Achieving Energy Efficiency With EFM32 Gecko Microcontrollers

By Tom R. Halfhill
Senior Analyst

December 2014

The Linley Group

www.linleygroup.com

Achieving Energy Efficiency With EFM32 Gecko Microcontrollers

By Tom R. Halfhill, Senior Analyst, The Linley Group

This paper describes Silicon Labs' EFM32 Gecko microcontroller family. This large product portfolio of 32-bit MCUs includes more than 240 devices based on the ARM Cortex-M0+, Cortex-M3, and Cortex-M4 processor cores. Despite this variety, all EFM32 MCUs are software compatible with each other, and EFM32 chips with the same package configuration are pin compatible. They also have many common peripherals and other features. (Note: This paper is sponsored by Silicon Labs, but all opinions and analysis are those of the author.)

Silicon Labs: Background

Most engineers know Silicon Labs for its analog and mixed-signal technology. The company has core expertise in integrating high-performance analog peripherals with digital circuits in CMOS. This expertise is embodied in a number of IP blocks such as analog-to-digital converters (ADCs), digital-to-analog converters (DACs), digital isolation, digital phase-locked loops, USB, RF, and sensors.

Less well known is that Silicon Labs has a long history in the MCU market. This experience dates back to 1999, when a company called Cygnal Integrated Products began work on a line of mixed-signal MCUs. By the time Silicon Labs acquired Cygnal in 2003, the company had released more than 50 versions of its eight-bit MCU. Then, in 2013, Silicon Labs acquired Energy Micro, a Norwegian MCU vendor founded in 2007. Energy Micro's ultra-low-power EFM32 Gecko MCUs merged with Silicon Labs' MCU product line, and the combined company has focused even more intensely on energy efficiency.

Any product portfolio with more than 240 devices is bound to seem confusing. As Table 1 shows, however, Silicon Labs has organized this portfolio into several product families of pin- and software-compatible chips. The EFM32 Zero Gecko family includes the lowest-power MCUs based on ARM's Cortex-M0+. The EFM32 Wonder Gecko family includes higher-end MCUs based on ARM's Cortex-M4. Between them are four additional families of EFM32 Gecko MCUs based on ARM's Cortex-M3 (Tiny Gecko, Gecko, Leopard Gecko, and Giant Gecko). All of Silicon Labs' ARM-based 32-bit MCUs are upwardly software compatible.

This paper focuses primarily on the Cortex-M3 EFM32 MCUs, which offer the most popular combinations of processing power, features, low power consumption, and low cost. These MCUs operate at CPU clock frequencies of up to 48MHz and have 2KB to 128KB of SRAM and 4KB to 1MB of flash memory. When active, a typical EFM32 Gecko MCU on a 3V power supply draws only 150 microamps per megahertz. In ultra-low standby mode, the same MCU draws a mere 0.9 microamps. Both numbers are below those of most other 32-bit MCUs and even beat some leading 16-bit MCUs.

	Zero Gecko	Tiny Gecko	Gecko	Leopard Gecko	Giant Gecko	Wonder Gecko
CPU Core	Cortex-M0+	Cortex-M3	Cortex-M3	Cortex-M3	Cortex-M3	Cortex-M4
CPU Freq (max)	24MHz	32MHz	32MHz	48MHz	48MHz	48MHz
SRAM	2–4KB	2–4KB	8–16KB	32KB	128KB	32KB
Flash Memory	4–32KB	4–32KB	16–128KB	64–256KB	512KB–1MB	64KB–256KB
Low-Energy Sensor Interface	No	Yes	No	Yes	Yes	Yes
Advanced Features	Peripheral Reflex System	Peripheral Reflex System, LCD	Peripheral Reflex System, LCD	Peripheral Reflex System, LCD, TFT, USB2.0	Peripheral Reflex System, LCD, TFT, USB2.0	Peripheral Reflex System, LCD, TFT, USB2.0, DSP+FPU
Variety	4 chips	11 chips	11 chips	20 chips	20 chips	20 chips
Packages	QFN-24, QFN-32, QFP-48	QFN-24, QFN-32, QFN-64, QFP-48, QFP-64, BGA-48	QFN-32, QFN-64, QFP-48, QFP-64, QFP-100, BGA-112, full wafer*	QFN-64, QFP-64, CSP-81, QFP-100, BGA-112, BGA-120, full wafer*	QFN-64, QFP-64, QFP-100, BGA-112, BGA-120, full wafer*	QFN-64, QFP-64, CSP-81, QFP-100, BGA-112, BGA-120, full wafer*

Table 1. Silicon Labs' EFM32 microcontroller product portfolio. *Uncut wafers are available for high-volume customers that do their own packaging; examples are chip-on-board designs and multichip modules (MCMs). (Source: Silicon Labs)

The Importance of Low Energy Consumption

All MCUs are low-power self-contained devices. With their CPU cores, on-chip SRAM, nonvolatile flash memory, and integrated peripherals, they can replace a discrete microprocessor that requires external DRAM, external flash, and peripheral chips. Thus, they save board space and reduce the bill of materials (BOM) while lowering system power consumption. However, "low power" is a broad statement that does not adequately describe the significant differences among the thousands of MCUs on the market. Silicon Labs' EFM32 Gecko MCUs reduce power consumption in both their active and idle states, which is particularly important for battery-powered systems.

