Technical memory

Mike Alwais reviews the design considerations in implementing a single-chip memory environment

Design engineers working on embedded systems usually select memories based on the intended memory contents: A simple division of storing executable code in non-volatile memory and data in volatile memory, except for archival purposes. Recently, the situation has become more fragmented. Hard disks and flash memory are now used for storage of images and music, which can be classified as data. For most embedded systems, however, the traditional mix of memory technologies with applications still holds.

Applications for memory technologies are generally divided between executable code and data tasks. ROM-based technologies, including mask-ROM, OTP-EPROM and NOR-Flash, are non-volatile and are oriented toward code storage applications. Other derivatives of ROM technology including NAND-Flash and EEPROM can serve as a non-volatile data memory. These are clearly a compromise since they perform both code and data storage with low-performance relative to alternatives. Flash's main virtue for data is low cost, rather than ease of use or performance.

RAM-based technologies such as SRAM serve as data memory and also as working-space for code execution when Flash is too slow. RAM provides an excellent blend of code and data functionality, but ordinary RAM provides only temporary storage. Space-constrained applications require maximum functionality in few devices. Even in applications with adequate board space the design engineer may be unhappy using up to three memory types in the system. Ideally, a single memory technology could serve for code and data. Since it must be non-volatile, ordinary RAM is omitted from consideration. This leaves the ROM family of technologies, which offer poor performance as data storage media, undesirable battery-backed SRAM, or Ferroelectric RAM (FRAM).

Code storage

In many older systems, code storage was either ROM or OTP-EPROM and required little thought. The memory was non-volatile and could not be altered in-system. Therefore, the basic parameters of memory size, access time and operating voltage were the main design issues.

The popularity of field alterable memories for embedded system code store introduces new variables associated with in-application programming. Code storage is a read-oriented environment. Updates are infrequent. Owing to the difficulty of programming Flash, working parameters and temporary variables that must be altered are stored in RAM. While Flash memory devices are still tricky to upgrade in a single-chip memory environment, they are considered relatively ideal for code storage.

Flash has substantial shortcomings as a data memory, as discussed below. A summary of the attributes needed and desired

"Ramtron FM20L08": The FM20L08 is a byte wide FRAM memory logically organised as 131,072 X 8 and accessed using a parallel interface

for code storage is listed below. The latter two points are particularly desirable for systems with just one or a few memory devices.

Essentially, code memory considerations can be classified as:

- Non-volatility
- Appropriate density
- Read access time
- Ability to prevent inadvertent writes
- Field programmable in parts or sections
- Programmable with concurrent read access

Data storage

In many respects, the requirements for data are the opposite of those for code. Data storage is a far more diverse task, requiring flexibility and easy write-access. It may be either non-volatile or volatile depending on the type of data. For many applications, the write functions of a data memory are at least as important as the read functions. An instructive example is the crash recorder and smart-seat sensor in an automobile. In this case, the instrumentation data is stored continuously in a circular buffer and also immediately as a crash occurs. Writes are frequent and must occur quickly. In most cases the data is never read. After an automobile collision it is read once. Write operations should not require complex protocols and should not take a long time. In addition, a high number of writes should be accommodated for virtually unlimited data collection.

The very term 'programming' implies that adding new content to Flash is significantly more difficult than adding it to RAM-based memories. Complex algorithms for writing, delays for erasing and relatively long write cycles are normal for Flash. Sectors program at comparatively slow speed, but must first be erased, which is extremely slow. EEPROM is a traditional choice for non-volatile data memory since it is simpler to use. However, it offers very slow write times and both it and Flash permit few write-cycles before wearing out.

FRAM is the first memory optimised for non-volatile data storage. FRAM writes occur at the same speed as the reads. It is non-volatile and offers an effectively unlimited number of write cycles. No algorithms or protocols are needed to write the memory and it is byte addressable. FRAM provides the most flexible solution with the fewest drawbacks for data storage. While more diverse than code, a list of potential data-memory attributes to be considered are follows:

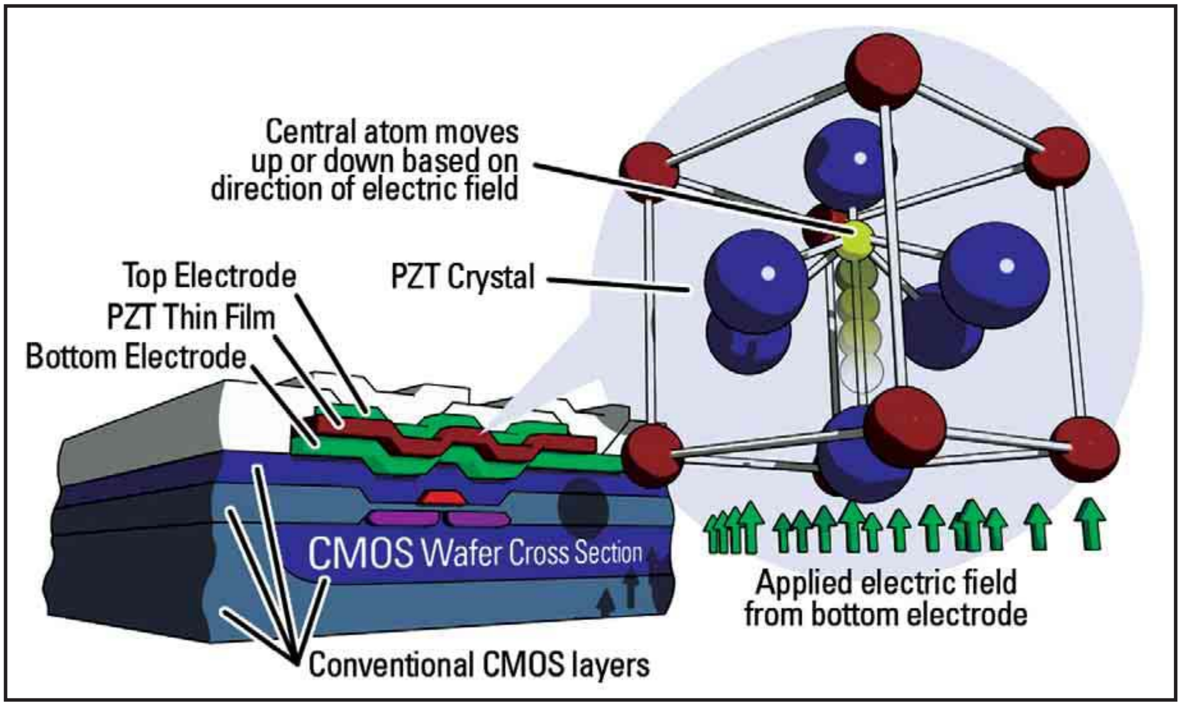
- Fast write access
- Many writes allowed
- Simple write protocol (none is best)
- Byte-addressable writes
- Ability to serve volatile and nonvolatile needs.

Single-chip code and data requirements

Blending code and data applications in a single device means that one memory provides almost mutually exclusive services. Code storage requires non-volatile memory and the need for occasional writes. Upgrading code does not call for a large number of write operations. While slow writes can be inconvenient, in some respects, the more difficult it is to write a code memory the better. An accidental write could be catastrophic. Data memory often needs a non-volatile storage that can be written without restriction. It requires relatively large numbers of writes and write speed can be an important factor.

FRAM technology is optimised for data memory applications. Its superior write-performance makes it preferable to Flash or EEPROM and its non-volatile aspect makes it preferable to battery-backed SRAM. Since it is non-volatile, it can serve as code memory as well. The main concern is to prevent inadvertent writes. The FM20L08 128Kx8 FRAM from Ramtron is intended to serve as a single-chip-solution for code and data applications. It is sufficiently larger to serve many applications, offers the performance benefits needed in a data collection system and has sophisticated block write protect allowing users to designate certain memory areas read-only.

When not write protected, writing FRAM is very similar to writing an SRAM. This is ideal for data storage. When serving as code storage it is critical to make sure that the system does not



accidentally write to an area of memory being used for executable code. A version of the FM20L08 features eight sectors that can individually be selected as write-protected, as discussed below.

High-density FRAM

The FM20L08 is a non-volatile ferroelectric random access memory (FRAM) that is designed as a drop-in replacement for standard asynchronous SRAMs. Organised as a 128K x 8 non-volatile memory that reads and writes like a standard SRAM, this 3-volt, 1-megabit FRAM comes in a lead-free, 32-pin TSOP (thin small outline plastic) package and offers virtually unlimited read/write cycles; it allows users to access each address a million times per second for hundreds of years with no wear out. The FM20L08 provides fully compatible SRAM timing with address transition detection (ATD) and is tailored to drop directly into an SRAM design with a JEDEC 128Kx8 SRAM pin-out and 60-ns access time. High-speed page mode operation runs up to a 33-MHz bus speed for a 4-byte burst. The device is available in both the industrial temperature range (40°C to +85°C) and the commercial temperature range (0°C to +70°C). Cycle times are 150 ns and 350 ns, respectively. With a standard SRAM pin-out, the FM20L08 operates like an SRAM, only better. It provides non-volatile storage; therefore, battery backup is not required, which lowers the cost. Operating at 3.3 volts, the FM20L08 uses less operating current than a standard SRAM.

Where it fits

Targeted for systems that collect and store data where power levels can vary or be lost suddenly, designers in the set-top box, automotive telematics or industrial applications space can opt for an FM20L08 over a standard SRAM product. This device is equipped with a voltage monitor that oversees the power supply voltage and will issue an active-low signal indicating that the memory is write-protected when VDD drops below a critical threshold. By locking out access to the memory when VDD is low, data corruption is easily avoided.

Equipped with software-controlled write protection, the FM20L08's memory array is divided into eight uniform blocks, each of which can be individually write protected; the settings are non-volatile. An address and command sequence drives the write-protection mode. To change write protection, the system host issues six read commands and two write commands. The specific sequence of read addresses must be provided to access the write-protect mode.

In summary, FRAM is the memory technology that best serves the combined code and data experience. It's true that other technologies can be made to work, however they suffer performance limitations.

The author is a Vice President, Ramtron International Corporation. For more information log on to www.ramtron.com