Non-Volatile CACHE for Host-Based RAID Controllers

A Dell Technical White Paper

Bill Lynn

Storage Advanced Engineering

Ansh Gupta

RAID Hardware Engineering

THIS WHITE PAPER IS FOR INFORMATIONAL PURPOSES ONLY, AND MAY CONTAIN TYPOGRAPHICAL ERRORS AND TECHNICAL INACCURACIES. THE CONTENT IS PROVIDED AS IS, WITHOUT EXPRESS OR IMPLIED WARRANTIES OF ANY KIND.

© 2010 Dell Inc. All rights reserved. Reproduction of this material in any manner whatsoever without the express written permission of Dell Inc. is strictly forbidden. For more information, contact Dell.

Dell, the *DELL* logo, the *DELL* badge, and PowerEdge are trademarks of Dell Inc. Other trademarks and trade names may be used in this document to refer to either the entities claiming the marks and names or their products. Dell Inc. disclaims any proprietary interest in trademarks and trade names other than its own.

Revision 2 January 2011

Contents

Figures

Introduction

Data protection is an absolute requirement in any enterprise-class storage system. To achieve a high degree of data protection, system administrators use a technology called Redundant Array of Independent Disks (RAID) to protect the data stored on their enterprise servers. RAID distributes data across multiple disk drives in such a manner that if a single disk fails the data can be recovered from the remaining disks. With RAID a disk failure does not result in the loss of user data.

A second requirement of an enterprise-class storage system is high performance. Servers must be able to access data as quickly as possible. Since server processors run thousands of times faster than the disk drives attached to them, RAID controllers use a small amount of high-performance memory called DRAM to hold data until it can be written to the disk drives. This memory is known as a cache. RAID controllers place data intended to be written on the disks in the cache until the disk drives are ready and the data can be written to the disk. This process is known as “write caching.”

There are two types of write caching, write-through and write-back. In write-through caching the data is transferred quickly to the cache memory, but the RAID controller does not acknowledge that the write-through is complete until the data is written to disk. This is a slow process since the RAID controller must wait for the disk drives to become ready and write the data before it can respond to the server that the data transfer has been completed. In write-back caching the RAID controller responds with complete command as soon as the data is transferred into the cache memory. From the server’s point of view the data transfer completes very quickly and the server processor can continue processing additional commands. The RAID controller can then complete the data transfer at a later time. This increases server storage performance but poses the problem of the data being at risk while it is being stored in the cache. If the server suffers an unexpected power loss then the data in the cache could be lost.

This paper discusses a method of protecting the DRAM cache memory of a RAID controller using non-volatile flash memory and a power source to transfer the cache data from the DRAM memory to the flash memory in the event of an unexpected power loss. The cache data is then restored from the flash memory to the DRAM cache memory upon the next power cycle. This method is known as a Cache-to-Flash Non-volatile Cache, or CTF NVCACHE.

Battery-Backed Cache

The concept of a write-back cache is not new and has been implemented on several generations of RAID controllers. Historically the DRAM cache memory has been protected by using a battery to provide a back-up power source to the memory in the event of an unexpected power loss. In this case, the DRAM memory is placed in a low-power self-refresh mode, and the battery provides power to maintain the data.

There are several limitations with battery-backed cache implementations. The major limitation is the fact that the cache data is maintained only as long as the battery is able to supply power. If the battery becomes fully discharged, the cache data is lost. The amount of time that the data can be maintained is dependent upon the capacity of the battery and the size of the memory. Traditionally the goal is to maintain cache data for at least 72 hours (Dell guarantees that the BBU will retain data in cache for 24 hours, which is also an industry standard). With servers moving into the Small/Medium Business (SMB) space and remote locations, 24 hours may no longer be adequate to guarantee a timely service call.

If the amount of time or the size of the memory is increased, the battery capacity must increase. As the capacity of a battery increases, the battery size increases and in some cases can become quite large. Also as the size of the battery increases the amount of time required to charge the battery increases and may take up to several hours. During the charge time the server should not use the write-back cache feature and the overall server performance will suffer.

Another limitation of battery-backed designs is that the memory is tethered to the battery. If the battery is disconnected, the data is lost. This presents difficulties if one wants to transfer the RAID controller from a failed server to a replacement server. In this case the battery and the RAID controller must be moved as a single connected unit.

As Moore's law predicts, the performance and capacity of a RAID controller increases with each generation. Cache memory size has increased from 256MB to 512MB to 1GB, and the next generation is moving to 2GB. This increasing cache memory size is making necessary larger and larger battery sizes. The cache battery size and the placement of the battery has now become a significant hurdle in new server designs.

CTF NVCACHE

Over the past several years, non-volatile flash memory technology has made tremendous gains. Increases in speed, density, and reliability coupled with decreases in cost and size have made the use of flash memory viable in enterprise-class storage system. Solid State Disk Drives (SSDs) are now becoming mainstream storage devices not only for notebook computers, but for enterprise servers as well. It is the increase in speed and density along with the decreasing cost of flash memory that make the CTF NVCACHE implementation viable.

