Application Note

Saving Data During a Power Failure Using the DataFlash® E-Series Family

Abstract

This application note is intended to provide readers with information on how to save critical data to the Flash memory of DataFlash E-Series devices in the event of a power failure. The power failure is simulated by removing power to the devices. This document does not discuss how the power failure is detected, how the microcontroller may respond to the failure, or how the power rail is controlled for both the DataFlash device and the microcontroller.

Product Highlights

- Single 3V Read/Write Operation (1.65V - 3.6V Supply Range)
- SPI Mode 0 and Mode 3 Compatible
- Fast Read Access Times: 85 MHz Maximum Clock Frequency
- Individual Hardware and Software Sector Protection
- Security: 128-byte Register
- JEDEC Standard Manufacturer and Device ID Read
- Endurance: 100,000 Program/Erase Cycles per Page Minimum
- Data Retention: 20 Years
- Packaging Options: SOIC, DFN, WLCSP, Die/Wafer
- Green (Pb/Halide-free) Packaging Options

Application Note 114   **11-Dec-2020**

Revision History

Revision	Date	Description
A0	11-Dec-2020	Initial Release

1   Introduction

The DataFlash E-series family of Flash memory devices are often used in applications such as server configuration, data logging, event counters and failure/error/status loggers. These and many other applications must be able to survive an unscheduled power failure and may also require that a small amount of data relating to the system status at the time of power failure.

One way of managing this type of event is to make use of one of the two 264-byte SRAM buffers within the device, leaving the other buffer for the application use. This buffer can be preloaded with status or mission-critical information ready to be saved should the system power supply fail. The buffered data can then be transferred to the memory at the point of power fail.

To increase the amount of time that VDD is maintained to the DataFlash device after the power has failed, a capacitor and Schottky diode are used to store enough charge to allow the writing of critical data to the DataFlash device before VDD falls below the minimum requirement of 1.7V (applies to wide voltage range DataFlash products). Figure 1 shows a diagram of a test circuit that can be used to save critical data after a power failure.
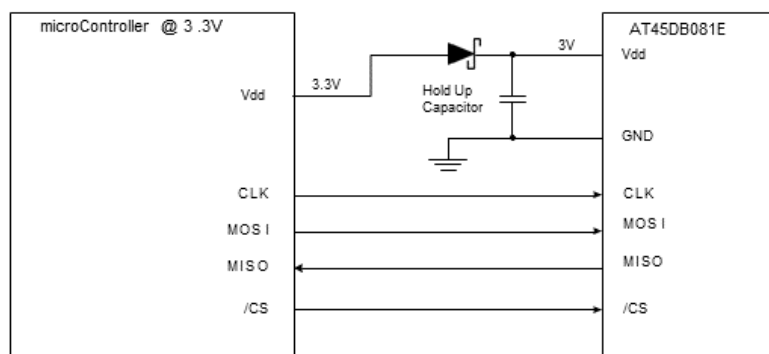

Figure 1: Test Circuit Block Diagram

2   DataFlash Overview

The AT45DB081E 8-Mbit Flash device was used to test the functionality described in this application note, although any member of the Dialog DataFlash E-series family can be used. Each member of the E-series DataFlash family contains the following logic blocks:
• Flash memory (8 Mbit in the AT45DB081E)
• Two independent SRAM's (each 264-bytes wide in the AT45DB081E)
• SPI I/O interface
• All necessary logic for executing SPI-based commands (see "Memory Transfer Command sequence" on page 7)

All of the memories (Flash/SRAM) in the AT45DB081E DataFlash device have a 264-byte page size. In this application note, one of the SRAM memories is used for situations such as a power failure, while the other SRAM is used during normal operation of the device.

