Analysis of Simultaneous Multithreading Implementations in Current High-Performance Processors

Kamil Kedzierski[1], Francisco J. Cazorla[2], Mateo Valero[1,2]

[1]Departamento de Arquitectura de Computadores,
Universidad Politécnica de Cataluña (UPC), Spain

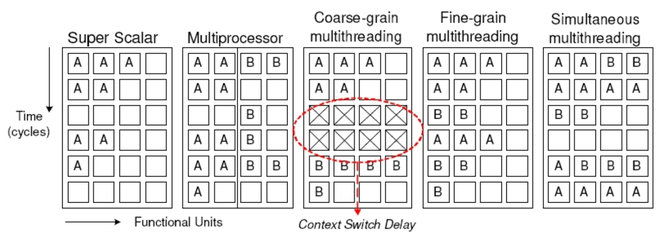
[2]Barcelona Supercomputing Center
Centro Nacional de Supercomputación (BSC), Spain

Introduction

Current superscalar processors take advantage of Instruction Level Parallelism (ILP) from a single thread, which allows them to execute several instructions during a clock cycle.

However, the amount of a parallelism in one thread is limited due to control and data dependencies, what has motivated the research on other forms of parallelism:

- Multiprocessor systems: several threads run in parallel at a given time on different sets of hardware resources, only sharing some levels of the cache hierarchy.
- Coarse-grain systems: the architecture swaps to a different thread when a given thread experiences a long latency event, such as cache miss.
- Fine-grain systems: the context switching occurs more often (in some implementations every clock cycle). In this processor the reason to undertake the context switching may not be necessarily long-latency event.
- Simultaneous multithreading: the only case where the processor is able to issue instructions from the different threads in the same cycle.

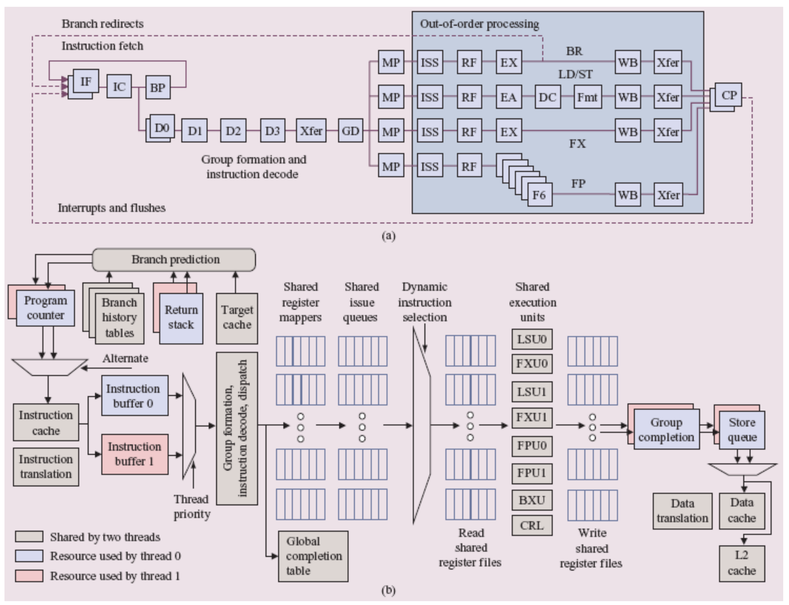


A possible classification of multithreaded architectures [Cazorla2005]

Architectures

What are the differences in SMT implementations?

Power5

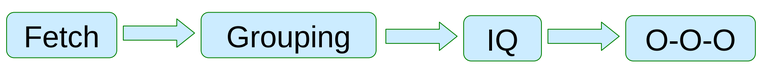


Power5 instruction pipeline (a) and data flow (b) [Sinharoy2005].

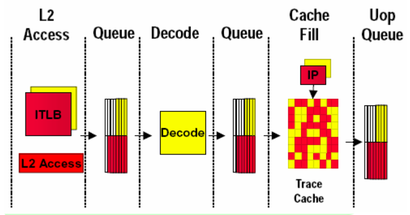
How to implement SMT:
- sharing resources
- partitioning resources
- duplicating resources

Resource	POWER5	Intel Xeon
PC	*duplicated*	*duplicated*
Instruction Cache	*shared*	*shared*
ITLB	*shared*	*duplicated*
DTLB	*shared*	*shared*
BHT	*shared*	*shared*
Return Stack Buffer	*duplicated*	*duplicated*
Decode	*shared*	*shared*
Instruction Buffer/ uCode Queue	*duplicated*	*duplicated*
Group Formation	*shared*	-
Mapping/Rename	*shared*	*duplicated*
IQ	*shared*	*partitioned*
Scheduler	-	*shared*
Register Read	*shared*	*shared*
FUs	*shared*	*shared*
Store Queues	*duplicated*	*partitioned*
Register Write	*shared*	*shared*
GCT/Commit	*duplicated*	*partitioned*

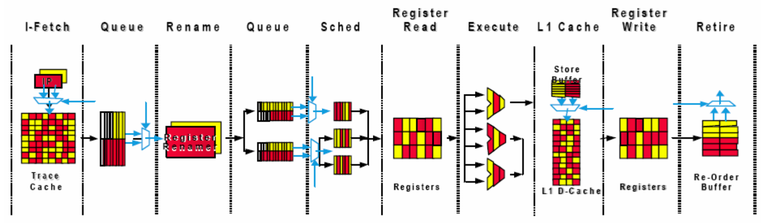
Simplified pipeline view



Pentium 4

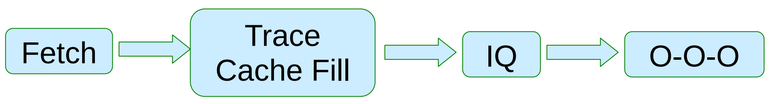


Additional stages in case of cache miss — used to fill Trace Cache [Marr2002].



