XAPP988 (v1.0) March 13, 2008

Correcting Single-Event Upsets in Virtex-4 Platform FPGA Configuration Memory

Authors: Carl Carmichael, and Chen Wei Tseng

Summary

Designers of space-based application must be concerned with the effect of single-event upsets (SEUs) on FPGA configuration memory. Changes to configuration memory can cause changes in the functionality and performance of the device. This application note describes the use of configuration scrubbing and readback in Virtex™-4 FPGAs for the purpose of detecting and correcting single-event upsets to the configuration memory array induced by cosmic rays.

It is essential for the reader to have a basic understanding of the Virtex-4 SelectMap interface as well as configuration and readback operations. An in-depth review of UG070, *Virtex-4 User Guide*, is strongly encouraged.

Introduction

In-orbit, space-based, and extra-terrestrial applications must consider the effects high-energy charged particles (radiation) can have on electronic components. In particular, SEUs can alter the logic state of any static memory element (latch, flip-flop, routing pip, or RAM cell). Since the user-programmed functionality of an FPGA depends on the data stored in millions of configuration latches within the device, an SEU in the configuration memory array can have adverse effects on the expected functionality.

A static upset in the configuration memory is not synonymous with a functional error — upsets might have no effect on functionality. Mitigation techniques, such as using TMRTool to implement triple module design redundancy, or triple FPGA deployment, can harden the application against single-events upsets. However, the upsets must be corrected so that errors do not accumulate. For information about TMRTool, please refer to the UG156, *Xilinx TMRTool User Guide*.

The Virtex-4 FPGA SelectMap interface provides the most efficient, post-configuration read/write access to the configuration memory array. The only other alternative interface is through the Boundary-Scan port, also known as the JTAG interface. Readback is a post-configuration read operation of the configuration memory, and configuration scrubbing is a post-configuration write operation to the configuration memory. Readback and scrubbing can be used to allow a system to detect and repair SEUs in the configuration memory without disrupting its operations.

This application note is presented with the assumption that the reader has a solid understanding of basic configuration and readback operations, bitstream format, configuration command structure of the Virtex-4 configuration logic, and SelectMap interface. A careful review of the latest user guide can provide this information (refer to UG071, *Virtex-4 Platform FPGA Configuration User Guide*).

© 2008 Xilinx, Inc. All rights reserved. XILINX, the Xilinx logo, and other designated brands included herein are trademarks of Xilinx, Inc. All other trademarks are the property of their respective owners.

Configuration Memory Architecture Overview

Virtex-4 configuration memory is arranged in frames, the smallest addressable segments that are tiled about the device. In contrast to previous generation of Virtex families, Virtex-4 architecture is composed of fixed-length frames, each consisting of 41 words (Figure 1). Virtex-4 default configuration sizes are shown in Table 1.

Table 1: **Virtex-4 Bitstream Size**

Device	Configuration Bit[1]
XQR4VLX200	51,367,808
XQR4VSX55	22,745,216
XQR4VFX60	21,002,880
XQR4VFX140	47,856,896

Notes:

1. Default bitstream size = configuration bits + overhead. This number should match the configuration length noted in the header information of the .rbt file generated by BitGen.

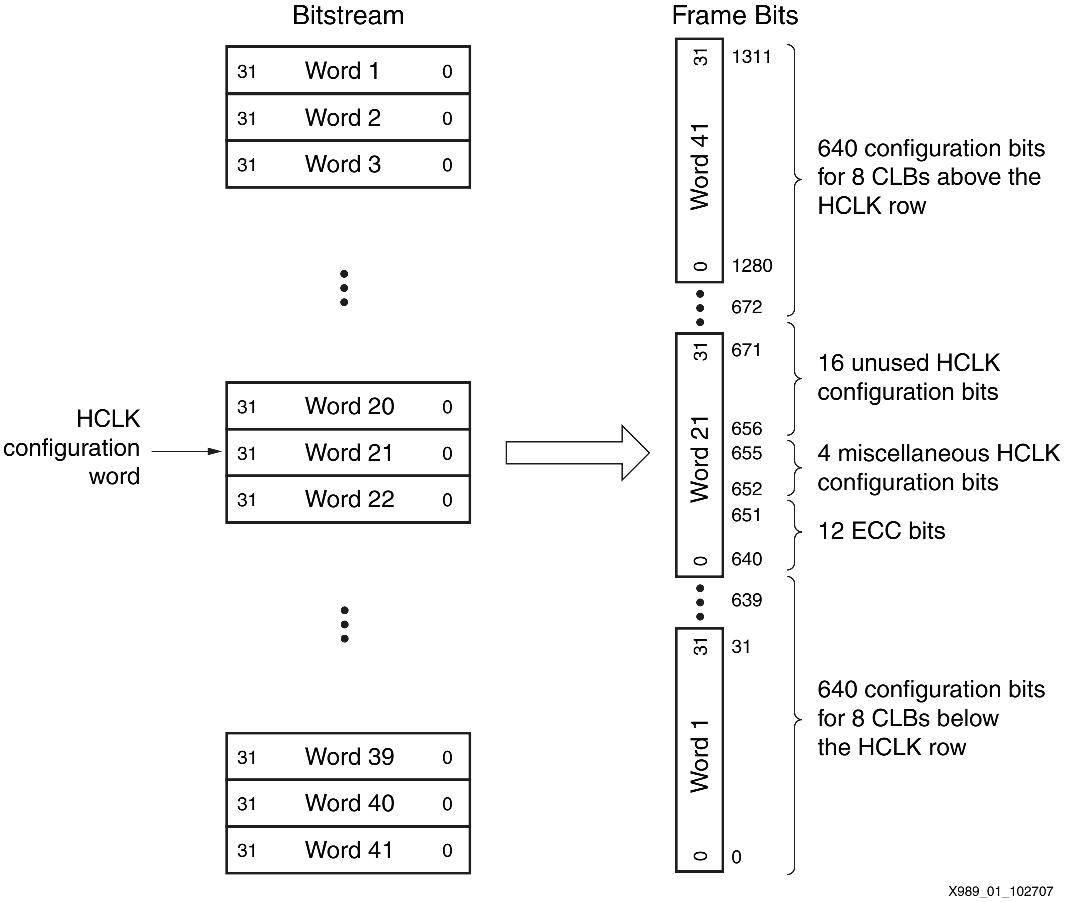

Figure 1: **Configuration Words in the Bitstream and Configuration Bits in a Frame**

To access and perform configuration or readback operations, 32-bit data packets are issued to target different configuration registers. Configuration data packets are made up of a 32-bit type-1 header (Table 2) that can be followed by either a 32-bit data body or a 32-bit type-2 packet header (Table 3).

