AN INTERCONNECTION NETWORK OF ASCA: A MULTIPROCESSOR FOR MULTI-GRAIN PARALLEL PROCESSING

Keisuke Iwai* Tomohiro Morimura * Takashi Fujiwara*
Katsuto Sakamoto** Takahiro Kawaguchi* Keiji Kimura***
Hideharu Amano* Hironori Kasahara***
* Department of Computer Science, Keio University
3-14-1, Hiyoshi Yokohama 223 Japan.
** Hitachi, Ltd. Software Development Center
5030, Totsukacho, Totsukaku, Yokohama, 223 Japan.
*** School of Science and Engineering 3-4-1 Okubo, Shinjuku, Tokyo, 169 Japan
{iwai,morimura,fujiwara,sakamoto,kawaguti,hunga}@aa.cs.keio.ac.jp
{kimura,kasahara}@oscar.elec.waseda.ac.jp

Abstract

The multi-grain processing scheme is hopeful method to exploit the highly effective performance of multiprocessors. Although this scheme is useful for various multiprocessors, maximum performance will be obtained on a dedicated multiprocessor architecture whose processor, memory system and interconnection network are tailored for it.

Such a processor, ASCA is proposed in this paper and initial evaluation results are shown.
KEYWORDS:*multiprocessor,multi-stage interconnection network, multi-grain parallelizing compilation scheme,explicit synchronization*

1 INTRODUCTION

Computing on multiprocessor systems, it is important to exploit an inherent parallelism from sequential programs and to schedule parallel tasks. In order to exploit further parallelism, schemes that use parallelism among tasks which are loops, subroutines and basic blocks with loop-level parallelism have been investigated [1,2]. Also to exploit further parallelism, a multi-grain automatic parallelizing compilation scheme has been proposed[3]. It can exploit parallelism of the user program in various levels: the coarse grain parallelism (macro dataflow computation), medium grain parallelism (loop-level parallelism), and near-fine grain parallelism (statements level parallelism)[4].

Although this scheme is useful for various multiprocessors, maximum performance will be obtained on a dedicated multiprocessor architecture whose processor, memory system and interconnection network are tailored for the scheme.

For this purpose, we propose a multiprocessor system ASCA(Advanced Scheduling oriented Computer Architecture) for efficient execution of multi-grain parallel processing.

ASCA provides various mechanisms for this purpose including a dedicated communication mechanism which is used efficiently both for coarse grain and near fine grain parallel processing, and a custom designed processor for static scheduling.

This paper focuses the ASCA and its interconnection network R-Clos. First, the structure of the R-Clos and conditions for emulating hierarchical multi-bus model is introduced. Then, the loss of the communication and performance of R-Clos are preliminary evaluated in actual condition of near-fine grain parallel processing with ASCA simulator.

2 ASCA

ASCA (Fig.1)adopts a multi-grain parallelizing compilation scheme. ASCA consists of processing clusters (PCs) and a centralized shared memory (CSM) to store shared variables. A PC includes four or eight processing elements (PEs), and connected with a novel interconnection network called Recursive Clos (R-Clos). R-Clos is a hierarchically extended Clos network with extra intermediate stage, and can emulate hierarchical multi-bus model easily.

A PE consists of a custom microprocessor called MAPLE, distributed shared memory (DSM), local memory(LM) and various controllers.

3 HIERARCHICAL MULTI-BUS MODEL

As an interconnection model for compiler and scheduler, we adopted a hierarchical multi-bus structure shown

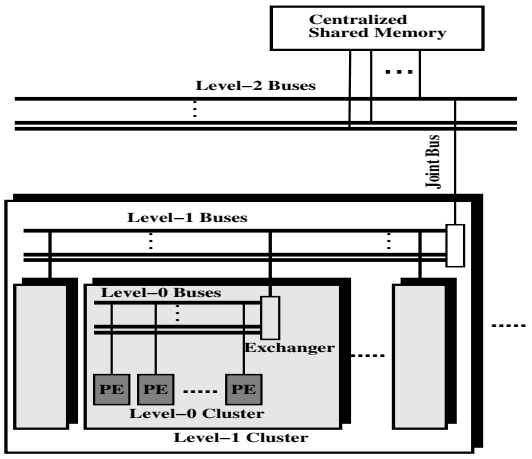

Figure 1: MULTIPROCESSOR SYSTEM ASCA

in Fig.2. In this model, the same number of buses as PEs are prepared in a cluster. One of any buses can be used as a joint for the higher level cluster.

