Multi-stage Interconnection Network Recursive-Clos(R-Clos) II: a scalable hierarchical network for compiler directed multiprocessor ASCA

Tomohiro Morimura[‡] Kensuke Tanaka[‡] Keisuke Iwai[‡] Hideharu Amano[‡]
[‡]Department of Computer Science, Keio University
3-14-1 Hiyoshi, Kohoku-ku, Yokohama, 223-8522 Japan
E-mail: {morimura,kensuke,iwai,hunga}@am.ics.keio.ac.jp

Abstract *In 1998, we proposed a multistage interconnection network Recursive-Clos(R-Clos) for a multiprocessor ASCA to schedule the data transfer statically. R-Clos consists of hierarchically clustering multiple Clos networks with extra intermediate stages. Although, R-Clos satisfy our goals as for schedulability and expandability, transfer ability between layers still insufficient. In this paper, we modify R-Clos, and modified R-Clos is named R-Clos II. R-Clos II provides multiple paths to other level-1 cluster, while R-Clos does only one path. A basic scheduling schemes and a heuristic scheduling schemes for level-1 (Clos) network are also introduced. The switch chip of R-Clos II was fabricated on 0.35 μm gate array with dedicated switch architecture for the static packet scheduling. The average latency of this switch chip is 40% lower than that of usual single channel input queuing switch.*

To compare R-Clos II with the original R-Clos, the throughput and the latency are also analyzed with computer simulation. The hardware cost of R-Clos II are only 8% larger than that of R-Clos.

Keywords: multiprocessor,multi-grain parallel processing,multi-stage interconnection network

1 Introduction

In multiprocessor systems, overhead of processor communication is often limits the total performance of parallel processing.

To achieve effective parallel processing including processor communication, we have proposed and developed a multiprocessor system ASCA[4] (Advanced Scheduling oriented Computer Architecture) tailored for multi-grain parallelizing scheme[1][2], one of effective parallelizing schemes. The aim of ASCA is to obtain high effective performance with adopting the multi-grain parallelizing and the compiler static optimization techniques. For the purpose, ASCA has a dedicated hardware structure which can show their behavior clearly to the compiler for static scheduling. An interconnection network is often critical in multiprocessors performance to save the communication overhead between processors. In 1998, Recursive-Clos(R-Clos)[6] was proposed as an interconnection network of ASCA. R-Clos is hierarchically extended Clos network, a 3-stage MIN(Multistage Interconnection Network), with adding extra intermediate stages.

Although high degree of bandwidth within a cluster and schedulability of R-Clos network are suitable for our goal, transfer ability between layers are insufficient, since routing paths to other clusters are restricted.

In the next section, ASCA system and R-Clos are introduced. In section 3, modified R-Clos, a solution of the insufficient bandwidth between layers, named R-Clos II, is proposed. In section 4, the basic scheduling scheme on R-Clos II and a heuristic scheduling scheme to level-1 (Clos) network are introduced. The switch architecture for static scheduling implemented on 0.35μ gate array are described in section 5. The evaluation of R-Clos II compared with R-Clos and Clos network are in section 6. Finally, we conclude about proposed R-Clos II and mention the future works.

2 Interconnection Network for ASCA

2.1 Multiprocessor system ASCA

ASCA is a scalable shared-memory multiprocessor consisting of a large number (256 - 4096) of processing elements(PEs) with a scalable hierarchical interconnection network. As shown in Fig.1, ASCA has a simple hierarchical clustering structure for corresponding to various level parallel processing: near-fine grain, medium grain (loop-division) and coarse grain (macro data-flow) parallelizing. The PE consists of custom RISC processor MAPLE[5] and data transfer controller(DTC)[7].

In ASCA, all data transfer between PEs or PE and memory are managed by the compiler through the processor called the DTC. Hence, data transfer overhead can be minimized with compiler's static scheduling. It is also required that its interconnection network provides non-blocking packet transfer for compiler's scheduling. For the purpose, we have proposed R-Clos.

