Third International Conference on Massively Parallel Computing Systems (MPCS98), Colorado Springs, Colorado, Apr. 6–9, 1998

Memory Centric Interconnection Mechanism for Message Passing in Parallel Systems

Yamin Li, Sanli Li, and Wanming Chu

Computer Architecture Laboratory
The University of Aizu
Aizu-Wakamatsu 965-8580 Japan
{yamin, lsl, w-chu}@u-aizu.ac.jp

Abstract

The Interconnection Network (IN) connecting computing nodes in parallel systems has become one of the key research issues in parallel computer architecture. Traditionally, the INs in parallel systems for message passing have been built on the basis of logic circuits with different topology structures. Currently, the bandwidth of data transmission for message passing in available parallel systems reaches the magnitude of Gbps. In this paper, we investigate a kind of interconnection network based on multiport memory for message passing parallel systems, which is termed Memory Centric Interconnection Mechanism (MCIM). The memory is a Multi-Port Fast Static Memory (MPFSRAM) which acts as mailbox for message passing. With considerably simpler Arbitration and Selection Units, MCIM can fully utilize the bandwidth of MPFSRAM for parallel data transfer to achieve the bandwidth of dozen Gbps for message passing with pipelined mode operation of data sending and receiving, meanwhile with much less complexity than that of the IN based on logic circuits, such as routers with hundreds of lines for each link in conventional Mesh structure parallel systems. Furthermore, the buffers in the mail box provide flexible measures for scheduling the message passing. This paper describes the principle of MCIM, the mechanism of data sending/receiving operations, the arbitration/selection of communication path, and the scheduling flowchart of message passing in the buffer. The paper also gives the experimental simulation results of the MCIM performance.

1. Introduction

The Interconnection Network (IN) for parallel system always remains as the key research issue in parallel computer architecture. Various kinds of INs have been developed with different topologies for MPP systems. Traditionally, such kinds of INs mainly employ logic circuit based technique for switching router design [1][2][3][4][5][6].

Another approach of constructing parallel supercomputing systems beyond MPP is the so-called Networked Parallel Computing (NPC) [7][8]. NPC provides a very promising way due to its feasibility and friendly user interface. The design of IN in NPC is also an important issue. In conventional NPC systems, multiple computing nodes are connected by Ethernet. Recently, instead of Ethernet, some specially custom designed IN switches have been developed for achieving fast data transmission.

However, the data transmission bandwidth of IN in both MPP and NPC, can only reach the magnitude of Gbps at present. Additionally, the custom designed IN for both MPP and NPC costs a considerable portion of system expenses due to its complexity with hundreds of data and address lines per link.

With the rapid development of micro-electronics and very high processor clock frequency, the network latency of an IN currently constitutes a significant part of the total overhead of sending/receiving operations in message passing.

SMP [9][10] with common bus is a good solution for Big Node of parallel systems. However, because of the limitation of physical bus bandwidth, the processor number of a Big Node in SMP with common bus cannot become “bigger”, if no extra measure is adopted.

This paper describes a Memory Centric Interconnection Mechanism (MCIM) for parallel systems. It employs a Multi-Port Fast Static Memory (MPFSRAM) for message passing, surrounded with the so-called Arbitrator and Selector Units (ASUs), which are shared by multiple computing nodes. Each node contains one pro-

cessor as well as multiple processors. The MCIM can fully utilize the high-bandwidth of MPFSRAM for message passing. The multiple ports offer the feasibility of pipelined mode of operations for the message sending and receiving, thus it can reduce the communication overhead of message passing. The MCIM is of much less complexity than the logic circuit based routers in most MPP systems. Comparison with SMP systems, it allows more processors contained in one "Big Node". In MCIM parallel systems, computing nodes share the ASUs, however, because each node only links a few of control signals to the ASUs, each ASU can be shared by more nodes than that of SMP using common bus. Moreover, MCIM provides more flexible mail-box buffers for dynamic allocation of the passing messages in both routing and scheduling among different nodes.

2. The Structure and Principles of MCIM in Parallel System

A MCIM based parallel system by using MPFSRAM of four ports (Port 0-3) is shown in Fig. 1. In the center of this structure, the MPFSRAM is divided into n Mail Boxes for message passing, here we use eight Mail Boxes. Connecting with each memory port, there is an Arbitrator and Selector Unit (ASU). Signal links connecting these four ASUs are needed for passing the control signals.

