Techniques to Enable FPGA Based Reconfigurable Fault Tolerant Space Computing

Grant L. Smith and Lou de la Torre
Honeywell Inc., Defense & Space Electronic Systems
13350 U.S. Highway 19 North
Clearwater, Florida 33764-7290
727-539-4311/ 727-539-2584
grant.l.smith@honeywell.com
lou.delatorre@honeywell.com

Abstract—[1,2] Reconfigurable computing using Field Programmable Gate Arrays (FPGAs) offer significant performance improvements over traditional space based processing solutions. The application of commercial-off-the-shelf (COTS) FPGA processing components requires radiation-effect detection and mitigation strategy to compensate for the FPGAs' susceptibility to single event upsets (SEUs) and single event functional interrupts (SEFIs). A reconfigurable computing architecture that uses external triple modular redundancy (TMR) via a radiation-hardened ASIC provides the most robust approach to SEU and SEFI detection and mitigation. Honeywell has designed a TMR Voter ASIC with an integrated FPGA configuration manager that can automatically reconfigure an upset FPGA upon TMR error detection. The automatic configuration manager also has features to support resynchronizing the upset FPGA with the remaining two FPGAs operating in a self checking pair (SCP) mode. Automating and minimizing reconfiguration times and resynchronization times enables high performance FPGA-based processors to provide high system availability with minimal software/system controller intervention.

TABLE OF CONTENTS

INTRODUCTION 1
FPGA RADIATION EFFECTS.....................................2
INTERPROCESSOR NETWORK 3
SYSTEM CONTROLLER...4
DETAILS OF THE FPGA INTERFACE.........................4
DETAILS OF THE VOTER LOGIC 5
CONFIGURATION MANAGEMENT DETAILS 5
TMR UPSET RATE 7
FPGA RE-SYNCHRONIZATION.................................9
TECHNOLOGY DEVELOPMENT 10
CONCLUSION 10
REFERENCES 10
ACKNOWLEDGEMENTS 11
BIOGRAPHIES 11

INTRODUCTION

Present and future space missions are requiring significant increases in on-board signal processing. The vast amounts of data being generated can not be transmitted via the downlink channels in a reasonable time. As the users of the information demand quicker access, more and more of the data reduction or feature extraction processing must be performed on the spacecraft. Increasing the spacecraft processing allows the down link data bandwidths to be used more effectively and the number of independent user channels to increase.

The traditional instruction based processor approaches are not able to compete with multi-million gate FPGA-based processing solutions in signal processing applications [3, 4, and 14]. The inherent ability of the FPGA to support a parallel approach to the signal processing task has been shown to offer up to 1000 times the processing performance of a standard 100MHz PowerPC based processor [7]. Systems with multiple FPGA-based processors could potentially meet the very large processing requirements of Space Based Radar (SBR), next generation adaptive beam forming and adaptive modulation space based communication programs.

As the name implies, an FPGA-based system can be easily reconfigured while in development or even after launch to meet new requirements. An FPGA-based reconfigurable space processing payload design can ideally be reused if architected to support changes to the typically unique data interfaces of each program. As the costs to develop space processing solutions continue to increase, there is strong support for leveraging payload development costs across multiple programs. The fact that requirements for many future programs are unknown dictates an easily scaleable, multiple-FPGA-based reconfigurable system approach.

From a con-ops (Concept of Operations) perspective, the reconfiguration capability of an FPGA-based system also offers the promise of extended mission life if mitigation

[1] 0-7803-9546-8/06/$20.00© 2006 IEEE

[2] IEEEAC paper #1592, Version 5, Updated December 24, 2005

methods can be designed to compensate for degraded elements in the system.

Reconfigurable processing solutions do come at a cost. Commercial of the shelf (COTS) SRAM-based FPGAs have shown sensitivity to radiation induced upsets. Although much work has gone into FPGA radiation-effect detection and mitigation strategies [12] with some very promising results, it is apparent that a purely COTS-based reconfigurable system approach is too unreliable for many space customers. A heterogeneous approach, mixing radiation-hardened ASICs with the COTs FPGAs appears to offer the best of both technologies [9].

The brute force approach for detecting and mitigating all known FPGA SEU and SEFI susceptibilities is to employ external triple modular (component) redundancy (TMR) as shown in figure 1-1. External TMR requires three separate FPGAs operated synchronously with one another. The control and data signals from each FPGA interface are voted against each other by a radiation-hardened ASIC. Triple component redundancy has several disadvantages such as higher power requirements, higher board real estate requirements, lower per-processor efficiency, and unusable system processing capacity. Some processing efficiency can be retained if provisions are designed in for a non-TMR mode of operation during inherently SEU-insensitive processing tasks such as data decompression. Environmentally adaptive techniques could also be used to predict opportunities for non-TMR modes. [11, 17]

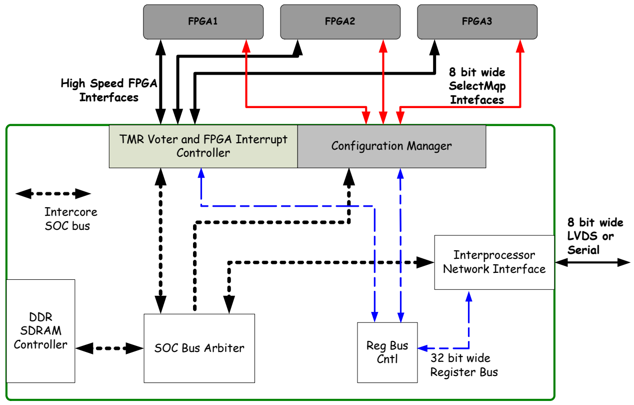

FPGA1
FPGA2
FPGA3
High Speed FPGA Interfaces
8 bit wide SelectMqp Intefaces
TMR Voter and FPGA Interrupt Controller
Configuration Manager
Intercore SOC bus
8 bit wide LVDS or Serial
Interprocessor Network Interface
DDR SDRAM Controller
SOC Bus Arbiter
Reg Bus Cntl
32 bit wide Register Bus

Figure 1-1 Radiation Hardened ASIC with FPGA Voter and Configuration Management Cores Shaded

Although device level TMR can also be used to mitigate errors due to electromagnetic interference (EMI), non-deterministic jitter, and poor signal integrity in the FPGA to ASIC data path, properly applied space level design processes, testing and qualification can typically reduce the probability of these error conditions to below that of the radiation hardened ASIC upset rate.

