Quantitative Modeling of Racetrack Memory, A Tradeoff among Area, Performance, and Power

Chao Zhang†, Guangyu Sun†, Weiqi Zhang†, Fan Mi‡, Hai Li‡, Weisheng Zhao§ †CECA, Peking University, China
‡University of Pittsburgh, U.S.A.
§Spintronics Interdisciplinary Center, Beihang University, China
Email: †{zhang.chao, gsun, zhangweiqi}@pku.edu.cn, ‡mf198942@gmail.com, ‡hal66@pitt.edu, §weisheng.zhao@u-psud.fr

***Abstract*—Recently, an emerging non-volatile memory called Racetrack Memory (RM) becomes promising to satisfy the requirement of increasing on-chip memory capacity. RM can achieve ultra-high storage density by integrating many bits in a tape-like racetrack, and also provide comparable read/write speed with SRAM. However, the lack of circuit-level modeling has limited the design exploration of RM, especially in the system-level. To overcome this limitation, we develop an RM circuit-level model, with careful study of device configurations and circuit layouts. This model introduces Macro Unit (MU) as the building block of RM, and analyzes the interaction of its attributes. Moreover, we integrate the model into NVsim to enable the automatic exploration of its huge design space. Our case study of RM cache demonstrates significant variance under different optimization targets, in respect of area, performance, and energy. In addition, we show that the cross-layer optimization is critical for adoption of RM as on-chip memory.**

I. Introduction

With the rapid development of computing systems, there is an urgent demand of increasing on-chip memory capacity. Memory researchers are now finding alternatives of static random access memory (SRAM), such as embedded DRAM (eDRAM), and spin-transfer torque random access memory (STT-RAM). They can achieve $2 \sim 4$ times storage density over SRAM. Recently, a new type of non-volatile memory (NVM), racetrack memory (RM), which draws attention of researchers [1], [2], [3], [4], further achieves a higher storage density. Compared with STT-RAM, RM achieves about 12 times storage density, and keeps similar read/write speed.

Dr. Parkin et al. [1] first proposed the racetrack memory (RM) in 2008. They pointed out the tremendous application potential of racetrack memory. In 2011, Dr. Annunziata et al. [5] demonstrated the first 200mm RM wafer (real fabricated chips). It was fabricated with IBM 90nm CMOS technology, and each die contained 256 racetrack cells, which approves the feasibility of RM fabrication. Venkatesan et al. first explored the application of RM as on-chip CPU caches [2], and on-chip GPGPU caches [4]. They found RM could achieve about $8\times$ storage density, and about $7\times$ energy reduction as on-chip CPU caches. And RM-based on-chip cache could improve GPGPU performance by 12.1% and reduce energy about $3.3\times$ over SRAM. Sun et al. [6] further proposed hierarchical and dense architecture for racetrack (HDART). Their RM-based cache achieved about $6\times$ chip area reduction, 25% performance improvement, and 62% energy reduction over STT-RAM.

However, system-level analysis cannot fully explore the RM circuit design space, because of the lack of a circuit-level model. Several works [2], [4] seems ignored the shrink potential of racetrack width. Sun et al. estimated the system-level performance with fixed RM configuration [6]. Without the circuit level model and its simulation tool, further research will also be limited. In order to fully explore the system-level design and optimization, we need a quantitative and automatic simulation tool.

NVsim [7] is the most popular NVM modeling tool. But it cannot support the RM efficiently by taking a racetrack as its cell. First, different RM "cells" would overlap and affect each other, which induces significant layout inefficiency. Second, in order to access different bits stored in a single racetrack cell, a cell need more than one access ports. Third, a multi-bit cell may require various shift effort (latency and energy) to access different bits. Because the multi-port cell and shift operation are not modeled in NVsim, we need a new tool to quantitatively model the RM circuit design.

Thus, we propose a RM circuit-level model, with careful study of device configurations and circuit layouts. In order to enable automatic exploration of the huge RM design space, we propose a macro unit (MU) design to integrate the RM into NVsim. We also analyze the interactive impact in MU parameters. Then we perform a cross-layer optimization for area, latency, and energy.

This paper is organized as follows. We first introduce the preliminary on racetrack memory and NVsim in Section II. Racetrack memory modeling is presented in Section III. RM cross-layer optimization is conducted in Section IV. We conclude our work after a cross-layer case study in Section V.

II. Preliminary

A. Racetrack Memory

Racetrack Memory (RM) is a new variation of magnetic random access memory (MRAM). It stores many bits in racetrack like stripes, which achieves ultra-high storage density. It inherits various distinctive attributes of MRAM, including non-volatility, high density, low leakage power, and fast read access, etc. [8], [9], [2], [6], [10], [11]. The compatibility with CMOS technology and its distinguish scalability make RM a promising candidate to replace SRAM as on-chip memory in the future [2], [6], [12].