Each ASU is shared by a group of k computing nodes, in Fig. 1, k is equal to 8. So, this structure consists of 32 nodes (P0-31). The node numbers (0-31) are referred as destination address when a node wants to send a message to the destination node. The value of the two most significant bits of a node number is just equal to the port number of MPFSRAM where the node is located.

The computing nodes send and receive messages through the MPFSRAM, the data width could be of 64 bits. As the latest VLSI technology report [11], the multiport SRAM write operation can be carried out at 350MHz, then one memory port is for WRITE (sending message) and the other port is used for READ (receiving message), the bandwidth of data transfer could reach 22.4Gbps. With the rapid development of VLSI technology, it is expected that the improved multiport SRAM will be available, thus it could gain higher data transfer bandwidth in the MCIM parallel systems.

Data communication path for message passing among different nodes in the MCIM parallel systems needs Arbitration and Selection Unit (ASU). The arbitration function is used by the sending nodes for contending the privilege of using the memory port and the mail box; the selection function is used by the receiving

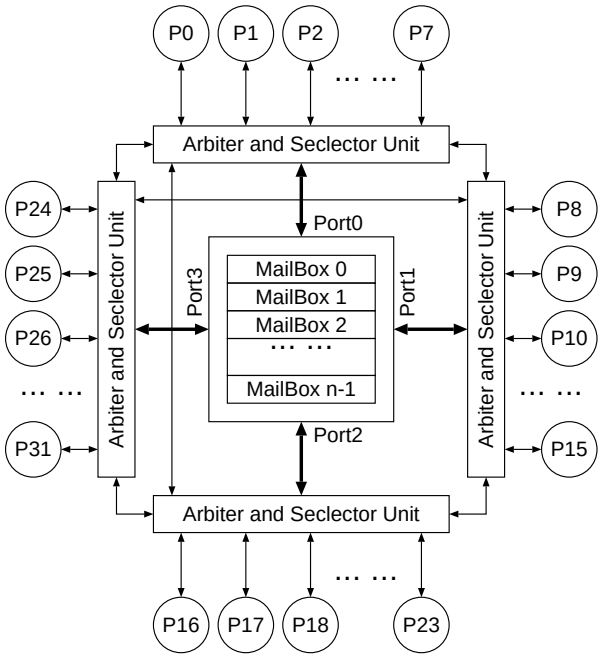


Figure 1. The MCIM Parallel System

nodes for contending the privilege of using the memory port only.

Referring to Fig. 1, if a node, say $P2$, wants to transmit message to a destination node, say $P9$, $P2$ should contend the Port 0. Owing to that other nodes also possibly intend to contend this port at the same time, the arbitration takes place according to the port status and the priority of these nodes. The destination node $P9$ is connected to the port 1. Similarly, $P9$ also needs to contend the port 1 for receiving data.

Each ASU has an one-bit register to indicate the status of the memory port (Busy or Idle). The Busy status means that this port is being used by a node for its data transmission. The Idle status indicates that this port is ready for use.

To coordinate with ASU, every node should have its priority. The priority will be used by the ASU to arbitrate which node can obtain the privilege of using the port when two or more nodes intend to contend this port simultaneously. The procedure for message passing through the mail boxes in MCIM is illustrated in Fig. 2.

Once a sending node obtained the privilege of using the port, it will set the status of the port "Busy". Furthermore, a free (idle) mail box has to be allocated for buffering the sending data. For the mail boxes' arrangement, one solution is to allocate the mail boxes with the same amount as the number of memory ports and to assign a fixed box to its corresponding port. For example,

