GRSCRUB: FPGA Configuration Supervisor White Paper

Ádria Barros de Oliveira, Lucas Antunes Tambara, Fabio Malatesta, Fredrik Sturesson, Jan Andersson, Martin Ronnback, Fredrik Johansson, and Sandi Habinc – Cobham Gaisler AB
www.gaisler.com

***Abstract*—The GRSCRUB is an external Field Programmable Gate Array (FPGA) configuration supervisor developed by Cobham Gaisler as an Intellectual Property (IP) core. The GRSCRUB IP features different capabilities, such as programming and scrubbing, which prevents the accumulation of errors in the configuration memory of SRAM-based FPGAs. The GRSCRUB IP is currently compatible with the Xilinx Kintex UltraScale and Virtex-5 FPGA families. This white paper describes the GRSCRUB IP functionalities and evaluates the system by emulating faults in the target FPGA. Two evaluation designs are used: a static design and a design based on a LEON3FT processor core. The results demonstrate the GRSCRUB IP capability in correcting all faults injected in the FPGA configuration memory. In addition, in the case of the LEON3FT-based design, the GRSCRUB IP scrubbing operation allows uninterrupted software execution in the presence of correctable errors in the FPGA configuration memory by preventing the error build-up.**

***Index Terms*—Scrubbing, GRSCRUB, GR716B, FPGA, UltraScale, Soft-error, SEE, Fault injection.**

I. Introduction

DEVICES operating in the space environment are vulnerable to ionizing particles that may affect the system and provoke errors. Even the space-grade Field Programmable Gate Arrays (FPGA) are susceptible to Single Event Effects (SEE) that may affect not only the user data but also the configuration memory of the device. Single Event Upsets (SEU) in the FPGA configuration memory may lead to persistent errors in the system, changing the architectural implementation of the design. Scrubbing is a well-known technique responsible for coping with errors in the configuration memory and avoiding their build-up.

Scrubbing can be defined as internal when the scrubber engine is embedded in the target FPGA being monitored, and external when the scrubber controller is located externally to the target FPGA in a different component. The literature presents several scrubbing implementations that mainly differ in the error detection, power consumption, resource usage, and correction speed [1-3]. The Xilinx Soft Error Mitigation Intellectual Property (SEM-IP) [4] is an internal scrubbing core compatible with most of Xilinx FPGAs. The SEM-IP main advantage is the high-speed for single error detection and correction. As demonstrated in [5], internal scrubbers are susceptible to get locked and have the correction capability compromised due to faults in the scrubber interface or multiple errors in the configuration memory. In this context, external scrubbers may provide higher robustness and the ability to deal with multiple errors.

The Cobham Gaisler's GRSCRUB IP core is an external FPGA configuration supervisor that features programming and scrubbing capabilities, which prevents the accumulation of errors in the configuration memory of SRAM-based FPGAs. The GRSCRUB IP targets soft errors affecting the FPGA configuration memory, and it is able to detect and correct single and multiple errors. However, one must notice that the GRSCRUB IP does not avoid bit-flips from happening or its effects on the design, as well as other scrubbers. Therefore, additional mitigation techniques at design level are recommended to decrease the number of single points of failure in the system and increase the fault masking. Table I presents a comparison between the GRSCRUB IP and the SEM-IP.

The GRSCRUB IP is currently compatible with the Kintex UltraScale and Virtex-5 Xilinx FPGA families. It accesses the FPGA configuration memory externally through the SelectMap (SMAP) interface, which provides better performance in comparison with JTAG, due to the parallel data access. The GRSCRUB is part of the Cobham Gaisler's IP library (GRLIB) [6]. Moreover, the GRSCRUB IP will be integrated into the next version of the GR716 Microcontroller

TABLE I
GRSCRUB AND SEM-IP COMPARISON

	GRSCRUB	SEM-IP [4]
Type of scrubbing	External	Internal
FPGA programming	Yes	No
Single error detection	Yes	Yes
Single error correction	Yes	Yes
Multiple errors detection	Yes	Yes
Multiple errors correction	Yes	No
Advantages	Robustness; MBU correction; multifunction	High correction speed

Á. B. de Oliveira, L. A. Tambara, F. Malatesta, F. Sturesson, J. Andersson, M. Ronnback, F. Johansson, and S. Habinc are with Cobham Gaisler AB, Gothenburg, Sweden. E-mail: {adria, lucas.a.tambara, fabio.malatesta, fredrik, jan, martin.ronnback., fredrik.johansson, sandi}@gaisler.com.

