Freescale Semiconductor
Application Note

Using the Kinetis Family Enhanced EEPROM Functionality

by: Melissa Hunter
Derrick Klotz

1   Introduction

Some of the Kinetis family processors include a FlexMemory feature. The FlexMemory allows users to configure some of the on-chip flash memory as enhanced EEPROM, additional flash memory, or a combination of the two.

This application note discusses the use of the FlexMemory as enhanced EEPROM. The advantages of the enhanced EEPROM, configuration of the memory, and endurance calculations are discussed.

2   Enhanced EEPROM (EEE) Features

The enhanced EEPROM (EEE) capability of the FlexMemory has a number of features that allow the replacement of external EEPROMs and improves upon their performance.

Contents

FlexMemory EEE features include:

- Configurability to allow the designer to make trade-off decisions based on the EEE endurance required and the total amount of EEE and flash memory required
- High endurance EEE memory can reach over 100 M cycles (depending on configuration)
- Supports byte, word, and longword accesses
- Fast write times (~100 us for a write to a pre-erased location, and 1.5 ms for an erase and program)
- EEE functionality across the full operating voltage range for the processor (1.71 V – 3.6 V)

2.1 How FlexMemory EEE works

To provide enhanced EEPROM functionality, the FlexMemory uses a RAM block (FlexRAM), a flash block (FlexNVM), and EEE state machine. When the EEE functionality is enabled, the FlexRAM becomes your EEE memory. The FlexRAM address space is where you access all of your EEE data. When the EEE is accessed, the EEE state machine keeps track of the data and backs it up as data records, stored in some portion of the FlexNVM used as an E-flash. Using a large block of E-flash to back up the data for a smaller amount of EEE data allows the FlexMemory EEE implementation to offer extremely high endurance.

The EEE state machine uses 32-bit records to backup data from the EEE into the flash. Sixteen bits of the record are used for the data, and the other sixteen bits are address and status information about the data. The data records are written and erased as needed. This means that if a location within the EEPROM has never been accessed, there will not be a data record for it. This helps to reduce the amount of data that needs to be backed up and can increase the memory endurance.

2.2 How FlexMemory EEE uses memory

Devices that support FlexMemory have two separate blocks of flash, the P-Flash block and the FlexNVM block. The P-Flash block is intended to be used as the program flash block, but it can actually be used for storing both instructions and data. The FlexNVM block is a configurable flash block that can be used as additional flash space (D-flash), as backup memory for supporting the enhanced EEPROM functionality (E-flash), or as a combination of the two.

NOTE

The portions of the FlexNVM not used as EEE backup memory (E-flash) are referred to as D-flash. This flash would typically be used for data storage space; however, like the P-flash, D-flash can actually be used for instructions or data.

The FlexMemory consists of the FlexNVM block, FlexRAM, and EEE state machine. These three blocks together are required to support the EEE functionality.

Figure 2-1 shows the flash blocks and the FlexMemory components on a 256 KB device with FlexMemory. In this case the device is unpartitioned, so the FlexNVM block has not been designated as D-flash or E-flash.

NOTE

Kinetis family device part numbers indicate only the size of the P-flash block, therefore the size of the FlexNVM is not explicitly listed in the part number. For example an MK60X256 device has 256 KB of P-flash and 256 KB of FlexNVM.
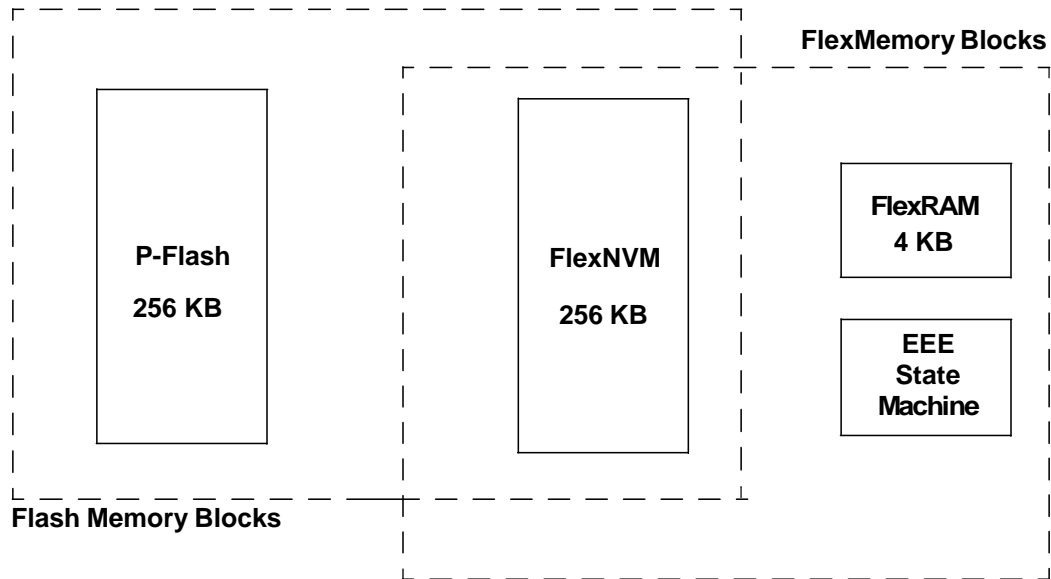