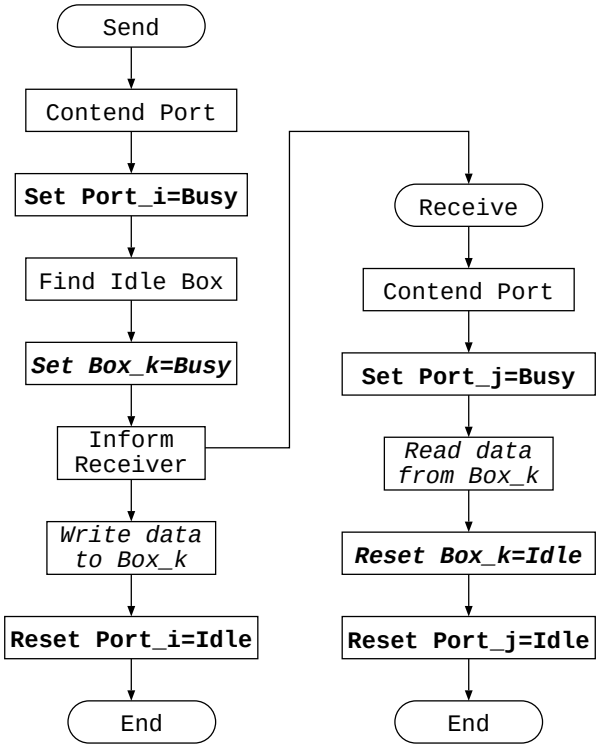


Figure 2. Procedure of message sending and receiving

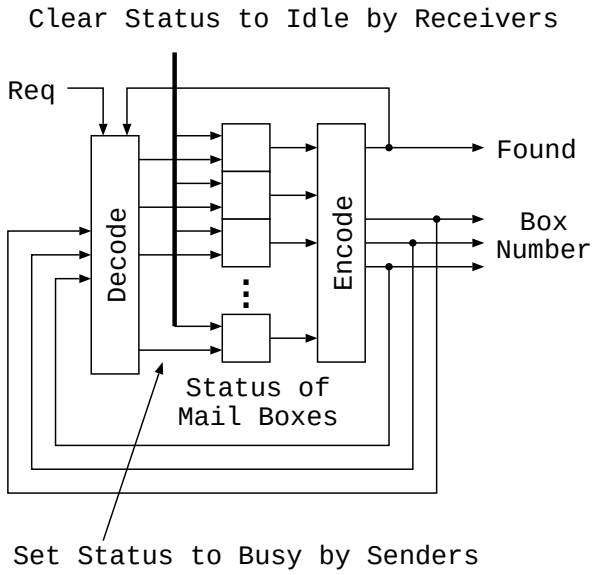


Figure 3. Circuit of finding an idle Mail-Box

the $P2$ can only use the mail box 0 for sending its message. $P9$ is informed to catch the message from this box.

This approach will decrease the flexibility of the use of the mail boxes. Because in the multiport memory, each port can access any location, we can remove the restriction of the fixed box allocation solution. In our model, we propose to allocate more mail boxes than the number of ports, and of course, each box can be accessed through any port.

It is easy for a node to get a free box. A simple circuit shown in Fig. 3 is used for this purpose. There are eight 1-bit registers with each bit indicating the corresponding mail box's status. A priority encoder outputs a possible idle box number. The "Found" signal indicates the availability of an idle box. Once an idle box is found, the status of this box will be set "Busy" by a decoder, and the sending node will start to send its data into this mail box. The box status will be cleared "Idle" by the receiving node once the node starts to read data in the box.

The ASU in the sending side has the responsibility to inform the receiving side ASU of the box number which is used by the sending node. There is a completely connected path between four ASUs. The destination ASU could be easily found just by checking the destination node number.

The destination ASU should inform the destination node of that a message for it has arrived. Then the destination node contends its memory port just as same as the sending node does in the first step. After the destination node gains this port, it will set the port status "Busy", read the data from the mail box, and clear the status of the box "Idle", once the data have been read. The "Busy" status of this port must be cleared by the node as well.

There is another solution for building the multiport memory in which the ordinary memory chips are used (see Fig. 4). In this solution, the mail box for any two nodes' communication is fixed. The basic memory chip has two separated ports; one is the Read port and the other is the Write port. If the nodes can be implemented with full duplex communication mechanism, this solution can provide four communication channels at the same time.

The complexity of the connection links for MCIM shown in Fig. 1 is considerably simple. After the arbitration and selection, the ASU allows only one computing node which has won the privilege of using its corresponding memory port to use this port, so, at one time, only one computing node can send its 64-bit data to this memory port, but the computing node does not need to connect its address lines with the memory port.

