Eidetic Systems Corporation

A Study and specification of a distributed fault-tolerant microprocessor system : an executive summary / T. Gomi, M. Inwood.

Government of Canada Gouvernement du Canada

Department of Communications

DOC CONTRACTOR REPORT DOC-CR-SP -82-048

DEPARTMENT OF COMMUNICATIONS - OTTAWA - CANADA

SPACE PROGRAM

TITLE: A Study And Specification Of A Distributed Fault-Tolerant Microprocessor System - An Executive Summary

AUTHOR(S): T. Gomi, M. Inwood

ISSUED BY CONTRACTOR AS REPORT NO: None

PREPARED BY: Eidetic Systems Corporation
P.O. Box 13440
Kanata, Ontario
K2K 1X7

DEPARTMENT OF SUPPLY AND SERVICES CONTRACT NO: 3ER.36100-1-0274
SN: OER81-03138

DOC SCIENTIFIC AUTHORITY: R.A. Millar

CLASSIFICATION: Unclassified

This report presents the views of the author(s). Publication of this report does not constitute DOC approval of the reports findings or conclusions. This report is available outside the department by special arrangement.

DATE: May 18, 1982

2

A STUDY AND SPECIFICATION OF A DISTRIBUTED FAULT-TOLERANT MICROPROCESSOR SYSTEM 10

AN EXECUTIVE SUMMARY

Contract No. OER81-03138

1

T. Gomi
M. Inwood
Eidetic Systems Corporation

May 18, 1982.

CONTENTS

ACRONYMS

AASC	Advanced Autonomous Spacecraft Computer
Ada	DoD defined Ada programming language
AI	Artificial Intelligence
ASM	Autonomous Spacecraft Maintenance
BA	Bus Adaptor (FTBBC/BIBB)
BC	Bus Controller (OBDH/RTU)
BIBB	Bus Interface Building Block (FTBBC)
BIU	Bus Interface Unit (AASC)
Core-BB	Core Building-Block (FTBBC)
CPDU	Control and Power Distribution Unit (OBDH)
CPU	Central Processing Unit
CTU	Central Terminal Unit (OBDH)
ESA	European Space Agency
FTBBC	Fault-Tolerant Building Block Computer
FTC	Fault-Tolerant Computing
FTCS	International Symposium on Fault-Tolerant Computing
HLM	High-level Module (FTBBC)
IIU	Input Output Interface Unit (AASC)
IOBB	Input/Output Building Block (FTBBC)
IPL	Interface Processor Link (AASC)
ITU	Intelligent Terminal Unit (AASC)
JPL	Jet Propulsion Laboratory
LAN	Local Area Network
MIBB	Memory Interface Building Block (FTBBC)
MIPS	Mega Instructions Per Second
NIU	Network Interface Unit (AASC)
OBDH	On Board Data Handling, an ESA standard for computer based on-board housekeeping methodology (ESA)
PCU	Processor Complex Unit
RAM	Random Access Memory
RBI	Remote Bus Interface (OBDH)
RTU	Remote Terminal Unit (OBDH)
SCCM	Self-Checking Computer Module (FTBBC)
TM	Terminal Module (FTBBC)
UDS	Unified Data System - a JPL designation for a standardization of fault-tolerant on-board computer system.
VLSI	Very Large Scale Integration

1. INTRODUCTION AND BACKGROUND TO THE STUDY

The early application of fault-tolerance to spacecraft was an expensive procedure. The burden it involved was only justified in deep-space missions where long-term survival was essential to the success of the project. Typically, systems were designed using well-proven technology which, by launch date, was becoming outmoded because of the lengthy development time. Thus each mission required a fresh start, with inevitable high development costs.

Many changes in spacecraft requirements have occurred since then. The economic climate no longer favours expensive ventures, launching costs are considerably higher, reusability, flexibility and increased reliability have become necessities. At the same time, missions are becoming more complex, are of longer duration, payload requirements have increased and there is a trend for the onus of control to be gradually transferred from costly and vulnerable ground facilities to on-board locations. There is now a far greater desire for fault-tolerance at all levels to support these requirements.

Concurrently, technological developments such as VLSI, are making it possible to meet the diversity of these needs and are making fault-tolerance both feasible and cost-effective. As microprocessors are more widely used, (they are, for example, now on-board all geosynchronous spacecraft), it is a natural progression to achieve self-checking by duplication of these processors. The concept of distributed systems has encouraged the development of reconfigurable, building-block architectures which led to the formation of building-block computers in the early 1970s, notably the Unified Data System/ Fault-Tolerant Building Block Computer (UDS/FTBBC) at the Jet Propulsion Laboratory and the European Space Agency's On-Board Data Handling (ESA/OBDH) system. The FTBBC, through dual processors and extensive hardware checking, is also a self-checking computer module. It was an attempt to minimize development costs and maximize reliability by using commercially available and proven elements, such as processors and memory arrays.

This study was prompted by these changing requirements and began as an examination of fault-tolerant building block concepts, with the intention of establishing the best features and ideas of these systems and augmenting them in the light of the many developments which have occurred since their inception. For example, the building-block approach has been considerably refined. Multiprocessors can now be added and removed, invisibly to the user and with no major architectural changes. Improvements in communications, such as packet switching technology, are having an impact on the efficiency and security of communication channels at all levels from global networking to internal computer data transfer.

