

EEPROM

SURVIVAL OF THE FITTEST

BRIAN DIPERT, TECHNICAL EDITOR

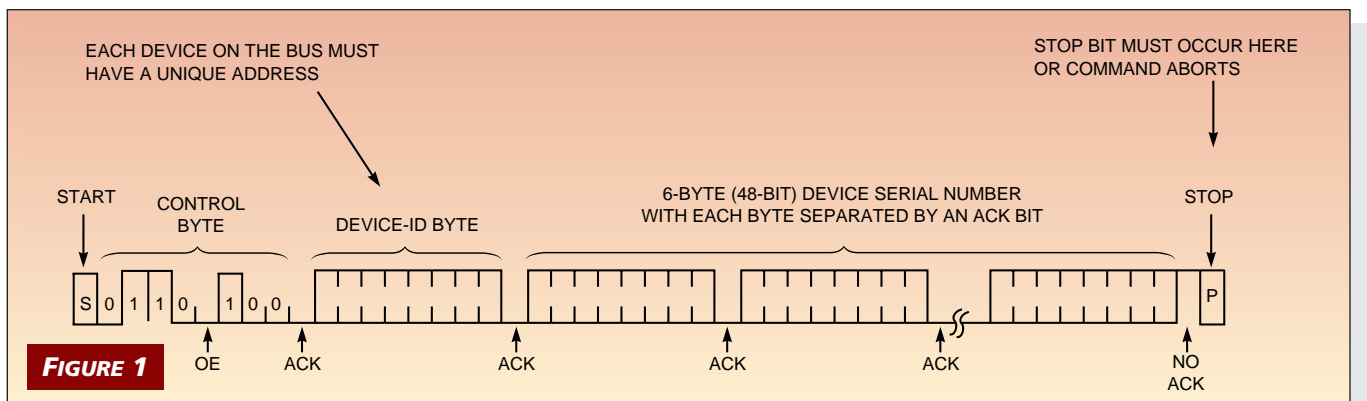
EEPROM is perhaps the most pervasive memory in all of electronics, a distinction that only SRAM can claim to match. Industry analysts focus their attention on cellular handsets and pagers, because these two applications consume a disproportionate percentage of total yearly EEPROM shipments. But the list of applications that use EEPROM is much longer, including networking equipment, universal remote controls, cordless phones, garage-door openers, cameras, automotive electronics, home audio and video, and smart cards. Think about how many devices you use every day that contain some amount of updatable data that remains intact even if you pull the power plug

EEPROMs are expanding and transforming their features to meet evolving application needs and competitive pressure. EEPROM technology is also diversifying and finding homes in some surprising places. Charles Darwin would be proud.

or take out the AA batteries. Odds are good that these products contain either an EEPROM or a battery-backed SRAM.

Unlike with flash memory, you can change EEPROM data from 1 to 0 and 0 to 1 on a bit-by-bit basis. EEPROM is also more cost-effective than SRAM at high densities and doesn't require constant power to retain stored information. Ferroelectric RAM (FRAM) holds promise as a long-term alternative but will need to overcome EEPROM's roughly 20-year head start in high-volume-production expertise to get there (Reference 1).

The first EEPROMs used an SRAM-like parallel interface, which remains the dominant pinout at highest densi-



Software-addressable I²C EEPROMs support unique 48-bit identifiers and as many as 255 devices on one bus (courtesy Microchip Technology).

EEPROMs AND DERIVATIVES

ties, although the market size is limited. A parallel interface has several advantages, including fast random read and write access to any location within the array, a straightforward hookup to the processor's address and data buses, robust hardware write protection, and no limitations on component density due to restrictive package dimensions. Also, a parallel interface is your only option if you plan to execute code directly from the EEPROM. However, many of today's EEPROM applications have board-space and height restrictions that prohibit use of the large parallel-memory packages. Parallel EEPROMs at 256 kbits and above also compete against flash memory, a more cost-effective alternative that, in combination with EEPROM-emulation software, delivers adequate read/write performance in some applications.

Serial-interface alternatives

Parallel EEPROMs also increasingly compete against their serial-interface brethren, causing several manufacturers to deprioritize or even abandon their parallel-EEPROM product lines in recent years. The more advanced the process lithography, the higher the density threshold below which the device's pin count, not the memory array, defines die size and cost. All other factors being equal, a five-pin serial device has a significantly smaller die and much lower power consumption than its 24-pin or larger parallel alternative. On-chip error-detection and -correction circuitry, a requirement for all high-density, high-cycling EEPROMs, is also more expensive to implement on parallel-interface devices and scales in complexity with density (see **box** "Tools and documentation simplify your job"). Increasing serial-interface bit rates narrow the read-performance gap between serial and parallel EEPROMs. If you frequently write to the EEPROM or the updated data stream is large, you may find that the memory's internal write speed, not the interface-transfer rate is the performance limiter. In addition to their smaller package and pinout, serial-interface EEPROMs require only a few signal lines between themselves and the system CPU or other logic. A two-

to-four-trace interconnection may be all the space you can spare in your next complex system-board layout.

I²C, developed by Philips for its 8051-variant microcontrollers, is the industry-standard serial-interface

@ a glance

- EEPROMs are a mature memory technology with widespread usage, but new applications and competition encourage vendors to innovate and diversify.
- Serial interfaces are growing in popularity at the expense of parallel-EEPROM alternatives.
- Power, performance, and predictability are among EEPROM's strengths compared with flash memory.
- EEPROM's analog heritage allows for its inclusion in interesting, unique, and innovative spin-offs.

option with the lowest pin count and the most popularity. It supports multiple masters and slaves and requires only three interconnections: a bidirectional serial-data/address signal (SDA, requiring open-drain or open-collector buffers), a serial-clock signal (SCL), and common ground signals. Slave devices on the 400-pF maximum-loading SDA line usually differentiate themselves with unique addresses, defined by tying device inputs high or low. I²C EEPROMs with no address-configuration inputs offer smaller die and reduced costs because of the lower active pin count but respond to all EEPROM I²C bus traffic.

