International Journal of Internet, Broadcasting and Communication Vol.10 No.4 6-11 (2018)
http://dx.doi.org/10.7236/IJIBC.2018.10.4.6

IJIBC 18-4-2

Technology of the next generation low power memory system

Doosan Cho

EE, Sunchon National University, Korea
E-mail: dscho@scnu.ac.kr

Abstract

As embedded memory technology evolves, the traditional Static Random Access Memory (SRAM) technology has reached the end of development. For deepening the manufacturing process technology, the next generation memory technology is highly required because of the exponentially increasing leakage current of SRAM. Non-volatile memories such as STT-MRAM (Spin Torque Transfer Magnetic Random Access Memory), PCM (Phase Change Memory) are good candidates for replacing SRAM technology in embedded memory systems. They have many advanced characteristics in the perspective of power consumption, leakage power, size (density) and latency. Nonetheless, nonvolatile memories have two major problems that hinder their use it the next-generation memory. First, the lifetime of the nonvolatile memory cell is limited by the number of write operations. Next, the write operation consumes more latency and power than the same size of the read operation.These disadvantages can be solved using the compiler. The disadvantage of non-volatile memory is in write operations. Therefore, when the compiler decides the layout of the data, it is solved by optimizing the write operation to allocate a lot of data to the SRAM. This study provides insights into how these compiler and architectural designs can be developed.

Keywords: *memory design, power consumption, cache memory, memory system, memory hierarchy, architecture*

1. Introduction

In current DRAM technology, fabrication technology is no longer possible in the area of 30nm or less. DRAM is the most widely used main memory. Until now, deepening of DRAM technology has progressed and it has been developed at the ratio of 100 times / 10 years for the past 30 years. In developing of the processor, it is used together until now, but the processor is facing commercialization of 6nm process and can no longer do with DRAM. DRAM development has stopped because of the underlying technology that stores one bit of data in a pair of capacitors and transistors. The read of data is based on the detection of the potential difference of the capacitor because the size of the capacitor can not maintain the minimum required filament of 20fF (femto Farad) at 30nm. At the same time, the leakage current also increases exponentially with deepening technology. The solution technology of this problem is nonvolatile memory which does not

Manuscript Received: Aug. 7, 2018 / Revised: Aug. 11, 2018 / Accepted: Aug. 18, 2018

Corresponding Author: dscho@scnu.ac.kr
Tel: +81-61-750-3577, Fax: +81-61-750-3570
EE, Sunchon National University, Korea

need capacitor. The main goal of this technology is to solve the power consumption / leakage power simultaneously due to deepening technology.

Static Random Access Memory (SRAM) also occupies the largest area of today's microprocessors (50% to 90%) and is the most important factor in leakage power in system chips. Since battery-powered handheld devices are the market for most system chips, memory leakage power issues must be addressed, and the severity of this problem is increasing exponentially as semiconductor fabrication technology becomes finer.

Over the past few decades, the scaling of conventional CMOS technology has evolved due to the need for high integration density and performance. Static power consumption is an important parameter in nanoscale integrated circuit design because the subthreshold current exponentially depends on the threshold voltage. Embedded memory now occupies a major portion of the chip area in the microprocessor, so minimizing leakage power is becoming an increasingly important design factor. Table 1 shows leakage of each SRAM / STT-RAM at 32nm fabrication process, room temperature (23 °C), cache line size 64B, and 1MB cache.

Table 1. Comparison both memory parameters

Associativity	2	4	8	16
SRAM [mW]	96.9	100.4	102.3	105.2
STT-RAM [mW]	1.1	1.5	1.5	2.2

Low leakage power and high-density spin torque transfer memory (STT-RAM) are ideal candidates to replace conventional SRAM / DRAM. However, STT-RAM is difficult to use directly with low-level cache or main memory because of its high write latency and large energy consumption. Several studies have been conducted to improve the write energy and performance of STT-RAM, including write buffers, early write termination (EWT) techniques, multi-retention time caches, and hybrid read / write centralized caches. In most studies, write latency is balanced with write energy or cache durability, which is still insufficient to completely replace existing technologies. It is necessary to develop a hybrid type memory technology that can complement this.

