Proactive useless data flush architecture for nonvolatile SRAM using magnetic tunnel junctions

Daiki Kitagata¹(a), Shuu’ichirou Yamamoto¹, and Satoshi Sugahara¹

Abstract A store energy and latency reduction architecture based on proactive useless data flush (PUDF) is proposed for nonvolatile SRAM (NV-SRAM) using magnetic tunnel junctions (MTJs). Prior to the store operation to the MTJs in the array, the PUDF architecture predicts store-unneeded blocks having useless data and shuts down these blocks in advance. As a result, the store energy and latency can be reduced depending on the proportion of store-unneeded blocks, which enhances the energy reduction efficiency of power gating (PG) using the NV-SRAM. The energy and latency characteristics are computationally analysed and experimentally verified using circuit parameters extracted from fabricated test-element-group (TEG) circuits of the NV-SRAM.

Keywords: CMOS logic systems, power gating, SRAM, nonvolatile SRAM, magnetic tunnel junction

Classification: Integrated circuits

1. Introduction

Thermally activated leakage currents of transistors cause the fundamentally unavoidable standby power problem for CMOS logic systems such as microprocessors and SoCs, which degrades/restricts the energy efficiency of these logic systems [1]. In particular, standby power of the constituent caches account for much proportion in that of logic systems, and thus its reduction is considerably indispensable. Application of emerging nonvolatile memory such as spin-transfer torque magnetoresistive RAM (STT-MRAM) to caches can achieve extremely low standby power owing to its nonvolatile retention ability [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]. However, its high store energy and long store latency restrict the application to the last-level cache or main memory. Power gating (PG) architectures employing a dedicated nonvolatile SRAM (NV-SRAM) using magnetic tunnel junctions (MTJs) can avoid the store energy and latency problems, since the store operation to the MTJs is done only before shut-down period [16, 17]. Namely, nonvolatile retention using the MTJs is not used during the normal SRAM operation mode, which can lead to excellent circuit performance comparable to conventional 6T- SRAMs (The detail is described in the next section). Therefore, this type of NV-SRAM can be applied to higher level caches.

However, the high store energy prolongs the break-even time (BET) that represents the minimum shutdown duration of PG, i.e., the temporal granularity of PG is restricted. To overcome the store energy problem, several architectures, e.g., early write termination, efficient data coding for reducing store bits, and so on, have been proposed for caches using STT-MRAM [18, 19, 20, 21, 22, 23, 24, 25, 26, 27]. These architectures can be also applied to NV-SRAM caches. Recently, simple store skipping (SSS) and hierarchical store-free (HSF) [17] architectures are also proposed, which are more suitable for NV-SRAM caches. In these architectures, the store operation to the MTJs is executed when data already-stored in the MTJs are not identical with those in the bistable circuits (6T cell parts) connected to them in the cell array. In this case, the store operation is carried out even for cells having useless data, as far as data inconsistency between the additional unneeded latency.

In this study, we propose a new store energy and latency reduction architecture based on proactive flush of useless data for the NV-SRAM using MTJs, which is analogous to cache line replacement techniques used in ordinary cache systems. PG performances of the proposed proactive useless data flush (PUDF) architecture are investigated.

2. Proposed architecture

Fig. 1 shows the circuit configuration of the NV-SRAM cell [17]. The cell is comprised of a conventional 6T cell and MTJs connected to its storage nodes through pass transistors. In the normal SRAM operation mode, the NV-SRAM cell can act as a high performance 6T cell, since the 6T cell part is electrically separated from the MTJs by shutting off M1 and M2. At the beginning of PG execution, data in the 6T cell part are stored in its corresponding MTJs before entering the shutdown state (hereafter, this operation mode is referred to as store operation mode). The two virtual supply voltages (VVDS’s) for the normal SRAM operation and sleep (low-voltage retention without the usage of MTJs) modes are generated by two power switches PS1 and PS2. The sleep mode is used for the store-waiting cells during the store operation mode [17]. Fig. 2 shows the system organization of the PUDF architecture. The NV-SRAM has a capacity of 32 kB, 256 kB, or 2 MB that are suitable for first-, second-,
and third-level caches, respectively. The cell array consists of 256 bit × 256 bit (= 8 kB) subarrays and each subarray is divided into 1 kB (64 bit × 128 bit) blocks. The power management unit (PMU) controls the power supply of these blocks using power switches.

