# ANALYSIS OF DESIGN ALTERNATIVES ON USING DYNAMIC AND PARTIAL RECONFIGURATION IN A SPACE APPLICATION

**Filip Veljković (1), Teresa Riesgo (1), Luis Berrojo (2), Raúl Regada (2), Ángel Álvaro (2), Eduardo de la Torre (1)**

(1) *Centro de Electrónica Industrial, Universidad Politécnica de Madrid, José Gutierrez Abascal, 2. 28006 Madrid, Spain, Email:{filip.veljkovic, teresa.riesgo, eduardo.delatorre}@upm.es*

(2) *Thales Alenia Space España, C/Einstein 7, 28760, Tres Cantos, Spain, Email: {luis-rafael.berrojovalero, raul.regadaalvarez, angel.alvaro}@thalesaleniaspace.com*

***Abstract*— Some of the biggest concerns in space systems are power consumption and reliability due to the limited power generated by the system's energy harvesters and the fact that once deployed, it is almost impossible to perform maintenance or repairs. Another consideration is that during deployment, the high exposure to electromagnetic radiation can cause single event damage effects including SEUs, SEFIs, SETs and others. In order to mitigate these problems inherent to the space environment, a system with dynamic and partial reconfiguration capabilities is proposed. This approach provides the flexibility to reconfigure parts of the FPGA while still in operation, thus making the system more flexible, fault tolerant and less power-consuming. In this paper, several partial reconfiguration approaches are proposed and compared in terms of device occupation, power consumption, reconfiguration speed and size of memory footprints.**

***Keywords—SRAM-based FPGAs; scalable dynamic and partial reconfiguration; scalability; on-board processor; space application;***

## 1. INTRODUCTION

In the past decade, FPGAs (Field Programmable Gate Arrays) have been progressively integrated into space system applications. The availability of low-cost, highly reconfigurable SRAM-based FPGA devices have made them a legitimate option for implementing Systems on Programmable Chips (SoPCs). Moreover, these FPGAs have been finding their way into space-based SoPCs due to their high performance and reconfigurability. The dynamic and partial reconfiguration (DPR) technique [1], available in this type of devices, represents one of the main reasons space engineering starts turning to the platform despite the fact that the vast majority of space digital systems is still based on Application Specific Integrated Circuits (ASICs). Although highly flexible, SRAM-based FPGAs are very susceptible to radiation effects due to their volatile nature. Once in orbit, maintenance and possible need for repairs become a difficult task. Therefore, taking the benefit of their volatility, SRAM-based FPGAs can be reconfigured in order to improve the performance or repair the existent fault. Furthermore, as the power consumption is always an important issue in space applications, self-reconfiguration provides the opportunity to save important amounts of energy and area having present in the device only necessary portions of logic at a given point of the operation. Bearing in mind all these advantages, along with the fact that they are generally less expensive than predominantly used ASICs, SRAM-based FPGAs are gradually taking the precedence in highly sensitive applications such as space and avionics.

Several published papers address the implementation of FPGA-based systems and DPR in space applications [2]-[4]. Research on this topic is mainly focused on high performance on-board processing which is achieved by updating processing modules real-time during the flight through space. However, in order to retain high reliability, self-reconfiguration should be performed safely and various factors should be taken into careful consideration [5]. Moreover, possible single event effects (SEEs) caused by radiation [6] have to be treated properly so that the system is able to detect the fault as soon as possible, characterize it, and eliminate it from the system such re-enabling regular operation.

Fault tolerant space-based SoPCs use DPR in order to heal the errors provoked by radiation, ageing or latent effects [7]. These systems are mostly based on redundancy where the entire design, or only a part of it, is duplicated or triplicated in hardware in order to propagate the correct output. Depending on the number of replicas, in literature we distinguish between double modular redundancy (DMR) and triple modular redundancy structures (TMR) [8]-[9]. In TMR, if one replica is affected by an SEE, two others can still propagate the correct result. However, if another domain gets affected, the structure starts producing erroneous outputs. In order to heal the system and thus extend the lifetime of the redundant architecture, only the faulty domain is reconfigured writing the original configuration to the determined part of the configuration memory [11]. Therefore, the smaller the domain, the faster and less power consuming is the healing process. Kretzschmar et al. in [12] introduce TMR architectures of different granularities in order to evaluate the robustness of the design. It is shown that the structure is able to propagate the correct result to the output even when two different domains are affected by an error at the same time, but in different domain stages. Accordingly, outputs are voted after each stage in order to mask a potential fault in one of them.

