Using nvSRAM in RAID Controller Applications

Introduction

The term RAID (Redundant Array of Independent Disks) first appeared in papers written by Garth Gibson, Randy Katz, and Dave Patterson of the University of California at Berkeley. Since that time the number of manufacturers of RAID systems has expanded to over 100 companies with product lines that range from high end commercial products to lower cost controllers for the home market.

The RAID advisory board (RAB) was formed in 1992 to help minimize confusion within the industry by standardizing terminology and maintaining standards for the classification and typing of controllers. The board is comprised of over 40 members and continues to promote the industry by encouraging the development of hardware components that are optimized for RAID applications. The goal of the RAB is to become a compliance verification and testing organization that will issue product approvals, act as a regulatory agency, and assure users that the RAID level claimed by the manufacturer meets RAB standards. The RAB also will perform testing to certify that vendor hardware meets RAB requirements for Array-Ready Disks, and verify Disk Array Performance Benchmarks.

RAID Theory Overview

For disk I/O intensive systems there are two characteristics that act as the primary system performance bottlenecks:

1. Data Seek Time
2. I/O Transfer Rates

It would seem logical that all that is required to reduce the time necessary for the computer to fetch data from the disk is to use multiple disks in parallel and distribute the data. While this solution sounds easy and cheap, the realities of life aren't so simple. As any experienced system administrator can tell you 80% of the total I/O load of a system is directed at 20% of the I/O resources. This so-called 80/20 rule requires that the I/O system be tuned to distribute the load over the bank of parallel disks. This evens out the number of I/O requests per disk and greatly speeds up the disk access. The trouble with disk tuning is that it requires a lot of system administration time to accomplish, and when done, there is no guarantee that it will stay balanced. In a dynamic system the I/O load will change with time and therefore will require constant tweaking to maintain peak efficiency. A better solution is to use an array of disks. The RAB defines an array of disks as "a collection of disks from one or more commonly accessible disk subsystems, combined with a body of Array Management Software (AMS)".

Array Management Software is usually defined as firmware that executes in a dedicated control system rather than the host computer and has two major functions. Function one is to map the storage space available and optimize system balance to maximize disk I/O performance. Function two is to present storage to the operating environment as virtual disks by converting I/O requests to virtual disk I/O requests. This gives the appearance of a single large disk to the system and frees the administrator from constantly having to tweak the data distribution. Disk arrays generally have improved I/O performance, and simpler storage management requirements than a string of parallel disks.

The next task facing the designer of RAID systems is to assure that data stored in the array can never be lost due to hardware failure. Major users of disk array systems such as banks, airlines, and credit agencies must be certain that they can never lose a disk in such a manner that the data stored on that disk is not recoverable. Even frequent and conscientious backing up of all disk storage does not recover new data that has been written since the last backup cycle was performed. A solution to this data reliability problem is the use of a RAID Controller. RAID Controllers are defined with 7 levels:

Level 0 - Data Striping
Level 1 - Disk Mirroring
Level 2 - Hamming Code
Level 3 - Parallel Transfer Disks with Parity

Level 4 - Independent Access Array
Level 5 - Independent Access Array with Rotating . . Parity
Level 6 - Recovery from the failure of up to 2 disks

RAID Level 0

A stripe set presents a single virtual disk whose capacity is equal to the sum of the capacities of its members. The reliability of the stripe is less than the reliability of its least reliable member and its read and write rates are high. RAID 0 is not a true RAID controller because it provides no redundancy. It is, however, a performance-oriented architecture that is inexpensive and therefore attractive to many low cost users. RAID 0 is a parallel transfer technology.

RAID Level 1

A mirror set also presents a single virtual disk; its capacity however is equal to that of its smallest member. Its reliability is very high, its read performance is usually better than that of a single member, but its write performance is somewhat slower. A RAID 1 system protects against disk failure by replicating all stored data at least once on a physically separate disk. RAID 1 can be implemented as either a parallel or independent array and is well suited to applications that are read intensive and where reliability requirements are high.

RAID Level 2

A parallel access array that uses Hamming Coding to provide error detection and correction capability to the array. This approach is very expensive and therefore almost never implemented into a system.

