EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH
CERN - SPS DIVISION

CERN SPS/85-28 (ACC)

MULTI-MICROPROCESSOR ARCHITECTURE FOR THE LEP STORAGE RING CONTROLS

J. Altaber, P.G. Innocenti and R. Rausch

Abstract The new CERN Large Electron Positron (LEP) storage ring control system follows the concepts developed for the Super Proton Synchrotron (SPS) accelerator but making use of present day technology. A multi-tasking computer is replaced by an assembly of microprocessor based modules performing a unique, single stream type of task. Each module is a general purpose processing unit (GPU) containing a 68000 microprocessor, private memory, protected memory accessible by others GPU's, a programmable and adressable interrupt logic and a distributed arbitration mechanism. Communication amongst GPU's is done by a message protocol, the medium being the VME multi-master parallel bus. Dedicated Input/Output modules can be associated privately with the GPU's thus forming specialized functional modules, or providing additional private memory. All these functional modules communicate over the VMEbus in a protected access mode with resource reservation to prevent processor interferences. A global system concept has been developed which will be detailed in this paper. The improvement in performance, flexibility, processor independence, minimization of integration effort required, as well as error diagnostics in such a multiprocessor architecture are also discussed.

Paper presented at the 6th Annual Workshop on Distributed Computer Control Systems, International Federation of Automatic Control, Monterey, 20-22 May, 1985.

Geneva, May, 1985.

MULTI-MICROPROCESSOR ARCHITECTURE FOR THE LEP STORAGE RING CONTROLS

J. Altaber, P.G. Innocenti and R. Rausch

European Organization for Nuclear Research (CERN), 1211 Geneva 23, Switzerland

Abstract. The new CERN Large Electron Positron (LEP) storage ring control system follows the concepts developed for the Super Proton Synchrotron (SPS) accelerator but making use of present day technology. A multi-tasking computer is replaced by an assembly of microprocessor based modules performing a unique, single stream type of task. Each module is a general purpose processing unit (GPU) containing a 68000 microprocessor, private memory, protected memory accessible by others GPU's, a programmable and addressable interrupt logic and a distributed arbitration mechanism. Communication amongst GPU's is done by a message protocol, the medium being the VME multi-master parallel bus. Dedicated Input/Output modules can be associated privately with the GPU's thus forming specialized functional modules, or providing additional private memory. All these functional modules communicate over the VMEbus in a protected access mode with resource reservation to prevent processor interferences. A global system concept has been developed which will be detailed in this paper. The improvement in performance, flexibility, processor independence, minimization of integration effort required, as well as error diagnostics in such a multiprocessor architecture are also discussed.

Keywords. Multi-microprocessor architecture, accelerator control, distributed arbitration, VME multiprocessor, function-to-function architecture, message passing protocol.

INTRODUCTION

CERN is constructing a Large Electron Positron (LEP) collider on the Swiss/French border. The new machine has a circumference of 28 kms and is located in an underground tunnel at 60 to 120 m depth below the surface of the ground. Equipment will be installed into the tunnel and be controlled from underground alcoves and from surface buildings located at eight equidistant areas. LEP is expected to start operation early 1989.

The LEP control system will use a fully distributed computer system made up almost entirely of closely coupled multi-microprocessor assemblies (Altaber, 1983b; Crowley-Milling, 1983).

The LEP machine will be operated from the same control room as the SPS accelerator, in operation since 1976. It follows the same philosophy of functional and geographical distribution but takes advantage of to-day's technology. Instead of conventional mini-computers the LEP control system will use assemblies of microcomputers built in VMEbus standard (Altaber, 1983a) by a token passing ring network conforming to the ISO P.8802.5 standard protocol, Fig. 1.

CERN is currently consulting computer manufacturers to identify and chose a distributed multi-microprocessor system together with a distributed real-time operating system industrially supported (CERN, unpublished, 1984).

The purpose of this paper is to describe the prototype multi-microprocessor architecture being developed at CERN, Fig. 2. The objective of this project is to verify the validity of these concepts for real-time control of a large accelerator and to assess its relative merits and limitations compared to a conventional implement-

ation (Altaber, 1979; Altaber, 1980; Altaber, 1982). The technical requirements to be satisfied by a multi-processor architecture, the inter-processor communication, and the reservation and protection mechanisms implemented are discussed. Finally the software environment for a fully distributed real-time operating system is presented.

