

Content-aware Encoding for Improving Energy Efficiency in Multi-Level Cell Resistive Random Access Memory

Hadi Hajimiri, Prabhat Mishra
CISE, University of Florida
{hadi, prabhat}@cise.ufl.edu

Swarup Bhunia
EECS, Case Western Reserve University
sxb21@case.edu

Branden Long, Yibo Li, Rashmi Jha
EECS, University of Toledo
rashmi.jha@utoledo.edu

Abstract— *Memory is an integral and important component of both general-purpose and embedded systems. It is widely acknowledged that energy of the memory structure is a major contributor in overall system energy. Recent advances with emerging non-volatile memory (NVM) technologies can potentially alleviate the issue of memory leakage power. However, they introduce new challenges and opportunities for dynamic power management in memory. In this paper, we consider resistive random access memory (RRAM), a promising NVM technology, and observe that a specific feature of the memory, namely, its multi-level cell (MLC) structure, can be used to significantly reduce its read access energy. Unlike conventional CMOS static random access memory (SRAM), the read access energy in RRAM largely depend on the stored content. Based on this observation, we present an efficient encoding technique for improving the energy efficiency for multi-level cell RRAM. Our simulation results with benchmark applications demonstrate an order-of-magnitude energy reduction with modest area overhead.*

I. INTRODUCTION

Power consumption has emerged as a primary design constraint for both general-purpose and embedded systems. The active power in an integrated circuit (IC) comprises of switching (dynamic) power and active leakage in logic and memory circuits. Reduction of active power has emerged as the primary design goal for the IC manufacturers and system designers to address the battery life issue in portable systems and to mitigate the temperature induced reliability concerns. Memory plays an important role in system energy, due to integration of increasingly larger memory closer to the processor in the memory hierarchy and faster memory clock. Hence, there is a critical need to significantly reduce active energy in memory. Memory active power has two components: 1) active leakage (which is typically much higher than standby power due to higher junction temperature in active mode); and 2) the read/write access energy. In order to reduce memory energy significantly, one needs to address both the leakage and dynamic energy during read/write operations.

With CMOS technology rapidly approaching the end of its roadmap [1], future computing systems are likely to be built with emerging non-volatile memory (NVM) technologies, such as resistive random access memory (RRAM) [7][8], spin torque transfer RAM (STTRAM) [11]-[13], phase change memory (PCM) [10], which show promising density, read/write performance and endurance. Among these technologies, RRAM has appeared highly promising primarily due to the benefits in terms of its non-volatility, integration density, read/write endurance, manufacturability, and access performance/energy. In recent times, various research efforts

have focused on RRAM fabrication, device modeling and optimization, and circuit and system level performance/energy analysis. However, circuit/architecture level design approaches for minimizing memory energy for these devices have not been adequately explored. In this paper, we study a specific characteristic of RRAM, namely its multi level cell (MLC) structure that enables a cell to store more than one bit information and is beneficial in greatly improving memory density. Next, we propose an architecture-level energy optimization approach that exploits the MLC nature of the RRAM cells to drastically reduce the read access energy. This is in addition to the fact that the non-volatile nature of RRAM cells virtually eliminates the memory core leakage. This makes RRAM as an attractive choice for implementing the memory arrays in a processor memory hierarchy.

Based on read/write process of RRAM memory cells, we exploit the intrinsically asymmetric nature of most NVM cells in order to improve the energy efficiency while maintaining or improving access performance and integration density. In particular, we observe that MLC structures, which lead to large improvement in density, provide largely varying access energy. From the RRAM read operation we find that a smaller current flows in the circuit corresponding to read “00” than read “11”. This is because the resistance of state “00” is higher than state “11”. A more resistive state with an identical voltage pulse will result in a lower instantaneous power (V^2/R) and therefore a lower read energy. Our analysis with respect to read access energy in a resistive crossbar shows that reading “00” and “01” from MLC comes at several orders of magnitude less energy than reading “10” and “11” as shown in Fig. 1. In order to exploit the nature of MLC for improving energy-efficiency, we propose an efficient encoding technique based on bit flips. Our experimental results demonstrate an order-of-magnitude reduction in memory access energy.

The rest of the paper is organized as follows. Section II presents an overview and contributions of our proposed approach. We have surveyed the related work in Section III. In Section IV, we model energy consumption of RRAM cells. Section V describes our encoding-based energy optimization technique. Section VI presents the experimental results for a set of benchmark applications. Section VII concludes the paper.

