= 1$ subarray to simulate the memory array organization. We also exclude the chip area of I/O pads and duplicated cells to make the fair comparison. As the write latency is not disclosed, we assume the write pulse duration is 20 ns. The validation result is listed in Table 2.4. The result shows that the area and the latency estimation error are within 3 and 5 %, respectively.

2.7.3 PCRAM Validation

We first validate the PCRAM model against a 0.12 μm MOS-accessed prototype. The array organization is configured to have 2 banks, each has 8×8 mats. Every mat contains only one subarray. Table 2.5 lists the validation result, which shows a 10 % underestimation of area and 6 % underestimation of read latency. The projected write latency (SET latency as the worst case) is also consistent with the actual value.

Table 2.4 NVSim's STT-RAM model validation with respect to a 65 nm 64 Mb STT-RAM prototype chip [40]

Metric	Actual	Projected	Error
Area	39.1 mm^2	38.05 mm^2	−2.69 %
Read latency	11 ns	11.47 ns	+4.27 %
Write latency	< 30 ns	27.50 ns	–
Write energy	N/A	0.26 nJ	–

Table 2.5 NVSim's PCRAM model validation with respect to a 0.12 μm 64 Mb MOS-accessed PCRAM prototype chip [1]

Metric	Actual	Projected	Error
Area	64 mm^2	57.44 mm^2	−10.25 %
Read latency	70.0 ns	65.93 ns	−5.81 %
Write latency	> 180.0 ns	180.17 ns	–
Write energy	N/A	6.31 nJ	–

Another PCRAM validation is made against a 90 nm diode-accessed prototype [21].

2.7.4 ReRAM Validation

We validate the ReRAM model against a 180 nm 4 Mb HfO_2-based MOS-accessed ReRAM prototype [35]. According to the disclosed data, the subarray size is configured to 128 Kb. We further model a bank with 4 × 8 mats, and each mat contains a single subarray. The validation result is listed in Table 2.7. Note that the estimated chip area given by NVSim is much smaller than the actual value since the prototype chip has SLC/MLC dual modes, but the current version of NVSim does not model the MLC-related circuitry.

2.7.5 Comparison to CACTI

We also test the closeness between NVSim and CACTI by simulating identical SRAM caches and DRAM chips. The results show that NVSim models SRAM and DRAM more accurately than CACTI does since some false assumptions in CACTI are fixed in NVSim.

2.8 Case Studies by Using NVSim

In this section, we conduct two case studies to demonstrate how we can use NVSim in two ways: (1) use NVSim to optimize the NVM designs toward certain design metric

Table 2.6 NVSim's PCRAM model validation with respect to a 90 nm 512 Mb diode-selected PCRAM prototype chip [21]

Metric	Actual	Projected	Error (%)
Area	91.50 mm^2	93.04 mm^2	+1.68
Read latency	78 ns	59.76 ns	−23.40
Write latency	430 ns	438.55 ns	+1.99
Write energy	54 nJ	47.22 nJ	−12.56

Table 2.7 NVSim's ReRAM model validation with respect to a 0.18 μm 4 Mb MOSFET-selected ReRAM prototype chip [35]

Metric	Actual	Projected	Error
Area[a]	187.69 mm^2	33.42 mm^2	–
Read latency	7.2 ns	7.72 ns	+7.22 %
Write latency	0.3–7.2 ns	6.56 ns	–
Write energy	N/A	0.46 nJ	–

[a] A large portion of the chip area is contributed to the MLC control and test circuits, which are not modeled in NVSim

and (2) use NVSim to estimate the performance, energy, and area before fabricating a real prototype chip, especially when the emerging NVM device technology is still under development and there is no standard so far.

2.8.1 Use NVSim for Design Optimization

NAND flash is currently the widely used firmware storage or disk in embedded systems. However, codes stored in NAND must be copied to random-accessible memory like DRAM before execution since NAND's page-accessible structure causes poor random access performance. If emerging NVM technologies such as STT-RAM, PCRAM, and ReRAM can be adopted in such systems, and their byte-accessibility property can eliminate the need of DRAM modules in such systems. But, the issue of directly adopting emerging NVM technologies as the NAND flash substitute comes from the observation that the current prototype has a much slower read/write latency than DRAM. In this case study, we use PCRAM as an example without the loss of generality. The technology node used in this case study is 90 nm. Table 2.8 shows the latency difference between an NAND chip, a DRAM chip, and a PCRAM prototype chip with the same 512 MB capacity.

