



XAPP864 (v1.0.1) March 5, 2009

SEU Strategies for Virtex-5 Devices

Authors: Ken Chapman and Les Jones

Summary

Xilinx® devices are designed to have an inherently low susceptibility to single event upsets (SEUs). This application note provides a substantial discussion of strategies and representative calculations for handling SEUs with an emphasis on reliability when addressing these low probability events.

This application note is accompanied by a reference design for use with the Virtex®-5 FPGA ML505 evaluation platform (but can port to other hardware) and an SEU controller macro, which can be included in any Virtex-5 FPGA design to implement an SEU detection and correction scheme. These supplements can be used to evaluate the different methods of dealing with SEUs.

Due to the infrequent and unpredictable nature of real SEUs, small scale testing of their effects and system verification is impractical. For this reason, the SEU controller macro and reference design can emulate an SEU by deliberately injecting an error into the FPGA configuration so that its subsequent detection and correction can be confirmed. Injection of errors can also be used to assess SEU mitigation circuits implemented in a design.

This application note focuses on the Virtex-5 family although much is applicable to the Extended Spartan®-3A family.

Introduction

SEUs have the potential to affect most digital electronic circuits. Xilinx takes this issue seriously, and by improving the underlying technology, Xilinx devices experience very low levels of SEUs. Xilinx also recognizes that SEUs are unavoidable within commercial and practical constraints, so the company provides built-in SEU detection in the Virtex-5 and Extended Spartan-3A families to simplify and improve the system design.

A Note for Experienced Users

The study of SEU behavior and the impact on Xilinx devices (the Rosetta Experiment) is an ongoing project that is providing an improved understanding of the subject. For this reason, readers familiar with SEU issues, especially with regards to Xilinx FPGAs, are advised to study the whole of this application note and review their choice of strategy when dealing with this phenomena. Xilinx continues to take the subject of SEUs very seriously and therefore the dynamics of this subject are changing. The good news is that the situation has improved, but even positive changes should initiate a review of strategy. If you have previously used techniques to handle SEUs in Virtex-4 devices and are now working with Virtex-5 devices, there are valuable differences due to the improved capabilities of the newer devices.

Applications requiring the highest level of reliability should use an appropriate SEU risk mitigation scheme. This application note discusses the factors that should be considered and presents strategies that can be used with Virtex-5 devices.

The Rosetta Experiment

The Rosetta experiment is an ongoing project at Xilinx that collects real measurements of SEUs and applies the knowledge gained when engineering each new product. The test data for

Virtex-5 and Extended Spartan-3A families confirms that these devices have significantly lower susceptibility to SEUs than their predecessors.

Details of the Rosetta experiment are available in [WP286](#), *Continuing Experiments of Atmospheric Neutron Effects on Deep Submicron Integrated Circuits*. Calculations in this application note are based on values stated in version 1.0, dated March 10, 2008 of the white paper.

Risk Assessment and Specification

As a starting point, a target specification for system reliability should highlight critical sections of a design and provide a guideline value for the required reliability of the FPGA design. This is typically expressed as the failures in time (FIT) rate, which is the number of failures that can be expected in 10^9 hours (approximately 114,155 years) or the mean time between failure (MTBF).

Based on the FIT data in the *Continuing Experiments of Atmospheric Neutron Effects on Deep Submicron Integrated Circuits* white paper, a Virtex-5 device shows a nominal 151 FIT/Mb for the configuration cells with a 95% confidence range of 101–215 FIT/Mb. For example, the XC5VLX50T device on the ML505 development board has approximately 10.84 Mb of configuration cells (see [Table 4](#) for configuration size; $10.84 \text{ Mb} = 11,365,856 / (1,024 \times 1,024)$) and therefore, a nominal susceptibility of 1,637 FIT or an MTBF of 70 years ($114,155 \text{ years} / 1,637 \text{ FIT} = 70 \text{ years MTBF}$). The device configuration FIT has a 95% confidence range of 1,095–2,330 FIT or an MTBF of 49–104 years.

These calculations reveal how infrequently SEUs occur. Even so, some critical applications must consider appropriate precautions and define post-SEU actions.

When more products are deployed, the probability of an SEU affecting any one of them increases proportionately. For example, if the above XC5VLX50T is used in 1,000 products, the nominal FIT across all products is 1,637,000 and represents one SEU every 25 days. This should not be confused with the probability of an individual device being affected. Also, the probability of an individual device incurring a second SEU is determined by the FIT (or MTBF) of the individual device and not the collection. This is an important consideration when assessing suitable strategies for an application. For example, the MTBF for even the largest Virtex-5 device indicates that it would incur two or fewer SEUs within a 20 year lifetime. The selected strategy should be based on this infrequent potential for disruption even if it was the MTBF for the collection of products to be shipped that motivated the consideration of a requirement to choose one.

The Rosetta experiment demonstrates that in the vast majority of cases, an SEU only changes (flips) a single configuration bit. Multi-bit upsets (MBUs) due to a single ionizing particle almost never occur. Also, there is a high probability that this bit will have little or no effect on the design because less than 20%, and typically less than 10%, of the configuration cells have any significance to a design implementation. An SEU affecting device resources that are not used (for example, unused CLBs, I/O, DCMs, block RAMs, etc.) will have no effect. The percentage of device resources used by a particular design is available in the MAP report, and the device FIT or MTBF calculation can be scaled to reflect the proportion of device used.

Every configuration frame consisting of 1,312 bits contains 12 built-in error correction code (ECC) bits. Any change to the ECC bits caused by an SEU has no effect on the active design. Each frame also contains 16 unused bits (bits 656 to 671 as shown in the figure captioned "Configuration Words in the Bitstream and Configuration Bits in a Frame" in [UG191](#), *Virtex-5 FPGA Configuration User Guide*), making a total of 28 bits that have no effect on any design. Hence, 2.13% of the configuration bits can be eliminated from the active design risk calculation in all situations.

A large number of unused configuration cells are also within the area of an otherwise used resource. For example, the programmable interconnect has many possibilities but only a few of those apply to a particular design. Consequently, an SEU that causes the connection of an unused segment of interconnect to another unused segment has no effect on a particular

design. Even a connection of an unused segment to a used segment is unlikely to have any noticeable effect.

Therefore, as a first very conservative approximation for the device configuration, FIT can be divided by five to provide a design configuration FIT, which for the XC5VLX50T device translates into a design configuration MTBF of 245 to 520 years with a 95% confidence level. This number improves when the design occupies less than 100% of the resources.

If an application needs reliability at this level, then other aspects of the system and design not related to SEUs almost certainly require analysis similar to the basic risk assessment already described. For example, intermittent faults have been traced back to the use of asynchronous resets and cross coupling between traces on PCBs when the fault was initially attributed to SEUs. The use of asynchronous resets in a high reliability design is not recommended and priority should be given to eliminating this common design practice before proceeding with SEU topics.

Comparison of Configuration and Data SEUs

If an SEU occurs in a Virtex-5 device, the state of a bit is flipped. That bit can be associated with the configuration of the device or it can be a change to the operational data of the current design. In this situation, data means anything that is a variable within the design, including the contents of RAM and flip-flops. Although the focus of this application note is the impact of SEUs on the integrity of configuration cells, SEU effects on data must be considered because they also influence the selection of precautions used for configuration.

In most cases, a single bit change within informational data can be tolerated and ignored, for example, a single corrupted ASCII character in an e-mail or one incorrect pixel in a video image. More significantly, any transitory data is overwritten by new information, meaning that the effect of the SEU is short lived. However, when the data takes the form of instructions or a state required in the continuing operation of the design, the effect of an error can be significant and prolonged, for example, a state machine might enter an illegal state or the IP address for communication be made incorrect. In such cases, risk assessment should be performed, and suitable precautions should be considered.

As described previously, the FIT or MTBF associated with the configuration cells of a Virtex-5 device can be calculated using the results from the Rosetta experiment. In a similar way, the FIT or MTBF can be calculated for the contents of the block memories of the device. The XC5VLX50T device has sixty 36 Kb block RAMs, having a total memory capacity of 2.11 Mb. Per [WP286](#), *Continuing Experiments of Atmospheric Neutron Effects on Deep Submicron Integrated Circuits*, the nominal FIT/Mb value for block RAM contents is 635, which yields a device block RAM FIT of 1,339 or an MTBF of 85 years.

Although there are over five times more configuration cells than block RAM memory bits, the FIT for the device block RAM (1,339) is similar to the FIT for the device configuration block RAM (1,673). The configuration cells are robust based on their requirement to remain static most of the time while block RAM memory must switch between states quickly for operational reasons. This makes them more susceptible to SEUs.

It is common for designs to use a large proportion of the available block RAM resources, which increases the probability that an SEU will affect the data within a design. In comparison, the typical amount of configuration cell usage is low and the design data FIT will dominate over the design configuration FIT. Therefore, when block RAM is used extensively in an application, the importance of data should be given consideration before that of configuration.

The [“Macro Size and Analysis of Reliability”](#) section provides an example of how to perform a risk analysis calculation for any design. It estimates the configuration and data FIT rates that can be compared or combined to determine the total FIT of the macro.

Data contained in flip-flops are least likely to suffer an SEU. Accelerated tests predict the FIT for flip-flops to be as low as 1 to 2 per megabit. Given this low value and the low number of flip-flops in a device, flip-flops can be normally omitted from risk calculations. For example, the

XC5VLX50T has approximately 0.03 Mb of flip-flops, which means a maximum device flip-flop FIT of 0.06 or an MTBF of nearly 2 million years, even if all flip-flops in the device are used.

However, if any data value is highly significant, including the data held in flip-flops, the design should contain precautionary logic to detect, correct, ignore, or recover from an SEU disturbance in a way that is appropriate to the application. Only the design has the ability to determine when data is corrupted because data is variable with a meaning specific to the application. Since the block RAM content has the potential to dominate the requirement for precautionary logic, the block RAMs of Virtex-5 devices are provided with an ECC option, which can be exploited. However, logic circuits that require absolute data integrity to operate correctly might have to be implemented with a degree of redundancy, such as employing a triple modular redundancy (TMR) technique.

Strategies for Handling Configuration SEUs

Given the previous risk assessment values, it is understandable why the vast majority of applications can ignore the whole subject of SEUs. Xilinx continues to commit time and resources to the Rosetta experiment to provide the information and data needed to assess the risks. This data should be used as a starting point, and claims not backed by similar data should be viewed cautiously.

