

ECEN 5823-001 / -001B

Internet of Things Embedded Firmware

Lecture #27

05 December 2017

Agenda

- Class announcements
- Google Sign up sheet
- Final Exam
- Final Report
- Memory for the IoT and IIoT markets

Class Announcements

- Course Project Update #2 is due at 11:59pm this Wednesday the 6th
- Any questions regarding the Course Project?
- Any questions regarding Bluetooth Mesh?
- Targeting Course Project and Final Report Rubric on Thursday the 7th

Course Project Demo Sign up

- Here is the link to sign up for the course project demos. If you are a distant learning team and there are no appropriate times, we can target Friday the 15th or some other time. Another alternative for distant students is a video conference call.

https://docs.google.com/a/colorado.edu/spreadsheets/d/19pyGBCRNCU37iTsneXGnOyGnfPwv31wxGi_yn6B3nuc/edit?usp=sharing

Final Exam

- When: Sunday the 17th at 1:30 to 4:00pm
- Where: ECCR 1B51
- Similar format to Mid-Term, D2L base
- Tentatively planning:
 - 40 questions (100% of Final Exam)
 - 2 hours for exam
 - Covering material since Mid-Term plus BLE
 - Any topic related to programming assignments plus course project
 - 5 bonus questions (up to +5% of Final Exam)
 - 5 minutes
 - Past quiz questions from quiz 7, 8, 9, and 10
- Distant Students can take the Final from 1:30pm on Sunday the 17th through 11:59pm on the 17th

Final Report

ECEN 5823 Project Update #2 Assignment Fall 2017

Objective: To update the status and provide additional information on the Course Project in ECEN 5823, Fall 2017.

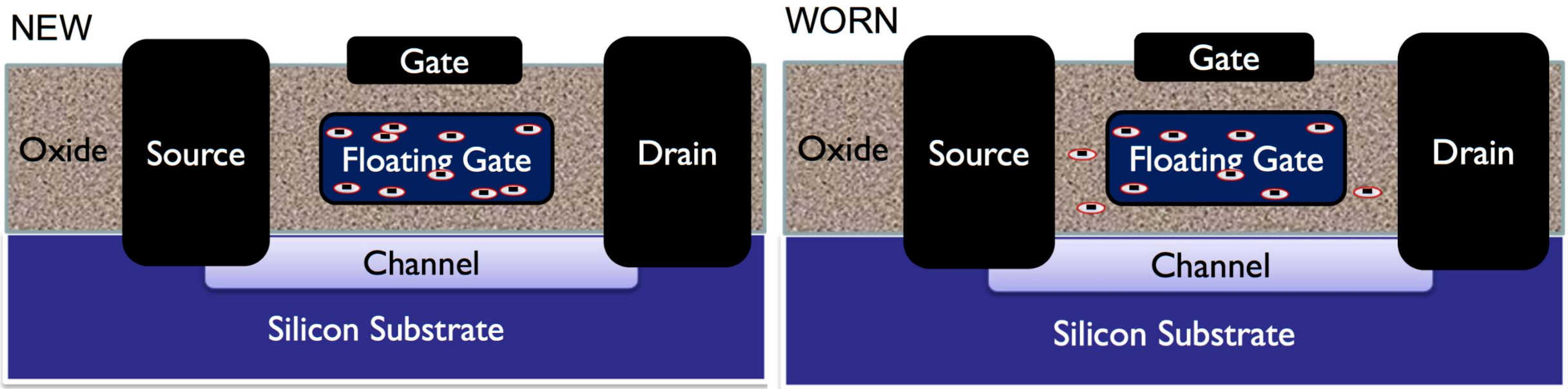
Note: You can use your course project proposal as a base for this project update.

Project Proposal | Due Date: Wednesday, December 6th, at 11:59pm via D2L drop box

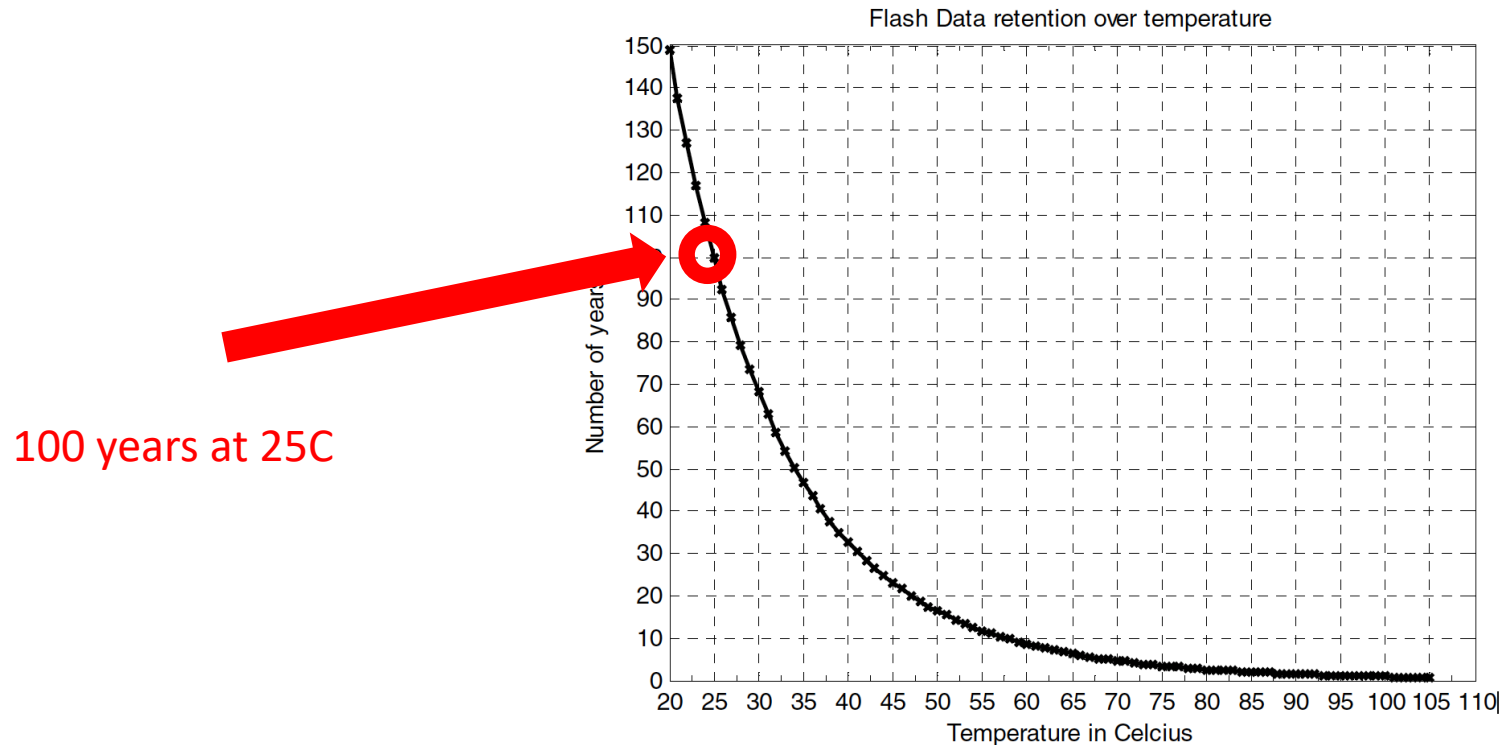
Team proposals: (Include and Provide update to the below items)

1. Describe what problem this project addresses
2. How does this project alleviate or solve the problem?
3. Functional block diagram of the team project
4. Summary of each individual project and how it plays a role in solving the problem
5. Project team members
6. Team project validation plan

Flash memory wear-out: Electrons trapped in the tunneling oxide preventing a reliable read of a “0” or “1.”



Data Retention versus Temperature



Data Retention errors increase as the leakage current increase electron migration with temperature. It can only happen to a floating gate with a positive charge, "1." The value of a "0" can become a "1," and never a "1" to a "0"

Figure 1: Flash Data Retention vs Temperature for 170°C 420-Hour Test

The corner cases for 85°C and 105°C are slightly over 2 years and less than 9 months,

Data Retention vs Temperature dependency

- With higher temperatures, leakage current increases and, thus, the charge on the floating gate is reduced more quickly than at lower temperatures. This temperatures dependence follows the Arrhenius equation

$$AF = e^{-\frac{E_a}{k} \left(\frac{1}{T_1} - \frac{1}{T_2} \right)}$$

Where

AF = Acceleration factor

Ea = 0.6 eV = Activation energy

k = 86.17×10^{-6} = Speed constant

T1 = Temperature 1 (K)

T2 = Temperature 2 (K)

Example of data retention versus temperature

- At T_2 25C, the data sheet specifies data retention at 100 years
- At T_1 50C, what is the estimated data retention in years?
- 100 yrs / AF
- $100 \text{ yrs} / e^{-\left(\frac{0.6 \text{ eV}}{86.17 \times 10^{-6}}\right)\left(\frac{1}{T_1} - \frac{1}{T_2}\right)}$
- $100 \text{ yrs} / e^{-\left(\frac{0.6 \text{ eV}}{86.17 \times 10^{-6}}\right)\left(\frac{1}{323} - \frac{1}{298}\right)}$
- $100 \text{ yrs} / e^{1.8085}$
- 100 yrs / 6.101
- 16.39 yrs

NXP LPC15xx data sheet

Table 13. Flash characteristics

$T_{amb} = -40\text{ }^{\circ}\text{C}$ to $+105\text{ }^{\circ}\text{C}$. Based on JEDEC NVM qualification. Failure rate $< 10\text{ ppm}$ for parts as specified below.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
N_{endu}	endurance	[1]	10000	100000	-	cycles
t_{ret}	retention time	powered	10	20	-	years
		not powered	20	40	-	years
t_{er}	erase time	page or multiple consecutive pages, sector or multiple consecutive sectors	95	100	105	ms
t_{prog}	programming time	[2]	0.95	1	1.05	ms

[1] Number of program/erase cycles.

[2] Programming times are given for writing 256 bytes to the flash. $T_{amb} \leq +85\text{ }^{\circ}\text{C}$. Flash programming with IAP calls (see *LPC15xx user manual*).

Data Retention of the NXP LPC15XX at 125C operating?