This structure can be extendable as the number of PEs increases. Because of its simple structure, a compiler can avoid conflicts of each bus using static scheduling in each level.

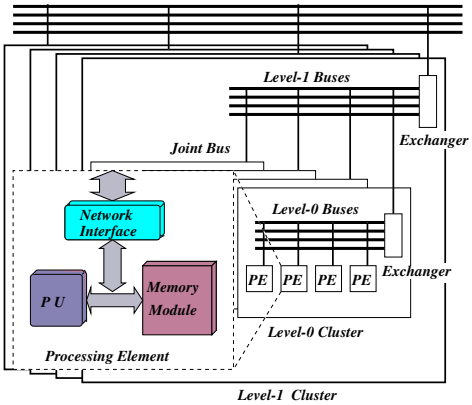

Figure 2: HIERARCHICAL MULTI-BUS MODEL

4 MULTI-STAGE INTERCONNECTION NETWORK R-CLOS

Although the hierarchical multi-bus model satisfies the requirements for ASCA, such a long buses are difficult to be implemented by electrical reason. Thus, we propose a Multi-stage Interconnection Network (MIN) called Recursive Clos (R-Clos) network for emulating the multi-bus model.

STRUCTURE: The diagram of R-Clos (case of R=4 and PE=64) network is shown in Fig.3. It consists of 3 stages switching element, Distributor, Exchanger, and Concentrator. Traditional Clos network is corresponding to the level-0 cluster. Extended stage links

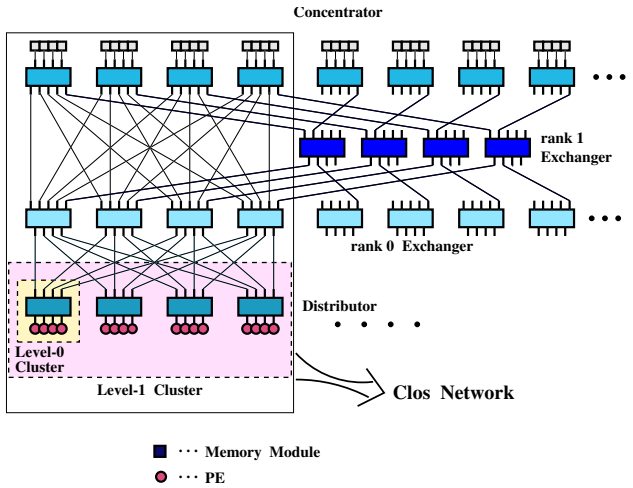

Figure 3: STRUCTURE OF R-CLOS NETWORK

up among basic stages like in Fig.3. Rank-0 Exchanger provides a port named *Extend Port* for connecting with extended stages. The number of extended stages depends on R, standing for the number of layers. That is, R-Clos with R=1 is corresponding to a simple Clos network.

If communication is done only between PEs in a level-0 cluster, this network is conflict free. When communication between other clusters is required, packets may conflict at extended stage. However, if only a PE is selected and issues the packet in a cluster based on the hierarchical multi-bus model, it is conflict free even in the extended stage. Thus, it can emulate multi-bus model without conflicting packets.

5 SIMULATION

Here, an initial evaluation result with a clock-level simulator which was developed for a multiprocessor OSCAR[6] is shown.

5.1 ENVIRONMENT

Simulator: In ASCA, the compiler can fix the time of issuing the packet so as not to conflict with each other in the near-fine grain parallel processing. However, packets for accessing CSMs will conflict in the R-Clos network. In this case, one of conflicting packets is stored in the buffer at the input of switching element. Here, R-Clos with 3 stages is used. The size of switching element of R-Clos network is equal to the number of PEs in a cluster. In this model, the minimum latency for accessing CSM is 4 clocks. Since instruction level simulation takes a lot of simulation time, the number of PE is set to be 6.

