SNAIL-2: a SSS-MIN connected multiprocessor with cache coherent mechanism

Takashi Midorikawa[†], Daisuke Shiraishi[†], Masayoshi Shigeno[†],
Yasuki Tanabe[†], Toshihiro Hanawa[‡]and Hideharu Amano[†]

[†]Department of Information and Computer Science, Keio University
3-14-1, Hiyoshi Yokohama 223 Japan
[‡]Department of Information Technology, Tokyo University of Technology
snail@am.ics.keio.ac.jp

Abstract

Two component architectures for MIN-connected multiprocessors: the Piled Banyan Switching Fabrics (PBSF) and MINC (MIN with Cache consistency mechanism) are evaluated with a real machine SNAIL-2 and an instruction level simulator. The PBSF is a high bandwidth MIN with three dimensional structure, and the MINC is a mechanism for controlling the consistency of private cache modules provided between processors and the MIN. Empirical implementation and simulation results show that the performance improvement of cache controlled by the MINC is significant, and throughput of the PBSF is sufficient if the cache is provided.

Key words: Interconnection Networks, Parallel Architectures, Performance Evaluation

1 Introduction

Multistage Interconnection Networks(MINs) have been well researched as an interconnection mechanism between processors and memory modules of a multiprocessor, particularly for middle scale multiprocessors. Most of them are blocking networks like the omega network[1], and packets are transferred in the store-and-forward manner between switching elements through bit-parallel(8-64bits) lines. In blocking networks, packets may collide each other even if their destinations are different. In this case, one of the conflicting packets is stored in the packet buffer equipped with each switching element. However, in the conventional MINs, their complicated structure and pin-limitation problem have been a stumbling block to implement. As a result, large scale crossbar switches tend to be used in recent real machines instead of MINs.

As a high speed architecture of MINs, we proposed a novel architecture called *Simple Serial Synchronized* (SSS)-MIN in 1992[2]. In the SSS-MIN, all packets are transferred in the serial and synchronized manner. Synchronized bit-serial communication simplifies the structure/control, and also solves the pin-limitation problem. With the simple structure, a highly integrated chip which works with a high frequency clock rate can be utilized. In order to enhance the performance, the mechanisms of *pipelined circuit switching* and *stage hopping* are introduced. As a result, the SSS-MIN achieves comparable or superior performance to the conventional ones with a much smaller hardware cost.

The first prototype multiprocessor using SSS-MIN, SNAIL was develped in 1994[4], and evaluated with a parallel benchmarks on a simple operating system[5].

Through the empirical evaluation, first, we improved the network topology, and proposed PBSF(Piled Banyan Switching Fabrics)[3] which provides three dimensional structure with multiple outlets. However, since the performance of recent processors is much improved, the network will bottleneck the system even by using the PBSF without cache between the MIN and processors. Thus, we proposed the cache consistency maintenance hardware for MIN called MINC (MIN with Cache Consistency mechanism)[7]. With MINC, the coherent private cache can be provided between the MIN and processors to reduce both the latency and traffic load in the MIN.

In order to evaluate the performance of the avobe two mechanisms, we have been developed the second prototype of multiprocessor with SSS-MIN called SNAIL-2. In this paper, a design, implementation and empirical evaluation of SNAIL-2 are presented with instruction level simulation results. In Section 2, the concept, structure and control of the SSS-MIN are introduced. The architecture of PBSF and MINC are described in Section 3. In Section 4, the design and implementation of the prototype LSI chip and SNAIL-2 are described. Section 5 is for showing performance evaluation results.

2 The SSS-MIN

2.1 Basic operation of the SSS-MIN

The basic operation of the SSS-MIN is illustrated in Figure 1. Like the Asynchronous Transfer Mode (ATM)-based packet switching systems for telecommunication, all packets are inserted into the SSS-MIN serially synchronized with a common frame clock. Since each switching element stores only one bit (or a few bits) of the packet, the SSS-MIN behaves like a set of shift registers with a switching capability. After a delay for passing through all stages, the packet headers come out at the output of the MIN.

When a conflict occurs, one of the conflicting packets is routed in an incorrect direction since the SSS-MIN provides no packet buffers in each switching element. The conflict bit in the routing tag is set when the conflicting packet is routed in the incorrect direction. Such a packet is treated as a *dead packet*, and never interferes with other packets.

When the first flit of an address packet reaches the output of the MIN, all switching elements are set to be either straight or exchange, and a path is formed from the input to the output of the MIN. Here, this path is called the *trace*.

When the *trace* is set, it can be judged whether the packet is correctly routed or not by referring the conflict bit at the output of the MIN. An acknowledge signal (ACK: ACKnowledge or NAK: Not AcKnowledge) is returned to the processor which issues the access request through this *trace* without being stored at each switching element, experiencing only the delay of wires and multiplexors. If the NAK signal is returned, the packet is re-inserted from the input buffer again at the next frame .

2.2 Pipelined circuit switching

The *trace* is utilized not only for the acknowledge signal but also for the transfer of data packets. The address transfer, acknowledge signal and data transfer can be performed in an overlapped manner. Figure 2 shows a timing diagram of these transfers. In frame i, an address packet is transferred. When the head of the address packet arrives at the output of MIN, the *trace* is established and the acknowledge signal is transferred immediately. If the ACK signal is returned, the data packet is also transferred using the same *trace* in frame $i+1$. If the pass through time of the MIN is large, a pipeline can be formed for these transfers. We call this operation overlapped or pipelined circuit switching.