In this paper we present three different dynamically and partially reconfigurable architectures to be used in an

on-board processor which was previously designed as an ASIC and used for a digital video communication with a satellite. One of the main goals was to implement compact partial configurations of different granularities and determine the most convenient ones to be used under different operational conditions. The paper gives a comparative analysis in terms of device utilization, reconfiguration speed and size of memory footprints. In addition, fault tolerance capabilities which could be exploited in future upgrades of the design are studied for each of the architectures.

This paper is organized as follows. In section 2, the entire architecture of the digital video broadcast on-board processor is presented and the partition of the design into the static and reconfigurable parts has been made. Section 3 contains the main contribution of the paper. Interface and content of the partial configurations for each of the design alternatives are shown and the reconfiguration mode is presented and analyzed. Obtained results and comparisons between alternatives are discussed and summarized in section 4. Finally, section 5 gives perspectives and conclusion of the paper.

## 2. Reconfigurable Digital Video Broadcast On-Board Processor (OBP)

Currently, OBP technology is still prohibitively costly to be integrated into a satellite system. In addition, the research into OBP systems has not matched the pace that was originally expected for various reasons. One of the main factors responsible for this is that engineers are hesitant to use ASIC technology in space due to the lack of flexibility once commissioned. Most satellite missions are designed to last more than 15 years, which means that all the components will be obsolete towards the end of the satellite's life and will not be compatible with newer communication schemes. A viable solution to this problem is reconfigurable OBPs based on reprogrammable hardware, which would extend the life of the satellite due to its capability of transforming itself into a newer architecture that would be compatible with the changing technological landscape.

