Title	キャッシュメモリの消費電力削減を目的とした自発的無効化命令の適用法に関する研究
Author(s)	山野，純嗣
Citation	
Issue Date	2008-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/4307
Rights	
Description	Supervisor:田中清史，情報科学研究科，修士

Research on Application of Software Self-Invalidation for Energy Reduction in Cache Memory

Junji Yamano (0610089)

School of Information Science,
Japan Advanced Institute of Science and Technology

February 7, 2008

Keywords: gated-V$_{dd}$, Self-Invalidation, last-touch memory reference instruction.

1 Introduction

Recently, semiconductor process technology is improving and transistors are shrinking in size. Therefore, processor's speeds have been dramatically upgraded. On the other hand, since leakage currents of a transistor are becoming too larger to ignore, the power consumption of a processor is increasing. This affects battery life of mobile devices which are required to have high-performance and long battery life in recent years. Consequently, it is important to reduce power consumption of a processor while maintaining its high-performance.

The power consumption of a cache memory accounts for majority of the power consumption of a processor. This is because a cache memory accounts for a large area of a processor. So, reducing to the power consumption of a cache memory leads to reducing the power consumption of a processor.

For these reasons, I aim at reducing power that the cache memory consumes without overheads in program execution in this research.

2 Related Works

Recntly, there are a lot of researches to reduce the power of a cache memory. The gated-V$_{dd}$ is one of the technique to reduce power of a cache memory. The gated-V$_{dd}$ is a transistor that has a high threshold between SRAM cell and GND, and controls power supply.

Self-Invalidation is a concept of invalidating previously a cache block that will be invalidated. The software Self-Invalidation applies this concept to reduction of the power consumption of a cache memory. The software Self-Invalidation introduces the last-touch memory reference instruction. The last-touch memory reference instruction is the same as load/store instruction. Add to this, it controls power supply to cache block. The memory reference instruction for a block that will be invalidated is replaced by the last-touch memory reference instruction. And the power supply of the cache block is cut off. But when finding the last reference to the cache block, it is necessary to grasp the memory references in the program execution completely. Therefore, the software Self-Invalidation needs the advance execution of a program to get the information. This technique lacks practicality.

3 Proposal Technique

In this research, I propose an application technique of software Self-Invalidation. This technique solves problems of software Self-Invalidation. The application technique predicts the memory reference instruction that can be replaced by the last-touch memory reference instruction, and replaces automatically. The application technique of software Self-Invalidation is classified into two methods.

- Static application technique of software Self-Invalidation
  When compiling a program, predict the instruction that can be replaced, and replace.

- Dynamic application technique of software Self-Invalidation
  While executing a program, predict the instruction that can be replaced, and replace.

In this research, I focus on the static application technique. This technique aims at getting the same effects as conventional software Self-Invalidation.

The static application technique focuses on an array type data reference in the loop of a program, and checks whether the data will be reused or not. If it has no reuse in the future, the compiler replaces the memory reference instruction to the data by the last-touch memory reference instruction. In this way, I get efficient application of the software Self-Invalidation. The power consumption of a cache memory is reduced.

4 Evaluation

I evaluate an effect of power reduction of the proposal technique by simulation. To make comparison with the conventional techniques easy, the target to evaluate is the same as conventional techniques, a 2-core chip multiprocessor that has private L1 data and instruction cache per processor core. Benchmark programs are RADIX, FFT, LU (contiguous), LU (non-contiguous), and CHOLESKY in SPLASH-2. I mainly evaluate the power consumed by the L1 data cache when applying the proposal technique to each program.

From results of simulation, the technique could reduce power consumption of L1 data cache by 20.75% on average. In addition, the proposed technique could decrease cache misses, which led to efficient usage of cache memory.

5 Conclusion

In this thesis, I examined a technique for an application of software Self-Invalidation which is a technique for reducing processor power consumption. I proposed the static application technique of software Self-Invalidation. The static application technique solves problems of software Self-Invalidation, and aimed at getting the same effects as software Self-Invalidation.

From results of evaluation, I obtained effects of power reduction by 20.75% on average.