FPGA based On-Board Computer System for the "Flying Laptop" Micro-Satellite

Felix Huber[(1)], Peter Behr [(3)], Hans-Peter Röser[(2)], Samuel Pletner[(3)]

[(1)] Steinbeis Transferzentrum Raumfahrt, Rötestr. 15, D-71126 Gäufelden, Germany, huber@tz-raumfahrt.de
[(2)] Universität Stuttgart, Institut für Raumfahrtsysteme, D-70550 Stuttgart, Germany, roeser@irs-uni-stuttgart.de
[(3)] Fraunhofer Institute for Computer Architecture and Software Technology (Fraunhofer FIRST), Kekulé Str. 7, D-12489 Berlin, Germany, peter.behr@first.fraunhofer.de, samuel.pletner@first.fraunhofer.de

ABSTRACT

The availability of new high density FPGA technologies enables innovative approaches to the architecture of on-board computer systems (OBCS) for demanding small satellite applications. In the paper, we present the architecture of a highly integrated satellite computer exploiting the powerful features of Xilinx's Virtex-II Pro FPGA technology. The computing resources of the "Flying Laptop" satellite are provided by a single OBCS performing the control functions of a traditional satellite bus controller (SBC) as well as all tasks of the payload data handling and processing system (PDH). The unification of the different computing functions on board the satellite onto a single but highly redundant computer system results in clear-cut system structure and provides a high degree of fault tolerance with minimal resource requirements.

The "Flying Laptop" micro satellite is under development at the Institute of Space Systems at the Universität Stuttgart. The primary mission objective is to demonstrate and qualify new small-satellite technologies for the future projects. In addition, the three main scientific payloads of the satellite will allow performing a number of ambitious earth observation experiments. According to the primary mission objectives, the OBCS itself will be a subject for evaluations of its innovative concepts and for qualification of its underlying hardware.

1 THE "FLYING LAPTOP" MICRO-SATELLITE

The total mass of the cubical satellite body with 60 cm edge length will be around 100 kg. The power supply system is based on three solar panels two of which are deployable. The triple-junction solar cells have an efficiency of approx. 26%. The peak power consumption of the satellite will be about 300 W which will be delivered by a battery pack consisting of 8 Lithium ion cells.

The attitude control system is designed to achieve a pointing accuracy of better than 2.5 arc seconds. In addition, the planned scientific experiments require a very high manoeuvrability which both are challenging requirements for a micro-satellite. The "Flying Laptop" satellite is three-axis stabilized by momentum wheels, which can be de-saturated using magnetic coils. The attitude motion is monitored by five different types of sensors: a 3-axis magnetometer, a coarse sun sensor system, four fibre optical rate sensors, a star tracker unit with two camera heads and a navigation system consisting of three GPS receivers.

2 PAYLOAD SYSTEMS

For BRDF measurements, a multi spectral camera system is formed by three single cameras, one for each channel (green, red and near infra-red). The ground sample distance (GSD) of the 1024x1024 Pixel images is 25m. The three cameras are mounted in a triangle mode on a reinforced carbon fibre composite plate for better alignment and thermal stability. In addition, the star cameras are mounted on the same plate which gives precise orientation ability of the imaging system.

The second payload is a thermal infra-red (TIR) camera system for images with a GSD of 50 m (320 x 240 Pixel). The TIR camera system uses a 50 cm Cassegrain mirror, a relay optic and a micro-bolometer sensor. Therefore, no cooling is necessary which makes the system small in size with low power consumption and very cost effective.

The third payload of the "Flying Laptop" satellite is a high performance Ka-band communication system. The travelling wave tube (TWT) amplifier of the Ka-band transmitter will deliver an RF transmission power of 57 W (170.5 W DC input) which is unique for a micro-satellite. The 50 cm primary mirror of the TIR camera will also be used as the reflector dish for the Ka-band communication system. This dual use of the 50 cm mirror is absolutely necessary to install the two complex payload systems on a micro-satellite platform.

