SINGLE EVENT UPSET AND LATCHUP SENSITIVE DEVICES IN SATELLITE SYSTEMS

Richard H. Maurer
James D. Kinnison
The Johns Hopkins University
Applied Physics Laboratory
Laurel, MD 20723-6099

Abstract: We present a decision tree to systematically evaluate the potential use of single event sensitive devices in spacecraft systems. We present several concrete examples of branches on the tree.

1. Introduction

Satellite systems must be able to survive the hazards of space, including single event upset and latchup. However, the choice of electronic components may also be constrained by the need to perform a particular mission, by schedule, or by cost. Using radiation hardened parts is not feasible in many cases, so the effect of the radiation environment on integrated circuits must be assessed.

The sensitivity of each device must be measured to achieve reliable satellite system performance. In general, all tests of this nature involve exposing an integrated circuit to a monoenergetic charged particle beam and measuring the frequency of event occurrence. In practice, this process can be implemented in a variety of ways, some of which are more fruitful than others. In any case, the data gathered in single event upset or latchup testing is used to estimate the on-orbit behavior of a device.

Inevitably, some crucial integrated circuit exhibits undesirable behavior; a device may latch up, or may be overly sensitive to single event upset. This typically occurs in the only device in a system which cannot be replaced. Therefore, some means of recovering from undesirable phenomena must be implemented. For instance, a resistor in series with a device which latches up can prevent damage to the device; in addition, latchup detection and correction circuits may be implemented which automatically recover from latchup. Similarly, system-level techniques can be used to compensate for a device which is overly sensitive to single event upset. A decision tree representing a systematic approach to evaluate the use of single event sensitive devices is given in Figure 1.

2. Latchup and Latchup Protection

The first decision point on the Single Event Effects Decision Tree (Figure 1) is the box containing the question, "Does the device latch?". Ideally, one does not want to fly any such device; but there may be compelling reasons for considering a latchup sensitive device (e.g. a gate array or digital signal processor). To perform the required mission may necessitate use of such a device.

CMOS VLSI digital devices generally need to be screened for latchup using a Californium system or an accelerator as a heavy ion source. If the device does latch up, the experimenter needs to determine important parameters including the latchup threshold, the latchup asymptotic cross section, the range of the latched state currents (usually in hundreds of milliamps) and the range of the latched state holding currents (usually less than 10 milliamps).

Hopefully, the latched state will have some distinctly different characteristics from the normal operating states of the device so that a latchup protection circuit can be designed, if necessary. The difference between the operating current and the latched state current is one parameter that can be sensed; the change in logic state on an output or driver pin is another.

In order to return a device from a potentially destructive latched state to a normal operating state, the supply current to the device must be limited by a resistor to prevent device burnout, and the supply current must be reduced to a level below the holding current. Implementing such protection in satellite hardware creates weight, volume and power penalties. There may also be some performance impact on the device itself especially with respect to speed of operation.

For some missions, such as a shuttle mission, the environment is benign and the mission duration is short. In such a case the program office may elect to accept the risk of the device latching and use it without any protection.

An intermediate case could be one in which adequate latchup protection is not possible because of an inability to sense a latched state or monitor the device. Even if the program were willing to support the latchup protection circuit design, a redesign or part substitution would be necessary.

Example: ADSP2100A[1]

A latchup protection, detection and removal circuit may also have to restart a device such as a processor to continue normal system operation after a latchup occurs. An example of such a circuit, designed for the Analog Devices ADSP2100A digital signal processor, is shown in Figure 2.

A small resistance in series with the device power pins protects the chip by limiting the latchup current, and prevents the VCC bond wires from melting. When a latchup occurs, the current through the resistor increases and the voltage applied to the ADSP2100A drops. This drop is sensed by a comparator, producing a signal that clocks a flip flop, creating a latch detect signal. This signal is used to turn off the series transistor, which removes power from the device VCC pins. At the same time, the entire board is reset. In order to extinguish the parasitic SCR that forms the latchup, all sources of current that sustain the SCR must be eliminated. In CMOS devices, the device input pins are connected to the chip VCC bus through a diode. Therefore, input pins driven high could supply current to sustain the SCR. To prevent this, the latch detect signal is used to tri-state or

force low all signals that drive ADSP2100A inputs.

