APPLICATION NOTE

RELEASED 11/06

Designing with the UT8ER512K32 Monolithic 16M RadHard™ SRAM

1.0 Introduction

This application note describes how to use the UT8ER512K32 Monolithic 16M RadHard SRAM in different system configurations, including a detailed look at the bus signals and CPU interface. The reader will also gain an understanding of how error detection and correction (EDAC) functions and improves error rate.

For single-event upset (SEU) mitigation, memory designers find it increasingly difficult to increase the density of storage elements in a memory devices using conventional 6T technology. Also, utilization of 10T and 12T storage elements provides little or no benefit in performance and decreases cell density. An alternative approach to increasing the number of storage elements is to use 6T elements in conjunction with a design methodology that relies upon error detection and correction, rather than upon error mitigation. The use of single-error correction and double-error detec-tion (SECDED) is an effective method to mitigate SEU in a design.

The UT8ER512K32 SRAM utilizes an internal error detection and correction scheme using a Hamming block code that can correct a single error, and detect a two-bit error, in any 32-bit word. The SRAM consists of 512k 32-bit words. There are also seven additional bits associated with each 32-bit word that are used for encoding. When a word is written to a memory location, the seven encoder bits are also written as defined by the block encoding algorithm. During normal operation, the address space is continually and sequentially "scrubbed" for errors. If a word contains a single bit-error, the error is corrected. If a word contains two bit-errors, the system is alerted upon reading the address at which the data is corrupted. This is discussed in detail in Section 2. The rate at which the address space is scrubbed for errors is called the scrub frequency, which is defined by the user. The implications of scrub frequency, both upon performance and device availability, are discussed in Section 3. Programming the scrub frequency is discussed in Section 4.

Figure 1 illustrates EDAC functionality upon the address space. The EDAC pointer contains the address of the word currently being evaluated for errors. When the scrub cycle is completed for that word, the pointer increments to the next address and a new scrub cycle begins. Again, the rate at which the address pointer increments is the scrub frequency. If an ionizing particle results in a single-event upset, the EDAC automatically corrects the error. Correction of the single bit- error is invisible to the rest of the system.

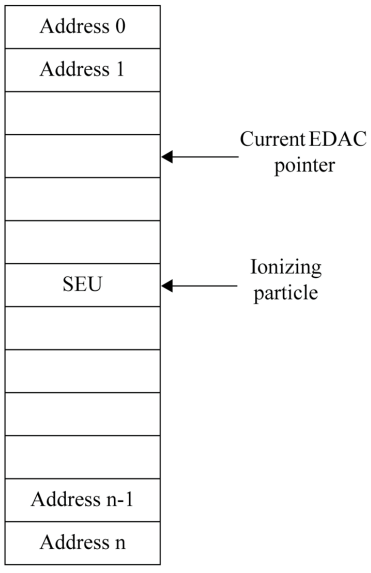

Figure 1. EDAC Pointer and Address Space

Designing with the UT8ER512K32 Monolithic 16M RadHard™ SRAM

2.0 Device Overview and Functionality

The UT8ER512K32 SRAM is available in two versions: a master device (UT8ER512K32M) and a slave device (UT8ER512K32S). The master contains a fully autonomous EDAC system and can be configured to control the scrub cycles of one of more slave devices. A slave device is dependent upon external control signals to initiate its scrub cycle. System configuration examples for the master and slave are detailed in Sections 2.2 and 2.3, respectively.

2.1. Additional EDAC Control Signals

Both master and slave utilize several non-traditional control signals for EDAC functionality. These control signals are shown in Table 2.

Table 1: Bus Signals Used with EDAC

Signal	Function in Master	Function in Slave
$\overline{\text{BUSY}}$	Output: driven high to alert system that a scrub cycle is about to begin.	No function
$\overline{\text{SCRUB}}$	Output: driven low during scrub cycle. Connected to one or more slave devices to initiate a scrub cycle.	Input: a low signal initiates the scrub cycle of the slave device.
MBE	Output: driven high during a read cycle if the 32-bit word contains two bit errors.	Output: driven high during a read cycle if the 32-bit word contains two bit errors.

