A Preliminary Evaluation on Energy Effciency of a Temperature-aware Multicore-processor

Hidenori Sato
Seiko Epson Corporation
hide@mickey.ai.kyutech.ac.jp

Toshinori Sato
PRESTO, JST
toshinori.sato@computer.org

Abstract— While lower supply voltage is effective for energy reduction, it causes performance loss. To mitigate the loss, we propose to execute only the part, which does not have any influence on execution speed, with low-speed. We are investigating a multithreaded execution, named Contrail Architecture, which divides an instruction stream into two streams. One is the speculation stream, which is the main part of a program and where speculation is applied. The other is the verification stream, which verifies every speculation. The energy consumption is reduced by the decrease in the execution time in the speculation stream and by the low-speed execution in the verification stream. The paper will present our preliminary evaluation on energy efficiency of a Contrail processor.

I. Introduction

The increasing popularity of portable and mobile computer platforms such as laptop PCs and smart cell phones is a driving force in the investigation of high-performance and power-efficient microprocessors. As the computing power of microprocessors for mobile devices increases, however, their power consumption also increases. While power is already a major design constraint in the area of mobile and embedded computer platforms, it has also become a limiting factor in general-purpose microprocessors. Multicore-processor architecture is one of the solutions and it has been already adopted in embedded microprocessors[6, 13, 25, 28].

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. Power consumption in a CMOS digital circuit is governed by the equation:

$$P = P_{active} + P_{leakage} \tag{1}$$

where P_{active} is active power and $P_{leakage}$ is leakage power. The active power P_{active} and gate delay t_{pd} are given by

$$P_{active} \propto f C_{load} V_{dd}^2 \tag{2}$$

$$t_{pd} \propto \frac{V_{dd}}{(V_{dd} - V_t)^\alpha} \tag{3}$$

where f is the clock frequency, C_{load} is the load capacitance, V_{dd} is the supply voltage, and V_t is the threshold voltage of the device. α is a factor dependent upon the carrier velocity saturation and is around 1.5 in modern CMOS technology. Based on Eq.(2), we can easily find that a power-supply reduction is the most effective way to lower power consumption. However, Eq.(3) tells us that reductions in the supply voltage increase gate delay, resulting in a slower clock frequency, and thus diminishing the computing performance of the microprocessor. In order to maintain high transistor switching speeds, it is required that the threshold voltage is proportionally scaled down with the supply voltage.

On the other hand, the leakage power can be given by

$$P_{leakage} = I_{leakage} V_{dd} \tag{4}$$

where $I_{leakage}$ is the leakage current. The subthreshold leakage current $I_{leakage}$ is dominated by threshold voltage V_t in the following equation:

$$I_{leakage} \propto 10^{-\frac{V_t}{S}} \tag{5}$$

where S is the subthreshold swing parameter. Thus, lower threshold voltage leads to increase subthreshold leakage current and power. Maintaining high transistor switching speeds via low threshold voltage

gives rise to a significant amount of leakage power consumption.

In order to achieve high performance and low power, 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. Another kind of parallelism, which is thread level parallelism, is utilized for energy reduction with maintaining processor performance[6]. In this paper, we propose an energy-efficient speculative multicore-processor.

II. Contrail Processor Architecture

To reduce the energy consumption, we divide an execution of an application into two streams. One is called the *speculation stream* and consists of the main part of the execution. However, it utilizes speculation to skip several regions of the execution. In other words, the number of instructions in the speculation stream is smaller than that in the original execution, resulting in energy reduction. In contrast, the other stream is called the *verification stream* and supports the speculation stream by verifying each speculation. The key idea is that the speculative execution in the original program translates its each critical path into a non-critical one and we move it from the speculation stream into the verification stream. Hence, the verification stream can execute slowly if the speculation is almost successful. We can reduce the clock frequency of the components for the verification stream. Furthermore, the supply voltage is also reduced. From these considerations, its energy consumption is significantly reduced. We coined this technique *Contrail* architecture[23].

A. Multicore-processor model

Each stream executes as a thread on a multicore-processor, each processing element (PE) of which independently works at the variable clock frequency and supply voltage[7, 8, 10]. The speculation stream is executed on a PE in high-speed mode and the verification stream is executed on other PEs in low-speed mode. In the ideal case, that means there are no mispredictions; the speculation stream finishes silently and waits for the verification process. In the case in which a misprediction is detected in a verification stream, all threads from the misprediction point to tail, including the speculation stream and any verification streams, are squashed, and processor state is recovered by the verification stream that detect the misprediction. And then, the verification stream becomes the speculation stream.

