ČESKOSLOVENSKÁ VĚDECKOTECHNICKÁ SPOLEČNOST

FAULT TOLERANT SYSTEMS & DIAGNOSTICS

DIAGNOSTIKA A ZABEZPEČENÍ ČÍSLICOVÝCH SYSTÉMŮ

1981

SYSTEM ARCHITECTURE AND PROTECTION MECHANISMS OF THE MuTEAM MULTIMICROPROCESSOR

R. Cardini°, M. La Manna°, L. Lopriore°°, L. Strigini°

°Selenia, Industrie Elettroniche Associate, S.p.A., Roma, Italy
°°Istituto di Elaborazione dell'Informazione, Consiglio Nazionale delle Ricerche, Pisa, Italy

Introduction

MuTEAM is a multimicroprocessor system, being developed in Pisa under the support of the National Computer Science Program of the Italian National Research Council /1/. The system is composed of several Computer Elements (essentially constituted of a processor and some memory), which are grouped in clusters. The Computer Elements of the same cluster communicate through a shared memory space, which is utilized for obtaining a fast message transmission among processes. The shared space is segmented, and a protection mechanism is provided, which associates an access control list to each segment: the list of a given segment specifies the access rights of each processor of the cluster on that segment.

In this paper, the MuTEAM system architecture and protection mechanisms are described in some detail. The provided fault treatment strategy and the operating system kernel are related in /2/ and /3/, respectively.

System architecture

The system is composed of several clusters, a cluster being essentially constituted of up to 16 Computer Elements (or nodes) communicating through two high speed parallel buses, namely the Cluster Bus and the Signalling Bus (Fig. 1). Clusters are connected by serial lines, which allow low speed communications among nodes of different clusters. Each Computer Element is mainly constituted of: i) a microprocessor with segmentation facilities, actually a Zilog Z8001; ii) a Private Memory Block of up to 1M bytes; iii) a Shared Memory Block of up to 1M bytes; and iv) a Communication Controller, for the transmission of interrupts to the other nodes of the same cluster (via

This work has been supported by the National Computer Science Program of the Italian National Research Council

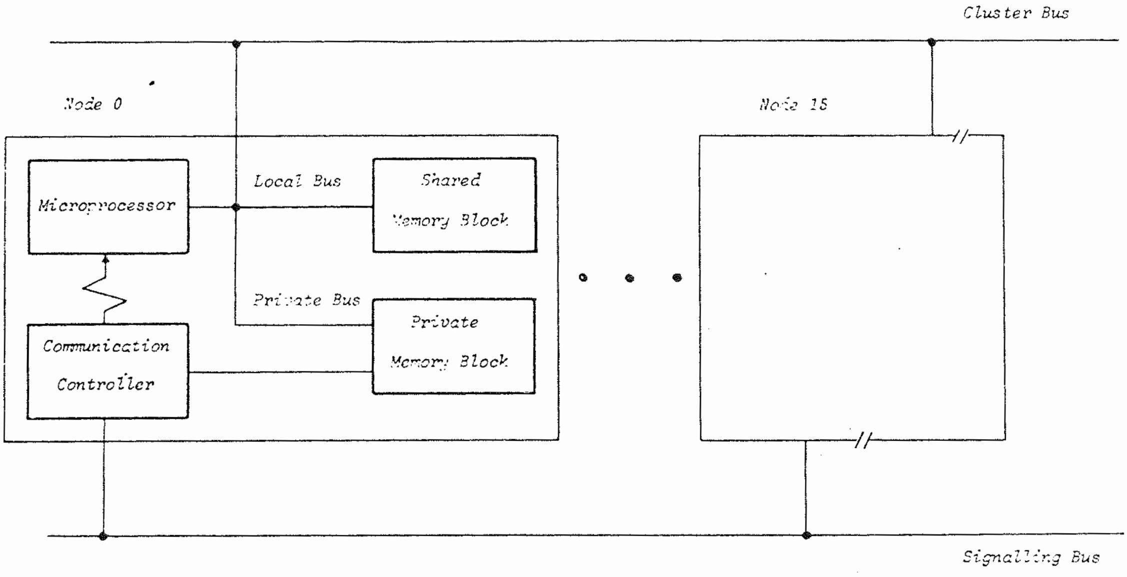

Fig. 1 Configuration of a cluster

the Signalling Bus).

