



HAL
open science

Ariane 5-ALF: an HW/SW Architecture driven Data Handling System evolution

O. Notebaert

► **To cite this version:**

O. Notebaert. Ariane 5-ALF: an HW/SW Architecture driven Data Handling System evolution. 2nd Embedded Real Time Software Congress (ERTS'04), 2004, Toulouse, France. <hal-02271071>

HAL Id: hal-02271071

<https://hal.science/hal-02271071v1>

Submitted on 26 Aug 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization



Session 5A: Hardware & Software Architectures

Ariane 5-ALF: an HW/SW Architecture driven Data Handling System evolution

Contact Author and Speaker: O.Notebaert (EADS Astrium)

EADS Astrium, 31 avenue des Cosmonautes – F-31402 Toulouse Cedex 4 (France)
Phone: +33 5 62 19 74 23, Fax: +33 5 62 19 78 97, E-mail: olivier.notebaert@astrium.eads.net

Co-authors: Arnaud Stransky & Vinh-Quy Ribal (EADS-ST), Thierry Planche (EADS-Astrium),
Hans Corin & Torbjorn Hult (Saab Ericsson Space), Georges-Albert Bonnerot (CNES)

Abstract

In the coming years, the Ariane 5 On-Board-Computer (OBC) will handle missions and performances enhancements together with the need for significantly reducing costs and the replacement of obsolescent components. The OBC evolution is naturally driven by these factors, but also needs to consider the SW system compliance. Indeed, it would be a major concern that the necessary change of the underlying HW should imply new development of the flight software, mission database and ground control system.

The Ariane 5 SW uses ADA language, which enables verifiable definition of the interfaces and provides a standardized level of the real-time behavior. To enforce portability, it has a layered architecture that clearly separates application SW and data from the lower level software. In addition, the on-board mission data is managed thanks to the extraction of an image of the systems database located in a structured memory area (the exchange memory). Used for all interchanges between the system application software and the launcher's subsystems and peripherals, the exchange memory is the virtual view of the Ariane 5 system from the flight SW standpoint. Thanks to these early architectural and structural choices, portability on future hardware is theoretically guaranteed, whenever the exchange memory data structures and the service layer interfaces remains stable. The ALF working group has defined and manufactured a mock-up that fulfils these architectural constraints with a completely new on-board computer featuring improvements such as the microprocessor replacement as well as an advanced integrated I/O controller for access to the system data bus. Lower level SW has been prototyped on this new hardware in order to fulfill the same level of services as the current one while completely hiding the underlying HW/SW implementation to the rest of the system. Functional and performance evaluation of this platform consolidated at system level will show the potential benefits and the limits of such approach.

Ariane 5 Data Handling Subsystem

The Ariane 5 Data Handling is part of the launcher's electrical system, which is required to remain operational in case of failure of any of its equipment. Additionally, the allowed autonomous reconfiguration time is limited, especially during the most critical mission phases. The Data Handling Subsystem has been therefore organised as dual redundant chain of sensors and actuators (Fig.1).

The communication system (SdC for *Système de Communication*) is implemented with Mil-std-1553B standard components and is the only data link between the computer pool and the equipments. This computer system operates in a hot active/standby configuration: in nominal situation, the Master computer (OBC1) controls the communications on both buses and executes the flight software. The Backup computer (OBC2) monitors the bus traffic in parallel and



maintains the flight dynamic data context running the same algorithms. The flight software is able to control and monitor any nominal or redundant remote unit and to eliminate faulty items from the functional chain. In addition, the Master computer has the capability to self-detect any abnormal local situation, electrically isolate itself from the communication bus and trigger a switchover to the other computer. In such situation, the Backup computer is able to take over the flight control in a very limited time frame.

On-Board Computer

The choice of using a standard serial bus as the backbone for the avionics system communication has driven the internal architecture of the On-Board-Computer. Indeed, in the late eighties, the best way to organize the system communication scheme with strong low-level synchronization constraints was to control a standard 1553B I/O device by a dedicated SW driver. In addition, the performance needed for the flight software required to de-couple the data acquisition and distribution function from the flight control application. The OBC has been therefore designed with three functional blocks (Fig.2): A Power supply (APS), an application processor board (UT) and an I/O controller (UES) providing access to exchanged data in memory. Both processor boards are implemented around 68020 microprocessors that meet the Ariane 5 standard radiation requirements.

On-Board Software

The Ariane 5 Flight Software is responsible for the launcher's flight control. Its properties have been enforced by the usage of the HOOD method (Hierarchical Object Oriented Design) during its development, in particular the layered architecture.