Table 2: **Type 1 Packet Header**

Type			WR	RD	Register Address														RSVD		Word Count										
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	1	x	x	R	R	R	R	R	R	R	R	R	x	x	x	x	x	0	0	x	x	x	x	x	x	x	x	x	x	x

Table 3: **Type 2 Packet Header**

Type			WR	RD	Word Count																										
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Configuration Registers

All bitstream commands are executed by reading or writing to the configuration registers (Table 4). A detailed explanation of selected registers follows.

Table 4: **Virtex-4 Configuration Register Address for Type 1 Packet**

Register Name	Read	Write	Address	Description
CRC	Y	Y	`00000`	CRC Register
FAR	Y	Y	`00001`	Frame Address Register
FDRI	N	Y	`00010`	Frame Data Register, Input (write configuration data)
FDRO	Y	N	`00011`	Frame Data Register, Output (readback configuration data)
CMD	Y	Y	`00100`	Command Register
CTL	Y	Y	`00101`	Control Register
MASK	Y	Y	`00110`	Masking Register for CTL
STAT	Y	N	`00111`	Status Register
LOUT	N	Y	`01000`	Legacy Output Register (DOUT for daisy chain)
COR	Y	Y	`01001`	Configuration Option Register
MFWR	N	Y	`01010`	Multiple Frame Write Register
CBC	N	Y	`01011`	Initial CRC Value Register
IDCODE	Y	Y	`01100`	Device ID Register
AXSS	Y	Y	`01101`	User Bitstream Access Register

Frame Address Register

Configuring and reading back configuration frames requires the frame address to be specified. Virtex-4 device is broken into top and bottom halves. Frame address register (FAR) initiates from the upper-center and left-most column of the device (Table 5). The FAR auto-increments starting with the minor address, followed by column address, row address, top/bottom bit, and finally block type.

The minor address increases until it reaches the targeted column type frame count. Then column address increments while minor address resets. Row address increments when column address has cycled through the right most column. Once the FAR increments through the top half of the device, it begins with the bottom half of the device and repeats the process from the center left. The bottom half of the device is the mirror image of the top half.

The overhead column is used to enable smooth row address transition. This column holds no configuration information and is composed of two frames. It exists only if frame data input register (FDRI) or frame data output register (FDRO) word counts cross the row boundary and must be accounted for in FDRI or FDRO word calculation. This overhead column also allows the readback of the last block RAM interconnect column type frame without requiring the first block RAM content frame be loaded into the configuration pipeline (which could risk corrupting block RAM content).

Since the overhead column does not physically exist in the hardware, it does not have a frame address. Therefore, at the end of the row, if frame address is read out while the device is transitioning to the next row, the frame address remains at the last physically existing address until the FAR increments to the next row.

As shown in Table 5, the Virtex-4 FAR has rearranged block-type assignments such that block RAM interconnects are occur prior to block RAM content, simplifying the readback process.

Note: The spreadsheet, (xapp988_virtex4_architecture.xls, provided with the reference design gives more details.

Table 5: **Frame Address Register**

Bit Index	Address Type	Description
31:21	Not Used	Cannot be written to and remains all 0s.
22	Top / Bottom	Selects between top-half or bottom-half rows. Default selects top-half rows.
21:29	Block Type	Block types are: `000` – CLB/IO/DSP/CLK/MGT `001` – block RAM interconnect `010` – block RAM content
18:14	Row Address	Selects a row of frames. Row address increases with the distance from the center in both the up and down directions.
13:6	Column Address	Selects a major column. Column addresses start at 0 on the left and increase going to the right.
5:0	Minor Address	Selects a single memory cell address line within the major column. The address count for each column type is: CLB (22) IO (30) DSP (21) CLK (2) MGT (20) Block RAM interconnect (20) Block RAM content (64)

Frame Data Register, Input

Writes to FDRI configure frame data at the frame address specified in the FAR. The FDRI is used for both full configuration and active partial reconfiguration scrubbing. The number of FDRI words must fall exactly on a frame boundary.

Frame Data Register, Output

The configuration logic reads back the number of words specified by the FDRO. The number of readback words must fall exactly on the frame boundary with the exception of the SelectMap 8-bit interface. The SelectMap 8-bit interface should add one word to the FDRO readout to account for the additional dummy word. More details on the readback operation are discussed in "Device Configuration Memory Readback (Optional)," page 16.

Command Register

The command register is used to instruct the configuration control logic to perform various functions. Table 6 lists the command register commands and codes.

Table 6: **Virtex-4 Command Register Codes**

Command	Code	Description
NULL	0000	**No Operation.** Used to clear configuration commands.
WCFG	0001	**Write Configuration Data.** Used prior to writing configuration data to the FDRI.
MFWR	0010	**Multiple Frame Write Register.** Used to perform a write of a single frame to multiple frame addresses.
LFRM	0011	**Last Frame.** Deasserts the GHIGH_B signal, activating all interconnect. The GHIGH_B signal is asserted with the AGHIGH command.
RCFG	0100	**Read Configuration Data.** Used prior to reading configuration data from the FDRO.
START	0101	**Begin Startup Sequence.** Initiates the startup sequence. The startup sequence begins after a successful CRC check and a DESYNC command are performed.
RCAP	0110	**Reset Capture.** Resets the CAPTURE signal after performing readback-capture in single-shot mode.
RCRC	0111	**Reset CRC.** Resets the CRC register
AGHIGH	1000	**Assert GHIGH_B Signal.** Places all interconnect in a high-Z state to prevent contention when writing new configuration data. This command is only used during shutdown reconfiguration and readback. Interconnect is reactivated with the LFRM command.
SWITCH	1001	**Switch CCLK Frequency.** Updates the frequency of the master CCLK to the value specified by the OSCFSEL bits in the COR.
GRESTORE	1010	**Pulse the GRESTORE Signal.** Sets/resets (depending on user configuration) IOB and CLB flip-flops.
SHUTDOWN	1011	**Begin Shutdown Sequence.** Initiates the shutdown sequence, disabling the device when finished. Shutdown activates on the next successful CRC check or RCRC instruction (typically an RCRC instruction).
GCAPTURE	1100	**Pulse GCAPTURE.** Loads the capture cells with the current register states.
DESYNC	1101	**Reset DALIGN Signal.** Used at the end of configuration to desynchronize the device. After desynchronization, all values on the configuration data pins are ignored.

Control Register

The control register (CTL) sets many device options as listed in Table 7. To write to the control register, the MASK register must first be written with '1' at locations where the control register data bits are to be enabled for the write process. An example of control register write is provided as part of the scrubbing process in Table 18.