The ADSP2100A processor board is under the control of the subsystem main processor. The main processor can set or clear the latch detect flip flop with software. After the latchup has been extinguished, the main processor clears the latch detect flip flop, which allows power to be applied to the device and inputs to be driven.

Unforeseen problems, such as a greater than expected device current increase due to total dose damage in the ADSP2100, could cause the supply current to exceed the threshold of the latch detect circuit. In order to prevent continuous triggering of the latch detect circuit due to total dose damage, the function can be disabled. The latchup disable/enable function is under software control of the subsystem main processor. When disabled, the latch current would limited by a resistor, but would not be automatically switched off in the event of a latchup.

The latch detect signal is used to generate an interrupt to the subsystem main processor. The interrupt service routine reloads the RAM based ADSP2100A software, clears the latch detect flip flop after a fixed length delay, and then clears the ADSP2100A reset, allowing the ADSP2100A to resume operation.

Software on the subsystem main processor is also used to detect latchups below the threshold of the latch detection circuit, bit flips that cause the ADSP2100A to malfunction and latchups that take place when the latch detect circuit is disabled. The ADSP2100A is programmed to generate periodic interrupts to the main processor. If the interrupt is not generated, a timer in the main processor times out and generates an interrupt. This interrupt service routine performs a similar procedure as the latch detect interrupt service routine to re-start the ADSP2100A.

A reliable latchup protection circuit must not be sensitive to the radiation environment. To this end, we use a radiation hardened main processor, a CMOS/SOS 1750A. In addition, the devices used in the latchup protection circuitry are not sensitive to single event upset or latchup and are hard enough to survive in the expected particle environment over the mission lifetime.

To verify that the latchup circuit functioned properly, we exposed samples of the commercial ADSP2100A to heavy ions at Brookhaven National Laboratory. In all, more than 60 latchups were generated in three samples, with no damage to any device. In addition, the devices stopped operating several times due to single event upsets; the protection circuitry detected this state and reset the upset device to restore proper function.

3. Transient Upset

For devices which prove to be latchup immune and, therefore, candidates for flight hardware, their susceptibility to soft errors or transient single event upset (SEU) needs to be assessed. A fairly inclusive data base is updated and published in odd numbered years in the IEEE Transactions on Nuclear Science by the

Radiation Effects group at JPL. Another approach is to test for SEU sensitivity using an accelerator at the Brookhaven or Berkeley facilities.

After determining the SEU threshold and cross section as a function of particle type and energy, we try to categorize and assess the kinds of upsets that occur with their consequent impact on the particular spacecraft system. If the transient upset causes a serious system malfunction such as a change of attitude, orientation or pointing, then a system or device level protection scheme similar to that devised to cope with latchup must be developed.

In contrast, if soft errors do not lead to system malfunction, the designer need only consider if the error rate for the device itself is too high for that device to perform acceptably. If the device error rate in the mission environment is low, one may use the chip as is. If the rate is too high, some error detection and correction (EDAC) scheme such as a parity check may be employed for the chip (often a memory in this case). With any EDAC comes an overhead exhibited as a loss of memory cells dedicated to error checking and a decrease in memory speed. Such performance impacts may or may not prove acceptable to the system design. Finally, it is possible that for the same mission a chip's upset rate might be acceptable for a science data system but not for a command or attitude control system.

In testing devices for single event upset it is important to exercise and monitor flip flops or logic gates as well as memory registers. Even for random access memories (RAMs) or read only memories (ROMs) the test engineer should consider if any peripheral logic or power structures could be upset in addition to the memory cells.

In general, while device level protection devotes nodes or registers of the device to its own self-monitoring (thus decreasing device performance), system level protection requires additional hardware and/or procedures which do not necessarily decrease device performance but add complexity to system operation. The designer must monitor the upset device with a second hardened, intelligent controller detecting faults exterior to the chip in question. To rely on the system-level-generated resets requires a thorough testing of fault detection and correction capability. All error states should be known, exercised and reset efficiently.

Example: FIFO Memory[2]

A typical 512 x 9 bit First In First Out Memory consists of a circular buffer of 512 words and control logic with a read and a write pointer to keep track of filled and empty cells. Three status flags can be used to monitor the state of the FIFO; the empty, half-full, and full outputs can be used to determine if the FIFO is empty, full, less than half full, or greater than half full.

