Power and Performance Advantages of the Highly Clustered Microarchitecture

Yukinori Sato, Ken-ichi Suzuki and Tadao Nakamura
Graduate School of Information Sciences, Tohoku University
6-6-01, Aramaki Aza Aoba, Aoba-ku, Sendai, 981-8579, Japan
{yukinori,suzuki,nakamura}@archi.is.tohoku.ac.jp

Abstract

As the enlarged and multi-ported register file results in high power consumption and slow access time, it is difficult to design high performance superscalar processors. Partitioning and clustering the enlarged and multi-ported register file into several smaller and less-ported register files is expected to overcome the register file issues. The much more a monolithic register file is partitioned into, the lower power and faster access register files can be realized. However, the partitioning causes losses of IPC (instructions per clock cycle) due to communication among partitioned structures. In this paper, we investigate appropriate degrees of partitioning in 3-way, 4-way, 6-way and 8-way clustered microarchitectures. The results show that highly clustered register files make their access time and energy dissipation significantly lower than conventional monolithic register files. If we assume that the access time for a RegFile decides the operating frequency of a processor, we find that highly clustered microarchitectures can achieve both higher performance and lower power. In 8-way configurations, a highly clustered configuration can achieve 1.59 times higher IPS and 83% lower power than a non-partitioned register file configuration.

1. Introduction

Microprocessor designers strive to increase instruction issue width because wider issue width enables a processor to exploit more ILP (instruction level parallelism) in programs. However, instruction issue width for integer functional units has saturated at three or four in current commercial high performance microprocessors [8]. In current high-performance processor design methodology using a single monolithic register file (RegFile), simultaneous reads and writes for a register are required. As instruction issue width of a processor is widened, the number of simultaneous accesses for a register increases. To accommodate the increase of simultaneous access, the number of ports of registers is increased. Not only the number of ports but also the number of registers in a RegFile is increased in order to accommodate the large number of in-flight instructions.

Since power consumption and access time are mainly dependent on the number of registers and the number of ports, the enlarged and multiported RegFile consumes huge power and slows its access time [2, 5]. The huge power consumption of RegFile is becoming a serious restriction of building high performance microprocessors [21]. The slow access time for the RegFile is becoming the critical path in the cycle time of a microprocessor, and degrades the operating frequency of the processor.

A natural way to deal with the RegFile issues is to build clustered microarchitectures [1, 13, 14]. In the clustered microarchitectures, the global structures of conventional superscalar processors are partitioned into simple smaller structures and each of them is arranged in a PE (processing element), which is also called a cluster in some papers [14]. Therefore, the single monolithic RegFile of superscalar processors is partitioned and clustered into smaller and less-ported RegFiles.

However, IPC (instructions per clock cycle) of clustered microarchitectures is lower than those of the non-clustered ones because the clustering causes extra communication delay among PEs. The amount of communication among PEs is dependent on a degree of partitioning; in other words, how many PEs an original datapath is partitioned into. Therefore, as a degree of partitioning is increased, the IPC is decreased due to communication.

On the other hand, if a degree of partitioning is increased, the much more number of entries and ports of the RegFile can be reduced. Therefore, the highly partitioning will be able to make the less power consumption and the faster access RegFile possible. Because the IPC, RegFile access time and RegFile power consumption are dependent on the degree of partitioning, the appropriate degree of partitioning, in terms of the performance and power consumption, should be investigated.

In this paper, we attempt to show a relationship between degrees of partitioning, performance and power consump-

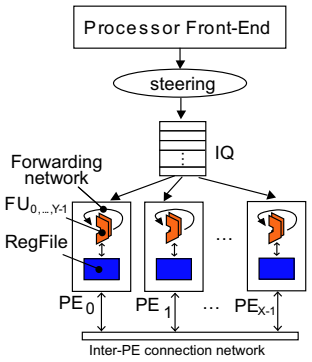


Figure 1. The overview of the clustered microarchitecture (X*Y configuration).

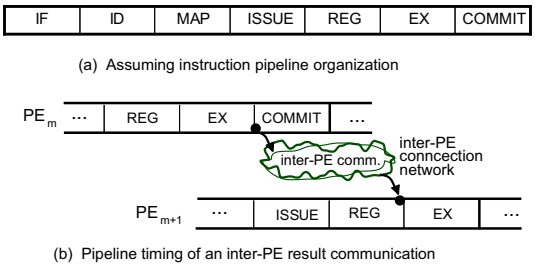


Figure 2. The timing of the pipeline.

