PAPER

Power Estimation of Partitioned Register Files in a Clustered Architecture with Performance Evaluation

Yukinori SATO[†,††a)], ***Nonmember***, **Ken-ichi SUZUKI**[†], ***and*** **Tadao NAKAMURA**[†], ***Members***

SUMMARY High power consumption and slow access of enlarged and multiported register files make it difficult to design high performance superscalar processors. The clustered architecture, where the conventional monolithic register file is partitioned into several smaller register files, is expect to overcome the register file issues. In the clustered architecture, the more a monolithic register file is partitioned, 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 register files. Therefore, degree of partitioning has a strong impact on the trade-off between power consumption and performance. In addition, the organization of partitioned register files also affects the trade-off. In this paper, we attempt to investigate appropriate degrees of partitioning and organizations of partitioned register files in a clustered architecture to assess the trade-off. From the results of execute-driven simulation, we find that the organization of register files and the degree of partitioning have a strong impact on the IPC, and the configuration with non-consistent register files can make use of the partitioned resources more effectively. From the results of register file access time and energy modeling, we find that the configurations with the highly partitioned non-consistent register file organization can receive benefit of the partitioning in terms of operating frequency and access energy of register files. Further, we examine relationship between IPS (instructions per second) and the product of IPC and operating frequency of register files. The results suggest that highly partitioned non-consistent configurations tends to gain more advantage in performance and power.

key words: *clustered architecture, partitioned register files, non-consistent register files, instruction level parallelism*

1. Introduction

As register files (RegFiles) in modern high-performance processors are enlarged and multiported to support wider issue out-of-order execution, power consumption of the RegFiles will become a serious restriction of realizing future high-performance microprocessors [1]. The enlarged and multiported RegFile also makes its access time slower. The access time of the RegFile is critical since it can impact the cycle time of the processor.

The power consumption and access time of a RegFile are mainly dependent on the number of registers and the number of ports [2], [3]. Reducing the number of registers and ports by means of partitioning a RegFile is one of the effective approaches to realize a reasonable RegFile.

Dynamically-scheduled clustered architectures are expected to overcome the register file issues [4]–[6]. In clustered architectures, 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 [5]. One of the global structures partitioned is the conventional single monolithic RegFile, and each of the partitioned RegFiles is provided in a single PE. The partitioning makes the RegFile power consumption lower and RegFile access time faster, because the number of entries and ports of the partitioned structures can be reduced. The much more the number of entries and ports is reduced by means of increasing the degree of partitioning, the less power consumption RegFile we will be able to realize, which contributes to solving the restriction caused by the enlarged and multiported RegFiles.

IPC (instructions per clock cycle) of clustered architectures is lower than that of non-partitioned configurations corresponding to conventional superscalar processors because the partitioning causes extra communication delay among PEs. The amount of communication among PEs is dependent on degree of partitioning. As the degree of partitioning is increased, the IPC is decreased due to the communication, while providing the lower power and faster access RegFile. Therefore, degree of partitioning has a strong impact on the trade-off between power consumption and performance.

The other factor that determines the trade-off between power consumption and performance in the partitioned configurations is an organization of partitioned RegFiles. There are two organizations of partitioned RegFiles: multiple coherent (MC) RegFiles [7] and non-consistent (non-C) RegFiles [8]. In the MC RegFiles, any register instances are replicated in each PE, so, every PE can utilize any register instances locally without the inter-PE register read. On the other hand, any register instances in the non-C RegFiles are not replicated. Therefore, the non-C RegFile organization is realized by the smaller number of registers and ports. However, when a PE uses a register instance stored in another PE, additional inter-PE communication is required, which causes performance degradation.

In this paper, we attempt to show a relationship between degrees of partitioning, performance and power consumption of RegFiles. Then, we perform cycle accurate simulation on various 8-way issue configurations varying the degree of partitioning and the organization of partitioned RegFiles. We also perform estimation of access time and access energy for RegFiles in each configuration using CACTI model [9]. Using these results, we further discuss perfor-

Manuscript received December 5, 2005.
Manuscript revised July 22, 2006.

[†]The authors are with Graduate School of Information Sciences, Tohoku University, Sendai-shi, 980–8579 Japan.
[††]The author is with Sendai Software Development Center, FineArch Inc., Sendai-shi, 980–6108 Japan.
a) E-mail: yukinori@archi.is.tohoku.ac.jp

DOI: 10.1093/ietisy/e90–d.3.627

mance and power consumption of partitioned RegFiles in a clustered architecture.

The rest of this paper is organized as follows: Section 2 shows the base microarchitecture used in this paper and discusses the effect of organization of partitioned RegFiles. Section 3 describes the experimental framework, the evaluation methodology and the results. Section 4 shows some related work. Section 5 concludes this paper.

2. Clustered Architecture

2.1 Microarchitecture Configuration

The microarchitecture of a clustered architecture is based on that of the aggressive out-of-order issue superscalar processors. Figure 1 shows an overview of the microarchitecture of the clustered architecture. We represent a configuration of the clustered architecture as X*Y, where X denotes the number of PEs and Y denotes the issue width in a PE. In this paper, we simulate the following four configurations as 8-way issue processors: 1*8, 2*4, 4*2 and 8*1. In the 1*8 configuration, any structures are not partitioned, so it corresponds to a conventional superscalar design. In partitioned configurations, a degree of partitioning corresponds to the number of PEs. If the number of PEs is increased, less ported RegFiles are realized, which enable faster access and lower power. However, data processing with the large number of PEs induces much more communication among PEs.