In practice, the depth of the pipeline is usually two since the pass through time of a packet is reduced by the stage hopping mechanism described later.

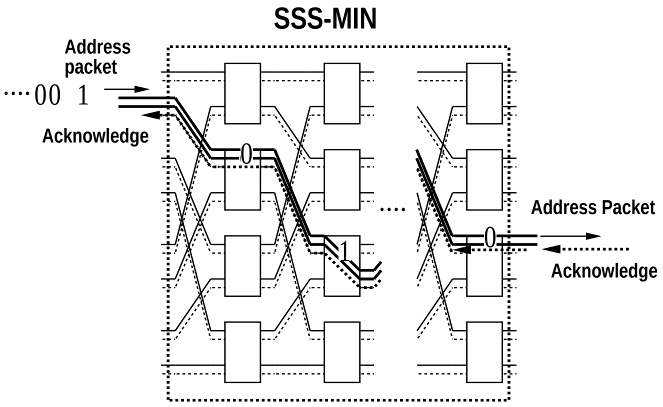


Figure 1: Structure of the SSS-MIN

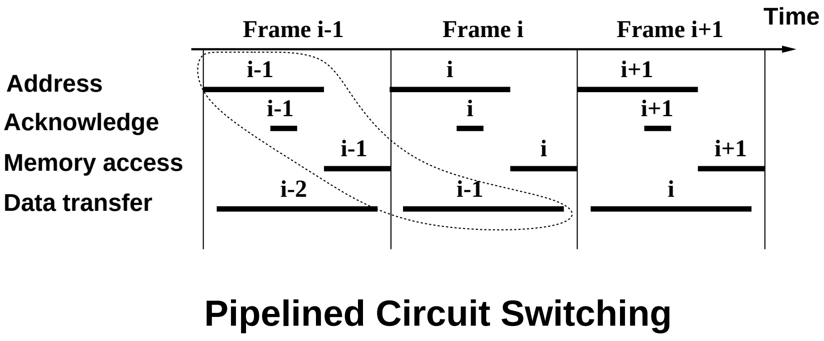


Figure 2: Timing diagram of packets/signal transfers in the SSS-MIN

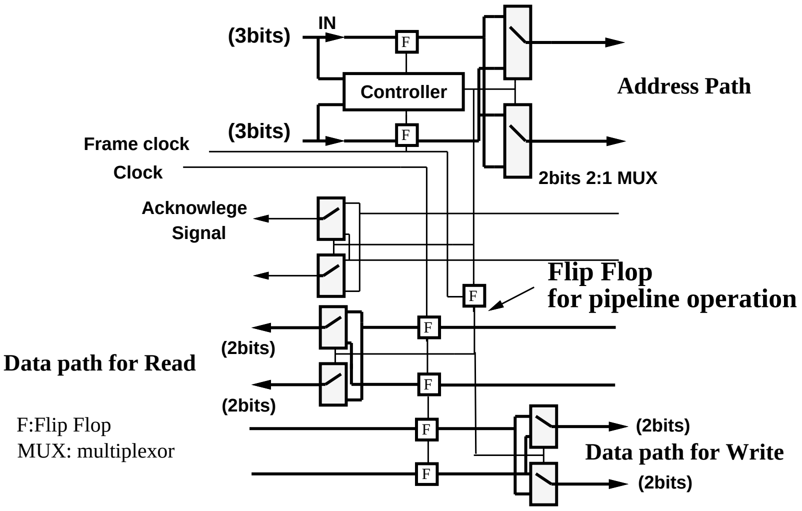


Figure 3: Structure of a switching element in the SSS-MIN

By sharing the *trace*, the structure of the SSS-MIN becomes simple unlike the conventional method which requires two independent MINs for bi-directional transfer. Figure 3 shows the structure of each switching element.

In order to allow the overlapped/pipelined operation, an address packet, a data packet, and the ACK/NAK signal use their own signal lines and multiplexors in a switching element. In the SSS-MIN, the state of an element (straight or exchange) is set by the address packet header. Since the ACK/NAK signals and data packet share the same *trace*, which is set by the address packet, only one controller is required on the address path.

Once the *trace* is established, the rest of the address packet and the data packet do not need to be stored in each switching element. Thus, the flip-flop can be omitted in some stages, and packets can hop over several stages. The latency is much reduced by this mechanism, called *stage hopping*.

The first prototype of the SSS-MIN called SNAIL was developed with the avobe design methodology. It consists of 16 microprocessors (MC68040) and 16 memory modules which are connected with 4 SSS-omega chips. Several parallel applications from SPLASH/SPLASH-2, and a simple operating system called EULASH[6] have been developed on it.

Empirical evaluation results of SNAIL suggest that followings should be improved.

- The throughput of the SSS-omega used in SNAIL is not sufficient for recent high speed processors.
- In order to reduce the latency and relax the load of switches, private cache must be provided between processors and MIN.