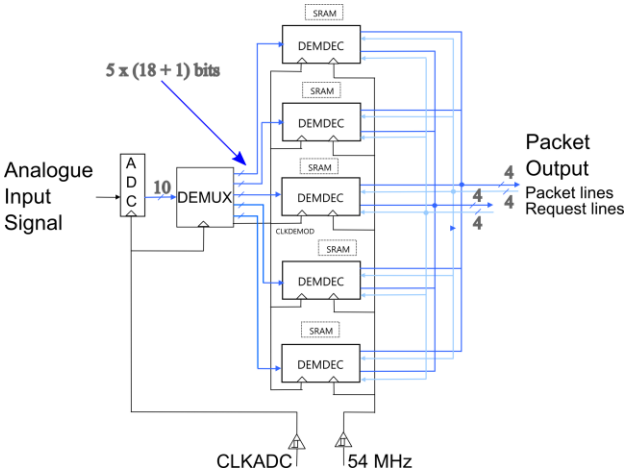


*Figure 1. Multi-Carrier Demultiplex Demodulate Decode unit - MC3D*

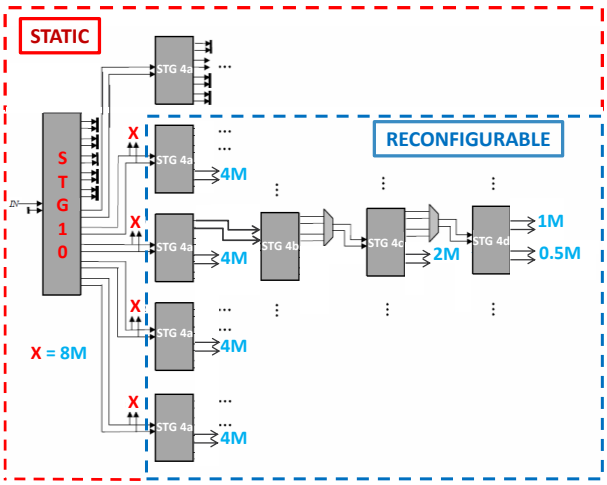


*Figure 2. Partitioning of the DEMUX architecture*

The reconfiguration methodologies that will be presented in the following sections are applied to an existing demultiplexer design (DEMUX), developed by Thales Alenia Space España and previously designed as an ASIC [13]. It forms a part of a base band processor which is designed for digital video communication with a satellite. The processor is composed of a series of MC3D units (Figure 1) which are intended to demultiplex, demodulate and decode carriers located within a transponder in order to generate a single multiplex of MPEG-2 packets following the digital video broadcasting standard. Hence, as presented, it is composed of an analog-to-digital converter (ADC), demultiplexer (DEMUX) and five demodulator-decoders (DEMDECs). The DEMUX architecture is selected and made reconfigurable as a consequence of its modular and scalable properties. The input is a 10-bit two's complement integer coming from the ADC, at a typical working frequency of 84.24 MHz, i.e., 240Rs, where Rs is the elementary symbol rate, typically 0.3510 MHz. This is also the frequency of the DEMUX input clock.

As presented in Figure 2, DEMUX is composed of the stage 10 (STG10) and 5 sub-bands (SB) corresponding to 5 different DEMDECs, within which the carriers are created. Four of them are with a bandwidth of 24Rs and the other one with a half that bandwidth. STG10 represents a series of components including 10-bin polyphase structure, configuration and output interfaces and at its outputs, carriers of 8 MHz can be extracted from the first 4 sub-bands. Furthermore, within each SB, after stages 4a, 4c, and 4d, 4, 2, 1 and 0.5 MHz carriers can be extracted. Depending on the configuration signal bits, which are received remotely, carriers of different frequency values are required from the system. Consequently, for the ASIC approach, some resources are present in the design although they are not necessary at some points of the operation. In order to cope with this issue, the architecture is divided into static and reconfigurable part. STG10, as it has to be present at all points of the operation, is left in the static part along with the 5$^{th}$ SB whereas the other 4 SBs are cut from the

architecture and placed in the reconfigurable part of the FPGA. Depending on the granularity level three different reconfiguration approaches are proposed and presented in the following section.

## 3. Reconfigurable Design Alternatives

As presented in Figure 2, the four reconfigurable SBs consist of several stages intended to generate carriers of different frequencies. Each of the SBs has the same architecture. STG4b outputs the material for the creation of lower frequency carriers (2, 1 and 0.5MHz) and, unlike the rest of the stages, is the only stage of a SB whose outputs are not actual carriers used later by the DEMDECs. In order to determine the best possible reconfiguration approach for the operation under different circumstances and application demands, our reconfigurable SBs are reconfigured in three different manners: using conventional, partially scalable and fully scalable partial configurations.

### 1) Conventional Dynamic and Partial Reconfiguration

The DPR technique today is mostly based on using conventional partial configurations where one part of the logic is substituted by another of different functionality such having present in the chip only the necessary hardware for that point of the operation. Hence, partial configurations are implemented such that include only those modules that are needed for a particular purpose. Accordingly, in order to make the DEMUX architecture reconfigurable and therefore save unnecessary portions of the on-chip area, we have created partial configurations consisting of only those stages that are needed for the creation of particular frequency carriers. The reconfigurable DEMUX architecture, for an example configuration where 0.5, 8, 4 and 2MHz carriers are demanded form the SBs, is presented in Figure 3. As can be seen, different partial configurations are configured depending on the remotely received configuration. Consequently, when 0.5 or 1 MHz carriers are requested from a SB, the on-chip area reserved for that SB is reconfigured using partial bitstream (PBS) 4 or PBS3, respectively. Both of these partial configurations consist of all four stages although PBS3 is significantly optimized in the last, stage 4, as no logic for the creation of 0.5 MHz carriers is necessary. In addition, all partial configurations are additionally optimized during the synthesis process as the complete hardware used for the creation of other frequency carriers is removed from the configuration.

