Chapter 24
Simultaneous MultiThreading Microarchitecture

Chen Liu
Florida International University, USA

Xiaobin Li
Intel® Corporation, USA

Shaoshan Liu
University of California, Irvine, USA

Jean-Luc Gaudiot
University of California, Irvine, USA

ABSTRACT
Due to the conventional sequential programming model, the Instruction-Level Parallelism (ILP) that modern superscalar processors can explore is inherently limited. Hence, multithreading architectures have been proposed to exploit Thread-Level Parallelism (TLP) in addition to conventional ILP. By issuing and executing instructions from multiple threads at each clock cycle, Simultaneous MultiThreading (SMT) achieves some of the best possible system resource utilization and accordingly higher instruction throughput. In this chapter, the authors describe the origin of SMT microarchitecture, comparing it with other multithreading microarchitectures. They identify several key aspects for high-performance SMT design: fetch policy, handling long-latency instructions, resource sharing control, synchronization and communication. They also describe some potential benefits of SMT microarchitecture: SMT for fault-tolerance and SMT for secure communications. Given the need to support sequential legacy code and emerge of new parallel programming model, we believe SMT microarchitecture will play a vital role as we enter the multi-thread multi/many-core processor design era.

INTRODUCTION
Ever since the first integrated circuits (IC) were independently invented by Jack Kilby (Nobel Prize Laureate in Physics in 2000) from Texas Instruments and Robert Noyce (co-founder of Intel®) around 50 years ago, we have witnessed an exponential growth of the whole semiconductor industry.

DOI: 10.4018/978-1-60566-661-7.ch024
Simultaneous MultiThreading Microarchitecture

Moore’s Law and Memory Wall

The semiconductor industry has been driven by Moore’s law (Moore, 1965) for about 40 years with the continuing advancements in VLSI technology. Moore’s law states that the number of transistors on a single chip doubles every two years, as shown in Figure 1, which is based on data from both Intel® and AMD. A corollary of Moore’s law states that the feature size of chip manufacturing technology keeps decreasing at the rate of one half approximately every five years (a quarter every two years), based on our observation shown in Figure 2.

As the number of transistors on a chip grows exponentially, we have reached the point where we could have more than one billion transistors on a single chip. For example, the Dual-Core Itanium® 2 from Intel® integrates more than 1.7 billion transistors (Intel, 2006). How to efficiently utilize this huge amount of transistor estate is a challenging task which has recently preoccupied many researchers and system architects from both academia and industry.

Processor and memory integration technologies both follow Moore’s law. Memory latency, however, is drastically increasing relatively to the processor speed. This is often referred to as the “Memory Wall” problem (Hennessy, 2006). Indeed, Figure 3 shows that CPU performance increases at an average rate of 55% per year, while the memory performance increases at a much lower 7% per year average rate. There is no sign this gap will be remedied in the near future. Even though the processor speed is continuously increasing, and processors can handle increasing numbers of instructions in one clock cycle, however, we will continue experiencing considerable performance degradation each time we need to access the memory. Pipeline stalls will occur when the data does not arrive soon enough after it has been requested from the memory.
29 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the product’s webpage:

www.igi-global.com/chapter/simultaneous-multithreading-microarchitecture/36424?camid=4v1


www.igi-global.com/e-resources/library-recommendation/?id=1

Related Content

A Grid Aware Large Scale Agent-based Simulation System
Dan Chen (2009). *Quantitative Quality of Service for Grid Computing: Applications for Heterogeneity, Large-Scale Distribution, and Dynamic Environments* (pp. 299-319).
www.igi-global.com/chapter/grid-aware-large-scale-agent/28283?camid=4v1a

A Two-Level Fuzzy Value-Based Replica Replacement Algorithm in Data Grids

Hardware Transactional Memories: A Survey
www.igi-global.com/chapter/hardware-transactional-memories/159039?camid=4v1a

Testing-Effort Dependent Software Reliability Model for Distributed Systems
www.igi-global.com/article/testing-effort-dependent-software-reliability/78150?camid=4v1a