The advent of Ada, a well-engineered, programming language designed for embedded, multitasking, realtime uses, and eminently suitable for space applications, has brought software and hardware designers together to achieve a new computer architecture. Through the enforcement of structured programming, and the use of software packages and libraries, Ada provides a high degree of data protection and a basis for resolving software fault-tolerance and reliability issues. This new architecture, based on Ada-like structures, achieves a consistency throughout which enables the placing of software protection schemes in hardware. Software and hardware integration is being further achieved by the placement of basic operating system functions within hardware, thus increasing hardware self-sufficiency.

Automated fault-handling capabilities are also being developed which allow hardware to detect, confine and diagnose faults that occur at various levels of computer hardware and execute fully automated recovery from them.

The effect of automation in space is similar to its terrestrial impact. In addition to dependability, a major benefit obtained is affordability. Areas which are prime targets for automation are the standardization of autonomous spacecraft maintenance (ASM) and automatic decision making (an Artificial Intelligence discipline), which are currently being addressed by NASA and other practitioners of space-related disciplines. An extension of the frontiers of knowledge in these areas could have a considerable impact on the cost of space utilization and an awareness of these developments has potential benefits in the long term.

2. CONTRACT OBJECTIVES

The contract objectives were:

- to review fault-tolerant spacecraft computer design concepts developed in the United States and Europe, as exemplified by the Fault-Tolerant Building Block Computer (FTBBC) developed at the Jet Propulsion Laboratory (JPL) and the On-Board Data Handling (OBDH) system developed by the European Space Agency (ESA).

- to specify a fault-tolerant computer hardware and software system suitable for use on future spacecraft missions.

- to review fault-tolerant concepts in the light of the technological developments mentioned in Section 1.

3. REPORTS PRODUCED

The following two reports were produced:

- The first report, "Review of Spacecraft Fault-Tolerant Computer Design Concepts", describes the result of an indepth study of JPL's FTBBC with emphasis on the fault-tolerant features of the design. The report also includes a brief description of the On-Board Data Handling (OBDH) System proposed by the European Space Agency (ESA), and a conceptual introduction of the Advanced Autonomous Space Computer (AASC) System proposed by Eidetic Systems Corporation. A feature by feature comparison of the above three fault-tolerant computer systems was made, along with a presentation of current issues facing further advancements of the technology. A set of design rules, the Fault-Tolerant Computing (FTC) rules, was also proposed and has been used as a tool in the evaluation of computer systems in this report.

- The second report, which is entitled "A Fault-Tolerant On-Board Computer System for Spacecraft Applications", describes in detail the design requirements for and conceptual specification of a computer system that would be suitable for a satellite application (AASC). The report discusses general hardware structure, requirements for the operating system which will consist of Ada packages, and methods for inter-subsystem communication. Operation principles of the fault-tolerant computer and interfacing requirements for application connection are also described. The report introduces the Autonomous Spacecraft Maintenance (ASM) requirements put forward by a study group sponsored by the United States Air Force as the basis for future autonomous spacecraft design, which was contributed to by both National Aviation and Space Administration (NASA) and JPL. Additional design considerations deemed important by the authors for such spacecraft are also discussed.

