ELectronics EXpress

IEICE Electronics Express, Vol.5, No.23, 1010–1016

Energy efficiency of scratch-pad memory in deep submicron domains: an empirical study

Hideki Takase[a)], **Hiroyuki Tomiyama, Gang Zeng, and Hiroaki Takada**

Graduate School of Information Science, Nagoya University, Furo-cho, Chikusa-ku, Nagoya 464–8603, Japan

a) *takase@ertl.jp*

Abstract: As the technology scales down to the deep submicron domain, the leakage energy in memory devices could contribute to a significant portion of the total energy consumption. Therefore, evaluation of energy consumption including the leakage energy is necessary. In this paper, we investigate the effectiveness of scratch-pad memory on energy reduction considering both the dynamic and leakage energy. The experiments are performed for 65 nm, 45 nm, and 32 nm technologies. The results demonstrate the effectiveness of scratch-pad memory in deep submicron technology. It is also observed that the leakage energy becomes less significant along with the technology scaling.

Keywords: energy consumption, scratch-pad memory, embedded systems, deep submicron

Classification: Integrated circuits

References

[1] S. Segars, "Low Power Design Techniques for Microprocessors," *IEEE International Solid-State Circuits Conference (Tutorial)*, 2001.

[2] S. Steinke, L. Wehmeye, B. Lee, and P. Marwedel, "Assigning Program and Data Objects to Scratchpad for Energy Reduction," *Proc. Design, Automation Test Europe*, Paris, France, pp. 409–417, March 2002.

[3] F. Algiolini, L. Benini, and A. Caprara, "An Efficient Profile-Based Algorithm for Scratchpad Memory Partitioning," *IEEE Trans. Computer-Aided Design Integr. Circuits Syst.*, vol. 24, no. 11, pp. 1660–1676, Nov. 2005.

[4] P. R. Pand, A. Nicolau, and N. Dutt, *Memory Issues in Embedded Systems-On-Chip*, Kluwer Academic Publishers, 1998.

[5] A. Janapsatya, A. Ignjatovic, and S. Parameswaran, "Exploiting Statistical Information for Implementation of Instruction Scratchpad Memory in Embedded System," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 14, no. 8, pp. 816–829, Aug. 2006.

[6] S. J. E. Wilton and N. P. Jouppi, "CACTI: An Enhanced Cache Access and Cycle Time Model," *IEEE J. Solid-State Circuits*, vol. 31, no. 5, pp. 677–688, May 1996.

[7] SimpleScalar LLC, [Online] http://www.simplescalar.com

DOI: 10.1587/elex.5.1010
Received October 07, 2008
Accepted October 31, 2008
Published December 10, 2008

[8] M. R. Guthaus, J. S. Ringenberg, D. Ernst, T. M. Austin, T. Mudge, and R. B. Brown, "MiBench: A free, commercially representative embedded benchmark suite," *Proc. Workshop on Workload Characterization*, Washington, USA, pp. 3–14, Dec. 2001.

1 Introduction

Energy minimization has become one of the primary goals in the design of embedded systems. These days, cache memory is used not only in general-purpose processors but also in embedded processors in order to improve their performance. However, cache is one of the most energy-hungry components in embedded processors [1]. More recently, scratch-pad memory (SPM) has attracted attention as an alternative to cache memory due to its energy efficiency. SPM consumes less dynamic energy than cache mainly because no tag comparison is necessary.

A number of techniques have been proposed for efficient usage of SPM in terms of energy consumption. The authors of [2] formulated the optimization problem to allocate program code to SPM as a 0/1 programming problem. They demonstrated the effectiveness of SPM against cache memory. In [3], a dynamic programming algorithm for determining allocation of program code to SPM is proposed. The authors of [4] proposed a data allocation method for SPM based on the number of cache conflict misses. In [5], a hardware mechanism for efficient overlay of SPM at runtime is proposed. However, most of them take account of dynamic energy only and neglect the leakage energy in their evaluations.