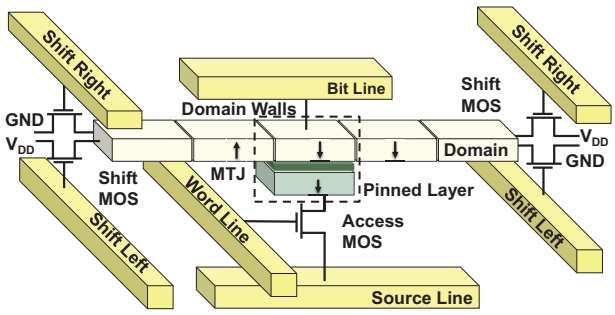

Fig. 1. An illustration of racetrack (RT) structure.

Racetrack memory (RM), also known as Domain Wall Memory (DWM), takes a magnetic nanowire stripe, called racetrack (RT), as its basic building block, as illustrated in Figure 1. The physical layout of the racetrack contains successive notches or pinning sites, where domain walls (DW) can rest. A bit in RT is represented by the magnetization direction of a domain. A phenomenon called spin-momentum transfer can shift the domain walls in lock step mode. The

978-1-4799-7792-5/15/$31.00 ©2015 IEEE

DWs move in most case opposite to the current direction. Note that the DW shift will not cause mechanical movement of the material, but the change of domains magnetisation direction.

Basic operations, including read, write and shift, are performed through ports. Read is done via access port by measuring the magneto-resistance. Write is performed by activating a high write current through the access port, to flip the magnetization direction of a specific domain. A "shift" operation is to drive the domain walls under a spin-polarized current pulse through the entire RT stripe, controlled by the shift ports.

An access port, consisting of a magnetic tunneling junction (MTJ) and a word line controlled transistor, executes the read and write, as illustrated in Figure 1. The MTJ is a sandwich structure: a pinned layer with fixed magnetization direction lies on the bottom, a MgO barrier is in the middle, and a domain in RT works as the top layer. The bit-line (BL), word-line (WL), and source-line (SL) are connected to the domain of MTJ, to the gate of access transistor, and to the source of access transistor, respectively. The transistor controls the current density through MTJ, which determines the latency of read and write in a port. The magnetization direction in the domain represents the stored value. When the magnetization direction of domain is antiparallel against the pinned layer of MTJ, a high resistance will be archived from source line to bit line, indicating logic "1". And a parallel direction between the domain and the pinned layer stores logic "0". As shown in Figure 1, domains with magnetization direction as "up, down, down" are formed successively. The shift port consists of a pair of transistors connected at both ends of the RT. The current through the RT will shift all domain walls opposite to the current direction, after shift control lines turn on corresponding pair of transistors in the shift ports. If a shift pulse pushes the domain walls to right, the data sequence read out from the port will be "100". And similarly it will be "001", if the domain walls move to left.

B. NVSim Modeling Framework

NVSim is a circuit-level model, which facilitates the NVM system-level exploration before real chip fabrication [7]. It takes device parameters as input, optimizes the NVM circuit designs, and evaluates the area, performance, and energy under given design specification. It supports various types of NVM, including STT-RAM, PCRAM, ReRAM, and Flash.

As shown in Figure 2, a chip in NVsim can be organized as three levels: bank, mat, and array. Bank is the top level unit, mat is the building block of bank, and array is the elementary structure. An array contains multiple cells and corresponding periphery circuitry. And a cell stores one bit and multiple cells should not be overlapped. The layout of a cell is dominated by the MTJ or the access transistor. The periphery circuitry includes decoders, multiplexers, sense amplifiers, output drivers and rout pathes. Routes from I/O interface ,via bank and mat, to array are modeled with address wires, broadcast data wires and distributed data wires. Thus, NVsim evaluates the circuit area, performance and power by estimating cells and their periphery circuitry. Different memory types change the input cell parameters, while keeping the modeling of periphery circuitry unchanged.

III. Racetrack Memory Modeling

Due to the tape-like cell shape, it is not area-efficient to organize RM cells as traditional array-like memory. Thus, we first introduce the concept of Macro Unit (MU), which is considered as the basic unit to build an RM array. And we use share degree to describe the bit density for an access port. We further model the basic operations, including read, write, and shift, based on the MU. The extra circuitry that enables the shift operation is also introduced. At last, we discuss the impact of MU parameters.

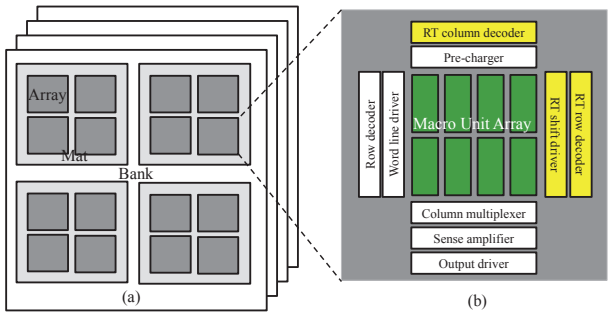

Fig. 2. An illustration of a racetrack memory bank. (a) Overview of the bank; (b) Detailed view of array. White and gray components are existing parts, and yellow ones are new for RM.

