Cactus Technologies® White Paper

CTWP005: Write Abort Handling for Cactus Technologies® Industrial-Grade Flash-Storage Products

Covered Products: -203,-303,-503 CF cards, -900S SATA products, -806,-808 SD cards, -300 USB products

1 Introduction

Cactus Technologies® industrial-grade flash storage products are designed to operate in environments where temperature, shock, vibration and voltage fluctuations occur often and can cause abnormal operations on the device. The special features of the on-board intelligent flash controller for Cactus Technologies® flash storage products can help to prevent these extreme conditions from causing drive corruption.

1.1 The Problem

Many industrial computers and embedded systems often experience unexpected power outages, surges, spikes, sags or brownouts. Sometimes the device is manually removed from the system. These can cause data and disk corruption, and in turn lead to field failures and product returns.

When power is unexpectedly removed during idle or read operation, no data loss will occur. Data losses can only occur when power is unexpectedly removed, either by power outage or by manual device removal, during write operations. This is known as *Write Abort*. If the write operation is "aborted" during flash erase operations, data losses beyond the data pending for write may occur.

1.2 Failure Symptoms

There are, in general, two types of failure symptoms that the user may come across when the storage device is affected by sudden power failures. These are 1) file system corruptions and 2) internal device data corruption.

File system corruptions are the result of the operating system not being able to completely update the file system records. Some file systems are more vulnerable to corruption issues than others. For example, a journaling EXT3 or EXT4 file system is much more robust against corruption when compared to a FAT file system. Some examples of FAT file system corruption includes corrupted FAT tables or lost clusters.

It should be noted that file system corruptions cannot be prevented by the storage device's firmware or hardware design. This is because the storage device is designed to be independent of the host system specifics, it has no knowledge of what file system is being used and does not know where the file system records are located.

Luckily, file system corruptions are generally not fatal. Most operating systems will perform a file system repair operation on the next power up after a sudden power lost event. Alternately, the user can run a command or utility to perform the repair operation.

Internal storage device data corruptions, on the other hand, can make the whole device unusable. This type of error is the result of the device's internal management metadata (such as logical to physical mapping table) getting corrupted during a sudden power loss event. When this happens, either existing data may get corrupted or the drive becomes totally unrecognized by the host. The only way to recover from such a failure is to perform a low-level format of the device, thus resulting in a loss of user data.

Since internal data corruption is fatal, it is the focus of the storage device's firmware and hardware design to ensure that internal data corruption does not occur under any circumstances.

In the following sections, we will explain some of the key features in Cactus Technologies® Industrial grade flash products that mitigates and/or eliminates Write Abort data corruption problems.

2 NAND Flash Basics

2.1 NAND Flash Structure

NAND flash is internally organized in blocks and pages. Each block contains multiple pages, depending on the device capacity; a typical number is 64 to 256 pages per block. Each page has a fixed size; current SLC NAND used in Cactus Technologies® industrial grade products has a page size of either 2KB or 4KB, depending on device capacity. NAND flash is programmed in pages but can be erased only on a block basis; thus, in order to overwrite data in a page, it is necessary to perform a read/modify/write operation, this may involve first erasing another block that contains old data and is marked for recycling.

2.2 Logical To Physical Mapping

Host system access the storage device using Logical Block Addressing (LBA), each LBA is also referred to as a sector. At the device level, the LBAs are mapped to a physical block location; this is known as Logical to Physical mapping. This mapping is maintained in a table (log2phys); every time data is written to the storage device, the log2phys table needs to be updated. Since updating the log2phys table is a time-consuming process, the changes are not written to the table immediately; instead, a separate logbook is used to log all pending changes to the log2phys table. When the logbook is full, the updates are written to the log2phys table and a new logbook is started.

3 Safe Power Loss Protection Features

Cactus Technologies® products utilize a patented firmware algorithm in order to ensure data integrity when transferring or writing data. Recognizing that write abort events will occur occasionally, the emphasis on the firmware design is to ensure that the device can restart and operate correctly after a write abort event has occurred. This is accomplished by internal voltage detector that detects loss of power, coupled with a robust, safe power loss protection algorithm.

3.1 Power Fail Detection

The controllers used in Cactus Technologies® industrial grade products contain an internal voltage detector. When the power supply level drops below 2.5V, the flash devices are immediate put into write protect mode by pulling their Write Protect pin low. This minimizes any further data corruption in the flash devices.

