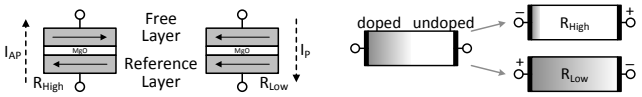
Architecting a Common-Source-Line Array for Bipolar Non-Volatile Memory Devices

Bo Zhao*, Jun Yang*, Youtao Zhang†, Yiran Chen*, Hai Li‡
*Dept. of ECE, University of Pittsburgh; †Dept. of CS, University of Pittsburgh; ‡Dept. of ECE, Polytechnic Inst. of NYU
*{boz6, juy9, yic52}@pitt.edu, †zhangyt@cs.pitt.edu, ‡hli@poly.edu

***Abstract*—Traditional array organization of bipolar non-volatile memories such as STT-MRAM and memristor utilizes two bitlines for cell manipulations. With technology scaling, such bitline pair will soon become the bottleneck of density improvement. In this paper we propose a novel common-source-line array architecture, which uses a shared source-line along the row, leaving only one bitline per column. We also elaborate our design flow towards a reliable common-source-line array design, and demonstrate its effectiveness on STT-MRAM and memristor memory arrays. Our study results show that with comparable latency and energy, the proposed common-source-line array can save 33% and 21.8% area for Memristor-RAM and STT-MRAM respectively, comparing with corresponding traditional dual-bitline array designs.**

I. Introduction

Due to scaling challenges in traditional memories, several new memory devices have recently emerged as promising candidates for next-generation memory technologies. These new devices present key advantages such as scalability, non-volatility, low leakage and resilience to errors over traditional DRAM or SRAM. However, each new device also exhibits unique characteristics that often require customized memory design in order to achieve high density, low latency, and high reliability. Among those new devices, there are a few that share a common feature in that changing the cell state from 0 to 1 requires a different current or voltage direction than from 1 to 0. For example, the Magnetic Tunnel Junction (MTJ) (Fig. 1(a)) used in Spin Torque Transfer Magnetic RAM (STT-MRAM) [4], the memristor [11] (Fig. 1(b)), the conductive bridging memory [7], and organic memory [8] are all such bipolar devices. In this paper we refer them as *bipolar non-volatile devices* (bi-NVD).

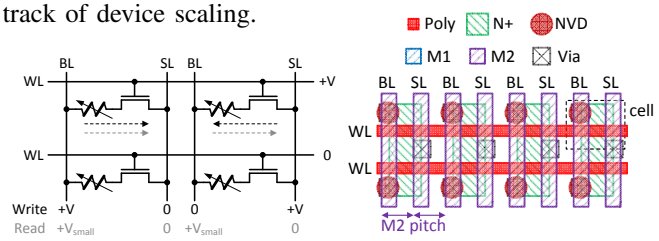


(a) MTJ (b) Memristor

Fig. 1. Conceptual view of MTJ and Memristor structures.

To provide bi-directional current to a bipolar device, a classic *dual-bitline* array structure utilizes bitline pairs to control voltages on two ends of a cell (Fig. 2(a)), similar to that of SRAM. However, with aggressive scaling of new memory devices, the wire width and spacing of bitline pair become the bottleneck to further shrinking memory area, diminishing the benefit of device scaling. In this paper we propose a novel common-source-line array architecture, in which one of the bitline pair is moved to rows, leaving only one bitline per column of cells. Therefore within the proposed array, cell size is again determined by the access device, similar to that in DRAM and PCM, leading density improvement back to the track of device scaling.

978-3-9810801-8-6/DATE12/ ©2012 EDAA



(a) Schematic (b) Layout

Fig. 2. Illustration of a dual-BL array.

In this paper, we describe our design flow for a reliable common-source-line architecture, and demonstrated the viability of common-source-line arrays using two upcoming devices, memristor and MTJ. Our results show that with comparable latency and energy, our common-source-line array can save 33% and 21.8% area for Memristor-RAM and STT-MRAM respectively, compared with corresponding dual-bitline arrays.