A. Macro Unit

Figure 3 (a) and (b) show the side and top views of one RM cell layout. There are one access transistor and two shift transistors. The racetrack is stacked on top of these transistors. Each end of the racetrack has two shift transistors. The insulator (MgO) and pinned magnetic layer are sandwiched between the racetrack and the access transistor, which makes up one access port. Apparently, there is a lot of unused space in the layout of one RM cell, because a transistor is normally wider than a racetrack . If we organize such a RM cell to achieve an array as traditional memory type, the area efficiency is quite low due to the blank space in the cell [2].

In order to utilize the layout efficiently, multiple RM cells can be overlapped with each other [6]. An example of two overlapped RM cells is shown in Figure 3 (c) and (d). All shift transistors and access transistors of both cells are aligned vertically, as illustrated in Figure 3 (d). In this work, such a layout with overlapped RM cells is called Macro Unit (MU). MU performs as the basic building block of RM array. Obviously, we can increase the number of RM cells in one MU to further improve area efficiency. Figure 3 (e) shows an MU, which is composed of four racetrack cells. In addition, the number of access ports on each racetrack is doubled. The dashed box in the figure illustrates the area of domains that can be accessed by the set of access ports. Note that the area above the dashed box is design overhead for racetrack shifting.

We define the device-level and circuit-level design parameters of an MU in Table I and Table II, respectively. F is the technology feature size. The range of circuit-level parameters in Table II is constrained by those device-level ones. For example, the number of domains is limited by the length of racetrack and length of domain. A 128F-long racetrack with 2F domain length can at most have 64 domains, including domains designed for overhead. When the device-level parameters are fixed, the layout of a MU is only determined by three circuit-level parameters, which include the number of domains in each racetrack (except the overhead domains), the number of access ports in an MU, and the number of racetracks in an MU. We name the three parameters as MU configuration parameters. Thus, in the rest of this paper, each MU design is labeled with MU-N_D-N_P-N_{RT}. For example, the MU in Figure 3 (a) and (b) is MU-6-1-1, the MU in Figure 3 (c) and (d) is MU-6-2-2.

Apparently, we can estimate the area of MU with these parameters. Due to the size mismatch of the transistor layer and the racetrack layer, the length and width of MU are outlined by the bigger ones. The area is estimated by Equation 1. And we also estimate the matching level by comparing the big and small edges. We label this value as MU area efficiency (η), defined by Equation 2. For example, MU area of MU-6-1-1 is 240 F^2, and the area efficiency is 4.17%.

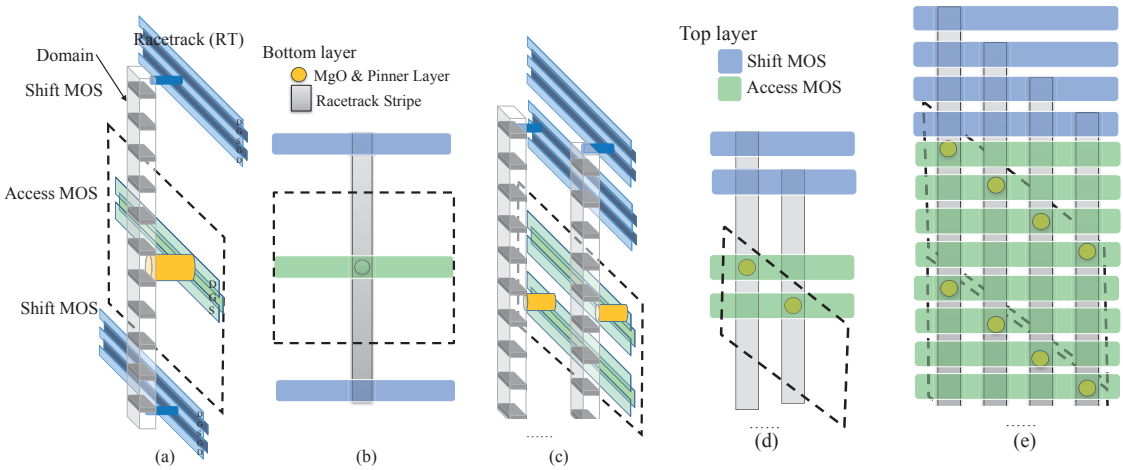

Fig. 3. The layouts of macro unit (MU). (a) Side view of one cell MU layout. (b) Top view of one cell MU layout. (c) Side view of MU layout for overlapped cells. (d) Top view of MU layout for overlapped cells. (e) Top view MU layout of four overlapped cells.

TABLE I
TYPICAL VALUE AND DEFINITION OF MU DEVICE-LEVEL PARAMETERS.