Figure 2-1. 256 KB device with FlexMemory before partitioning

2.2.1   FlexMemory with EEE Functionality Disabled

Figure 2-2 shows how the memory blocks function, if the EEE functionality is disabled on a processor that supports FlexMemory.
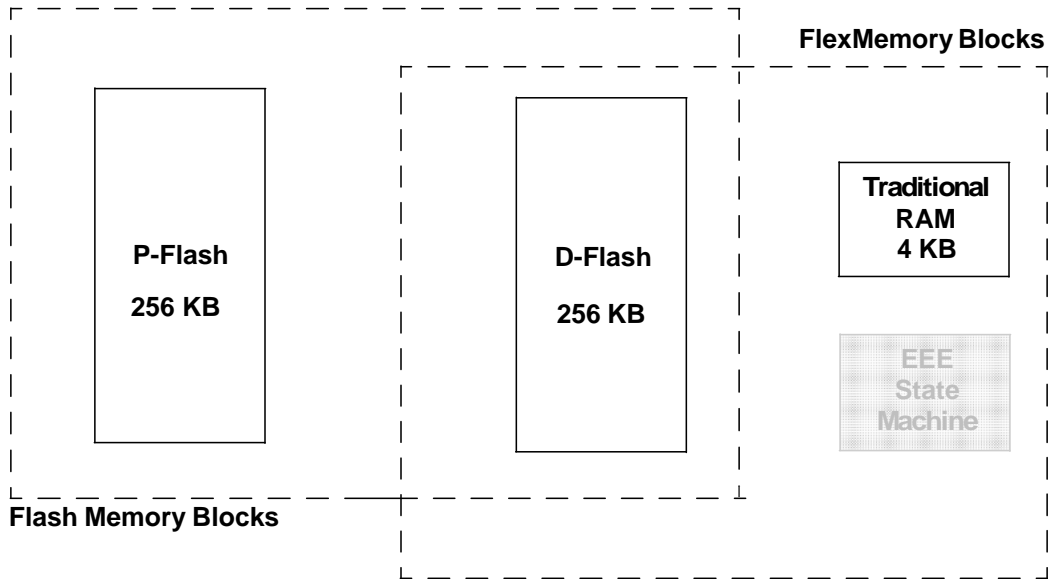
Figure 2-2. 256 KB device with FlexMemory with EEE disabled

The P-Flash memory is always a P-Flash. Its functionality does not change for any FlexMemory configuration. Because the EEE functionality is not used in this case, the entire FlexNVM is allocated as D-flash space (no E-flash is needed). The FlexRAM becomes a 4 KB traditional RAM. This means it can be used as extra memory space, but keep in mind that it does run at the flash clock speed and not the core speed (the tightly coupled memory RAM, TCM, runs at the core speed). The EEE state machine is present in the device, but not active.

2.2.2 FlexMemory with EEE functionality enabled

Figure 2-3 shows how the memory blocks function, if the EEE functionality is enabled and the entire FlexNVM is used to back up the EEE data.
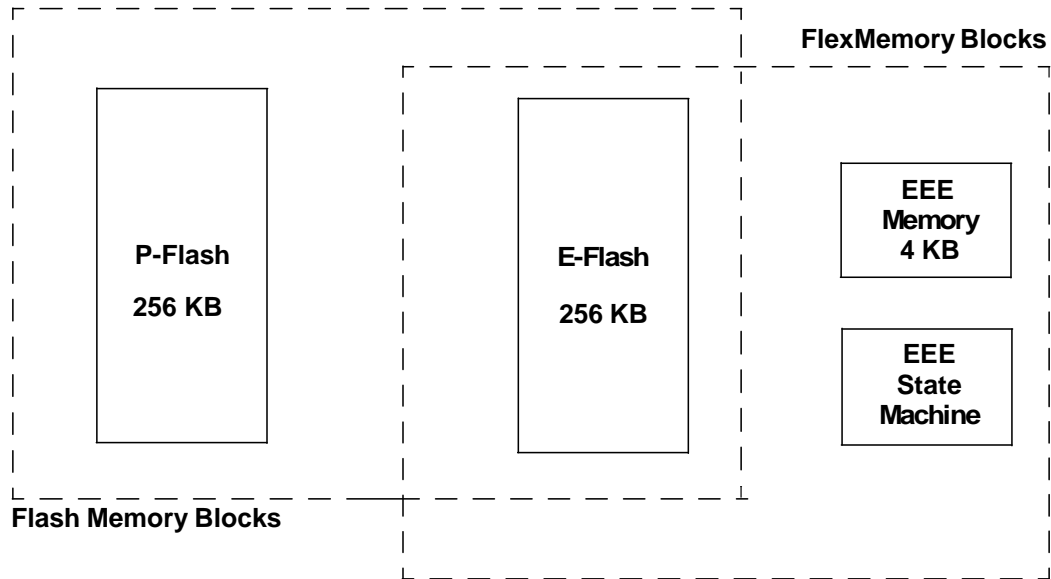
Figure 2-3. 256 KB device with FlexMemory with EEE enabled (Entire FlexNVM used to backup EEE data)

