MBP-light: A Processor for Management of Distributed Shared Memory

Inoue Hiroaki[†], Ken-ichiro Anjo[†], Jun Tanabe[†], Katsunobu Nishimura[†],
Mitsuru Satoh[‡], Kei Hiraki[*], Hideharu Amano[†]

[†]Keio University, [‡]Fujitsu Laboratories LTD., [*]University of Tokyo
[†]3-14-1, Hiyoshi, Kohoku-ku, Yokohama, Kanagawa, JAPAN
E-mail: mbp@aa.cs.keio.ac.jp

Abstract

MBP(Memory Based Processor)-light is a dedicated processor for management of cache coherent distributed shared memory (DSM) in a massively parallel processor called JUMP-1. Unlike traditional complicated controllers to manage DSM, it provides a simple but powerful 16-bit RISC processor as a core. On the other hand, memory controller, bus interface and network packet handler are implemented completely with a hardwired logic to require high-speed operation.

Using 0.4μm CMOS embedded array technology and the TBGA (Tape Ball Grid Array) package, it works at 50MHz clock. Compared with chips used in Sequent's NUMA-Q, it supports almost the same performance with much smaller amount of hardware.

1 Introduction

JUMP-1 is a prototype of a massively parallel processor developed by a collaboration of 7 Japanese universities[1]. The major goal of this project is to establish techniques to build an efficient cache coherent Distributed Shared Memory (DSM) on a massively parallel processor.

A lot of novel techniques are introduced in the DSM of JUMP-1 for this purpose. In order to satisfy both high degree of perfomance and flexibility, a dedicated processor called MBP(Memory Based Processor)-light is proposed for management of the DSM of JUMP-1. Unlike traditional complicated controllers to manage DSM, it provides a simple but powerful 16-bit RISC processor as a core. It can handle packets efficiently by using a special instruction set called *buffer-register architecture*. On the other hand, memory controller, bus interface and network packet handler are implemented with a hardwired logic for high-speed packet management.

In Section 2, the structure of JUMP-1 is introduced. The architecture MBP-light is illustrated in Section 3 and the detail of the chip implementation is described in Section 4.

2 The Structure of JUMP-1

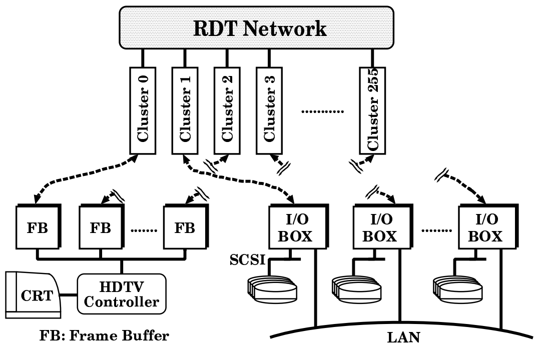

Figure 1: The Structure of JUMP-1

As shown in Figure 1, JUMP-1 consists of 256 clusters connected each other with an interconnection network called RDT(Recursive Diagonal Torus)[4, 5]. The RDT includes both torus and a kind of fat tree structure with recursively overlayed two-dimensional square diagonal tori structure. Each cluster provides a high speed point to point I/O network[9] connected with disks and high-definition video devices.

The cluster is a bus-connected multiprocessor (Figure 2) including four RISC processors (SuperSPARC+), MBP-light which is directly connected to a cluster memory, and the RDT router chip for the interconnection network[6]. MBP-light, the heart of JUMP-1, is a custom designed processor which manages DSM, synchronization, and packet handling. The first prototype of JUMP-1 consists of 256 clusters (1024 processors).