The Ka-band transmitter is expected to achieve a downlink data rate of up to 500 Mbps during a ground station fly-over. In addition to evaluating the Ka-band transmitter technology for future lunar missions, a number of scientific experiments to study the Ka-band wave propagation are planned. The dominant effects to signal attenuation at frequencies beyond 10 GHz are clouds and all kind of precipitation. In addition, gas absorption, and tropospherical refraction play an increasing role with higher frequencies. By measuring signal propagation and the atmospheric attenuation in the Ka-band frequency range under different conditions the causal rela-

tions of different and combined effects will be systematically explored.

Another experiment will use the Ka-band transmitter for global precipitation measurements. The experiment is based on the strong correlation between the amount of precipitation and the difference of the attenuation to Ku- and Ka-band signals. That's why the "Flying Laptop" will also be equipped with a low power Ku-band transmitter. The main advantage of this method is that the measured amount of precipitation is independent of the rain characteristic, e.g. the distribution of the raindrop size. It is further planned to use the Ka-band transmitter as a powerful radar source with can be received by ground stations within a spot diameter of 25 km.

Further scientific experiments will be performed by using the three GPS receivers of the satellite navigation system and the star sensor camera unit of the attitude control system. More detailed descriptions of all scientific mission objectives can be derived form the "Flying Laptop" website.

3 FPGA BASED ON-BOARD COMPUTER SYSTEM

The variety of scientific instruments and the broad range of the scientific mission objectives are creating challenging demands on the performance and functionality of the on-board computing infrastructure. By following the system-on-chip (SoC) design approach and by exploiting the capabilities of the new high density FPGA technologies, the complete functionality of the OBCS could be integrated onto a single printed circuit board (PCB) of standard Euro card size (100mm x 160mm). The highly integrated and configurable system-on-chip (SoC) approach allows integrating most functions of the node within a single chip and offers significant advantages over a conventional system design especially for the development of space computer hardware. The reduced number of components reduces weight and volume of the OBDH system and results in better reliability figures. In addition, the SoC design provides higher operation speeds and reduced power consumption due to on-chip vs. off-chip connections. By exploiting the high degree of parallelism within the FPGA fabric lower clock frequencies are required which further can reduce the power consumption significantly. Another advantage of the programmable FPGA technology is its high flexibility that allows for in-orbit reconfiguration of the hardware functions e.g. to adapt the system to changing mission requirements or to update the FPGA configuration to fix design errors.

Another unique feature of the OBCS is the software-less implementation of all control and processing functions. The complete functionality of the OBCS is modelled in Handel-C, a C-based design language, and Celoxica's hardware compiler tool chain is used to directly generate the FPGA configuration file from the Handel-C description. This high level design approach maximally exploits the functional parallelism of a design by mapping the functions of a system onto dedicated parallel operating hardware units.

Figure 1: PCB of one node computer

In general, when designing an OBCS based on COTS components, the design strategy has to be based on the assumption that, because of the single event radiation effects (SEE), failures within the system must not be considered as an exception but, in principle, as normal operation. Such a design principle requires fault tolerance features to be added to the system. Redundancy and parallelism allow for realizing highly dependable systems even if individual components do not fulfil the requirements defined for space qualified technology.

Paying respect to the risks of the COTS approach, the OBCS is realized as a highly redundant replicated four-lane computer system. The four identical computer nodes of the OBCS form a synchronous symmetric

multi computer system synchronously executing in a master/monitor configuration. In principle each single node is able to execute all functions and to perform all control tasks of the OBCS. This implies that all nodes must be connected to all instruments and devices of the satellite. In addition, direct communications can be performed between the nodes to efficiently exchange control and status information.

The main advantages of the four node approach are:

- extended lifetime of the OBCS in the event of a permanent error in the node hardware
- short outage times (higher availability) in the event of SEU induced errors
- redundant computations allow for voting resulting in higher reliability

Due to the SEU sensitivity of the SRAM-based FPGA technology arbitrary errors of the node computer must be expected. The FDIR strategy of the OBCS is based on:

- Error detection by comparing the checksums generated by the replicated nodes
- Error isolation by voting on the checksums and selecting a node which provided the correct checksum to drive the output lines
- Error recovery by enforcing a faulty node to perform the reset/restart procedure and subsequently re-synchronize with the other nodes

Within this straight forward FDIR strategy, error handling mainly relies on the implementation of the robust restart capability. It must be ensured, that after resetting a node, the FPGA can reliably be configured by loading a valid configuration into the configuration memory. This task is performed by a dedicated configuration controller (CC) implemented in a separate FPGA and a radiation hard configuration EEPROM.