PBS2 consists of only three stages. Hence, when 2 MHz carriers are requested from a SB, considerable amount of FPGA area is saved comparing to the non-reconfigurable design. In an ideal case, 2 MHz carriers would be taken after stage 2. However, as that is not the case, PBS2 also includes the stage 3 logic. Furthermore, PBS1 includes only one stage, stage 1, and is configured

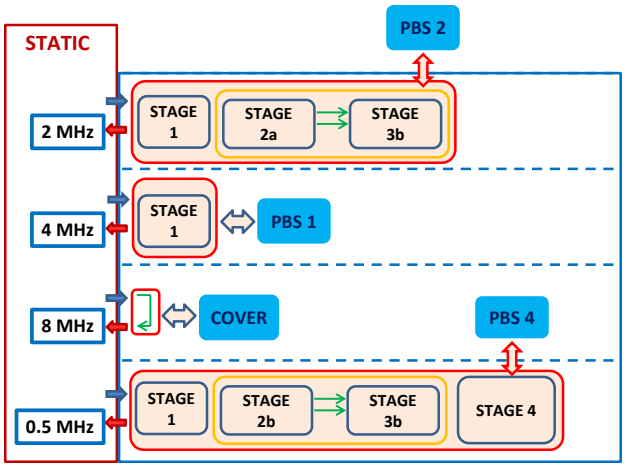


*Figure 3. Conventional Partial Configurations – Reconfigurable DEMUX architecture*

whenever 4 MHz carriers are demanded from a SB. As this is the least area consuming PBS significant amount of the on-chip area is saved comparing to the non-reconfigurable design. Finally, in order to make the design completely reconfigurable from the point of carrier's extraction, when 8 MHz carriers are demanded, a simple COVER is configured which takes the carriers created in the static part and returns them back to the output interface. Consequently, the entire area reserved for a SB is left without any logic.

When reconfiguring the device using conventional partial configurations no redundant logic is ever present in the reconfigurable part of the design. Therefore, this is the best possible DPR solution for area and power consumption savings comparing to the following ones.

### 2) Partially Scalable Dynamic and Partial Reconfiguration

The second proposed DPR approach takes the benefit of both modular and scalable nature of the design, making possible the reconfiguration of a smaller part of the hardware thus not only lowering the global design power consumption, like in the previous case, but also the time and power consumption during the partial reconfiguration process. In the partially scalable DEMUX, shown in Figure 5, partial configurations are implemented such that are able to connect one to another in order to reconfigure less area at each configuration change. Hence, stage 1 is a partition by itself (PBS_S1) and it is not included in other partial configurations used for the lower frequency carrier creation. It is the least area consuming partial configuration as it contains only one stage like in the conventional approach but also provides a possibility to be attached to the following one such supporting the scalability concept. Unlike conventional PBS2 which is consisted of three stages, scalable PBS_S2 contains 2 stages and is able to connect to PBS_S1 in order to create and extract 2 MHz carriers from a SB.

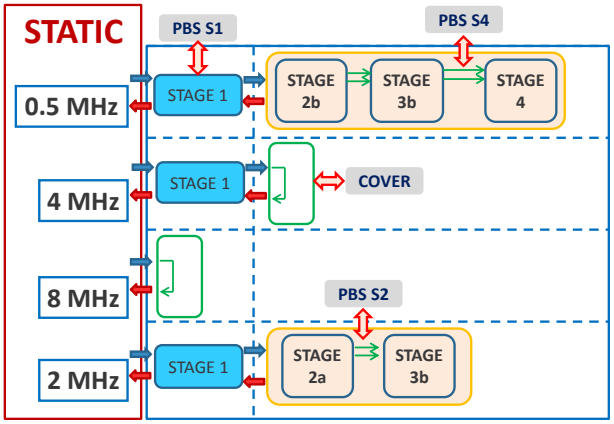


*Figure 5. Partially Scalable Partial Configurations – Reconfigurable DEMUX architecture*

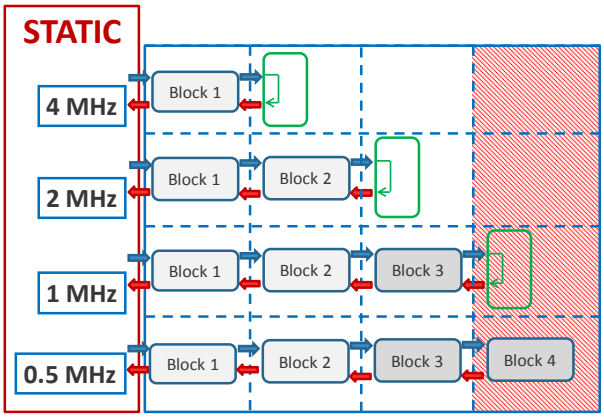


*Figure 4. Scalable Partial Configurations – Reconfigurable DEMUX architecture*

In an ideal case, PBS_S3 and PBS_S4 would consist of only stage 4 and would be able to attach to the PBS_S2 in order to create and extract 1 or 0.5 MHz carriers. However, due to the unavailability of the on-chip area for a single reconfigurable SB, both of them include stages 2, 3 and 4 and like PBS_S2 connect to the PBS_S1 when 1 or 0.5 MHz carriers are requested.