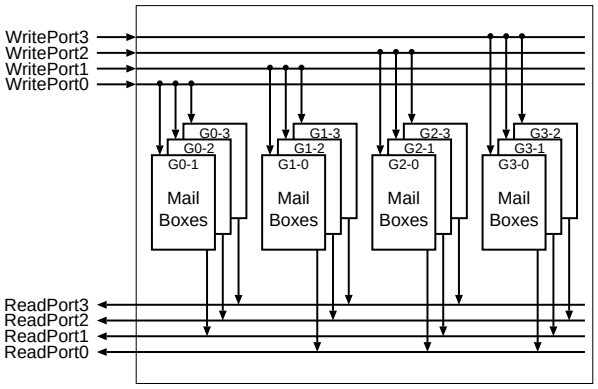
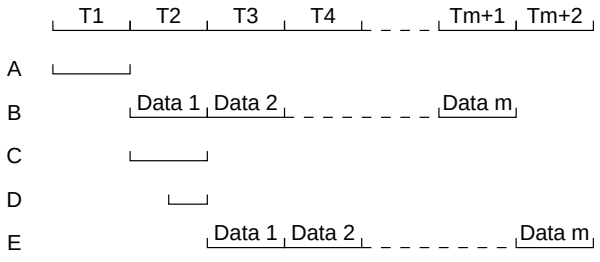


Figure 4. Using general memory chip to build Mail-Box

Instead, it employs the concatenation of Box Number and the value of counter as the memory address. The Box Number indicates the base address of the message location.

Since we employ the multiport memory, the sending process and the receiving process can occur concurrently. Meanwhile, it is a message passing mechanism, the data words of the message can be sent and received concurrently in most cases.



A: The sending nodes participates into arbitration
B: The sending node sends message
C: The destination node takes selection
D: The destination node acknowledges
E: The destination node receives message

Figure 5. The timing of sending and receiving data

The timing chart of sending and receiving data is shown in Fig. 5. When a sending node intends to send a message, it will send a request for participating into the arbitration in period T_1. When the sending node won the memory port, it starts sending the data of message in sequence in the periods of T_2,T_3..... T_{m+1}, i.e. to write the data into the contiguous memory locations addressed by a counter of the selected mail box. In period T_2, the destination node accomplishes the selection procedure and then notifies the destination node of the data receiving. The receiving node will issue an acknowledgment signal by the end of T_2, and will start to receive the data of message in sequence, i.e. to read the data from the selected mail box in a sequencing order in the periods of T_3,T_4.....T_{m+2}, m is the length of the passing message.

In Fig. 5, it is shown that the data receiving starts only with a delay period of two cycles after the start of data sending. And the data sending occurs concurrently with the data receiving in a pipeline mode.

In the MCIM parallel systems, if it happens that the message passing between the different nodes of the same groups, it also needs the procedure of arbitration and selection. According to their priorities, one node of them will win the privilege of sending data. Because all the nodes in the same groups connect their data lines together with the appropriate control gates, when the correspondingly selected gates are strobed, the sending node will directly send the data message into the receiving node by using DMA mechanism, it doesn't need to pass data through the mail box of the MPFSRAM.

3. Physical Layout and Scalability

The MCIM parallel system shown in Fig. 1 has 32 processors if each node contains only one processor. If each node consists of four processors, for which many computer vendors provide such support of chipsets, then, this parallel system can offer higher peak-performance.

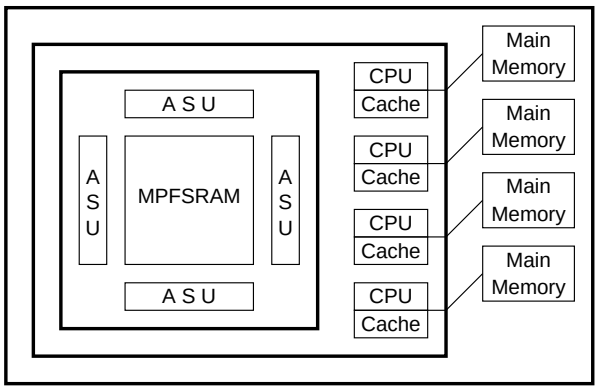


Figure 6. The illustration of the physical layout for the MCIM system

In the physical layout of the MCIM parallel system,

the MPFSRAM is located in the center of a board and surrounded by four ASUs, which should be linked with MPFSRAM as nearly as possible. The 32 processors of four groups with each groups corresponding to one ASU could be built on one large board just beneath the daughter board of MPFSRAM and ASUs, as shown in Fig. 6, where only four CPUs of one group are shown.

These two boards construct a compact physical installation for MCIM system. The connection lines would be rather short for the high-frequency units, and the high-bandwidth of MPFSRAM could be fully utilized. On the large board beneath the daughter board, the "cache" actually is the secondary cache, because the primary cache is on chip. The secondary cache works with a relatively slow speed, so it is feasible to link these secondary caches more easily to a third board, on which main memories of the computing nodes are located. The number of the memory boards depends on the scale of the required parallel system.