To address the first requirement, we proposed PBSF(Piled Banyan Switching Fabrics)[3] which provides three dimensional structure with multiple outlets, and developed a prototype switch[9]. The latter problem can be solved with the dedicated cache control network called MINC(MIN with Cache Consistency mechanism)[7], and the prototype chip was also developed[10].

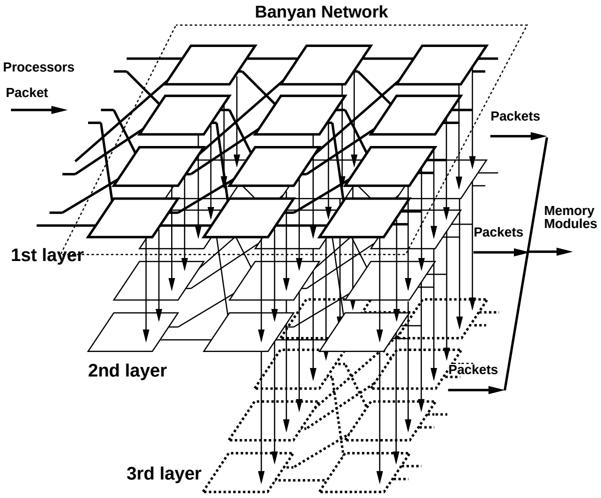


Figure 4: Piled Banyan Switching Fabrics

3 PBSF and MINC

3.1 Piled Banyan Switching Fabrics

The SSS-MIN architecture can be applied to any topology of MIN. In SNAIL, SSS-omega network is used. However, the pass through ratio of common blocking MINs like omega network is not sufficient for for SSS-MIN. Piled Banyan Switching Fabrics or the PBSF[3] is proposed as a connection topology both with a low letency and high bandwidth for SSS-MIN.

In the PBSF, banyan networks are connected in the three dimensional direction (Figure 4). A switching element except in the highest and lowest layers provides four inputs/outputs(two for horizontal, and two for vertical direction.).

Packets are inserted into the highest layer banyan network, and transferred to the horizontal direction. When two packets collide with each other, a packet is fed to the corresponding switching element in the next lower layer banyan network with a clock delay. The vertically transferred packet may collide with both packets which are transferred to the horizontal direction. In this case, one of horizontally transferred packets is selected and sent the next lower layer network.

When three (one from the vertical, and two from the horizontal direction) packets for the same direction conflict in a switching element, a packet is routed to the correct direction, a packet is routed vertically, but the other packet cannot be routed to any direction. Such a packet is routed to the incorrect direction, and treated as a “dead-packet”. On the lowest layer, such a packet is discarded. Only the processor (or the interface units between processors) which issued the correctly routed packets receives the ACK signal through the trace. Other discarded packets are inserted again in the next frame from the input buffer. Small packets buffer is provided for each outlets of the banyan network. When the buffer is full, the arrived packet is regarded as a discarded packet, and inserted again.

3.2 MINC

3.2.1 The directory management method

Unlike several networks which combine MINs and caches[8], the key idea of the MINC is a cache directory scheme called the RHBD[11] which was proposed for a massively parallel processor JUMP-1[12]. This technique can be easily applied to the MIN because of its embedded tree structure.

Using the RHBD, since multicast does not require to access the directory in each hierarchy, quick message transfers can be performed. However, processors which don't share the cache line receive unnecessary messages and they may cause the congestion of the network. To reduce these packets, the pruning cache is introduced in switching elements of a stage.

RHBD scheme

In this scheme, the bit-map of the hierarchical directory is equipped only in the main memory module and reduced using two techniques.

- using the common bit-map for all nodes of the same level of hierarchy (tree), and
- a message is sent to all children of the node (thus, broadcasting) when the corresponding bit in the map is set.

By the combination of these techniques, several schemes are delivered[11]. We adopted the simplest scheme (SM:Single Map), since it is advantageous when the number of processors is not so large.

The reduced directory is not stored in each hierarchy but stored only in the root. Message multicast is done according to the reduced bit-map attached to the message header. Using this method, since multicast does not require to access the directory in each hierarchy, quick message transfers can be performed. However, unnecessary packets are transfered. Although they are just discarded in the cache controller of the processor, it may cause the congestion of the network.

Pruning cache

The major disadvantage of the RHBD is that it requires unnecessary message multicast since it only uses a single bit map or broadcasting in each level of the hierarchy. From the simple evaluation, the RHBD generates more than 100 times messages as really required messages in the worst case[11].

To cope with this problem, "the bit-map pruning cache" is proposed for the MINC chip[7]. The bit-map pruning cache is a small cache of the bit-map referred by the line address of multicasting packets.

When a processor issues the first read access of a cache line, the read request is transferred to the memory module. At the same time, the corresponding pruning cache entry and the bit-map is set. The bit-map is updated when other processors transferred read request of the same cache line to the memory module.

When the invalidation message or updating data is multicast, the bit-map pruning cache is checked. If there is an entry which matches the address, the bit-map of the pruning cache is utilized for multicasting. Otherwise, multicast is done according to usual RHBD schemes(Figure 5.) After the invalidation,the entry is immediately deleted.

3.2.2 Multiprocessors based on the MINC

Figure 6 illustrates a switch-connected cache coherent multiprocessor based on the MINC. This system consists of the following components.