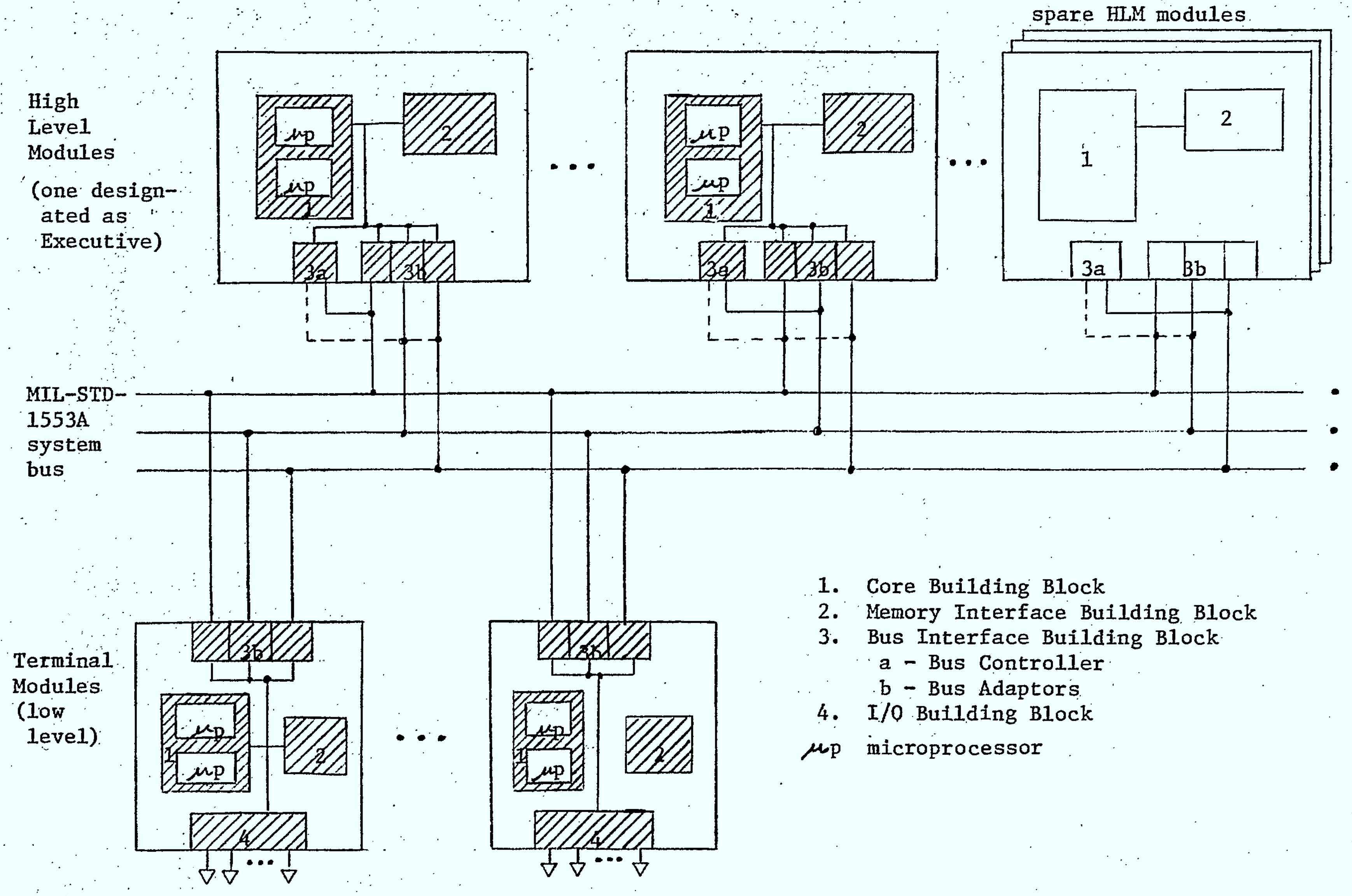
4. TECHNICAL SUMMARY

4.1 JPL's UDS and FTBBC

In 1970, JPL conducted a study entitled "Thermoelectric Outer Planet Spacecraft" (TOPS) aimed at establishing a technical basis for developing reliable spacecraft for deep space missions which are characterized by long duration, lack of maintenance opportunities, highly hostile environment, and high development and operation costs. In addition to the Pioneer and Voyager spacecraft, out of this study came a project called the Unified Data System (UDS). It defined a distributed computer architecture suitable for on-board computing which required a high degree of fault-tolerance. The project completed its system level experiments after successfully operating a proof-of-concept breadboard in 1977 using several minicomputers and a few special purpose i/o devices including a TV camera. The UDS experiment is the first implementation of a truly distributed multiprocessor system in an on-board application.

A follow-up to the UDS at JPL was the Fault-Tolerant Building Block Computer (FTBBC) project led by Dr. D. Rennels. Its objective was to implement in actual hardware what the UDS project, which was no more than a system level exercise, had identified as the Self-Checking Computer Module (SCCM). Unlike conventional mini/microcomputers, the computer module consists of building blocks with various fault-tolerant features. Figure 1 shows the building blocks, which represent a CPU complex, memory modules, bus interface modules, and i/o interface units, each of which will be implemented as a VLSI component. The intention is to allow the designer of a spacecraft computer system to select an appropriate set of these building blocks and build a self-checking computer suitable for individual needs. An SCCM thus composed may be designated as a High Level Module (HLM) or a Terminal Module (TM). An HLM will act as a processing module and as a controller to TMs belonging to one of several system buses. A TM will be involved mainly in acquiring data from sensors or sending out control signals to actuators. Redundancy is provided for both HLMs and TMs.

The strength of the FTBBC approach is, first of all, its successful introduction of building block concepts to formulate a sophisticated computer system. As noted above, the system (UDS) itself consists of these self-checking computers arranged in a highly distributed fashion, providing "building block" characteristics at a higher level. Experi-



1. Core Building Block
2. Memory Interface Building Block
3. Bus Interface Building Block
   a - Bus Controller
   b - Bus Adaptors
4. I/O Building Block

µp microprocessor

Figure 1. FTBBC - System Configuration

ence in building hardware components for an advanced computer made the JPL group highly suited to building space-qualified computer hardware. The design excels in dealing with issues associated with fault detection and fault containment.

On the other hand, there appears to be an over emphasis on hardware design and insufficient on defining the software functionality of the computer. Throughout the project, a general lack of concern for a properly designed top-down software structure is obvious. The choice of MIL-STD-1553A bus, though almost unavoidable because of military sponsorship (U.S. Navy) and other circumstantial reasons, was inappropriate in that it eventually became a source of several system level constraints that affected the upper level system design. For example, the sets of HLM-bus-TMs created because of the 1553 bus architecture are judged to contribute to the lack of dynamism when planning redundancy at HLM levels. Also, there is no clear definition of what software or hardware would do once a fault is detected and contained at a lower level locale. The lack of concern for fault-diagnosis and execution of recovery procedures at higher levels of the fault hierarchy is considered to be dangerous when the satisfactory over-all performance of a computer system is important.