There are applications (often small parts of larger designs) for which even the smallest risk is unacceptable, and some precautions and actions must be considered. An SEU disturbance to a configuration cell has the potential to change the definition of the design itself, and unlike informational data, is not overwritten by a new value. However, this possibility should not prevent even high reliability systems from benefiting from Virtex-5 FPGA technology. The strategies described in this application note assume that this is the requirement and that even small risks are unacceptable. It will also focus on the effects of SEUs on configuration cells.

The strategies discussed are scheduled maintenance, emergency maintenance, running repairs, and a combination of all. Because an SEU is a highly unlikely event, the designer must carefully consider the disadvantages of adopting a particular strategy as well as its advantages.

Scheduled Maintenance Strategy

Now consider the possible effects resulting from an SEU that flips a configuration cell that directly impacts an active design. The effect on the design can either be almost instantaneous, or irregularities might not be noticed for a significant time. For example, a disturbance to the main system clock distribution can have a marked effect almost immediately but a change to a circuit used to display the hours on a clock display might not become apparent for literally hours.

The point is that some parts of an application are more critical than others, and it is the critical parts to which precautions should be applied. It is useful to assess the time an SEU takes to affect a function and to consider the consequences that a failure can have. An important question to ask is if the system is required to maintain normal operation after the event or is it only required to fail safely? The answer helps to determine the appropriate precautions and actions to take.

An SEU is a soft error, meaning that its effect can be reversed and has no lasting damage. This is significantly different from a hard error such as a broken wire to a connector, which typically requires the replacement or physical repair of some part of a system. For this reason, the FIT or MTBF associated with SEUs should not be confused with that of product life expectancy. Whenever an FPGA is configured (for example, following the application of power or cycle of PROGRAM_B), all configuration cells are defined for the required design regardless of any previous state, and any error caused by an SEU is removed.

Although there are some applications that operate continuously for very long periods of time, very few are expected to operate continuously without interruption for the entire product lifetime. Realistically, most applications experience relatively frequent power cycles or periods of inactivity when maintenance can be performed. The scheduled maintenance strategy is

intended to fully exploit these opportunities. Obviously, a power cycle inherently results in reconfiguration of the device and requires no further thought. But the best use of the scheduled maintenance strategy exploits every available opportunity to reconfigure the device during normal operation and reconfigure whenever it is not playing an active role in the system. No attempt is made to determine if an SEU has occurred; the reconfiguration simply repairs any corruption, if it exists. This is the same concept as performing regular maintenance on an aircraft, where certain parts are replaced at regular intervals even if they appear to be perfectly serviceable.

Even with the confidence that any errors are corrected by the next scheduled device reconfiguration, what happens for the period of time between an SEU disturbance and the next device reconfiguration? This is when precautions must either maintain normal service or fail safe as the application requires. Some degree of redundancy is required in the design, such as the use of TMR techniques, which accommodate a failure in one circuit because the remaining two circuits continue to provide normal functionality.

By including the redundancy that the application requires, a single bit error in the configuration resulting from an SEU should be acceptable. The probability of the device receiving another SEU before the next scheduled maintenance should also be considered. This is easy to determine because the same MTBF figure applies, and the shorter the time to the scheduled maintenance, the smaller the opportunity for a second SEU to happen. The previous calculations showed that the design configuration MTBF for an XC5VLX50T device was at least 245 years, so if the device experiences an SEU, it is not likely that a second SEU will occur in the lifetime of the device, let alone the hours until the next service.

Even if a second SEU were to occur before the next service reconfiguration, it would have to cause a single bit error in a specific place to have an adverse effect. Given that typically less than 10% of configuration bits are directly used by a design, there is less than 1% probability that both SEUs would affect the design. If a design also employed TMR techniques, a second error in the module that has already failed does not matter. Likewise, an error occurring in a different part of the application outside of the remaining modules of the previously affected TMR function is covered by its own TMR circuits.

In summary, scheduled reconfiguration definitely corrects any errors that have occurred, and if a design contains preventative measures to cope with an SEU, this scheme should be acceptable.

Emergency Maintenance Strategy

The concept behind the emergency maintenance strategy is the same as a car's brake warning light, which indicates the brake pads might be worn. The appropriate course of action is to take the car to the garage to have the brakes checked and replaced as soon as possible rather than wait until the next scheduled service.

For an FPGA, this means bringing forward the next reconfiguration of the device when an SEU impacts the configuration cells. An analysis must determine if the warning is worthy of immediate reconfiguration or for how long it can be delayed, but the objective is to reconfigure as soon as practically possible.

The key to the emergency maintenance strategy is detecting if an SEU has occurred in the configuration cells. The Virtex-5 and Extended Spartan-3A devices provide a built-in readback CRC facility to make this possible. A comprehensive description is provided in [UG191](#), *Virtex-5 FPGA Configuration User Guide* and [UG332](#), *Spartan-3 Generation Configuration User Guide*. In simple terms, the built-in circuit continuously scans (reads) the configuration cells of the device and computes a 32-bit CRC value. A 32-bit CRC is able to detect any change in as many as 2^{32} bits, which far exceeds the size of the largest devices. If the CRC value computed by a scan differs from the CRC value computed by the first scan immediately following device configuration, a change to the configuration has occurred and the disturbance is indicated by driving the INIT_B pin Low.

Virtex-5 devices also provide an internal signal that is driven High, which can be observed on the CRCERROR output of the FRAME_ECC_VIRTEX5 primitive.

Reaction to the disturbance varies according to the application. Two factors must be recognized. First, the error signal indicates a change in the configuration of the device and does not (and cannot) indicate a change to informational data of the application. If parts of the design are data critical, these parts still need precautionary circuits of their own. Second, the detection of an SEU by the readback CRC circuit takes some time. Table 1 shows the scan times for each Virtex-5 device.

The time to report an SEU depends on the relative positions of the scan read location and the location of the SEU when it occurs. An SEU positioned later in the scan is reported in less than one device scan while an SEU positioned earlier in the scan requires the next device scan to complete. Hence, the time to detection is up to two device scans, with an average of one device scan. If the maximum detection time can be tolerated before an otherwise instant reconfiguration of the device is initiated, then the emergency maintenance strategy avoids the requirement for any special circuitry. If the detection time is considered significant (for example, 20 ms equates to 3,000,000 clock cycles in a design employing a 150 MHz clock), then some precautionary circuits should be part of the design. Depending on the balance of precautions included in the design and the requirements of the application, the urgency of reconfiguration can be assessed, and reconfiguration can be delayed until a suitable point in design operation.

Because less than 20% of the configuration cells have a direct effect on a typical design, even if the readback CRC has reported a configuration change, then statistically more than four out of every five configuration SEUs will have no effect. Forcing emergency maintenance reconfiguration for every instance could be more of an issue than continuing to operate with precautionary logic, as described in the regular maintenance strategy.

Based on the values for the XC5VLX50T calculated previously, then even the low end of the MTBF range is 49 years for an SEU affecting any of the configuration cells and indicates that emergency maintenance will be a rare event anyway. In comparison, the low-end MTBF for an SEU affecting a configuration cell defining the design is at least 245 years suggesting that emergency reconfiguration can be avoided.

In summary, consider using the emergency maintenance strategy when the design can tolerate a configuration fault for the duration of the readback CRC soft error detection time plus the reaction time needed to initiate the device reconfiguration. If this is the case, the design can avoid all precautionary circuits associated with configuration errors, simplifying the design, and keeping it small enough to fit in a lower density device. This is a significant advantage because a smaller device is statistically less susceptible to SEUs.

Readback CRC Considerations

Figure 1 is a representation of the dedicated readback CRC, which is activated by the POST_CRC constraint. For highest reliability of detection, the internal configuration oscillator more commonly associated with CCLK in master modes should be employed, and the INIT_B pin should be used to observe the configuration status. INIT_B is also driven Low in response to an attempt to program with a corrupted configuration image or in preparation for programming, so it is already important to monitor this signal in any high-reliability system. An external controller requires adequate intelligence to interpret the reason for INIT_B being Low, based on the configuration status of the device.

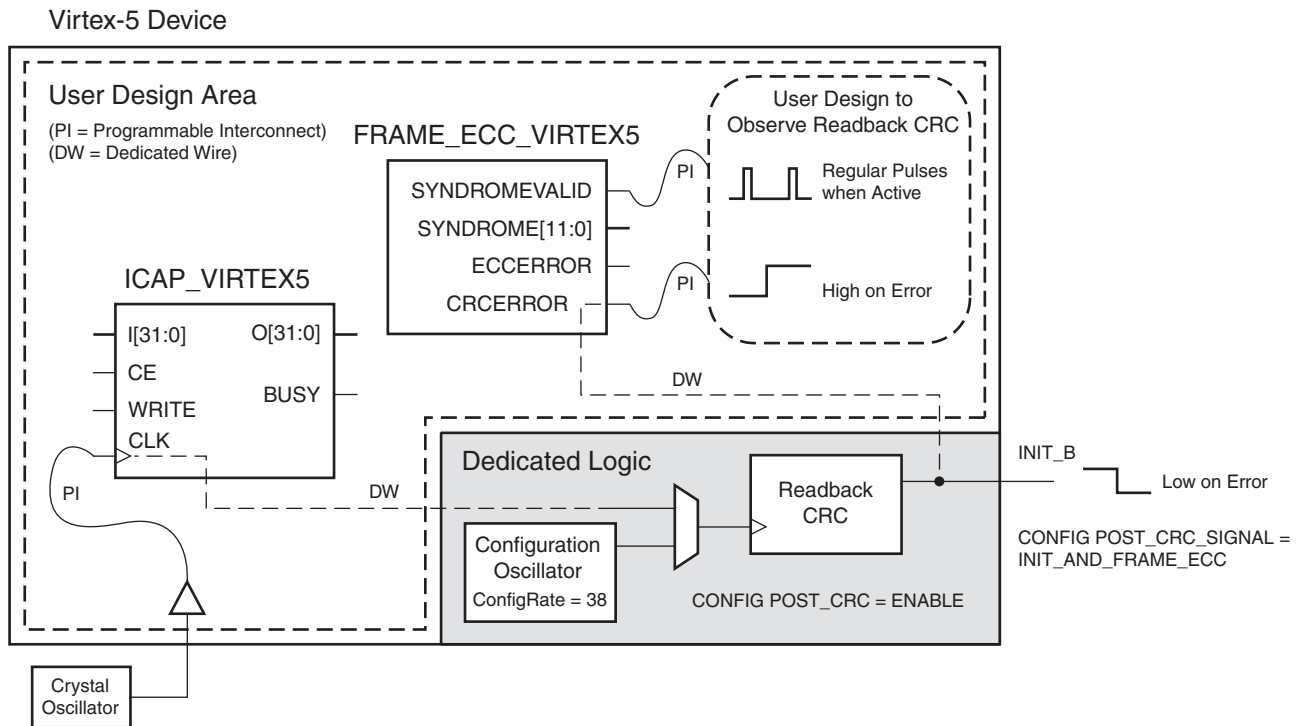


Figure 1: Readback CRC Arrangement in a Virtex-5 FPGA

The optional FRAME_ECC_VIRTEX5 primitive provides internal access to the same (even though inverted) CRC error status as presented externally on the INIT_B pin and can be used to observe a configuration error and take actions appropriate for the application. It is possible to disable the INIT_B pin using a POST_CRC_SIGNAL = FRAME_ECC_ONLY constraint so that only the internal signal is used to report a configuration error. However, it must be appreciated that the programmable interconnect associated with the routing of the internal CRCERROR signal and any user logic that is intended to respond to it is also susceptible to SEUs. Therefore, the INIT_B pin represents the highest reliability reporting point. For the very highest reliability systems, both reporting points can be used with different connections and circuits providing a degree of redundancy.