(GR716B), which is a mixed-signal fault-tolerant microcontroller Application-Specific Integrated Circuit (ASIC) based on the LEON3FT SPARC V8 processor.

This white paper presents the GRSCRUB IP features and its distinct scrubbing execution modes. The scrubbing capability is evaluated under fault injection by emulation, targeting a Xilinx Kintex UltraScale FPGA. The results show that the GRSCRUB IP is able to correct all faults injected in the FPGA configuration memory. In addition, the tests in a LEON3FT-based design demonstrate that the GRSCRUB IP scrubbing operation allows uninterrupted software execution in the presence of correctable errors in the FPGA configuration memory by preventing the error build-up.

II. Soft error mitigation in SRAM-based FPGAs

Radiation-induced soft errors are errors provoked by ionized particles that affect the system without damaging the device permanently. SRAM-based FPGAs are particularly susceptible to soft errors due to the memory elements used to configure the design logic and architecture. SEUs affecting such elements may lead to persistent errors in the system, changing the architectural implementation of the design. Single Events Transients (SET) are transient pulses that propagate through the combinational logic and may be captured by a memory cell, changing the storage data. Soft errors can also directly affect the memory data, registers, and flip-flops, and cause Silent Data Corruptions (SDC), which are incorrect results outputs. The Single Event Functional Interrupt (SEFI) occurs when a soft error affects the control logic or a state register and leads to hangs or crashes in the design.

In this context, applications that demand a high level of reliability, such as space applications, require fault mitigation methods to cope with the radiation-induced effects. The GRSCRUB IP aims to maintain the FPGA configuration memory consistent by repairing the logic and correcting bit-flips, avoiding the accumulation of faults. The GRSCRUB IP targets soft errors affecting the only FPGA configuration memory. The memory elements that store dynamic data, such as Block RAMs (BRAM), distributed memory, and Flip-Flops (FF), are not protected by the IP. Moreover, the GRSCRUB IP does not avoid bit-flips from happening or its effects on the design, but it detects and corrects the configuration errors. One can combine the GRSCRUB IP with additional mitigation techniques at the design level to increase the overall system reliability.

Soft errors affecting the dynamic elements can be mitigated by applying fault tolerance techniques such as redundancy or Error Correction Code (ECC). Triplicating logic is an efficient method to cope with the effects of single faults in the design. Additional user level techniques can also be applied to deal with SDCs. Moreover, a periodic reset may be required to reestablish the system state and restore the initial state of flip-flops. Since SEFIs may also affect internal control elements of the FPGA or the configuration interface, a complete power cycle might be required to restore the system.

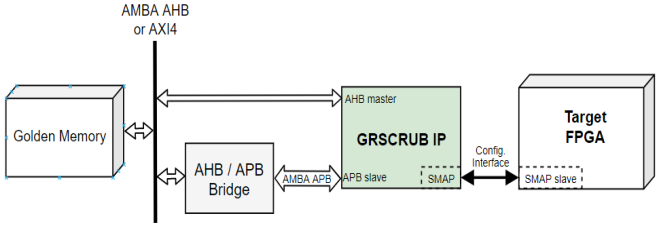

Fig. 1. GRSCRUB IP system block diagram.

III. GRSCRUB - FPGA configuration supervisor

A. *Xilinx FPGAs architecture*

The architecture of Xilinx FPGAs is divided into frames. Each frame contains data divided into 32-bit words. The GRSCRUB IP works with frames related to the FPGA configuration memory, which includes Configurable Logic Blocks (CLB), input and output (I/O) interconnections, and clock lines. As described in the previous section, GRSCRUB IP verifies the static data in the configuration memory only, and the dynamic memory elements are not protected.

B. *GRSCRUB IP system setup*

Fig. 1 shows the block diagram of a GRSCRUB-based system, which can be the GR716B Microcontroller or a design implemented in a flash-based FPGA, integrated with the target FPGA. The configuration memory of the target FPGA is accessed externally through the slave SelectMap configuration interface. The GRSCRUB IP can access SelectMap through all the supported bus widths (i.e., 8-, 16-, or 32-bit). The slave SelectMap clock is provided externally by the system in which the GRSCRUB IP is embedded. The GRSCRUB IP is a multiple clock domain design, which includes the internal system clock, and the SelectMap clock used for synchronization.