This paper presents an experimental study on the effectiveness of SPM with considering not only dynamic energy but also leakage energy. It is well known that leakage energy becomes more dominant as the feature size shrinks. In terms of leakage energy, the effectiveness of SPM is not obvious. On one side, in comparable capacity, SPM consumes less leakage power than cache since tag region is not needed. However, the use of SPM instead of cache often results in longer memory access latency, especially in case the size of working set is considerably larger than that of SPM. Thus, SPM has both positive and negative effects on leakage energy. This study assumes 65 nm, 45 nm and 32 nm technologies where leakage energy is significant. To our knowledge, this is the first work which explicitly studies the energy effectiveness of SPM for the deep submicron era.

2 Experimental setup

2.1 Experimental procedure and tools

This paper focuses on energy consumption for instruction accesses. We tested three on-chip instruction memory subsystems: (a) only with cache, (b) with both cache and SPM, and (c) only with SPM. In (a), all the program code is placed on the main memory. In (b) and (c), a part of the program code is

DOI: 10.1587/elex.5.1010
Received October 07, 2008
Accepted October 31, 2008
Published December 10, 2008

placed in the SPM, and the other part is in the main memory. Instructions in the cache or SPM can be fetched by the CPU in a single cycle, otherwise ones can be fetched in multiple cycles.

Our experimental procedure is depicted in Fig. 1. Dynamic and leakage power of memories are calculated based on CACTI 5.0 [6]. We use the SimpleScalar/ARM [7] for the instruction-level simulation. Benchmark programs are cross-compiled into binary code, which is fed by a simulator to generate instruction access traces. Based on the memory access traces, code allocation is decided. The execution cycles are calculated by an in-house cache simulator. We use six benchmark programs from the MiBench suite [8].

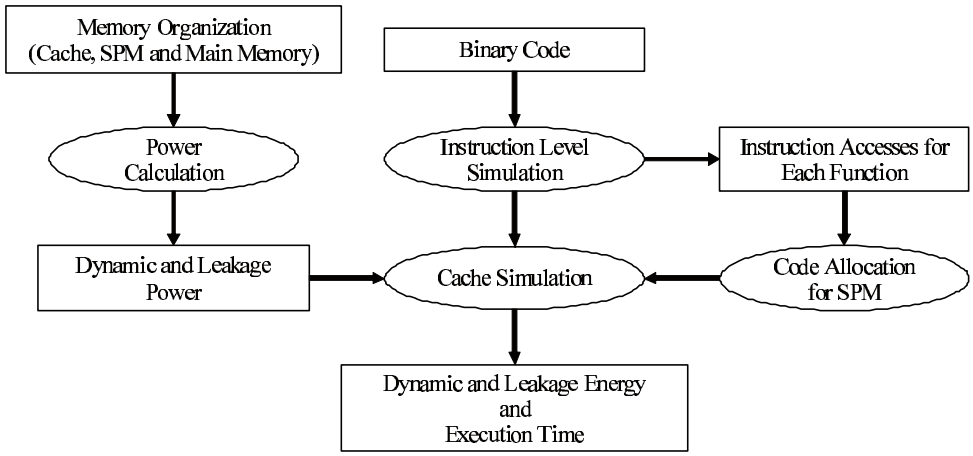

Fig. 1. Experimental procedure

2.2 Energy model

The total energy consumed in the memory system E_{Total} is defined as

$$E_{Total} = E_C + E_S + E_M \tag{1}$$

where E_C, E_S and E_M are the energy consumption of cache, SPM and main memory, respectively. The energy of each memory component is the sum of dynamic and static energy:

$$E_C = E_{C\_dyn} + E_{C\_lkg} \tag{2}$$
$$E_S = E_{S\_dyn} + E_{S\_lkg} \tag{3}$$
$$E_M = E_{M\_dyn} + E_{M\_stby} \tag{4}$$

Dynamic energy consumption of cache E_{C_dyn} is defined as follows.

$$E_{C\_dyn} = E_{C\_read} \times N_{C\_hit} + (E_{C\_read} + E_{C\_write}) \times N_{C\_miss} \tag{5}$$

Here, E_{C_read} and E_{C_write} denote the cache energy on a read access and a write access, respectively. N_{C_hit} and N_{C_miss} denote the number of cache hits and cache misses, respectively.