Table 7: **Control Register**

Name	Bits	Description
RESERVE	31	Reserved
ICAP Select	30	Selects top or bottom ICAP access: 0: Top ICAP 1: Bottom ICAP
RESERVE	29-9	Reserved
GLUTMASK_B	8	Global LUT mask signal: 0: Disable readback and reconfiguration to SRL16 and LUTRAM 1: Enable readback and reconfiguration to SRL16 and LUTRAM
RESERVE	7-6	Reserved
Security Bits	5-4	Security level: 00: Read/Write OK (default) 01: Readback disabled 1x: Readback and writing disabled (except CRC register)
Persist	3	The configuration interface defined by M2:M0 remains after configuration. Typically used only with the SelectMAP interface to allow reconfiguration and readback: 0: No (default) 1: Yes
RESERVE	2-1	Reserved
GTS_USR_B	0	Active Low high-Z state for I/Os: 0: I/Os placed in high-Z state 1: I/Os active

Status Register

The status register (STAT) reflects numerous global signals and the current state of the FPGA. Description of each bit position is provided in Table 8.

Table 8: **Status Register**

Name	Bits	Description
Reserved	31-17	Reserved
DEC_Error	16	FDRI write attempted before or after decrypt operation: 0: No DEC_ERROR 1: DEC_ERROR
ID_Error	15	Attempt to write to FDRI without successful DEVICE_ID check: 0: No ID_ERROR 1: ID_ERROR
Done	14	Value on DONE pin.
Release_Done	13	Value of internal DONE signal: 0: DONE signal not released (pin is actively held Low) 1: DONE signal released (can be held Low externally)
INIT	12	Value on INIT pin
INIT_Complete	11	Internal signal indicating initialization has completed: 0: Initialization has not finished 1: Initialization finished
Mode	10-8	Status of the MODE pins (M2:M0).
GHIGH_B	7	0: GHIGH_B asserted 1: GHIGH_B deasserted
GWE	6	0: FFs and block RAM are write disabled 1: FFs and block RAM are write enabled
GTS_CFG_B	5	0: All I/Os are placed in high-Z state 1: All I/Os behave as configured
EOS	4	End of Startup signal from startup block: 0: Startup sequence has not finished 1: Startup sequence has finished
DCI_Match	3	The logical AND function of all the MATCH signals (one per bank). If no DCI I/Os are in a particular bank, the bank's MATCH signal = 1: 0: DCI not matched 1: DCI is matched
DCM_Lock	2	The logical AND function of all DCM LOCKED signals. Unused DCM LOCKED signals = 1: 0: DCMs not locked 1: DCMs are locked
Part_Secured	1	0: Decryptor security not set 1: Decryptor security set
CRC_Error	0	0: No CRC error 1: CRC error

Configuration Option Register

The configuration options register (COR) is used to set certain configuration options for the device. The name of each bit position in the COR is described in Table 9.

Table 9: **Configuration Option Register**

Name	Bits	Descriptions
Reserved	31-29	Reserved.
CRC_Bypass	28	0: CRC enabled. 1: CRC disabled.
Reserved	27-26	Reserved.
Done_Pipe	25	The FPGA waits on DONE (delayed by one StartupClk cycle). Use this option when StartupClk is running at high speeds: 0: No pipeline statue for DONEIN 1: Add pipeline stage for DONEIN
Drive_Done	24	0: DONE pin is open drain 1: DONE is actively driven High
Single	23	New captured values are loaded on each successive CAP assertion on the CAPTURE_VIRTEX4 primitive. Capture can also be performed with the GCAPTURE instruction in the CMD register: 0: Readback is not single-shot. 1: Readback is single-shot. The RCAP instruction must be loaded into the CMD register between successive readbacks.
OSCFEL	22-17	Select CCLK frequency in Master configuration modes.
SSCLKSRC	16-15	Startup-sequence clock source: 00: CCLK 01: UserClk (per connection on the CAPTURE_VIRTEX4 block) 1x: JTAGClk
Done_Cycle(1)	14-12	Startup cycle to release the DONE pin.
Match_Cycle(1)	11-9	Startup cycle to stall in until DCI matches.
Lock_Cycle(1)	8-6	Startup cycle to stall in until DCMs lock.
GTS_Cycle(1)	5-3	Startup cycle to deassert the global three-state (GTS) signal.
GWE_Cycle(1)	2-0	Startup cycle to deassert the global write-enable (GWE) signal.

Notes:

1. 000: Startup cycle 1
   001: Startup cycle 2
   010: Startup cycle 3
   011: Startup cycle 4
   100: Startup cycle 5
   101: Startup cycle 6

IDCODE Register

Each device has an embedded IDCODE (Table 10) register that can be read and written used for identifying the device. When writing to the IDCODE register, the embedded value does not get overwritten. The write operation simply initiates a comparison process where the written value is compared against the embedded value. If the value (other than the revision field of the IDCODE) matches, the following FDRI operation is allowed to execute; otherwise, the device declares a CRC error, halts the ensuing FDRI operation, and pulls INIT Low.

Table 10: **IDCODE Register Field Explanation**

Revision Code[1]				Family Code[2]							Device Size[3]									Manufacturer ID											[4]
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
X	X	X	X	F	F	F	F	F	F	F	S	S	S	S	S	S	S	S	S	0	0	0	0	1	0	0	1	0	0	1	1

Notes:

1. Revision code of IDCODE is used to indicate the device revision and is *not* checked by the configuration logic.
2. Family code is `0b0001011` for LX, `0b0010000` for SX, and `0b0001111` for FX.
3. Device size is the summation of device rows and columns.
4. The last bit is always 1 to conform to the JTAG IDCODE specification.

Table 11: **Virtex-4 Qpro-R IDCODE**

Device	IDCODE (HEX)
LX200	`X1734093`
SX55	`X20B0093`
FX60	`X1EB4093`
FX140	`X1F14093`

Single-Event Upset Overview

Most upsets within the configuration memory do not have any effect because the majority of configuration memory cells are not relevant to a typical user design. While routing control bits account for over 60% of the configuration bits, a typical design uses only approximately 10 to 20% of the available routing resources. However, an upset to utilized configuration memory bits can interrupt the user design. This type of regional or device-wide interference caused by an SEU is referred as single-event functional interrupt (SEFI). To quickly and properly identify SEFIs, a combination of different detection methods should be implemented. A strong detection method can quickly and accurately identify a SEFI, while a weak detection method might only detect an event slowly or not at all.

Virtex-4 FPGAs have been fully characterized for SEFI events in proton and heavy ion testing. The observed SEFIs and their distinguishing characteristics are as follows:

- **Power-On-Reset (POR)**

  A POR-SEFI causes similar behavior to that of the device powering on, or as if PROG is asserted.
- **SMAP and JCFG**

  The SMAP and JCFG SEFIs correspond to the use of the SelectMAP and JTAG ports, respectively. Although they have different cross-sections (probability of occurrence), they are considered exclusive to each other under the assumption that only one configuration port would be utilized by a system for FPGA configuration management. Both SEFIs result in the loss of post-configuration control (readback and configuration scrubbing).
- **FAR**

  A SEU resulting in the inability to read or write the frame address register, or inadvertently triggered the FAR's auto-increment function.
- **Global Signals**

  Numerous device-wide configuration controls can be upset. Signals such as global write enable (GWE), global tri-state control (GTS), GHIGH, etc., can be upset, resulting in loss of configuration logic or device function.