Processing Unit (PU) with a private cache : The private cache is a simple write-through cache which stores copy of the shared memory module. The structure of the cache controller is almost same as the snoop cache controller for bus-connected multiprocessors.

Data Transfer Network : Cache lines, writing scalar data and vectors are transferred with this network. Any type of high bandwidth network like crossbars, MINs, or data exchanger chips can be used.

The MINC chip: Cache coherent messages must be multicast according to the bit-map of the RHBD. The MINC chip is a dedicated network chip which transfers only a part of address and messages to maintain cache consistency of the private cache.

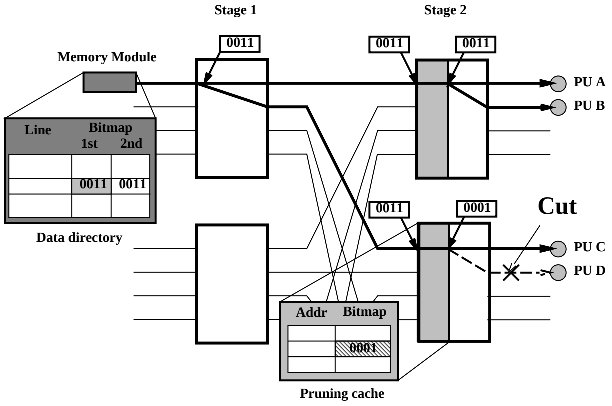


Figure 5: Pruning cache based on the RHBD

Memory module : The shared memory and the bit-map of the RHBD are stored here. The memory controller manages the cache directory and generates packets for the MINC chip and the data transfer network.

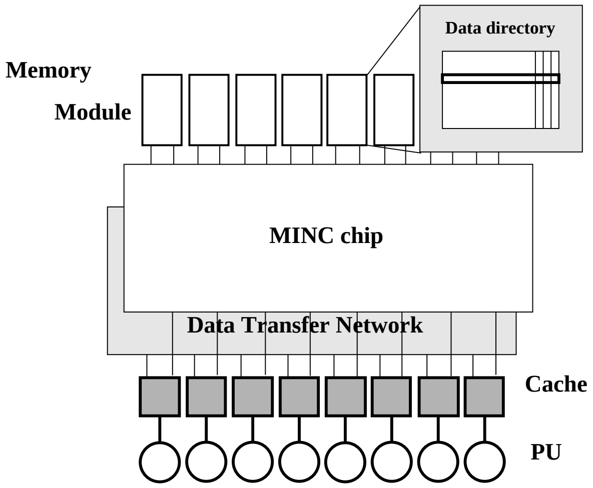


Figure 6: The multiprocessor with the MINC

3.2.3 Multicasting the invalidation messages

When a processor writes into the shared data on its cache, other copies must be invalidated. In this case, the invalidation message must be transferred to all caches which hold the copy of data. First, the write address is transferred to the memory module using the data transfer network, and the RHBD bit-map which indicates the location of copies is pushed into its packet header. Then, this invalidation packet is inserted into the MINC chip.

Each switching element of the MINC checks the bit-map within the packet header, and multicast the invalidation message. Since multiple multicasting messages may conflict each other, acknowledge bit-map is returned immediately to the input buffer(Figure 7.) In the next frame, the packet is transferred again only to the destination corresponding acknowledge bit is 0.

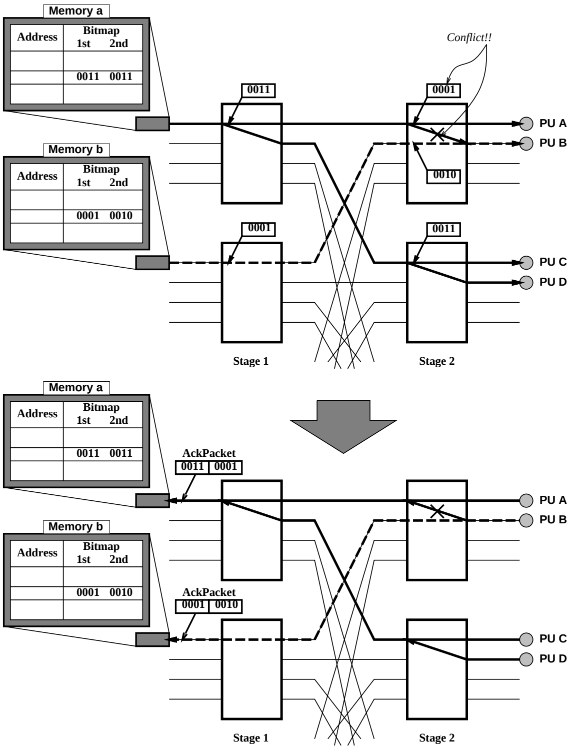


Figure 7: Multicast and Acknowledge

4 SNAIL-2

4.1 The structure of SNAIL-2

SNAIL-2 has been developed to evaluate the performance of the PBSF and the MINC with practical applications. As shown in Figure 8, 16 processing units are connected with the 16 shared memory modules using the PBSF. A processing unit is connected with a host workstation through the Ethernet.

SNAIL-2 consists of Processing Unit(PU), the interconnection network, and the cache coherent network.