Two types of error can occur in these devices. The first type, data errors are easily detected. However, the registers which contain the read and write pointers can also upset. These control errors are not so easily handled. For instance, if the read pointer is advanced by an upset, all the data between the previous and current location is lost. If the write pointer is

upset, a block of invalid data may be read, or a block of valid data may be overwritten. The worst control error, however, occurs when the read pointer is advanced beyond the write pointer (or when the write pointer is set behind the read pointer) by an upset. In this case, the entire data block is lost, and the device must be reset before it will function again. Control errors make up about 10% of the total cross section for a typical device.

Example: 93L422 Static RAM[3]

As an example of using a device with a high upset rate without protection, consider the notorious 93L422 static RAM. Fifteen of these devices were used in a digital filter bank in an environment causing a predicted eight upsets per device per day. The consequent 120 upsets a day average to one upset every 12 minutes. The digital filter bank accumulates 128 waveform sample amplitudes over 40 pulses in each burst and the burst period is 8893 microseconds. Thus, in one second, each waveform sample represents the average of 4498 pulses. We would not expect the average performance to be affected in the event of a single upset during any one pulse.

In this case it was also of interest to estimate the effect of a single upset in determining whether a perceptible perturbation in the data might result. A simulation of an upset in the output memory was run. The conclusion was that the soft error would not produce a perturbation in the data for averaging times greater than or equal to one second.

Example: 80C86 Microprocessor[4,5]

For the case of the 80C86 microprocessor with a predicted rate of one upset every five days, the designer of the command system decided it was intolerable to reload the command system several times a week and a replacement part was found.

Example: 80186 Microprocessor[4,5,6,7]

Finally, we describe the use of the Intel 80186 microprocessor in an adaptive tracker. This device was predicted to upset about once every three days for a low earth orbit mission. However, since the 80186 was also susceptible to proton induced upsets, a rate of one upset every five hours could be reached should a large solar flare occur. A watchdog timing scheme was designed to cope with upsets.

In the adaptive tracker, SEUs are detected by two watchdog timers. The "burst rate" timer must be reset approximately every 8.5 milliseconds, and the "track rate" timer every 50 milliseconds. In the event that either timer is not reset in time, a processor reset is generated.

When a reset occurs, the microprocessor begins executing the system bootup routine. This bootup routine interrogates the command word to determine what type of reset has occurred. In the event of an error reset, which is the default, the initialization routine assumes the state of the processor is contained in write-protected RAM. This state includes the last mode command executed, (Idle, Standby, Calibrate or Track), and the values for all of the control variables.

On reset, the initialization software brings the adaptive tracker up in the idle mode. This portion of the recovery requires about 16 milliseconds. The bootup routine then copies the last mode command from write-protected RAM to the command word. Finally, the bootup routine flags the command processor ready to run, and turns control over to the table manager program.

At this point, the table manager program will invoke the command processor, as it does whenever a command is received, and the command processor will execute the command word. In executing the command word, the command processor will set the synchronizer parameters, and then signal whatever task is necessary to transfer to the correct mode of operation. This processing requires under 2 milliseconds to complete. Thus, the adaptive tracker processor can recover from an error reset to the previous mode of operation in under 20 milliseconds.

4. Summary

We have presented a single event decision tree and discussed some general principles of handling single event sensitive devices. We have given several examples:

1) ADSP2100A - device latches, adequate protection is possible, performance degradation acceptable, use with protection;

2) 93L422 - device does not latch but has high upset rate without causing system malfunction, use as is;

3) 80C86 - device does not latch but has high upset rate which causes serious system malfunction, effective protection not possible, redesign with replacement part;

4) 80186 - device does not latch but has high upset rate which might cause serious system malfunction under certain environmental conditions, system level protection is effective, use with protection.

It is sometimes possible to use devices which are sensitive to single event effects in satellite systems. Effective use of these devices requires three things: good susceptibility data, adequate protection mechanisms, and systematic evaluation of the appropriateness of the protection scheme. With these, the risk associated with using unhardened integrated circuits in the orbital charged particle environment can be minimized.

References

1. J. D. Kinnison et al., "Radiation Characterization of the ADSP 2100A Digital Signal Processor," IEEE Trans. Nucl. Sci. 38, 1398-1402, December 1991.