Figure 1 shows a block diagram of the CTF NVCACHE implementation.

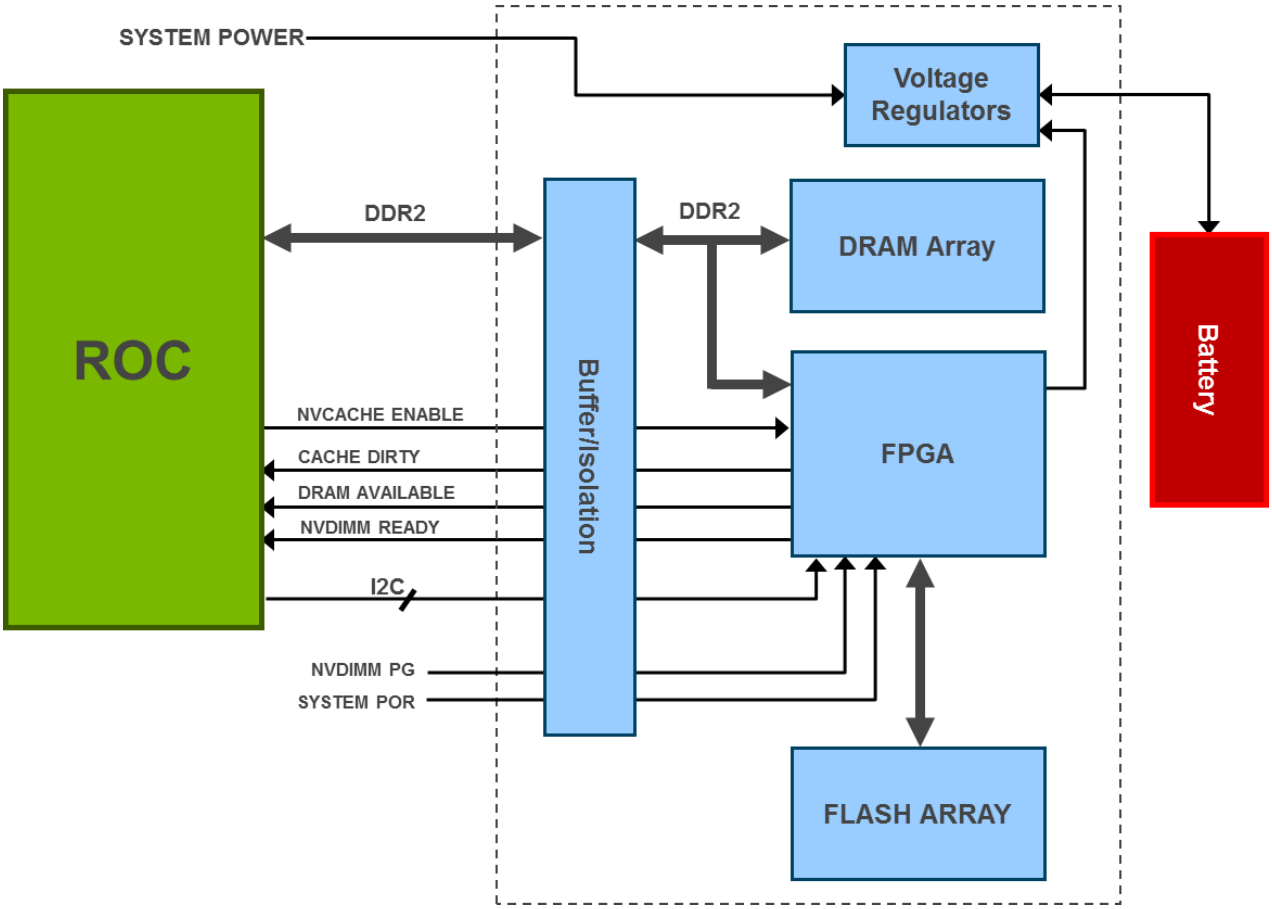

Figure 1. CTF NVCACHE Block Diagram

As shown in Figure 1, cache data is stored in the DRAM array during normal operation. A Field Programmable Gate Array (FPGA) is used to monitor the system power and in the case of an unexpected power failure, transfer the cache data from the DRAM array to the flash array. A small battery or super capacitor is used to supply power until the transfer is complete. Once the transfer is complete the FPGA shuts off power and the cache data is maintained, essentially indefinitely, until power is restored. When power is restored, the FPGA restores the cache data from the flash array to the DRAM array and notifies the RAID controller there is cache data that needs to be written to the disk drives.

NVCACHE Control Signals

The FPGA is connected to the DRAM memory bus and uses a series of control signals to enable and control the CTF operation. These signals include NVCACHE Ready, DRAM Available, NVCACHE Enable, Cache Dirty, and NVCACHE PG (Power Good). An I2C interface is also used by the RAID controller to configure and monitor the FPGA.

