In-Flight Reconfigurable FPGA-Based Space Systems

Norma Montealegre, David Merodio, Agustín Fernández and Philippe Armbruster

European Space Agency, European Space Research and Technology Centre
Keplerlaan 1, 2201 AG Noordwijk, The Netherlands
Norma.Montealegre@esa.int

***Abstract*—This paper gives an overview of in-flight reconfigurable FPGA-based space systems. Firstly, an introduction is presented regarding issues of FPGAs in space systems such as: types of FPGAs being used, the increasing use of FPGAs to the detriment of non-programmable devices, the project phases in which FPGAs are being used, the types of FPGA reconfiguration being considered, and the applications of in-flight reconfiguration of FPGAs. Secondly, this paper introduces the architecture of in-flight reconfigurable FPGA-based space systems platforms and a prospective self-repairing multi-FPGA system that uses in-flight reconfiguration for failure recovery. Thirdly, the essentials of the reconfigurable systems of two European space missions to be launched in the near future are explained here, focusing on their in-flight reconfiguration capability.**

I. Introduction

Reconfigurable systems are systems that have the capability of changing their configuration fully or partially in a controlled manner. This capability can be added to electronic systems by means of using field programmable gate arrays, known as FPGAs. An FPGA is a chip with the capability of being configured after manufacturing thanks to its programmable logic [1]. An FPGA can be either: one-time (hard) programmable, using antifuses; or many times (soft) programmable, using memory for storing the configuration of its components.

One-time programmable FPGAs are the most resilient to radiation effects in the space environment because they use antifuses. This is the reason why they are used frequently in the production of electronic boards for space systems [2]. Besides, in the early stage of board development, the FPGA chip can be soldered and its functionality can be programmed later when the circuit design becomes ready. A procedure that makes the development of the electronic board cheaper and faster. For space systems, FPGA vendors offer space-grade FPGAs. One example are the space-grade one-time programmable FPGAs of the RTAX family from Microsemi. The ever increasing use of one-time programmable FPGAs in space systems is taking the market share of non-programmable devices [2], [3]. One-time programmable FPGA chips are used mostly instead of non-programmable Application-Specific Integrated Circuit chips, in short ASICs, for reducing development costs or when the application is not highly demanding in terms of power, speed and size [2].

Many-times programmable FPGAs can be configured more than once, therefore they are also called reconfigurable FPGAS and thus, are the matter of this paper. In space systems, reconfigurable FPGAs can be used during system development before manufacturing the final electronic board, namely for the engineering model; and for the production of the final electronic board, namely for the flight model.

During space systems development, before manufacturing the final electronic board, reconfigurable FPGAs are used in [2]: prototype electronic boards; intermediate boards before producing the final electronic board with a one-time programmable FPGA; in fault injection hardware equipment used for the verification and validation of electronic components for space systems; or in the development phase of Application-Specific Integrated Circuits. An ASIC, similarly to FPGA circuits, is described in a hardware description language such as VHDL or Verilog. Therefore, an ASIC design can be first implemented for debugging and testing using a reconfigurable FPGA, before being manufactured as a chip. When the ASIC design is large and does not fit in the biggest available reconfigurable FPGA chip, it is possible to prototype it in a customized or commercially available multi-FPGA board, e.g., the HAPS board from Synopsys or any other multi-FPGA board from the DINI Group or HiTech Global. However, the use of a multi-FPGA board for ASIC prototyping requires to apply manual or automatic design partitioning techniques [4].

If the spacecraft is planned to carry a reconfigurable FPGA on-board, both the production of the electronic board carrying the reconfigurable FPGA and the design of the circuitry to be programmed in the FPGA follow the phases of a space system project. The phases of a space system project are: Phase 0 (Mission analysis), Phase A (Feasibility), Phase B (Preliminary Definition), Phase C (Detailed Definition), Phase D (Qualification and Production), Phase E (Utilization) and Phase F (Disposal). Considering those phases, a reconfigurable FPGA in the final electronic board can be reconfigured either in phase D, before spacecraft launch (on-ground); or in the phase E, during system operation after spacecraft launch (in-flight). After spacecraft launch, FPGA reconfiguration can be triggered in two ways. Firstly, it can be triggered by an external signal to the FPGA raised either by the on-board

978-1-4673-7501-6/15/$31.00

platform of the spacecraft or by a telecommand coming from the ground station. Secondly, it can be triggered by an internal signal of the FPGA raised by an error detection module, a timer, etc. Finally, regarding the storage place of the different configuration bitstreams for the FPGA, those can be stored on-board in the spacecraft or remotely in the ground station and sent to the spacecraft when required.

At present, static random access memory-based reconfigurable FPGAs and flash memory-based reconfigurable FPGAs are commercially available. Unfortunately, the ionizing radiation present in space has an effect in semiconductor materials and therefore also in FPGAs. SRAM-based FPGAs are very sensitive to ionizing radiation compared with flash-based FPGAs that are more robust to radiation [2]. However, flash-based reconfigurable FPGAs do not support partial reconfiguration to date yet. That is to say, they do not support the reconfiguration of just a portion of the FPGA while the rest of the FPGA is operating. The Total Ionizing Dose tolerance of an FPGA chip is a unit that reflects the level of radiation that an FPGA chip is able to withstand before failing. So, the Total Ionizing Dose tolerance of space-grade reconfigurable FPGAs is much higher than of normal FPGAs. The most recent space-grade reconfigurable FPGAs in the market are: the Xilinx Virtex-4QV, the Xilinx Virtex-5QV and the Atmel ATF280E, which are SRAM-based reconfigurable FPGAs; and the Microsemi RTProASIC4, which is a recently announced flash-based reconfigurable FPGA. In addition, radiation mitigation techniques used during system development help to harden FPGA designs against radiation such as: the introduction of error detection and correction codes in memories, the use of manual redundancy techniques in the design such us double or triple modular redundancy, and the use of automatic insertion of redundancy with the tools XTMR and RoRA [2].