2. Hybrid cache architecture

In order to develop a successful hybrid memory, a design methodology must be developed that determines, for example, cache size / line size / associativity, etc., in the case of cache memory. First, the benchmark programs in the target application are profiled and the appropriate memory parameters are selected based on the results. In the case of hybrid memory, the optimal memory parameters (size / number of lines / associativity) can be determined by also prioritizing the segmentation of the STT-RAM and the data to be placed in the SRAM cache during the profiling phase. Such a partitioning mechanism can be implemented by hardware or a compiler. We plan to proceed with both approaches in this study.

First, we design a hybrid cache capable of data encoding by using low-cost peripheral circuit (STT-RAM uses CMOS transistors for peripheral circuit) to improve cache energy, standby time and durability simultaneously. The data aware hybrid cache selects data suitable for the STT-RAM and the SRAM cache based on the ratio of the data bit value '1', and stores the data in a partitioning manner. These partitioning functions can be implemented in hardware or in a compiler. We primarily designed a hardware accelerator to implement this as a compiler and speed it up. The data partitioning allows the STT-RAM cache to store data with most '0's and most of the' 1 'bits in the SRAM cache. When 0 is written to the STT-RAM cell, the energy required for writing is 3.5 to 6.5 times smaller than that of writing '1'. Therefore, a higher '0' ratio of STT-RAM improves cache performance and energy efficiency. For efficiency of the SRAM, we will use an asymmetric 5T-SRAM cell, which can store '1' data at very low power. Writing a large number of zero data in a cache containing mostly zeros improves the number of write operations, write energy and durability of the STT-RAM. These tasks can be achieved by developing compiler data layout optimization techniques. Therefore, development of data placement optimizing compiler technology is also included in the center of this study.

To this end, we first developed compiler-driven memory system design automation tools to develop optimal next-generation memory systems. As shown in Figure 1, if the target application is passed to the compiler tool (compiler / analyzer / optimizer / assembler / profiler) as an input, the compiler generates code for the default system and passes it to the simulator. The developer optimizes the memory system parameters by feeding back the simulation results (system chip performance - energy consumption and performance per module). At this time, the memory parameters are transferred to the design automation tool using a hardware description language developed by the self-developed hardware, and the compiler and the simulator reflecting the memory parameters are automatically synthesized and the above process is repeated. The optimization process is repeated to complete the design when the desired system is attained. The advantage of the design automation tool we are implementing is that it is compiler-centric, so we can analyze the execution results in a timely manner when changing memory system components.