3.2 Shadow Block

When data needs to be written to a flash block, the device firmware will map an additional shadow block to the physical block. Thus, a logical block is temporarily mapped to two physical blocks. New data is written to the shadow block while the original data remains untouched. Eventually, under certain safe conditions, the device firmware will resolve and merge the data in the two physical blocks, thus returning to a one to one logical to physical mapping again.

By keeping an old copy of the previous known good data, firmware can recover to a previous known good state in the event that the current write commits are corrupted by a sudden power loss event.

3.3 Fail Safe Write Operation

In Cactus Technologies® industrial grade products, data is written in a way that minimizes the delta between an old and a new state. The data system is coherent at all times. Utilizing the logbook and the shadow blocks, should the last entry of the log be corrupted, the controller recovers the last valid entry. The device is then able to recover to the last known good state; this is possible because the original data is still not erased. This process minimizes data loss due to power failures. Should power loss happen at the very same time when data is written to the flash, this data in transit might get lost. In no case, however, will the overall data system be corrupted. In summary, the process of writing new data to the device is as follows:

1. accept data from host
2. transfer data to internal RAM buffer
3. firmware creates shadow block for the target LBA
4. firmware creates entry in logbook to update log2phys table later
5. program data into shadow block

6. wait for safe condition, then merge shadow block with original physical block

Note that the log2phys table is written using the same shadow block methodology; thus, in the event that power is lost while this table is being written, the last known good copy is always available. The logbook is written page by page; if power is lost while the logbook is being updated, the last entry may be corrupted. In this case, the firmware will simply ignore this entry; this means the last operation will not take place but system integrity is still maintained.

When the logbook is full, the transaction records will be transferred to the log2phys table and a commit record will be written to dedicated commit blocks. Only when the commit record is written and verified will the old log2phys table be marked for recycling and put into the erase block pool and a new logbook started. If the commit record is corrupted due to a write abort event, the old log2phys table and logbook will be re-used, thus ensuring that device can recover to its last known good state.

Extensive power cycling tests have been performed on this algorithm to verify that no data loss will occur due to sudden power failures.

3.4 Redundant Firmware

This feature is available only on -503 CF cards, -900S SATA products, -808 SD cards and -300 UFD products. The controllers used in Cactus Technologies® industrial grade products contain an internal boot ROM. Multiple copies of the firmware are stored in different, fixed locations; these files are protected by a unique signature and CRC checksum. During boot process, the boot ROM will search for firmware files at the fixed locations and check their validity using the signature and checksum. Invalid firmware files are not used and will be updated with the valid copy.

3.5 Disable Use of Early Write Acknowledge

In order to improve write performance, some vendor products choose to return a ready status to the host as soon as data has been transferred to the internal data buffers. This data has not been written to flash memory yet and can be lost if there is a sudden power lost. There is no protocol for the storage device to notify the host that such data has not been successfully written to the flash media. Therefore, use of early write acknowledge is risky if power integrity cannot be guaranteed.

Cactus Technologies® industrial grade products do not use early write acknowledge; all host data are written to flash media before firmware returns ready status to the host.

3.6 Minimize Use of External DRAM Cache

Many of our competitor's products utilize extensive DRAM caching to improve performance. Typically, the DRAM cache stores lookup tables and other internal metadata but in many cases, it also stores user data to be written to flash memory. In the event of a sudden power lose, the data stored in these DRAM caches are vulnerable and can cause significant data integrity loss. In the worst-case scenario, the device's internal metadata is corrupted and the device will no longer be recognized by the host system and the only way to recover is to do a low level format on the device, causing all previously stored data to be lost.

Cactus Technologies® recognizes the vulnerability of using external DRAM cache, therefore, we intentionally try to avoid using DRAM caching wherever we can. All our -203, -303, -503 CF products, our -806, -808 SD products, our -900S SATA products and our -300 UFD products do not use external DRAM caching.

Cactus Technologies® recognizes that the lack of DRAM caching results in lower read/write performance of our products. However, we view the superior robustness against write abort data protection as a higher priority, particularly in industrial applications where high reliability is generally more important than high performance.

3.7 No Internal Power Backup

Some vendor products try to compensate for the vulnerability of DRAM cache by adding internal power backup capacitors. As most Cactus Technologies® products do not use DRAM cache, there is no need for us to add backup capacitors. Furthermore, we have found that these backup capacitors are general ineffective in preventing data corruption during a write abort event; this is because the duration of backup power provided by such capacitors are generally not long enough to guarantee that all critical data is completely written to the flash media. This is particularly so for small form factor products such as CF cards, SD cards and UFD devices, where there isn't room to add large capacitors.