If an alternative clock is desirable (typically for predictably fast detection times), it can be supplied by a connection to the CLK input of the optional ICAP_VIRTEX5 primitive (or the USRCCKO input of the STARTUP_VIRTEX5 primitive not shown). Reliable operation of the readback CRC logic is now dependant on the integrity of the alternate clock supply and its connection to the ICAP_VIRTEX5 primitive. The clock should be provided as directly as possible (as illustrated in Figure 1) because any additional logic and programmable interconnect in the path is also susceptible to SEUs. If the external clock stops or if it fails to reach the readback CRC circuit, then device scanning stops and it is not able to report any errors.

The FRAME_ECC_VIRTEX5 primitive also provides access to the SYNDROMEVALID signal. While the primary purpose of this signal is in relation to the 12-bit SYNDROME and ECCERROR outputs, they are not used in this scenario. Instead, the SYNDROMEVALID signal can be used to confirm that the readback CRC circuit is actively scanning, and in so doing, confirms the integrity of the clock being used.

Throughout each device scan, the SYNDROMEVALID signal pulses High for one clock cycle at intervals of 41 readback CRC clock cycles, followed by one 49 clock cycle interval at the end of each complete scan. When the internal master configuration clock is employed, the precise timing of the pulses will be unknown, but a simple watchdog circuit can be implemented to determine the presence or absence of the pulses without being concerned with their exact

timing or spacing. The use of the same clock source for the watchdog circuit and the readback CRC is detrimental to system reliability so the use of the internal master configuration clock is also advantageous in this respect. Although a watchdog circuit itself is also susceptible to SEUs, any such occurrence still results in an error report being generated by the readback CRC circuit a short time after the SEU, so this case is naturally covered. If the watchdog circuit fails to detect pulses, action will need to be tailored to the specific application. It might be appropriate to consider a failure of the readback CRC as less of an emergency than an actual readback CRC configuration error. When using an external clock, the failure could be associated with the external oscillator or connections and such a hard error cannot be fixed as easily as reconfiguration repairs a soft error within the FPGA.

Readback CRC Scan Times

The number of clock cycles required for each scan of a particular device is fixed, but the frequency at which the circuit is clocked, and hence the time to complete each scan of the device, is dependant on the user settings and/or the particular device.

The most reliable configuration detection is achieved when the readback CRC circuit is driven by the internal oscillator normally associated with master mode configuration (CCLK). This highest reliability relates to the fact that no external oscillator is required and eliminates the risks posed by physical components and connections as well as avoiding the risks associated with SEU strikes to the programmable interconnect used to route the clock signal as part of the user design. However, the internal oscillator is subject to variations in manufacturing process, operating voltage, and temperature (PVT), and has the potential to deviate up to $\pm 50\%$ from its nominal value, which is defined by the ConfigRate option in BitGen. The nominal ConfigRate values available are 2, 6, 9, 13, 17, 20, 24, 27, 31, 35, 38, 42, 46, 49, 53, 56, and 60 (MHz). The selection of a higher value results in a shorter readback CRC scan time, so it is probably best to specify a value greater than the default value of 2 (MHz). Care should be taken when selecting higher frequencies to ensure that it is compatible with your configuration method as well as ensuring that the readback CRC maximum frequency of operation is not exceeded should the clock be 50% above its nominal value. For this reason, the maximum value of ConfigRate selected should be 38, yielding a frequency in the range 19 to 57 MHz with corresponding range of readback CRC times (Table 1).

Table 1: Readback CRC Clock Cycles and Scan Times

Device	Clock Cycles per Readback CRC Scan ⁽¹⁾	Readback CRC Scan Time at 60 MHz ⁽²⁾ (mS)	Longest Readback CRC Scan Time using ConfigRate = 38 ⁽³⁾ (mS)
XC5VLX30	226,122	3.77	11.90
XC5VLX50	339,200	5.65	17.85
XC5VLX85	573,392	9.56	30.18
XC5VLX110	764,534	12.74	40.24
XC5VLX155	1,062,030	17.70	55.90
XC5VLX220	1,450,382	24.17	76.34
XC5VLX330	2,175,590	36.26	114.50
XC5VLX20T	152,609	2.54	8.03
XC5VLX30T	236,782	3.95	12.46
XC5VLX50T	355,190	5.92	18.69
XC5VLX85T	589,382	9.82	31.02
XC5VLX110T	785,854	13.10	41.36
XC5VLX155T	1,083,350	18.06	57.02

Table 1: Readback CRC Clock Cycles and Scan Times (Cont'd)

Device	Clock Cycles per Readback CRC Scan ⁽¹⁾	Readback CRC Scan Time at 60 MHz ⁽²⁾ (mS)	Longest Readback CRC Scan Time using ConfigRate = 38 ⁽³⁾ (mS)
XC5VLX220T	1,471,702	24.53	77.46
XC5VLX330T	2,207,570	36.79	116.19
XC5VSX35T	300,578	5.01	15.82
XC5VSX50T	450,884	7.51	23.73
XC5VSX95T	805,862	13.43	42.41
XC5VSX240T	1,852,838	30.88	97.52
XC5VFX30T	305,826	5.10	16.10
XC5VFX70T	611,686	10.19	32.19
XC5VFX100T	880,646	14.68	46.35
XC5VFX130T	1,100,816	18.35	57.94
XC5VFX200T	1,570,922	26.18	82.68

Notes:

1. The number of clock cycles does not easily correlate with the configuration image sizes shown in the data sheet. This is due to the highly optimized nature of the readback CRC scan, which avoids variable user data (for example block RAM contents).
2. Maximum external clock rate for readback CRC circuit operation is 60 MHz.
3. ConfigRate = 38 is the highest value guaranteed to remain below 60 MHz. It can be as low as 19 MHz yielding the scan times shown.

Alternatively, a user clock can be supplied to the readback CRC circuit, which allows the scan time to be predictable and typically faster. However, note that the reliability of the detection is slightly degraded by the dependence on the external clock, the clock's connections to the device, and the risk that SEUs could impact the programmable interconnect routing the clock from the pin to either the ICAP or STARTUP primitives. It is therefore sensible to ensure that the clock source is reliable and that all connections (especially programmable) are kept as simple and as short as practically possible.

As a further precaution and general indication that readback CRC is operational, it is possible to monitor the SYNDROMEVALID output of the FRAME_ECC_VIRTEX5 primitive. This signal pulses High once every 41 clock cycles (49 cycles at the end of each complete device scan) and can therefore be used as a heartbeat indicating all is well. If the clock is lost for any reason (including a hard failure loss of the external clock), then the heartbeat will cease, and the system must take appropriate action.

Running Repairs Strategy

The running repairs strategy is useful in applications where it is desirable to maintain operation following an SEU while carrying out a rapid localized repair of single bit errors. Given the high probability (>80%) that an SEU has no effect on a particular application, this strategy avoids the interruption to service associated with the emergency reconfiguration strategy, which invokes a full device reconfiguration. However, when an SEU does impact the application, the repair is very important, and the limitations of the running repairs strategy must be considered.

As with the emergency maintenance strategy, the built-in readback CRC feature of a Virtex-5 device is instrumental in first detecting changes to the configuration cells. Using this dedicated circuit ensures that the detection is reliable and rapid. Although a configuration upset is detected and reported, only the behavior of the design and system can deduce if the SEU has any effect on the operation. As described earlier, an SEU has no effect in the majority of

cases. However, when an SEU does impact the application, the potential exists for undesirable application behavior from the time the SEU occurs until it is detected and corrected.

A logical error caused by an SEU can also result in prolonged corruption of informational data but this might be tolerable if the repair is made quickly and has no lasting effects. For example, if a configuration bit in the path of a streaming video stream is corrupted, it might cause a disturbance to the displayed images for a few frames until the repair is complete and new video data overwrites the older corrupted images. However, a logical error which adversely affects the control paths or operating states of a design can have lasting effects and precautionary logic might be required in this situation. One potential solution is to issue a reset to all critical circuits upon detection of an error and then remove the reset after the error has been corrected. Although this interrupts normal operation, it will be of significantly shorter duration than the time required for a full reconfiguration. This is especially true when the higher density devices are used.

Having detected an error, it must be located. The exact location can be determined using the configuration Error Correcting Codes and calculator circuit which is included in every Virtex-5 device. As described in [UG191](#), *Virtex-5 FPGA Configuration User Guide*, there is a 12-bit ECC value embedded in each frame of configuration consisting of 41 words of 32 bits (1,312 bits). As each frame is read over 41 clock cycles, the built-in ECC logic calculates the 12-bit ECC value for the current content which combines with the embedded 12-bit value to expose any error (even if the error is within the 12 ECC bits). It is then possible to use this 12-bit value, which is presented by the FRAME_ECC_VIRTEX5 primitive to identify the location of a single bit error in the frame which has been read.

In principle, the correction is straightforward. The corrupted frame of 1,312 bits is read into a buffer, the single bit error is isolated by interpreting the 12-bit ECC value, and then it is corrected. The correction simply inverts the bit to reverse the inversion caused by the SEU. Finally, the corrected frame is written back into the configuration cells. In practice, this task is quite complex, so this application note provides an SEU controller macro to perform all the necessary steps (see [“The SEU Controller Macro”](#) for details). A reference design is also provided, which contains the macro and can be used to evaluate the macro's capability and features.