NVCACHE Ready is an output signal from the FPGA that indicates that the NVCACHE subsystem is ready to be put into write-back cache mode. The NVCACHE Ready signal indicates that the flash memory components have been erased, initialized, and are ready for cache data transfer from the DRAM to the flash. NVCACHE Ready only represents the readiness of the FPGA and flash array. The battery is

monitored by the RAID controller. At this point the NVCACHE is referred to as being “clean” (no cache data has been written to the flash array). The RAID controller may not go into write-back cache mode until NVCACHE Ready is asserted and the NVCACHE is clean.

DRAM Available is an output signal from the FPGA that indicates that the DRAM array is available to be read from or written to by the RAID controller. The RAID controller may not access the DRAM until DRAM Available is asserted. At power up the FPGA will hold DRAM Available de-asserted until it can determine the state of the flash array. If the flash array is clean then the FPGA will assert DRAM Available and the RAID controller will continue with normal operations. If the FPGA determines that flash array contains valid cache data the FPGA will restore the contents of the flash array to the DRAM array. DRAM memory that contains valid cache data is known as a dirty cache. Once the restore operation is complete the FPGA will assert DRAM Available and the RAID controller will flush or write the cache data to the disk drives.

NVCACHE Enable is an input signal to the FPGA from the RAID controller indicating that the system has entered write-back cache mode and that the RAID controller will start writing cache data to the DRAM cache. At this point the cache memory is referred to as being “dirty”, meaning that there is valid data in the cache memory. In normal operation the RAID controller may not go into write-back mode and assert NVCACHE_ENABLE until it has detected both DRAM Available and NVCACHE Ready.

If the NVCACHE is clean at power up, the NVCACHE Enable signal is used to instruct the FPGA to enter NVCACHE (Write Back) mode and transfer the data in the DRAM array to the flash array in the event of an unexpected power loss. If the NVCACHE is dirty at power up, the FPGA is not allowed to erase and initialize the flash array until NVCACHE Enable is toggled.

During power up the NVCACHE signal shall be de-asserted by HW on the RAID controller. The RAID controller will then determine the state of the NVCACHE (clean or dirty). If the NVCACHE is clean the RAID controller shall continue with normal operation. If the NVCACHE is dirty the RAID controller may not assert NVCACHE Enable until all dirty data has been written to disk and the controller has gone into write-through cache mode. Once the data has been written to disk, the RAID controller will generate a pulse of 500us on NVCACHE Enable signaling FPGA to erase and initialize the flash and assert NVCACHE Ready. Once the flash array is erased and initialized, the FPGA shall assert NVCACHE Ready indicating to the RAID controller that it is ready to enter NVCACHE mode.

Cache Dirty is an output signal from the FPGA to the RAID controller that is used to indicate that the FPGA has entered non-volatile mode and there is valid or dirty data in the cache. In normal operation, this signal is asserted by the FPGA once NVCACHE Enable is detected. During power-up, the FPGA will read a predetermined location in the flash array to determine if there is dirty data in the flash and set the Cache Dirty signal appropriately.

NVCACHE PG is an input signal to the FPGA from the RAID subsystem that indicates that the power supplied to the NVCACHE subsystem is good. During a dirty power down or a power loss event, the RAID controller will put the DRAM array into self-refresh mode before de-asserting NVCACHE PG. The NVCACHE PG signal is also used to switch the NVCACHE subsystem from normal power to battery or super capacitor power.

NVCACHE State Diagram

The FPGA implements what is known as a “finite state machine.” In a finite state machine the behavior of the system is defined by a series of states in which the system can exist and how the system moves from state to state. Within a state the system can perform one or more tasks that are specific to that state.

Seven major states are available within the NVCACHE subsystem. These states include Power Up, Idle, Backup, Restore, Erase, Glitch, and Power Down. Transitions from one state to another are controlled by the NVCACHE control signals and some internal flags provided by the FPGA. Figure 2 shows a simplified state transition diagram for the NVCACHE subsystem.

Power Up is the state that the NVCACHE subsystem enters when power is applied to the RAID controller module. In the Power Up state the FPGA reads a header location in the flash array to determine if the flash array is clean or dirty.

If the flash array is dirty, the FPGA asserts the Cache Dirty and de-asserts the DRAM Available and NVCACHE Ready control signals. The FPGA then transitions the NVCACHE subsystem into the Restore state.

If the flash array is clean, the FPGA checks to see if the flash array has been erased and initialized properly. If the flash array has not been erased or has not been initialized properly the FPGA asserts DRAM Available, de-asserts Cache Dirty and NVCACHE Ready, and transitions the NVCACHE subsystem into the Erase state.

If the flash array is clean, erased, and properly initialized, the FPGA will assert DRAM Available, NVCACHE Ready, de-assert Cache Dirty, and transition the subsystem into the Idle state. See Figure 2.