The last two are the most area consuming partially scalable configurations as they consist of three stages instead of only one, stage 4, and their partial scalability is reflected in the fact that they can only attach to the PBS_S1 and cannot provide connectivity to other scalable configurations. This is directly related to the unavailability of the on-chip area as these partially scalable configurations cannot be optimized during the synthesis process and no logic can be removed due to the fact that it might be used by the following reconfigurable modules. Therefore, further granulation of the reconfigurable sub-band was impossible without modifying the hardware of the rest of the stages.

The achieved scalability is partial as the lower frequency carriers (2, 1 and 0.5 MHz) are obtained by reconfiguring the sub-band reconfiguration zone placed next to the stage 1. However, comparing to the conventional DPR, obtained partial configuration enable faster and therefore less power consuming reconfiguration process and also occupy smaller amount of memory storage.

### 3) Fully Scalable Dynamic and Partial Reconfiguration

The third approach shown in Figure 4 is a fully scalable partial reconfiguration where the configurations are based on fine-grained implementations of the DEMUX SB. Each block represents a partition by itself and is able to attach to the following block in order to create carriers of lower frequencies. Therefore, stages are "erased" or configured one next to another at each configuration change.

In the fully scalable solution one stage is never repeated in other partial configuration. Nevertheless, as the original DEMUX stages were not suitable for this kind of implementation, certain modifications of the second and third stage had to be made. A modified stage 2 which represents the block 2 fully scalable partial configuration is presented in Figure 6. The original stage was unable to create 2 MHz carriers, which was the biggest limitation for the fully scalable design. Therefore, the FILT3 modules, which were previously part of the stage 3, are moved to the previous stage thus creating a fully scalable partial configuration able to generate 2 MHz carriers by itself. These modules could not be completely removed from the stage 3 as some parts are used for the creation of 1 MHz carriers. Consequently, block 3 configuration has similar architecture consisting of its own, but modified FILT3 modules and FILT4 modules which are similarly moved from the stage for and used to generate 1 MHz carriers.

Unlike in the previous two alternatives where the SB reconfiguration zone consists of one or two zones, each SB of the fully scalable DEMUX is divided in 3 reconfiguration areas, one for each partial configuration. As presented in Figure 4, the fourth reconfiguration zone, which should be used for the configuration of the block 4, is left out due to the lack of area in the FPGA. That is because the fully scalable partial configurations cannot be optimized during the synthesis process as the logic in each anterior block is used for the creation of lower frequency carriers. Like in the partially scalable approach, after each scalable block a simple cover is configured to return the demanded carriers to the static part of the design.

Fully scalable partial reconfiguration provides the best results in terms of the reconfiguration speed, flexibility and power consumption during the reconfiguration process. On the other hand, as there are a significant number of additional reconfiguration zones a slightly larger portion of the FPGA area needs to be reserved for the reconfigurable part of the design. Moreover, comparing to the conventional DEMUX, there is always redundant logic present on the chip for each of the configurations as nothing can be removed from the block considering that it might be used later by the

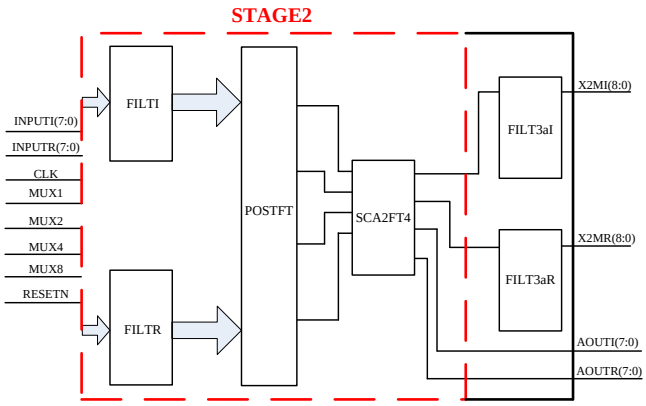


*Figure 6. Block 2 - Fully scalable partial configuration*

following blocks. Nevertheless, the implementation of this type of partial configurations introduces important reductions in the size of memory footprints required for the bitstream storage.