- At T_2 105C (378K), the data sheet specifies data retention at 10 years
- At T_1 125C (398K), what is the estimated data retention in years?
- 10 yrs / AF
- $10 \text{ yrs} / e^{-\left(\frac{0.6 \text{ eV}}{86.17 \times 10^{-6}}\right)\left(\frac{1}{T_1} - \frac{1}{T_2}\right)}$
- $10 \text{ yrs} / e^{-\left(\frac{0.6 \text{ eV}}{86.17 \times 10^{-6}}\right)\left(\frac{1}{398} - \frac{1}{378}\right)}$
- $10 \text{ yrs} / e^{0.9257}$
- 10 yrs / 2.524
- 3.96 yrs

How to get desired data retention times for systems in elevated temperature with long life?

- Data Retention time is based on when the cell is written, so rewriting the cell before the Data Retention time becomes an issue will “restart” the Data Retention clock.
Since the failure mechanism is a “1” being read as an erroneous “1”, the cell must be erased and then programmed!
- Is this a good solution? Maybe.
 - Can you guarantee or design that over the life span of the product that the number of Erase/Program cycles specified will not be surpassed
 - Erasing a flash cell is a relatively a high current operation
 - Can the system provide the current required?
 - Does the battery have enough charge to meet the battery life cycle requirements?
 - Writing to the flash takes a relatively long time, and the processor is limited capabilities / resources during the flash erase, programming, and refreshes

Example of the high current to program flash

- Silicon Labs' EFM32LG Leopard Gecko
- Typical current consumption without flash programming @ 14MHz HFRCO
 - EM0 = 3.02mA
 - EM1 = 1.11mA
 - EM2 = 0.0017mA
 - EM3 = 0.0013mA
 - Em4 = 0.0009mA

Table 3.7. Flash

Symbol	Parameter	Condition	Min	Typ	Max	Unit
EC _{FLASH}	Flash erase cycles before failure		20000			cycles
RET _{FLASH}	Flash data retention	T _{AMB} <150°C	10000			h
		T _{AMB} <85°C	10			years
		T _{AMB} <70°C	20			years
t _{W_PROG}	Word (32-bit) programming time		20			μs
t _{PERASE}	Page erase time		20	20.4	20.8	ms
t _{DERASE}	Device erase time		40	40.8	41.6	ms
I _{ERASE}	Erase current				7 ¹	mA
I _{WRITE}	Write current				7 ¹	mA
V _{FLASH}	Supply voltage during flash erase and write		1.98		3.8	V

¹ Measured at 25°C

High current of programming flash memory

- Possible solutions are:
 - Program or refresh FLASH only when connected to an external power source
 - Increase the current capability of the battery
 - Increase the charge or capacity of the battery
 - Manage power of the system
 - Example:
 - BLE Radio can consume 15mA during transmit
 - Writing to the flash consumes 7mA
 - Processor in EMO while writing to the FLASH is 3mA
 - Total current during these operations combined is 25mA
 - Only perform writing to the flash when the radio is turned OFF

Limiting peak current
during FLASH
programming or refresh
to 10mA

High current of programming flash memory

- Silicon Labs' Leopard Gecko EFM32LG32 write to flash example
- Worst case is a page erase plus a full page write
- Page size is 2048 bytes or 512 4-byte words
 - Erase page 20.8ms
 - Write 512*20uS 1.0ms
 - Total time of **21.8ms**
- To minimize the current spike on the battery for a BLE application that programmed while operating, the **ConnInterval** or **SlaveInterval** should guarantee at least 21.8ms with the radio off to program the flash

Table 3.7. Flash

Symbol	Parameter	Condition	Min	Typ	Max	Unit
EC _{FLASH}	Flash erase cycles before failure		20000			cycles
RET _{FLASH}	Flash data retention	T _{AMB} <150°C	10000			h
		T _{AMB} <85°C	10			years
		T _{AMB} <70°C	20			years
t _{W_PROG}	Word (32-bit) programming time		20			μs
t _{PERASE}	Page erase time		20	20.4	20.8	ms
t _{DERASE}	Device erase time		40	40.8	41.6	ms
I _{ERASE}	Erase current				7 ¹	mA
I _{WRITE}	Write current				7 ¹	mA
V _{FLASH}	Supply voltage during flash erase and write		1.98		3.8	V

¹ Measured at 25°C

Limitations due to the number of erase cycles before failure

- Silicon Labs' EFM32LG Leopard Gecko
- If a page of the flash had to be updated 1 time per hour based for sensor readings
 - $24 \times 365 = 8,760$ /yr
 - Requires a "fresh" flash sector **2.25 years**
- Or, a page of flash had to be updated every 1 minute
 - $60 \times 24 \times 365 = 525,600$
 - Requiring a "fresh" flash sector every **13.89 days**

Table 3.7. Flash

Symbol	Parameter	Condition	Min	Typ	Max	Unit
EC _{FLASH}	Flash erase cycles before failure		20000			cycles
RET _{FLASH}	Flash data retention	T _{AMB} <150°C	10000			h
		T _{AMB} <85°C	10			years
		T _{AMB} <70°C	20			years
t _{W_PROG}	Word (32-bit) programming time		20			μs
t _{PERASE}	Page erase time		20	20.4	20.8	ms
t _{DERASE}	Device erase time		40	40.8	41.6	ms
I _{ERASE}	Erase current				7 ¹	mA
I _{WRITE}	Write current				7 ¹	mA
V _{FLASH}	Supply voltage during flash erase and write		1.98		3.8	V

¹ Measured at 25°C

Limitations due to the number of erase cycles before failure

- Possible solutions
 - Allocate enough flash to insure enough good “flash” pages for the life of the product
 - Example:
 - The case of a flash page is updated every hour
 - The product is an industrial application with a lifecycle projection of 20 years
 - Number of pages based on the previous example = $20 \text{ yrs} / 2.25 \text{ yrs/page}$
 - ~ 8.89 pages
 - Total amount of flash dedicated for this storage = $2,048 \text{ bytes/page} * 8.89 \text{ pages}$
 - 18,207 bytes
 - **Solution:** Purchase a microcontroller that had an additional 9 pages of flash available to allocate for data logging
 - **Solution:** Utilize an external Flash

Limitations due to the number of erase cycles before failure

- Possible solutions

- Allocate enough flash to insure enough good “flash” pages for the life of the product
- Example:
 - The case of a flash page is updated every minute
 - The product is an industrial application with a lifecycle projection of 20 years
 - Number of days over product life cycle = $20 * 365 = 7240$ days
 - Number of pages based on the previous example = $7240 / 13.89 \sim 8,864$ pages
 - Total amount of flash dedicated for this storage = $2,048 \text{ bytes/page} * 8,864 \text{ pages}$
 - 18,153,472 bytes
 - **Solution:** ~~Purchase a microcontroller that had 18MB of additional memory~~
 - **Solution:** External Flash or change product specifications

Writing to the flash takes a long time

- Silicon Labs' Leopard Gecko EFMLG32 write to flash example
- During writes to flash in the Leopard Gecko, no access to flash memory is allowed
 - Not even instructions to execute
 - It will stall the processor
- Must plan the write to flash when access to flash will not be required
- Page size is 2048 bytes or 512 4-byte words
 - Erase page 20.8ms
 - Write 512*20uS 1.0ms
 - Total time of **21.8ms**
- Must plan writes when no time critical interrupts can occur
 - For BLE operations, it should be planned between Connection Events so the ConnInterval or ServerLatency should be greater than the time to write to flash

Table 3.7. Flash

Symbol	Parameter	Condition	Min	Typ	Max	Unit
EC _{FLASH}	Flash erase cycles before failure		20000			cycles
RET _{FLASH}	Flash data retention	T _{AMB} <150°C	10000			h
		T _{AMB} <85°C	10			years
		T _{AMB} <70°C	20			years
t _{W_PROG}	Word (32-bit) programming time		20			μs
t _{PERASE}	Page erase time		20	20.4	20.8	ms
t _{DERASE}	Device erase time		40	40.8	41.6	ms
I _{ERASE}	Erase current				7 ¹	mA
I _{WRITE}	Write current				7 ¹	mA
V _{FLASH}	Supply voltage during flash erase and write		1.98		3.8	V

¹ Measured at 25°C

Correction to TI application note SLAA334

- Let's take a look at the TI SLAA334 statement

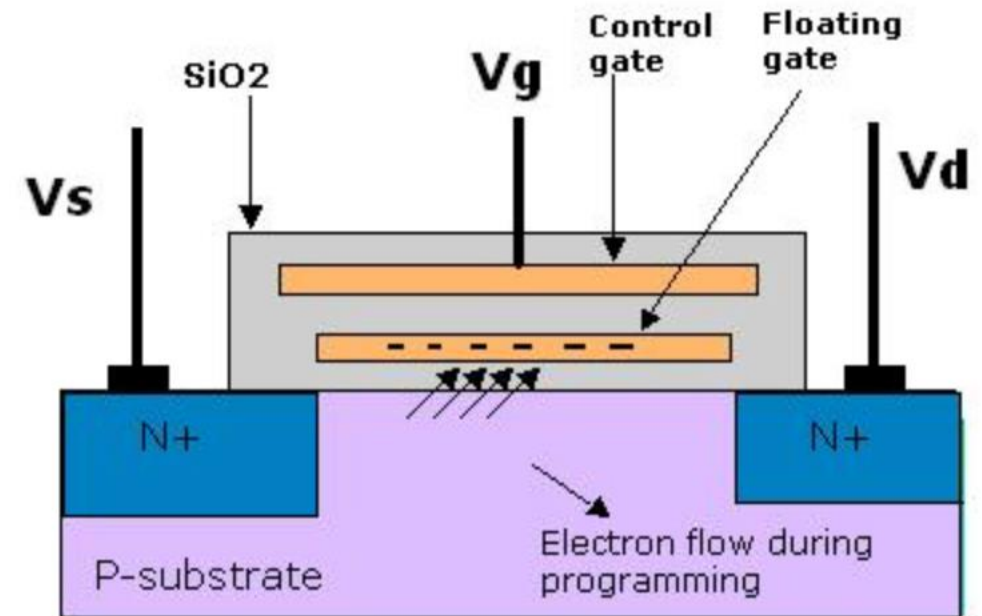
3.1 *Data Retention*

3.1.1 Leakage Mechanism

Data retention is limited by leakage current through the insulating oxide. Leakage can only occur if the floating gate is fully charged. Therefore, leakage only can flip an erased cell with the logic level 1 to a programmed cell with the logic level 0. According to Manabe [1], there are several phenomena that cause leakage.

