CMP on SoC: Architect’s View

Shuichi Sakai
The University of Tokyo
7-6-1 Hongo Bunkyo-ku
Tokyo 113-8656 JAPAN
TEL. +81-3-5841-6653
sakai@mtl.t.u-tokyo.ac.jp

ABSTRACT

This briefly sketches the current issues of chip multiprocessor (CMP) design for system LSI chip from the viewpoint of computer architects.

Categories and Subject Descriptors

C.1.4 [**Parallel Architecture**]

General Terms: Performance, Design, Reliability, Security

Keywords: CMP (Chip Multiprocessor), parallel processing, SoC (System on Chip), dependability, I/O centric

1. INTRODUCTION

Moore’s Law and the evolution of our IT society accelerate the advancement of System on Chip (SoC) technology. This briefly sketches the current and near future issues of chip multiprocessor (CMP) [1][2] design for system LSI chip from the viewpoint of computer architects.

2. PARADIGM SHIFT

For traditional computer architects, designing multiprocessors was mainly for speedup. It has also been true for designing SoC. However, lots of change occurs because of the various requirements from our society, which causes a paradigm shift in this field.

2.1 Performance

Performance classically means speed (throughput). In parallel processing, speedup proportional to the number of processing elements (PEs) is regarded as ideal. It has been measured by the execution time of some benchmark programs such as SPEC and Linpack. These benchmarks are mainly for measuring the speed of CPU, especially the performance of memory-register-ALU architecture. In SoC, however, we have to pay much attention to the I/Os, since CPUs (especially SoCs) are connected to so many kinds of I/O devices such as cameras, microphones, GPS, Internet, signal I/O devices, pointing devices and secondary storages. So **I/O centric approach** becomes one key for performance.

Another key is **functional distribution**. Even if we use the homogeneous multiprocessors in a single chip, each processor does not necessarily execute a part of a single program, but the different program for performing different functions.

The interface toward the Internet is definitely the central problem.

The third additional issue is **realtimeness** or **a short response time**. Although getting high throughput is not easy, to keep realtimeness is usually more difficult. In real world ubiquitous information processing, realtimeness may become a top priority.

2.2 Power Consumption

No doubt to **lower power consumption** is one of the most significant issues in information systems, especially in PDAs and embedded processors. It is for extending battery-life at one charge longer.

In desktop computers and servers, low power becomes also critical, since the increase of power consumption will soon be beyond the limit for cooling CPU packages.

2.3 Dependability

CPUs must be **dependabl**e, i.e. they should be reliable enough and safe enough for general people to depend on. Reliability mainly means the stability of the system, and the safety mainly means the complete protection from the attacks.

2.4 Design Simplicity

We all are afraid that future processors are too complex to design and test. On the other hand, our society requires a rapid development of the next generation processors. On that condition, we have to take care **simplifying the processor architecture** and shortening the design period.

3. HOW TO ACHIEVE

The issues listed above are not independent of each other; they are tightly related. This section roughly shows how to optimize some of them without disturbing the others, or how to make trade-off of more than one issue.

For performance, of course, to **extract parallelism** and to **equally distribute** it among CPUs are important. **Granularity** should be appropriate: if it is too small, overhead of invoking and finishing threads may affect the whole performance; if it is too large, workloads could not be distributed equally.

In addition, **speculation** is a key technology for gaining performance. **Speculative multithreading** [3] was proposed for it and there are several techniques for efficient speculation [4]. In addition to the control speculation, **value predictions** and **memory speculations** are the new important techniques.

ISSS ’02, October 2-4, 2002, Kyoto, Japan
ACM 1-58113-562-9/02/0010.

Functional distribution should be achieved by two ways: by hardware and by software. For special functions which cannot be performed by usual CPUs, we have to attach special hardware. Otherwise one (or more) of the CPUs of CMP might dedicate itself to it by software. Today's evolving hardware and software enable **software implementation of special functions,** or the general purpose CPU comes to involve special hardware (such as a graphic unit), so in many cases implementation of special functions becomes simpler, i.e., just to attach another CPU for performance.

For power reduction, in addition to the effort from the device side, architectures and compilers also support. To **reduce speculation** and to **make speculation accuracy higher** are two of the key techniques; the others are to **control the number of CPUs** for balancing performance and power consumption.

For security, our CPU has to cope with the virus patterns and invasion patterns adaptively. They might be downloaded as vector data from security management companies. To exploit them, a **daemon processor** might be dedicated to updating the pattern and checking whether we are attacked or not.

For reliability, CMP multiprocessing can be exploited for **duplicate execution**. It will be carried out by cooperation of hardware and software (compilers and operating systems); to make duplicate codes is a compiler work and to assign them to CPUs is an OS work.

For design simplicity, **IPs of a single CPU** might be used for constructing CMPs. However, **interface logic** such as synchronization logic and communication interface logic is needed. To **reduce the architectural design overhead** of it is one problem. Another problem is to reduce time period of changing compilers and operating systems for efficiently exploiting multiple threads.

We should again take care that, in every issue, cooperation of architecture, compilers and operating systems is becoming much more important. Middleware technologies become now also significant, since some IPs are dominant in hardware and software.

4. ACKNOWLEDGMENTS

Our thanks to STARC (Semiconductor Technology Academic Research Center) for supporting our research on next generation low-power CMP.

5. REFERENCES

[1] Olukotun, K., Nayfeh, B., Hammond, L., Wilson, K. and Chang, K. The Case for a Single Chip Multiprocessor, in Proceedings of 7th International Symposium on Architectural Support for Programming Languages and Operating Systems, 2-11, 1996.

[2] Sohi, G.S., Breach, S. and Vijaykumar, T.N. Multiscalar Processors, in Proceedings of 22nd Annual International Symposium on Computer Architecture, 414-425, 1995.

[3] Krishna, V. and Torellas, J. A Chip Multiprocessor Architecture with Speculative Multithreading, IEEE Transactions on Computers, Vlo.48, No.9, 866-880, 1999.

[4] Iwama, C. Barli, N.D., Sakai, S. and Tanaka, H. Improving Conditional Branch Prediction on Speculative Multithreading Architectures, in Proceedings of 7th International Euro-Par Conference, at Manchester, UK, 413-417, 2001.