Improving Energy Efficiency via Speculative Multithreading on MultiCore Processors

Sato, Toshinori
System LSI Research Center, Kyushu University

Tanaka, Yuu
Kyushu Railway Company

Sato, Hidenori
Seiko Epson Corporation

Funaki, Toshimasa
Graduate School of Computer Science and System Engineering, Kyushu Institute of Technology

他

出版情報：16th International Workshop on Power and Timing Modeling, Optimization and Simulation, pp.553-562, 2006-09. Springer
バージョン：
権利関係：

Improving Energy Efficiency via Speculative Multithreading on MultiCore Processors

Toshinori Sato[1], Yuu Tanaka[2], Hidenori Sato[3], Toshimasa Funaki[4], Takenori Koushiro[5], and Akihiro Chiyonobu[4]

[1] System LSI Research Center, Kyushu University, 3-8-33-3F Momochihama, Sawara-ku, Fukuoka, 814-0001 Japan
toshinori.sato@computer.org
http://www.slrc.kyushu-u.ac.jp/~tsato/
[2] Kyushu Railway Company, 3-25-21 Hakataekimae, Hakata-ku, Fukuoka, 812-8566 Japan
yuu-ta@is.naist.jp
[3] Seiko Epson Corporation, 3-3-5 Owa, Suwa, 392-0001 Japan
hide@mickey.ai.kyutech.ac.jp
[4] Graduate School of Computer Science and System Engineering, Kyushu Institute of Technology, 680-4 Kawazu, Iizuka, 820-8502 Japan
{t-funaki,chiyo}@mickey.ai.kyutech.ac.jp
[5] Toshiba Corporation, 1 Komukaitoshiba-cho, Saiwai-ku, Kawasaki, 212-8582 Japan
takenori.koshiro@toshiba.co.jp

Abstract. The advance in semiconductor technologies has increased the number of transistors on a die, resulting in the continuous improvement in microprocessor performance. However, the increase in power consumption and hence in power density is about to stop the progress in microprocessor performance. While supply voltage reduction is commonly known as an effective technique for power savings, it increases gate delay and thus causes performance degradation. The increasing transistors can be utilized for maintaining performance while reducing power consumption. We are considering a speculative multithreaded execution on MultiCore processors. We propose to execute only the part of the program, which has the impact on program execution time, on power-hungry cores. In order to enable this, we divide the instruction stream into two streams. One is called speculation stream, which is the main part of a program and where speculation is applied. It is executed on power-hungry cores. The other is the verification stream, which verifies every speculation. It is executed on low-power cores. The energy consumption is reduced by the decrease in the execution time in the speculation stream and by the low-power execution in the verification stream. We call this technique Contrail architecture. The paper will present the energy efficiency of a Contrail processor based on detailed simulations.

1 Introduction

The current trend towards increasing mobile devices requires high-performance and low-power microprocessors. Generally, high performance and low power conflict with each other and it is very difficult to achieve both high performance and low power simultaneously. While power is already the first-class design constraint in embedded systems, it has also become a limiting factor in general-purpose microprocessors, such used in data centers.

The energy consumed in a microprocessor is the product of its power consumption and execution time. Thus, to reduce energy consumption, we should decrease either or both of them. As commonly known, for CMOS circuits, a power-supply reduction is the most effective way to lower power consumption. However, it increases gate delay, resulting in a slower clock frequency. That means processor performance is diminished. In order to keep transistor switching speed high, it is required that its threshold voltage is proportionally scaled down with the supply voltage. Unfortunately, however, lower threshold voltage leads to increase subthreshold leakage current. Maintaining high transistor switching speeds through low threshold voltage gives rise to a significant amount of leakage power consumption.

In order to achieve both high performance and low power simultaneously, we can exploit parallelism [3]. Two identical circuits are used in order to make each unit to work at half the original frequency while the original throughput is maintained. Since the speed requirement for the circuit becomes half, the supply voltage can be decreased. In this case, the amount of parallelism can be increased to further reduce the total power consumption. MultiCore processors are one of the solutions for high performance and low power and they have been already adopted in embedded microprocessors [6, 12, 21, 24]. Thread level parallelism is utilized for power reduction with maintaining processor performance [6]. In this paper, we propose an energy-efficient speculative MultiCore processor.

