MSP430™ FRAM Quality and Reliability

Priya Thanigai, William Goh   *MSP430 Applications*

ABSTRACT

FRAM is a nonvolatile embedded memory technology and is known for its ability to be ultra-low power while being the most flexible and easy-to-use universal memory solution available today. This application report is intended to give new FRAM users and those migrating from flash-based applications knowledge on how FRAM meets key quality and reliability requirements such as data retention and endurance.

Contents

List of Figures

1   Introduction to FRAM

Ferroelectric Random Access Memory (FRAM) is an ultra-low power nonvolatile memory technology with write speeds similar to static RAM (SRAM). The technology has been in the industry for over a decade, implemented as stand-alone memory. FRAM's first introduction as an embedded memory in a general-purpose ultra-low-power MCU was on Texas Instruments 16-bit MSP430™ product line, the MSP430FR57xx family.

Some of the key attributes of FRAM are:

- FRAM is nonvolatile; that is, it retains its contents on loss of power.
- FRAM is a true random-access memory. The memory is not segmented; addressing of data for read or write at word or byte level happens directly in the same way as SRAM.
- The embedded FRAM on MSP430 devices can be accessed (read or write) at a maximum speed of 8 MHz. Above 8 MHz, wait states are used when accessing FRAM. Typical write speeds can exceed 2 MBps with FRAM compared to approximately 14 kBps on flash devices [1].
- Writing to FRAM and reading from FRAM require no setup or preparation such as pre-erase before write or unlocking of control registers.
- FRAM write accesses are extremely low power, because writing to FRAM does not require a charge pump.
- FRAM writes can be performed across the full voltage range of the device.
- FRAM meets and exceeds reliability requirements on data integrity. It provides practically unlimited endurance for read and write operations on the order of 10^{15} write or erase cycles.

MSP430 is a trademark of Texas Instruments.
All other trademarks are the property of their respective owners.

1.1 *FRAM Cell*

A single FRAM cell can be considered a dipole capacitor that consists of a film of ferroelectric material (ferroelectric crystal) between two electrode plates. This crystalline structure is made up of a PZT (lead-zirconium-titanate) complex (see Figure 1). Applying an electric field across the crystal causes the mobile atom to move from one stable state to another stable state in the crystal lattice. The stable state is retained after power is removed, which makes FRAM nonvolatile.
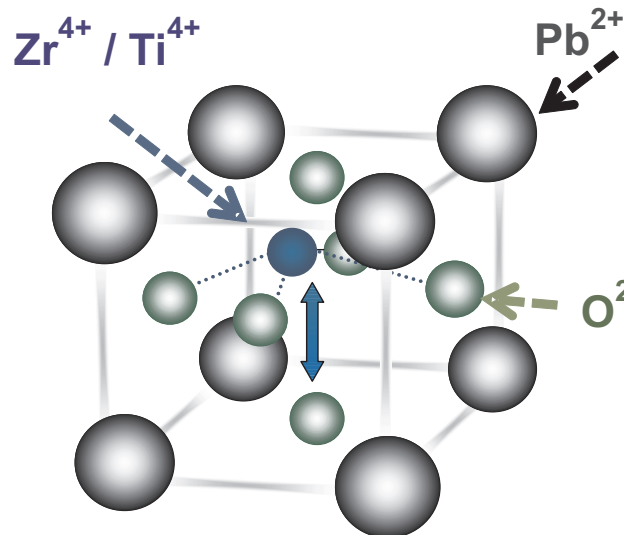

Figure 1. Structure of PZT Crystal

1.2 *Writing to FRAM*

Storing a 1 or 0 (writing to FRAM) requires polarizing the crystal in a specific direction using an electric field. This makes FRAM very fast, easy to write to, and capable of meeting high endurance requirements.

1.3 *Reading from FRAM*

Reading from FRAM requires applying an electric field across the crystal similar to a write. Depending on the state of the crystal, it may (or may not) need to be repolarized, thereby emitting a large or small induced charge. This charge is compared to a known reference to estimate the state of the crystal.

In the process of reading the data, the crystal that is polarized in the direction of the applied field loses its current state [2]. Therefore, every read is accompanied by a write-back to restore the state of the memory location. With TI's MSP430 FRAM MCUs, this is inherent to the FRAM implementation and is transparent to the application. The write-back mechanism is also protected from power loss and is ensured to complete safely under all power conditions. The MSP430FRxx power management system achieves this by isolating the FRAM power rails from the device supply rails in the event of a power loss. The FRAM power circuitry also uses a built-in low-dropout regulator (LDO) and a capacitor that store sufficient charge to complete the current write-back in the event of a power failure [2].