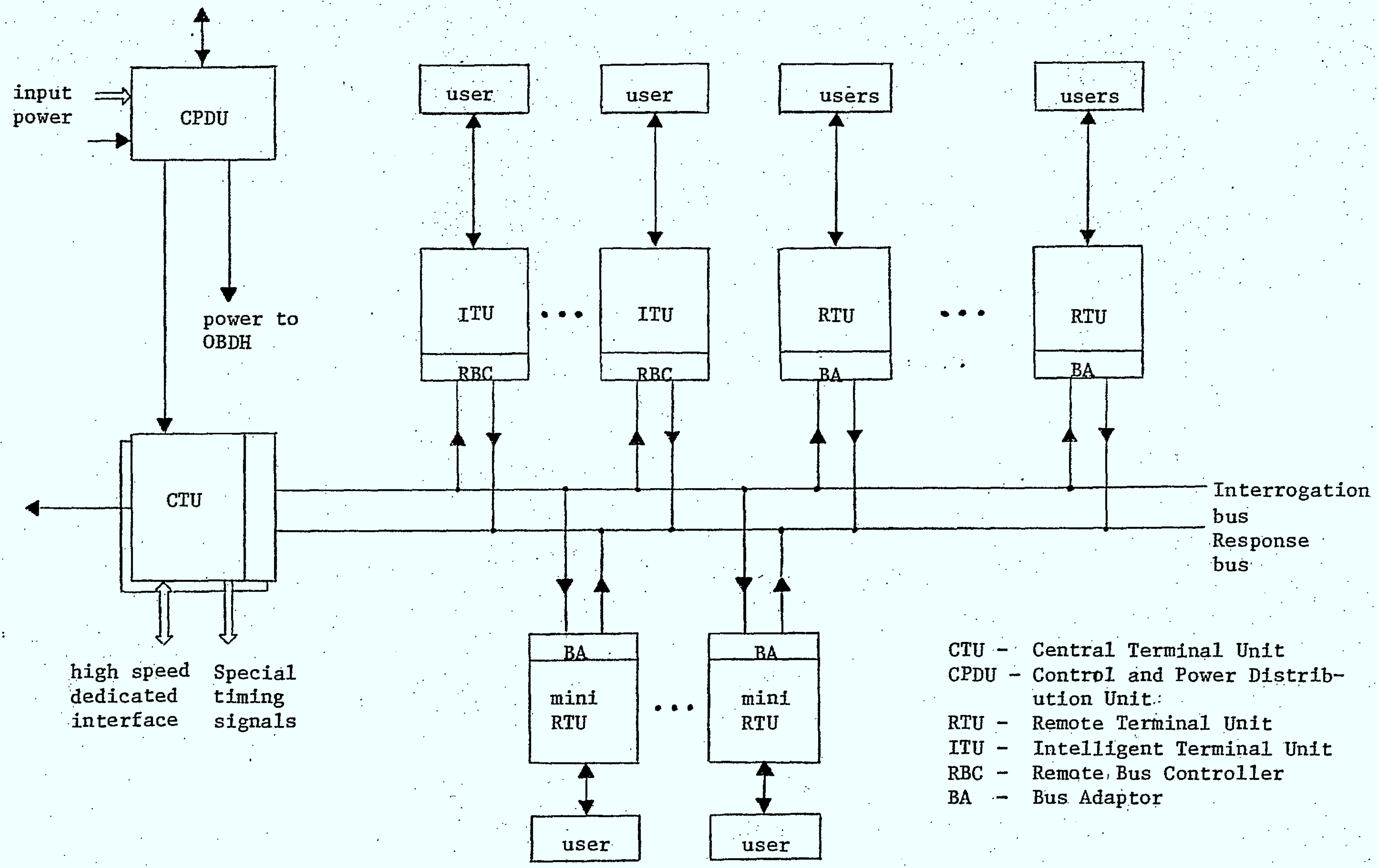
JPL is currently experiencing some difficulties in obtaining space-qualified components for all of their building blocks. Financial support for the project by NASA and the Navy is almost coming to an end in the general atmosphere of drastic spending cuts in the area of space R & D. The United States Air Force is showing some interest in the program and is expected to offer support for the continuation of the development. The project is now moving from breadboarding to VLSI die design. The onboard computing facility for the Galileo Spacecraft is said to be a version of the UDS architecture, though it has not adopted the FTBBC as its element computer.

4.2 ESA's OBDH System

The On-Board Data Handling System is a product of the European Space Agency's Technical Research Programme. It has been designed as a "re-usable" on-board computer for unmanned satellites. It is now, after some modifications which include the use of the Ferranti F100L processor, operating as the L-Sat on-board computer and is the only one of the three computers reviewed here which is actually in use. The ESA project involved the political, technological and economic consensus of all the participating countries and has proved to be a well-coordinated, goal-oriented project. The design of the project was well defined by 1973 and covered a computer system, plus a set of standards. The shared development has had several benefits. It has enabled the expense load to be spread, an expertise base to be formed and the system to be completed quickly. It has also forced some decisions to be taken at an early stage, such as those related to standards and the distributed, modular nature of the system.

Figure 2 shows the standardized units which comprise the system. The Control and Power Distribution Module (CPDU) acts as an executive module; the Central Terminal Unit (CTU) is a high level module; there are three varieties of Remote Terminal Unit (RTUs) to accommodate the differing needs of users; and an ESA standard bus. The CTU contains some building block options for users. The bus system allows a reasonable degree of distribution and provides some broadcasting facilities between units on the system. Some internal redundancy is provided within the CPDU, the RTUs and also the bus system but the CTU relies entirely on a standby spare for protection. A few software detection features are incorporated, such as watchdog timers and terminal status monitors. The system can be configured to differing mission requirements.