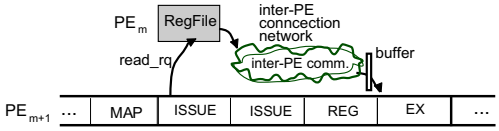


Figure 3. An inter-PE register read.

tion. We observe IPC and register access counts by cycle accurate execution driven simulations. We also estimate RegFile access energy and access time using CACTI model [17]. Finally, we evaluate the performance of each configuration measured by IPS (instructions per second) and the energy and power consumption of RegFiles using various benchmark programs.

The rest of this paper is organized as follows. In section 2, we show the overview of a clustered microarchitecture assumed in this paper. Section 3 describes an experimental framework, evaluation methodology and results. Section 4 describes future work including the application of highly clustered microarchitectures. Section 5 shows some related work. Section 6 concludes this paper.

2. Clustered microarchitectures

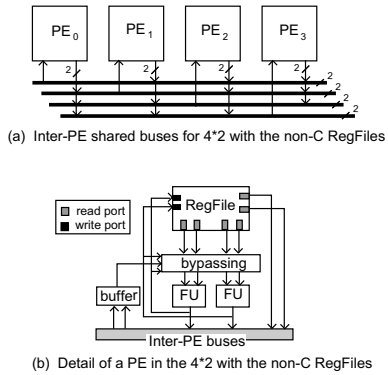
2.1. The structures of microarchitectures

The microarchitecture of a clustered microprocessor is based on that of the aggressive out-of-order issue superscalar processors. Fig. 1 shows the overview of the clustered microarchitecture assumed in this paper. We represent a configuration of the clustered microarchitecture as X*Y, where X denotes the number of PEs and Y denotes the issue width in a PE. The total number of issue width of the configuration is equal to the product of X and Y.

A PE is a group of partitioned datapath components such as a RegFile and a few functional units (FUs), and it is also called a cluster in some papers [1, 14]. We use the non-consistent (non-C) RegFile organization where a physical register in each RegFile has its own register instance and any register instances are not replicated among PEs. It is well known that the non-C RegFile organization can reduce the number of registers and ports for each RegFile more than the other RegFile organizations [12, 19].

Fig. 2 (a) shows the pipeline stages used in this paper. The pipeline organization is based on that of Alpha21264, which is composed of 7 stages (IF, ID, MAP, REG, ISSUE, EX, COMMIT). The processor front-end fetches multiple instructions at once in IF stage and decodes them ID stage. In MAP stage, instructions are dynamically steered to appropriate PEs and architectural registers are renamed to the physical registers. And then, the steered instruction is dispatched to the IQ (issue queue). In ISSUE stage, instructions in the IQ are checked whether their operands are ready and their corresponding FUs are available. If the operands and the FU of an instruction is available (i.e. the instruction is waked-up and selected), the operands are read in REG stage, and then the instruction is executed in its given latency in EX stage. The detailed timing of the pipeline is the same as that in [19].

Clustered microarchitectures induce inter-PE communication when an instruction uses some operand stored in the other PEs. We assume that it takes 2 extra cycles for this inter-PE communication. The timing of an inter-PE communication between dependent instructions is classified into following two cases based on the timing of a wakeup and select process. The first case is that the transferred operand is unready and will be produced by another PE. Fig. 2 (b) shows the pipeline timing of this case. After the transferred operand is produced, the waiting instruction is waked-up and selected. The second case is that the transfered operand has already been stored in a RegFile of another PE. In this case, the operand is read using an inter-PE register read process. Fig. 3 shows the pipeline timing of this inter-PE register read. When a PE is found to use a register instance stored in another PE in ISSUE stage, the register read is requested to the remote PE. After the data transfer starts, the instruction is waked-up and selected.



(a) Inter-PE shared buses for 4*2 with the non-C RegFiles

(b) Detail of a PE in the 4*2 with the non-C RegFiles

Figure 4. The clustered microarchitecture with the non-C RegFile organization.

In order to realize effective localized processing on clustered microarchitectures, a group of dependent instructions should be processed in a single PE. On the other hand, to realize parallel processing, groups of dependent instructions should be processed simultaneously in multiple PEs. If a program is partitioned into groups of dependent instructions ideally, there are no communication among PEs. However, the additional latency for the inter-PE communication and load imbalance among PEs will cause IPC degradation compared with the non-partitioned configurations. Therefore, we attempt to distribute instructions into the appropriate PEs.