The comparison shows that the PCRAM prototype chip is much slower than its DRAM counterpart. To overcome this obstacle, it is necessary to optimize PCRAM chips for latency at the expense of area efficiency by aggressively cutting word lines and bit lines or inserting repeaters. Such area/performance trade-off is also available for DRAM designs. However, in this case study, we keep the DRAM chip parameters

Table 2.8 Using PCRAM as direct replacement of NAND

A typical 90 nm 512 Mb NAND (*source* K9F1208X0C datasheet)	
Access unit	Page
Read latency	15 μs
Write latency	200 μs
Erase latency	2 ms
A 90 nm 512 Mb PCRAM (*source* [21], Table 2.6 for more details)	
Access unit	Byte
Read latency	78 ns (59.76 ns, NVSim estimation)
Write latency	430 ns (438.55 ns, NVSim estimation)
A typical 90 nm 512 Mb DRAM (*source* K4T51043Q datasheet)	
Access unit	Byte
tRCD	15 ns
tRP	15 ns

unchanged since the current DRAM specification is already the sweet spot explored by DRAM industry for many years. But for PCRAM, such performance optimization is necessary.

Table 2.9 shows the comparison before and after NVSim optimization. The result shows that the PCRAM read latency can be reduced from 59.76 to 16.23 ns by only cutting subarrays into smaller size (from 1024 × 1024 to 512 × 32). Although the PCRAM write latency does not reduce too much due to the inherent SET/RESET pulse duration, write latency is typically not in the critical path and can be tolerated using write buffers. As a result, the optimized PCRAM chip projected by NVSim can properly replace the traditional NAND+DRAM solution in the embedded system. The latency optimization is at the expense of increasing chip area, which rises from 93.04 to 102.34 mm^2.

2.8.2 Use NVSim for Early-Stage Estimation

Considering the facts that the research of some emerging NVM technologies (e.g., ReRAM) is still in an early stage and there are only a limited number of NVM prototype chips available for high-level computer architects understanding the technologies, we expect NVSim would be helpful in providing performance, energy, and area estimations at an early design stage. In this case study, we demonstrate how NVSim can predict the full design spectrum of a projected ReRAM technology when such a device is fabricated as an 8 MB memory chip. Table 2.10 lists the projection.

Table 2.11 tabulates the full design spectrum of this 32 nm 8 MB ReRAM chip by listing the details of each design corner. As shown in the result, NVSim can optimize the same design toward different optimization targets by exploring the full design space, which means that NVSim automatically tunes all the design knobs such as

Table 2.9 New PCRAM parameters after NVSim latency optimization

Parameter	Before optimization	After optimization
Subarray size	1024×1024	512×32
Area	$93.04\,mm^2$	$102.34\,mm^2$
Read latency	59.76 ns	16.23 ns
Write latency	438.55 ns	416.23 ns

Table 2.10 The projection of a future ReRAM technology

	MOS-accessed	Cross-point
Cell size	$4F^2$	$20F^2$
Maximum NMOS driver size	$100F$	
RESET voltage and pulse duration	2.0 V, 100 ns	
SET voltage and pulse duration	−2.0 V, 100 ns	
READ input	0.4 V voltage source, or 2 μA current source	
LRS resistance	10 kΩ	
HRS resistance	500 kΩ	
Half-select resistance	–	100 kΩ

array structure, subarray size, sense amplifier design, write method, repeater design, and buffer design. If necessary, NVSim can also be explored to use different types of transistor or wire models to get the best result.

2.9 Related Work

Many modeling tools have been developed during the last decade to enable system-level design exploration for SRAM- or DRAM-based cache and memory. For example, CACTI [39, 43] is a tool that has been widely used in the computer architecture community to estimate the performance, energy, and area of SRAM and DRAM caches. Evans and Franzon [8] developed an energy model for SRAMs and used it to predict an optimum organization for caches. eCACTI [25] incorporated a leakage power model into CACTI. Muralimanohar et al. [29] modeled large-capacity caches through the use of an interconnect-centric organization composed of mats and request/reply H-tree networks.