Correction to TI application note SLAA334

- Let's think of a model of a NAND memory cell
- A "1" occurs when current flows from the source to the drain
- A "0" occurs when the free electrons in the substrate are moved and trapped in the floating gate, thus current cannot flow from the source to drain



Data Retention

In flash storage, data retention is the measure of how long the integrity of data can be guaranteed after being written to the flash drive without suffering from data corruption. Once a flash cell is charged, the electrons stored in the cell leak across the NAND gate over time, causing the charge on the cell to decrease. With enough leakage, the voltage level on the cell will drift into the neighboring region, causing the incorrect binary value to be read.

Because SLC flash memory is only divided into two voltage regions, it has more margin for charge loss before a bit flip occurs (a 0 becomes a 1), as shown in Figure 2.

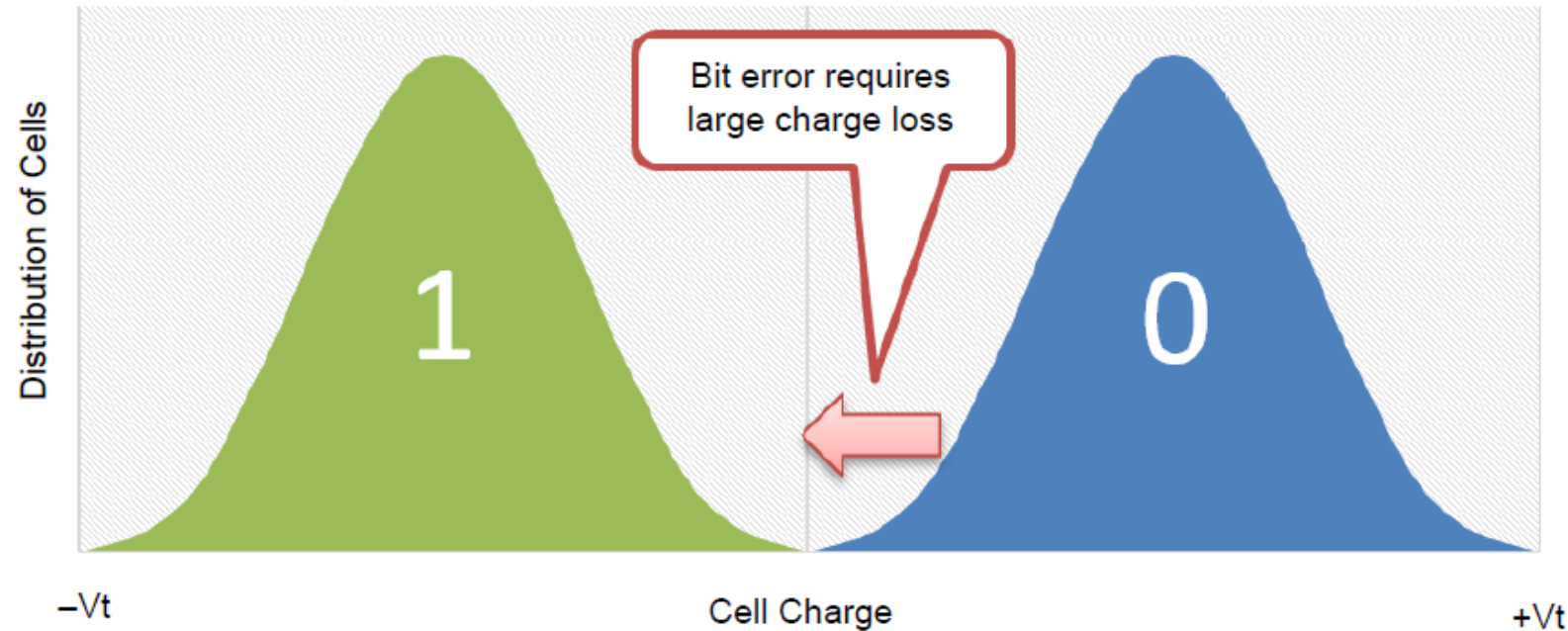


Figure 2 SLC Flash Data Storage

On the other hand, MLC can tolerate much less charge loss before data errors occur because it has a similar voltage range divided into four regions, as shown in Figure 3.

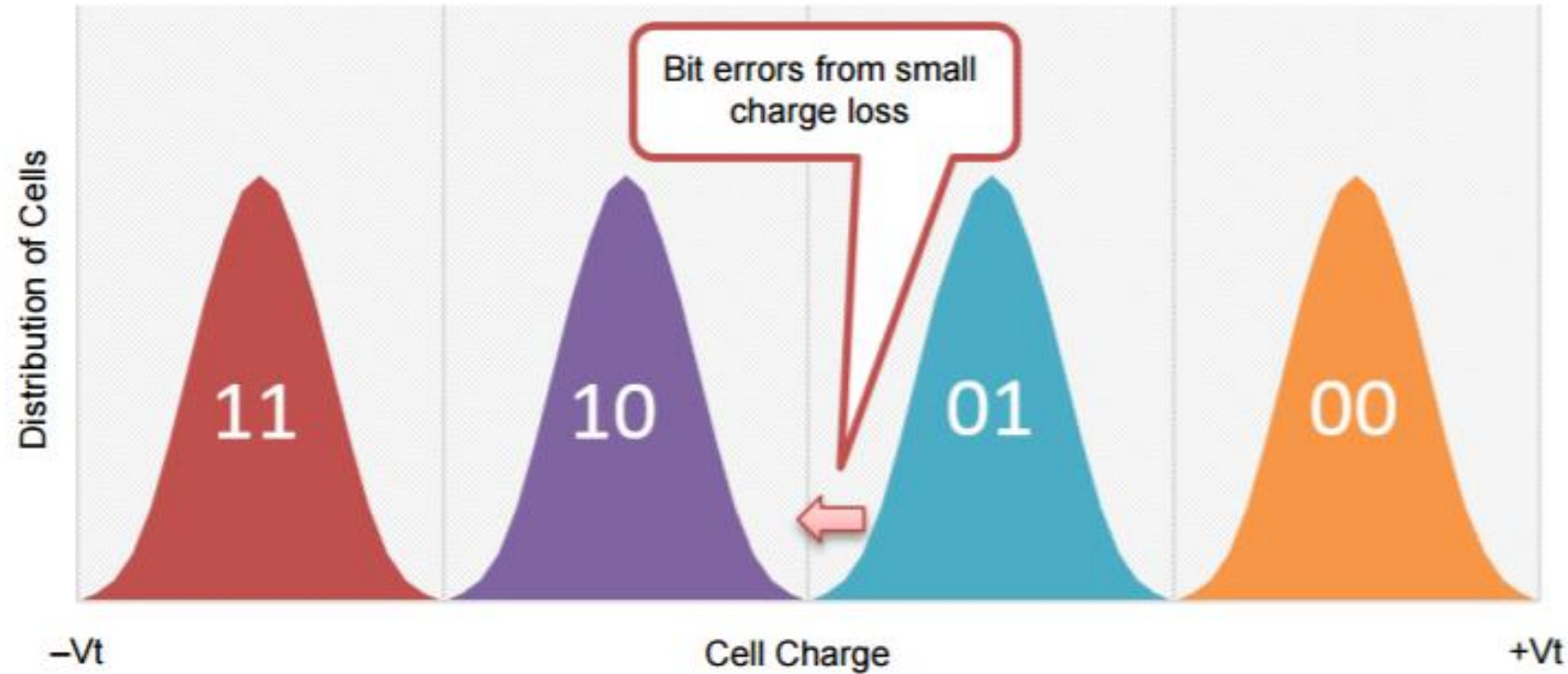
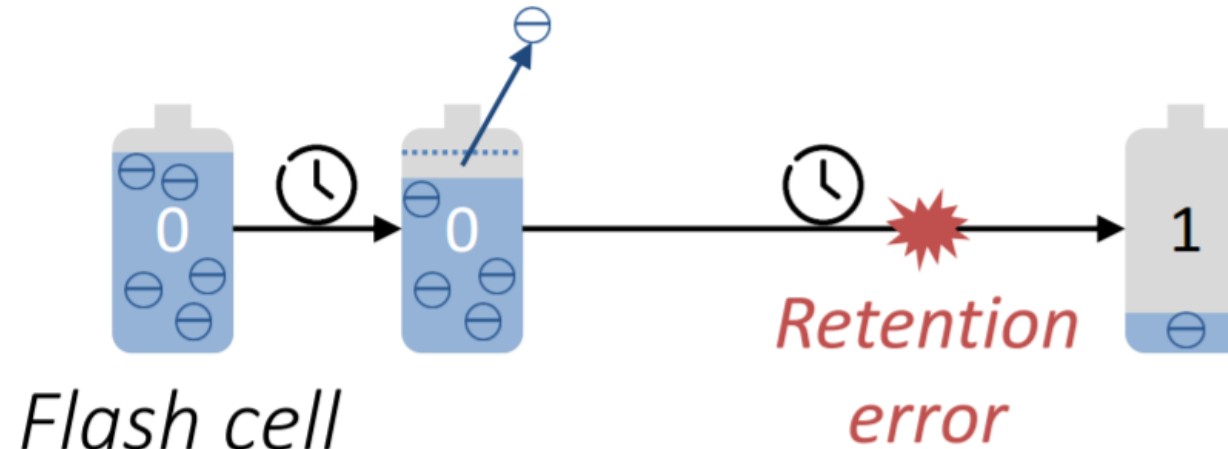


Figure 3 MLC Flash Data Storage

Retention Loss

Charge leakage over time



*One dominant source of flash
memory errors [DATE '12, ICCD '12]*

Correction to TI application note SLAA334

- Understanding MSP430 Flash Data Retention, SLAA392

SLAA392

INSTRUMENTS

2.3.1 Accelerated Tests

To test the flash data retention at various temperatures we make use of accelerated tests on the flash. These tests are wholly based on Arrhenius law and equation. The Arrhenius theory allows the test of any device under accelerated environments for short periods and predicts the behavior under normal conditions for longer periods. Similar tests are performed on the MSP430 flash to test and predict data retention. During each test, an unprogrammed device is subjected to these tests. A flash failure is indicated when any of the flash cells change from an unprogrammed state (logic 1) to logic 0. The Arrhenius equation is shown in Equation 1.

