



## **ARM® Cortex®-M3 & M4 MCU Architecture**

### **Introduction**

Many embedded developers are familiar with the ARM Cortex processor architecture, but few have the opportunity to become intimately acquainted enough of this popular architecture to take full advantage of its unique features and capabilities. This is especially true for the new ARM Cortex-M4 processor, which boasts an improved architecture, native digital signal processing (DSP) capabilities and an optional floating-point accelerator, which a savvy programmer or hardware engineer can exploit to their advantage. Let's take a closer look at some of the more interesting (and often-overlooked) features found in Cortex-M3 based microcontrollers (MCUs) as well as new M4 variants.

Since many target applications for Cortex-M based MCUs are portable and derive their power from batteries or energy harvesting systems, most of the ideas we will explore involve techniques for reducing a design's overall energy consumption. In many cases, however, these energy conservation techniques are also helpful tools for designing processor-optimized applications that provide:

- More cost-effective solutions
- More processing margin available for upgrades and new features
- Performance and features that help products stand out in crowded markets.

### **ARM Cortex Basics**

Much like the original 16-bit processor cores created by Advanced RISC Machines (ARM) in the 1980s, the ARM Cortex series is based on a Harvard-style RISC machine with a modest silicon footprint that enables high performance as well as code and memory efficiency. The architecture has evolved considerably over the past decade, branching into three distinct sub-families (or profiles) created to meet the requirements of a particular application space:

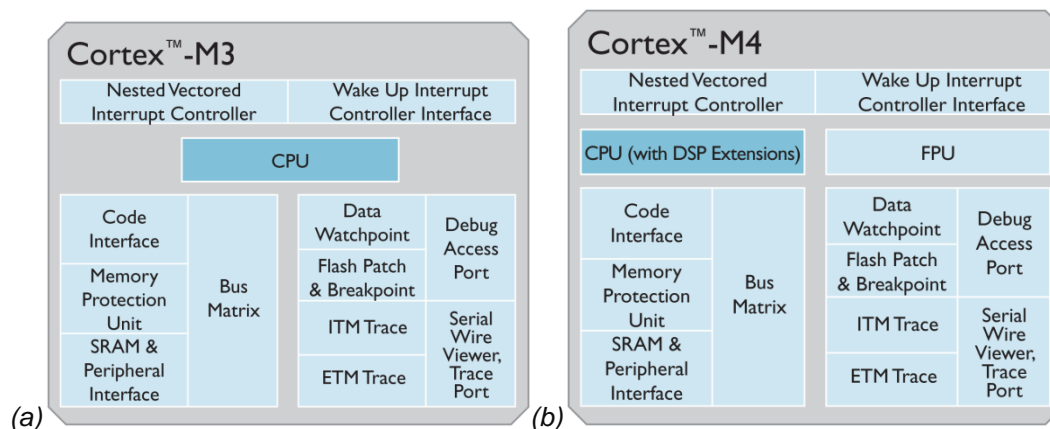
- A-profile products are optimized for high-performance open application platforms
- R-profile processors include features for enhanced performance and reliability in real-time applications
- The M-profile processor series was developed for use in deeply-embedded MCUs in applications where performance must be balanced with energy efficiency and low solution cost. Popular applications for the Cortex M series include smart metering, human interface devices, automotive and industrial control systems, white goods, consumer electronics products and medical instrumentation.

## The Cortex-M3 vs. Cortex-M4 Story

The idea behind the Cortex-M3 architecture was to design a processor for cost-sensitive applications while providing high-performance computing and control<sup>1</sup>. These applications include automotive body systems, industrial control systems and wireless networking/sensor products. The M3 series introduced several important features to the 32-bit ARM processor architecture including:

- Non-maskable interrupts
- Highly-deterministic, nested, vectored interrupts
- Atomic bit manipulation
- Optional memory protection (MPU)

In addition to excellent computational performance, the Cortex-M3 processor's advanced interrupt structure ensures prompt system response to real-world events while still offering low dynamic and static power consumption<sup>2</sup>.



