

Little-known flash-memory features protect data and IP

FEATURES FROM BLOCK LOCKING TO ENCRYPTED-PASSWORD-ACCESS MECHANISMS CAN PREVENT UNINTENTIONAL DISRUPTION, MALICIOUS DAMAGE, OR COPYING.

You design a system, and somebody messes it up. The damage is sometimes unintentional. For example, a service provider may install its software on your device and corrupt your original code. On the other hand, hackers and IP (intellectual-property) thieves go out of their way to overwrite, copy, or clone data stored in your systems. Whatever the cause, the resulting damage or theft represents no less of a problem. It's not surprising that designers need a way to protect system integrity. What may be surprising is that within its bits and blocks, flash memory holds the key to protecting firmware and even hardware designs.

Flash devices offer a number of data-protection measures, each with its own advantages for read, write, or erase protection. The security options add layers of security to slow down would-be hackers and thieves and provide protection from unintentional modifications. Some flash-security features don't even add cost to the final design, and, although the strongest flash-protection features may cost more than standard flash, they are far more affordable than a nonflash-hardware-encryption engine, hidden operations, authenticated operations, or software-encryption applications.

Manufacturers and even devices from the same manufacturer offer different features. Designers must select the right flash device for the final application after considering a number of factors, such as the built-in security options, performance, density, size, and cost.

FINDING THE RIGHT APPROACH

Evaluating the options starts with identifying the problem that you want to solve. Features perform specific functions, and some come with added cost. First, determine what you must protect amid the bits, data, and code. For example, you might need to protect electronic-system serial numbers, security keys, boot code, or financial information for services access, such as for pay TV. Once you know what you need to protect, determine whether a software or a physical disruption is likely to affect those bits, data, or code. A software attack may come from the Internet or a system application, for example. A physical attack, for instance, could involve the direct removal of

a flash device from a PCB (printed-circuit board).

Finally, identify whether the threat is unintentional or intentional. Unintentional alterations, such as those that bugs in software cause, are typically easier to prevent because the cause of the problem is not elusive or persistent. If the attack comes from a hacker or a thief, quantify how much effort the attacker is willing to make. The amount of time and money a hacker is willing to spend affects how much security the design requires. With these data points, determine which flash-security features provide the right level of protection against the source and intent of the attack. For example, if you must protect the design from data corruption from an Internet attack, block locking provides moderate protection, and OTP (one-time-programmable) blocks provide the best protection (Figure 1). If an IP thief aggressively targets the design by removing the flash device and attempting to read the data using a PROM (programmable-read-only-memory) programmer, protecting the design may warrant paying more for flash-data-encryption features (Figure 2).

DETAILED FEATURE REVIEW

From block locking to advanced encrypted-password access, you can choose the features that address the type and source

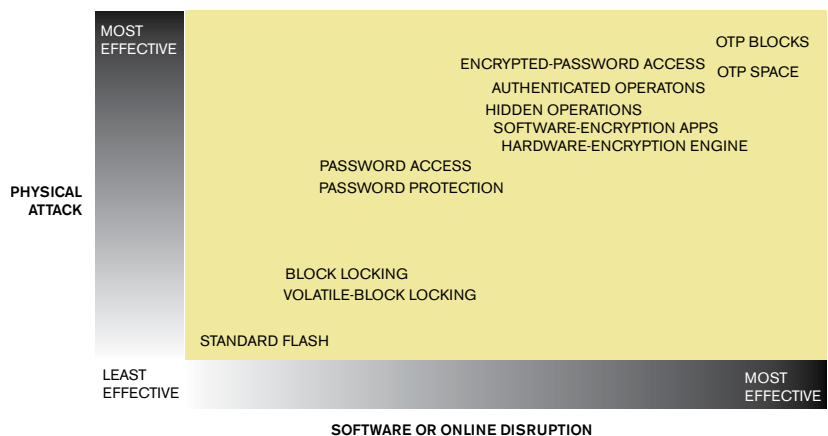


Figure 1 Security alternatives offer differing levels of protection against physical attack and software disruption, whether due to malicious corruption or unintentional data alteration. Note that adding such features as encrypted-password access, authenticated operations, hidden operations, software-encryption applications, and hardware-encryption engines increases system cost.

of an expected attack, with acceptable cost impact on the design. One such feature is password access. Some flash devices offer password-access features that slow down thieves, creating barriers that make the design a less desirable target for copying or cloning. IP thieves must quickly and easily copy system data. Password access adds time, cost, and effort to low-overhead cloning operations. Password access locks either the entire array or selected blocks in the main array from program, erase, or read access, depending on the flash device. You can individually set each block to the desired protection level. Before the system leaves the factory for the end customer, the manufacturer must store a 64-bit password in the password area of the flash device and program a matching password into the system microcontroller or other hidden storage.