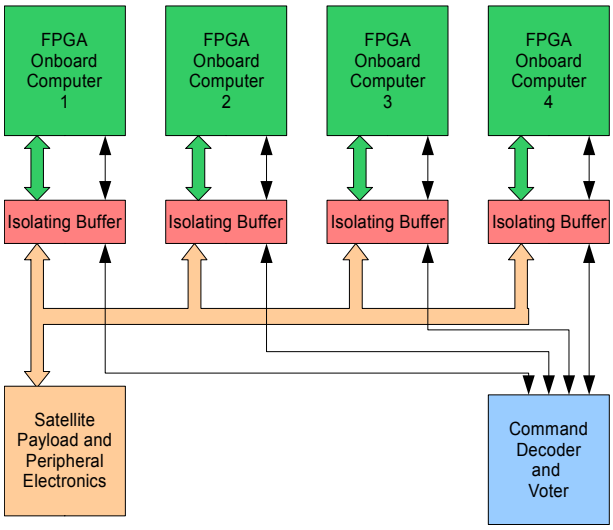

Figure 2: Block diagram of the OBCS

As shown in the block diagram of the OBCS, an additional component called the command decoder and voting unit (CDV) is connected to the four computing nodes. The specific functions of the CDV are essential for the viability of the whole satellite; therefore the CDV is implemented by a space qualified radiation hard FPGA device. The reliability of the CDV is assumed to be sufficiently high, thus no replication is required. The CDV includes the mass memory, provides an interface to the receiver of the up-link, and is performing the following main tasks:

- (CCSDS) decoding of the uplink data stream
- Forwarding of commands to the node computers
- Voting on the checksums, generated by the node computers
- Selecting the master node (based on voting results)
- Enforce reset/recovery or power cycling of a faulty node
- Isolating permanently failing nodes (by power off)
- writing new configuration data into the EEPROMS of the nodes

Replicating the nodes of the OBCS to ensure sufficient high availability requires that each node is able to access the payload instruments and all other devices controlled by the OBCS. Sharing of a device interface requires either dedicated point-to-point connections or a shared bus system. In the OBCS of the “Flying Laptop” the four node computers share the bus lines to connect to the peripherals and dedicated connections are used between the nodes and the CDV. The block diagram of the OBCS in Fig. 2 shows the shared I/O bus to the devices, the dedicated connections between the four nodes, the CDV, and the isolating node interfaces.

The redundant nodes in general are receiving the same digital inputs and are executing the same functions, thus producing the same results. The CDV selects by dedicated control lines one of the four nodes to execute in the master mode. Only the master node is able to drive the output lines to the devices of the satellite. The CDV performs the voting function on checksums generated by all active nodes. For the checksum information of a node can not only be received by the CDV but also by the three other nodes a Byzantine voting scheme may be implemented as an alternative to the centralized voting by the CDV.

Nodes delivering a wrong or no result are assumed to be faulty and are enforced to perform the reset/restart function. If the current master fails to produce the correct checksum, the CDV will select another node as master which will perform the correct output. To avoid accumulation of latent errors in the master node, the CDV may cyclically select a new master while enforcing the old master to execute the reset/restart function.

The implementation of the master/monitor principle within the four node system implies the following states of the nodes that must be determined by the CDV:

- Inactive (power off, isolated from the shared bus
- Master (selected to drive the output lines of the shared I/O buses)
- Monitor synchronized (runs synchronously, provides correct checksums, may be selected as a new master)
- Monitor not synchronized (performing the restart/recovery function, cannot be selected as a new master).

4 NODE COMPUTER

The highly integrated node computer is based on the VIRTEX-II Pro FPGA XC2VP50 of Xilinx. All control and data processing functions of the satellite, which are not directly executed by the CDV, are implemented within this FPGA.