**Figure 1. Comparison of the Cortex-M3 and M4 Processor Cores.**

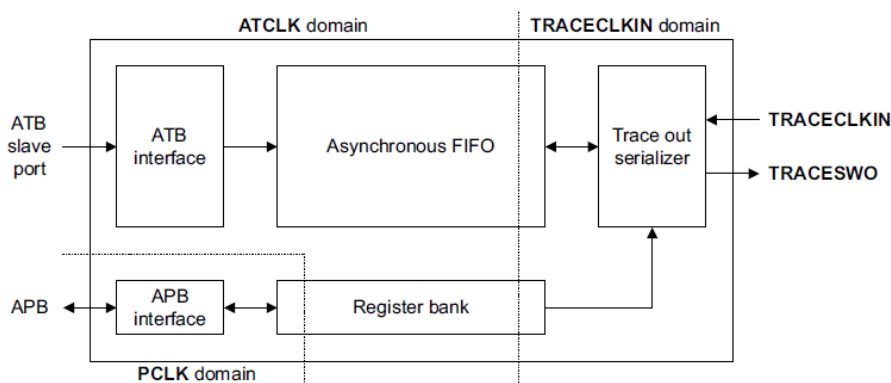
The Cortex-M3 and M4 processors share many common elements including advanced on-chip debug features and the ability to execute the full ARM instruction set or the subset used in THUMB2 processors. The Cortex-M4 processor's instruction set is enhanced by a rich library of efficient DSP features including extended single-cycle cycle 16/32-bit multiply-accumulate (MAC), dual 16-bit MAC instructions, optimized 8/16-bit SIMD arithmetic and saturating arithmetic instructions. Overall, the most noticeable difference between M3 and M4 is the optional single-precision (IEEE-754) Floating Point Unit (FPU) available with the M4.

## Serial Secrets Stimulate Slick Solutions

The success or failure of an embedded design often rests on finding the right balance between system performance, energy consumption and solution cost. In many cases, developers can use the Cortex-M processor's unique features to optimize for product cost or energy appetite while maintaining, or even improving, its performance. For example, the Cortex-M core has native serial I/O capabilities that can be

used to save energy, simplify development and free up peripherals to be used for other application tasks.

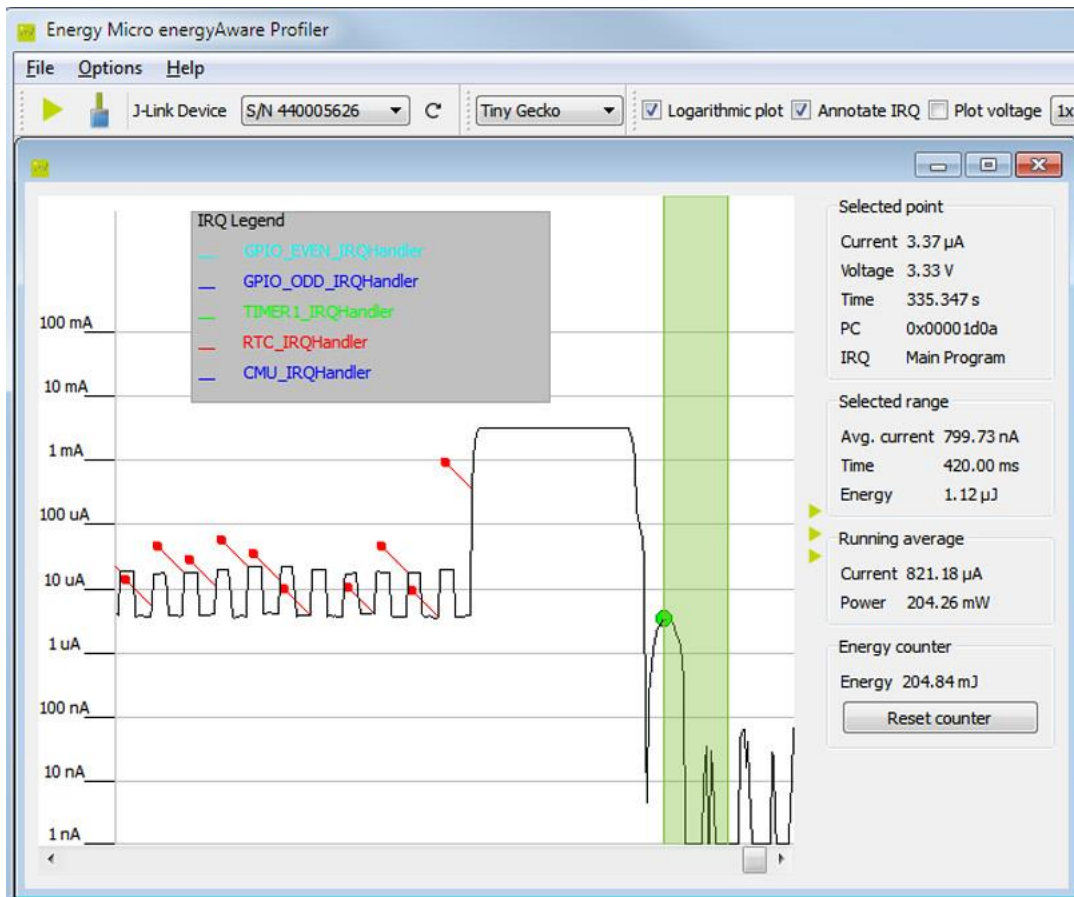
Besides the traditional Serial Wire Debug functions, ARM Cortex-M based microcontrollers also offer an instrumentation trace interface through its single-pin Serial Wire Viewer Output (SWO)<sup>3</sup>, as shown in Figure 2. This port can be used to pass “printf-format” debug messages directly to application code. SWO allows the debug messages to be viewed directly from any standard IDE. Additionally, these messages can be viewed through a standalone SWO viewer such as Segger's J-Link SWO Viewer software<sup>4</sup> or the energyAware Commander from Silicon Labs<sup>4</sup>. Since the SWO output is built into the core HW itself, this is an inherent benefit of the Cortex-M core. SWO doesn't waste any of the MCU's regular UARTs, which might already be committed to the application.