I²C is unique in that it operates using a level-sensitive signal protocol. Most address, data, and acknowledgment transitions on SDA occur on the low phase of SCL, but two exceptions to this rule exist. When a master drives SDA from high to low on SCL's high phase, the action signifies a start condition. A low-to-high transition of SDA while SCL is high references a stop condition. I²C is a comparatively noise-intolerant approach for several reasons. These reasons include level-driven inputs, shared signal lines susceptible

to multidevice collisions, and the subsequent potential for incorrect decoding of read operations vs write operations and invalid detection of start and stop bits.

I²C originally had a 100-kHz maximum clock frequency, partially because of the delays required to minimize bus contention on the shared address/data line. Newer industry-standard EEPROMs run as fast as 400 kHz, and some vendor-proprietary parts operate even faster. Philips just announced an I²C spec enhancement that enables devices with different operating voltages to share a common bus. Also, the original I²C specification address scheme limits the total EEPROM density (on one chip or spread over multiple chips) to 16 kbits and the number of EEPROMs to eight. These limits existed because these first-generation EEPROMs used a common 1010 device-type identifier in the 8-bit command byte, which also includes a read/write bit, followed by an 8-bit location address. Numerous enhanced I²C schemes boost density, some by using multiple location-address bytes and others by replacing the device-type identifier with extra address bits. These incompatible improvements make the schemes unusable in some multichip designs.

Microchip Technology's 1-kbit 24LCS61 and 2-kbit 24LCS62, the first members of the company's software-addressable I²C EEPROMs, are the latest attempt to bring some order to this chaos. This I²C enhancement, which Microchip is submitting to JEDEC and other industry-standards bodies for review, enables dynamic connection of as many as 255 devices, practically limited by maximum bus-loading specifications. Combining 255 devices per bus and an 8-bit address specifying the location to be accessed within the device results in a maximum EEPROM density of just less than 512 kbits. High-density EEPROMs can claim and respond to multiple software device IDs. Software-addressable I²C also specifies a 48-bit serial number, unique to each device and programmed by the manufacturer before shipment (**Figure 1**). Software-addressable and standard I²C devices can coexist on the same

bus, because the 24LCS61 and 24LCS62 respond to a 0110 device-type identifier. Microchip considers large multi-board systems, such as data-communications and telecommunications equipment, as a potential application for software-addressable I²C EEPROMs.

Microwire, the oldest and the slowest growing of the three industry-standard serial interfaces, was developed by National Semiconductor. Conceptually, Microwire is similar to its alternative, serial peripheral interface (SPI). Both SPI and Microwire require a minimum of four interconnect traces. data-in (DI) and data-out (DO) pins can connect to each other, although brief bus contention on each read cycle occurs when the last address bit is 1. Inputs latch on the rising edge of CLK, an active-high chip select (CS) allows the master to enable a specific slave, and a common GND completes the interface. Some Microwire EEPROMs also provide an organization (ORG) pin, which, when high, enables you to read and write 2 bytes at a time, and a program-enable (PE) input, intended for hardware write protection.

Inherent in Microwire's active-high inputs is its data-corruption Achilles' heel. Signals can float to a high state in the absence of a device that actively

drives them, such as during system reset and power transitions. Unfortunately, the valid erase command also comprises a string of 1s followed by the desired device address. If you occasionally find the last location within your Microwire EEPROM set to all 1s, you've probably encountered this problem, caused by glitches on CLK. Speaking of CLK, the highest frequency offered today on a Microwire EEPROM is approximately 2 MHz. A lack of widespread direct support in embedded processors and controllers, as well as density limitations similar to those in I²C, also restricts Microwire's popularity.

SPI is less prevalent than both I²C and Microwire when you consider total unit shipments, but SPI is the fastest growing in popularity of the three interfaces. Because of its high performance, SPI is also the bus of choice for high-density serial EEPROMs, in which bandwidth is most critical. SPI EEPROMs run as fast as 5 MHz, and Atmel's Serial DataFlash, a model of EEPROMs to come, offers speeds to 10 MHz. SPI's growing success also derives from the bus's inclusion on a number of embedded processors from Motorola (Schaumburg, IL), the company that developed SPI.

SPI's signals have different names

from their Microwire counterparts, but the functions are similar: serial in (SI), serial out (SO), serial clock (SCK), and chip select (CS). One key difference is that SPI EEPROMs latch inputs on SCK's falling edge, a characteristic that also improves SPI's head room to run at high clock frequencies. The SPI specification defines a maximum of only four slaves for each master. SPI EEPROMs include write protect (WP) and hold, both active-low inputs. Addressing Microwire's write-protection shortcomings, SPI command sequences also have much longer bit strings, and none of them is all 0s or all 1s.

Respond to the flash challenge

Over the past several months, a number of flash-memory vendors have introduced devices that attempt to integrate EEPROM functions by using various hardware and software techniques (References 2 and 3). Vendors target these devices primarily at cellular phones and pagers, which together constitute roughly half of the total flash-memory-market revenue and roughly one-fourth of total units shipped. The devices take advantage of the fact that wireless system code rarely fills an entire 8- or 16-Mbit flash memory and that today's phones use rela-

TOOLS AND DOCUMENTATION SIMPLIFY YOUR JOB

An impressive assortment of application notes and reference hardware and software is available from a number of EEPROM vendors. Instead of doing all the work yourself and scratching your head when the design doesn't function as you intended, spend a little time surfing the vendor Web sites and save yourself the debugging headaches. You'll find an impressive assortment of application notes and reference hardware and software available from a number of EEPROM manufacturers.