The block diagram in Fig. 3 shows the Virtex-II Pro computing device, the dedicated configuration controller (CC) and the I/O interface logic. The Virtex-II Pro has access to four parallel operating banks of fast SSRAM for intermediate results and to three banks of DDR SDRAM to be used for application data or optionally as program memory of the PowerPC core. Additional memory devices of the node are three banks of NOR flash for multiple versions of the Virtex-II Pro configuration, the radiation hard EEPROM for the configuration of the configuration controller, and a NAND flash memory for permanent storage of application data and optionally of the PowerPC code.

4.1 Node interface

The availability and reliability of the redundant OBCS highly depends on the feature to securely isolate a faulty node so that it cannot obstruct the correct operation of the rest of the system. As shown in the block diagram all I/O connections are implemented by special isolating buffer devices that completely disconnect all interface signals from the shared I/O and communication buses if the CDV switches off the power supply of a node. By this, individual nodes can be switched on and off according to the current mission requirements and also to isolate a faulty node containing a permanent hardware error.

In addition to the isolating feature of the bidirectional buffer devices, control inputs allow to enable or disable the input and output buffers separately. In addition, the output enable control signal of the isolating buffer devices is used to control the output drivers of the nodes such that on shared bus lines only the selected master node is able to drive the output lines. While after power-on the input buffers are always enabled, the output buffers to shared bus lines by default are disabled through pull up/down resistors. Only the output drivers of the master node will be enabled by the CDV via the direct master monitor control lines.

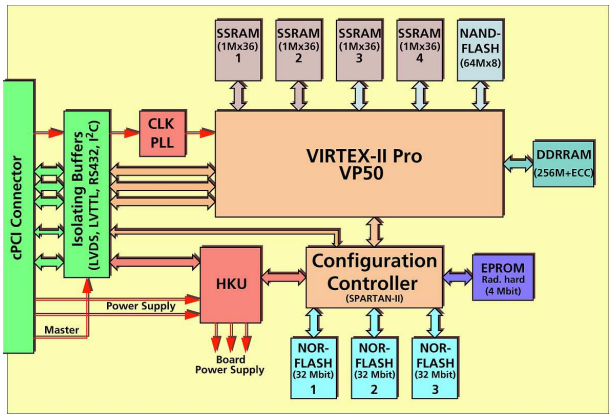

Figure 3: Block diagram of the node computer

4.2 Configuration Controller

In a system-on-chip design based on the Xilinx Virtex-II Pro FPGA, after power-on, first the system itself must be configured by initializing the FPGA. Thus, the initial system configuration must be performed by dedicated hardware. Further, the straight forward FDIR strategy of the OBCS mainly relies on the implementation of a robust restart capability within the nodes. It must be ensured, that after resetting a node the Virtex-II Pro FPGA can reliably be configured by loading a valid configuration into the configuration memory. To realize this vital function of the nodes a dedicated configuration controller (CC) was implemented within in a separate FPGA (Spartan-II). The configuration of the Spartan-II FPGA is assumed to be fixed and is stored in a radiation hard serial EEPROM.

Multiple versions of the configuration file for the Virtex-II Pro FPGA can be held in three physically separate NOR-Flash memory banks. This allows for secure in orbit reconfiguration of the satellite functions. The CDV has access to the flash memories via the CC either to load new versions of configuration data or to restore a corrupted configuration file. In addition, the configuration data can be directly streamed from CDV into the configuration memory of the FPGA. Loading the flash memories under control of the CDV ensures that a node always can be newly configured even if all configuration data in the flash memory has been corrupted. This is true as long as there is no permanent hardware error in the node and the configuration of the CC in the serial EEPROM is not corrupted.

After power-on or RESET, first, the CC is configured from the configuration data in the serial EEPROM.

Then the CC will configure the VIRTEX-II Pro FPGA by transferring the configuration file stored in the memory bank Flash 0. The complete configuration of the node from Flash 0 takes only about 200 ms. If the configuration file stored in Flash 0 is corrupted or a different configuration has to be loaded, the CDV - by appropriate commands - can enforce the configuration from another memory bank or to load the configuration file directly in the streaming mode.