There are a number of configuration options that can be used when the EEE functionality is enabled. Figure 2-3 shows an example of where the entire FlexNVM is used as E-flash memory. The FlexRAM becomes the EEE memory space (up to 4 KB). Any reads and writes of the EEE data uses this 4 KB memory space because the E-flash is not directly accessible. The EEE state machine automatically manages all of the writes to the EEE memory space, and generates flash program and erase operations as needed into the E-flash.

2.2.2.1 E-flash and EEE memory configuration details

Figure 2-3 shows the memory blocks at a high level and shows the respective functionality of the configurable memory blocks. This section describes the actual use of the E-flash and FlexRAM blocks to create a EEE subsystem in more detail and provide examples of some different configurations.

2.2.2.1.1 EEE configuration parameters

The FlexMemory allows for many different memory configuration options. The amount of EEE data and the amount of E-flash memory used to backup that EEE data are all programmable. This allows you to make trade-offs between the memory size and endurance of the EEE.

There are three programmable options that are used to define the exact memory use for a system. These parameters are:

1. EEE size — This is the total size of the EEE data needed. The total EEE size can be 32 bytes to 4 KB, programmable in power of 2 increments (32, 64, 128, ...).

2. EEE split — The EEE state machine treats the EEE as two independent subsystems (subsystem A and subsystem B). Each subsystem is always backed up by the same amount of E-flash memory, but the size of the EEE memory does not have to be equal. Subsystem A can be programmed to be

1/8, 1/4, or 1/2 of the total EEE size with the remainder assigned to subsystem B. If the EEE split is programmed as 1/2 then the EEE memory is evenly divided between the two subsystems, therefore the same amount of E-flash is used to backup the same amount of EEE data. This creates two subsystems with equal endurance. If the EEE split is programmed to 1/8 or 1/4, then subsystem A uses the same amount of E-flash as subsystem B, but it now backs up less EEE data. The change in proportion between the amount of EEE data and backup space means that subsystem A will have higher endurance than subsystem B. The EEE split parameter allows for subdividing data into high endurance and medium endurance areas, where normal flash space (P-flash or D-flash) would be used as normal endurance areas.

3. FlexNVM partition — This parameter defines how much of the FlexNVM is used as normal flash (D-flash) and how much is used as EEE backup memory (E-flash). If EEE is used, then at least 32 KB of the FlexNVM must be allocated as E-flash. To get the maximum possible endurance from the EEE, then the entire FlexNVM can be used as E-flash.

2.2.2.1.2   Calculating EEE endurance

For many applications, the endurance requirements for the EEE data drives the decision of how the FlexMemory must be configured. Understanding how the parameters affect the endurance is usually required to determine what EEE configuration to use.

The estimated minimum endurance for the EEE can be calculated using the equation:

$$Endurance = \frac{EFlash - (2 \times SPLIT \times EEESIZE)}{SPLIT x EEESIZE} \times RecordEfficiency \times BaselineEndurance \qquad \textbf{Eqn. 2-1}$$

Where:

- E-flash — The amount of FlexNVM allocated for EEE backup memory for the subsystem
- SPLIT — The split factor for the subsystem, determines how many bytes of EEE are in the subsystem
- EEESIZE — The total size of the EEE, The number of bytes of FlexRAM used in both subsystems.
- Record Efficiency — 0.5 for 32-bit and 16-bit writes, and 0.25 for 8-bit writes
- Baseline Endurance — Normal endurance expected from the flash (10 k cycles)

This equation calculates the minimum number of guaranteed writes allowed to each EEE data location assuming that the entire EEE size is used and that data locations are written equally and do not mix data sizes. If some EEE locations are written less fequently, the calculated minimum number of writes allow for more writes to other locations.

The equation can be used to get a general idea of the amount of E-flash that may be required for the system. However, the equation is simplified and intended to be used as a general guideline. Freescale also has a EEE endurance calculator application available. The calculator application takes into account details of how the EEE data is updated instead of assuming all data values are updated equally using the same access size. The calculator also uses a more complex and accurate tool for modeling the endurance. For applications with very specific EEE endurance requirements use of the calculator tool is highly recommended. The calculator can be downloaded from the Freescale website by typing "C90TFS_FLEXMEM_CALC" in the keyword search.

2.2.2.1.3　EEE memory configuration example 1 — maximum endurance

This example discusses the EEE configuration that yields the maximum possible endurance for EEE data. To get the maximum possible endurance you need to have as much E-flash as possible backing up the smallest possible amount of EEE data.

