NINTH EUROPEAN ROTORCRAFT FORUM

Paper No. 40

A TOP DOWN APPROACH TO STRUCTURED SOFTWARE DESIGN FOR THE M.A.R.A. AIRBORNE COMPUTER

A. DI GIOVANNI
and
L. PADELLA

SELENIA INDUSTRIE ELETTRONICHE ASSOCIATE S.p.A.
(POMEZIA)

ITALY

September 13-15-1983
STRESA, ITALY

ASSOCIAZIONE INDUSTRIE AEROSPAZIALI
ASSOCIAZIONE ITALIANA DI AERONAUTICA ED ASTRONAUTICA

A TOP DOWN APPROACH TO STRUCTURED SOFTWARE DESIGN FOR THE M.A.R.A. AIRBORNE COMPUTER

A. Di Giovanni and L. Padella — SELENIA, Industrie Elettroniche Associate, S.p.A.

A B S T R A C T

The fast growth of processing structures making use of microprocessors (the so called microprocessor revolution) has stimulated a parallel growth in the S/W development methodologies for said processors. This paper describes the results of five years of S/W definition activity carried-out in Selenia for the SL/AYK-203 airborne computer (MARA line), in order to provide the application S/W designers with powerful, general and simple to use strategies. The main tasks considered as study guidelines have been the following:

— design of a H/W architecture with capabilities of real-time, modularity, reliability, adaptability to the technological evolutions.

— implementation of a basic S/W architecture with the same H/W characteristics.

— old methodologies proceedings in order to get a new approach to meet a low cost/quality ratio.

— definition of a complete S/W factory (open to ADA) based on widely known commercial computers.

1. MARA FAMILY

The MARA acronimous "Modular Architecture for Real Time Applications" exactly expresses the anticipated objectives.

We are not concerned with a computer or a system, but rather with a modular architecture, i.e. personalizable units which may be connected by means of a set of uniform principles in an extremely diversified way with a high degree of flexibility. Its characteristics of simultaneous processing management and of high throughtput, fault tolerance, real-time make MARA in the mono-version, multi-version, matched-pair or local network configurations, particularly suited to avionic applications.

There are many reasons which gave rise to the design of a sectional modular system with a high degree of versatility.

The role of MARA has a considerable importance in the military avionic field. In this sector it is very important to have an extremely modular instrument which, in the face of very diversified requirements, allows easy adaptability of equipment and know-how. Furthermore a very attractive solution is the availability of an expandable architectural scheme oriented towards a high throughput and a large memory capacity (or a mix of both) without:

— having to redesign H/W units;

— meeting costs of new prototypes and new lines of H/W and S/W development;

— having to acquire a new know-how.

In the MARA project two aspects of the degree of modularity of a system, normally absent in the widely known computers and often considered opposing one another, have been recognized as fundamental:

— the availability of a concentrated processing power modularly upgradable;

— the capability of physically distributing the subsystems of a system.

The solution which has been realized by MARA has passed the old "diverse-components combination" approach, which inevitably increased the complexity of the architecture and contained within itself the limits of its development. MARA solution, on the other hand, defines uniform elementary units, capable of being combined in many ways, but without changing the basic philosophy.

The key elements of the success of this approach are bound to sub-objectives such as:

— technological elements;

— architectural aspects;

— extreme attention to containing the costs of the equipment;

— instruments and methodologies for a S/W development at a low cost/quality ratio.

1.1 Loose and Tight Coupling

In the MARA architecture are present both fundamental types of computer interconnection: loose and tight coupling.

These types of coupling can be met in H/W as well as in S/W. From a H/W point of view several MARA nodes can be connected in a local network (e.g. via the 1553B bus) and then they interact loosely.

But each node of the network can be a mono-processor or a multi-processor, i.e. with a common bus and a memory (or peripherals) in common, so as to interface tightly with the other processors constituting the node.

Similarly from a S/W point of view the same type of collaboration exists between tasks. A collaboration referred to as "loose" requires that the parallel tasks do not have objects in common, but can collaborate exclusively by an exchange of messages or by explicit resources managers, which in effect eliminate the parallelism between tasks to guarantee a correct resources utilization.

The other so called "tight" connection allows the parallel tasks to have a direct visibility of the common resources and leaves the application programmer to manage these resources.

