High-reliable Data Acquisition and Control System

A. Balázs, I. Hernyes, I. Horváth, Z. Pálos,
P. Spányi, S. Szalai, L. Várhalmi

KFKI Research Institute for Particle and Nuclear Physics
Division of Space Technology

Abstract

Our institute in common with the Cosmology Group of the Eötvös University participates in an international astronomical space project called Spectrum-X-Gamma. In the frame of this project a scientific satellite is under construction. The data acquisition system of the spacecraft is developed by the Division of Space Technology. It is based on a central onboard computer with the goal of controlling the scientific instruments and the telemetry system.

The computer has a modular design and it is based on Intel’s 80C86 processor. Because reliability is a key issue, all modules are redundant and can work either in cold or in hot reservation modes. High speed bidirectional serial bus links maintain the communication between the modules and the external instruments, providing a safe yet flexible connection. Various hardware and software techniques are applied to ensure fault tolerancy of the system.

1. Introduction

The following lecture presents an application of microprocessors in a rather special field: in space technology. In a wide international cooperation an astronomical spacecraft called Spectr-X-Gamma is planned to be put into orbit in 1995. It will be equipped with a set of latest technology telescopes and detectors, operating in X-ray, Gamma-ray and ultra-violet bands, to observe celestial bodies in the high-energy region of the electromagnetic spectrum. The reason why the instruments will be flown on a satellite is that the Earth’s atmosphere is not transparent for these wavelengths. Scientists hope that the spacecraft will provide them with a lot of new information on astrophysics and cosmology.

The more than 10 scientific instruments will collect a big amount of observational data (exceeding 1 Gigabytes each day), which will be transmitted to the ground by the spacecraft’s telemetry system. Gathering of the data from the instruments and organizing its transmission to the Earth will be carried out by the central data acquisition and control system of the spacecraft. The Division of Space Technology of our institute is responsible for the development and for the construction of this part of the satellite.

The expected lifetime of the spacecraft is at least 3 years. Since there is no way of repairing failed components while the spacecraft is in orbit, the equipments must meet very high reliability criteria. The data acquisition and control system has a key role, because it
controls all scientific instruments and without it it would be impossible to transmit any observational data to the ground. Consequently, it must be not only reliable but also fault tolerant, in order to maintain its activity should any failure occur. Fault tolerance means the ability to detect failures of elements of the system, and to continue working in a degraded mode in such cases.

2. The task of the data acquisition and control system

The data acquisition and control system consists of an onboard computer and a data bus which is connected to all scientific instruments of the satellite. The onboard computer has the following main tasks:

- autonomously controlling the operating modes of the instruments, according to a preprogrammed sequence
- distributing individual commands to the instruments upon receiving them from the ground
- providing the instruments with the onboard time code (which is needed for data calibration)
- collecting the observational and housekeeping data from the instruments and transmitting it to the ground

In addition to these main tasks, the onboard computer must continuously check the state of the instruments and, if necessary, detect possible errors, it must generate and transmit housekeeping data frames and carry out reconfiguration actions in emergency situations.

The computer is microprocessor based. Because the speed and performance requirements are not critical in this application, Intel's, 80C86 processor is used. Although the onboard computer contains most of the components of a traditional microprocessor based system, several hardware and software methods are applied to ensure its reliability and flexibility.

3. The System Interface Bus

In order to reduce complexity of the onboard cabling, the onboard computer and all scientific instruments are connected to a common serial bus link, called System Interface Bus. This provides a relatively simple connection scheme, allowing that all instruments and the onboard computer can exchange information between each other. The electrical parameters and the communication protocol of the link is based on the MIL-1553 standard, which defines a time division multiplex data bus system, invented originally for military applications, such as aircraft control.

The key elements of this standard are the bus controller, the remote terminals, the serial data bus cable and the isolation couplers. In our case the onboard computer has the role of the bus controller, and the scientific instruments behave as remote terminals. The bus controller's main function is to provide data flow control for all transmission on the bus. In this role, the bus controller is the sole source of communication. The system uses a command-response method. The remote terminals (namely the scientific instruments) are allowed to transmit or receive data only if they are requested to do so by the controller. The transmission technique is half-duplex, the operation is asynchronous.

The data bus itself is a shielded twisted pair cable. Its electrical properties allow data traffic with a raw bit rate of 1 Mbit/sec. The data is Manchester bi-phase coded, thus the
lack of DC components allows transformer coupling. The information exchange is based on messages with maximal length of 1024 words. (One word consists of 16 valuable bits).