The MCIM system is scalable, because we could extend each node to a multiprocessor node. In addition, if it is required to construct a large scale MPP system, we could take the MCIM system depicted above as a Big Node, and to connect many Big Nodes together by the other interconnection network, for instance, by Hypercube IN as used in SGI-CRAY Origin 2000, where a S2MP consisting of 64 processors is used as a Big Node.

4. Simulation and Experiential Results

For the purpose of studying the data transfer bandwidth and time delay of sending/receiving operations, we have developed a simulation tool, by which we can investigate the bandwidth and delay with the relationship to the access frequency of memory port and to the message length. The simulation tool can generate the messages accessing the memory ports with variable time interval between messages, as well as to generate the messages with variable lengths.

In simulation, we assume that the interval of message generation and the length of message comply with Poisson's distribution, and the destination address is given with the distribution model of equality. In our experiment we assume that the number of computing nodes is 32; there are four read ports and four write ports separated each other.

In Fig. 7 and Fig. 8, the simulation results are illustrated. The horizontal axis is the expected value of time interval between sequential data transmissions for each node. This value indicates the access frequency of memory port by the computing nodes. Expected Length is the expected value of the message length, and the influences of which on the system performance is related to the ex-

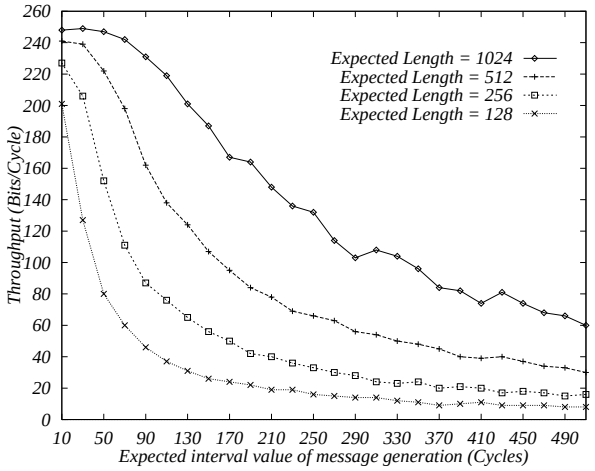


Figure 7. Simulation results for bandwidth

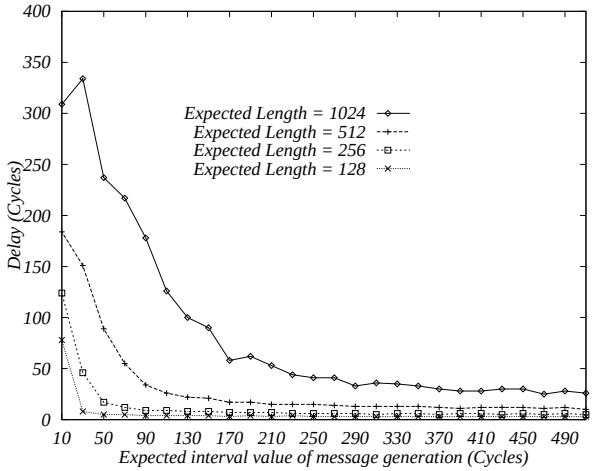


Figure 8. Simulation results for delay

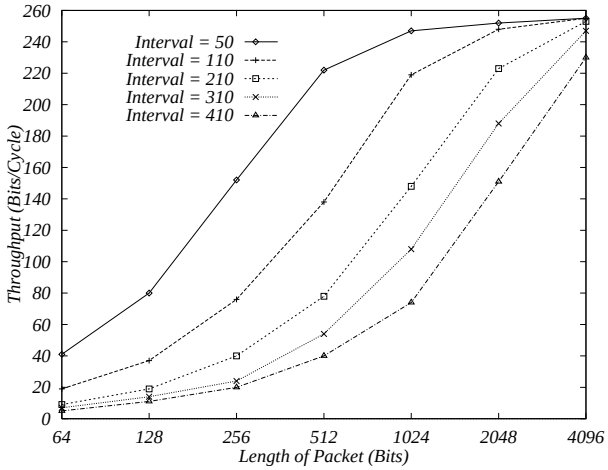


Figure 9. Simulation results for bandwidth

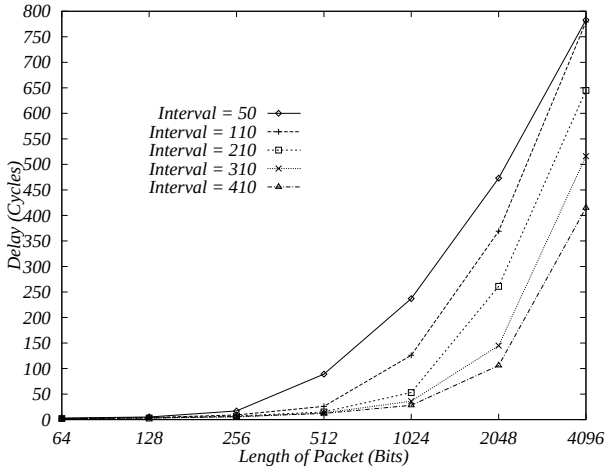


Figure 10. Simulation results for delay

pected interval of memory port access. In these figures, the unit of bandwidth is in bits/cycle, the unit of interval is cycle.

Fig. 9 and Fig. 10 show the bandwidth and delay with variable packet lengths. We found that the packet length of 256-bit or 512-bit is a better tradeoff between the delay time and the bandwidth.

5. Conclusion

This paper investigated the Memory Centric Interconnection Mechanism (MCIM) for message passing parallel systems. MCIM uses the Multi-Port Fast Static Memory as the mail-box, surrounded with the Arbitrator and Selector Units (ASUs), which are shared by multiple computing nodes.

