



Where's the solution?

The proliferation of NAND flash architectures, the migration to deep submicron processes and the introduction of new features have made NAND subsystem design more challenging.

Designers integrating NAND have three options: discrete solutions using flash controllers and separate NAND devices; embedded NAND controllers, with separate NAND devices; or managed NAND solutions.

NAND flash was once dominated by Samsung and Toshiba, who used similar architectures, a 512byte sector size and similar command interfaces and packaging. System designers with a simple controller could use the ATA/IDE bus to translate address and data signals into memory accesses and swap out one vendor's devices for the other.

However, the technology has evolved. As densities passed 4Gbit, manufacturers found the 512byte programming page was no longer optimal, moving instead to a 2kbyte page. More recently, some have adopted a 4kbyte page. At the same time, multilevel cell (MLC) architectures, storing up to 4bit per cell, are appearing.

Whilst these features are intended to

Discrete or managed NAND?

Analysing the tradeoffs.

By Yuping Chung.

improve performance, they present problems for those looking to multiple source NAND flash.

NAND controllers address this problem by offering a simple interface between host and flash. OEMs typically purchased controllers and NAND ics from different suppliers and mounted them on a board. Once the system powered up, a flash file system in the controller recognised the memories, executed handshaking and performed low level formatting.

More recently, vendors have offered managed NAND solutions, which combine an ATA controller with one or more NAND flash die in a multichip package, bringing cost and space savings.

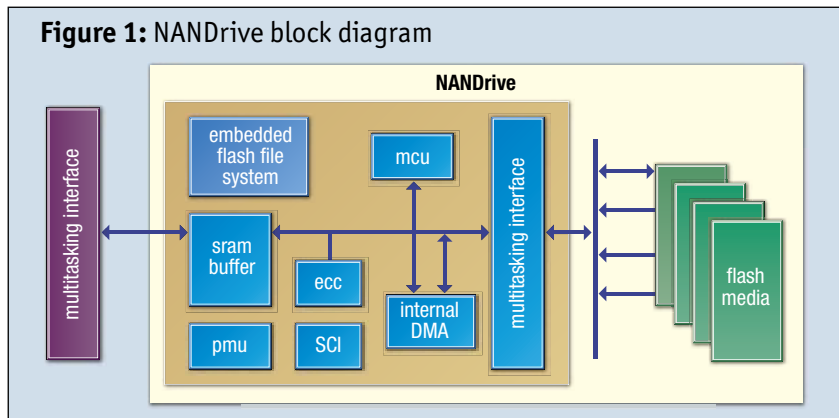
Both discrete and managed NAND solutions use an mcu to decode host commands and translate them into flash commands. To boost performance, the mcu

is usually supported by a small sram buffer between host and flash. To eliminate latency, these solutions typically use a DMA block to interface between the mcu and the data channel running between the host ATA IDE bus and flash.

The key differentiator in any NAND flash subsystem is the embedded flash file system block, which manages handshaking. For designers, this simplifies the task of writing data to memory by compensating for architectural differences between various flash devices. It can also store customer specific features or add bug fixes when required. Finally, the flash file system typically stores a number of media management functions that help ensure data integrity and longer life for the NAND flash memory subsystem.

Not all implementations of the flash file system block are the same, however. Many controllers implement the block in mask programmable rom. Controllers from vendors such as SST use an in circuit reprogrammable flash memory. Given the rate of change in NAND technology, this distinction is crucial. Designers using a controller that stores firmware in rom must





generate an expensive mask change to reprogram the device every time they update the firmware. This also applies to SoCs with integrated NAND controllers.

Inevitably, designers using these devices cannot keep up with the changes and must therefore limit their component selection. Designers using controllers that store their firmware in embedded flash, on the other hand, can modify their device

charge associated with it and is more prone to temperature variations. Eventually, temperature variations could result in a level change leading to soft errors. To compensate, NAND flash subsystems must provide more powerful ECC engines. Some, such as SST's discrete controller and NANDrive product lines, offer an 8bit hardware ECC engine.

Bad block management is also

repeated erase and program cycles, all flash memories no longer retain data. But endurance levels have decreased with smaller geometries and multilevel cells. Flash memory controllers compensate using wear levelling – algorithms which track memory usage by block or page.

Power conservation in battery powered portable applications is also a key concern. NAND flash solutions typically feature a power management unit which scans the memory subsystem and powers down inactive functions, extending battery life.

Decision criteria

Given their relatively similar features, choosing between discrete or managed NAND memory subsystems can be a daunting task. But a few key considerations simplify the decision.

The first is memory capacity. NANDrives from SST support up to 8Gbyte, with higher densities expected. If a higher density memory subsystem is needed, a discrete solution is an alternative.

In some applications, inventory management or supply risk may be an issue. Discrete solutions allow the use of NAND ics from a range of suppliers to avoid allocation issues. On the other hand, managed NAND solutions can simplify inventory management by offering the opportunity to source a single device specification. Managed NAND solutions also use stacked packaging techniques to offer extremely compact devices.

Finally, designers must review in house skills. Discrete NAND solutions can offer the flexibility to build custom configurations, but designers must devote resources and time. The team will need knowledge of how a discrete NAND controller functions and the specific interface requirements of NAND devices from different suppliers. Development teams with limited NAND expertise and those facing tight development schedules may find the 'plug and play' characteristics of a managed solution advantageous. ■

Author profile:

Yuping Chung is business director for Silicon Storage Technology.

“Choosing between discrete or managed NAND memory subsystems can be a daunting task.” Yuping Chung, SST

quickly and inexpensively. This brings more flexibility, allowing components from a wider selection of vendors to be used.

Key features

Whether designers are planning a managed or discrete approach, there are a number of key features. One is embedded error correction (ECC). Random read errors commonly occur with NAND flash, due to read/write operations in adjacent cells and ECC circuits combat this problem.

This has become increasingly important as NAND manufacturers have turned to MLC techniques. Manufacturers typically specify one bit error for every 512byte page in SLC devices but, MLC architectures can have up to four bit errors per 512byte block.

Typically, an MLC device operates at the same voltage as comparable SLC NAND devices, so each level in a MLC cell has less

important. Whilst NOR flash guarantees the integrity of all memory bits, NAND devices are designed to operate with bad blocks. Memory manufacturers typically specify which blocks are bad. At initialisation, firmware based bad block management functions identify these blocks and map them out of the memory array.

Endurance presents another issue. After