Note: All SEFIs can be recovered by pulsing the PROG pin of the FPGA. All non-SEFI SEUs can be corrected with configuration scrubbing.

SEU Detection Methods

The following detection methods (summarized in Table 12) are devised for each known SEUs and SEFIs with a unique detection characteristic. Combining these methods provides an all-encompassing solution for mitigating SEUs and SEFIs. However, some applications might not employ all methods depending on the designer's intent to automatically scrub or implement a readback detection-only circuit, or whether configuration bit errors are to be counted and tracked. While there are many options for the designer to consider, this document provides an example combining all of the following methodologies into a complete mitigation solution:

- **Monitoring DONE**

  After device configuration, the DONE pin should remain High. If at any time, DONE is de-asserted, a full reconfiguration should follow.
- **Writing and reading FAR**

  This operation should be performed before each scrubbing and readback operation and can quickly identify SelectMAP interface problems or configuration logic disturbances that can result in inability to update FAR or corruption to FAR's auto-increment feature.
- **Calculating and comparing a readback CRC**

  When performing readback, the readback data can be used to calculate a CRC value on a frame or device basis. The calculated CRC value should then be compared against the golden CRC value. If the values mismatch, scrubbing operation should follow. A golden CRC value needed for this comparison can be calculated using a predetermined value generated during configuration from the PROM data, generated from an initial readback after configuration verifies with no bit errors, or stored after two consecutive readback and scrub operations yields the same CRC value.

- **Monitoring BUSY**

  During write operations, the BUSY pin should stay Low. The BUSY pin is pulled High during the start of a read operation; de-assertion of BUSY indicates that valid readback data is available on the data bus. BUSY remaining High for more than 32 clock cycles after a transition from write to read is a good indication of an SMAP SEFI.

- **Verify readback data**

  During readback, the readback data is compared against the configuration data to ensure no differences. CRC values can be calculated from the readback data at this time.

  Note: Verification operations require access to the non-volatile configuration memory storage device.

- **Checking status register and control register values**

  After device configuration, the device should reflect the default status register value of `0x00007EFC,` and the default control register value of 0x2000109. Readback register values differing from the expected values can indicate a device upset. Refer to Figure 4, page 15 for more information.

Table 12: **Virtex-4 SEU Detection Methods**

SEU Detection Method	Strong for Detecting
Monitoring DONE	POR SEFI
Writing and reading a predefined value to FAR	SelectMAP SEFI FAR SEFI
Calculating and comparing CRC value on the readback data	Non-SEFI SEU
Monitoring BUSY	SelectMap SEFI
Read and verify readback data against configuration data	Non-SEFI SEU
Monitoring status and control registers	Global signal SEFI POR SEFI

SEU and SEFI Mitigation

There are various SEU mitigation schemes that can be combined to accommodate a particular application. Before choosing a technique to mitigate SEUs, designers should consider both the criticality of the design as well as the hostility of the environment.

Table 13 shows some example error rates for a Virtex-4 XQR4VSX55 and earlier Virtex families in a geosynchronous orbit (solar quiet conditions) for the configuration memory, block memory, and POR-SEFI. For a fully TMR design with a configuration management engine, the resulting functional device error rate is dominated by the summation of the POR and SMAP SEFI rates.

Table 13: **GEO (36,000 km) Error Rates Using CREME96**

FPGA Family	POR	Configuration	Block RAM	Units
Virtex	1.56e–6	5.79e–8	1.42e–7	Errors/device-day
Virtex-II	1.32e–5	4.87e–7	4.62e–7	Errors/device-day
Virtex-4 (SX55)	1.06e–7	3.21e–7	1.21e–6	Errors/device-day

If the application cannot afford any possible downtime or sudden disruption within the SEFI error rates for the applicable orbit, then the mitigation scheme should employ redundant FPGAs, a configuration management engine, plus a downstream, radiation-hardened output voter.

If the device SEFI upset rate is acceptable for the mission and the environment is harsh, the design should be triplicated with TMRTool and a configuration management engine employed to prevent SEU accumulation in the configuration memory. For more information on TMRTool, refer to http://www.xilinx.com/products/milaero/tmr/index.htm

Some applications have an inherent duty cycle as a part of normal operation, either powering down on a predetermined schedule or frequently idling the FPGA, allowing for soft-reboots or full reconfiguration. For applications where the FPGA can be fully reconfigured at a higher rate than the configuration memory upset rate, a mitigation scheme might not be required.

For more information on selecting an appropriate mitigation scheme, please refer to XAPP987, *Single-Event Upset Mitigation Selection Guide*.

Note: Power-cycling is not necessary for repairing SEUs or SEFIs in a Xilinx Virtex-4 FPGA. All SEUs to the configuration memory can be mitigated by scrubbing. All SEFIs are recovered by pulsing the PROG pin and fully reconfiguring the FPGA. Single-event transients (SETs) and SEUs in user storage elements can be mitigated through design redundancy (XTMR).

Configuration Management Engine

For most orbital applications, a configuration management engine is required to prevent SEUs from accumulating in the configuration memory and to detect and recover from SEFIs. SelectMap (SMAP) provides the most efficient and comprehensive device access for mitigation implementation; thus, this application note focuses on using the SMAP 8-bit interface (SMAP8) for the configuration management engine.

Note: Alternative interfaces include JTAG and the SelectMap 32-bit (SMAP32) interfaces (which are not covered in this document). The fundamentals of implementation are the same for all interfaces with JTAG having to account for the additional JTAG command overhead and SMAP32 needing to align the data word accordingly.

Various implementations can be deployed to detect and correct SEUs. Figure 2 demonstrate a common implementation in which the configuration management engine is hosted by an external radiation-hardened device.