Microchip Technology's Serial EEPROM Designer's Kit (SEVAL) includes a programming board and a number of device samples. It communicates with a PC via the serial port (the kit includes a cable) and lets you write any desired information to the memory, as well as test the memory's cycling capability. When you debug your CPU's interface to the EEPROM, it's good to start with a known data pattern, and the Serial EEPROM Designer's Kit is ideal for this reason. I installed SEVAL and had it up and running in less than 5 minutes.

Speaking of cycling, the company's Windows-based Total Endurance Utility lets you graphically and numerically see how

EE-PROM useful life and part-per-million failure rate vary with density, cycling mode, data pattern, voltage, temperature, and rewrite amount and frequency. After running it, I found myself wishing that other EEPROM and flash-memory vendors were as forthcoming with their reliability information!

If you'd like to evaluate flash memory as a potential EEPROM replacement, Intel's Flash Data Integrator (FDI) software is a good place to start. The developer's kit bundles source code, application notes, device samples, and a thick user's guide. FDI includes both the background file-system software and a foreground interrupt manager, because Intel's Advanced Boot Block flash memories don't support simultaneous read-while-write functions. AMD is working with DataLight (Arlington, WA) and its FlashFX file system to emulate EEPROM. Other flash-memory companies rely on Intel's user-modifiable FDI; if the memory supports reading of code blocks in parallel with the program or erasing of data sectors, you don't need to worry about the foreground media manager. You can run an interactive demo of FDI on Intel's Web site, www.intel.com.

EEPROMs AND DERIVATIVES

tively small 16- to 64-kbit EEPROMs. Nothing particularly unique about the architectures of these enhanced flash memories precludes their use in other applications, however.

Integrating two chips into one solves a key portable-equipment concern: board space. Data-read accesses can *potentially* be faster, because the data-storage portion of the flash memory shares the existing code memory's parallel interface. However, EEPROM-emulation software and file-format overhead, which are especially slow when you use a linked-list file system, may preclude any performance improvement. Also, if the flash memory doesn't allow reads from the code portion while programming or erasing the data sectors, you have to add in the sometimes unpredictable delay necessary to suspend the internal state machine. Even if the communications protocol specifies a known time interval between interrupts, you must consider the potential for asynchronous interrupts caused by user interaction. If you're considering using a flash memory that emulates read-while-write using suspend/resume functions, you must ensure that you can immediately service interrupts via careful silicon and software characterization, delay interrupt servicing as a lower priority function, or selectively block interrupts during data or code updates. You also must include necessary external RAM in your design to execute the suspend/resume routine and file-management software.

The other controversial selling point of the "two-in-one" flash memory is cost savings. Is the integrated flash memory cheaper than a separate code flash memory and data EEPROM? The answer depends on your design, how much EEPROM you're replacing, the average data-file size that you rewrite with each update operation, and how often you execute an update. All other factors being equal, a flash memory at a given density is cheaper than an equivalent EEPROM. However, flash memory can't match EEPROM's full-byte alterability. EEPROM emulation uses available space less efficiently, meaning that you need more flash-memory density than you would need

for the equivalent function done in EEPROM. The more data you update, coupled with increases in how often you update the data, the more flash-memory space you need to reserve to avoid excessive media cycling and slow block erase. You also need room in the flash memory to store the EEPROM-emulation algorithms.

Today's cellular phones store four main types of data in EEPROM: never-

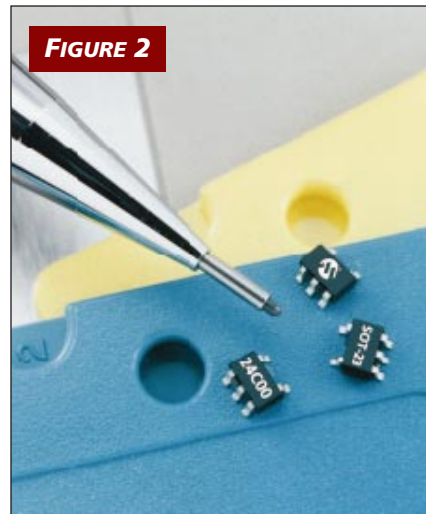


FIGURE 2

SOT-23 squeezes 128 bits of EEPROM into a package that is 0.110 in. long, 0.059 in. wide, and 0.035 in. high (courtesy Microchip Technology).

updated system information initialized during phone manufacturing; rarely updated service-feature data, such as the extent of your allowable coverage area; occasionally updated phone numbers and other user information; and frequently updated cellular-protocol parameters. The phone rewrites the last type of data each time the phone connection transfers from one base station to another, even when the phone is in standby. Newer phones add pager-like incoming and outgoing text-message functions, such as Short Message Services, as well as local storage of incoming voice-mail messages. This function is especially important in Europe, where travel between countries, each with its own telecomm providers, is common. Although the average amount of stored data may grow in the future, the average rewrite frequency may decrease because of the

characteristics of what the system is writing. This combination would result in flash memory's becoming a more attractive EEPROM alternative. Nokia's (Finland) Japan PDC cellular phone is one of the first wireless units to store voice data and execute code from one flash memory—in this case, Intel's (Folsom, CA) Smart 3 Advanced Boot Block architecture.