2 Contrail Processor Architecture

The advance in semiconductor technologies has increased the number of transistors on a die, as known as Moore's law. We propose to utilize the increasing transistors in order to maintaining processor performance while reducing its power consumption. The key idea is to execute only the part of a program, which has the impact on the program execution time, with high power. We divide the execution of the program into two instruction streams. One is called **speculation stream** and is the main part of the execution. We utilize speculation to skip several regions of the stream to reduce its execution time. In other words, the number of instructions in the speculation stream is reduced from that in the original execution. While it is executed with high power, the small number of instructions and hence the short execution time result in energy reduction. In contrast, the other stream is called **verification stream** and supports the speculation stream by verifying each speculation. Since the verification stream just performs verifications, which will not have much impact on the program

execution time, it can be executed slowly. We reduce the clock frequency and hence the supply voltage delivered to the verification stream. From these considerations, its energy consumption is significantly reduced. We coined this technique Contrail architecture [20].

2.1 Contrail – a Speculative MultiCore Processor

We realize Contrail architecture on a MultiCore processor. Each stream is executed as an independent thread on the MultiCore processor. Each processor core has its own clock and power supply, and works at the variable clock frequency and supply voltage [7, 8, 10]. The speculation stream is executed on a power-hungry core at high clock frequency, and the verification stream is executed on low-power cores at low clock frequency. If misspredictions do not occur frequently, we expect a considerable amount of power reduction. When a misspredict occurs in the speculation stream, it is detected by the verification stream. All threads from the misspredict point to tail, including the speculation stream and any verification streams, are squashed, and processor state is recovered by the verification stream that detects the misspredict. And then, the verification stream becomes the speculation stream. In the cases, the additional power is consumed regarded as misspredict penalties.

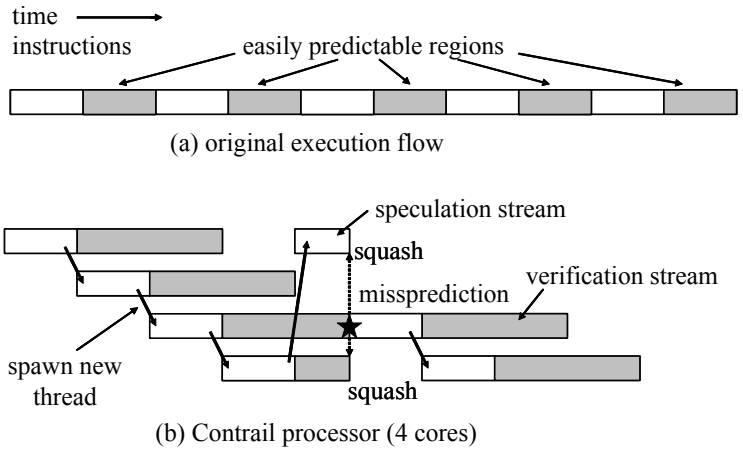

Fig. 1. Execution on a Contrail processor

Fig. 1 explains how a program is executed on a Contrail processor. In this explanation, we assume that half of the regions in the original execution flow is easily predictable and is distributed uniformly as explained in Fig. 1(a). This is a reasonable assumption, since Pilla et al. [17] reported that approximately 60% of dynamic traces can be reused with the help of the value prediction. We also assume the clock frequency for the verification cores at half that for the speculation core. Under these assumptions, the execution is divided into speculation and verification streams in the Contrail processor as depicted in Fig. 1(b). The predicted regions are removed from

the speculation core, and are moved into and hence are executed in the verification cores. Determining trigger points is based on the confidence information obtained from the value predictor. That is, thread partitioning is dynamically performed by hardware without any assistance of compilers. When an easily predictable region is detected, the head thread spawns a new speculation stream on the next core and it turns into a verification stream. That means only one core executes the speculative stream and the other cores execute the verification stream in a distributed manner. One of the possible implementations of the Contrail processor is a ring-connected MultiCore processor such as MultiScalar architecture [9], as shown in Fig. 2. Each verification stream stays alive until all instructions in the corresponding region removed from the speculation stream are executed. After that, the verification core is released for the future speculation stream.