The PUDF architecture is based on the prior shut-down of blocks having useless data that are predicted in similar manner to cache line replacement techniques. In conventional cache systems, all the data memorized in them need not be used again, i.e., there exist useless data in caches (although data coherency is required between the intended and its lower-level caches/memories). Data that have a high possibility for being useless can be predicted by several techniques such as FIFO (first in, first out), LRU (least recently used), and other replacement architectures [28]. In the proposed PUDF architecture, the store-unneeded blocks predicted to have useless data are memorized as useless-data flags (UDFs) in the register file in the PMU, in which these store-unneeded blocks can be judged by a technique analogous to FIFO, LRU techniques and so on used in cache line replacement architectures. For instance, the control unit (see Fig. 2) keeps access logs of each block during the normal SRAM operation mode based on the LRU architecture. At the beginning of the shutdown mode, the unit judges whether the blocks have useless data or not using the logs, in which a threshold to evaluate frequency in use of data would be employed for the judgement. Then, the control unit sets the UDFs to the register file in the PMU on the basis of the judgement. After that, the store-unneeded blocks are shut down in advance in accordance with the UDFs before executing the store operation to the MTJs. Then, data in the cells of the residual blocks are sequentially stored in their MTJs and the store-completed blocks are shut down in a sequential order. Therefore, the PUDF architecture can reduce the excess energy consumption with the additional latency with respect to the store operation for useless data. Note that the register file for the UDFs is reset when the shutdown mode is exited. Also note that the PUDF architecture can be used with the HSF architecture [17]. In the HSF architecture, store-unneeded blocks are memorized as store-free flags, in which these store-unneeded blocks are encoded by the presence or absence of write access during the normal SRAM operation mode. By the combination, the store operation can be skipped for the blocks that the 6T cell parts of the constituent cells have the same data as the corresponding MTJs and for the blocks having useless data even when each 6T cell part has different data from the corresponding MTJs.

For cache applications, the data and tag arrays are configured with the NV-SRAMs. In this case, data (e.g., index addresses, valid bits, etc.) in tag array blocks for the useless data blocks in the data array can be also flushed, i.e., store-free shutdown based on the PUDF and HSF architectures can be executed for the tag blocks. The valid bits for these tag blocks are reset after exiting the shutdown mode.

3. Performance evaluation

Energy and latency characteristics of the PUDF architecture for the NV-SRAM are analysed using HSPICE, in which a low-power transistor model for 65 nm CMOS process is used [29]. The design of the cell is described in detail elsewhere [17]. Energy performance of the NV-SRAM is analysed from BET that is an energy performance index of PG. BET is given by the period when the excess energy ($\Delta E_{L,NL}$) due to leakage currents during the normal SRAM operation mode and the energies ($\Delta E_{STR}$ and $\Delta E_{RST}$) required for PG execution (data store to the MTJs, shutdown, wake-up, and data recovery from the MTJs) are compensated by the saved energy ($\Delta P_{save} : BET$) due to the shutdown [17, 30]. The BET is given by

$$BET = (\Delta E_{L,NL} + \Delta E_{STR} + \Delta E_{RST})/\Delta P_{save}. \quad (1)$$

The BET is appraised compared the leakage and store/restore energies of the NV-SRAM with the leakage energy of a reference 6T-SRAM [17]. Note that, in this study, the transistor sizes of the equivalent 6T-SRAM cell are designed to be the same as those of the 6T cell part in the NV-SRAM cell. (1) can be also rearranged by

$$BET = BET_{SR} + BET_{NL} = BET_{SR} + \eta T_{NL}, \quad (2)$$

where $\eta$ denotes the ratio of the leakage energy to the store energy.
in which $\text{BET}_{\text{SR}}$ and $\text{BET}_{\text{NL}}$ are the BET components predominated by the store/restore energies and the excess leakage energy during the normal SRAM operation mode, respectively, and $\eta$ is the leakage-related factor and $C_{\text{NL}}$ is the duration for the normal SRAM operation mode. Hereafter, the $\text{BET}_{\text{SR}}$ component in (2) is used for the following energy performance evaluation, which can represent the energy characteristics of the PUDF architecture.

Store latency $\tau_{\text{STR}}$ is also an important index for PG, since it also restricts the granularity of PG. Here, this quantity is simply estimated by sum of the store operation duration for cell array blocks having store-requested MTJs, and thus the quantity gives the minimum store latency. Note that the store operation is executed for every word segment (64 bit) in all the store-requested blocks, in which the duration for the 1-word store operation is 20 ns [17].