II. OVERVIEW AND CONTRIBUTIONS

Fig. 2 shows the overall approach, which integrates memory characterization and power modeling with content-aware encoding and subsequent system-level energy estimation steps. We study the effectiveness of the proposed encoding for MLC

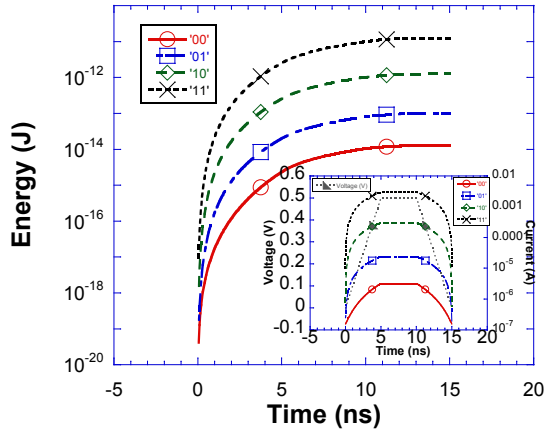


Fig. 1: Memory read access energy for RRAM MLC (2 bits per cell) for four different states with a read-voltage pulse duration of 15ns. Read energy is calculated by first obtaining the resistance of the device in different states from experimental data (shown in Fig. 5 (a)) after programming the device in different states. These resistance values were then used along with the read-voltage pulse of amplitude 0.5 V, and width 15ns to get the instantaneous read currents, shown in the inset. The instantaneous read power was calculated and integrated along the pulse-width and plotted as read-energy vs. time over the entire 15ns pulse-width.

RRAM based main memory, although such an approach can be used in other levels of memory hierarchy. As a by-product, RRAM array also helps to mitigate the memory leakage issue. Unlike alternative volatile memory technologies such as SRAM and DRAM, which require constant connection to VDD to retain stored content, RRAM reduces the core leakage power to virtually zero. This is consistent with other resistive NVM technologies [17]. The leakage power of the read/write circuitry remains comparable to that of a volatile memory. However, the leakage for a conventional SRAM-based memory array is typically dominated by the core. Hence, elimination of core leakage results in large saving in memory leakage. Since RRAM does not incur any leakage overhead, reduction of read energy consumption directly translates to overall energy efficiency for application binaries.

In a processor-based system, a large part of the memory dynamic energy is typically dissipated in reading data. In particular, for instructions of a program, there is no writing once the program page is fetched to main memory from the last level of memory. In our simulations with a set of media benchmarks, we observed 88.7% read operation compared to 11.3% write operations in L2 cache. Hence, a system-level design approach that aims at minimizing the read access energy of embedded memory can translate to large saving in total energy. Based on the skewed content-dependent read energy of RRAM devices, we infer that if the system is biased for reading more “00” than “11”s, we can have considerable saving in total dynamic energy with RRAMs.

This analysis provides the motivation to encode information (both instruction and data) before storing them to memory in a way that maximizes the “00” and “01” content in order to dramatically improve the access energy. Interestingly, increasing

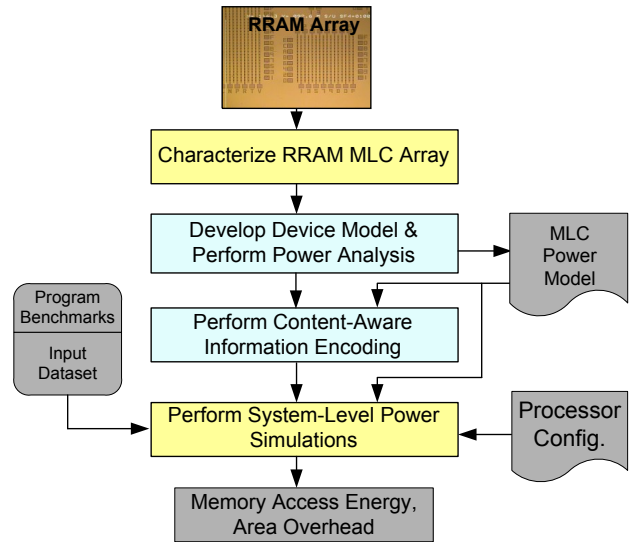


Fig. 2: Overall flow for the proposed approach that leverages on intrinsic asymmetry of resistive memory.