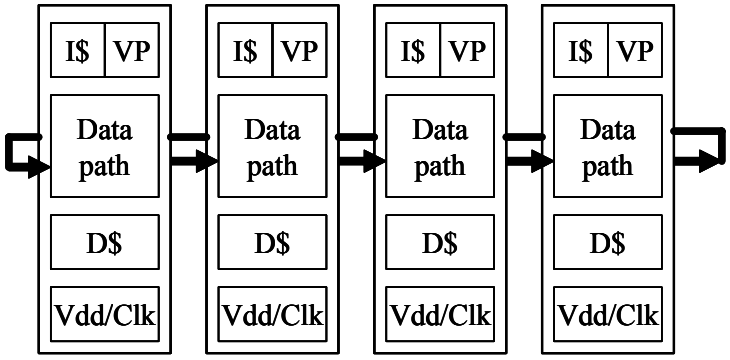

Fig. 2. Contrail processor

We should consider the cost of spawning a new thread. If it is larger than the cost of verifying a value prediction on a single-threaded processor, the MultiCore implementation is not a good choice. However, from the following observations, we determined to adopt the multithreaded MultiCore processor model rather than the single-threaded one. In the single-threaded model, only datapath alternates between high-speed and low-speed modes. The other blocks, especially instruction-supply front-end, should be always in high-speed mode. This reduces the efficiency of the variable voltage and frequency scaling technique. On the other hand, every component of each processor core can alternate two modes in MultiCore model, and thus the improvement in energy efficiency is expected.

2.2 Active Power Reduction

As shown in Fig. 2, each core has its dedicated voltage/frequency controller and value predictor. The potential effect of the Contrail processor architecture on energy-efficiency is estimated as follows: We determine the clock frequency and supply voltage for the verification cores at half that for the speculation core as used in Fig. 1. Here, we will focus on active power, and thus energy consumption is calculated as

follows: For the speculation cores, energy consumption becomes half that of the original execution since the number of instructions is reduced by half. In contrast, for the verification cores, the sum of every execution time remains unchanged since the execution time of each instruction is two times increased while the total number of instructions is reduced by half. Its energy consumption is decreased due to the reduction of the clock frequency and the supply voltage. It is reduced to 1/8 of the original. Thus, the total energy savings is 37.5%. It is true that the energy efficiency of Contrail processors depends on the value prediction accuracy and the size of each region. However, we believe that the potential effect of Contrail processors on energy savings is substantial from the estimate above.

Recent studies regarding power consumption of value predictors find that complex value predictors are power-hungry [1, 16, 19]. One of the solutions for reducing power consumed in value predictors is using simple value predictors such as last-value predictor [15, 19]. However, value prediction is not the only technique for generating speculation stream. Other techniques are probably utilized for this purpose, and the key idea behind the Contrail architecture will be adopted.

2.3 Leakage Power Consideration

While the Contrail processor architecture has good characteristics on active power reduction, its leakage power consumption might be increased since it has multiple cores. As mentioned in the previous sections, only one core has to be fast and the remaining cores can be slow. Thus, we can reduce or even cut the supply voltage for the slow cores. Similarly, the threshold voltage of transistors for the slow cores can be raised, resulting in significant leakage reduction. There are several circuits proposed to reduce leakage current by dynamically raising the threshold voltage, for example by modulating body bias voltage [2, 13, 25]. From these considerations, we expect to keep the leakage power consumed by the Contrail processor comparable to or even smaller than that consumed by a single-core processor.

2.4 Related Works

Speculative multithreading [5, 9, 18, 22, 26] is very popular in microprocessor architecture. One of the differences from the previously proposed speculative multithreaded processors is that the Contrail processor does not require any mechanism to detect memory dependence violations. Since the Contrail processor strongly relies on value prediction, any memory dependence violations cannot occur. Instead, it suffers from value misspredictions. This simplifies hardware complexity. This is because value misspredictions can be detected locally in a core, while detecting memory dependence violations requires a complex mechanism such as ARB or Versioning Cache [9]. Another difference from the previously proposed pre-computing architectures [22, 26] is that the Contrail processor architecture does not rely on redundant execution. In the ideal case, the number of executed instructions is unchanged. Another difference is that its target is the improvement in energy efficiency instead of that in performance.

3 Evaluation

This section explains evaluation methodology, and then presents simulation results.