The relationship between $\overline{\text{BUSY}}$ and $\overline{\text{SCRUB}}$ is shown in the timing diagram in Figure 2. $\overline{\text{BUSY}}$ is driven low as an alert to the system that a scrub cycle is about to begin. The delay between the assertion of $\overline{\text{SCRUB}}$ and the assertion of $\overline{\text{BUSY}}$ is user-programmable. The range is 0ns to 750ns. In the master, the assertion of $\overline{\text{SCRUB}}$ indicates that a scrub cycle is in process. Once the scrub cycle begins, it is important to not attempt access to the SRAM as the results will be unpredictable. $\overline{\text{BUSY}}$ can be used as an indicator to a host processor to complete a memory cycle, or it can be used as a wait-state generator.

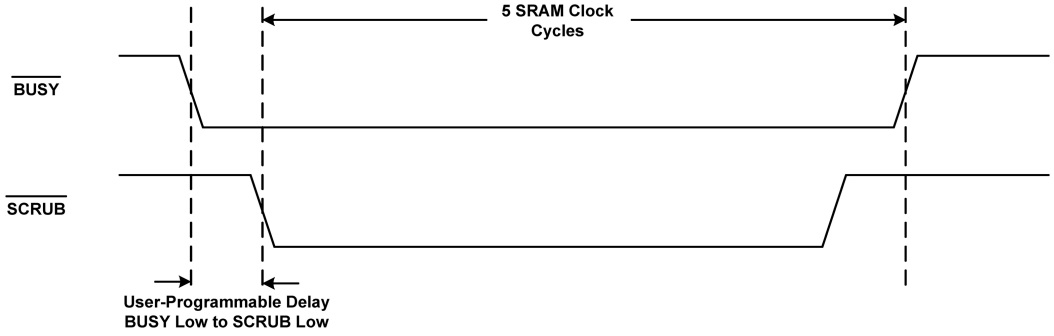

Figure 2. Bus Timing Diagram for Scrub Signals

In a master/slave configuration, $\overline{\text{BUSY}}$ and $\overline{\text{SCRUB}}$ are used as follows: $\overline{\text{BUSY}}$ is asserted as an alert to the system that a scrub cycle is about to begin. $\overline{\text{SCRUB}}$ is asserted by the master to make the slaves begin their scrub cycles. As shown in the Figure 2, the entire scrub process takes 5 internal SRAM clock cycles. A single clock cycle is between 50ns and 75ns; therefore, the maximum time for a scrub cycle is 375ns.

Designing with the UT8ER512K32 Monolithic 16M RadHard™ SRAM

2.2. Single-Device Operation with a Master

Figure 3 shows a partial schematic of a master device interfaced to a host. The control signals are representative of a SPARC processor. By default, masters operate in "auto-scrub". That is, the master controls scrub operation without the need for external control signals. $\overline{\text{BUSY}}$ is buffered through two flip-flops to provide signal stability. The output of the second flip flop is tied to the bus-ready ($\overline{\text{BRDY}}$) signal of the host processor in order to generate wait states. Figure 4 shows the bus timing diagram for $\overline{\text{BRDY}}$.