Once an error has been corrected, the readback CRC circuit restarts and confirms that the error is repaired after one complete scan of the device. This means that although the confirmation of the correction takes additional time, the error is corrected sooner. Although the ECC bits enable any single bit error to be corrected, it is not possible to correct multiple bit errors within the same frame (1,312 bits). In the unlikely event two errors occur in the same frame, the error condition will persist and the system must determine a suitable course of action.

Multiple errors can also be repaired providing they are distributed (a single bit error in each frame), but this situation is extremely unlikely except when forcing multiple configuration errors deliberately during tests.

As mentioned before, even multiple bit errors that cannot be repaired will probably have no real effect on the operation of the active design, but extremely high reliability systems must plan for this possibility. If an application must maintain operation even in this extreme case, TMR techniques are probably used extensively throughout the design. In such cases, the level of redundancy almost certainly means that the design can continue operating until the next scheduled reconfiguration of the device.

Ultimately, the SEU controller macro should only be included in a design when the benefits and limitations of the running repairs strategy are understood. If the design contains adequate redundancy, then a regular maintenance strategy or delayed emergency maintenance strategy should be adequate and addition of the SEU controller might add unnecessary complexity.

Combined Strategy

Systems requiring the very highest reliability benefit from using a combination of elements from the scheduled maintenance, emergency maintenance, and running repairs strategies. This multi-layered approach continues the redundancy concept by using each scheme to provide cover for the next.

To provide a good foundation, the design of the system has to be superior in all respects. The susceptibility of Xilinx products to SEUs is low and published values define the level of risk. These contributions should be exploited.

Preventing failures is preferable to having to cope with them during service. Every aspect of the design implementation contributes to the level of reliability (or unreliability), so poor design practices (such as the use of asynchronous reset in HDL code) must be avoided.

When considering the most beneficial combinations for an FPGA-based system, the operation of an aircraft is a useful analogy when reliability is paramount. Once in service, regular maintenance of the aircraft includes the replacement of parts even if there is no visible evidence of a problem. This prevents failures due to wear over time as well as fixing defects missed during in situ inspections. Reconfiguring an FPGA at regular intervals offers similar advantages by correcting possible unseen errors and ensuring a clean start for each period of operation.

Before each flight, checks are made to ensure all critical systems on the aircraft are working properly. If the check reveals a fault, the aircraft is grounded until the fault is corrected. Using the built-in readback CRC circuit of the Virtex-5 device to continuously check the FPGA configuration performs a similar function. If a system is in a standby mode and an error is detected, an emergency reconfiguration can be invoked to repair the fault. This avoids the potential of accumulating errors over time and ensures the device is in perfect condition when it enters an operational mode.

Once an aircraft is in flight, any failure is undesirable. An active warning light alerts the crew to the nature of the fault so that they can take appropriate actions. These actions vary depending on the severity and location of the failure. It might be possible to contain the problem and continue to the planned destination, or a diversion and an emergency landing might be required. The most challenging scenario is when the plane must maintain flight because there is no suitable landing site, for example, in the middle of the Atlantic. In all cases, once the plane has landed, the aircraft is removed from service and repaired before being flown again. This applies even if the fault was considered to be temporary and resolved by the crew during the flight.

In a Virtex-5 device, the warning is provided when the built-in readback CRC circuit detects an SEU. In most cases, the SEU does not affect the operational design. However, the cases that do impact operation require that appropriate action is taken, depending on the severity of the impact. If the device can be safely reconfigured immediately, this is the most obvious response to the warning. If continuous operation must be maintained, then redundancy and localized repairs should be considered. These techniques can be used separately or in combination depending on the design's specific requirements.

When redundancy is included in the design, a decision needs to be made as to how much is suitable to cover the potential risk and maintain operation until the system can be repaired. The Virtex-5 FPGA built-in ECC logic exploited by the SEU controller macro provided with this application note enables localized repairs to be made during operation. The worse case time to detect and repair an error can be estimated using the table of readback CRC scan times (Table 1) because the scan time for detection is the dominant factor. In the worst case scenario, detection would take two scans. In comparison, the error would appear to be corrected almost immediately. Although corrected, final confirmation will require one more scan of the device. It is important to remember that following correction, the effects the error had on operational states and data might prolong and a localized reset to circuits might be appropriate. Regardless of all the measures employed to maintain operation, it would still then be wise to carry out a full

device reconfiguration at the earliest convenient opportunity as this ensures that all data and states as well as the device's configuration are known to be good.

The SEU Controller Macro

The SEU controller macro facilitates the “Running Repairs Strategy.” The macro also facilitates emulating SEUs within the Virtex-5 device by injecting errors in a controlled and predictable way into the configuration memory. It also provides a means to evaluate and test the readback CRC circuit and the error correction capabilities of the macro which is impossible with real SEUs.

The macro is provided as a VHDL file named `SEU_cntlr.vhd` and a Verilog file named `SEU_cntlr.v`. The separate reference design provides a location for the macro and enables the macro's features for evaluation. Supplemental documentation describing the reference design is included in the file download package available at <https://secure.xilinx.com/webreg/clickthrough.do?cid=115162>

The macro has input and output ports shown in Figure 2 and the VHDL and Verilog component instantiation templates are shown in Figure 3 and Figure 4 respectively. The signals are described in detail in Table 2, which should be reviewed in conjunction with the description of each mode.

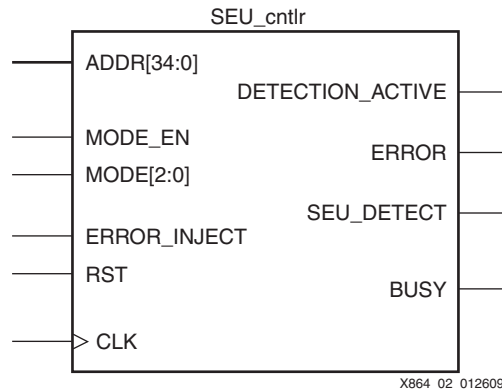


Figure 2: The SEU Controller Macro

Inside the macro, the `ICAP_VIRTEX5` and `FRAME_ECC_VIRTEX5` primitives are used in the same way as described previously to clock and observe the readback CRC circuit, which performs the SEU detection. The macro also includes a controller that connects to the other ports of these primitives to perform the operations necessary to locate and correct SEU errors using the built-in ECC facility. For test purposes, the connection to ICAP is used to facilitate the controlled injection of configuration errors.

Note that because the macro and readback CRC are linked with the configuration of the device, the use of alternative configuration ports must be avoided. For example, the use of JTAG to access configuration will interrupt operation and changes made to the configuration will subsequently be interpreted as errors. Likewise, the `PERSIST` option must be set to `NO`, otherwise the ICAP primitive will be disabled.

Component Declaration

```

component seu_cntlr
  port (
    mode : in std_logic_vector(2 downto 0);
    mode_en : in std_logic;
    busy : out std_logic;
    detection_active : out std_logic;
    seu_detect : out std_logic;
    error : out std_logic;
    error_inject : in std_logic;
    addr : in std_logic_vector(34 downto 0);
    rst : in std_logic;
    clk : in std_logic );
end component;

```

Component Instantiation

```

<instance_name>: seu_cntlr
  port map (
    mode => mode,
    mode_en => mode_en,
    busy => busy,
    detection_active => detection_active,
    seu_detect => seu_detect,
    error => error,
    error_inject => error_inject,
    addr => addr,
    rst => rst,
    clk => clk );

```

X864_03_022609

Figure 3: SEU Controller Macro VHDL Instantiation

Component Instantiation

```

seu_cntlr <instance_name>(
    .mode (mode),
    .mode_en (mode_en),
    .busy (busy),
    .detection_active (detection_active),
    .seu_detect (seu_detect),
    .error (error),
    .error_inject (error_inject),
    .addr (addr),
    .rst (rst),
    .clk (clk));

```

X864_04_030409

Figure 4: SEU Controller Macro Verilog Instantiation

[Table 2](#) describes the input and outputs signals of the SEU controller macro. These signals are identical in both VHDL and Verilog formats.

Table 2: SEU Controller Input and Output Signals

Signal	I/O	Description
MODE[2:0]	I	<p>This 3-bit value specifies the operational mode of the macro as defined in Table 3. The value provided on the MODE[2:0] is only read by the macro on the rising edge of CLK when the MODE_EN signal is active-High.</p> <p>Note: Following device configuration, or a successful macro reset, the operational mode of the macro will be 000.</p>
MODE_EN	I	<p>Apply an active-High signal to this input to instruct the macro to read the value provided on the MODE[2:0] on the next rising edge of CLK. The macro will take up to 40 cycles of CLK after application of MODE_EN for the new mode to be adopted, so this number of clock cycles must be allowed before application of ERROR_INJECT pulses in appropriate modes.</p> <p>Note: If the macro is busy performing an operation (BUSY=1) then the new mode will not be adopted until the current task completes.</p>
RST	I	<p>Active-High input to instruct the macro to reset on the rising edge of CLK. The reset clears the operational mode to 000 and initializes the error injection pointer to zero.</p> <p>Note 1: Macro reset is not possible when the BUSY signal is active. This is to enable all operations in which the device configuration is being accessed to complete and ensure the integrity of the device configuration and the configuration state machine.</p> <p>Note 2: Due to the higher priority of JTAG port, any use of JTAG to conduct configuration related operations results in the macro being unable to interact with the configuration memory. If the macro detects this exclusion, it will enforce an internal reset of the macro as if a RST had been applied. In practice, the macro is intended for use in deployed products when the use of JTAG is not used during periods of normal operation.</p>
CLK	I	<p>Input clock used by all elements within the macro and used to clock the built-in readback CRC logic of the Virtex-5 device. As such, this should be a reliable free running clock of 60 MHz or less (typically 50 MHz) and distributed via a clock buffer and low skew network within the device (normally inserted automatically by synthesis tools). To maximize system reliability, the clock should be provided as directly as possible to the macro avoiding additional logic (for example, DCMs and PLLs) and interconnect whenever possible. It is also good practice to monitor the DETECTION_ACTIVE output as this will also confirm that a clock is correctly applied.</p>
SEU_DETECT	O	<p>An active-High level on this output signifies that a configuration error has been detected by the readback CRC circuit. This signal has a direct connection to the CRCERROR output of the FRAME_ECC_VIRTEX5 primitive used within the macro and therefore it internally reflects the status (even though inverted) also being indicated on the external INIT_B pin unless that pin has been disabled (not recommended). The SEU_DETECT output will revert to the normally Low state after a successful correction of the error by the macro when operating in mode 1 and following one complete readback CRC scan confirming the correction has been successful.</p>