Figure 2 shows a simplified block diagram of the AT45DB081E DataFlash device.
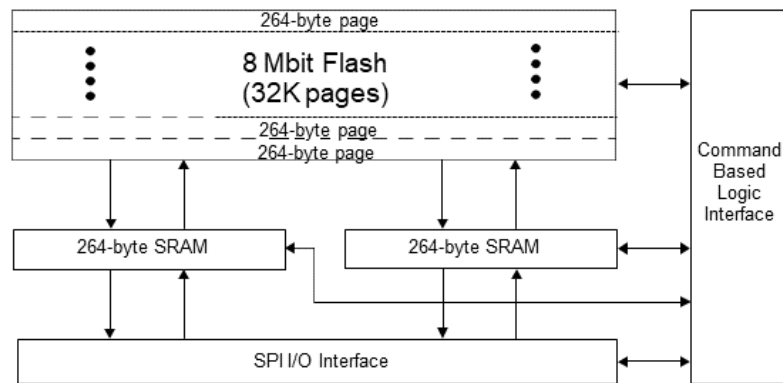
Figure 2: E-Series DataFlash Simplified Block Diagram

3   Test Circuit

In the test circuit in Figure 1, the Vdd pin of the microcontroller is connected to the power input of the DataFlash device through a Schottky diode and a capacitor. Once power to the DataFlash device fails, the capacitor continues to supply power for a finite period of time. The voltage at the Vdd pin falls from the 3V normal level, to 1.7V (minimum level required to maintain state) during this time.

After the power is removed the DataFlash device continues to function and data can be moved from the on-chip SRAM to Flash memory. The time until the voltage falls below the minimum level is a function of frequency and the size of the capacitor. As shown in Table 1, this time can be anywhere from 12ms to 20ms.

Note 1

3.1   Calculating the Capacitor Size

The examples in the following sections transfer 264 bytes of data from the SRAM data buffer to the Flash memory at different frequencies using different sizes of capacitors.

To determine the approximate capacitor size required to transfer this amount of data, it is necessary to determine the amount of charge that will be required. This can be done using the formula;

$Q = i * t$

where,
Q = the amount of charge in Coulombs
i = the amount of current required
t = the time required

In the following examples, 7 milliamps (mA) of current are required to transfer 264 bytes of data in 5 milliseconds (ms) at a frequency of 1 MHz followed by the programming cycle which takes 14mA for 2ms. Therefore,

$Q = 7mA \times 5mS + 14mA \times 2ms = 0.063mC$

As shown, 0.096 mC (milliCoulombs) of charge is required to transfer the required data. This value is then divided by the difference between the normal and minimum Vdd values. This is expressed in the following equation, where C is the capacitance required in microFarads.

$C = 0.096mC / (3.3 - 1.7) = 39.3 \ \mu F$

Application Note 114   **11-Dec-2020**

From this calculation, it is determined that a minimum capacitor size of 60uF is required to facilitate the data transfer of 264 bytes in less than 7ms. Therefore, both 100µF and 50µF capacitors are used in the following examples to compare datasheet values with measured values.

Note that it is taking 5ms to load the data at 1MHz. This 5ms can be substantially reduced by increasing the SPI clock frequency. The increase in frequency makes little difference to the current consumption of the DataFlash device but reduces the energy consumed during this period.

3.2   Diode Selection

As shown in Figure 1, a diode is placed in the path between the Vdd pins of the microcontroller and the DataFlash device. A Schottky diode was chosen due to its low voltage drop of 0.15 - 0.45 Volts, versus a drop of 0.6 - 0.7 Volts for a conventional diode, allowing for higher switching speeds and better system efficiency. The type and manufacturer of the diode is not critical, as long as it is a Schottky diode.

4   Test Results Overview

Table 1 shows the test results based on the following variables:
• Transfer type
• Capacitor size
• Clock speed
• Buffer pre-load status

Table 1: Test Results Based on Capacitor Size and Clock Speed

Condition	Transfer Type	Capacitor Size	Clock Speed	VDD Decay Time (3.3V -> 1.7V)	Time Required to Save 264 Bytes to DFLASH
Buffer pre-loaded with known data.	Without Erase	100 µF	1 MHz	>20 ms	1.5 ms
Buffer pre-loaded with known data.	With Erase	100 µF	1 MHz	>20 ms	7 ms
Buffer pre-loaded with known data.	Without Erase	50 µF	1 MHz	12 ms	1.5 ms
Buffer pre-loaded with known data.	With Erase	50 µF	1 MHz	12 ms	Fail
Load buffer with data, then write to Flash memory.	Without Erase	100 µF	1 MHz	>20 ms	7 ms
Load buffer with data, then write to Flash memory.	Without Erase	50 µF	2 MHz	12 ms	3.4 ms