On the application processor, the flight control layer (LN3) and the launchers electrical equipments management layer (LN2) constitute the Flight Program Software (FPS) that can be mission dependant (Fig.3).

The service layer (LN1) provides mission independent operating system and services environment. It handles communications with

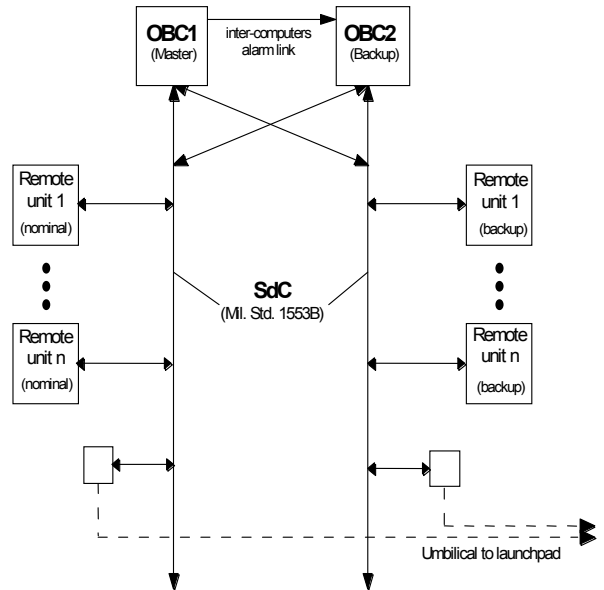


Fig. 1: Ariane 5 Electrical Communication Chains

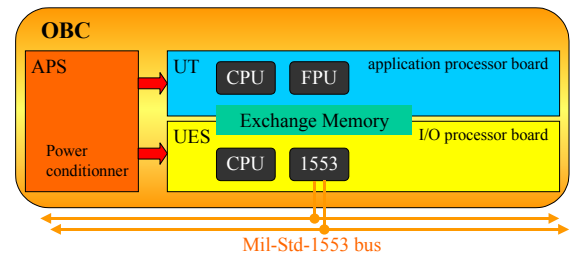


Fig. 2: Ariane 5 On-Board Computer

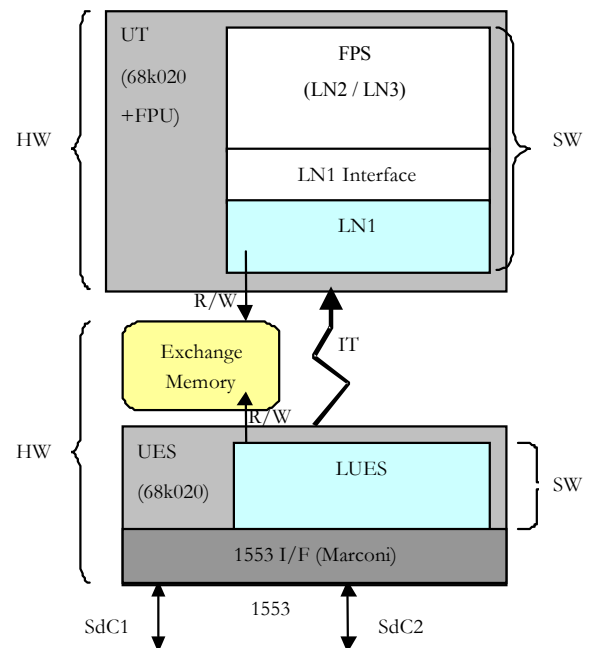


Fig. 3: Ariane 5 Data Handling Software architecture



the UES software (LUES), which manages all services protocols and devices on the I/O processor board. Shared data between application and I/O processor are stored in a structured exchange memory, which is protected against concurrent access.

The flight program ensures functions such as mission sequencing, flight control, guidance, navigation and electrical system monitoring, failure management and reconfiguration. It is mostly written in ADA and uses the Alsys ARTK® run-time and ADA environment. The LN1 is implemented as a library, which provides access to all resources of the data-handling subsystem.

