Micro-RDC
Microelectronics Research Development Corporation

Programmable Scrubber for FPGAs

(Preliminary Feature Sheet)

Scrubbing is the process of removing errors from a memory's content by rewriting it periodically with correct values. Scrubbing is used in conjunction with Triple Mode Redundancy (TMR) to mitigate the effects of radiation in SRAM-based FPGAs. Micro-RDC's programmable scrubber for FPGAs is the first industry available scrubber flexible enough to support a wide range of FPGA families and different implementation options (Radiation-Hardened-By-Design (RHBD) Structured ASIC (S-ASIC) or RTL for implementation in any reconfigurable device). RTL can be implemented in any generic FPGA.

General features

- Based on a RISC processor with 8-bit data bus and 14-bit wide instructions, up to 64K instructions and a operation frequency of 50MHz.
- CRC16 calculation capability for error detection. Stores up to 1024 checksums.
- Error Detection And Correction (EDAC) optional
- Frame-based scrubbing capable.
- Read-detect-scrub capable.
- Single Event Failure Interrupt (SEFI) detection capabilities.
- Capable of fully configuring the device upon request.
- Supports SelectMAP32, SelectMAP8, ICAP32 and JTAG configuration interfaces.
- Expandable to interface with different forms of non-volatile storage.
- Reporting and statistics gathering capabilities through a RS232 port.
- Modular design: Can add or remove peripherals as needed. Basic module comprises the RISC controller, a Read Only Memory (ROM) interface, a CRC16 calculation module, a communication module (RS232) and a configuration memory interface (SelectMAP32).
- Available in IP form (to be implemented in a reconfigurable device) or in a rad-hard S-ASIC device

Setup options

Two configuration setups are possible: External scrubber of Self-scrubber. The setup of an external scrubber consist of a separate device from the FPGA subject to scrubbing as shown in Figures 1 and 2. In this case the scrubber can perform as a master or slave configuration controller.

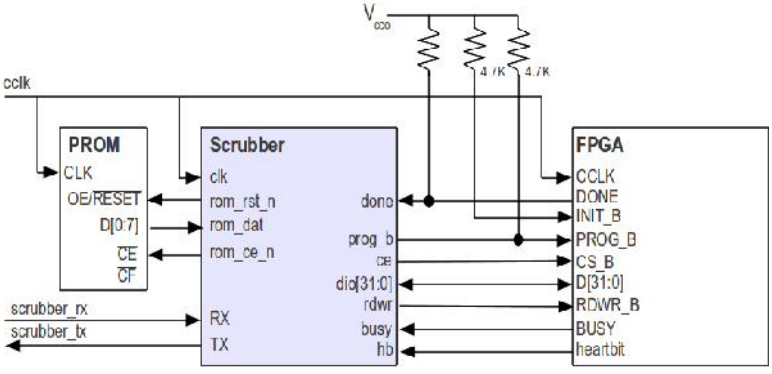


Fig. 1. External setup where the scrubber performs as a master configuration controller.

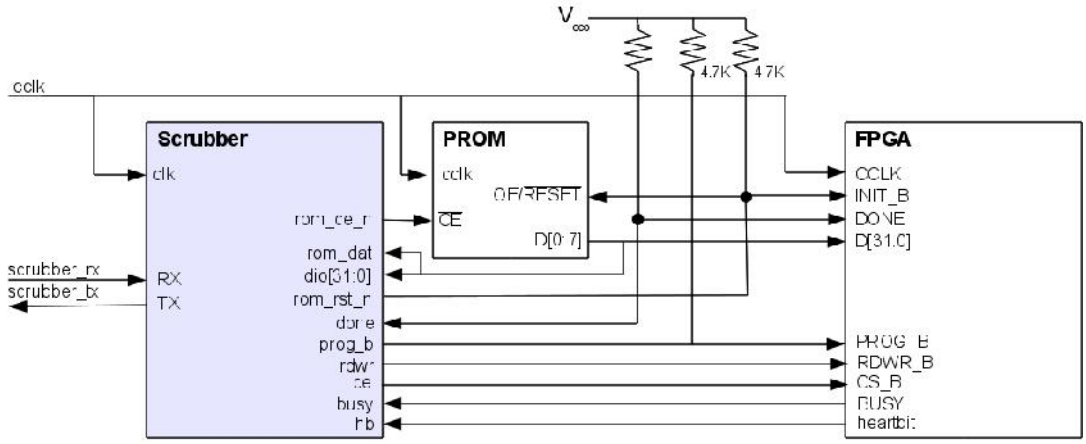


Fig. 2. External setup where the scrubber performs as a slave configuration manager.