Low active power enables the MCU to conserve energy when running at any CPU frequency within its range, even while exercising on-chip memory and integrated peripherals. After completing an essential task, the MCU can enter a sleep mode that dramatically reduces energy consumption while remaining alert to external events that will trigger another wakeup. A less-capable processor may draw less current while operating but will use more energy overall when it awakens and labors to finish a complex task. (If halving the power consumption means running the processor in a higher power state for twice as long, the net energy saving is zero—or even less than zero, if other system functions required to support the CPU are also active in the higher power state.) Thus, the key to conserving total system power is to put the MCU to sleep for as long as possible, quickly awaken it to perform its necessary duties as rapidly as possible, and then put it to sleep again. Figure 2 illustrates this runtime profile. In some applications, this profile enables a Gecko MCU to subsist on a coin-size battery for ten years or more. Longevity is crucial for embedded systems installed in places that make battery replacements difficult, impossible, or prohibitively expensive.

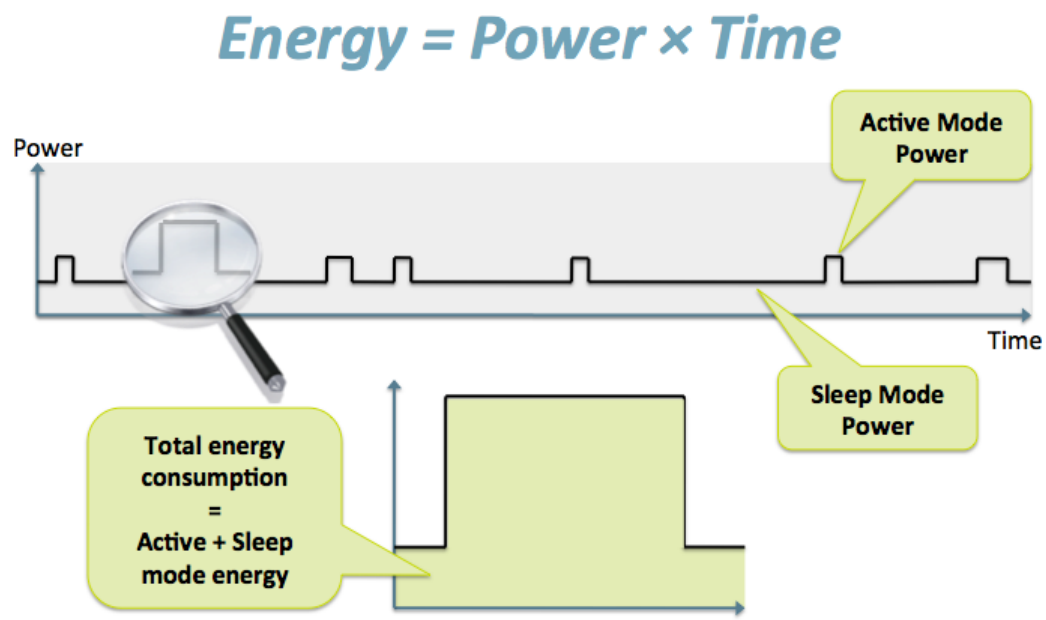

Figure 1. Runtime power profile in active and sleep modes.

Low power consumption also improves system reliability by reducing heat dissipation. Components last longer when their operating temperatures remain cool and steady, and passively cooled systems last longer and run quieter than systems needing exhaust fans.

Gecko's power/performance profile is well suited to Internet of Things (IoT) applications. A smart thermostat, for example, spends most of its time in sleep mode, waiting for a temperature variation that activates the heating or air-conditioning equipment. The same is true for many other intelligent connected devices: smoke alarms, security systems, programmable lighting controllers, health monitors, and so forth. Some of these devices can be mains powered, making energy consumption less important. However, others must run on a tiny battery or at least have a battery backup. Still others may subsist on a small photovoltaic cell or a parasitic power source. For these types of applications, a low-energy MCU is not just a virtue but also a requirement.