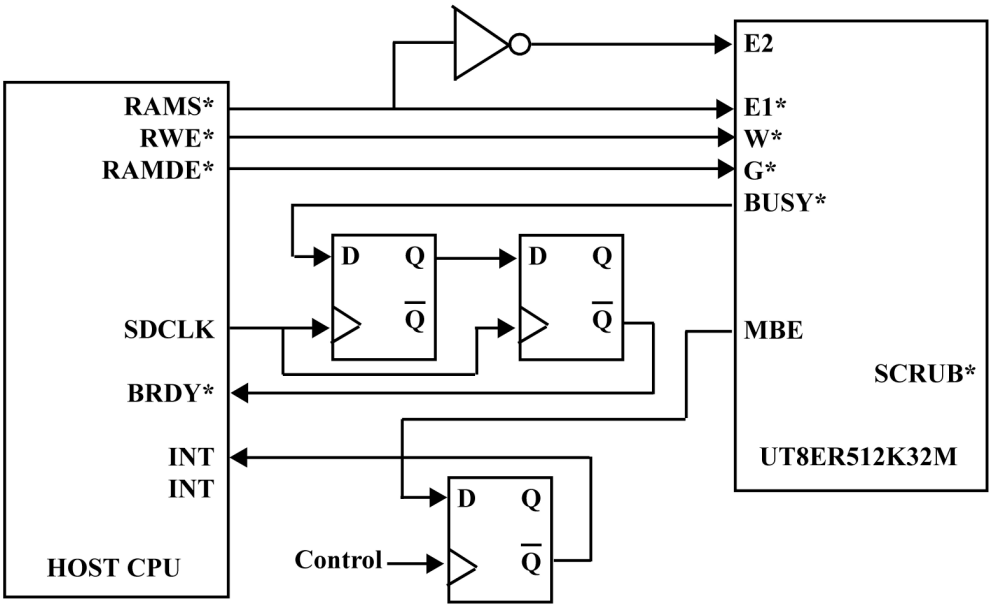

Figure 3. Master Device with Auto-Scrub and Wait-State Generation

Figure 5 shows the bus timing diagrams for the multiple-bit error (MBE) signal. In this configuration MBE is latched to a flip-flop using a control signal that could consist of the expression (!$\overline{\text{RWE}}$ * SDCLK * BRDY). That is, MBE is latched when data is being read, the bus is ready, and when the clock transitions from low to high. MBE could also be polled. Here it is shown being used to generate an interrupt.

Designing with the UT8ER512K32 Monolithic 16M RadHard™ SRAM

Figure 4 shows the bus signals during a typical read or write cycle that is interrupted by a scrub cycle. $\overline{BUSY}$ is asserted low and is sampled on the rising clock edge. Two cycles later $\overline{BRDY}$ is deasserted, which puts the host CPU into wait-state mode. After the scrub cycle completes, $\overline{BRDY}$ is asserted following two rising clock edges to signal the host CPU that the bus is available. Once $\overline{BRDY}$ is asserted, the data bus is either sampled or written on the next rising clock edge.

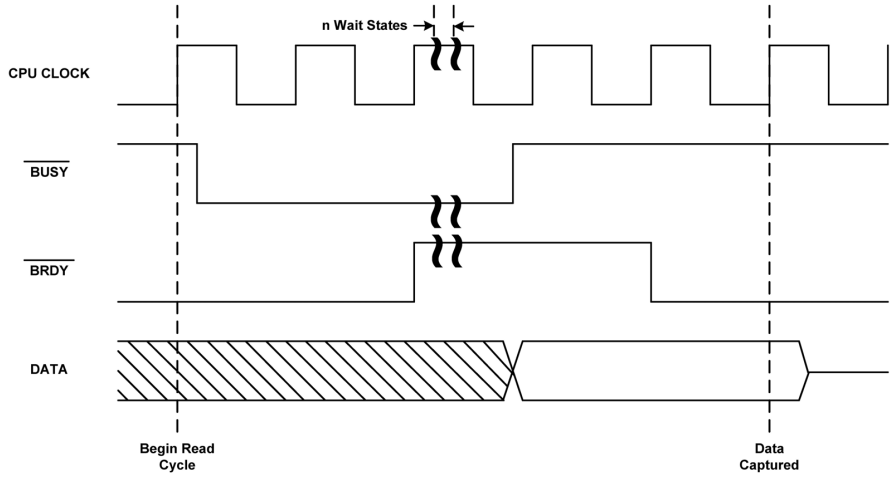

Figure 4. Typical Read or Write Timing Diagram

In this configuration, there is a delay of two clock cycles between $\overline{BUSY}$ asserted and $\overline{BRDY}$ asserted. Therefore, the $\overline{BUSY}$ to $\overline{SCRUB}$ delay should be set to no more than two clock cycles to ensure that the host enters wait-state mode before $\overline{SCRUB}$ is asserted.