An instruction steering scheme plays the role of dynamic program partitioning, that is to say, instruction distribution to appropriate PEs. In this paper, we use the !ready (not ready) instruction steering scheme because this can achieve higher IPC than the other basic schemes [18]. This scheme utilizes the status of operands of dependent instructions (i.e. source operands are ready or unready) in MAP stage. In order to reduce the IPC loss due to communication, this scheme steers instructions with at least on unready operand to the same PE as its dependent instruction. To improve the load balance among PEs, instructions without unready operands are steered to the minimum loaded PE. To determine the minimum loaded PE, we use the heuristics based on the number of waiting instructions in each PE.

2.2. Details of the inter-PE network and partitioned RegFiles

Fig. 4 shows the details of the clustered microarchitecture with the non-C RegFile organization assumed in this paper. A shared bus network is used as the inter-communication network fabric. Each shared bus corresponds to a write to one PE, and each PE is able to send data to any bus. Fig. 4 (b) shows the details of a PE in the 4*2 non-C RegFile configuration. We assume that the number of shared buses is the same as that of the total issue width which is equal to the product of the number of PEs and issue width in a PE. For example, in the 4*2 configuration, the total number of buses is eight and each PE has two buses for writes to the PE. In addition, we set the number of read ports for the inter-PE network in a RegFile to the same as the issue width in a PE. In the case of the 4*2 configuration, up to 2 registers are allowed to be read in a clock cycle from a single RegFile for the input of the inter-PE network. Our model of the inter-PE communication network on the non-C RegFiles is similar to the model in [22].

In this paper, we assume that an FU is composed of a set of one ALU and one multiplier, and the ALU and the multiplier share the ports of the RegFile. The number of ports required to feed data for one FU is two read ports and one write ports. In each PE, the number of FUs is set to the same as issue width in the PE. Hence, the total number of FUs in a clustered microarchitecture is equal to the total issue width.

Based on these detailed structures, we summarize the relationship between the degree of partitioning in terms of the number of registers and ports for an X*Y configuration (X: the number of PEs, Y: the issue width in a PE).

For the number of registers, there are two metrics in the clustered microarchitectures: the number of total registers N_{reg_total}, and the number of physical registers in a PE N_{reg_PE}. The number of total registers is:

$$N_{reg\_total} = X \cdot N_{reg\_PE} \tag{1}$$

In each RegFile, the total number of required ports is composed of that for functional units and for inter-PE communication networks. For functional units, 2Y read ports and Y write ports are required. The ports required for the inter-PE communication network of our non-C RegFile model is assumed to be Y read ports. Therefore, the total number of ports is:

$$ReadPorts = 3Y \tag{2}$$

$$WritePorts = Y \tag{3}$$

Table 1 shows the number of required ports for the 8-way and 4-way issue configurations. We note that non-clustered configurations (1*8, 1*4) do not require the ports for the inter-PE network, and they require only ports for functional units.

Since the energy dissipation and access time of a RegFile are mainly dependent on the number of ports and registers, the degree of partitioning is an essential parameter for the power and performance of the clustered microarchitecture. As a RegFile is highly partitioned and clustered, the RegFiles will be made their energy consumption lower and

Table 1. The number of ports of the RegFiles.

8-way config.	
8*1	3R + 1W
4*2	6R + 2W
2*4	12R + 4W
1*8	16R + 8W

4-way config.	
4*1	3R + 1W
2*2	6R + 2W
1*4	8R + 4W

Table 2. Main architectural parameters.

IQ, LQ, SQ size	64
ROB size	256
L1I cache	128KB, 2way
L1D cache	128KB, 2way
RegFile model	non-C
Inter-PE comm.	2 cycle latency

their access time faster. However, highly clustering causes extra communication among PEs. Therefore, we must clarify how much the partitioned RegFile achieves faster access and lower energy consumption, and how much the communication degrade its IPC.

3. Experiments

In this Section, we demonstrate power and performance advantages in the highly clustered microarchitecture.