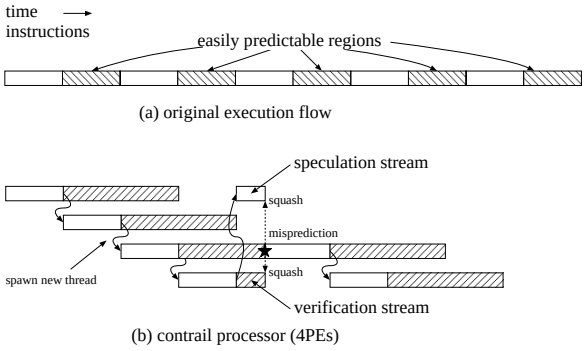

FIG. 1 Execution on a Contrail processor

We explain how the program is logically executed on a Contrail processor, using Figure 1. We assume that half of the original execution of an application is ideally predicted and is distributed uniformly as explained in Figure 1(a). This is a reasonable assumption, since it has been reported that 59% of dynamic traces can be reused with the help of the value prediction[20]. We also assume the clock frequency for the verification PEs at half that for the speculation PE. Under these assumptions, the execution is divided into speculation and verification streams in a Contrail processor in a distributed manner as depicted in Figure 1(b). The predicted regions are skipped in the speculation stream and execute in the verification streams while enlarging their execution time. Determining trigger points is based on the confidence information obtained from the value predictor. When an easily predictable region is detected, the head thread spawns a new speculation stream on the next PE and it turns into a verification stream. Thus, the Contrail processor is logically build as a ring-connected multicore-processor such as MultiScalar architecture[9], as shown in Figure 2. Physical interconnects do not necessarily follow the logical topology. Each verification stream stays alive until all instructions in the corresponding region removed from the speculation stream are executed. After that, the verification PE is released for a future speculation stream.

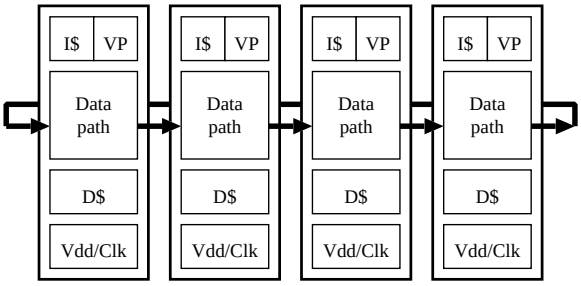

FIG. 2 CONTRAIL PROCESSOR

The cost of spawning a new thread might be larger than that of verifying a value prediction on a single-threaded processor. However, in single-threaded processor 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 effect of the variable frequency and voltage control technique. On the other hand, every component of each processor core can alternate two modes in multicore model, and thus improving energy efficiency is expected. From these observations, we determined to adopt multi-threaded model rather than single-threaded model.

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 mispredictions. This simplifies hardware complexity. This is because value mispredictions can be detected locally in an PE, 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[26, 30] 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.

B. Active power reduction

As shown in Figure 2, each PE 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 PEs at half that for the speculation PE. When we focus on active power, energy consumption is calculated as follows: For the speculation stream, energy consumption becomes half that of the original execution since the number of instructions is reduced by half. In contrast, for the verification streams, the sum of every execution time remains unchanged since the execution time of each instruction increases doubly while the total number of instructions is reduced by half. Its energy consumption is decreased by the reduction of the clock frequency and the supply voltage. Based on Eq.(2), it is reduced to $\frac{1}{8}$. 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 predicted region. However, we believe that the potential effect of Contrail processors on energy savings is substantial.

Recent studies regarding power consumption of value predictors found that complex value predictors are power-hungry[1, 19, 22]. One of the solutions for reducing power consumed in value predictors is using simple value predictors such as last-value predictor[17, 22]. 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 applied.

C. Temperature awareness

The Contrail processor has a good characteristic on temperature awareness[24]. Figure 3 is a bird's-eye view showing how the speculation and verification streams are executed on a Conrail processor chip consisting of four processor cores. As you can see, there is only one hot core and it is rotating. This feature resembles the activity migration[12] and the cluster hopping[4]. Because hot spots are rotating, heat is diffused over the chip and its temperature will be down.

