The Fault Tolerant Computer System of The Brazilian Scientific Application Microsatellites

Alderico Rodrigues de Paula Jr.(1) -
Claudio Roland Sonnenburg (2)
Instituto de Pesquisas e Desenvolvimento - IP&D
Universidade do Vale do Paraíba - UNIVAP
AV. Shishima Hifume 2.911
12244-000 São José dos Campos, SP , Brazil
ipd@univap.br

ABSTRACT

A fault tolerant computer system has been conceived to become the standard framework that will be utilized by the future family of Brazilian small satellites for scientific applications. Based on the proposed standard, a computer system with three processing modules was developed for the First Brazilian Scientific Application Microsatellite (SACI-1 - Satélite Científico). Each processing module is based on a 32- bit Transputer that is connected to the two other processing modules, as well as, to the satellite subsystems through a 10 Mbit serial bus. A set of fault handling mechanisms is implemented in the computer system which enable it to tolerate a single fault and most of the double faults. Four switches controlled by a watch-dog-timer are used in each processing module to allow the serial links to bypass the processor when a catastrophic fault is detected in that module. This technique allows the computer system to degrade gracefully to two processing modules, or even to one processing module. The computer system is designed to execute all on board tasks with three or two processing modules and the essential tasks with one processing module. A large mass memory that uses memory chips not qualified for space was implemented in each processing module. Since these memory chips are susceptible to cosmic radiation, a special circuit was used to detect latchups and to power off the memory chip for 10 msec when a latchup is detected. Additionally, these memory chips present a high single event upset (SEU) rate, when in orbit, producing single, double and, less frequently, triples errors in a memory word. To overcome this problem, the scientific experiment data frames are coded before being stored in the mass memory using an error correction code. The on board software was organized around a set of processes that communicate among themselves through a routing process. The essential software was formally specified and its correctness certified using the CSP-Z formalism.

----

(1) PhD in Computer Science - UCLA 1982
Professor and Researcher in Computer Science at Universidade do Vale do Paraíba
E-mail: alderico@univap.br
(2) PhD in Computer Engineering - University of Michigan 1974
Professor and Researcher in Computer Science at Universidade do Vale do Paraíba
E-Mail: sonnen@univap.br

I. Introduction

The Brazilian National Institute for Space Research (INPE - Instituto Nacional de Pesquisas Espaciais) is responsible for the development of a family of microsatellites for scientific applications. In order to minimize cost and assembly time, it was decided that a modular microsatellite bus, that can easily be adapted for different mission requirements, should be developed. The development of the first satellite (SACI-1 - Primeiro Satélite Científico Brasileiro) started in 1994 in cooperation with some Brazilian Universities.

Four experiments were selected for the first mission [NER95a]:

a) Airglow Photometer: This experiment has the objective of measuring the intensity of the terrestrial airglow emission, in global ranges, of oxygen OI 557.7 nm , OI 630 nm and OH.

b) Plasma Bubble Experiments: The main objective of this experiment is to study plasma bubbles in the Earth ionosphere, investigating their generation, development and decay, especially over the Brazilian and South Atlantic region.

c) Solar and Anomalous Cosmic Rays Observation in Magnetosphere (ORCAS): This experiment aims to investigate the anomalous cosmic radiation flux, measuring the flux, its composition and the spatial and temporal variation of the ions from He to Ne and protons and electrons with energy up to 50 MeV.

d) Geomagnetic Experiment (Magnex): The objective of this experiment is to investigate the phenomena related to the current aligned with the trans-equatorial magnetic field and the plasma electrodynamics that involves the Earth, specially in the region of the South Atlantic Anomaly.

The experiment boxes are assembled with the microsatellite bus in 400 x 400 x 600 mm parallelepiped structure. The bus is a sandwiched type structure that holds the power supply, telecommunication and computer subsystems. The microsatellite is spin stabilized. Its total mass is 60 KG and the on board equipment power consumption is 30W on average.