Figure 2 (a) shows the pipeline structure used in this paper, which are based on those of Alpha21264 [7]. The processor front-end fetches multiple instructions at once in IF stage and decodes them in ID stage. In MAP stage, architectural registers of the decoded instructions are renamed to physical registers. The register renaming mechanism we adopted is also based on that of Alpha21264. At the same time, steering logic dynamically chooses a PE for the execution of each instruction based on an instruction steering scheme. In ISSUE stage, the steered instruction is dispatched to IQ (issue queue), and observed whether the operands of each instruction are ready or not. When all of the required operands are ready, the instruction is waked up and resources of the steered PE are checked. If the resources are available, the instruction is selected and the unit is reserved for execution. After the operands are read in REG stage, the instruction is executed in its given latency in EX stage.

In the case where the instruction uses at least one unready operand, the instruction must wait until the results of the preceding instructions are provided. When the preceding dependent instruction is executed in the same PE as the waiting instruction, the waiting instruction is executed at the next cycle of the execution of the preceding instruction using a forwarding network. On the other hand, when the waiting instruction is allocated in a different PE from the preceding dependent instruction, the result from different PE must be transfered to the PE where the waiting instruction

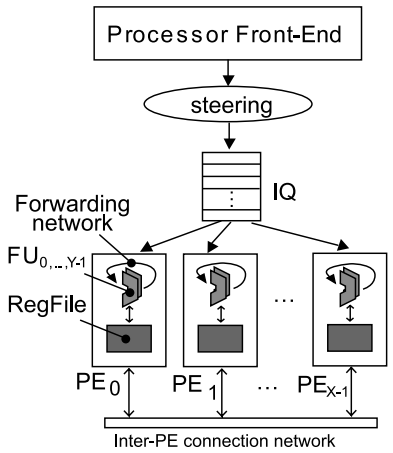
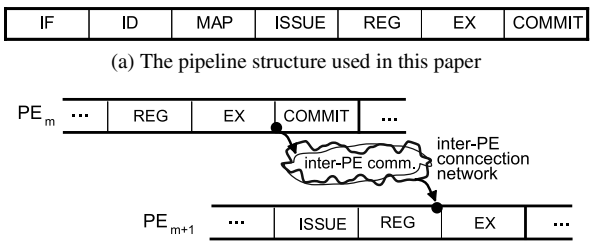


Fig. 1 The base clustered architecture (X*Y configuration).



(a) The pipeline structure used in this paper

(b) Pipeline timing of an inter-PE result communication

Fig. 2 The timing of the pipeline.

is allocated. We assume that it takes 2 extra cycles for this inter-PE communication as shown in Fig. 2 (b).

In order to avoid performance loss due to inter-PE communication, an instruction steering scheme attempts to distribute instructions to appropriate PEs. In addition to inter-PE communication, load imbalance among PEs also degrades IPC of the clustered microarchitectures [5], [6]. In this paper, we use the !ready (not ready) instruction steering scheme because this scheme utilizes the status of operands to reduce the IPC loss due to communication and load balancing, and this can achieve higher IPC than the other schemes [10].

In order to prevent undesirable inter-PE communications, the !ready scheme steers instructions with at least one unready operand to the same PE as its dependent instruction. To improve the load balance among PEs, instructions with no unready operand are steered to the minimum loaded PE. The status of operands is always monitored to realize the out-of-order execution in conventional processors. To monitor the minimum loaded PE, we use the heuristics based on the number of waiting instructions in each PE because this heuristics can obtain better performance than the DCOUNT heuristics [5].

In this paper, we assume all the PEs share a single large IQ to isolate a problem of the lack of available IQ entries from our evaluation as shown in Fig. 1. If we partition the IQ across the PEs, we have to take into account the utilization of each IQ. The instruction steering scheme used in this

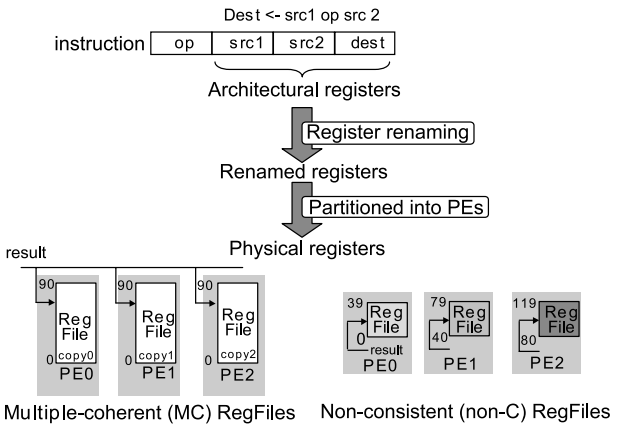


Fig. 3 Register mapping process in partitioned RegFiles.

work tries to even up the utilization of each IQ. As a result, the IPC degradation due to the partitioned IQ will be negligible. The effect of the partitioned IQ will be evaluated in our future work including considerations for the effect of wakeup delay on circuit level [4].

2.2 Organizations of Partitioned RegFiles

Most of current instruction set architectures are based on a single set of registers. However, a clustered architecture provides physically partitioned sets of registers, and each PE provides a set of registers. Therefore, a register mapping mechanism needs to map architectural registers into partitioned physical registers in an effective manner. Figure 3 shows register mapping process in partitioned RegFiles. Originally, operands of an instruction are specified by architectural registers. To resolve WAR and WAW dependencies, the architectural registers are renamed to renamed registers. In clustered architectures, physical registers are partitioned into PEs, so, renamed registers must be mapped to appropriate physical registers. There are two partitioned RegFile organizations to decide the mapping of renamed registers: multiple coherent (MC) RegFiles [7] and non-consistent (non-C) RegFiles [8]. The main difference between them is whether a renamed register is mapped to multiple physical registers or one physical register.

