Architecture of the SSB Core Avionics System

Dr. Sergio Montenegro, Lutz Dittrich
DLR, Institute of Space Systems, Am Fallturm 1, 28359 Bremen, Germany,
http://www.dlr.de/irs
email: Sergio.Montenegro@dlr.de , Lutz.Dittrich@dlr.de

Introduction

DLR (German Aerospace Center) is currently developing a low cost satellite bus named Standard Satellite Bus (SSB) for LEO satellites in the 100 kg class. The SSB shall be used for different DLR internal scientific missions. This paper presents the heritage, design philosophy and planned architecture of the SSB core avionics system.

Heritage

The SSB core avionics system is a further step in the development line of a software and hardware architecture which was first used in the successful BIRD-Mission (Bispectral InfraRed Detection).
The core avionics system of BIRD consists of four identical computer nodes. One node is the master node, one node is in hot stand-by the other two nodes are cold-redundant [1]. Different peripheral devices like S-band receiver, reaction wheels, star sensors, GPS receiver, IMU and magnetic torquers are connected to the computer nodes via an FPGA. These peripheral devices all have different interfaces and use different protocols. The CPU nodes are powered by an embedded real-time operating system named BOSS.
The next step in the development line is the technology test carrier TET of the German on-orbit verification program. TET is based on BIRD-hardware but has an improved software system. The most important new feature of TET's core avionic system is it’s message oriented middleware (MOM). This middleware provides a standardized interface for all peripheral devices to the application programmer.

Design Philosophy

The core avionics system planned for the SSB should be reliable, dependable and fault-tolerant. To achieve these design targets and to avoid design error the system should be kept as simple as possible.
The most effective and safe way to implement a complex parallel system is to compose it as a network of simple sequential co-operating tasks, which communicate by offering and consuming services.
One challenge in typical core avionics development is the diversity of different software-hardware interfaces to access the peripheral devices. The improved core avionics concept targets these problems and aims to provide a very simple integrated solution of software and hardware. The border between both shall vanish. In this

concept the functionality is provided by a network of services. Some of them are implemented in classical CPU-software, some in FPGA-software and some in hardware devices, for example in sensors or actuators. All services provide the same interface. There shall be no difference in how it was implemented (CPU, FPGA or hardware) and where it runs. The core avionics system shall be a distributed computer system. No single node is required to be dependable. The service providers are connected by a dependable hardware network, which is the heart of the system. Software services can be distributed on all computer nodes and may migrate from one node to another for example in case of failures, overloading or for power management purposes.

In the same way like the hardware network, there shall be a software network, which interconnects all services, including software tasks running on the same node, on a different node, FPGA programs and even hardware devices. The global interconnection network is called the Middleware. Tasks communicate with each other using the middleware.

Architecture

The architecture of our new core avionics system consists of computing nodes, mass memory, a diversity of different sensors and actuators and a dependable multicast middleware switch. All computing nodes, the mass-memory and the peripheral devices are connected with each other via the switch (see fig. 1).