Correction to TI application note SLAA334

- Let's take a look at the TI SLAA334 statement

3.1 *Data Retention*

3.1.1 Leakage Mechanism

Data retention is limited by leakage current through the insulating oxide. Leakage can only occur if the floating gate is fully charged. Therefore, ~~leakage only can flip an erased cell with the logic level 1 to a programmed cell with the logic level 0.~~ According to Manabe [1], there are several phenomena that cause leakage.

Writing to the flash takes a long time

- Silicon Labs' Leopard Gecko EFMLG32 write to flash example
- During writes to flash in the Leopard Gecko, no access to flash memory is allowed
 - Not even instructions to execute
 - It will stall the processor
- Must plan the write to flash when access to flash will not be required
- Page size is 2048 bytes or 512 4-byte words
 - Erase page 20.8ms
 - Write 512*20uS 1.0ms
 - Total time of **21.8ms**
- Must plan writes when no time critical interrupts can occur
 - For BLE operations, it should be planned between Connection Events so the ConnInterval or ServerLatency should be greater than the time to write to flash

Table 3.7. Flash

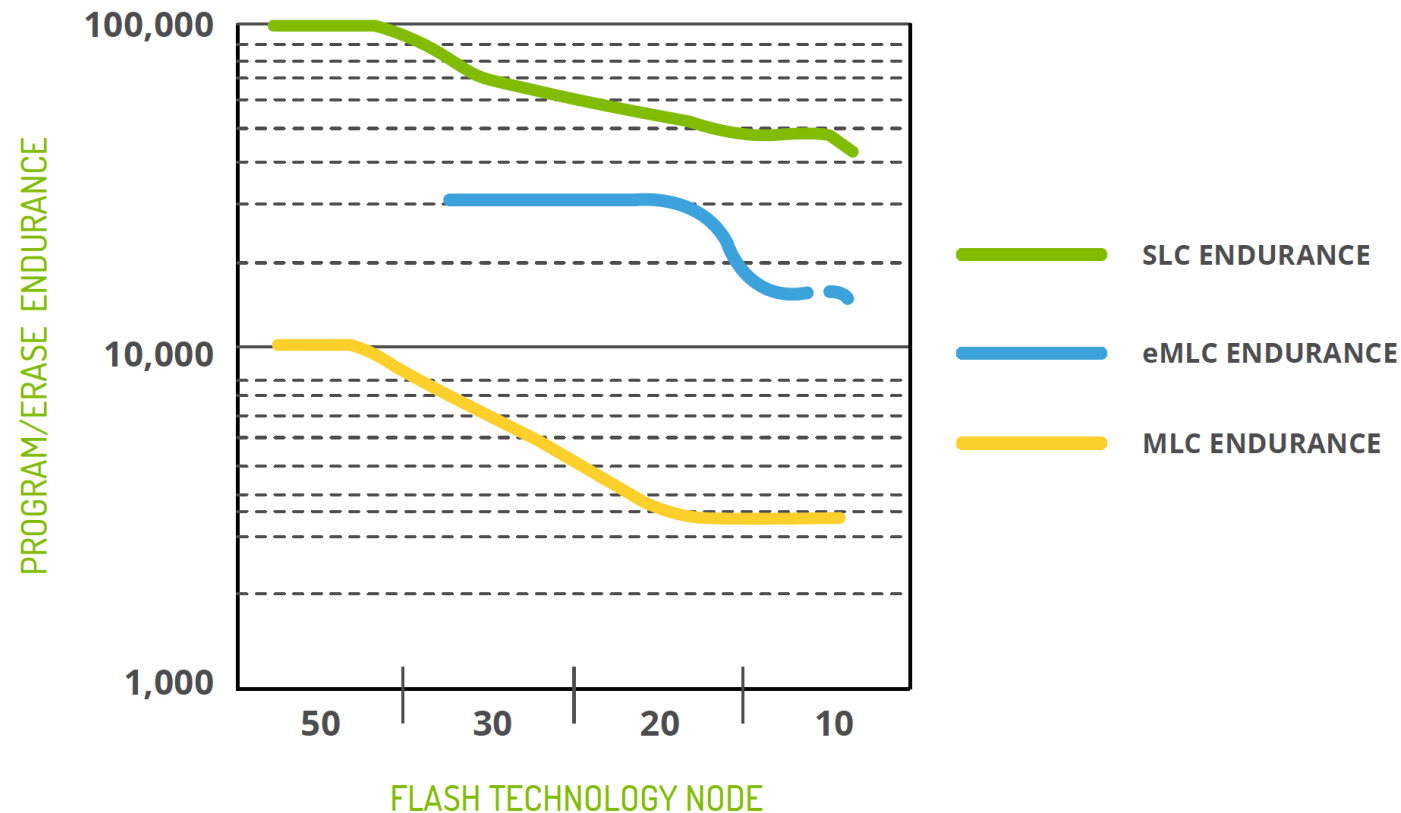
Symbol	Parameter	Condition	Min	Typ	Max	Unit
EC _{FLASH}	Flash erase cycles before failure		20000			cycles
RET _{FLASH}	Flash data retention	T _{AMB} <150°C	10000			h
		T _{AMB} <85°C	10			years
		T _{AMB} <70°C	20			years
t _{W_PROG}	Word (32-bit) programming time		20			μs
t _{PERASE}	Page erase time		20	20.4	20.8	ms
t _{DERASE}	Device erase time		40	40.8	41.6	ms
I _{ERASE}	Erase current				7 ¹	mA
I _{WRITE}	Write current				7 ¹	mA
V _{FLASH}	Supply voltage during flash erase and write		1.98		3.8	V

¹ Measured at 25°C

Writing to the flash takes a long time

- **Solution:** Find a time that critical interrupts can be turned off
 - In a BLE application, is there a time between ConnIntervals or ServerLatency?
 - Is there a time in the day that operation is not critical?
 - Example: micro converter on a solar panel during the solar panel daily wakeup
 - The product gets plugged in to charge
 - Example: A fit bit watch does not need to perform its primary role while plugged in to be charged
- **Solution:** Use an external Flash
 - External flash does not tie up a microcontroller's critical resources such as program memory
 - External flash such as eMMC have integrated algorithms to minimize data retention and other errors such as write disturb
 - External flash with the correct capacitance coupling can provide guaranteed write completion
 - Negative, an additional part resulting in higher part cost and board real estate

Endurance versus Moore's Law



Endurance goes down as the area to store the electrons in the floating gate gets smaller. Less electrons, less margin or separation between a "0" and a "1" state.

NAND Technology types

- SLC
 - Single Level Cell
 - Each NAND cell is one bit, or two states (0,1)
- MLC
 - Multi Level Cell
 - Each NAND cell represents two bits, or four states (0,1,2,3)
- TLC
 - Tertiary Level Cell
 - Each NAND cell represents three bits, or 6 states (0,1,2,3,4,5)
- 3d
 - Memory cells stacked on top of each other (3d)
 - Can be SLC or MLC

NAND Technology comparisons

- SLC

- Relatively fast read and write capabilities
- Good endurance
- And relatively simple error correction algorithms
- More expensive than MLC and TLC since one bit for the same area

- MLC

- Twice the density of SLC, thus less cost per bit than SLC
- Roughly 1/3 the speed of SLC
- And, Roughly 1/10 the reliability of SLC
- Most common flash today – good balance between cost and performance

NAND Technology comparisons

- TLC

- 3x the density of SLC
- Much lower performance
- Reliability is lowest of all NAND types
- Good for applications that have low amounts of writes such as MP3 players

- 3d NAND

- Technology based on stacking NAND cells
- Increases the number of layers and steps in the manufacturing process
- Enables costs to continued to go lower
- Reliability increased over standard MLC due to larger feature size, better insulating material, and charge trap design