THE PRINCIPLES

In the past, several multiprocessor architectures have been proposed differing mainly by the degree of coupling and ranging from a network of distributed processors to tightly coupled microcomputers connected to a high speed multimaster parallel bus (Ali, 1979; Barthmaier, 1980). The CERN project described here is of the closely coupled type and uses the high bandwidth of the standard VMEbus for interprocessor communication without the aid of a central controller.

Modularity and expandability, the primary goals in the hardware design, are ultimately aimed at drastic reductions in design time and cost for customized systems of varying degrees of complexity and software simplification. Although the microprocessor is a relatively slow device, the parallelism obtained in closely coupled multiprocessors greatly enhances system speed and throughput. In addition, future expansion of the system is accomplished easily by adding more microprocessors and without any significant increase in the overall cost.

Another attractive capability of the multi-processor approach over the single multitasking minicomputer is the potential for increasing the reliability. Reliability can be increased by allowing more than one microprocessor to perform the same critical tasks in parallel.

Fig. 1. LEP computer network

Function-to-Function Architecture

A Function-to-Function Architecture (FFA) (Conrad, 1981) has been implemented for the CERN multi-microprocessor project. The FFA is a generation of memory-intensive microcomputer boards and software modules completely interchangeable within the system structure. FFA is the technique for distributing functionally powerful computing elements throughout a system. The FFA provides familiar structures and simplified procedures allowing users to benefit from highly advanced microcomputer technology.

In a conventional system, the three functions, shown as F1, F2 and F3 in Fig. 3, are executed in turn by multiplexing the hardware resources between the functions as required. Thus only one function at a time can be active. Even if one CPU was dedicated to each function, the overall system performance would not improve appreciably because all activity is bound by the bandwidth of the bus.

The function-interconnect system of Fig. 3 represents the form of FFA, and resolves the bus crowding problem. Each functional module contains sufficient processing, memory and I/O capability to execute the required functions collected within the module. The effective bandwidth of an FFA system can be as high as the sum of its individual microcomputer bandwidths. The bandwidth of the bus that interconnects these microcomputers is a function of the volume of inter-processor messages and shared data traffic. Modular computing functions can contain the hardware and software for implementing one or several pre-programmed functions.

The CERN multi-microprocessor architecture is aimed at the control of the LEP accelerator. A large number of these multi-microprocessor systems will be used and they will act as Process Control Assemblies (PCA). Each PCA contains several functionally dedicated processing units, some of them are associated privately with Input/Output cards or private memory extension boards, Fig. 4. All processing units are identical in the present implementation, using a Motorola 68010-12 MHz processor and show to the function interconnect bus the same logical structure, Fig. 5. More powerful processors may be used in the future if required for some specific function but the physical and logical connection to the bus will be identical to the present one.

Some of the functional modules are:

- A Data-Link unit (DL) interfacing each PCA to the Token Passing Ring network conforming to the ISO-P.8802.5 protocol.
- A Supervisory Unit (SU) managing the activities in a PCA.
- One or more NODAL Interpreter modules (NI) interpreting the high level commands.
- A FORTRAN Processing Unit when fast data processing is required.
- An Equipment Directory unit (ED) which acts primarily as the local data-base.
- Several Multidrop Bus controller (BC) modules conforming to the MIL 1553-B protocol linking

Fig. 2. The LEP controls architecture

many Equipment Control units to the PCA for control and monitoring.

Additional functions may be added to a PCA to complement the basic functionality, i.e. graphics, man-machine interfaces, mass storage drivers, etc. Modules can evolve at their own pace changing and improving with advances in technology; the architecture provides a clear path to 68020 32-bit multiprocessor configurations.

Message Passing Protocol

The function-to-function architecture implies a message passing protocol between micro-computer boards for data exchange. This concept is extensively supported in the MULTIBUS II architecture (Beaston, 1984). Two intermodule communication protocols are generally used in multiprocessor systems "pass-by-reference" and "pass-by-value". In "pass-by-reference" the communicating modules exchange pointers to gain addressability to the shared data structures, while in "pass-by-value" modules exchange a copy of the data structure. Both methods have their relative merits and inconveniences.

The CERN multiprocessor project uses a flexible event interrupt method to communicate the type of message to be transferred and the "pass-by-value" method for the data exchange between the source processor and the destination processor.