5.2 RESULTS OF SIMULATIONS

We analyzed access pattern of actual application shown in Fig. 5.1 which was parallelized by multi-grain auto-

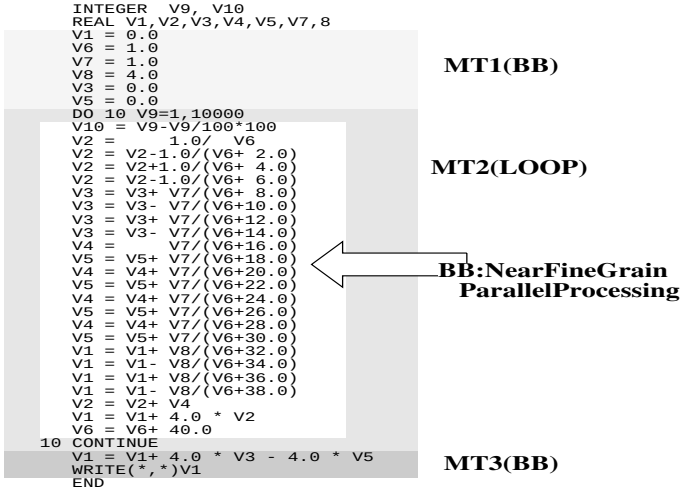

Figure 4: EXAMPLE PROGRAM

matic parallelization compiler[3]. Results of the average latency with this pattern are shown in Table.1.

Although 1,120,498 unit times are required to finish this application, the number of accessing packet is not so large compared with the execution clocks. Latency of between PEs communication is almost same as the steps for passing three stages of R-Clos network. It shows that packets rarely conflicts in communication between PEs. A few conflicts are detected in CSM according to the simulation result. However, the latency is not so stretched.

Furthermore we scheduled communication in the program to prevent packet conflicts in the multi-bus model. Explicit synchronization code for the end of an iteration of loops is inserted by hand. Results are shown in Table 2. All conflicts are occurred in the outside loop which is not scheduled. It shows that R-Clos can be completely schedulable based on the multi-bus model.

These simulation results also demonstrate that R-Clos for ASCA system provides a sufficient bandwidth for an actual application code which is scheduled near-fine grain parallel processing by the compiler.

Table 1: ACCESS PATTERN

	To PE	To CSM
access packet	60,005	190,015
conflict packet	6	53,332
latency(Ave.)[u.t.]	6.0001	4.2806

Table 2: SCHEDULED PROGRAM

	To PE	To CSM
access packet	60,005	190,015
conflict packet	6	1

6 CONCLUSION

In this paper, a multiprocessor system ASCA and its network R-Clos are presented and preliminary evaluation results are shown. Simulation results show that the communication mechanism of ASCA can be completely schedulable and most of packets conflicts can be avoided.

Now, hardware of processing element and network of ASCA are under development. Implementation of the first prototype and extended simulations are our future work.

References

[1] C. D. Polychronopoulos, M. B. Girkar, M. R. Haghighat, C. L. Lee, B. P. Leung and D. A. Schouten, Parafrase-2: An Environment for Parallelizing, Partitioning, Synchronizing, and Scheduling Programs on Multiprocessor, *Proc. of the Int. Conf. on Parallel Processing*, St. Charles IL, pp. II39-48, August 1989.

[2] M. W. Hall, J. M. Anderson, S. P. Amarasinghe, B. R. Murphy, S. W. Liao, E. Bugnion and M. S. Lam, Maximizing Multiprocessor Performance with the SUIF Compiler, *IEEE Computer*, December 1996.

[3] M. Okamoto, K. Aida, A. Yoshida, H. Kasahara and S. Narita, Development of A Practical Level Multi-grain FORTRAN Compiler, *IPSJ SIG Notes*, Vol.96, No.106, pp.43-48, Oct. 1996.(Japanese)

[4] W. Ogata, K. Fujimoto, M. Oota and H. Kasahara, Compilation Scheme for Near Fine Grain Parallel Processing on a Multiprocessor System without Explicit Synchronization, *Proc. of IEEE Pacific Rim Conf. on Communications, Computers, and Signal Processing*, pp.327-332, May 1995.

[5] M. Okamoto, K. Yamasita,H. Kasahara and S. Narita, Hierarchical Macro-Dataflow Computation Scheme on a Multiprocessor System OSCAR, *Proc. of IEEE Pacific Rim Conf. on Communications, Computers, and Signal Processing*, pp.44-49, May. 1995.

[6] H. Kasahara, S. Narita and S. Hashimoto, OSCAR's Architecture, *Trans. of IEICE*, Vol. J71-D-I, No. 8, Aug. 1988.

[7] C. Clos, A study of non-blocking switching networks, *Bell system Tech.J. 32*, pp.406-424, Mar. 1953.

[8] T. Morimura, K. Iwai, H. Amano, The Multi-stage Interconnection Network R-Clos for emulating the hierarchical multi-bus model, *IEICE Tec. Rep.* Vol.97, No.225, pp.85-92, Aug. 1997.(Japanese)