Resynchronization of all three FPGAs after one FPGA has a TMR fault can be accomplished several ways. It is important to minimize the time that the remaining pair operates in a self checking pair (SCP) mode. If the same radiation-hardened ASIC that performs the voting function also contains FPGA configuration management logic then the resynchronization can be accomplished automatically without system processor intervention. Resynchronization may require the use of a system controller (traditional instruction based processor on a radiation hardened single board computer) depending on the complexity of the algorithms being processed by the FPGAs. Resynchronization could also be complicated by the need for the two remaining FPGAs operating in a SCP mode to share state information with the reconfigured FPGA if performing complex state-machine-based functions.

In this paper, we review FPGA radiation effects and discuss system architecture with a fault tolerant interprocessor network at a high level. The features of our configuration management logic and TMR voter logic in the FPGA interface ASIC are discussed in detail. The reliability equations for a TMR system are reviewed and a curve is presented that allows the reader to determine if the external TMR technique can support their mission requirements. It will be shown that the availability of the processing assembly is a strong function of the time required to resynchronize an upset FPGA. Several resynchronization strategies required to support FPGA-based reconfigurable fault tolerant space computing are also discussed in detail.

FPGA Radiation Effects

In general, fault tolerant reconfigurable computing can be implemented with various degrees of processing availability based on the specific mission requirements and the time that can be allotted at the system level to an upset recovery process. Some critical applications such as real time attitude control cannot tolerate any interruptions to the algorithmic processing. Other critical applications such as store and forward communication applications are naturally immune to short processing interruptions due to the buffering of data and retry protocols implemented at the system level. This paper focuses on techniques required to support critical missions that are intolerant of any processor interruption.

FPGA-based processors can have enhanced SEU immunity using a combination of selective internal triplication of logic (Internal TMR, also called XTMR™ by Xilinx), configuration memory scrubbing and configuration memory refreshing [12]. Configuration SRAM memory scrubbing, or configuration memory refreshing while the FPGA is operational, can improve SEU immunity of the logical fabric but there are restrictions on the types of logical elements that can be used. [13] The largest portion of configuration SRAM is dedicated for the logic fabric and routing resources in the FPGA. The next largest portion of configuration memory is dedicated to the BlockRams inside the device. Although the FPGA's BlockRam could be accessed and checked for errors while the device is operational, access contention would likely cause problems for the functioning algorithms.

Virtex-4 FPGAs have increased the performance and complexity of their Input/Output Blocks (IOB). The FX family of devices has included multi gigabit transceivers (MGTs) capable of up to 10Gbps. Virtex-4 IOBs can be expected to become more susceptible to SEUs and potentially SEFIs with the increase in complexity and control bits. Fortunately, unique frames (addresses) of the configuration memory are used to program the IOBs so that periodic scrubbing and or periodic refreshing can remove IOB upsets while the logical fabric of the device is operational.

Virtex-4 FPGAs have shown a 6X improved immunity to atmospheric SEU events over Virtex-II devices. It has been reported that the improvement appears to be mainly due to the reduction from 0.15-0.12um to 0.09um feature size although the Virtex-4 does support BlockRam error checking and correction (ECC) and frame error checking (EC) logic. [16]

Internal TMR, configuration scrubbing, or refreshing cannot mitigate upsets to the internal control logic of the FPGA. The power-on-reset (POR) logic, Digital Clock Managers (DCM) and the hard-macro PowerPC (included in the Virtex-II Pro and FX series of Virtex-4 devices) all have dedicated control logic. Upsets to the control logic can result in single event functional interrupts (SEFIs). There have been several reported FPGA SEFI mechanisms in Xilinx FPGAs [12]. Some of them are as follows,

- A power-on-reset (POR) SEFI that causes an unexpected erasure of the entire part
- A SelectMap SEFI that changes and potentially locks up the SelectMap interface
- A JTAG SEFI that locks up the JTAG interface

Recent testing on the Virtex Pro series of devices has also revealed SEFI's in the hard-macro PowerPC 405 core inside the device. The same PowerPC 405 core(s) are also in the FX series of Virtex-4 devices

The number and cross section of the bits that cause these SEFIs are very low compared to the FPGA fabric configuration bits [12] but internal TMR techniques, scrubbing and refreshing cannot provide availability better than the FPGA's SEFI limit. It is shown in the probability analysis section of this paper that external TMR can provide processor availability better than the SEFI limit but only in recovery scenarios with relatively short resynchronization times.

Some radiation-induced effects such as single-event-transients (SETs) require further study [15] but may be able to be ignored by an external TMR voter since they are not persistent. More dynamic testing is required to fully characterize SET effects.

Interprocessor Network

An interprocessor communications network is required support high speed system data flows encountered in distributed processing applications using multiple reconfigurable processing assemblies. For several recent applications, Honeywell has chosen the RapidIO commercial industry standard [12].

RapidIO is recognized by many as the leading-edge fault tolerant high performance COTS standard interconnect. State-of-the-art payload-data-processor interconnects are mostly based upon multi-drop configurations such as Module Bus, PCI and VME. Multi-drop systems distribute available bandwidth over each module in the system but also produce points of contention among participant nodes often resulting in system level bottlenecks. In contrast, RapidIO implements a packet-switched, point-to-point interconnect allowing, multiple full-bandwidth point-to-point links to be simultaneously established between end-nodes in a network. RapidIO reduces contention and delivers more bandwidth to the application.