The OBDH has some of the same drawbacks as the FTBBC. Because of its age, it does not incorporate any of the recent developments in fault-tolerant techniques and indeed, does not have the same degree of hardware fault-detection as the FTBBC. The technology used in it is now becoming obsolete. It uses little VLSI and none is yet planned. As in the FTBBC, design has concentrated heavily on hardware. However, it makes no provision for inflight reconfiguration. The system relies entirely on ground control and although the proposed use of "packetized telemetry" will be a move towards on-board processing, its low throughput is an impediment to the transfer of fault-tolerant control on board the spacecraft. The operating system is ESA designed. It does not allow multitasking and is not distributed. There are no apparent software protection schemes and no visible software hierarchy, which implies difficulty in integrating software fault-tolerance into the system. When checked against the FTC rules, several critical violations become apparent. Not-



CTU - Central Terminal Unit
CPDU - Control and Power Distribution Unit.
RTU - Remote Terminal Unit
ITU - Intelligent Terminal Unit
RBC - Remote Bus Controller
BA - Bus Adaptor

Figure 2. OBDH - System Configuration

ably, the CPDU forms a single point of failure, CTU failure could cause local paralysis and, as a fixed-fault arbiter, there is no check on its well-being.

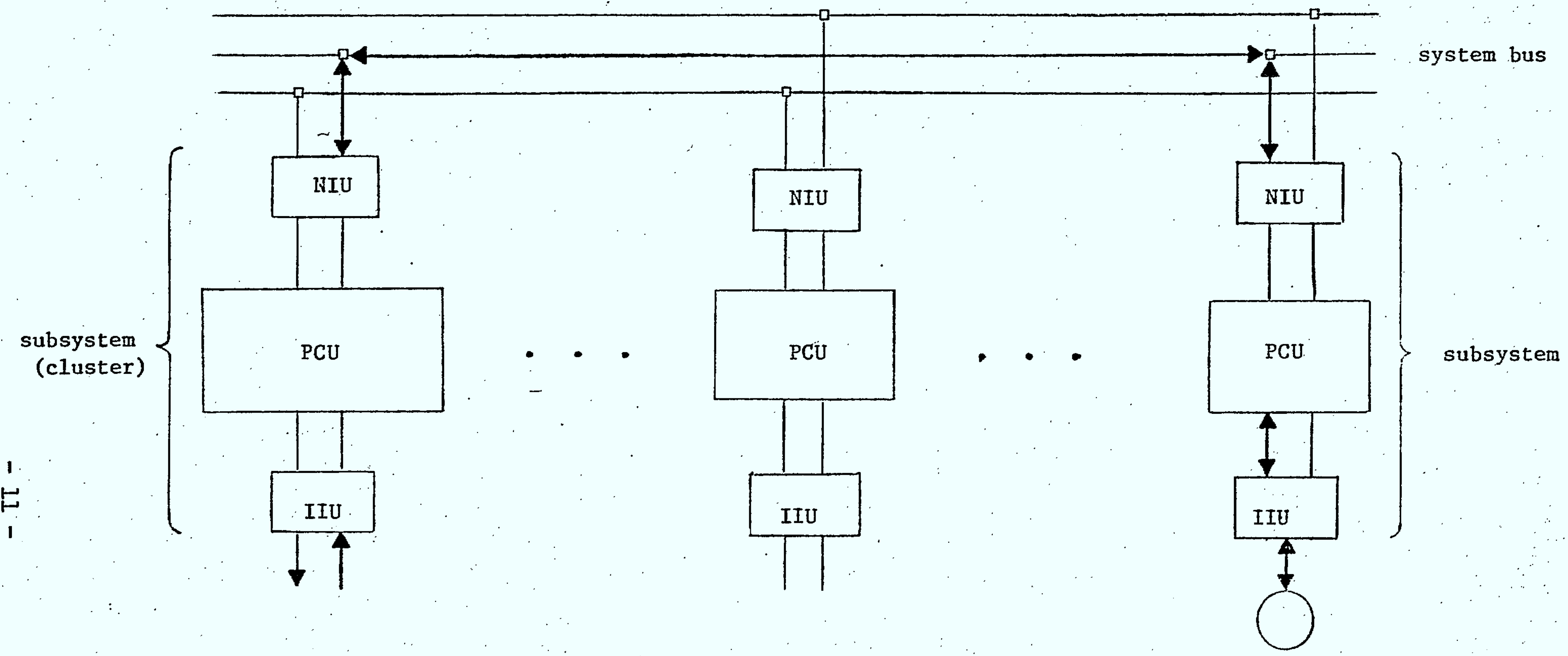
Development of the OBDH is being continued. The modifications necessary for its incorporation into L-Sat have pinpointed the need for changes which are currently under way. A joint NASA/ESA workshop is considering the standardization of data formats, which involves the introduction of "packetized telemetry". The bus is also being improved, involving the introduction of processor interrupts and some local area network features.

4.3 Proposed AASC

In 1981, the Communication Research Centre of the Department of Communication began a series of studies to review microprocessor-based fault-tolerant technologies in use in space applications. The Advanced Autonomous Space Computer (AASC) concept emerged following this effort. It is a conceptual definition of an on-board computing facility that is adaptable to future satellite applications. At this stage the system exists only as a concept. Two separate evaluations of its capabilities and performance characteristics are being undertaken to ensure its readiness for implementation.

