PAPER *Special Issue on Architecture, Algorithms and Networks for Massively Parallel Computing*

The RDT Router Chip: A versatile router for supporting a distributed shared memory

Hiroaki Nishi[†], **Ken-ichiro Anjo**[†], *Nonmembers*, **Tomohiro Kudoh**[††],
and **Hideharu Amano**[†], *Members*

SUMMARY

JUMP-1 is currently under development by seven Japanese universities to establish techniques for building an efficient distributed shared memory on a massively parallel processor. It provides a coherent cache with *reduced hierarchical bit-map directory scheme* to achieve cost effective and high performance management. Messages for coherent cache are transferred through a fat tree on the RDT(Recursive Diagonal Torus) interconnection network. RDT router supports versatile functions including multicast and acknowledge combining for the reduced hierarchical bit-map directory scheme. By using 0.5μBiCMOS SOG technology, it can transfer all packets synchronized with a unique CPU clock(50MHz). Long coaxial cables(4m at maximum) are directly driven with the ECL interface of this chip. Using the dual port RAM, packet buffers allow to push and pull a flit of the packet simultaneously.

key words: router, interconnection network, cache coherent distributed shared memory

1. Introduction

JUMP-1 is a massively parallel processor prototype developed by a collaboration between seven Japanese universities[4]. The major goal of this project is to establish techniques for building an efficient distributed shared memory on a massively parallel processor. To achieve this goal, a sophisticated methodology called Strategic Memory System (SMS) is proposed[9][4]. In the SMS, the reduced hierarchical bit-map directory schemes [7] are used for efficient cache management of distributed shared memory.

However, in order to implement the reduced hierarchical bit-map directory schemes efficiently, a high performance versatile network is required. We proposed a novel interconnection network called RDT (Recursive Diagonal Torus)[12][11], and developed a sophisticated router chip for this network. Unlike other router chips which simply transfer packets between processors, a hierarchical multicast mechanism and acknowledge combining mechanism are provided on this router chip.

By using $0.5\mu m$ BiCMOS technology, the clock rate (60MHz) which is equal to the MPU (SuperSparc+) clock is achieved in spite of its complicated operations. Connected cables are directly driven through the ECL interface of the chip. By using dual port RAM, packet buffers can push and pull flits of a packet simultaneously. Hybrid design approach of schematic and VHDL enables the development of this complicated chip with almost 100,000 gates within a year.

In Section 2, JUMP-1 and its interconnection network RDT are briefly introduced. In Section 3, the Reduced Hierarchical Bit-map Directory (RHBD) scheme and implementation of the reduced hierarchical bit-map directory schemes on the RDT is introduced. In Section 4, the implementation of the RDT router chip is described. In Section 5, testing environments of this router chip is described.

2. JUMP-1 and the RDT

2.1 Structure of JUMP-1

As shown in Figure 1[4], JUMP-1 consists of clusters connected to an interconnection network called the RDT[12]. Each cluster provides a high speed point to point I/O network connected with disks and high-definition video devices.

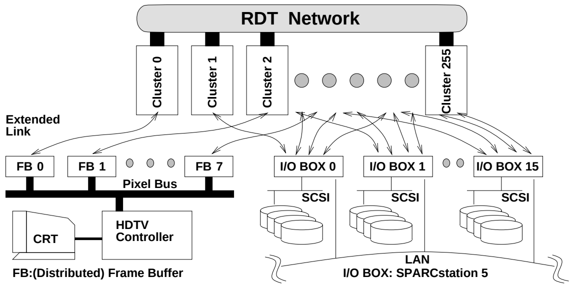


Fig. 1 Structure of JUMP-1

Each cluster is a bus-connected multiprocessor (Figure 2[4]) including four coarse-grained processors (CPU), two fine-grained processors (Memory Based Processor or MBP) each of which is directly connected to main memory and the RDT router chip. CPU is an off-the-shelf RISC processor (SUN SuperSparc+)

Manuscript received January 1, 1995.
Manuscript revised May 14, 1995.

[†] The authors are with the Department of Computer Science, Keio University, 3-14-1 Hiyoshi, Kohoku-ku, Yokohama, 223 Japan

[††] The author is with the Department of Information Technology, Tokyo Engineering University. (Presently with Real World Computing Partnership, 1-6-1 Takezono, Tsukuba, Ibaraki, 305 Japan

which performs the main calculation of the program. MBP (Memory Based Processor), the heart of JUMP-1, is a custom designed fine-grained processor which manages distributed shared memory, synchronization, and packet handling. The first prototype of JUMP-1 consists of 256 clusters, or 1024 processors.