Each of these variables is defined in the following subsections:

4.1   Transfer Type

Two transfer types are shown in Table 1, Without Erase and With Erase. This indicates whether the Flash memory portion of the DataFlash device is erased before the data in the buffer is written to the Flash memory. As shown, for the experiments that required the Flash memory to be erased first, the time required to save the 264 bytes is significantly longer. In the case using a 50uF capacitor where the Flash must be erased prior to writing, the operation cannot be completed before the voltage falls below the required threshold using a 1MHz clock.

4.2   Capacitor Size

Two capacitor sizes are shown in Table 1, 50uF and 100uF. Note that with the 50 ·F capacitor, the decay time is less, but still long enough to complete the transfer to the Flash memory in most cases.

4.3   Clock Speed

Clock speeds of 1MHz and 2MHz were used.

4.4 Decay Time

The decay is the time required for the voltage at the Vdd pin of the DataFlash device to fall below the 1.7V threshold required to complete the data transfer operation. Note that this time is directly proportional to the size of the capacitor used.

4.5 Write Time

The write time is the time required to move data into the Flash memory, either from the on-chip SRAM buffer, or from the microcontroller. In the last two entries of Table 1, data is not preloaded, but is moved into the SRAM, then to the Flash memory. In both cases, the write time is significantly less than the voltage delay time, and the time required to complete the operation depends on the size of the capacitor and the clock speed. In the last entry of Table 1, the voltage decay time is reduced due to the smaller capacitor size, but the actual transfer time is cut in half due to the faster clock speed. These two experiments are described in more detail in the following section.

5 Test Data

This section shows the actual test data for the experiments corresponding to the last two entries of Table 1. In both of these experiments, data is first transferred to the on-chip SRAM and then to Flash memory. In both cases it is assumed that the page has been previously erased. The main variables between these experiments are capacitor size and clock speed. In both cases, the data is placed into the top location of the Flash memory as described in the Command Sequence section below.

5.1 Experiment 1: 100µF / 1MHz, Load Buffer then Flash Memory without Erase

Figure 3 shows the oscilloscope output for the following conditions:
• 100 µF capacitor
• 1 MHz clock speed
• Load to buffer, then to Flash memory
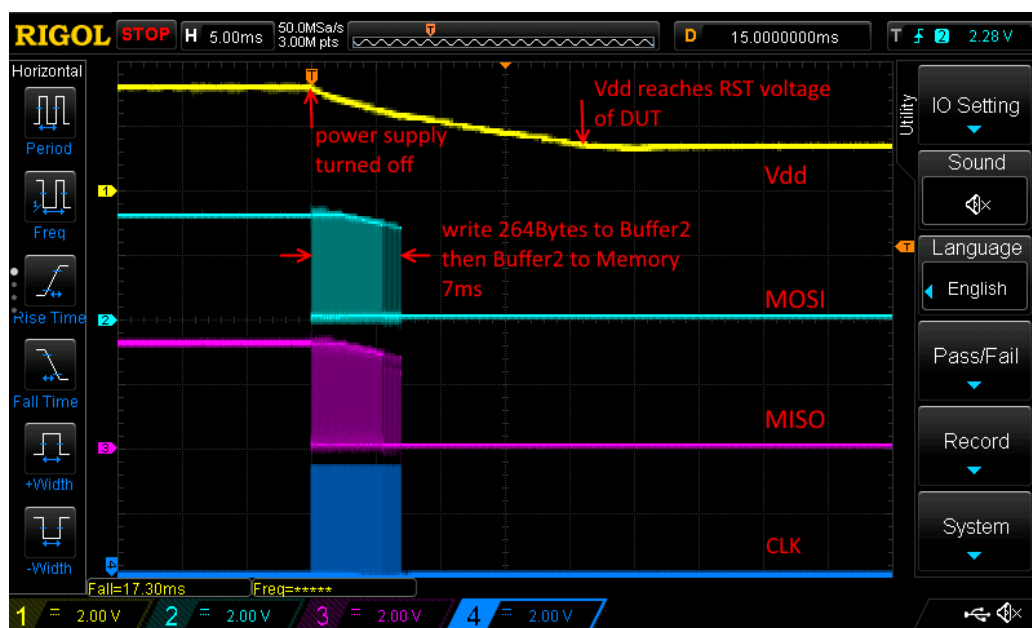• No erase operation performed on Flash memory prior to data transfer