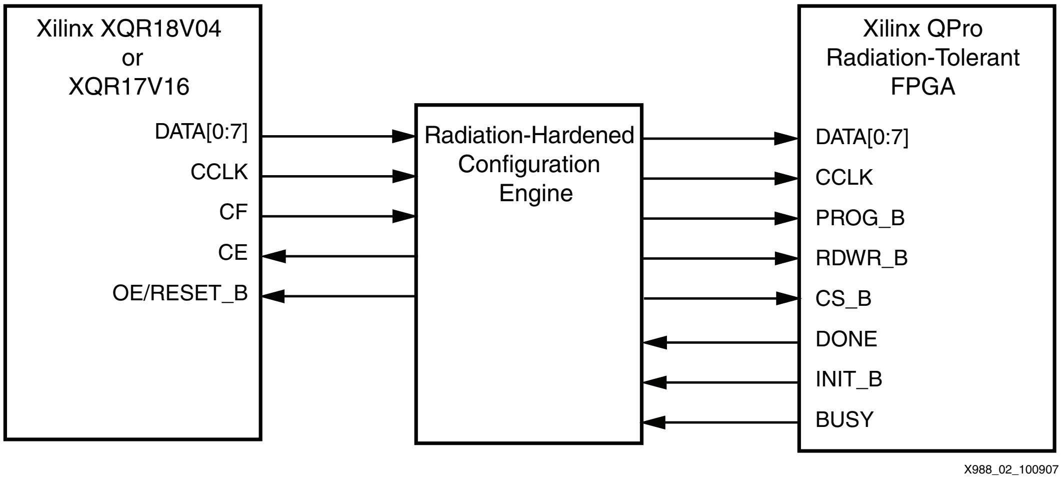

Figure 2: **Overview of an External Device Hosting Configuration Manager for a Virtex-4 Device**

Figure 3 illustrates a configuration management flow to ensure prompt detection and correction of SEU in the configuration memory or SEFI.

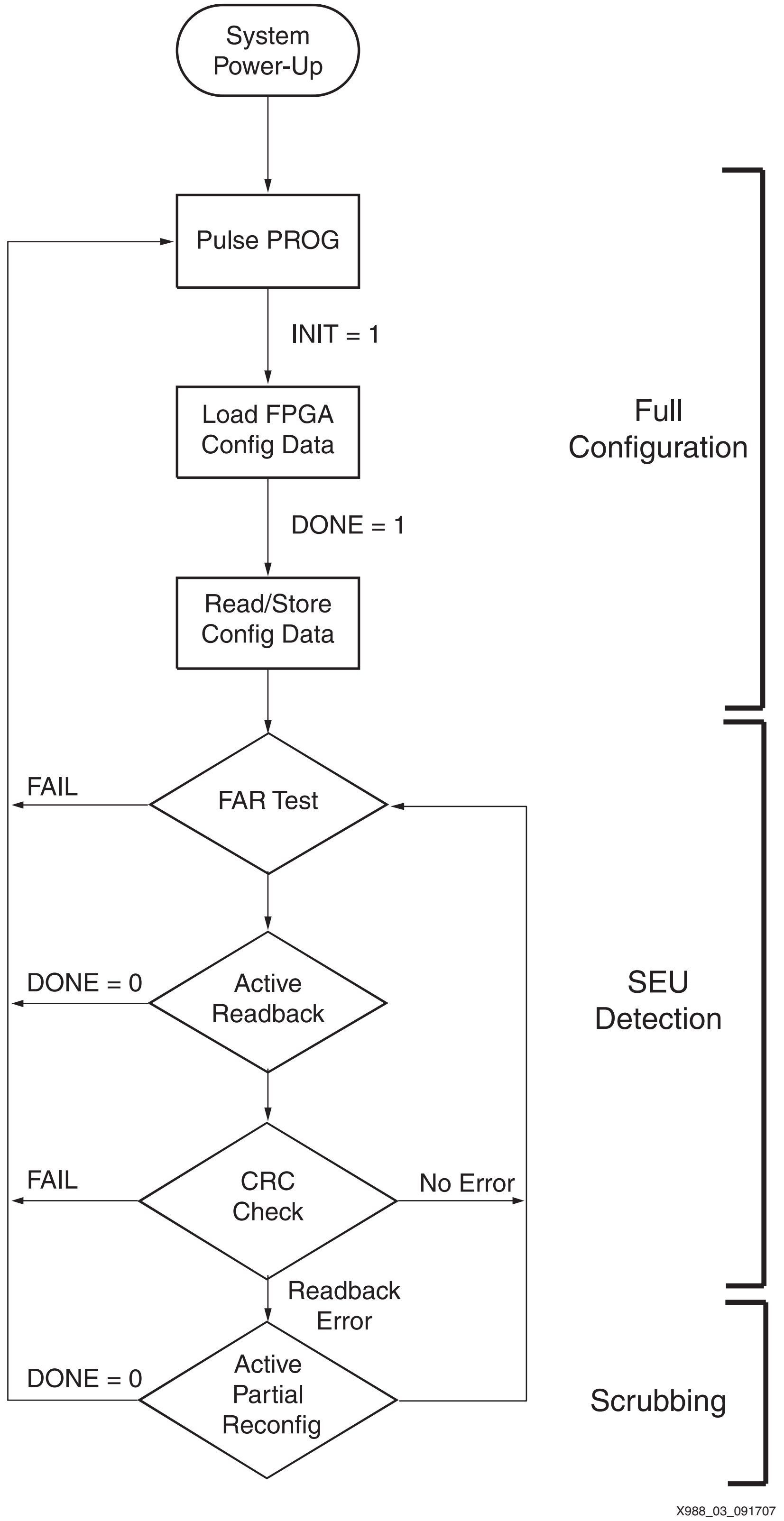

Figure 3: **Configuration Management Flow Chart**

The reference design is built for maximum configuration logic visibility as well as rapid SEFI recovery and implements the flow illustrated in Figure 3. For flight applications, it is strongly recommended to update the reference design for specific mission needs. For more information regarding the reference design, refer to the readme file in the reference design zip file.

A VHDL reference design for the flow is also available:

http://www.xilinx.com/support/documentation/application_notes/xapp988.zip.

Full Configuration

Upon system power up, the configuration management engine starts programming the FPGA as soon as INIT pin is taken High or after the T_{POR} requirement specified in the device data sheet. It is strongly recommended to always pulse the PROG pin for a minimum of 300 ns before any configuration to fully initialize the FPGA and accommodate SEFI recovery. The engine can load the configuration data from radiation-hardened memory devices, typically a PROM, to configure the FPGA.

INIT and DONE can be monitored at the end of the configuration process to ensure configuration completes successfully. If INIT or DONE is Low at the end of configuration, the device should be reconfigured.

SEU Detection

Upon completion of configuration, the engine can now enter the optional but recommended SEU detection phase. Performing SEU detection allows the configuration memory storage device to be powered off or stay idle until scrubbing or full reconfiguration is needed. It can also reduce the chance of inadvertent corruption of the FPGA if the engine, the memory storage device, or the FPGA itself experiences an upset.

If the readback process is to be skipped, it is strongly recommended to keep FAR test, control register check, and status register check which can detect configuration interface (SMAP), FAR, and global signal SEFI.