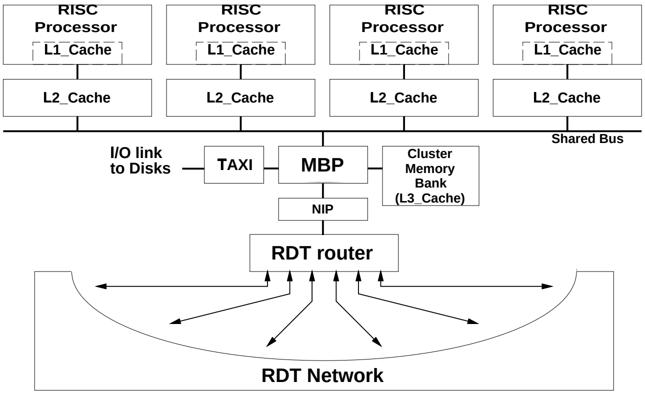


Fig. 2 Structure of a JUMP-1 cluster

2.2 Interconnection network - RDT

The RDT[10], [12] is a network consisting of recursively overlayed two-dimensional square diagonal tori. In order to reduce the diameter, bypass links are provided in the diagonal direction. When four links are added between a node (x, y) and nodes $(x \pm n, y \pm n)$ (n: *cardinal number*) respectively, additional links result in a new torus-like network. A new torus-like network is formed at an angle of 45 degrees to the original torus, and the grid size is $\sqrt{2}n$ times that of the original torus. We call this new torus-like network the rank-1 torus. On the rank-1 torus, we can form another torus-like network (rank-2 torus) by providing additional links in the same manner. Figure 3 shows rank-1 and rank-2 tori when n is 2. The RDT consists of such recursively formed tori.

Recursive Diagonal Torus RDT(n,R,m) can be defined as a class of networks in which each node has links to form base (rank-0) torus and m upper tori (the maximum rank is R) with cardinal number n. Note that, each node can select different rank of upper tori from others.

A large degree makes implementation difficult. JUMP-1 must be scalable to the system with ten thousand nodes, therefore m is to be 1 (degree = 8). For this number of nodes, the rank of the most upper torus is 4. Thus, the RDT(2,4,1) is adopted here.

In the RDT, each node can select different rank tori from others. Thus, the structure of the RDT(2,4,1) also varies with the rank of tori which are assigned to each node. This assignment is called the *torus assignment.*

In this assignment, a node has eight links, four for the base (rank-0) torus and four for one of the upper rank (1-4) tori (most links to upper rank tori are omitted in Figure 4). Note that all nodes have neighboring nodes which is connected three other ranks except each own rank. Therefore, the torus of any rank can be used after at most single message transfer between neighboring nodes. This property reduces the diameter and average distance between nodes.

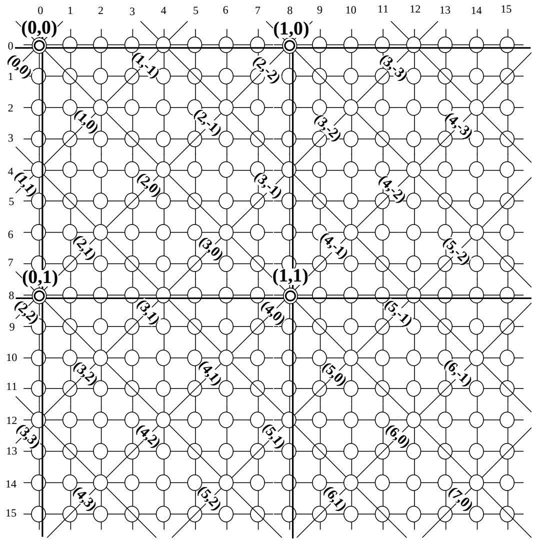


Fig. 3 Upper rank tori

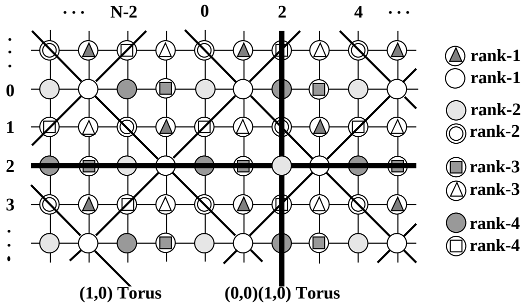


Fig. 4 Torus assignment used in the JUMP-1

3. Reduced Hierarchical Bit-map Directory Scheme

3.1 Hierarchical bit-map directory scheme

Most of conventional non-bus based shared memory multiprocessors equip a cache directory whose entries are associated with the cache lines. However, in a massively parallel machine both with a large number of processors and a large address space, the large amount of memory required for the directory will be unacceptable. In JUMP-1, directory entries are associated with pages while the data are transferred by a cache line[9]. Using this strategy, both the increase of the directory memory and the congestion of the network caused by large message transfers can be avoided.

However, in this case, number of destinations of the coherence maintenance messages increases especially when an update type protocol is used. If the number of destinations is considerable, it will take a long time to send a message if they are sent one after another.