**Figure 2. The Dedicated ARM Cortex SWO Interface Saves I/O Pins and Speeds Up Debugging.**

Another important advantage of SWO-based debugging is that it allows the MCU to maintain an active debug connection when it enters its lowest sleep modes where, in most cases, the logic for traditional debug connections is inoperative. The instrumentation trace of the SWO can also be used for sampling the program counter to help IDEs create statistics on how much time is spent in each of the program functions. These statistics can be combined with current measurements to help fine-tune a design's energy consumption.

Cortex-M based MCU vendors are beginning to recognize this benefit, and some manufacturers have already incorporated power profile and current measurement hardware into their development platforms for this purpose. For example, all starter and development kits for the EFM32 Gecko MCUs from Silicon Labs include live power measurement outputs, which can be coupled with the program trace in the energyAware Profiler tool<sup>6</sup>. Figure 3 shows how this allows the designer to pinpoint which program functions are the highest energy drains and allows fast debugging of other energy-related problems.



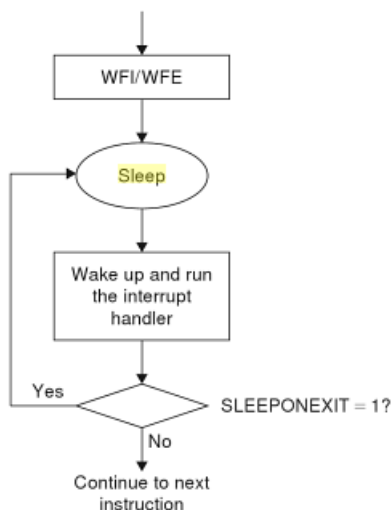
**Figure 3. Software and hardware tools that pinpoint which functions are using the most current eliminate the need oscilloscopes and multimeters and enable fast debugging.**

## Sleep Smart and Make Every $\mu\text{W}$ Count

The ARM Cortex-M processor’s Sleep-on-Exit instruction is another “twofer” feature that can save both CPU cycles and energy. This is especially useful in interrupt-driven applications where the processor spends most of its time either running interrupt handlers or sleeping between interrupt events. When entering an interrupt service routine (ISR), the MCU must spend several instruction cycles pushing the present thread’s state onto the stack and then “popping” it upon return. In applications where the processor returns directly to its sleep after an ISR, a conventional MCU must still recover its stored state information before the thread code can put the device to sleep. Likewise, its state must be pushed to the stack again when the next interrupt wakes the device.