3.1. Methodology

We modified sim-alpha [6] to model the clustered microarchitecture with all the architectural features discussed in Section 2. In this paper, we focus on the evaluation of integer datapath because the advantage in the integer datapath is easily applied to the floating point datapath. We simulate the 3-way (1*3 and 3*1), 4-way (1*4, 2*2 and 4*1), 6-way (1*6, 2*3, 3*2 and 6*1) and 8-way (1*8, 2*4, 4*2 and 8*1) issue processors. The number of ports for each configuration is set based on the equations in Section 2.2. The number of physical registers in a PE (N_{reg_PE}) is also varied. The number of instructions fetched and decoded in a single cycle is set to the same as the issue width. The other architectural parameters are shown in Table 2. The rest of parameters such as latency of the caches and that of functional units are following that of Alpha21264.

We modeled the access time and access energy of partitioned RegFiles using CACTI-2.0 tool set [17] at 0.07 μm technology. Basically the model is intended to evaluate cache system, so we discarded the tag path of the model and set the width of a register to be 64 bits as depicted in [20].

As a metric of performance, we utilize the IPS (instructions per second), which is the product of IPC and operating frequency. There are a number of critical paths that will set the operating frequency of processors, and the critical paths are ordinarily included in the issue queue, the register renaming unit and the RegFile. In reality, the delay of these structures is dependent on their implementation, but they will have structures similar to those found in the RegFile (such as numbers of ports or numbers of entries) and tend to scale together according to the issue width. Therefore, we assume that the operating frequency of processors is decided by the access time for RegFiles depicted in [7].

In order to estimate the total energy consumption, we count the total number of register accesses (N_{access}) in the cycle-accurate simulation. The number of register accesses includes all of the read and write accesses to the registers, and excludes the case of the operand fetch through the forwarding (bypassing) network in a PE similar to [15]. Using the register access energy (E_{access}) obtained by CACTI-2.0 model, we estimate total energy consumption (E_{total}) as follows:

$$E_{total} = E_{access} \cdot N_{access} \tag{4}$$

Furthermore, we estimate the power consumption of RegFiles using the operating frequency assumed. When the total number of executed instructions is N_{inst}, the power consumption of RegFiles is:

$$Power_{RegFiles} = (E_{total} \cdot IPS)/N_{inst} \tag{5}$$

We select a subset of 4 benchmarks (djpeg, cjpeg, rawdaudio, rawcaudio) from the MediaBench benchmark suite [11]. This benchmark suite captures the main features of commercial multimedia applications. Benchmarks which tend to achieve high instruction-level parallelism have been selected. We also select a subset of 7 benchmarks (gzip, vpr, gcc, mcf, perlbmk, bzip, twolf) from the SPEC2000CPU int benchmark suite [9]. The rest of SPEC2000 benchmarks could not be adapted to the simulation environment used. All the benchmarks were compiled for the Alpha binary using Compaq's C compiler v6.5 on Tru64 UNIX V5.1B with -O4 -fast -non_shared options. Each program of the MediaBench was executed until the completion and 100 million instructions of each program of the SPEC2000int were executed after forwarding 1 billion instructions.

3.2. Performance Evaluations

Fig. 5 shows IPC of the various 8-way and 4-way issue configurations for the MediaBench suite. The IPC is the average number of MediaBench benchmark programs. The x-axis is the number of physical registers per PE. The solid line represents 8-way configurations and the dotted line represents 4-way configurations.

The result shows that 1*8 and 1*4 configurations achieve the highest IPC in the same issue width configurations.

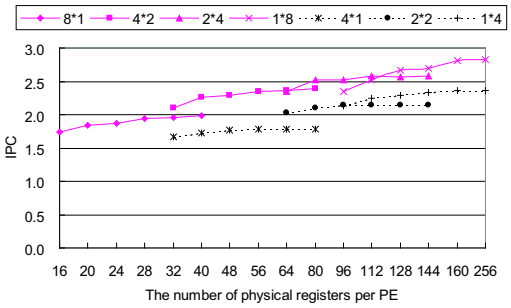


Figure 5. IPC for the MediaBench suite.

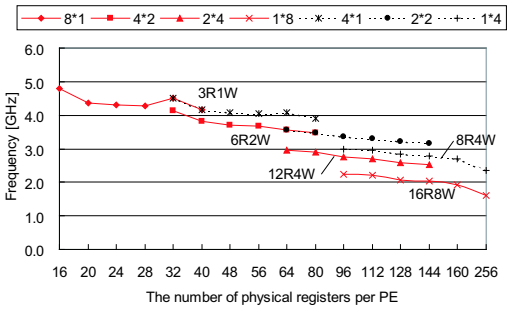


Figure 6. The operating frequency of the RegFiles.