EEPROM companies are doing all they can to avoid being rendered obsolete by flash memory. To address board-space concerns, they squeeze EEPROMs into smaller packages, even as they increase density. For example, Microchip Technology's 24C00, a 128-bit I²C EEPROM, comes in a five-lead SOT-23 package (Figure 2). The company's higher density 24C128 128-kbit I²C EEPROM provides two compact package options: an eight-lead, 150-mil SOIC, and a low-profile, 14-lead TSSOP. Several companies also offer EEPROMs in bare-die form, an option that's attractive in ultrathin smart cards. Chip-scale packaging, under development by several EEPROM vendors including SGS-Thomson, will soon deliver another ultrasmall alternative, close to die in size but much easier to handle.

Where these packages are still too large and for designs that value fast parallel-EEPROM access, Atmel, SGS-Thomson, and Waferscale Integration (WSI) have developed chips that combine flash memory and true EEPROM on one die. Atmel was first to market with its ConcurrentFlash product line in 1996. The 29C432, the first—and still the only—member of the family, combines 4 Mbits of flash and 256 kbits of EEPROM. SGS-Thomson recently announced its first Flash+ memory, the M39432, also combining 4 Mbits of flash and 256 kbits of EEPROM (Figure 3 and Reference 4). Available in a 40-lead TSOP, the M39432 will also come in a BGA package beginning next year. SGS-Thomson's Flash+ process uses a simple, one-transistor cell for the flash-memory portion of the array. This first product in the Flash+ family contains duplicate peripheral circuitry, such as decoders, state machines, and sense amplifiers, for both memories on the die. However, Mario Licciardello, SGS-

EEPROMs AND DERIVATIVES

Thomson's general manager for non-volatile- and application-specific-memory, indicates that future devices will combine functions wherever possible to further reduce the memory's cost and power. Future Flash+ memories will also include flash-memory densities of 2 to 8 Mbits, with EEPROM sizes of 64 to 256 kbits.

WSI combines 128 kbytes of flash memory and 256 kbits of parallel EEPROM on the company's PSD8XXF1, which also includes 2 kbytes of SRAM, user-programmable logic, and a Joint Test Action Group (JTAG) interface. WSI divides the EEPROM into four 8-kbyte blocks, which you can use for both data storage and system boot code. The company developed the PSD8XXF1 in conjunction with SGS-Thomson and uses the Flash+ technology. On-chip decoding circuitry and dynamic address-remapping capability enable code execution out of the EEPROM coincident with update of the flash-memory array, even with μ Cs that don't directly allow code fetches from the data portion of the system-memory map. One of the chief shortcomings of a single-die integrated EEPROM and flash memory is its lack of flexibility compared with a two-chip alternative. Unless your memory-density requirements exactly match those offered in a single-die combo chip, the chip will be too size-restrictive, or you'll pay extra for memory bits you don't need.

One-package, two-die alternatives, such as Atmel's Combo Flash and devices from White Microelectronics, are similarly application-specific and perhaps have higher power and cost. However, these alternatives enable the semiconductor vendor to choose from a variety of stand-alone EEPROM and flash memories in meeting your density needs. Even a chip that emulates EEPROM in flash memory is more flexible, because any of the small blocks not used for data are available for lower cost code storage. Because memory cost derives from both die size and volume, a commodity memory may cost less than its more proprietary alternative, even if the commodity memory has a larger and more complex architecture. When combining code and data stor-

age in one chip, you must ensure that data writes don't accidentally corrupt system software, or you could face a costly recall, replacement, and redesign.

EEPROM vendors also see power consumption as a key differentiator for their products. Although flash memories that read and write at 1.8V have only recently emerged (for example, AMD's (Sunnyvale, CA) AM29SL800),



Figure 3
The Flash+ M39432 includes 4 Mbits of flash memory and 256 kbits of parallel EEPROM on one die, operating concurrently at 3.3V and delivering 120-nsec read performance (courtesy SGS-Thomson).

1.8V EEPROMs have been available since 1992. Seiko Instruments recently announced sampling of 0.9V, 512-bit and 1-kbit EEPROMs. Combined with the low complexity, and consequently the low power, of a serial-interface memory, these EEPROMs require only nanoamps of standby current and less than a milliamp of read and write current. One of Microchip Technology's application notes even discusses how to directly power EEPROMs from the μ C port pins or other logic outputs.

Serial EEPROMs use the clock signal only for inputting and outputting data, so you can run them as slow as you like, make the clocks asymmetrical, or even stop the clocks for power management, even when the EEPROM is internally rewriting data. Keep in mind, though, that you need to consider both power and energy consumption when evaluating serial vs parallel memories. The low-bandwidth serial-EEPROM interface means that read and write bus

cycles take longer than parallel equivalents, and time is a key part of the energy equation. Longer bus cycles also mean that other system logic, such as ASICs and the μ P, must also remain "awake" longer. Watch out, too, for multidevice data collisions on I²C and Microwire, because these collisions also increase dynamic power consumption.

Predictable write performance is another notable EEPROM attribute. When you emulate EEPROM with flash memory, the time needed to update data can vary widely from one series of writes to the next (Reference 5). Update time depends on how frequently the flash-file software needs to erase a block and how effectively the software can hide block erase as a background task. You could always buffer incoming data using an external SRAM or FIFO buffer to alleviate system-performance impacts. However, this approach increases system cost if you don't have sufficient extra memory bits or logic gates available in your design.