On Board Computer Evolution

Rationale for HW evolution

The current launcher system has to face different goals to remain competitive. In this framework, the Ariane 5 launcher still continues to evolve, offering new services at lower costs (mainly launch of heavier payloads, at higher orbits). This includes more constraints on the different sub-systems - including avionics - that may be summed up as:

- Decrease of the costs: this has led in the OBC to mainly reduce the components and boards numbers, and to enhance the production, tests and validation tools to gain efficiency.
- Increase of the mission duration: this impacts both the avionics behaviour and global performances (provide same quality during a longer mission), as well as the radiation aspects (higher particles rates in the different atmosphere layers during longer duration).
- Management of obsolescence: to provide a better reusability of sub-systems to various missions and strategies, the new architectures shall reduce dependence on components or technologies.
- Increase the processing capability both to improve CPU load margin, which is currently very low, and to be able to offer new functions.

The current OBC shall be upgraded according to these requirements and with the constraint of risk minimisation. Studies are currently performed in order to quantify and minimise these risks but one good way to validate the architecture is to develop a mock-up and analyse the system impacts from it.

Impacts of changes on flight SW

As some of the computer parts change, it is not possible to strictly preserve the same whole flight software within a new design. At least, the lower level SW including the Board Support package (BSP) and possibly the operating system interface needs to be adapted to the new target. Within a layered SW architecture, the lower layer software can implement all necessary adaptations so as to present an unmodified interface at the upper layer level. To this aim, the following constraints on SW porting have been identified from existing design to ensure best compatibility:

- Mission data is structured in the OBC memory to be managed by both low-level and high-level software, and is an output of the mission database. To preserve these structures is mandatory.
- Low-level services interfaces shall be kept, as well in terms of syntax as in terms of minimal performances and global behaviour.





- Ground-Board communication protocol: though this is a part of the ground benches, this protocol may have many impacts on the software structure, and has to be preserved.
- Real-time structure: as a critical embedded software, the real-time tasking is a precious asset to be preserved.
- Test systems should be kept equivalents in order to achieve reusability of the validation and qualification external equipments as well as test scenarios. However, impacts cannot be avoided for tools that are machine dependent (e.g. μ P emulators).

On the static and logical level, applying these rules should allow to recompile the existing application SW and being able to run and qualify it on the new target. However, for a real-time application that has specific dynamic constraints, it probably requires special attention towards the stability of the dynamic behaviour when upgrading the HW target to another one, even more powerful.

Engineering Approach

In order to allow a system efficient use of new computer technology on Ariane, the ALF¹ Working group has been set-up by the French Space Agency (CNES). DHS specialists at system, SW and HW level form a co-engineering working group that trades-off and proposes architectural and technological evolutions with respect to the assigned objectives (costs, functions, obsolescence treatment, system compatibility). The group approach is to validate proposed designs by prototyping the solutions and eventually trading-off the consequences at the launcher's system level. The primary objectives of the new OBC baseline are:

- Replacement of its obsolescent 68020 processor;
- OBC cost to be reduced (objective 50%);
- Processing power to be enhanced (objective 400%);
- No impact on external interfaces;
- Porting effort of application SW reduced to minimum.

In response to the obsolescence of its processor and to the need for increased performance, the ERC32SC processor has been selected. This choice is compatible with the European Space Agency (ESA) microprocessor policy and allows for further technology upgrade with SW compatibility (e.g. future Leon processor and System on Chip).

For cost reduction and also as a contribution to an increased performance, a single processor board interfaced with a dedicated advanced I/O electronics has been selected instead of the current double processor architecture. Hence, the OBC I/O board (UES) including its processor and software is to be replaced by an ASIC (called IOK for I/O Kernel). The LN1 SW, responsible for handling the communication with the former UES board, needs to be modified in order to cope with the new hardware interface and to hide this new implementation to the upper software layer.

¹ The ALF working-group, created within CNES R&T activity, prepares technology evolution of launcher avionics and data-handling system. The ALF R&D working group members are CNES, EADS-ST, EADS-Astrium and Saab-Ericsson Space (supported by the Swedish National Space Board).



In order to provide an equivalent testability level, an “emulator” connector has been implemented on the evaluation Mock-up. This connection allows plugging a Microprocessor Interface Adapter (EADS-Astrium MIA product) to monitor and control the CPU activities. The MIA is the closest you can get to a classical In-Circuit Emulator, where the user has full control over the target system to be debugged.

In order to demonstrate the system compatibility of the new OBC concept and to allow evaluating the achieved performance of the integrated system, the OBC and lower level SW system has been prototyped.