Figure 3: Oscilloscope Output of Experiment 1

As shown above, once power from the microcontroller is removed, the 100 µF capacitor supplies power to the DataFlash device for approximately 20ms. During this time, commands are sent to the DataFlash device to initiate the transfer from the microcontroller to the on-chip SRAM, and from the SRAM to the Flash memory. The command and subsequent 264-byte data transfer takes 7ms, which is well within the maximum time of 20ms.

5.2   Experiment 2: 50 µF / 2 MHz, Load Buffer then Flash Memory without Erase

Figure 4 shows the oscilloscope output for the following conditions:

• 50 µF capacitor
• 2 MHz clock speed
• Load to buffer, then to Flash memory
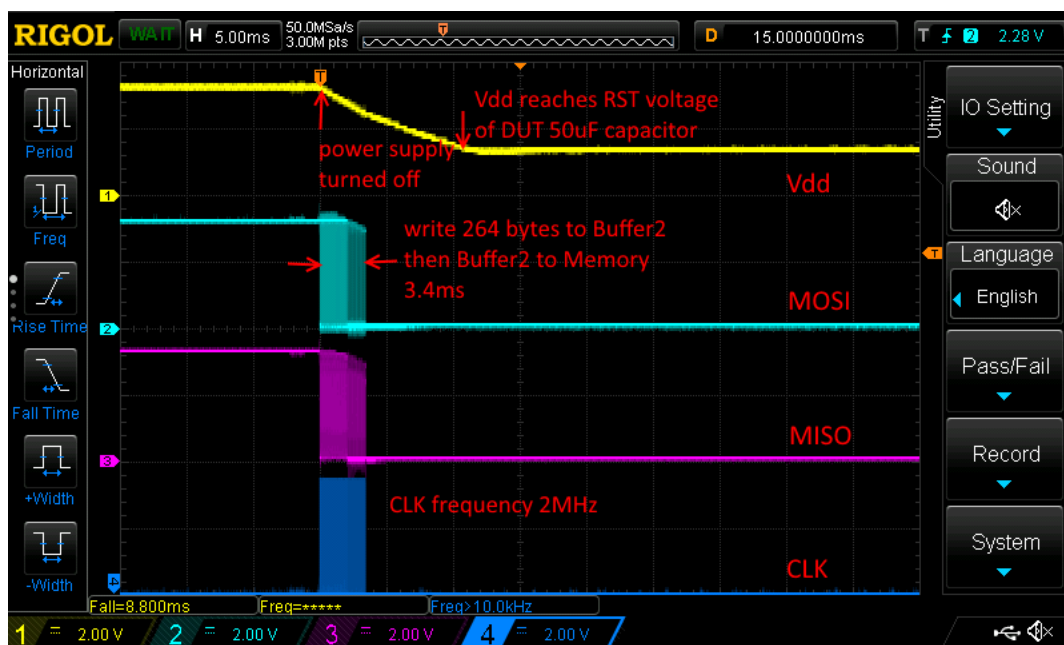• No erase operation performed on Flash memory prior to data transfer

Figure 4: Oscilloscope Output of Experiment 2

As shown above, once power from the microcontroller is removed, the 50 µF capacitor supplies power to the DataFlash device for 12ms. During this time, commands are sent to the DataFlash device to initiate the transfer from the microcontroller to the on-chip SRAM, and from the SRAM to the Flash memory. The command and subsequent 264-byte data transfer takes 3.4ms, which is well within the maximum time of 12ms.

