BIRD- Spacecraft Bus Controller

Sergio Montenegro
GMD, Institut FIRST
Kekuléstr. 7, 12489 Berlin, Germany
http:/www.first.gmd.de/~sergio
Wolgang Bärwald
DLR, Institut für Weltraumsensorik und Planetenerkundung,
Rutherfordstr. 2, 12489 Berlin, Germany

The spacecraft bus is controlled by the dependable board computer of the satellite bus. To achieve a high dependability, safety, and lifetime, the board computer is formed of four identical computers (nodes). As shown in the block diagram below, the redundant nodes and all the devices of the satellite that have to be controlled by the board computer are interconnected by several bus systems with different protocols.

The architecture of the redundant control computer is totally symmetric, that means, each of the nodes is able to execute all control tasks. One node (the worker) is controling the satellite while a second node (supervisor) is supervising the correct operation of the worker node. The two other node computers are spare components and are disconnected. If an anomaly of the worker node is detected by the supervisor node the supervisor becomes the worker and takes over the control of the satellite. The old worker node is enforced to execute a recovery function and, if there is no permanent error detected, it becomes the supervisor node. If the recovery procedure fails or if a permanent hardware error is detected, the faulty node computer will be switched off and replaced by one of the spare nodes. By this strategy up to 3 permanent node failures can be tolerated while the board computer stays operable.

The node computers are based on the PowerPC MPC623 processor. Each node has 4 MByte of FLASH memory and 8 MByte DRAM memory. The FLASH memory is large enough to hold several versions of the control software, each protected by a checksum. The DRAM is parity protected and duplicated which allows for error detection and correction. The I/O devices, the interfaces to the satellite bus systems and the logic of the telemetry control system are realized within a complex FPGA device. To detect errors within the control logic, there are redundant implementations of the critical state machines and each data within the FPGA is protected by parity bits. All devices are protected against latch-ups.

The control software is based on our own real time operation system (BOSS). The highly modular operating system software was implemented by using newest software technology and the critical parts have been formally verified. The applications running on top of BOSS are implemented by using object oriented technology, resulting in a highly modular application software. The BOSS operation system is available not only on the target system but also as a guest level implementation on top of Linux. This allows to implement and test all the application software on Linux workstations and move them to the target system without any change.

To achieve well structured application system, we defined a software back plane formed out of two software busses. Each application implements an interface to each software bus. One software bus is used to distribute commands that have to be executed by the applications. The other bus collects the status information of the applications which have to be sent down to the control stations on the earth. The principle of a software back plane allows an easy configuration of the system by simply plugging the software components in and out of the back plane.