Table 2: SEU Controller Input and Output Signals (Cont'd)

Signal	I/O	Description
DETECTION_ACTIVE	O	<p>This output signal has a direct connection to the SYNDROMEVALID output of the FRAME_ECC_VIRTEX5 primitive used within the macro and can be used to confirm that readback CRC is actively scanning the device configuration in order to detect SEUs. During normal readback CRC scanning of the device, the DETECTION_ACTIVE output will repeatedly pulse High at regular intervals of 41 cycles of CLK with the exception of one interval of 49 CLK clock cycles as each device scan is completed.</p> <p>Note 1: Pulses will become irregular when the macro is in the act of correcting or injecting errors.</p> <p>Note 2: A prolonged absence of pulses probably indicates a failure of CLK (source or connectivity). Therefore, a watchdog monitor circuit using a different clock should be used to monitor this signal.</p>
ERROR_INJECT	I	<p>This active-High input should only be used in conjunction with modes 4, 5, 6, and 7 (MODE[2:0] = 100, 101, 110, or 111) to initiate the action associated with the particular mode. Each cycle of the ERROR_INJECT signal requires a hand-shaking interaction with the BUSY output using the following sequence:</p> <ul style="list-style-type: none"> • Confirm BUSY is Low or wait for it to become Low. • Assert (drive High) ERROR_INJECT. • Wait for BUSY to become High. • Deassert (drive Low) ERROR_INJECT. <p>Other than performing the above handshaking sequence synchronously to CLK there are no specific timing requirements.</p> <p>Note 1: Do not apply ERROR_INJECT when in modes 0, 1, 2, or 3 as doing so can lead to unexpected behavior.</p> <p>Note 2: Allow at least 40 cycles of CLK after selecting a new mode before application of ERROR_INJECT pulses.</p>
BUSY	O	<p>Active-High output indicating the macro is actively engaged in performing a task such as error correction or error injection. The main purpose of this signal is in the hand-shake sequencing of the ERROR_INJECT input and to determine when an operation has been completed so that the macro is ready to continue with the same or a different task.</p> <p>Note: SEU detection is performed by the dedicated Virtex-5 FPGA built-in readback CRC and therefore, under normal operating conditions, the macro is simply waiting for an error to be reported. Hence BUSY is Low under normal conditions.</p>

Table 2: SEU Controller Input and Output Signals (Cont'd)

Signal	I/O	Description
ADDR[34:0]	I	<p>This 35-bit input is only used in conjunction with mode 4 (MODE[2:0] = 100) and defines the location at which an error will be injected. The ADDR input should be stable before and during the application of the ERROR_INJECT pulse in this mode. The time required for the macro to inject an error is proportional to the value of ADDR specified and BUSY will be active during this operation.</p> <p>Note 1: The largest value of 0x7FFFFFFFFF hex will take approximately 4 minutes to complete when using a CLK = 50 MHz</p> <p>Note 2: The largest value that should ever be required is 0x00433C500 hex which equates to the size of the largest XC5VLX330T device and will take approximately 0.5 seconds with CLK = 50 MHz.</p> <p>Note 3: Table 4 shows the maximum ADDR values for each device.</p> <p>Note 4: If ADDR exceeds the size of the device, the excess address value wraps (aliases) to the start of the device.</p>

Table 2: SEU Controller Input and Output Signals (Cont'd)

Signal	I/O	Description
ERROR	0	<p>The macro will drive this output High to indicate when an operation has been unsuccessful. The significance of the error status output depends on the mode under which the macro is operating when it occurs.</p> <p>The greatest significance is when the macro is operating in the automatic correction mode 1 (MODE[2:0] = 001) in a deployed product. In this situation, the macro is expected to wait for the detection of an SEU by the readback CRC circuit and then repair the configuration error if one is detected. Therefore, the macro ERROR signal would be expected to remain Low at all times. If ERROR were to become High while in the automatic correction mode, it would signify that the macro has been unable to repair the detected configuration error. This should only occur in the unlikely event that multiple bits within a single configuration frame consisting of 1,312 bits have been corrupted (ECC can only be used to correct single bit errors). The only way this situation is likely to occur is through the deliberate injection of multiple bit errors using the error injection modes.</p> <p>Modes 4 and 7 (MODE[2:0] = 100 or 111) are used to inject configuration errors by inverting the bit at the location specified by ADDR or the index pointer. In these modes the ERROR_INJECT input is used to initiate the injection of an error and the BUSY signal confirms when the operation is taking place. If the macro is unable to inject an error at the current index then the ERROR signal will be High when the BUSY signal returns Low. In this case the ERROR signal will remain High until modes 0 or 1 are selected or a subsequent successful operation is completed in mode 4, 5, 6, or 7.</p> <p>There are two possible reasons why error injection can fail to happen:</p> <ol style="list-style-type: none"> 1. Not all bits in the configuration memory map can be changed. For example, when the design uses a LUT6 for distributed memory, the write controls associated with that LUT6 are passed from the configuration interface to the design. So even though the bits of the LUT6 have a position within the configuration memory map, they have become non-writable from the configuration interface (ICAP_VIRTEX5 primitive). All frames contain 16 unused bits which can not be changed. 2. Successful injection of an error inverts (toggles) the bit located at the selected location. However, If a second error injection is performed at the same location (without previously correcting it using mode 1) then the second otherwise successful injection of an error will actually cancel the original error. Providing there are no other errors in the same frame the ECC logic will indicate that the frame is now good and the macro interprets this to mean that error injection was unsuccessful. This is good experiment to perform with the reference design as the first error injection does result in a configuration error which is then detected and reported by the readback CRC logic (SEU_DETECT = 1). A second error injection at the same location will correct the error, ERROR will be High but SEU_DETECT will return Low.

Summary of Macro Modes

Macro modes are summarized in [Table 3](#).

Table 3: Summary of SEU Controller Modes

Mode	MODE[2:0]	Description
0	000	Detection only (default at power-on or following reset)
1	001	Detection and automatic correction.
2	010	Reserved (Do not use)
3	011	Reserved (Do not use)
4	100	Inject configuration error at location = ADDR
5	101	Increment the error injection index (index = index + 1)
6	110	Reset the error injection index (index = 0)
7	111	Inject configuration error at location = index

Notes:

1. The MODE[2:0] are confirmed by the MODE_EN input.
2. Modes 4, 5, 6, and 7 require ERROR_INJECT to be asserted to initiate the operation.
3. Detection is performed continuously in all modes unless the macro is actively performing an operation, which requires access to the configuration memory.

Description of Macro Modes

Mode 0 – Detection Only

This is the macro default mode following device configuration or the application of a RST. In this mode, the built-in readback CRC circuit continuously scans the device configuration cells to detect any changes that might occur. The macro initially ensures that the RBCRC_EN bit within the Configuration Options Register (COR1) is set. This means that readback CRC is activated even if the POST_CRC = ENABLE constraint is omitted or disabled during bitstream generation. If an SEU (or any configuration error) is detected, the SEU_DETECT signal is driven High and the INIT_B pin on the device will be driven Low (unless this is specifically disabled which is not recommended). No other actions will be taken if the macro remains in this mode.

When only detection is required, the macro is not required. The POST_CRC = ENABLE constraint is all that is required to enable the readback CRC circuit in its highest reliability form. The optional ICAP_VIRTEX5 and FRAME_ECC_VIRTEX5 primitives can then be included in your design, if required. Although only detection might be required in deployed systems, the ability of the macro's error injection modes to emulate SEUs on demand is a useful tool during system development and testing. Therefore, the macro can be used during development and removed later. Care should be taken when removing the macro to ensure that the POST_CRC = ENABLE constraint is specified correctly and that the built-in readback CRC circuit is still operating (SYNDROMEVALID signal pulsing on the FRAME_ECC_VIRTEX5 primitive is the best way to confirm this).

Mode 1 – Detection and Automatic Correction

This mode is the principle reason for the SEU controller macro and the macro should generally be placed in this mode within a deployed system. The macro initially ensures that the RBCRC_EN bit within the Configuration Options Register (COR1) is set so that readback CRC will be activated even if the POST_CRC = ENABLE constraint is omitted or disabled during bitstream generation. The built-in readback CRC then continuously scans the device, checking the configuration image. If an SEU is detected, the SEU_DETECT signal is driven High and the INIT_B pin on the device is driven Low (unless this is specifically disabled, which is not recommended).

On detection of a configuration error, the SEU controller macro interrupts the readback CRC scan and accesses the configuration memory to identify the location of the erroneous bit using the built-in error correcting code and logic. The identified bit is corrected (inverted) and written

back into the configuration memory. The readback CRC logic then resumes device scanning with completion of the first scan clearing the INIT_B pin and SEU_DETECT signal to confirm a successful repair.

It is advisable to use the SEU_DETECT signal (and/or INIT_B pin) to apply appropriate resets to critical parts of the design or system. These should be used to prevent the propagation of any data corruption caused by the configuration error between the time of its creation by the SEU and its correction by the macro. It is also recommended that the detection of any SEU is recorded and used to invoke a full reconfiguration of the device at the earliest convenient time to guarantee all states within the device.

In the unlikely event that SEUs result in multiple bit errors within a frame (1,312 bits), correction is not possible and the ERROR output of the macro becomes active while in this mode. In this case, full reconfiguration of the device must be invoked to correct the errors and your system will need to determine when this can be performed.

Figure 5 shows the minimal connections required to use the macro in detection and automatic correction mode. MODE_EN is tied High to ensure that the detection and automatic correction mode (001 applied to MODE[2:0]) is used after configuration. It also ensures a return to this mode should another access of the configuration (for example, JTAG) temporarily interrupt SEU detection and force an internal reset of the SEU macro. The 35-bit ADDR and ERROR_INJECT inputs are tied Low because they are unused in this mode, and RST is tied Low for continuous operation. SEU_DETECT can be used to control circuits that might have been affected during the SEU incident. ERROR, and to a lesser extent SEU_DETECT, should be used to invoke a full reconfiguration of the device at the earliest convenient time.