CAES PIONEERING ADVANCED ELECTRONICS
4350 Centennial Blvd. | Colorado Springs CO 80907 USA | 1-800-645-8862 | cobhamaes.com

Designing with the UT8ER512K32 Monolithic 16M RadHard™ SRAM

Figure 5 shows the bus signals when MBE is asserted during a read cycle. MBE is asserted high when two bit-errors are detected during a read access. MBE is valid at the same time data is valid; therefore, it should be latched to the rising clock edge when the data bus is being read.

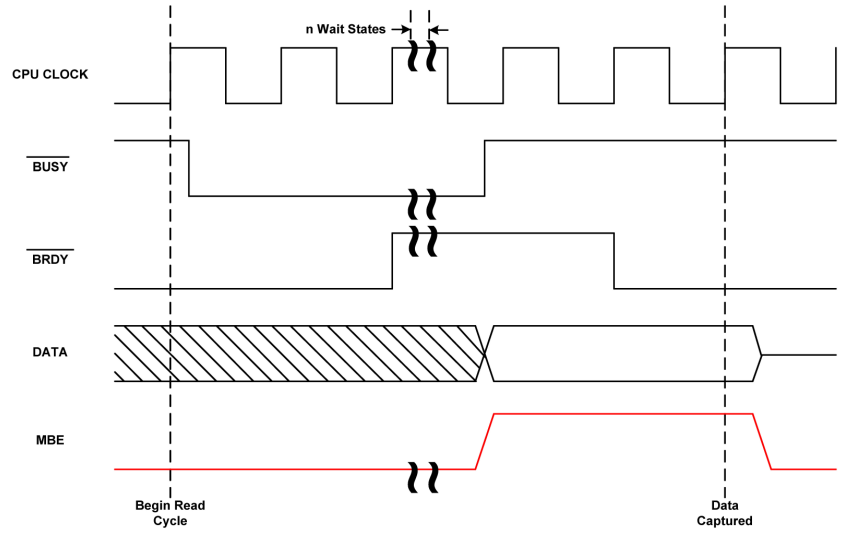

Figure 5. Read Cycle with MBE

Designing with the UT8ER512K32 Monolithic 16M RadHard™ SRAM

2.3. Multi-Device Operation in a Master/Slave Configuration

Figure 6 shows a master/slave configuration with auto-scrub. The master controls the scrub cycles of the slaves via the $\overline{\text{SCRUB}}$ signal. Refer to Figure 2 for the $\overline{\text{SCRUB}}$ timing diagram. When the master begins its scrub cycle, it first asserts $\overline{\text{BUSY}}$ as an indicator to the system that the scrub cycle is about to begin. The actual cycle begins after $\overline{\text{SCRUB}}$ is asserted. The time delay between $\overline{\text{BUSY}}$ asserted and $\overline{\text{SCRUB}}$ asserted is user-programmable and can range from 0ns to 750ns. $\overline{\text{SCRUB}}$ is asserted by the master long enough for the asynchronous slaves to complete their scrub cycles. Other than their $\overline{\text{SCRUB}}$ cycle dependency, slaves are interfaced in hardware the same as a master.

The MBE signals from each SRAM are independent; therefore, each MBE indicator should be read independently by the host CPU.