In contrast, you can rewrite individual EEPROM locations in a fixed amount of time, regardless of whether you change 0s to 1s or 1s to 0s. While the EEPROM executes an internal rewrite cycle, the μ P can handle other system tasks in parallel, thereby maximizing efficiency. EEPROM write performance slowly and predictably degrades with increased byte cycling (the number of rewrites). Many EEPROMs support small, 16- to 64-byte page-buffer writes with lower bus overhead than a string of single-byte writes. This minimization results in even faster average rewrite times for multiple sequential data bytes. On-chip address counters are another EEPROM performance enhancement. These counters automatically increment the address for the next access if you don't explicitly reinitialize them with the corresponding address bit-stream overhead after a read or write.

EEPROM vendors have also developed schemes to guard against unintended rewriting of stored information during system power transitions or due to system noise on control signals. Longer and more complex command bits reduce the probability that the EEPROM will decode a random glitch

EEPROMs AND DERIVATIVES

event as a valid opcode, although they also increase overhead. Most EEPROMs also specify a lockout voltage below which the memory does not respond to intended or unintended data-alteration attempts. As vendors increase the usable operating range of their parts, however, the lockout voltage becomes less meaningful. Internal noise filters and input Schmitt triggers enable the memory to ignore short-duration control-input pulses. Write-protect pins block rewrite of the entire array, which is effective in conjunction with the power-good output of the system power supply or monitoring circuits. Additionally, it's always a good idea to reset the EEPROM (via a start-stop bit sequence in I²C, for example) as part of your system's normal power-up and reset algorithm.

Proprietary enhancements

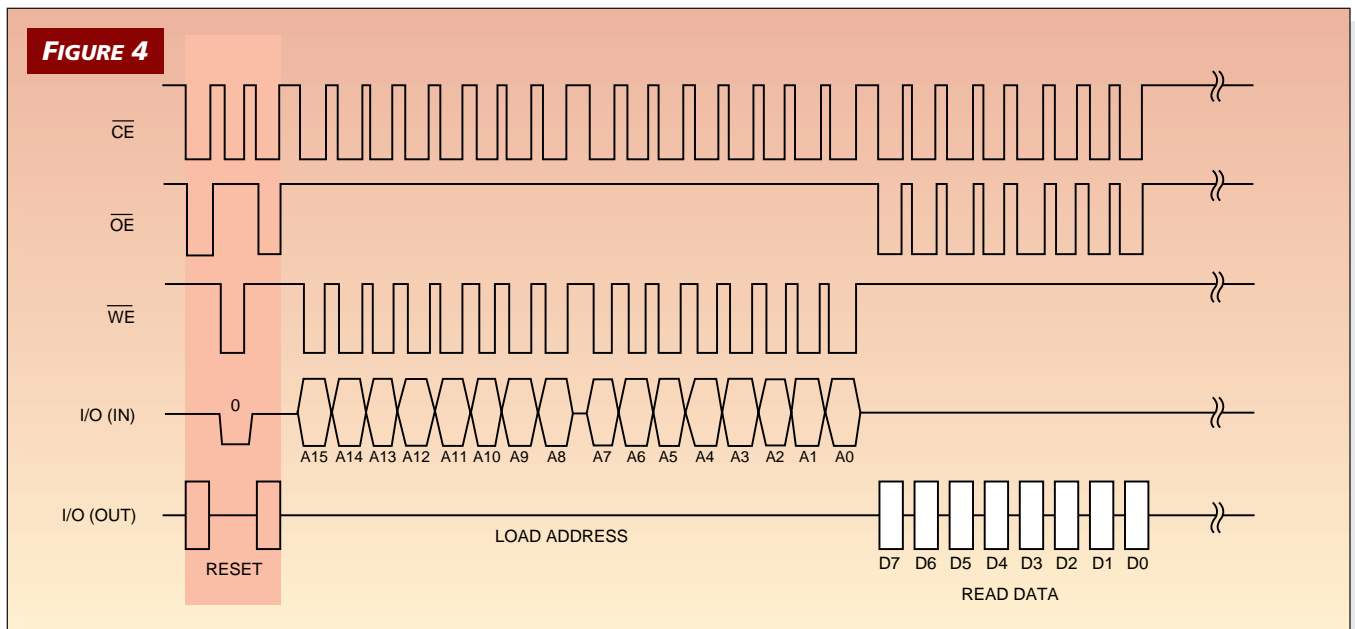
Block locking is another technique to prevent unintentional and malicious corruption of EEPROM contents. This approach is also useful in cases when the system processor makes an invalid jump out of normal code execution and into the midst of the EEPROM update algorithm. Without block locking, the processor might scramble EEPROM contents before the watchdog timer

restores order. In some cases, you find vendor-to-vendor standardization on block-lock sizes, locations, techniques, and commands. However, this luxury will become less common, because the EEPROM manufacturers are all striving to differentiate themselves and capture your business. This assortment of incompatible approaches unfortunately can cause engineers to design to the least common denominator: no block locking. For example, Microchip Technology's EEPROM Marketing Manager Bryan Liddiard, estimates that no more than 10% of the chips his company ships each year feature block locking. Two cases in which you *will* find compatibility are serial-presence-detection chips for DRAM modules—an extrapolation of the previous SIMM jumper-identification scheme—and VESA plug-and-play EEPROMs. Both options ensure compatibility, because industry-standards committees developed the memory definitions.

Serial-presence-detection chips are otherwise-standard I²C EEPROMs that contain an additional device-type-identifier-driven command. This command irreversibly locks the lowest 128 bytes of the array from subsequent alteration (JEDEC standard 21-C). A module supplier issues this command

after first storing various information, such as DRAM type, speed, organization, and manufacturer. Hardware write protection of the entire array via an input pin can supplement this partial array software-protection scheme. VESA plug-and-play EEPROMs, similarly based on I²C, enable a computer monitor to communicate its attributes—such as size, dot pitch, video settings, and preferred scan frequencies at different resolutions—to the graphics card and other subsystems. This communication can be one-way (DDC1 (Display Data Channel 1)) or bidirectional (DDC2, normally backward-compatible for DDC1 graphics cards). Bidirectional communication enables the system to configure the monitor. Noise immunity is especially critical in the monitor's hostile EMI environment. More complex VESA chips have dual ports, allowing simultaneous chip communication with the monitor's μ C over I²C and with the rest of the system over the ACCESS.bus.