An additional complexity of using large backup capacitors in products such as SSDs is that the start up time is often increased, this is due to the need to charge up the backup capacitors. This longer startup time can sometimes cause the drive to be not detected by the host system during bootup time. Changing the startup time delay may not be an option for systems that are already out in the field.

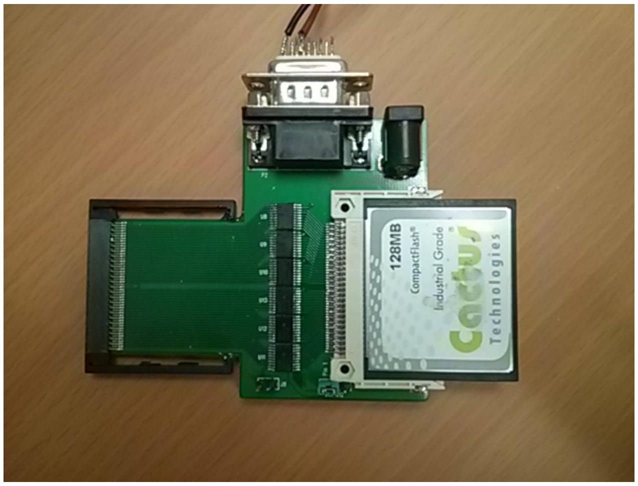
Cactus Technologies® has conducted power cycling tests on some vendor products that contain internal backup capacitors and found that they do not completely prevent data corruption during write abort events. Thus, the use of internal backup capacitors can create a false sense of security against write abort data corruption.

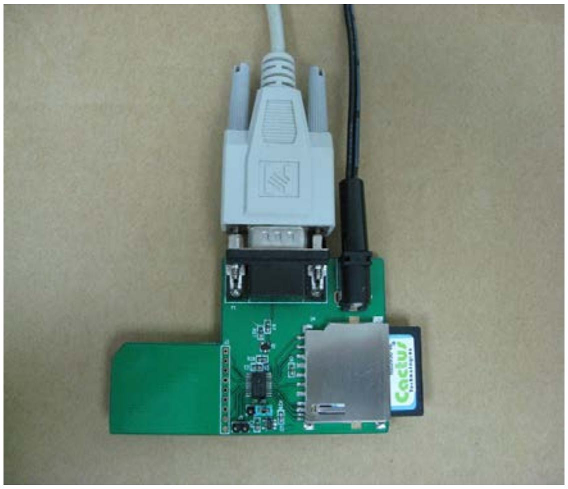
The power backup switch over circuit must also be properly designed, otherwise it creates more problems that it is intended to solve.

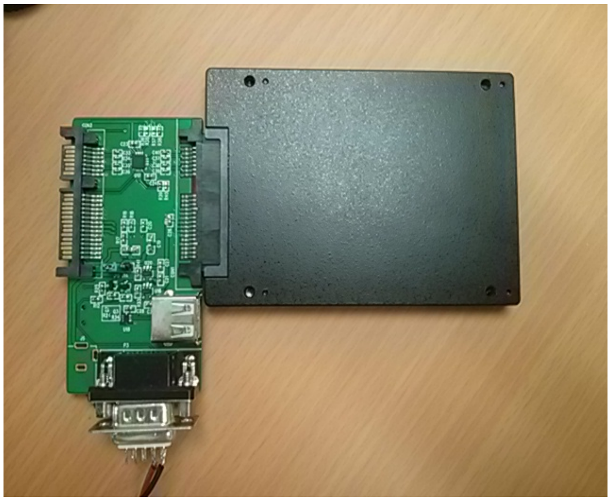
Here at Cactus Technologies®, we have concluded that robust firmware design is a far more effective approach to maintaining data integrity during write abort events.

3.8 Rigorous Testing

In addition to the firmware features discussed above, Cactus Technologies® has built test fixtures and conducted extensive power cycling tests to verify that our products is robust against data corruption in a sudden power fail scenario. The pictures below show our test fixtures for testing SD Cards, CF cards and SATA SSDs and CFast cards:







The test fixture has a DC-DC controller and a bus isolation circuit, and is controlled via a serial port connected to a PC. Special test code written by Cactus Technologies® randomly shuts off power and isolates the interface bus from the DUT. This ensures that the DUT is not powered via interface bus signals. A typical test sequence is as follows:

1.Pre-copy some directories and static data to the DUT.

2.Verify directories and data in DUT.