In addition to the CC function for the Virtex-II Pro FPGA the Spartan-II FPGA is used to implement the command and status interface to the CDV, the logic to gather and store the housekeeping data of the node, and an independent watch dog function. The block diagram in Fig. 4 shows the main functions and interfaces of the Spartan-II FPGA.

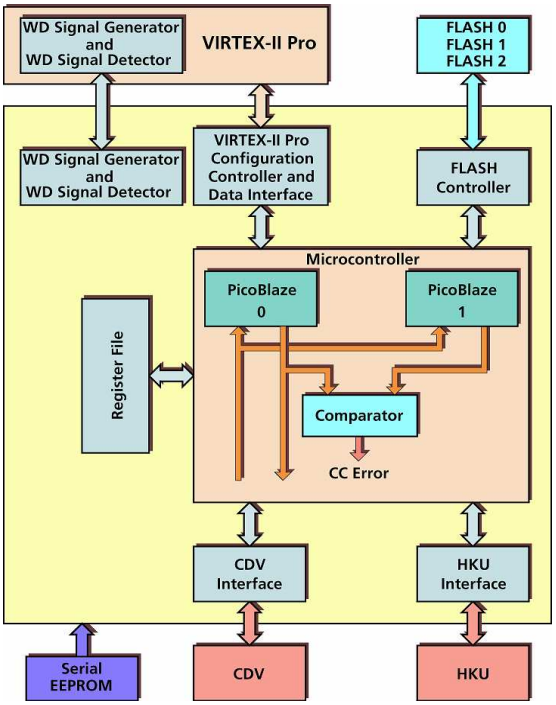

Figure 4: Block diagram of the Spartan-II FPGA

The PicoBlaze Micro-Controller IP core (MC) is used to flexibly implement the different functions within the Spartan-II FPGA. The firmware of the PicoBlaze controller is stored in the Block RAM of the CC FPGA. To improve the reliability of the CC the PicoBlaze controller is duplicated executing in a master/checker configuration. By sharing all input lines and buses, it is enforced that both MCs receive the same input data. By comparing the output data of both controllers, each fault can be detected by a mismatch of any of the output signals. This straightforward mechanism is completely transparent to the firmware and it can be ensured that all possible SEU errors in the PicoBlaze section of the CC FPGA will be detected immediately.

The communication between the CDV and the CC is performed via a serial UART interface by a specific reliable protocol. Further, the CC directly can notify the CDV via a number of dedicated status lines, e.g. to signal successful initialization of the Virtex-II Pro FPGA or to indicate an over-current or over-temperature condition or to inform the CDV about a self-reset of the node.

To flexibly exchange housekeeping data and other control and status information between the CDV and the Virtex-II Pro FPGA the Spartan-II FPGA includes a register file. While the Virtex-II Pro can only perform read operations, the CDV can use specific commands to perform read and write operations on the register file via the CC. Because neither the CDV nor the CC is in-flight reconfigurable a fixed set of commands had to be defined. But by using the communication mechanism via the register file, new commands can still be defined for new versions of the Virtex-II Pro configuration. These additional commands will be transparently forwarded by the CDV and written into specific locations of the register file.

To allow for mutual monitoring of the two FPGAs (Spartan-II and Virtex-II Pro) within the node, a watchdog timer is implemented in each FPGA. The watchdog outputs of both FPGAs are connected to the reset signal of the node. In addition, the CDV is informed about the event of self-reset of a node via a dedicated status signal. The periodic generation of the signal, which retriggers the external watch-dog timer is based on the combination of internal alive signals generated by all the controllers within the FPGA. Such an effective watchdog implementation ensures that all unexpected node failures caused by unspecific errors will result in a watch-dog time out and a subsequent reset and re-initialization of the faulty node.

The different error detection mechanisms implemented within the Spartan-II FPGA are working independent of the general voting mechanism of the CDV. They significantly improve the reliability and availability of the OBCS because serious failures during the configuration of a node are detected immediately and will automatically restart the initialization procedure of a node.

4.3 Main FPGA

The Virtex-II Pro FPGA holds the complete ACS, TM/TC processing, real time image processing of the RG, NIR and TIR cameras and generation of scientific downlink data for the S- and Ka-Band. All these functions use a dedicated individual area within the FPGA and run strictly in parallel. Thus, no race conditions or interrupt latencies exist and allow for the real-time synchronous operation. This is accomplished by a direct implementation of the algorithms in hardware without the need for an operating system using the Handel-C compiler.