## 4. Results and Discussion

The proposed reconfigurable architecture is implemented on a Xilinx Virtex-5 XC5VFX130T FPGA. The selection of the device is done considering its compatibility with the Space-grade Virtex-5QV FPGA. The implementation of the static and reconfigurable portions of the design is performed in ISE Design Suite 14.2. Implementations are carefully constrained using PlanAhead and FPGA Editor such that there is no possibility for any kind of logic overlapping between the static and reconfigurable zones. In order to ensure permanent and proper connection between these parts of the design, hard bus macros are created in FPGA Editor and placed at the predetermined positions on the configuration border in both static and reconfigurable configurations. They consist of only two slices which can fix up to 8 bits of data and are split and recomposed at each partial reconfiguration. The static part includes 32 bus macros, 8 for each reconfigurable SB. Hence, each conventional partial configuration have 8 bus macros on the border with the static part of the design. On the other hand, fully scalable partial configurations have 8 bus macros on both sides of the configuration in order to ensure proper communication with the neighbor reconfiguration zones.

As previously explained, the static part should consist of the logic that has to be present during the entire operation. Apart from the fixed part of the DEMUX architecture, the implemented static part of the entire system includes several other peripherals implemented to support the DPR and control the communication on-chip. Therefore, two memory controllers have been implemented in order to enable the communication with the external RAM and flash memories. A MicroBlaze processor is implemented in order to control the reconfiguration and evaluation processes. However, in the final solution, it should be substituted by the LEON2-FT2 in order to improve the fault tolerance of the design. The

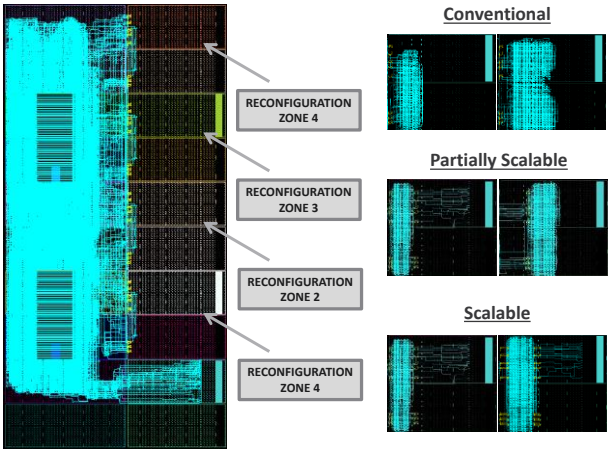


*Figure 7. Implemented static part of the design and partial configurations of three different alternatives*

TABLE 1
THE CHIP UTILIZATION, THE MAXIMUM RECONFIGURATION TIME AND THE SIZE IN MEMORY OF THE CONVENTIONAL PARTIAL CONFIGURATIONS

| * | PBS1 | PBS2 | PBS3 | PBS4 | Cover | |
|---|---|---|---|---|---|---|
| Occupied Slices | 280<br>1 % | 661<br>3 % | 1125<br>5 % | 1305<br>6 % | 16<br>1% | |
| Reconf. Time [us] | 103.3 | 191.9 | 280.44 | 280.44 | 29.52 | Total |
| Size [KB] | 132 | 211 | 295 | 295 | 24 | 957 |

main part of the system responsible for making the DPR possible is the enhanced HWICAP [14]. It is based on the Xilinx HWICAP but it also includes two important features in the reconfiguration process. The first one is the speed, i.e. faster reconfiguration is achieved as the enhanced HWICAP can rewrite the desired portion of the configuration memory using the partial configuration read directly from the RAM. The second implies a possibility to have relocatable partial configurations. In other words, partial configurations implemented in one part of the FPGA can later be configured in any other part of the chip which has the same architecture. Hence, as the entire right part of the device used for the implementation of the reconfigurable DEMUX has the same architecture, all partial configurations are implemented in the lower right corner of the chip, stored in the external memory and configured on demand in four reconfiguration zones corresponding to the number of reconfigurable SBs. The implemented static part of the system, along with 2 partial configurations for each of the design alternatives, is presented in Figure 7.