When an ARM Cortex-M based microcontroller’s Sleep-on-Exit is enabled, the device will enter sleep directly after the ISR finishes without returning to the thread (see Figure 4). This preserves the processor in the interrupt state, saving the precious machine cycles normally required to push the normal state

onto the stack during wake-up. Eliminating the stack push and pop cycles saves both the time and energy otherwise consumed by unneeded instruction cycles, as well as any code a conventional MCU would need to manage the stack between its sleep and wake states. And, should the processor be awakened by a halt debug request, the unstacking process will be carried out automatically.



Bits	Name	Type	Reset Value	Description
4	SEVONPEND	R/W	0	Send Event <b>on</b> Pending; wakes up from WFE if a new interrupt is pended, regardless of whether the interrupt has priority higher than the current level
3	Reserved	–	–	–
2	SLEEPDEEP	R/W	0	Enable SLEEPDEEP output signal when entering <b>sleep</b> mode
1	SLEEPONEXIT	R/W	0	Enable SleeponExit feature
0	Reserved	–	–	–

**Figure 4. The ARM Cortex-M Sleep-on-Exit capability reduces power consumption by avoiding unnecessary program execution and by reducing unnecessary stack push and pop operations.**  
 Courtesy of “*The Definitive Guide to the ARM Cortex-M3*”.

## Run Faster, Sleep Deeper with the ARM Cortex-M4

Like many MCUs, Cortex-M3/4 processors can often achieve energy savings in interrupt-driven applications by running at a relatively high clock rate. This counter-intuitive but commonly used energy-saving tactic works well if the processor spends much of its time in a sleep mode where the savings from its reduced active time far outweighs its slightly higher operating current. Put simply, expending 10 percent more power for 20 percent less time represents an overall energy savings.

This technique can be applied to any Cortex-M series processor, and applications that involve compute-intensive tasks also can often benefit from the Cortex-M4 processor’s added capabilities. Its single-cycle DSP instructions and optional floating point accelerator can greatly reduce the number of execution cycles required for functions such as digital signal conditioning, filtering, analysis or waveform synthesis.

Some applications simply need the processing horsepower of a DSP. For example, some security systems employ a device that senses glass breakages using acoustic analysis. Breaking glass is accompanied by a distinctive series of sounds and vibrations that culminate in a resonance at the characteristic natural frequency of the glass, in this case around 13kHz. Most systems employ a sensor interface that only wakes up the processor when telltale frequencies are detected, but designs using a Cortex-M4 DSP-enabled CPU achieve additional energy savings by performing the actual glass break analysis more quickly than software-based solutions.

Even greater energy savings can be realized in these applications using M4-based MCUs that include advanced sleep modes and autonomous peripherals that perform many routine tasks while the CPU remains asleep. For instance, the Cortex-M4 equipped Wonder Gecko MCU7 has five distinct low-energy modes including a 20 nA shut-off state and 950 nA deep sleep mode (running real-time clock, with full RAM and register contents retained and brown-out detector enabled).

The same features that enable energy savings can also yield other advantages. For example, applications such as ultrasonic/acoustic water meters, which must operate for years on a small battery, require the MCU to remain in sleep mode as long as possible. In addition to helping to reduce the MCU's wake time, the Cortex-M4 DSP and floating-point math instructions also eliminate the need for expensive ultrasonic flow transducers by using sophisticated filtering functions to extract the necessary information from the output of inexpensive acoustic sensors. In this application example, the Wonder Gecko MCU's peripherals provide additional energy savings by acting as an analog state machine that wakes the Cortex-M4 processor only when needed.

## Conclusion

While by no means complete, this modest collection of tips and tricks should give you a good idea of how you can exploit some of the Cortex-M series' lesser-known features to your advantage in your next design. The References section at the end of this article provides the details you'll need to put these and other important features of the ARM Cortex-M family to work.