II. Standard Computer System Architecture

One of the objectives of the SACI program is the development of a microsatellite bus that could easily be adapted for different scientific application missions. Based on this premise, a standard modular computer system that could satisfy the following requirements was developed [DEP95]:

a) The computer system should be a fault tolerant distributed system able to survive any single hardware fault without degradation and also survive most double hardware faults. When a second fault occurs, the computer system should be able to execute, at least, the critical tasks. The computer reliability should be better than 0.98 for a two year mission.

b) The interface with the satellite subsystems should be completely redundant. Any hardware fault in one of the interfaces should not affect the communication between the computer and the satellite subsystems.

c) The computer system should have a large mass memory, capable of storing the data collected from the experiments when the satellite is not in contact with a ground station. At any given time, less than 0.01% of the data frame stored in the mass memory can be affected by unrecoverable errors.

d) The computer system should be easily adapted for different mission performance requirements without a complete redesign.

e) The computer system should be compact and should have a small power consumption.

f) The computer system components should be able to support a cosmic radiation total dose of up to 20 Krads for a two year mission and all components should be latchup free for radiation particles up to 50 Mev or should have a circuit to recover from latchups.

Different computer architectures were analyzed for the scientific microsatellite computer system. The selected one was the modified meshed architecture. Each processing module should be able to communicate with four other processing modules or with the satellite subsystems interface through a high speed serial link. All interfaces should be connected to two different processing modules as presented in Figure 1.

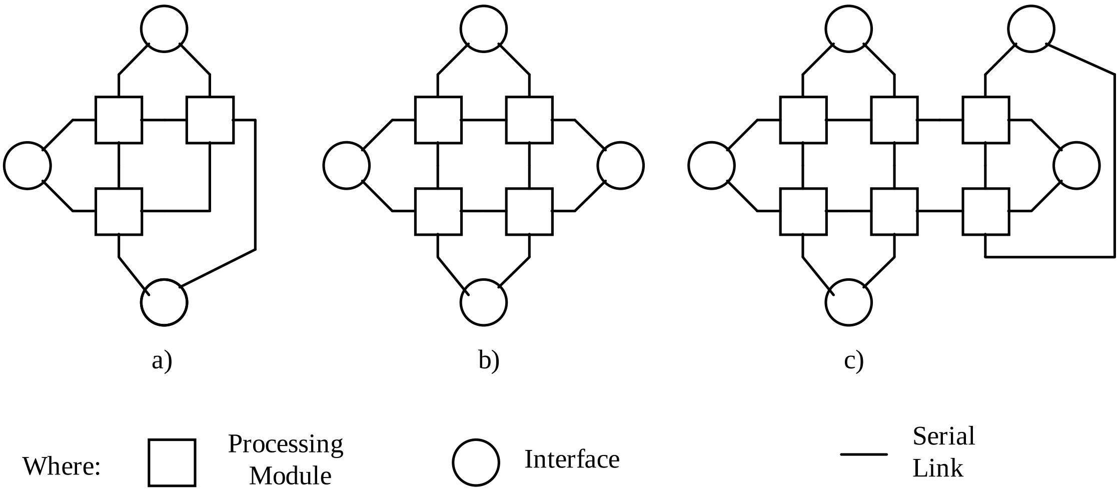
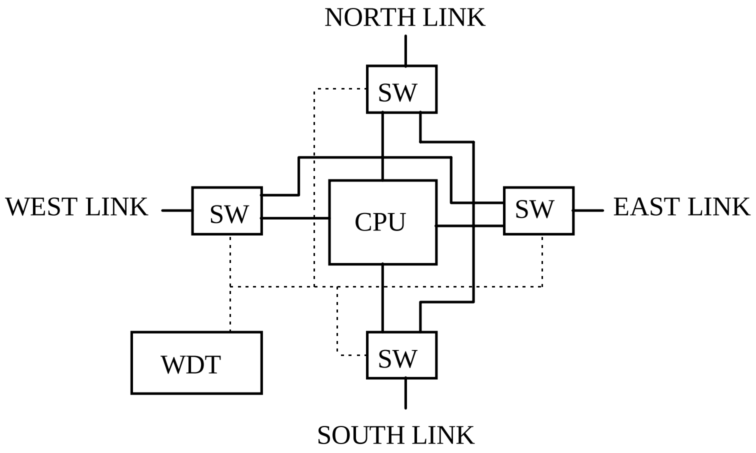


Figure 1 Modified mesh architectures. a) Three processing modules and three interfaces b) Four processing modules and four interfaces c) Six processing modules and five interfaces

To allow graceful degradation [CAS93], the meshed architecture was modified with the addition of a watch-dog-timer (WDT) to each processing module and a set of switches used to disconnect the processing module from the mesh and to connect the north link to the south link and the east link to the west link when a catastrophic fault is detected. This fault is detected by the WDT when it is not reset during a predefined time interval. The switching scheme is presented in Figure 2.