II. Proposed Common-Source-Line Array Design

In memory technologies requiring specially-processed memory device such as DRAM, PCM and MRAM, the memory device is stacked on top of its access transistor that is made as small (narrow) as possible to achieve high density. In such a case, the cell area of a dual-bitline array is usually wire pitch dominant. In other words, the transistor (diffusion) width plus spacing is smaller than two times the bitline (metal wire) pitch. This is illustrated in Fig. 2(b) which shows the layout of a group of eight cells. Here the bi-NVD is at metal 1 level, within the via/contact stack from diffusion to bitlines [12]. As can be seen in the figure, the cell width is determined by the pitch of bitline (BL) and source line (SL) [5], not the transistor width. For ease of fabrication and cost control, in most STT-MRAM prototypes [2], [4], [9], the MTJs are implemented in the top metal layer, after the formation of all metal layers. The dominance of wire pitch is even more pronounced in such designs as the pitch of higher metal levels are usually several times that of bottom metal levels.

A. Common-SL Layout

To reduce the area of such an array, we propose to turn the SLs 90° such that they span across all columns, as illustrated in Fig. 3 (with the cross-sectional view along its bitline). That is, all cells in a row share a single SL, eliminating the areas taken by N SLs previously, where N is number of columns.

Hence, cell width is narrowed down to the transistor (diffusion) width plus diffusion spacing, and the area of an array can be considerably reduced compared to a dual-BL array.

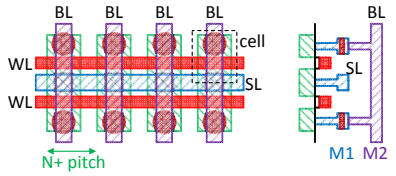
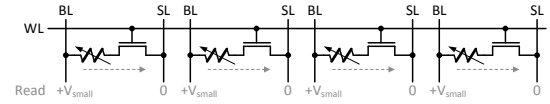


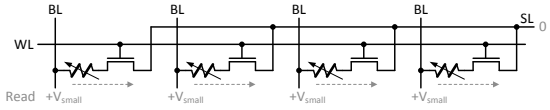
Fig. 3. Layout of common-SL array.

B. Read and Write

With the common-SL design, the memory accesses become different. Fig. 4 shows the schematic and read operation comparison between dual-BL and common-SL arrays. These two array designs essentially share the same read scheme.



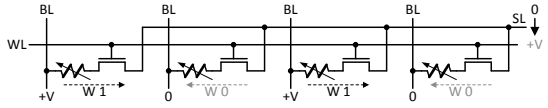
(a) Schematic of dual-BL array



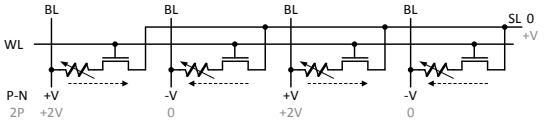
(b) Schematic of common-SL array

Fig. 4. Schematic comparison between dual-BL and common-SL arrays.

For writes, in the common-SL array, writing different cells in a row are no longer independent due to the shared SL. Hence, writing bit 1 and 0 must be performed in two separate rounds. As shown in Fig. 5(a), the bitlines voltage are first set according to the values to be written. Then the SL is set to 0 for writing bit 1, and in the next round to +V for writing bit 0. As we can see, this write scheme doubles the write latency due to the common-SL.



(a) Flip-V scheme



(b) P-N (or 2P) scheme

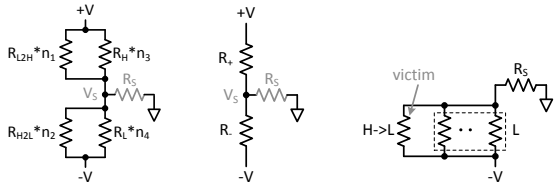
Fig. 5. Two write schemes in common-SL array.

