A Stady of TLB Preloading by Using Page Address Prediction

Tomoaki Ukezono (110019)

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

February 14, 2002

Keywords: TLB, prediction, preload, MMU.

Abstract

In recent years, the improvement in the clock frequency of microprocessors and the enlargement of memory in computer systems tend to extend the size of working set in programs. This tendency degrades efficiency of computation in processors managing virtual memory. Processors with virtual memory management usually include TLB (Translation Lookaside Buffer). The TLB is a processor built-in cache of a page table in virtual memory, which makes address translation and memory protection fast. A TLB reach is the range that the TLB can map in memory at a time and is one of indicators that influence the performance of TLB. The TLB performs replacement of its entries when the size of working set in a program exceeds the TLB reach. When the size of working set is much larger than the TLB reach, the replacement happens frequently, which has a possibility of causing thrashing where the TLB cannot work. Increasing the number of TLB entries or expanding the size of page for preventing thrashing from occurring causes degradation of the clock frequency memory fragmentation. In this thesis, we propose the hardware TLB preloding scheme using page address prediction that settles this problem.

1 Introduction

Generally, today's processors provide a virtual memory mechanism. The virtual memory raises diversity and flexibility of computers, while it makes program execution somewhat inefficient. A TLB (Translation Lookaside Buffer) is a cache of a page table and supports the virtual memory. A TLB reach is the range that the TLB can map in memory at a time.

Execution of programs with data sets whose size exceeds the TLB reach causes TLB thrashing, leading to serious inefficiency in computation. This study aims at improvement of the TLB performance by prediction without expanding the TLB reach.

2 Page Address Prediction

We require appropriate prediction policy when improve TLB performance by use prediction. In this papar, We propose a scheme of improvement of TLB performance extended TLB Preloading using linear page address prediction. This scheme is that predicting ± 1 page table entry address from current access page table entry adderess and preloading page table entry from page table in advance. After loaded page table entry, that page table entry is inserted a buffer separated from TLB. We avilable to use our scheme's characteristics by so doing, and then improve TLB performance. We show the characteristics of proposed mechanism as follows.

1. This mechanism can lelieves first page access TLB miss.
2. This mechanism can improve TLB reach problem by reduce TLB inserted page table entry(PTE)
3. This mechanism can put in conventional TLB system only by little modifying.
4. This mechanism don't falls conventioal TLB performance, and improves TLB preformance only when hit its prediction.

3 Configuration of Linear Page Address Predictor For Nonlinear Page Access

We propose configuration of linear page address predictor for nonlinear page access in order to accommodate nonlinear page access. Our proposals are WRS(Wide Range Support) configuration and MOS(Multiple Operand Support) configuration. WRS configuration can accommodate nonlinear

page access generateed in near page on virtual address space by exteding prediction range. The conventional linear predictor prepared prediction buffer for ± 1 prediction. WRS configuration extends this prediction buffer for ± 2 , ± 3 ,and more. MOS configuration can accommodate nonlinear page access generated by arithmetic operands by preparing linear predictor in parallel. We propose WRS implementation design that use memory burst transfer that is being used at cache between main memory.

4 Implementation Hardware

In this papar, we designed logical circuit to estimate our proposed mechanism performance. A design is described by hardware description language "VHDL". we designed the any circuit in RTL level design, such as instruction execution pipeline , MMU(Memory Management Unit) and proposed mechanism.

5 Methodology

In this papar, we examined our designed mechanism both the quantity of hardware and its delay. we can estimated the quantity of hardware and the delay by using FPGA Logical Synthesis Compiler, and we can examined our desinged mechanism performance by using Logical Circuit Simulator.

6 Estimate

Our proposed mechanism is feasible design small enough, and reduce TLB misses in a large data processing program that is sure to cause TLB TLB thrashing.

7 Summary

Our proposed mechanism can suppress many TLB misses in spite of implementation of low hardware quantitiy, and don't falls conventioal TLB performance in any cases.