Ten Techniques for Saving Energy

Almost all 32-bit MCUs have ARM Cortex-M cores, which would seem to leave little room for differentiation among MCU vendors. In addition, all vendors must obey the same laws of physics, which dictate that power consumption varies linearly with CPU clock frequency and quadratically with voltage. In practice, however, vendors can reduce power consumption in various ways: by using better clock trees, automated power gating, more-efficient logic synthesis, different operating modes, and so on. Also, the other digital and mixed-signal logic consumes significant power and therefore offers additional opportunities for improving efficiency. The EFM32 Gecko family employs ten power-saving techniques:

- Optimized circuit design and logic synthesis keep active power as low as possible. As noted above, a typical EFM32 MCU draws only 150 microamps per megahertz when active.
- Performance-enhancing features enable EFM32 MCUs to rapidly complete their tasks when active so they can return to their sleep state as soon as possible. The Cortex-M3 core employed in most EFM32 MCUs achieves 3.32 CoreMarks per megahertz on EEMBC's popular benchmark test—only 2% less than the more powerful Cortex-M4.
- EFM32 MCUs can enter low-power sleep modes faster than most MCUs and can awaken in as little as two microseconds. One reason for this performance is that two of the frequently used sleep modes preserve the register contents and on-chip SRAM. As a result, EFM32 MCUs waste very little energy and time in transitional states and respond quickly to external events.
- In standby, EFM32 MCUs draw very little current. Shutdown mode draws as little as 20 nanoamps at 3V; deep-sleep mode draws only 900 nanoamps at the same voltage while maintaining the SRAM, flash memory, and several peripherals.
- All EFM32 peripherals are autonomous, so they can continue operating while the CPU sleeps. For example, the low-energy UART can autonomously filter input and accept only the input containing a certain address. The DMA controller can then autonomously store the accepted input in SRAM and awaken the CPU only after receiving a whole package.
- A user-configurable on-chip network (the Peripheral Reflex System) enables the peripherals to communicate directly with each other without involving or even waking the CPU or DMA controller.
- Five distinct energy modes enable flexibility when conserving power and balancing performance.
- EFM32 MCUs have optimized peripherals that benefit from Silicon Labs' experience with digital and mixed-signal design. For example, the 12-bit ADC can measure one million samples per second while drawing only 350 microamps.
- Even while sleeping, EFM32 MCUs can monitor numerous external sensors. All EFM32 MCUs (except the Zero Gecko and original Gecko families) have a low-energy sensor interface called LESENSE that can simultaneously monitor up to 16 sensors while the CPU slumbers in deep-sleep mode (the third-lowest power state). Other peripherals can monitor external sensors as well. The low-energy sensor interfaces are user configurable and can monitor capacitive, resistive, and inductive inputs.
- Silicon Labs' EnergyAware Profiler is an important development tool included in the company's Simplicity Studio development platform. It gives programmers real-time feedback on current consumption and correlates it with the code being executed, enabling developers to profile and optimize their application.

Some of these techniques deserve closer examination. In particular, Silicon Labs has optimized the various power states and on-chip peripherals in ways that set EFM32 MCUs apart from competing products.

EFM32 Energy Modes

Figure 2 is a high-level block diagram of an EFM32 Gecko MCU with the various function blocks colored to show their operational status in the five energy modes. (Note: this status varies slightly among different Gecko-family MCUs; this figure represents Leopard Gecko and Giant Gecko.) The default energy mode is EM0, in which all function blocks are awake and fully operational. In this mode, the ARM Cortex-M3 is actively fetching and executing instructions from on-chip SRAM or flash memory. As noted, a Gecko MCU typically draws only 150 microamps per megahertz in EM0.

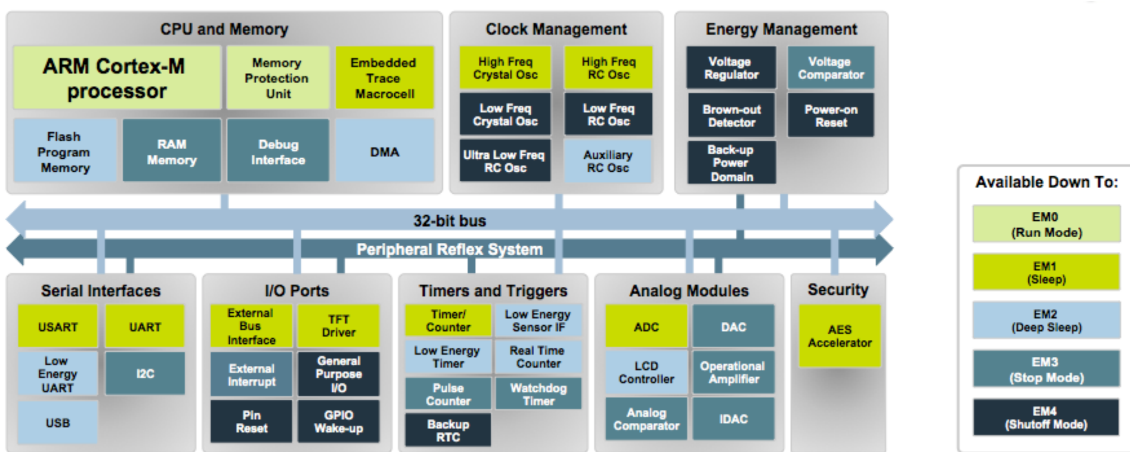

Figure 2. EFM32 Gecko block diagram and the five energy modes.