All devices (controllers and terminals) are connected to the bus through bus couplers. The couplers, which contain transformers and passive protection circuits, isolate the bus and the devices. The purpose of these elements is to prevent a short on a terminal from shorting the entire bus.

For safety reasons, the System Interface Bus is duplicated and used in a cold reserved manner. Only one bus is active at a time, and the controller can easily overswitch to the reserve bus, upon detecting a malfunction on the main one.

4. The internal architecture of the onboard computer

According to our experiments, applying the bus standard described above provides a safe and error-propagation free method for information exchange. That is why the data bus is used not only for connecting the onboard computer with the external instruments, but also for linking together the internal modules of it. The main design principle of the computer was modularity.

The functional block diagram of the onboard computer is shown in Fig. 1. It consists of modules having different tasks and being loosely connected with each other. Each module has its own separate power supply and can be switched on and off individually. Because the connection between them is maintained through transformer-coupled buses, they are galvanically isolated. This high level of isolation prevents the system from error-propagation from a failed module to the good ones. Modularity offers one more advantage during the development phase: Every module of the computer can be tested separately, using the same uniform test equipment. The computer is composed of the following modules:

- Processor Modules (PM1,..PM3)
- interface modules:
  - Fast Telemetry (FTM1,..FTM3)
  - Slow Telemetry (STM1,..STM3)
  - Analog Telemetry (ATM)
  - Command and Time Receiver (CR1,..CR3)
- Power Supply (PS)

Only the Processor Modules contain microprocessor, representing all the intelligence of the system, while the interfaces perform the necessary buffering and data frame formatting toward the radio transmitter and from the radio receiver of the spacecraft. The three kinds of telemetry interfaces are used for transmitting different types of information in different formats: science data (FTM), digital housekeeping data collected from the instruments (STM), and analog information about the power conditions of the onboard computer (ATM). The incoming information (command and time codes) are received by the CR interface. The Power Supply has 27 V DC input, and contains separate DC/DC converters for each module.

The modules are connected together through the System Interface Bus (SIB). There is another serial bus link between the Processor Modules, called Processor Unit Bus (PUB). This can be regarded as an internal service bus (technological bus). During ground testing it helps debugging and modifying the onboard software, during flight it is used for inter-
Fig. 1  The simplified functional block diagram of the onboard computer
processor communication if more than one Processor Modules are operating at a time. Physically the Processor Unit Bus is very similar to the SIB.

5. Reservation methods

To increase the overall reliability and the lifetime of the system, its main parts are reserved. The modules of the computer are triple redundant: there are 3 identical Processor Modules and also 3-3 identical modules for the different types of interfaces. The system can work in any combination of the modules: any Processor Module can work together with any other interface module. Various reservation techniques could be applied, depending on the required operation mode of the data acquisition system:

If uninterrupted operation is a critical issue, the onboard computer can work in a hot reserved manner, when two or all the three modules of each type are powered. This could be needed when important observations or other actions are carried out and momentary breaks in the operation or data losses are not allowed. Since all communications between the modules are done through the serial System Interface Bus, it is ensured at software level that only one bus controller is working actively at a time, while the reserve modules are only "listening" to the bus. In hot reserved mode the software recognizes and isolates the failed module, and automatically over switches to the redundant one.

When temporary breaks in program execution and short time data losses are allowed to happen, cold reservation can be applied. In this mode only one module is powered at a time, and recognition of possible faults is done by self-diagnostic functions. Overswitching to the reserve module needs manual interaction by sending remote ground-commands, so-called relay commands. Although cold reservation implies a lower level of autonomy, for most of the lifetime of the Spectr-X-Gamma spacecraft this will be the preferred mode, because of two reasons: This needs less power consumption, and the electrical components tolerate radiation much better when not powered. Both issues are critical: power is always limited onboard of a spacecraft; and due to the location of the satellite's orbit the radiation flux will be relatively high.

It is also possible to implement mixed reservation: When only the most important and most complex elements of the computer (e.g. the Processor Modules) are hot reserved, and the rest of the modules operate in cold reserved mode.

The advantage of using the serial bus (SIB) as the only communication link between the modules is that changing the active module or the reservation mode needs only to switch on/off selected modules. The system reconfigures itself automatically, by applying software tools only.