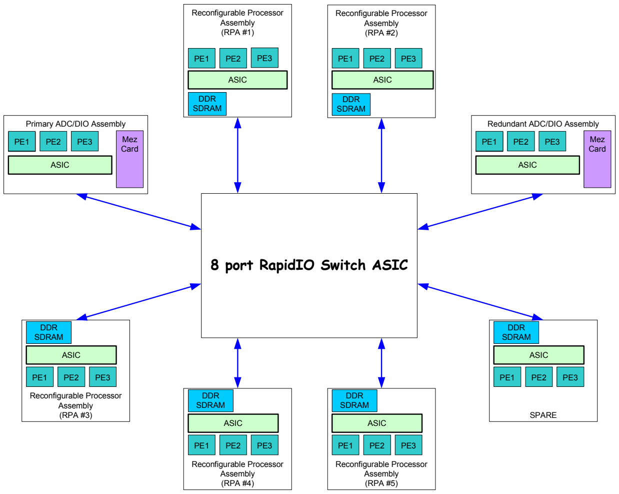

Reconfigurable Processor Assembly (RPA #1)
PE1
PE2
PE3
ASIC
DDR SDRAM
Reconfigurable Processor Assembly (RPA #2)
Primary ADC/DIO Assembly
Mez Card
Redundant ADC/DIO Assembly
8 port RapidIO Switch ASIC
Reconfigurable Processor Assembly (RPA #3)
Reconfigurable Processor Assembly (RPA #4)
Reconfigurable Processor Assembly (RPA #5)
SPARE

Figure 1-2 RapidIO Interprocessor Network

Figure 1-2 shows a RapidIO system network based on two building blocks: RapidIO end-nodes, and a RapidIO switch. Each end-node in the system is outfitted with a RapidIO network interface having a point-to-point link to a shared RapidIO switch. The switch receives and routes packets to the appropriate destination. The non-blocking nature of RapidIO allows concurrent routing of multiple packets. For example: input data can be routed to and stored in the globally available memory of one of the reconfigurable processing assemblies at the same time as the reconfigurable processor is sending results to another processing assembly. By using multiple switches in a system, topologies consisting of tens to hundreds of nodes can be achieved. Note that in figure 1-1 two (primary and redundant) of the assemblies are A/D and D/An external interface cards. The reconfigurable processing assemblies use an M of N redundancy to meet mission requirements.

RapidIO interfaces are based on LVDS source synchronous DDR signaling technology and can achieve bandwidths of many tens of Gbits/s for each active link. RapidIO can be implemented with a parallel or serial interface. A 16-bit wide parallel RapidIO link operating at 250MHz is capable of 16 Gbits/s providing >15x performance increase over a 33 MHz 32-bit CompactPCI-based interface. Honeywell is prototyping a radiation hardened Quad SERDES that will support 4 serial RapidIO links with an aggregate signaling rate of 24Gbits/sec. (3.125 Gb/sec per differential signal pair). The RapidIO switch implemented to date is an eight port, eight bit wide parallel device.

A notable benefit of the RapidIO protocol is its extensive fault tolerant error-detection and recovery mechanisms. By combining retry protocols, cyclic redundancy codes (CRC) and single/multiple error detection, RapidIO handles most network errors without system controller intervention. This inherent error handling and recovery capability proves ideal for space applications that require a highly reliable interconnect.

System Controller

In case a TMR error is detected, a rapid reconfiguration of the faulted FPGA is critical to provide the highest possible processing availability. A system controller could be interrupted when the TMR error is detected, but interrupt response times are typically much longer than can be achieved with dedicated configuration management logic. Configuration manager logic in the radiation-hard ASIC can relieve the system controller of the detailed reconfiguration and periodic partial refreshing processes.

A good candidate for a fault tolerant system controller is the Honeywell radiation-hardened RHPPC Single Board Computer (SBC). The radiation-hardened SBC is based on Motorola MPC603e microprocessor technology [3]. The system controller is used as a platform for critical system control software such as responding to error conditions and the initialization of the radiation-hard application specific integrated circuit (ASIC) that hosts the TMR FPGA interface and configuration manager. In a typical flight system, the FPGA configuration Bitstreams would be stored as compressed files on the system controllers EEPROM. The system controller would uncompress these files, properly format them, and transfer them to the local DDR SDRAM on the reconfigurable processor assembly.

Details of the FPGA Interface

A simplified block diagram of a typical Honeywell reconfigurable processing assembly that contains three separate FPGAs and a radiation-hardened ASIC is shown in figure 1-1. Note that the TMR Voter logic and configuration manager sections are shaded. As discussed above, the ASIC also has a high speed interprocessor network interface, an internal high speed system-on-chip (SOC) bus, a command and control register bus, a DDR SDRAM controller and a SOC high speed bus arbiter.

The FPGAs have two separate interfaces to the radiation-hard ASIC, a high speed TMR FPGA interface used by the voter logic and a slower FPGA configuration SelectMap interface used by the configuration manager logic. Both of these interfaces are shown as full duplex. The high speed TMR FPGA interface is routed using unidirectional paths to mitigate the FPGA IOB upsets that can turn a bi-directional IO into a constant input or a constant output. [12] Tristate FPGA IO have also been shown to switch to a constant output, potentially causing bus contention and reliability concerns [15] if the interface uses the same net for both directions of data flow.

A source synchronous double-data-rate (DDR) interface can be supported since each data source also sends clock. Sending clock relieves the routing constraints and skew control concerns typical of high speed interfaces and improves the integrity of the voting operation. Each signal is voted against the corresponding signal from the other FPGAs.