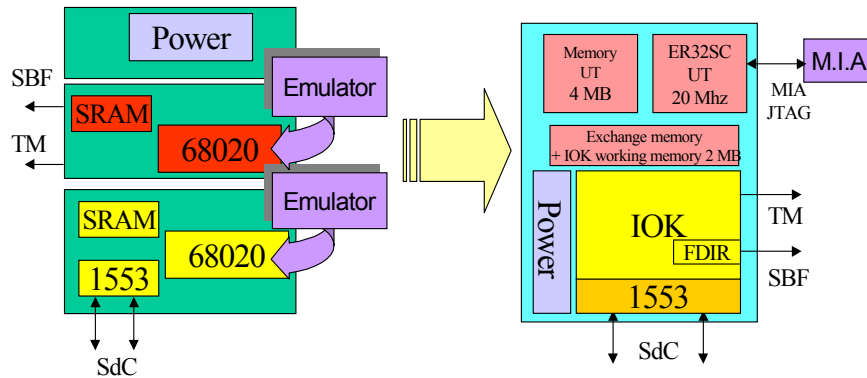


Fig. 4: From a 3-board to the new single board OBC

The OBC prototype

A prototype processor board has been designed (Fig.5) in order to represent the future target computer, the I/O Kernel being implemented in an FPGA. The I/O Kernel is actually the key technological part of the new OBC. It makes it possible to take a significant step in reducing the amount of software and hardware needed, like going from a triple-board computer to a single-board computer, while at the same time reducing mass and power consumption.

As the I/O Kernel implements all the functions earlier performed by a complete CPU-based I/O processor it is structured in a similar way, comprising the following main functional blocks (Fig.6):

- CPU interface
- Exchange Memory interface
- On-Board Time Manager
- I/O Protocol Manager
- Ariane 5 Data Structure Interpreter
- 1553 data bus protocol engine

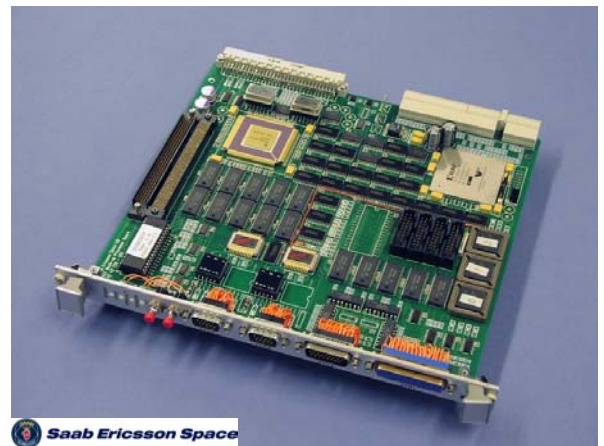


Fig. 5: The ALF ERC32 single board OBC



The specific Ariane 5 programme requirements concerning 1553 data bus management and the adaptation to the existing data structures are handled by the I/O Protocol Manager and the Data Structure Interpreter. In the I/O Protocol Manager the data bus Transfer Frames are defined and maintained.

As the 1553 Protocol Engine is a generic existing IP module, there is a need to have some kind of translation from the 1553 message data structure used in the Ariane programme to the message structure used by the 1553 module.

An additional specific support function is the On-Board Timer that is used by the Protocol Manager to trigger bus transfers at specific intervals and to date completion of certain transfers. The On-Board Timer also includes a set of programmable timers that are used for scheduling purposes by the Application Software. This represents a significant simplification to lower level SW dated functions including higher precision since such services now rely on HW clocks with no added SW counter handling.

In addition to the 1553 Protocol Engine, the CPU interface and Exchange Memory Interface modules are reused from existing SAAB-Ericsson-Space ASIC designs that are also reused, for instance, in the Herschel/Planck satellites and the Vega small launcher. This has been a significant cost saving for the ALF study and clearly shows the advantages of having basic technology, like for processor and data buses, common between many programmes.

The SW prototype

Since the concept is to hide the underlying HW changes by the lower layer software (LN1), only the LN1 formally requires re-coding and evaluation. The upper layers source code will be ported and checked as a second step.

Lower level software

There are two low level software products that have to interface to the underlying I/O kernel of the new OBC: the LSSI Init and the LN1 (Fig.7).