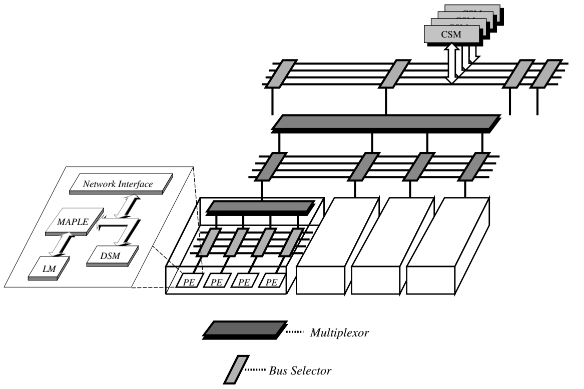


Figure 1: Overview of ASCA

2.2 Multi-stage Interconnection Network R-Clos

R-Clos has been proposed to satisfy the following properties:

1. to provide large bandwidth global network to CSM(centralized shared memory) and high speed local network between neighboring PE's,
2. to give an expandability for large size multiprocessor system, and
3. to guarantee a certain transfer latency for scheduled packet by compiler.

For high speed local network, we focus on 3-stage MIN, Clos network[3], for connecting between neighboring PE's and its memories. As shown in Fig.2, Clos network consists of three stage switching fabrics, distributer, exchanger, and concentrator.

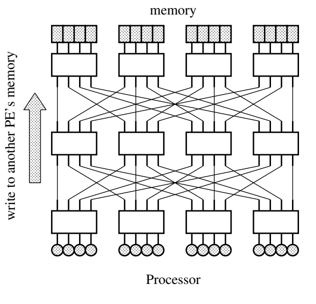


Figure 2: Clos Network (16 PEs)

R-Clos is constructed by hierarchical connecting between each local Clos Network with adding extra intermediate stages(named extended stages) recursively as shown in Fig.3. Expanding to the larger size system, the exchanger switches have an extra input/output port(named extended port) to connect the other cluster network. Packets are transfered to another clustering network through these extended ports and stages.

Here, R-Clos (i,k) represents level-i clustering network with radix-k, the definition of R-Clos (i,k) is as follows:

Definition 1 [Definition of R-Clos (i,k)] *R-Clos $(0,k)$ is a $k \times k$ crossbar switch and R-Clos $(1,k)$ is simple Clos network which consists of k $k \times k$ switch fabrics at each three stages. R-Clos $(2,k)$ are formed k Clos network with connecting together by adding a extended stages which consists of k $k \times k$ switch fabrics(named level-2 exchanger). Therefore 64 PEs are connecting to R-Clos $(2,4)$. That is R-Clos (i,k) are formed k R-Clos $(i-1,k)$ in same hierarchical manner recursively.*

Having hierarchical clustering subnetwork, each level-n subnetwork within R-Clos are called level-n network.

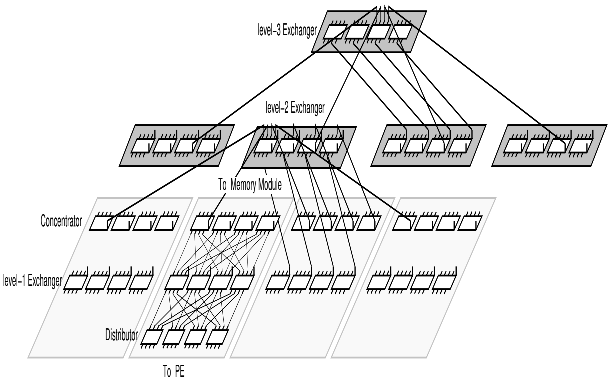


Figure 3: Structure of R-Clos (256 PEs)

If the number of processing elements connected with R-Clos is P and the size (the number of inputs/outputs) of switching element is k, the level R is fixed with the following expression.

$$R = \log_k P - 1$$

The number of additional stages S is as follows:

$$S = (R-1)(R \geq 2).$$

For scheduling the packet transfer to remove its conflicts by compiler, R-Clos can emulate simple hierarchical multi-bus which is connecting between each layer's crossbar (multiple buses) network (Fig.4). Although, R-Clos satisfy our requirements for scalable and schedulable, a vulnerable point still remains and we point out the problem of R-Clos in the next section.