Parameter	Description	Default Value
W_{RT}	Width of racetrack	1F
G_{RT}	Gap distance between two racetracks	1F
L_D	Length of the domain in a racetrack	2F
L_{RT}	Length of racetrack	128F
T_{RT}	Thickness of racetrack	6nm
W_{NMOS}	Gate width of transistor (MOS)	10F
L_{NMOS}	Length of transistor (MOS)	4F
G_{NMOS}	Gap distance between two transistor	1F
W_{via}	Width of connection via for SL	1F
ρ	PMA racetrack nanowire resistivity	$4.8 \times 10^{-7} \Omega m$
TMR(0)	TMR with 0 V_{bias}	150 %
J_{shift}	Critical current density for shift	$6.2 \times 10^{7} A/cm^2$
$J_{nucleation}$	Critical current density for write	$5.7 \times 10^{6} A/cm^2$
RA	Resistance-area product	$10 \Omega \mu m^2$

$$\begin{aligned} W_{big} &= MAX\{N_{RT} \times (W_{RT} + G_{RT}), W_{MOS}\} \\ L_{big} &= MAX\{L_{RT}, N_P \times (L_{MOS} + G_{MOS})\} \\ Area &= W_{big} \times L_{big} \end{aligned} \quad (1)$$

$$\eta = \frac{H_{small} W_{small}}{H_{big} W_{big}} \quad (2)$$

B. Share Degree

Because the number of ports that can be actually fabricated is much fewer than that of domains, several domains will have to share one access port. We define the number of domains that share the same port as a circuit-level parameter, share degree (N_{shr}), in this work. With increased N_{shr}, more bits can be stored under the same area, which increases the data density significantly. But larger sharing degree also means that it costs more shift operations to get a required domain. This constrain of MU will finally affect system-level performance.

N_{shr} is calculated by Equation (3). And N_{PPR} stands for the number of ports attached to a racetrack, which is estimated as $N_{PPR} = \frac{N_P}{N_{RT}}$. The share degree is equal to the number of domains in the dash box shown in Figure 3. RM takes this part as valid bits, and keeps them in overhead region when these bits are shifted out. We can find that the share degree is a combination of MU configuration parameters.

$$N_{shr} = \frac{N_D}{N_{PPR}} = \frac{N_{RT} N_D}{N_P} \quad (3)$$

Because of the share degree, there is a tradeoff in access transistor width. We draw the share degree accompanied with read/write latency, corresponding to access transistor width in Figure 4. As the W_{MOS} increases, share degree increases in stairs, which means the maximum shift distance for a read/write will increase. Thus, to keep fast read/write access, access transistor width should neither be too large or too small. It is set at $8F$ typically, when the read latency is about $4ns$, write latency is $8ns$, and share degree is 8.

TABLE II
DEFINITION OF MU CIRCUIT-LEVEL PARAMETERS.

Parameter	Description
N_D	Number of domains for storage in a RT
N_P	Number of access ports in an MU
N_{RT}	Number of racetracks in an MU
η	Area efficiency of MU
N_{PPR}	Number of ports attached to a racetrack
N_{shr}	Number of domains sharing a port

C. Read and Write Operations

The mechanism of racetrack data read and write is similar to STT-RAM. A data access port consists of an access transistor and an MTJ. Read is performed via access port by measuring the magneto-resistance through the MTJ, after applying read voltage. Write is performed by activating a high write current pulse through MTJ, to change the magnetization direction of a domain in the racetrack. Read latency consists of latency on address decoding, bit line sensing, and output driving. Write latency consists of delay on address decoding and magnetization direction transformation.

In order to model the read and write operation of RM, we reuse peripheral circuitry for STT-RAM in NVsim. Simulation of read and write latency with various access transistor width is shown in Figure 4. Large W_{MOS} provides higher read/ write current density, but introduces larger capacity reflected in a word line. If W_{MOS} increases, magnetization direction switching latency will decrease, but the RC delay in the routing pass will increase. Thus, the difference between RM read latency and write latency is reduced when the width of access transistor is increased.

D. Shift Operation

Shift operation in RM is achieved by pushing and pulling domain walls under shift current pulse. Higher current density consumes more power to achieve lower shift latency. As shown in Figure 1, the shift transistors at both ends of RT provide the required current. For example, when a RT is selected to be shifted right, array periphery circuity will supply a positive pulse to the driver transistors at the bottom-left and the bottom-right. If there is no longer of the driven pulse, the transistors will be turned off, and all domains in the racetrack will stop moving.

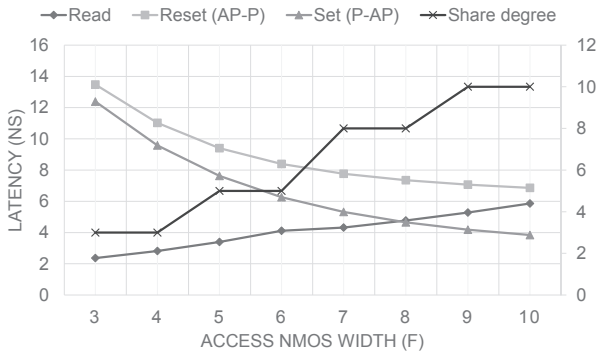

Fig. 4. The relation between read/write latency and access transistor width. Share degree is also shown with right axis. 32MB RM is evaluated with read/write latency optimized solutions.