Instead, we propose to concurrently write all cells in a row, achieving a write latency comparable to traditional dual-BL array. This is achieved by applying both +V and -V to corresponding BLs and 0 to SL, producing current/voltage in two directions simultaneously. This is illustrated in Fig. 5(b) as positive-negative voltage (P-N) scheme. Alternatively, one can also shift all voltages by V leading to an equivalent scheme with no negative voltages, shown as the $2\times$ positive voltage (2P) scheme in the figure. We will use this P-N scheme for better illustration in the remainder of this paper.

III. Design for Reliability

In the proposed common-SL array architecture, SLs are shared among cells in the same row, which are read and written simultaneously on each access. As discussed earlier, the read/write operations on individual cells are no longer independent, while such isolation is guaranteed in a dual-BL array. This is not a problem in reading a row because all the cells are exposed to the same voltage configuration. However, the write operation is more complicated. We now formulate the problem using a static model.

In a write operation, a cell's resistance experiences one of the four state changes: from high to low (H2L), from low to high (L2H), staying high (H), and staying low (L). For these four cases, we extracted the effective resistance of a cell, including both the bi-NVD and the access transistor. Therefore, writing a group of cells sharing a common SL can be generalized into an equivalent circuit as shown in Fig. 6(a), assuming a positive BL writes high resistance and a negative BL writes low resistance. The node in the middle represents the common SL and its resistance R_s. Here $n_1 \sim n_4$ are the number of cells in the four state changes respectively, and $N = n_1+n_2+n_3+n_4$, which is the number of cells sharing a common SL. It can be further simplified into the circuit shown in Fig. 6(b).



(a) (b) (c) Worst-case pattern

Fig. 6. Equivalent circuits of write operation.

Fig. 6(b) is essentially a voltage divider circuit. The voltage on the common SL, shown as V_s, is supposed to stay grounded to provide identical voltage drop on all cells. However, the imperfection of common SL and the global sources that drive it, represented by R_s, breaks such balance and introduces *voltage drift* on the SL node. Such drift places negative impact on write operations, especially on those cells with less voltage drops. For Memristor-RAM, the memristor state is a function of applied flux, which is the integral of voltage over time [3], reduced voltage implies increased latency for a full shift of memristor state. For STT-MRAM, reduced voltage may directly causes write failures as the induced current may not be larger than the switching current of MTJ. We now apply KCL to express V_s analytically:

$$V_s = \frac{V \cdot (R_- - R_+)}{R_+ + R_- + \dfrac{R_+ R_-}{R_s}} \tag{1}$$

From this expression and Fig. 6, we can derive that V_s is determined by the following parameters:

1) Old data stored in each cell and new data to be written
2) Number of cells, N, sharing a common SL
3) Driving capability of SL node, R_s

Parameter 1) decides the distribution of $n_1 \sim n_4$. Hence, it determines R_+ and R_-, together with parameter 2). However, 1) is governed by data patterns generated by applications, and thus is hard to control at design time. On the other hand, we do have control over both N and R_s. Therefore, to mitigate V_s drift, we will first find the worst-case data pattern, i.e. the pattern that leads to largest V_s drift with given N and R_s, and then find an array design with proper N and R_s that work reliably/robustly under such worst cases.

In the following parts of this section, we will use memristor as example to demonstrate the design flow, which is then extended to STT-MRAM for the results in Section IV. The parameters of these two devices used in our analytical models and circuit simulations are summarized in Table I and II.