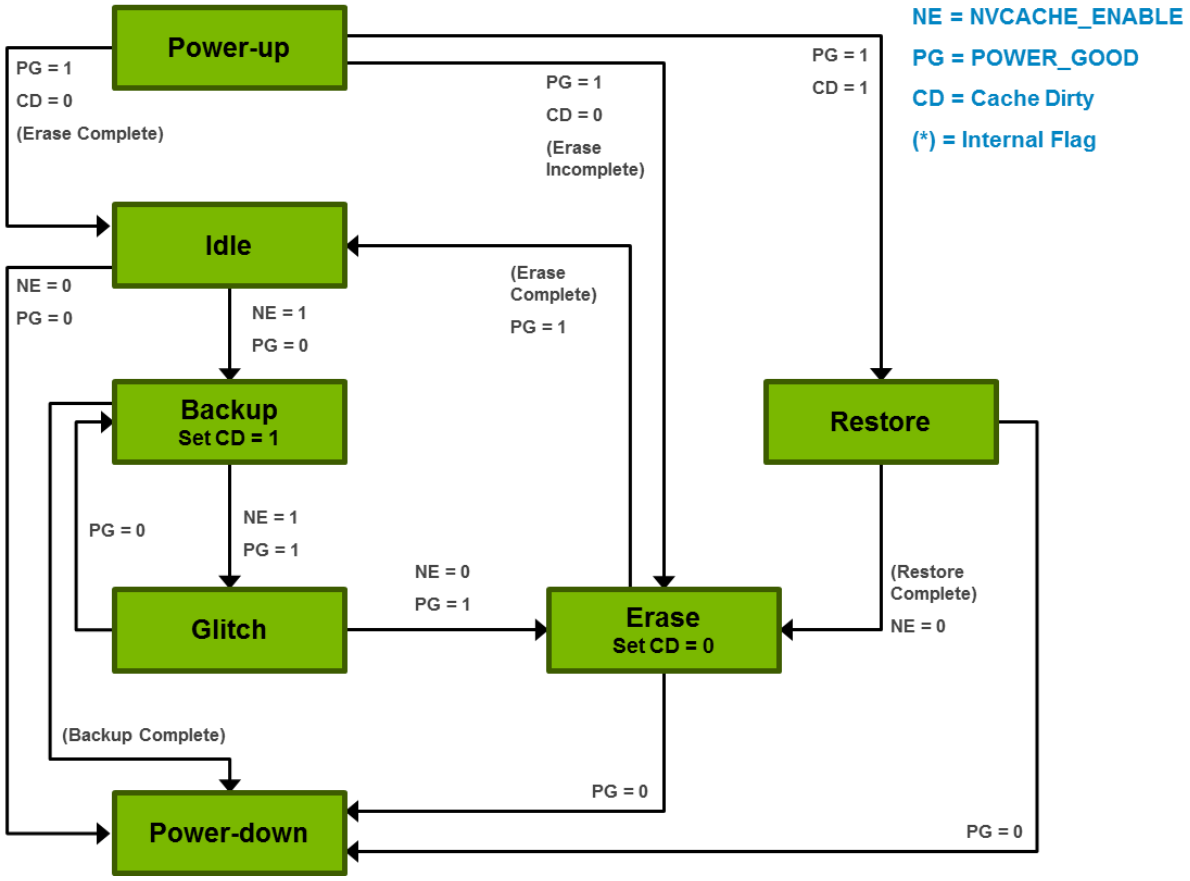

Figure 2. NVCACHE State Diagram

Idle is the state in which the NVCACHE subsystem exists after Power Up or Erase has completed and before the RAID controller decides to go into write-back cache mode. In this state, the NVCACHE Enable control signal is de-asserted and the subsystem is idle. If the system experiences a power loss (i.e., NVACACHE PG is de-asserted) the NVCACHE subsystem will transition to the Power Down state and gracefully power down without doing a backup of the cache data to the flash array.

Backup is the state the NVCACHE subsystem enters when it is in the Idle state and the RAID controller asserts NVCACHE Enable indicating that it is entering write-back cache mode. At this time the FPGA will prepare to execute a backup of cache data and assert the Cache Dirty control signal. The FPGA will then wait for either NVCACHE PG or NVCACHE Enable to be de-asserted.

If NVCACHE Enable is de-asserted, the RAID controller has exited write-back cache mode and there is no write cache data in the DRAM array. At this point the FPGA will de-assert Cache Dirty and the NVCACHE subsystem will transition back to the Idle state.

If NVCACHE PG is de-asserted, the system has suffered a power loss and needs to perform a backup of the write cache data. At this point, the power for the NVCACHE subsystem switches from system power to battery or super capacitor power. The FPGA de-asserts the DRAM Available control signal and then reads the write cache data from the DRAM array and writes it into the flash array. This process may take up to 30 seconds (the time may increase with increased DRAM size) to complete. Once the write

cache data is written to the flash array the FPGA writes the backup status into the flash headers and transitions the NVCACHE subsystem into the Power Down state.

If NVCACHE PG is asserted during the backup process (i.e., the power comes back on) the FPGA will pause the backup process and transition the subsystem into the Glitch state.

Glitch is the state the NVCACHE subsystem enters when it is in the process of backing up write cache data and system power is restored (i.e., NVCACHE PG is asserted). At this point the FPGA will assert the DRAM Available control signal, de-assert the NVCACHE Ready control signal, and wait until NVCACHE PG is de-asserted or NVCACHE Enable is pulsed/toggled for 500us.