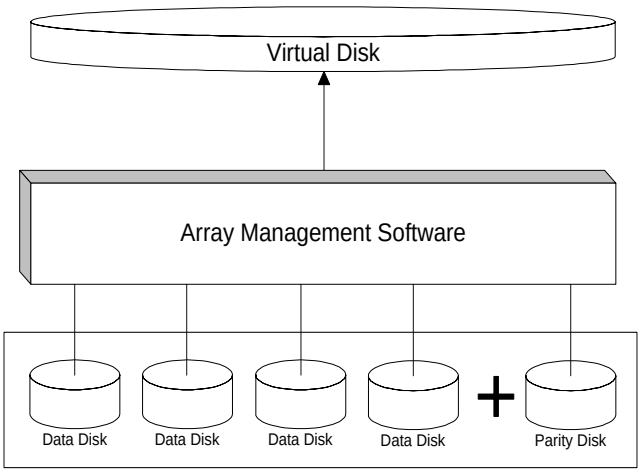

Figure 2
Example of a Typical RAID Level 3 or 4 Controller
From The Roadblock Edition 1-1

RAID Level 3

RAID 3 is optimized for high data transfer rates and is a parallel transfer technique with parity. Each data sector is subdivided, and data is scattered across all data disks with redundant data being stored on a dedicated parity disk. Reliability is much higher than a single disk and the data transfer capacity is the highest of all listed RAID types. RAID 3's weakness lies in its relatively slow I/O rates that make it unsuitable for most transaction processing unless assisted by some other technology such as cache. The parity disk stores redundant information about the data chunks stored in corresponding locations on the data disks. The redundant information is typically in the form of a bit-by-bit Exclusive OR function of corresponding data chunks from the other disks. Typical applications for RAID 3 include large data objects such as CAD files, graphical images, seismic or telemetered data streams.

RAID Level 4

RAID level 4 is an independent access array in which data sectors are distributed in a similar manner to disk striping systems. Redundant data is stored on an independent parity disk (similar to RAID 3). Its data reliability is much higher than a single disk (comparable to RAID 2, 3, and 5) and its data transfer capacity is moderate. RAID 4 is a high I/O read rate technology with moderate write speeds, but is not well suited for high data transfer applications due to the parity disk write bottleneck Two of the four operations required to perform a virtual disk write are directed at the parity disk; for this reason RAID 4 arrays are seldom implemented. Possible applications would include systems that are read intensive and do not require high data transfer rates.

RAID Level 5

RAID level 5 is an independent access array with rotating parity. Data sectors are distributed in the same manner as disk striping systems but redundant information is interspersed with user data across multiple array members rather than stored on a single parity disk as in RAID 3/4 systems. This relieves the write bottleneck associated with RAID level 4 controllers. RAID 5 arrays have high data reliability, good data transfer rates and high I/O rate capability. It is well suited to applications such as online customer services, inquiry-type transaction processing, group office automation, etc.

RAID Level 6

RAID 6 is a non-Berkeley level controller that is designed for extremely high data reliability. RAID 6 is an independent access array concept that requires two parity blocks be updated for each block written. This requires an extra parity disk but gives

the added data safety of requiring 3 disks to fail before data will be lost. RAID 6 data transfer and I/O capability is lower than RAID 5 for writes, but data reliability is highest of all RAID architectures. Presently RAID level 6 is not widely used because of the higher costs associated with the added complexity, and the high penalty paid in system I/O performance due to long write times.

Additional RAID Implementations

RAID 10 is a combination of RAID 0 & 1. This architecture gives high I/O performance and good data reliability. It is accomplished by using RAID 0 (data striping) to enhance I/O rates and by using RAID 1 (disk mirroring) for high data reliability. RAID 10 requires costly hardware (disk and port) to implement, and is primarily used in applications where the data has high value and can justify a mirrored storage system.

RAID 53 is a combination of RAID levels 0 & 3 and provides RAID 3-like data transfer performance, and striping-like I/O request rates at RAID 3 or 5 costs. RAID 53 is used where both high data request rates and high data transfer performance is required such as airline reservation systems, financial and banking applications, etc.

nvSRAM Applications & System Architecture

In modern RAID systems the Array Management Software can run in the host or in a dedicated embedded controller. Most modern systems are using embedded controllers, including many manufacturers using the Intel i960 chip as the engine.

In the past RAID systems were designed to use a distributed block of disk to maintain system configuration and to store system recovery address vectors. The primary problem with this type of architecture is that if a power failure occurs, and the controller's volatile system memory is lost, the entire disk array must be scanned upon power up to reestablish configuration and to redefine data locations. On a large array this is very time consuming, requiring many minutes to accomplish. Service-oriented industries cannot afford this length of down time and must come up and be operating very quickly once power is restored. In the latest generation of RAID systems the restart vectors are stored in nonvolatile semiconductor memory on the controller board itself. Due to the fact that the Array Management System is constantly moving data among the individual array members to optimize I/O balance, maximize I/O rates, and assure redundancy, the RAID controller is constantly tweaking the address vector tables. Also, the system configuration data is being