Even so, the less time spent in EM0, the better. Ideally, a well-designed system will enter this mode just long enough to perform any duties requiring the CPU's processing muscle and then drop down to a lower-power mode. In EM1 mode, wakeup is almost immediate—just a few clock cycles. In the EM2 and EM3 modes, any interrupt will return the chip to EM0 in about two microseconds. After completing the appropriate interrupt service routine, the CPU will resume executing code at the point where it went to sleep. EM4 (stop mode) is different—waking up requires a reset and takes a little longer, depending on the event triggering the reset.

EM1 is sleep mode. To enter this mode, the control program executes either a Wait For Interrupt (WFI) or Wait For Event (WFE) instruction. Sleep mode stops the CPU clock but keeps all peripherals, on-chip SRAM, and flash memory available. For example, a timer can tell the ADC to sample an analog signal, convert it to a digital bitstream, and store the data in memory—all without awakening the CPU. The Peripheral Reflex System (PRS) and DMA enable these capabilities. A typical application would be high-speed data acquisition in which the ADC continues to run at full speed and uses DMA to log data to memory. Eventually, of course, the CPU must process the converted data for some purpose. The control program can wait until the ADC converts a user-specified amount of data, then generate an interrupt to put the CPU into EM0 mode. Once awake, the CPU can quickly process the data, act on it, and go back to sleep.

EM2 is deep-sleep mode. To enter this mode, a program first sets the SLEEPDEEP bit in the system control register and then executes the same WFI or WFE instruction that puts the chip into EM1 sleep mode. EM2 deep sleep shuts off the CPU and high-frequency oscillators, leaving only the asynchronous or low-frequency peripherals available. Those operational blocks include the SRAM and flash memory and the DMA, LCD, and USB2.0 controllers. Although the fast UART and USART controllers go to sleep in this mode, a lower-energy 32KHz UART remains available and draws only 150 nanoamps at 9.6Kbps. Likewise, the regular 16-bit timer goes to sleep, but a low-energy 16-bit timer, a 24-bit real-time counter, and a 32-bit backup counter remain awake. Ditto for the low-energy sensor interface. The main function blocks that are *not* operational in EM2 are the ADC, the AES cryptography accelerator, the 480x320-pixel TFT/OLED driver, and the 16-bit external bus interface. Note, however, that the DAC and analog comparator remain available in EM2 mode even though the ADC is asleep. EM2 deep-sleep mode allows an EFM32 MCU to draw even less current than it does in EM1 sleep mode while still driving an LCD screen, monitoring a sensor input, moving data through the low-energy UART, and performing other critical functions. For example, a smoke alarm can periodically sample the sensor, process it, and go back to sleep.

EM3 is stop mode, but not everything stops. To enter this mode, the program first sets the SLEEPDEEP bit in the system control register and then executes the WFI or WFE instruction. If the low-frequency crystal or RC oscillator was off when the program issued the instruction, the chip enters EM3 stop mode. In other words, the difference between EM2 and EM3 is whether the low-frequency oscillators are available. When the oscillators stop, the low-energy timer, low-energy sensor interface, and real-time counter are inactive. EM3 also disables flash memory (but not SRAM) and the DMA, LCD, USB, and low-energy UART controllers. An important exception, however, is that an ultra-low-frequency oscillator keeps the watchdog timer running. Other available function blocks include the pulse counter, voltage comparator, analog comparator, DAC, I^2C controller, GPIO (including GPIO interrupts), and the Peripheral Reflex System. These exceptions are important because they enable some critical operations to continue. For instance, the I^2C controller can check inputs for a matching memory address; the voltage comparator can monitor an external source and trigger an interrupt if the input voltage reaches a defined threshold; and the analog comparator can wake up the CPU if an external signal exceeds a reference value. These capabilities are useful for industrial control applications.

EM4 is shutoff mode, which draws as little as 20 nanoamps. To avert any chance of entering this mode by accident, a program must consecutively write a series of values into a control register in the energy management unit. Only a few blocks continue to function in EM4, including the reset logic and (optionally) the voltage regulator and brownout detector (BOD). If the regulator and BOD remain available, the chip draws about 100 nanoamps. Only a reset can exit EM4 mode. Usually, an external event triggers the reset, but timed wakeups are possible by preconfiguring the backup real-time counter (a 32-bit counter) and some other function blocks (such as the voltage regulator). When the regulator and BOD are disabled, an EFM32 MCU in EM4 mode draws less current than even the self-discharge rate of a typical coin-size battery. The MCU can remain on standby for years, then spring back to life, resume fetching instructions from internal flash memory, and continue processing as if nothing had happened. For example, a remote control can sleep until the user wakes it up by pressing a button.