6. The Processor Modules

The basic elements of the onboard computer are the Processor Modules. Functional block diagram is shown in Fig.2. They are the real copy of each other, having both electrically and mechanically the same construction. They contain system programs and all of the application tasks to be performed by the computer in their own local program memory, in order to be able to replace each other in every single respect. The computer itself is capable of recognizing of its own faulty operation - either the faulty module itself, or working in hot reserved mode, by more modules cooperating with each other. Depending on
Fig. 2 A Processor Modul
the reservation mode, overriding the failure is done either by human decision with remote commands, or automatically - providing minimal interruption in operation.

A Processor Module itself is not redundant, though in some functions it has redundant components as well (i.e. clock generator, bus interface) and has memory protection. The required reliability and expected lifetime is ensured on module level. This, of course, does not mean that the Processor Modules do not have to be reliable, otherwise loosing of system-sources might be speeded up. The appropriate reliability can be achieved by worst case design, applying qualified and radiation tolerant electrical components. The type of microprocessor is 80C86, this is one of the few processor types approved for usage in space.

The memories are the most sensitive to the radiation in the outer space, thus it is very desirable to take care of program and data protection. That is why the memory is protected by Hamming-code by applying an error detector/corrector circuit. Every Processor Module has 32 kbytes of electrically erasable PROM (EEPROM) and 32 kbytes of RAM, both memory types are protected. In addition, the module contain also a small-capacity ROM, to store the core of the operating system and the most vital tasks, because the ROM is the least endangered by radiation. Except the ROM, the memory of the Processor Module can be reprogrammed, during development phase its internal software could be updated without disassembling the computer; and even during flight there will be a limited way of reprogramming.

Being the controller of the System Interface Bus, the Processor Module has direct access to it, and it has the right to select and overswitch between the redundant busses. Bus handling is implemented by direct memory access (DMA), this is the fastest method requiring minimal software overhead.

When more than one Processor Modules are powered, they use the Processor Unit Bus for communication. Every module has a unique module identifier. On the basis of it, they can decide which one of them is the active and which is the hot reserve one. Through the PUB the active powered module can keep providing the reserve one with crucial data and critical parameters during operation, thus the computer can continue working even if a module fails. The information exchange on the PUB is based on a token passing method, providing a continuous dialog between the modules. In cold reserved mode the PUB is not used (except during ground tests).

7. Error detection, reconfiguration

Error detection during operation is expected to be efficient and quick. To meet this expectation, it is supported by the error detection circuitry on processor module level. Its task is to qualify the Processor Module itself. Although in principle it could be doubtful if the module itself is capable of making such a decision regarding to the correctness of its own operation. Fortunately, a large number of possible faults can be recognized with the help of some simple hardware and/or software elements. Using a few simple methods, being principally different from each other, increases the range of recognizable faults and the probability of the recognition as well (i.e. memory error detector, memory write attempt into PROM, hardware watchdog etc.). As a rule, there are some possible faults, which can not be detected by the localized error detector circuitry.

The error detection is performed by built-in diagnostics and can be divided into two groups: self-diagnostic and cross-diagnostic. The duty of self-dianostic is to qualify each
single Processor Module. It is performed by both hardware and software means. The hardware part of it is a watch-dog logic, which generates an alarm if it is not reset within a certain time constant (i.e. due to program crash). In addition, a self-diagnostic program is running in the background and checks the state of the Processor Module periodically. The hardware and software error detector mechanisms complete each other in the sense that the software checks those components of the local module that the watch-dog circuit is unable to check.

In fact, some faults can escape the self-diagnostic or they are simply out of its "visible" range. To discover such errors, the next level of error detection is implemented, as cross-diagnostic. As a result, the Processor Modules keep checking each other via the Processor Unit Bus. For this reason, from time to time it will be desirable to operate the onboard computer in hot reserved mode - even if otherwise cold reservation would be sufficient.

Taking into account that the self-diagnostic offers the fastest fault recognition, it is important to find the most reasonable division of labour between self- and cross-diagnostics. Thus the components being responsible for the self-diagnostic are only to be tested in the other modules by the cross-diagnostic. These are:

- processor
- the memory segments containing the self-diagnostic programs
- to make sure that the self-diagnostic in the other module runs properly

Besides the Processor Modules themselves, the additional interface modules of the computer must be also tested regularly via the System Interface Bus.

In this way the two types of diagnostics give a good coverage of sensing a wide range of possible malfunctions and that is the basis of ensuring faultless operation.

Upon detecting critical and not correctable errors, the operation of the faulty module must be suspended and a redundant module must replace it. While working in cold reserved mode, all what the Processor Module can do is to isolate itself from the buses and to wait until the reserve unit is switched on from the Earth, by sending high-priority ("relay") commands. If the failure occurred in the hot reserved state of the computer, a reconfiguration process begins - a very similar action what is done after general power on. In the frame of this process, first the currently powered modules are identified, then one of the Processor Modules is selected as the new active controller.