Table 1-1. FPGA Interface Key Features

FPGA Interface Key Features
FPGA Input Synchronizer
Rotational Arbiter (for non-TMR mode)
TMR/DMR Voter
Fault Decoder
Cumulative Fault Counter
Fault Rate Counter
Receive FIFO
Receive Controller
Command FIFO
Command FIFO Controller
Transmit FIFO
Transmit Controller
FPGA Data Output Synchronizer
SOC High Speed Data Interface (to DDR SDRAM)
SOC Register Interface
Event Generator

The input synchronizer uses the clock sent by the FPGA to sample the input data and control signals. In our present design, each FPGA interface is routed with equal length to support voting on a clock cycle by clock cycle basis.

The rotational arbiter is only used in a non-TMR mode, where each FPGA competes for access to the high speed SOC bus. The frame signal is used to signal the voter when a FPGA wants to gain access to the SOC bus. The multiplexer selects which FPGA gets access. Since this paper is focused on enabling fault tolerant reconfigurable computing for space environments, we will not discuss non-TMR features.

DETAILS OF THE VOTER LOGIC

The details of the FPGA Voter logic are shown in figure 1-4. Note in the diagram the signals from the FPGA are prefixed with pe_, which stands for processing element (PE). The voter circuit uses combinational logic to compare each signal against the corresponding signal from the other two FPGAs. As seen from the logic diagram, if two of three corresponding signals are a one (zero) then this block produces a one (zero).

The fault detection block is used to determine which FPGA is miscomparing. The output pattern from this block is all ones if all FPGAs agree. If one FPGA miscompares then two of the signals will be zero. The two that agree cause one of the signals to remain a one. The two agreeing FPGAs would continue to operate in a self-checking-pair (SCP) or dual-modular-redundant (DMR) mode. Once an FPGA has been determined to be at fault, miscompares between the two remaining FPGAs in SCP mode signal a fatal error. The probability of a SCP fatal error is the limit to the processing availability of an external TMR system. The probability is analyzed in detail later in the paper. When a SCP error occurs, the system controller must begin a complete recovery sequence on all three FPGAs.

The FPGA Voter logic also contains a cumulative error counter that can be used over the life of the mission to gather statistics on the SEU or SET rate (or BER of the interface). This cumulative error counter is not used to determine a faulty FPGA. A separate error-rate counter is used to determine if more than an acceptable number of miscompares have occurred in a row. It has been reported that up to 45% of FPGA upsets are SET related. [15].

The miscompare block is followed by the Receive FIFO that allows a buffering of FPGA data before a request is made to the SOC bus arbiter for access to the SOC bus. Once the FPGA is granted access, the FPGA requested read or write transactions are executed.

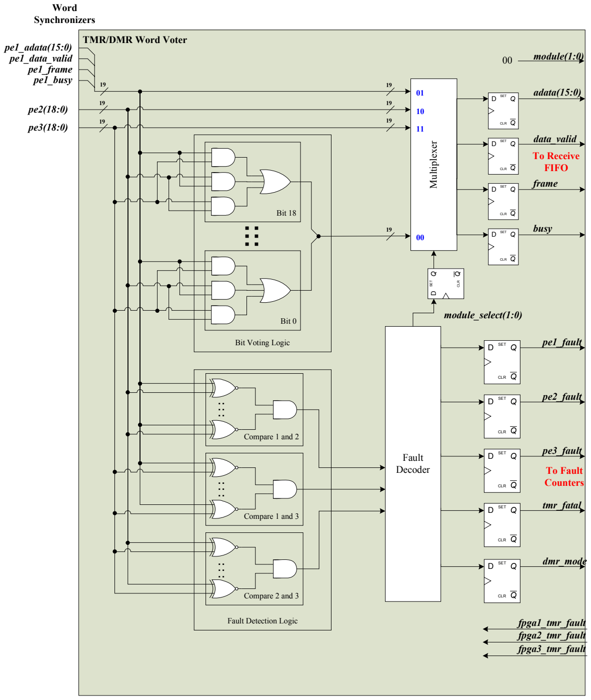


Figure 1-4. FPGA Interface with Voter Detail

CONFIGURATION MANAGEMENT DETAILS

The configuration management block performs several functions with minimal system controller interaction but does require some initialization. As can be seen in figure 1-1, the configuration manager logic is a part of a larger ASIC that also includes a DDR SDRAM controller. As discussed in the System controller section, the FPGA configuration Bitstreams are stored in the local DDR SDRAM on the reconfigurable processor assembly. The system controller must also program a couple of registers in the configuration manager with the location and size of the bitstream. After initialization, the configuration manager can be commanded to simultaneously configure all three FPGAs in parallel (at board start up or periodically to guarantee synchronization) or to individually configure a single FPGA. The configuration manager can also be set up to automatically initiate a sequence of commands to the FPGA that has been determined at fault by the FPGA voter logic. Configuring from DDR SDRAM supports rates much faster than can be obtained with legacy radiation hardened memories. The following commands are supported by the configuration manager.

- Reset
- Erase
- Config
- Abort
- Start

Since some of these commands would temporarily disable the ability of the FPGA to process, and processing availability is critical, they are protected in the configuration manager with a write protect bit but plus an Arm and Fire sequence. A block diagram of the configuration manager logic is shown in figure 1-5.

As can be seen in the block diagram, the configuration manager has three separate SelectMap interfaces, one for each of the FPGAs (signals designated PE_). This is provided so that one of the FPGAs can be reconfigured independently of the other two (e.g. if it had a TMR fault) while the other two FPGAs continue to operate in SCP mode. Separating the SelectMap interfaces also protects against the possibility that an upset or SET on one interface could cause a corruption to the bitstream being loaded into the other two.