The ability of all processors to share portions of physical memory is fundamental to the design of a closely coupled multiprocessor system. CERN's design allows a processor access to a local physical memory that, while reserved primarily for its own use, is also shared among other processors. (This use of shared local memory is similar to the structure used in CM^{*}, a multiprocessor developed at the Carnegie-Mellon University (Swan, 1977)). Arbitration logic resolves competition for use of each processor's memory bus. The principle of a safe "circuit-switching" whereby a direct signal path is established between a source memory and a destination processor is implemented, Fig. 6. The shared local memory is partitioned in small blocks of 256 words each allocated to one corresponding source processor. The memory mapping and the address generation and decoding is arranged by hardware logic in such a way that any processor can physically only access its own predefined and reserved area in the shared memory of any other processor. Thus corruption of data is prevented

Fig. 3 The function-to-function architecture

by hardware protection and moreover is reinforced by a "Read mostly" approach of the messages out of shared memory. This organization of the memory mapping allows the maximum number of pluggable processors (20) in a VME crate. The multicrate architecture developed at CERN for the VMEbus extends theoretically this capability up to 256 processors.

Resource Allocation and Protection

With functional partitioning of the tasks only the data that must be shared by individual processors passes over the function-interconnect bus. Resources may be either private, thus only accessible by a single processor or resources may be global and shared by several processors. In this latter case a reservation mechanism by semaphore and TEST and SET functions associated with a source signature mechanism is implemented (ESONE, 1983). One byte of the 24 bit address on the communication bus, representing the source processor identifier, is automatically loaded into a signature register of the resource during a successful reservation access. From this instant on this signature becomes part of the full 24 bit address decoding while before reservation the resource responds to 16 bit address only. From then on only the successful processor can access this resource or release it. Deadlock situations are monitored by the supervisory unit.

THE VMEbus USED FOR MULTIPROCESSING

In 1982 CERN choose the VMEbus and the 68000 microprocessor family for all future projects following the recommendations of an internal Working Group. Thus the multi-microprocessor project for the LEP control system has been implemented in VME. The large industrial support allowed us to concentrate our effort on the implementation of the distributed multiprocessor concepts and to use standard products available from many VME manufacturers for all common facilities like: SASI drivers, display functions, diagnostics modules, memory extensions, crates, busses and power units, etc.

While the VMEbus specification provides good facilities for several processors plugged in a crate its architecture is not ideal for true distributed processing with up to twenty closely coupled microcomputers communicating by message protocol. To achieve its goal CERN has studied and developed new facilities fully compatible

Fig. 4. Process Control Assembly

with the VMEbus specification. In the future some of these facilities will be provided by the serial VMSbus chip set. These compatible developments and operation procedures include a distributed arbitration mechanism, a programmable interrupt system, a private memory extension over P2 connector, and a multimaster-multicrate highway linking up to seven VME crates. All these extensions allow the use of standard VME manufacturers' products.

Distributed arbitration

This development allows to mix in the same VME crate modules working with the daisy chain arbitration and modules using a distributed arbitration approach in conjunction with a special system controller plugged in slot 1 (Rausch, 1984). Most new multiprocessor busses have adopted the distributed arbitration mechanism: FASTBUS, SMBUS, MULTIBUS II, NUBUS, P896 FUTURE BUS. Distributed arbitration offers position independence and the possibility to implement easily fairness and dynamic high priority. The fairness algorithm ensures that every board gets its fair chance to access the bus; no board is forever locked-out by a higher priority master. The high priority algorithm allows the designer to define a priority structure among all of the boards in the system. Modules handling real-time events are usually given the highest priority while less critical boards get a lower priority. In the CERN project this capability is used extensively for passing urgent messages and event-like interrupts between processors. Changing between the two algorithms can be done dynamically; the designer is free to choose one or the other, or a combination of the two. Distributed arbitration avoids the burden of manipulating numerous daisy-chain jumpers on the backplane in case of system reconfiguration, as processor's relative priority is position independent. But the biggest advantage given by distributed arbitration is the diagnostic possibilities. While it is difficult with the daisy-chain mechanism to know which processor has generated the current bus cycle at any instant it

Fig. 5. General Purpose Processor Unit

is straight forward with distributed arbitration. Each bus cycle is accompanied with the priority code of the current master processor, thus bus activity monitoring can be selective and very flexible; a bus error can be trapped and attributed to the faulty master.