To get the maximum amount of E-flash for a 256 KB Kinetis device, use the full 256 KB of FlexNVM as E-flash memory. Half of the E-flash (the first 128 KB) is used to store the backup data records for EEE subsystem A, and the other half of the E-flash (the second 128 KB) is used to store the backup data records for EEE subsystem B.

To get the smallest possible amount of EEE data, configure the EEE size for the smallest possible value— 32 bytes. Then the EEE split option can be set to 1/8. This means that EEE subsystem A has 4 data bytes (the smallest possible size for a single EEE subsystem). Subsystem B is 28 bytes.

Using the endurance Equation 2-1 from Section 2.2.2.1.2, "Calculating EEE endurance" the expected endurance for both of the EEE subsystems can be calculated.

For subsystem A assuming 16-bit and 32-bit accesses, the expected endurance is:

$$EnduranceSubsystemA = \frac{(128 \times 1024) - (2 \times 1/8 \times 32)}{1/8 \times 32} \times 0.5 \times 10000 = 163830000 \approx 164 Mcycles \qquad \textbf{\textit{Eqn. 2-2}}$$

For subsystem B assuming 16-bit and 32-bit accesses, the expected endurance is:

$$EnduranceSubsystemB = \frac{(128 \times 1024) - (2 \times 7/8 \times 32)}{7/8 \times 32} \times 0.5 \times 10000 = 23395714 \approx 23 Mcycles \qquad \textbf{\textit{Eqn. 2-3}}$$