Also shown in figure 1-5 is the configuration manager's SOC bus master capable of issuing block read requests to the on-chip DDR SDRAM memory controller. A ping/pong buffer in the configuration manager stores up to 256 byte blocks of bitstream data. While one buffer's contents are being sent out to the FPGA, the other buffer can be filled by the SOC bus master. If reading back the configuration SRAM of the FPGA is desired, the SOC bus master would issue write requests to store the data into the on-board DDR SDRAM.

To support an automatic resynchronization process the configuration manager accepts four inputs from the FPGA voter interface, a TMR Enabled signal and one each for each FPGA.

To alert the system controller on the progress of any commands (e.g. erase complete) the configuration manager also outputs several event signals to the ASIC interrupt generation logic. These interrupts are sent via RapidIO as doorbells to the system controller. The configuration manager block also contains a counter that can be programmed to periodically refresh portions of an FPGA's configuration memory (e.g. the IOB bits).

Note that the FPGA Reset signals (not traditionally associated with the FPGA SelectMap interface) are controlled by the configuration manager logic. The simultaneous release of Reset (Start command) is used to start up all three FPGAs at the same time so that the voter logic can function in TMR mode. In addition, the configuration manager can be set up to hold each FPGA's DONE pin low until all three FPGAs have attempted to release DONE.

Key features of the configuration manager logic are summarized in Table 1-2.

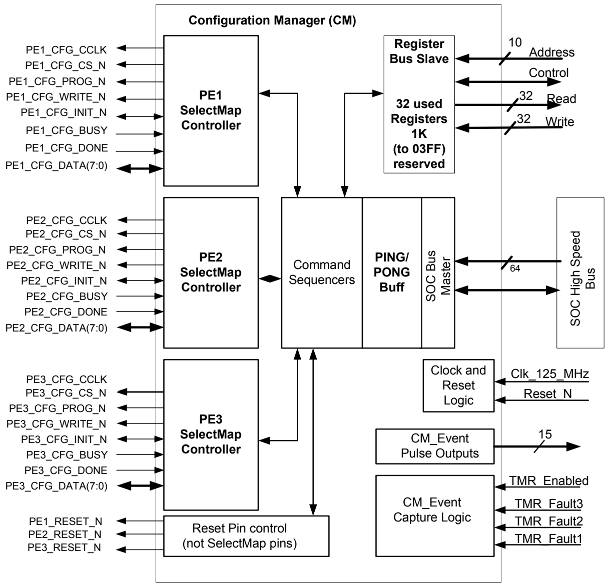


Figure 1-5. Configuration Manager Block Diagram

Table 1-2. Configuration Manager Key Features

Configuration Manager Key Features
Three independent FPGA SelectMap Interfaces
Configuration Command Controller
FPGA BUSY and INIT_B monitoring
Erase Command Controller
Reset Command Controller
Abort Command Controller
SOC High Speed Data Interface (used to get Bitstreams from DDR SDRAM)
SOC High Speed Interface Data buffers
Automatic Sequencing Controller
CRC Error Detection Logic
Timeout Detection Logic
Partial Refresh Timer (used for IOB refresh)
Event Generator

The configuration manager uses a start address and size programmed by the system controller to get the Bitstream from the DDR SDRAM via the SOC Bus Interface. PING/PONG buffers in the configuration manager allow the bitstream to be sent to the FPGA uninterrupted. While one buffer is being filled the other buffer is being transferred into the FPGA. When the entire file been sent successfully, the configuration manager issues a config_complete interrupt to the system controller.

The Virtex-II devices can assert BUSY to flow control the configuration data being sent into the device. In the Virtex-4 devices, BUSY is only used during the readback of the configuration memory to flow control the turn around of the SelectMap data bus. The configuration manager monitors the BUSY signal from the FPGAs during configuration so that it can be used for Virtex-II devices at the highest possible configuration clock rates. If any of the three FPGAs asserts BUSY, transfer is halted until BUSY is negated. If one FPGA hangs with BUSY asserted, a timer times out and lets the two remaining FPGAs continue with the configuration process. Configuring one FPGA at a time would multiply by three the time it takes to recover from a TMR or SCP event in some resynchronization scenarios.

The FPGAs automatically check the CRC of the Bitstream as it is loaded into the part. A CRC mismatch is signaled by the FPGA with an assertion of INIT_B. The configuration manager monitors the INIT pin and sends an interrupt if the FPGA reports a CRC error.

When commanded via an ARM and Fire sequence from the system controller the configuration manager causes an FPGA Erase by enabling a timer to pulse the FPGA's PROG pin low for the minimum acceptable time of 300ns. The INIT_B pin is also monitored so that an interrupt can be sent to the system controller when the FPGA announces that it has completed its erase cycle.

An important command from a fault tolerant perspective is the Abort command. Radiation testing has revealed one or more SEFI mechanisms in the SelectMap interface. One way to detect the SEFI is to attempt to readback the Frame Address Register in the FPGA. Failure to read the correct value indicates that a SEFI has occurred. Another mitigation method that has been reported is to periodically command an Abort to the SelectMap interface. After the Abort is received by the FPGA, the SelectMap data bus is turned around and the FPGA outputs four status bytes. The configuration manager reads and stores these bytes for the system controller to interpret but the primary use of the Abort command is to reset the SelectMap control logic, clearing any upset control bits.

TMR Upset Rate

The external FPGA TMR Voter logic can support processor availabilities better than the SEFI limit of a single device, if after a TMR event, the remaining two FPGAs continue to operate normally in a self checking pair (SCP) mode. The limit to the processing availability in the external TMR case is the chance that two or more TMR errors occur within an application dependant recovery time.

Radiation events are a random process with a Poisson distribution, characterized by an event rate, λ (*l*ambda). The upset rates for Virtex devices are a complex function of the radiation type, energy of the particles (LET spectra) radiation fluence and angle of incidence. Several other system design parameters such as shielding, orbit profiles etc. significantly contribute to the actual flight upset rates. The SEE Consortium [12] has been publishing reports to support the estimation of the upsets rates. Honeywell has studied several orbital profiles and used several models such as CREME to determine expected upset rates on the Xilinx FPGAs.