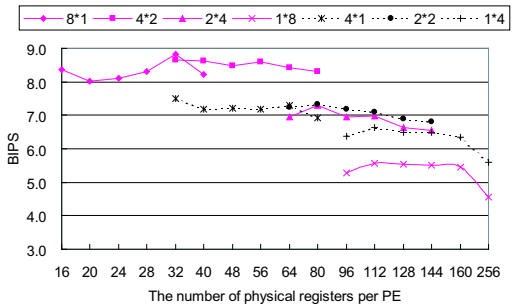


Figure 7. BIPS for the MediaBench suite.

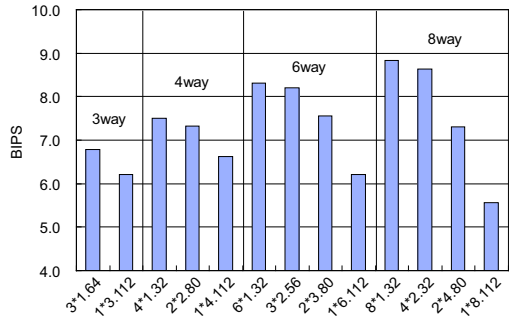


Figure 8. BIPS of 3-way to 8way issue configurations for the MediaBench suite.

Since the 1*8 and 1*4 configurations are not partitioned, they can accomplish better IPC without any inter-PE communication losses. We can observe that the more RegFiles are partitioned, the lower the IPC becomes. This degradation is caused by inter-PE communication. However, highly clustered configurations enable faster access and lower power RegFiles as discussed later.

It is also observed that IPC is increased when the number of registers per PE is increased in the same configuration. This is because the large number of registers can resolve more output dependencies and anti-dependencies using a register renaming technique. However, a large number of registers will enlarge access time and consume much energy as shown later.

In return for the IPC losses due to the highly clustering, the highly clustered configurations enable faster accesses to RegFiles, which allows higher operating frequency of the processors. Fig. 6 shows the operating frequency of the RegFiles in each configuration. The number of ports depicted in the figure is determined by the equations in Section 2.2.

It is observed that the more a RegFile is partitioned, the higher frequency it can achieve. Therefore, we can understand that the smaller number of ports is effective in realizing a higher operating frequency. The smaller number of registers also makes RegFile accesses faster. We note that the peaks in the 16 and 32 registers are caused by the fact that the CACTI model prefers multiples of 16 as the number of registers per PE. Also, the number of ports has a larger impact on the frequency. Since the number of ports is determined by the configuration of a PE, we have to consider configurations of the clustered microarchitecture carefully.

The large number of PEs due to the highly clustering requires more communications among PEs, resulting in IPC degradation. On the other hand, the highly clustering makes RegFile accesses faster and the faster RegFile allows higher operating frequency of the processor. To illustrate this trade-off more clearly, we use the IPS metric.

Fig. 7 shows BIPS (Billions of Instructions Per Second) for the MediaBench suite. Here, we assume that the access time for the RegFile is critical to the operating frequency of a processor and the operating frequency of a processor is equal to that of the RegFile.

The result shows that the 8*1 and 4*2 configurations achieve higher IPS in the 8-way configurations, and the 4*1 and 2*2 configurations achieve higher IPS in the 4-way configurations. We can understand that as the IPS tends to be

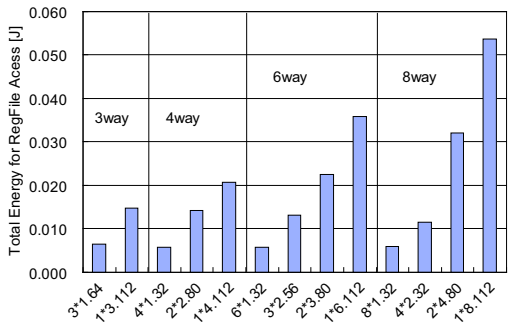


Figure 9. The total energy for register file accesses in the MediaBench suite.