4.1.1 Processing Unit (PU) with a private cache

A MIPS compatible 32bit RISC processor with FPU and MMU (IDT 79R3081E-50MJ) is used as a CPU. It provides 2Mbyte local memory, the ROM for booting, Ethernet for connecting the host workstation, and 1Mbyte private cache between the interconnection network. The private cache is two-way set associative cache with write-through policy, and only available for the shared memory module as the latency of the local memory is enough small. The cache controller, which is implemented in Altera's CPLD (Flex10K), has almost the same structure as a snoop cache controller commonly used in bus-connected multiprocessors. 32Kbyte dual port RAM is used for the tag memory, while 1Mbyte single port SRAM is used as a data memory of cache. The size of available cache data memory can be selectable for the performance evaluation.

4.1.2 The interconnection network

The first key component of SNAIL-2 is the PBSF. It was developed by the pilot program of the chip design and fabrication service organization for LSI design education in Japan. In the pilot

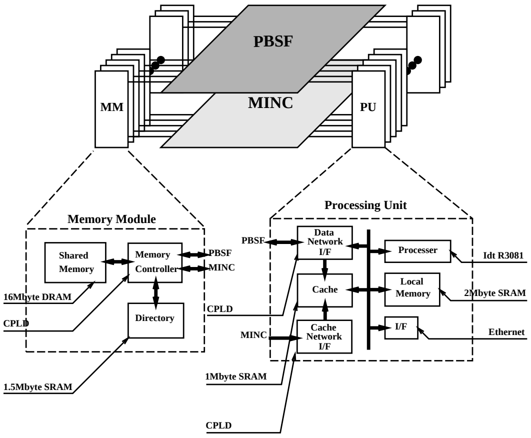


Figure 8: the structure of the SNAIL–2 system

program, the type of chip is limited to be the $0.5\mu m$ CMOS SOG with 16k gates at maximum. Since the PGA package with 176 pins (116 pins for signals) can be used, the size of the network is set to be 16×16 which is the maximum one with this number of pins. The structure and function of the PBSF implemented in the chip is optimized so as to maximize the performance under the pin and gate limitation according to results of the estimation with the computer simulation.

As a result, we decide that the structure of PBSF chip has two layers and multiplexed output, and that the function is 2bit packet priority control, and message combining on the first layer. It provides 2bits for the forwarding path, that is, for the address packets with writing data if necessary, and 1bit for backward path, that is, for the data packets. 17356 gates are implemented on the chip, and is confirmed to work with at least 90MHz clock. That is, the total throughput of the chip is 540Mbytes/sec (34Mbyte/sec for each port).

Eight PBSF chips are used in parallel for performance enhancement. Cache lines, writing scalar data and vectors are transferred with this chip.

4.1.3 Cache coherent network

For maintain the consistency of the private cache, cache coherent messages must be multicast according to the bit-map of the RHBD. The second key component, the MINC chip, is used to transfer only a part of address and messages to maintain cache consistency of the private cache.

The MINC chip used in SNAIL-2 consists of input buffers and 2-stage bi-directional omega networks using 4x4 switching elements, thus, 16-input/output is supported. Considering the limitation of the RAM inside the chip, 256 entries two set associative pruning cache is provided in each switching element. The capacity miss rarely occurs with this size of cache.

The MINC chip was also developed by the pilot program of the VDEC design curriculum. In the pilot program, the type of chip is limited to be the $0.4\mu m$ Chip Express's LPGA(Laser Programmable Gate Array) which has 100k gates (recommended 50k gates and 64Kbit memory cells) at maximum in the PGA package with 391 pins (264 pins for signals). Since the speed limitation of the LPGA, the maximum frequency is 50MHz.

4.1.4 Memory Modules (MMs)

SNAIL-2 provides sixteen 16Mbyte-DRAM memory modules (256Mbyte in total) interleaved in the line size of the private cache (4 words). The memory controller implemented in the Altera's CPLD manages packet analysis/generation, synchronization operations, block transfer and memory refresh. The bit-map of the RHBD used for the cache control is also managed with the memory controller. Fetch&Dec operation, which is an atomic operation of reading and decrementing a word, is provided for the synchronization.

4.2 Implementation

SNAIL–2 consists of two kinds of board. One is the PU-MM board with two PUs and two MMs(Figure9). The other is the network board with the PBSF chip and the MINC chip(Figure10). In the maximum configuration, eight PU-MM boards can be connected with Network board by two flat cables.