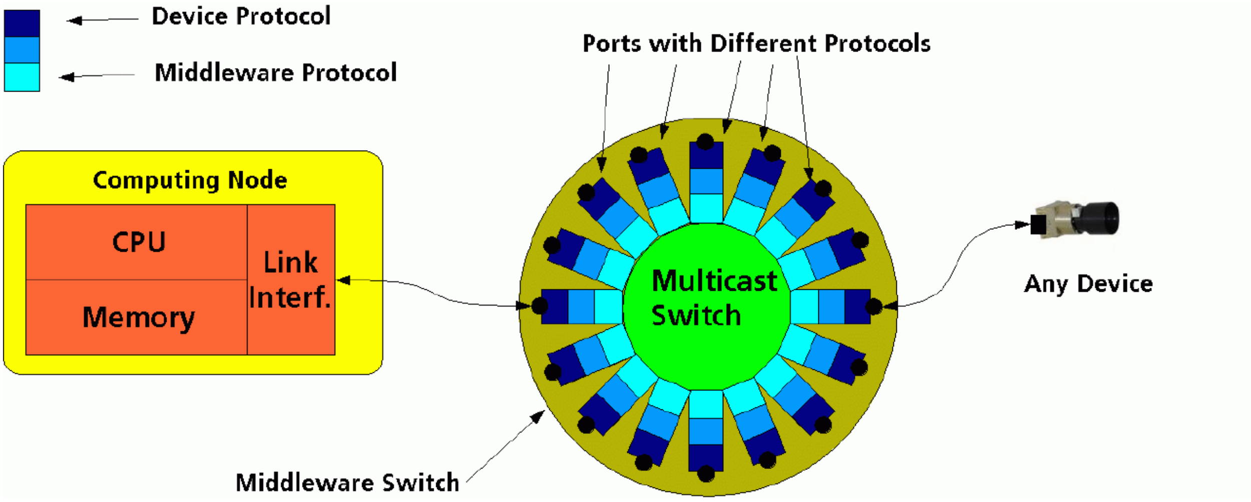

Figure 1

The middleware switch is a multicast capable bus. Each port provides a protocol translation layer to connect a device with its own interface and protocol to the middleware bus.

To decrease latency and to improve throughput it is possible to use multiple middleware switches and to connect them via special routing nodes (see fig. 2).

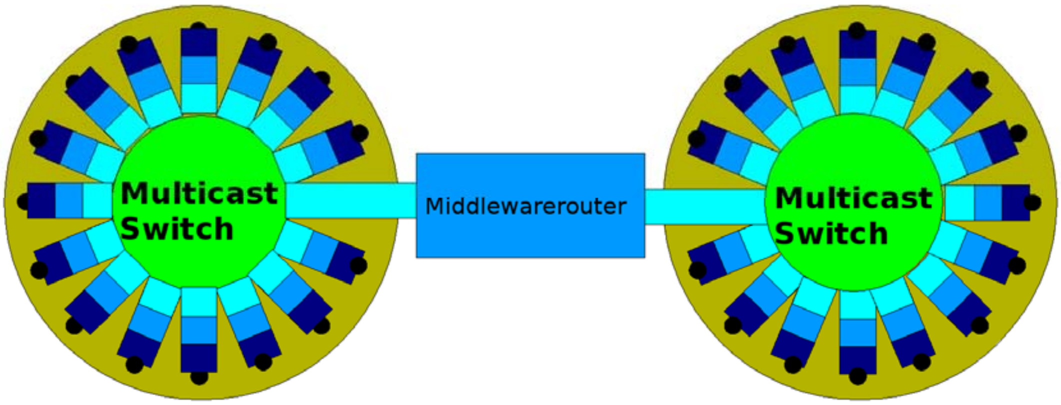

Figure 2

To implement the system, it is planned to use Xilinx Viretex4 FX12 FPGA devices. These devices provide an embedded PowerPC 405 core surrounded by blocks of RAM and FPGA fabric. The PowerPC cores in the Virtex4FX FPGAs have the advantage, that their instruction- and data cache can utilize parity-code-protection[2] to prevent faults induced by single event effects (SEE) , which was not possible in Xilinx Virtex II Pro devices.

Together with scrubbing techniques[3] and extensive use of EDAC for memory protection very effective measures can be implemented to mitigate SEEs. So the overall system could be very reliable and dependable.

Conclusion

The presented core avionic system is a successor of a flight proven core avionic system of BIRD. It will be designed to improve reliability, dependability, flexibility and ease of use.

Literature and References

[1] Montenegro, S.; Bärwald, W.: BIRD-Spacecraft bus controller. Small Satellites for Earth Observation, Digest of the 3rd International Symposium of the International Academy of Astronautics, Berlin, April 2-6, 2001
[2] PowerPC Processor Reference Guide; Xilinx, Inc.: UG011(v1.2) January 19,2007;
[3] Radiation Effects & Mitigation Overview; Xilinx, Inc.: http://www.xilinx.com/esp/mil_aero/collateral/presentations/radiation_effects.pdf