FAR Test

To perform a FAR test, the configuration management engine writes a predefined value to the FAR register followed by reading back the FAR value. If the value matches the expected value, the engine proceeds to device readback. If the FAR readback data does not match the expected value, then a SEFI has likely occurred and the configuration engine should re-initiate a full device configuration. Upon a FAR test failure, the designer can chose to repeat the test before declaring an SEFI condition as it is possible, however unlikely, that the FAR readback was corrupted by an SET rather that an actual SEFI.

Table 14: **Recommended Sequence for FAR Test. Using SMAP 8 Interface**

CCLK Cycles	CS/WR	Config Data (HEX)	Explanation/Caution

1	CS=0
	WR=1
			Abort sequence to desync the FPGA. Ensures that if the FPGA is out of the 32-bit word boundary due to an SEU or SET, it can be realigned.

1	CS=0
	WR=0		

8	CS=1
	WR=0		

4	CS=0
	WR=0
		`FF FF FF FF`	Dummy word.

4		`AA 99 55 66`	Sync word.
4		`00 00 00 00`	Dummy word.
4		`30 00 80 01`	Write to CMD.
4		`00 00 00 07`	Reset CRC.
4		`30 00 20 01`	Write FAR.
4		`00 37 C1 00`	Predefined FAR value.
4		`28 00 20 01`	Read FAR.
4		`00 00 00 00`	Dummy word.

1	CS=1
	WR=1
			Switch from write to read mode. As soon as CS deasserts, BUSY toggles High. BUSY must be pulled up with 1 KΩ resister.

X+4	CS=0
	WR=1
			The rising CCLK edge after BUSY returns Low, indicating the first valid data byte is available.

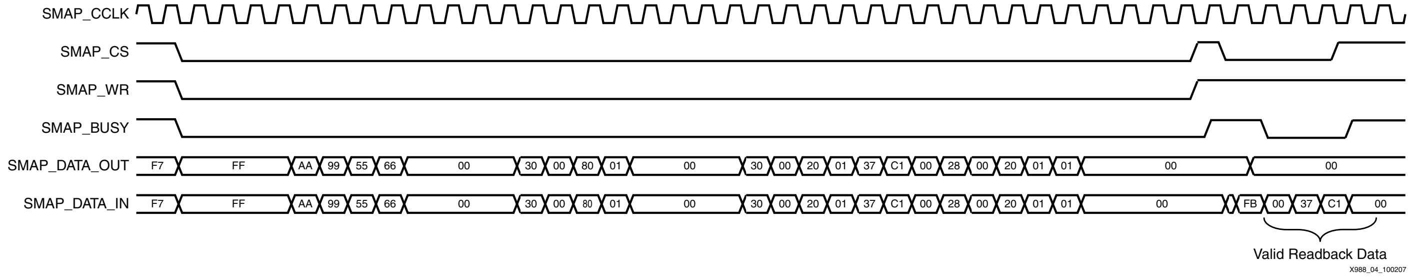
SMAP_CCLK
SMAP_CS
SMAP_WR
SMAP_BUSY
SMAP_DATA_OUT
F7 FF AA 99 55 66 00 30 00 80 01 00 30 00 20 01 37 C1 00 28 00 20 01 01 00 00
SMAP_DATA_IN
F7 FF AA 99 55 66 00 30 00 80 01 00 30 00 20 01 37 C1 00 28 00 20 01 01 00 FB 00 37 C1 00
Valid Readback Data
X988_04_100207

Figure 4: **Sample FAR Check Sequence**

Status and Control Register Check

After successful FAR test, the status (STAT) and control (CTL) registers should be read back and verified. The FPGA should be fully reconfigured if bits 5, 6, or 7 of the status register return '0'. If CTL returns mismatching value, the configuration management engine should immediately start the scrubbing process. If the error persisted, the device needs to be reconfigured.

The default value for STAT is `0x00007EFC` and for CTL is `0x20000109` (when the persist option is set by BitGen). The value of CTL is `0x20000009` if GLUT_MASK_B is enabled. Refer to Table 7 and Table 8 for more information regarding CTL and STATUS register fields.

Table 15: **Recommended Sequence for Status and CTL Register Readback**

CCLK Cycles	CS/WR	Config Data (HEX)	Explanation/Caution

1	CS=0
	WR=1
			Abort sequence to desync the FPGA. Ensures that if the FPGA is out of the 32-bit word boundary due to an SEU or SET, it can be realigned.

1	CS=0
	WR=0		

8	CS=1
	WR=0		

4	CS=0
	WR=0
		`FF FF FF FF`	Dummy word.

4		`AA 99 55 66`	Sync word.
4		`00 00 00 00`	Dummy word.
4		`30 00 80 01`	Write to CMD.
4		`00 00 00 07`	Reset CRC.

4		`28 00 E0 01`
		`or`
		`28 00 A0 01`
			Read STAT and CTL.

4		`00 00 00 00`	Dummy word.

1	CS=1
	WR=1
			Switch from write to read mode. As soon as CS deasserts, BUSY toggles High. BUSY must be pulled up with 1 KΩ resister.

X+4	CS=0
	WR=1
			The rising CCLK edge after BUSY returns Low, indicating the first valid data byte is available.

Device Configuration Memory Readback (Optional)

Although optional, it is recommended to perform a device readback. Scrubbing only when necessary reduces the chance of corrupting the FPGA due to an upset to the controller logic or the FPGA's configuration logic. The worst-case scenario for controller logic or FPGA configuration logic upsets during readback only trigger scrubbing by the configuration management engine.

In Virtex-II devices, readback or configuration scrubbing of SRL16 and LUTRAMs result in conflicts with user operation and corruption of memory contents. With Virtex-4 devices, scrubbing or readback does not interfere with SRL16 and LUTRAM operation if the GLUTMASK_B bit of CTL is set to '0'. Setting this bit causes the configuration logic to skip all SRL16 and LUTRAM configuration memory locations.

By taking advantage of the overhead column, the readback operation can read out the last block RAM interconnect frame without loading the first block RAM content frame into the configuration pipeline registers. This special configuration architecture allows one readback sequence to cover all configuration memory other than block RAM content; thus simplifying the readback process.

It is common practice to calculate a CRC value at the end of the readback process and compare the value against the golden CRC value to determine if upset has occurred. If the values mismatch, scrubbing is used to remove the upset. The Virtex-II CRC verification algorithm recommended in XAPP779, Corr*ecting Single-Event Upsets in Virtex-II Platform FPGA Configuration Memory*, assumes the correction of SEU failed if a failing CRC value of a repeated attempted readback matches a previous failing CRC value. However, this type of failure, also referred to as CRC SEFI, has not been observed in Virtex-4 designs.

Moreover, with the introduction of GLUTMASK_B setting, there exist unused configuration memories that can not be overwritten, but can be upset and readback. However, upsets to these bits are of no consequence. Therefore, the Virtex-4 CRC calculation algorithm is modified such that the golden CRC is updated after two consecutive scrubbing and matching CRC values to account for irrelevant upsets in the unused configuration memory.