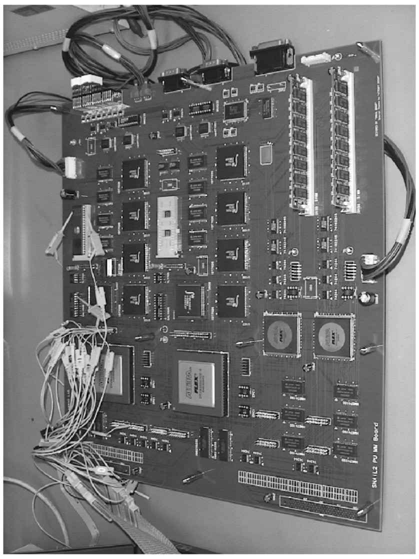


Figure 9: The PU-MM board

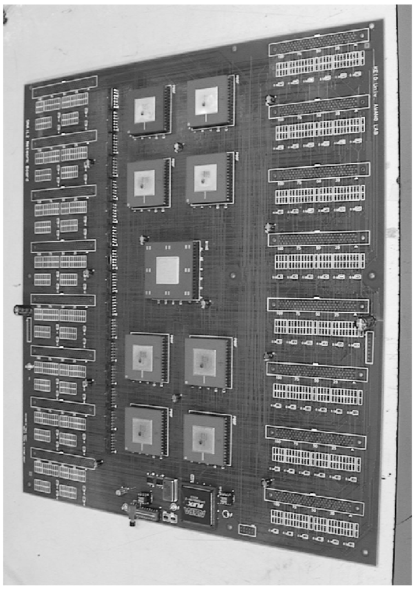


Figure 10: The network board

5 Performance evaluation

5.1 Empirical evaluation

5.1.1 Conditions and Applications

Although SNAIL-2 is designed to be 16 PUs with 16 memory modules, now a system with 4 PUs/ 4 memory modules is available. Although the system are designed to work with 50MHz clock, A low frequency clock (20MHz) is used in this evaluation by the electrical implementation problems around the cables which connects PU-MM boards and Network board.

For evaluation, we implemented four applications: Radix, FFT, LU and Ocean from SPLASH-II benchmark suits[13].

- Radix: A parallel radix sorting program. Since the shared data is not so large, both data exchange and synchronization is not frequently required. 524288 items are sorted.

- FFT: A parallel fast $\sqrt{n}$ basis Fourie's transform with 6-step algorithm. The number of data exchange between processors is minimized. The size is set to be 2^{16}.

- LU: LU decomposition program for a 256 × 256 matrix.

- Ocean: Ocean simulation program. A large data is shared and frequent data exchange is required. The target grid size is set to be 130 × 133.

The barrier synchronization in the application is implemented with Fetch& Dec operation in SNAIL-2. The local data is stored in the local memory, and instructions and shared data is placed in the shared memory.

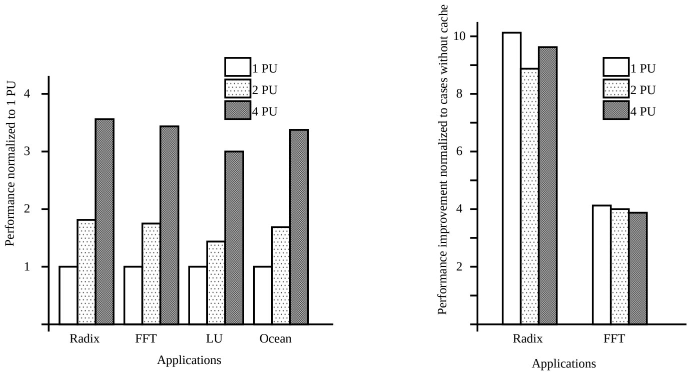
5.1.2 Evaluation results

Table 1 shows access latency to memory modules in SNAIL-2. By electrical problems around cables, the frame clock must be stretched to be 100 clocks in this evaluation, although it can be 40 clocks in the original design. Since, the cache controller provides a write buffer, a single data writing does not hold the processor. However, continuous data writing causes the stall shown in Table 1 since the write through policy is used. If conflict occurs in the network, the access latency is strected to the end of the next frame.

Table 1: Access latency of SNAIL-2 (clocks)

Read	Cache hit	9
	Cache Miss-hit	180-278
	Cache is not used	148-246
Write (Continuous)	Cache hit	143–241
	Cache Miss-hit	143-241
	Cache is not used	160-258

Figure 11 a) shows the performance normalized to those with 1 PU when cache is not used. Almost liner performance improvement is observed except LU that causes load unbalancing in the final stage of the program execution. Since the current system provides only four PUs, network congestion does not degrade the performance.



a) Performance improvement without cace

b) The effect of cache

Figure 11: Perfomance improvement with cache

Figure 11 b) shows the performance improvement normalized to the results without cache. With bugs around the shared memory, applications which use a large shared data is not working with cache. In the current stage of development, only Radix and FFT can be evaluated now. The performance is improved four to ten times those without cache system. The large performance improvement is caused by the large overhead of accessing shared memory as shown in Table 1.

5.2 Evaluation with instruction level simulation

5.2.1 The instruction level simulator

Since, the prototype has not been completely available, there are a lot of limitations for evaluation. The clock frequency and size are limited, and the frame clock must be stretched with the current boards. Moreover, technologies used in SNAIL-2 are already out-of date from the current viewpoint of technology.

For extensive performance evaluation, we developed an instruction level simulator based on the design of SNAIL-2. A simulator development libraries called "ISIS"[14] is used, and all components including the PBSF and the MINC chip are modeled in the clock level accuracy. The same applications: Radix, FFT and LU are used. As shown in Table 2, the default parameters for simulation is set to be the same as SNAIL-2, except the frame clock rate which is inadequately set in the real machine.

Table 2: Simulation environment

Number of PUs	1〜16
Cache Size	256KB / PU
Cache way number	2-way
Cache line size	32 byte
The number of layers of the PBSF	2-layer
Frame clock	40 clock

5.2.2 The performance estimation of full size SNAIL-2

Figure 12 shows performance normalized to that of 1 PU versus the number of PUs.