The implementation of this VME compatible arbitration uses three of the four pairs of Bus-Request Bus-Grant In/Out lines. BR3 and BG IN/OUT3 are not touched but are used for ordinary VME arbitration. All three BG IN/OUT pairs of pins are strapped at all module slots including at the first and the last positions. This gives six lines (BR0, 1, 2 and BG 0, 1, 2) which are terminated on each side of the dataway by the standard VME resistor network. These six lines are used for priority encoding; five lines for priority 0-31 and the 6th for exceptional high priority access, typically for urgent messages or an event-like interrupt cycle as explained later. Processors developed for the CERN multiprocessor project support either arbitration methods by replacement of a little daughter board. To provide full mixing capability the system controller in slot 1 combines the daisy-chain arbitration BR3 with the distributed arbitration and leaves the BR3 chain for top priority as specified by the VME standard.

Programmable Interrupt System

The VMEbus provides seven Interrupt Request lines IRQ1-7 and a single daisy-chained Interrupt Acknowledge line. In a multiprocessor system the seven IRQ lines may be activated each by any processor and each request line ends in one of seven processor boards. Unless several sources share a given interrupt line and software polling is used, seven is the maximum manageable number of processors one can use in a VME crate system. Each processor has first to become the master of the bus before it can acquire the interrupt vector. This is a serious limitation for the implementation of a flexible multiprocessor architecture. The CERN project has therefore used a programmable and addressable interrupt-event concept which is nowadays also applied on other standard busses (Beaston, 1984).

The principle is the following: when a source processor needs to interrupt a destination processor it requests bus mastership with high priority, i.e. its normal priority code plus activation of the sixth priority code line. The source processor performs a programmable write

Fig. 6. Remote Reference Memory

cycle associated with a 16 bit interrupt vector data word; the high byte contains the source processor code (3 bits for crate number and 5 bits for the processor number) and the low byte contains the interrupt vector code. The 16 bit address is generated with the standard short I/O address modifier code. All processors contain an Interrupt FIFO (16 bit wide, 32 words deep) receiving these interrupt vector codes. Thus each processor can interrupt or be interrupted by up to 20 processors in a single crate system and by up to 140 processors in a multicrate system with seven VME crates. The multimaster-multicrate highway carries these interrupt vector words as any other programmed Input/Output transaction.

This simple but very powerful interrupt mechanism uses the dynamical high priority arbitration implemented either with the normal four level daisy-chain priority or with distributed arbitration as explained above. It allows straight forward interrupt source identification provided that a coherent module and crate memory address-mapping has been adopted during system configuration. This method can be seen as memory-mapped interrupts where the source module performs a write into the register address space of the destination to trigger an interrupt. This creates so many "virtual" lines that the designer has not to worry about running out of lines among modules.

Local Bus

In the CERN multi-microprocessor project the VMEbus is used mainly as a function to function interconnect bus carrying messages from source to destination processor. Considering the small size of the VME card and the functionality required from the various microcomputer boards an extension capability is often required for additional memory or private I/O. This extension can be done via rows a and c of the P2 connector, either by following the VMXbus specification or by using these 64 pins freely as user I/O pins according to the VMEbus specification. For private memory extension on a local bus the full microprocessor speed must be usable without introducing wasteful wait states. As VMXbus specifies multiplexing of address and data it has been considered too stringent to multiplex A/D at twice the microprocessor speed to benefit from the full microprocessor power. Thus to satisfy CERN's requirements a straight forward expanded local private bus for the 68000 processor family has been specified on the 64 user I/O pins and named VMCbus (VMCbus Specification Manual, 1984).

The VMCbus is an extension of the 68000 processor bus to provide a high speed secondary path which is optimised for connecting up to six boards in a subsystem configuration. This subsystem can transfer data from board to board over its VMCbus interface without waiting for and without

burdening the primary bus (VMEbus). A set of two or more modules interconnected via the VMCbus behaves like one big VMEbus module with its own internal bus, freeing the VMEbus for other transfers.

The 68000 vectorised and autovectorised interrupt mechanism is provided over the VMCbus allowing any card connected to the VMCbus to interrupt the processor card directly, thus saving unnecessary mastership arbitration in a multiprocessor system over the VMEbus with improved response time. The 68000 Direct Memory Access mechanism is supported over the VMCbus and multiple DMA masters can communicate over the VMCbus with the processor's private memory.

The VMCbus specification permits to use directly all 68000 and 6800 family peripheral chips as the signal timing and addressing are those of the 68000 processor.

A board that is inserted from the front of the card rack may have a VMEbus interface, a VMCbus interface, or both. For example, a global I/O board might have only a VMEbus interface, a CPU or a memory board might have both; while a private memory, a private I/O or a math-processor might have only a VMCbus interface. Any board that has a VMCbus interface uses the two outer rows of the J2 connector.