The modified mesh architecture satisfies the specified requirements when tree or more processing modules are utilized. It can degrade gracefully and it can be easily adapted for different mission performance and reliability requirements by adding additional processing module.



WHERE: SW = SWITCH; CPU = CENTRAL PROCESSING UNIT; WDT = WATCH-DOG-TIMER

Figure 2. Switching scheme of a processing module

III. The Computer Architecture of the First Brazilian Microsatellite

The following tasks were assigned to the onboard computer [DEP95a]:

a) Reception of the telecommands from the Control Center. The onboard computer should decode the telecommands and, then, generate the commands to the satellite subsystems and experiments.

b) Acquisition of the telemetry signals from the satellite subsystems and the data from the experiments. The onboard computer should format acquired data in frames, store the frames in the memory and send the frames when the satellite is in contact with a ground station.

c) Onboard processing of the data requested by the experiments.

An analysis was done at the beginning of the project to select off-the-shelf microprocessors that satisfied the radiation requirements and that were suitable to be used in the proposed architecture [DEP95b]. As a result of this analysis, the Transputer T805, manufactured by INMOS, was selected. This T805 has an internal floating point unit and it can access directly an external memory of up to one Gbytes. Additionally, it has four serial links that can communicate at up to 20 Mbits/sec simultaneously with the Transputer processing unit. The Transputer has a 2Kbytes internal memory. As this memory is sensitive to single event upsets, it could not be used onboard and it was consequently deactivated.

The performance analysis [DEP95b] demonstrated that, in order for the computer system to execute all specified onboard tasks and to store all acquired data, two processing modules each having a program memory of 128Kbytes and a mass memory of 8 Mbytes were necessary. The dependability analysis demonstrated that the reliability of each processing module for a two years mission was 0.98. However, the reliability of the computer system, using two processing modules, was 0.96, which was less than what had been specified for the computer system. To satisfy the reliability requirements, three processing modules were necessary, resulting in a reliability of 0.998 for a two year mission.

The designed onboard computer is composed of three processing modules based on the INMOS Transputer T805 and three internally redundant interfaces (SRI, UAC and TCTM) that are interconnected by a 10 Mbps serial line. The SRI is an interface with the experiments that exchanges messages in serial form. The UAC is used to generate the commands to the satellite subsystems and also to acquire the telemetry signals. Finally, the TCTM receives the telecommand frames from the Control Center and sends the telemetry frames to the ground stations. The block diagram of the onboard computer is shown in Figure 3

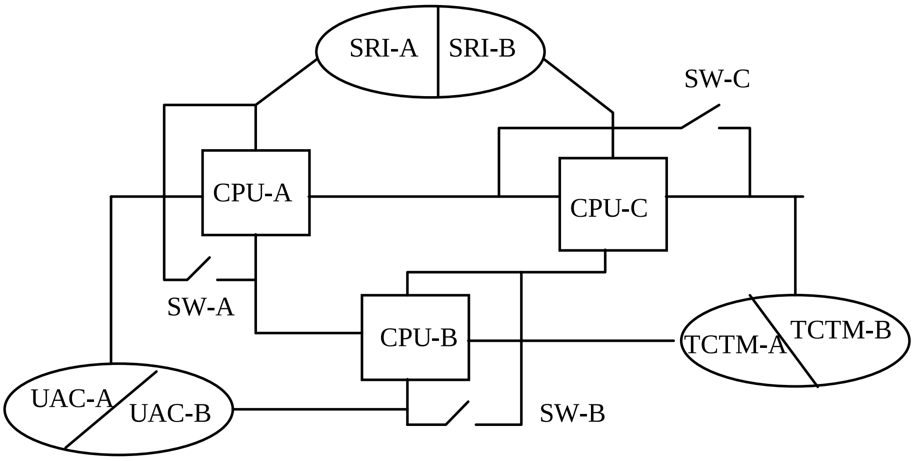


Figure 3 SACI onboard computer block diagram

The SACI computer system was designed to degrade gracefully to two or to one processing modules. As shown in Figure 3, a subset of the switch defined for the standard modified mesh architecture was used. Initially, all the three processing modules are powered on, and the execution of the tasks distributed among them. When a catastrophic fault occurs in one processing module, the computer system degrades to two processing modules. In this case, the tasks are relocated and all tasks can continue normal execution. Figure 4a presents the computer system when the Processing Module C fails and the system is reconfigured.