The PROM based LSSI Init SW is responsible for uploading the flight software into the launchers OBC's during the countdown operations. It is a critical software product for mission achievement but it is a standalone product without any upper layer software: the ground-board protocol will need to be preserved by a future OBC in order

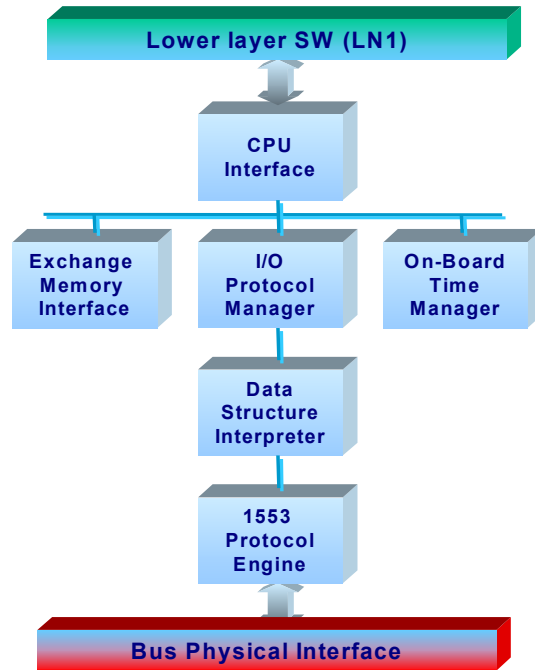


Fig. 6: The New OBC I/O kernel Architecture

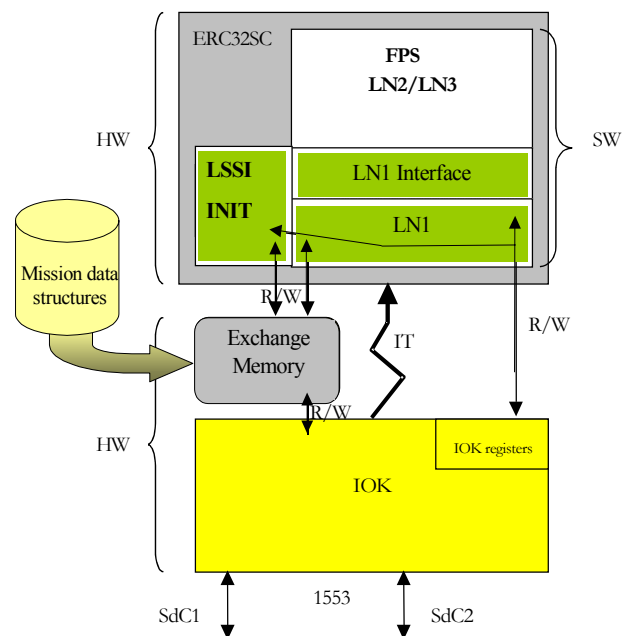


Fig. 7: The LN1 SW adapts the new HW into the system Data-handling landscape





to remain compatible with the ground support equipment, but this has not been identified as a critical feasibility issue requiring preliminary prototyping.

The SW effort has therefore been concentrated on the LN1 prototyping, which is the key item to insert the new hardware in the system with limited impact while providing significant performance increase. The LN1 provides several services to LN2 and LN3 such as time and delays management, mode control, data communications, monitoring, reconfiguration management and initialisation.

Software development approach and tools

The development has been performed in two steps.

The LN1 SW has been first quickly prototyped using COTS development and test tools and operating system (Vxworks®, Tornado® and NetRom®). This preliminary version was aimed at quickly develop the Board Support Package and set-up the first SW to HW communications with the version 1 of the IOK that implemented the SW interfaces for bus control but with limited functionalities. This environment was convenient to start and learn from the HW and eventually update the IOK specification in order to optimise for overall best performance of the LN1+IOK services.

On a second step, the LN1 has been ported on the target Ada environment and adapted to ADA I/F with a second IOK version (V2) that supports the full set of requirements. The choice on the ERC32SC ADA compilers and runtime was driven by the system compatibility requirements: the selected baseline for the ADA compiler and runtime is the AONIX T-SMART (SAE®) tool. In order to preserve the capability of the LN1 library to suspend and resume calling tasks, a patch providing the RESUME, SUSPEND and get task ID has been implemented by Alslys. Utilisation of a GNAT environment was also foreseen but not selected.

The C cross compiler, also provided by AONIX in the AdaWorld environment has also been used for LN1. An ADA interface for interface compatibility with the application layer has been provided as an independent SW package. The use of C on the lower SW layer is not visible to application.

SW Communications with the I/O Kernel

From the design standpoint, the handling of data communications and error reporting (Fig.8) has been quite simplified in comparison with the previous two-processors hardware. Indeed, the large number of control and monitoring signals that can be made available on a device like the IOK makes a direct interface with the board management software much easier. For instance, message passing from SW to IO device is handled through data queues instead of interruption as per previous design. As a result, we expect an improved performance of LN1 services by a higher factor than the pure estimated microprocessor ratio of 4 – when no task re-scheduling has to be done.