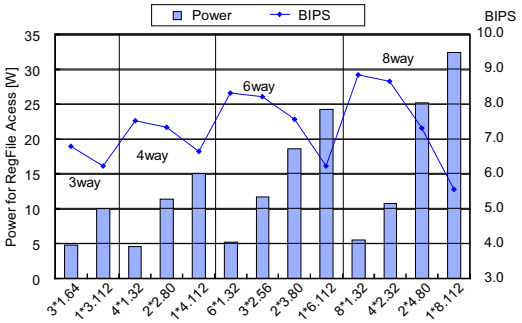


Figure 10. The power for register file accesses and BIPS for the MediaBench suite.

higher in the highly clustered configurations. The reason is that the highly clustered configurations can overcome the disadvantage of lower IPC using the benefit of its higher frequency. Therefore, We find that the RegFile of highly clustered microarchitectures can defeat the problem of design using the conventional multiported RegFile.

Fig. 8 shows the BIPS of 3-way, 4-way, 6-way and 8-way total issue width configurations. We select the number of registers that can achieve the maximum IPS for each configuration. The number following each configuration name in the figure represents the number of registers per PE. It is observed that if we increase the total issue width without partitioning and clustering, IPS is decreased. If we increase the total issue width with partitioning, the IPS is increased according to the degree of partitioning. IPS gain of the 8*1, 6*1, 4*1 and 3*1 configurations compared with the 1*3 configuration are by 42%, 34%, 21% and 9%, respectively.

3.3. Power estimation of partitioned RegFiles

Fig. 9 shows the total energy for RegFile accesses to complete a benchmark program. The total energy represented in the figure is the average number of MediaBench benchmark programs.

We observe that the highly clustered configurations reduce the total energy dissipation significantly. It is remarkable that the energy of the 8*1 configuration is reduced to 11% compared with the non-clustered 1*8 configuration. The number of total RegFile accesses is almost constant among all configurations. Therefore, energy dissipation for a RegFile access can contribute the efficient energy dissipation of RegFiles. The small number of ports and registers of highly clustered configurations enables such a low energy dissipation. Especially, the number of ports that is mainly decided by the issue width in a PE dominates the bulk of

Table 3. The BIPS, power and energy normalized by 1*8 configuration.

	MediaBench			SPEC2000CINT		
Config.	8*1	4*2	2*4	8*1	4*2	2*4
GIPS	1.59	1.56	1.32	1.67	1.69	1.25
Power	0.17	0.33	0.78	0.19	0.36	0.77
Energy	0.11	0.21	0.60	0.12	0.21	0.62

total energy dissipation. We note that the total energy dissipation of 8*1 configuration is slightly lower than that of the 3*1 configuration. This is because the number of physical registers per PE of 8*1 configuration is halved compared with that of the 3*1 configuration.

Fig. 10 shows the total power for RegFile accesses and BIPS for the MediaBench suite. Here, we assume that access time for the RegFile decides the operating frequency of the processor. The result shows that the highly clustered configurations have still advantage in terms of power consumption. The power consumption metric has a disadvantage for the configuration obtaining higher IPS because the time required to complete a program is shortened when the IPS is high. The shorter execution time increases the power if the total energy is constant. Nevertheless, the advantage of the lower RegFile energy consumption in highly clustered configurations compensate such a situation.

It is remarkable that the RegFile for the 8*1 configuration can be realized by a little increase of the RegFile power consumption compared with that of the 3*1 configuration, and about half of the non-partitioned 1*3 configuration while the 8*1 can achieve much more performance.

Table 3 shows IPS, energy and power of 8-way configuration for the SPEC2000int benchmark suite normalized by 1*8 configuration. Similar to the MediaBench suite, we can understand that the highly clustered configuration reduce energy and power consumption significantly together with high IPS. In the 8-way configurations, the highly clustered

configuration can achieve 1.59 times faster processing with 83% lower power consumption than the conventional non-partitioned configuration in MediaBench and 1.67 times faster processing with 88% lower power in SPEC2000int.

Finally, we can conclude that the RegFile access time and RegFile energy dissipation of highly clustered microarchitecture is enough low to defeat the RegFile issues of wider issue processors. Moreover, if we assume that the access time for the RegFile decides the operating frequency of the processor, we find that highly clustered microarchitectures can achieve higher performance and lower power.

Now it is time to move from the design using a centralized and monolithic RegFile to the design using localized and clustered RegFiles. In the past, we experienced the transition from CISC to RISC, that is to say, from the complex instruction set to simple instruction set. The potential of RISC is enhanced by the assist of effective hardware designs and compiler techniques. We believe that the highly clustered microarchitecture has a potential to be enhanced by the assist of effective hardware designs and compiler techniques, and will succeed like RISC processors.