Reconfiguration of FPGAs can be useful for: adapting the behavior of the FPGA circuit, transforming the function of the FPGA circuit, or correcting errors in the FPGA circuit. Adapting the behavior or transforming the function of the FPGA circuit can be used before spacecraft launch (on-ground) when there is a change or addition of requirements, or after spacecraft launch (in-flight) when there are changes in the space environment and new user demands arise. Correcting errors in a circuit can be necessary before spacecraft launch (on-ground) when design errors are detected or after spacecraft launch (in-flight) when persistent radiation effects, wear-out or silicon defects are present.

Reconfigurable FPGAs are attractive in the space sector due to their flexibility and high performance, but still they are susceptible to radiation. Thus, they are considered suitable mostly for non-mission critical parts in the payload rather than in the on-board platform. Their usual application is for data processing. For instance, in on-board signal processors of telecommunication satellites or in sensor data processing modules inside instruments of science missions. Thereby, in-flight reconfiguration of reconfigurable FPGAs is useful for algorithm upgrading or performance optimization of those data processing modules. One example of a mission which is planning to use in-flight reconfiguration of the reconfigurable SRAM-based FPGAs in its reconfigurable on-board processor is the Heinrich Hertz Satellite Mission, scheduled to be launched in 2017 in a Geostationary Earth Orbit. In-flight reconfiguration of FPGAs can also serve for power and resources reduction by means of time-space partitioning of the processing algorithm. One mission working on the implementation of such paradigm is the Polarimetric and Helioseismic Imager of the Solar Orbiter Mission to be launched in 2018. Furthermore, in-flight reconfiguration of FPGAs is also attractive for fault tolerance issues like error correction. Periodical full reconfiguration for correcting errors in the configuration memory of reconfigurable SRAM-based FPGAs, namely scrubbing, is a technique already known and used in most missions that include reconfigurable SRAM-based FPGAs. Partial reconfiguration for correcting errors is a technique which nowadays can be implemented using for instance the Soft Error Mitigation Controller IP Core from Xilinx [5] for detecting bit flips caused by radiation, better known as Single Event Upsets (SEUs), in the configuration memory of the FPGA.

Radiation-hardened, space-grade, SRAM-based, reconfigurable FPGAs have been used already in space missions. For example, regarding satellites, the Australian micro-satellite FedSat, launched in 2002 in a Sun-synchronous circular orbit (altitude 780 km), carried a Xilinx XQR4036XL in its adaptive instrument module. That module was designed with the ability of performing reconfiguration of the FPGA circuitry while the spacecraft is in-flight. That was the first time that this ability was introduced in a space mission [6]. The CFESat satellite, launched in 2007 in a Low Earth Orbit (altitude 560 km), was equipped with nine Xilinx Virtex XQVR1000 FPGAs in three reconfigurable computer modules used to perform signal processing experiments and SEU detection experiments [7]. Examples regarding science missions to Mars are: the Mars Exploration Rovers Discovery and Spirit, launched in 2003, which used Xilinx XQRV1000 FPGAs to control the motors of the wheels, steering, antennas, cameras and other instruments. Besides, its lander used the Xilinx XQR4062XL FPGAs for controlling the pyrotechnics during descending and landing on the surface of Mars [8]. The Mars Reconnaissance Orbiter, launched in 2005, had a radiation-hardened Xilinx Virtex 300E FPGA in its CCD Processing and Memory Module to perform control, signal processing and data compression [9]. The Mars Science Lab with its rover Curiosity, launched in 2011, used radiation-hardened Xilinx Virtex-II FPGAs for their image processing pipelines [10]. Those were the first most prominent missions that used reconfigurable FPGAs. In the meanwhile commercial reconfigurable FPGA platforms for space systems based on Xilinx Virtex-4 and Virtex-5 FPGAs appeared on the market such as the RA-RCC, used in the LEO satellite Tac-Sat-3, and the Proton-3, without mentioning some other commercial platforms already available for CubeSats [11].

Regarding European science missions: the joint mission of the NASA and the German Aerospace Center named Gravity Recovery and Climate Experiment (GRACE) and composed

of two 220 km apart twin satellites, launched in 2002 in a polar orbit (altitude 500 Km), has been the first mission that carried a reconfigurable FPGA for sensor data processing, the Xilinx XQR4036XL FPGAs [12]. The first Venus exploration mission of the European Space Agency named Venus Express, launched in 2005, used a radiation-tolerant Xilinx Virtex-1 FPGA for its monitoring camera [13]. That instrument has been developed by the University of Braunschweig in Germany. However, it has not been designed to perform in-flight reconfiguration. Thereafter the capability for in-flight reconfiguration has been added to that design and it evolved into the reconfigurable FPGA platform which is being used now for the instrument PHI of the Solar Orbiter mission [13]. That platform has also been considered to be used in future missions such as the Proba 3 [11].

In-flight reconfigurable FPGA-based space systems require hardware, firmware and software support. In the following sections, the architecture of platforms that support in-flight reconfigurable FPGA-based systems for space applications, i.e, the one mentioned in the last paragraph, developed by research institutes and industry in cooperation with the European Space Agency will be introduced. Thereafter, the two above mentioned space missions to be launched in the near future, namely the Heinrich Hertz Satellite Mission and the Solar Orbiter Mission, which will include support for in-flight reconfiguration of its FPGA-based reconfigurable system, are going to be introduced focusing on their design for in-flight reconfiguration.