Figure 2-4 shows how the FlexNVM and FlexRAM blocks are used for this configuration.
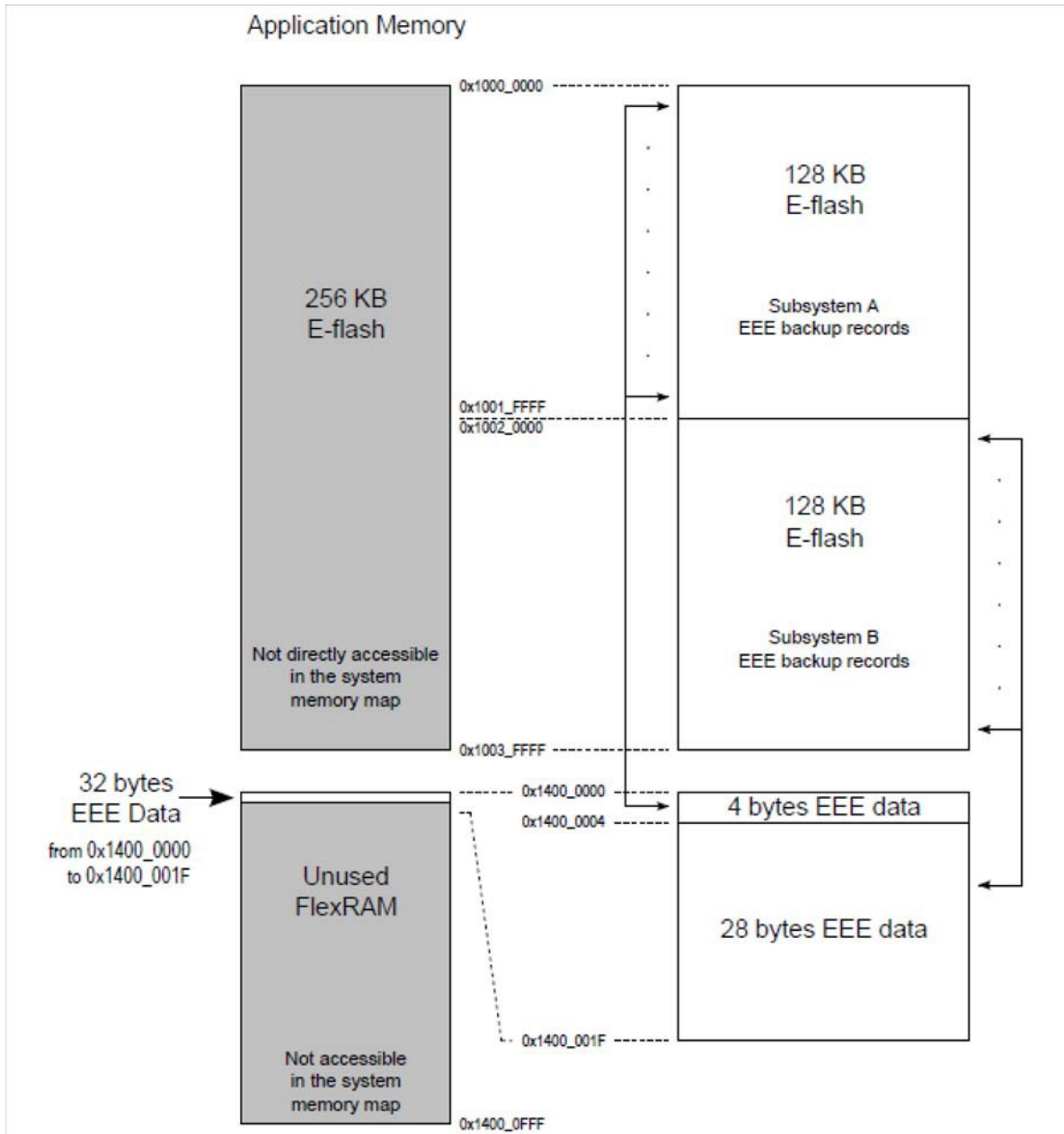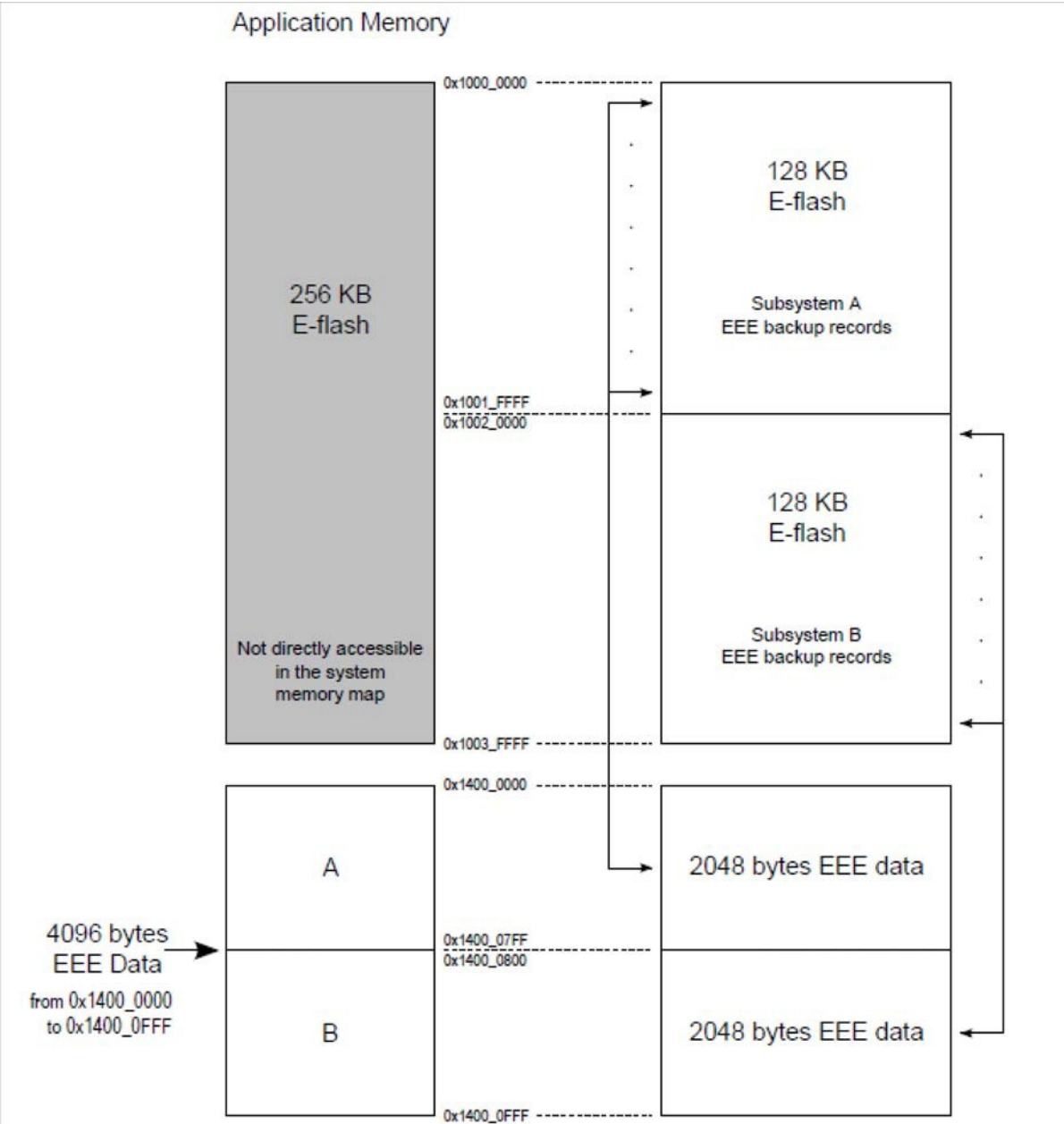
Figure 2-4. EEE configuration with E-flash = 256 KB, Subsystem A = 4 bytes, and Subsystem B = 28 bytes

2.2.2.1.4 EEE memory configuration example 2 — maximum EEE

This example discusses the EEE configuration that yields the maximum amount of EEE data space with the best possible endurance for that amount of data. The maximum possible EEE data size is 4 KB. If the entire FlexNVM is configured as E-flash (256 KB), then you can get the best endurance possible across the 4 KB EEE data set. For this example, the EEE split is configured as 1/2 therefore the endurance is equal for the entire 4 KB of EEE.