Also, dynamic energy of SPM E_{S_dyn} is defined as

$$E_{S\_dyn} = E_{S\_read} \times N_{S\_hit} \tag{6}$$

DOI: 10.1587/elex.5.1010
Received October 07, 2008
Accepted October 31, 2008
Published December 10, 2008

where E_{S_read} and N_{S_hit} denote the SPM energy on a read access and the number of access to SPM, respectively.

The dynamic energy consumption of main memory E_{M_dyn} depends on the existence of cache memory. In case cache exists (memory subsystem of (a) and (b) described in Section 2.1), E_{M_dyn} is defined as

$$E_{M\_dyn} = E_{M\_burst\_read} \times N_{C\_miss} \tag{7}$$

where $E_{M\_burst\_read}$ denotes the energy on a burst read access to main memory. In case cache does not exist as (c) in Section 2.1, E_{M_dyn} is defined as

$$E_{M\_dyn} = E_{M\_random\_read} \times N_{S\_miss} \tag{8}$$

where $E_{M\_random\_read}$ denotes the energy on a random read access to main memory and N_{S_miss} the number of SPM misses (in other words, the number of random accesses to main memory).

The static energy depends on not only the static power but also the execution time. The leakage energy of cache E_{C_lkg} and SPM E_{S_lkg} are given by

$$E_{C\_lkg} = P_{C\_lkg} \times EC/f \tag{9}$$
$$E_{S\_lkg} = P_{S\_lkg} \times EC/f \tag{10}$$

while the stand-by energy of main memory E_{M_stby} is given by

$$E_{M\_stby} = P_{M\_stby} \times EC/f \tag{11}$$

where P_{C_lkg} and P_{S_lkg} denote the leakage power of cache and SPM respectively, and P_{M_stby} the stand-by power of main memory, and EC and f the execution cycles and the clock frequency, respectively.

2.3 Code allocation to SPM

This work assumes that allocation of program code to SPM is static; the contents of SPM are not changed during an execution of a program. Code allocation is performed at a function-level granularity. Our allocation method is based on the work in [2] where the allocation problem is formulated as the following knapsack problem.

maximize $\sum_i N(func_i) \times x_i$
subject to $\sum_i C(func_i) \times x_i \leq S$
$func_i$: the i-th function in a given program
$C(func_i)$: the code size of $func_i$
$N(func_i)$: the number of executed instructions in $func_i$
S: the size of SPM
x_i: a 0/1 variable whose value is 1 if $func_i$ is allocated to SPM

In our experiments, GNU ILP solver is used by finding x_i.

DOI: 10.1587/elex.5.1010
Received October 07, 2008
Accepted October 31, 2008
Published December 10, 2008

2.4 Memory parameters

The values of read access energy and leakage power for cache, SPM and main memory are obtained by using CACTI [6]. We assume that cache and SPM is made of SRAM while main memory is of DRAM. The technology size is varied over 65, 45 to 32 nm. The burst and random access times of main memory are also derived from CACTI. The room temperature 27°C is assumed.

In any memory system organization described in Section 2.1, the total size of cache and SPM is fixed to 8 KB. This size ranges over approximately 10 to 20% of the size of the selected benchmark programs. The cache size is selected from 0, 1, 2, 4, and 8 KB, and the SPM size is determined accordingly.

As seen in Section 2.2, the leakage energy depends on the clock frequency, and in turn, the clock frequency should be selected by taking the technology size into account. Considering the roadmaps of state-of-the-art embedded processors in industry, we assume that the clock frequency at 45 nm is 1 GHz, and those at 65 and 32 nm are derived based on the technology scaling.

3 Results and Discussions

In this work, we conducted two sets of experiments. First, we tested the effectiveness of SPM at the 45 nm technology. Then, we investigated the effect of technology scaling on the energy consumption.