2. J. D. Kinnison et al., "A Summary of Recent VLSI SEU and Latchup Testing," presented at IEEE Nuclear and Space Radiation Effects Conference, Data Workshop, New Orleans, 15 July 1992.

3. J. L. MacArthur, "Bit Upsets in the TOPEX Altimeter Digital Filter Bank," APL Memorandum S2R-84041, 28 March 1984.

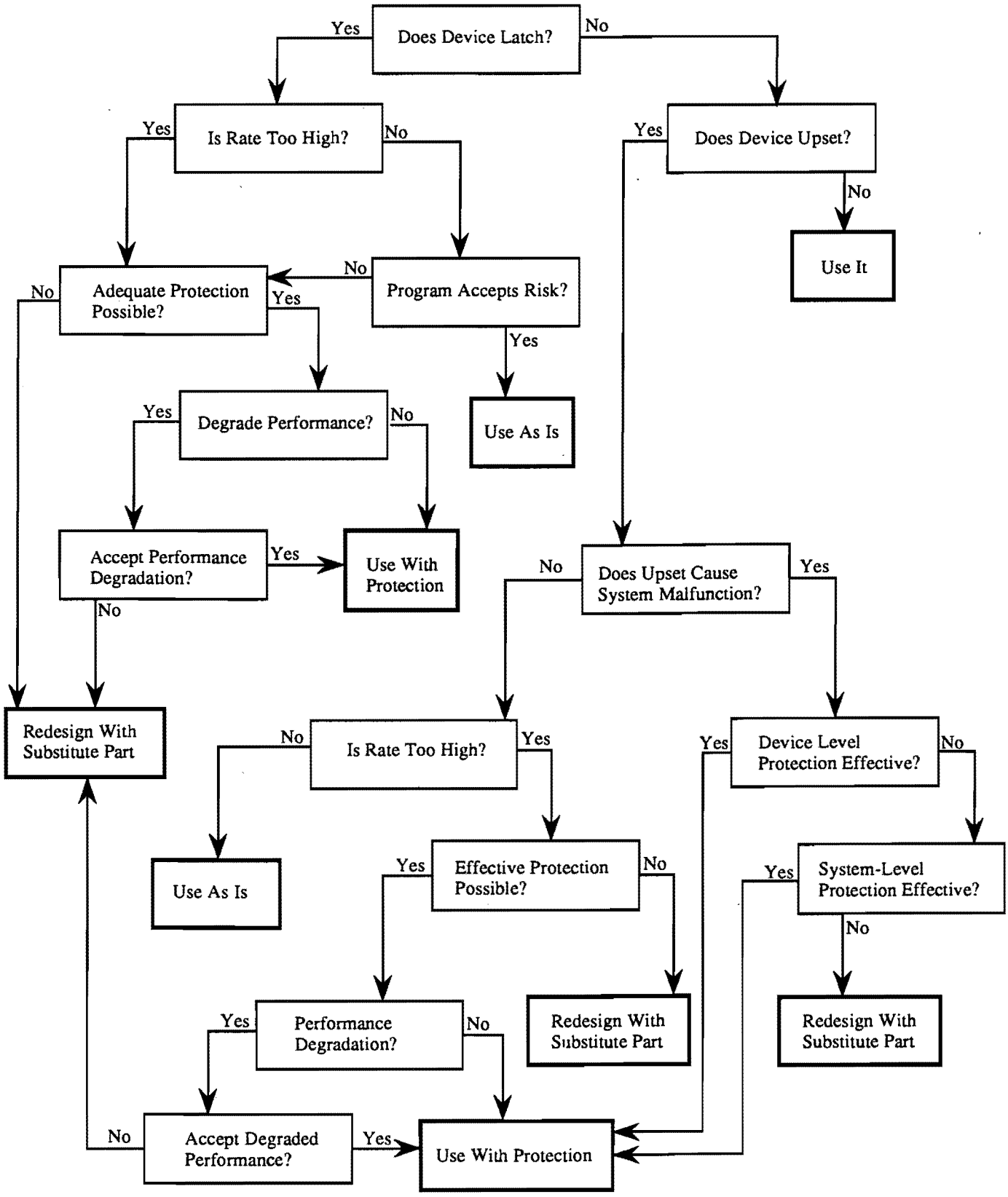
4. R. H. Maurer, J. D. Kinnison and P. L. McKerracher, "A Summary of Recent VLSI SEU and Latchup Testing," presented at 1990 Single Event Effects Symposium, Los Angeles, 24 April 1990.