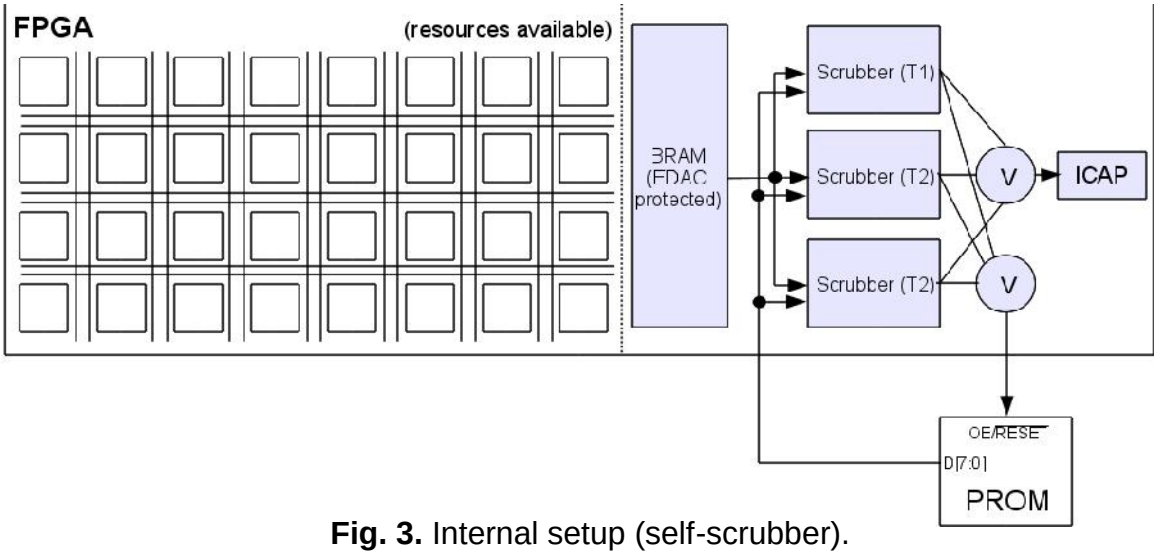


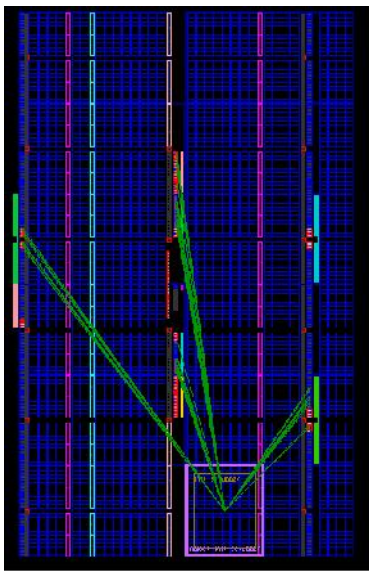
Fig. 3. Internal setup (self-scrubber).

When working as a master configuration manager, the scrubber sits between the read-only-memory (ROM) where the bitstream is stored and the FPGA. The scrubber is responsible for programming the FPGA upon power up and for checking and correcting errors in the configuration memory. Note that this configuration requires significant changes with respect to the classic Slave SelectMAP configuration mode.

The slave configuration manager setup is defined as the less intrusive deployment of the scrubber. Little changes are necessary between the classic Slave SelectMAP configuration mode and this setup. In this setup, the PROM is able to configure the FPGA by itself in case the scrubber is absent or disabled. The only provision necessary is to connect the FPGA's DONE pin to the PROM's CE, RDWR_B and CS_B to ground. The rest of the connections can be left alone and the PROM will correctly configure the FPGA upon power up even with the scrubber absent.

The “external” setups described in Figures 1 and 2 call for the implementation of the scrubber in a separated device. Current alternatives are a rad-hard FPGA or an RHBD ASIC design or S-ASIC fabric. Micro-RDC’s preferred solution is the later, leveraging its RHBD Via-configurable S-ASIC technology (see next section for details).

In the self-scrubber configuration, the RISC Cntrl and its peripherals are triplicated. A voter for the ICAP interface is implemented to vote out possible errors due to upsets in the user logic or the configuration memory. The scrubber program resides in a Block RAM protected by Error Detection (2 bits) And Correction (1 bit) (EDAC) codes. This is implemented in the instruction memory behavioral model, and it is transparent to the scrubber. The scrubber is locked to a very specific area of the FPGA, which the FPGA’s payload is not allowed to use. An example of the scrubber being confined to a section of a Virtex 5 FPGA is shown in Figure 4. The scrubbing algorithm can scrub the scrubber and the FPGA’s payload areas.



Device Utilization Summary (estimated values)			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Registers	750	28800	2%
Number of Slice LUTs	785	28800	2%
Number of fully used Bit Slices	243	1292	18%
Number of bonded IOBs	48	220	21%
Number of Block RAM/FIFO	1	48	2%
Number of BUFG/BUFGCTRLs	1	32	3%

Fig 4. Scrubber locked to a small section of the FPGA’s fabric. Resources estimated are shown for a single copy of the scrubber. A TMR implementation uses ~3x the amount of resources.

RHBD 90nm CMOS

Microelectronics Research Development Corporation (Micro-RDC), a leading developer of technology for radiation effects hardening-by-design microelectronics, and its partner ViASIC have developed a family of 90nm low power Design-Hardened Structured Application Specific Integrated Circuits (DH SASICs) that can be produced on a commercial IC fabrication line (See Figure 5).