II. Platforms for developing in-flight reconfigurable FPGA-based space systems

In this section, firstly, the architecture of a regenerative on-board processor demonstrator for software defined radio in telecommunication satellites is presented briefly. Secondly, the essentials of the two versions of the project with the title Dynamic Reconfigurable Processing Module is introduced shortly. The aim of that project has been to raise the Technology Readiness Level of in-flight dynamic partial reconfiguration of SRAM-based reconfigurable FPGAs for on-board data processing by building technology demonstrators. Thirdly, the extension of one of those platforms with a framework for adaptive redundancy, developed in the frame of a doctoral thesis, is introduced. At the end, a failure recovery technique in a multi-FPGA system using in-flight reconfiguration, which is being investigated at ESA, is presented.

A. Regenerative On-Board Processor

A Regenerative On-Board Processor prototype for Software Defined Radio has been developed by Thales Alenia Space, Space Engineering and ULISSE under ESA contract. Its main objective has been to build a demonstrator of a signal processing system for telecommunication satellites able to be reconfigured when a change is demanded. It consists of: two Reconfigurable Signal Processing Chains, a Reconfiguration Controller, a Default Configuration Storage, and a Dynamic Configuration Storage. One of the Reconfigurable Signal Processing Chain is active while the other is in standby mode, switching states when reconfiguration is required to avoid the interruption of the communication service. The Reconfiguration Controller is in charge of: the acquisition of an encrypted reconfiguration bitstream from a certified ground station, its storage in the Dynamic Configuration Storage, its integrity verification by means of CRC, the reconfiguration process itself, and the switching of processing chains [14]. This prototype has been tested in laboratory successfully.

B. Dynamically Reconfigurable Processing Module 1

In the frame of a Technology Readiness Level project, a demonstrator for the Dynamically Reconfigurable Processing Module has been developed by TWT, Bielefeld University, Swiss Space Tech and Politecnico di Torino, under ESA contract [15]. That platform is based on the RAPTOR-X64 modular rapid-prototyping baseboard and the daughterboard DB-V4 developed by the University of Bielefeld [16], together with the DB-SPACE daughterboard specially developed for the platform also by the University of Bielefeld. The RAPTOR-X64 board comprises a PCI-X and a USB interfaces for communication with a computer. The DB-V4 daughterboard is a module for the RAPTOR-X64 board, which hosts 4 GB RAM Memory and a Xilinx Virtex-4 FX100 FPGA with a system-on-chip infrastructure for self-configuration that uses the ICAP internal configuration port of the FPGA. The DB-SPACE daughterboard is a module which hosts: an Atmel AT7913E chip with a LEON2FT processor as system controller; a Xilinx Spartan 6 XC6SLX100 FPGA for implementing all interfaces for external communication; another Xilinx Spartan 6 XC6SLX100 FPGA for implementing the infrastructure necessary for communicating the FPGA in charge of external communication and the Partial Reconfigurable FPGAs in the DB-V4 modules; and a Xilinx XC2C384 CoolRunner-II CPLD as a reconfiguration controller for all FPGAs in the platform. A Partial Reconfigurable FPGA in a DB-V4 daughterboard uses a CoreConnect Processor Local Bus (PLB) for internal communication. Differently, the FPGAs in the DB-SPACE daughterboard use an AMBA Advanced Extensible Interface 4 (AXI4) for internal communication [17].

This platform has the advantage of having all external interfaces implemented in only one place, namely the FPGA is charge of the external communication. The designer can configure only the required interfaces and only those which fit in the FPGA. The inter-FPGA communication in the platform is a kind of ad-hoc bus-based communication. And there is only one partial reconfigurable region in the Partial Reconfigurable FPGA of the DB-V4 module, which is divided in fine-granular homogeneous tiles that communicate using a Wishbone bus-based inter-tile communication infrastructure. This approach requires special placing and routing tools that care about the homogeneity of the tiles.

C. Dynamically Reconfigurable Processing Module 2

This version of the Dynamically Reconfigurable Processing Module consists of a complete platform concept for sensor

data processing with support for in-flight reconfiguration developed by Astrium Ltd and the Institute of Computer and Network Engineering at the Technical University of Braunschweig in cooperation with the European Space Agency [18].

Basically, it consists of three components [11]: one or more Dynamically reconfigurable FPGA (DFPGA) modules, a SpaceWire router, and a system controller. A DFPGA module is a module that can contain: two or more SRAM-based reconfigurable FPGAs placed in separate piggyback boards for an easy change of FPGA model, and a radiation hardened one-time programmable FPGA placed in the main board and in charge of the configuration of the reconfigurable FPGAs in the module. The SpaceWire router is used for connecting one or more of those DFPGA modules, making the system scalable. The system controller has the function of running the main processing software, triggering fully or partial reconfiguration of all the reconfigurable FPGAs in the DFPGA modules, and establishing communication with the spacecraft.

The reconfigurable FPGAs in the DFPGA module are able to implement data processing using partial reconfigurable modules. Those reconfigurable modules are connected using SoCWire. SoCWire is a parallel and synchronous version of the serial asynchronous SpaceWire interface, which has been conceived at the Institute of Computer and Network Engineering at the Technical University of Braunschweig [19]. It allows on-chip communication (intra-FPGA communication) inside the one-time programmable FPGA and the reconfigurable FPGAs by using SoCWire switches and interfaces. Furthermore, off-chip communication (inter-FPGA communication) among all the reconfigurable FPGAs in the DFPGA module and the one-time programmable FPGA is also possible, for example by connecting the SoCWire switches of the reconfigurable FPGAs to the SoCWire switch in the one-time programmable FPGA.