The use of a ribbon cable or alternatively a small PC backplane to bus the P2 connectors allows any group of up to six adjacent slots to function as a subsystem. The user can install two or more P2 cables or backplanes to create several VMCbuses in a single card rack. Each of these VMCbus subsystems can operate independently of the primary system bus (VMEbus) and independently of each other. It also permits some slots to be used for other purposes such as I/O signals.

Multimaster-multicrate highway

An accelerator control system needs provision for connecting numerous equipments to the local minicomputer. At the SPS for example, up to eight CAMAC crates are controlled by the minicomputer I/O bus and up to eight serial multidrop busses (MPX system) connect 32 crates each to serial drivers located in CAMAC crates. In addition local intelligence may have to be associated logically with some I/O subsystems, similarly to CAMAC-ACC's. An homogenous communication amongst these intelligent VME I/O crates is necessary for large and powerful systems. This example demonstrates the need for a multicrate highway allowing several VME crates to interface equipment either directly in a point to point distribution or via a multidrop serial bus over long distances.

As a consequence the LEP multiprocessor assembly must be expandable over more than one VME crate. To satisfy this requirement a VME multimaster-multicrate highway has been developed (Rausch, 1984).

For the multimaster-multicrate highway (Fig. 7) the following design criteria have been fulfilled:

- All standard VMEbus cycles with all address modifiers are supported.
- Full 32-bit address and data are transferred over the highway in an A/D multiplexed mode.
- Differential drivers and receivers according to RS-485 use a twisted pair line for each signal allowing interconnection of seven VME crates over distances of 30 meters.

Fig. 7 Multicrate-multimaster highway

- Each VME crate can support several processor modules capable of gaining mastership of the highway and accessing memory or I/O resources in any other crate. A daisy-chain arbitration organises the priority amongst the VME crates. Each highway bus linker module has top priority in the crate it is located, minimising arbitration time in the destination crate. Crates not concerned by a data transfer over the highway do not participate in the arbitration; only the source and destination crates are involved in a transaction.

- Data transfer can be fully interleaved at a cycle by cycle basis, but sending small data blocks saves arbitration time.

- Interrupt-like events are sent over the highway in the same fashion as described in chapter 3.2. Any slave or any processor can send its interrupt vectors with source identification to any other processor in a single high priority transaction.

- When a processor tries unsuccessfully to access a remote resource over the highway due to heavy traffic it uses the microprocessor's re-run facility to try again later. Bus error conditions and deadlock situations have been taken care of.

MERITS OF THE ARCHITECTURE

Compared with conventional computer systems the function-to-function multiprocessor architecture implemented for the LEP controls system offers a number of present and future advantages:

- This architecture allows simple designs of complex systems by permitting system requirements to be broken down into identifiable, manageable tasks associated with real world needs (DL, BC, SU, ED, MS, MI, FP modules, etc.).

- It provides excellent system throughput, with very low bandwidth over the function-interconnect system bus (VMEbus) and very high bandwidth through the private local bus (VMCbus) of each separate processing module.

- Low cost, high density electronics following VLSI improvements is offered. Tomorrow's components will be functional modules tailored to an

application using just the components necessary for processing, memory and I/O operations.

- Functional modules can be grouped into clusters to raise significantly the performance level of a particular function. (Several MI modules in a PCA allow parallel interpretation of commands expressed in high level language.)

- Functional modules can be duplicated and clustered to increase a function's reliability. (In a PCA two DL modules allow connection on a dual communication network for redundancy.)

- This architecture allows hardware/software transportability of functions without impacting system operation. Versions of the same function can be implemented in software, embedded in software on-chip with a microprocessor, or for highest performance, designed in hardware. (In the future, a language processor module, MI or FP, may be implemented with a special microprocessor with an instruction set optimized for this language.)

- Designs are simple to upgrade or modify provided the system bus interconnection logic remains the same. Functions literally can be plugged into or be removed from an existing system with no adverse effects on control or scheduling operations.

- Alternatively FFA functions are bus-independent. Functions developed for a particular bus can operate without modification on other computer system buses. Dedicated bus-coupling circuitry adjusts for differences.

- This architecture offers economy by permitting functions to be implemented in any technology that can perform the desired function. This also applies to software and hardware. A function constructed in software can be converted to hardware to satisfy the higher performance requirements of a particular part of a system.

SOME SOFTWARE ASPECTS

The above description of the hardware structure for the LEP control system has an important consequence on the conception of the software.