In addition, CACTI has also been extended to evaluate the performance, energy, and area for STT-RAM [6], PCRAM [7, 26], cross-point ReRAM [44], and NAND flash [27]. However, as CACTI is originally designed to model an SRAM-based cache, some of its fundamental assumptions do not match the actual NVM circuit implementations, and thereby, the NVM array organization modeled in these CACTI-like estimation tools deviates from the NVM chips that have been fabricated.

Table 2.11 The predicted full design spectrum of a 32 nm 8 MB ReRAM chip

Optimization target	Area	Read latency	Write latency	Read energy	Write energy	Leakage power
Area (mm^2)	**0.664**	5.508	8.071	2.971	3.133	1.399
Read latency (ns)	107.1	**1.773**	1.917	5.711	6.182	426.8
Write latency (ns)	204.3	200.7	**100.6**	202.8	203.1	518.2
Read energy (nJ)	1.884	0.195	0.234	**0.012**	0.014	4.624
Write energy (nJ)	13.72	25.81	13.06	12.82	**12.81**	12.99
Leakage (mW)	1372	3872	7081	6819	7841	**26.64**
Array structure	Xpoint	Xpoint	1T1R	Xpoint	Xpoint	1T1R
Subarray size	512 × 512	128 × 128	1024 × 2048	512 × 512	256 × 256	2048 × 4096
Routing scheme	Non-H-tree	H-tree	H-tree	H-tree	H-tree	Non-H-tree
SA placement	External	Internal	Internal	Internal	Internal	External
SA type	CIVS	CS	CS	VDS	VDS	VDS
Write method	SbR	EbR	Normal	SbR	SbR	Normal
Interconnect	Normal	Repeated	Repeated	Low swing	Low swing	Normal
Output buffer optimized for	Area	Latency	Latency	Area	Area	Area

2.10 Conclusion

STT-RAM, PCRAM, and ReRAM are emerging memory technologies for future non-volatile memories. The versatility of these upcoming NVM technologies makes it possible to use these NVM modules at other levels in the memory hierarchy, such as execute in place (XIP) memory, main memory, or even on-chip cache. Such emerging NVM design options can vary for different applications by tuning circuit structure parameters such as the array organizations and the peripheral circuitry types or by using devices and interconnects with different properties. To enable the system-level design space exploration of these NVM technologies and facilitate computer architects leverage these emerging technologies, it is necessary to have a quick estimation tool. While abundant estimation tools are available as SRAM/DRAM design assistants, similar tools for NVM designs are currently missing. Therefore, in this work, we build *NVSim*, a circuit-level model for NVM performance, energy, and area estimation, which supports various NVM technologies including STT-RAM, PCRAM, ReRAM, and conventional NAND flash. This model is successfully validated against industrial NVM prototypes, and this new *NVSim* tool is expected to help boost NVM-related studies such as the next-generation memory hierarchy.