2   FRAM Characteristics

2.1   *Data Retention*

The primary goal of performing data retention tests is to ensure that FRAM can meet the retention specification for nonvolatile memory characterized by a ten year lifetime at 85°C. The following sections explain the main types of thermal wear with regard to FRAM and the testing procedure to check against these effects and to ensure lifetime data retention.

Two mechanisms must be considered when evaluating FRAM retention reliability [3]: thermal depolarization (see Section 2.1.1) and imprint (see Section 2.1.2).

2.1.1   Thermal Depolarization

The integrity of the information stored in an FRAM cell is directly proportional to the amount of polarization that the cell is capable of maintaining. When exposed to high temperatures, FRAM cells lose their ability to stay polarized and are unable to store information for as long as the high-temperature condition exists. In other words, thermal depolarization is referred to as a reduction of the spontaneous polarization that occurs as the ambient temperature of the ferroelectric material increases towards the Curie temperature. In the case of the specific composition of material used in the MSP430 FRAM MCU family, the Curie or transition temperature at which the FRAM cell is completely depolarized is approximately 430°C [3].
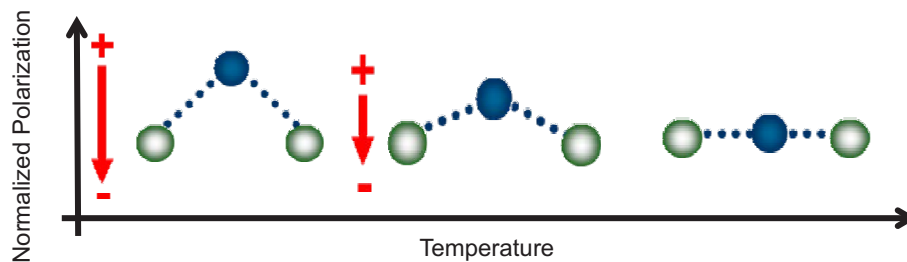

Figure 2. Reduction of Polarization With Temperature in an FRAM Cell

The effect of the depolarization is not permanent. Below the transition temperature, the material regains its ability to become polarized. Because every read requires a write-back, the material can be restored to its full polarization potential after the first read-write-back operation.

While the high transition temperature is not within the bounds of the normal operating temperature of the device, it is possible that the device is subjected to high temperatures during assembly processes such as hand or reflow soldering. Therefore, it is important to take precautions while subjecting the device to reflow as explained in Section 2.2. Also, it is important to note that within the device operating limits of -40°C to +85°C, depolarization is at an insignificant level and does not result in any data loss.

2.1.2   Imprint

Ferroelectric memories experience an effect in which data in one logic state can strengthen when the memory cell is exposed to high temperatures over long periods. This effect (the stabilization of polarization into a particular state) is known as imprint. However, imprinting also weakens the ability of the FRAM cell to store the complementary state data [3]. Unlike thermal depolarization, the effect of imprinting is permanent and is not lessened with a reduction in temperature. Note that while complete thermal depolarization occurs at the transition temperature of 430°C, imprinting occurs at lower temperatures, such as when data is written to FRAM and the material is subjected to a high-temperature bake for a long duration.

2.1.3 FRAM Data Retention Test Flow

To clarify the test procedure for data retention, the following two terms are defined:

- *Same-state* refers to the logic state of FRAM; that is, the state of polarization of the ferroelectric crystal prior to the high-temperature bake when testing for imprint.

- *Opposite-state* refers to the polarization of the crystal in a direction opposite to that in which it was imprinted.

Figure 3 shows the test procedure for data retention. To test for imprint, a data pattern with a set logic state is written onto FRAM, and then the device is exposed to a high-temperature bake at 125°C. This bakes the bit-cells in one logic state referred to as same-state. This bake is followed by a read-restore to further strengthen the same-state data. Following this, opposite-state data is written to FRAM. This is followed by a thermal depolarization bake to weaken/stress the opposite-state data. This bake is performed at the maximum operating temperature for the device (85°C for the MSP430). After the depolarization bake is completed, the data is read out to ensure the integrity of the opposite-state data.