By using the hierarchical bit-map directory scheme of COMA (Cache Only Memory Architecture)[13], a message is transferred to different destinations simultaneously (i.e. multicast) using a tree structured multicasting paths (multicasting tree).

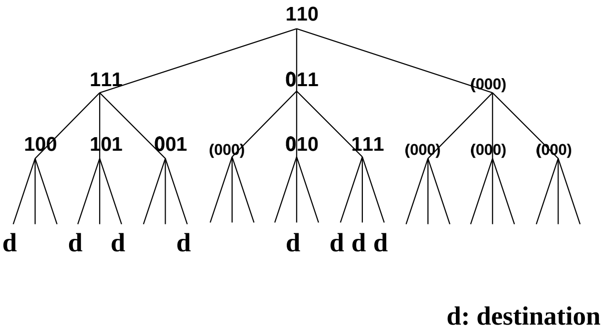


Fig. 5 Hierarchical bit-map directory scheme

Figure 5 illustrates the concept of hierarchical bit-map directory scheme. Leaves of the tree correspond to processors and messages are first sent from the root of the tree. Each node can multicast a message to its branches at a time. To specify the destination leaves precisely, an n-bit bit-map is required for each node when an n-ary tree is used. Thus, for n-ary tree with height m, total of $\sum_{k=1}^{m} n^k$bits are required for each entry. Although the required amount of memory is larger than that of the full-map directory scheme in which n^m bits are required (since there are n^m leaves), messages can be multicast using the tree structured path in the hierarchical bit-map directory scheme.

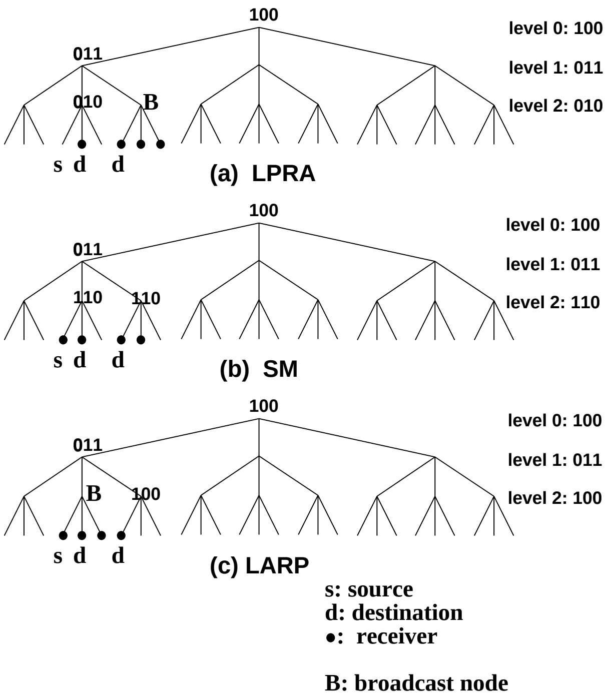
To maintain cache consistency, acknowledge packets are usually required. These packets are transferred from the destination nodes (leaves) to the source node (root) and inform the completion of the invalidation (or the update). Unlike other directory methods, acknowledge packets can be collected and combined at each hierarchy to reduce the network traffic when the number of destination nodes is large.

3.2 The RHBD

Since JUMP-1 is a massively parallel processor, $\sum_{k=1}^{m} n^k$ bit directory entry (for n-ary tree with height m) is not feasible. In order to cope with this problem, the Reduced Hierarchical Bit-map Directory scheme (RHBD) was proposed[8][7]. In this scheme, the bit-map is reduced using the following techniques:

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

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, messages are quickly transfered since no directory access is required at each hierarchy.



(a) LPRA

(b) SM

(c) LARP

Fig. 6 Hierarchical bit-map directory schemes

By the combination of the above two techniques, three schemes, LPRA, SM, and LARP are derived:

LPRA scheme: When multicast starts, the message is sent from the source node to the root of the multicast tree. In the LPRA (Local Precise Remote Approximate) scheme, the bit-map is used only at nodes which are the root of the subtree including the source node. For nodes in other subtrees, the message is broadcast to all children. Figure 6(a) shows an example of this scheme for the 3-ary tree. In this figure, 's' is the source node, and 'd' is the destination to which packet is sent. • indicates nodes which receive the data. It is desirable that the number of nodes which have • but not 'd' is small as possible. Using this scheme, the bit-map is used for local nodes of the source node, while the message is broadcast in the remote subtree marked **B**. This scheme is advantageous when a node sends a message to the group of a remote nodes.

SM scheme: In the SM(Single Map) scheme, all nodes at a level use a unique bit-map as shown in Figure 6(b), and thus, no broadcast is made (unless a bit-map is all-1). This scheme is advantageous when the number of destination is not so large.

