Application Report
SPNA109 – July 2008

Discriminating Between Soft Errors and Hard Errors in RAM

Kevin Lavery

ABSTRACT

Memory checks detect both hard and soft errors. The application response to these different error types should differ because hard failures are caused by the circuit itself while soft errors are caused by some external particles that destroys the data but does not degrade the circuit functionality.

1 Introduction

A strong safety concept requires periodic self-checks of memory and logic to insure that the microcontroller system is operating correctly. As these checks verify that the memory element can be read and written, the tests are destructive to the memory contents; often then, these checks are run at ignition when failures can be checked prior to entering the safety algorithm. For RAM, the values stored in the memory are constantly changing, and the exact contents of this memory cannot be checked against some expected/known pattern.

The memory requires some level of duplication for protection, provided that the source of the errors does not affect both memories in the same way. This duplication of the memory data can be accomplished through a fully redundant memory, ECC logic for the memory, or parity bits for the memory. Any of these items which check the memory’s data will detect hard failures as well as soft failures.

In this context hard failures are errors that occur through process defects and/or circuit bugs – hard failures are repeatable with the correct sequence of actions within the microcontroller. Soft errors occur through no failure of the circuit or defect but due to an external source that causes the data to change. An example of a cause of soft errors is high energy atmospheric neutrons. This application report discusses how to distinguish hard and soft errors in protected RAM.

2 Memory Redundancy Types

Checking memory during the application is very valuable, and for non-volatile memory, the contents can be checked against a checksum or some other fixed reference. However, volatile memory has no fixed reference and so requires extra circuitry to allow an in-application memory check.

- A redundant memory provides a full map of the exact data. The redundant memory map has a large penalty in terms of area; it allows an exact identification of every wrong data bit, but it is not possible to determine which memory map is correct. Redundant memory will never mask a failure and so catches every failing bit (not caused by a common source). The primary deficiencies in a fully redundant memory are 1) the inability to correct the failure and 2) a large area penalty.
- Parity adds a parity bit for every *n* bits of memory; TI uses parity on the peripheral RAM – 1 bit of parity for every 8-bit word or every 32-bit word. The parity bit is a flag based upon whether the number of 1s in a word is even or odd. The parity flag is a simple 1-bit redundancy that allows for detection of any single bit failure; the key point is that any write operation is responsible for calculating and writing a new polarity flag. So, even though the RAM data is constantly changing, the polarity bit is changing in a synchronized way for *valid write operations*.

  The parity bit may catch multiple bit corruptions (in a word [1]) though it is only guaranteed to detect a single bit failure; parity provides detection for a single bit failure but cannot recover the correct data.

- ECC adds several bits for every *n* bits of memory; TI uses 8 ECC bits for every 64-bit RAM word. With these ECC bits, any single or double bit error is detected; errors with more than 2 bits incorrect in the 64-bit word are not guaranteed to be detected. If the error is a single-bit error, the ECC can uniquely detect and correct the failure. Again, the fact that bits are spread along a row [1] has a large benefit in making the ECC-solution robust.

3 Difference Between Hard and Soft Errors (and Quasi-Soft Errors)

Checking the memory of the device is intended to find all errors. However, the type of error – hard, soft, or quasi-soft – will play a large part in the application's response to the error.

Figure 1 and Figure 2 show a latching memory element. In Figure 1, data is driven onto the left port of the latch (and in SRAM, the complementary data is written onto the right port). The data is latched in the cross-coupled inverters which store the data until the next time data is written. Figure 2 shows the same cross-coupled inverters at a transistor-level. This basic scheme is common to SRAM cells and flip-flop designs.

- A hard error is caused by a real error in the circuit – whether a design bug or a process defect. Because the hard error is related to a real circuit error, the circuit can be expected to fail repeatedly in the same manner. As an example, consider a highly resistive contact at the red X in Figure 2. When Data is driven high (and nData driven low), the latch is fully functional with both inverters driving at full strength. However, if Data is driven low (and nData driven high), the bottom inverter (driving Data) is weakened by the resistive contact and if the contact is resistive enough the latch will not hold the low value.

Table 1. Results of Defect in Figure 2

Data	Result
Low	Weakened latching – possibly failing
High	Good

- In a soft error, the data is corrupted without any circuit error – in this discussion, a soft error is caused by a physical particle (e.g. alpha, neutron) passing through a critical volume of the device. The neutron may break the silicon nucleus into different atomic weight nucleons (spallation). These nucleons move in different directions, conserving mass-energy and momentum, and the moving, charged nucleons produce a wake of charge separation. This wake of charge separation may recombine or some of the separated charge may be captured by an active node of the circuitry.

  As an example, consider a neutron interacting with the drain node of the transistor driving Data in Figure 2 (shown with a green dot). Let one of the spallation nucleons pass through the critical volume near the drain of the nmos transistor so that the drain can capture some of the separated charge.

  – If Data is high and nData is low, then the drain node where the transistor passes through is biased high while the p-well is biased to ground. Figure 3 shows the voltage difference that can capture some of the separated negative charge into the drain of the transistor.

    This captured charge produces a pulse of current that can flip the voltage of the Data node from high to low.

    Once the Data node flips from high to low, it begins to drive the nData node from low to high.

    If the pulse of current lasts long enough to maintain the Data node low until nData is driven high, the latch has completely flipped and the stored data is lost.

  – Conversely, if Data is low and nData is high, then the drain node will be at nearly the same voltage as the p-well, and there is little voltage difference that can capture some of the separated charge. In the absence of an external voltage difference, the separated charges will simply recombine.