3.1 Methodology

We implemented the Contrail processor simulator using MASE [14], a derivative of SimpleScalar tool set. Its PISA instruction set architecture (ISA) is based on MIPS ISA. The baseline processor and each core in the Contrail processor are a 2-way out-of-order execution processor. Only fetch bandwidth is 8-instruction wide. The number of cores on the Contrail processor model is 4. In the current simulator, the followings are assumed. Instruction and data caches are ideal. Branch prediction is perfect. Ambiguous memory dependences are perfectly resolved. In contrast, we model a value predictor in details. We use a 2K-entry last-value predictor [15]. For thread spawning policy, the fixed interval partitioning [5] is used, because it does a good job with load balance. The interval is 32 instructions. This value is determined based on the previous study [23]. The overhead on spawning a new thread is 8 cycles.

We evaluate a scaling for supply voltage and clock frequency based on Intel Pentium M processor [10]. The verification cores work at lower frequency and voltage (800MHz, 1.036V), and the speculation core and the baseline processor work at higher frequency and voltage (1.6GHz, 1.484V). Since leakage power strongly depends upon temperature, we should use a pessimistic assumption. We use temperature of 100°C, where the leakage power is equal to the active power [4]. This is a reasonable assumption, since it is reported that the leakage power is comparable to the active power in the future process technologies [2]. The verification cores exploit the body bias technology. It is assumed that the leakage power consumed by the cores, where reverse body bias is applied, is reduced by 2x [2]. And last, we assume that the leakage power consumed by idle cores is negligible.

We use 11 programs from SPEC2000 for evaluating general-purpose applications, 13 programs from MediaBench for multimedia application, and 3 programs from MiBench [11] for embedded applications.

3.2 Results

We only show the average of simulation results for three benchmark suites, respectively, due to the lack of space.

Fig. 3 presents execution cycles of the Contrail processor, which are relative to those of the baseline single-core processor. Last-value predictors have only a few contributions on single-core processor performance [15]. This is same for the Contrail processor, while performance improvement is not the goal of this architecture. The combination of value prediction and multithreading architecture achieves the improvement of around 10% in performance.

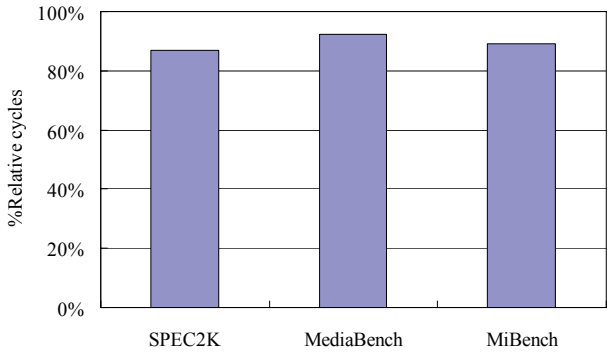

Fig. 3. Relative execution cycles

Fig. 4 presents the relative power consumption. Each bar is divided into two parts. The lower and upper parts indicate the average active and leakage power, respectively. As you can see, power consumption is slightly increased. This is because the execution cycle is reduced by speculation. From Figs 3 and 4, it is observed the cycle reduction rate is larger than the power increase rate. Thus, energy consumption is reduced. The results are very different from those for other low power architectures, which achieve power reduction at the cost of performance loss.

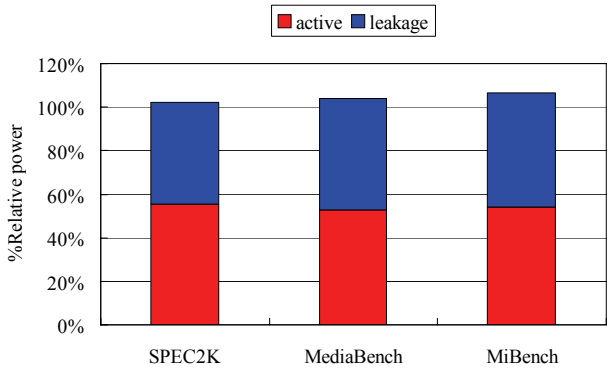

Fig. 4. Relative power consumption

Fig.5 shows Energy-Delay^2 product (ED^{2}P). The vertical line indicates ED^{2}P of the Contrail processor relative to that of the baseline single-core processor. It is observed that the improvement of 23% in ED^{2}P is achieved on average. As mentioned above, the misspredicition penalties are included in the results.