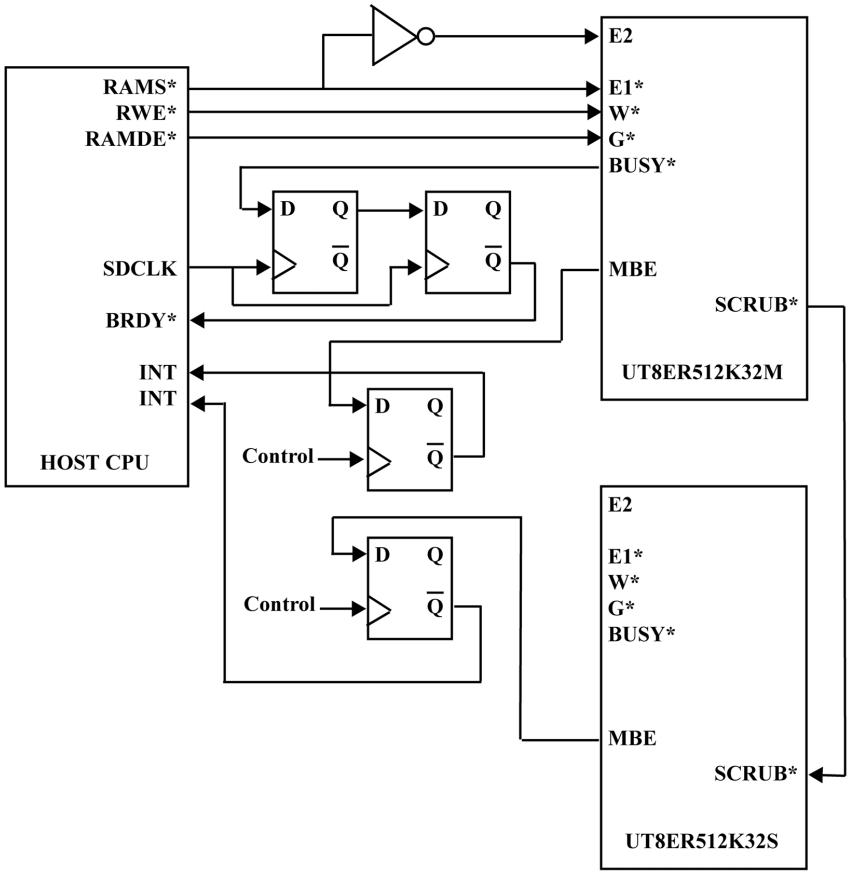

Figure 6. Master / Slave Configuration with Auto-Scrub

Designing with the UT8ER512K32 Monolithic 16M RadHard™ SRAM

2.4. Multi-Device Operation with Slaves

Figure 7 shows a configuration using only slave devices. The scrub configuration is "scrub-on-command", as slaves require a host to control their scrub cycles. A typical configuration might use a timer output from the host CPU to control the scrub cycle. Asserting the $\overline{\text{SCRUB}}$ input(s) low initiates a scrub cycle. It is important to hold $\overline{\text{SCRUB}}$ low for 375ns in order to ensure completion of the scrub cycle. Once $\overline{\text{SCRUB}}$ is deasserted, the internal EDAC pointer increments to the next address.

This configuration has several advantages: It allows the CPU to dynamically control the scrub rate without having to program the control registers each time the rate is changed. Also, it allows the user to create custom scrub algorithms for different modes of operation. For example, during a time of low access where memory retention is important, the scrub rate could be increased. Likewise, during periods of heavy memory access, the scrub rate could be lowered, or turned off, to ensure minimal intrusion on the memory cycles by the scrub cycles.