The loose collaboration is, of course, much easier and safer for the application programmer to use; but very often the tight one allows faster and more efficient solutions. This type of cooperation derives from the possibility that the several processors of a multi-micro configuration collaborate in the same work to obtain a modular processing power.

On the other hand, the loose collaboration techniques are more appropriate when a physical distribution of the collaborating tasks is requested. In this case, because the interactions occur via the operating system, the basic S/W hides from the user the inter-computers communications. This is the case of tasks located on physically separated nodes and therefore without a common memory.

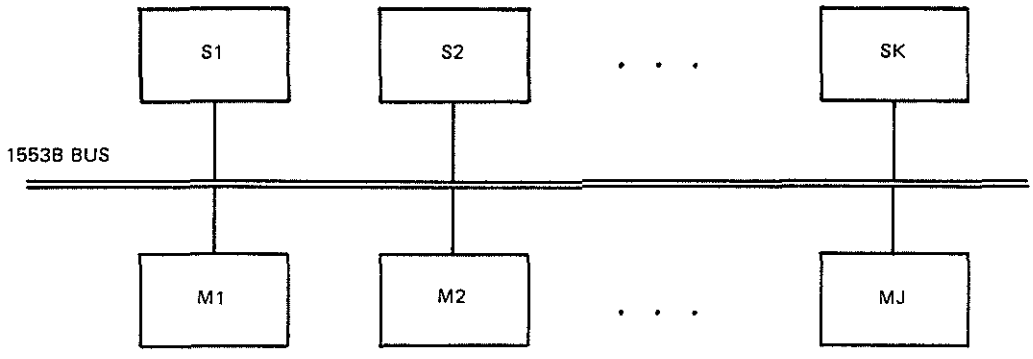
2. SL/AYK-203 ARCHITECTURE

The MARA solution to the modularity requirement offers essentially:

— two levels of modularity in both the H/W and S/W structure of a system; in both cases the first level of modularity supports the loose coupling, while the second supports the tight one;

— a set of fundamental mechanisms both in the H/W and S/W, which allow to define, after the application S/W development, the H/W configuration of the system and the related S/W modules distribution.

At the first level of H/W modularity the loose coupling is thus found, so that a system can be seen as local network in which "n" MARA nodes are present (fig. 1).
The local network offers a total coupling between all the nodes present; the access time to a single node depends only on the traffic on the inter-node connection bus and not on the relative position of the transmitter and receiver.



Si = single-processor node (i = 1,k)
Mi = multi processor node (i = 1,j)

fig. 1 — LOCAL NETWORK CONFIGURATION

The second level of H/W modularity, i.e. the tight coupling, is found within the environment of the single node. In fact in a multi-processor node each processor can access to a local bus containing its private memory resources, but it can also access to a nodal bus, containing memory and I/O resources common to all the processors (fig. 2).

The H/W configuration of each node is realized and modified by using exclusively standard modules listed in the catalogue.

The maximum number of processors in a node is 16.

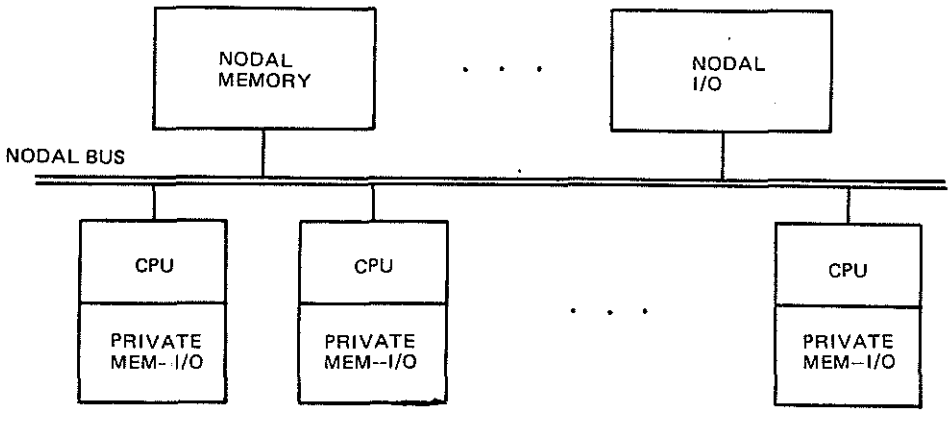


fig. 2 — SL/AYK-203 NODE ARCHITECTURE

One of the basic elements of a versatile architecture, i.e. oriented to different types of problems and durable in time, is that of being easily adaptable to new components, in particular micro-processors, whose performance is continually and rapidly improving.