In the MC RegFiles, a renamed register is mapped to all of the partitioned RegFiles. The numbers on the left of the RegFiles in Fig. 3 indicate the identifiable numbers for renamed registers. In the MC RegFiles, physical registers placed at the same position of each RegFile are always replicated. This replication requires a large number of registers for containing the same register instance in each partitioned RegFile, and additional dedicated write ports for writing all the produced results to all the RegFiles.

In the non-C RegFiles, a physical register in each RegFile has its own register instance since a result is written into only one physical register [8]. This organization can reduce the number of registers for each RegFile because a register instance is not replicated, and the number of ports because

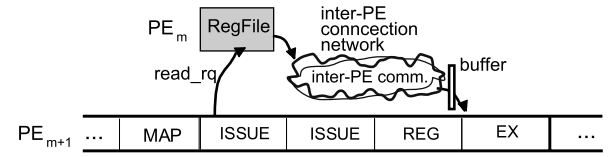


Fig. 4 The pipeline timing of inter-PE register read in the non-C RegFile organizations.

results are not delivered to the other RegFiles. Therefore, as illustrated in Fig. 3, the identifiable numbers of renamed registers are not replicated.

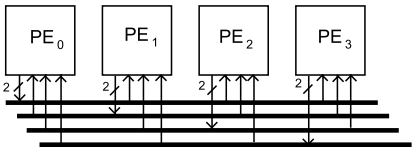
The non-C RegFile organization requires the additional inter-PE communication when a PE is found to use a register instance stored in another PE. We assume that a remote register read requires two cycles for the communication. Figure 4 shows the pipeline timing of an inter-PE register read. Since additional inter-PE communication will cause performance degradation, we must prevent inter-PE communication by distributing register instances into proper PEs.

The other difference between the MC and non-C RegFiles is a management scheme of free registers and committed registers. A free register is a register that is not used in any instructions at the time. In the non-C organization, the numbers of available free registers are different among PEs. To accommodate this difference among PEs, we prepare each PE its own free register list and introduce an instruction reallocation mechanism. The reallocation mechanism works as follows: When a PE selected by an instruction steering scheme lacks available free registers, the instruction is reallocated to one of the other PEs with available registers not to stall the instruction steering. This reallocation causes extra inter-PE communication delay, so the lack of available free registers in particular PEs affects performance in the non-C RegFile organizations.

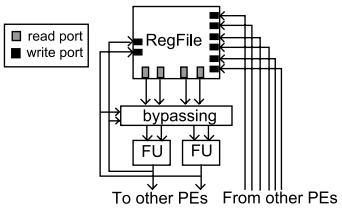
A committed register is a register that stores a value committed in COMMIT stage, and always placed in a dedicated register. The number of committed registers is the same as the number of architectural registers defined in the instruction set, i.e. 32. In the MC RegFile organization, the committed registers are also replicated across the partitioned RegFiles. On the other hand, in the non-C RegFile organization, the committed registers are not always distributed into all the partitioned RegFiles. If the committed registers are placed on a particular PE, the register pressure is increased due to the lack of available free registers on the PE. To avoid the converging, we assume the configuration that committed registers are partitioned into PEs and each PE has the same number of committed registers. For example, in an 8 PE configuration, 32 committed registers ($0-$31) are partitioned as follows: PE_0 has $0-$3, PE_1 has $4-$7, ..., PE_7 has $28-$31.

2.3 Inter-PE Networks of the Partitioned RegFiles

The number of registers and ports are important parameters that decide access time and access energy of RegFiles.

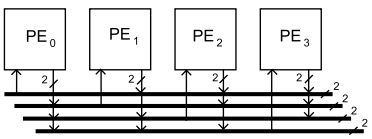


(a) Inter-PE dedicated lines for 4*2 with the MC RegFiles

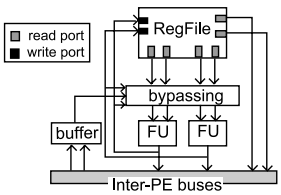


(b) Detail of a PE in the 4*2 with the MC RegFiles

Fig. 5 The MC RegFile organization.



(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

Fig. 6 The non-C RegFile organization.

These parameters are dependent on not only the degree of partitioning but also the structure of the inter-PE communication network, which is dependent on the organization of the partitioned RegFiles.

The MC RegFile organization requires a fully-connected network to deliver all the produced results to all of the RegFiles. To realize the network, dedicated signal lines and write ports for each result are required. Figure 5 (a) illustrates an overview of the inter-PE dedicated lines of the 4*2 with MC RegFile configuration. In this case, every PE can produce up to 2 results per cycle, so the required number of write ports in a single PE from the other PEs is 6. Figure 5 (b) illustrates the details of a PE in this configuration. In this case, total requirement for the ports in a PE is 4 reads and 8 writes.

The non-C RegFile organization is configured using a shared bus network as illustrated in Fig. 6 (a). Each shared bus corresponds to a write to one PE, and each PE is able to send data to any bus. Unlike the dedicated lines for the MC RegFiles, there can be various network configurations for non-C RegFiles. We assume the inter-PE communication network that is made up of as many buses as the total issue width, because we want to make a fair comparison by equalizing the maximum number of transfered data per clock cycle to that of the MC RegFile organizations. 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.