D. Leakage power consideration

While the Contrail processor architecture has good characteristics on active power reduction and temperature awareness, 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

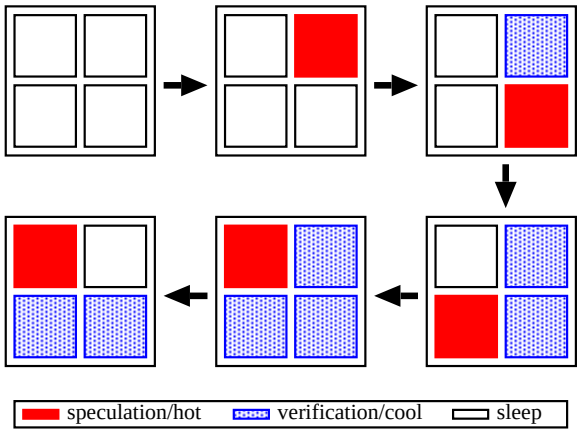

FIG. 3 PE ROTATION

for the slow cores. Similarly, the threshold voltage of transistors for the slow cores can be raised, resulting in 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, 14, 29]. Leakage power strongly depends upon temperature. Since it is expected that the Contrail processor reduces chip temperature, its leakage power will be also reduced. From these considerations, the Contrail processor's leakage power consumption will be comparable to or even smaller than that of a single-core processor.

III. EVALUATION

In our previous study[27], we evaluated the potential of the Contrail processor on energy efficiency under the assumption of perfect value prediction. In this paper, we consider penalties due to value mispredictions.

A. Methodology

We implemented the Contrail processor simulator using MASE/SimpleScalar/PISA tool set[15]. Its instruction set architecture (ISA) is based on MIPS ISA. The baseline processor and each PE are a 2-way out-of-order execution processor. Only fetch bandwidth is 8-instruction wide. The number of PEs 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. We use a 2K-entry last-value predictor[17]. For thread spawning policy, the fixed interval partitioning[5] is used, because it does a good job with load balance. Considering other partitioning policies[18] is remained for the future study. The interval is 32 instructions. This value is determined based on the previous study[27]. The overhead on spawning a new thread is 8 cycles.

We evaluate a scaling for supply voltage and clock frequency based on SAMSUN's ARM processor[16]. The verification PEs work at lower frequency and voltage (600MHz, 0.7V), and the speculation PE and the baseline processor work at higher frequency and voltage (1.2GHz, 1.1V). As mentioned above, leakage power strongly depends upon temperature. Since evaluation here is very preliminary, 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 PEs exploit the body bias technology. It is assumed that the leakage power consumed by the PEs, where reverse body bias is applied, is reduced by 2×[2]. And last, we assume that the leakage power consumed by idle PEs is negligible.

We use CRC, FFT, and StringSearch from MiBench suite[11] for this evaluation, because our primary interests are on embedded applications. MiBench is developed for use in the context of embedded, multimedia, and communications applications. It contains image processing, communications, and DSP applications. We use original input files provided by University of Michigan. All programs are compiled by the GNU GCC with the optimization options specified by University of Michigan. Each program is executed to completion.

B. Results

Figure 4 shows how the last-value predictor works. Each bar is divided into three parts. The bottom part indicates the percentage of instructions whose value is correctly predicted. The middle part indicates the percentage that is mispredicted. And the top part indicates the percentage that is not predicted due to low confidence. It is observed that the characteristics strongly depend upon programs. The only thing we can find from the figure is that the percentage of misprediction is small.

Figure 5 shows the execution cycle relative to that of the baseline single-core processor. We have al-

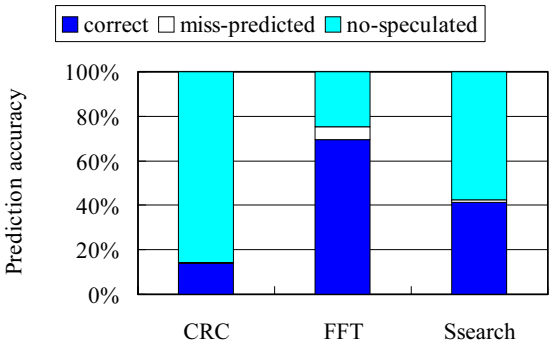

FIG. 4 %VALUE PREDICTION ACCURACY

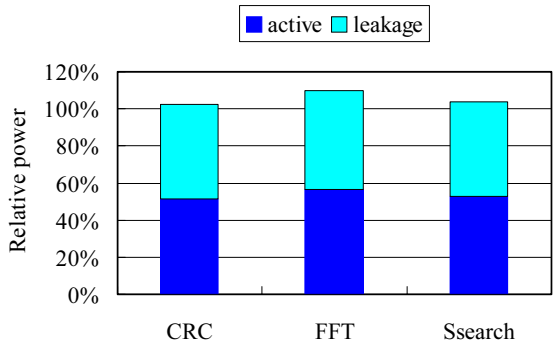

FIG. 6 RELATIVE POWER CONSUMPTION

ready know that the last-value predictor has only a few contributions on single-core processor performance. The 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.