The processor of a given Computer Element can directly access both the ˇShared and the Private Memory Blocks of that Computer Element, by means of a Local Bus and a Private Bus, respectively. Moreover, it can access the Shared Memory Blocks of all the other nodes of the same cluster, by utilizing the Cluster Bus. Accesses to a non-local Shared Block are slower than accesses to the Local Shared Block, owing to contentions for the Cluster Bus with the other processors of the cluster.

Addressing mechanisms

A process space, composed of 128 segments of up to 64K bytes each, is associated with each running process. The address generated by the processor of a given node is relative to the space of the process running on that processor. The address is composed of a 7-bit segment number and a 16-bit offset, which specifies a byte in that segment. The segments of all the process spaces are mapped into a unique cluster virtual space, composed of 32K segments of 64K bytes each (the segments of the cluster virtual space are implemented in the Private and in the Shared Memory Blocks of the cluster, 1K segments for each block). The mapping operation is performed in each node by a proper Translation Unit, which is mainly constituted of 128 registers of 15 bits. Each register contains all the information required to map a segment of the space of the running process into a segment of the cluster virtual space. More precisely, a register is composed of three fields, namely the P/S, ND and NB fields. The 1-bit P/S field specifies if the segment is implemented in the Private Memory Block of that node, or in a Shared Memory Block. The 4-bit ND field is significant only if the segment is implemented in a Shared Memory Block, and specifies the node containing the block. At last, the 10-bit NB field specifies the name of a segment in the memory block addressed by the P/S and ND fields.

According to the contents of the P/S and ND fields, the cluster virtual address is sent to the Private Memory Block of that node, or to a Shared Memory Block. More precisely, if the segment is implemented in the Private Memory Block, the address is sent to the block via the Private Bus. If the segment is implemented in the Shared Memory Block of the same node, the address is sent to the block via the Local Bus. At last, if the segment is implemented in the Shared Memory Block of a different node, the address is sent to the Cluster Bus.

A proper Relocation Unit is provided in each node, which converts the cluster virtual address generated by the Translation Unit of the same node (and coming from the Local Bus) or by the Translation Unit of a different node (and coming from the Cluster Bus) into a physical address in the relative Shared Memory Block. The Relocation Unit mainly consists of 1K Relocation Registers, a register for each segment of the block. A Relocation Register is 16 bit long, and its contents multiplied by 16 represent the segment base in the block.

A similar Relocation Unit converts the cluster virtual address generated by the Translation Unit of a given node (and coming from the Private Bus) into a physical address in the Private Memory Block of the same node.

Protection mechanisms

Accesses to the Shared Memory Block of each node are controlled by a dedicated Protection Unit (Fig. 2). The unit contains a set of 1K Protection Registers, a register for each segment of the block. A Protection Register is 48 bit long, and is partitioned into three 16-bit fields, namely the L, R, and W fields. The L field specifies the length of the segment, in bytes. The R and W fields implement an access control list /4/ for the segment. More precisely, if the i-th bit of the R field is set, then the i-th processor in the cluster can access the segment for read operations, i=0,1,...,15. Similiarly, if the j-th bit of the W field is set, then the j-th processor in the cluster can access the segment for write operations, j=0,1,...,15.