RAM memory is attached to the reconfigurable FPGAs for temporal storage during processing. A bigger amount of highly reliable RAM memory is attached to the configuration controller for: buffering acquired sensor data for later processing, and for storing the actual configuration bitstreams to use them, for instance, to reconfigure the FPGA configuration memory periodically in order to avoid SEUs. That memory can be accessed directly from any reconfigurable module. Besides, non-volatile memory is attached to the system controller for storing all fully or partial configuration bitstreams.

The firmware in the configuration controller FPGA is based on IP cores from the GRLIB IP library [20]. So, a LEON3 processor core running at 20 MHz, a CAN and a SpaceWire interfaces are connected by an AMBA Advanced High-performance Bus (AHB). The low speed GPIOs, RS422 and SPI interfaces are connected through an AMBA Advanced Peripheral Bus (APB). Both buses are connected through an AHB/APB bridge. The controller for the memory that buffers acquisition data is isolated from the microprocessor using another AHB bus, connected through an AHB/AHB bridge running at twice the frequency of the processor. The SoCWire switch is connected also through AHB/SoCWire bridges to the last AHB bus, making a fast data transfer possible from and to the reconfigurable modules.

A demonstrator of the DRPM has been developed using: an Atmel AT7913E SpaceWire Remote Terminal Controller, which is provided with a LEON2FT processor; a SpaceWire router built with an Atmel AT910E SpaceWire router chip; and a DFPGA module with two SRAM-based Xilinx Virtex-4 FPGAs and a flash-based Microsemi ProASIC3 FPGA, to be replaced with a compatible one-time programmable Microsemi RTAX FPGA in a flight model [21].

This platform is being employed for the PHI instrument of the Solar Orbiter mission, which is explained in section III-B.

D. Adaptive redundancy management in the Dynamically Reconfigurable Processing Module 2

If a processing unit is implemented as exactly one partial reconfigurable module, redundancy at that level, such as duplication or triplication can be managed by the system controller. However, in the Dynamically Reconfigurable Processing Module developed by Astrium Ltd and the Institute of Computer and Network Engineering at the Technical University of Braunschweig, processing data can be accessed directly by the partial reconfigurable modules, overpassing the system controller. Therefore, failure detection and redundancy management can be better implemented somewhere outside the partially reconfigurable modules. This issue is addressed in [22], where it is proposed to embed a comparator, in case of duplication, or a voter, in case of triplication of the processing module, inside one of the SoCWire switches that serves for intra- and inter-FPGA communication. More precisely, between the SoCWire interfaces at the ports of the SoCWire switch and the routing matrix in that switch. At that place, a so called Module Switch Matrix is added for rerouting from any port to a comparator/voter. Besides, an extra Routing Table is added for routing the output data of the comparator/voter to a desired output port. The Module Switch Matrix and the Routing Table should be re-programmed each time a new redundancy configuration is implemented. For that, a Configuration Port is also added to the SoCWire switch, which, besides serving for re-programming those modules, it serves for reporting a failure to the system controller in order to trigger scrubbing of the faulty reconfigurable module. This research is being executed in the context of a doctoral thesis at the European Space Agency and the University of Leicester. The intention of the work is to switch from a configuration that implements a type of redundancy to another, and that in-flight, in order to save power or increase the availability of the system. The use of comparators or voters in the routing switches makes easier to add or to remove redundant processing units.

E. Prospective self-repairing multi-FPGA system

Support for reconfiguration in a system provides with the capability to make changes in the design in the future. Support for dynamic partial reconfiguration may help to fit a big design in a smaller number of FPGAs by using time-space partitioning. And, having multiple reconfigurable FPGAs may