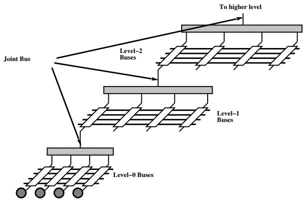


Figure 4: Hierarchical Multi-Bus

3 Multi-stage Interconnection Network R-Clos II

3.1 The problem of R-Clos : Insufficient bandwidth between layers

Since R-Clos has a hierarchical structure, its transfer ability between layers affects the total system performance. In this section, we analyze the hierarchal structure of R-Clos and point out its problem, and indicate the solution for it.

Assume that the source and destination nodes are in the different level-1 network of R-Clos network. In this case, packets are transferred to extended stages through the extra output port of second stage switch (denote it level-1 exchanger). Here, a routing tag T is given by the following:

$$T(\ d_1,\ \underbrace{k, \ldots,\ k}_{r},\ \underbrace{d_{r+1},\ d_r,\ \ldots,\ d_2}_{r},\ d_0) \qquad (1)$$

Note that $d_m d_{m-1} d_1 d_0$ represent labels of the destination with radix-k digits and tag k represents the extended output port for higher layer's switch (higher level exchanger). Packets are routed with these tags in order from the first stage (distributer) to the last stage (concentrator).

According to routing tags, each routing path to other level-1 (Clos) network is settled statically by the output port labeled with the same number as d_1 corresponding to the connected distributer. Therefore, R-Clos is divided into k subnetworks consisting of the same colored switch fabrics illustrated in Fig.5 after the packets traversed distributers. That is, when a packet is sent to output port d_1, it is routed only within same colored switch fabrics in Fig.5. Therefore, they have only one path to the destination belonging to the other level-1 (Clos) network. This means that all packets are blocked at the distributer if requiring output port has been already blocked by another packet, regardless of the fact the other output ports related to the higher level are available. If packets can use these empty output ports to level-2 exchanger, blocking can be avoided.

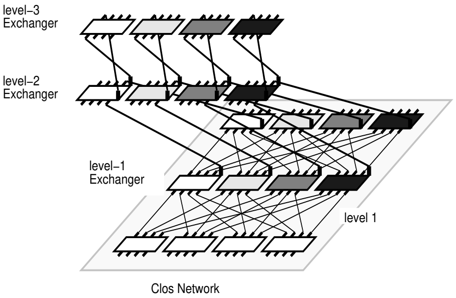


Figure 5: Division into sub networks

3.2 An improvement of connecting between layers: R-Clos II

Before describing the solution, some definitions are introduced. The input stage, middle stage, and output stage are called distributer, exchanger and concentrator respectively. A sequential number is assigned into a switching element with radix-k digits in every stage, and a switching element assigned with number j in distributer, exchanger and concentrator are called D_j, $E_{i,j}$, and $C_{0,j}$ where i represents the exchanger's level.

As shown in Fig.6, direct connection between level-2 stage and level-0 stage causes the division into sub-network.

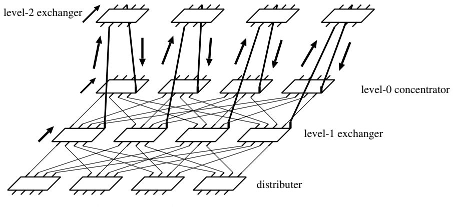


Figure 6: direct connection between level-0 and level-2 network

Therefore, we modify these connections the following:

1. Add an extra stage among the level-2 exchanger and level-0 concentrator.
2. Additional extra stage consists of k $k \times k$ switch fabrics. They are called level-1 concentrators, and expressed as $C_{1,j}$ $(0 \leq j \leq k)$
3. Connect the output port of level-2 exchanger to the input port of level-1 concentrator according to the following:
    - the output port m of $E_{2,j}$ is connected to the input port j of $C_{1,m}$.
4. Connect the output port of level-1 concentrator to the input port of level-0 concentrator according to the following:
    - the output port m of $C_{1,j}$ is connected to the input port k (extended port) of $C_{0,m}$ within R-Clos $(1, j)$.