The test is then repeated until the testing time is reached. The opposite-state bake identifies potential imprint issues if any while the read ensures that the FRAM cells have retained the ability to be polarized with opposite-state data as well preserved enough polarization to be read without data loss.

MSP430 FRAM data retention is tested for a cumulative bake time of 1000 hours at 125°C. See the device-specific data sheet for data retention ($t_{retention}$). Figure 4 is an example from the MSP430FR5739 data sheet.
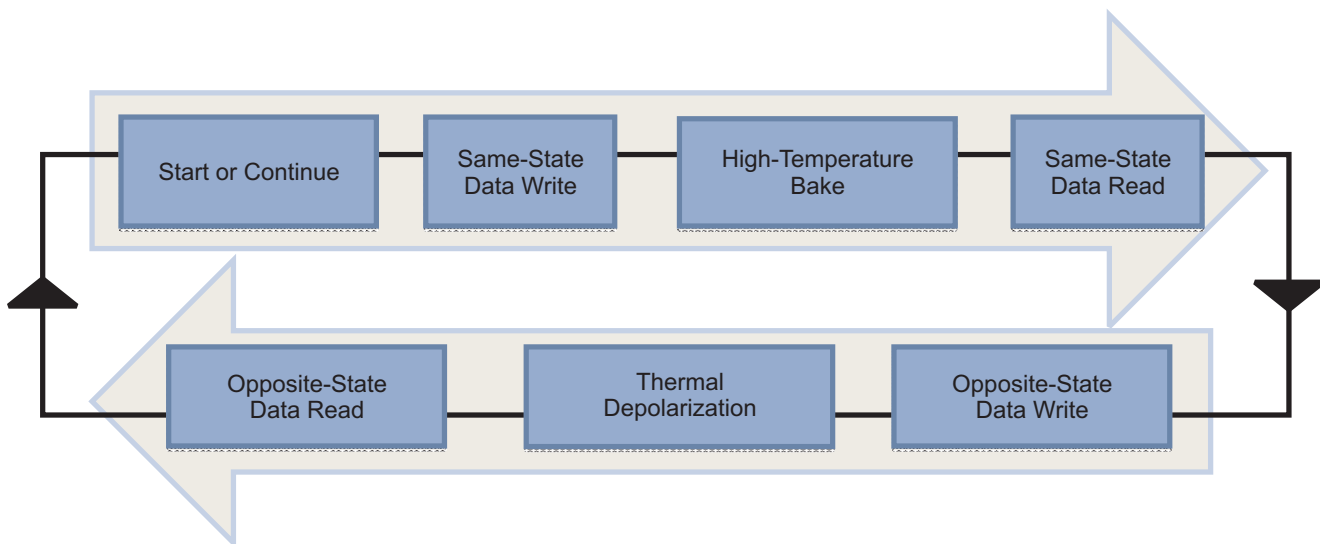
Figure 3. Test Flow for FRAM Data Retention

		$T_J = 25°C$	100	
$t_{Retention}$	Data retention duration	$T_J = 70°C$	40	years
		$T_J = 85°C$	10	

Figure 4. $t_{Retention}$ Example

2.2 Reflow and Soldering Recommendations

2.2.1 All MSP430 FRAM Except MSP430FR57xx

According to the data sheet, there is no limitation on programming the device prior to reflow. Higher temperatures can be applied during board soldering according to the current JEDEC J-STD-020 specification with peak reflow temperatures not higher than classified on the device label on the shipping boxes or reels.

2.2.2 MSP430FR57xx

> **NOTE:** The FR57xx family of devices must only be programmed after the reflow process has occurred.

According to the FR5739 device data sheet, data retention on FRAM cannot be ensured when the specified maximum storage temperature ($T_{stg} = 95°C$) is exceeded. During the soldering process, it is required to follow the current JEDEC J-STD-020 specification with peak reflow temperatures no higher than classified on the device label on the shipping boxes or reels.

Note that factory programmed information, such as calibration values, are designed to withstand the temperatures reached in the current JEDEC J-STD-020 specification.