The GRSCRUB IP accesses through an AMBA AHB or AXI4 bus, a memory that stores the golden configuration bitstream and the mask data of the design implemented in the target FPGA (Golden memory). The golden bitstream is used both to configure the FPGA at start-up and to repair the configuration memory in the event of soft errors. The mask data information is provided by the synthesis tool and contains a description of all dynamic bits in the design. During data verification in the scrubbing operation, the GRSCRUB IP does not verify the dynamic bits in the frames, and the mask data is used to mask only these specific bits.

C. *GRSCRUB IP operation modes*

The GRSCRUB IP implements five operation modes:

1) *Idle mode*: the IP is in idle waiting for an operation command.
2) *Programming mode*: the IP programs the configuration bitstream into the target FPGA.
3) *Scrubbing mode*: the IP executes a scrubbing operation. As described further, two scrubbing methods are supported: blind and readback scrubbing. The IP can be configured to scrub the entire FPGA configuration memory or just selected frames.

4) *Mapping mode*: the IP identifies and maps the frame addressing of the target FPGA. The frame addressing defines the frames positioning in the target FPGA, required for any scrubbing operation. Only frames that refer to configuration blocks are mapped, i.e., the memory block frames are not considered. The frame addresses are saved in the Golden memory and are accessed by the IP in scrubbing mode during reading and writing operations.
5) *Golden Cyclic Redundancy Check (CRC) mode*: the IP computes the golden CRC codes for the current frame data of the target FPGA configuration memory. The CRC code can be selected as a data check in the readback scrubbing mode. A CRC code is computed to each frame of the configuration memory, and it is verified against the golden CRC copy.

The GRSCRUB IP scrubbing operation mode supports both blind and readback scrubbing methods. In the blind scrubbing mode, the GRSCRUB IP rewrites the configuration frames without any data verification. The blind scrubbing can be performed periodically, continually refreshing the configuration data. In the readback scrubbing mode, the GRSCRUB IP verifies the integrity of each frame of the FPGA configuration memory, and then, in the event of errors, rewrites the frame with the correct data read from the Golden memory. Differently from the blind scrubbing, the readback mode allows detecting errors and correcting the frame only if necessary. The readback can also be executed periodically.

The error detection can be performed through CRC verification or by comparing a frame bit-by-bit against its golden version stored in the Golden memory. The latter option is defined as Full Frame Check (FFC). The CRC is an error detection code that applies redundancy to check inconsistencies. A standard 32-bit CRC (CRC32C) algorithm is computed for each FPGA frame, and it is compared to the golden code saved in the Golden memory. The CRC and FFC data verifications do not check the masked bits. Each data verification method can be configured to be enabled or not.

D. GRSCRUB IP additional features

The configuration interface of the target FPGA can also be affected by soft errors, which may lead to catastrophic results during the scrubbing operation. For instance, in case the FPGA Frame Address Register (FAR) is affected by an SEU during a blind scrubbing execution and its value changes to another valid address, all the subsequent frames would be overwritten wrongly, compromising the entire design. In [7], the authors observed high-current events in Xilinx FPGAs due to SEEs affecting the configuration interface, which led the blind scrubbing to write multiples frames in incorrect addresses.

In order to avoid such events, the GRSCRUB IP verifies the integrity of the configuration interface of the target FPGA before each new scrubbing execution. The verification is performed by reading a specific frame and checking its address. If the returned address matches the expected one, the interface is considered stable and, therefore, the scrubbing cycle starts. Otherwise, an error is reported. In addition, setting up the configuration interface for each scrubbed frame could be a safer approach instead of configuring all frames at once. For instance, writing one frame at a time during blind scrubbing avoids overwritten the entire memory in case of errors in the FAR register. Both blind and readback scrubbing can be configured to enable or disable such features.

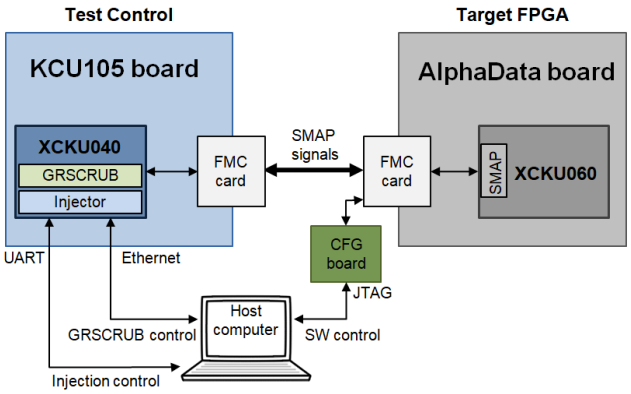

Fig. 2. GRSCRUB IP experimental setup.