“00” and “01” counts is also expected to have large positive impact in memory reliability. This is because storing a reduced range of values improves the reliability of a multi-level cell, since variation-induced degradation in dynamic range (due to reduced I_{ON}/I_{OFF}) is not likely to cause failures in the cells storing “00” and “01”. We believe it would affect the read, write and data retention reliability of the memory cells. In order to exploit the nature of MLC in reliability and energy-efficiency simultaneously, we propose an efficient encoding technique based on bit flips at varying level of granularity. It can drastically reduce read access energy in RRAM array.

In particular, the paper makes following key contributions:

- 1) It presents a study of read access energy of resistive memory which shows a content-dependent variation in access energy due to corresponding variations in resistive states. It models the access energy behavior for MLC RRAM cells for representative cell design.
- 2) Exploiting the skewed access energy pattern, it presents a low-cost content-dependent information encoding approach that aims at maximizing the memory access energy saving. To the best of our knowledge, this is the first effort that aims at improving memory energy efficiency exploiting intrinsic asymmetry of MLC cells, which provide different read access energy depending on the stored content.

III. RELATED WORK

Encoding techniques are widely used to achieve energy, area, reliability, and performance optimizations. Xie et al. [2] introduced a compression technique capable of compressing flexible instruction formats in VLIW architectures. Seong et al. [3] presented bitmask-based compression that improves dictionary-based compression using bitmasks. Mirhoseini et al. [4] developed a novel coding method to minimize Phase Change Memory (PCM) write energy. It minimizes the energy required for memory rewrites by utilizing the differences between PCM read, set, and reset energies. S. Cho and Lee

[9] proposed Flip-N-Write to replace a PRAM write operation with a more efficient read-modify-write operation. These techniques only consider write energy for single level cell PRAM memory. In this work we present a content-aware encoding technique that optimizes the read energy (program and read-only data) utilizing the differences in energy of reading different values in multilevel RRAM cells (MLC). The existing write-energy minimization techniques are complementary to our approach, and can be used together to reduce both read and write energy of memory in case of diverse datasets.

IV. RRAM POWER ANALYSIS

After an initial electroforming step by DC voltage sweep, the device was reset to high-resistance state (HRS) the state denoted by 00. Fig. 3 (a) shows that low-resistance state (LRS) denoted by 11 can be achieved by setting the device using a compliance current of 5 mA and a different LRS denoted by 10 can be achieved by setting the device using a compliance current of 1 mA from an initial state of 00. Fig 3 (b) shows that, HRS 00 and 01 could be achieved by resetting the device from state 11 using reset voltages of -4.0 V and -2.6 V, respectively. The compliance currents were much higher than the desired values ($< 100\mu\text{A}$) due to larger dimensions of these devices.

Based on the above results, Fig. 4 summarizes the write strategies for storing multiple bits in these devices by switching the device to multiple resistance states. Multiple LRS can be achieved using different compliance currents while multiple HRS can be achieved using different reset voltages. All states are not interchangeable. For example, a device in the state 01 can be brought to 00 simply by applying a higher reset voltage. However, to toggle a device from state 00 to 01, one has to first set the device to one of the LRS values and then apply an appropriate reset voltage. Similarly, a device in state 10 can be switched to 11 by simply setting it again with a higher compliance current. However, a device in state 11 can be toggled to a state 10 only by first resetting it to an HRS (e.g. 00) and then setting again using an appropriate compliance current.

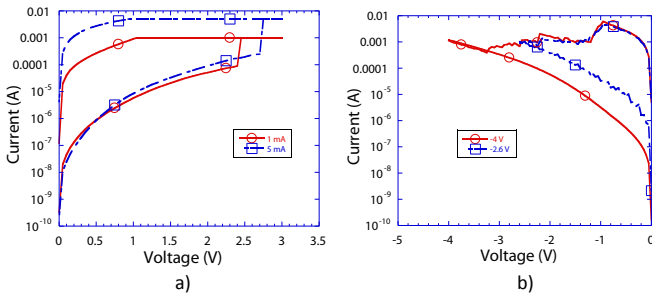


Fig. 3: (a) States 11 and 10 achieved by setting the device from state 00 using compliance currents of 5 mA and 1 mA, respectively, (b) State 00 and 01 can be achieved by resetting the device from state 11 using reset voltages of -4.0 V and -2.6 V, respectively.

