Available online at www.sciencedirect.com
ScienceDirect
Microprocessors and Microsystems 32 (2008) 79–94
MICROPROCESSORS AND MICROSYSTEMS
www.elsevier.com/locate/micpro

Partial resolution for redundant operation table

Byung-Soo Choi, Jun-Dong Cho *

School of Information and Communication, Sungkyunkwan University, Republic of Korea

Available online 4 July 2007

Abstract

In this work, we discuss several drawbacks of the conventional wide-width redundant operation table such as the waste of area cost and power consumption. We found that the waste of area cost and power consumption is caused by storing meaningless bits of the narrow-width operand values. Based on this analysis, we propose a way to avoid these storing of meaningless information of the narrow-width operands. The proposed method, *partial resolution method*, divides the conventional wide-width redundant operation table into two tables as the wide-width table for the half entries and the narrow-width table for the other half entries. The wide-width and the narrow-width redundant operation tables store different dynamic instructions whose operand values are wide and narrow, respectively. Since the narrow-width redundant operation table stores smaller number of bits, it requires lower area cost and also power consumption compared with the wide-width redundant operation table. The partial resolution method decreases the area cost by about 7% and 20% for the integer and the floating-point tables, respectively, and reduces the dynamic power consumption by about 34% and 30% for the integer and the floating-point tables, respectively, compared with the conventional wide-width redundant operation table with 2K entries. Meanwhile, the performance simulation with a high-end microarchitecture model and SPEC2000 benchmarks shows that the partial resolution method affects the performance very little, and even increases slightly in terms of IPC (Instruction per Cycle) value.

Keywords: Redundant operation table; Microarchitecture; Instruction-level parallelism; High performance; Low implementation cost; Low power

1. Introduction

Many methods for the high performance microprocessors have been proposed to increase the instruction-level parallelism. With the higher instruction-level parallelism, many operations executes speculatively. Meanwhile many of the speculative and the non-speculative operations may be redundant. Note that an operation is redundant when its inputs are the same with that of previous execution. It is important to indicate that the speculative or the redundant operations limit the performance improvement, and increase the power consumption as well [1–3]. In other words, a high performance method has advantages such as the higher speedup and disadvantages such as the more redundant operations in general. Since a redundant operation generates the same result value with a previous instance of the same operation, it wastes the power consumption and the execution time. To reduce such drawbacks of the redundant operations, many optimization methods have been proposed (refer [4–6] and references therein). One of the solutions is the elimination method of the redundant operations by the simple lookup operations of a table (refer [7–22] and references therein), which is called a redundant operation table. Note although there are many different names of the conceptually same table, the redundant operation table is used in this work just for a consistency. Usually a redundant operation table stores two operands as a tag and one result as an output of the operation.

Although the redundant operation table was originally proposed to reduce the waste of the computation time and the power consumption caused by the redundant operations, it has also its own drawbacks. Since a redundant operation table requires large number of bits in the tag and the result parts, it also increases the area cost and the power consumption of the processor itself. For

* Corresponding author.
E-mail addresses: bschoi3@gmail.com (B.-S. Choi), jdcho@skku.ac.kr (J.-D. Cho).

0141-9331/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.micpro.2007.06.003

example, a redundant operation table requires more than the half of the area cost for the instruction and the data caches in the high-end microprocessor Alpha 21264, and hence adds more than the half of the total power consumption. Section 4 explains more precisely several drawbacks of the redundant operation table. Hence unless the area cost and the power consumption of the redundant operation table are minimized, the motivation of proposing the redundant operation table is questionable. Surprisingly there have been no studies to touch such drawbacks of the redundant operation table as far as we know.

Hence in this work we propose a way to resolve the above-mentioned drawbacks. It is better to understand the structure of the redundant operation table to find any clues for achieving our purpose. The redundant operation table stores two operands and a result value of the previously executed operations [16,17]. Usually, two operands and one result in the conventional wide-width redundant operation table require a sufficiently large number of bits in order to support the full precision of the register value for the target processor. Meanwhile, compared with the data cache, the redundant operation table assumes two operands as a tag for a corresponding entry. Since the number of operands is usually two and the number of result is one, the tag part stores more bits than the result part in the redundant operation table. Since the tag part stores a large number of bits, it requires higher area cost and more power consumption than the result part. Therefore, in this work, we focus on the tag part, and introduce an optimization method for it.

In this paper, we make the following contributions.

Analyze operand values: We analyze all executed or dynamic instructions, which require two operands, of the chosen SPEC2000 benchmark programs. A preliminary analysis of the operand values for the integer and the floating-point operations reveals that the most operands can be represented with a small number of bits. Therefore, a full precision tag is not required for all instructions, and we call it an operand locality.

Propose partial resolution method: To exploit the mentioned operand locality, we propose a *partial resolution method* for the tag part in the conventional wide-width redundant operation table. Note that the concept of the partial resolution method looks similar to the partial-tag method [23], which was proposed for the value predictors. However, two ways are quite different since the partial-tag method for the value predictors stores only partial information of operands, but the partial resolution method for the redundant operation table must store exact information of operands. Hence, the partial-tag method for a value predictor cannot be directly used for the redundant operation table. To implement the partial resolution method, we propose a wide–narrow-width redundant operation table, which consists of two redundant operation tables: the wide-width redundant operation table for the half entries and the narrow-width redundant operation table for the other half entries. The wide-width redundant operation table works the same as the conventional wide-width redundant operation table except it stores not all instructions but some instructions whose operand values require a wide bit width. On the other hand, the narrow-width redundant operation table stores some instructions whose operands require a narrow bit width, but sufficient, to represent the exact information of the operands. Therefore each instruction must be stored in the one of two tables depends on the width of the operand values.

Analyze partial resolution method: The partial resolution method decreases the area cost by about 7% and 20% at the maximum for the integer and the floating-point redundant operation tables, respectively, compared with the conventional ones. Also it reduces the dynamic power consumption of the conventional wide-width redundant operation table by about 34% and 30% at the maximum for the integer and the floating-point operations, respectively. Note that to analyze the integer and the floating-point redundant operation tables at the same time, we simulate a chosen set of SPEC2000 floating-point benchmarks only since it contains both the integer and the floating-point operations. The performance simulation with a high-end microarchitecture shows a negligible performance variation and even a slight improvement in terms of IPC (instruction per cycle) values.

This paper is organized as follows. Related works are discussed in Section 2. Section 3 illustrates the way for the elimination of redundant operations and its table structure as a redundant operation table. Section 4 discusses several drawbacks of the conventional wide-width redundant operation table, and explains the motivation of this work. Section 5 describes the proposed partial resolution method to exploit the locality of operands, and explains its implementation as the wide–narrow-width redundant operation table. The effects of the partial resolution method for the area cost, the power consumption, and the performance are analyzed in Section 6. Section 7 concludes this research with a discussion about the advantages of the partial resolution method, and talks about several future works.

2. Related works

Many ways have been proposed to eliminate redundant executions statically or dynamically. Generally a static method is based on a compiler optimization method to reduce trivial computations or to replace a complex operation into a simplified operation. On the other hand, a dynamic approach utilizes a table, which is called a redundant operation table in this work, to memorize the instances or the traces of operations. In the performance improvement point of view, the dynamic ways are better than the static ways since the former can find all redundant operations while the latter cannot. Although it is almost impossible to discuss all related works in this paper, we briefly discuss several, but mostly related, previous works in this section.