When the system receives a command to read, modify, or erase data in the protected blocks, the system processor looks for a match between the number in the microcontroller and the one in the flash device. If the passcode is not valid with both the microcontroller and the flash device, the would-be hacker cannot read or modify the data. If the system detects a matching passcode, a user can read or modify individual blocks. Depending on the flash device, designers can choose from various protection modes, including read, modify, and substitution prevention.

Password protection is both a service-theft deterrent and an IP-copying and -cloning deterrent. Duplicated flash chips can provide premium services to users who don't pay for them, representing lost revenue for the service provider. Password-based read protection is a simple, cost-effective way to thwart attempts to distribute pirated flash chips that enable access to premium services. If the designer uses password protection on the flash device, he leaves the pirate with inoperable chips. When a would-be service thief attempts to read the data stored on the flash device, the device attempts to validate the 64-bit password. Without the password, the device returns only values of zero, rendering the copied chip inoperable. Even if the thief can snoop, discover the password, and copy the data from the chip, the 64-bit password in the pirated chip will not match the password in the microcontroller in the system into which the thief inserted it, again making the chip inoperable.

In the case of cloning, IP thieves must replicate and produce a design before an updated version of the original makes the clone obsolete. Flash-memory-password protection can create a significant enough delay to make the cloner seek an easier target because the flash protects the hardware signature of the system. A flash device with a 64-bit password limits access to legitimate sources. Without the password, an IP cloner who uses a PROM programmer to read the flash chip will read back only zeros.

As with the service-theft deterrent, the cloner can resort to bus-snooping to discover the password, but that step adds time, cost, and effort. The delay gives the legitimate manufac-

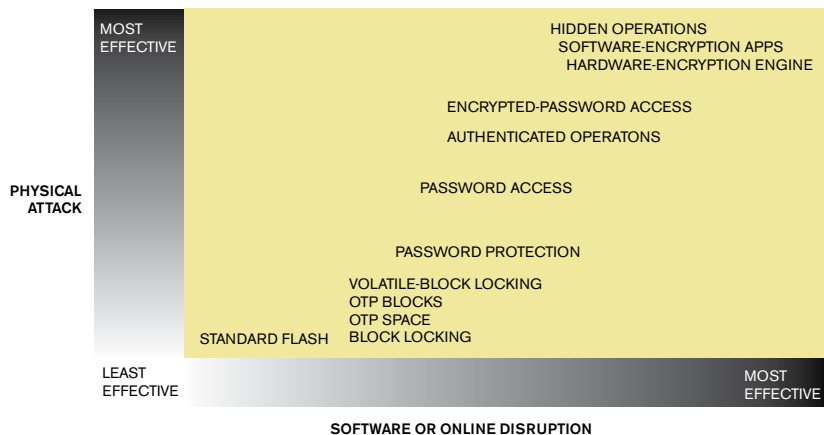


Figure 2 Various flash-security features can protect against cloning and intellectual-property theft. Note that adding such features increases system cost.

turer more time in the market before the clone can compete for revenue, and a deterrent may create enough delay for the original manufacturer to refresh its design before the cloner has an opportunity to produce a viable product. This built-in flash feature offers a cost-effective method for countering the revenue impact of IP loss.

Encrypted-password access for higher-level IP security is another method of protecting IP. A 64-bit password slows down a cloner or a service thief, but an encrypted password adds a significantly higher level of data protection. A few flash devices include an encrypted-password feature through an algorithm that manufacturers implement in silicon. The manufacturer encrypts passwords that pass from the flash and the microcontroller to the processor. The processor deciphers the passwords using the algorithm and confirms a match. A bus snooper, on the other hand, can only read—but not decipher—the encrypted password. Without the unencrypted password, the flash chip is unreadable, and the IP remains protected. Flash devices with an encrypted password typically cost more than those without one because manufacturers implement the algorithm on silicon, which adds to the component cost. However, the cost of the flash chip may be nominal compared with lost service revenue. For products with long refresh cycles, encrypted passwords are essential tools in preventing clones from reaching the market before the manufacturer updates its model.