Fig. 5 (a) shows DC I-V characteristic of devices put into four distinct states namely 00, 01, 10, and 11. Clearly, distinct resistance of the device can be observed corresponding to

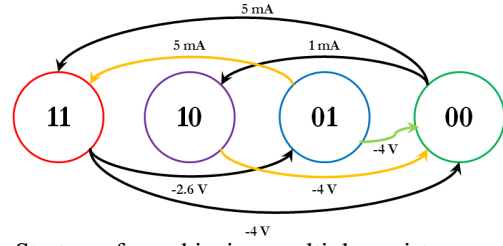


Fig. 4: Strategy for achieving multiple resistance states and 2-bit MLC in RRAM devices.

each state which demonstrates the potential for achieving MLC using RRAM devices. Fig. 5(b) shows resistance values extracted at -0.5 V corresponding to each of these states. If same read pulse is used to read these states then it is intuitive that HRS (00 and 01) will demonstrate lower read energy than LRS (10 and 11). This result was simulated using a trapezoidal read voltage pulse of 0.5 V amplitude and 5 ns width with rise and fall time of 5 ns. The total read energy integrated during the read pulse duration is shown in Fig. 1. Interestingly, read energy for 11 state was almost 3 orders of magnitude higher than 00 state. Therefore, for memory read intensive computations, storing data in different HRS values of devices will be beneficial to achieve low-power operations.

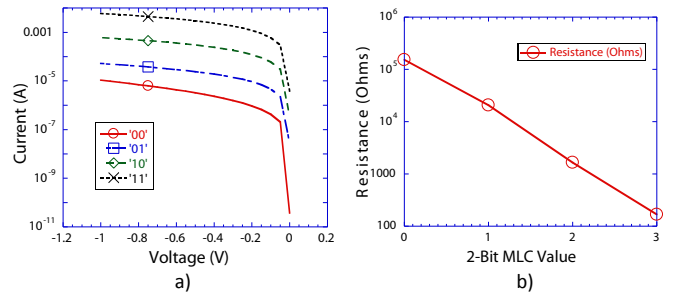


Fig. 5: (a) DC I-V characteristic of the device in different states, (b) resistance in different states extracted at -0.5 V. Clearly, distinct resistance states can be observed.

In addition to saving energy using this approach, we believe that storing data in HRS values of the device will also improve reliability and considerably reduce the failure rate due to cycle-to-cycle variability in RRAM devices. This is because to achieve multiple LRS values in RRAM devices, the compliance current needs to be carefully controlled. The LRS value of RRAM device is very sensitive to the compliance current used to set the device [14]. A lower compliance current leads to the formation of smaller dimension filaments resulting in a higher resistance in LRS while a higher compliance current leads to the formation of bigger filament resulting in a lower resistance in LRS of the device [14]. Different compliance currents can be achieved by connecting a transistor with RRAM device and applying an appropriate gate voltage to limit the maximum current through the RRAM device, shown in Fig. 6. However, it is well-known that the charging of parasitic capacitor at transistor-RRAM junction when RRAM switches from HRS to LRS leads to an overshoot of current over the compliance current [6][15], shown in Fig. 6. This can cause uncontrolled filament dimensions and variability in LRS.

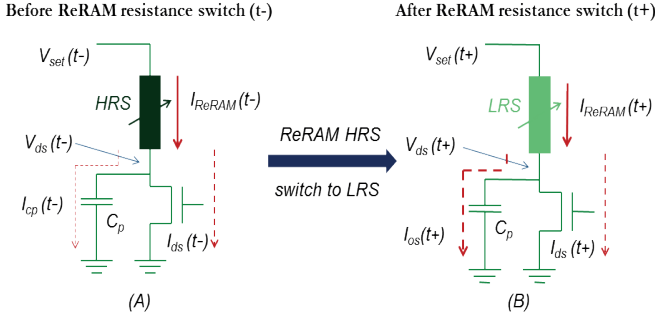


Fig. 6: Schematic diagram for illustrating the origin of current overshoot effect by charging of parasitic capacitor at transistor-RRAM junction when device resistance state switches from HRS (A) to LRS (B).

Fig. 7 shows overshoot current ($I_{overshoot}$) vs. the compliance current (I_{comp}) simulated for different parasitic capacitances (C_p) at transistor-RRAM node. Clearly, the overshoot current increases with an increase in the compliance current for a given parasitic capacitance. This indicates that the variability in the dimension of the filament will be much larger for devices set with higher compliance current than lower compliance currents. Therefore, the devices in the lowest resistance states (i.e. 11 in this case) will cause highest variability. This variability can manifest itself as variability in the resistance value corresponding to the state 11 between different devices and cycles or when devices are reset to an HRS from state 11. Therefore, from device standpoint, to minimize the failure during operation due to cycle-to-cycle and device-to-device variability, it is desired to set the device using a low compliance current and achieve multiple states by resetting the device with different reset voltages. This indicates that storing data in HRS values of devices (i.e. 00 and 01 in this case) will not only provide energy saving but also has potential for improved reliability.