As shown in Figure 3, the AASC consists of a set of system buses linking computer clusters. Each cluster would house a spacecraft subsystem (such as AOCS, power management, telemetry and command, and heat management) and provide a highly dependable hardware and software environment for its operation. Figure 4 shows a cluster with its three subunits: Network Interface Unit (NIU), Processor Control Unit (PCU), and Input-Output Interface Unit (IIU).

The design objectives of the AASC include flexibility, expandability, and reliability. The structure may be modified to suit a mission by selecting the number and placement of clusters. The cluster itself may be adjusted in its size (throughput), i/o capability, or level of fault-tolerance by choosing appropriate building blocks within subunits. The open-ended design of the system bus would meet any foreseeable expansion of requirements in future applications. Mission to mission adaptability of the AASC is judged to be excellent.

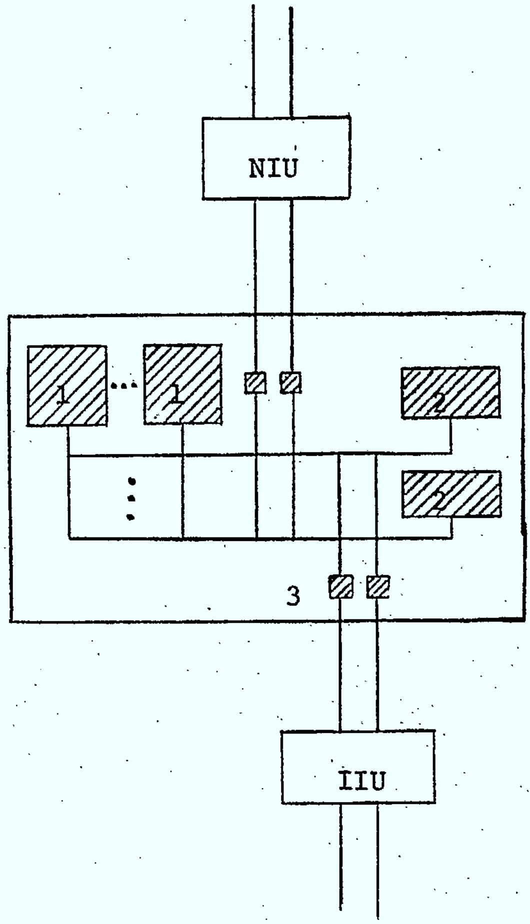


NIU - Network Interface Unit

PCU - Processor Complex Unit

IIU - I/O Interface Unit

Figure 3. AASC - System Configuration



1. Processor module
2. Memory module
3. Interface processor link

Figure 4. AASC - PCU detail

Fault-tolerance within the AASC exists in a hierarchical fashion. At the top, the system bus structure has its own fault-tolerant features. Clusters are supported by spare clusters that are based on the principles of non-dedicated redundancy. Spare computer and memory components are used in a way similar to but more generalized than in the FTBBC. Each VLSI component of the cluster possesses similar redundancy features and extensive fault-handling capabilities to automatically detect, confine and recover from hardware errors independent of the software. More complex faults are handled by software. Thus, total compliance to the FTC Design Rules is possible throughout the system.

The AASC hardware constitutes a typical Ada execution environment. All software written for the AASC will be implemented using that language. This assures a high degree of software portability, adding to the mission to mission adaptability of the system. Software developed for a particular mission will be catalogued using Ada's package and library concepts, and will be made available for subsequent flights or missions. Thus, as development proceeds, the program library will become a major software resource, to be tapped by future software designers in formulating software for new missions. This will significantly cut software development costs, which are rapidly becoming a sizeable portion of the total expense.

AASC software development will also utilize another feature of the Ada language designed to improve the quality of embedded real-time programming. The strict enforcement of structured programming, including the precise definition of data types before use, will make the process of software design, implementation and testing more efficient, and provide an effective monitoring tool for the administration of the development project.

The AASC architecture adopted a standard networking principle widely used in other computer system applications with similar communication needs. Achievements made in this active technological area will be utilized in on-board applications. On-board inter-cluster communication has all the aspects of a typical distributed processing application, which has extremely high reliability requirements and wide traffic variance, such as computer systems used in banking applications. This approach will result in the use of a proven technology, with its development costs long written off, and with high availability of optimized components and techniques. Again, contributions to over-all cost reduction should be substantial.

The utilization of commercial technological developments in space rather than custom-built components was an implied objective of the FTBBC project. The AASC expands this principle to cover all of its hardware, and most of its software. The operating system, for example, is commercially available for sophisticated industrial applications, a use of ready made products enabled by recent design developments. The only difference would be the extensions and optimization effort needed to fit the standard version to the AASC.

The microprocessor adopted for the AASC has a fault-tolerant feature especially designed for operations requiring extreme levels of reliability, such as space missions, deep sea explorations, and nuclear reactor monitoring and control. The processor also has sophisticated program and data protection schemes enforced in VLSI which are outlined in the Ada language definition. These reliability features enable the development of a space-borne computer system that is self-sufficient and susceptible only to outside forces of destruction, such as meteoric collision.

While the AASC concept is undergoing evaluation, its major elements such as: operating system; VLSI components including microprocessor, memory controller, bus controller, and i/o controller; communication protocols; and network components; are being developed by manufacturers and the general computer community. A sizable investment is being made by the U.S. government and industry to space-qualify a large number of hardware components including those being considered for the AASC.