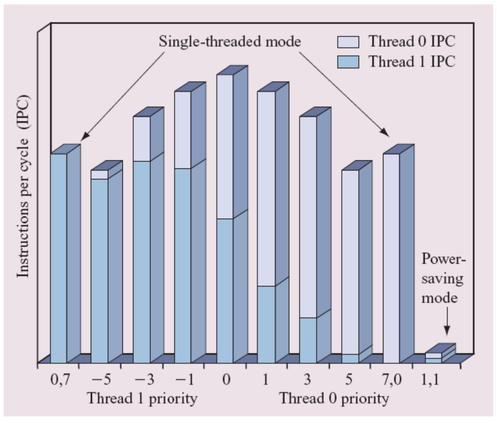
Detailed pipeline view of the Intel Pentium 4 [Burns2002].

Simplified pipeline view



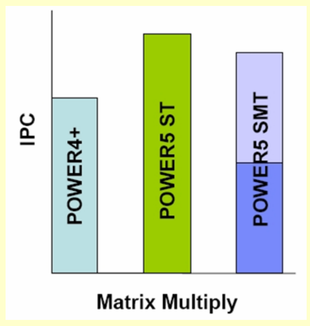
Performance

Power5



Conceptual view of the effects of priority on performance. The performance increase is up to 41% [Sinharoy2005].

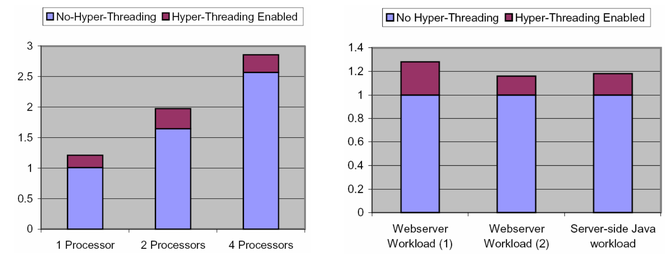
Is SMT always the best choice?



Performance results of the matrix multiply benchmark in Power5 architecture [Kalla2003].

Even beneficial in many cases for some programs SMT is counter productive. Two threads requiring big caches and competing for the same cache memory resource were limiting its performance. As the result, the overall achieved performance was less than in the single-thread mode.

Pentium 4



Average OLTP (left) and 3 different web server (right) benchmarks performance on Xeon machine. The performance gain varies from 16% to 28% [Marr2002].

Conclusions

Simultaneous multithreading is a solution to increase performance, which is commonly used in a processor implementation. The progress of computational machines has reached the point, where the resources in current architectures may not be fully utilized due to data and control dependency in a given workload. Thread Level Parallelism is a good answer on how to increase performance without drastically increasing the resources.

Future work: power analysis of the SMT implementation in Power5-like architecture (IBM's Turandot Simulator).

References

[Kalla2004] R. Kalla, B. Sinharoy, J. M. Tendler, "IBM POWER5 Chip: A Dual-Core Multithreaded Processor", IEEE MICRO 2004, pages 40-47

[Sinharoy2005] B. Sinharoy, R. N. Kalla, J. M. Tendler, R. J. Eickemeyer, and J. B. Joyner "POWER5 system microarchitecture", IBM Journal of Research and Development 2005, pages 505-521

[Mathis2005] H. M. Mathis, A. E. Mericas, J. D. McCalpin, R. J. Eickemeyer, and S. R. Kunkel "Characterization of simultaneous multithreading (SMT) efficiency in POWER5", IBM Journal of Research and Development 2005, pages 555-564

[Marr2002] Deborah T. Marr, Frank Binns, David L. Hill, Glenn Hinton, David A. Koufaty, J. Alan Miller, Michael Upton "Hyper-Threading Technology Architecture and Microarchitecture", Intel Technology Journal, Volume 6, Issue 1, 2002

[Burns2002] David Burns "Pre-Silicon Validation of Hyper-Threading Technology", Intel Technology Journal, Volume 6, Issue 1, 2002

[Kalla2003] Ron Kalla, BalaramSinharoy, Joel Tendler, "Simultaneous Multi-threading Implementation in POWER5", A Symposium on High Performance Chips (HotChips), 19th August 2003,

[Hinton2001] G. Hinton, D. Sager, M. Upton, D. Boggs, D. Carmean, A. Kyker, and P. Roussel. "The microarchitecture of the Pentium 4 processor", Intel Technology Journal, Volume 5, Issue 1, 2001

[Merritt1999] Rick Merritt, "Designers cut fresh paths to parallelism", EETimes http://www.eetimes.com/story/OEG19991008S0014

[Cazorla2005] Francisco J. Cazorla Almeida, "Quality of Service for Simultaneous Multithreading Processors (QoS for SMT Processors)", PhD Thesis, DAC, UPC, 2005

This work has been supported by the Ministry of Science and Technology of Spain under contract TIN-2004-07739-C02-01 and grant AP2005-3776 (Kamil Kedzierski), the HiPEAC European Network of Excellence, an Intel fellowship, and an IBM fellowship.