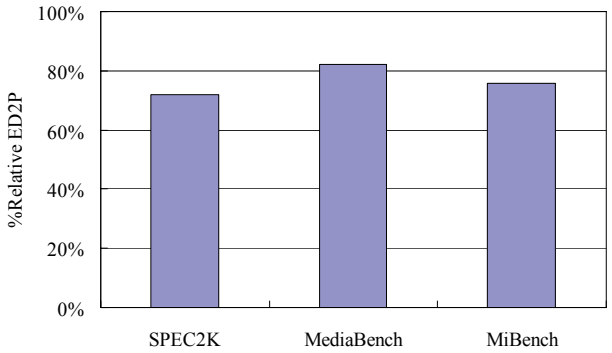

Fig. 5. Relative energy-delay^2 product

Fig. 3 shows that the Contrail processor achieves performance improvement, which is not our primary goal. Since performance improvement is not required, we can still slow down clock frequency, resulting in further power reduction. This is a hierarchical frequency scaling. A global control signal uniformly throttles every local clock, which originally has two modes; high-speed mode for the speculation core and the low-speed one for verification cores. Since the hierarchical change in supply voltage will be difficult to implement, we only change clock frequency. Under this scenario, power consumption is reduced as shown in Fig. 6, if we could ideally control the global clock. While this technique does not affect energy, the power reduction is desirable for temperature awareness.

4 Conclusions

The current trend of the advance in semiconductor technologies will impose the diminishing return in single-thread performance since a huge amount of the increase in power consumption and hence in power density is predicted. Multithreading and dual-power functional units will be promising techniques to reduce energy in the future microprocessors [18]. We proposed such an energy-efficient speculative MultiCore processor, which we call Contrail processor. It exploits thread level parallelism, resulting in mitigating performance loss caused by the supply voltage reduction. Only the part of the program, which has the impact on program execution time, is executed on power-hungry cores. From the detailed simulations, we found that the Contrail processor achieves approximately 23% of ED^{2}P savings while processor performance is slightly improved.

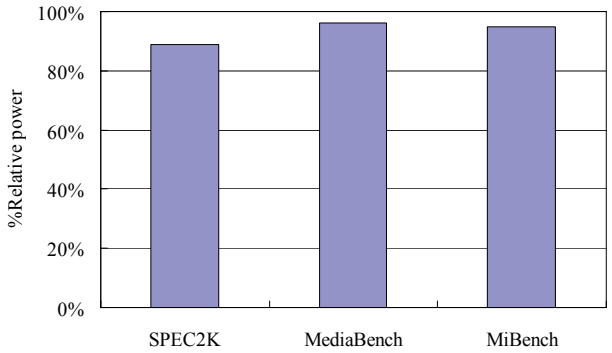

Fig. 6. Further power reduction via frequency control

Acknowledgements

This work is partially supported by Grants-in-Aid for Scientific Research #16300019 and #176549 from Japan Society for the Promotion of Science.

References

1. Bhargava, R., John, L.: Latency and energy aware value prediction for high-frequency processors, 16th International Conference on Supercomputing (June 2002)
2. Borkar, S.: Microarchitecture and design challenges for gigascale integration, 37th International Symposium on Microarchitecture, Keynote (December 2004)
3. Chandrakasan, A.P., Brodersen, R.W.: Minimizing power consumption in digital CMOS circuits, Proceedings of IEEE, 83(4) (April 1995)
4. Chaparro, P., Gonzalez, J., Gonzalez, A.: Thermal-effective clustered microarchitecture, 1st Workshop on Temperature Aware Computer System (June2004)
5. Codrescu, L., Wills, D.: On dynamic speculative thread partitioning and the MEM-slicing algorithm, 8th International Conference on Parallel Architectures and Compilation Techniques (October 1999)
6. Edahiro, M., Matsushita, S., Yamashina, M., Nishi, N.: A single-chip multi-processor for smart terminals, IEEE Micro, 20(4) (July-August 2000)
7. Fleischmann, M.: LongRun power management, white paper, Transmeta Corporation (January 2001)
8. Flynn, D.: Intelligent energy management: an SoC design based on ARM926EJ-S, 15th Hot Chips (August 2003)
9. Franklin, M.: Multiscalar processors, Kluwer Academic Publishers (2003)
10. Gochman, S., Ronen, R., Anati, I., Berkovits, A., Kurts, T., Naveh, A., Saeed, A., Sperber, Z., Valentine, R.C.: The Intel Pentium M processor: microarchitecture and performance, Intel Technology Journal, 7(2) (May 2003)