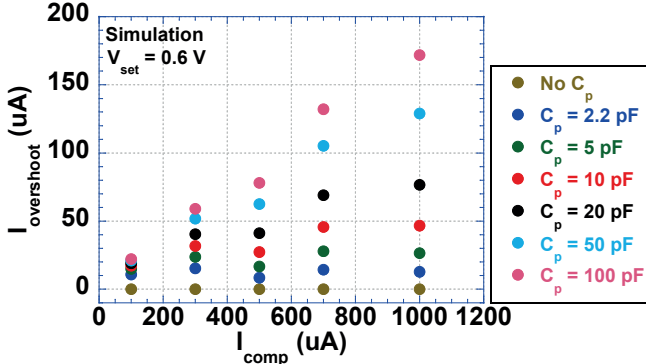


Fig. 7: Current overshoot ($I_{overshoot}$) over the compliance current (I_{comp}) due to parasitic capacitance (C_p) vs. I_{comp} . This indicates that higher I_{comp} leads an increased $I_{overshoot}$ for a particular C_p .

Table I shows read energy comparison for RRAM devices if operated as Single Level Cell (SLC) versus MLC. SLC refers to the condition where RRAM stores just two states either 0 or 1. In this case, 0 can correspond to the highest resistance state (i.e. 00 of MLC) while 1 can correspond to the lowest resistance state (i.e. 11 of MLC) while skipping the intermediate states. Therefore, for SLC in this scenario,

TABLE I: Simulated read energy comparison of single cell RRAM versus Multilevel RRAM cell.

Two-Bit Value	Total Read Energy for Two-RRAM (J)	Multilevel RRAM Cell Read Energy (J)
'00'	2.54E-14	1.2702e-14
'01'	1.25E-11	9.9770e-14
'10'	1.25E-11	1.2577e-12
'11'	2.49E-11	1.2438e-11

the read energies for the state '0' is 1.2702E-14 J and state '1' is 1.2438E-11 J for 0.5V, 15 ns read pulse, as evident in Fig.1. The second column in the table represents the read energy for two-bits (i.e. 00, 01, 10, 11) in SLC RRAM devices. This energy was obtained by adding the read energies for two separate states for various combinations. For example, the read energy for two bits containing '10' would be the sum of the read energy for bit '1' and bit '0' (i.e. 1.2438E-11+1.2702E-14 = 1.25E-11 J) and so on. The third column is the read energy for the MLC approach that simply uses the final energy readings from Fig.1. for 0.5V, 15ns read-pulse to show the read energies for all four states if intermediate states of RRAM is utilized for MLC. It can be observed that for all 2-bit patterns using a multilevel RRAM cell is more energy efficient and suggests energy savings even if no encoding is used. In the following section we propose an encoding technique to drastically reduce the read energy exploiting the low-energy characteristic of multilevel RRAM cell.

V. CONTENT-AWARE CODING FOR ENERGY EFFICIENCY

To take advantage of the aforementioned characteristics of RRAM we propose a novel encoding of the program binary to improve the overall energy consumption. The program coding and decoding flow is illustrated in Fig. 8 where the encoding is done offline (prior to execution) and the encoded program is loaded into the memory. The decoding is done during the program execution (online). As shown in the figure, the decoder is placed between memory and processor cache. Let's assume program binary is composed of N-bit words where N can be 2, 4, 8, 16, 32, 64, or 128. Every pair of bits in the program code can be stored in one memory cell. For example, a 4-bit word can be stored in two memory cells. The key idea is to flip all bits of a word if the number of '11' and '10' patterns stored in memory cells for that word is greater than the number of '01 and '00's¹. Recall that read energy consumption of a cell storing '11' is three orders-of-magnitude higher than a cell storing '00' (about 100 times of reading '01' and 10 times of reading '10'). An additional memory cell (flip bit) is added for each word to indicate whether the word has been flipped or not.

Algorithm 1 presents major steps of our encoding algorithm. It slices the input program into N-bit words. Each word in the input program translates into a (2+N)-bit word in the output program (extra 2 bits as the flip indicator)².

¹In our experiments we use actual energy numbers from Table I to decide whether it is beneficial to flip all bits.