All partial configurations are compatible with the presented static part of the system and the choice of the alternative that is going to be used is completely dependent on the application preference. Reconfigurable SBs are implemented in the fixed zones and each one occupies 2 clock regions in the right part of the chip. Table 1, Table 2 and Table 3 present the design

TABLE 2.
THE CHIP UTILIZATION, THE MAXIMUM RECONFIGURATION TIME AND THE SIZE IN MEMORY OF THE PARTIALLY SCALABLE PARTIAL CONFIGURATIONS

| * | S1 | S2 | S3 | S4 | Cover | |
|---|---|---|---|---|---|---|
| Occupied Slices | 340 1 % | 484 2% | 944 4% | 1104 5% | 16 1% | |
| Reconf. Time [us] | 88.2 | 103.3 | 191.9 | 236.16 | 29.52 | Total |
| Size [KB] | 88 | 132 | 207 | 242 | 24 | 693 |

TABLE 3.
THE CHIP UTILIZATION, THE MAXIMUM RECONFIGURATION TIME AND THE SIZE IN MEMORY OF THE SCALABLE PARTIAL CONFIGURATIONS

| * | Block 1 | Block 2 | Block 3 | Cover | |
|---|---|---|---|---|---|
| Occupied Slices | 340 (1 %) | 488 (<2%) | 674 (<3%) | 16 (1%) | |
| Reconf. Time [us] | 88.2 | 103.3 | 163.7 | 29.52 | Total |
| Size [KB] | 88 | 100 | 147 | 24 | 359 |

occupation statistics, the maximal reconfiguration time and the size in the memory for each partial configuration used in three different design alternatives. Although provide such flexibility that no redundant logic is present in the chip, conventional partial configurations occupy the largest amount of memory which is a significant disadvantage comparing to the scalable solution and considering the price of the rad-hard memory used in space applications. Moreover, changes from one configuration to another are more time consuming due to the fact that the stages are repeated in different partial configurations such implying larger area for reconfiguration and therefore more time for the reconfiguration process. Although the total size of the fully scalable partial configurations cannot provide a fair comparison with the previous solutions, it shows a significant improvement in terms of the amount of saved rad-hard memory used for bitstream storage. Therefore, the decision on whether to use conventional or one of the scalable solutions should be a consequence of the trade-off between the power consumption, reconfiguration speed and the size of memory footprints that are stored in an external, radiation hardened, memory. In addition, by making the design reconfigurable the total amount of the memory used for both full and partial configuration is reduced by 0.73 MB in the case of the reconfigurable DEMUX with conventional partial configurations and by 1.34 MB when fully scalable partial configurations are used.

## 5. CONCLUSIONS

In this paper we have presented three different alternatives on using DPR in a digital video broadcast on-board processor. The proposed reconfigurable designs are implemented in the ML510 development platform based on the Xilinx Virtex-5 XC5VFX130T FPGA. The choice of the chip is made based on the compatibility with an equivalent space qualified FPGA. Proposed techniques can be applied to any design with modular and scalable properties. The paper analyses the impact of different granularities on reconfigurable sub-bands based on coarse, medium and fine-grained partial configuration implementations. Embedded self-reconfiguration techniques provide savings in terms of resource utilization, size of the memory footprints and power consumption. Partial configurations for all the alternatives are carefully constrained, compact and reliable. Moreover, scalable solutions offer faster configuration changes and require smaller amount of the external rad-hard memory comparing to the conventional solutions. Taking into account that all the configurations can be relocated in the entire right part of the chip and that the same are used for each of the SBs, future work will address the improvement of fault tolerance capabilities by configuring particular SBs several times and such adapting on-line to harsh environmental conditions using redundancy. The scalable partial configurations can be used in such adaptive fault tolerant structures in order to cure the system by reconfiguring as less area as possible. Consequently, the redundant system would be able to operate properly even if several domains are affected by an error in different stages by introducing majority voters after each stage. In addition, the time available for repairing the redundant structure by reconfiguration would increase as there would be less area to be reconfigured and the probability that the particle affects the same stage in several domains is lower comparing to the case when conventional modules are used.