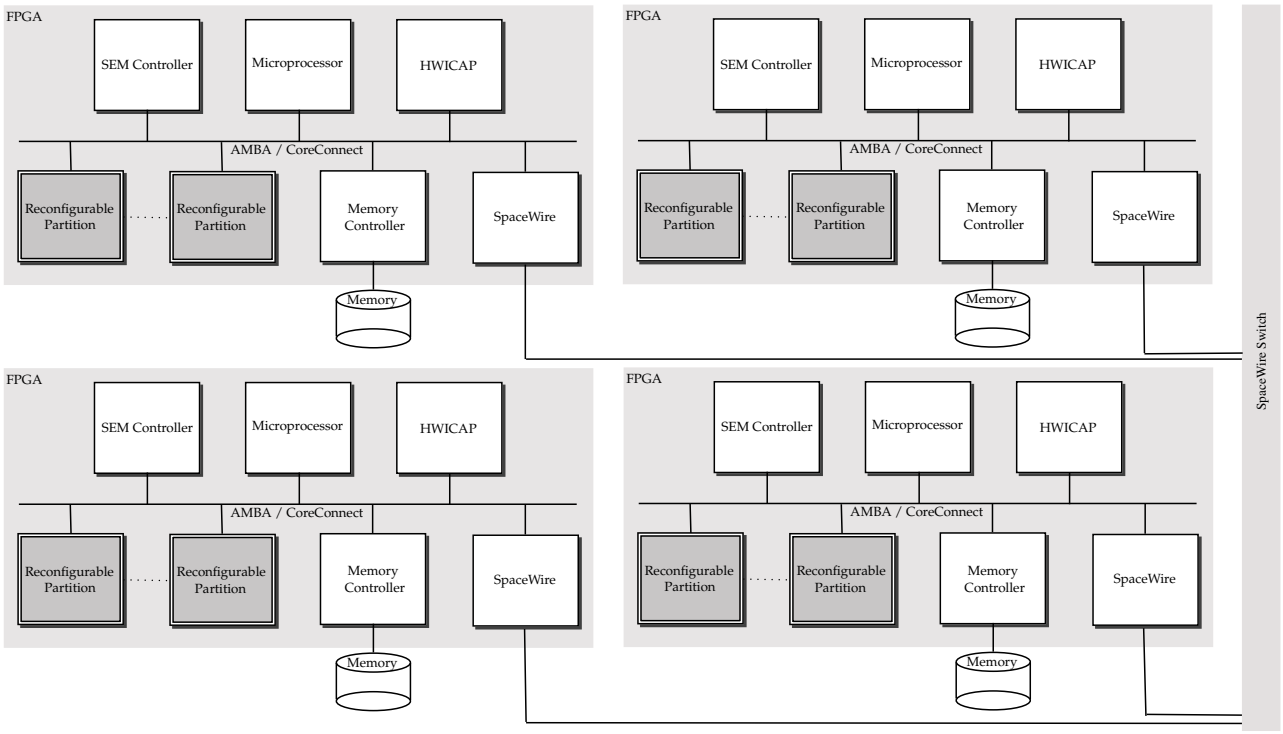

Fig. 1. Architecture of the self-repairing multi-FPGA prototyping platform

help to recover a system from a failure by reconfiguring the affected module in a neighboring FPGA, if enough space is available and the application allows it.

Recovering an FPGA-based system from Single Event Upsets happening in the configuration memory of the FPGA due to radiation, requires to reconfigure the affected FPGA with the same configuration bitstream either partially or fully. However, this procedure does not help when the detected fault appears to be permanent. That is to say, when the same error is detected after reconfiguring the FPGA a couple of times. In such a case, one possibility is to move the affected hardware module in the affected FPGA to a neighboring FPGA. However, that procedure only works when there exists a neighboring FPGA with enough space available for the affected hardware module. Searching for an FPGA with space available for the affected module is not a straightforward problem. To find a solution to that problem, the support of the system architecture and its components is necessary. Consequently, an investigation about the necessary technical steps for having a multi-FPGA system that supports the recovery of the system from a permanent fault is being conducted in the Data Systems Section at ESA. The issues being considered in that study are presented below.

Firstly, a system that relies on reconfiguration for recovering from a fault, should assure that the reconfiguration controller is reliable and will do its work when required. For that, the reconfiguration controller of a reconfigurable FPGA can be implemented in an external one-time programmable FPGA. Another alternative is to implement the reconfiguration controller in the same reconfigurable FPGA. Most probably using the Internal Configuration Access Port available in Xilinx FPGAs, but with the requirement of employing radiation mitigation techniques in the implementation of the reconfiguration controller. The last one is a less reliable alternative in comparison with the implementation of the external reconfiguration controller, considering a reconfigurable FPGA is radiation prone and multiple permanent faults could be present in the reconfigurable FPGA device which might affect the reconfiguration controller as well. Nevertheless, an internal reconfiguration controller allows a simplified architecture, which might be convenient when: it is the goal to have a self-repairing FPGA device; or in the prototyping phase of a system, as the external FPGA or component for implementing the reconfiguration controller can be disregarded.

Secondly, searching for an FPGA with space available for the affected hardware module would imply to have the knowledge of the resource allocation in all reconfigurable FPGAs. That knowledge can be stored and managed by a central system controller or it can be acquired on demand by the affected FPGA from each one of the other reconfigurable FPGAs. Thus, it is important to have inter-FPGA communication, whether directly among FPGAs or through a central system controller. In both cases, a communication interface such as SpaceWire or SoCWire can be used together with the necessary on-chip or off-chip switches.

Thirdly, hardware modules are inhomogeneous regarding resource consumption. To be able to find an available area in a neighboring FPGA might demand to have different predefined design configurations with inhomogeneous hardware modules

that take into account the place where a permanent fault might come into. An alternative would be to create a design with homogeneous hardware modules right from the beginning and having the partial bitstreams for all possible configurations stored somewhere, while taking into account the following. Considering that all reconfigurable FPGAs are equal, the same partial bitstream of a hardware module can be used in a neighboring FPGA, if the area available is exactly in the same place. However, when the module has to be reconfigured in a different place, a partial bitstream for that area necessarily has to be available.

Finally, if a hardware module requires the use of external inputs/outputs, that requirement should be considered and managed in the way that the access to those external inputs/outputs is still possible after the hardware module is reconfigured in a neighboring FPGA. That is difficult to achieve. However, one alternative may be to avoid having direct access to external inputs/outputs from the reconfigurable modules in the reconfigurable FPGAs. Instead, external inputs/outputs can be connected to the reconfigurable hardware module through a switch and communication interfaces, e.g., SoCWire.

Multi-FPGA platforms may have several architectures. Some important architectural parameters to consider at the time of prototyping a multi-FPGA system able to recover from a permanent fault are: the placement and type of the system controller, the type of the inter-FPGA communication interface, the placement of the router or switch in the inter-FPGA network, and the placement and type of the reconfiguration controller. Regarding the system controller, it may be implemented with a microprocessor as a standalone chip or as a core inside a radiation hardened one-time programmable FPGA. Another possibility is to have a distributed system control implemented for example with a microprocessor in each one of the reconfigurable FPGAs. The inter-FPGA communication interface may be implemented using different communication interfaces, e.g., SpaceWire or SoCWire. The respective switch or router for the FPGA network may be placed externally as a standalone component, be implemented as a core in the device that hosts the system controller or in the device that hosts the reconfiguration controller, or be implemented in each reconfigurable FPGA. Finally, the reconfiguration controller may be placed either internally using the ICAP configuration interface in the same reconfigurable FPGA, or externally using the SelectMAP configuration interface in an extra one-time programmable FPGA or in the device that hosts the system controller.