3 If some corrupted files in directory “\FOUND.000”, delete them, as these are results of file system repair (as discussed in Section 1).

4.Start a random timer to disconnect the DUT.

5 Write a large file (File A) to DUT.

6.Delete “File A” in DUT

7.Write another large file (File B) to DUT

8.Delete “File B” in DUT

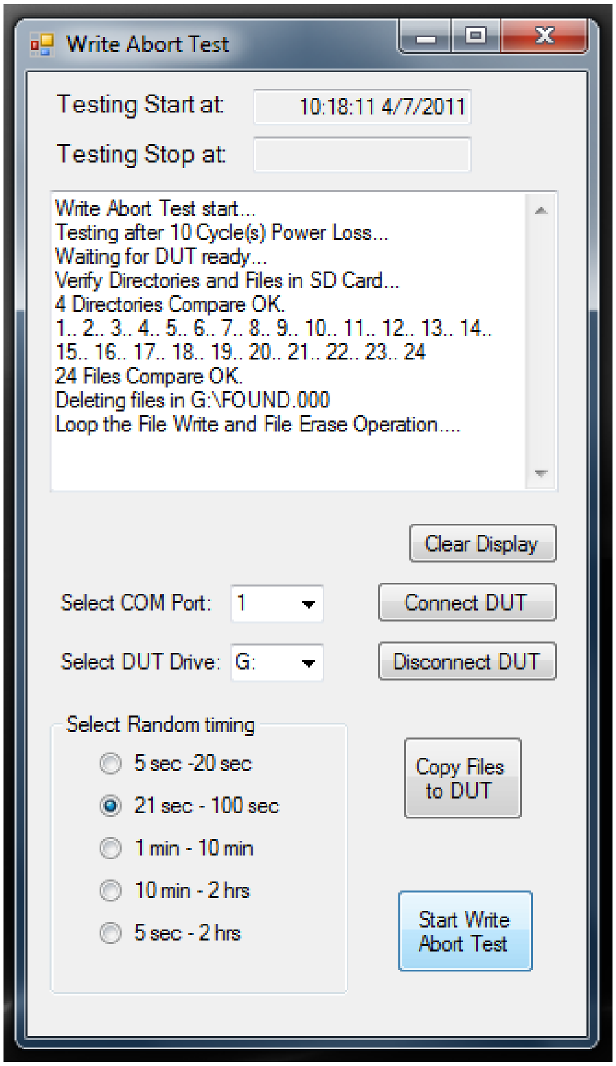
9.GOTO step 5. Repeat loop until the random timer activated.

10.The random timer disconnect the DUT, Wait 3 sec and re-connect the DUT.

11.Scan and repair corrupted files. This is standard OS file system process.

12.GOTO steps 2 and repeat loop.

The following picture is a screen capture showing a typical test run:



Testing Start at: 10:18:11 4/7/2011

Testing Stop at:

```
Write Abort Test start...
Testing after 10 Cycle(s) Power Loss...
Waiting for DUT ready...
Verify Directories and Files in SD Card...
4 Directories Compare OK.
1.. 2.. 3.. 4.. 5.. 6.. 7.. 8.. 9.. 10.. 11.. 12.. 13.. 14..
15.. 16.. 17.. 18.. 19.. 20.. 21.. 22.. 23.. 24
24 Files Compare OK.
Deleting files in G:\FOUND.000
Loop the File Write and File Erase Operation....
```

Select COM Port: 1

Select DUT Drive: G:

Select Random timing

○ 5 sec -20 sec
● 21 sec - 100 sec
○ 1 min - 10 min
○ 10 min - 2 hrs
○ 5 sec - 2 hrs

Cactus Technologies® performs ongoing testing of new controllers and firmware to ensure data protection against sudden power fail is not compromised.

4 Host Design Considerations

Even with the robust firmware design features and rigorous testing noted above, it is impossible to guarantee that no data corruption will ever happen due to a write abort event. Therefore, we strongly recommend host systems to have some sort of data verification scheme, such as using MD5 or similar checksum method or by comparing the file against a known good source on the host. If the verification fails, the host should re-write the data to the device or perform other error-recovery steps to ensure data integrity.

If the system design allows, adding backup power supply and designing the system to avoid manual device removal can also prevent write abort from occurring in the first place.

4.1 Version History

Version	***Date***	***Change***
1.01	October 9, 2006	Initial Version
2.0	September 25, 2013	Rewrite
3.0	Oct. 31, 2016	Rewrite and update