If NVCACHE PG is de-asserted, the NVCACHE subsystem will transition back to the backup state and complete the backup process.

If NVCACHE Enable is pulsed/toggled indicating that the RAID controller has written the write cache data to disk and exited write-back cache mode, the FPGA will de-assert the Cache Dirty control signal and transition into the Erase state. By implementing the Glitch state the RAID controller does not have to wait for an entire backup cycle to complete if the system suffers a minor power glitch.

Erase is the state the NVCACHE subsystem enters when either a restore cycle has completed or the system has suffered a minor power glitch. The NVCACHE subsystem can only enter the Erase state after all write cache data has been written to disk and the RAID controller has exited write-back cache mode. In the Erase state the FPGA will erase and initialize the flash array. Once the flash array is erased and initialized, the FPGA will assert the NVCACHE Ready control signal and transition the NVCACHE subsystem to the Idle state.

Restore is the state the NVCACHE subsystem enters when the system powers up and the FPGA determines that there is valid write cache data written in the flash array. In the Restore state the FPGA will hold the DRAM Available control signal de-asserted until all of the write cache data has been read from the flash array and written to the DRAM array. Once the restore cycle is complete the FPGA will assert DRAM available indicating to the RAID controller that valid write cache data is available to be written to disk.

When the RAID controller completes writing the cache data to disk, it will pulse/toggle the NVCACHE Enable control signal for 500us indicating that it is safe to transition to the Erase state. The FPGA will de-assert the Cache Dirty control signal and transition the NVCACHE subsystem to the Erase state.

If power is lost during a restore cycle (i.e., NVCACHE PG is de-asserted) the FPGA will halt the restore operation and transition the NVCACHE subsystem to the Power Down state.

Power Down is the last state the NVCACHE subsystem enters in the cache-to-flash sequence. In the Power Down state the FPGA goes through some housekeeping tasks and then gracefully shuts off power from the battery or super capacitor. At this time all write cache data has been saved in the flash array and the system does not have a time limit as to when the data needs to be restored.

NVCACHE Scenarios

There are four scenarios that are of particular interest where the NVCACHE will transition through the state diagram in response to different power conditions. These scenarios include power-up clean, power-up dirty, power fail, and power glitch. Walking through these different scenarios will illustrate

how the NVCACHE functions under different conditions. The following discussion describes in detail the different NVCACHE scenarios.

Scenario 1: Power Up Clean

Power Up Clean is the scenario where the system powers up and there is no cache data in the NVCACHE subsystem. This is the sequence the NVCACHE subsystem will execute during a normal boot of the system. Figure 3 shows the sequence of events that occurs when power is applied to the system.