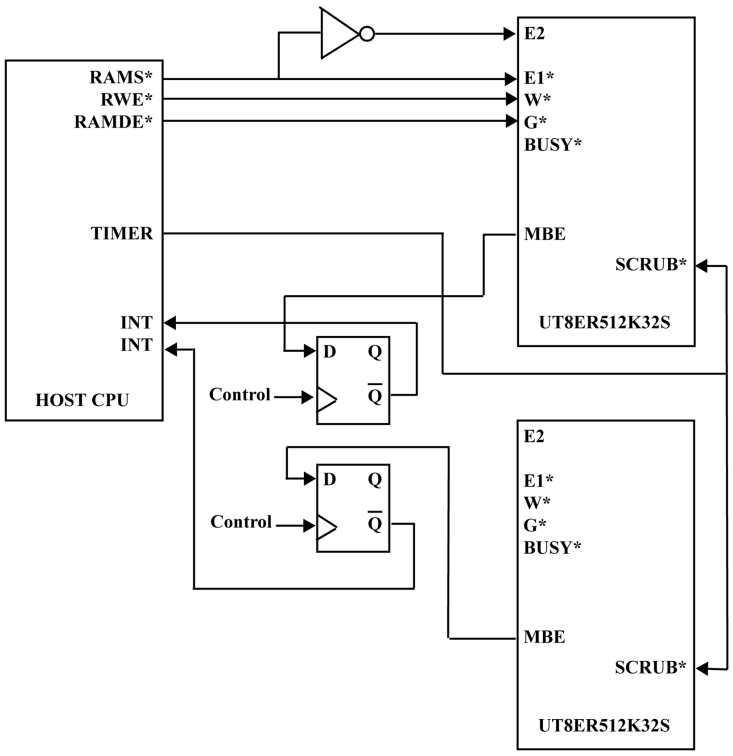

Figure 7. Slave Devices with Scrub-on-Command

Designing with the UT8ER512K32 Monolithic 16M RadHard™ SRAM

3.0 Error Detection and Correction (EDAC) for Reduced Error Rates

EDAC configuration affects both error rate and system throughput. As the scrub rate increases, the effective error rate decreases, while throughput decreases. The following sections describe how the scrub rate affects both of these.

3.1 Scrub Effects on Error Rate

The SEU error rate of the SRAM without EDAC functionality is 6.3×10^{-7} errors per bit-day. With the EDAC enabled, the error rate is a function of the scrub frequency as shown in Figure 8.

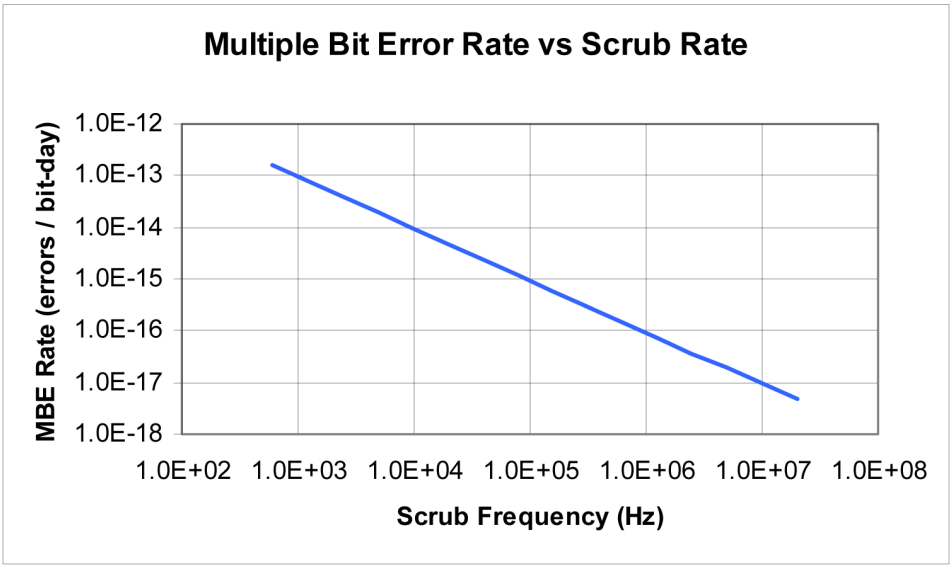

Figure 8. Bit-Error Rate vs Scrub Frequency

As the EDAC can correct a single-bit error and indicate a two-bit error, an error is defined as more than two corrupted bits at an address, the EDAC bits inclusive. At the default scrub frequency of 156 kHz, the error rate is 6.01×10^{-16} errors per bit-day in the Adams 90% worst-case geosynchronous environment.

CAES PIONEERING ADVANCED ELECTRONICS
4350 Centennial Blvd. | Colorado Springs CO 80907 USA | 1-800-645-8862 | cobhamaes.com

Designing with the UT8ER512K32 Monolithic 16M RadHard™ SRAM

3.2 Scrub Effects on Performance

As the scrub frequency increases, the available access time necessarily decreases. Therefore, there is a trade-off between error rate improvement and system throughput. Figure 9 shows the relationship between system throughput and the scrub frequency.