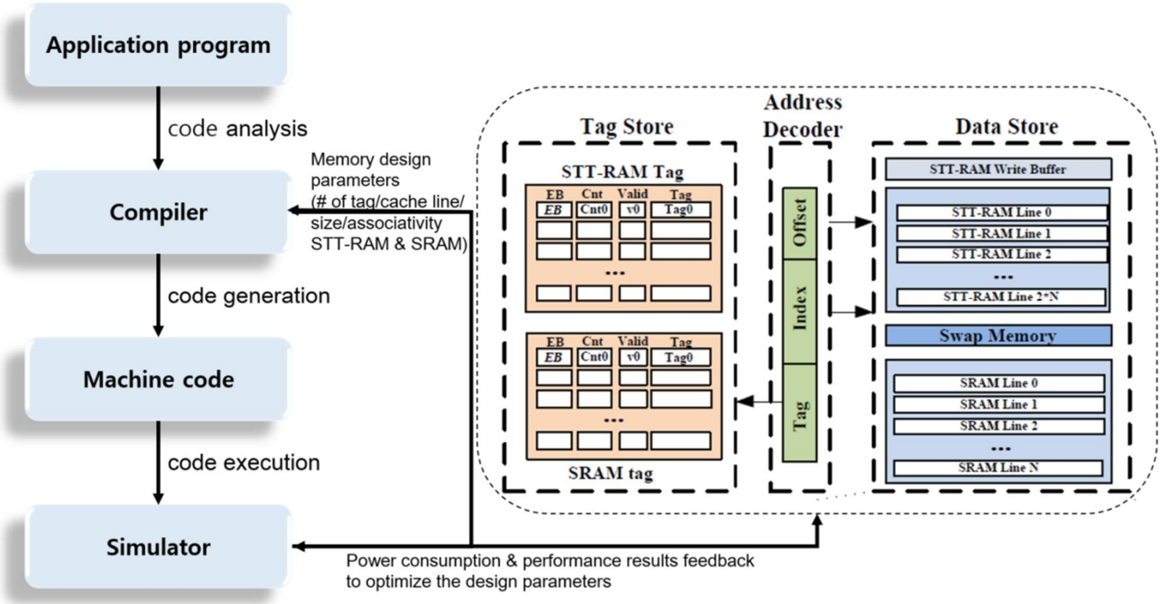

Figure 1. Overview of the proposed work

Currently, there are hardware description languages (HDL) such as Verilog, VHDL, and SystemC, which are widely used programming languages for representing various hardware application specific integrated circuits (ASICs). The problem with these languages is that it requires a lot of effort and time from the developer to fully describe a design and verify its performance because it requires detailed description to the hardware netlist level sub-structure at the time of design. And the implementation typically takes from weeks to months. Another problem is that hardware described in HDL must use tools such as simulators and compilers to verify its performance requirements. The simulator has EDA (Electronic Design Automation) tools that are automatically generated from HDL, but since there is no compiler, the software is manually created at the machine code level in a typical development environment. Therefore, it takes much time for development / verification. A design automation tool that provides a compiler has been introduced recently, but requires separate development effort or a license of several hundred million dollars.

The advantage of configuring a hardware design tool centered on a compiler is that it can be quickly verified by automating handwork and feedbacks design optimization information to developers to support development. The main purpose of the traditional compiler was to convert the program to machine code so that the best performance can be achieved when the given hardware is used. In this code conversion process, the compiler analyzes the memory usage pattern of the target application by using static analysis (control flow / data flow analysis) and dynamic analysis (dynamic analysis: run-time profiling) Based on the information, optimized cache hierarchy, tag storage configuration parameters, and data storage configuration parameters can be optimally determined in perspective of energy consumption / performance.

Considering that the hybrid memory system has a very complicated structure and a very large design space, it is necessary to repeatedly search a number of design changes and verifications in order to design an architecture optimized for a specific application. In searching memory design space, the development time is also consuming a lot. Using the proposed compiler centric design tool can shorten and accelerate this design process.

The development of Internet / Big Data / Artificial Intelligence is promoting the wave of portable, low power electronic devices. As hardware structures continue to change in order to meet various application fields and rapidly changing needs, there is a need for the ability to design / verify new features frequently required by the compiler. The compiler-centric design tool we design and implement has the following three features.

- Analyze memory access patterns of target application through compiler's static analysis (control flow / data flow analysis) and dynamic analysis (dynamic analysis: run-time profiling), and generate memory design parameters

- An optimizing compilation technique that generates optimized memory system parameters and optimized application code using various analysis techniques of the compiler.

- Interactive design support technology that effectively executes the memory system parameters and application code generated by the compiler to feed performance results back to the user and effectively support the user to adjust memory system parameters directly.

By using the above-mentioned features, it is possible to greatly reduce the time required for design and verification work based on a conventional hardware description language (HDL). In the conventional method, the hardware is designed in HDL by using EDA (electric design automation) tool, the manually generated application code is executed by using the semi-automatically generated simulator, and the HDL is corrected by using the feedback information. It proceeds for several months. Using the proposed tool, the design support tool analyzes the target application code, provides the initial design parameters, and simulates the power consumption and performance information so that the user can modify the design parameters according to the desired specification. Along with information. If the parameter is modified by the developer, the compiler and the simulator are automatically updated to provide the verification result immediately, so that the development work can be shortened to several days.

3. Related works

Leakage current is being an important problem in deepening manufacture process, since it is exponentially increased. Nonvolatile memory has characteristics such as nearly zero leakage current and ultralow power consumption. Thus, it can be used as the next generation memory component to solve the scaling problem. Some studies focus on utilization of nonvolatile memory components in multimedia applications.