Both the latency and energy of a shift operation depend on the value of shift current. The latency can be calculated by Equation 4. The latency (T) depends on the length of domain (L_D) and the velocity of DW movement (V). We model the DW movement velocity with Equation 5, given by previous work [13].α and β are material related coefficients. And μ_B, e, P, M_S represent Bohr magnetron, electron charge, and spin-polarization, and demagnetization field, respectively. j_p stands for the shift current density. The DW movement velocity is determined by the shift current density, which is controlled by the size of driver transistor. Thus, the shift latency is influenced both by racetrack length and the shift driver transistor. And in order to shift racetrack faster, racetrack should be as short as possible.

$$T = \frac{L_D}{V} \tag{4}$$

$$V = \frac{\beta}{\alpha}\frac{\mu_B P j_p}{e M_S} \tag{5}$$

Shift energy is consumed to move all Domain Walls in an RT. If the shift distance is one domain length, the shift energy is estimated by Equation (6). The energy is the production of shift current square, resistance of racetrack stripe, and the shift duration. When the shift distance is n domain length, shift energy is multiplied by n times.

$$E_{shift} = I_p^2 \rho \frac{L_{RT}}{T_{RT} W_{RT}}\ T \tag{6}$$

E. Periphery Circuitry in an Array

Conventional array components, including row decoder, word line driver, pre-charger, column multiplexer, sense amplifier and output driver, are dedicated to domain access. All MU Ports are connected to word lines selected by row decoder. And only ports of different RTs can share one word line. The word lines select specific ports in MU to access the data in domain. Column multiplexer converges the bit lines, selects the required data, and sends them to sense amplifier. The sense amplifier and output driver drive the output signal to output interface finally.

Due to the shift operation, we need to add some new components to the array, including RT row decoder, RT column decoder and RT shift driver, shown as yellow parts in Figure 2(b). RT row/column decoder is dedicated to decode which RT in the array we should shift. The row and column decoders are connected to the shift transistors, each at one end of RT. We can share the shift transistor with RTs in a MU, to reduce the number of lines generated by RT decoders. But this will reduce the shift current and prolong the shift latency. Because the fanout of decoder is generally power of two, it's better to set N_P and N_{RT} following these numbers.

F. Interactive impact of MU parameters

As mentioned before, device-level parameters and circuit-level parameters have impact on each other. Device-level parameters limit the valid value space of circuit-level parameters. In this subsection, we mainly explore the constrains on MU configuration parameters.

1) Number of Domains: The maximum number of domains to storage (N_{Dmax}) can be fabricated in an RT is determined by the feature size, the number of ports connected to an RT, and the length of RT. With more ports connected to an RT, the max shift distance for a domain to be accessed reduces. Thus, few domains are required to store overhead bits, which saves more domains to store bits. N_{Dmax} is calculated in Equation (7). The relationship between N_{Dmax} and L_{RT} is linear, but affected by the number of ports per racetrack.

$$N_D \le N_{Dmax} = \frac{N_{PPR}}{1 + N_{PPR}}(L_{RT}/L_D - N_{sep}N_{PPR}) \tag{7}$$

2) Number of Ports: The maximum number of ports allowed in an MU, N_{Pmax}, is limited by the length of racetrack, given by Equation 8. Note that access transistor cannot be stacked at racetrack overhead segment. Thus, the overhead segment length, $N_D L_D 2^{1-N_P/N_{RT}}$, should be deducted. With the equation , we find the typical value of N_{Pmax} for PMA racetrack in Table I is 32.

$$N_P \le N_{Pmax} = \frac{L_{RT} - N_D L_D 2^{1-N_{PPR}}}{L_{NMOS} + G_{MOS}} \tag{8}$$

3) Number of Racetracks: The maximum number of RTs in an MU, N_{RTmax}, is determined by the width of access transistor, described as Equation (9). The space for connection between source line and access transistor source field should also be reserved, and we estimate the width of the via space as W_{via}.

$$N_{RT} \le N_{RTmax} = \lfloor \frac{W_{MOS} - W_{via}}{W_{RT} + G_{RT}} \rfloor \tag{9}$$

If there is only one racetrack layer exists in an MU, the G_{RT} should be kept as a positive value, which limits the maximum RT in an MU. If multiple layers of racetrack can be fabricated and aligned, the gap distance G_{RT} could be smaller, or even becomes a negative value.

IV. Cross-Layer Optimization

From Section III, we can find that device level design parameters have impact on area, performance, and energy of a MU. Consequently, it further enlarges circuit-level design space of RM significantly. Thus, we perform a cross-layer optimization with the aforementioned model integrated into NVsim. The optimization targets include area, latency, energy for each basic operation.

We design a 32MB RM data array under 65 nm process node in our extension of NVsim. To support simulation of MU, the major modification is in organization of memory array. We reuse most simulation framework for peripheral circuitry in NVsim and add extra components highlighted in Figure 2. In addition, we enable cross-layer optimization by exhaustively searching both device level and circuit level design parameters. Value and physical equations of device-level parameters are collected from previous works [14], [5], [13], [15], [16], [17], and are listed in Table I.

A. Comparison among different optimization goals

We compare the solutions for different targets, including area, leakage power, and latency/energy for read, write, and shift. The solutions for different targets are shown in Table III. Area optimized solution occupies only $6.89mm^2$ chip area. Read, write and shift latency can as low as $3.78ns$, $10.23ns$, $2.13ns$, respectively. Read, write and shift Energy can as low as $224pJ$, $998pJ$, $124pJ$, respectively.