For the implementation of a demonstration and experimentation prototype, two multi-FPGA hardware platforms are being taken into account: several FPGA prototyping boards connected together through a switch, and the multi-FPGA board HAPS from Synopsys. The first architecture being considered at the moment and shown in figure 1 is a multi-FPGA system with: a distributed system control, an inter-FPGA communication using the SpaceWire interface, a standalone SpaceWire switch for connecting the FPGAs, and reconfiguration controllers implemented in each FPGA using the ICAP interface. The advantage of such an architecture is its high scalability. If implemented with single-FPGA prototyping boards, a new one can be added anytime, with the only restriction of having a port available in the standalone switch at that moment. The second architecture being considered is a multi-FPGA system having: only one system controller implemented in one of the reconfigurable FPGAs, an inter-FPGA communication using the SoCWire interface, a switch implemented together with the system controller, and reconfigurable controllers in each FPGA using ICAP. The interest in such an architecture is to have a hardware accelerating platform, similar to the Dynamic Reconfigurable Processing Module used in the PHI instrument of the Solar Orbiter, but with centralized Input/Output placed next to the system controller. That might allow to simplify the problem of communication of the reconfigurable hardware modules in the reconfigurable FPGAs with the external world. A similar architecture is being investigated in the High Performance COTS Based Computer Architecture project at the European Space Agency, which is an architecture for data processing in space systems. That architecture comprises multiple digital signal processors for computation acceleration, instead of FPGAs, and a system controller implemented in a radiation hardened one-time programmable FPGA called SmartIO because it executes a smart I/O management [23].

III. Space missions that integrate in-flight reconfigurable FPGA-based systems

European space missions, mentioned in section I, that make use of in-flight reconfiguration and that will be launched in the near future, are described in the following subsections.

A. Fraunhofer On-Board Processor of the Heinrich Hertz satellite mission

The Heinrich Hertz communications satellite, in short H2Sat, is a German satellite that is planned to be launched in 2018 in a Geosynchronous Equatorial Orbit and to have an expected life of 15 years [24]. Its goal is to verify new hardware, software and communication technologies in-orbit in order to cope with the constant change of telecommunication standards. The satellite will include a regenerative transponder implemented with a reconfigurable on-board processor. First, a regenerative transponder is a system that gathers signals over a range of uplink frequencies and after regenerating them it retransmits them on a different set of downlink frequencies to receivers on Earth. Second, a reconfigurable on-board processor is an in-flight reconfigurable FPGA-based module that serves for the regeneration of the uplink signal. Then, the regenerative transponder implemented with a reconfigurable on-board processor will be capable of being updated in-flight when a new telecommunication standard is released. That just by updating the configuration memory of the FPGAs with new configuration bitstreams sent from the ground station to the satellite over a so called virtual Telemetry/Telecommand link, in short vTM/TC [25]. It is also intended the updating of the configuration memory of the FPGAs for adapting the implemented redundancy according to the radiation level,

		Regenerative On-Board Processor	DRPM 1	DRPM 2	Adaptive redundancy framework	Self-repairing multi-FPGA system
Application	Software Defined Radio	x				
	Sensor Data Processing		x	x	x	
	Fault Tolerance				x	x
Characteristic	Reconfiguration Controller	External	Internal, External	External	DRPM 2	Internal, External
	Time-Space Partitioning	-	Capable	Capable	DRPM 2	Capable
	I/Os	Centralized	Centralized	Distributed	DRPM 2	Centralized
	Architecture	Two processing chains	One FPGA for I/Os	Antifuse-based FPGA as reconfig. controller	SoCWire switch extension	Multi-FPGA

TABLE I
FUNDAMENTAL DIFFERENCES AMONG THE PRESENTED IN-FLIGHT RECONFIGURABLE FPGA-BASED PLATFORMS

		H2Sat	Solar Orbiter
Application	Software Defined Radio	x	
	Sensor Data Processing		x
	Fault Tolerance	x	
Characteristic	Orbit	GEO	HEO
	Reconfiguration Controller	Internal, External	External
	Time-Space Partitioning	-	x
	I/Os	Centralized	Distributed
	Architecture	Master and Slave FPGAs	DRPM 2

TABLE II
FUNDAMENTAL DIFFERENCES AMONG THE PRESENTED IN-FLIGHT RECONFIGURABLE FPGA-BASED SYSTEMS OF CURRENT MISSIONS

measured by counting the number of Single Event Upsets present in the Block RAM memory of one of the FPGAs [26].