4.4 Comparison of the UDS/FTBBC, OBDH and AASC Systems

The art of fault-tolerant computing (FTC) was immature when the FTBBC and OBDH systems were conceived. The FTBBC excels in the detection and confinement of faults but has almost no facilities for diagnosis and recovery. The OBDH system is similarly limited in its FTC capabilities. Fault-tolerance in the AASC hardware is an integral part of its design and is founded on knowledge accumulated during the past several years. Software fault-tolerance in the AASC is based on the advanced control theory being developed in similar applications demanding high reliability.

To achieve autonomy in satellite maintenance, the FTBBC would require a considerable degree of alteration to its hardware design. The OBDH system, which was designed around a tight ground control scheme, has little or no such potential. The AASC, with its advanced hardware autonomy already in place, needs only the addition of proper software functions. Its abundant CPU power and memory space will be instrumental in supporting the sophisticated algorithms needed to achieve a satisfactory level of autonomy.

The six FTC design rules discovered during the study are frequently violated by both the FTBBC and OBDH systems. Weaknesses in the OBDH system are more severe, such as the inclusion of a single point of failure in the design. The conceptual design of the AASC has so far successfully avoided infringements. While these design rules are still subject to the test of validity, most of them have been adopted among designers and researchers as empirical factors that affect the stability of a complex computer system.

The size variance from mission to mission will affect performance of the FTBBC, in particular at the higher end. The microelectronics technology used in the OBDH system is becoming partially obsolete. Both systems may face problems in adapting to future missions that are larger and more sophisticated. Having benefited from this experience and the recent leap forward in the progress of VLSI technology, the AASC design has clearly avoided these restrictions.

As the OBDH system is already operational, the space qualification problem has obviously been solved for that system. The FTBBC is still in search of some space-hardened components. The majority of the components considered for the AASC are very new and hence, are not yet space-qualified. However, a major effort is underway to create space-qualified versions of these components because of their recognized suitability for future space applications.

In both the FTBBC and OBDH systems, processors are arranged to provide execution vehicles for dedicated satellite subsystem functions. Any portability that might exist in either system between application processes and processors is very limited. The flexibility of the AASC is best demonstrated by the arrangement of processors in the system. They are pooled to provide the appearance of a non-dedicated computer resource. Assignment of an application process to a particular processor takes place only at execution time. The separation of application needs and system capability provides the opportunity to engineer the system (processors) independent of throughput and fault-tolerance modes. The AASC has a wide range of adjustment available for these requirements.

The MIL-STD-1553A bus used in the FTBBC design has some serious deficiencies including one which results in a violation of the FTC design rules. It also restricts the performance and expandibility of the system. The system bus used in the OBDH system is well-engineered but its design objectives are short-sighted. With an increase in application size it will soon run into capacity and fault-tolerance performance problems. The local area network architecture of the AASC is adapted from similar information processing requirements seen elsewhere and reflects a natural progression towards a type of on-board processing expected in future satellites and other types of space-borne structures. This technology has, and will continue to be, the subject of much study. Resources available through development in this area will certainly be beneficial to spacecraft designers.

The FTBBC and OBDH systems both have custom-made operating systems. They are both simple in structure and limited in capability. To support the highly sophisticated process structure of future on-board applications, the AASC adopted a commercial real-time operating system. This also obviates the necessity for up to 200 man-years of engineering effort required for the development of such a mature system.

The superiority of modern structured high-level languages over their low-level counterparts is well established in industry. In recognition of this, Ada has been adopted as the design, as well as implementation and documentation language for the AASC. With the drastic improvement in capability and decrease in the cost of computer hardware, the overhead issue (memory space, execution speed) which was the main justification for avoiding modern languages is now a minor concern. Software engineering has high-lighted other important factors affecting software development which can be dealt with effectively in an Ada environment.

A good on-board computer system must provide most of the services demanded by applications. It must also place the minimum of arbitrary restrictions on the type or form of application. At the same time, however, it should enforce discipline on the application design. The FTBBC imposes some serious structural restrictions on applications. The OBDH system, whilst well-engineered, gives little more freedom in the interface between subsystems and the computer. The process-oriented interface which the AASC provides for applications, permits the design of a proper application process hierarchy independent from the hardware and software reality of the system. On the other hand, the AASC demands a strict compliance to structured programming principles and system defined inter-process protocol. In return, inter-module protection is enforced by the hardware without overheads.

The OBDH devised its own test procedure, both for hardware and software. The FTBBC does not yet have one. With the use of Ada and structured programming, the AASC has future possibilities in the automation of software validation. As the reliability of VLSI components is rapidly increasing and the level of integration constantly being raised, the need for reliability has moved to a higher level. Issues such as software validation are becoming of more importance in the testing of future on-board computer systems. this aspect will become the most important issue in testing future on-board computer systems.