2.3 Endurance

Flash memory is specified in terms of write endurance, which is defined as the measure of the number of erase and write cycles that a flash array can achieve while retaining data integrity. Per IEEE Standard Definitions and Characterization of Floating Gate Semiconductor Arrays [4], endurance is defined as "The measure of the ability of a nonvolatile memory device to meet its data-sheet specification as a function of accumulated nonvolatile data changes." [4]

Flash memory on MSP430 flash-based devices has a typical write endurance of 10^5 write or erase cycles [5]. In comparison, the endurance of a single FRAM cell in the MSP430FRxx family is 10^{15} write or erase cycles [6]. However, FRAM is inherently different from flash in that the total endurance is impacted by read cycles as well. As explained in Section 1.1, every read of an FRAM byte also necessitates a write-back and, therefore, there is no difference between read and write endurance on FRAM. Any access from FRAM, either a read or a write, has an impact on the endurance. On the other hand, the endurance of flash-based devices is affected only by a write access. The following section describes the impact of read and write cycles on lifetime endurance.

To understand the endurance specification on FRAM, it is important to know that while the flash limit is derived from experimental results that show a failure of flash memory beyond 10^5 write or erase cycles, there is no known fail limit in the case of FRAM. Therefore, it is possible to categorically state that FRAM is capable of exceeding 10^{15} write or erase cycles. The exact minimum endurance is not yet quantifiable, because testing to the fail limit on FRAM requires several years. As of this writing, no failure has been detected on FRAM units that have executed continuous read and write cycling; therefore, based on test data, Texas Instruments is confident in specifying the *minimum* endurance for MSP430 FRAM to be 10^{15} write or erase cycles [6].

2.3.1 Calculating the Impact of Read and Write Cycles on Endurance

Consider an example code as shown in Example 1. This code stresses the device by implementing the worst case for the maximum number read and write cycles and the impact of those cycles on the endurance limit. The formula derived using the worst-case test can be used for any application, and a lifetime in years can be calculated.

The example code shows a series of JMP instructions that direct code execution to specific locations. The code itself is seemingly useless in an application as it does not execute any instructions in between the JMP statements. So why was this specific test case used?

To understand the impact of read/write cycles on FRAM it is important to know that:

- Every read of FRAM includes a write-back, so FRAM is being written to every time it is accessed, regardless of whether the access is a read or a write.

- The number of instructions executed cannot be directly related to the number of FRAM accesses on the MSP430FRxx device. This is because the FRAM controller uses a four-word two-way associative cache to buffer instructions prior to execution. Therefore, instructions are fetched 64 bits at a time and executed from cache memory until a cache miss occurs. When there are no more instructions to execute in the cache and an FRAM access is required to execute the next instruction it termed as a cache miss. The access of the cache is transparent to the application and the filling of the cache occurs within the time taken for one FRAM access cycle. Therefore, we can see that in a typical use case for every 4-word (64-bit) instructions that are executed, only one access needs to be performed to FRAM.

However it is possible to stress the device by using code that causes a cache miss on every access. While this implementation is not practical in a real-world application, it is done by reaching outside the 64-bit boundary every time an instruction is executed (see Example 1). In between the JMP statements, nop instructions are used to pad the memory just enough so that the next JMP statement is outside the cache and requires an FRAM access. The nop instructions themselves are never executed and serve the sole purpose of manipulating the location of the JMP. Using assembler directives such as ".align 8", it is possible to align the first line of code to the start of cache, and this has been done to make the calculation easier.

Example 1. Code Example Forcing a 0% Cache Hit Ratio

```
Main
        nop
        nop
        nop
Label_A   jmp Label_B   ; cache miss
        nop
        nop
        nop
        nop
        nop
        nop
Label_B   jmp Label_C   ; cache miss
        nop
        nop
        nop
        nop
        nop
        nop
Label_C   jmp Label_A   ; cache miss
        nop
        nop
        nop
```

Figure 5 shows the frequency of FRAM accesses. An FRAM access occurs every three MCLK cycles, because one cycle is used to fetch from FRAM and two MCLK cycles are required to execute the JMP instruction. Because the code uses the JMP to transition between three labels represented by A, B, and C, the fetch at each address has an impact on the endurance of that location.
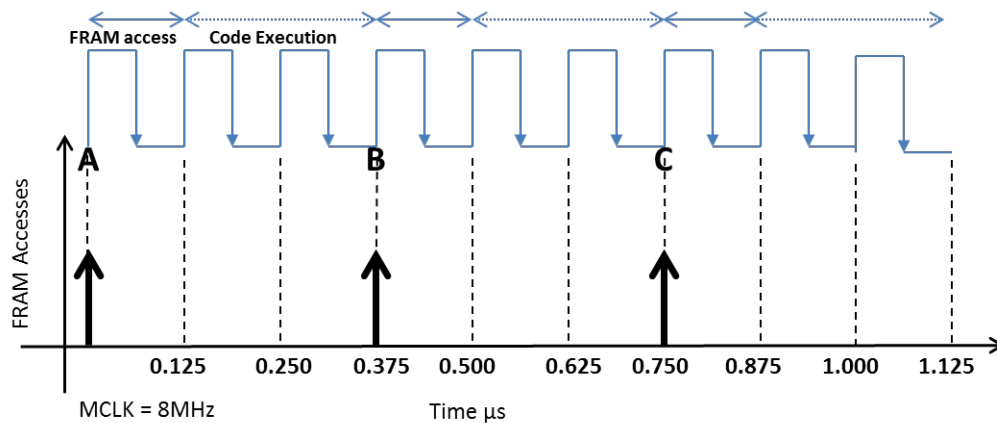
Figure 5. FRAM Accesses Performed Over Time