For this satellite, the Fraunhofer Institute for Integrated circuits and the Friedrich-Alexander-Universitaet Erlangen-Nuenberg in Germany are developing the Fraunhofer On-Board Processor using space-grade SRAM-based Xilinx Virtex-5QV FPGAs[27]. The Fraunhofer On-Board Processor presented in [28] is composed of four hardware modules: a Radio Frequency Card, a Power Supply Unit, and two DSP cards. Each DSP card has an Analog to Digital Converter, an FPGA and a Digital to Analog Converter. The FPGA in charge of the 36 MHz narrowband acts as master and the FPGA in charge of the 450 MHz broadband focuses on the broadband digital signal processing and acts as slave. After power up, both FPGAs are configured with initial bitstreams stored in a non-volatile magnetoresistive RAM by an external processor. The initial configuration of the master FPGA includes a System-On-Chip which is placed in a static area and contains a LEON3FT processor together with some cores of the GRLIB IP Library. With the initial configuration, the master FPGA is capable of executing partial reconfiguration of itself and of triggering partial or full reconfiguration of the slave FPGA. The master FPGA is also capable of scrubbing the configuration memory of the slave FPGA against SEUs. Besides, it is in charge of the vTM/TC link, which is a link multiplexed with the normal user uplink, and of executing radiation level measurement using one of its BRAM blocks.

B. Polarimetric and Helioseismic Imager instrument of the Solar Orbiter mission

The Solar Orbiter is a science mission to be launched in 2017 that will observe the polar regions of the Sun and conduct measurements of the region of space dominated by the Sun called Heliosphere [29]. One of its instruments is the Polarimetric and Helioseismic Imager which is based on the Venus Monitoring Camera from the Venus Express mission. It contains two sensors with 2048 x 2048 14 bit pixel each, which will provide a large amount of data, of approximately 3,2 Gbits/min/dataset each. The telemetry link is only able to send approximately 100 Mbits/dataset to the Earth. Therefore, it has become necessary to reduce the data to be sent to the ground station by moving the processing of the raw data from the ground station to the spacecraft, on-board, using a digital processing unit (DPU), which furthermore is required to be flexible to changes after the spacecraft has been launched.

That DPU is being developed by the Institute of Computer and Network Engineering at the Technical University of Braunschweig. The unit will execute the following processing tasks: image stabilization, data acquisition, data preprocessing and compression, and Radiative Transfer Equation inversion. The required processing capacity and flexibility to changes will be achieved by using a microprocessor and reconfigurable FPGAs. The microprocessor is for controlling the processing flow and the FPGAs for accelerating the data processing by hosting the four processing modules. It has been decided to use only two SRAM-based reconfigurable FPGAs for implementing the four processing modules in a time-space partitioning manner using partial reconfiguration and two configuration modes, in order not to overpass the budget of power and heat dissipation allocated for the instrument. This implementation requires buffering all the acquired data for further use in the data processing mode. With such an implementation, it is the first time that a scientific space instrument employs a reconfigurable FPGA-based platform that will make use of in-

flight partial reconfiguration for resources sharing [11].

The Dynamically Reconfigurable Processing Module architecture, presented in section II-C is being employed for implementing that DPU. The hardware hosts one DFPGA module with two reconfigurable Xilinx Virtex-4 XQR4VSX55 FPGAs and a one-time programmable Microsemi RTAX2000SL FPGA [30]. The controller of the system is being implemented using an ASIC chip, the GR712RC dual-core LEON3FT processor, which is able to communicate with the Solar Orbiter platform through a SpaceWire interface. For receiving, storing and verifying a new bitstream coming from the ground station, the On-board Command Language (OCL) will be used, which is a software framework already employed in some past space missions for software updating [30].

IV. Conclusion

This paper presented several issues regarding in-flight reconfigurable FPGA-based systems in conjunction with development platforms and examples of space missions including such systems. Tables I and II summarize the main characteristics of the presented platforms and missions. The European Space Agency has been working together with the industry and research institutes in the area of reconfigurable space systems. From that work, the Dynamic Partial Reconfiguration Module has been originated. Using that platform, some further research is being conducted in the field of fault detection, isolation and recovery. One example is the development of a framework for applying adaptive redundancy, or the use of reconfiguration for the recovery of failures in multi-FPGA systems, which show the applicability of in-flight reconfiguration of FPGA-based system also for fault tolerance in space systems.

References

[1] J. Astola and R. Stankovic, *Fundamentals of Switching Theory and Logic Design - A Hands on Approach*. Springer, 2006.

[2] A. F. León, "European Space Technology Harmonisation Technical Dossier - Microelectronics: ASIC and FPGA," European Space Agency, Tech. Rep., January 2012.

[3] R. B. Gardenyes, "Trends and patterns in ASIC and FPGA use in space missions and impact in technology roadmaps of the European Space Agency," Master's thesis, TU Delft, 15 August 2012.

[4] D. Amos, A. Leasea, and R. Richer, *FPGA-Based Prototyping Methodology Manual - Best Practices in Design-For-Prototyping*. Synopsys Press, 2011.

[5] Xilinx, "LogiCORE IP Soft Error Mitigation Controller v4.1 - Product Guide," Xilinx, Tech. Rep., 2 April 2014.

[6] eoPortal Directory, "FedSat," January 2015, directory.eoportal.org/web/eoportal/satellite-missions/f/fedsat.

[7] M. Caffrey, K. Morgan, D. Roussel-Dupre, S. Robinson, A. Nelson, A. Salazar, M. Wirthlin, W. Howes, and D. Richins, "On-Orbit Flight Results from the Reconfigurable Cibola Flight Experiment Satellite (CFESat)," in *Symposium on Field Programmable Custom Computing Machines*. IEEE, April 2009, pp. 3–10.

[8] D. Ratter, "FPGAs on Mars," *Xcell journal*, vol. 50, Fall 2004.

[9] A. S. McEwen, E. M. Eliason, J. W. Bergstrom, N. T. Bridges, C. J. Hansen, W. A. Delamere, J. A. Grant, V. C. Gulick, K. E. Herkenhoff, L. Keszthelyi, R. L. Kirk, M. T. Mellon, S. W. Squyres, N. Thomas, and C. M. Weitz, "Mars Reconnaissance Orbiter's High Resolution Imaging Science Experiment (HiRISE)," *Journal of Geophysical Research: Planets*, vol. 112, no. E5, 2007.