.

Additionally, the system is designed to degrade to one processing module when any two processing modules fail. In this case, not all tasks will be executed, only the most important ones are selected. Figure 4b presents the computer system after Processing Modules C and B failed.

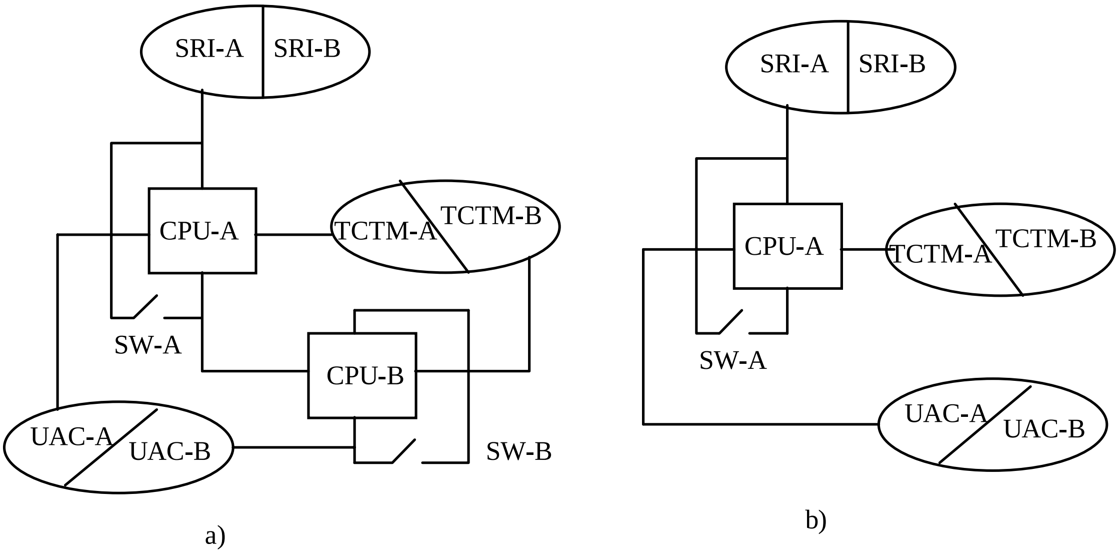


Figure 4 a) Computer system after the failure of the module C b) Computer system after the failure of the modules B and C

The on board software is composed of a set of processes that exchange messages among themselves through a routing process [SAN97]. The active processes in each processing module are defined by a task table that can be updated by the routing process or by telecommands sent from the ground station. The processes are organized in three classes: application, system and interface, as shown in Figure 5.