Figure 6 (b) shows details of a PE in the 4*2 non-C RegFile configuration. We set the number of read ports for buses 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 read in a clock cycle from a single RegFile for the input of the buses. Our model of the inter-PE communication on the non-C RegFiles is similar to the model in [11]: when an operand from a remote PE is ready, the operand is send to the consumer PE as soon as possible. To support this communication, we use a small buffer to save delivered operands from remote PEs. The size of this kind of buffer has been reported to be small enough not to affect the operating frequency and the total hardware cost [11].

We note that this shared bus network causes resource conflicts if there are no available shared buses or read ports for the execution. The resource conflicts in the network incur the delay of communications and the waiting instruction is stalled until the data arrives. In Sect. 3.3, we evaluate effects of communication bandwidth by varying the number of the shared buses and read ports for inter-PE network.

In order to understand differences among various configurations, we summarize the relationship between the degree of partitioning and the organization of RegFiles in terms of the number of registers and ports. Here, we assume a clustered architecture of X*Y configuration, where X represents the number of PEs, and Y the issue width in a PE.

In a clustered architecture, there are two metrics for the number of registers: the total number of physical registers which includes replicated registers, and the total number of registers for register renaming which excludes all the replicated registers. The total number of physical registers (N_{reg_phy}) is represented using the number of physical registers in a PE (N_{reg_PE}) as follows:

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

On the other hand, the total number of registers for register renaming (N_{reg_rnm}) is different between organizations of RegFiles.

$$N_{reg\_rnm}(MC) = N_{reg\_PE} \tag{2}$$

$$N_{reg\_rnm}(nonC) = X \cdot N_{reg\_PE} \tag{3}$$

From these equations, we find that the MC RegFile organization requires the X times larger number of physical registers over the non-C RegFiles organization comparing the two RegFile organizations with the same number of total registers for register renaming.

The number of ports is also different among the degree of partitioning and the organization of RegFiles. Table 1 shows the number of required ports for each organization. In each RegFile, the required ports consist of the ports for the local functional units and the inter-PE communication network. For the functional units, 2Y read ports and

Table 1 The number of ports in each configuration.

Config.	MC RegFiles	Non-C RFs
1*8	16 R + 8 W (Not partitioned)	
2*4	8 R + 8 W	12 R + 4 W
4*2	4 R + 8 W	6 R + 2 W
8*1	2 R + 8 W	3 R + 1 W
X*Y (X≥2)	2Y R + XY W	3Y R + Y W

Table 2 Main architectural parameters.

Fetch and decode	8 instructions
Branch predictor	Tournament branch predictor
IQ, FQ, LQ, SQ size	64
ROB size	256
Issue width	8 issue in total
L1Icache	128KB, 2way
L1Dcache	128KB, 2way

Y write ports are provided, where Y is the issue width in a PE. The number of ports required for the inter-PE communication network differs between the organizations of partitioned RegFiles. In the MC RegFiles, (X-1)Y write ports are required for the inter-PE network, while the non-C RegFiles require only Y read ports. The total number of ports is 2Y read ports and XY write ports for the MC RegFile configuration, and 3Y read ports and Y write ports for the non-C RegFile configuration.

3. Experiments

3.1 Methodology

We developed a cycle-accurate execution-driven simulator to evaluate the various configurations of clustered architecture. Baseline simulator is sim-alpha [12], which is one of the extention versions of SimpleScalar tool set [13]. The sim-alpha models the detailed microarchitecture of Alpha21264, which is one of the clustered architectures composed of dual integer PEs (clusters) with the MC RegFiles.

We modified sim-alpha to model an 8-way clustered architecture with all of the architectural features discussed in the previous section including the degree of partitioning, the organizations of partitioned RegFiles and the number of registers per PE. 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 [9] at 0.07 μm technology. Basically the CACTI model is intended to evaluate cache system, so we discarded the tag path and set the width of a register to be 64 bits as depicted in [14].

In order to estimate the total energy consumption of partitioned RegFiles, we count the total number of register accesses (N_{access}) in cycle-accurate simulation. The number of register accesses includes all of the read and write accesses to registers and excludes the case of operand fetch through a 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: $E_{total_{RF}} = E_{access} \cdot N_{access}$.

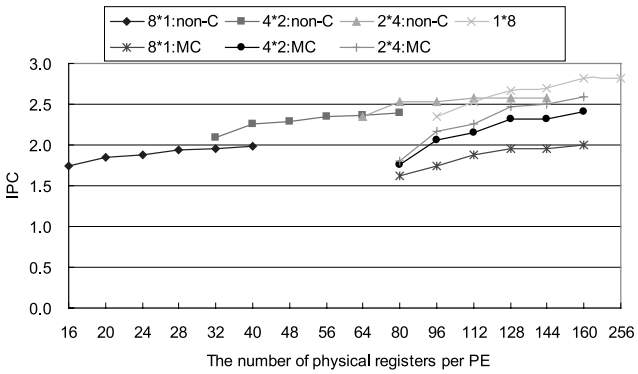


Fig. 7 IPC for the MediaBench suite.

We select a subset of 4 benchmarks (djpeg, cjpeg, rawdaudio, rawcaudio) from the MediaBench benchmark suite [16]. 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 [17]. 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 IPC

Figure 7 shows IPC averaged in the MediaBench suite. The results obtained in the SPEC2000CINT suite also show the similar characteristics. The specifier following the colon in the legend of figure indicates the organization of the partitioned RegFiles. The x-axis is the number of physical registers per PE.

The result shows that IPC of 1*8 configuration is the highest of all. The 1*8 configuration does not partition any structures, so the 1*8 configuration can perform ideal 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. 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 reduce IPC losses due to the lack of available free registers.