We call this modified R-Clos network R-Clos II. Unlike original R-Clos, it provides an extra stage (level-1 concentrator) which merges the divided subnetworks illustrated in Fig.7.

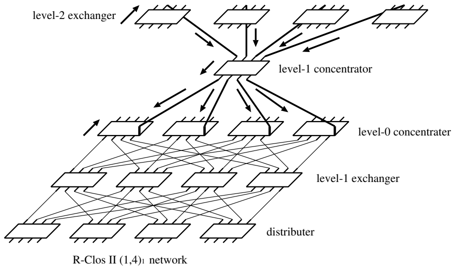


Figure 7: connection after modification

All packets separated with sub-networks are gathered at level-1 concentrator and delivered to a certain level-0 concentrator attached to its destination. As a result, the routing restriction for other level-1 network is removed.

The following routing tags are used in R-Clos II:

$$T(\,*,\, \underbrace{k, \ldots,\, k}_{r},\, \underbrace{d_{r+1},\, d_r,\, \ldots,\, d_2}_{r},\, d_1,\, d_0) \qquad (2)$$

Note that $*$ represents that packets to other level-1 network are allowed to be transferred to any output port. It means that there is no blocking at distributer. Therefore, the routing paths to other level-1 networks become k from only one.

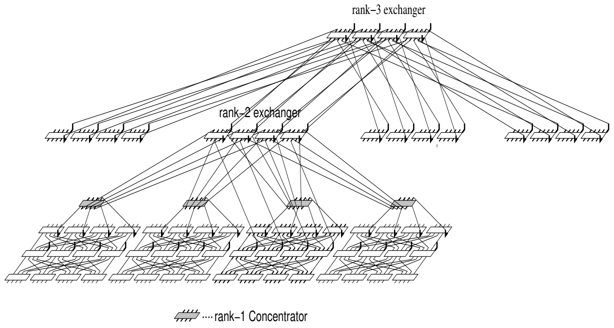


Figure 8: Structure of R-Clos II - 256 PEs -

Fig.8 shows the overview of R-Clos II connecting 256 PEs.

Since proposed R-Clos II is made by little modification of the connection between layers, its property is similar to the original R-Clos except that multiple routes can be constructed and additional latency by adding an extra stage is required.

Little hardware increasing are estimated only by adding the some switch fabrics. The number of additional switch fabrics are expressed by the following equation. $SN_{extra} = \frac{N}{k^2}$, where N represents the system size (the number of connecting PEs). The hardware estimation for R-Clos II compared with that of R-Clos is shown in Section 5.

An important requirement for the interconnection network in ASCA is that a simple scheduling scheme can be adopted. In next section, we describe the basic idea of scheduling the data transfer on this network.

4 Scheduling Schemes for R-Clos II

4.1 Topological Advantage

In case of scheduling on R-Clos, a single routing path to each other cluster is formed for each k layer-to-layer link, and one valid path can be used in spite of having k links between layers.

As mentioned in the previous section, R-Clos II provides k valid paths to other cluster networks, therefore, scheduling flexibility is enhanced.

4.2 Basic Scheduling Scheme for R-Clos II

The following is a basic scheduling scheme for R-Clos II.

- Each output of a switch fabric is administrated by a table with the following items:
  - current time,
  - holding port - which input port holds this output port,
  - holding time,
  - required port queue - a queue which registers the input ports requiring this output port,
  - state - indicating whether it is in HOLD or RELEASE.
- All packets which will be issued t_{sch} are registered to each tables at the compile time.
- Packet transfer is arranged so as not to conflict.
- The above processes are done for all t_{sch}.

This basic sequence is simple but time-consuming. Now, we introduce an effective scheduling scheme within level-1 (Clos) network by extending the above method.

4.3 Scheduling Scheme on level-1 network within R-Clos II

Here, we describe a heuristic scheduling scheme at t_{sch} on the level-1 network. The aim of this scheme is to improve the switch utilization by communication optimization. First, a set of packets issued t_{sch} are defined as P_{sch}. The scheduling scheme uses following data structures:

- AL_n ... Access List
  A list which registers the destination level-0 network associated with a source level-0 network.

