A Fault-Tolerant Middleware Switch for Space Applications

Sergio Montenegro, Ebrahim Haririan
German Aerospace Center (DLR)
Institute of Space Systems
Robert-Hooke-Str. 7
28359 Bremen, Germany
sergio.montenegro@dlr.de
ebrahim.haririan@dlr.de

Dependability is a main issue for space applications. After more than 30 years of research, a general solution to achieve dependable computing, has not yet been found. There are many proposals on how to achieve fault tolerance, robustness or fault prevention etc. but not a single global accepted solution. The main risk factors in a typical core avionics development are complexity, software-hardware interfaces and difficulties to handle many different interfaces in a single system. These topics will be addressed in order to get high dependability.

Typical data systems for space applications are computer-centric. The central component is a computer to which all (many) devices are attached. The computer has to handle devices, communication, computing, and storage of data. The new DLR Institute RY in Bremen started an ambitious project to create a new concept of core avionics systems that targets fault tolerance as a natural part of the concept. This core avionics system will control the compact satellite AsteroidFinder. In our approach, the central component will not be the computer but a distributed fault-tolerant network system. We provide dependability to the network. A set of undependable redundant components like devices, simple computing units, mass memory units etc. can be attached to this network. Any of these devices may fail. The network manager will deactivate the failed device and activate a redundant one producing the same services as did the one failed.

The Network is built using relative simple middleware switches, which at the first implementation are fully implemented in an FPGA. SRAM-based FPGAs are very susceptible to SEUs stem from radiation effects in space applications. Considering robustness as a key to success in space missions, a need for fault-tolerant design is inevitable. We have selected an internal TMR (Triple Module Redundancy) approach in spite of more power consumption together with impacts in timing. Due to high increase of area with TMR, sort of partial redundancy is applied into the middleware switch to minimize fault-tolerance overhead. Every Flip Flop of the system is replicated and voted according to the TMR approach. This includes all registers and state machines. To implement TMR state machines, states require an explicit encoding scheme. Among all, Hamming-3 encoding is a better choice for a very good reliability. Using Hamming-3 encoding, three bits should be changed in any state in order for the state machine, to malfunction. An additional Active-Supervisor architecture even makes the middleware system, radiation-hardened up to a more reasonable level of reliability.

The programming paradigm is based on a distributed network of services. The most effective and safe way to implement a complex parallel system is to compose it as a network of simple sequential tasks. These tasks may be executed by software - e.g. steering control - or by hardware components - for example providing temperature measurements. We aim to unify software and hardware so that there will be no difference if services are provided either by software or hardware.