When the total number of registers for register renaming is equal, IPC of MC organizations is higher than that of non-C organizations. For example, in the case that the total number of registers for register renaming is 128, IPC of the 8*1:MC with 128 registers per PE configuration (the total

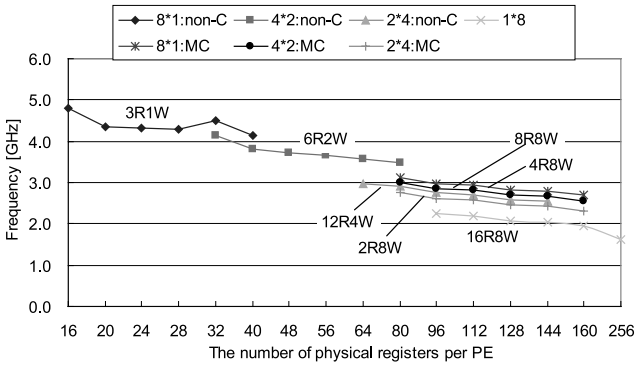


Fig. 8 Operating frequency of RegFiles.

number of physical registers is 1024) is higher than that of the 8*1:non-C 16 registers per PE configuration (the total number of physical registers is 128). The IPC losses of a non-C organization is caused by extra communication delay due to the inter-PE register read and the lack of available free registers in particular PEs.

However, it is hard for MC organizations to increase the number of registers for register renaming due to their replicated structures. On the other hand, a non-C organization can increase the number of registers for register renaming at less cost. As the number of register per PE is increased in non-C organizations, we can find out that they can achieve higher IPC with the smaller number of total physical registers than MC organizations. Therefore, we can understand that non-C RegFile organizations exploit the partitioned resources more effectively because it does not duplicate any register instances.

Furthermore, we find that the range of IPC is mainly decided by the number of ports. Even if we increase the number of registers, we observe that IPC gain is saturated. The number of ports is determined by the organization of RegFiles and the degree of partitioning. Therefore, we find that these two are more important parameters.

In return for the IPC losses, the partitioning makes RegFile accesses faster due to the smaller number of physical registers and ports. Figure 8 shows operating frequency of each RegFiles. The number of ports depicted in the figure is determined by degree of partitioning and organization of partitioned RegFiles as shown in Sect. 2.3. It is observed that the more a RegFile is partitioned, the higher frequency the RegFiles can achieve. The smaller number of registers also makes RegFile accesses faster. However, as shown in Fig. 8, the number of ports has a larger impact on the frequency than the number of registers.

In terms of organization of RegFiles, operating frequency of non-C RegFiles is higher than that of MC RegFiles even in the same degree of partitioning. This is because the non-C RegFiles organization can reduce more ports. It is also observed that if we do not partition any structure at all, the operating frequency becomes a half of the highly partitioning configuration with non-C RegFiles. The higher operating frequency of RegFiles is useful for avoiding the performance bottleneck caused by slow access time of RegFiles. Therefore, the highly partitioned non-C RegFile configuration will be one of the solutions for future high performance microprocessors.

Table 3 Various bus configurations for non-C RegFile organization.

(a) 8*1:non-C (the # of registers in a PE = 32)

Bus config.	b2r2	b2r1	b1r1(base)
nIPC	0.94	0.92	0.91

(b) 4*2:non-C (the # of registers in a PE = 32)

Bus config.	b2r2(base)	b2r1	b1r1
nIPC	0.99	0.95	0.89

(c) 2*4:non-C (the # of registers in a PE = 80)

Bus config.	b4r4(base)	b4r2	b4r1	b2r2	b2r1	b1r1
nIPC	1.00	1.00	0.97	0.99	0.95	0.92

3.3 Effect of Communication Bandwidth

As discussed in Sect. 2.3, the inter-PE communication network in non-C RegFiles can have various configurations. Then, we analyze impacts of inter-PE communication bandwidth in non-C RegFile organization. Table 3 shows normalized IPC for various bus configurations in non-C RegFile organization. For each number of PEs (8, 4 and 2), we chose the number of physical registers per PE to achieve the highest product of IPC and operating frequency of RegFiles, assuming unlimited number of buses. In the notation of 'Bus config.,' the numbers following 'b' and 'r' are the number of shared buses destined for a PE and the number of read ports per PE dedicated for inter-PE communication, respectively. 'nIPC' is the relative IPC normalized by that of the ideal unlimited bus configuration. For example, a 4*2b2r1 configuration represents two buses for a PE and one read port for inter-PE communication.

It is observed that there is IPC degradation when the bandwidth is limited. In Sect. 3.2, we simulated baseline configurations indicated '(base)' in Table 3. In each baseline configuration, the total number of shared buses is equal to the issue width in a PE, which is eight in this 8-way clustered architecture. In 8*1:non-C configurations, IPC of the baseline bandwidth configuration is decreased by 9% compared with the ideal bandwidth configuration. IPC of the baseline bandwidth for 4*2:non-C and 2*4:non-C configurations is the almost same as the unlimited bus configurations. In 4*2 configurations, IPC is decreased when the bandwidth is reduced from the baseline. However, in 2*4 configurations, IPC is not decreased when the bandwidth is reduced to some degree.

We can understand that the bandwidth of the shared bus network surely affects IPC of clustered architecture. However, if the bandwidth is decreased, the number of ports required for the buses can be reduced. The less number of ports will make RegFile access time faster. Moreover, we can see that decreasing the number of register read ports does not always decrease its IPC. Therefore, we should reduce the number of register read ports that do not affect

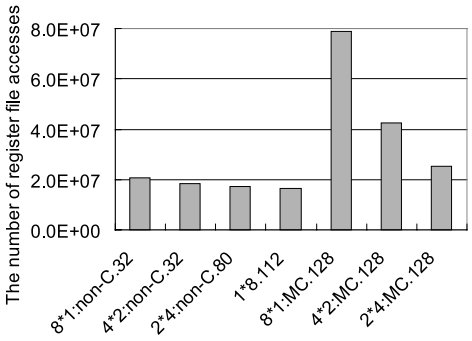


Fig. 9 The number of register file accesses.