²One bit is enough to indicate whether a word is flipped. Since we consider MLCs of 2 bits, the flip indicator requires two bits (one cell) – 00 to indicate original word and 01 to indicate flipped word.

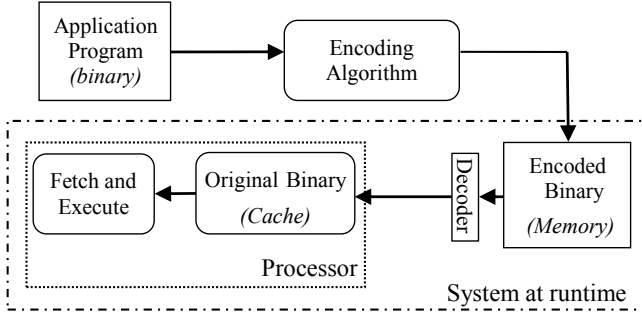


Fig. 8: Overview of the proposed content-aware information encoding scheme.

Algorithm 1: Content-aware encoding algorithm

```

1 Inputs: N: word size, P: program binary
2 Output: EP: encoded program binary
3
4 Initialize EP as a binary stream
5 Array W = slice P into  $\lceil \text{size}(P)/N \rceil$  N-bit words
6 for each word in W do
7   flipped_word = flip all bits in word
8   e = energy required for reading the word
9   e' = energy required for reading the flipped_word
10  if e' < e then
11    new_word = 01:flipped_word // concatenate
12    01 with flipped word
13  else
14    new_word = 00:word
15  end
16  Append new_word to EP
17 end
18 return EP

```

Fig. 9 a) shows an example of transforming an 8-bit word with its corresponding flip bit. Here the flip bit is 01 indicating that it is beneficial to flip all bits in the word to reduce the number of 11 patterns and thus reducing the energy consumption of storing this word in memory. The flipped words are flipped again by the processor at retrieval time. In this example, area overhead of 25% is introduced for an 8-bit word code transformation (4 cells to 5 cells). The word is flipped only if the overall energy consumption of the flipped word is lower than the input word. However, as it can be seen in the figure, flipping every bit in the word can also transform a 00 pattern into 11. By dividing the 8-bit word into two 4-bit words and encoding each word individually we gain more control over how the bits are flipped. Fig. 9 b) shows encoding of the same 8-bit data where we chose the word size to be 4 bits. Although 4-bit encoding of the same data can reduce amount of unwanted bit flips, it also increases the number of additional flip bits, increasing the area overhead.

VI. EXPERIMENTS

A. Experimental Setup

In order to evaluate the effectiveness of our approach we used SimpleScalar cycle accurate detailed microarchitectural

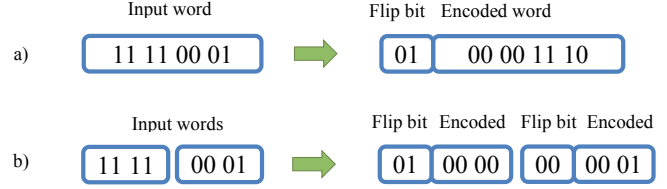


Fig. 9: Examples of encoding with bit flips: a) word size=8, b) word size=4

simulator compiled for Alpha instruction set [16]. The memory hierarchy was composed of separate level one instruction and data caches and main memory. We used a 4KB cache with line size of 32 bytes and associativity of 2 for both instruction and data caches. Cache hit latency was set to 1 cycle. Memory latency was set to 18 and 2 cycles (first chunk and remaining chunks) and memory access bus width was 8 bytes. We selected applications from MiBench and MediaBench embedded benchmark suites with their default inputs obtained from the benchmark suites. The encoded programs were placed in memory and new program addresses was calculated based on the word size. Area overhead was calculated by counting the increase in program size due to addition of flip bits. We fed gathered memory read statistics into our memory power model to obtain the energy consumption.

B. Energy Efficiency

Fig. 10 shows energy consumptions of this encoding for several word sizes using MLC RRAM. This energy includes the extra energy consumption accounted for the larger program and area due to encoding. The energy consumption is normalized to the energy of using single cell RRAM. It can be observed that reducing word size increases energy savings but it also increases area overhead. Energy savings of up to 88% (85% on average) is achieved if area overhead of 25% is allowed. In other words, our approach can provide an order of magnitude reduction in energy with modest area overhead. In the extreme case where 100% area overhead is acceptable for the system, 99% (two orders of magnitude) energy savings on average can be achieved.