[10] Xilinx, "Mars Curiosity Rovers MAHLI images a dusty penny on Mars with 14 m resolution," 24 June 2014, http://forums.xilinx.com/t5/Xcell-Daily-Blog/Xilinx-of-Mars-Happy-First-Martian-Anniversary-Curiosity/ba-p/480750.

[11] F. Bubenhagen, B. Fiethe, T. Lange, H. Michalik, and H. Michel, "Reconfigurable platforms for Data Processing on scientific space instruments," in *NASA/ESA Conference on Adaptive Hardware and Systems (AHS)*. IEEE, June 2013, pp. 63–70.

[12] K. Manthey, D. Krutz, and B. Juurlink, "A new real-time system for image compression on-board satellites," *On-Board Payload Data Compression Workshop*, 25 - 24 October 2014.

[13] B. Osterloh, H. Michalik, B. Fiethe, and F. Bubenhagen, "Enhancements of reconfigurable System-on-Chip Data Processing Units for Space Application," *NASA/ESA Conference on Adaptive Hardware and Systems (AHS)*, pp. 258–262, August 2007.

[14] C. Morlet, F. Iacomacci, F. Autelitano, and F. Quaranta, "Reconfigurable implementation for On-Board digital Processors," in *International Workshop on Signal Processing for Space Communications (SPSS)*. IEEE, October 2008, pp. 1–7.

[15] European Space Agency, "Dynamically Reconfigurable Processing Module," January 2015, www.esa.int/Our_Activities/Space_Engineering_Technology/Onboard_Data_Processing/DRPM_-_Dynamically_Reconfigurable_Processing_Module.

[16] University of Bielefeld, "RAPTOR Family," January 2015, www.ks.cit-ec.uni-bielefeld.de/en/projects/raptor-family.html.

[17] J. Hagemeyer, A. Hilgenstein, D. Jungewelter, D. Cozzi, C. Felicetti, U. Rueckert, S. Korf, M. Koester, F. Margaglia, M. Porrmann, F. Dittmann, M. Ditze, J. Harris, L. Sterpone, and J. Ilstad, "A Scalable Platform for Run-time Reconfigurable Satellite Payload Processing," in *NASA/ESA Conference on Adaptive Hardware and Systems (AHS)*. IEEE, June 2012, pp. 9–16.

[18] European Space Agency, "Dynamically Reconfigurable Processing Module," January 2015, www.esa.int/Our_Activities/Space_Engineering_Technology/Onboard_Data_Processing/Dynamically_Reconfigurable_Processing_Module_DRPM.

[19] B. Osterloh, H. Michalik, B. Fiethe, and K. Kotarowski, "SoCWire: A Network-on-Chip Approach for Reconfigurable System-on-Chip Designs in Space Applications," in *NASA/ESA Conference on Adaptive Hardware and Systems (AHS)*. IEEE, June 2008, pp. 51–56.

[20] "GRLIB," January 2015, www.gaisler.com/index.php/downloads/leongrlib.

[21] F. Bubenhagen, B. Fiethe, H. Michalik, B. Osterloh, P. Norridge, W. Sullivan, C. Topping, and J. Ilstad, "Enhanced Dynamic Reconfigurable Processing Module for Future Space Applications," in *International SpaceWire Conference*, June 2010, pp. 475–482.

[22] F. Siegle, T. Vladimirova, O. Emam, and J. Ilstad, "Adaptive FDIR framework for payload data processing systems using reconfigurable FPGAs," in *NASA/ESA Conference on Adaptive Hardware and Systems (AHS)*. IEEE, June 2013, pp. 15–22.

[23] M. Patte, R. Grimoldi, and R. Trautner, "High Performance COTS Based Computer," in *Data Systems In Aerospace Conference (DASIA)*. Eurospace, June 2014, pp. 242–247.

[24] OHB, "Heinrich Hertz Satellite," January 2015, www.ohb-system.de/heinrich-hertz-english.html.

[25] A. Hofmann, R. Glein, B. Kollmannthaler, and R. Wansch, *An On-Board Processor for in Orbit Verification Based on a Multi-FPGA Platform*. Springer Berlin Heidelberg, 2011, pp. 147–157.

[26] R. Glein, B. Schmidt, F. Rittner, J. Teich, and D. Ziener, "A Self-Adaptive SEU Mitigation System for FPGAs with an Internal Block RAM Radiation Particle Sensor," in *22nd Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*. IEEE, May 2014, pp. 251–258.

[27] Fraunhofer Institute for Integrated Circuits, "Fraunhofer On-Board Processor," January 2015, www.iis.fraunhofer.de/en/ff/kom/proj/FOBP.html.

[28] F. Rittner, R. Glein, T. Kolb, and B. Bernard, "Broadband FPGA payload processing in a harsh radiation environment," in *NASA/ESA Conference on Adaptive Hardware and Systems (AHS)*. IEEE, July 2014, pp. 151–158.

[29] European Space Agency, "Solar Orbiter," January 2015, sci.esa.int/solar-orbiter.

[30] B. Fiethe, F. Bubenhagen, T. Lange, H. Michalik, H. Michel, J. Woch, and J. Hirzberger, "Adaptive Hardware by Dynamic Reconfiguration for the Solar Orbiter PHI Instrument," in *NASA/ESA Conference on Adaptive Hardware and Systems (AHS)*. IEEE, June 2012, pp. 31–37.