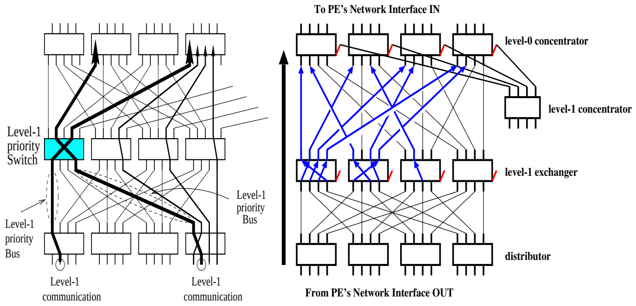
  For example, when packets issued from the level-0 network 0 are sent to labeled $1, 3$ level-0 networks, AL_0 is set as follows.
  $$AL_0 = 1,\ 3$$
- ALT_{cur} ... Current Access List Table
  A table which holds all AL within a level-1 (Clos) network.
- VPC_n ... Valid Port Counter
  A counter which indicates the number of valid input ports of level-0 n cluster. It means how many paths to level-0 cluster n from outer level-0 clusters are available.

Let denote a packet to a local level-0 cluster **level-0 packet** and a packet to an outer level-0 cluster **level-1 packet**. Level-0 packet is given the highest priority for quick transferring between the neighbors, and these packets should not be disturbed by level-1 packets from the outer level-0 cluster.

In order to avoid the conflicts between level-0 packet and level-1 packet, in R-Clos II, arbitrary level-1 exchanger switch is selected as **level-1 priority switch**(Fig.9(a)). In Fig.9(a), level-1 priority switch is the lowest number labeled (the most left side switch). Level-1 packets can use this switch in preference to level-0 packets. This switch is used in the following policies:

- level-1 packets must be routed to this switch.
- level-0 packets may use this switch only when there is no level-1 packet from local cluster.

Using this heuristic, level-1 packets are gathered to the level-1 priority switch and no conflicts between level-1 and level-0 are occurred. The scheduling for R-Clos II is built by extending this priority switch to form multiple paths to outer clusters. First, the compiler assigns each level-1 packet from each level-0 cluster to the level-1 priority switch basically. Next, the compiler chooses an AL from ALT_{cur} as a scheduling target. Then, next level-1 priority switch is arbitrary selected and the packet associated with the element of AL is assigned to this switch. Once assigning to the level-1 priority switch, VPC associated with the level-0 cluster is decremented. So as not to disturb the level-0 packets transfer, the compiler checks the VPC and if VPC is set to 0, further more level-1 packets are not acceptable to the cluster associated with VPC. These unacceptable elements are removed from AL and the packets associated with these elements are also removed from P_{sch} and are added to P_{sch+1} for re-scheduling. After processing an element of AL, compiler chooses the next AL as a scheduling target. The above process is iterated until all elements in ALT_{cur} are scheduled.



(a) single paths by level-1 priority switch

(b) multiple paths after the scheduling scheme

Figure 9: Routing paths of level-1 packets

Fig.9(b) shows an example of the status after adopting this scheduling. In this example, level-1 priority switch is selected in order from the left side. Therefore, level-1 packet's multiple routes are formed from the lowest number labeled (in figure, from left side) switch fabrics.

5 Architectural Supports for the scheduling : MGF Switch Architecture

The architectural supports for the strict static scheduling are also required. In ASCA, the static scheduled packets and non-scheduled packets are mixed in the same traffic. While packets in coarse-grain parallel processing are dynamically issued caused by indeterministic factors such as branches, packets in near-fine grain parallel processing are transfered according to the scheduling.

Therefore, we propose a new switch architecture design, called MGF(Multi channel queuing with Greedy Forwarding) architecture, that can treat two different kinds of communication. The former type uses a short packet for near-fine grain parallel processing in a cluster. Since departure and arrival time of packets in this category is strictly scheduled by the compiler so as not to conflict each other. The switch must pass these packets as quick as possible. We call such packets scheduled packet.