11. Guthaus, M.R., Ringenberg, J.S., Ernst, D., Austin, T.M., Mudge, T., Brown, B.: MiBench: a free, commercially representative embedded benchmark suite, 4th Workshop on Workload Characterization (December 2001)
12. Kaneko, S., Sawai, K., Masui, N., Ishimi, K., Itou, T., Satou, M., Kondo, H., Okumura, N., Takata, Y., Takata, H., Sakugawa, M., Higuchi, T., Ohtani, S., Sakamoto, K., Ishikawa, N., Nakajima, M., Iwata, S., Hayase, K., Nakano, S., Nakazawa, S., Tomisawa, O., Shimizu, T.: A 600 MHz single-chip multiprocessor with 4.8 GB/s internal shared pipelined bus and 512 kB internal memory, International Solid State Circuits Conference (February 2005)
13. Kuroda, T., Fujita, T., Mita, S., Nagamatsu, T., Yoshioka, S., Sano, F., Norishima, M., Murota, M., Kato, M., Kinugasa, M., Kakumu, M., Sakurai, T.: A 0.9V, 150MHz, 10mW, 4mm^2, 2-D discrete cosine transform core processor with variable-threshold-voltage scheme, International Solid State Circuit Conference (February 1996)
14. Larson, E., Chatterjee, S., Austin, T.: MASE: a novel infrastructure for detailed microarchitectural modeling International Symposium on Performance Analysis of Systems and Software (November 2001)
15. Lipasti, M.H., Wilkerson, C.B., Shen, J.P.: Value locality and load value prediction, 7th International Conference on Architectural Support for Programming Languages and Operation Systems (October 1996)
16. Moreno, R., Pinuel, L., Del-Pino, S., Tirado, F.: A power perspective of value speculation for superscalar microprocessors, 19th International Conference on Computer Design (September 2000)
17. Pilla, M.L., da Costa, A.T., Franca, F.M.G.. Navaux, P.O.A.: Predicting trace inputs with dynamic trace memoization: determining speedup upper bounds, 10th International Conference on Parallel Architectures and Compilation Techniques, WiP session (September 2001)
18. Rattner, J.: Electronics in the Internet age, 10th International Conference on Parallel Architectures and Compilation Techniques, Keynote (September 2001)
19. Sam, N.B, Burtscher, M.: On the energy-efficiency of speculative hardware, International Conference on Computing Frontiers (May 2005)
20. Sato, T., Arita, I.: Contrail processors for converting high-performance into energy-efficiency, 10th International Conference on Parallel Architectures and Compilation Techniques, WiP session (September 2001)
21. Shiota, T., Kawasaki, K., Kawabe, Y., Shibamoto, W., Sato, A., Hashimoto, T., Hayakawa, F., Tago, S., Okano, H., Nakamura, Y., Miyake, H., Suga, A., Takahashi, H.: A 51.2GOPS, 1.0GB/s-DMA single-chip multi-processor integrating quadruple 8-way VLIW processors, International Solid State Circuits Conference (February 2005)
22. Sundaramoorthy, K., Purser, Z., Rotenberg, E.: Slipstream processors: improving both performance and fault tolerance, 9th International Conference on Architectural Support for Programming Languages and Operating Systems (November 2000)
23. Tanaka, Y., Sato, T., Koushiro, T.: The potential in energy efficiency of a speculative chip-multiprocessor, 16th Symposium on Parallelism in Algorithms and Architectures (June 2004)
24. Torii, S., Suzuki, S., Tomonaga, H., Tokue, T., Sakai, J., Suzuki, N., Murakami, K., Hiraga, T., Shigemoto, K., Tatebe, Y., Obuchi, E., Kayama, N., Edahiro, E., Kusano, T., Nishi, N.: A 600MIPS 120mW 70A leakage triple-CPU mobile application processor chip, International Solid State Circuits Conference (February 2005)
25. Transmeta Corporation: LongRun2 technology, `http://www.transmeta.com/longrun2/`
26. Zilles, C., Sohi, G.S.: Master/slave speculative parallelization, 35th International Symposium on Microarchitecture (November 2002)