For example, to calculate the worst-case endurance for location A:

Location A is accessed once every 1.125 µs or (0.888 × 10^6) times per second.

The total number of accesses allowed at location A (as specified in the data sheet) is 10^{15}.

Time to reach the minimum endurance limit = 10^{15} / (0.88 × 10^6) seconds, which is approximately 36 years.

The same is true for locations B and C.

Now consider adding a one-word instruction to every label. This doubles the lifetime of each location to 72 years.

What about an instruction that makes continuous calls to a single location such as a `JMP$` `(while(1))` in C programming? This instruction has no impact on endurance, because the instruction is executed only out of the cache and no cache misses (that is, FRAM accesses) occur. This is termed a 100% cache hit.

Any real-world application is not likely to use code that simply jumps between labels continuously with no code execution in between. In most cases, the application code tests flags, branches to functions, uses interrupt service routines, and so on. Therefore, we can easily see that the endurance for any single FRAM location significantly exceeds the product lifetime.

2.4 Soft Error Rate (SER)

Soft error rate (SER) is the rate at which a device or system encounters or is predicted to encounter soft errors. It is typically expressed as either number of failures-in-time (FIT) or the mean time between failures (MTBF). The unit adopted for quantifying failures in time is called FIT, equivalent to one error per billion hours of device operation. MTBF is usually given in years of device operation. Soft errors in nonvolatile memory are caused by neutron events. These events may produce alpha particles and gamma rays as a side effect, causing a soft error when in close proximity to a circuit node [7].

FRAM uses crystal polarization instead of charge to store data. Therefore, it is expected that an FRAM bit cell would not experience the same data loss as an SRAM bit during a neutron event. Two kinds of experiments were conducted to test the SER for FRAM:

* Static mode, in which an unpowered device is subjected to radiation and then the pre-written memory pattern is tested.
* Dynamic mode, in which a powered device performing continuous reads is tested for bit failure.

The static mode test results yielded a FIT rate of <0.051/Mb and the dynamic mode yielded an SER <0.16/Mb [3]. The tests indicate that the SER for FRAM is orders of magnitude lower than SRAM.

3 Summary

This document provides a comprehensive list of production considerations for the MSP430FRxx family of devices. The document discusses FRAM qualification results and tests in detail. For device-specific guidelines on ESD, see the device data sheet. For general good practice for ESD-safe design, see the application report *MSP430 System-Level ESD Considerations* ([SLAA530](#)).

4 References

1. *Maximizing FRAM Write Speed on the MSP430FR573x* ([SLAA498](#))
2. Michael Zwerg, Adolf Baumann, et al., *An 82 µ A/MHz Microcontroller with Embedded FeRAM for Energy-Harvesting Applications* (http://ieeexplore.ieee.org)
3. J. Rodriguez, K. Remack, J. Gertas, L. Wang, C. Zhou, K. Boku, J. Rodriguez-Latorre, K. R. Udayakumar, S. Summerfelt, T. Moise, D. Kim, J. Groat, J. Eliason, M. Depner, F. Chu, *Reliability of Ferroelectric Random Access Memory Embedded within 130nm CMOS* (http://ieeexplore.ieee.org)
4. Peter Forstner, *Flash Memory Characteristics* ([SLAA334](#))
5. MSP430F2274 device data sheet ([SLAS504](#))
6. MSP430FR5739 device data sheet ([SLAS639](#))
7. Wikipedia results for "Soft Error Rate" (http://www.wikipedia.com)
8. *MSP430 System-Level ESD Considerations* ([SLAA530](#))