TABLE II
RESOURCE USAGE OF GRSCRUB IP IMPLEMENTED IN THE XCKU040 FPGA

LUT	FF	Carry	DSP	BRAM
4,550	2,678	117	5	1

IV. EXPERIMENTAL EVALUATION SETUP

The GRSCRUB IP test setup consists of a host FPGA embedding the GRSCRUB IP in a system similar to the one presented in Fig. 1, and the Device Under Test (DUT), which is the target FPGA under evaluation, a Xilinx UltraScale FPGA. Fig. 2 presents the block diagram of the experimental setup.

A. Test controller

A Xilinx KCU105 evaluation board [8] is used as the test controller. The board features a Xilinx Kintex UltraScale XCKU040 FPGA, in which the GRSCRUB IP and a fault injection engine are implemented. Table II presents the resource usage of the GRSCRUB IP embedded in the XCKU040 FPGA.

The fault injection engine is controlled via UART and is responsible for emulating upsets in the configuration memory of the target FPGA. It also uses the SelectMap interface to access the configuration frames and flip bits, one at time. The target frame and target bit inside the frame are selected randomly. The injection engine reads the selected frame, flips the target bit, and then rewrites the frame to the FPGA. Since both GRSCRUB IP and injector uses the SelectMap interface to access the FPGA configuration memory, only one can be enabled at a time.

TABLE III
RESOURCE USAGE OF STATIC AND LEON3FT DESIGNS IMPLEMENTED IN THE XCKU060 FPGA

Design	LUT	FF	Carry	DSP	BRAM
Static	23	521	3	0	0
LEON3FT	8,852	6,016	43	4	53

Besides the GRSCRUB IP and the fault injection system, the test controller design also contains other IP cores from the GRLIB IP library [6], such as AHB bus, DDR3 memory controller, Debug Support Unit (DSU), Ethernet, and UART. In this setup, the GRSCRUB IP is controlled through Ethernet using the Cobham Gaisler's GRMON3 debug monitor [9] that configures the IP to execute the operation modes presented in Section III.

Two FPGA Mezzanine Card (FMC) breakout boards are used to allow the communication between the test controller and the DUT board. The SelectMap signals from the target FPGA are accessed and controlled via the FMC cards.

The test controller frequency is *100 MHz*, and the provided SelectMap clock is *10 MHz*. The maximum SelectMap clock frequency depends on the system setup. Due to the cabling to connect both boards and long signal paths, the SelectMap frequency is restricted in the experimental setup. Higher speeds can be achieved in a system integrating the target FPGA and GRSCRUB IP on the same board.

The fault injection campaigns aim first to evaluate the GRSCRUB IP and test the scrubbing functionality, and second to ensure that the IP operates transparently in dynamic designs. In all test campaigns, the GRSCRUB IP programs the target FPGA, and then the test controller starts the execution. For each injection run, one or more random faults are injected in the configuration memory of the target FPGA. In sequence, the GRSCRUB IP is released to scrub the faulty bits. At the end of the scrubbing execution, the configuration memory is verified to check if all bits were corrected. After that, a new injection run starts, and the loop is repeated.

B. Target FPGA – Xilinx Kintex UltraScale XCKU060

An AlphaData ADM-SDEV-BASE development kit [10] embedding a Xilinx Kintex UltraScale FPGA (XCKU060-1-FFVA1517I industrial part, equivalent to the XQRKU060-CNA1509 space-grade part) is the adopted target FPGA. An FMC card is also attached to the board, providing access to the JTAG interface. The JTAG connection is used to control the software execution when required.

Two test designs were implemented for the evaluation experiments, as described below:

- *Static design:* the design does not implement any dynamic function, and therefore most of the configuration bitstream is empty. The functionality of the design is not evaluated since the goal is only to validate the GRSCRUB IP features. The fault injection targets all FPGA configuration frames, and the IP also monitors the entire configuration memory.
- *LEON3FT-based design:* the design implements a LEON3FT processor core. In addition to the LEON3FT processor, the design also contains other IP cores from GRLIB [6], such as DSU, fault-tolerant SRAM module, AHB bus, JTAG, and UART. The 16 KB Instruction and Data L1 caches and the processor Register File (RF) are implemented in BRAMs and are protected by Error Detection And Correction (EDAC). The LEON3FT runs a test software that monitors and tests the Integer Unit (IU) of the processor. The software is controlled using the GRMON3 via JTAG. The floorplanning of the design is constrained to a specific area, and both fault injection and GRSCRUB IP only target this area.

The resource usage of Static and LEON3FT designs implemented in the target FPGA are presented in Table III.