3.1 Effectiveness of SPM at 45 nm

Fig. 2 shows the energy and performance results at 45 nm. The lines indicate the normalized execution times. The bars show the energy consumption which are analyzed into four factors: stand-by energy and dynamic energy of main memory, leakage energy and dynamic energy of SRAM (including both cache and SPM).

From this figure, the use of cache only results in the largest energy consumption for all the programs. This demonstrates the effectiveness of SPM at 45 nm.

It is observed that the effect of SRAM leakage energy is not trivial. For patricia, the SRAM leakage energy greater than its dynamic energy. Thus, it is not feasible to ignore the leakage energy when we evaluate the energy consumption of memory systems.

The SRAM leakage energy tends to decrease as the SPM size increases. This demonstrates the effectiveness of SPM on the leakage energy reduction. The execution time tends to become worse as the cache size decreases. This is because the decreased cache size causes more cache misses. On the other hand, the increase in the execution time is not significant. Again, patricia is an exception where combination of cache and SPM gives the highest performance as well as the lowest energy.

DOI: 10.1587/elex.5.1010
Received October 07, 2008
Accepted October 31, 2008
Published December 10, 2008

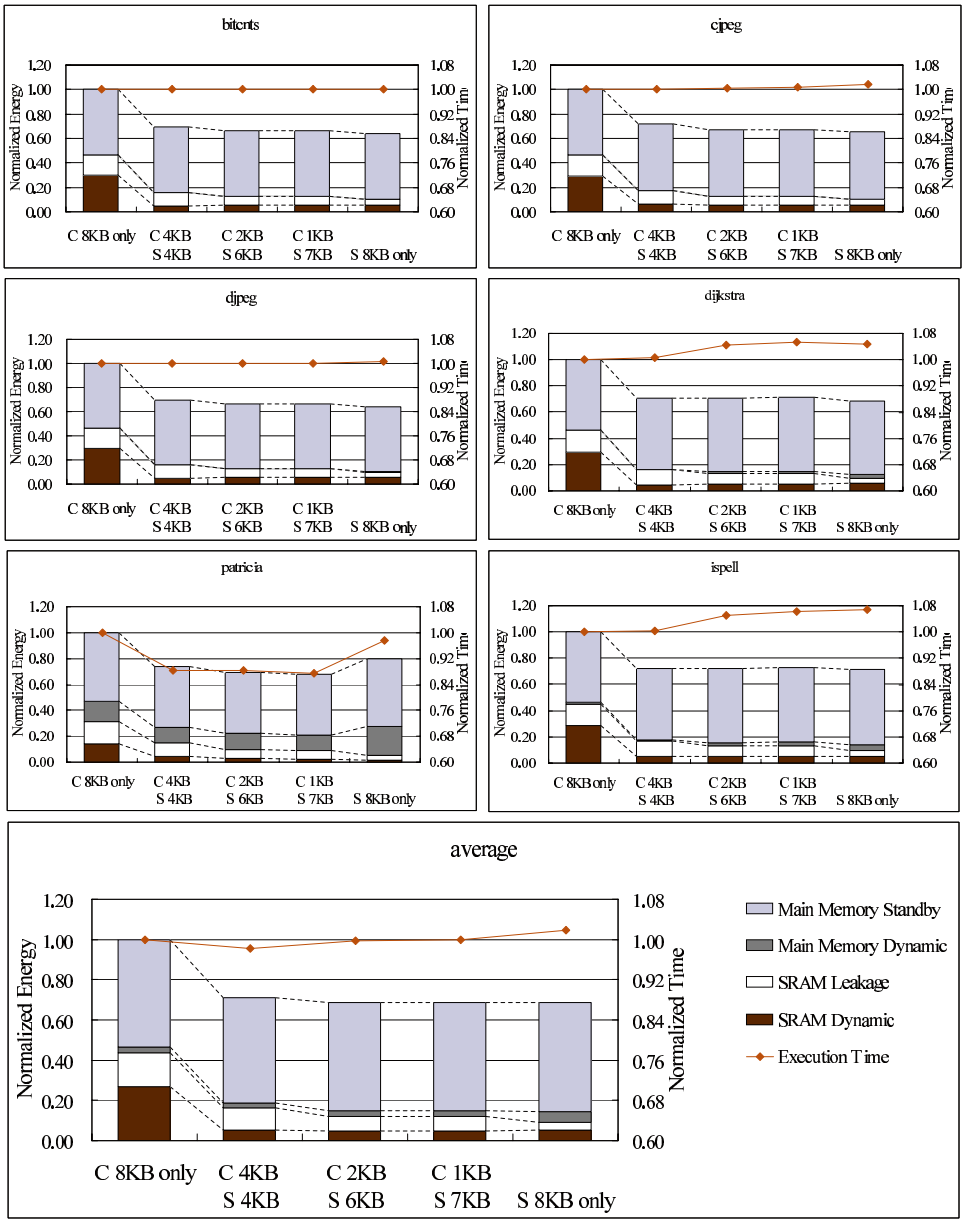

Fig. 2. Energy consumption and performance at 45 nm

3.2 Effects of technology scaling