Using the endurance Equation 2-1 from Section 2.2.2.1.2, "Calculating EEE endurance" the expected endurance can be calculated. Because the size of subsystem A and subsystem B are the same, the endurance is the same for both subsystems.

For subsystem A or B assuming 16-bit and 32-bit accesses, the expected endurance is:

$$EnduranceSubsystemA/B = \frac{(128 \times 1024) - (2 \times 1/2 \times 4 \times 1024)}{1/2 \times 4 \times 1024} \times 0.5 \times 10000 = 310000 = 310kcycles \qquad \textbf{Eqn. 2-4}$$

Figure 2-5 shows how the FlexNVM and FlexRAM blocks is used for this configuration.

Figure 2-5. EEE configuration with E-flash = 256 KB, Subsystem A = 2 KB, and Subsystem B = 2 KB

2.2.2.1.5 EEE memory configuration example 3 — mix of D-flash and E-flash

This example configures the FlexNVM to be used as a mix of D-flash and E-flash. In this example the FlexNVM is divided in half to get 128 KB of D-flash memory and 128 KB of E-flash. The EEE data size is configured as 2 KB with the split between subsystem A and subsystem B as 1/2 again.

When comparing this example to example 2, the E-flash and EEE sizes are both reduced by half. Because the E-flash to EEE size ratio is the same, the expected endurance is also the same. Using the endurance Equation 2-1 from Section 2.2.2.1.2, "Calculating EEE endurance" the expected endurance can be calculated for both of the EEE subsystems. Notice that the result is the same as in Section 2.2.2.1.4, "EEE memory configuration example 2 — maximum EEE.

For subsystem A or assuming 16-bit and 32-bit accesses, the expected endurance is:

$$EnduranceSubsystemA/B = \frac{(64 \times 1024) - (2 \times 1/2 \times 2 \times 1024)}{1/2 \times 2 \times 1024} \times 0.5 \times 10000 = 310000 = 310kcycles \qquad \textbf{\textit{Eqn. 2-5}}$$

Figure 2-6 shows how the FlexNVM and FlexRAM blocks are used for this configuration.
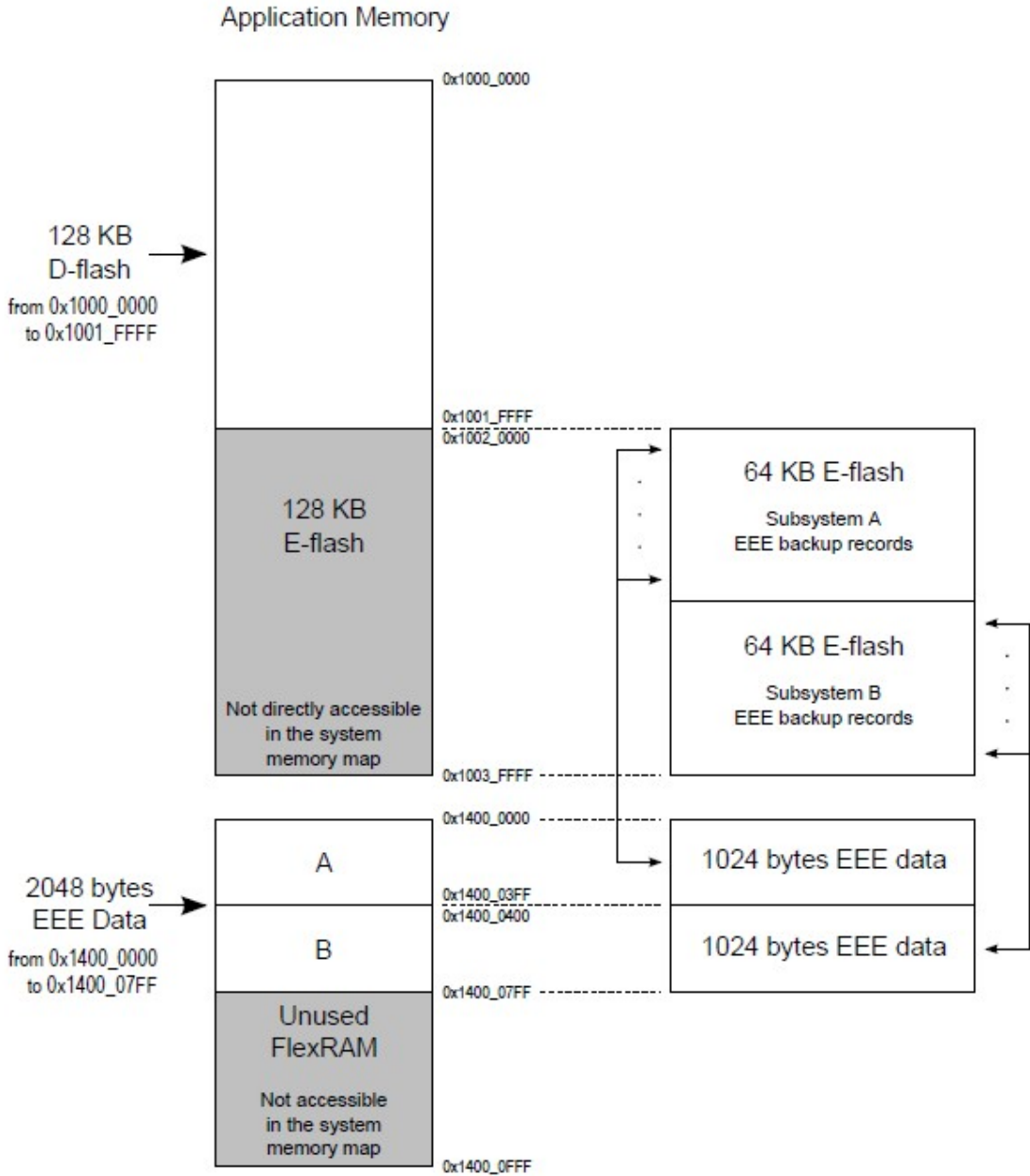
Figure 2-6. EEE configuration with E-flash = 128 KB, Subsystem A = 1 KB, and Subsystem B = 1 KB