Table 16: **Recommended Sequence for Configuration Memory Readback**

CCLK Cycles	CS/WR	Config Data (HEX)	Explanation/Caution

1	CS=0
	WR=1
			Abort sequence to desync the FPGA. Ensures that if the FPGA is out of the 32-bit word boundary due to an SEU or SET, it can be realigned.

1	CS=0
	WR=0		

8	CS=1
	WR=0		

4	CS=0
	WR=0
		FF FF FF FF	Dummy word.

4		AA 99 55 66	Sync word.
4		00 00 00 00	Dummy word.
4		30 00 80 01	CMD write.
4		00 00 00 07	RCRC command.
4		30 00 C0 01	MASK register write.(1)
4		00 00 01 09	MASK register value.(1)
4		30 00 A0 01	Control register write.(1)
4		20 00 00 09	Control register value.(1)
4		30 00 20 01	FAR register write.
4		00 00 00 00	Initial frame address
4		30 00 80 01	CMD write.
4		00 00 00 04	RCFG command.
4		28 00 60 00	FDRO command.
4		48 xx xx xx	Number of words to be read out.(2)
4		00 00 00 00	Dummy word.

1	CS=1
	WR=1
		–	Switch from write to read mode. As soon as CS deasserts, BUSY toggles High. BUSY must be pulled up with 1 KΩ resister.

X(3)	CS=0 WR=1	–	The rising CCLK edge after BUSY returns Low, indicating the first valid data byte is available.

Notes:

1. GLUT_MASK_B bit writes to MASK and CTL are optional. However, if the design uses SRL16/LUTRAM, the GLUT_MASK_B bit must be set before readback.
2. The number of readback words needed to account for dummy frame. For the SelectMap 8 interface, an additional word needs to be added in addition to the dummy frame. The number of words can be derived from the architecture figures provided in the reference design zip file. Table 17 also provides a quick reference.
3. For Selectmap 8 interface, an additional dummy clock should be accounted for in addition to the dummy word and dummy frame.

Table 17: **FDRO Word Count for Virtex-4 Qpro-R**

Device	Real FDRO Word Count for SMAP8(1)	Real FDRO Word Count for all other Interfaces(1)
LX200	1,382,521	1,382,520
SX55	498,889	498,888
FX60	486,425	486,424
FX140	1,115,365	1,115,364

Notes:

1. Real FDRO word count includes all configuration memory bits covered by FAR block address 000 and 001. The word count assumes the readback process covers the entire block address in one read sequence; therefore, it includes overhead frames at the end of each row.

Scrubbing (Active Partial Reconfiguration)

After a non-SEFI SEU event is detected, the FPGA should to be scrubbed to prevent SEU accumulation. After scrubbing completes, readback should immediately follow to ensure SEUs are corrected. Moreover, the scrubbing process is non-intrusive and does not interrupt the user design.

In contrast to earlier Virtex architectures where scrubbing operations can potentially corrupt LUT memory primitives such as SRL16s and LUTRAM, if GLUT_MASK bit is set, Virtex-4 applications can make free use of SRL16 and LUTRAM for better resource utilization and timing performance.

Table 18: **Recommended Scrubbing Commands**

CCLK Cycles	CS/WR	Config Data (HEX)	Explanation/Caution

1	CS=0
	WR=1
			Abort sequence to desync the FPGA. Ensures that if the FPGA is out of the 32-bit word boundary due to an SEU or SET, it can be realigned.

1	CS=0
	WR=0		

8	CS=1
	WR=0		

4	CS=0
	WR=0
		`FF FF FF FF`	Dummy word.

4		`AA 99 55 66`	Sync word.
4		`00 00 00 00`	Dummy word.
4		`30 00 80 01`	CMD write.
4		`00 00 00 07`	RCRC command.
4		`00 00 00 00`	Dummy word.
4		`00 00 00 00`	Dummy word.
4		`30 01 80 01`	Write to IDCODE.
4		`xx xx xx xx`	Device IDCODE.
4		`30 00 C0 01`	MASK register write.[1]
4		`7F FF FF FF`	MASK register value.[1]
4		`30 00 A0 01`	Control register write.[1]
4		`20 00 00 09`	Control register value.[1]
4		`30 00 80 01`	CMD write.
4		`00 00 00 06`	RCAPTURE command
4		`00 00 00 00`	Dummy word.
4		`30 00 20 01`	FAR register write.
4		`00 00 00 00`	Initial frame address.
4		`30 00 80 01`	CMD write.
4		`00 00 00 01`	WCFG command.
4		`30 00 40 00`	FDRI command.
4		`50 xx xx xx`	Number of words to be written.[2]
X		–	Scrub data.[3]

Notes:

1. GLUT_MASK_B bit writes to MASK and CTL are optional. However, if the design uses SRL16/LUTRAM, the GLUT_MASK_B bit must be set before readback.
2. The number of FDRI words is device dependent and can be derived from the architecture figures provided in the reference design zip file. Table 19 also provides a quick reference.

Table 19: **FDRI Word Count for Virtex-4 Qpro-R**

Device	Real FDRI Word Count[1]
LX200	1,382,520
SX55	498,888
FX60	486,424
FX140	1,115,364

Notes:

1. Real FDRI word count includes all configuration memory bits covered by FAR block address `000` and `001`. The word count assumes the readback process covers the entire block address in one read sequence; therefore, it includes overhead frames at the end of each row.

Fault Injection Design Consideration

Fault Injection provides a cheaper alternative to on site cyclotron testing. This testing can cover all known user accessible configuration memory cells and provide an estimate on the design failure cross section along with acceleration radiation test. However, it can not simulate upsets to the configuration logic itself.

Design Consideration for Configuration and Readback

High-Current SEFI

One SEFI exists that results in excessive current draw through V_{CCINT} has been observed during accelerated radiation testing. The cross section for this high-current SEFI is at least an order of magnitude lower than the total SEFI cross section when scrubbing is constantly performed. However, extended current draw on V_{CCINT} can damage the device and affect its reliability. The extent of the current draw is in direct correlation with the application — the more resources used in the device, the higher the current draw upon experiencing this SEFI.

The high-current SEFI is the result of upsets to the internal configuration logic such that scrubbing corrupts configuration rather than correcting upsets. In order to further lower the cross section of this SEFI, it is strongly recommended to perform scrubbing only if readback reports an issue. The worst-case scenario for internal configuration logic upsets during readback only results in the configuration management core issuing a follow-on scrubbing sequence.

To further protect the system, a current limiting device can be used. If the FPGA ever experience this SEFI, the current limiting device can clamp the current draw, lowering V_{CCINT} to below the power-on-reset threshold. Once V_{CCINT} falls below this threshold, the FPGA loses its configuration and the internal contention clears. V_{CCINT} can then be returned to its normal voltage and current supply level while the FPGA reconfigures.