NAND failure mechanisms

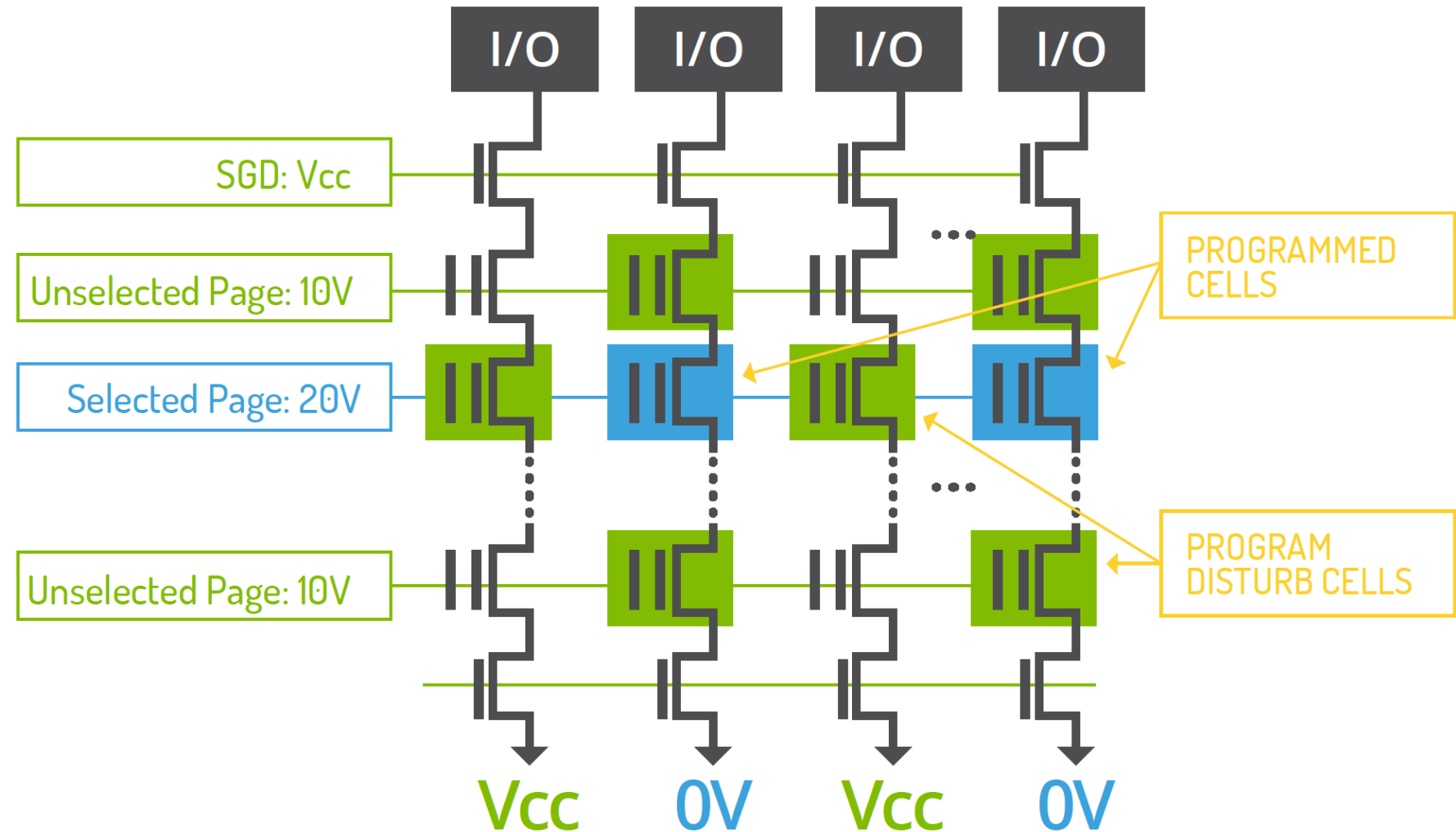
- Erase/Program cycle endurance
 - Same as NOR
- Data Retention
 - Same as NOR
- Reliability versus Moore's Law
 - Same as NOR
- Read Disturb
 - NAND failure mode
- Write Disturb
 - NAND failure mode

Program disturb occurs in neighboring cells of the ones being programmed. This happens because the neighboring cells are exposed to voltage levels which are higher than normal. This setup causes these cells to appear to be weakly programmed. Fig.5 illustrates a representation of this problem:

Write Disturb

A write disturb error would result in a "1" bit going to a "0."

Note: MLC NAND is more sensitive to Program/Write disturb due smaller voltage margin per state.

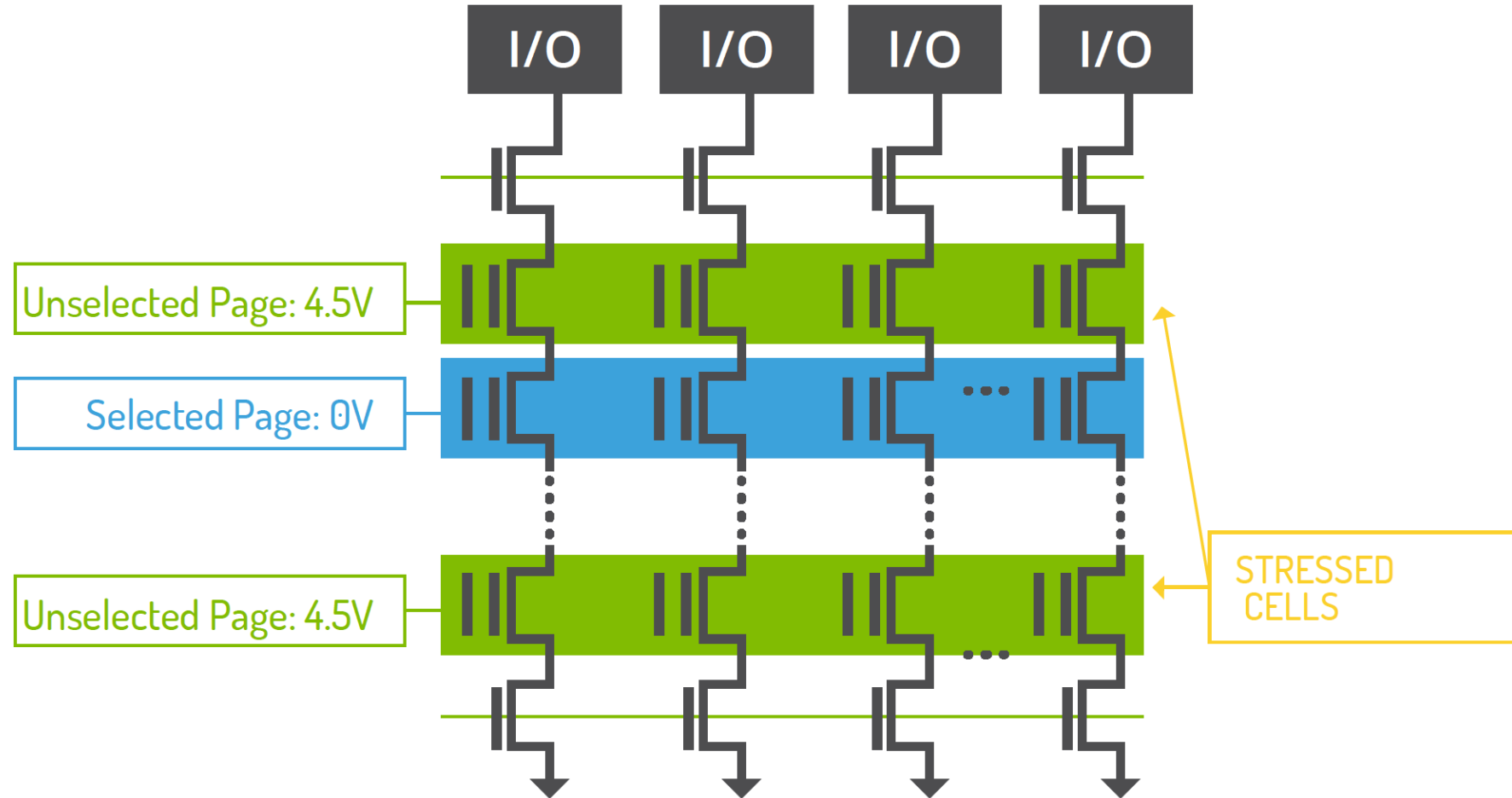


Read Disturb

Read disturb happens in neighboring cells of the ones being read due to stray charge being coupled to the floating gates of the unselected cells. This problem is not as severe as write disturb but is getting worse as flash geometry shrinks. Fig. 6 illustrates this scenario:

A read disturbed error would result in a "1" bit going to a "0."

Read disturb errors only occur on the adjacent cells, and not the cell read.



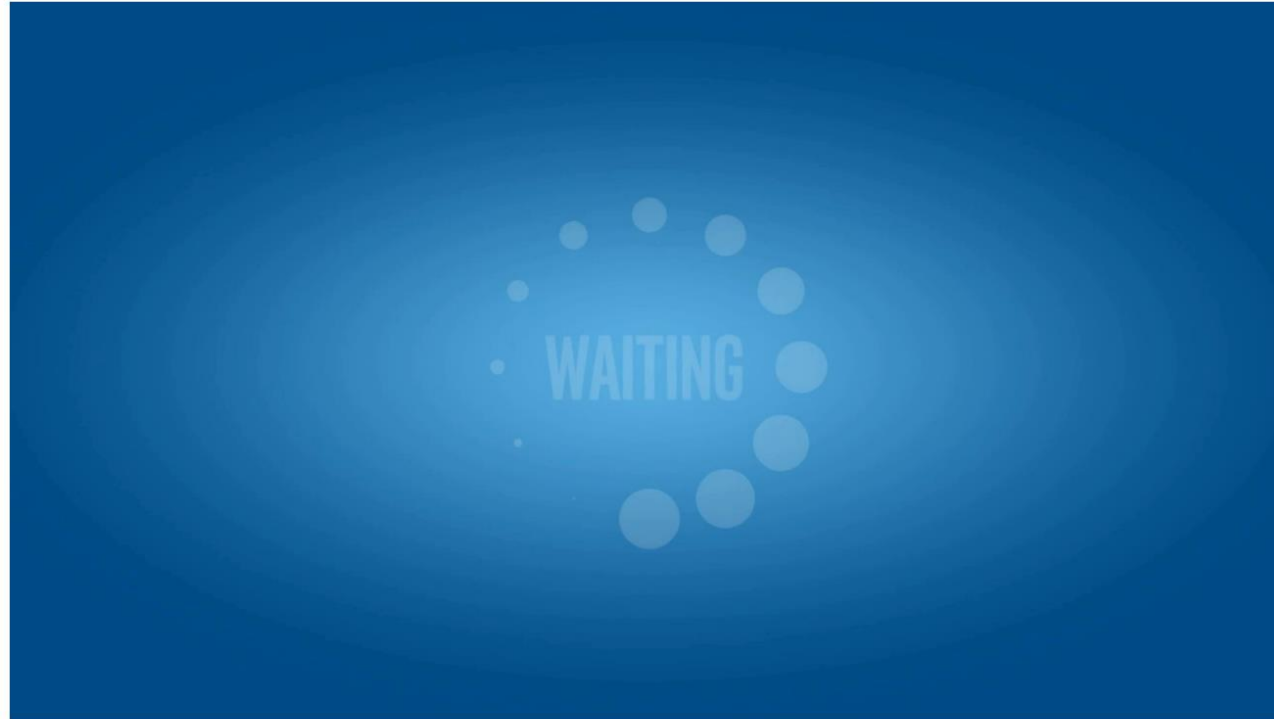
Dynamic (Global) Wear Leveling

- **Dynamic wear** leveling is a method of pooling the available blocks that are free of data and selecting the block with the lowest erase count for the next write. This method is most efficient for dynamic data because only the non-static portion of the NAND Flash array is wear-leveled. A system that implements dynamic wear leveling enables longer NAND Flash device life than a system that does not implement wear leveling.