The math discussed in this paper for the SCP fault probability using external TMR is a subset of a more general mission reliability study for systems that have more than one spare element. External device level TMR is simply a case where at least two of three processors are required to meet mission requirements.

The general mathematical representation for the reliability (Probability of Success) of a unit regardless of the failure rate distribution is:

$$P_s\,(t) = e^{-\int_0^t \lambda(\tau)\,d\tau}$$

where $\lambda(\tau)$ is the failure rate of the unit as a function of time. If $\lambda(\tau)$ is constant, then the expression reduces to:

$$P_s\,(t) \quad = e^{-\lambda\int_0^t d\tau}$$

$$= e^{-\lambda t}$$

The reliability of a system consisting of *m* primary active units (devices) and a standby spare unit (device) can be calculated as follows.

Let lambda, λ, represent the primary processing element (device) upset (failure) rate and let it be a constant.

Let λ_s represent the spare processing device upset rate and let it be a constant.

In our case the standby device is always active and has the same upset rate, $\lambda_s = \lambda$ although a more general derivation case allows for different upset rates.

Of course there is a chance that none of the units ever upset, but that figure is small since our reconfigurable processors are based on the relatively sensitive SRAM based FPGAs.

Two recovery process times can be analyzed. In an event driven scenario, the recovery time starts when one of the primary units upsets. In a periodic scenario, the resynchronization opportunities are fixed and preplanned. In this case, the probability that a unit upsets at time t_1 prior to the resynchronization time T must be determined.

In the event driven scenario, the processing reliability (or processing availability) in such a system is the probability that a primary unit and the spare unit survive the entire resynchronization process period T after one of the primary units upsets.

In the periodic resynchronization scenario, the processing reliability (or processing availability) is the probability that none of the units upset *and* the probability that only one of the units upset at some time $\boldsymbol{t_1}$ prior to ***T.*** The remaining primary unit and the spare unit survive the time remaining ($\boldsymbol{T\text{-}t_l}$) until the resynchronization opportunity. This scenario is similar to a standard mission reliability analysis where ***T*** represents the mission life requirement.

For the periodic resynchronization scenario

The reliability (or processing availability) in such system is the probability that the *m* primary units survive the entire resynchronization period ***T*** $\{P_{\text{s-prim}}(T)\}$; *or* that one of the primary units upsets at some time $\boldsymbol{t_1}$ prior to ***T*** $\{Q_{\text{p}}(t_1)\}$ *and* the standby unit has not upset prior to $\boldsymbol{t_1}$ $\{P_{\text{s-stby}}(t_1)\}$ *and* this standby unit survives the rest of the resynchronization period $\{P_{\text{s-stby}}(T - t_1)\}$.

$$R(T) = P_{\text{s-prim}}(T) + Q_{\text{p}}(t_1)\, P_{\text{s-stby}}(t_1)\, P_{\text{s-stby}}(T - t_1)$$

The probability of *m* primary units surviving the entire resynchronization period is:

$$P_{\text{s-prim}}(T) = e^{-m\lambda T} = R(t)$$

The upset (mortality) rate of the primary units is defined as:

$$m_{\text{o}}(t) = -\frac{dR(t)}{dt} = m\lambda e^{-m\lambda t}$$

The probability that one of the primary units fail at some time $\boldsymbol{t_1}$ prior to ***T*** is then:

$$Q_{\text{p}}(t_1) = \int_0^T m_o(t_1)dt_1 = m\lambda \int_0^T e^{-m\lambda t_1}\, dt_1$$

The probability that the standby unit has not upset prior to t_1 is:

$$P_{\text{s-stby}}(t_1) = e^{-\lambda s t_1}$$

The probability that the standby unit survives until the resynchronization time is:

$$P_{\text{s-stby}}(T - t_1) = e^{-\lambda(T - t_1)}$$

Therefore, in combining the four probabilities, the reliability of the system is then:

$$R(T) = e^{-m\lambda T}$$

$$+ m\lambda \int_0^T e^{-m\lambda t_1} e^{-\lambda s t_1} e^{-\lambda(T - t_1)} dt_1$$

$$= e^{-m\lambda T} + m\lambda e^{-m\lambda T} \int_0^T e^{-\lambda s t_1}\, dt_1$$

$$= e^{-m\lambda T} + \frac{m\lambda e^{-m\lambda T}}{\lambda_s}\left(1 - e^{-\lambda s T}\right)$$

R(*T*) represents the probability that the external TMR approach would provide uninterrupted processing during the resynchronization time window.

To use the periodic resynchronization equations for external device level TMR, *m* is equal to 2.

So, if $\lambda = \lambda_s$ and *m* = 2, then the above expression reduces to:

$$= e^{-2\lambda T} + 2e^{-2\lambda T}\left(1 - e^{-\lambda T}\right)$$

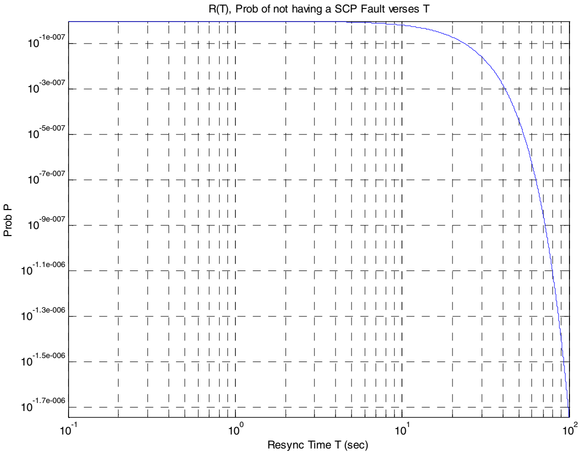
$$= 3e^{-2\lambda T} - 2e^{-3\lambda T}$$

Equation 1 Probability of uninterupted processing during *T* using external TMR.