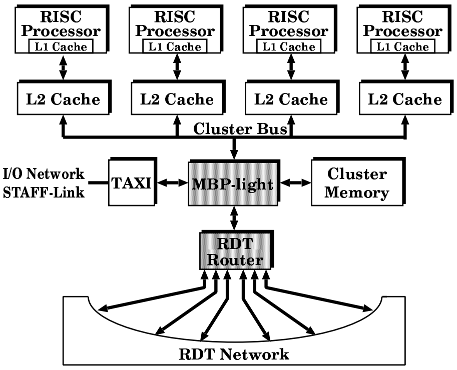

Figure 2: The Structure of JUMP-1 Cluster

The detail scheme of DSM management for JUMP-1 is described in [2].

3 The Structure of MBP-light

In traditional DSM systems, a directory entry is managed with every cache line. Although such a mechanism works efficiently in those systems with a limited number of processors, it is not suitable for a system with thousands of processors. For example, a large amount of memory for cache and directory is required. The invalidation protocol based on one-to-one data transfer often causes a network congestion when many processors share the same data.

In order to address these problems, the following methods are used in JUMP-1.

Each processor (SuperSPARC+) shares a global virtual address space with two-stage TLB (Translation Look-aside Buffer) implementation. The directory is attached not to every cache line but to every page, while the data transfer is performed by a cache line. A part of cluster memory is available as L3 (Level-3) cache.

Reduced Hierarchical Bitmap Directory schemes[3] are also proposed to manage efficiently cache based on multicasting. Using this directory schemes, various types of cache coherence protocols can be utilized, including not only invalidate type protocols but also update type protocols for applications which require frequent data exchange.

To manage such a sophisticated DSM efficiently, the MBP-light consists of the MBP Core processor and hardwired logic blocks which are RDT Interface and MMC (Main Memory Controller), as shown in Figure 3.

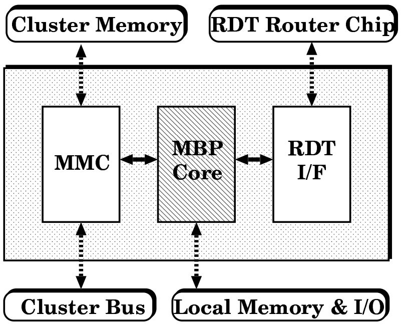

Figure 3: The Structure of MBP-light

3.1 MBP Core

Unlike traditional processors to manage DSM (*e.g.* Magic in Stanford FLASH[7] and SCLIC in Sequent's NUMA-Q[8]), a simple 16-bit RISC architecture with four stage pipeline is adopted in MBP-light.

The MBP-light mainly analyzes a receiving packet, accesses a table, transforms the address, and generates a sending packet. A header of a packet in JUMP-1 is sometimes complicated and occupies several flits of the packet. Also, tags included in a data part of a packet relates to a protocol control. Therefore, it is convenient to treat all packet buffers as a register. However, since a width of packet buffers is 68-bit, it requires an enormous hardware to treat such buffers as common general purpose registers in the processor.

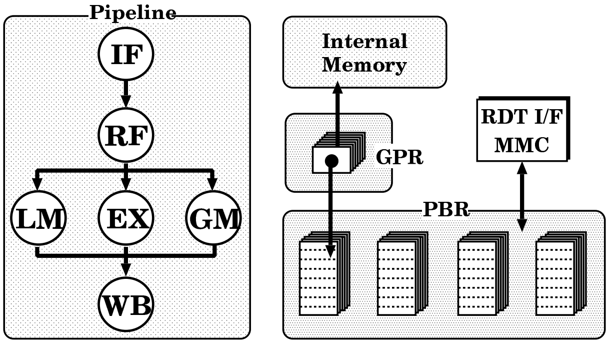

Figure 4: The Structure of MBP Core

To solve this problem, MBP Core provides 16 GPRs (General Purpose Registers) of 16-bit width and 112 PBRs (Packet Buffer Registers) of 68-bit width. The PBR is indicated by the content of the GPR, and accessed in the processor pipeline like a common register. While operations and data transfer between PBR-GPR and PBR-PBR are allowed, the content of the PBRs is transferred directly as a packet from/to MMC or RDT Interface. As operations are mainly done between such a packet buffer and a register, we call this structure the *buffer-register architecture*.