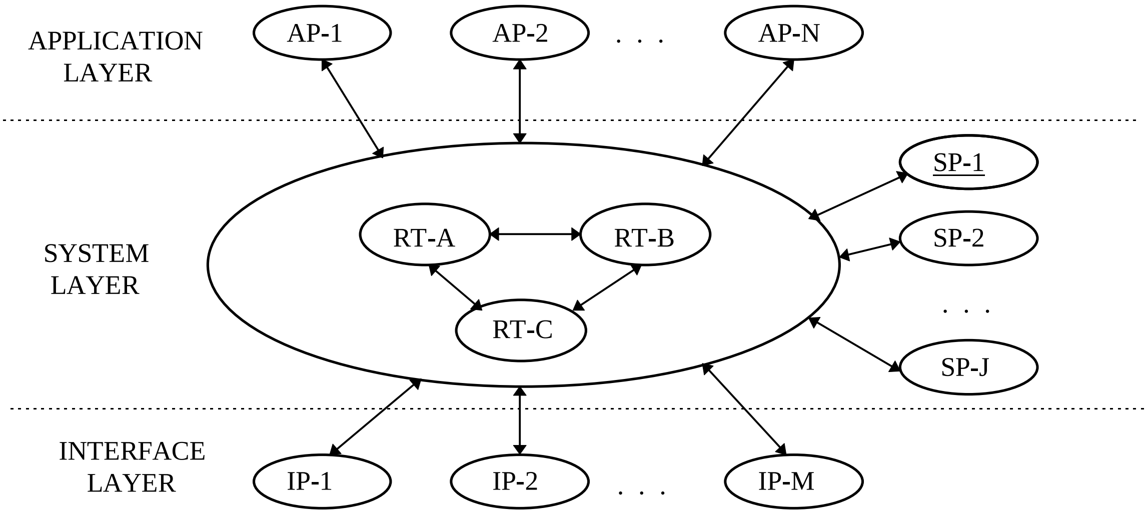


Figure 5 Layered organization of onboard processes in layers

The application processes (AP-1, AP-2,...,AP-N) execute the onboard data processing required by the experiments. The system processes (SP-1, SP-2, ..., SP-J) execute general tasks such as telemetry and telecommand processing, fault analyzes and diagnosis routines and house keeping tasks. Finally, the I/O processes (IP-1; IP-2, ..., IP-M) control the data exchange among the processing modules and the satellite subsystem interfaces.

The routing process is subdivided in three processes, RT-A, RT-B and RT-C, one for each processing module. The main functions of the routing processes are to route the messages exchanged among the processes, process synchronization and detection of malfunction during the message exchange.

The on board routines are stored in PROMs. If, after the launch, an error is detected in one onboard process, it can be deactivated by a telecommand sent by the Control Center and a new process may be loaded and activated. Additionally, a new process to execute a new task may be loaded when the satellite is in orbit.

The critical part of the onboard software (composed of the routing processes and fault treatment routines) was described using the formal language CPS-Z [MOT97]. The CPS-Z language is the union of the Z language [SPI92] used to describe sequential systems with the language CSP [HOA85] used for concurrent system description. After the formal description, it was formally proved that the critical parts of the software were deadlock free [MOT97].

IV. SACI-1 Fault-Tolerant Mechanisms

The SACI-1 fault-tolerance mechanisms were organized in two hierarchical layers: the upper layer that corresponds to the Control Center, and the lower layer that corresponds to the onboard computer. Some of the fault tolerance mechanisms operate automatically, without human intervention, using one or more layers. Others may need the intervention of the Control Center operator or even the specialist group.

The Control Center receives the service telemetry sent by the onboard computer, verifies the consistency and range of the telemetry parameters and the fault reports. If some discrepancy is detected and, when the Control Center is programmed to solve the detected problem, it sends the commands to reconfigure the onboard computer. Otherwise, it sets an alarm to inform the operator that a problem was detected onboard. When the operator is unable to solve the problem, he should consult the specialist team. Based on the specialist's analysis, the operator can program the Control Center computer to send commands to reconfigure the computer system or to execute an additional diagnosis program onboard to obtain more information about the fault.

The lower layer that corresponds to the onboard computer is subdivided in hardware, routing and application program layers. Some mechanisms use only one layer, others use two or more layers. Most of the fault tolerant mechanisms are organized in three phases:
a) Error detection
b) Fault analysis
c) Recovery

The error detection mechanisms are divided in two classes: concurrent with the operation and not concurrent with the operation, such as the diagnosis program. When an error is detected and it is not corrected by the hardware, the fault analysis routine is activated. The main purposes of this routine are to determine the cause of the error, whether it was generated by a transient phenomena, by a permanent hardware fault or by a software bug and, also to estimate the damage in the data structure. The fault analysis routine is programmed to deal with some classes of faults. If it cannot solve the problem in this layer, it sends a report to a higher layer fault tolerant mechanisms. Based on the fault analysis results, the system can return to the normal operation or activate the failure flag line. The failure flag of a processing module is connected to the interrupt input port of the other two modules. Therefore, when the failure flag of one processing module is activated, it causes a system general reset.

After a reset operation, all surviving processing modules execute the initialization program that contains a diagnosis routine that evaluates the health of the processing module. If a catastrophic fault is detected, the processing module maintains the failure flag activated. The modules in good health read the failure flags from the other modules and, then, load an execution task table based on the failure flag line of the other modules. The task table can also be updated by Control Center commands. In the remaining part of this section, the fault detection mechanisms utilized in the processing module will be discussed in more detail.