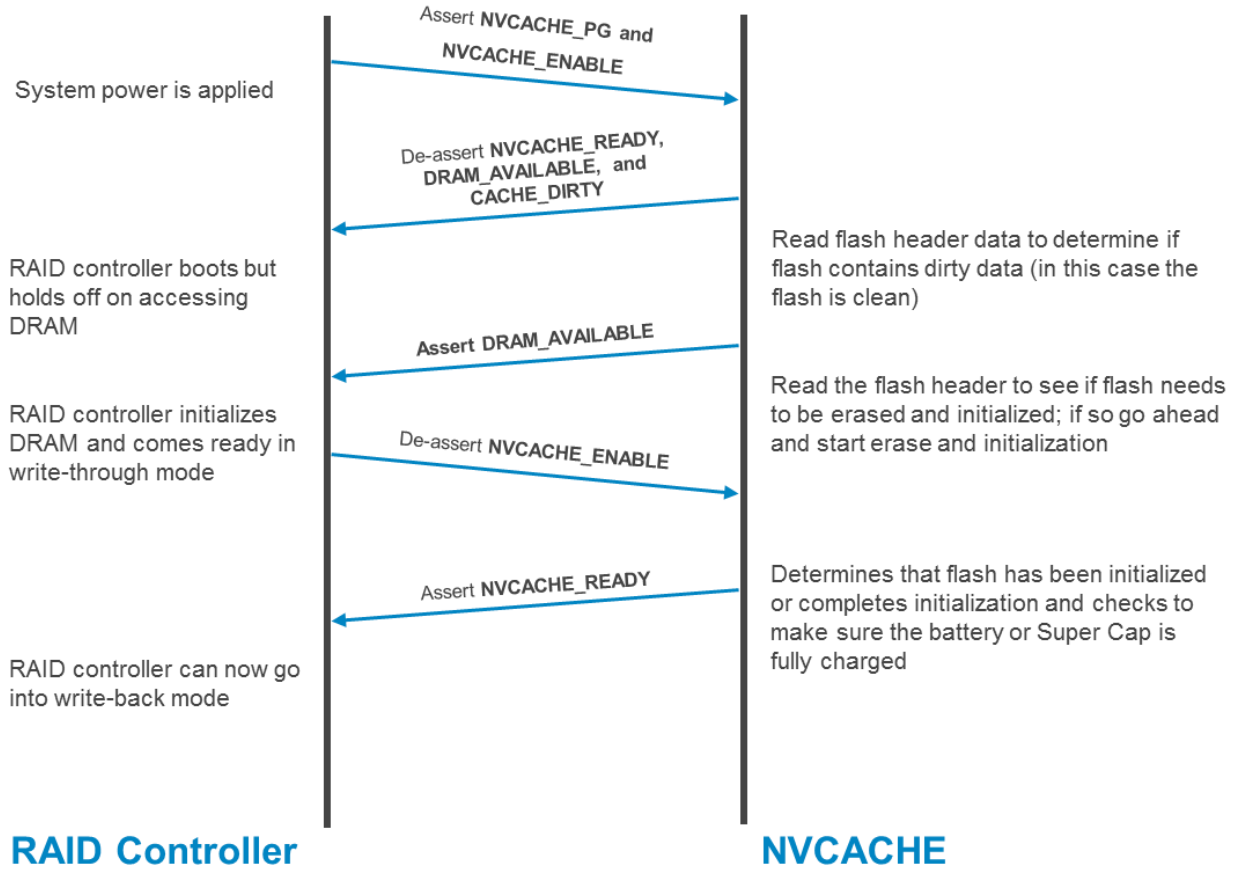

Figure 3. Power Up Clean

In this scenario the system hardware will ensure the default state of NVCACHE Enable is de-asserted and then assert NVCACHE PG to indicate that the system has good power. At this point the NVCACHE subsystem is in the Power-up state. The NVACHE subsystem will de-assert NVCACHE Ready and DRAM Available. The NVCACHE subsystem will read the flash headers to determine the state of the flash. If the flash has no valid cache data, the NVCACHE subsystem will de-assert Cache Dirty and assert DRAM Available indicating to the RAID controller that it can continue with its boot sequence. The RAID controller toggles NVCACHE_EN signal for 500us to signal FPGA to erase and initialize the flash. If the flash has been erased and initialized, the NVCACHE subsystem will assert NVCACHE Ready and transition to the Idle state. If the flash has not been erased or initialized, the NVCACHE subsystem will

transition to the Erase state and erase and initialize the flash. Once initialization is complete the NVCACHE subsystem will assert NVCACHE Ready and transition to the Idle state. Once The NVCACHE subsystem has entered the Idle state by asserting NVCACHE Ready, the RAID controller is free to go into write-back cache mode.

Scenario 2: Power Up Dirty

Power Up Dirty is the scenario where the system powers up and there is valid cache data in the NVCACHE subsystem. This is the sequence the NVCACHE subsystem will execute after it has experienced an unexpected power loss and the RAID controller was in write-back cache mode. Figure 4 shows the sequence of events that occurs when power is applied to the system.