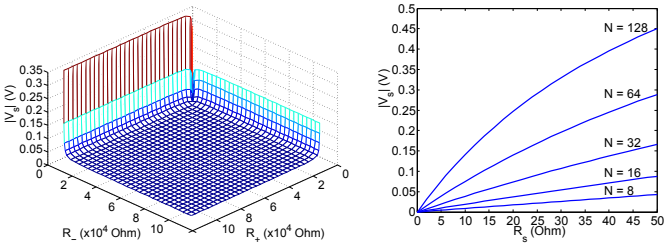
TABLE I
MEMRISTOR PARAMETERS [1], [3], [11]

High/Low R	100K/100 Ω
Carrier Mobility	10^{-7} m^2/V·s
Thickness	10 nm
Planar size	50×50 nm

TABLE II
STT MTJ PARAMETERS [4], [6]

R_L (P)	2KΩ
R_H (AP)	4KΩ
I_{P2AP}	70μA
I_{AP2P}	50μA

A. Mitigating V_s Drift



(a) V_s vs. R_+ and R_-, R_s=30Ω (b) V_s vs. N and R_s

Fig. 7. V_s drift for Memristor-RAM.

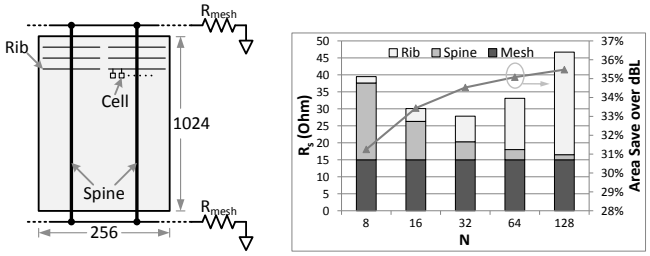
To find the worst-case data pattern, we plotted V_s drift in the entire range of R_+ and R_-, for N={8,16,32,64,128} with a constant R_s=30Ω. Fig. 7(a) shows for N=128 and all plots agree on the same tendency: the absolute value of V_s reaches its extremes when R_- reaches its minimum. The worst case happens when R_+ is at its maximum. The drift is relatively moderate with all other R_+s and R_-s.

The worst-case data pattern is n_1=n_3=0, n_2=1, n_4=N-1, as shown in Fig. 6(c). In this case, only one cell is altered and this cell is a victim of the parallel R_Ls that lower the effective resistance. We then use this worst-case data pattern and study V_s drift as a function of N and R_s, as plotted in Fig. 7(b). As we can see, $|V_s| \propto N$, $|V_s| \propto R_s$. Further, with large N and R_s, V_s drift can reach ∼450mV, which is prohibitive as the voltage on a victim cell is nearly halved from what it should be. We will show next on how to design the array for lowest R_s and best N to achieve high reliability.

B. The Array Design

R_s is the resistance between SL node and the "ideal" ground. It includes the resistance of on-chip mesh networks for ground/power supply, and we denote this resistance as R_{mesh}. Based on the chip size of [12] and the analysis in [10], we estimated the worst-case R_{mesh} to be ∼15Ω.

Other parts of R_s come from within an array. Fig. 8(a) depicts an example of a common-SL array design. It has 1024 rows and 256 columns of cells, same as the prototype in [12]. The *rib* wires in wordline direction are the common SLs of width N. The *spine* wires in bitline direction connect the ribs to ground (power) meshes outside the array. Here we show the configuration of N=128, 64 on each side of its spine. Hence, in a 256-column wide array, $\frac{256}{N}$=2 spines are needed.



(a) (b) R_s tradeoffs

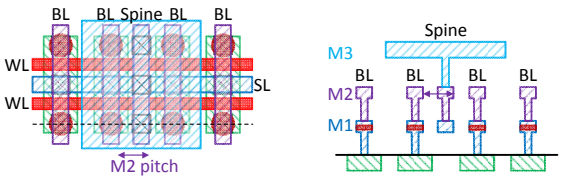
Fig. 8. An array example with N=128.