Fig. 3 shows the average energy and execution time for 65, 45 and 32 nm technologies. As the technology shrinks, both the energy consumption and the execution time are reduced. An interesting observation is that the percentage of the SRAM leakage energy is decreased along with the technology scaling. To our knowledge, this trend has not widely been recognized. One of the major reasons for the trend is the clock frequency. As the technology shrinks, the clock frequency is improved. Then, an application program finishes its execution in a shorter time, and thus, the leakage energy is reduced. This implies that the memory should be powered-off when no task is ready to run. Otherwise, the leakage energy will be a serious problem.

DOI: 10.1587/elex.5.1010
Received October 07, 2008
Accepted October 31, 2008
Published December 10, 2008

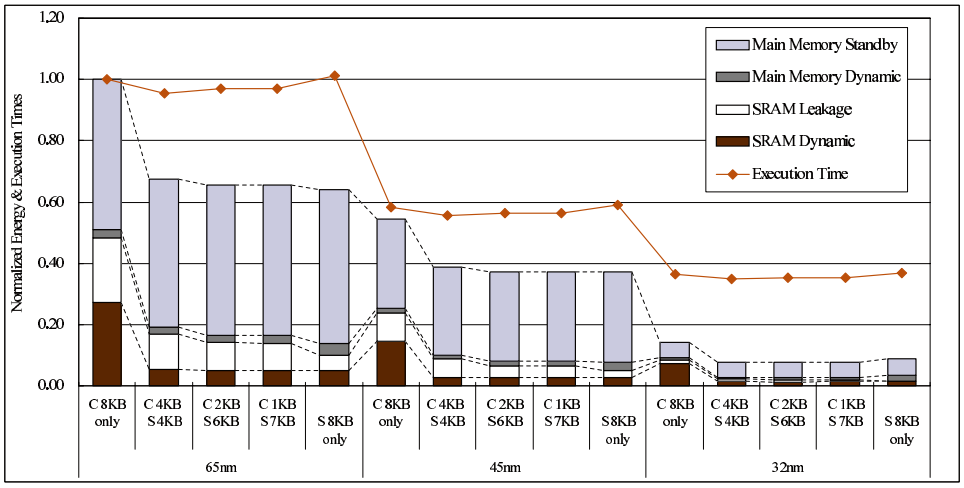

Fig. 3. Effects on technology scaling on energy consumption and performance (average of six programs)

4 Conclusions

Scratch-Pad Memory (SPM) has been considered as a promising solution for energy optimization in embedded systems. However, most of past studies focused on dynamic energy reduction, and neglected the leakage energy in their evaluations. This is the first paper for presenting experimental studies on the energy effectiveness of SPM at 65, 45 and 32 nm with consideration of not only the dynamic energy but also the leakage energy. The results demonstrate the effectiveness of SPM in the deep submicron technology. It is also observed that the leakage energy becomes less significant along with the technology scaling.

Acknowledgments

This work is supported in part by Core Research for Evolutional Science and Technology (CREST) from Japan Science and Technology Agency.

DOI: 10.1587/elex.5.1010
Received October 07, 2008
Accepted October 31, 2008
Published December 10, 2008