Each processing module is composed of a CPU based on the Transputer T805 made by INMOS, a watch-dog-timer and link switches (WDT-SW), an interrupt and I/O circuit (INTER-I/O), a main memory (MAIN MEM) and a mass memory (MASS MEM) [CAS95] as depicted in Figure 6.

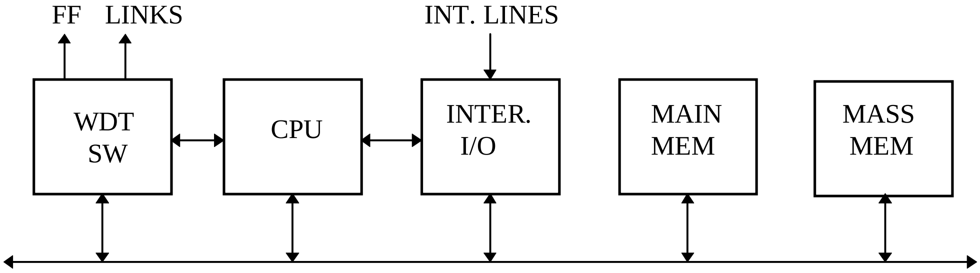


Figure 6 Processing module block diagram

a) CPU

The main fault detection mechanism in the CPU is the watch-dog-timer (WDT) that uses the hardware and the routing layers. If the WDT is not reset within a predefined time interval, it generates an interrupt. If after the interrupt, it is not reset again during a predefined time interval it generates up to seven initialization pulses to restart the processing module. If, after the seven reset pulses, the processing module is not able to return to normal operation, or when the fault handle routine detects a catastrophic failure, the WDT is not reset again and after a predefined time interval the failure flag (FF) signal is activated which causes a general system reset.

This mechanism, in addition to be able to detect different types of hardware faults and transitory errors, it is also able to detect some software bugs, as, for example, when the program is in an endless loop.

b) Main Memory

In the onboard computer, the main memory is composed of memory chips that are sensitive to space radiation. When a high energy particle reaches a memory chip, a single event upset (SEU) may occur, flipping one or more memory bits. To overcome this problem, a circuit to detect double errors and correct single errors (EDC) is utilized. The single errors are corrected automatically by the EDC and an interrupt, that can be masked, is generated. This interrupt is used only for bookkeeping purpose. When a double error is detected, an interrupt request is sent to the CPU and the fault handling routine is activated. When the double error is a critical program code or critical variable region, the processing module is reset. Otherwise, the fault handling routine reloads the routine in which the double error was detected. If the double error occurs several times in same address, the fault handling routine calls the reconfiguration routine to block the memory module in which the hardware fault is located.

c) Mass Memory

When the SACI-1 project started, high capacity memory chips qualified for space applications were not available in the market. So, the design team decided that industrial memory chips that were radiation resistant up to 20 Krads and that presented a low probability to produce latchups due to a high energy particles were to be used. After analyzing different memory chips it was decided to use a memory chip made by Hitachi with a capacity of 512 Kwords of 8 bits. This chips were organized in two independent banks of four Mbytes. For each bank, a circuit to limit the latchup current and to disconnect the power supply of the bank for 10 msec, when a latchup was detected, was utilized.

Since the mass memory is only used to store the experiment and telemetry frames for few hours and the stored data will not be used onboard, a decision has been made not to use additional hardware to detect or to correct errors onboard, but to code the data frames using a error detection and correction code. With the utilization of high density memory chips such as the 512 Kbytes chips, a single cosmic particle can cause double or triple errors in a single word of the memory. To circumvent this problem, the frames are coded using a Hamming code in each bit column of the frame. For each 1 Kbyte data frame, 10 parity bytes were added.

As double or triple errors may also occur in the column bit direction, it was decided to spread the data frame over the memory chip, so that, no consecutive words of the same data frame are located in the same region of the memory chip. Using the column-wise Hamming code and spreading the data frame over the memory chip, it was demonstrated [DEP96] that, if a memory bank generates up 100 SEU a day, less than 0,01% data frames will have errors that can not be corrected in ground. When the single error in the mass memory is corrected (washed) onboard at each four hours interval, the probability of occurrence of a frame with a not recoverable error is less than $1.0x10^{-7}$.

d) System and Application Layers

In the system layer, the main fault detection mechanisms are the diagnosis program that is executed periodically and the verification of the address of messages exchanged between the processes that is accomplished by the routing process. The error detection mechanisms in the application layer depend on the specific application and will not be discussed in this paper . Generally, these mechanisms consist on verifying the consistency of the computed data. A detailed description of the fault tolerant mechanisms, implemented in the SACI-1 onboard computer, is presented in [DEP95c].