## REFERENCES


[1] http://www.xilinx.com/support/documentation/sw_manuals/xilinx14_5/ug702.pdf

[2] Osterloh, B.; Michalik, H.; Fiethe, B.; Bubenhagen, F., "Architecture verification of the SoCWire NoC approach for safe dynamic partial reconfiguration in space applications," Adaptive Hardware and Systems (AHS), 2010 NASA/ESA Conference on , vol., no., pp.1,8, 15-18 June 2010

[3] Yimao Cai; Yuanfu Zhao; Lidong Lan, "Implementation of a reconfigurable computing system for space applications," System Science, Engineering Design and Manufacturing Informatization (ICSEM), 2011 International Conference on , vol.2, no., pp.360,363, 22-23 Oct. 2011

[4] Quinn, H.; Graham, P.; Morgan, K.; Caffrey, M.; Krone, J., "A test methodology for determining space-readiness of Xilinx SRAM-based FPGA designs," AUTOTESTCON, 2008 IEEE , vol., no., pp.252,258, 8-11 Sept. 2008 doi: 10.1109/AUTEST.2008.4662621

[5] Osterloh, B.; Michalik, H.; Fiethe, B.; Bubenhagen, F., "Architecture verification of the SoCWire NoC approach for safe dynamic partial reconfiguration in space applications," Adaptive Hardware and Systems (AHS), 2010 NASA/ESA Conference on , vol., no., pp.1,8, 15-18 June 2010

[6] E. Petersen, "Single Event Effects in Aerospace", Wiley-IEEE Press, 2011.

[7] Bolchini, C.; Miele, A.; Sandionigi, C., "A Novel Design Methodology for Implementing Reliability-Aware Systems on SRAM-Based FPGAs," Computers, IEEE Transactions on , vol.60, no.12, pp.1744,1758, Dec.2011 doi: 10.1109/TC.2010.281

[8] C. Carmichael. Triple Module Redundancy Design Techniques for Virtex® Series FPGA: Application Notes 197. San Jose, USA: Xilinx, 2000.

[9] de Lima Kastensmidt, F.G.; Neuberger, G.; Hentschke, R.F.; Carro, L.; Reis, R., "Designing fault-tolerant techniques for SRAM-based FPGAs," Design & Test of Computers, IEEE , vol.21, no.6, pp.552,562, Nov.-Dec.2004 doi: 10.1109/MDT.2004.85

[10] Kastensmidt, F.L.; Sterpone, L.; Carro, L.; Reorda, M.S., "On the optimal design of triple modular redundancy logic for SRAM-based FPGAs," Design, Automation and Test in Europe, 2005. Proceedings , vol., no., pp.1290,1295 Vol. 2, 7-11 March 2005

[11] Azambuja, J.R.; Sousa, F.; Rosa, L.; Kastensmidt, F.L., "Evaluating large grain TMR and selective partial reconfiguration for soft error mitigation in SRAM-based FPGAs," On-Line Testing Symposium, 2009. IOLTS 2009. 15th IEEE International , vol., no., pp.101,106, 24-26 June 2009

[12] Kretzschmar, U.; Astarloa, A.; Lazaro, J.; Garay, M.; Del Ser, J., "Robustness of different TMR granularities in shared wishbone architectures on SRAM FPGA," Reconfigurable Computing and FPGAs (ReConFig), 2012 International Conference on , vol., no., pp.1,6, 5-7 Dec. 2012

[13] Veljković, F.; Mora, J.; Riesgo, T.; Berrojo, L.; Regada, R.; Álvaro, A.; de la Torre, E.,"Prospections of Reconfiguration Capabilities using Space Qualified SRAM-based FPGAs for a Satellite Communications Application," Communications Satellite Systems Conference, 2013. ICSSC 2013. 31st AIAA International, 14-17 Oct. 2013

[14] Otero, A.; Morales-Cas, A.; Portilla, J.; de la Torre, E.; Riesgo, T., "A Modular Peripheral to Support Self-Reconfiguration in SoCs,"Digital System Design: Architectures, Methods and Tools (DSD), 2010 13th Euromicro Conference on, vol., no., pp.88,95, 1-3 Sept. 2010