5. J. D. Kinnison et al., "Single Event Survivability of Unhardened VLSI Devices, "Proceedings of the 1990 Advanced Microelectronics Technology Qualification, Reliability and Logistics Workshop, pp 239-248, San Diego, 29 August 1990.

6. C. W. Spaur, "TOPEX Radar Altimeter Signal Processor SEU Recovery," APL Memorandum S2F-890126, 14 March 1989.

7. R. H. Maurer et al., "Space Radiation Qualification of a Microprocessor Implemented for the Intel 80186," Proceedings of the 2nd Annual AIAA/USU Conference on Small Satellites, Logan, Utah, 18-21 September 1988.

Figure 1. Single Event Effects Decision Tree



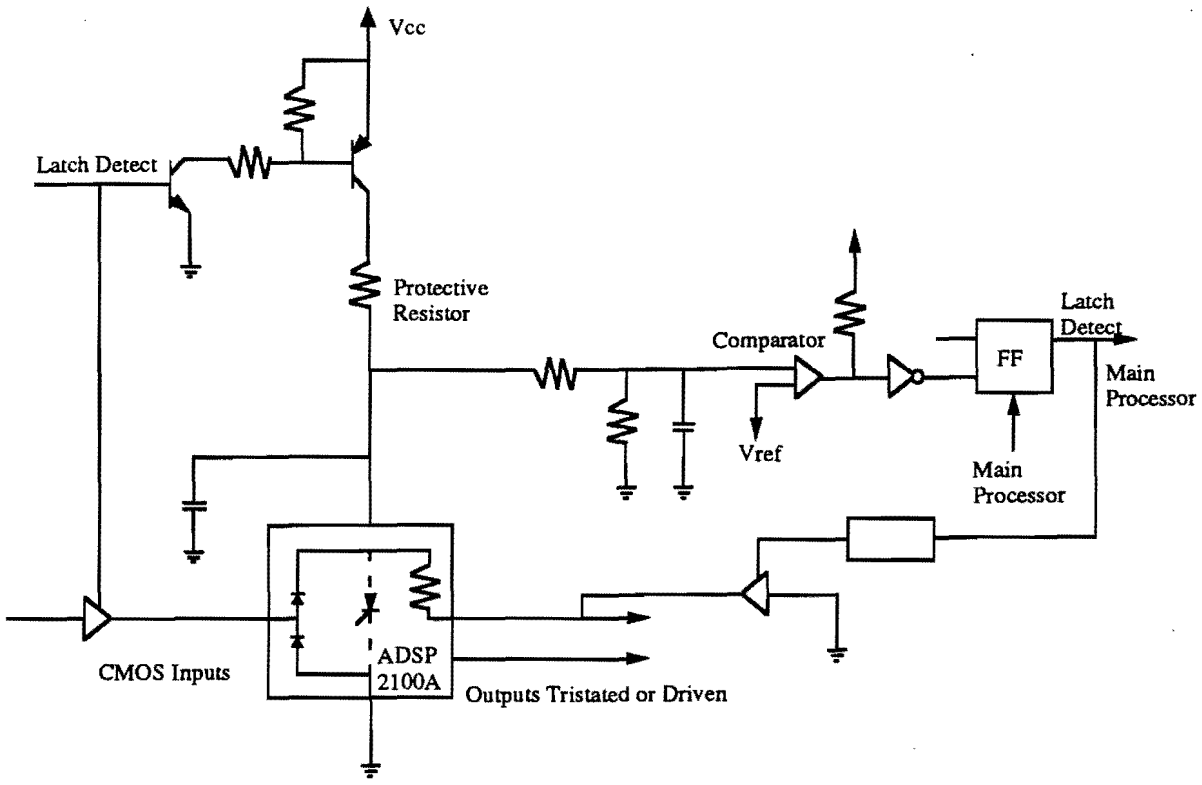


Figure 2. Block Diagram of the latchup protection circuit for the ADSP2100A.