If we use a device upset rate of 1 upset per day (1.157E-05 upsets per sec) and a resynchronization time ***T*** of 10 seconds, the probability of surviving ***T*** seconds without a system upset is:

R(10) = 0.9999999598

The chart below shows the probability of not having a SCP fault (providing uninterupted processing) as a function of the recovery time ***T*** with lambda equal to 1 upset per device day.



Note that the Y axis units of $10^{-xe(-y)}$ are a shorter way to represent numbers very close to 1. For example $10^{-8e(-7)}$ = 0.99999920000032

From the equations and the above chart, it is apparent that to improve system processing availability, either *m* can be increased or resynchronization time, ***T,*** can be decreased. (Assuming lambda, $\boldsymbol{\lambda}$, is fixed).

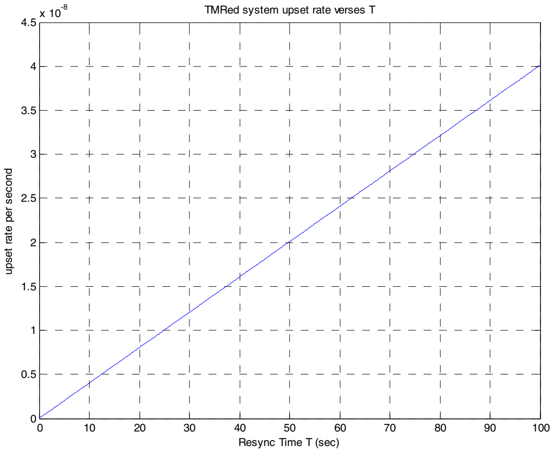
A new TMRed system upset rate can be derived from this probability estimate by using the relationship:

$$R(t) = e^{-\lambda t}$$

Solving for λ at ***T***=10 yields an effective TMRed system upset rate of 4.0E-9 faults per second or 3.472E-04 faults per day. This is an improvement of almost 3000x the upset rate for a single device (1 upsets per day).

The chart below shows the new TMRed system upset rate (per second) as a function of the recovery time ***T*** with device upset lambda equal to 1 upset per device day.

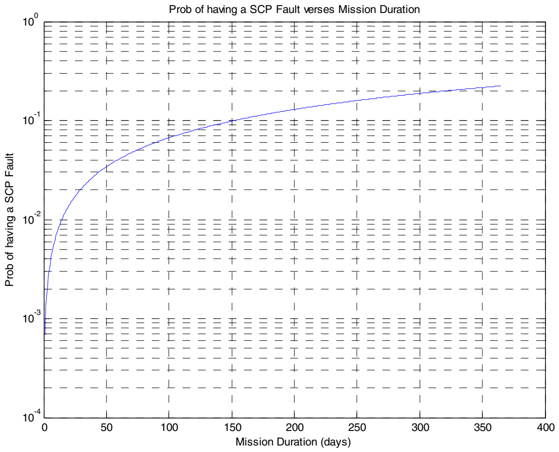
It has been discussed that device upset rates are very mission dependant. There have been reports on Geosynchronous Earth Orbits (GEO) with expected FPGA upset rates of less than 10 per device day and Space Station Orbit upset rates of less than 3 per device day [12, 15]. It has also been reported that up to 90 percent of the FPGA configuration memory is unused or only used for routing resources and if these bits are upset they would be less prone to causing a functional error [15]. FPGA upset simulators, like those available from LANL, can be used to estimate the upset rates on the specific cores instantiated into the proposed flight design.



Note that it scales linearly as expected.

Using a TMRed system daily upset rate 3.470E-4 allows us to calculate the probability of having a SCP fault over a mission life of 1 to 365 days.

Even if a SCP fault does happen, all three FPGAs can be reconfigured and brought on line relatively quickly based on several application dependant issues. No radiation testing to date on the Virtex-II devices has been seen to require a power cycle [12]. All upsets including SEFIs can be corrected with an erase and reconfiguration.



Note that after 150 days the chance of having a SCP fault (loss of processing) is approximately 10%. (device upset rate of 1/day and a resynchronization time equal to 10 sec).

FPGA RE-SYNCHRONIZATION

Different signal processing applications have different optimum procedures for bringing a faulted FPGA back into synchronous operation with the two remaining FPGAs (that are continuing to operate in a SCP mode). As can be seen above, it is important to minimize the resynchronization times and re-enable the TMR voter logic. Resynchronization can be verified by having the system controller monitor the FPGA interface TMR cumulative error counter and see that it is not incrementing.

The simplest resynchronization scenario occurs when there is an inherent periodic opportunity to disable processing on all three FPGAs and perform a simultaneous reconfiguration on all three in parallel. This is equivalent to a power up scenario. Obviously, the DDR SDRAM must be preloaded with the configuration bitstream and any data the FPGAs need before the configuration is commanded.

Another resynchronization scenario is where blocks of data being processed by the FPGAs are periodically refreshed. This scenario is typical of a communications, radar or image processing application. The configuration of the faulted FPGA can be performed in the background while the two other FPGAs are processing the last data set. A simple reset and start command can be used to bring all three FPGAs into synchronous operation. All are now ready to operate on the new set of data stored in DDR SDRAM. A reset and start (release of Reset) can be performed in as little as 100us. In this scenario, a periodic reconfiguration of all three FPGAs is still desired to scrub out any upsets to the FPGA's configuration memory that have gone undetected by the voting logic.