Such a structure allows the cutting of the global control problem into units of practical size which can be solved separately. This fragmentation capability is firstly exploited by assigning a PCA to each geographical functional subsystem such as the beam instrumentation, accelerating system, vacuum, power converter. Each of these PCA's will communicate via the token ring essentially by means of messages containing programs or data in a canonical form.

A further level of fragmentation will be achieved within a PCA via the function-to-function architecture. This architecture is based on autonomous GPU-based modules, which emulate a real-time multi-tasking conventional operating system. The communication between the units is made on the VMEbus by means of messages which contain requests for service and data.

This implies that the elementary software package is a function which, when it is well defined, is simple to develop and debug. The overall distributed operating system is made by combining, in a PCA, the appropriate functions.

The essential element for such a system to work is a well defined communication convention between GPU's in a PCA, and between PCA's.

The information which flows on both communication systems is basically of the same nature, they are messages. The ISO Open System Interconnection reference model is ideally suited for this kind of problem, it offers a coherent framework for this implementation of a uniform scheme for the data flow in the overall system (Altaber, 1984).

Another important aspect of the hardware structure comes from the fact that a function being logically defined is independent of the hardware module which realizes it physically. If tomorrow a better hardware module is found for performing a specified function we could use it without any major change in the design.

For this statement to be fully true it requires that the software which was in the module can be mostly re-used. The achievement of this requires the use of a high level language widely accepted in order to guarantee the availability of compilers. Waiting for ADA to come, it has been decided to use the Modula-2 language which offers all the necessary features for writing the operating system programs and which can be translated later into ADA with automatic packages.

REFERENCES

- Ali, G., A.D. Hirschman, and R. Swan (1979). Standard Modules offer Flexible Multiprocessor System Design. Computer Design, May, 1979.
- Altaber, J., F. Beck, M.C. Crowley-Milling, and R. Rausch (1979). Suggested Principles for the Control of Future Accelerators. IEEE Trans. on Nucl. Sc., Vol. NS-26, No. 3.
- Altaber, J., F. Beck, and R. Rausch (1980). Accelerator Control Systems without Minicomputers (1980). XIth Intern. Conf. on High-Energy Accelerators, CERN, Geneva.
- Altaber, J., F. Beck, M.C. Crowley-Milling, and R. Rausch. Truly Distributed Control Using one Microprocessor per Real-Time Task (1982). IVth Intern. Conf. on Trends in On-Line Computer Control Systems, Warwick.
- Altaber J., M.C. Crowley-Milling, P.G. Innocenti, and R. Rausch (1983a). Replacing Mini-Computers by Multi-microprocessors for the LEP Control System. Particle Accelerator Conference, Santa Fé.
- Altaber, J., and R. Rausch (1983b). A Multiprocessor Bus Architecture for the LEP Control System. Particle Accelerator Conference, Santa Fé.
- Altaber, J., V. Frammery, C. Gareyte, R. Rausch, and P. van der Stok (1984). The Message Architecture of the LEP Control System. Nuclear Science Symposium, Orlando.
- Barthmaier, J.P. (1980). Multiprocessing System Mixes 8 and 16 bit Microcomputers. Computer Design, February, 1980.
- Beaston, J., Intel Corp. (1984), Multibus II An Architecture for 32-bit systems. Electronic Engineering, March, 1984.
- Conrad, M., W.D. Hopkins, L. Spry, A. Orban, A. Nawaz, and I. Ansari (1981). Functional Architecture Threatens Central CPU's, Electronic Design, September, 1981.
- Crowley-Milling, M.C. (1983). The Control System for LEP, Particle Accelerator Conference, Santa Fé.
- E3S - ESONE Standard System Specification (1983). Proposal submitted to the ESONE General Assembly, Berlin.
- Rausch, R., (1984). Bus Multichassis-Multimaitre compatible VME. Conférence sur les Systèmes à Microprocesseurs, Ecole Polytechnique, Paris.
- Rausch, R., J.M. Sainson and G. Surback (1984). Distributed Arbitration Compatible with the VMEbus Standard. CERN internal report SPS/ACC/Note/83-18, CERN, Geneva.
- Swan, R.J., et al. (1977). The Implementation of CM* Multiprocessor. AFIPS Conference Proceedings, pp. 645-655.
- VMEbus Specification Manual (1984). CERN internal report, SPS/ACC/Tech.Note/85-2, CERN, Geneva.
- Preliminary Specification for the Supply of Software and Hardware for the Multiprocessor Control Assemblies (1984). Tender I-1398/LEP, CERN, Geneva.