We also conservatively assume the resistance of a rib (common SL) is seen by all cells sharing it, regardless of their relative positions. Hence, we have

$$R_s = R_{mesh} + R_{spine} + R_{rib}$$

Early analysis from Fig. 7(b) shows that keeping both R_s and N small helps reducing V_s drift. While R_{mesh} has been determined by the global ground network, the tradeoff between R_{spine} and R_{rib} is dependent on N, which is studied in Fig. 8(b). Spines are constrained by total width of N cells and are thus wider at larger N. So R_{spine} decreases with increasing N. R_{rib} however increases linearly with N and thus becomes dominant at larger N. As spines are additional area overheads, layout techniques can be apply to make the area overhead per spine less than a Metal 2 pitch, while still maintaining low resistance, as shown in Fig. 9. The idea is to use narrowest wires to reach ribs on Metal 1, then back them up using wide Metal 3 wires that traverse over the cells. The area savings with different N, and thus different number of spines, are also studies as shown in Fig. 8(b). We see that even with the largest area overhead (N=8), the common-SL array still holds more than 30% of area reduction. Area estimations are based on generic 45nm design rules [13].

Combining the results in Fig. 8(b), we picked N=16 for a memristor common-SL array to achieves low R_s, good area reduction, and small V_s drift.

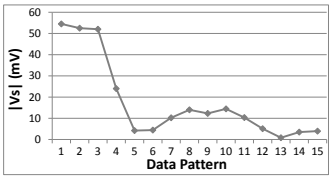


(a) Layout (b) Cross-Section along dashed line

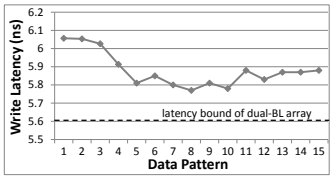
Fig. 9. Physical design of spines.

IV. EVALUATION RESULTS

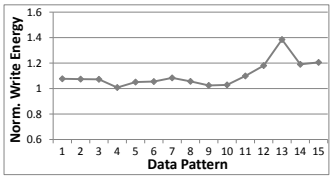
With 45nm PTM model [14] and device parameters given in Table I and II, we built subsets of common-SL and dual-BL



(a) V_s drift



(b) Write latency



(c) Write energy

Fig. 10. Simulation results of Memristor-RAM (N=16).

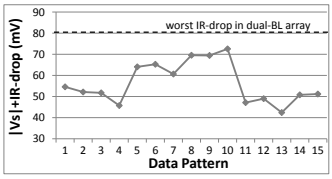


Fig. 11. V_s drift + IR-drop vs. data pattern in STT-MRAM (N=8).

arrays for both Memristor-RAM and STT-MRAM, and simulated them in HSPICE. The RC parasitics of all wire routings are properly modeled as well. We selected 15 representative data patterns for evaluation, including the most difficult ones that cause worst V_s drifts, and all other possibilities of cell state transition.

A. Memristor-RAM

Fig. 10(a) shows the absolute values of V_s drifts over all data patterns. Pattern 1$\sim$4 represents cases of relatively large V_s drifts, and all the rest patterns generate small V_s drifts. As a result of large V_s drift, victim cells take longer latency to be fully written, so pattern 1$\sim$4 are slower to write (Fig. 10(b)). Latencies of all patterns are slightly longer than dual-BL array latency, mainly due to the weaker access transistor to sustain higher voltage. Longer nominal write delay (6.1ns as upper bound) also increases cell energy, as shown in Fig. 10(c). This is especially evident for patterns 12$\sim$15. These data patterns have more H2L cells. They take small latencies on a write but drain large currents till the end of 6.1ns. Such cell energy increases will be neglectable when considering energies of peripheral circuits and I/Os all together.

To summarize, for Memristor-RAM design, the proposed common-SL array achieves similar delay and energy, while saving 33% area over a traditional dual-BL memory array.