LARP scheme: The LARP scheme is a complimentary scheme of the LPRA. In this scheme, broadcast is done at nodes which are the root of the subtree including the source node while a unique bit-map is used in other subtrees like the SM scheme. This scheme is advantageous when the mapping can make the best use of the locality of communication.

3.3 RHBD on RDT

Since the RDT includes a fat tree of tori with multiple root nodes, the RHBD can be implemented in the distributed manner without causing congestion around the root node. The pattern of message transfers for emulating a fat tree is shown in Figure 7.

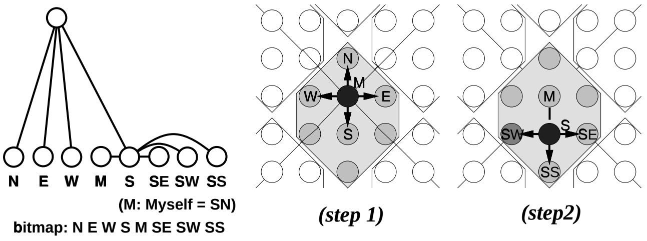


Fig. 7 8-ary tree and bit-map pattern for multicast on RDT

Two steps are required: (1) each node transfers a message to four neighbors, (2) a neighbor (the south in this figure) transfers the message to three neighbors. Thus, if all nodes with rank-i tori execute this pattern, the message is transferred to all nodes with rank-(i-1) tori. By repeating this data transfer from the maximum rank to the rank-0, 8-ary tree is formed on the RDT. In this case, a rank in the RDT directly corresponds to the level of the tree. Moreover, in the RDT, the upper rank torus can be used within a step of message routing. Thus, a message can be directly transferred from the sender node to the root node without using the tree structure. Figure 7 also shows the 8-ary tree relevant to this bit-map pattern for the hierarchical bit-map directory scheme.

In the RDT, nodes which receive the message through the tree whose root rank is 'i' are located around the source node. For larger 'i', the number of such nodes becomes large, thus the area in which a message is multicast becomes wider. We call such an area "territory" of a multicast. Figure 8 shows territories of a multicast from rank-0 and rank-1. Since the territory is always formed around a source node, message multicast to local nodes are performed from a lower rank (thus, with only a small territory).

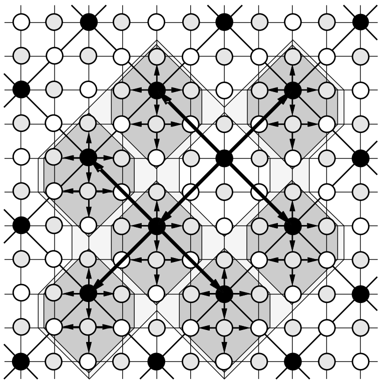


Fig. 8 Territory of a multicast

From the initial performance evaluation, the performance of the RHBD is comparable or better than those of the limited pointer scheme[1] and the chained directory scheme[5][6] when the number of destinations is large and the processes are mapped into processors considering the communication of locality [7].

However, for the efficient implementation of this scheme, the router chip with high speed multicast and acknowledge packet combining mechanism is required.

4. RDT Router Chip

4.1 Structure of the router chip

The structure of RDT router chip which provides all functions of the RHBD is shown in Figure 9. The core of the chip is a 10 × 11 crossbar which exchanges packets from/to ten 18-bit wide links, that is, four for the rank-0 torus, four for the upper rank torus, and two for the MBPs which manage the distributed shared memory of JUMP-1. In JUMP-1, two RDT router chips are used in the bit-sliced mode to form 36-bit width for each link.

All packets are transferred between router chips synchronized by a unique 60MHz clock. In order to maximize the utilization of a link, packets are bi-directionally transferred. The maximum packet length is 16 flits (36-bit width 16-flit length) which can carry a cache line. 3-flit header which carries the bit-map of RHBD is attached to every packet, and the length of the body is variable.

Unlike common router chips[2], following facilities are provided to support distributed shared memory system on a large scale multiprocessor with the RHBD:

- efficient deadlock free multicasting using asynchronous wormhole routing,

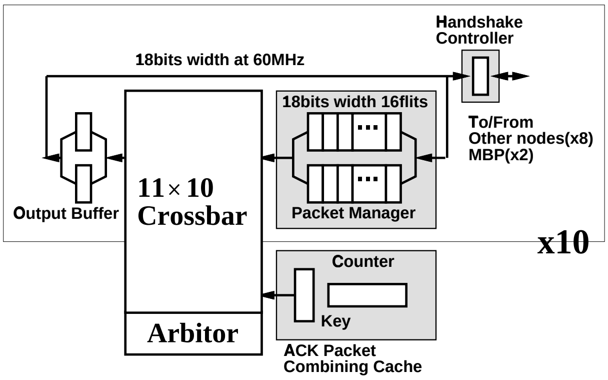


Fig. 9 The structure of the RDT router

- acknowledge packet combining,
- shooting down/setting up, and
- error/handling mechanism.