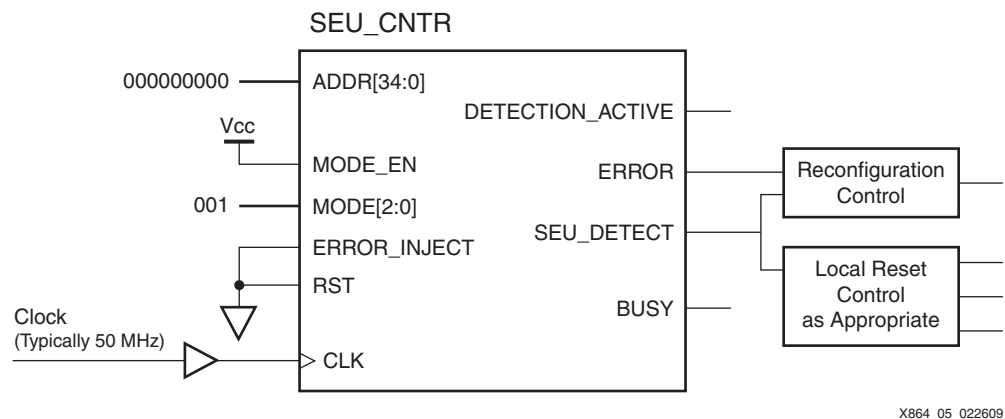


Figure 5: **Minimum Recommended Connections for SEU Controller when Used for SEU Detection and Automatic Correction**

Mode 2 and Mode 3 – Reserved

These modes are reserved for future expansion and should not be used. Simple selection of these modes will result in SEU detection only as described in mode 0 but any subsequent changes to the ADDR input or application of ERROR_INJECT stimulus might lead to unpredictable behavior.

The Error Injection Modes (Modes 4, 5, 6, and 7)

Real SEUs are so rare and unpredictable that evaluating their effects on a design within a Virtex-5 device is extremely challenging. [WP286](#), *Continuing Experiments of Atmospheric Neutron Effects on Deep Submicron Integrated Circuits*, also discusses the LANSCE experiments in which devices are exposed to a controlled source of radiation resulting in increased numbers of SEU. However, the SEUs are still highly unpredictable and have somewhat different characteristics from that of natural radiation. The error injection modes of the SEU controller macro facilitate the emulation of SEUs by directly accessing the configuration via the ICAP_VIRTEX5 primitive within the macro. When desired, a frame of

configuration memory is read, one bit of that frame is inverted to emulate the bit flip caused by an SEU, and then the frame is written back into the active configuration memory. This enables SEUs to be emulated on demand and with predictability and repeatability.

For convenience, the macro presents two schemes for defining the location at which an error is to be injected. Both schemes achieve the same overall effect, but one scheme might be more appropriate for a given experiment. It is also possible to use both schemes sequentially within the same experiment because their operational parameters are independent.

It is important to recognize that when selecting modes 4, 5, 6, and 7, the readback CRC circuit continues operating except for a short interruption when the macro accesses the configuration memory in order to inject an error. This interruption only occurs following the application of an ERROR_INJECT pulse. Therefore, within milliseconds of injecting an error, the readback CRC circuit completes a scan of the device and detects the error, and the SEU_DETECT signal (and INIT_B pin) becomes active.

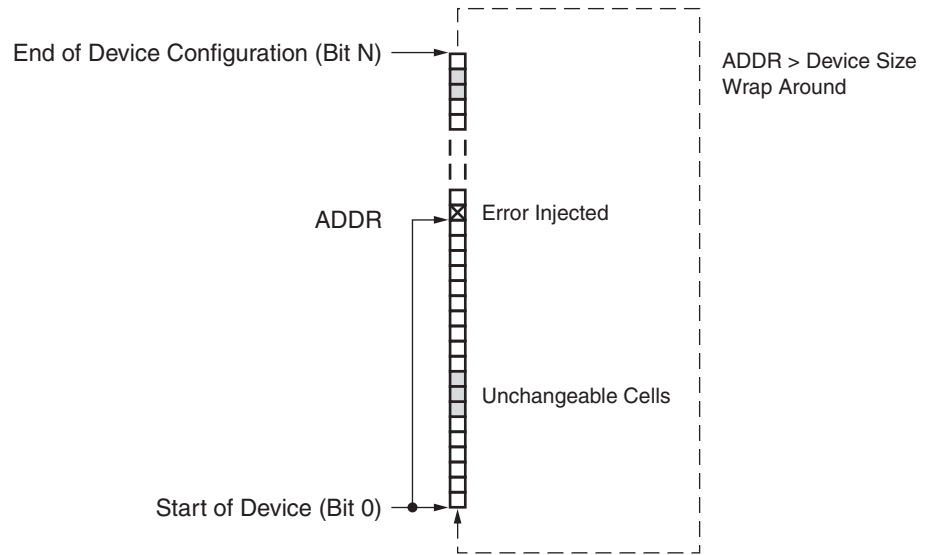
After the required error has (or errors have) been injected, the effect, if any, on the active design can be observed, and the detection of the error by the readback CRC detection can be confirmed. The ability of the macro to perform correction is tested by selecting mode 1. It is possible to inject multiple errors that cannot be corrected automatically but this can also provide useful experimental results. Initiating a full device reconfiguration is the only way to recover from this situation and can help to test the way your system responds to the ERROR output, which would be asserted.

It is highly recommended that regular reconfiguration of the device under test is performed during experiments. This helps prevent misleading results caused by the accumulation of errors and/or the effect those errors have had on variable data within the active design. It is a completely valid test to emulate the accumulation of errors with or without correction over a period of time, but this does require extremely good test management to extract meaningful and constructive results.

Mode 4 – Inject Configuration Error at Location = ADDR

This mode independently provides one scheme for the injection of errors. In one macro operation, it allows an error to be injected at any writable location of the configuration memory by specifying that location using the 35-bit ADDR input. Be aware when using this mode that this single operation can take up to about 0.5 seconds in the largest XC5VLX330T device. Such relatively long times from the initiation of the operation to the instant the error is actually injected must be taken into account when observing the effect on the system.

To specify an appropriate value for ADDR, it is necessary to consider the configuration memory of the device as being organized as N locations of 1-bit where N is the total number of configuration cells in the device. This is fundamentally the same bits observed by the readback CRC and excludes the contents of block memory (block RAM) in the same way. The practical range of ADDR values is from zero up to the total number of configuration bits in the device, provided in [Table 4](#). Larger values of ADDR can be used to specify the error injection location, but the operation will take longer to complete (several minutes for very large values) and the final location of the injected error will alias into the addressable range, that is, ADDR = ADDR – *device size* until ADDR is within the range of the device.

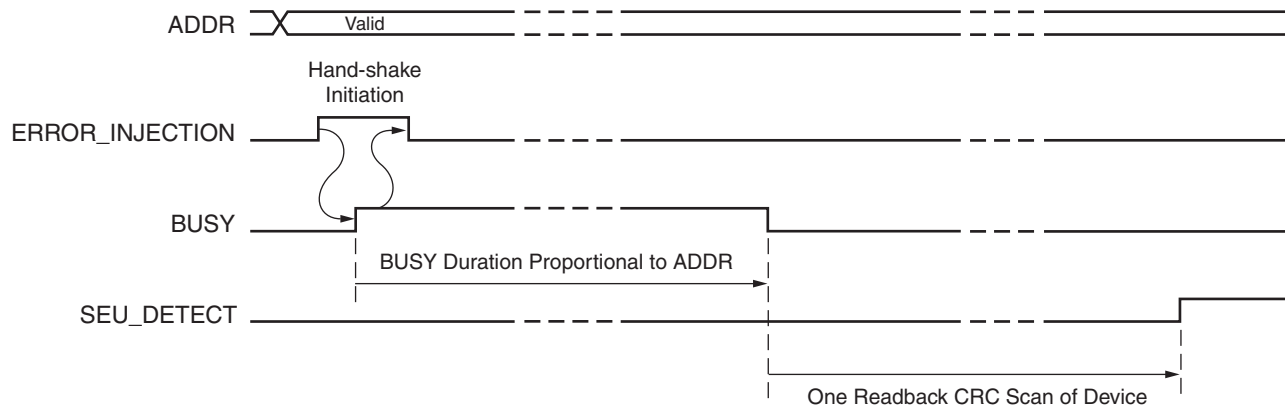


X864_06_022609

Figure 6: Mode 4 Configuration Error Injection Location Defined by ADDR.

To inject an error, select mode 4 by setting MODE = 100 and apply a pulse to the MODE_EN control. The readback CRC continues to perform detection only. If not prepared previously, apply the desired 35-bit value to ADDR input and ensure that it remains stable throughout the subsequent operation. When the error is to be injected, drive ERROR_INJECT High to start the operation. ERROR_INJECT should remain High until the BUSY output of the macro goes High confirming the start of the operation after which ERROR_INJECT can be deasserted. BUSY remains High for the duration of the error injection operation during which time the readback CRC scanning is also disabled. Some less consistent pulses will be present on the DETECTION_ACTIVE output as the macro accesses the configuration memory.

The time to complete the operation and for BUSY to return Low depends on the magnitude of the ADDR value. However, when BUSY does go Low, the readback CRC scanning resumes and the injected error is detected on completion of the first scan of the device.



X864_07_022609

Figure 7: Waveforms (Not to Scale) of a Successful Error Injection (ERROR = 0) Using Mode 4

Some values of ADDR will correspond with locations that are unable to be changed. In these cases, no error will be injected and the ERROR output is High when the BUSY signal returns Low to signify when this has been the case. The failure to inject an error is also an indication that not all SEUs will affect the logic configuration of the device, so such failures are also of value when using mode 4.

There are two reasons why certain locations cannot be changed. First, each frame contains 16 unused bits (Bits 656 to 671 shown in [UG191](#), *Virtex-5 FPGA Configuration User Guide*, see the Frame Bits section), which cannot be written but are predictable in their locations. There are also some frames, particularly near the end of the device, that have more unused bits. Second, when the design uses a LUT6 for distributed memory (RAM or SRL functions), the write controls associated with that LUT6 are passed from the configuration interface to the design. So even though the bits of the LUT6 have a position within the configuration memory map, they have become nonwritable from the configuration interface. The number and locations of these bits are totally design dependant. While an SEU would have an effect on these bits, the change would be to the design's variable data and not the logical configuration. The readback CRC also ignores bits associated with distributed memory, so even if an error could be injected in these locations, it would not be detected and reminds us that a design must be implemented to tolerate data errors when necessary.