B. STT-MRAM

For STT-RAM common-SL arrays, the V_s drift will likely to create write failure since the effective voltage on cells may be less than the threshold. However, if we can keep the drift within half of the natural IR-drop of a dual-BL array (denoted as $\frac{1}{2}$IR-drop), then we can gain the same reliability on writes. In a dual-BL array, the existence of resistive BL and SL introduces *IR-drops* into the write/read current path. Such IR-drops decrease the effective voltage applied on a cell, especially when it is physically far from its write driver. In contrast, the source line resistance is minimized by our spine+rib design in the proposed common-SL array, leaving only BL resistance in write/read current path, which effectively halves these IR-drops. Hence, if we can control the V_s drift such that it does not exceed $\frac{1}{2}$IR-drop, or, the total effective voltage drop does not exceed that in a dual-BL:

$$|V_s| + IR_{cSL} \leq IR_{dBL} \qquad (IR_{cSL} \approx \frac{1}{2} IR_{dBL})$$

then a common-SL array can guarantee the same write reliability as a dual-BL array

A common-SL STT-MRAM suffers worse V_s drift than a Memristor-RAM. This is because in addition to extreme imbalances, in STT-MRAM, R_+ and R_- are usually smaller due to the lower device resistances, resulting smaller denominator in Equation 1. For lower V_s drift, we choose N=8 in the array design. We can see from Fig. 11 that V_s drift together with IR-drop falls below the IR-drop in dual-BL array, demonstrating the effectiveness of our common-SL design.

As long as the reliability is guaranteed, a common-SL array of STT-MRAM enjoys similar latency and energy to its dual-BL counterpart. Its potential for area reduction is smaller compared with Memristor-RAM because its cell transistors are made wider to deliver a current larger than the 70μA switching current (Table II). After accounting for all the overheads, the proposed common-SL array architecture on a STT-MRAM achieves a 21.8% area reduction over dual-BL array.

V. Conclusion

Traditional dual-bitline array structure significantly limits area density of bipolar non-volatile memories such as STT-MRAM and memristor. To liberate the scaling potentials of these memory technologies, in this paper we proposed a novel common-source-line array architecture, which effectively eliminates the constraints from array designs. We then elaborated our design flow towards reliable common-source-line arrays for both Memristor-RAM and STT-MRAM. Our results show that with comparable latency and energy, our common-source-line array can save 33% and 21.8% area for Memristor-RAM and STT-MRAM respectively, comparing to corresponding dual-bitline arrays.

References

[1] J. Borghetti, et al., "Electrical transport and thermometry of electroformed titanium dioxide memristive switches", *JAP*, 2009.
[2] D. Halupka, et al., "Negative-Resistance Read and Write Schemes for STT-MRAM in 0.13um CMOS", *ISSCC*, 2010.
[3] Y. Ho, G. Huang, P. Li, "Nonvolatile Memristor Memory: Device Characteristics and Design Implications", *ICCAD*, 2009.
[4] M. Hosomi, et al., "A Novel Nonvolatile Memory with Spin Torque Transfer Magnetization Switching: Spin-RAM", *IEDM*, 2005.
[5] T. Kawahara, et al., "2Mb SPRAM with Bit-by-Bit Bidirectional Current Write and Paralilelizing-Direction Current Read", *ISSCC*, 2007.
[6] T. Kishi, et al., "Lower-current and Fast switching of A Perpendicular TMR for High Speed and High density STT MRAM", *IEDM*, 2008.
[7] M. Kund, et al., "Conductive Bridging RAM (CBRAM): An Emerging Non-Volatile Memory Technology Scalable to sub 20nm", *IEDM*, 2005.
[8] G. Muller, et al., "Status and Outlook of Emerging Nonvolatile Memory Technologies", *IEDM*, 2004.
[9] R. Nebashi, et al., "A 90nm 12ns 32Mb 2T1MTJ MRAM", *ISSCC*, 2009.
[10] K. Shakeri, J. Meindl, "Compact physical IR-drop models for chip/package co-design of GSI", *IEEE TED*, 2005.
[11] D. Strukov, et al., "The missing memristor found", *Nature*, 2008.
[12] K. Tsuchida, et al., "A 64Mb MRAM with Clamped-Reference and Adequate-Reference Schemes", *ISSCC*, 2010.
[13] Cadence, 45nm Generic PDK data sheet and device models.
[14] ASU, Predictive Technology Model, http://www.eas.asu.edu/$\sim$ptm/