Static Wear Leveling

- **Static wear** leveling utilizes all good blocks to evenly distribute wear, providing effective wear leveling and thereby extending the life of the device. This method tracks the cycle count of all good blocks and attempts to evenly distribute block wear throughout the entire device by selecting the available block with the least wear each time a program operation is executed. Static data is managed by maintaining all blocks within a certain erase count threshold. Blocks that contain static data with erase counts that begin to lag behind other blocks will be included in the wear-leveling block pool, with the static data being moved to blocks with higher erase counts.

Intel and Micron breaking the NAND technology limitations – 3D XPoint



3D XPoint™ Technology Revolutionizes Storage Memory

Intel engineers help shatter all the rules with 3D XPoint™ technology, a simple, stackable, and transistor-less design that creates fast, inexpensive, and nonvolatile storage memory with low latency to unleash your processor's true potential.

eMMC

- eMMC
 - Embedded MultiMediaCard Memory
 - MMC (MultiMediaCard)
 - Released in 1997 by SanDisk and Siemens AG
 - Based on NAND memory
 - Much smaller than other non-volatile cards in 1997 based on NOR Flash technology such as CompactFlash
 - Designed to solve NAND memory issues for the system designer
 - Perform ECC
 - Increase reliability
 - Static wear leveling
 - Dynamic (Global) wear leveling
 - System design becomes independent to NAND die changes resulting in ECC changes



Micron example of wear leveling benefits

- Consider a case without wear leveling. In a NAND Flash device with 4,096 total blocks and 2.5% allowable bad blocks in a system that updates 3 files comprised of 50 blocks each at a rate of 1 file every 10 minutes (or 6 files per hour), where a NAND host reuses the same 200 physical blocks for these updates, the NAND Flash device will wear out in under 1 year, leaving over 95% of the memory array unused.
- No wear leveling:

Only 200 blocks are reused:
$$\frac{10,000 \text{ cycles} \times 200 \text{ blocks}}{50 \text{ blocks per file} \times 6 \text{ files per hour} \times 24 \text{ hours per day}} = \sim 278 \text{ days or } < 1 \text{ year}$$

Dynamic (Global) wear leveling example

- In a 4,096-block MLC device with a 10,000-cycle count, 75% static data, and a program and erase rate of 50 blocks every 10 minutes (or 6 files per hour), dynamic wear leveling results in device wear-out after approximately 4 years, with 75% of the blocks nearly unused.

Wear leveling only dynamic data: $\frac{10,000 \text{ cycles} \times 1,024 \text{ blocks}}{50 \text{ blocks per file} \times 6 \text{ files per hour} \times 24 \text{ hours per day}} = \sim 1,422 \text{ days or } < 4 \text{ years}$

Static wear leveling example

- Using the same example of a 4,096-block MLC device with a 10,000-cycle count, 75% static data, and a program and erase rate of 50 blocks every 10 minutes (or 6 files per hour), static wear leveling provides the best chance of extending the device life span beyond 15 years.

Wear leveling static and dynamic data:
$$\frac{10,000 \text{ cycles} \times 4,096 \text{ blocks}}{50 \text{ blocks per file} \times 6 \text{ files per hour} \times 24 \text{ hours per day}} = \sim 5,689 \text{ days or } >15 \text{ years}$$




Preventing Read Disturbance

- InnoDisk Firmware is designed to resolve this issue by wear leveling and refresh (“re-charges”).
- With wear leveling feature, not only spread the program/erase count evenly on all blocks, but also can reduce the read access frequency to prevent Read Disturbance by reprogramming the data to different blocks.
- Alternatively, ECC (Error Correcting Code) can detect and fix the data where the electrical properties may have been altered by refresh. When error bits in a block reach a threshold of say 17 error bits out of 24 bits, the block is automatically refreshed. i.e. the data is deleted and re-written.
- This stops the controller from constantly reading blocks with too many error bits and prevents read disturbance.

Wear Leveling Summary

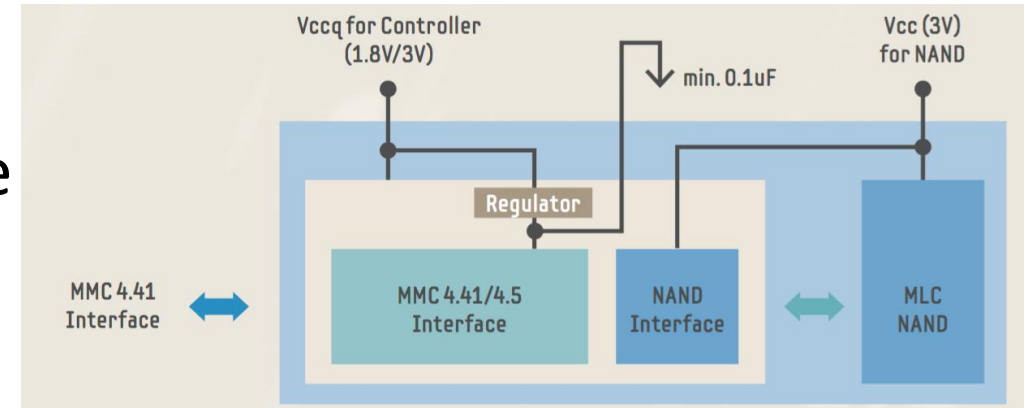
Comparison [\[edit \]](#)

The following table compares static and dynamic wear leveling:^{[\[3\]](#)}

Item 	Static 	Dynamic 
Endurance	Longer life expectancy	Shorter life expectancy
Performance	Slower	Faster
Design Complexity	More complex	Less complex
Typical Use	SSDs ^{[2]}	USB Flash Drives

eMMC architecture

- eMMC memory is 2 die in one package
 - An EMMC controller
 - NAND memory
- eMMC advantages
 - Industry standard in both hardware and software
 - Enables the eMMC solution to be multi-source to enhance availability and reduce cost
 - Off loads the NAND management from the system firmware or hardware
 - System hardware and firmware does not need to change as NAND memory shrinks and changes ECC scheme – the change is supported by the eMMC controller

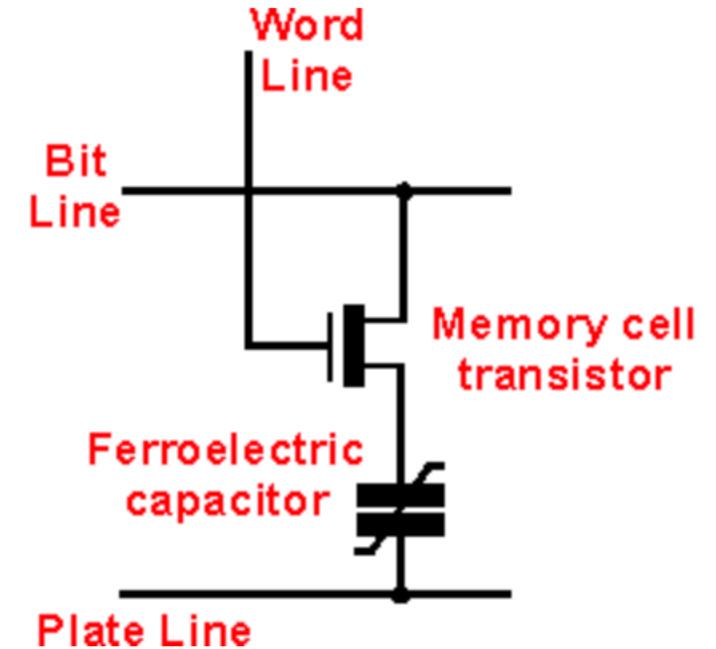


Example of NAND management offload

- Hyperstone S8 NAND Management Features
 - hyReliability™ Flash Memory Management optimizing reliability, power fail safety, endurance, data retention, and performance
 - Read Disturb Management, dynamic data refresh to maximize data retention and refresh data subject to read disturbance
 - Static and Global Wear leveling to maximize write endurance
 - Bad Block Management
 - Complete Flash Translation Layer (FTL) for random Flash data access including mapping of logical block addresses (LBA) to physical block addresses (PBA)

FRAM

- Ferroelectric Random Access Memory (FRAM), also known as FeRAM or F-RAM, is a memory technology that combines the best of Flash and SRAM. It is non-volatile like Flash, but offers fast and low power writes, write endurance of 10^{15} cycles, code and data security that is less vulnerable to attackers than Flash/EEPROM



Basic Ferroelectric memory cell

Radio-Electronics.com "FRAM
Ferroelectric Random Access Memory
Tutorial"

FRAM Technology

- Molecular Structure
 - FRAM is a random access memory, meaning that each bit is read and written individually. This non-volatile memory is similar in structure to DRAM, which uses one transistor and one capacitor (1T-1C), but FRAM stores data as a polarization of a ferroelectric material (Lead-Zirkonate-Titanate). As an electric field is applied, dipoles shift in a crystalline structure to store information.
 - The use of crystal polarization as opposed to charge storage enables state retention, **lower** voltage requirements (as low as 1.5V) and **fast** write speeds when compared against Flash, EEPROM and SRAM technologies used in typical microcontroller.

FRAM – Molecular Structure

In contrast to the complex charge storage mechanism used in EEPROM and flash, FRAM stores information through the use of a spontaneous, stable electric dipole found in the ferroelectric crystal. Intrinsically, the dipole atom within a ferroelectric crystal has either positive or negative orientation, as shown in Figure 2-4.

Applying an electrical field polarizes the material by creating large regions of the crystal with Ti/Zr ions all oriented the same direction (domains). By applying a voltage of opposite polarity above the coercive voltage, the Ti/Zr ions will have enough energy to overcome the energy barrier in the center of the cell to move to the other low-energy site as Figure 2-5 illustrates.