Peripheral Reflex System Operates Autonomously

Although all MCUs (and most other types of microprocessors) also have multiple low-power operating modes, no other 32-bit MCUs can draw as little current as EFM32 Gecko MCUs do while remaining capable of responding to such a wide variety of events. One key to this capability is the Peripheral Reflex System, which enables the on-chip peripherals and other function blocks to autonomously receive inputs from the outside world, interact with other function blocks, and access on-chip memory through the DMA controller—all while the CPU sleeps.

The Peripheral Reflex System is an on-chip network that connects reflex producers to reflex consumers. As the names imply, the producers generate signals to which the consumers respond. Reflex producers include the ADC, DAC, timers, counters, low-energy sensor interface, UARTs, USB port, GPIO pins, analog comparator, and voltage comparator. Reflex consumers include the ADC, DAC, timers, low-energy sensor interface, UARTs, and pulse counter. Note that some function blocks (such as the ADC and DAC) are both producers and consumers.

Figure 3 shows how the Peripheral Reflex System enables an EFM32 MCU to monitor external events without involving the Cortex-M3 CPU. As an example, the analog comparator and a timer can be used to monitor a signal to detect if it exceeds a predefined threshold for more than a predefined duration. If the signal is above the threshold, the comparator can send a high signal through the Peripheral Reflex System to one of the timers, which measures the condition's duration by resetting and starting its count on the positive edge of the reflex signal and stopping on the negative edge. If the count exceeds the predefined comparator value, the timer can trigger an interrupt that wakes the CPU. Otherwise, the CPU can remain asleep. This system is much more power efficient than using the CPU to monitor the comparator's output directly.

Numerous other examples are possible by connecting the reflex producers to the reflex consumers in different ways. The Peripheral Reflex System is truly a network that enables direct connections between any producer and consumer. These interactions can continue even while portions of the MCU sleep in the EM1, EM2, or EM3 modes (depending on which peripherals are available in a particular mode). In addition, the DMA controller provides direct access to memory in EM1 (sleep) and EM2 (deep sleep). Without bothering the CPU, it can move data between SRAM, flash memory, peripherals, and the external bus interface. This 12-channel controller also supports scatter-gather operations from multiple sources (vectored I/O) and double-buffered ping-pong operations (page flipping).