All four nodes execute the same algorithms in parallel at a clock cycle precision and in addition to their ACS and scientific output, a hash checksum of the internal state variables is permanently presented via dedicated cross strapped lines to each of the other nodes. Every node is checking the hash codes of the other nodes against its own values and presents the result to the other nodes and the CDV in order to identify a faulty node. In principle, this checking could be performed after every clock cycle but could also be performed only immediately before the new ACS commands are sent out to the actuators.

After a reset, the algorithms in a node resynchronise to the current state vector by reading in the state variables of the other nodes which are also exchanged periodically. After some iterations of the ACS, the node is synchronised again and signals this to the CDV, after which it is again included in the voting process.

In parallel to the ACS, the scientific functions are executed. This is mainly the clocking of the CCD cameras to read in the pixels and a (pre-)processing of the data. This includes the rotation and scaling of the raw images using real-time edge detection and summing of several images in order to achieve a better S/N ratio. The image processing functions make use of pipelined library functions that are are part of the Handel-C compiler and store intermediate results in the SSRAM. Each of the four banks of the SSRAM can be accessed independently of the others at a sustained data rate of 533 Mbytes/s. Thus, while one image is being processed, a new image can already be read in from the cameras allowing to use the cameras at their maximum clock speed. The processed images will either be sent directly to the downlink modems or will be stored in the CDV, where a mass memory will be located. In order to achieve the best possible scientific outcome of the images, only processing that would have to be performed on the ground anyhow before the data can be used will be performed. However, due to the high processing speed of the node, a real-time classification of the pixels could be also performed according to pixel values with an artificial colouring of detected areas.

5 HANDEL-C PROGRAMMING MODEL

The nodes execute their functions directly in hardware without the need for a micro processor or an operating system. This is made possible by the use of the Handel-C programming language which uses a C-like syntax to describe hardware performing complex algorithms. The Handel-C compiler performs the complete synthesis of the design directly from the C source code without the need for RTL programming such as VHDL. The advantage of this direct conversion is that the compiler can perform a global optimisation on the structure and typically shrinks the design by 50% by making use of knowledge of the internal primitives that a certain FPGA provides and by eliminating common sub expressions. Moreover, the compiler can perform a re-timing optimisation in order to allow for higher clock speeds.

The basic properties of the Handel-C compiler are

- All ANSI-C Elements are synthesized
- **One source line** is executed per clock cycle
- Fixed deterministic timing down to clock cycle resolution
- True parallelism
- Any bit width for variables
- External RAM or other H/W included transparently
- Same code is used for PC simulation and synthesis

An additional Platform abstraction layer (PAL) hides hardware specific functions from the user's main code and is pulled in at link time before synthesis. For each dedicated hardware platform, a PAL library functions as a wrapper to the hardware specific implementation (Platform support layer, PSL) and after synthesis eventually evaluates to nothing. Thus, no extra penalty is payed by the insertion of this extra layer. On the hand, the user code in then independent from the underlying hardware and can be compiled directly for different targets including simulation. For the simulation, the design is also fully synthesized but the primitives of the FPGA are then simulated by a normal C program that runs under the control of the compiler. In this mode, the synthesized design can be step-by-step examined on a clock cycle level with full stimulation of external I/O signals.

The speed gain over a software based implementation is enormous, typically a speed gain of 40-100 is achieved. On the other hand, the FPGA could run at a much slower clock speed in order to save power. The currently implemented parts of the ACS include the safe mode and de-tumble mode that, for safety reasons, only make use of the magnetometers and magnetic torquers. Each of the algorithms uses roughly 1.1% of the FPGA's gate equivalences including debug code to feed in simulated orbital propagation data for stimulus and performs 20000 iterations per second, which is well beyond the design cycles of 10/s given the speed of the star tracker cameras.

The expected speed of the main ACS algorithm using the full set of sensors is in the order of 1000-2000 iterations per second, leaving enough time to perform a cross checking of the data with the other nodes before the new actuator commands are sent out to the buses.