Xicor's Block Lock, Block Decode, and IDLock memories contain non-volatile bits that enable write protection on quarter-array or even finer granularity. The Xicor X76F product family goes one step further by gating array access with passwords. For exam-



Micro Port Saver EEPROMs support 10-Mbps bus-transfer rates and offer minimal-glue interfaces to embedded μ Ps without adequate I/O pins (courtesy Xicor).

EEPROMs AND DERIVATIVES

ple, the X76F128 contains a 128-kbit main section and a 512-bit auxiliary array. Four separate and unique 64-bit passwords protect the main array and auxiliary array from both read and write operations. Eight invalid password attempts result in automatic erasure of both arrays. A fifth 64-bit password restores normal device operation.

Xicor also sells the X76F128, as well as other product-family members, in a smart-card form factor.

Taking advantage of its European presence and the smart-card popularity in that part of the world, SGS-Thomson has focused on highly secure EEPROMs. The ST13xx families, intended for prepaid phone-card usage, contain

several distinct subarrays, hardwired security logic and fuses, and an optional authentication secret key. Atmel and Microchip Technology also supply smart-card and RFID ICs (Reference 6). Atmel's AT88SC10X devices keep track of the number of incorrect security-code attempts and permanently invalidate the memory after four consecu-

FOR MORE INFORMATION...

For more information on the products listed in this article, circle the appropriate numbers on the Information Retrieval Service card or use EDN's Express Request service. When you contact any of the following manufacturers directly, please let them know you read about their products in EDN.

Advanced Linear Devices⁵

Sunnyvale, CA
1-408-747-1155
fax 1-408-747-1286
www.aldinc.com
Circle No. 322

Analog Devices⁵

Norwood, MA
1-781-329-4700
fax 1-781-821-4273
www.analog.com
Circle No. 323

Atmel^{1, 2, 4, 9, 10, 12}

San Jose, CA
1-408-441-0311
fax 1-408-436-4300
www.atmel.com
Circle No. 324

Catalyst Semiconductor^{1, 2, 3, 5, 7, 10, 11}

Sunnyvale, CA
1-408-542-1000
fax 1-408-542-1402,
www.catsemi.com
Circle No. 325

Dallas Semiconductor^{5, 7}

Dallas, TX
1-972-371-4000
fax 1-972-371-3717
www.dalsemi.com
Circle No. 326

Fairchild Semiconductor^{1, 8, 9, 10, 11}

Santa Clara, CA
1-408-721-7989
fax 1-408-721-3790
www.fairchildsemi.com
Circle No. 327

Hitachi²

Brisbane, CA
1-415-244-7848
fax 1-415-583-4207
www.hitachi.com
Circle No. 328

Information Storage Devices⁶

San Jose, CA
1-408-369-2400
fax 1-408-369-2422
www.isd.com
Circle No. 329

Integrated Silicon Solution¹

Santa Clara, CA
1-408-588-0800
fax 1-408-588-0805
www.issiusa.com
Circle No. 330

Microchip Technology^{1, 2, 9, 10, 11}

Chandler, AZ
1-602-786-7200
fax 1-602-899-9210
www.microchip.com
Circle No. 331

National Semiconductor⁷

Santa Clara, CA
1-408-721-5000
www.national.com
Circle No. 332

Philips Semiconductors¹

Santa Clara, CA
1-408-991-2207
fax 1-408-991-2311
www.philips.com
Circle No. 333

Seiko Instruments USA¹

Torrance, CA
1-310-517-7770
fax 1-310-517-7792
www.seiko-usa-ecd.com
Circle No. 334

SGS-Thomson Microelectronics^{1, 2, 3, 4, 9, 10, 11}

Lincoln, MA
1-617-259-0300
fax 1-617-259-9423
www.st.com
Circle No. 335

Silicon Storage Technology²

Sunnyvale, CA
1-408-735-9110
fax 1-408-735-9036
www.ssti.com
Circle No. 336

Simtek³

Colorado Springs, CO
1-719-531-9444
fax 1-719-531-9481
www.simtek.com
Circle No. 337

Space Electronics²

San Diego, CA
1-619-452-5499
fax 1-619-452-4167
Circle No. 338

Summit Microelectronics^{1, 5, 7}

Saratoga, CA
1-408-867-8960
fax 1-408-867-8970
www.summitmicro.com
Circle No. 339

Turbo IC^{1, 2}

San Jose, CA
1-408-324-0288
fax 1-408-324-0289
www.turbo-ic.com
Circle No. 340

Waferscale Integration⁴

Fremont, CA
1-510-656-5400
fax 1-510-657-5916
www.wsipsd.com
Circle No. 341

White Microelectronics^{3, 4}

Phoenix, AZ
1-602-437-1520
fax 1-602-437-9120
www.whitemicro.com
Circle No. 342

Winbond Electronics^{1, 2}

Hsinchu, Taiwan
+886-3-5770066
fax +886-3-5792668
www.winbond.co.tw
Circle No. 343

Xicor^{1, 2, 3, 5, 7, 10}

Milpitas, CA
1-408-432-8888
fax 1-408-432-0640
www.xicor.com
Circle No. 344

VOTE . . .