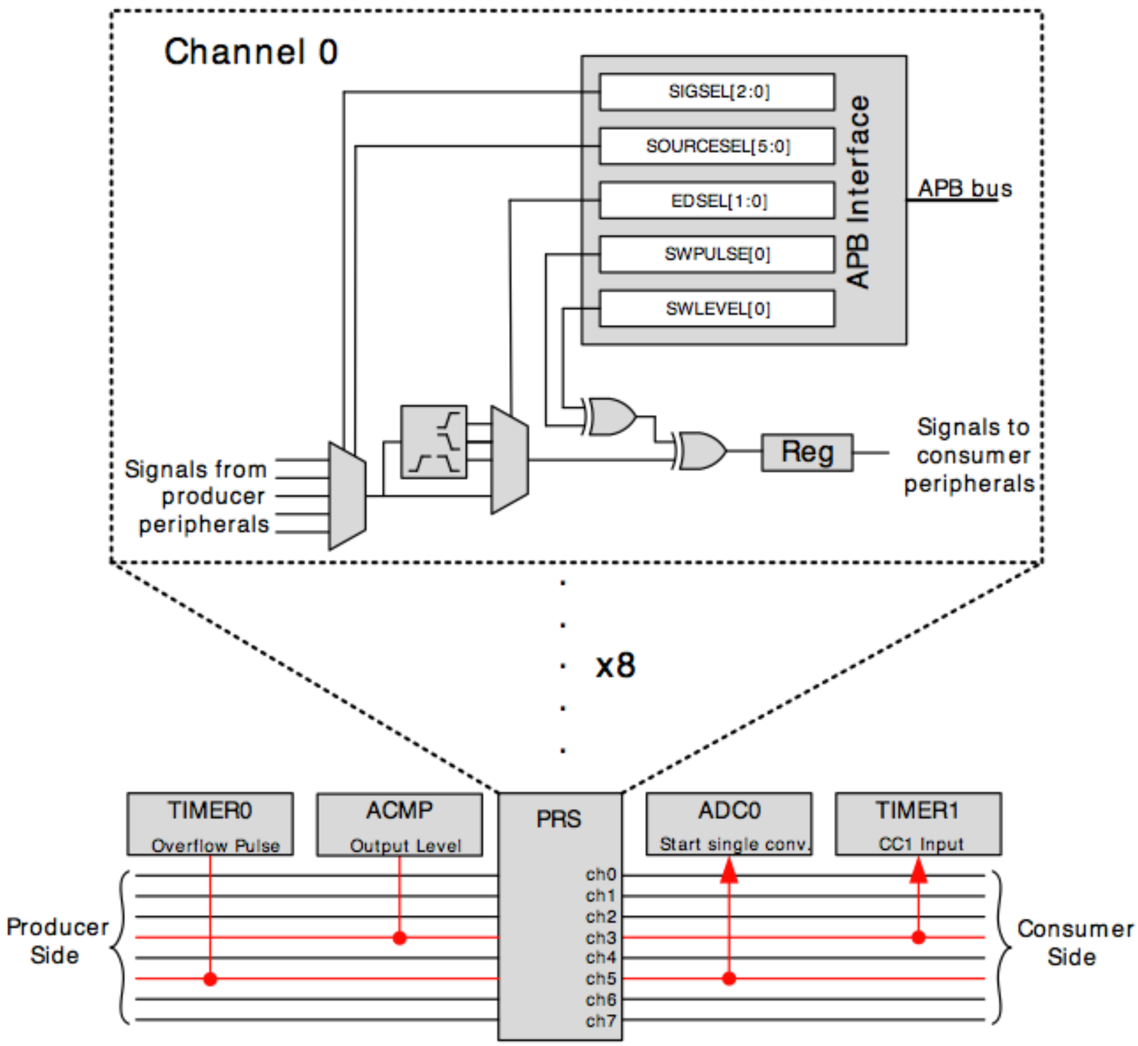

Figure 3. Processing external signals through the Peripheral Reflex System.

Low-Energy Peripherals Complete the Picture

Bypassing the CPU with the Peripheral Reflex System saves significant power but would be much less effective without low-energy peripherals. Silicon Labs brings its extensive analog-design experience to this challenge. Consequently, EFM32 MCUs have some of the industry's lowest-energy integrated peripherals. (Note: the peripheral mix varies among the Tiny Gecko, Gecko, Leopard Gecko, and Giant Gecko MCUs.)

The ADC and DAC are good examples because all Cortex-M3 Gecko family MCUs have these peripherals and also because they often see heavy use in embedded applications. Silicon Labs' 8-channel ADC can sample inputs at 6-, 8-, or 12-bit resolution. An integrated temperature sensor and hardware-controlled warmup period ensure accurate results. At its highest resolution, it can convert one million samples per second while drawing only 350 microamps. The dual-channel DAC also operates at 12-bit resolution and can convert 500,000 samples per second while drawing only 200 microamps.

Another frequently used function block is the low-energy sensor interface (available in Tiny Gecko, Leopard Gecko, and Giant Gecko). It can measure up to 16 sensors for resistive and capacitive inputs at currents down to 1.3 microamps or inductive inputs at currents down to 1.5 microamps. The sensor interface operates autonomously and remains active in sleep (EM1) and deep-sleep (EM2) modes. When it detects a sensor event, it can alert another peripheral device (such as the ADC) on the Peripheral Reflex System or trigger an interrupt that awakens the CPU.

The low-energy sensor interface also has a built-in state machine for preprocessing sensor results. It can automatically decode quadrature rotation data based on inductive-capacitive (L-C) sensors. The state machine can feed this data to the pulse counter, which can keep track of the rotation state autonomously. The CPU will awaken only under certain conditions, such as a direction change or after a certain number of revolutions.

As Figure 4 shows, autonomous operation is more power efficient than waking up periodically to check for sensor events. Another feature, conditional wakeup, enables higher-level reasoning about external events. Whereas an L-C sensor measurement is merely a first-order result, multiple measurements can reveal rotational movement and direction, and accumulated movements can reveal a position. External events can trigger wakeups whenever the MCU measures a predefined value (every time), or on direction changes (every *n* number of times), or even on accumulated movements. Alternatively, the low-energy UART can wake the system on every byte received or only when the system needs to wake up to handle an event.