4. Future Work

Estimating and understanding the total power consumption of clustered microarchitectures is quite important. Not only the RegFile, but also other global structures such as issue queue, inter-PE communication network, processor front-end and so on can be partitioned and clustered. Partitioning and clustering these structures also degrade the IPC. In addition, we should consider the power consumed by the instruction steering mechanism. As the degree of partitioning is increased, the complexity of an instruction steering mechanism might be increased.

To evaluate clustered microarchitectures more precisely, we should reconstruct the timing of pipeline stages. Partitioning and clustering global structures result in the other critical paths which are not considered in conventional superscalar pipeline models. Therefore, we plan to estimate more effects of degree of partitioning in terms of power and performance.

Current technology trends indicate that the fraction of the static power will exceed that of the total dynamic power as technology drops below the 65 nm feature size [3, 10]. The static power consumption caused by transistor leakage is increased linearly with the number of transistors whether the transistor is active or not. Therfore, it would be interesting to take into account area consideration in highly clustered microarchitectures. Furthermore, shutting off the inactive part of the system using power gating or multiple threshold voltage technique is one of the promising approaches to reduce the static power consumption [3, 10]. In the highly clustered microarchitecture, we can manage to turn unused PEs off by controlling the total issue width. In future, we plan to evaluate the various aspects of power consumption of the highly clustered microarchitecture.

Wider issue and highly clustered processors are motivated by the multithreading. While the 8-issue processor is not fully utilized in a single thread environment due to the lack of inherent ILP in most applications, it can easily take full advantage of wider issue capability in a multithread environment [4, 16].

5. Related Work

In order to obtain higher performance in dynamically-scheduled clustered microarchitectures, there are many proposals for instruction steering schemes and their comparisons in literature [1, 14, 18]. However, the degree of the partitioning is also one of the other important factors that determine performance of clustered processors. Previously, we have reported the evaluation of a clustered microarchitecture with various degrees of partitioning and organizations of RegFiles in terms of performance [19]. In this paper, we focus on power consumption as another important factor which is the most significant constraint to realize high performance processors. The results obtained in this paper illustrate the trade-off between power and performance more clearly.

Our model of the partitioned RegFiles is similar to the non-C (non-consistent) model in Zyuban and Kogge [22]. They evaluated the clustered microarchitecture with the non-C register files in terms of energy efficiency. However, they do not discuss the relationship between the IPC decrease and the degree of partitioning.

Aggarwal and Franklin [1] evaluated the IPC of a clustered microarchitecture varying the issue width in a PE. However, in their evaluation, the number of PEs is fixed when the issue width of a PE is varied, and they do not consider the power consumption and access time of register files.

6. Conclusions

In this paper, we have compared the power and performance of various configurations of the clustered microarchitecture, especially focusing on its register files, and further investigated reasonable degree of partitioning. The partitioning of a single monolithic register file into several smaller ones makes the register file access energy lower and access time shorter because the number of entries and ports can be reduced. On the other hand, the partitioning increases the amount of communication among PEs and this causes IPC degradation.

We have observed that IPC is decreased as a single monolithic register file is highly partitioned and clustered.

This is because the partitioning requires more communication among PEs. On the other hand, the highly partitioning makes register access time shorter and register access energy lower. In order to describe the performance trade off between the RegFile access time and IPC, we have utilized the IPS (instructions per second) metric assuming that the access time of the register file is critical path of processor operating frequency. The results have shown that the highly clustered configurations can achieve higher IPS than the moderate partitioning configurations.

We have estimated the total energy dissipation of the clustered register files in each configuration. From the results, we have found that the highly clustering makes total energy dissipation of the register files significant lower. Also, we have estimated the power consumption of register files. In the case that the access time of register file decides the operating frequency of the processor, the highly clustered configuration can achieve lower power consumption of register file than that of the alternatives.

These results have shown that the highly clustered microarchitecture has various advantages in terms of power consumption and performance. We believe that the highly clustered microarchitecture is an attractive design to overcome a list of difficulties to realize wider issue processors.

References

[1] A. Aggarwal and M. Franklin. Scalability of instruction distribution algorithms for on-chip clustering. *IEEE Transactions on Parallel and Distributed Systems*, 16(10):944–955, October 2005.