References

1. Ahn, S.J., et al. (2004). Highly manufacturable high density phase change memory of 64Mb and beyond. *Proceedings of the International Electron Devices Meeting* (pp. 907–910).
2. Burr, G. W., et al. (2010). Phase change memory technology. *Journal of Vacuum Science and Technology B*, *28*(2), 223–262.
3. Chen, Y.C., et al. (2003). An access-transistor-free (0T/1R) non-volatile resistance random access memory (RRAM) using a novel threshold switching, self-rectifying chalcogenide device. *Proceedings of the International Electron Devices Meeting* (pp. 750–753).
4. Chen, Y.S., et al. (2009). Highly scalable hafnium oxide memory with improvements of resistive distribution and read disturb immunity. *Proceedings of the International Electron Devices Meeting* (pp. 105–108).
5. Chien, W., et al. (2009). Multi-level operation of fully CMOS compatible WO_x resistive random access memory (RRAM). *Proceedings of the International Memory, Workshop* (pp. 228–229).
6. Dong, X., et al. (2008). Circuit and microarchitecture evaluation of 3D stacking magnetic RAM (MRAM) as a universal memory replacement. *Proceedings of the Design Automation Conference* (pp. 554–559).
7. Dong, X., et al. (2009). PCRAMsim: System-level performance, energy, and area modeling for phase-change RAM. *Proceedings of the International Conference on, Computer-Aided Design* (pp. 269–275).
8. Evans, R. J., & Franzon, P. D. (1995). Energy consumption modeling and optimization for SRAM's. *IEEE Journal of Solid-State Circuits*, *30*(5), 571–579.
9. Fishburn, F., et al. (2004). A 78nm $6F^2$ DRAM technology for multigigabit densities. *Proceedings of the Symposium on VLSI Technology* (pp. 28–29).
10. Grupp, L.M., et al. (2009). Characterizing flash memory: Anomalies, observations, and applications. *Proceedings of the International Symposium on Microarchitecture* (pp. 24–33).
11. Hanzawa, S., et al. (2007). A 512kB embedded phase change memory with 416kB/s write throughput at 100μA cell write current. *Proceedings of the International Solid-State Circuits Conference* (pp. 474–616).

12. Horowitz, M. A. (1983). *Timing models for MOS circuits*. Tech. rep. California: Stanford University.
13. Hosomi, M., et al. (2005). A novel nonvolatile memory with spin torque transfer magnetization switching: Spin-RAM. *International Electron Devices Meeting* (pp. 459–462).
14. International Technology Roadmap for Semiconductors: Process Integration, Devices, and Structures 2010 Update. http://www.itrs.net/
15. International Technology Roadmap for Semiconductors: The Model for Assessment of CMOS Technologies And Roadmaps (MASTAR). http://www.itrs.net/models.html
16. Ishida, K., et al. (2009). A 1.8V 30nJ adaptive program-voltage (20V) generator for 3D-integrated NAND flash SSD. *Proceedings of the IEEE International Solid-State Circuits Conference* (pp. 238–239,239a).
17. Kang, S., et al. (2007). A 0.1 μm 1.8V 256Mb phase-change random access memory (PRAM) with 66MHz synchronous burst-read operation. *IEEE Journal of Solid-State Circuits*, *42*(1), 210–218.
18. Kau, D., et al. (2009). A stackable cross point phase change memory. *Proceedings of the IEEE International Electron Devices Meeting* (pp. 27.1.1-27.1.4).
19. Kawahara, T., et al. (2007). 2Mb spin-transfer torque RAM (SPRAM) with bit-by-bit bidirectional current write and parallelizing-direction current read. *IEEE International Solid-State Circuits Conference* (pp. 480–617).
20. Kim, K. H., et al. (2010). Nanoscale resistive memory with intrinsic diode characteristics and long endurance. *Applied Physics Letters*, *96*(5), 053,106.1-053,106.3.
21. Lee, K. J., et al. (2008). A 90nm 1.8V 512Mb diode-switch PRAM with 266MB/s read throughput. *IEEE Journal of Solid-State Circuits*, *43*(1), 150–162.
22. Lee, M.J., et al. (2007). 2-stack 1D–1R cross-point structure with oxide diodes as switch elements for high density resistance RAM applications. *Proceedings of the IEEE International Electron Devices Meeting* (pp. 771–774).
23. Liang, J., & Wong, H. S. P. (2010). Cross-point memory array without cell selectors: Device characteristics and data storage pattern dependencies. *IEEE Transactions on Electron Devices*, *57*(10), 2531–2538.
24. Lin, W., et al. (2010). Evidence and solution of over-RESET problem for HfO_x based resistive memory with sub-ns switching speed and high endurance. *Proceedings of the International Electron Devices Meeting* (pp. 19.7.1-19.7.4).
25. Mamidipaka, M., Dutt, N. (2004). eCACTI: An enhanced power estimation model for on-chip caches. Tech. Rep. TR04-28, Center for Embedded Computer Systems.
26. Mangalagiri, P., et al. (2008). A low-power phase change memory based hybrid cache architecture. *Proceedings of the Great Lakes Symposium on VLSI* (pp. 395–398).
27. Mohan, V., et al. (2010). FlashPower: A detailed power model for NAND flash memory. *Proceedings of Design, Automation and Test in, Europe* (pp. 502–507).
28. Moon, Y., et al. (2009). 1.2V 1.6Gb/s 56nm $6F^2$ 4Gb DDR3 SDRAM with hybrid-I/O sense amplifier and segmented sub-array architecture. *Proceedings of the International Solid-State Circuits Conference* (pp. 128–129).
29. Muralimanohar, N., et al. (2008). Architecting efficient interconnects for large caches with CACTI 6.0. *IEEE Micro*, *28*(1), 69–79.
30. Oh, H. R., et al. (2006). Enhanced write performance of a 64-Mb phase-change random access memory. *IEEE Journal of Solid-State Circuits*, *41*(1), 122–126.
31. Oh, J.H., et al. (2006). Full integration of highly manufacturable 512Mb PRAM based on 90nm technology. *Proceedings of the International Electron Devices Meeting* (pp. 49–52).
32. Pellizzer, F., et al. (2004). Novel μTrench phase-change memory cell for embedded and stand-alone non-volatile memory applications. *Proceedings of the International Symposium on VLSI Technology* (pp. 18–19).
33. Raoux, S., et al. (2008). Phase-change random access memory: A scalable technology. *IBM Journal of Research and Development*, *52*(4/5),
34. Seevinck, E., et al. (1991). Current-mode techniques for high-speed VLSI circuits with application to current sense amplifier for CMOS SRAM's. *IEEE Journal of Solid-State Circuits*, *26*(4), 525–536.