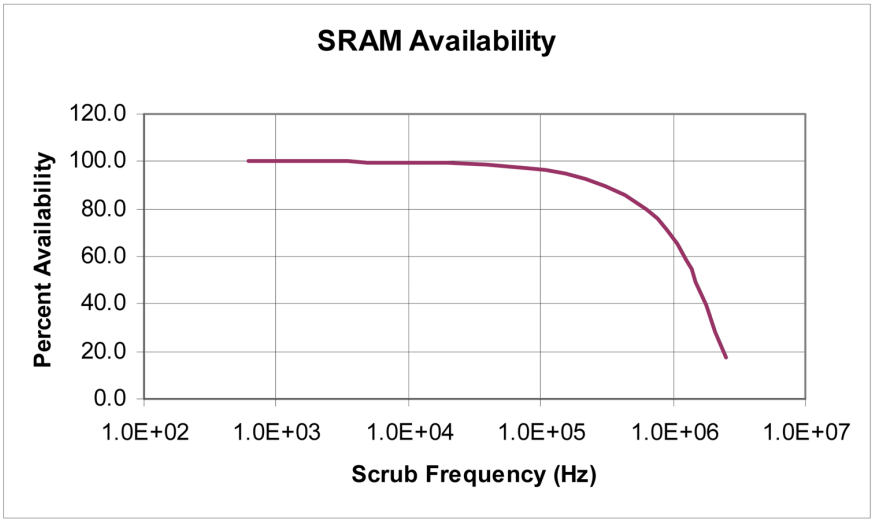

Figure 9. Percent Availability vs Scrub Frequency

System availability is 95% at the 156 kHz default scrub frequency. That is, the device will be unavailable for access 5% of the time assuming constant read or write cycles. Availability drops off to below 20% with a scrub frequency of 2.5 MHz. However, if data integrity is more important than data access, this could be advantageous in heavy ionization environments as the error rate improves to 3.75×10^{-17} errors per bit-day.

CAES PIONEERING ADVANCED ELECTRONICS
4350 Centennial Blvd. | Colorado Springs CO 80907 USA | 1-800-645-8862 | cobhamaes.com

Designing with the UT8ER512K32 Monolithic 16M RadHard™ SRAM

4.0 Configuration Options

Configuration options are selected by writing to the control register. Table 3 shows the parameters and available options. Section 4.3 explains how to program the control register.

Table 2: EDAC Configuration Table

Data Bit	Parameter	Min	Max	Function
0-3	Scrub Rate	0	15	0 = 20 MHz 6 = 312 kHz 11 = 9.76 kHz
				1 = 10 MHz 7 = 156 kHz 12 = 4.88 kHz
				2 = 5 MHz 8 = 78 kHz 13 = 2.44 kHz
				3 = 2.5 MHz 9 = 39 kHz 14 = 1.22 kHz
				4 = 1.25 MHz 10 = 19.5 kHz 15 = 0.61 kHz
				5 = 625 kHz
4-7	$\overline{\text{BUSY}}$ Low to $\overline{\text{SCRUB}}$ Low	0	15	0 = 0ns 6 = 300ns 11 = 550ns
				1 = 50ns 7 = 350ns 12 = 600ns
				2 = 100ns 8 = 400ns 13 = 650ns
				3 = 150ns 9 = 450ns 14 = 700ns
				4 = 200ns 10 = 500ns 15 = 750ns
				5 = 250ns
8	EDAC Bypass	0	1	0 = normal operation 1 = EDAC is bypassed
9	Read / Write Register Control	0	1	0 = bits 0-8 are written to control register 1 = bits 0-8 are asserted to the data bus

4.1. Scrub Rate Options

The scrub rate can be programmed between 610 Hz and 20 MHz. This is the rate at which the EDAC address pointer increments. The scrub rate is programmed via bits 0-3 in the control register. This option has no functionality in a slave.