2.2.2.1.6   EEE memory configuration example 4 — 1/4 EEE split option

This example configures the FlexNVM to be used as a mix of D-flash and E-flash. In this case 192 KB of the FlexNVM is used as D-flash, and 64 KB is used as E-flash. The EEE data size is configured as 2 KB, but this time the EEE split option is configured as 1/4. Therefore, subsystem A is 512 bytes and subsystem B is 1536 bytes.

Using the endurance Equation 2-1 from Section 2.2.2.1.2, "Calculating EEE endurance" the expected endurance for both of the EEE subsystem can be calculated.

For subsystem A assuming 16-bit and 32-bit accesses, the expected endurance is:

$$EnduranceSubsystemA = \frac{(32 \times 1024)\,(2 \times 1/4 \times 2 \times 1024)}{1/4 \times 2 \times 1024} \times 0.5 \times 10000 \quad 310000 = 310kcycles = \qquad \textbf{\textit{Eqn. 2-6}}$$

For subsystem B assuming 16-bit and 32-bit accesses, the expected endurance is:

$$EnduranceSubsystemB = \frac{(32 \times 1024)\,(2 \times 3/4 \times 2 \times 1024)}{3/4 \times 2 \times 1024} \times 0.5 \times 10000 \quad 96666 \approx 97kcycles = \qquad \textbf{\textit{Eqn. 2-7}}$$

Figure 2-7 shows how the FlexNVM and FlexRAM blocks are used for this configuration.

Figure 2-7. EEE Configuration with E-flash = 64 KB, Subsystem A = 512 bytes, and Subsystem B = 1536 bytes

3 Using the EEE

3.1 FlexMemory partitioning

To use the EEE features, the memory has to be partitioned. The partitioning process tells the state machine how much EEE memory will be used and how much of the FlexNVM flash will be used to back up the EEE.

The flash has a special *program partition* command that is used to configure the EEE. The program partition command is used to program the three EEE configuration parameters described in Section 2.2.2.1.1, "EEE configuration parameters,"— the EEE size, EEE split, and FlexNVM partition. These three parameters are programmed into a special location within the flash block itself. Because this is a non-volatile storage location, the partitioning needs to be done only once in the entire lifetime of the device. Before launching the program partition command the FlexNVM and D-flash IFR must be in an erased state. It is recommended that a new device is partitioned as the first step in factory programming.

NOTE
Partitioning must only be done once. If the flash is re-partitioned for a different configuration, then recorded data is lost and you are not guaranteed to get the expected endurance.

NOTE
Partitioning information, EEE data, and EEE location information is lost if a device is mass erased. Use of a backdoor key is highly recommended when enabling security on a device where the EEE is being used. The backdoor key allows for a means of disabling security temporarily without needing to perform a mass erase. If a mass erase is used instead then the endurance of the EEE can no longer be guaranteed.

3.2 EEE startup

After a reset the EEE configuration options written during the partitioning process are automatically loaded. If the EEE is enabled, then the state machine loads the FlexRAM with EEE data from the E-flash during the system boot. The amount of time needed to copy data from the E-flash into the FlexRAM can vary depending on the configured size of the EEE and the amount of backup E-flash that needs to be parsed. The FTFL_FCNFG[EEERDY] flag is cleared until the load of the EEE data is complete, therefore software must wait for the EEERDY flag to set before attempting to access the EEE data in the FlexRAM. If an interrupt driven option is needed instead of software polling, then the CCIF interrupt could be used instead of polling EEERDY.

NOTE
For further reference on EEE startup along with initialization code, refer to the *Kinetis Quick Reference User Guide*, Chapter 8 "Using the FlexMemory".