Red curves in Figs. 3(a) and 3(b) show $\text{BET}_{\text{SR}}$ and $\tau_{\text{STR}}$, respectively, for the 32 kB, 256 kB, and 2 MB NV-SRAMs as a function of useless-data proportion $R_{\text{PUDF}}$. Using the PUDF architecture, both $\text{BET}_{\text{SR}}$ and $\tau_{\text{STR}}$ can be effectively reduced with increasing useless-data proportion. The blue curves in Figs. 3(a) and 3(b) show the results when the HSF architecture is simultaneously employed with the PUDF architecture. $\text{BET}_{\text{SR}}$ and $\tau_{\text{STR}}$ are further diminished depending on store-free proportion $R_{\text{HSF}}$ of the HSF architecture. Namely, the reduction of $\text{BET}_{\text{SR}}$ and $\tau_{\text{STR}}$ is enhanced by combining the PUDF architecture with the HSF architecture. Fig. 4 shows the effect of the PUDF architecture on $\text{BET}_{\text{SR}}$ for the 32 kB, 256 kB, and 2 MB NV-SRAM arrays with the peripherals, in which the HSF architecture are also additionally examined and a condition of $R_{\text{PUDF}} = 50\%$ and $R_{\text{HSF}} = 50\%$ is assumed. The computationally simulated results and the experimentally analyzed results using circuit parameters extracted from a fabricated NV-SRAM TEG [17] are shown in the figure. The NV-SRAM TEG and analysis procedure were described in detail in Ref. [17]. $\text{BET}_{\text{SR}}$ can be effectively reduced using the PUDF architecture and its reduction ability can
be enhanced by the combination with the HSF architecture. These BETSR reductions analyzed using the measured parameters almost coincide with the simulation results. The simulation and experimental results are also identical except $R_{PUDF} = 50\%$ and $t_{HSF} = 50\%$ (i.e., regardless of $R_{PUDF}$ and $t_{HSF}$ values). For the 32 KB NV-SRAM, the reduction of BETSR is proportional to the total store-skipped block proportion, since the store energy dominates BETSR. As the array size increases, the reduction rate of BETSR is much enhanced. This is because the contribution of the leakage energy of the store-waiting blocks to BETSR increases with increasing array size (although the above-described sleep mode is employed for the store-waiting blocks), and the leakage energy can be effectively reduced by the prior shutdown of the store-unneeded blocks. For practical cache applications, setting of $R_{PUDF}$ and $t_{HSF}$ is an important issue, since these quantities depend on not only cache size but also workload/application. Several benchmarks would be required to estimate these quantities. Finally, we would like to emphasize that the standby power of this type of NV-SRAM can be extremely lowered by PG (more than 99% reduction) [17] and the PG granularity determined by BET can be finet by the proposed architecture.

4. Conclusion

We propose the PUDF architecture for store energy and latency reduction for NV-SRAM using MTJs. The architecture predicts store-unneeded blocks having useless data and shuts down these blocks prior to the store operation to the MTJs. Thus, the architecture can effectively reduce the store energy and latency of the NV-SRAM, leading to highly energy-efficient PG for logic systems using it. This architecture can be also applied to STT-MRAM caches.

Acknowledgments

This work was supported by VLSI design and education center, VDEC, the University of Tokyo in collaboration with Renesas Electronics Corp. The authors would like to thank Dr. Y. Shuto, Imaging Science and Engineering Laboratory (ISEL), Tokyo Institute of Technology.