4.2 Hierarchical multicast

4.2.1 Method of multicast

In this router chip, the asynchronous wormhole routing is adopted to cope with the frequent multicasting. Although a packet can be forwarded to the buffer in the next node while receiving the packet like a usual wormhole routing, a buffer which can hold the maximum sized packet is provided for each virtual channel. When the target buffer is occupied, the entire packet can be stored in the buffer inside the chip. This technique is not the virtual cut through since infinite number of buffers are not provided. However, communication to the buffer outside the chip which may cause the severe performance degradation can be avoided.

The simplest way for multicasting is to wait until all required opposite buffers become empty and multicast at a time. However, the opportunities of multicast are badly reduced when the number of the multicast destinations is increased. In the RDT router chip, each buffer provides a bit-map corresponds to required destinations of a packet. The packet is sent whenever the empty opposite buffers are found, and then the corresponding bits of the multicast bit-map are reset.

4.2.2 Deadlock avoidance

In the RDT, multicast is performed without deadlocks using the modified e-cube routing[3]. With this method, a packet is transferred according to following rules:

- A packet must be multicast in the order of descending ranks. When a packet is multicast at a rank, the order of multicast shown in Figure 10 prevents the deadlock. Rooting on a rank is in the order of descending number of dimensions, and the condition of the deadlock-free routing is satisfied.

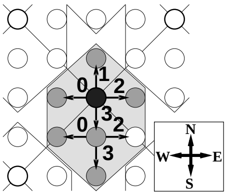


Fig. 10 Deadlock-free broadcast

- The multicast shown in Figure 10 includes two continuous transfers to the same (south) direction. Since the RDT involves the wrap around loop, this may cause a deadlock. Like the e-cube routing for the k-ary n-cube, this deadlock can be avoided by two virtual channels. The channel number is changed when the packet is transferred through the wrap around link to south direction.

- In the RDT used in JUMP-1, each node provides links of only one upper rank torus. If the current node does not provide the required upper rank torus, a packet must be transferred to the node which provides the upper torus through the base torus. Since this use of the base torus may cause deadlock, special virtual channels are required. In the RDT used in JUMP-1, an upper rank is assigned to a node as shown in Figure 4. In this assignment, special virtual channels are only required for horizontal directions (east and west links).

For this modified e-cube routing, two virtual channels are required for south direction of all torus and two for east/west direction of base torus. To resolve this, the RDT router chip provides two virtual channels to each link.

These channels are automatically selected in the deadlock free mode. In the user selection mode, they can be used freely. In this router, the FIFO assumption is ensured since the route of the multicast is fixed.

4.2.3 Implementation of RHBD

Figure 11 shows the format of the multicast packet. It consists of three-flit header for multicasting, a compressed address for acknowledge combining, variable sized body (12-flit at maximum) and a flit for the vertical parity. The first flit includes the most important information: the *Virtual channel*, the *RHBD mode*, the *Status*, the *Top rank of Multicast*, and the multicast bit-map for the current rank. The size of the body is carried in the second flit of the header (*Length*). Bitmaps for multicasting in each rank are stored in the first three flits. Two bits (*RHBD mode*) in the first flit are used to select the RHBD scheme, LPRA/SM/LARP.

Thus, three schemes are selectable for every packet, and the mixture of packets with different schemes is allowed as well. *Status* represents current rooting state (routing step) of the packet.

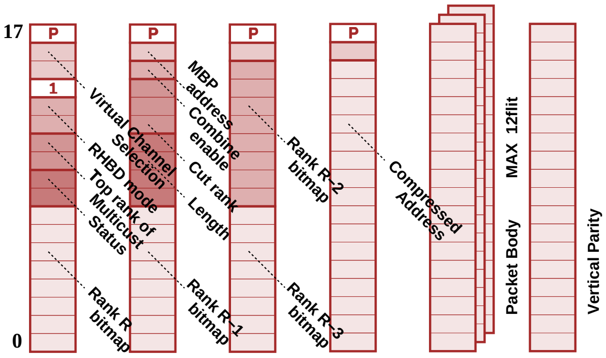


Fig. 11 Multicast packet format

The packet is multicast by two hardwired logic blocks called the bit-map generator and multicast/handshake controller attached to the packet buffer(Figure 12). The multicast/handshake controller is a sequencer which manages the handshake for receiving and sending packets while the bit-map generator is a combinatorial logic which decides the link to which the packet is actually sent.