TABLE IV
FAULT INJECTION RESULTS FOR STATIC DESIGN AND GRSCRUB IP IN DIFFERENT SCRUBBING MODES

GRSCRUB IP Scrubbing	# Inj. faults per run	# Total runs	# Total faults corrected
Blind	1	2,000	2,000
Blind	10	15,735	157,350
Readback FFC	10	12,086	120,860
Readback CRC	10	7,220	72,200

V. EVALUATION RESULTS

Tables IV present the fault injection results for the Static design implemented in the target FPGA. The blind, readback FFC, and readback CRC scrubbing modes of the GRSCRUB IP were evaluated. Single or multiple random faults were injected per run, and then the GRSCRUB IP scrubbing mode was enabled to correct the faults. In all tests, the GRSCRUB IP was able to detect and correct all injected faults.

The tests with the LEON3FT-based design implemented in the target FPGA demonstrated that 99.6% of the software runs were successful. The software executed continuously while single random faults were injected in the target FPGA. After each injection, the GRSCRUB IP readback FFC scrubbing was enabled to clear the bit-flip. A total of 11,399 faults were injected, and the GRSCRUB IP was able to correct all injected faults. The large amount of injected faults not leading to errors in the target design confirms that the GRSCRUB IP scrubbing operation allows uninterrupted software execution in the presence of correctable faults in the FPGA configuration memory by preventing the error build-up. The software errors presented refer to critical points of failure related to non-protected modules in the target FPGA design (the literature usually refers to such bits as "critical bits") that lead to errors before the GRSCRUB IP be able to correct the fault. One must notice that such software errors are application-dependent, i.e., different software benchmarks may lead to different results. In this context, the GRSCRUB IP minimizes the latency of single points of failure in the system, but it does not avoid errors happening and neither their effects on the design. Additional mitigation techniques at the design level are recommended to decrease the number of single points of failure and increase the fault masking.

VI. Conclusion

The Cobham Gaisler's GRSCRUB IP is an FPGA configuration supervisor that features programming and scrubbing capabilities. The GRSCRUB IP will be included in the new version of the Cobham Gaisler's GR716B Microcontroller, and it is also available as an IP core in the GRLIB. Fault injection tests targeting a Xilinx Kintex UltraScale FPGA demonstrated the GRSCRUB IP capability to correct all injected faults in the FPGA configuration memory. Tests in a LEON3FT design confirms that the GRSCRUB IP scrubbing operation allows uninterrupted software execution in the presence of correctable errors in the FPGA configuration memory by preventing the error build-up. The GRSCRUB IP reduces the persistent effects of errors in critical points of failure. However, the impact on the design is not mitigated. Therefore, additional mitigation techniques at the design level are recommended for that and to increase the fault masking.

References

[1] J. Heiner *et al.,* "Fault Tolerant ICAP Controller for High-Reliable Internal Scrubbing," *2008 IEEE Aerospace Conference*, Big Sky, MT, 2008, pp. 1-10.

[2] F. Brosser *et al.,* "Assessing scrubbing techniques for Xilinx SRAM-based FPGAs in space applications," 2014 International Conference on Field-Programmable Technology (FPT), Shanghai, 2014, pp. 296-299.

[3] A. Stoddard *et al.,* "A Hybrid Approach to FPGA Configuration Scrubbing," in IEEE TNS, vol 64, no 1, pp 497-503, Jan 2017.

[4] Xilinx, "Soft Error Mitigation Controller," v4.1 LogiCORE IP Product Guide, Vivado Design Suite, PG036, Apr. 2018.

[5] M. Berg *et al.,* "Effectiveness of Internal Versus External SEU Scrubbing Mitigation Strategies in a Xilinx FPGA: Design, Test, and Analysis," in IEEE TNS, vol. 55, no. 4, pp. 2259-2266, Aug. 2008.

[6] C. Gaisler, GRLIB IP Core User's Manual, Version 2020.1, Mar. 2020.

[7] D. S. Lee *et al.,* "An Analysis of High-Current Events Observed on Xilinx 7-Series and Ultrascale Field-Programmable Gate Arrays," IEEE Rad. Effects Data Workshop (REDW), Portland, OR, USA, 2016, pp. 1-5.

[8] Xilinx, "KCU105 Board User Guide," UG917 (v1.10), Feb 2019.

[9] C. Gaisler, "GRMON3 User's Manual," GRMON3-UM (Version 3.2.2), Mar. 2020.

[10] AlphaData, "ADM-SDEV-BASE/XCKU060 User Manual," V1.4, 2020.