

Reconfigurable Avionics for Hubble Servicing Mission

Ed Cheung¹, Will Clement², and Ray Bietry³

¹ Jackson & Tull

² Clement Engineering

³ Orbital Science Corporation

The Hubble Space Telescope Project periodically constructs new instruments and avionic components for installation into the orbiting observatory. These components, called Orbital Replacement Units (ORUs), are carried to orbit on carriers aboard the Space Shuttle. We describe a new reconfigurable avionics that will fly along with the ORUs on their carriers. It will allow ground personnel to control heaters and read carrier telemetry. The new avionics will replace ageing components for which there is no existing replacement, and allow flexibility to our program to carry ORUs that are added to the manifest at the last minute.

In order to provide tolerance to Single Event Upsets events, our system will employ three Xilinx XCV600 reprogrammable FPGAs. Their outputs will be voted upon by one anti-fuse 54SX72 Actel FPGA. All I/O performed by the Xilinx units will be via high-speed serial busses communicating with the Actel. Messages received by the Actel via these busses control the actual I/O ports on the Actel. Thanks to the voting arrangement, the impairment of a single Xilinx will be suppressed, and the telemetry system will continue to function normally.

Each Xilinx FPGA will contain needed peripherals such as a Command Decoder, and a Telemetry Generator to communicate with the Space Shuttle on-board avionics. They are controlled by a microprocessor implemented as an IP core running embedded C code. An important aspect of this redundant FPGA system is the temporal synchronization of the three processors so that their outputs can be voted in a simple majority-wins manner. This synchronization is repeated every few minutes to counteract the effect of any SEUs on the state of the processor or its peripherals.

Since the Xilinx FPGAs are not truly radiation hardened, a key aspect of this triply-redundant design is the ability to detect and correct configuration errors. The task is facilitated by the fact that we use three identical Xilinx FPGAs rather than incorporating triply-redundant logic in a single, larger FPGA. Rather than comparing the configuration contents of the FPGAs to the original program load stored in PROM (which contains non-configuration data as well, and therefore requires a mask file), the use of three XFPGAs allows us to read back and compare the contents to one another. This simplification could be used to reduce the total power-up time of the PROM as well as eliminating the need for storage of the very-large mask file. Furthermore, using a single FPGA to hold all three instantiations of the logic requires that reconfiguration be performed in a manner which does not halt FPGA operation. Using three separate FPGAs, this design can detect and correct any single FPGA error without interruption of operations.

We describe here the mechanics of designing a system around triply-redundant Xilinx FPGAs. Details of readback and configuration for our design are given, as well as references to the Xilinx documentation from which they were drawn.