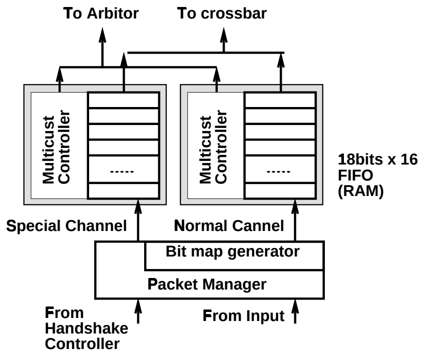


Fig. 12 The structure of the packet buffer

When header of a packet is received, a bit-map generator generates the bit-map for multicasting in this router from the information mostly in the first flit: *Status*, *Mode*, and *Bitmap* for the corresponding rank. As soon as the bit-map is fixed, the multicast/handshake controller tries to send the packet. The packet is immediately multicast to which the receiver is ready, and during the multicasting, the next packet can be inserted to the buffer.

In order to reduce the time of the arbitration, arbitration of the crossbar and the bi-directional transfer lines are performed simultaneously. The priority of crossbar is resolved using a round robin algorithm to avoid starvation. The arbitration and the packet transfer are overlapped, and the next master resides when the current master is sending a packet.

Usually, the RDT router chip can make a multicast bit-map in a single clock cycle from the information in the first flit. When the bit-map included in the second or third flit of the header is required, two or three clocks are required. The router chip needs one clock to check whether the opposite buffer is empty or not, one clock for arbitration, one clock to pass through the crossbar, and one clock to pass through the line and enter the input buffer of the next router. Thus, the message transfer latency is five clocks at minimum, and seven clocks at maximum.

After *Rank R bitmap* in the first flit is used, it is replaced by the *Rank R*-1 *bitmap* in the second flit. Similarly *Rank R*-1 *bitmap* is replaced by the *Rank R*-2 *bitmap*, and *Rank R*-2 *bitmap* is replaced by *Rank R*-3 *bitmap*, respectively. With this mechanism, the bitmap used in the multicast is always stored in the first flit.

4.3 Acknowledge Packet Combining

Acknowledge packets are often required for each multicast in order to maintain cache consistency. If all acknowledge packets are directly transferred to the root node of the multicast, it may cause a severe congestion around the root. To address this problem, acknowledge combining mechanism is provided. When a packet is multicast, a key (compressed address) and the numbers of immediate destinations of the packet are stored in a combining buffer. As shown in Figure 13, the acknowledge packet consists of three flits which carry the routing information, compressed address, and 9-bit body.

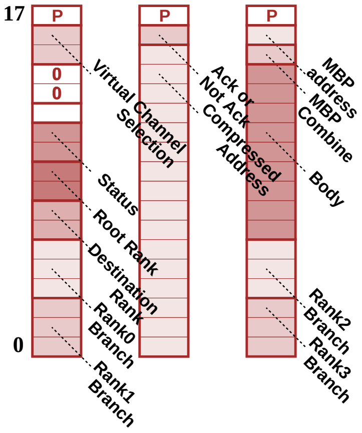


Fig. 13 Acknowledge packet format

In the router, acknowledge packets are automatically combined if the compressed address of each acknowledge packet matches the address of the buffered multicasting packet. Figure 14 illustrates the routing and combining of acknowledge packets. When the multicast starts, the compressed address (in the forth flit of the multicast packet as shown in Figure 11) and the number of packets to be combined are registered in the combining buffer of node A. When an acknowledge packet is returned from a leaf, the compressed address in the second flit of the packet is compared to the registered address independently in each buffer. If the compressed address matches, the counter of the combining buffer is decremented. When the counter becomes zero, the combining buffer generates an acknowledge packet to the upper hierarchy.

In order to avoid the deadlock, acknowledge packets are transferred through a different route from the multicast as shown in Figure 14. The information for routing is stored in the first and third flit of the packet. If another multicast is performed when the acknowledge packet combining buffer is occupied, the combining is done at the MBP.

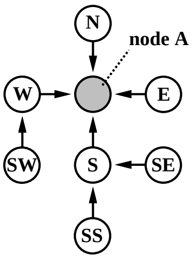


Fig. 14 Routing of acknowledge packets

4.4 Packet Shooting Down and Setting Up

In the case of job switching or debugging, it is sometimes required to flush out all packets in the network. After doing another job or finishing the debugging process, these packets must be returned. These mechanisms are called the packet shooting down and setting up.

When the request for the shooting down is issued, the router changes its mode into *thepacketshootingdownmode*. The request is not only issued by the MBP for job switching or debugging, but also issued when a transmission error inside the router occurs. A simple barrier synchronization line which connects all RDT router chips of every node in cascade can also carry the request. In this mode, all packets in packet buffers are sent to the MBP. After all buffers become empty, the router becomes *thepacketsettingupmode*. In this mode, packets are returned from the MBP. Finally, after finishing the setting up mode, the router starts transmitting. In this way, the FIFO assumption is ensured even if shooting down and setting up are performed.