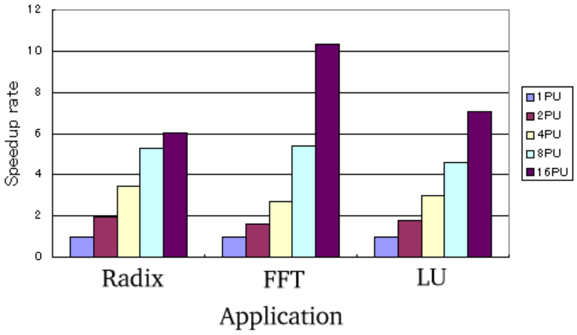
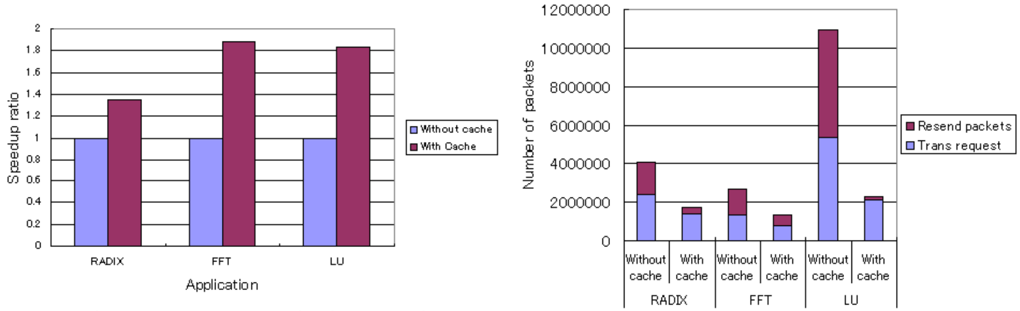


Figure 12: Performance improvement with full size SNAIL-2

From this figure, it appears that the performance is improved even with the full size of SNAIL-2 (16 PUs) especially in FFT. In this figure, the reduction of frame clocks does not much affect to the relative performance improvement.

5.2.3 The effect of Cache

As well as with the real machine, we evaluate the efficiency of the cache with a full system size (16 PUs) of SNAIL-2. Figure 13 a) shows the performance of system with cache normalized to that without cache. From this figure, cache on SNAIL-2 improves the execution time from 30% to 80%. The effect becomes small compared with the results using real machine, since the access time without cache is reduced because of relatively high speed frame clock.



a) Application exection time speed up by using cache

b) Traffic reduction by using cache

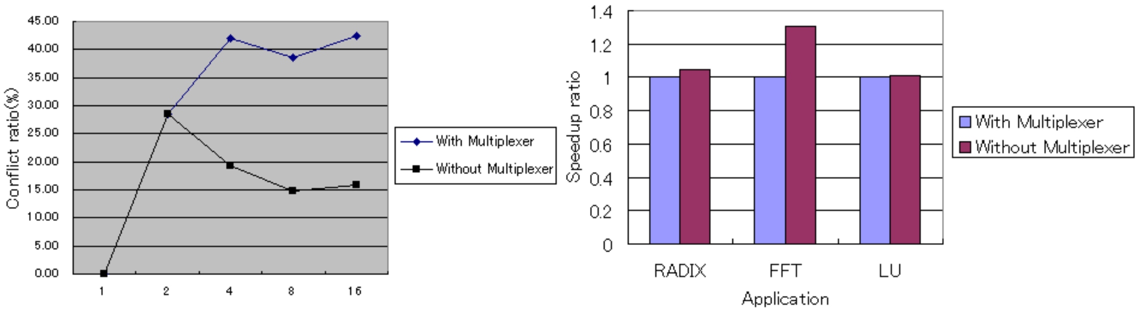
Figure 13: Cache Effect

Figure 13 b) shows packet conflicting ratio. In the case using cache, number of accesses to shared memory is much reduced, and it also lighten the network load, thus packet conflicts is reduced. In the SSS-MIN, packet conflicts causes re-sending in the next frame, and a large extra latency is added to the access time with such packets. Since such re-sending packets are also reduced with the effect of cache, it much contributes the perforamnce improvement.

5.2.4 The effect of outputs multiplexing in the PBSF

Current PBSF chip used in SNAIL-2 provides a multiplexer in its output with the pin-limitation of the chip described in Section 3. That is, the prototype PBSF chip does not actually privide muliple outlet, and this may limit the performance. In order to investigate this influence, the SNAIL-2 using the PBSF chip without the multiplexer is simulated.

Figure 14 a) shows the packet conflicting ratio of the PBSF with and without the multiplexer. In this evaluation, the number of layer is two. The packet conflict ratio with the multiplexer reaches 40%, while it is reduced to 20% without the multiplexer. The execution time is also reduced as shown in Figure 14 b).



a) FFT packet conflict reduction

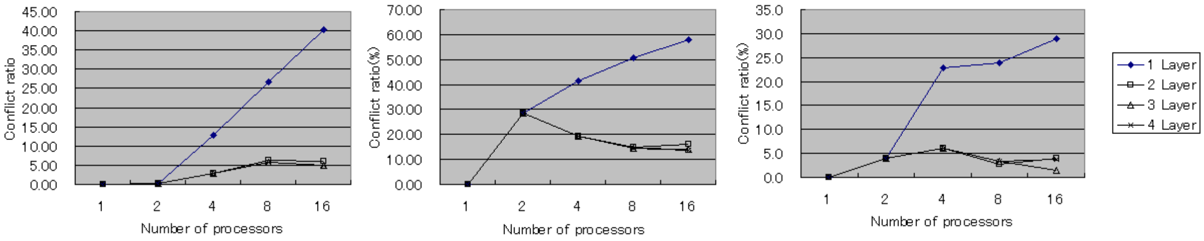
b) Application execution time

Figure 14: The effect of outputs multiplexing

These figures show that the outputs multiplexing in the current PBSF chip degrades performance, and an adequate number of output pins must be provided in the next implementation.