In this experiment, the capacitor size is smaller than in experiment 1, resulting in a faster voltage delay time. However, the data transfer time is approximately twice as fast due to the increased clock speed. As a result, the overall data transfer occurs in approximately half the time as the previous experiment. Note that increasing the clock frequency does not have a big impact on the current consumption so that by using a faster clock, a smaller capacitor may be used, up to the 85 MHz maximum clock speed.

6 Memory Transfer Command Sequence

This section describes the command sequence that can be used to transfer the critical data from the microcontroller to Flash memory. The following sequence is intended to show proof of concept of how critical data can be saved in the event of a power failure.

The memory transfer command sequence is divided into the following steps. Many of these steps are optional and are shown for verification purposes. Each step is described in the following subsections.

Saving Critical Data
1. Remove power to DataFlash device
2. Move critical data to Buffer 2 SRAM
3. Initiate Buffer 2 to Flash Memory Page Program
4. Check for completion of transfer to Flash memory
5. Disable SPI interface

Verification
1. Restore power to DataFlash
2. Enable SPI interface
3. Refill Buffer 2 with critical data
4. Compare contents of Flash and Buffer 2 SRAM using the DataFlash device resources
5. Check for completion of compare
6. Alternatively compare contents of Flash and Buffer 2 SRAM using microcontroller resources

6.1 Remove Power to DataFlash Device

In this experiment, power to the DataFlash device was removed manually to simulate the power failure.

6.2 Move Critical Data to the Buffer 2 SRAM

As described in the section "DataFlash Overview" on page 2, the E-series devices have two independent SRAM memories that can be used for any purpose. In this example, one SRAM is used for normal operation, and one is used to store critical data that can be written to Flash in the event of a power failure. In this example, SRAM Buffer 2 is used to store the critical data.

To fill the Buffer 2 SRAM, a command of 87h is sent by the microcontroller over the SPI bus, followed by a 24-bit address (00_0000h). Once the SRAM buffer is addressed, 264 bytes are data are copied from the microcontroller to the SRAM.

Figure 5 shows the SPI command sequence for transferring the critical data to the Buffer 2 SRAM of the DataFlash device. In this figure, the high byte of address is driven onto the bus after the command.
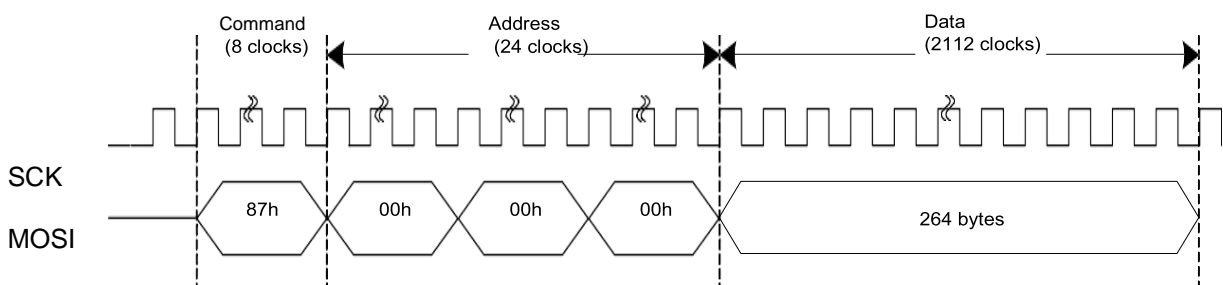

Figure 5: Transferring data to the Buffer 2 SRAM

6.3   Initiate Buffer 2 to Flash Memory Page Program

To transfer the critical data from the Buffer 2 SRAM to the Flash memory, the microcontroller sends command 89h*, Buffer 2 to Main Memory Page Program Without Built-in Erase*. This command moves the 264 bytes of critical data from the Buffer 2 SRAM to the Flash memory.

Figure 6 shows the command 89h and corresponding address being transferred across the SPI bus. In this example, the data is being placed in the uppermost address of the Flash memory. Since the Flash memory page size is 264 bytes in this example, only one 24-bit address is required. In this figure, the high byte of address is driven onto the bus after the command.