For this reason the following basic choises have been considered as the relevant and qualifying elements:

— the various processor units are general purpose;

— the architecture is not based on a micro-processor but on a complete component family (with the evident possibility of up-grading);

— no bound exists to a specific technology.

2.1 Fault Detection And Tolerance

The fault detection characteristics are one of the most qualifying aspects of the MARA system, which provides this performance not as an element which is difficult to manage, but as a set of basic elements which are completely transparent to application programmer.

The fault detection is obtained both by means of H/W and S/W facilities. From a H/W point of view, a complete set of controls and loop-backs is provided. But the most important fault detection instrument is represented by a powerful protection mechanism, which protects each functional S/W environment from the others, using four privilege levels. The protection environments oblige the application programmer to respect specified access rights in order to avoid violation errors and consequent invocation of the Exception Handler. In this way any H/W fault or S/W malfunction is immediately detected and a recovery task is triggered. The protection mechanism is therefore very useful during the S/W testing phase.

From a S/W point of view, a set of powerful on-line diagnostics and a complete redundancy management service, are provided.

Concerning fault tolerance, a typical approach is given by the role-switching in a matched-pair configuration (fig. 3).

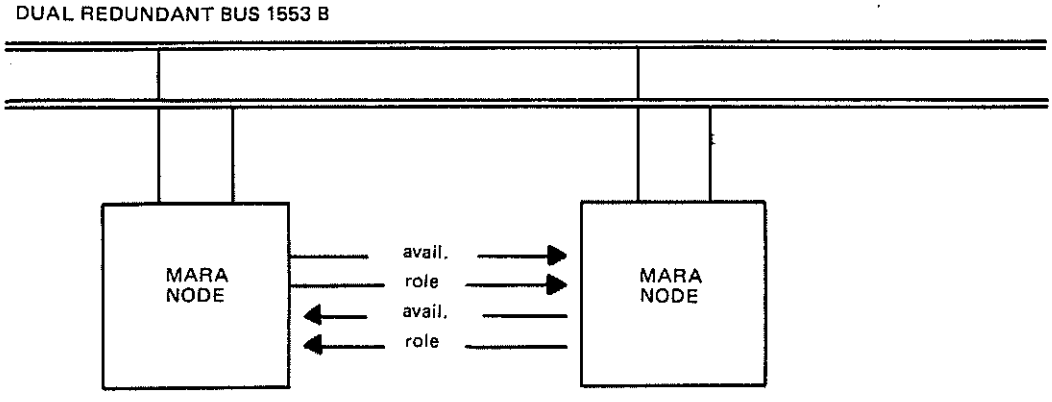


fig. 3 — FAULT-TOLERANT CONFIGURATION

3. S/W APPROACH

The characteristics of modularity, reliability and real-time, which are the objectives of the MARA H/W architecture, are also reflected in the S/W one.

The two highest levels of modularity (loose and tight) have their virtualization also at S/W level. At the first level (loose coupling) the application S/W is organized as a certain number of FUNCTIONS. Each function is a set of programs, tasks, data structures, considered by the disigner as operatively homogenous.

A function is therefore a set of agents (i.e. everything that corresponds to instructions executed by a processor) operating on a set of objects (passive data structures declared by the programmer).

In the loose mode the functions communicate one another by means of executive services which allow the exchange of messages via a path network. A path is a simple virtualization of a classic point-to-point full duplex line of communication.

Each function must be assigned to a node but clearly several functions can be located on the same node.

The important advantage of having standardized the interface between the functions by means of paths is that of being able to perform a communication independently of the physical functional location; in other words an absolute separation between the functional structure and the physical one is realized.

It is just this feature which makes the use of a mono-processor, multiprocessor or network completely transparent to the application user.

The operating system service which manages the communications via paths, is called "path-manager" while that which manages the inter-node communications is called "network manager". The use of the network manager is optional for the user dependently of the application.

As the second level of H/W modularity allows the user to organize each node in the desidereed manner, so the second level of S/W modularity (tightly coupling) allows each function to be organized as a set of tasks which collaborate in a tight mode. In this case each function of a node is protected by the others using the protection mechanisms offered by the H/W.