35. Sheu, S.S., et al. (2011). A 4Mb embedded SLC resistive-RAM macro with 7.2ns read-write random-access time and 160ns MLC-access capability. *Proceedings of the IEEE International Solid-State Circuits Conference* (pp. 200–201).
36. Smullen, C., et al. (2011). Relaxing non-volatility for fast and energy-efficient STT-RAM caches. *Proceedings of the International Symposium on High Performance Computer, Architecture* (pp. 50–61).
37. Sutherland, I. E., et al. (1999). *Logical effort: designing fast CMOS circuits*. Morgan Kaufmann.
38. Thoziyoor, S., et al. (2008). A comprehensive memory modeling tool and its application to the design and analysis of future memory hierarchies. *Proceedings of the International Symposium on Computer, Architecture* (pp. 51–62).
39. Thoziyoor, S., et al. (2008). CACTI 5.1 technical report. Tech. Rep. HPL-2008-20, HP Labs.
40. Tsuchida, K., et al. (2010). A 64Mb MRAM with clamped-reference and adequate-reference schemes. *Proceedings of the International Solid-State Circuits Conference* (pp. 268–269).
41. Udipi, A. N., et al. (2010). Rethinking DRAM design and organization for energy-constrained multi-cores. *ACM SIGARCH Computer Architecture News*, *38*(3), 175–186.
42. Wei, Z., et al. (2008). Highly reliable TaO_x ReRAM and direct evidence of redox reaction mechanism. *Proceedings of the International Electron Devices Meeting* (pp. 293–296).
43. Wilton, S. J. E., & Jouppi, N. P. (1996). CACTI: An enhanced cache access and cycle time model. *IEEE Journal of Solid-State Circuits*, *31*, 677–688.
44. Xu, C., et al. (2011). Design implications of memristor-based RRAM cross-point structures. *Proceedings of Design, Automation and Test in, Europe*, (pp. 1–6).
45. Yang, J. J., et al. (2008). Memristive switching mechanism for metal/oxide/metal nanodevices. *Nature Nanotechnology*, *3*(7), 429–433.
46. Yoshitaka, S., et al. (2009). Cross-point phase change memory with $4F^2$ cell size driven by low-contact-resistivity poly-Si diode. *Proceedings of the Symposium on VLSI Technology* (pp. 24–25).
47. Zhang, Y., et al. (2007). An integrated phase change memory cell with Ge nanowire diode for cross-point memory. *Proceedings of the IEEE Symposium on VLSI Technology* (pp. 98–99).