3.3 Reading and writing the EEE

The EEE data is read and written by accessing the FlexRAM address space. The EEE space is allocated starting at the beginning of the FlexRAM. The addressable space is the FlexRAM base address (0x1400_0000) up to the programmed EEE size. Any space in the FlexRAM that is not used as EEE must not be accessed while the EEE functionality is enabled. For example, if the EEE size is configured as 32 bytes total, then read or write accesses to any addresses between 0x1400_0000 and 0x1400_001F are allowed, but 0x1400_0020 to 0x17FF_FFFF generates a bus error.

Because the EEE data is accessed through a RAM, the data is readable and writable at any size, byte, word, or longword. Although any access size is possible, the records used to back up the EEE data use a word sized data field. This means that byte writes are possible, but they make less efficient use of the E-flash. That is why the endurance Equation 2-1 in Section 2.2.2.1.2, "Calculating EEE endurance" uses a different efficiency factor for 8-bit accesses than for 16-bit or 32-bit accesses.

3.3.1 EEE writes

Writes to the EEE space launch a EEE operation to store the data within the E-flash memory. Because this is a flash program operation, software must test the CCIF bit to determine if any other flash operations are in progress before writing to the EEE space. Because multiple concurrent writes and read-while-write operations within the same flash block are not allowable, accesses to the EEE or D-flash space are not allowed until the EEE write is complete.

NOTE

The P-flash memory is a completely separate logical block, therfore read accesses to the P-flash can continue normally while a EEE write is in progress.

3.3.2 EEE reads

When the EEE is read, the data is supplied by the FlexRAM, so no flash operations are triggered. However, EEE reads are not allowed while a EEE write is in progress. Software must either test the EEERDY bit before read operations or wait for EEERDY after a write access before allowing software to continue. In many cases it is most efficient for software to test EEERDY (or CCIF) both before and after writes and block other EEE operations until EEERDY sets after the write. This way a special function is needed for EEE writes, but a EEE read does not require any special software. Another advantage to this approach is that no additional delay or flag checking is required if you have multiple EEE read accesses with no EEE write cycles in between.

A special case for a EEE read that must be considered is the first access to the EEE after a reset. For the first read of the EEE after reset, the EEERDY bit may need to be tested to make sure that the state machine has completed the initial load of data from the E-flash to the FlexRAM. If the system start-up time is long, this guarantees that the initial data load has time to complete before the first EEE read, then a test of the EEERDY flag before the first read may not be required. However, it is safer to explicitly test the EEERDY bit before the first read access to the EEE.

4 EEE Performance

In addition to the flexibility and high endurance provided by the FlexMemory EEE implementation, the FlexMemory EEE is faster than a typical EEPROM. A traditional external EEPROM typically requires around 5 ms for the maximum program time. By comparison, the EEE can be erased and written in 1.5 ms worse case scenario.

The EEE can also be pre-erased by writing 0xFF to the EEE data locations. Pre-erasing locations help to reduce the program time because it guarantees that an erase cycle will not be needed. Typical program time

to a pre-erased data location is ~100 us. This feature allows for quick data logging in time critical situations.

One typical use case is a system where fault data or operating information needs to be stored when an imminent power loss is detected. The amount of data that needs to be saved times the maximum write time determines how much decoupling must be provided in the system to maintain minimum operating power long enough for the data to be stored. The significant decrease in the EEE program time gained by pre-erasing data locations means that in this situation less decoupling is required and more data can be stored before losing power.

5 Brownout Detection

The EEE state machine includes logic that can detect if any EEE data has not been fully programmed. This feature is referred to as brownout detection in documentation, but the name is not completely accurate. Any situation where EEE data is detected as not fully programmed due to either a brownout or any reset during the write process is treated the same. If a reset happens while a EEE write is in progress, then the data can become corrupted. The EEE state machine tests vulnerable EEE data record for values that may not be fully programmed. If an incomplete record is detected, the state machine marks the data record as *compromised* and replaces it with the previous valid data record for the associated EEE address during the next EEE write. This ensures that if a EEE write is interrupted for any reason, you will get the last value that was properly written to the EEE. Depending on how far into the write the reset occurred, this value can be either the previous value or the new value, but you will not get a corrupted value.

* Revision history

Revision Number	Date	Substantive changes
0	3/2011	Initial Release
1	3/2015	Corrections to errors in equations.

THIS PAGE IS INTENTIONALLY BLANK

How to Reach Us:

Home Page:
www.freescale.com

Web Support:
http://www.freescale.com/support

USA/Europe or Locations Not Listed:
Freescale Semiconductor, Inc.
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
+1-800-521-6274 or +1-480-768-2130
www.freescale.com/support

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.freescale.com/support

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor China Ltd.
Exchange Building 23F
No. 118 Jianguo Road
Chaoyang District
Beijing 100022
China
+86 10 5879 8000
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor Literature Distribution Center
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Document Number: AN4282
Rev. 1
03/2015