This revolutionary approach supports the development of digital ICs for space, using hardened-by-design technology combined with the use of a Single-Reticle Multi-Project-Wafer (SR-MPW) commercial fabrication approach to provide advanced high-performance ICs, qualified for space, in a much shorter amount of time and at a fraction of the cost of developing even larger feature-sized process hardened custom ICs.

Time and costs for Micro-RDC's DH SASIC have been reduced by using Via-configurability techniques that allow the user to create a new semi-custom ASIC design by just modifying one Via-level mask.

The key features of the Design-Hardened S-ASIC fabric are:

- Hardness Estimate: TID > 3Mrad(Si), SER < 1e-10 Errors/bit-day, SEL > 75 MeV-cm2/mg(LET).
- Design-Hardened Fabric w/Distributed Dual Port SRAM (Logic Cell w/Memory in Each Tile)
- Logic using 1ns and/or 2ns delay, patented Temporal Latch structures provide SEU/SET mitigation.
- Design-Hardened PLL, Input Frequency: 10MHz to 125 MHz.
- Design-Hardened Configurable 8Kx8 Block SRAM Memory w/EDAC
- Design-Hardened Via-Programmable ROM.
- Design-Hardened Via-Programmable I/O
- Dual Voltage Supply; 1.2V core and 1.8V to 2.5V I/Os.

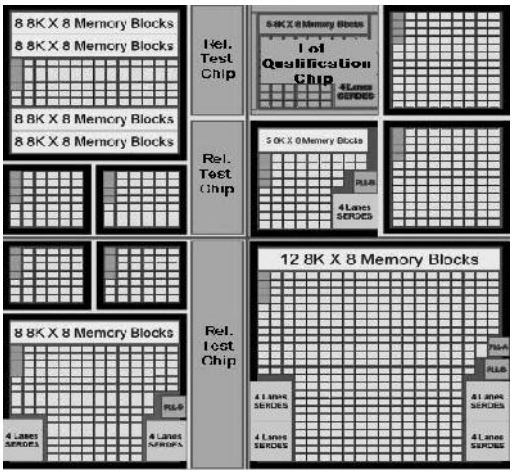


Fig 5. Micro-RDC's Via-configurable Structured ASIC MPW Reticle

Read-Only-memories supported

The scrubbing approach is only as effective as the storage of the bitstream's golden copies. If that data is corrupted the scrubbing approach will fail. The scrubber currently supports different interfaces to access a ROM and retrieve the information it needs to perform scrubbing. The default scrubber configuration considers a free running ROM (i.e XCFNN devices). In this case the scrubber must wait and identify the information it needs from the data broadcasted by the ROM. To accomplish this, the data in the ROM is partitioned and "tag IDs" are added as shown in Figure 6. Alternatively, the scrubber also supports memories with a NAND flash and simple RAM interfaces. For support of other interfaces please contact Micro-RDC.

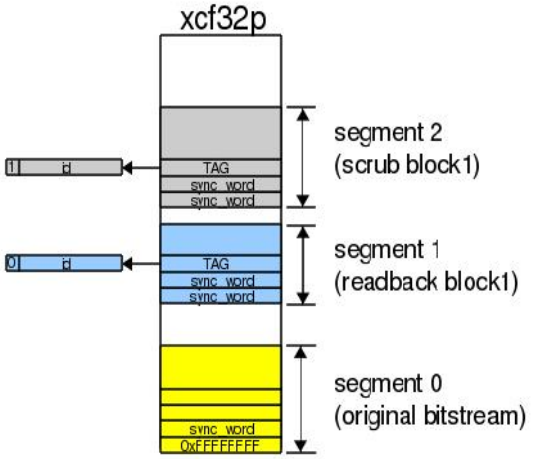


Fig 6. ROM data partitioning to store custom data to be used by the scrubber.

Default scrubbing algorithm

The scrubber is shipped with a default, full-device scrubbing algorithm. Since the scrubber is programmable, the user can implement variations of this algorithm (i.e. frame-based strategies, partial scrubbing, blind scrubbing, etc.)

Scrubber Pin-Out

Port	Width	Direction	Description
clk	1	Input	Input clock signal
rx	1	Input	RS232 input port.
tx	1	Output	RS232 output port.
rom_ce_n	1	Output	Optional chip enable signal to control ROM
rom_data	7	Input	ROM data part.
rom_rst_n	1	Output	Optional reset signal for the ROM. It can be used to restart the address counter if it is a free running ROM.
dio	32	Input/Output	Data bus to the SelectMAP or ICAP interfaces.
rdwr	1	Input	Read or write control signal for the SelectMAP and ICAP interfaces.
ce	1	Output	Chip enable signal for the configuration memory interface.
busy	1	Input	Busy signal from the configuration memory interface.
prog_b	1	Output	Signal used to reboot the FPGA.
hb	1	Input	FPGA's heartbit signal used for SEFI testing.