C. Performance Overhead

Fig. 11 illustrates the performance overhead of using our encoding technique³. Several embedded applications were used from MiBench, MediaBench embedded benchmarks and Spec2000 application suite. The performance overhead for many benchmarks (g721_enc, adpcm_dec, adpcm_enc, bitcnt, crc32, dijkstra, and patricia) is less than 0.2% even when 2-bit word size is selected. Less than 1% performance overhead is observed for all benchmarks with 4-bit and larger word sizes. Selecting 2-bit word size results in the highest performance overhead (1.6% using epic benchmark). Clearly, this performance overhead is negligible especially since we are able to achieve two orders of magnitude energy improvement.

³Performance overhead is defined as: (execution time of the encoded program / execution time of the original program) x 100 - 100

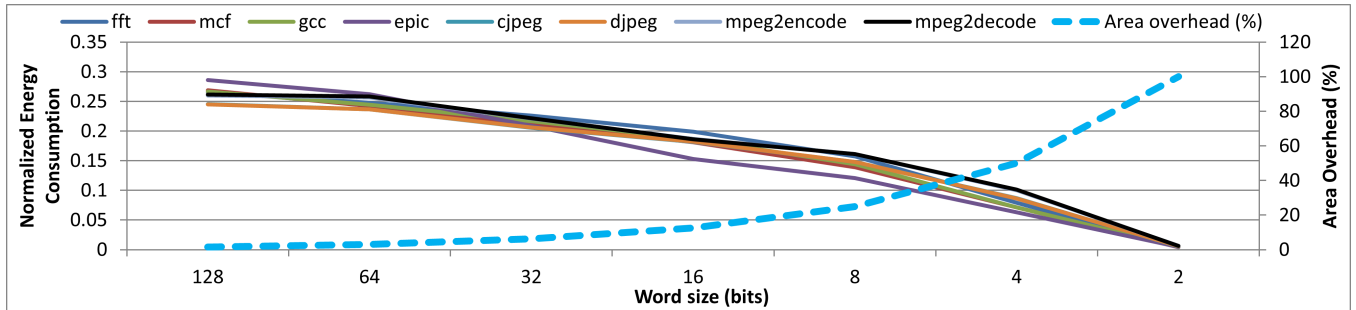


Fig. 10: Energy consumption and area overhead for various word sizes for different benchmarks normalized to single cell RRAM.

VII. CONCLUSION

We have presented a novel system-level design approach that minimizes dynamic energy of memory through content-aware encoding. It considers resistive memory, an emerging NVM technology that shows promising density, access performance and endurance. We exploit the fact that read energy of a 2-bit MLC for ‘11’ state is three orders of magnitude higher than ‘00’ state due to corresponding difference in read current, which depends on the resistance for a state (i.e. stored value). We presented an efficient encoding scheme based on bit flips exploiting the differences of read energy of various MLC states. The encoding scheme can be applied at various level of granularity which trades off energy savings with area overhead. Our experimental results demonstrated that the proposed encoding technique can provide an order-of-magnitude energy savings with modest area overhead. Although, we have considered main memory as a case study, the approach can be easily extended to other levels of memory in a typical processor memory hierarchy. Moreover, the approach is scalable across higher-density MLC memory, which can store 3 or more bits per cell.

While our study is based on a specific resistive memory, the approach applies to all variants of resistive memory technologies. Furthermore, it easily extends to other emerging NVM technologies, such as spin torque transfer RAM (STT-RAM), which also exhibit MLC feature. STT-RAM cells are also known to exhibit content-dependent read access energy due to variation in cell resistance for storing ‘1’ or ‘0’ (for a SLC) [18] and similarly for multi-level states (for MLC).

REFERENCES

- [1] International Technology Roadmap for Semiconductors (ITRS), <http://www.itrs.net>
- [2] Y. Xie, W. Wolf, and H. Lekatsas, “A Code Decompression Architecture for VLIW Processors”, *MICRO*, 2001.
- [3] S. Seong, P. Mishra, “Bitmask-Based Code Compression for Embedded Systems”, *IEEE Trans. CAD*, 2008.