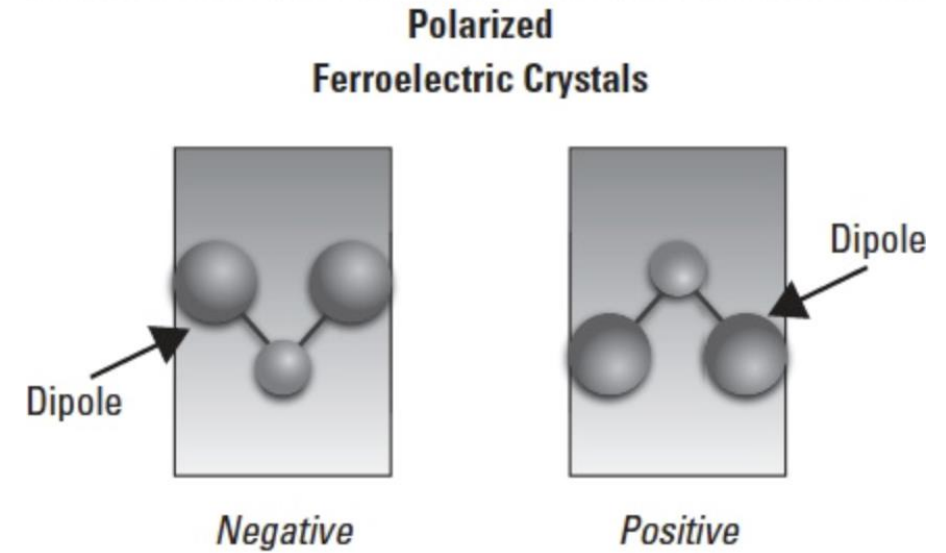


Figure 2-4. Dipole positions of ferroelectric crystals.

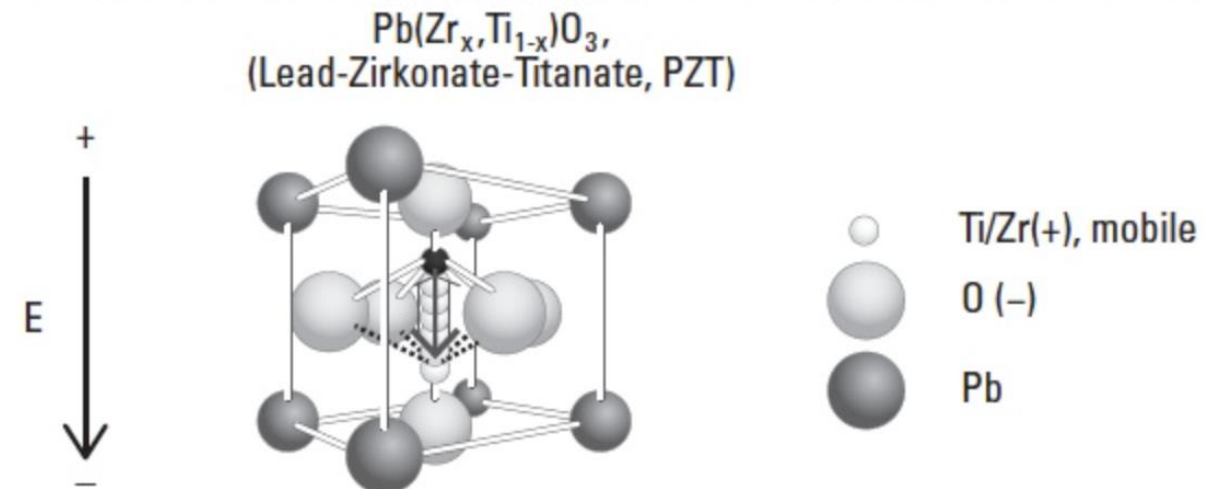
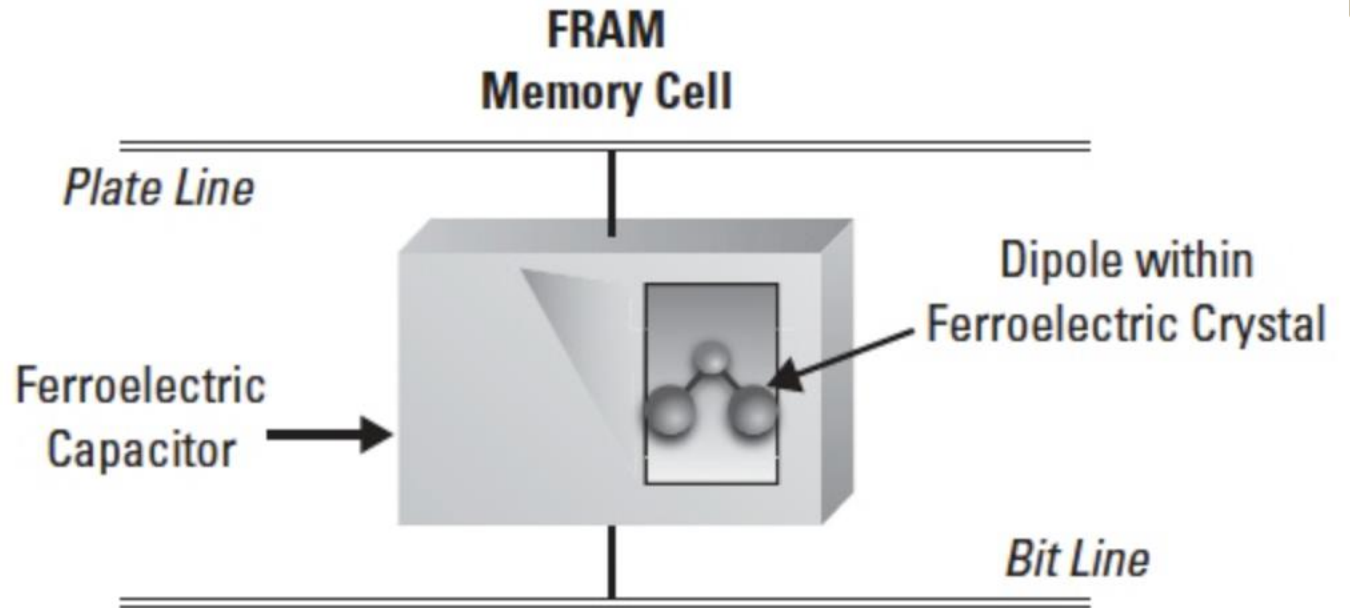


Figure 2-5. Lead-Zirconate-Titanate structure.

FRAM Technology

- Reliability/Security advantages of FRAM Technology
 - The lack of a charge pump removes a key vulnerability against physical attacks.
 - FRAM is also resistant to electric/magnetic fields as well as radiation. Since FRAM state is not stored as a charge, alpha particles are not likely to cause bits to flip and the FRAM Soft Error Rate (SER) is below detectable limits.
 - On top of this resistance to external interference, FRAM is anti-tearing, meaning power lost during a write/erase cycle will not cause data corruption.

FRAM Read/Write Operations



- Read and write operations represent the fundamental way that data is accessed and stored in semiconductor memory.
- An FRAM memory cell consists of a ferroelectric capacitor containing crystalline PZT, which contain many ferroelectric domains, each of which has the same dipole orientation.
- The capacitor is connected to by a plate line and bit lines (see Figure 2-9) and a transistor switch to access the capacitor. For PZT materials, this is a titanium or zirconium ion in a lead/oxygen crystal lattice.

FRAM Read Operation

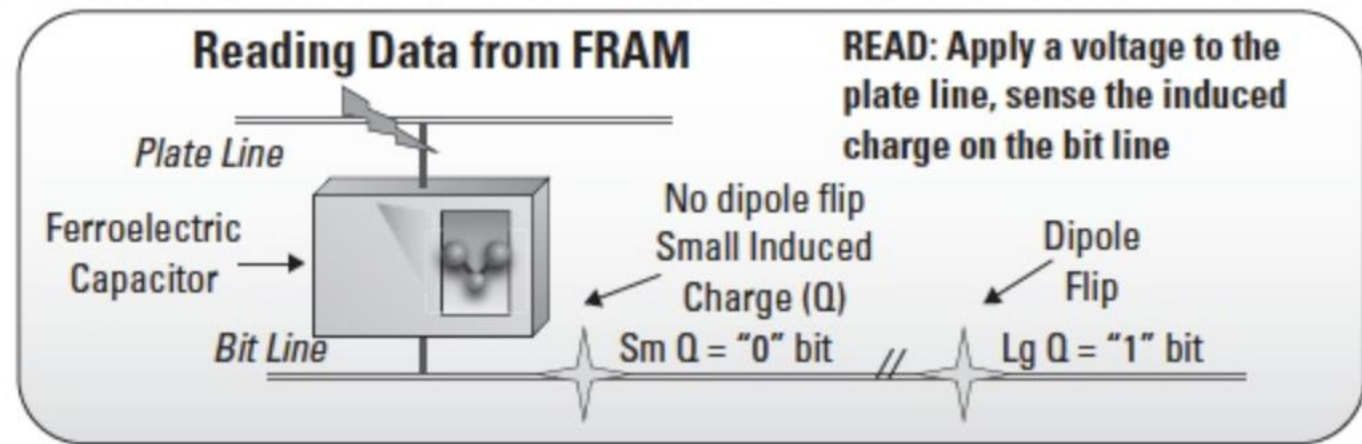


Figure 2-10. Reading a FRAM cell

- To read the data from a FRAM memory cell, a voltage is applied to the plate line; the key here is that you are **potentially changing the state in order to perform a read, FRAM always requires a memory state refresh after a read.** the voltage causes dipoles to align. If the voltage is high, then a large induced charge is sensed on the bit line. If the voltage is low, then a small induced charge is sensed on the bit line. So, in reading the data from an FRAM cell, a small induced charge is a 0 bit and a large induced charge is a 1 bit (see Figure 2-10).

FRAM Write Operation

- Writing to FRAM is also simple. To write a 1, you apply a voltage to the bit line to force a change in the orientation of the dipole to a positive 1 bit. To write a 0, you apply voltage to the plate line to move state to 0.

