Proceedings of the 5th WSEAS Int. Conf. on Software Engineering, Parallel and Distributed Systems, Madrid, Spain, February 15-17, 2006 (pp208-210)

A SMT-ARM Simulator and Performance Evaluation

Dong Lan[(1)] Ji Zhenzhou[(1)] Suixiufeng[(1)] Hu Mingzeng[(1)] Cui Guangzuo[(2)]
Dept. of Computer Science and Engineering, Harbin Institution of Technology[(1)]
Harbin 150001, CHINA
Modern Education Center of Beijing University[(2)] Berjing 100086, China

Abstract: - Exponential growth in the number of on-chip transistors with smaller size, make each generation of embedded microprocessors capable to supply more processing ability. In this paper a microarchitecture approach is proposed to make a simultaneous multithreading extension on ARM ISA processor. By exploiting both Instruction Level Parallelism and Thread Level Parallelism, the architecture can be expected to achieve better tradeoff between performance and hardware cost. The organization of the architecture is described. Detailed simulations of microarchitecture show IPC is improved with the multithreading extension of the simplesim-arm architecture greatly, especially for some benchmark combination.

Key-Words: Simultaneous Multithreading, TLP (Thread Level Parallelism), ILP (Instruction Level Parallelism), ARM, Simulator

1 Introduction

Simultaneous multithreading technique (SMT) has been becoming the main stream in the industry of microprocessors [1]. SMT combines the ILP and TLP so as to utilize the hardware resource more efficiently compared with traditional multithreaded processor and Superscalar processor [2]. SMT also has less hardware cost than the comparative multiprocessors. With the improvement of the processing ability of micoprocessors, microarchitecture trends have interesting implications for multi-workload. Future embedded microprocessors will be more prone to be multitask systems especially multithreaded light work environment. Based on basic SMT architecture and simplescalar-ARM simulator, a SMT-ARM simulator is implemented. 6 benchmarks from mibench has been tested for the comparison of IPC. IPC is improved with the increase of the number of thread. Even for two threads combination, IPC is improved above 26% for general benchmarks combination. Because the benchmark and cache utilize feasibility of embedded system are somehow different from the other architectures, the impact of the performance by the Cache configuration are studied in detail in this paper. A quantitative illustration of the cache configuration impact to IPC of this architecture is also discussed here. We analyze the cache configuration impact to IPC from the number of bank, module, associate respectively and monolithically. It can do a help for the configuration choice with a fixed cache size.

2 SMT-ARM Simulator

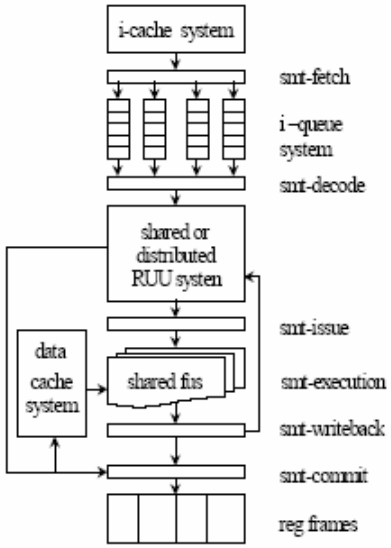

Fig. 1 Microarchitecture of SMT

The architecture is just like SMT architecture in Fig.1[8]. The resource set containing register file, tables and queues, used to keep the context of one thread. The fetch stage fetches instructions from the il1-cache, giving priority to some thread. A technique, called ICOUNT [6], achieved better performance than other simple policies use this algorithm is explored here. From these fetch buffers,

a lot of instructions are decoded and dispatched to reservation stations orderly. From the reservation stations, the instructions are issued to a shared pool of functional units. Each thread has an individual set of registers to store a different context. Many features were inherited from the original Sim-out-of-order simulator [7] and SMT simulator[8], such as out-of–order and speculative execution, branch prediction and register renaming and hardware thread contexts. The simulator is a SMT processor supported ARM ISA.

3. Experiment

The simulation configuration is illustrated in table1 and table2. From Fig.2 It illustrated that IPC of the two threads is improved 26% in benchmarks group1, 30% for group2, and 49% for group3.

