

Hardware Performance Monitoring in Multiprocessors

by

Guy G. F. Lemieux

A thesis submitted in conformity with the requirements
for the degree of Master of Applied Science
Graduate Department of Electrical and Computer Engineering
University of Toronto

Copyright © 1996 by Guy G. F. Lemieux

Hardware Performance Monitoring in Multiprocessors

Guy G. F. Lemieux

Master of Applied Science, 1996

Department of Electrical and Computer Engineering
University of Toronto

Abstract

Multiprocessors are often quoted as being capable of a ‘peak performance,’ but in practise it is difficult to utilize this potential. Consequently, software applications must be well tuned to run efficiently.

In this thesis, factors affecting performance in cache-coherent multiprocessors, particularly those which use a sequentially-consistent memory model, are analyzed. Cost-effective hardware features are suggested to measure these factors without intruding on software performance. Many of the proposed features can be readily incorporated into future processor designs, and others are easy to implement in external hardware. The measurements can be used to help tune a program’s performance as well as other purposes such as simulation verification, workload characterization, or runtime performance decision support.

To demonstrate cost feasibility and other implementation concerns, the processor card hardware performance monitor developed for the NUMAchine multiprocessor is also described.

Acknowledgements

I am indebted to the many people who have helped me along in the course of developing this thesis. In particular, I wish to thank Stephen Brown and Zvonko Vranesic for the opportunity and their support and encouragement.

Next, I am very grateful for the sacrifices of my Grandmother, who worked hard to help support me throughout my university education. My love is with her always.

Of course, my friends have contributed to maintaining my sanity and providing distractions. Thanks, Kelvin, for the many great squash games! I have also enjoyed the company of Steve, Zeljko, Derek, Dan, Alex, Robin, Rob, Mitch, Dean, Ben, Gordon, Wing-Chi, and many, many others. Most importantly, I will always cherish Martha's love, support and companionship.

The helpfulness of discussions and criticisms with Naraig, Orran, Todd, and Keith is appreciated.

Financial support from OGS, ITRC, UofT, and Hewlett-Packard is gratefully acknowledged.

Table of Contents

Abstract.....	ii
Acknowledgements	iii
Table of Contents	iv
List of Figures.....	vii
List of Tables	viii
Chapter 1 <i>Introduction</i>	1
1.1 Motivation	1
1.2 Objectives	3
1.3 Contributions	4
1.4 Overview	5
Chapter 2 <i>Background</i>.....	6
2.1 Monitoring in Software	6
2.1.1 gprof	6
2.1.2 Mtool	7
2.1.3 Paradyn.....	9
2.1.4 QPT and CProf	10
2.1.5 MemSpy	11
2.1.6 Summary	12
2.2 Hardware Monitoring in Processors	12
2.2.1 Intel Pentium and Pentium Pro.....	13
2.2.2 DEC Alpha 21064	14
2.2.3 MIPS R4400 and R10000	15
2.2.4 Hewlett-Packard PA-RISC	16
2.2.5 Sun SPARC	17
2.2.6 IBM/Motorola/Apple PowerPC 604	18
2.2.7 IBM POWER2	18
2.2.8 Processor Summary.....	20
2.3 Hardware Monitoring in Systems.....	21
2.3.1 University of Toronto Hector	21
2.3.2 Stanford DASH	23
2.3.3 MIT M-Machine.....	25
2.3.4 Cray Research Inc.	26
2.3.5 Convex Computer Corporation	27
2.3.6 Kendall Square Research.....	29
2.4 Summary.....	31
Chapter 3 <i>Multiprocessor Hardware Performance Monitoring</i>	32
3.1 Performance Losses: Software View	32

3.1.1	Inefficient Algorithms	33
3.1.2	Extra Parallel Work	33
3.1.3	Synchronization.....	33
3.1.4	Load Imbalance	34
3.1.5	Inefficient System and Library Calls.....	34
3.1.6	Amdahl's Law	34
3.1.7	Summary	35
3.2	Performance Losses: Hardware View.....	36
3.2.1	Instructions	36
3.2.2	Instruction Scheduling and CPI.....	37
3.2.3	Memory Stalls	38
3.2.4	Other Stalls.....	38
3.2.5	Exceptions and Interrupts.....	39
3.2.6	Branch Mispredictions	40
3.2.7	Special Cases.....	40
3.2.8	Summary	41
3.3	Processor View of Memory Stalls	42
3.3.1	Counting Cache Misses	43
3.3.2	Classifying Cache Misses.....	45
3.3.3	Measuring Cache Miss Latency	54
3.3.4	Summary: Processor View of Memory Stalls	61
3.4	Hardware Support for Software Tools	63
3.4.1	Timer Support.....	63
3.4.2	Basic Block Profiling	63
3.4.3	Pointing Losses Back to Code and Data	67
3.4.4	Summary: Hardware Support for Software Tools.....	67
3.5	Summary.....	68