A task is the simple virtualization of a processor; each task must be executed by a processor or by more processors; a processor can be shared by several tasks, even belonging to different functions. The competition between the various tasks for the possession of each H/W processor is resolved by the scheduler which uses several priority levels and a time slicing algorithm at each level, to equally share the processing power between all the tasks of the same priority.

3.1 MACH Operating System

The MACH (Modular All Core Cooperation Handler) operating system provides the suitable primitives to facilitate the cooperation between tasks belonging to the same function, placing three types of objects at the disposal of the programmer:

— spin locks: to obtain short periods of mutual exclusion

— semaphores: to obtain long periods of mutual.exclusion or to obtain syncronization between tasks

— mailboxes and messages: to perform the communications between tasks

The inter-function and inter-node communications are also supported by MACH respectively as path servicing and networking. Moreover the operating system also provides other fundamental services as interrupts, real-time clocks and memory-segments management, etc.

All the above said features are performed by the Real Time Executive, the first of the two main blocks in which MACH has been devided. The second main block BOSS (Built-in On-line System Supervisor) is that part of the operating system which realizes the man-machine interface and supervises the correct functioning of the node. In particular the fundamental duties of the System Supervisor are as follows:

— to take control after the H/W power-on and diagnostic the basic H/W in order to determine the equipment operational conditions and decide on the activities to perform

— to provide a set of powerful debugging instruments which may be used in ON-LINE mode (i.e. as a Executive's task) during the S/W debugging phase, and selected either in local or in network configurations

— to act as default exception handler when the user's one is missing

— to manage the role-switching in the matched-pair configurations in a user-transparent way

— to realize a standard protocol with the S/W development centers, acting as a monitor for the S/W factory tools (i.e. remote debugging assistant), which need interactions whith the target machine

— to realize a standard protocol with a remote test-set (either intelligent or not) for maintenance purposes.

Real-Time Executive and System Supervisor have been developed by means of an efficient structured design and a simple modular growth, according to the "abstraction-levels" methodology. The H/W privilege levels reserved for these packages are the first and the second of the general protection scheme. Levels "two" and "three" are at disposal of the user for the application programs (fig. 4).

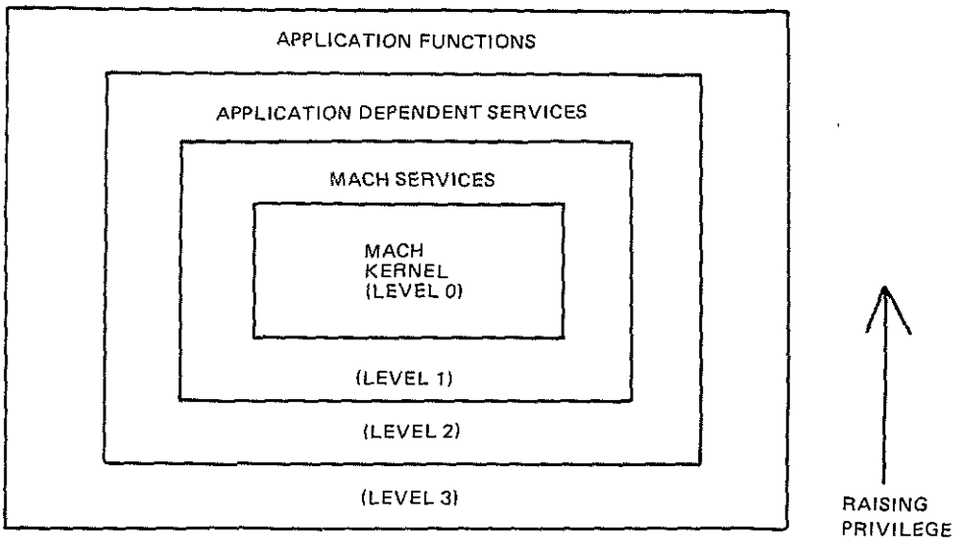


fig. 4. — PRIVILEGE LEVELS SCHEME

The Executive services (common to the other lines of the MARA family and the Supervisor features have been specifically designed for airborne configurations, in order to meet the fundamental constraints of: minimum CPU and storage overhead, very short response times, high reliability.