Table 1: COMPARISON OF FAULT-TOLERANT ON-BOARD COMPUTER SYSTEMS (SUMMARY)

FEATURES	UDS/FTBBC	ESA-OBDH	AASC
Fault Tolerant features	good	few	very good
ASM potential	some	low	high
FTC rules - failings	some	critical	none
Mission adaptability	average	average	good
Space qualification	some problems	in effect	by 1986
Processor	TI9900 or MC68000	Harris 6100	iAPX 432
- capacity	1.2 MIPS approx.	0.8 MIPS approx.	variable
- multiprocessing	yes	yes	automatic
System bus	MIL-STD-1553A	ESA standard	iAPX packet bus and LAN
- multi-terminal communication (broadcasting)	none	some	yes
Operating system			
- design	FTBBC	OBDH	standard
- concurrent execution	limited	none	good
Programming language	UDS Design Language and assembler	Assembler and FORTRAN	Ada
Application interface			
- processor independence	average	average	high
- structured programming	possible	design phase only	enforced
- distributed processing	some	none	good
- hierarchical structure	limited	none visible	flexible
- inter-module protection	poor	poor	very good
System testing quality	average	average to high	high
Software validation	difficult	very difficult	future possibility

5. CONTACTS ESTABLISHED

Several important contacts with authorities at the forefront of fault-tolerance and its associated disciplines were established, both in industry and academic circles. These are listed below:

Dr. David A. Rennels was interviewed at the Jet Propulsion Laboratories. Dr. Rennels heads the team which built the FTBBC and is the General Chairman of the 1982 International Symposium on Fault-Tolerant Computing Systems (FTCS-12). He discussed past and future trends in fault-tolerance. He was also questioned about the development of the FTBBC and the basis on which decisions were made during that process.

The Program Chairman for FTCS-12 is Dr. George C. Gilley, of the Aerospace Corporation. Dr. Gilley was in charge of the UDS project, the forerunner of the FTBBC, and was a leading member of the NASA/USAF study group on ASM in which Dr. Rennels also took part. During an interview in Los Angeles, Dr. Gilley contributed much useful background information on ASM.

Dr. J.S. Albus and his team at the National Bureau of Standards, Washington, DC. Dr. Albus' laboratory was visited, his work on an advanced control theory was discussed, and a demonstration of its application was observed.

Stefan Ciarrocca and Peter Dubock of the European Space Research and Technology Centre (ESTEC), Noordwijk, Holland, who provided information on the OBDH system.

Researchers in the theory of fault-tolerance have also been contacted. Their work has been studied and a continuing interest is being taken in developments arising therefrom. These contacts and their fields of study include:

- J. Kuhl and S. Reddy, of the University of Iowa;
  a proposed diagnosis model for a fully distributed system

- J. McPherson, University of Wisconsin, Madison;
  a set of theorems to describe system correctness in the presence of faults

- J. Black, University of Waterloo;

a study of data structure robustness

- W.G. Wood, University of Newcastle-upon-Tyne;
  aids to the recovery process in a distributed system

- C.L. Kan and S. Toida, University of Waterloo;
  the hierarchical nature of faults and fault-tolerance within a system

- P. Azema, Laboratoire d'Automatique et d'Analyse des Systemes du C.N.R.S. of Toulouse;
  application of computer networking technology in establishing system level fault-tolerance

- D. Pradhan, Oakland University of Rochester, Michigan, and S. Reddy, University of Iowa;
  a topology to permit efficient routing and distributed fault-diagnosis.

6. CONTRACT CONCLUSIONS

A concensus of opinion among leading authorities in the field maintains that future demands for on-board processing will increase and will require a greater degree of flexibility, reliability, expandability, throughput, and fault-tolerance.

Three fault-tolerant computers have been compared using these points of reference. In spite of their building-block architecture, the results show major problems in the upgrading of both the FTBBC and the OBDH systems to an acceptable level. Notwithstanding its conceptual status, the AASC, based on its rating in the comparison, shows a robustness of design, adaptability of architecture and resilience in fault situations which would enable it to meet these needs.

It is the recommendation of many influential members of the fault-tolerant community that the answer to problems facing the future use of spacecraft such as economy, complexity and security, lies in the development of on-board control and an implied use of advanced control theory, automated decision making and higher levels of machine intelligence. It would, therefore, be prudent to ensure that current developments in fault-tolerant on-board computing not only meet immediate needs but contain the framework on which long-term solutions may be built. An on-board computer should not be precluded by its nature, as in the present case of the FTBBC and OBDH systems, from utilizing such advances. In the opinion of Eidetic Systems Corporation, the AASC would be compatible with these future possibilities.

7. RECOMMENDATIONS FOR FOLLOW-ON WORK

In order to verify the AASC concept and obtain hands-on experience with the new computer architecture which is proposed in it, the following phased theoretical and practical activities are recommended. Such actions are expected to be carried out step-by-step, the results from each step being fed into the next stage in the design cycle.

Step 1. Concept Review.

An elaboration should be made of the features contained in the design of the AASC, and related technologies, in respect of the anticipated requirements for future spacecraft.

Step 2. Conceptual Breadboard Development and Tests.

A flexible breadboard system should be designed and built to test the basic concepts of the AASC. This should consist of at least two computer stations linked by a local area network. The aim would be to test algorithms within the AASC, but not the AASC hardware, through the use of software simulation.

Step 3. Refined Breadboard Development and Tests.

The NIU and IIU should be developed, integrated into the breadboard developed during Step 2, and tested. The IIU would be tested using a real or simulated spacecraft i/o.

Step 4. Full Breadboard Development.

The PCU hardware should be developed using various sizes and combinations. The number of stations on the breadboard system should be increased. The software controlling the PCU should be installed.

Step 5. Full Breadboard Experiment.

A detailed test plan should be produced based on the previous results. It should include testing PCU characteristics and should use the full breadboard system.

Step 6. Recommendations for Prototype Design.

Following Step 5, any necessary revisions should take place and recommendations be listed for developing a prototype AASC.

GOMI, T.
A study and specification of a distributed fault-tolerant microprocessor system....