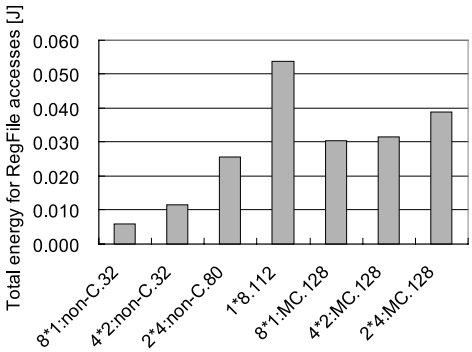


Fig. 10 The total energy for register file accesses.

the IPC. In the following section, we evaluate 8*1b1r1, 4*2b2r2, and 2*4b4r2 configurations as base non-C RegFile configurations.

3.4 Energy Dissipation for Partitioned RegFiles

Figure 9 shows the number of RegFile accesses for each configuration. In each configuration, we choose the number of registers per PE to make the product of its IPC and RegFile operating frequency the highest, and the number of RegFile accesses is averaged among MediaBench suite. In MC organizations, as degree of partitioning is increased, there is a significant increase of the number of register accesses compared with that of non-C organizations. The reason of this increase is that all the results must be written to all the RegFiles in order to keep coherence of RegFiles. This large number of accesses will increase the total energy dissipation and power consumption for RegFiles.

The number of register accesses in non-C organizations is decreased dramatically compared with that of MC organizations. Within non-C organizations, there are little increases of the numbers of accesses as the number of PEs is increased. This is because more dependent instructions are executed using forwarding logic in less partitioned configurations.

Figure 10 shows the total energy for RegFile accesses. The total energy represented in the figure is the average number of MediaBench suite. We observe that non-C organizations reduce the total energy dissipation significantly. The energy is reduced to 11% in the 8*1 configuration, and to 21% in the 4*2 configuration compared with a non-partitioned 1*8 configuration. The small number of registers and ports in a non-C organization makes it possible to realize such a low energy dissipation. It is also remarkable that the energy for the RegFile accesses can be reduced linearly with the degree of partitioning. For example, the energy becomes about one half when the number of PEs is increased from 4 to 8. Therefore the highly partitioning is effective in reducing the energy dissipation of RegFiles.

The total energy of MC organizations is decreased compared with the non-partitioned 1*8 configuration. However, the amount of the reduction is smaller than that of non-C organizations. This is because the number of ports, physical registers and access counts for the RegFiles are not so reduced compared with those of the non-C configuration.

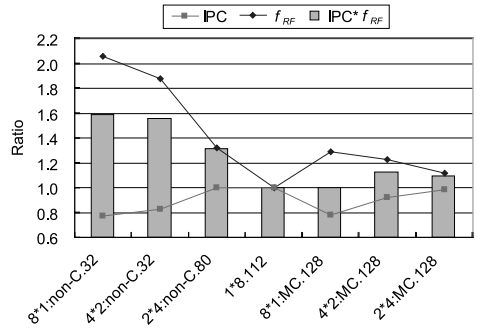
3.5 Power and Performance Consideration

IPC is an useful metic to compare performance of different architectures. However, IPC does not include a factor of operating frequency of a processor. There are a number of factors that will affect the operating frequency of a processor, but the critical path is generally considered to appear in issue queue, register renaming unit, forwarding logic, or RegFile.

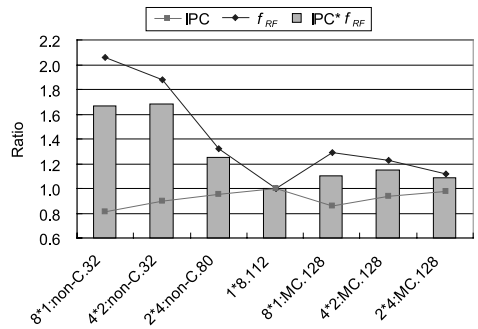
Improving operating frequency of RegFiles is useful for avoiding the possible performance bottleneck. Some current microprocessors use a pipelined RegFile to achieve a high operating frequency. However, the pipelined RegFile in turn degrades the IPC because its multi-cycle access time increases branch and load speculation miss penalties [18]. Moreover, a pipelined RegFile requires an extra level of forwarding logic, which increases the complexity of forwarding logic [2]. Since forwarding logic tends to affect the operating frequency of a processor [4], we should avoid increasing its complexity. Therefore, a non-pipelined RegFile is a preferable solution for a high performance microprocessor.

In order to give further consideration of performance including the access frequency of a RegFile, we multiply IPC and operating frequency of a RegFile. Figure 11 shows the products of IPC and operating frequency of partitioned RegFiles ($IPC * f_{RF}$) normalized by that of 1*8 configuration. The results represented in this figure are the peak of each configuration. The results show that 8*1:non-C and 4*2:non-C configurations achieve higher $IPC * f_{RF}$ than any other configurations. The MC RegFile organizations cannot achieve high $IPC * f_{RF}$ due to the low operating frequency compared with the non-C organizations. Furthermore, if the degree of partitioning is increased in non-C organizations, higher $IPC * f_{RF}$ can be obtained.