Table 2. Results of Soft Error in Figure 2

Data	Result
Low	No voltage difference in affected drain; no data corruption
High	A voltage difference exists between the affected drain and its p-well. The voltage difference draws electrons into the drain and potentially flips the bit.

The point of this example is that even though the data has corrupted, the circuitry remains completely functional. Furthermore, the bit that got affected by the neutron was random and independent of the functionality of the board or the device.

A quasi-soft error occurs when the data in a circuit is corrupted by external stimuli but that corruption is repeatable or quasi-repeatable. An example of this type of error is electromagnetic interference. These types of errors are typically found in a device qualification stage rather than in the field. As such, they are not addressed in this document.

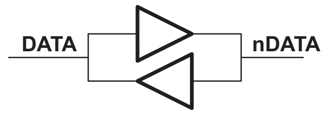

Figure 1. Latching (Memory) Element

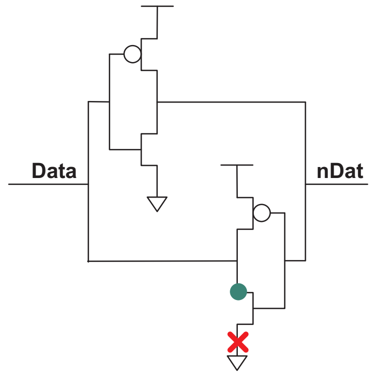

Figure 2. Latching Element at a Transistor Level

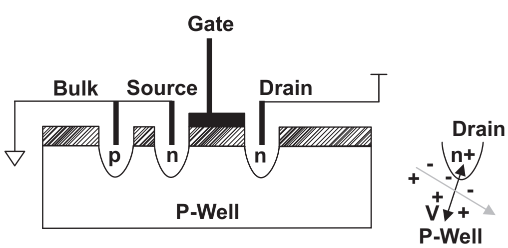

Figure 3. Configuration of Inactive nmos Transistor in an Inverter When pmos is Driving High

4 Soft Error Signature

Soft errors, by their nature, are rare occurrences. In the case of thermal neutrons produced from ^{10}B or alpha particles produced from trace radioactive elements in the device packaging, the quantity of these unstable elements is controlled in the device manufacturing and packaging. The neutron flux is not possible to prevent (about 13 neutron per square centimeter per hour in New York City), but fortunately the likelihood of a neutron interacting with the material it passes through is very small. The following produces a telltale signature for soft errors:

- Low probability of soft errors in the device
- Random distribution of the errors

In contrast to the soft-error signature, a hard error is localized to the defective circuit.

5 Algorithm for Determining Whether the Failure Type is Hard or Soft

The probability of a soft error is very small and the distribution of errors through the RAM array is random. Because of this failure signature, most devices should pass their entire lifetime without ever seeing a soft error [2]. Therefore, when a RAM error is detected, the location of the failure(s) should be saved into RAM. If the same RAM location(s) flips again, then it could be assumed that the failure is due to a deterministic failure source – either a hard failure or a quasi-soft error.

- On RAM arrays with ECC, single-bit failures [3] can be corrected during a RAM verify algorithm. As the RAM is read, it can be re-written into the array. Since the ECC corrects the single bit failures, the re-write corrects any incorrect bit. Any error is flagged, and the failing address should be preserved.
- On RAM arrays with parity, as the RAM is read out, the parity bit is compared to determine if the word has the correct parity. If a parity error occurs, the RAM must be re-initialized (or perhaps that word can be re-initialized). In either case, the RAM failing address is preserved.

The preserved RAM failing address should be saved to (EEPROM) memory and compared to any future RAM failures. The key point is that the failure rate due to soft errors in the RAM is small. While a single event can cause multiple words to have failures, events are quite rare [3]. This observation sets two criteria for classifying a RAM failure as hard:

- The RAM arrays experience an unreasonably high number of error events (failures at different times). These events could occur at different addresses.
- A specific RAM address sees multiple failures.

While a hard failure would likely appear multiple times, a soft error should never reappear.

6 Conclusion

Due to the dramatically different failure signatures of a hard error and a soft error, it is generally possible to distinguish these error types within the application. While soft errors are rare, it makes no sense to fail the device based upon a soft error. Therefore, a simple strategy of retaining the failing address and comparing future failures to that address distinguishes hard and soft errors.

7 References

1. The RAM's parity/ECC is based upon the logical word. However, soft errors typically corrupt physically close cells. Therefore, a RAM array that physically separates logically related bits provides an increased protection. It is typical to separates bits of the same word by interspersing words along a RAM row. This interspersing separates bits of the same RAM word. So, in order for a multiple-bit upsets to corrupt several bits of the same logical word, the upset must extent many bits along a row, and testing has shown that this failure type is rare. The point is: by separating logically-related bits, the chances of a single error affecting multiple bits of the same word decreases.
2. Given 1000 – 1500 FIT/Megabit on a device with 48Kbyte of RAM and an operating lifetime of 104 hours, one out of every 175 to 250 devices should be expected to see a RAM bit flip at some point in their operating lifetime. Obviously, multiple soft-error events on a given device would be quite rare. And even more rare would be a failure that occurred at the same bit location within the RAM. Since multiple words of the RAM array are interspersed, a multiple cell upset may create single-bit failures in multiple words.
3. Since multiple words of the RAM array are interspersed, a multiple cell upset may create single-bit failures in multiple words.

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Clocks and Timers	www.ti.com/clocks	Digital Control	www.ti.com/digitalcontrol
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
RFID	www.ti-rfid.com	Telephony	www.ti.com/telephony
RF/IF and ZigBee® Solutions	www.ti.com/lprf	Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265