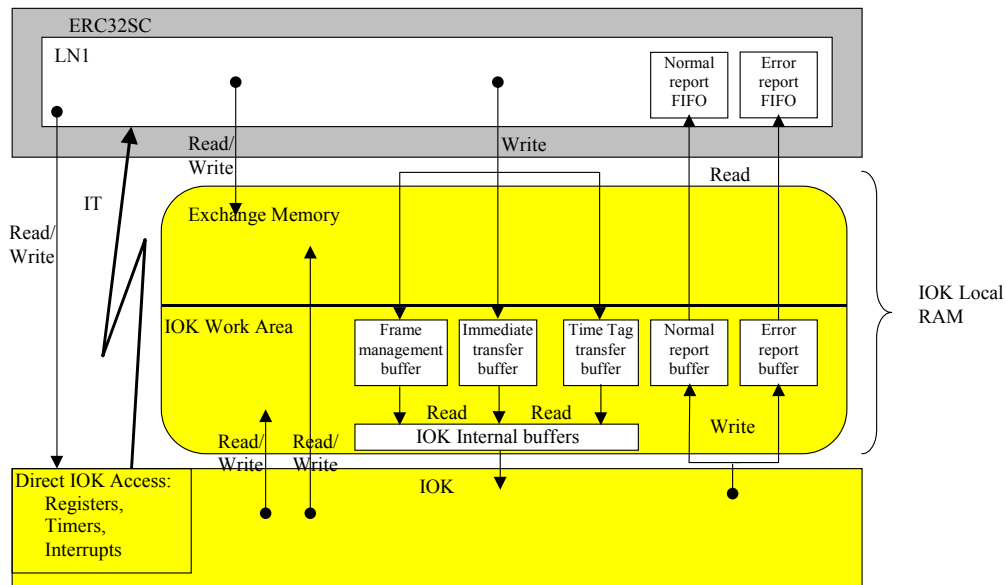


Fig. 8: SW communications with the I/O Kernel

Data structures compatibility

On the Ariane 5 program, the on-board data structures are elaborated from a database through the flight SW system generation tool (OML). This tool allows for extraction of data from the database and for generation of 68020 assembly packages. All data structures useful for the launcher mission are described in the system database and need to be stored into the exchange memory: individuals transfer, series, frames, configuration tables... These packages are loaded during launch operations through the 1553 interface.

The constraint was to keep this memory structure unmodified for data interface with the IOK for best compatibility. To this aim, a tool allowing converting the 68020-assembly code in SPARC assembly code was developed in AWK language.

To each 68020 assembly module corresponds a SPARC assembly module. Each SPARC file are compiled and can be loaded in the IOK memory and a ".h" library file includes all these definition to get them visible to the LN1 software (Fig.9).

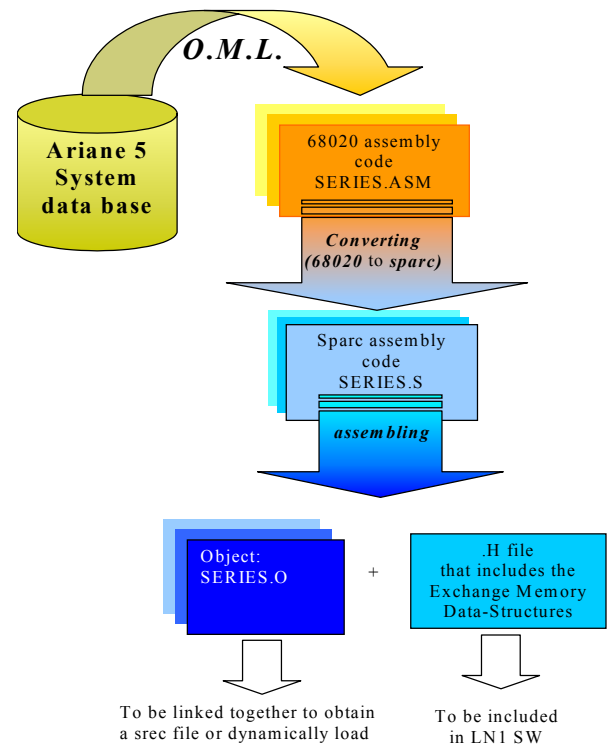


Fig. 9: Conversion of data structures into Sparc memory loadable file



Mock-Up Evaluation

Evaluation objectives

The objective of the performance evaluation is to demonstrate the validity of the ALF concept, i.e. that the developed technology brings an improvement of the LSSI (LN1+IOK) performances that enhances the CPU resources for the application layer with a stable interface. In order to achieve this, four evaluations levels have been defined:

- Characterization of the ALF Mock-up performance at LN1 service interface to application;
- Evaluation of the ALF Mock-up performance in a flight software representative situation and comparison with the current A5 DHSS (OBC+LSSI);
- Evaluation of the porting effort for application software;
- Demonstration that the Alf Mock-up is compatible with the Ground I/F.