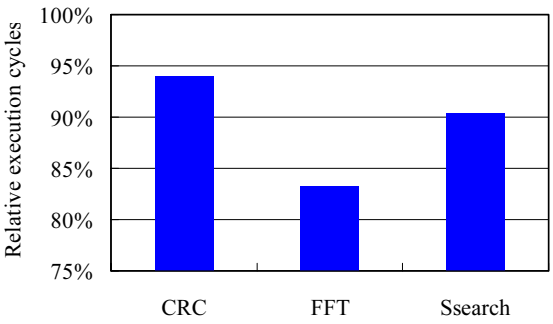

FIG. 5 RELATIVE EXECUTION CYCLES

Figure 6 shows the relative power consumption. Each bar is divided into three parts. The lower and upper parts indicate the average active and leakage power, respectively. It should be noted that power consumed by the value predictor is not considered. The consideration is remained for the future study. As you can see, power consumption is slightly increased. This is because the execution cycle is reduced by speculation. From Figures 5 and 6, 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.

Figure 7 shows Energy-Delay^2 product (ED^{2}P). ED^{2}P is a good metric for evaluating tradeoff in performance and energy. The vertical line indicates ED^{2}P relative to that of the baseline single-core processor. It is observed that the improvement of 25% in ED^{2}P is achieved on average. As mentioned above, the misprediction penalties are included in the results. We have seen from Figure 4 that the percentage of correct prediction is small. In addition, we have seen from Figure 6 that power consumption is increased. Nonetheless, the Contrail processor improves energy efficiency. This is a notable characteristic of this architecture.

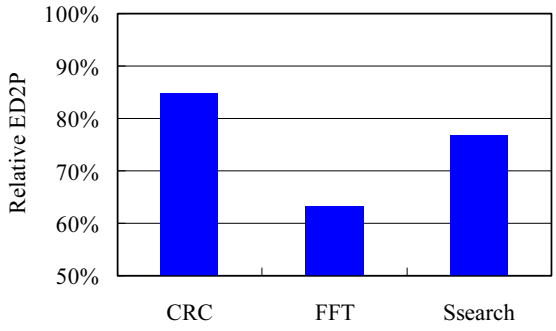

FIG. 7 RELATIVE ENERGY-DELAY^2 PRODUCT

As we have already seen in Figure 5, the Contrail processor achieves performance improvement, which is not its primary target. If performance improvement is not required, we slow down clock frequency and further reduce power. This is hierarchical frequency control. Global control signal uniformly throttles every local clock, which originally has two modes; high-speed mode for the speculation PE and the low-speed one for verification PEs. Since hierarchically changing supply voltage will be difficult to implement, we only change clock frequency.

Under this scenario, power consumption is reduced as shown in Figure 8 if we could ideally control the global clock. While this technique does not affect energy, the power reduction is desirable for temperature awareness.

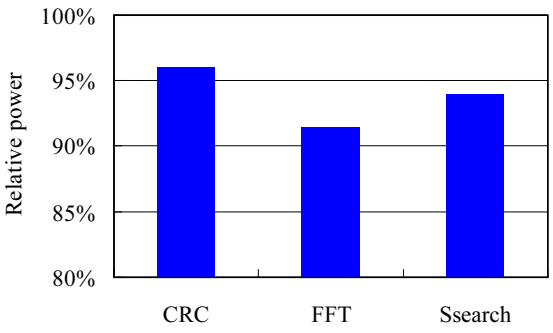

FIG. 8 Power reduction via frequency control

IV. Summary

It is expected that multithreading and dual-power functional units are key techniques for energy reduction[21]. We have proposed such an energy-efficient speculative multicore-processor. Our proposed architecture exploits thread level parallelism, resulting in mitigating performance loss caused by the supply voltage reduction. In this paper, we show preliminary evaluation results of the Contrail processor. We found that the Contrail processor has a potential of approximately 25% ED^2P savings while processor performance is slightly improved.

References

[1] R. Bhargava, et al., "Latency and energy aware value prediction for high-frequency processors," 16th Int. Conf. on Supercomputing, 2002.

[2] S. Borkar, "Microarchitecture and design challenges for gigascale integration," 37th Int. Symp. on Microarchitecture, Keynote, 2004.