TABLE III
COMPARISON AMONG DIFFERENT OPTIMIZATION TARGETS.

Optimization Target	Area	Read Latency	Write Latency	Shift Latency	Read Energy	Write Energy	Shift Energy	Leakage Power
Bank Area (mm^2)	**6.89**	11.12	11.12	12.82	8.70	8.70	8.72	37.88
Read Latency (ns)	5.83	**3.78**	3.78	3.91	17.68	17.68	17.68	6.64
Write Latency (ns)	12.49	10.23	**10.23**	10.27	24.36	24.36	24.36	12.60
Shift Latency (ns)	5.31	4.95	4.95	**2.13**	8.15	5.80	5.29	6.09
Read Energy (pJ)	236.63	337.62	337.62	380.37	**224.18**	224.64	225.55	540.26
Write Energy (pJ)	1032.00	1140.00	1140.00	1179.00	998.27	**998.27**	998.27	1330.00
Shift Energy (pJ)	214.61	328.62	328.62	325.87	166.10	132.93	**123.87**	515.90
Leakage Power (mW)	163.72	407.09	407.09	407.09	95.75	110.61	140.31	**52.74**
MU Configuration	MU-64-32-4	MU-64-32-2	MU-64-32-2	MU-16-8-2	MU-64-8-4	MU-32-8-4	MU-16-8-4	MU-64-1-1

Leakage power can be as low as $52.74mW$. We find that the shift latency for a domain is comparable with read latency, and the energy consumption is smaller than read dynamic energy.

Considering optimized solutions for area and read latency, we find that increasing number of racetrack in MU not only increases the storage density, but also increases the read/write latency. In order to achieve a better read and write latency, the number of racetracks should not be too large. This is because adding more racetracks in a MU will significantly increase the capacity of row decoder output and thus increase the latency. In order to keep low leakage power, number of ports should be small enough to reduce transistor number. But at this point, the performance might be influenced, due to the share degree is very high (64 in this configuration).

B. Analysis of area efficiency

Racetrack memory is expected to demonstrate high storage density, but single cell layout (Figure 3(a)) leads to quite low area efficiency. Thus we further explore the MU configuration parameters to improve the area efficiency.

As shown in Table IV, different MU configurations achieve different area efficiency. The chip area optimized solution for a 32MB data array is MU-64-32-4. Its MU has 4 RTs, 64 access ports, and each RT has 64 storage domains. It occupies only $6.051mm^2$ chip area with MU efficiency $\eta = 79.01\%$, which indicates a low equivalent cell size ($5.062F^2$).

Comparing MU-1-1-1 and MU-32-1-1, we find that long racetrack benefits very few for area. This is because MU area efficiency does not change so much even with long racetracks. Comparing MU-32-8-4 and MU-32-16-4, we find that increasing port density in MU can reduce the chip area, instead of expanding the cell. Comparing MU-64-32-2 and MU-64-32-4, we find that more RTs in an MU increases the area efficiency significantly. Thus, MU configuration parameters can significantly impact the circuit area.

C. Comparison between RM and Other Memories

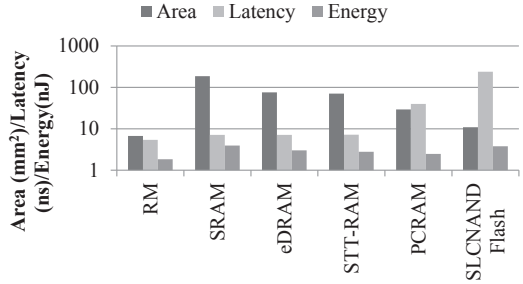

Fig. 5. Comparison with other memories as 32MB, on chip area, read latency and read energy.

With the optimized solutions, we compared the area, latency and energy of RM with other memories. We took SRAM, STT-RAM, PCRAM, Flash, and DRAM as references. All the designs were optimized for 32MB read latency optimized solution.

The chip area, read latency and read energy are evaluated in the Figure 5. RM shows about $12\times$ storage density compared with STT RAM, and about $28\times$ over SRAM. And the read latency of RM is comparable with SRAM, and about 75% over STT-RAM. The read energy for RM is the lowest in all the competitors, about 40% over SRAM. The advantage on read latency and energy is largely due to its small layout, which reduces the cost in periphery circuitry. Thus, RM is competitive to be deployed into memory hierarchy.

V. RM CACHE CASE STUDY

In order to explore the tradeoff of RM configuration in memory hierarchy, we conduct a case study on RM based Last Level Cache (LLC). The system is configured with a 4-core 2.0GHz CPU, private 32KB/32KB L1 instruction/data caches and a shared LLC. L1 is implemented with SRAM, and the read/write latency for L1 cache is 2-cycle. L2 is implemented with RM, and the read, write and shift latency is listed in Table III. Benchmarks are from SPEC CPU 2006 suite. Evaluation on gem5 [18] is conducted with a 10-billion instruction segment after fast forwarding the initial segment for each benchmark.