These results mean that non-C RegFiles are enough fast not to become a critical path of the processor. Consequently, these suggest that non-C organizations might overcome the disadvantage of lower IPC using the benefit of its higher frequency of RegFiles. Further, increasing degree of partitioning contributes to the reduction of the complexity of



(a) MediaBench average



(b) SPEC2000CINT average

Fig. 11 Normalized $IPC * f_{RF}$.

forwarding logic. These also imply that the highly partitioned non-C configuration tends to gain more advantage in performance.

The latency of the structures to become a critical path is dependent on their implementation, but it tends to scale together according to their issue width because the numbers of ports or entries of the structures are increased as their issue width becomes wider. This scaling comes from the fact that the most of these structures are composed of multiported register cells. Therefore, their access time tends to scale together with that of RegFiles. Farkas and Chow [19] assumed operating frequency of a RegFile as that of a microprocessor, and evaluate IPS (instructions per second), which is the product of IPC and operating frequency. Based on the assumption, $IPC * f_{RF}$ will be an useful metric for evaluating the performance of microprocessors.

Since power consumption of RegFile is predicted to be a major restriction of realizing high-performance processors, we evaluate their power consumption based on the above consideration of performance. Power consumption is defined as energy dissipation per second. Based on energy dissipation of RegFiles, the power consumption of RegFiles ($Power_{RF}$) is estimated using the total number of executed instructions (N_{inst}), total energy consumption for RegFile accesses ($E_{total_{RF}}$) and IPS as follows:

$$Power_{RF} = E_{total_{RF}} / ExecutionTime \quad (4)$$

$$= E_{total_{RF}} \cdot (IPS / N_{inst}) \quad (5)$$

In the previous section, we found that energy dissipation for RegFile accesses in the non-C configuration is reduced linearly as the degree of partitioning is increased. Based on the fact that N_{inst} is almost constant, the other factor that decides the power consumption of RegFiles is the IPS. In the above discussion, we suggested that the highly partitioning have potential to make its IPS higher. The equation of $Power_{RF}$ indicates that $Power_{RF}$ is increased as the IPS becomes higher.

To evaluate how the higher IPS affects $Power_{RF}$, we estimate the power consumption of the partitioned RegFiles assuming that the operating frequency of a RegFile decides that of the microprocessor. The result of the estimation is that power consumption is reduced to 17% in the 8*1 configuration, and to 33% in the 4*2 configuration compared with the non-partitioned 1*8 configuration in the average of MediaBench suite. We find that the power consumption of a RegFile is dominated by its energy dissipation. This also means that the highly partitioned non-C configurations are still attractive solution to reducing the power consumption significantly.

4. Related Work

In order to obtain higher performance in dynamically-scheduled clustered architectures, there are many proposals for instruction steering schemes and their comparisons in literature [5], [6], [10]. However, degree of the partitioning and organization of the partitioned RegFiles are also the other factors that determine performance of a clustered architecture.

Zyuban and Kogge evaluate a clustered architecture with non-C RegFile organization in terms of energy efficiency [11]. However, they do not discuss the relationship between sources of IPC decrease and degree of partitioning and organizations of partitioned RegFiles.

Our communication model of non-C RegFiles is similar to the model in [11]. They use a remote access buffer (RAB) to feed data to remote PEs while our model uses a buffer to save data from remote PEs. Intrinsically, these two timing models are the same.

A partitioned RegFile organization in [5] inserts copy instructions between dependent instructions dynamically. The required register instances are replicated across PEs, so this can be seen as partially non-C RegFile organization. The timing model of processing dependent instructions in partially non-C RegFile organization is the same as that of our non-C RegFile organization. However, this organization must add the extra write ports to receive data from remote PEs and the more registers to hold the copied instances while this organization does not require any extra buffers for inter-PE communication network.

Seznec and Rochecouste proposed register Write Specialization and register Read Specialization for clustered architecture [14]. Based on MC RegFile organization, they force functional units to write and read the specific registers in specific PEs. The register write specialization enables the number of ports of registers to reduce. The register read specialization can reduce the number of replicated registers.

This RegFile organization is referred as MC write specialization RegFile organization.

Brown and Patt evaluated performance of MC write specialization RegFile organization and partially non-C RegFile organization [20]. They concluded that organization with the write specialization RegFiles can achieve about 10% higher IPC than that with the partially non-C RegFiles. However, they did not vary the number of registers in partially non-C RegFile organization. In this paper, we find that if we adjust the number of registers, the non-C RegFile organization can achieve higher IPC than that of the MC RegFile organization. In future, we should evaluate partially non-C RegFile organization varying the number of registers per PE.

Previously, we have reported evaluation of a clustered architecture with various degrees of partitioning and organizations of RegFiles [21]. In this paper, we further consider power and performance metrics. In addition, we also evaluate effects of the bandwidth of a inter-PE communication network fabric. The results obtained in this paper are effective in understanding potential of a clustered architecture more clearly.

5. Conclusions

In this paper, we have investigated appropriate degree of partitioning and organizations of partitioned register files in clustered architectures. From execute-driven simulation, we have observed that IPC is decreased as degree of partitioning is increased. We have found that organization of RegFiles and degree of partitioning have a larger impact on IPC rather than the number of registers per PE, and configurations with non-consistent register files can make use of their partitioned resources more effectively. At the same time, we have evaluated access time and energy dissipation of register files using CACTI model. We have found that the highly partitioned non-consistent register file configurations can receive benefit of the partitioning in terms of operating frequency and access energy of register files more than the other configurations.

In order to give further consideration of performance and power, we have examined relationship between IPS and the product of IPC and operating frequency of register files. The results have suggested that highly partitioned non-consistent configurations tend to gain more advantage in performance and power.