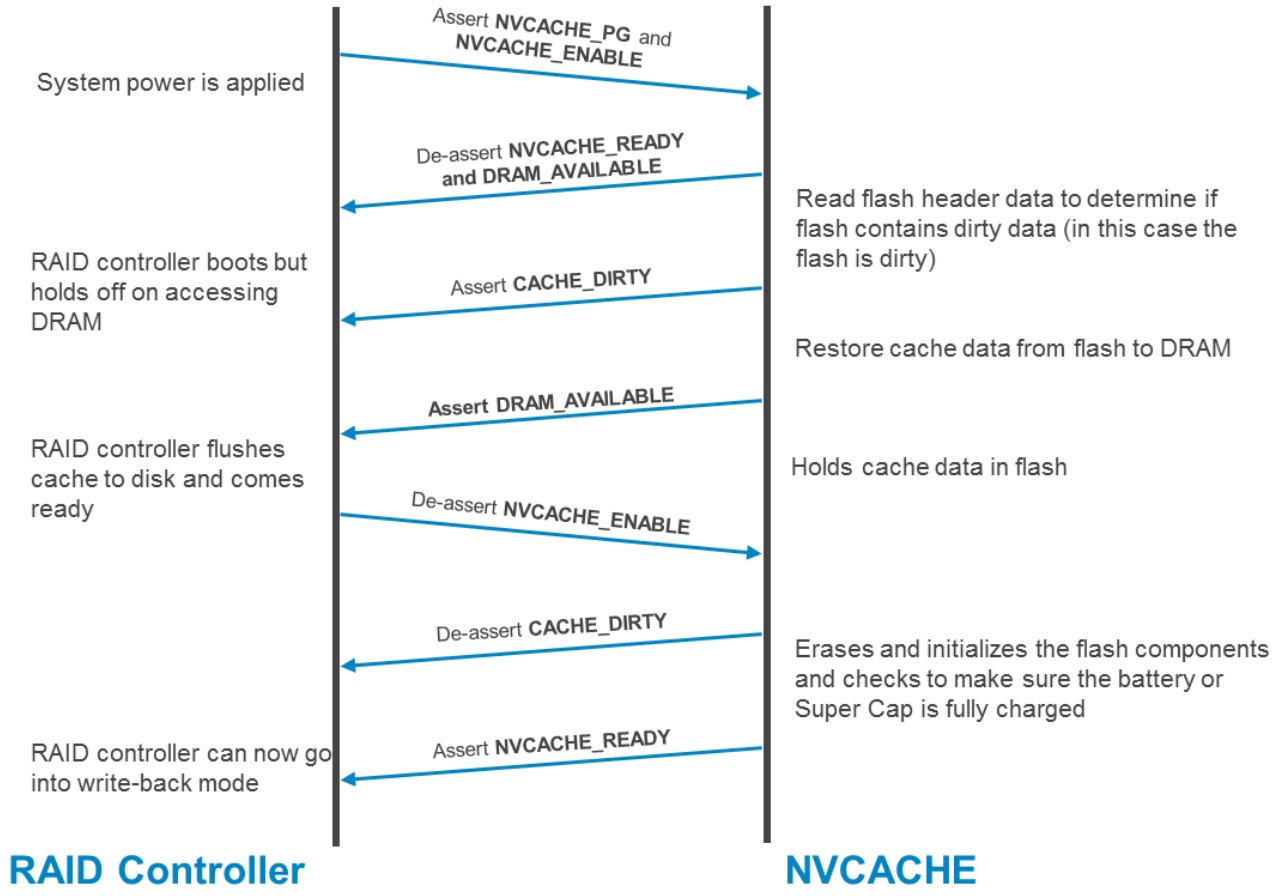

Figure 4. Power Up Dirty

In this scenario the NVCACHE subsystem goes through the same routine except when it reads the flash headers it finds that there is valid cache data in the flash array. At this point The NVCACHE subsystem will hold DRAM Available and NVCACHE Ready de-asserted, assert Cache Dirty, and transition into the Restore state. The NVCACHE subsystem will then restore all of the cache data from the flash array to

the DRAM array. Once the data is restored the NVCACHE subsystem will assert DRAM Available indicating to the RAID controller that it may continue booting. Once The RAID controller has booted and come online it will hold NVCACHE Enable de-asserted and proceed to write all cache data to disk. This process is known as flushing the cache. Once all data has been flushed the RAID controller will generate a pulse of 500us on NVCACHE Enable indicating to the NVCACHE subsystem it is safe to transition to the Erase state. At this point the NVCACHE subsystem will erase and initialize the flash array, assert NVCACHE Ready, and then transition to the Idle state.

Scenario 3: Power Fail

Power Fail is the sequence the NVCACHE subsystem will execute when it experiences an unexpected power loss and the RAID controller is in write-back cache mode. Figure 5 shows the sequence of events that occurs when the system detect a power loss condition.

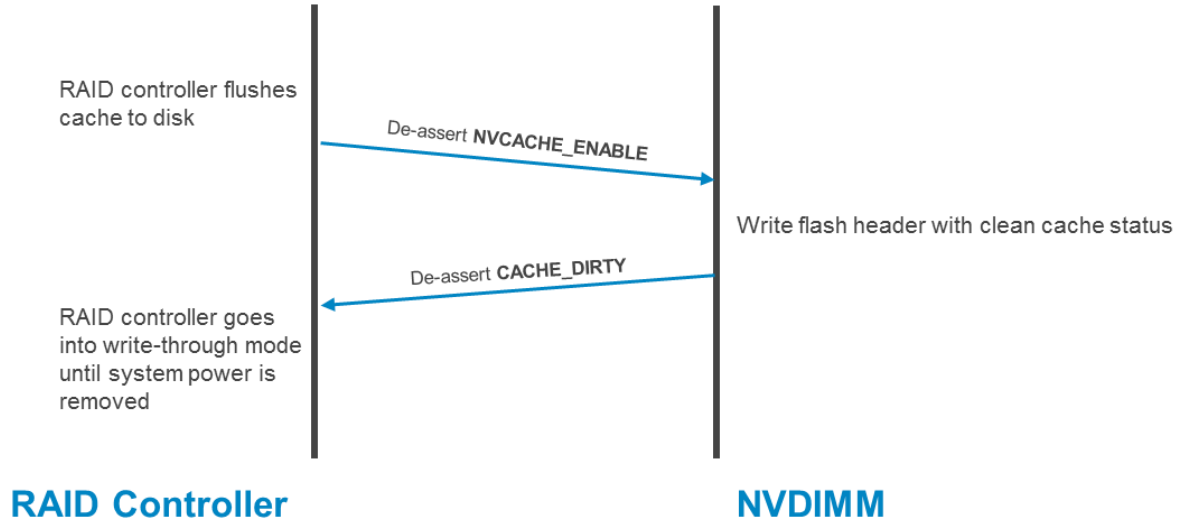

Figure 5. Power Fail

In this scenario the RAID controller will have NVCACHE Enable asserted indicating that it is in write-back cache mode. The system will de-assert NVCACHE PG when it detects a power fail condition and the NVCAHCE subsystem will switch from system power to battery power. The NVCACHE subsystem will also de-assert NVCACHE Ready and DRAM Available indicating that the DRAM array is isolated from the RAID controller and the FPGA is starting the cache transfer. The FPGA will write TRANSFER STARTED into the flash header to indicate that a backup cycle has started. The FPGA will then transfer the contents of the DRAM array into the flash array. Once the transfer is completed the FPGA will write transfer complete into the flash header. At this point the backup cycle is complete and the cache data is fully protected. The NVCACHE sub-system will then gracefully shutdown.