4.5 Error Handling Mechanism

Reliability is important for a complex router for massively parallel machines. As shown in Figure 11 and Figure 13 parity bits are attached to the header flits to handle the error. The multicast packet also carries the vertical parity flit for body checked in the destination node. Another parity checker is also provided for the internal status of all buffers. Since two chips are used in the bit-sliced manner in JUMP-1, the inconsistency of the internal status parity of two chips indicates an error. If any error is detected, the router changes its mode to *theshootingdownmode*.

Each buffer also provides a watch-dog timer for flushing a packet which stays in a buffer too long. The firing time can be selected in the range from $100\mu sec$ to $100msec$.

4.6 Chip Implementation

$0.5\mu m$ Hitachi BiCMOS SOG which provides maximum of 125K gates is utilized. Lines are directly driven by the ECL interface of this chip. By using internal dual port RAMs, packet buffers can push and pull a flit of a packet simultaneously. The specification of RDT router chip is shown in Table 1. The package of RDT router chip provides 299 pins including 260 signal. Up to 2m cable is driven directly with the ECL I/O buffer, and Bi-CMOS cell is widely used to secure large fanout and high gate speed. 19W power consumption is caused by these Bipolar Cells. To cope with this power consumption, a large heat sink is attached.

The required number of gates are shown in the Table 2. Random logics require 50,000 gates in total while areas corresponding to about 4,000 gates are required for dual-port RAM. Crossbar and arbiter, which are simple but require high performance, are designed in schematic while the complicated controllers are described with VHDL.

Table 1 Specification of RDT router

Power consumption	19.4W
Total Pins	299(Signal 260)
Rate of gate utilization	63
Clock rate	50MHz

Table 2 Number of gates of RDT router chip

Block name	Gates	Blocks	Total	Description
Crossbar	2,927	1	2,927	Schematic
Arbiter	2,736	1	2,736	Schematic
Multicast controller	1,558	10	15,580	VHDL
I/O controller	397	10	3,970	VHDL
Bit-map generator	2,288	10	22,880	VHDL
Acknowledge combining	2,009	1	2,009	VHDL
RAM for buffer	2,021	20	40,420	RAM
Total			90,522	

5. Testing Environment

5.1 JUMP-1 network board

The RDT router has been tested on the JUMP-1 network board shown in Figure 15. On this board, eight RDT router chips and four cluster boards are mounted. Although the RDT router itself works at 60MHz clock rate, the first version of JUMP-1 will work at 50 MHz clock rate. Packet transfer test at 50MHz clock has been already finished. The test was done using 2m coaxial cables which are the maximum length required for JUMP-1.

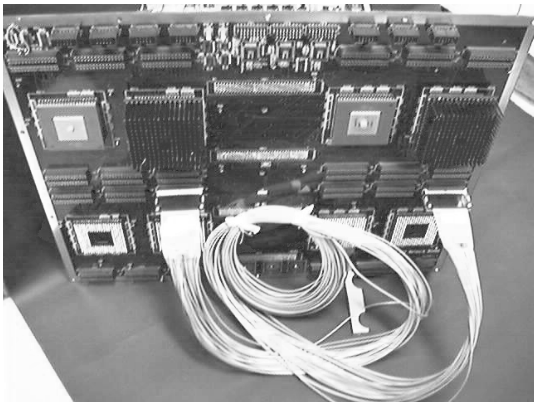


Fig. 15 Photo of JUMP-1 network board

5.2 Workstation Cluster JUMP-1/3

Now, a workstation cluster called JUMP-1/3 which uses the RDT router chip is operational. Eight workstations (SUN Work station SS5) are connected with the Distributed Shared memory Management (DSM) board and the network board consisting of two RDT router chips. The DSM board is a double-width SBus card provided with a shared memory and a simple microprocessor (TMS320C40) which manages the distributed shared memory. Packets for the distributed shared memory management are generated by the microprocessor, and sent to the FIFO on the attached network board. A hardwired logic on the network board manages the FIFO and the RDT router chip, and the distributed shared memory management protocol similar to that of JUMP-1 is implemented. Now, JUMP-1/3 is working at 25MHz, and functions of the RDT router including multicasting and acknowledging packet combining are under verification.

6. Conclusion

The RDT router chip which supports all RHBD schemes is implemented. Using $0.5\mu m$ BiCMOS SOG technology, versatile functions including hierarchical multicasting, combining acknowledge packets, shooting down/restart mechanism, and time-out/set-up mechanisms work at 60MHz clock rate. By using dual port RAM, packets can be sent while receiving, and the ECL input/output line of the chip can drive 2 m coaxial cables directly.