- [4] Mirhoseini, A. Potkonjak, M. ; Koushanfar, F. , “Coding-based energy minimization for Phase Change Memory”, *Design Automation Conference (DAC)* , 2012.
- [5] H. Y. Lee et al. , “Low Power and High Speed Bipolar Switching with A Thin Reactive Ti Buffer Layer in Robust HfO₂ Based RRAM”, *IEEE International Electron Devices Meeting (IEDM)* , 2008.
- [6] B. Long, Y. Li, R. Jha , “Switching Characteristics of Ru/HfO₂/TiO₂-x/Ru RRAM Devices for Digital and Analog Non-Volatile Memory Applications”, *IEEE Electron Device Letters* , Vol. 33, No.5, 2012.
- [7] Pierre Fazan, Global Semiconductor Alliance, “Future RAM Emerging Memory Technologies and Their Applications”, <http://www.gsaglobal.org/events/2010/0316/docs/7.GMC-PierreFazan.pdf>
- [8] Greg Atwood, Micron Technologies, “Current and Emerging Memory Technology Landscape”, [Online] <http://www.micron.com/>
- [9] S. Cho and H. Lee, “Flip-N-Write: a simple deterministic technique to improve PRAM write performance, energy and endurance”, *MICRO* , pages 347357, 2009.
- [10] A. Pirovano et al, “Reliability study of phase-change nonvolatile memories”, *IEEE Transactions on Device and Materials Reliability*, Sept. 2004, vol 4, issue 3, pp. 422427.
- [11] M. Hosomi et al, “A Novel Nonvolatile Memory with Spin Torque Transfer Magnetization Switching: Spin-RAM”, *IEDM Tech*, 2006.
- [12] S. Salahuddin et al, “Self-Consistent Simulation of Hybrid Spintronic Devices”, *IEDM Tech*, pp.1-4, Dec., 2006.
- [13] A.D. Smith et al, “STT-RAM A New Spin on Universal Memory”, *IEEE Transactions on Device and Materials Reliability*, Future Fab Intl. Vol. 23, July 2007.
- [14] B. Long, Y. Li, S. Mandal, R. Jha, and K. Leedy, “Switching dynamics and charge transport studies of resistive random access memory devices”, *Applied Physics Letters*, Vol.101, Issue 11, Sept. 12, 2012
- [15] D. C. Gilmer, G. Bersuker, et al., “Effects of RRAM Stack Configuration on Forming Voltage and Current Overshoot”, *3rd IEEE International Memory Workshop*, pp.1-4, 2011.
- [16] D. Burger, T. Austin, S. Bennet, “Evaluating future microprocessors: the simpliscalar toolset”, *Report CS-TR-1308*, University of Wisconsin-Madison. Computer Science Department Technical (July 2000).
- [17] X. Guo, E. Ipek, and T. Soyata, “Resistive Computation: Avoiding the Power Wall with Low-Leakage, STT-MRAM based computing”, *ISCA 2010*.
- [18] S. Paul, S. Chatterjee, S. Mukhopadhyay and S. Bhunia, “Nanoscale Reconfigurable Computing Using Non-Volatile 2-D STTRAM Array”, *IEEE Nano*, 2009.
- [19] X. Dong and Y. Xie, “AdaMS: Adaptive MLC/SLC phase-change memory design for file storage”, *ASP-DAC*, 2011.

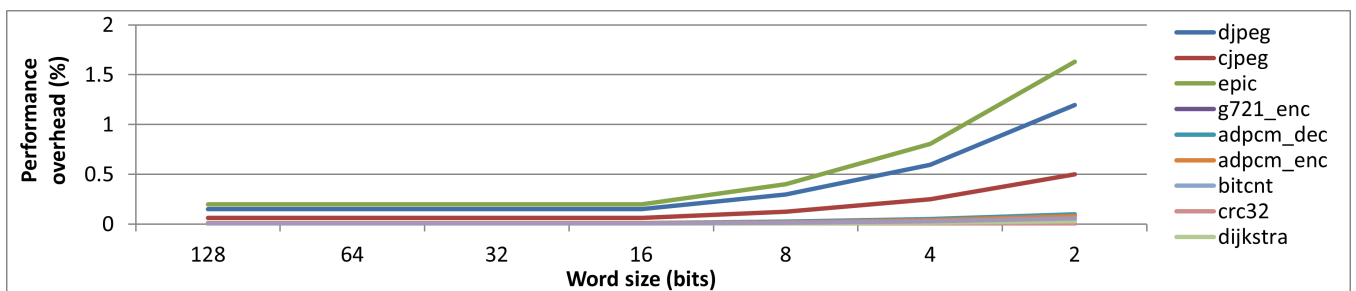


Fig. 11: Performance overhead of using content-aware encoding for various word sizes in case of different benchmarks.