[3] A. P. Chandrakasan, et al., "Minimizing power consumption in digital CMOS circuits," Proc. IEEE, 83(4), 1995.

[4] P. Chaparro, et al., "Thermal-effective clustered microarchitecture," 1st Workshop on Temperature Aware Computer System, 2004.

[5] L. Codrescu, et al., "On dynamic speculative thread partitioning and the MEM-slicing algorithm," 8th Int. Conf. on Parallel Architectures and Compilation Techniques, 1999.

[6] M. Edahiro, et al., "A single-chip multi-processor for smart terminals," IEEE Micro, 20(4), 2000.

[7] M. Fleischmann, "LongRun power management," white paper, Transmeta Corporation, 2001.

[8] D. Flynn, "Intelligent energy management: an SoC design based on ARM926EJ-S," 15th Hot Chips, 2003.

[9] M. Franklin, "Multiscalar processors," Kluwer Academic Publishers, 2003.

[10] S. Gochman, et al., "The Intel Pentium M processor: microarchitecture and performance," Intel Tech. Jour., 7(2), 2003.

[11] M. R. Guthaus, et al., "MiBench: a free, commercially representative embedded benchmark suite," 4th Workshop on Workload Characterization, 2001.

[12] S. Heo, et al., "Reducing power density through activity migration," Int. Symp. on Low Power Electronics and Design, 2003.

[13] S. Kaneko, et al., "A 600 MHz single-chip multiprocessor with 4.8 GB/s internal shared pipelined bus and 512 kB internal memory," Int. Solid State Circuits Conf., 2003.

[14] T. Kuroda, et al., "A 0.9V, 150MHz, 10mW, 4mm^2, 2-D discrete cosine transform core processor with variable-threshold-voltage scheme," Int. Solid State Circuit Conf., 1996.

[15] E. Larson, et al., "MASE: A novel infrastructure for detailed microarchitectural modeling," Int. Symp. on Performance Analysis of Systems and Software, 2001.

[16] M. Levy, "Samsung twists ARM past 1GHz," Microprocessor report, 16(10), 2002.

[17] M. H. Lipasti, et al., "Value locality and load value prediction", 7th Int. Conf. on Architectural Support for Programming Languages and Operation Systems, 1996.

[18] P. Marcuello, et al., "Thread partitioning and value prediction for exploiting speculative thread-level parallelism," IEEE Trans. Comput., 53(2), 2004.

[19] R. Moreno, et al., "A power perspective of value speculation for superscalar microprocessors," 18th Int. Conf. on Computer Design, 2000.

[20] M. L. Pilla, et al., "Predicting trace inputs with dynamic trace memoization: determining speedup upper bounds," 10th Int. Conf. on Parallel Architectures and Compilation Techniques, Work in Progress, 2001.

[21] J.Rattner, "Electronics in the Internet age," 10th Int. Conf. on Parallel Architectures and Compilation Techniques, Keynote, 2001.

[22] N. B. Sam, et al., "On the energy-efficiency of speculative hardware," Int. Conf. on Computing Frontiers, 2005.

[23] T. Sato, et al., "Contrail processors for converting high-performance into energy-efficiency," 10th Int. Conf. on Parallel Architectures and Compilation Techniques, Work in Progress, 2001.

[24] T.Sato, "Future directions of processor architecture," 6th Int. Workshop on Innovative Architecture for Future Generation High-Performance Processors and Systems, Panel, `http:www.mickey.ai.kyutech.ac.jp/~tsato/cosmos/papers/iwia03_panel.pdf`, 2003.

[25] T. Shiota, et al., "A 51.2GOPS, 1.0GB/s-DMA single-chip multi-processor integrating quadruple 8-way VLIW processors," Int. Solid State Circuits Conf., 2005.

[26] K. Sundaramoorthy, et al., "Slipstream processors: improving both performance and fault tolerance" 9th Int. Conf. on Architectural Support for Programming Languages and Operating Systems, 2000.

[27] Y. Tanaka, ey al., "The potential in energy efficiency of a speculative chip-multiprocessor," 16th Symp. on Parallelism in Algorithms and Architectures, 2004.

[28] S. Torii, et al., "A 600MIPS 120mW 70A leakage triple-CPU mobile application processor chip," Int. Solid State Circuits Conf., 2005.

[29] Transmeta Corp., "LongRun2 technology," `http://www.transmeta.com/longrun2/`.

[30] C. Zilles, et al., "Master/slave speculative parallelization," 35th Int. Symp. on Microarchitecture, 2002.