It is also possible to protect the device with frame-based scrubbing to limit configuration data corruption to only one frame in case of upset. However, frame-based scrubbing more than doubles the scrubbing period and requires configuration management design to provide frame addresses for each frame scrub.

Block RAM

An upset condition has been observed in Virtex-4 block RAM such that a large amount of block RAM content data may be corrupted due to SEU. For applications requiring block RAM, contact Xilinx technical support for mitigation information.

Distributed RAMs (LUTRAM) and SRL16

For readback and partial reconfiguration operations, there are special considerations for designs utilizing distributed RAMs and SRL16s. When a LUT is configured as either SRL16 or distributed RAM (LUTRAM), readback or active partial reconfiguration can corrupt the contents of such primitives if GLUTMASK_B bit is left at its default setting of '1'. Therefore, it is important to ensure the GLUTMASK_B bit is set to '0'. Block RAM (BRAM) content; however, is not covered by the GLUTMASK_B setting. Therefore, block RAM content configuration columns still need to be avoided during readback or scrubbing.

When GLUTMASK_B of the CTL register is set to '0', scrubbing skips over the LUTRAM and SRL16 configuration location. In addition, this setting also disables configuration writes to some unused configuration memory cells while readback of such cells are still allowed. If these unused configuration memory cells are used and GLUTMASK_B is set to '0', it is possible for these unused configuration memory cells to be upset to '1', readback as '1', but not corrected. In order to prevent the combination of GLUTMASK_B setting and upset to unused memory cells from continuously triggering a CRC error, it is recommended to implement a CRC-check algorithm different from that of the Virtex-II devices (refer to "Golden CRC Calculation" for more information).

Golden CRC Calculation

It is strongly recommended to run scrubbing only if necessary to avoid high-current SEFIs. Numerous algorithms can be used for detecting configuration upsets, with 16-bit or 32-bit CRC comparisons of configuration data the most commonly used. Using ECC implemented externally can provide additional information on single-bit error locations and an indication of multiple-bit errors. CRC calculation can be performed on at the device level or on a frame-by-frame basis to further enhance integrity.

In contrast to Virtex-II devices, CRC SEFIs have not been observed in Virtex-4 devices; therefore, the golden CRC calculation can be slightly modified to better suit the Virtex-4 architecture where the golden CRC can be updated if two consecutive CRC values match after intermediate scrubbing.

DCI Considerations

To prevent excessive current draw on Virtex-4 designs, additional care should be considered for VRP and VRN pins. It is recommended that all VRP and VRN pins are left floating and the design drives a constant '1' into instantiated DCIRESET primitive.

JTAG

If the SelectMap interface is chosen for configuration management, the JTAG interface still must be properly handled. Since JTAG has the highest priority in accessing the configuration, SelectMap can lose control to the configuration interface if JTAG access is unintentionally triggered by an SEU. To prevent the JTAG interface from unexpectedly accessing configuration interface, the JTAG TAP controller must be placed in Test-Logic-Reset during Selectmap configuration management operation. This setting can be achieved by having a free-running clock on TCK while TMS is pulled High. TDI can be left any state, but preferably pulled High.

Flip-Flops

While CLB and IOB flip-flops have programmable features selected by configuration latches, the flip-flop register storage elements are separate from configuration latches and cannot be accessed through configuration. Therefore, readback and partial reconfiguration do not affect the data stored in these registers. However, if the CAPTURE_VIRTEX4 component is used, then a dedicated configuration latch is updated with the current flip-flop register value when capture is triggered. This action could result in readback mismatch if this data is not masked for readback comparison.

Self-Hosting Configuration Management Core Design Considerations

When the configuration management core is hosted in a Virtex-4 FPGA, additional design care must be taken. The state machine of the core must be protected and periodically reset to avoid transitioning into unknown state due to upset.

Registers holding readback and scrubbing commands should be periodically reset because scrubbing can correct the set and preset value of the registers but not the value held in the register. An alternative method is to store the configuration commands in radiation-hardened configuration data storage devices.

In addition, a heart-beat signal is recommended to indicate the core's status. For more information regarding self hosting configuration management setup, refer to XAPP989, *Correcting Single-Event Upsets with a Self-Hosting Configuration Management Core*.

Additional Configuration Pins Requiring Pull-Ups and Pull-Downs

The following configuration pins require pull-ups and downs for proper operation:

- DONE – 470Ω pull-up
- INIT – 4.7 ΩK pull-up
- BUSY, PROG_B, PWRDWN_B – 1 ΩK pull-up
- CS, RDWR, V_{BATT} – 1 ΩK pull-down

Bitstream Reorganization

The default BitGen settings generate a configuration bitstream that might not be ideal for scrubbing. By rearranging the configuration bitstream and updating configuration command sets, the configuration bitstream can be optimized for scrubbing and readback usage. For more information on how to manipulate the bitstream for scrubbing, please consult Xilinx technical support.

Revision History

The following table shows the revision history for this document:

Date	Version	Description of Revisions
03/13/08	1.0	Initial Xilinx release.

Notice of Disclaimer

Xilinx is disclosing this Application Note to you “AS-IS” with no warranty of any kind. This Application Note is one possible implementation of this feature, application, or standard, and is subject to change without further notice from Xilinx. You are responsible for obtaining any rights you may require in connection with your use or implementation of this Application Note. XILINX MAKES NO REPRESENTATIONS OR WARRANTIES, WHETHER EXPRESS OR IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, IMPLIED WARRANTIES OF MERCHANTABILITY, NONINFRINGEMENT, OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT WILL XILINX BE LIABLE FOR ANY LOSS OF DATA, LOST PROFITS, OR FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, OR INDIRECT DAMAGES ARISING FROM YOUR USE OF THIS APPLICATION NOTE.

XILINX PRODUCTS (INCLUDING HARDWARE, SOFTWARE AND/OR IP CORES) ARE NOT DESIGNED OR INTENDED FOR USE IN ANY APPLICATION REQUIRING SAFETY CRITICAL PERFORMANCE, SUCH AS LIFE-SUPPORT OR SAFETY DEVICES OR SYSTEMS, CLASS III MEDICAL DEVICES, NUCLEAR FACILITIES, APPLICATIONS RELATED TO THE DEPLOYMENT OF AIRBAGS, OR ANY OTHER APPLICATIONS THAT COULD LEAD TO DEATH, PERSONAL INJURY OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE (INDIVIDUALLY AND COLLECTIVELY, "CRITICAL APPLICATIONS"). FURTHERMORE, XILINX PRODUCTS ARE NOT DESIGNED OR INTENDED FOR USE IN ANY APPLICATIONS THAT AFFECT CONTROL OF A VEHICLE OR AIRCRAFT.