4.2. Warning to Busy Timing Options

The delay between $\overline{\text{BUSY}}$ asserted to $\overline{\text{SCRUB}}$ asserted can be programmed to between 0ns and 750ns. In practice, the delay time should be long enough to ensure that the host processor can enter a wait-state mode, or to complete housekeeping tasks, before $\overline{\text{SCRUB}}$ is asserted. The delay is programmed via bits 4-7 in the control register. This option has no functionality in a slave.

4.3. EDAC Bypass

EDAC functionality can be enabled or disabled. The functionality of the device depends upon whether it is a master or a slave.

CAES PIONEERING ADVANCED ELECTRONICS
4350 Centennial Blvd. | Colorado Springs CO 80907 USA | 1-800-645-8862 | cobhamaes.com

Designing with the UT8ER512K32 Monolithic 16M RadHard™ SRAM

4.3.1. Master Device with EDAC Enabled

With EDAC enabled, memory is continually scrubbed for errors at the scrub rate. Furthermore, if the memory location being read contains a single-bit error, the EDAC automatically corrects the error during the read cycle even if the EDAC has not yet scrubbed that location,.

4.3.2. Master Device with EDAC Disabled

With EDAC bypassed, memory is not scrubbed for errors and data is not corrected during a read cycle.

4.3.3. Slave Device with EDAC Enabled

With EDAC enabled, a memory location containing a single-bit error is automatically corrected during the read cycle.

4.3.4. Slave Device with EDAC Disabled

With EDAC bypassed, data is not corrected during a read cycle.

4.4. Control Register Programming

The control register is accessed by writing a series of values to the address bus as shown in Figure 10. The contents of the control register are written following the fifth address. The contents of the address bus are written to the control register if bit 9 is zero. The contents of the control register are written to the data bus if bit 9 is one.

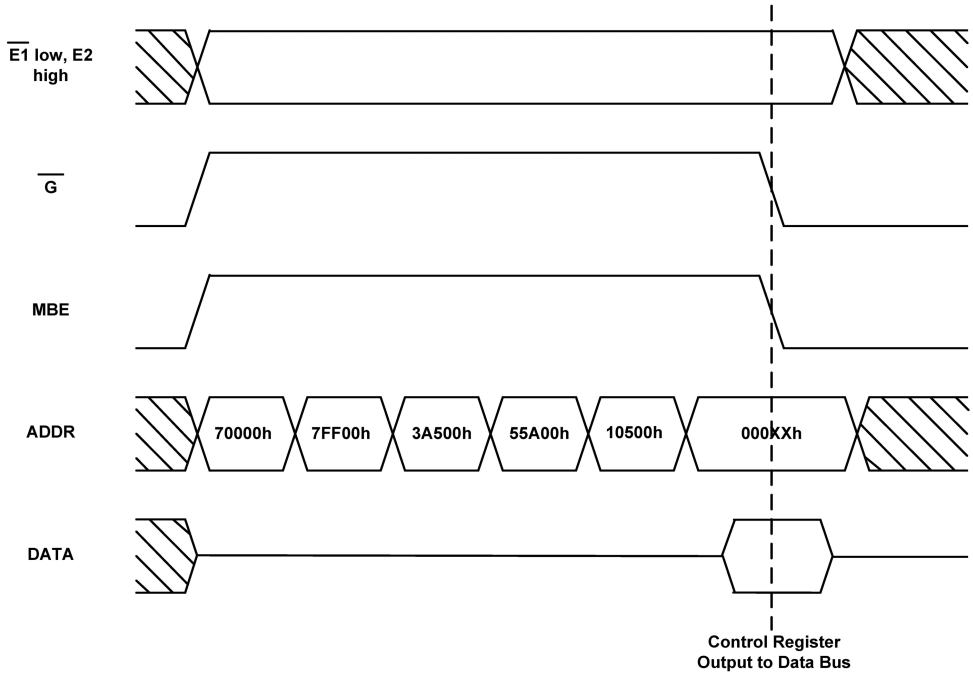

Figure 10. Accessing the Control Register