Table 4: Error Injection Configuration Size for Each Virtex-5 Device

Device	Configuration Device Size (Maximum ADDR)	
	Decimal	Hexadecimal
XC5VLX30	7,235,680	0x0006DB500
XC5VLX50	10,854,176	0x000A48F80
XC5VLX85	18,348,320	0x00116E980
XC5VLX110	24,464,864	0x00173E200
XC5VLX155	33,984,736	0x002052500
XC5VLX220	46,412,000	0x002C2C500
XC5VLX330	69,618,656	0x004242780
XC5VLX20T	4,883,264	0x00049FE00
XC5VLX30T	7,576,800	0x00072E980
XC5VLX50T	11,365,856	0x000AC5E40
XC5VLX85T	18,860,000	0x0011EB840
XC5VLX110T	25,147,104	0x0017E4B00
XC5VLX155T	34,666,976	0x0020F8E00
XC5VLX220T	47,094,240	0x002CD2E00
XC5VLX330T	70,642,016	0x00433C500
XC5VSX35T	9,618,272	0x000921000
XC5VSX50T	14,428,064	0x000DB1800
XC5VSX95T	25,787,360	0x001881000
XC5VSX240T	59,290,592	0x003868F80
XC5VFX30T	9,786,208	0x00094A000
XC5VFX70T	19,573,728	0x001294000
XC5VFX100T	28,180,448	0x001AC9400
XC5VFX130T	35,225,888	0x00217B900
XC5VFX200T	50,269,280	002FCE800

Notes:

1. Larger values of ADDR can be used but are not recommended in normal usage of mode 4.
2. These figures can also be used when estimating device configuration FIT rates.

It might be helpful to know a little about the operation being performed by the macro in mode 4 because this will explain the relatively long duration of the operation when using large values of ADDR. The macro begins by reading the first frame (1,312 bits) of configuration from the device and initializing a temporary pointer to zero corresponding with the first bit of that frame. It then repeatedly increments the temporary pointer value until it equals the ADDR value at which point it will invert the bit and write the frame back into the configuration memory to inject an error. Each time the incrementing of the pointer reaches the end of a frame, the next frame is read from the configuration memory enabling a natural sequential read of the entire device. However, that sequential read will frequently encounter the many frames associated with the contents of block memory (block RAM) as well as some other non-configuration related frames. When these frames are identified by the macro they are ignored in the same way that the readback CRC circuit does. The temporary pointer is only incremented when configuration frames are identified. This mechanism ensures that errors are only injected in to locations defining logical configuration of the device.

Mode 5 – Increment the Error Injection Index (Index = Index + 1)

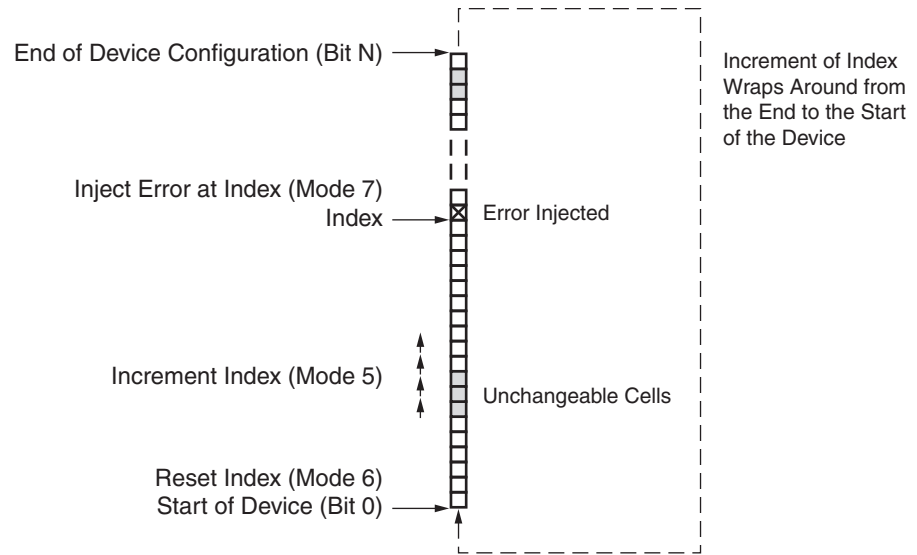
Mode 6 – Reset the Error Injection Index (Index = 0)

Mode 7 – Inject Configuration Error at Location = Index

These three modes work in combination to provide the alternative scheme for error injection. Preparation for error injection using these modes can be time consuming and might require some careful tracking, but it does have the advantage that the actual error injection operation is predictably fast in comparison to mode 4 error injection which is dependant on the value of the specified address.

As shown in [Figure 8](#), mode 7 is used to inject the error into the configuration memory. The location of that error is defined by the current value of an internal index which is a pointer to a specific configuration cell of the device. It is necessary to visualize the configuration memory of the device as being organized as N locations of 1-bit where N is the total number of configuration cells in the device. This is fundamentally the same number of bits observed by the readback CRC and excludes the contents of block memory (block RAM) in the same way. The index initially points to the first bit of the first frame in the configuration memory and can be restored to this position by mode 6. The index can be made to point at any location by using mode 5 to increment its value as many times as required. If incrementing the value should reach the number of configuration bits in the device (N), the next increment will effectively reset the index zero. Therefore, this scheme requires the combined use of modes 5 and 6 to set the index value defining the location for error injection followed by the use of mode 7 to inject the error.

It is useful to recognize that when the index has been incremented n times since it was last reset, an error injected using mode 7 will be at the same location as an error injected using mode 4 with $ADDR = n$.

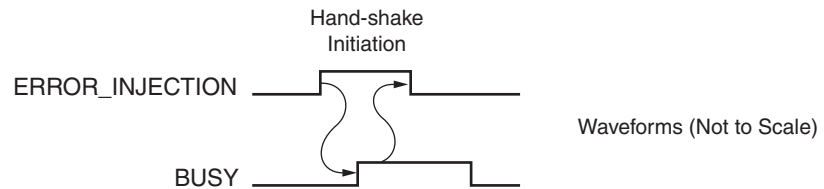


X864_08_022609

Figure 8: Use of Modes 5, 6, and 7 to Inject a Configuration Error

To reset the index to the first location in the device, select mode 6 by setting $\text{MODE} = 110$ and apply a pulse to the MODE_EN control. Once in this mode, the ERROR_INJECT input needs to be driven High to invoke the reset. To increment the index value, select mode 5 by setting $\text{MODE} = 101$ and apply a pulse to the MODE_EN control. Once in this mode, the ERROR_INJECT input needs to be driven High to invoke the increment. During each index increment operation, the macro ensures that only the configuration frames defining logic configuration are permitted. The frames associated with the contents of block memory (block RAM) as well as some other non-configuration related frames are identified and ignored in the same way that the readback CRC circuit does during device scans.

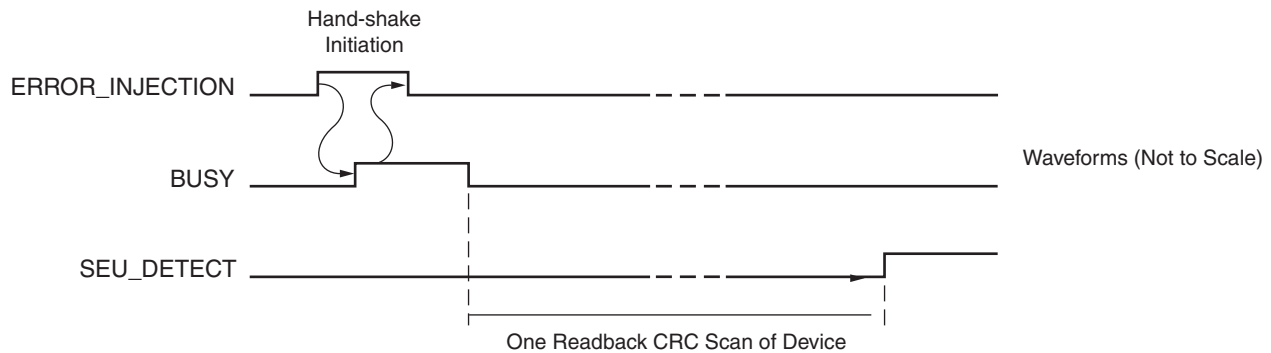
In modes 5 and 6, the ERROR_INJECT is being used as an operation control; it does not inject an error into the configuration memory. ERROR_INJECT should remain High until the BUSY output of the macro is observed to go High confirming that the operation is taking place after which ERROR_INJECT can be deasserted (Figure 9). BUSY will remain High for the duration of the operation. The device will perform readback CRC detection except when an operation is actually taking place ($\text{BUSY} = 1$). Less consistent pulses will be present on the DETECTION_ACTIVE output during operations as the macro accesses the configuration memory.



X864_09_022609

Figure 9: Waveforms Performing Index Reset Using Mode 6 and Index Increment Using Mode 5

To inject an error at the indexed location, select mode 7 by setting $\text{MODE} = 111$ and apply a pulse to the MODE_EN control. Once in this mode, the ERROR_INJECT input needs to be driven High to invoke injection of the error. As with modes 5 and 6, the handshake sequence must be completed using BUSY and the readback CRC detection will be interrupted (Figure 10). However, once the readback CRC resumes and completes the first scan of the device, the injected error will be detected and reported by the SEU_DETECT signal (and INIT_B pin).



X864_10_022609

Figure 10: Waveforms of a Successful Error Injection (ERROR = 0) Using Mode 7.

1. Not all indexed locations can be changed. If mode 7 is unable to inject an error, the ERROR output will be High as the BUSY signal returns Low. The failure to inject an error is also an indication that not all SEUs will affect the logic configuration of the device, so such failures are also valid results when using mode 7. There are two reasons why certain locations can not be changed. First, each frame contains 16 unused bits (Bits 656 to 671 shown in [UG191](#), *Virtex-5 FPGA Configuration User Guide*, see the Frame Bits section), which cannot be written but are predictable in their locations. There are also some frames, particularly near the end of the device, that have more unused bits. Second, when a LUT6 is used for distributed memory (RAM or SRL functions) by the design, the write controls associated with that LUT6 are passed from the configuration interface to the design. So even though the bits of the LUT6 have a position within the configuration memory map, they have become non-writable from the configuration interface. The number and locations of these bits are totally design dependant. While an SEU would have an affect on these bits, the change would be to the design's variable data and not the logical configuration. The readback CRC also ignores bits associated with distributed memory, so even if an error could be injected in these locations, it would not be detected. This is a reminder that a design must be implemented to tolerate data errors when necessary.