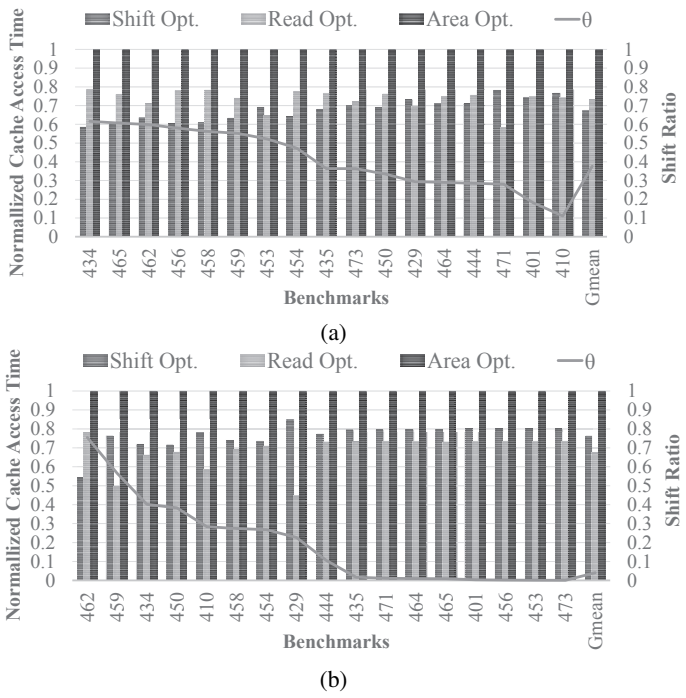

Fig. 6. Normalized cache access time for shift, read and area optimized solutions. (a) Performance without swap. (b) Performance with swap. Shift ratio is drawn as line with corresponding right axis. Cache access time is shown as bars with corresponding left axis.

We compared cache performance by cache access time with area-optimized, read-optimized and shift-optimized solutions. In order to

TABLE IV
AREA EXPLORATION FOR DIFFERENT MU CONFIGURATIONS.

MU Configuration	MU-1-1-1	MU-16-8-4	MU-32-1-1	MU-32-8-4	MU-32-16-4	MU-64-32-2	MU-64-32-4
Bank Area (mm^2)	37.07	7.901	36.766	7.89	6.7	11.12	6.051
MU Efficiency η (%)	37.5	59.259	1.172	29.63	71.111	58.82	79.012
Equivalent Cell (F^2)	32	6.75	32	6.75	5.625	8.50	5.062

make a fair comparison, area solution is 64MB, which keeps the chip size similar to others. We simulated both with and without shift-reduction technique "swap" proposed in Sun's work [6], to investigate the impact of architecture improvement techniques on MU configuration preferences. The comparison is shown in Figure 6.

In order to study the impact of shift operation, we define the shift ratio (θ) as the portion of shift requests in all requests. Note that we treat shift for multiple domains as multiple shift requests. The ratio is sensitive to data locality and mapping. We label the shift ratio in Figure 6 by lines, and the relationship between shift ratio and performance is obvious. Applications with high shift ratio generally prefer shift-optimized solutions. That is because applications with high shift ratio inevitably consume more time on shift. Because MU with small number of domain (N_D) is preferred by shift-optimized solution, applications with high shift ratio could perform better. And the RM-based on-chip memory could be better optimized for shift, if MU is organized with small N_D.

Comparing the performance before and after the "swap" technique, the circuit optimization preference changes from shift-optimized to read-optimized one. Without the architecture optimization "swap", the average shift ratio is 37.6%, and shift optimized solution achieves the best performance. But after the influence of "swap", the average shift ratio drops to 4.1%, and read optimized solution outperforms. This reason is architecture level optimization reduces the pending time caused by shift, and changes the RM circuit preference from MU-16-8-2 to MU-64-32-2. This case study shows the cross-layer interaction between RM parameters and system level applications.

VI. CONCLUSIONS

In summary, this paper presents a systematical and quantitative modeling of racetrack memory (RM) based on the Macro Unit (MU). We model the interaction between device parameters and MU structure design factors in details, and introduce the share degree to evaluate the RM performance. Based on NVsim, we perform a cross-layer optimization for area, latency and energy. The equivalent cell size could be $4.78F^2$, and the density is about 28 times of SRAM. Case study demonstrates a cross-layer interaction between RM cell parameters and system level applications.

VII. ACKNOWLEDGEMENTS

This paper is supported by NSF China (No.61202072), National High-tech R&D Program of China (No.2013AA013201), and NSF (CNS-1342566).

REFERENCES

[1] S. S. P. Parkin, M. Hayashi, and L. Thomas, "Magnetic domain-wall racetrack memory," *Science*, vol. 320, no. 5873, pp. 190–194, 2008.

[2] R. Venkatesan, V. Kozhikkottu, C. Augustine, A. Raychowdhury, K. Roy, and A. Raghunathan, "Tapecache: a high density, energy efficient cache based on domain wall memory," in *Proceedings of the 2012 ACM/IEEE international symposium on Low power electronics and design*. 2333707: ACM, pp. 185–190.