V. Conclusion

The proposed mesh architecture is very convenient for small satellite fault tolerant computer systems. This architecture can be easily adapted for different performance and reliability requirements. Additionally, it can degrade gracefully up to one processing unit. The reconfiguration is easily implemented by switching the communication links. The main fault tolerant mechanism in each processing unit is the watch-dog-timer (WDT). When the WDT detects a malfunction generated by a transient or permanent fault, it initially generates a CPU interrupt. If the processing unit is not able to recover, the WDT generates up to seven processing unit initialization pulses. If after this procedure, the processing unit does not return to normal operation, the WDT removes the processing unit from the mesh.

The critical software routine was developed using formal methods based on the CPS-Z language. This approach allows starting the software validation in the specification phase and the generation of the code under strict control, resulting in more reliable software. The non critical routines were organized in application, system and interface processes. If, after the satellite launch, an error is detected in one of these processes, the faulty one can be replaced by a new process sent from the Control Center.

At present, the SACI-1 is in the qualification test phase and the computer system is being submitted to exhaustive tests . The satellite is programmed to be piggyback launched with the Chinese-Brazilian Earth Satellite (CBERS) using the Chinese Long- March 4 launcher in the first semester of 1998.

VI. References

[CAS93] Castro, H. S, “Fault Tolerance through Reconfigurability: Application in Space Instrumentation”, PhD Thesis, University of Sussex , England, June, 1993.

[CAS95] Castro, H. S.; Monteiro, A. M. V.; RochaJ. R. I. and Girão, Q. M., “Trisputer: Computador de Bordo do SACI - Microssatélite Brasileiro”, UFC Technical Report, Fortaleza, Brazil, Mach, 1995.

[DEP95a] De Paula, A. R.; “Preliminary Specification for the Scientific Satellite Computer System”, INPE Technical Report, São José dos Campos, Brasil, Mach. 1995.

[DEP95b] De Paula, A. R.; Lonneux, L.“ Preliminary Design Review of SACI-1 Onboard Computer”, INPE Technical Report, São José dos Campos, Brasil, June, 1995.

[DEP95c] De Paula Jr., A. R, “Aspectos de Tolerância a Defeitos do Sistema de Computação do Primeiro Microssatélite de Aplicações Científicas do INPE", VI Simpósio de Computadores Tolerantes a Falhas, Canela, RS, August, 1995.

[DEP96] Paula Júnior, A. R.; Saturno, M.; Vargas, F.; Velasco R., “A Strategy Allowing of Use Commercial Circuits in Mass Memory of Microsatellites.”, International Workshop in Computer Aided Design, Test and Evaluation for Dependability, Beijing, China, July, 1996.

[HOA85] Hoare,C. A. R., “Communicating Sequential Processes” Prentice Hall, 1985.

[MOT97] Mota. A. C. “Formalização e Análise do SACI-1 em CSP-Z”; Master Dissertation, UFPe, Recife, Brazil, September, 1997.

[NER95] Neri, J. A. C. F.; Rabay, S.; Dos Santos; W. A.; De Souza, P. N.; Fonseca, I. M.; De Paula, A. R. ; “Key Technological Solutions towards the SACI-1Microsatellite Design”; Tenth Annual AIAA/USU Small satellite Conference; Logan, USA, September, 1996.

[SAN97] Santiago, V. A.; Monteiro, A. M. V.; Castro, H. S.; Thompsom, J. A. and Girão, A. M.; “SACI_FTR: Concepção, Projeto e Implementação do Software Gerenciador Básico para o Computador de Bordo do Primeiro Microssatélite Brasileiro de Aplicações Científicas (SACI-1)” VII Simpósio de Computadores Tolerantes a Falhas, Campina Grande, PB, July, 1997.

[SPIE92] Spivey, M., “The Z notation”, Prentice Hall, 1992.