Please also use the Information Retrieval Service card to rate this article (circle one):
High Interest 594
Medium Interest 595
Low Interest 596

Super Circle Number

► For more information on the products available from all of the vendors listed in this box, circle one number on the reader service card.

Circle No. 345

¹Serial EEPROMs; ²Parallel EEPROMs, PCMCIA attribute memory; ³Nonvolatile RAMs; ⁴Flash-plus-EEPROM memories; ⁵Programmable analog circuits, digital potentiometers, ADCs, and DACs; ⁶Audio record/playback devices; ⁷CPU, memory, and system supervisors, including devices that integrate reset-temperature-voltage monitoring and real-time-clock, and watchdog-timer functions; ⁸ISA plug-and-play controllers; ⁹VESA plug-and-play memories; ¹⁰Secure EEPROMs, encryption chips, and smart-card ICs; ¹¹Serial-presence-detection memories; ¹²FPGA configuration memories.

EEPROMs AND DERIVATIVES

tive invalid accesses. For the ultimate in security, Microchip Technology's KeeLOQ EEPROM-based encoders and decoders use sophisticated, 64-bit, key-code-hopping algorithms. Microchip's devices are for use in remote keyless-entry systems; vehicle alarms; automatic garage-door openers; and other remote-controlled, high-security systems. Fairchild Semiconductor's HiSeC chips fall into this category of high-security, rolling-code generators.

In addition to I²C, Microwire, and SPI, other proprietary low-pin-count devices are available from several EEPROM companies. These devices usually have generic names, such as Two-Wire and Four-Wire. You can connect these chips to almost any microcontroller's port pins, although the chips don't normally support sharing the bus with other devices. They offer fewer functions than industry-standard alternatives but may deliver lower cost, higher bandwidth, or both. If multi-vendor-supply capability is important to you, however, carefully evaluate functional compatibility among manufacturers' devices before proceeding down this path. (You should do so even if the companies use the same marketing name.) Atmel's AT17CXX FPGA configuration chips are pinout- and functionally compatible with other vendors' ROMs and EPROMs but offer in-system reprogrammability as many as 10,000 times. The chips operate at 3.3 and 5V and come in 65-, 128- and 256-kbit densities.

Xicor's Micro Port Saver (MPS) serial interface targets embedded processors, such as x86 CPUs and DSPs, that don't directly offer a large number of I/O pins (Figure 4). Instead of using a more expensive parallel EEPROM or creating a port from external logic, you can directly connect MPS EEPROMs to your processor's data bus and read/write control signals, in combination with a decoded address chip select. With serial-data bandwidth as high as 10 Mbps, MPS EEPROMs can run even faster than today's SPI memories. The primary complication in using MPS devices is software: You must use repeated register-shift commands to reconstruct within the CPU the data byte read from the EEPROM, and you must also decon-

struct that data before writing to the EEPROM. Dedicated microcontroller I²C, Microwire, and SPI ports automatically do this parallel-to-serial or serial-to-parallel conversion for you in hardware, and a parallel EEPROM handles information in its native byte form.

Xicor's 68C and 88C EEPROMs directly connect to a processor's multiplexed address/data bus, such as that in Intel 8051 and Motorola 68HC11 μ Cs,

Instead of adding EEPROM to another memory, such as flash memory, why not include it on the system CPU?

without the need for address latches. You still might need latches, though, to properly decode external peripheral chip selects. These memories, similar to WSI's PSD8XXF1, support simultaneous data read or code execution from part of the EEPROM array while you rewrite another portion. Some 68C and 88C members also reconstruct lost ports to create the controller's external address/data bus. Other 68C/88C EEPROMs support automatic programming without processor intervention via the Self Loading Integrated Code (SLIC) protocol.

Analog on EEPROM foundation

Instead of adding EEPROM to another memory, such as flash memory, why not include it on the system CPU? This approach is certainly possible, and a number of embedded-controller manufacturers offer versions of their devices with EEPROM onboard. However, you'll probably find that unless your EEPROM density needs are small, a two-chip approach is the more economical alternative. Manufacturing EEPROM is more complex than manufacturing a standard memory or logic process because EEPROM involves additional metal or polysilicon layers

(or both), more fabrication steps, and thicker silicon oxides.

Although EEPROM is not fiscally conducive to the inclusion of other "pure" digital circuitry, it is analog-friendly. Rich Palm, marketing vice president at Summit Microelectronics, points out that commodity EEPROM processes as a matter of course have good capacitors, resistors, and diodes for the memory's internal program and erase subsystems—three main ingredients for analog circuits. He also observes that more than 50% of all μ Cs have a serial EEPROM sitting next to them in the system, and these designs also commonly include basic analog devices, such as voltage monitors, ADCs, and DACs. Some analog manufacturers, such as National Semiconductor with its CPU supervisors, use EEPROM cells to retrim internal circuits after high-temperature die encapsulation.

Palm points out that if the EEPROM is already on the analog chip, and if the system design engineer is likely to want some EEPROM anyway, it makes good business sense for manufacturers to add a little more user-accessible EEPROM to their chips. Summit Microelectronics and other companies offer a number of devices that fit this description, including voltage monitors with precision voltage thresholds, active high and low reset outputs, and customer-specific reset delays. Some devices also respond to external reset input transitions, and still others include watchdog timers or real-time clocks. Summit's latest part, the S39421, is for use in hot-swap-board environments, such as Compact PCI and VME. The S39421 energizes the board's 3, 5, and 12V lines with valid input voltages present and in response to a detected card-insertion event. Delays from insertion detection to voltage output, as well as the output-voltage ramp slew rate, are programmable using EEPROM bits. The S39421 also includes 1 kbit of user-defined Microwire EEPROM, potentially for PCI configuration storage, and a data-dumper mode for downloading configuration data to the card's ASIC or controller.