[2] R. Balasubramonian, S. Dwarkadas, and D. H. Albonesi. Reducing the complexity of the register file in dynamic superscalar processors. In *Proceedings of the 34th annual international symposium on Microarchitecture*, pages 237–248, 2001.

[3] J. A. Butts and G. S. Sohi. A static power model for architects. In *Proceedings of the 33rd annual ACM/IEEE international symposium on Microarchitecture*, pages 191–201, New York, NY, USA, 2000.

[4] J. D. Collins and D. M. Tullsen. Clustered multithreaded architectures - pursuing both ipc and cycle time. In *18th International Parallel and Distributed Processing Symposium*, page 76b, 2004.

[5] J.-L. Cruz, A. González, M. Valero, and N. P. Topham. Multiple-banked register file architectures. In *Proceedings of the 27th annual international symposium on Computer architecture*, pages 316–325. ACM Press, 2000.

[6] R. Desikan, D. Burger, and S. W. Keckler. Measuring experimental error in microprocessor simulation. In *Proceedings of the 28th annual international symposium on Computer architecture*, pages 266–277, 2001.

[7] K. Farkas, N. Jouppi, and P. Chow. Register file design considerations in dynamically scheduled processors. In *High-Performance Computer Architecture*, pages 40–51, 1995.

[8] J. L. Hennessy and D. A. Patterson. *Computer Architecture A Quantitative Approach , 3rd ed.* Morgan Kaufmann Publishers, 2002.

[9] J. L. Henning. SPEC CPU2000: Measuring CPU Performance in the new millennium. *IEEE Computer*, 33(7):28–35, 2000.

[10] N. S. Kim, T. Austin, D. Blaauw, T. Mudge, K. Flautner, J. S. Hu, M. J. Irwin, M. Kandemir, and N. Vijaykrishnan. Leakage current - moore's law meets static power. *IEEE Computer*, 36(12):68–75, 2003.

[11] C. Lee, M. Potkonjak, and W. H. Mangione-Smith. MediaBench: a tool for evaluating and synthesizing multimedia and communicatons systems. In *Proceedings of the 30th annual ACM/IEEE international symposium on Microarchitecture*, pages 330–335, 1997.

[12] J. Llosa, M. Valero, and E. Ayguade. Non-consistent dual register files to reduce register pressure. In *Proceedings of the 1st IEEE Symposium on High-Performance Computer Architecture*, pages 22–31, 1995.

[13] S. Palacharla, N. P. Jouppi, and J. E. Smith. Complexity-effective superscalar processors. In *Proceedings of the 24th annual international symposium on Computer architecture*, pages 206–218, 1997.

[14] J.-M. Parcerisa and A. González. Reducing wire delay penalty through value prediction. In *Proceedings of the 33rd annual international symposium on Microarchitecture*, pages 317–326, 2000.

[15] I. Park, M. D. Powell, and T. N. Vijaykumar. Reducing register ports for higher speed and lower energy. In *Proceedings of the 35th annual ACM/IEEE international symposium on Microarchitecture*, pages 171–182, 2002.

[16] S. E. Raasch and S. K. Reinhardt. The impact of resource partitioning on SMT processors. In *Proceedings of the 12th International Conference on Parallel Architectures and Compilation Techniques*, page 15, Washington, DC, USA, 2003.

[17] G. Reinman and N. P. Jouppi. CACTI 2.0: An integrated cache timing and power model. Technical report, WRL Research Report 2000/7, 2000.

[18] Y. Sato, K. Suzuki, and T. Nakamura. An operand status based instruction steering scheme for clustered architectures. In *Proceedings of the 2005 International Conference on Computer Design (CDES'05)*, pages 168–174, 2005.

[19] Y. Sato, K. Suzuki, and T. Nakamura. Partitioned register file designs for clustered architectures. *Journal of Information*, pages 119–134, January 2006.

[20] A. Seznec, E. Toullec, and O. Rochecouste. Register write specialization register read specialization: a path to complexity-effective wide-issue superscalar processors. In *Proceedings of the 35th annual ACM/IEEE international symposium on Microarchitecture*, pages 383–394, 2002.

[21] V. Zyuban and P. Kogge. The energy complexity of register files. In *Proceedings of the 1998 international symposium on Low power electronics and design*, pages 305–310, 1998.

[22] V. V. Zyuban and P. M. Kogge. Inherently lower-power high-performance superscalar architectures. *IEEE Transactions on Computers*, 50(3):268–285, 2001.