Acknowledgment

The authors would like to express their sincere gratitude to the members of the Joint-University project for their valuable advises and discussions, especially, Dr. Nakajo of Koube University, Dr. Kei Hiraki and Dr. Takashi Matsumoto of Tokyo University, Japan for generous discussions. The authors also express their thanks to Mentor Graphics Japan for supporting design tools. The placement and routing of the SOG is done by Hitachi-Microcomputer Co. Ltd., and fabricated by Hitachi Co. Ltd. The authors also express their sincere gratitude to all staffs of Hitachi/Hitachi-Microcomputer Co.Ltd. for supporting the design.

A part of this research was supported by the Grant-in-Aid for Scientific Research on Priority Areas, #04235130, from the Ministry of Education, Science and Culture.

References

[1] A. Agarwal, R. Simoni, J. Hennessy, and M. Horowitz. An evaluation of directory schemes for cache coherence. In *15th ISCA*, 1988.

[2] W.J. Dally and C.L. Seitz. The torus routing chip. In *Distributed Comput., Vol.1, No.3*, 1986.

[3] W.J. Dally and C.L. Seitz. Deadlock-free message routing in multiprocessor interconnection networks. In *IEEE Trans. on Comput. vol. 36 no. 5*, 1987.

[4] K. Hiraki, H. Amano, M. Kuga, T. Sueyoshi, T. Kudoh, H. Nakashima, H. Nakajo, H. Matsuda, T. Matsumoto, and S. Mori. Overview of the JUMP-1, an MPP prototype for general-purpose parallel computations. In *Proc. of the International Symposium on Parallel Architectures, Algorithms and Networks (ISPAN'94)*, 1994.

[5] D.V. James, A.T. Laundrie, S. Gjessing, and G.S. Sohi. Distributed-directory scheme: Scalable coherent interface. *IEEE Computer*, 1990.

[6] M. Thapar and B. Delagi. Distributed-directory scheme: Stanford distributed-directory protocol. *IEEE Computer*, 1990.

[7] T.Kudoh, H.Amano, T.Matsumoto, K.Hiraki, Y.Yang, K.Nishimura, K.Yoshimura, and Y.Fukushima. Hierarchical bit-map directory schemes on the RDT interconnection network for a massively parallel processor JUMP-1. In *Proc. of the 1995 ICPP*, 1995.

[8] T.Matsumoto and K.Hiraki. A shared-memory architecture for massively parallel computer systems. In *IEICE Japan SIG Reports, Vol. 92, No. 173, CPSY 92-26 (in Japanese)*, 1992.

[9] T.Matsumoto and K.Hiraki. Distributed shared-memory architecture using memory-based processors. In *Proc. of Joint Symp. on Parallel Processing'93 (in Japanese)*, 1993.

[10] Y. Yang, H. Amano. Message Transfer Algorithms on the Recursive Diagonal Tours. In *IEICE Transaction on Infor-*

mation and Systems, Vol.E79-D, No.2, 1993.

[11] Y. Yang and H. Amano. Message transfer algorithms on the recursive diagonal torus. In *Proc. of the IEEE International Symposium on Parallel Architectures, Algorithms and Networks (ISPAN'94)*, 1994.

[12] Y. Yang, H. Amano, H. Shibamura, and T.Sueyoshi. Recursive diagonal torus: An interconnection network for massively parallel computers. In *Proc. of 1993 IEEE Symposium on Parallel and Distributed Processing*, 1993.

[13] E.Hagersten, A.Landin and S.Haridi, "DDM - A Cache-Only Memory Architecture," In *IEEE Computer, Vol.25, No.9, 1992. pp.44-56.*

Hiroaki Nishi received the B.E., M.E. from Keio University, Japan, in 1994, 1996, respectively. He is a Ph.D. candidate in the Department of Computer Science, Keio University, Japan, and Research Fellow of the Japan Society for the Promotion of Science. His research interests include area of designing a interconnection network and its router.

Ken-ichiro Anjo received the B.E., from Keio University, Japan, in 1996. He is Master candidate in the Department of Computer Science, Keio University, Japan. His research interests include area of workstation cluster.

Tomohiro Kudoh received the B.E., M.E., and Ph.D. degrees from Keio University, Japan, in 1986, 1988, and 1992, respectively. Since 1991 to 1996, and 1996 to 1997, he was an assistant professor and associate professor in Department of Information Technology, Tokyo Engineering University, respectively. He joined Real World Computing Partnership in 1997, and is now chief of the Parallel & Distributed Architecture Laboratory. The work in this paper was done when he was at Tokyo Engineering University. His research interests include the area of parallel and distributed processing, and parallel discrete event simulation.

Hideharu Amano received the B.E., M.E., and Ph.D. degrees from Keio University, Japan, in 1981, 1983, and 1986, respectively. He is now an associate professor in the Department of Electrical Engineering, Keio University. His research interests include the area of parallel processing.