Dallas Semiconductor's DS1624 is a 13-bit digital temperature sensor with 256 bytes of on-chip EEPROM for stor-

EEPROMs AND DERIVATIVES

ing various thermal-compensation or other data. Several companies also offer EEPROM-based digital potentiometers (Reference 7). These parts may not handle the high power of some of their mechanical counterparts and also restrict you to a limited number of discrete settings. These limitations contrast with the seemingly infinite number of continuous settings within a range in a mechanical potentiometer. However, you can set digital potentiometers electronically rather than with a screwdriver, perhaps with test equipment in the system-manufacturing line. You can also in-system-modify the potentiometers' values without disassembly or a technician service call. DACs with integrated EEPROM, if they automatically download EEPROM values to the DAC inputs on power-up, are an alternative to resistor-based digital potentiometers.

Other analog ICs based on EEPROM technology include ISD's ChipCorder audio-storage and -playback devices (the company is migrating to lower cost flash memory in the future), Advanced Linear Devices' Electrically Programmable Analog Devices (EPADs), and even oscillators (Reference 8). EPADs are dual or quad op amps that offer individually user-programmable offset voltages within a 2000-mV range and with 0.1-mV resolution. Fairchild Semiconductor has taken EEPROM technology in a different, distinctly nonanalog direction with its NM95MS ISA plug-and-play controllers. Four family members support varying numbers of logical devices, DMA and interrupt modes, single-chip integration, and Windows 95 certification compatibility. The NM-95MS18, via its N_PNP input, also handles non-plug-and-play systems, such as Windows NT and 3.1, Unix, and plug-and-play-unaware BIOSes. The alternative, a plug-and-play enabler built into the device driver supplied with the ISA board for these operating systems, can create havoc in multi-board systems. However, if Intel and Microsoft have their way as documented in the PC '98 specifications, the ISA bus will soon cease to exist in PCs.

Several other memory types also build on an EEPROM core. Nonvolatile RAMs (NVRAMs) combine an SRAM

array with a matching-density EEPROM array on one chip. NVRAMs offer fast parallel reads and writes (serial NVRAMs also exist) and infinite SRAM write cycling, with EEPROM as a limited-cycle nonvolatile data backup. Some chips also called NVRAMs don't include EEPROM but rely on a battery for SRAM data retention, which can be confusing. Some EEPROM-based NVRAMs require the user to issue software commands to the part to transfer

If Intel and Microsoft have their way as documented in the PC '98 specifications, the ISA bus will soon cease to exist in PCs.

EEPROM contents to and from the SRAM array. Other devices automatically copy EEPROM contents to the SRAM on power-up. These devices also write SRAM information back to the EEPROM in response to a hardware input-active transition (often an external monitoring circuit's early detection of power loss).

Simtek's AutoStore NVSRAMs integrate voltage monitoring with precise thresholds, and you can supplement system power with a large external capacitor to ensure a complete SRAM backup. NVSRAMs also include several levels of protection to minimize the probability of unintended store and recall operations. Xicor's SerialData Flash memories are EEPROMs with some features untested. Flash memories from companies such as Atmel, Bright Microelectronics (Sunnyvale, CA), Nexcom Technology (Sunnyvale, CA, now a division of Integrated Silicon Solution), Sandisk (Sunnyvale, CA), Silicon Storage Technology, Turbo IC, and Winbond are EEPROM-like in their array and periphery designs. These memories offer lower cost than full-featured EEPROM alternatives but

give up byte-level-rewrite capability in exchange. Their erase blocks, however, tend to be among the smallest of all flash memories. EDN

References

1. Dipert, Brian, "FRAM: ready to ditch niche?" *EDN*, April 10, 1997, pg 93.
2. Dipert, Brian, "Flash device targets mixed code and data storage," *EDN*, April 24, 1997, pg 16.
3. Dipert, Brian, "Read-while-write architectures come with a hardware twist," *EDN*, May 22, 1997, pg 11.
4. Dipert, Brian, "Flash targets small footprint, low voltage," *EDN*, Oct 9, 1997, pg 26.
5. Dipert, Brian, "Data storage in a flash," *EDN*, July 3, 1997, pg 65.
6. Schweber, Bill, "RFID components add sophistication, features," *EDN*, Dec 4, 1997, pg 11.
7. Schweber, Bill, "Gain control goes silicon," *EDN*, May 11, 1995, pg 113.
8. Travis, Bill, "Crystal oscillators: looking good in wireless systems," *EDN*, Nov 20, 1997, pg 38.
9. Axelson, Jan "Using serial EEPROMs: part 1: general principles," *Circuit Cellar INK*, July 1997, pg 66.
10. Axelson, Jan "Using serial EEPROMs: putting it all together," *Circuit Cellar INK*, August 1997, pg 70.

Acknowledgments

Rich Palm and Rick Orlando from Summit Microelectronics provided me not only useful information on integrating analog circuitry with EEPROM, but also a historical perspective of the EEPROM industry from their numerous years of experience at several companies. Thanks also to Eric Sells and the EEPROM folks at Microchip Technology. Finally, I learned a great deal about proper PC initialization from Eric Roesinger's (Thomson Consumer Electronics, Indianapolis, IN) postings on the Internet newsgroup comp.arch.embedded.

You can reach Technical Editor Brian Dipert at 1-916-454-5242, fax 1-916-454-5101, edndipert@worldnet.att.net, <http://members.aol.com/bdipert>.