Scenario 4: Power Glitch

Power Glitch is the sequence the NVCACHE subsystem will execute when it experiences an unexpected power loss, the RAID controller is in write-back cache mode, and system power is restored before the backup cycle is completed. Figure 6 shows the sequence of events that occurs when the system power is restored before the backup cycle is completed.

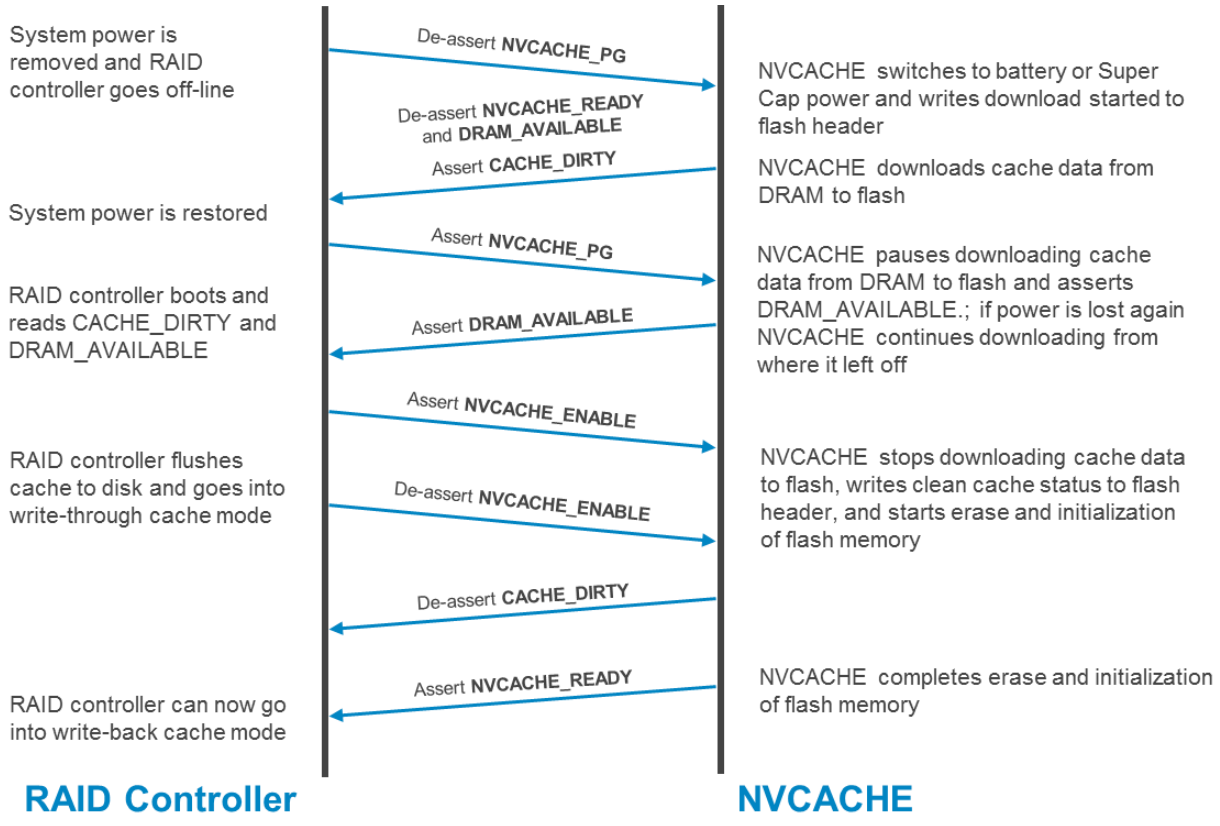

Figure 6. Power Glitch

This scenario is basically the same as the power fail scenario with the exception of having system power restored before the backup cycle is complete. In this case the FPGA will pause the backup process and assert DRAM Available indicating to the RAID controller that the DRAM array may be accessed. The RAID controller will note that CACHE Dirty is asserted and flush the contents of the DRAM array to disk. After flushing the cache the RAID controller will pulse NVCACHE Enable for 500us, and the NVCACHE subsystem will de-assert CACHE Dirty and erase and initialize the flash array. The NVCACHE subsystem will then assert NVCACHE Ready and transition into the Idle state. If system power fails again before the RAID controller completes flushing the cache to disk, the NVCACHE subsystem simply continues the backup operation from the point where it paused.

Summary

This paper has shown that a Cache-to-Flash NVCACHE implementation offers a secure and robust method to protect write cache data on a modern RAID controller. The Dell PERC H700 and H800 RAID controller products offer a CTF NVCACHE feature that increases data retention time between power failures. A Dell PERC RAID controller can be safely removed from a failed system and installed in a new system without having to worry about keeping the battery attached and the controller powered.