Chapter 4 Hardware Implementation 70

4.1	Introduction to NUMAchine	70
4.2	Processor Card Organization	73
4.3	Monitor Organization	75
4.3.1	Local Bus Controller	75
4.3.2	Pipeline Status	77
4.3.3	Counters & Interrupts.....	80
4.3.4	SRAM Memory	81
4.3.5	Count & Increment.....	82
4.3.6	Latency Timer	82
4.3.7	Configuration Controller	83
4.4	Programmable Configuration	83
4.4.1	PhaseID Register and PhaseID Watch.....	84
4.4.2	Command Watch and Command Filter	84
4.4.3	Address Watch and Address Filter	85
4.4.4	General-Purpose Counter Modes	87
4.4.5	SRAM Counter Mode	91
4.4.6	Master Counter Enables	95

4.5 Summary.....	95
Chapter 5 <i>Conclusions</i>	98
5.1 Contributions	98
5.2 Future Work.....	101
Appendix A <i>Memory Card Monitoring</i>	102
Bibliography	104

List of Figures

FIGURE 2.1. SUPERMON block diagram.....	22
FIGURE 2.2. DASH performance-monitoring hardware	24
FIGURE 3.1. An example of Amdahl's Law limitations on speedup.....	35
FIGURE 3.2. Time-based basic block profiling without special hardware support.....	66
FIGURE 3.3. Time-based basic block profiling with special hardware support.....	66
FIGURE 4.1. The NUMAchine station.....	71
FIGURE 4.2. The NUMAchine hierarchy	71
FIGURE 4.3. Memory consistency state transition diagrams	73
FIGURE 4.4. NUMAchine processor card datapath organization	74
FIGURE 4.5. Processor card performance-monitoring subsystem	76
FIGURE 4.6. Counters & Interrupts datapath, with write path highlighted	81
FIGURE 4.7. MCC registers with one Count-Enable Circuit.....	88
FIGURE 4.8. State machines to detect invalidation hits and misses	91
FIGURE 4.9. SRAM address generation. The outlined portion is inside the MCC	93
FIGURE A.1. Memory card organization.....	102

List of Tables

TABLE 2.1.	Intel Pentium performance-monitoring counter inputs.....	14
TABLE 2.2.	DEC Alpha 21064 performance-monitoring counters and inputs	15
TABLE 2.3.	MIPS R4400 externally observable events.....	16
TABLE 2.4.	MIPS R10000 performance-monitoring counters and inputs	17
TABLE 2.5.	PowerPC 604 performance-monitoring counters and inputs.....	19
TABLE 2.6.	Overview of performance-monitoring features on current processors	20
TABLE 2.7.	CRAY Y-MP C90 performance-monitoring counter.	28
TABLE 2.8.	Convex Exemplar SPP-1200 performance-monitoring counter inputs	29
TABLE 2.9.	KSR performance-monitoring counters.....	30
TABLE 4.1.	Breakdown of 64-bit address space in NUMAchine	77
TABLE 4.2.	Pipeline Status bits PS[6..5] specify one of three operating modes	79
TABLE 4.3.	Pipeline Status bits PS[4..0] select which events to monitor.....	79
TABLE 4.4.	Master Configuration Controller user configuration registers	86
TABLE 4.5.	General-purpose counter events.....	89
TABLE 4.6.	SRAM counter events.....	92
TABLE 4.7.	SRAM counter modes.....	92
TABLE 4.8.	Miss types encoded in the Miss Type Register.....	94
TABLE 4.9.	Response State Register encoding	94
TABLE 4.10.	Comparison of recommended versus implemented features	96
TABLE 4.11.	Approximate cost of monitoring components	97
TABLE A.1.	Memory state hit table	103