Speculative Execution in Modern Computer Architectures

David Kaeli
Department of Electrical and Computer Engineering
Northeastern University and
Pen-Chung Yew
University of Minnesota
Department of Computer Sciences and Engineering

October 21, 2003

Abstract

As the progress of VLSI technology soon will allow more than 1 billion transistors on a single chip, many new computer architectures have been proposed to take advantage of such an abundance of transistors and further enhance processor performance.

Two main basic techniques have been used successfully so far to improve processor performance. They include (1) improving clock rate and (2) exploiting instruction-level parallelism through pipelining, out-of-order execution and multithreading.

However, to expose the potential instruction-level parallelism present in programs, control flow and data flow constraints in the program must be overcome. The rapid increase in levels of integration has enabled designers to utilize sophisticated speculation techniques to reduce the latency typically encountered during instruction delivery and execution. Most high-performance microprocessors today utilize different forms of control and data speculation.

This book is intended to serve as an authoritative guide describing many of the advances in speculative execution techniques. The text will describe both cutting-edge research projects in the area of speculative execution, as well as a number of commercial implementations that have illustrated the value of this latency-hiding technique.

1 Features

This book features a comprehensive coverage of advanced and timely topics on speculative execution techniques in modern and future computer architectures. These techniques are essential in exploiting instruction level parallelism for higher performance. The book will begin with reviewing various control speculation techniques that use instruction cache prefetching, branch prediction, branch predication and multi-path execution. It is followed by dataflow speculation techniques such as data cache prefetching, address value speculation, data value speculation, precomputation and coherence speculation. More recent speculative multithreaded approaches will also be covered with an emphasis on profile-guided speculation, recent speculative microarchitectures and compiler techniques needed to support such architectural approaches.

2 Intended Audience

The subject of this book covers the core materials that are important to researchers, students, and practicing professionals in the fields of computer architecture, compilers, and system designs. This book assumes that the readers have general knowledge about computer architectures, compilers and application programs. It can be used as a textbook for senior and graduate students and a reference book for practicing professionals.

3 Market Information

One of the motivation for editing this book is the lack of adequate reference books that cover the most recent development in the field of high performance computer architectures. There are currently several textbooks on advanced computer architectures on the market. However, their coverage is mostly focussed on instruction set architectures, pipelined microarchitectures as well as cache memory hierarchy. There is no reference book that focus on the most dynamic subject of speculative execution that holds the key to the future high performance computer architectures. Almost none of the major computer architecture conferences in recent years can go without including some papers touched on the subject of speculative execution. However, there has been no reference book so far that attempts to summarize and give a general overview of this important subject. We feel this book could fill that important vacuum.

4 Table of Contents

4.1 Control Speculation

4.1.1 Instruction Cache Prefetching

This chapter will review techniques that attempt to preload the instruction cache prior to an actual request.

4.1.2 Branch Prediction

This chapter will cover many of the branch prediction techniques used in common microprocessors today.

4.1.3 Trace Caches

This chapter will discuss this popular technique for caching many basic blocks from the frequently executed path.

4.1.4 Branch Predication

This chapter will discuss the concept of predicated execution, and show how predicates can be used to reduce the complexity associated with speculative control flow execution.

4.1.5 Multi-path Execution

This chapter will discuss some of the options to pursue multiple paths of execution in order to hide the latency associated with unpredictable control flow.

4.2 Dataflow Speculation

4.2.1 Data Cache Prefetching

This chapter will review techniques that attempt to preload the data cache prior to an actual request.

4.2.2 Address Value Speculation

This chapter will discuss the ability of a microprocessor to speculate on the value of a load address.

4.2.3 Data Value Speculation

This chapter will discuss the ability of a microprocessor to speculate on the value that will be loaded from memory.

4.2.4 Precomputation

This chapter will present a discussion on the benefits of caching past computation results.

4.2.5 Coherence Speculation

This chapter will present the recent ideas about speculative coherence and consistency schemes targeting shared memory systems.

4.3 Related Techniques

4.3.1 Profile-guided Speculation

This chapter will discuss how profiling and be effectively used to guide speculation to impact performance.

4.3.2 Multi-threading and Speculation

This chapter will discuss how to use speculative threads to hide execution latencies.

4.3.3 Compilation and Speculation

This chapter will review a number of the compilation techniques that can be used to aid speculation.

4.4 Recent Speculative Microarchitectures

This chapters will review some of the recently proposed and commercially available speculative microarchitectures which include superspeculative architecture, the WARP engine, Resource Flow and Intel's Itanium processors.