3.2 Traditional Design and MARA Approach

In the design of a real time application system, the designer finds that he must suitably resolve two types of problems largely interlaced:

— the system functional architecture definition

— the system H/W architecture definition

The best adherence to the real needs and the best possible quality may be obtained if the software designer can ignore the detailed H/W needs in defining the project functional architecture, thus transferring his attention to the construction of an abstract model of the problem solution.

With MARA the two problems become completely uncorrelated, with the enormous advantage that each aspect may be modified independently from the other.

Concerning the traditional approach to the S/W system design, the designer's proceeding way consists of the following phases:

— translation of the problem to a functional abstraction

— logical decomposition

— physical decomposition

— program development

This approach presents several main disadvantages:

— no connection exists between functional abstraction and logical decomposition

— the physical decomposition is a too heavy phase

— too many decisions are necessary at the same time, because it is impossible to separate the definition of the modules from their physical allocation

— the result of this method is an unmodifiable system, because the physical decomposition precedes the S/W development.

From the MARA point of view instead, a functional decomposition phase follows immetiatly the functional abstraction one; in this way the functions defined during the first phase become directly known by the operating system. The next stes is the development of said functions (using programming techniques) and the system configuration and tuning.

MARA approach leads to an easy and rational S/W development which produces a completely adherent system to the functional decomposition and allows an optimized system configurability and modifiability.

It must also emphasize that a multimicro or even a multinode presents with new possibilities and at the same time with new problems to the designer:

— what is the best way of using all these processors ?

— what is the best way of dividing the application functions between the various computers ?

— what is the best way of parallelizing a traditional sequential algorithm so as to esploit the real parallelism offered by the H/W ?

— what is the best way of dividing a function between the common memory and the local memories of the single processors ?

For all these reasons the designer must be able to express the parameters and the implementation aspects of a function which can be changed so that one does not have to modify or rewrite the programs to vary the configuration. The same requirement applies for those functions which are of a general nature and must be allowed to be reused in other systems. In other words it is necessary for a function to be configurable, i.e. adaptable to the needs of a particular system without having to modify its source modules.

This has led to the idea of developing a System Configuration Language (SCL) which runs on the S/W factory. This enables the MARA designer:

— to arrange the system S/W architecture in terms of functions;

— to completely develop the S/W in terms of compilation units;

— to define the H/W architecture on the basis of performance requirements

— to distribute the S/W functions on this H/W architecture.

The definition of a system with SCL is articulated in two parts:

— a part, called "descriptor", which contains a general description of the programs which realize the system, expressed in a parametric and H/W independent way;

— a part, called "configurator", able to generate a particular configuration for the system.

This means that, to modify the configuration of a certain system, it is sufficient to modify the configurator section of the system description without rewriting any modules.

The output of the SCL compiler is interpreted by another tool (running on the S/W factory), which combines the system description (made in SCL) and the user files (previously compiled and assembled) to produce automatically a directly loadable and executable output for the SL/AYK-203 or the Eprom fusing tapes.

4. S/W FACTORY

By Software Factory we mean a set of methods and tools which represent the support to all the necessary functions for the development, maintenance and management of S/W projects in their complete work cycle.

The factory used by Selenia for the development of programs for the SL/AYK-203 is based on the VAX family computers (fig. 5) and presents the following particular characteristics:

— it can be considered a universal factory, i.e. may be used for different microprocessor families

— it is multi-user

— it is oriented towards the maximum portability

— it is equipped with a powerful set of instruments at the state of art

— it is oriented towards requirements and problematics related to the future ADA language and to the software environment necessary to support it.

A large variety of tools are availabel on the S/W factory such as assemblers, high level languages (PASCAL, FTN,...), linkers, locators, editors, system configurator (SLC), fusing tapes generator, etc. A remote debugging assistant is also available to allow the user's programs running on the SL/AYK-203 to be directly VAX cross debugged, using a VAX-target serial link.