On the other hand, packets used for communication between the central memory and PEs tends to require high bandwidth but the latency can be stretched. They are called non-scheduled packets. The switch provides two 5 x 5 crossbar switches and has two channels for separating the two different type of packets. As shown in Figure 10, schedule packet uses s-channel, while non-scheduled packets use c-channel. Unlike other switching fabrics with virtual channel mechanism, this switch provides two set of crossbars. Therefore both of communication can be sent to output independently in order to improve the throughput of the switch. Then, output module selects a channel and the packet of selected channel are sent to the outside of the switch. Schedule packets have always privilege to be sent to output port.

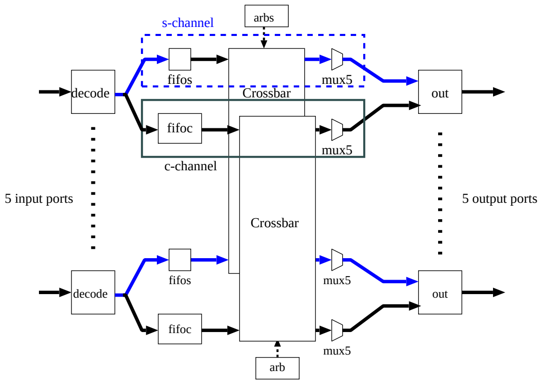


Figure 10: Overview of R-Clos Switch Chip

Moreover, for efficient transferring, the switch adopts the virtual cut-through switching.

The switch chip consists of 5 units, decoder, FIFO queue, arbiter, multiplexor and output selector. While the packet buffer in c-channel of this chip can be stored 8 16-bits flits, that in s-channel can be stored 2 16-bits flits because there is no conflicts between the schedule packets.

We implemented this MGF architecture on 0.35μ process Hitachi Gate Array supported by VDEC. First prototype chip is fabricated in March. The specification of this switch is shown in Table.1.

Table 1: The Specification of Switch Chip

Chip Feature	Hitachi 0.35um gate-array poly 1 Metal 5 190 pins 143k gates
Maximum clock	51.58 MHz
Logics	48167 BC(96334 gates)
Area utilization	67.37 %
Outside data width per link	16 bits
Inside data width per link	32 bits
The number of used pins	185

Simulation by the Verilog-HDL description demonstrates that the average latency of MGF switch are 40% lower than that of single queuing(depth 5) normal switch in exchange for 40% larger costs of hardware.

In next section, we evaluate R-Clos II compared with original R-Clos and Clos network under random traffic with computer simulations. The amount of hardware is also estimated in the section.

6 Evaluation of the transfer performance of R-Clos II

6.1 Evaluation Scheme and Conditions

Although R-Clos II should be used under the scheduling mentioned in the previous section, the throughput and latency are estimated with computer simulation in order to analyze the fundamental switching capacity of R-Clos II. The following two traffic patterns are used in the simulation:

- Random uniform traffic
  All packets are transferred to completely random destinations.
- Localized traffic
  A certain percents of packets are randomly transferred only within a local level-1 cluster (Clos network), and the rest of the packets are randomly transferred to PEs in other clusters.

Simulation conditions are as follows:

- Each switch fabric has input queue which stores 5 packets at maximum.
- If a packet is blocked by another packet, it is stored in the input queue.
- It takes 4 steps to traverse a switching element.

6.2 Results of Simulations

Fig.11 and Fig.12 shows the throughput vs. average latency in 64PEs and 256PEs respectively. The results of Clos are also plotted. Under 64PEs, throughput of R-Clos shows slightly better performance than R-Clos II, in spite of the single routing path. As for average latency, the performance of R-Clos are better than R-Clos II considerably because of the additional latency of one extra stage. On the contrary, under 256PEs, throughput of R-Clos II shows slightly better performance than R-Clos. It demonstrates the costs of getting the multiple routing paths for scheduling in exchanger for the additional extra latency under unscheduled traffic.