Macro Size and Analysis of Reliability

The SEU controller macro occupies approximately 95 logic slices (with some variation due to mapping in a complete design) and 2 block memories of 18 Kb (block RAM in Virtex-5 FPGAs is actually 36 Kb, but these can be divided to form two smaller memories). The macro also includes the ICAP_VIRTEX5 and FRAME_ECC_VIRTEX5 primitives required for the detection, correction, and error injection tasks.

Since the macro is itself part of the FPGA, it is also susceptible to SEUs. For this reason, the same analysis must be applied to the macro as well as the whole device or any part of the design.

Starting with the configuration cells, we know that the nominal value from [WP286](#), *Continuing Experiments of Atmospheric Neutron Effects on Deep Submicron Integrated Circuits* is 151 FIT/Mb. However, we need a way to convert the 95 logic slices and their associated interconnect together with the configuration and interconnect (not data contents) of the 2 block RAMs into a number of configuration bits. To facilitate this process, [Table 5](#) can be used to generate a reasonably accurate estimate. For the macro, that value is $(95 \times 1,181) + (2 \times 585) = 113,365$ bits, or 0.108 Mb. This results in a nominal macro configuration FIT of 16.33 or an MTBF of approximately 6,992 years.

Table 5: Approximate Number of Configuration Bits Associated with the Most Common Device Features

Device Feature	Approximate Number of Configuration Bits
1 logic slice	1,181
1 block RAM (36 Kb)	1,170
1 block RAM (18 Kb)	585
1 I/O block	2,657
1 DSP48E slice	4,592

Notes:

1. In all cases, the device feature includes all the programmable interconnects associated with getting signals to and from the feature.
2. These features account for approximately 95% of the total configuration cells of each device.
3. Block RAM does not include the data contents of the memory which must be analyzed separately.

An alternative, somewhat pessimistic way to estimate configuration FIT of a design is simply to evaluate the percentage of a given device that is occupied and use that proportion of the total configuration bits for the device. For the SEU controller macro in an XC5VLX50T device, that equates to 1.32% of the slices and 1.67% of the block RAMs. The table showing the readback CRC frames for each device in this application note states that the XC5VLX50T device has 8,663 frames, which each consist 1,312 bits—meaning a total of 11,365,856 configuration bits. Therefore, the macro is estimated to be associated with approximately 1.5% of 11,365,856 bits, which is 170,488 bits (or 0.163 Mb)—leading to a nominal configuration FIT of 25.55 or a MTBF of 4,650 years. This estimate is 50% more pessimistic than the previous, more accurate estimate, but might be a quicker way to obtain a first approximation, especially when evaluating designs that use a wider selection of features not covered by [Table 5](#).

Regardless of how the nominal configuration FIT is estimated, less than 20% of configuration cells (actually less than 10% in typical designs) would directly impact the active design if an SEU occurred. Therefore, it is reasonable to scale the estimates by a factor of 5. In this case, scaling the more accurate estimate reveals an operational configuration FIT for the SEU macro of approximately 3.27 FIT or an MTBF of 34,960 years.

Next, the susceptibility of the block RAM contents to SEUs must be considered. This is initially a straightforward calculation because there are two block RAMs, each of 18 Kb, which is a total of 36,864 bits (or 0.0352 Mb). The nominal block RAM data FIT from [WP286, Continuing Experiments of Atmospheric Neutron Effects on Deep Submicron Integrated Circuits](#), is 635, resulting in a block RAM data FIT for the SEU controller macro of 22.35 or an MTBF of 5,107 years. However, closer analysis of the macro operation reveals that less than half of the block RAM data contents can be considered critical, and therefore the meaningful data FIT of the macro is approximately 11 or an MTBF of 10,377 years. The number of data bits associated with flip-flop and distributed RAM contents are so small that no meaningful values can be generated.

Combining the configuration (FIT = 3.27) and data (FIT = 11) values, the complete operational susceptibility of the SEU controller macro to a potentially disruptive SEU is a FIT of approximately 14.3 or an MTBF of 7,983 years. Even with this low potential for an SEU influencing the SEU controller macro, what are the potential affects of such an event if it happens?

Configuration SEU detection is performed by the dedicated readback CRC logic, which is almost totally independent of the macro and therefore a configuration error will almost certainly be detected. There are some minor threats to detection for which precautions can be taken. If an SEU affects the clock supplied to the ICAP_VIRTEX5 primitive within the macro, then the readback CRC logic will be unable to continue. Monitoring the DETECTION_ACTIVE output of the macro, which is a direct connection to the SYNDROMEVALID output of the

FRAME_ECC_VIRTEX5 primitive, can be used to confirm that readback CRC is operational. When readback CRC detects a configuration error, then the CRCERROR output on the FRAME_ECC_VIRTEX5 primitive used by the macro is driven High. If the SEU in some way prevents the macro from observing the CRC error status, it will not be able to react to it as intended. However, the INIT_B pin on the device package will still be driven Low (unless disabled by user constraint), facilitating an external observation of the configuration error independently to the macro. The concept of adding supervisory logic and/or components to monitor a macro that is already supervisory in nature is really quite extreme and requires careful consideration of its reliability as well as deciding what system-level reaction it should invoke. However, it is sensible to use an external component to perform this monitoring as independently as possible and to combine this with the overall monitoring and control of device configuration. This will then facilitate scheduled and emergency maintenance strategies in support of the principle running repairs strategy.

Any SEU that does occur could impact the configuration or the informational data of the macro. In either case, the expected observation is that an SEU to the macro is not an SEU to the system application and therefore of no significant consequence unless it results in subsequent disruptive behavior.

If the SEU affects data, there will be no readback CRC error generated, and the macro should remain in a waiting state. There is a small possibility that the macro can behave erratically as a result of the change to data and that might subsequently prevent a correct response to a subsequent readback CRC error report. If the SEU affects the configuration, then a readback CRC error will be generated. Depending on the precise nature of the logical error, the macro might function adequately to correct that error. If not, then as with a data error, some kind of erratic behavior could be expected, but it is unlikely that this could have further effect on the system application other than generating unusual status signals. Remember that the readback CRC detection will drive the INIT_B Low (unless disabled by user constraint), and a prolonged Low on this pin is a solid indication that the macro is not correcting an error. This would also be the case if the macro were not placed in the correction mode (mode = 001), so this situation can be emulated.

The greatest concern is that an SEU to the macro could corrupt more of the device configuration by either unintentional injections of errors (normally used during test and evaluation), or corruption caused by erroneous operations during attempted error correction writes via the ICAP_VIRTEX5 primitive. In practice, operations resulting in writes to the configuration cells require precise implementation of a series of tasks and means that there is naturally no single point of failure, and erratic behavior would not be able to cause this kind of corruption.

Reference Design Files

The reference design package, including Verilog and VHDL reference designs, is available on the Xilinx website:

<https://secure.xilinx.com/webreg/clickthrough.do?cid=115162>

SEU Controller Macro

Table 6: SEU Controller Macro Matrix

Parameter	Description
Developer Name	Xilinx
Target Devices (stepping level, ES, production, speed grades)	Virtex-5 FPGAs
Source Code Provided	Yes (HDL only)
Source Code Format	VHDL and Verilog

Table 6: SEU Controller Macro Matrix (Cont'd)

Parameter	Description
Design Uses Code/IP from an Existing Reference Design/Application Note, Third Party, or CORE Generator™ software	Yes. Incorporates PicoBlaze™ processor
Simulation	
Functional Simulation Performed	Verified in hardware (simulation models for configuration unavailable)
Timing Simulation Performed	
Testbench Used for Functional Simulations Provided	
Testbench Format	
Simulator Software Used/Version (for example, ISE® software, Mentor, Cadence, other)	
SPICE/IBIS Simulations	
Implementation	
Synthesis Software Tools Used/Version	XST
Implementation Software Tools Used/Versions	ISE software, version 10.1, service pack 3
Static Timing Analysis Performed	
Hardware Verification	
Hardware Verified	Yes
Hardware Platform Used for Verification	ML505 Virtex-5 FPGA evaluation platform

Reference Design

Table 7: Reference Design Matrix

Parameter	Description
Developer Name	Xilinx
Target Devices (stepping level, ES, production, speed grades)	Virtex-5 FPGAs
Source Code Provided	Yes (VHDL and PicoBlaze assembler source code)
Source Code Format	VHDL
Design Uses Code/IP from an Existing Reference Design/Application Note, Third Party, or CORE Generator software	Yes. Incorporates PicoBlaze processor
Simulation	
Functional Simulation Performed	Verified in hardware (simulation models for configuration unavailable)
Timing Simulation Performed	
Testbench Used for Functional Simulations Provided	
Testbench Format	
Simulator Software Used/Version (for example, ISE software, Mentor, Cadence, other)	
SPICE/IBIS Simulations	
Implementation	

Table 7: Reference Design Matrix (Cont'd)

Parameter	Description
Synthesis Software Tools Used/Version	XST
Implementation Software Tools Used/Versions	ISE software, version 10.1, service pack 3
Static Timing Analysis Performed	
Hardware Verification	
Hardware Verified	Yes
Hardware Platform Used for Verification	ML505 Virtex-5 FPGA evaluation platform

Conclusions

The selection of an SEU strategy should begin with a realistic assessment of the risk of occurrence for which Xilinx provides some of the most comprehensive data in the industry. The effect that an SEU can have on your design should be assessed and compared with the requirements for your system to maintain operation should such an event occur.

Virtex-5 devices contain a built-in readback CRC circuit, which automates the detection of SEU errors and enables the system to take appropriate action.

Designs required to maintain operation must evaluate the merits of redundancy within the design, and the ability to perform local error correction using the provided SEU controller. The time from SEU detection to correction is rapid, but redundancy might still be considered necessary to bridge this period. Having redundancy might also imply that localized correction is not required for these rare events, but the macro can help keep the redundancy circuits smaller, reducing costs, and actually decreasing the statistical probability of an SEU in the first place.

Revision History

The following table shows the revision history for this document.

Date	Version	Description of Revisions
02/20/09	1.0	Initial Xilinx release.
03/05/09	1.0.1	Tech Pubs revision.

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.