A more complicated scenario is where the FPGA's are performing attitude control or other critical processing that has state-space or adaptive control algorithms being executed. In this case, the entire set of data required by an FPGA is not continuously refreshed. Some of the data depends on the history of the algorithm. The system controller must be used to coordinate a backup of this state information for use by the faulted FPGA. After the back up is successful, all three FPGAs can be reconfigured or only the faulted FPGA can be reconfigured but all three will need to the interrupted by the system controller to start up with the same state information.

Technology Development

The concepts described in this paper have been developed largely under Honeywell IRAD funding. Some of the hardware has been developed with customer funding across several different flight reconfigurable computing programs. Honeywell's foundry is actively working to space-qualify a 676-pin-count package for our radiation hardened ASICs to support fast and wide interfaces to the relatively high pin count FPGAs. The Xilinx FPGAs in this architecture use the class V flow CF1144 package under development with the help of Honeywell package qualification experts and Aerospace Corp. [10].

A software development unit (SDU) system prototype consisting of COTs parts has been successfully built, tested and shipped. All components were successfully integrated with application software to represent a functional fault tolerant reconfigurable processing system. The prototype is being used to further refine the signal processing algorithms executing on Xilinx FPGAs. The ASICs designs have been prototyped using Xilinx Virtex-II Pro100s and are targeted toward Honeywell's HX5000 .15um radiation hardened CMOS foundry.

Conclusion

Fault tolerant FPGA based reconfigurable computing for critical space processing applications can be achieved through the use of device level TMR and a separate radiation hardened voter ASIC with integrated configuration management logic. An example fault tolerant reconfigurable system was introduced and some of the features such as the fault tolerant interprocessor communications network were discussed. The radiation effects of the Virtex family of FPGAs were reviewed in preparation for a detailed discussion of features required by our radiation hardened voter and configuration management logic. The reliability equations that allow us to predict processing availibilites as a function of resynchronization times for an external TMR system have been reviewed. Several charts plotting the probability of a self checking pair fault (loss of processing) have been provided as well as a final chart on the probability of experiencing a loss of processing over a mission duration. It is shown that minimizing the resynchronization time is critical to improving the mission success. Several scenarios outlining approaches to FPGA resynchronization were reviewed.

References

[3] Gary R. Brown, "Radiation Hardened PowerPC 603e ™ Based Single Board Computer," 20th Digital Avionics Systems, 2001. Oct 2001

[4] AccelChip Inc, White Paper. "Comparison of Methods for Implementing DSP Algorithms". www.accelchip.com

[5] Thomas Cesear, AccelChip Inc. White Paper. "Optimizing Performance of DSP Systems through Block Level Design". www.accelchip.com

[6] E. R. Prado et al.,"A Standard Approach to Spaceborne Payload Data Processing", IEEE Aerospace Conference, March 2001.

[7] F. Irom et al., "Single-Event Upset in Evolving Commercial Silicon-on-Insulator Microprocessor Technologies, Nuclear and Space Radiation Effects Conference 2003

[8] Xilinx Corporation, "QPro Virtex 2.5V Radiation Hardened FPGA", Nov. 2001, Xilinx Paper, http://www.xilinx.com/

[9] J.S. Donaldson, "Push the DSP Performance Envelope", Xilinx Xcell Journal, Spring 2003

[10] Mark Dunn, Xilinx Class V Flow Document, Custom Requirements for Xilinx Ceramic Flip Chip Package Assembly. Honeywell Internal Document Oct. 2005

[11] P. Ellis and C. J. Walter, "Fault Tolerant Discovery and Formation Protocols for Autonomous Composition of Spacecraft Constellations", IEEE Aerospace Conference, 2003, pp 837-852.

[12] Xilinx Corp. “Xilinx Single Event Effects 1st Consortium Report, VirtexII Static SEU Characterization”, Jan. 2004

[13] C. Yui, G. Swift, Carl Carmichael, Rocky Koga and Jeffrey George, “SEU Mitigation Testing of Xilinx VirtexII FPGAs”, Candice Poster Session NSREC Monterrey, 2003

[14] Steven Leibson, Technology Evangelist, Tensilica, Inc “Configurable Processors What, Why, How”. June 2005, http://www.tensilica.com/

[15] Michael Caffrey, Paul Graham, Eric Johnson, and Michael Wirthlin, Los Alamos National Labs and BYU, "Single-Event Upsets in SRAM FPGAs" MAPLD 2002 Paper P8.

[16] Austin Lesea, Xilinx Corporation “Virtex-4 Reduces Soft Errors Rates Six Times” October 2005 Tech eXclusive Article, http://www.xilinx.com/

[17] Jeremy Ramos, Dean Brenner Honeywell Corp "EAFTC: An Enabling Technology for COTS based Space Computing" MAPLD 2005

ACKNOWLEDGEMENTS

We would like to thank the following people and organizations for their contributions to this work: Sana Rezgui of Xilinx Corporation and the entire Honeywell team including Jeremy Ramos, Cliff Kimmery, Tom Brown, Mark Allen, Lee Hoffman, Dave Campagna, and Angela Alexander.

BIOGRAPHIES

Grant L. Smith has been a Principal Systems Engineer with Honeywell Defense and Space Electronic Systems since 2004. He has 20 years of experience in the fields of optical sensor signal processing, gigabit rate DWDM optical communications, RF/microwave communication system design, digital signal processing, RFIC design and reconfigurable payload processing development for space systems. Mr. Smith received a B.S. and M.S. Degree in Electrical Engineering from the University of South Florida in 1985 and 1987. He also spent several years working toward his Ph.D. in Electrical Engineering in III/V optical/microwave device growth and characterization.

Lou de la Torre is a Staff Reliability Engineer with Honeywell Defense and Space Electronic Systems. He has over 18 years of experience in the field of Reliability Engineering of Space Electronic Systems. Mr. De la Torre earned his B.S. degree in Electrical Engineering from the University of Florida in 1986 and his M.S. degree in Electrical Engineering from the University of South Florida in 1993.