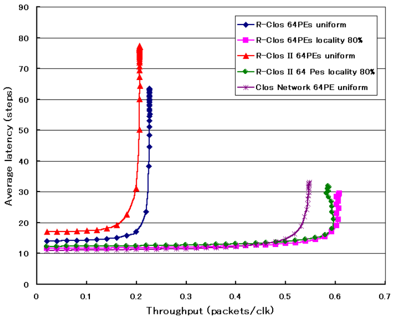


Figure 11: Throughput(packets/clk) vs. Average latency (64PE)

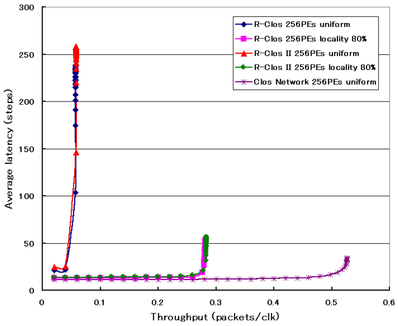


Figure 12: Throughput(packets/clk) vs. Average latency (256PE)

6.3 Hardware Estimation

Here, we consider hardware cost of R-Clos. Let P be the number of connected nodes, that is, the number of inputs/outputs of the network. Let k be the number of inputs/outputs of the switches in basic stage(within level-1 (Clos) network). To estimate the hardware cost, we calculate the number of cross-points. Table2 shows the cross-points of R-Clos II, R-Clos network, and Clos network calculated by the above equations where k is 4. As shown in Fig.2, R-Clos II are only 8% greater than original R-Clos regardless of the system size.

Table 2: Number of cross-points ($k = 4$)

PE	$CP_{rclosII}$	CP_{rclos}	CP_{clos}	CP_{rclII}/CP_{rcl}
16	192	192	192	100%
64	1024	960	1536	106%
256	4224	3968	12248	106%
1024	17024	16000	98304	106%

7 Conclusion

In this paper, we modified R-Clos by adding an extra stage between level-2 and level-0 and change connections between these stages. Modified R-Clos are called R-Clos II. Modification with a little hardware adding(only 8% increased by calculation of cross-points), the routing paths to other Clos network increase k paths from a path. It makes the compiler's scheduling more powerful and give more liberty of rearranging the packet transferring.

The probabilistic simulation results showed that the average latency of R-Clos II is larger than R-Clos because of adding one extra stage. However, for larger system, the degree of increasing latency becomes smaller and as for the throughput, R-Clos II is slightly better performance. It's a trade-off between the compensation of additional stage latency and multiple paths to other level-1 networks. Since R-Clos II is assumed to use in scheduled traffic by compiler, the performance will become better than that of random traffic. Additionally, the switching architecture MGF for scheduling was proposed and it was implemented on 0.35 μm Gate Array. By the synthesis of this chip, the latency of this switch chip are 40% less than normal usual single-queuing switch. In future works, we will implement the R-Clos II connecting with these switch chips and measure its practical performance under running the real programs.

References

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

[2] W. Ogata, K. Fujimoto, M. Oota, H. Kasahara: *"Compilation Scheme for Near Fine Grain Parallel Processing on a Multiprocessor System without Explicit Synchronization'*,

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

[4] K. Iwai, T. Morimura, T. Fujiwara, K. Sakamoto, T. Kawaguti, K. Kimura, H. Amano, H. Kasahara: *"AN INTERCONNECTION NETWORK OF ASCA:A MULTIPROCESSOR FOR MULTI-GRAIN PARALLEL PROCESSING"*, In*Proc. of IASTED International Conference APPLIED INFORMATICS AI'98*, Feb 1998.

[5] T. Fujiwara, K. Sakamoto, T. Kawaguti, K. Iwai, H. Amano: *"A CUSTOM PROCESSOR FOR THE MULTIPROCESSOR SYSTEM ASCA"*, In*Proc. of IASTED International Conference APPLIED INFOMATICS AI'98*, Feb 1998.

[6] T. Morimura, K. Iwai,H. Amano: *"Multistage Interconnection Network Recursive-Clos(R-Clos): Emulating the hierarchical multi-bus"*. Proc. of the ISCA 11th International Conference of PARALLEL AND DISTRIBUTED COMPUTING SYSTEMS, pp.99-104, Sept. 1998

[7] T. Abe, T. Morimura, T. Suzuki, et al: *"ASCA Chip Set: Key components of multiprocessor architecture for multi-grain parallel processing, An International Symposium on Low-Power and High-Speed Chips, COOL Chips IV* to appear , April. 2001