The AT45BD081E device contains 32,768 pages (32K), with 264 bytes per page. In this example, a 24-bit byte address of 1F_FE00h accesses the first byte of the uppermost page in memory. Since an entire page is written in this example, all 264 bytes of the page are updated during the memory transfer.
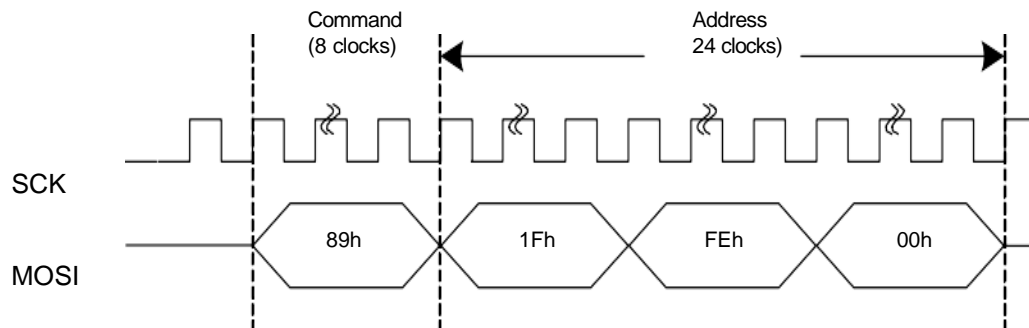

Figure 6: SRAM Buffer 2 to Flash Memory Transfer

Once the command is sent to the DataFlash device, the data transfer is performed by the device itself. No further action by the microcontroller is required.

6.4   Check for Completion of Flash Memory Transfer

As an optional step, the microcontroller can check the progress of the data transfer to the Flash memory. This is done by accessing the *Status* register of the DataFlash device. For the purposes of this example, this was done to verify that the memory transfer was completed successfully.

To read the *Status* register, the active-low /CS pin must first be asserted and then the opcode D7h must be clocked into the device. After the opcode has been clocked in, the device begins outputting *Status* register data on the MISO pin during every subsequent clock cycle. After the second byte of the *Status* register has been clocked out, the sequence repeats itself, starting again with the first byte of the *Status* register, as long as the / CS pin remains asserted and the clock pin is being pulsed. The data in the Status register is constantly being updated, so each repeating sequence may output new data.

Note that the RDY/BUSY status appears on both bits 15 and 7 of the *Status* register output. When these bits are low, the internal transfer is in progress. When either of these bits is high, the operation has completed. Deasserting the /CS pin terminates the *Status* register read operation and places the MISO pin into a high impedance state.

Figure 7 shows a *Status* register read. In this diagram, the entire contents of the register are driven onto the MISO pin four times. The first three times the RDY/BUSY bits are zero, indicating the operation is in progress. The fourth time the RDY/BUSY bits are one, indicating the operation is complete. Once the microcontroller recognizes bits 15 or 7 as being high, the /CS pin is driven high to disable the SPI interface.
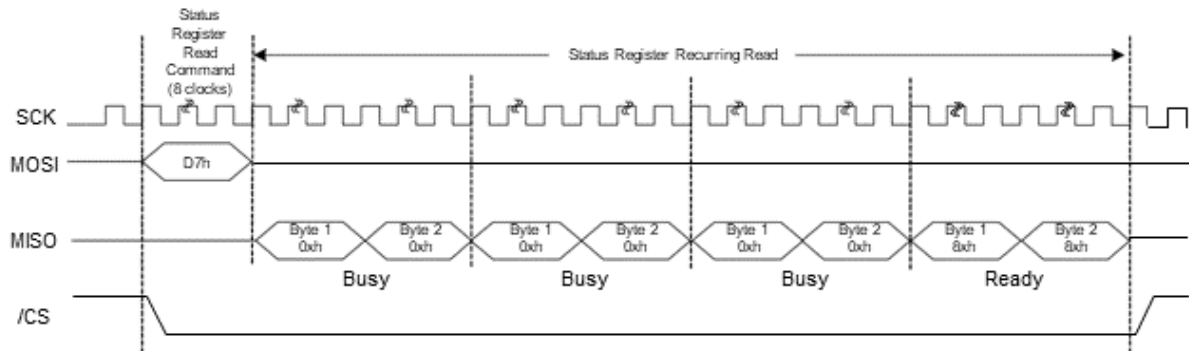
Figure 7: Reading the Status Register

6.5   Disable SPI Interface

The SPI interface on the DataFlash device is disabled when all SPI pins are driven low due to the power being removed.