References

[1] S. Jain, et al.: “A 280 mV-to-1.2 V wide-operating-range IA-32 processor in 32 nm CMOS,” ISSCC Dig. Tech. Papers (2012) 66 (DOI: 10.1109/ISSCC.2012.6176932).
[2] X. Dong, et al.: “Circuit and microarchitecture evaluation of 3D stacking magnetic RAM (MRAM) as a universal memory replacement,” ACM/IEEE Design Autom. Conf. (2008) 554.
[3] X. Wu, et al.: “Power and performance of read-write aware hybrid caches with non-volatile memories,” Conf. on Design, Autom. and Test in Europe. European Design and Autom. Association (2009) 737 (DOI: 10.1109/DATE.2009.5909762).
[4] X. Guo, et al.: “Resitive computation: Avoiding the power wall with low-leakage, STT-MRAM based computing,” ACM SIGARCH Comput. Archit. News 38 (2010) 371 (DOI: 10.1145/1816038.1816012).
[5] M. Rasquinha, et al.: “An energy efficient cache design using spin torque transfer (STT) RAM,” ACM/IEEE ISLPED (2010) (DOI: 10.1145/1840845.1840931).
[6] N. Goswami, et al.: “Power-performance co-optimization of throughput-oriented core architectures using spin memory,” IEEE HPCA (2013) 342 (DOI: 10.1109/HPCA.2013.6522331).
[7] M. Komalan, et al.: “Feasibility exploration of NVM based L-cache through MSHR enhancements,” DATE (2014) 1 (DOI: 10.7873/DATE.2014.034).
[8] K.-W. Kwon, et al.: “AWARE (Asymmetric Write Architecture with REdundant blocks): A high write speed STT-MRAM cache architecture,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst. 22 (2014) 712 (DOI: 10.1109/VLSI.2013.2255694).
[9] S. Senni, et al.: “Exploration of magnetic RAM based memory hierarchy for multicore architecture,” IEEE Computer Society Annual Symp. on VLSI (2014) 248 (DOI: 10.1109/ISVLSI.2014.29).
[10] F. Oboril, et al.: “Evaluation of hybrid memory technologies using SOT-MRAM for on-chip cache hierarchy,” IEEE Trans. Comput.-Aided Design Integr. Circuits Syst. 34 (2015) 367 (DOI: 10.1109/TCAD.2015.2391254).
[11] M. R. Jokar, et al.: “Sequioa: A high-endurance NVM-based cache architecture,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst. 24 (2015) 954 (DOI: 10.1109/VLSI.2015.2429054).
[12] F. Oboril, et al.: “Normally-Off STT-MRAM cache with zero-byte compression for energy efficient last-level caches,” IEEE ISLPED (2016) 236 (DOI: 10.1145/2934583.2934629).
[13] S. Senni, et al.: “Exploring MRAM technologies for energy efficient systems-on-chip,” IEEE J. Emerg. Sel. Topics Circuits Syst. 6 (2016) 279 (DOI: 10.1109/JETCAS.2016.2547680).
[14] J. G. Alzate, et al.: “2 MB array-level demonstration of STT-MRAM process and performance towards L4 cache applications,” IEEE IEDM (2019) 2.4.
[15] G. Hu, et al.: “Spin-transfer torque MRAM with reliable 2 ns writing for last level cache applications,” IEEE IEDM (2019) 2.5.
[16] Y. Shuto, et al.: “Nonvolatile static random access memory based on spin-transistor architecture,” J. Appl. Phys. 105 (2009) 07C933 (DOI: 10.1063/1.3076895).
[17] D. Kitagata, et al.: “Design and energy-efficient architectures for non-volatile static random access memory using magnetic tunnel junctions,” Jpn. J. Appl. Phys. 58 (2019) SBBB12 (DOI: 10.7567/1347-4065/ab0095).
[18] P. Zhou, et al.: “Energy reduction for STT-RAM using early write termination,” IEEE ICCAD (2009) 264.
[19] A. Nigam, et al.: “Delivering on the promise of universal memory for spin-transfer torque RAM (STT-RAM),” IEEE ISLPED (2011) 121 (DOI: 10.1109/ISLPED.2011.5993623).
[20] S. Yazdanshenas, et al.: “Coding last level STT-RAM cache for high endurance and low power,” IEEE Comput. Archit. Lett. 13 (2013) 73 (DOI: 10.1109/L-CA.2013.8).
[21] C. W. Smullen, et al.: “Relaxing non-volatility for fast and energy-efficient STT-RAM caches,” IEEE HPCA (2011) 50.
[22] A. Jog, et al.: “Cache revive: Architecting volatile STT-RAM caches for enhanced performance in CMPs,” ACM/IEEE Design Automation Conference (2012) 243 (DOI: 10.1145/2228360.2228406).
[23] M. H. Samavatian, et al.: “An efficient STT-MRAM last level cache architecture for GPUs,” ACM/IEEE Design Automation Conference (2014) 1.
[24] A. Jadidi, et al.: “High-endurance and performance-efficient design of hybrid cache architectures through adaptive line replacement,” Proc. IEEE/ACM ISLPED (2011) 79 (DOI: 10.1109/ISLPED.2011.5993611).
[25] J. Ahn and K. Choi: “Lower-bits cache for low power STT-RAM caches,” IEEE ISCAS (2012) 480 (DOI: 10.1109/ISCAS.2012.6272069).
[26] Z. Wang, et al.: “Adaptive placement and migration policy for an STT-RAM-based hybrid cache,” IEEE HPCA (2014) 13 (DOI: 10.1109/HPCA.2014.6359933).
[27] W. Cheng, et al.: “Architecture and data migration methodology for L1 cache design with hybrid SRAM and volatile STT-RAM configuration,” Microprocess. Microsyst. 42 (2016) 191 (DOI: 10.1109/HPCA.2014.6359933).
[28] D. A. Patterson and J. L. Hennessy: Computer Organization and Design: The Hardware/Software Interface (Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2013) fifth edition.

[29] VLSI Design and Education Center (VDEC), University of Tokyo, http://www.vdec.u-tokyo.ac.jp/English/index.html.

[30] S. Yamamoto, et al.: “Nonvolatile delay flip-flop using spin-transistor architecture with spin transfer torque MTJs for power-gating systems,” Electron. Lett. 47 (2011) 1027 (DOI: 10.1049/el.2011.1807).