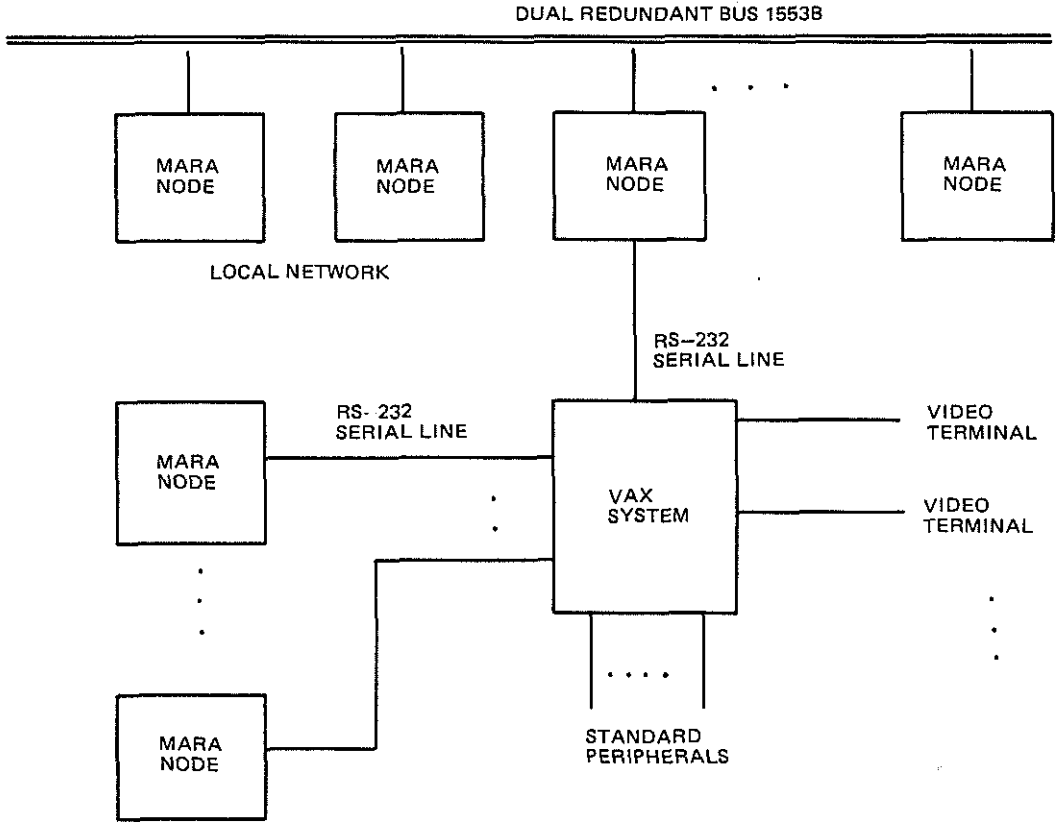


fig. 5 — SOFTWARE FACTORY CONFIGURATION

5. CONCLUSION

The SL/AYK-203 project developed in Selenia in the last years along the MARA line, is a typical example of integrated H/W-S/W design, in order to satisfy the following needs:

— availability of modular processing systems with the dual aim of application flexibility and of possible modular growth where necessary.

— availability of distributed processing networks, as may be required for modern military aircrafts making use of standardized H/W-S/W mechanisms

— availability of processing structures featuring a high-efficiency execution of code generated from modern hig-level languages.

SL/AYK-203 meets all these requirements by providing:

— H/W modularity, also in terms of processing power and configurability.

— very powerful fault detection and tolerance mechanisms

— complete transparency of both basic and application S/W versus the H/W configuration

— powerful set of logistic support tools including built-in test facility and telediagnosis capability

— complete S/W factory, available on widely known commercial computers

— H/W growth capability open to future evolutions, with complete portability of the S/W packages developed for today's H/W.

The practical results of the development program are fully in accordance with the above requirements.

REFERENCES

1) *Ian R. Martin:* "MARA overview for system designer's '.
Selenia — SPE 80012 internal — 2/6/80

2) *G. Lisi:* SL/AYK-203 Technical Specifications".
Selenia — ST 83057 internal — 31/5/82

3) *A. Di Giovanni:* "M.A.C.H./1 Executive".
Selenia — DS 83003 internal — 2/5/82

4) *A. Di Giovanni:* B.O.S.S. Supervisor".
Selenia — DS 83004 internal — 10/6/82

5) *Ian R. Martin:* "La modularità sistemistica del sistema MARA".
Presented at the CONVEGNO NAZIONALE ANIPLA '82. 17, Vela street, Torino (ITALY), November 24-26

6) *G. Ballaben:* "Il linguaggio di descrizione per la generazione di sistemi multiprocessor MARA".
Presented at the CONVEGNO NAZIONALE ANIPLA '82. 17, Vela street, Torino (ITALY), november 24-26