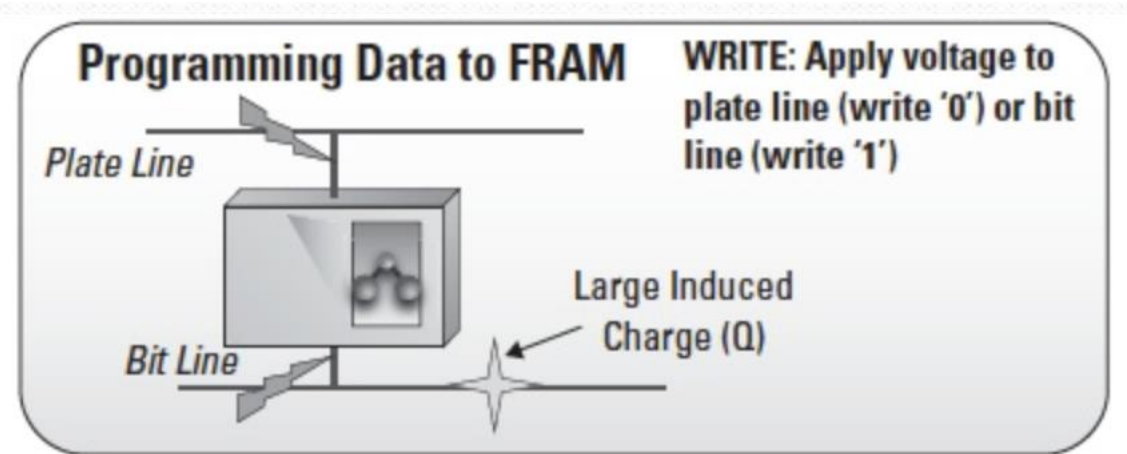


Figure 2-11. Writing to a FRAM cell.

FRAM memory comparisons

All-in-one: FRAM MCU delivers max benefits				
Specifications	FRAM	SRAM	EEPROM	Flash
Non-volatile <i>Retains data w/o power</i>	Yes	No	Yes	Yes
Write speed <i>(13 KB)</i>	10ms	<10ms	2 secs	1 sec
Average active Power [μA/MHz] <i>16 bit word access by the CPU</i>	100	<60	50,000+	230
Write endurance	10 ¹⁵	Unlimited	100,000	10,000
Soft Errors	Below Measurable Limits	Yes	Yes	Yes
Bit-wise programmable	Yes	Yes	No	No
Unified Memory <i>Flexible code and data partitioning</i>	Yes	No	No	No

* Based on devices from Texas Instruments

FRAM use cases

- Remote sensing or data logging
 - Lower energy
 - Fast writes
 - Low voltage and current is needed to change FRAM data
 - 10 billion times more cycles than Flash
- Over the air updates
 - Updating FRAM takes 100x less time and 250x less energy/bit
 - No pre-erase required
 - Data can be written on-the-fly
 - Data can be written to FRAM right out of the COMM channel, with no buffering required

FRAM use cases (continued)

- Energy Harvesting
 - Low active duty cycle for non-volatile writes
 - Low average and peak write power leads to low average and peak power consumption of the MCU
 - Faster wakeup time
 - Variables stored in non-volatile FRAM Over the air updates
- Data Security
 - No charge pump needed
 - Resistance to external fields
 - Memory protected from some types of physical attacks
 - State retention on power fail, fast writes and 10 write cycles
 - FRAM is not susceptible to Soft Errors
 - Update security keys quickly and send notifications in case of certain state changes

FRAM Advantages/Disadvantages

- **Advantages**

- Lower power usage
- Faster write performance
- Much larger number of write-erase cycles

- **Disadvantages**

- Lower storage density
- Overall capacity limitation
- Higher cost

The over riding disadvantage is cost. The FRAM cell structure is limited on how small the structure can be made. One limitation is that as structures become small, they tend to stop being ferroelectric. This effect is related to the ferroelectric's "depolarization field." Currently TI is building FRAM at 130nm linewidths where flash is being build in line widths as small as 16nm.

EEPROM

- EEPROM – Electrically Erasable Programmable Read Only Memory
- A form of NOR flash that has been optimized for:
 - Byte writeable
 - Increased endurance
 - Possibly ECC
- Due to its optimization, it is more expensive to implement on a microcontroller than NOR flash
 - Larger cell size
 - Limits the EEPROM portion of a controller for data storage

Dynamic Memory (DDR3, DDR3L, LPDDR3)

- DDRx – often referred to as (JEDEC) standard or commodity DRAM or just DRAM (DDR, DDR2, DDR3. etc.) JEDEC standard JESD79E, etc
- LPDDRx – Referred to as low power, mobile or wireless DRAM (LPDDR, LPDDR2, LPDDR3). Also defined by JEDEC standard JESD209A, etc
- In most system, the type of DRAM will be limited by the micro processor, DSP, or FPGA memory controller, thus the DRAM should be included in the decision of these devices.
- Most common DRAM in today's mobile devices will be DDR2 or DDR3

DDR3 comparisons

- LPDDR3

- Core voltage: 1.2v (1.8v WL required)
- I/O voltage: 1.2v
- Max Data rate: DDR1600
- Pin Config: 16x, 32x
- Partial Array Self-Refresh: individual bank and segment masking for partial-bank modes
- Deep Power Down Mode: Yes

- DDR3

- Core voltage: 1.5v
- I/O voltage: 1.5v
- Max data rate: DDR2100
- Pin Config: 4x, 8x, 16x
- Partial Array Self-Refresh: Optional
- Deep Power Down Mode: No

- DDR3L

- Core voltage: 1.3v
- I/O voltage: 1.3v
- Max data rate: DDR2100
- Pin Config: 4x, 8x, 16x
- Partial Array Self-Refresh: Optional
- Deep Power Down Mode: No

SDxx memory cards provide:

- SPI bus to interface to micro controllers
- Card Security
 - Commands to disable writes
 - Write-protect notch
 - Card password
 - DRM copy-protection
- Real World Performance Issues:
 - Write Amplification
 - File Fragmentation

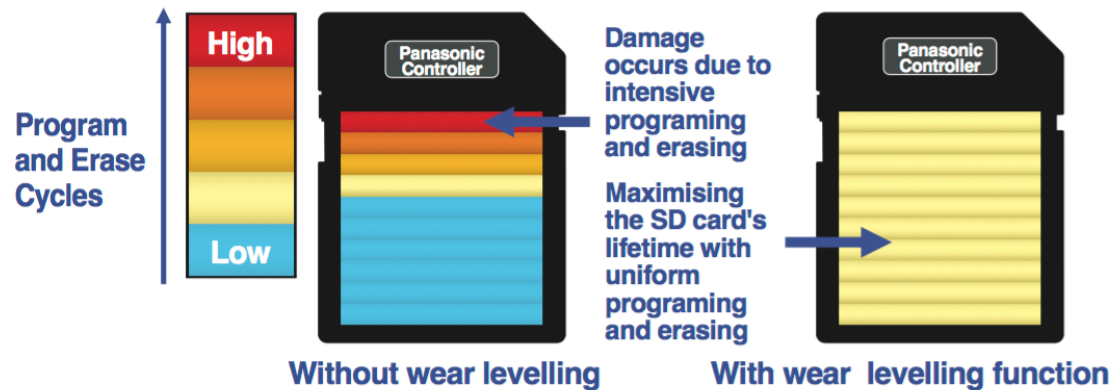
Example of Industrial SD memory card

Data Programming and Erase Endurance

Wear Levelling

● Maximising SD Memory Life

Static wear levelling controls written data, including fixed data. Various use cases eliminate intensive data writing and maximise the lifetime of the SD card.



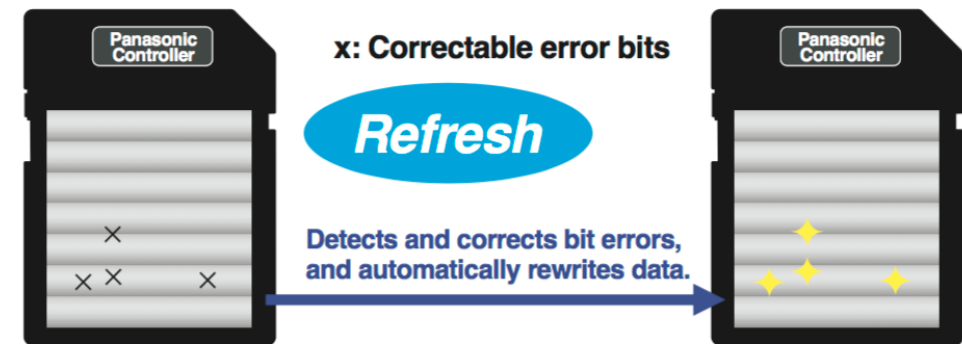
Secure Storage

Bit Error Auto Refresh

● Withstanding Repeated Reading Operations

Automatically refreshes the bit errors that accumulate over time, before they exceed the threshold. (Accumulated bit errors are detected from read data.)

* This function does not guarantee permanent data retention.



Example of Industrial SD memory card (Cont.)

Intelligent Data Writing

- **Dispersion of Writing Stress to NAND Flash Memory**

Intelligent data writing disperses the writing stress to NAND flash memory, to reduce program disturbances.

Recovery

- **Protects saved data and device**

Unique Panasonic algorithms minimise data damage in the event of a power interruption. Even in the event that an error is generated, the controller recovers the data, restoring it to the condition prior to the error, and preventing errors from reaching the entire SD memory area.

* Power Fail Robustness Mode firmware also available for more robust MLC system

Example of Industrial SD memory card (cont.)

Panasonic SD memory features high endurance against static electricity, magnetism, and X-rays.



Temperature Resistance

Operation is assured even under harsh temperature conditions.

A usable temperature range of -40 °C to 85 °C maintains stable performance everywhere, from extremely cold to intensely hot climates.



Electrostatic Resistance

ICE 61000-4-2 compliance: Clears Electrostatic Discharge Immunity Tests of 150-pF energy storage capacitance, 15-kV aerial discharge, and 330-Ω discharge resistance.



Impact Resistance

High endurance against bending and twisting.

Bending load resistance	20 N (Newton) min. (SD standard: 10 N)
-------------------------	---

Twisting torque resistance	0.3 N•m (Newton meter) min. (SD standard: 0.15 N•m)
----------------------------	--



Magnetic Resistance

Minimal damage from magnetic forces.

Operable after being set onto a 1,000-gauss DC magnetic field for approx. 1 minute.



X-Ray Resistance

Data is protected from X-rays.

ISO 7816-1 compliance: Operable after 0.1 Gy (gray) of X-ray irradiation.



Water Resistance

JIS IPX7 compliance: Operable after submerging the product in water (tap water, 1-m depth) for 30 minutes.

* micro SD – Excluding SD adaptor use.
* Card only.



Built-in Fuse

The internal card fuse protects against excess current and abnormal heating.

Even if excess current or abnormal heating were to occur due to internal card damage caused by the device being used or the environment, the built-in fuse will operate to prevent the SD Memory Card from overheating or igniting.