Test environment

Performance evaluation with the current Ariane 5 OBC is performed with the Software Development Models (SDM98) that allow connection of two Lauterbach® 68020 CPU emulators (one on each CPU board) providing full non-intrusive debug and measurement access to the microprocessor instruction level. For the ERC32SC processor, an emulator does not exist off the shelf, so the ERC32 MIA based test equipment was developed in parallel with the ALF study and supports similar non-intrusive debug and test capabilities. A 1553B bus monitor and simulator has also been synchronized to the MIA test equipment thus sharing the same clock reference: precise measurements of transit time from application call to transfer on the communication bus can therefore be measured with an accuracy better than 2µs. For comparison and compatibility tests, the old and new OBC's test configurations are coupled on the communication bus (Fig.10). One computer plays the role of the Master OBC and the other one plays the role of the Backup OBC in standby/monitoring mode. This way, the OBC's are in the same configuration as on the real launcher. For these tests, the master OBC runs the so called *golden frame* that represents the traffic on the Ariane 5 communication bus at the end of the booster phase, which is considered to be one of the most critical phase from a SW application performance standpoint.

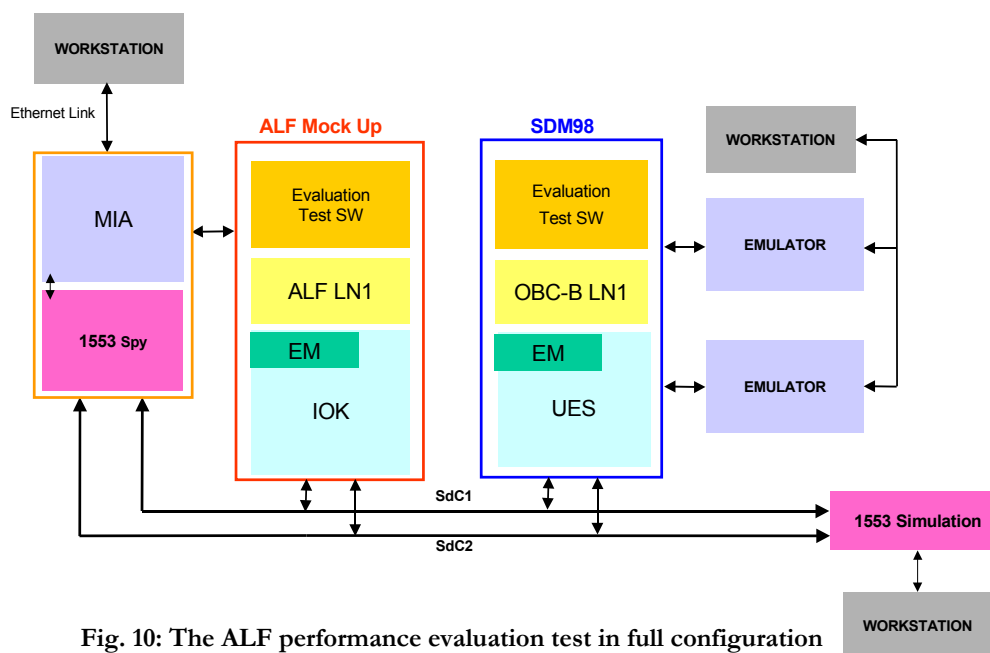


Fig. 10: The ALF performance evaluation test in full configuration



Test activity

In order to measure the performance of the LN1 services, full tracking of SW calls and data through the different layers transitions has been performed (Fig.11). This activity has provided values to the end-to-end service access time and also a very good view of the performances costs within SW layers (interfaces, ADA-run-time, re-scheduling, queues handling...) and in HW (memory accesses, bus transfers...).