In future, we plan to estimate latency and impact for the partitioning and clustering of the other global microarchitectural structures such as as issue queue, inter-PE communication network and processor front-end.

References

[1] V. Zyuban and P. Kogge, "The energy complexity of register files," Proc. 1998 International Symposium on Low Power Electronics and Design, pp.305–310, 1998.

[2] J.L. Cruz, A. González, M. Valero, and N.P. Topham, "Multiple-banked register file architectures," Proc. 27th Annual International Symposium on Computer Architecture, pp.316–325, 2000.

[3] R. Balasubramonian, S. Dwarkadas, and D.H. Albonesi, "Reducing the complexity of the register file in dynamic superscalar processors," Proc. 34th Annual International Symposium on Microarchitecture, pp.237–248, 2001.

[4] S. Palacharla, N.P. Jouppi, and J.E. Smith, "Complexity-effective superscalar processors," Proc. 24th Annual International Symposium on Computer Architecture, pp.206–218, 1997.

[5] J.M. Parcerisa and A. González, "Reducing wire delay penalty through value prediction," Proc. 33rd Annual International Symposium on Microarchitecture, pp.317–326, 2000.

[6] A. Aggarwal and M. Franklin, "An empirical study of the scalability aspects of instruction distribution algorithms for clustered processors," Proc. IEEE International Symposium on Performance Analysis of Systems and Software, pp.172–179, 2001.

[7] R.E. Kessler, "The alpha 21264 microprocessor," IEEE Micro, vol.19, no.2, pp.24–36, 1999.

[8] J. Llosa, M. Valero, and E. Ayguade, "Non-consistent dual register files to reduce register pressure," Proc. 1st IEEE Symposium on High-Performance Computer Architecture, pp.22–31, 1995.

[9] G. Reinman and N.P. Jouppi, "CACTI 2.0: An integrated cache timing and power model," tech. rep., WRL Research Report 2000/7, 2000.

[10] Y. Sato, K. Suzuki, and T. Nakamura, "An operand status based instruction steering scheme for clustered architectures," Proc. 2005 International Conference on Computer Design (CDES'05), pp.168–174, 2005.

[11] V.V. Zyuban and P.M. Kogge, "Inherently lower-power high-performance superscalar architectures," IEEE Trans. Comput., vol.50, no.3, pp.268–285, 2001.

[12] R. Desikan, D. Burger, and S.W. Keckler, "Measuring experimental error in microprocessor simulation," Proc. 28th Annual International Symposium on Computer Architecture, pp.266–277, 2001.

[13] D. Burger and T.M. Austin, "The simplescalar tool set, version 2.0," Compututer Architecture News, vol.25, no.3, pp.13–25, 1997.

[14] A. Seznec, E. Toullec, and O. Rochecouste, "Register write specialization register read specialization: A path to complexity-effective wide-issue superscalar processors," Proc. 35th Annual ACM/IEEE International Symposium on Microarchitecture, pp.383–394, 2002.

[15] I. Park, M.D. Powell, and T.N. Vijaykumar, "Reducing register ports for higher speed and lower energy," Proc. 35th Annual ACM/IEEE International Symposium on Microarchitecture, pp.171–182, 2002.

[16] C. Lee, M. Potkonjak, and W.H. Mangione-Smith, "MediaBench: A tool for evaluating and synthesizing multimedia and communicatons systems," Proc. 30th Annual ACM/IEEE International Symposium on Microarchitecture, pp.330–335, 1997.

[17] J.L. Henning, "SPEC CPU2000: Measuring CPU performance in the new millennium," Computer, vol.33, no.7, pp.28–35, 2000.

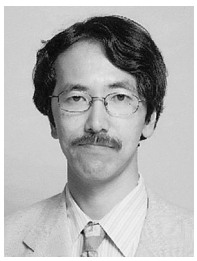
[18] E. Borch and E. Tune, "Loose loops sink chips," Proc. 8th Ingternational Symposium on High-Performance Computer Architecture, pp.299–310, 2002.

[19] K. Farkas, N. Jouppi, and P. Chow, "Register file design considerations in dynamically scheduled processors," Proc. 2nd IEEE Symposium on High-Performance Computer Architecture, pp.40–51, 1996.

[20] M.D. Brown and Y.N. Patt, "Demand-only broadcast: Reducing register file and bypass power in clustered execution cores," tech. rep., The University of Texas at Austin, TR-HPS-2004-001, 2004.

[21] Y. Sato, K. Suzuki, and T. Nakamura, "Partitioned register file designs for clustered architectures," Journal of Information, vol.9, pp.119–134, Jan. 2006.

Yukinori Sato received the BS degree, and the MS and Ph.D. degrees on Information Sciences from Tohoku University in 2001, 2003, 2006 respectively. He is a joint research member at the Graduated School of Information Sciences, Tohoku University. He is also engaged in embedded processor system design at Sendai Software Development Center of FineArch Inc.. His research interests include high-speed and low-power microprocessor architecture and its compiler design.



Ken-ich Suzuki received the B.E degree, Master degree on information sciences, and Ph.D. from Tohoku University in 1992, 1994, 1997, respectively. He worked for Miyagi National College of Technology as an assistant professor from 1997 to 2003. From 2003, he is an assistant professor at Graduate School of Information Sciences, Tohoku University.

Tadao Nakamura received the IEEE Computer Society Taylor L. Booth Award in 2004. He has been Organizing Committee Chair of the IEEE COOL Chips conference series fully sponsored by the IEEE Computer Society. He is an IEEE Fellow.