In addition, the EFM32 Gecko and Wonder Gecko MCU families from Silicon Labs provide examples of how the choice of an ARM-based MCU with the right combination of I/O, accelerators and other advanced peripherals can improve a design's performance, energy consumption and solution cost. For more details about these energy-friendly MCUs, see the Additional Reading list<sup>8</sup>, which provides additional information on the unique autonomous peripheral system and other advanced capabilities of EFM32 MCUs.

## References

- 1 - *The Definitive Guide to the ARM Cortex-M3*; 2nd Edition; Joseph Yiu; Newnes; 480 pages; 2009; ISBN 978-1-85617-963-8 - <http://en.wikipedia.org/wiki/Special:BookSources/9781856179638> (Online Sample - [http://books.google.com/books?id=mb5d\\_xeINZEC&printsec=frontcover&dq=isbn:9781856179638](http://books.google.com/books?id=mb5d_xeINZEC&printsec=frontcover&dq=isbn:9781856179638))
- 2 - "*Cortex-M3 Technical Reference Manual Revision r2p1*" - ARM Limited, <http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.ddi0337h/index.html>
- 3 - "*About the Serial Wire Output*" - CoreSight Components Technical Reference Manual, ch11.1 ARM Limited - <http://infocenter.arm.com/help/topic/com.arm.doc.ddi0314h/Chdhgebe.html>
- 4 - *J-Link SWO Viewer* is a tool which allows showing terminal output of the target performed via the SWO pin. More information available here - [www.segger.com/j-link-swo-viewer.html](http://www.segger.com/j-link-swo-viewer.html)
- 5 - More information about Energy Micro's energyAware Commander is available at <http://www.energymicro.com/lizard-labs/energyaware-commander>
- 6 - More information on EFM32 debug and SWO is available here: [http://cdn.energymicro.com/dl/an/pdf/an0043\\_efm32\\_debug\\_trace\\_capabilities.pdf](http://cdn.energymicro.com/dl/an/pdf/an0043_efm32_debug_trace_capabilities.pdf)
- 7 - A video demonstrating how Energy Micro's energyAware Profiler can be used to correlate source code execution and map energy consumption in real-time is available at this link: <http://www.youtube.com/watch?v=9EDq5W1deIM>
- 8 - Additional Reading  
"ARM DSP Instruction Set Extensions" ARM Limited - <http://www.arm.com/products/CPUs/cpu-arch-DSP.html>  
"The History of The ARM Architecture: From Inception to IPO" – Markus Levy, <http://www.reds.ch/share/cours/ReCo/documents/TheHistoryOfTheArmArchitecture.pdf>  
More helpful tips on making the most of the ARM Cortex-M3/4 architecture and Energy Micro's proprietary LESENSE interface may be found in - Application Note # AN0028 - [http://cdn.energymicro.com/dl/an/pdf/an0028\\_efm32\\_lesense\\_capacitive\\_sense.pdf](http://cdn.energymicro.com/dl/an/pdf/an0028_efm32_lesense_capacitive_sense.pdf)

Additional information on Silicon Labs 32-bit MCUs can be found at [www.silabs.com/MCUs](http://www.silabs.com/MCUs)

# # #

Silicon Labs invests in research and development to help our customers differentiate in the market with innovative low-power, small size, analog intensive, mixed-signal solutions. Silicon Labs' extensive patent portfolio is a testament to our unique approach and world-class engineering team. Patent: [www.silabs.com/patent-notice](http://www.silabs.com/patent-notice)

© 2014, Silicon Laboratories Inc. ClockBuilder, CMEMS, DSPLL, Ember, EZMac, EZRadio, EZRadioPRO, EZLink, ISOModem, Precision32, ProSLIC, Silicon Laboratories and the Silicon Labs logo are trademarks or registered trademarks of Silicon Laboratories Inc. ARM, Cortex-M4 and Cortex-M3 are trademarks or registered trademarks of ARM Holdings. ZigBee is a registered trademark of ZigBee Alliance, Inc. All other product or service names are the property of their respective owners.