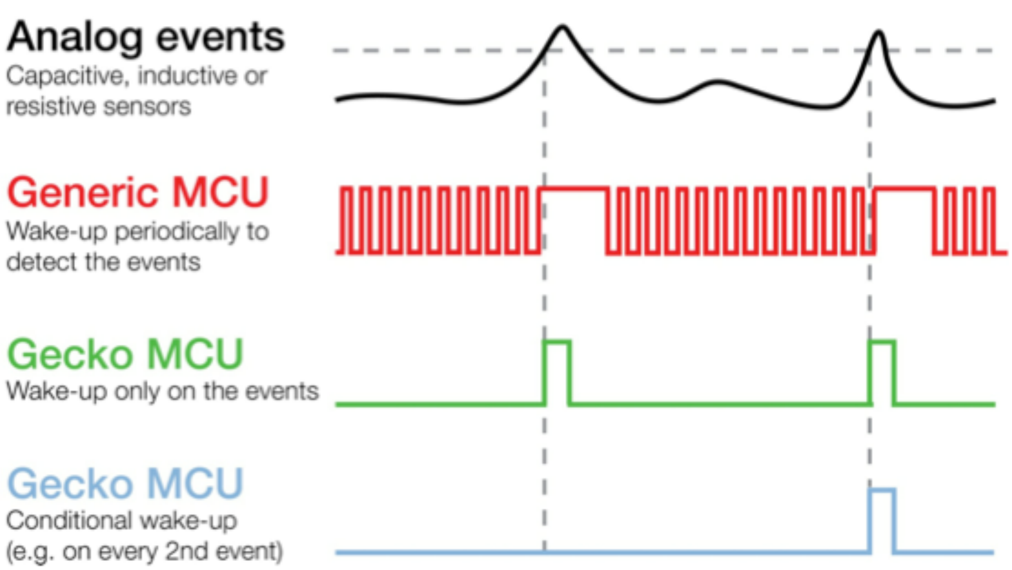

Figure 4. Silicon Labs' low-energy sensor interface spends more time sleeping to conserve energy.

Silicon Labs has brought similar design expertise to other Gecko peripherals, including the analog comparator, voltage comparator, operational amplifiers, TFT/OLED driver, and LCD controller. For instance, EFM32 Gecko MCUs can drive a 4x40-segment LCD while drawing only 550 nanoamps. They can drive a 1x40-segment LCD while drawing only 250 nanoamps. The LCD controller remains active down to EM2 deep-sleep mode, enabling a system to display its operational status even while the CPU and many peripherals are inactive.

Energy-Aware Development Tools

All modern MCUs have integrated development environments (IDEs) that ease programming. Whereas most IDEs readily offer feedback that helps optimize code size, run-time performance, and correctness, Silicon Labs' Simplicity Studio also displays profiles of system power consumption. The EnergyAware Designer and EnergyAware Profiler tools enable hardware and software engineers to rapidly find and fix the power-sapping "hot spots" as well as the performance-sapping code segments. Graphical displays provide detailed views of object-code execution and power consumption as the program runs. In addition, the Leopard Gecko and Giant Gecko MCUs include an embedded trace macrocell for enhanced real-time tracing and debugging.

EFM32 Gecko Finds a Market Niche

The world is awash in 32-bit MCUs from several vendors, including Atmel, Freescale, Microchip, NXP, STMicroelectronics, and Texas Instruments. Those companies are larger than Silicon Labs and offer hundreds of products that capture larger shares of the MCU market. Most of their MCUs are based on the same ARM Cortex-M0+, Cortex-M3, and Cortex-M4 cores that Silicon Labs uses in its EFM32 MCUs. Finding a way to stand out in this crowded market isn't easy, but the EFM32 family offers a compelling combination of performance, power consumption, features, and autonomous peripherals.

Table 2 compares a Tiny Gecko (EFM32-TG840F32) with MCUs from three of Silicon Labs' many competitors: Atmel, Freescale, and STMicro. An almost infinite number of such comparisons are possible; this one strikes a balance between performance and power consumption. Such comparisons don't necessarily result in a uniform selection of CPU cores. For instance, Atmel's low-power SAM4LC2C products use Cortex-M4, which is larger and slightly more powerful than Tiny Gecko's Cortex-M3, but they are nevertheless a close match on power consumption. Freescale offers no Cortex-M3 MCUs at all; its Kinetis-L KL02 device with Cortex-M0+ is the low-power substitute. STMicro's STM32L-151 uses Cortex-M3, so it's a closer match. This table scales all the clock speeds to match Tiny Gecko's performance at 25MHz (83 CoreMarks).

Atmel emerges as the closest competitor in this group. Despite the SAM4LC2C's larger CPU core, it beats Tiny Gecko's active power. And, like Tiny Gecko, the SAM4LC2C can operate some of its peripherals while the CPU sleeps. This capability (Atmel calls it "SleepWalking") is not as extensive as Tiny Gecko's, however. To preserve the SRAM contents and keep some peripherals running, Atmel's wait mode draws more than three times as much current as Tiny Gecko's EM2 deep-sleep mode—a comparable power state. The SAM4LC2C does support a lower energy mode that retains the logic state, but it disables the peripherals. By contrast, Tiny Gecko's next-lower energy mode (EM3) not only retains the logic state but also the SRAM contents and several peripherals, such as the DAC and analog comparator.