OTP FLASH

Some flash devices include a system-level OTP area in which you can permanently lock the bits after programming. Once you lock the bits, a hacker cannot program or erase the blocks the bits are mapped to. The OTP has one factory-programmed segment with a unique, unchangeable number. The other segment is blank so that the designer can program it. Flash devices with OTP typically come in varying configurations of segments with as many as 2112 bits.

Several flash devices include an additional OTP feature that allows you to permanently lock blocks in the regular memory array. This implementation of OTP prevents modifications that disrupt system integrity. For instance, a service provider

typically adds its own code to a set-top box after delivery from the manufacturer. A designer cannot know how the service provider's code will affect the system. To protect the erasure or alteration of boot code, the designer can store the boot code in an OTP block and set it to be permanently locked. Then the service provider's code cannot write or erase to the locked block, thereby preventing those inconvenient customer-tech calls.

Most flash devices include some form of hardware-write-protection capability that can prevent programming and erasure of either a block or the entire device. Hardware-write protection works by setting a pin to a certain voltage, either through hard-wiring or by toggling a bit from an I/O pin on a processor. Before executing a modify command, the flash chip checks the pin that corresponds to program/erase protection. If the pin is not at the correct voltage for modification, the chip will not execute the command, and the code or data will not change.

When a valid voltage is present in the program-supply voltage, you can modify the blocks in the main array. If you ground the program-supply voltage, you cannot program or erase the blocks. When you ground the supply voltage, attempts to program or erase will fail, and grounding sets the appropriate status-register fail bit.

Another version of hardware-write protection protects the highest or lowest block against program and erase operations. To protect the highest or lowest block, set $V_{pp}/WP = V_{il}$, where V_{pp} is the program-supply voltage, WP is write protection, and V_{il} is the input-low voltage. In this situation, the block is in lock-down mode, and you cannot modify it. To remove a lock-down situation, set $V_{pp}/WP = V_{ih}$, where V_{ih} is the input-high voltage. In this case, you can lock or unlock the block.

HARDWARE PROTECTION

The hardware-based approach provides an inexpensive layer of protection against the malicious code that slithers in through the Internet. Malicious code cannot modify or erase data stored on a flash device that is locked at the hardware level unless it first resets the pin voltage. If a hacker tries to

bypass a router, for example, the malicious code will reach the flash device, and the device will check the pin for the block that stores the boot code. Finding the voltage at a level that doesn't allow modification, the chip does not execute the malicious code, and the router continues to work. Volatile- and nonvolatile-block-locking features use software commands to lock and unlock blocks, protecting data from inadvertent modification. In volatile-block locking, bits in a volatile array are mapped to main-memory-array blocks. You can individually modify, set, and clear these volatile-protection bits. However, they can protect only those blocks that you have not locked with nonvolatile-array bits. When you cycle the system power or reset the hardware, the volatile-protection bits revert back to their original unlocked or locked state.

Nonvolatile-block locking keeps blocks locked or unlocked, as the designer defines, even after a power cycle or reset. A nonvolatile-protection bit is mapped to and can individually lock each block. You can clear nonvolatile-protection bits through a clear-bits command or an erase command. You can use nonvolatile-block locking to ensure that blocks remain locked against inadvertent overwrites even after an unexpected power cycle or reset occurs.

Flash-security features vary by manufacturer and by device. You should consider these features, along with density, performance, technology, lithography, cost, size, and packaging. These factors all work together to enable a design, and you can use them to protect its performance and position in the market. Flash-security features offer an affordable, secure alternative to protecting IP, content, data, or system integrity. **EDN**

AUTHOR'S BIOGRAPHY

Bill Stafford is the director of segment marketing at Numonyx, where he develops application strategies and product requirements for the embedded-flash-memory market and enables the flash-memory ecosystem. A 25-year veteran in the electronics field, he has experience in product engineering, field quality, and marketing for flash memory, PCBs, systems, and aircraft components. Before joining Numonyx, Stafford spent 22 years at Intel Corp after time with the US Department of Defense.