A reconfiguration does not necessarily means a substantial brake in program execution, that is, the program does not have to be executed from the "very beginning". Regular saving the carefully selected crucial system variables and critical parameters into the passive powered module, and then taking them after a reconfiguration ensures that the computer can carry on its normal operation without any substantial trouble.

8. The software of the onboard computer

The function of the computer's software system is to provide data flow control between the Earth data communications systems and the scientific instruments. It includes the reception, decoding, storing and distribution of ground-commands, collecting and sending the instruments' science and housekeeping data to the Earth, among other less important tasks, such as continuous maintenance and operation of the hardware.

The software system consists of several more or less independent parts. Some of these
parts were written in 8086 assembly, the remaining parts in C programming language.

The most basic part of the system is a simple real-time multitasking operating system, developed also by our institute. This part of the program gives the possibility of separating the application parts of the software according to their functional role, and provides these parts with several features, helping them to coordinate their activity.

The application parts are called tasks. The operating system maintains a table, where the necessary data for the supervisor (the task scheduler) about each task is stored. Such data are e.g. the current stack pointer, the activity description bits, etc.

Each task has its own stack, in which its register contents are saved when a scheduler interrupt occurs. At that moment the supervisor saves the current task’s description information, sets its description bits, and starts scanning through the mentioned table for the first task being ready to run. This task will then be set to running state, then its stack pointer and other registers’ content will be restored if it was interrupted, or if it is a recently started task, the control is simply given to its entry address.

Another task scheduling function of the supervisor is the task timing facility. In order to avoid a task being excluded from the CPU usage by another time consuming task, a certain time limit can be defined for each task. If this time interval is elapsed, the supervisor takes the control away from the task currently running, and gives it to the next ready-to-run one. There is a primitive call for the tasks which has the same effect: so that if the task recognizes that it has nothing to do at that moment, calling this primitive gives the CPU to the other tasks.

The supervisor provides the tasks with several calling primitives by which they can use the semaphore system of the operating system. A time-out timer can be attached to each of these semaphores so that deadlock situations can be avoided. Using these semaphore control primitives, a task can coordinate itself, or if several tasks share the same semaphore, they can implement mutual exclusion, or communicate with each other. The semaphore control primitives include testing a semaphore’s state, allocating, releasing and waiting for it.

The most important part of the onboard software is the operating system, but it just gives the opportunity to execute different functions in parallel. The real activities are done within the application tasks.

The following functions are separated into different tasks in the computer:
- physical service of the System Interface Bus
- handling of the Processor Unit Bus
- processing ground commands
- scheduling of periodic bus functions
- testing the operative memory
- supporting debugging features

The task which handles the System Interface Bus is able to check the quality of the bus communication and to overswitch to a reserve bus. This task communicates with the other ones using a large ring-buffer. Some software subroutines provide the opportunity of sending commands to this task. These commands are stored in the buffer in a first come, first served way.

The PUB handler task performs the communication between the Processor Modules.
is implemented at task level, meaning that any task can send message to any other one within any currently working module. This task is in charge for building up of the token ring communication in case of any error, or when a module is switched off or on.

The memory of the Processor Module is continuously tested by a task which, using the error detector and corrector circuitry, recognizes bit errors, and if possible, writes back the corrected data to the erroneous location. This memory error corrector task runs in the background, when no other activities are in progress.

There is an additional task, which helps the debugging of the software during development phase. This task uses the PUB for communication with another external device, such as a personal computer, implementing the human interface. This task is able to execute a very limited set of commands: for example the memory can be examined and/or changed, and some system level functions can be executed, such as task activation or deactivation.

These tasks usually interact with each other. For example, in every 125 ms it shall be tested whether a new command is available. When this interval is elapsed, the command processing task first of all issues a request to the bus service task to read a command in. When a command arrives, this service task passes it to the requester. Similarly, when a command must be sent to an instrument, the command processor task issues another type of request which commands the bus service task to send a command to the instrument. The scheduler task works in a similar way, interacting with the bus service task.

9. Summary

A high-reliable microprocessor based system was presented, which was intended for data acquisition and instrument control onboard of a future spacecraft. A combination of hardware and software solutions - including modularity, redundancy, automatic self-diagnostic and reconfiguration ability - were implemented to ensure safe and uninterrupted operation of the system in a harsh environment.

The development is founded by the National Committee for Technological Development.

Literature