Table1 Hardware resource configuration

Ifpsize	8
Inst-l1-cache	32k
Data-l1-cache	64k
Unif-l2-cache	2MB
Int-alu	8
Int-mult	2
Fp-alu	8
Fp-mult	2
Ld/st	1
Ruu-size	32
Lsq-size	12

Table2 Hardware latency

Types of latencies	Number of Cycles	
L1 hit; L2 hit; Tlb miss	1; 6; 30	
L2 miss (for n+1 chunks)	(18+n*2)	
Int-alu functional unit	1	
Fp-alu functional unit	2	
Ld/st functional unit	1	
Int-mul functional unit	Div oper	20
	Mult oper	3
Fp-mul Functional unit	Sqrt oper	24
	Div oper	12
	Mult oper	4

Table 3 Mibench selected for test

Benchmark	Instruction Count Million
Sha	20.6
bitcnts	49.6
Cjpeg	28.1
Pgp	259.2
stringsearch	38.9

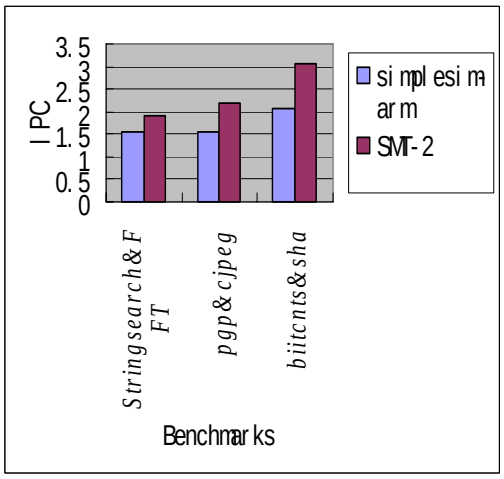

Fig.2 IPC Comparison of simplesim-ARM and SMT-2

4 Cache Configuration Impact to performance

Since embedded processor has its feasibility of application area, the cache configuration impact to system performance may be something different from other processor systems. So experiment is done under the configuration with different cache block size at the same cache size. >From Fig.4 it shows that the IPC is improved with the block size increasing.

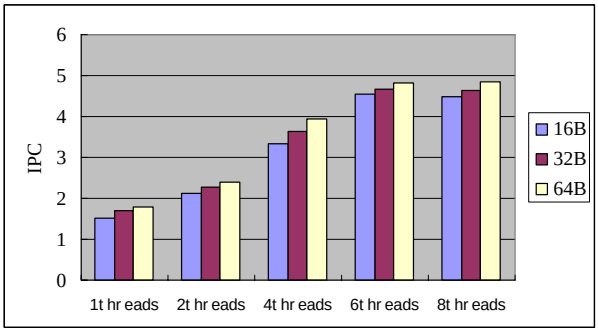

Fig.4 IPC comparison with the block size variables when cache size is 4KB

5 Summary

With the development of electronic technology growth, multithreaded extension to embedded processors is a promising direction in future. With simulator implement of such kind architecture and analyze in this paper, it can be concluded there are

great IPC improvement space making the multithreaded extension to embedded ISA.

References

[1] Theo Ungerer, Borut Robic, Jurij Silc, A survey of processors with explicit multithreading, *ACM Computing Surveys*, Vol.5 No.1, 2003, pp. 29-63

[2] Lo, J. et al, Converting Thread-Level Parallelism to Instruction-Level Parallelism via Simultaneous Multithreading, *ACM Transactions on Computer Systems*, Vol.15,No.3, 1997, pp.322-354

[3] Eric Tune, Rakesh Kumar, D.M.Tullsen,BradCalder, Balanced Multithreading Increasing Throughput via a Low Cost Multithreading Hierarchy, *Proceedings of the 37th annual IEEE/ACM International Symposium on Microarchitecture***,** 2004**,** pp.183 - 194

[4] Rakesh Kumar, D.M. Tullsen, Compiling for instruction cache performance on a multithreaded architecture, *Proceedings of the 35th annual ACM/IEEE international symposium on Microarchitecture*,2002,pp.419-429

[5] Jason Cong, Ashok Jagannathan, Glenn Reinman, Yuval Tamir, Understanding the energy efficiency of SMT and CMP with multiclustering, *Proceedings of the international symposium on Low power electronics and design ISLPED '05*, 2005, pp.48-53

[6] D. Tullsen et al, 'Exploiting Choice: Instruction Fetch and Issue on an Implementable Simultaneous Multithreading Processor', *Proceedings of the 23 rd Annual International Symposium on Computer Architecture*, 1996, pp.191-202

[[7]Burger D., Austin T.M., The SimpleScalar Tool Set, Version2.0, *ACM SIGARCH Computer Architecture*, Vol.25, No.3 ,1997, pp.13-25

[8] Ronaldo Gonçalves, Eduard Ayguadé, Mateo Valero and Philippe Navaux A Simulator for SMT Architectures : Evaluating Instruction Cache Topologies. *In 12th Symposium on Computer Architecture and High Performance Computing*, 2000, pp.279-286