5.2.5 The number of layers in the PBSF

In the current implementation, the layers in the PBSF chip is fixed to be 2. For investigating the optimal number of layers, we changed the number of layers from 1 to 4 in the simulation, and show the packet conflicting ration in Figure 15.



a) Radix

b) FFT

c) LU

Figure 15: Packet conflict rate

In all applications, 2 layers much improve the packet conflicting ratio compared with 1 layer. However, the performance improvement is saturated with 3 or 4 layers. Considering the required hardware and output pins, the PBSF with 2 layers is optimal in this situation.

6 Conclusion

Two component architectures for MIN-connected multiprocessors: the PBSF and MINC are evaluated with a real machine SNAIL-2 and an instruct level simulator. Empirical and simulation results show that the performance improvement of cache controlled by the MINC is significant, and the PBSF can pass the traffic without congestion if the cache is provided.

Since it took a long time for development of SNAIL-2 prototype, the empirical evaluation results shown here is based on out-of date device technologies from the current viewpoint. Using the instruction level simulator, extensive simulation researches for a larger system with the most recent technologies are required.

References

[1] D.H.Lawrie, “Access and Alignment of Data in an Array Processor,” IEEE Trans. on Comput. vol. c-24, No.12, Dec. 1975.

[2] H.Amano, L.Zhou, K.Gaye, “SSS(Simple Serial Synchronized) - MIN: a novel multi stage interconnection architecture for multiprocessors,” Proc. of the IFIP 12th World Computer Congress, Vol.I, pp.571-577, Sept. 1992.

[3] T.Hanawa, H,Amano,Y.Fujikawa, “Multistage Interconnection Networks with multiple outlets,” Proc, of the International Conference on Parallel Processing, pp.I 1-8, Aug. 1994.

[4] M.Sasahara and et.al., “SNAIL: a multiprocessor based on the Simple Serial Synchronized Multistage Interconnection Network architecture,” Proc, of the International Conference on Parallel Processing, pp.I 110-117, Aug. 1994.

[5] J.Yamamoto, T.Fujiwara, T.Komeda, T.Kamei, T.Hanawa, H.Amano, ”Performance evaluation of SNAIL: A multiprocessor based on simple serial synchronized multistage interconnection network architecture,” Parallel Computing 25, 1081–1103, 1999.

[6] J.Yamamoto, D.Hattori, J.Yamato, T.Tokuyoshi, Y.Yamaguchi and H.Amano, “A preprocessing system of the EULASH: an environment for efficient use of multiprocessors with local memory,” Proc. of IASTED/ISMM International Conference on Parallel and Distributed Computing and Systems, pp.68-71, Oct. 1995.

[7] T.Hanawa, T.Kamei, H.Yasukawa, K.Nishimura, H.Amano, ”MINC: Multipstage Interconnection Network with Cache control mechanism,” IEICE Trans. on Information and Systems,” Vol.E80-D, No.9, pp.863-870, 1997.

[8] R. Iyer and L.Bhuyan, “Design and Evaluation of a Switch Cache Architecture for CC-NUMA Multiprocessors,” IEEE Trans. on Comput. vol. 49, No. 8, pp.779-797, Aug. 2000.

[9] T.Kamei, M.Sasahara, H.Amano, “An LSI Implemenation of the Simple Serial Synchronized Multistage Interconnection Network,” Proc. of Asian South Pacific Design Automation Conference, pp.671-672, Jan. 1997.

[10] T.Midorikawa, T.Kamei, T.Hanawa, H.Amano, “The MINC Chip,” Proc. of Asian South Pacific Design Automation Conference, pp.337-338, Feb. 1998.

[11] T.Kudoh, H.Amano, T.Matsumoto, K.Hiraki, Y.Yang, K.Nishimura, K.Yoshimura, Y.Fukushima, ”Hierarchical bit-map directory schemes on the RDT interconnection network for a massively parallel processor JUMP-1,” Proc. of International Conference on Parallel Processing, pp. I-186–I-193, 1995.

[12] H.Tanaka, ” The Massively Parallel Processing System JUMP-1,: IOS Press,ISBN90-5199-262-9, 1996.

[13] S.Woo, M.Ohara, E.Torrie, J.Singh and G.Gupta, ” The SPLASH-2 Programs: Characterization and Methodological Considerations,” Poc. of the 22nd International Symposium on Computer Architecture,” pp.24-36, Jun. 1995.

[14] M.Wakabayashi, H.Amano, ”Environment of Multiprocessor Simulator Developemnt,” Proc. of International Synposium on Parallel Architectures, Algorithms and Networks, pp,64-71, 2000.