The MBP Core consists of a pipeline with four stages treating 21-bit instructions and 16-bit data, as shown in Figure 4. 21-bit × 64K local memory which stores instructions and local data is connected. The MBP Core takes the I/O mapped approach, and another 64K address is provided for I/O devices and dedicated hardware for the barrier operation. The cluster memory for data and tag is also accessed through the PBR.

Three sets of PBRs are also used as a cyclic buffer in RDT Interface. While RDT Inteface receives or sends a packet by using a set of buffer, the MBP Core can use another set of buffer.

A packet transfer between cluster bus is done in the different manner. A packet which is stored in any P-BRs can be directly block transferred from/to cluster bus by executing a bus transfer instruction through MMC (Main Memory Controller). Since this transfer sometimes takes more than 10 clocks, out-of-order completion mechanism is introduced. Succeeding instructions which do not use the corresponding PBRs can be executed during the block transfer. Cluster memory is also accessed in the same manner.

3.2 RDT Interface

The MBP-light is directly connected with the RDT router chip, and manages network packet transfer. The RDT Interface consists of the Packet Handler which sends/receives packets, Ack Generator for generation of an acknowledgment packet, and Ack Collector which collects acknowledgment packets.

Ack Generator provides two cache systems called Net Cache and Ackmap Cache accessed together when a packet with coherent message is received. Net Cache, a direct mapped cache with 512 entries, is accessed by the accessing address of the DSM in the packet header. It stores the information whether the accessing line is cached in the cluster (in L2 or L3 cache) or not. At the same time, the Ackmap Cache is accessed by the source cluster number which sent the packet, and the bitmap which shows the returning path is obtained. Using the above information, an acknowledgment packet (when the accessed data is cached) or not-acknowledgment packet (the accessed data is not cached) is generated by the hardwired logic.

3.3 Main Memory Controller

MMC (Main Memory Controller) manages the cluster memory consisting of SDRAM for data and SRAM for tags. It also controls the cluster bus which connects four SuperSPARC+ processors through the L2 cache. When a processor misses the L2 cache and the cluster memory must be accessed, MMC checks the tag. Depending on the status of a cache line, MMC interrupts to MBP Core, and the software on the core processor is invoked.

4 Chip Design

MBP-light is implemented on the Toshiba's 0.4μm CMOS 3-metal embedded array TC203E340. In order to cope with a large number of pins, TBGA (Tape Ball Grid Array) package is used. The design of MBP-light is described in VHDL, synthesized with Mentor's Autologic-II, and verified with Toshiba's VLCAD.

Table 1: The Specification of MBP-light

Maximum clock (MHz)	50
Random logics (the number of gates)	106,905
Internal memory (bits)	44,848
Area utilization (%)	38.3
The number of pins	352
Consuming Power (W)	3.1

Using the behavior level simulator, a program of the MBP Core processor for a simple protocol has been developed in parallel with the hardware design. The specification of the MBP-light is shown in the Table 1. A half of total gates are used for the embedded memory as a lot of buffers and tables are required in the MBP-light.

Table 2: The Protocol Processing Time

Request	Line State (Home CL[1])	JUMP-1	NUMA-Q
L3 cache hit	–	760 ns	800 ns
Read miss	Valid	6.9 μs	4-10 μs
Read miss	Invalid	15.3 μs	–

Compared with SCLIC / OBIC chips used in Sequent's NUMA-Q, only a third of random logic and a fourth of memory are required for MBP-light. The performance of protocol processing for DSM is almost the same as those of NUMA-Q as shown in Table 2. These tables demonstrate that MBP-light manages DSM quickly with less amount of hardware.