Zhou et al. [1] proposed an architecture-level technique that can extend lifetime of phase-change memories to an average of 13 to 22 years. Lee et al. [2] mitigated the drawbacks of PCM by analyzing the effects of buffer sizing, row caching, write reduction, and wear leveling (lifetime reduction). They concluded that PCM is a viable candidate to replace DRAM for scalable main memory. Hu et al. [3] [4] proposed minimizing write activity in NVM through data migration, scheduling, and recomputation. Shi et al. [5] adopted a smart victim cache to reduce write activity in flash main memory. Catherine H. Gebotys [6] aimed to reduce the number of memory accesses through register allocation in order to reduce the total energy of processor using low power memory.

There are many studies on scratch pad memory as a software-controlled cache. They shows that 60% reduction of power consumption by decreasing main memory accesses [7]. However, the problem of increasing leakage current still remains in the studies. To overcome the problem, non-volatile memory is actively studied, and it is the closest to the next generation memory technology. It consumes nearly zero leakage power, small standby power, and relatively less operation power. There is only concern of the overhead of write operations. In order to solve the write operation overhead of NVM, a hybrid cache is proposed to efficiently utilize the advantages of NVM [8]. Traditional studies focus on reduction of write operations to NVM. Thus, they optimize data assignment to place write intensive data on SPM and read intensive data to NVM.

4. Conclusion

In summary, the design of a memory system is a process of finding and modifying a structure that best meets given constraints, taking into account many factors such as performance, price, power, and size. New and complex objects with the advent of web applications and the rising demand for low power users, the complexity of designing system chips is increasing and the rate of change is also accelerating. Therefore, many system designers now require a flexible form of design tool to more easily design their systems according to the situation. In this point of view, the compiler based memory system design environment proposed in this study is not only the most important core technology that facilitates the development of the optimal memory system but also the base technology of the optimum design of the system chip.

Acknowledgement

This research was supported by the Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education (NRF - 2018R1D1A1 B07050054), and Sunchon National University Research Fund Ⅲ in 2018.

References

[1] P. Zhou, B. Zhao, J. Yang, and Y.-T. Zhang, "A Durable and Energy Efficient Main Memory Using Phase Change Memory Technology," Computer Architecture News, vol. 37, issue 3, pp. 14–23, 2009.
DOI: 10.1145/1555754.1555759

[2] B. Lee, P. Zhou, J. Yang, Y. Zhang, B. Zhao, E. Ipek, O. Mutlu, D. Burger, "Phase-Change Technology and the Future of Main Memory," IEEE Micro vol. 30, issue 1, pp. 143-143, 2010.
DOI: 10.1109/MM.2010.24

[3] J. Hu, C. Xue,W. Tseng, Q. Zhuge and E. H.-M. Sha, "Minimizing Write Activities to Non-volatile Memory via Scheduling and Recomputation," in *Proc. 8th IEEE Symposium on Application Specific Processors*, pp. 7–12, 2010.
DOI: 10.1109/SASP.2010.5521139

[4] J. Hu, C. Xue, W. Tseng, Y. He, M. Qiu and E. H.-M. Sha, "Reducing Write Activities on Non-volatile Memories in Embedded CMPs via Data Migration and Recomputation," in *Proceedings 47th IEEE/ACM Design Automation Conference*, pp. 350 – 355, 2010.
DOI: 10.1145/1837274.1837363

[5] L. Shi, C. Xue, J. Hu, W. Tseng and E. H.-M. Sha, "Write Activity Reduction on Flash Main Memory via Smart Victim Cache," in *Proceedings ACM/IEEE 20th Great Lakes Symposium on VLSI*, pp. 91 – 94, 2010.
DOI: 10.1145/1785481.1785503

[6] C. H. Gebotys, "Low Energy Memory and Register Allocation Using Network Flow," in *Proceedings of the 34th Annual Design Automation Conference,* pp. 435–440, 1997.
DOI: 10.1145/266021.266192

[7] Hyungmin Cho, Bernhard Egger, Jaejin Lee, Heonshik Shin, "Dynamic data scratchpad memory management for a memory subsystem with an MMU," in *Proc. of ACM LCTES*, pp. 13-15, 2007.
DOI: 10.1145/1254766.1254804

[8] S. Kang and A. G. Dean, "Leveraging both data cache and scratchpad memory through synergetic data allocation," in *Proc. of IEEE Real-Time and Embedded Technology and Applications Symposium*, pp. 119-128, 2012
DOI: 10.1109/RTAS.2012.22