6.6   Restore Power and Enable the SPI Interface

In this example, a few seconds elapsed before power to the DataFlash device was reapplied. The SPI interface in the microcontroller is then reinitialized (procedure not covered in this document). This allows for subsequent communication with the DataFlash to verify a successful storage of the critical data as described below.

6.7   Move Critical Data to the Buffer 2 SRAM

To validate the experiment, after power is restored to the DataFlash, the critical data is again moved into the Buffer 2 SRAM to be compared with the contents of the Flash memory. In this example, the same critical-data pattern was copied to the Buffer 2 SRAM by the microcontroller.

To fill the Buffer 2 SRAM, a command of 87h is sent by the microcontroller over the SPI bus, followed by a 24- bit address (00_0000h). Once the SRAM buffer is addressed, 264 bytes are data are copied from the microcontroller to the SRAM. This process is identical to that described in "Move Critical Data to the Buffer 2 SRAM" on page 7.

6.8   Read and Compare Critical Data from the Flash memory

There are two ways to verify that the critical data was saved in Flash memory.
• Compare contents of Buffer 2 SRAM and Flash memory using DataFlash resources
• Compare contents of Buffer 2 SRAM and Flash memory using Microcontroller resources

6.9   Compare Contents of Buffer 2 SRAM and Flash Memory Using DataFlash Resources

To verify that the critical data was successfully transferred to the Flash memory after the power to the DataFlash device was removed, the contents of the corresponding page in the Flash memory can be compared to the original critical data. In this example, the compare was done using resources contained within the DataFlash device.

Once power has been restored to the DataFlash device and the SPI interface enabled, the critical data was again written from the microcontroller to the Buffer 2 SRAM using the exact same procedure as described above.

Once the Buffer 2 SRAM has been filled, the compare operation can be performed by executing command 61h, *Main Memory Page to Buffer 2 Compare*.

Figure 8 shows the transfer of command   61h used to perform the compare. In this example, the data was placed in the uppermost address of the Flash memory. Since the Flash memory page size is 264 bytes in this example, only one 24-bit address is required.
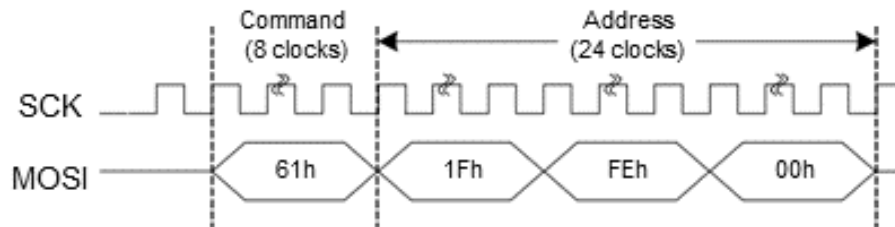

Figure 8: Flash Memory Page to Buffer 2 Compare

In this operation, the command 61h causes the DataFlash device to fetch the data from Flash memory at the address indicated by the incoming address field, then compare those contents to that in Buffer 2 SRAM.

6.10   Check for Completion of Compare Operation

The microcontroller can check the progress and completion of the compare operation by reading the *Status* register. This process is similar to the that described in the previous subsection, "Initiate Buffer 2 to Flash Memory Page Program" on page 8. The microcontroller sends the D7h command on the SPI bus and reads the *Status* register output until either of the RDY/BUSY bits (15 or 7) are one, indicating the operation is complete. Note that the state of the *Status* register is continually output onto the SPI bus as shown in Figure 7.