In comparison with MPP systems, MCIM is able to implement the pipelined mode of data word sending and receiving for message passing, so it results in less latency.

The features of multiport memory and the flexibility of mail-box usage in MCIM could reduce the blockage possibility of message transmission which constitutes an essential problem in the logic circuit based interconnection networks of the conventional MPP systems.

MCIM is also more cost-effective due to its more easily available MPFSRAM than the custom designed routers beyond the less complexity in connection links for ASUs. Meanwhile, it provides higher bandwidth of data transfer gained from the latest technology of MPFSRAM.

In order to achieve very high peak performance, the MCIM parallel system could be considered as a Big Node, and MCIM could be combined with the technology of the logic-circuit based interconnection network to construct a Tera-Flops supercomputer.

References

[1] T. Y. Feng, “An Survey of Interconnection Networks”, *IEEE Computer*, Vol. 14, No. 2, 1981, pp12-27.

[2] W. J. Dally, “Performance of K-ary X-cube Interconnection Networks”, *IEEE Transaction on Computers*, Vol. 39, No. 6, 1990, pp775-785.

[3] M. Jarczyk, T. Schwederski et al, “Strategies for Massively Parallel Simulation of Interconnection Networks”, *Proceedings of Int’l Conf. on Parallel Processing*, pp I.21-25, 1994.

[4] W. J. Dully and C. L. Seitz, “Deadlock-Free Message Routing in Multiprocessor Interconnection Networks”, *IEEE Transaction on Computers*, Vol. C-36, No. 5, May 1987, pp547-533.

[5] R. W. Hockney, “The Communication Challenge for MPP: Intel Paragon and Meiko”, Parallel Computing, Vol. 20, No.3, Mar. 1994, pp 389-398.

[6] Z. W. Xu and K. Hwang, “Modeling Communication Overhead: MPI and MPL Performance on the IBM SP-2” *IEEE Parallel & Distributing Technology*, Vol. 21, No. 1, Spring, 1996, pp.9-23.

[7] M. A. Blumrich, C. Duknicki, E. W. Felten, K. Li, and M. R. Mesanina, “Virtual-Memory-Mapped Network Interfaces”, *IEEE MICRO*, Vol. 15, No. 1, Feb. 1995, pp21-28.

[8] T. V. Eicken et al, “Active Messages? A mechanism for Integrated Communication and Computation”, *Proceedings of the 19th Int’l Symp. on Computer Architecture*, 1992, pp256-266.

[9] M. Dubois and F. A. Briggs, “Tutorial Notes on Shared Memory Architectures for Multiprocessors”, *Proc. 17th Symposium of Computer Architecture*, Seattle, WA, 1990.

[10] M. Dubois and S. Thakkar, “Scalable Share-Memory Multiprocessor”, *Kluwer Academic Publishers*, Boston, MA. 1992.

[11] T, Takayanagi et al, “350 MHz Time-Multiplexed 8-port SRAM and Word-Size Variable Multiplier for Multimedia DSP”, *International Solid State Circuit Conference ’96*, Feb. 1996, pp150-151.