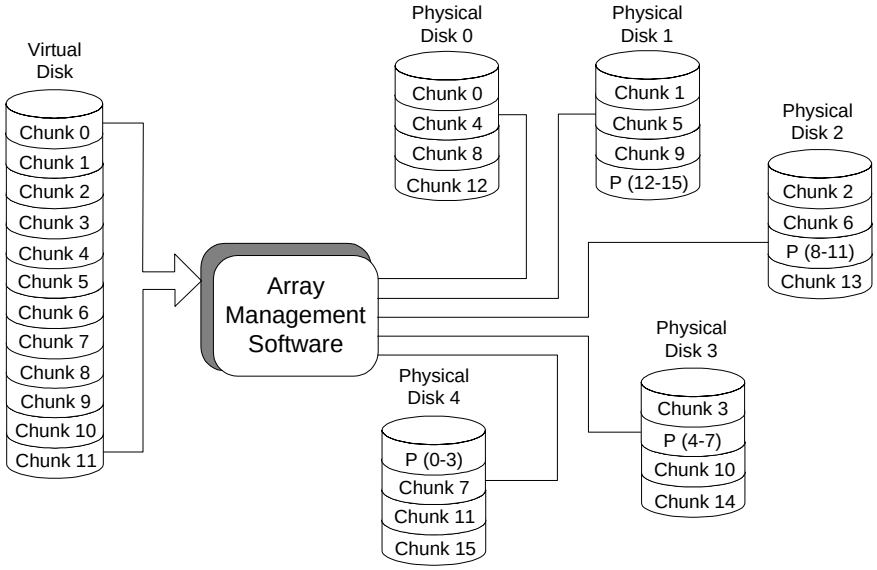

Figure 2
Example of a Typical RAID Level 5 Controller
From The RAID Book Edition 1-1

stored simultaneously in several different locations, and parity information is being maintained to allow recovery in case of disk failure. This constant moving of data and reconfiguring of the array, requires that the configuration address vectors be stored in a nonvolatile technology that is rapidly rewritable. This memory must be fast enough to run at processor bus speeds so the RAID controller does not have to waste time blindly searching the disk array for its configuration data.

The SIMTEK family of nonvolatile SRAM products are ideally suited to RAID applications. They run at processor speeds, require no batteries for nonvolatility, are standard size and shape for automatic manufacturing, and are flow solderable. The high number of nonvolatile stores (1 Million), fast read and write times (20 ns), and fast store times (10 ms) allows the design of highly efficient embedded controllers for both commercial and home applications.

Applying SIMTEK nvSRAM To Modern RAID Controllers

The STK10Cxx Family

The SIMTEK STK10Cxx family of nvSRAMs is a high performance nonvolatile memory family that is designed for easy interface to embedded controllers. The STK10Cxx family is designed for dynamic applications that require easy transfer from SRAM to nonvolatile memory under processor control. Using the 10Cxx family for *store* and *recall* of address vectors, configuration databases, and error detection/correction codes is an ideal application.

Data within the SRAM is stored to nonvolatile memory by simply requesting an SRAM write with the $\overline{NE}$ line active. This transfers data from the SRAM to shadow EEPROM in 10 ms. Data in the EEPROM is then completely nonvolatile and requires no batteries, capacitors or other energy sources to remain stored for 10 years.

The STK11Cxx Family

The SIMTEK STK11Cxx family of nvSRAMs is designed for use in applications where data must be safely maintained in a fast nonvolatile memory, but does not require the dynamic *store* capability of the STK10Cxx family.

The STK11Cxx family of parts uses a software *store* and *recall* system that allows the flexibility of a fast nonvolatile SRAM memory, but offers the data security of a less dynamic technology. This part has high applicability to RAID controllers for address vector and configuration storage.

The STK12C68/STK14C88 Family

The SIMTEK STK12C68/STK14C88 *AutoStore*™ family combines the flexibility of the 10Cxx family, the data security of the 11Cxx family, and adds the capability to perform power down *AutoStore*™ to assure that data is never lost.

The 12C68/14C88 family uses the same *STORE* and *RECALL* techniques that are used with the 10Cxx and 11Cxx parts, but also gives the design engineer the flexibility of automatically storing on power down. *AutoStores*™ are autonomously performed (unless inhibited) when system power falls below V_{SWITCH} (about 4.25 V). This assures that data is always safe and requires no batteries or other power sources that are prone to failure. RAID controllers that are dynamically keeping track of address vectors, system configuration, and disk error recovery information are ideal applications for this family of parts.

Conclusion

Modern RAID Controllers offer high data reliability and increased I/O performance in one package. This requires that the RAID system designer utilize the latest in high performance embedded processors, state-of-the-art software, and sophisticated control algorithms.

The flexibility of fast nonvolatile memory for the storage of RAID address vectors, system configuration information, and adaptive algorithm memory is becoming more apparent with each new generation of system. SIMTEK's family of fast nvSRAMs are ideally suited to this type of high performance application. They run at processor speeds with no wait states, reliably store data in a nonvolatile semiconductor memory without batteries, and look to the processor like a standard SRAM. Ease of implementation and ability to rapidly change data, as well as the ability to use modern manufacturing techniques, helps to reduce costs and to speed product to market.