[1] CL is an abbreviation for cluster

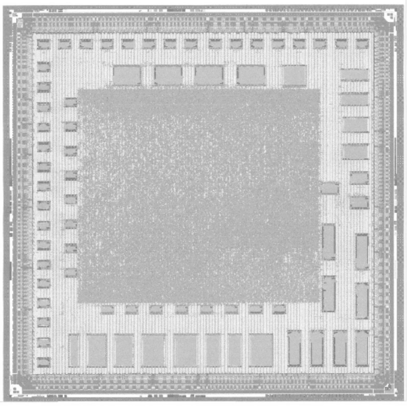

Figure 5: The Layout of MBP-light

The layout of MBP-light is shown in Figure 5. A lot of embedded RAMs are placed near four edges of the die surrounding random logics in the middle square part. Large RAMs are corresponding to the cache memory in RDT Interface, while small ones are used for PBRs.

The picture of the cluster board is shown in Figure 6. The MBP-light is a small metal chip placed on the center of the board. Since the TBGA package is used, no heat sink is attached.

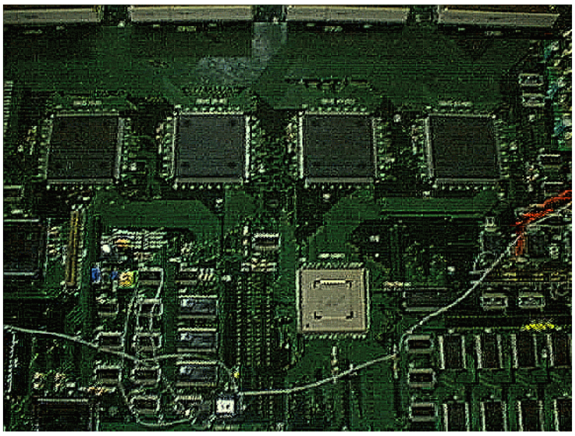

Figure 6: MBP-light in The Cluster Board

5 Conclusions

A dedicated processor called MBP-light for management of distributed shared memory is introduced, and its performance are evaluated.

Compared with gate numbers and protocol processing time, MBP-light is of great advantage to SCLIC/OBIC which manages the DSM on NUMA-Q.

Now, a cluster of JUMP-1 with engineering sample of MBP-light chip is under debugging. The prototype of JUMP-1 with 64 processors is scheduled to be available within this year.

6 Acknowledgments

The authors would like to express their sincere gratitude to the members of the Joint-University project for their valuable advises and discussions. The authors also express their thanks to Mentor Graphics Japan for supporting design tools.

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. This research is also supported by Parallel and Distributed system Consosium.

References

[1] K. Hiraki et. al.,"Overview of the JUMP-1, an MPP Prototype for General-Purpose Parallel Computations," IEEE International Symposium on Parallel Architectures, Algorithms and Networks, pp.427–434, 1994

[2] T. Matsumoto et. al., "Distributed Shared Memory Architecture for JUMP-1: A General-Purpose MPP Prototype," International Symposium on Parallel Architectures, Algorithms and Networks, pp.131–137, 1996.

[3] T. Kudoh et. al.,"Hierarchical bit-map directory schemes on the RDT interconnection network for a massively parallel processor JUMP-1," International Conference on Parallel Processing, August, pp.I-186–I-193, 1995

[4] Y. L. Yang et. al.,"Recursive Diagonal Torus: An interconnection network for massively parallel computers," IEEE symposium on Parallel and Distributed Processing, December, pp.591–594, 1993

[5] Y. L. Yang and H. Amano,"Message Transfer Algorithms on the Recursive Diagonal Torus," IEICE transactions on Information and Systems, February, pp.107–116, 1996

[6] H. Nishi et. al.,"Router Chip: A versatile router for supporting a distributed shared memory," IEICE-ISPAN, 1997.

[7] J. Kuskin et. al., "The Stanford FLASH Multiprocessor," The 21st ISCA, pp.302 – 313, 1994

[8] T. Lovett and R. Clapp,"STiNG: A CC-NUMA Computer System for the Commercial Marketplace," The 23rd ISCA, pp.308–317, 1996.

[9] H. Nakajo et. al., "An I/O Network Architecture of the Distributed Shared-Memory Massively Parallel Computer JUMP-1," International Symposium on Supercomputer, 1997.