For the evaluation of the ALF mock-up in a flight representative situation and the comparison with the present OBC, the application SW is derived from the robustness tests that have been already used on the present Ariane 5 OBC. It has been ported for the Sparc environment and adapted in order to reproduce the same type of access to the LN1 as the flight SW. Exchange memory accesses are added on top of this traffic in order to simulate high application data usage and to test the limits of the computer processing capability. The performances of the ALF computer can be compared with the present OBC, thus providing a good view of the enhanced processing capability of the new platform and the improved frame transfers accuracy. Switching the OBC roles allows evaluating and comparing the achieved performance in both the master and the standby/monitoring mode.

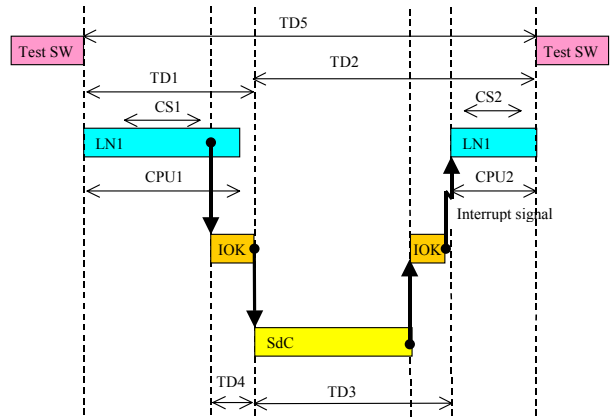


Fig. 11: LN1 services timing decomposition

Tests results

At this stage of the activities, the evaluation has not yet been completed but some preliminary results are available which show a good performance improvement with the ALF mock-up. Some typical measurements on the ALF Mock-up latency time (TD1 on Fig.11) compared with current OBC are displayed below on table 1:

Type of Measurement (from LN1 call to the first transfer on the bus)	Service duration ratio ($TD1_{Ariane\ 5\ OBC} / TD1_{ALF}$)
Start a frame of TFT's (series transfers)	10,8
Send an Immediate Transfer without data (Transmit TFI)	8,4
Send an Immediate Transfer with 16 words data (Receive TFI)	6,4

Table 1: Preliminary figures for typical performance improvement



System Level Evaluation

In the first phase, a theoretical evaluation of the flight program behaviour is performed within the ALF framework, focused on analysing the global performances of the computer from the ALF mock-up evaluation results given at the LN1 level. But the final validation of the mock-up will be done on the Flight Program Software (FPS) in real conditions. Indeed, the evaluation performed on the ALF mock-up i.e.:

- for LN1 replacement: measurement of LN1 services performance;
- for Real-time kernel replacement: measurement of tasking performance (task commutation, IT handling...), shared variables services (semaphores, timers...);
- for I/O system replacement: measurement of I/O unitary services, global I/O rate, synchronous and asynchronous messages management performances;
- for mathematical library replacement: all mathematical functions, global mathematical algorithms;
- for CPU replacement: logic algorithms, memory access algorithms and benchmarks with representative parts of the flight program code;

is not sufficient to guarantee the full portability of the operational FPS. What is currently envisioned for 2004 task is therefore to verify the correct operation of the ported FPS prototype on the ALF mock-up.

The used FPS will be a qualified generic version of the A5ECA FPS (ECA 3.2 software version) and the most stringent flight phases will be verified (ground and EAP boosters phases).

The points that will especially be under review are:

- Check-out of the SAE runtime operation with FPS tasks (ARTK is currently used);
- Verification of LN1 services operation in an operational context;
- CPU load measurement during ground and EAP phases for an operational nominal flight case.

The following tasks require to be performed with a new development environment:

- Update of LN2 and LN3 to remove 68020 assembler;
- Update LN3 to take into account the real time kernel change and, as a consequence, jitters and delays adaptation;
- Generation of an exchange memory for the ERC32 target (Sparc V7 type);
- Compilation with a new compiler of the whole FPS (over 50000 Ada code lines);
- Update of the FPS production procedures.

These tasks will assess the compatibility of the ALF mock-up with the Ariane 5 programme needs on A5ESCA or A5ESCB launchers.

Conclusion & Perspectives

At the end of the current phase, the ALF Mock-up will be ready for evaluation of the flight program software porting on this environment. The evaluated concept and I/O Kernel will also be a basis for further evolutions such as, for instance, the evolution of the microprocessor toward the Leon processor or system on chip.





The ALF working group approach provided a real and evaluated case showing that an ad-hoc strategy, applied at first HW/SW development stage, eases the evolution of a complex system with reasonable constraints on the HW and SW definition. Moreover, it showed that joint engineering efforts of customer, system, SW, test equipment and HW design actors, organized into a common working group, enforces the coherence of the overall designed system.