	Silicon Labs Tiny Gecko EFM32-TG840F32	Atmel SAM4LC2C	Freescale Kinetis-L KL02	STMicro STM32L-151
CPU Core	Cortex-M3	Cortex-M4	Cortex-M0+	Cortex-M3
CPU Frequency	25MHz	24.4MHz	34.3MHz	25MHz
CoreMarks/MHz	3.32CM/MHz	3.4CM/MHz	2.42CM/MHz	3.32CM/MHz
CoreMarks/mA	22.4CM/mA	38.6CM/mA	24.2CM/mA	15.6CM/mA
Active Power	3.7mA	2.2mA	3.4mA	5.3mA
Standby Power	0.9μA*	3.0μA†	86μA‡	1.2μA§
Wakeup Time	2.0μsec	1.5μsec	4.0μsec	8.0μsec
Autonomous Peripherals	Several	Some	Some	Some
Autonomous Peripheral Network	Yes	Yes	No	No

Table 2. Silicon Labs' Tiny Gecko and three similar microcontrollers. Clock frequencies and active power are scaled to match Tiny Gecko's performance of 83 CoreMarks at 25MHz. *EM2 deep-sleep mode while operating the RTC, brownout detector, 32KB SRAM, flash memory, and some peripherals. †Wait mode with memory retention and some peripherals. ‡Very-low-power wait mode without memory retention or peripherals. §Stop mode with RTC and memory retention but no peripherals. (Source: vendors)

Freescale's Kinetis-L KL02 uses the simpler Cortex-M0+ core, which consumes less power but delivers less performance than Cortex-M3 and Cortex-M4. To match Tiny Gecko's performance of 83 CoreMarks at 25MHz, the KL02 must run at 34.3MHz. (Its maximum speed is 48MHz.) Although it consumes a trifle less active power than Tiny Gecko at this scaled clock speed, in standby mode it draws nearly two orders of magnitude more power—86μA versus 0.9μA. And that rating is for Freescale's very-low-power wait mode, which sacrifices memory retention and stops all the peripherals. Another consideration is that existing ARM code will need recompiling, because Cortex-M0+ supports only subsets of Thumb and Thumb-2 instructions, not the full ARMv7-M instruction set.

STMicro's STM32L-151 consumes only a little more power than Tiny Gecko in both active and standby modes. Its attractive standby power comes at a price, however—it must enter a stop mode that retains memory and keeps the real-time clock running but disables the peripherals. Although the STM32L-151 supports some lower-power modes, they shut down the SRAM in addition to the peripherals, unlike Tiny Gecko's EM2 and EM3 modes. In addition, the STM32L-151 takes longer to wake up than the other MCUs in this group.

Another factor in Tiny Gecko's favor is its low-energy brownout detector, which stays available all the way down to EM4 shutoff mode. Brownouts happen when the voltage from the system's power supply (often a battery) dips just below the threshold required for correct operation but remains high enough for the system to continue operating. During a brownout, the system's behavior will be uncertain, because the behavior of the

CPU and other peripherals is not guaranteed. For example, the CPU may set an I/O pin high when it should be low, potentially causing a fatal error in a critical application. The other MCUs in this comparison maintain their brownout detectors in the standby modes listed in Table 2 but not necessarily in the lower-energy states that correspond to Tiny Gecko's EM3 mode. Tiny Gecko's brownout detector avoids such calamities by enabling the MCU to recover or reset itself.

All the MCUs in this comparison have similar integrated peripherals. There are some variations, of course, and different versions of these chips have slightly different features, so a system requirement for a specific peripheral or feature will influence the designer's choice more than small differences in power consumption or CPU performance. Overall, however, Tiny Gecko offers the best combination of CPU performance, active power, standby power, wakeup time, and integrated peripherals capable of operating autonomously in low-energy modes.

Gecko is Well Suited to Modern Systems

In summary, Gecko MCUs are a good choice for small embedded systems that spend most of their time sleeping and then awaken for brief bursts of activity. Their lowest-energy standby modes sip trivial amounts of power that can sustain the chips for years on coin-size batteries, photovoltaic cells, or parasitic power sources. Their integrated peripherals can handle many external events without CPU intervention, conserving even more power. When an event does require processing, the CPU awakens quickly and delivers strong performance while active. When the task is finished, the CPU and other function blocks immediately resume their slumber.

This power/performance profile is typical of many modern embedded systems and is particularly prevalent in IoT applications. Silicon Labs' competitors can match some of these characteristics but rarely all of them in a single device. In addition, the EFM32 family's wide variety of MCUs, peripherals, I/O interfaces, packages, and other features gives system designers a wealth of choices. In a market crowded with MCUs from larger vendors, Silicon Labs' EFM32 Gecko family merits a close look.

Tom R. Halfhill is a senior analyst at The Linley Group and a senior editor of Microprocessor Report. *The Linley Group offers the most comprehensive analysis of the microprocessor industry. We analyze not only the business strategy but also the internal technology. Our in-depth reports cover topics including embedded processors, mobile processors, network processors, base-station processors, and Ethernet chips. For more information, see our web site at www.linleygroup.com.*