[3] R. Venkatesan, M. Sharad, K. Roy, and A. Raghunathan, "Dwm-tapestri - an energy efficient all-spin cache using domain wall shift based writes," in *Proceedings of the Conference on Design, Automation and Test in Europe*. 2485718: EDA Consortium, pp. 1825–1830.

[4] R. Venkatesan, S. Ramasubramanium, S. Venkataramani, K. Roy, and A. Raghunathan, "Stag: Spintronic-tape architecture for gpgpu cache hierarchies," in *International Symposium on Computer Architecture (ISCA)*, June 2014.

[5] A. J. Annunziata, M. Gaidis, L. Thomas, C. Chien, C.-C. Hung, P. Chevalier, E. O'Sullivan, J. Hummel, E. Joseph, Y. Zhu, T. Topuria, E. Delenia, P. Rice, S. Parkin, and W. Gallagher, "Racetrack memory cell array with integrated magnetic tunnel junction readout," in *Electron Devices Meeting (IEDM), 2011 IEEE International*, Dec 2011, pp. 24.3.1–24.3.4.

[6] Z. Sun, W. Wu, and H. Li, "Cross-layer racetrack memory design for ultra high density and low power consumption," in *Proceedings of the 50th Annual Design Automation Conference*. 2488799: ACM, pp. 1–6.

[7] X. Dong, C. Xu, N. Jouppi, and Y. Xie, "Nvsim: A circuit-level performance, energy, and area model for emerging non-volatile memory," *Emerging Memory Technologies*, pp. 15–50, 2014.

[8] 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 *HPCA*, 2011.

[9] Z. Sun, X. Bi, and H. Li, "Process variation aware data management for stt-ram cache design," in *Proceedings of the 2012 ACM/IEEE international symposium on Low power electronics and design*, ser. ISLPED '12. New York, NY, USA: ACM, 2012, pp. 179–184.

[10] P. Zhou, B. Zhao, J. Yang, and Y. Zhang, "Energy reduction for stt-ram using early write termination," in *Proc. IEEE/ACM Int. Conf. Computer-Aided Design - Digest of Technical Papers ICCAD 2009*, 2009, pp. 264–268.

[11] J. Li, P. Ndai, A. Goel, H. Liu, and K. Roy, "An alternate design paradigm for robust spin-torque transfer magnetic ram (stt mram) from circuit/architecture perspective," in *Proc. Asia and South Pacific Design Automation Conf. ASP-DAC 2009*, 2009, pp. 841–846.

[12] L. Thomas, S.-H. Yang, R. Kwang-Su, B. Hughes, C. Rettner, W. Ding-Shuo, T. Ching-Hsiang, S. Kuei-Hung, and S. S. P. Parkin, "Racetrack memory: A high-performance, low-cost, non-volatile memory based on magnetic domain walls," in *Electron Devices Meeting (IEDM), 2011 IEEE International*, pp. 24.2.1–24.2.4.

[13] S. Fukami, T. Suzuki, Y. Nakatani, N. Ishiwata, M. Yamanouchi, S. Ikeda, N. Kasai, and H. Ohno, "Current-induced domain wall motion in perpendicularly magnetized cofeb nanowire," *Applied Physics Letters*, vol. 98, no. 8, p. 082504, 2011.

[14] S. Fukami, T. Suzuki, K. Nagahara, N. Ohshima, Y. Ozaki, S. Saito, R. Nebashi, N. Sakimura, H. Honjo, K. Mori, C. Igarashi, S. Miura, N. Ishiwata, and T. Sugibayashi, "Low-current perpendicular domain wall motion cell for scalable high-speed mram," in *VLSI Technology, 2009 Symposium on*, pp. 230–231.

[15] Z. Sun, X. Bi, H. H. Li, W.-F. Wong, Z.-L. Ong, X. Zhu, and W. Wu, "Multi retention level stt-ram cache designs with a dynamic refresh scheme," in *Proceedings of the 44th Annual IEEE/ACM International Symposium on Microarchitecture*, ser. MICRO-44. New York, NY, USA: ACM, 2011, pp. 329–338. [Online]. Available: http://doi.acm.org/10.1145/2155620.2155659

[16] Y. Zhang, W. S. Zhao, D. Ravelosona, J.-O. Klein, J. V. Kim, and C. Chappert, "Perpendicular-magnetic-anisotropy cofeb racetrack memory," *Journal of Applied Physics*, vol. 111, no. 9, pp. –, 2012.

[17] W. S. Zhao, J. Duval, D. Ravelosona, J. O. Klein, J. V. Kim, and C. Chappert, "A compact model of domain wall propagation for logic and memory design," *Journal of Applied Physics*, vol. 109, no. 7, pp. 07D501–07D501–3, 2011.

[18] N. Binkert, B. Beckmann, G. Black, S. K. Reinhardt, A. Saidi, A. Basu, J. Hestness, D. R. Hower, T. Krishna, S. Sardashti, R. Sen, K. Sewell, M. Shoaib, N. Vaish, M. D. Hill, and D. A. Wood, "The gem5 simulator," *SIGARCH Comput. Archit. News*, vol. 39, no. 2, pp. 1–7, 2011.