A proper Access Right Violation Checker generates an access right

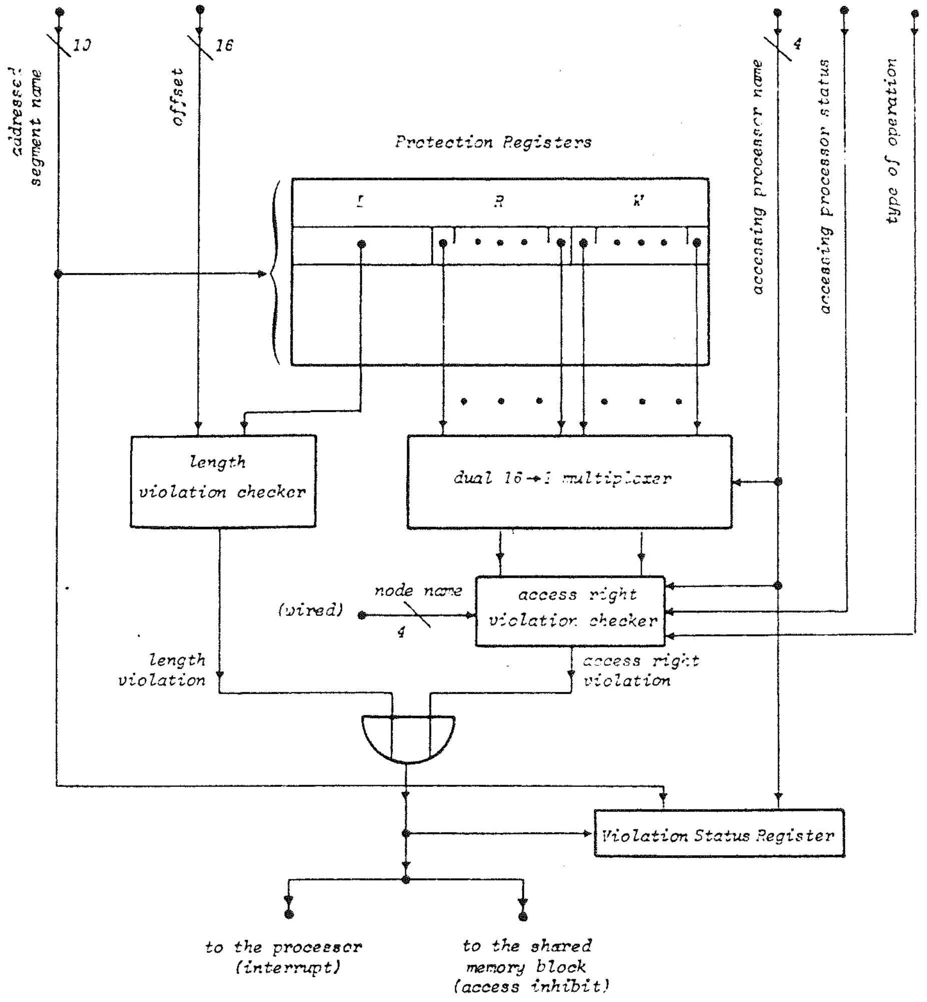

Fig. 2 Block diagram of the Protection Unit

violation if the type of the access attempt does not match the rights of the accessing processor on the addressed segment. The violation is also generated if an access attempt for write operation is generated by a processor belonging to a different node, and that processor is not in the supervisor state. For its aims, the checker utilizes the information coming from two input lines of the Protection Unit, namely the operation type line and the processor status line, which specify if the access attempt on the addressed segment is for read or write operation, and if the accessing processor is in the normal or supervisor state, respectively. Moreover, a Lenght Violation Checker generates a length violation if the address specifies an offset in the addressed segment, which exceeds the segment length.

In both cases of access right and length violation, the following actions occur: i) the access to the memory block is inhibited; ii) the status of the violation (that is, the name of the accessing processor and the name of the addressed segment) is stored into a proper Violation Status Register; iii) an interrupt request is sent by the Protection Unit to the processor of the same node. All the registers of the Protection Unit are mapped into the Special I/O addressing space of this processor. So, the interrupt handler will be able to read the name of the processor which caused the violation in the Violation Status Register. If the processor belongs to a different node, the handler will send it a proper notification message.

A further Protection Unit in each node controls the accesses to the Private Memory Block of that node. The unit mainly consists of 1K registers of 16 bit: the contents of each register specify the length of a segment of the Private Memory Block. A dedicated logic generates a length violation if the address specifies an offset within the addressed segment, which exceeds the length of the segment itself.

Aknowledgements

Several persons have been involved in the design of the MuTEAM architecture, and, among them, P. Corsini, of Istituto di Elettronica e Telecomunicazioni, Università di Pisa; G. Frosini, of Istituto di Informatica, Università di Ancona; G. Cioffi and P. Velardi, of Istituto di Automatica, Università di Roma; A. Orlandi, of Oto Melara, S.p.A., La Spezia; A. Ciuffoletti, of Selenia, S.p.A., Roma; and the authors of the companion papers /2,3/.

References

/1/ "The MuTEAM Project: Preliminary Presentation", Technical Report, MUMICRO Series, National Computer Science Program, 1980 (In Italian).

/2/ P. Ciompi, F. Grandoni, L. Simoncini, "Self-diagnosis in Multimicroprocessor Systems: the MuTEAM Approach", this issue.

/3/ F. Baiardi, A. Fantechi, A. Tomasi, M. Vanneschi, "Protection and Error Confinement in a Message-passing Environment", this issue.

/4/ J.H. Saltzer, M.D. Schroeder, "The Protection of Information in Computer Systems", Proc. IEEE, Sept. 1975, pp. 1278-1308.