At this point the microcontroller also reads bit 6 of the *Status* register output (COMP bit) to determine if the compare passed of failed. If this bit is set, the compare has failed. If the bit is cleared, the compare was successful.

6.11   Alternative: Compare Contents of Buffer 2 SRAM and Flash Memory Using Microcontroller Resources

As an alternative to the compare operation described in the previous subsection, the microcontroller can also fetch the data from Flash memory and copy it across the SPI bus, then perform the compare operation inside the microcontroller.

This would result in a read of the appropriate location in the Flash using command D2h, *Main Memory Page Read*. Data is copied from the Flash at the address provided and sent to the microcontroller over the SPI bus. Figure 9 shows a page read of the Flash memory. In this example the 264-byte page is located at the top of the Flash memory array.
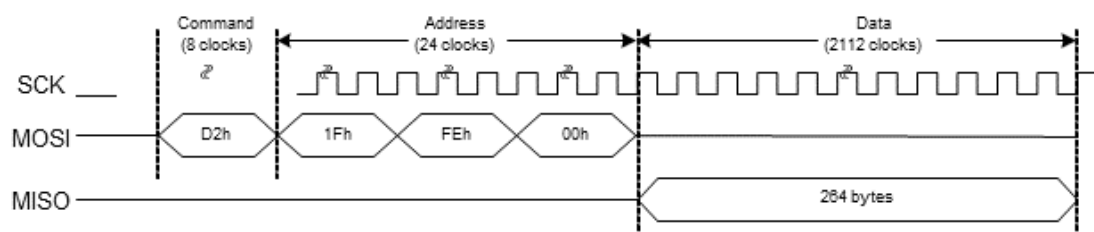

Figure 9: Flash Memory Page Read

Conclusions

This document describes how the Dialog DataFlash family of serial Flash devices can be used to save critical data in the event of a power failure. Using minimal hardware (capacitor and diode), a sequence of steps can be taken to save the critical data once the power fails. This technology can be used in a wide array of applications, including power meter failure, enterprise system power failure, data logger power failure, drone black box recorder, and any other application that needs to log critical data in the event of a power failure.

Appendix A

Assuming constant current:

$$i_c = -I$$

$$\int_{V0}^{V1} v_c(t)dt = \frac{1}{C}\int_0^t i_c dt$$

$$V_1 - V_0 = -\frac{I}{C}t \Rightarrow C(V_0 - V_1) = I \cdot t$$

$$C = \frac{I \cdot t}{V_0 - V_1}$$

More generically:

$$C = \frac{\sum_k I_k \cdot t_k}{V_0 - V_1}$$

For $V_0 = 3.3V, V_1 = 1.7V$

$$C = \frac{\sum_k I_k \cdot t_k}{3.3 - 1.7} = \frac{\sum_k I_k \cdot t_k}{1.6}$$

Disclaimer

Information in this document is believed to be accurate and reliable. However, Dialog Semiconductor does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information. Dialog Semiconductor furthermore takes no responsibility whatsoever for the content in this document if provided by any information source outside of Dialog Semiconductor.

Dialog Semiconductor reserves the right to change without notice the information published in this document, including without limitation the specification and the design of the related semiconductor products, software and applications.

Applications, software, and semiconductor products described in this document are for illustrative purposes only. Dialog Semiconductor makes no representation or warranty that such applications, software and semiconductor products will be suitable for the specified use without further testing or modification. Unless otherwise agreed in writing, such testing or modification is the sole responsibility of the customer and Dialog Semiconductor excludes all liability in this respect.

Customer notes that nothing in this document may be construed as a license for customer to use the Dialog Semiconductor products, software and applications referred to in this document. Such license must be separately sought by customer with Dialog Semiconductor.

All use of Dialog Semiconductor products, software and applications referred to in this document are subject to Dialog Semiconductor's Standard Terms and Conditions of Sale, available on the company website (www.dialog-semiconductor.com) unless otherwise stated.

Dialog and the Dialog logo are trademarks of Dialog Semiconductor plc or its subsidiaries. All other product or service names are the property of their respective owners.

Contacting Dialog Semiconductor

Application Note 114   **11-Dec-2020**