Part 3 of a 3-part Series of Cortex:

ARM® Cortex: M3 Challenges the Economics of 8-bit Microcontrollers

Author:
J. Scott Gardner, President and Principal Analyst, Advantage Engineering LLC

Synopsis:
The rules have changed for system designers. Forget the comfortable notion that designing for low-power and low-cost eliminates the need to consider 32-bit microcontrollers. As if system designers didn't already have a tough enough task, they now must carefully consider a wider assortment of CPU architectures before identifying the best solution that meets cost, power and performance constraints. Now that ARM-based microcontrollers sell for as little as a dollar, an economic barrier has shattered, and RISC microprocessors now have to be seriously considered for almost any microcontroller application.

The primary goal of this article is to analyze some of the technical trade-offs facing system designers as they work to identify the pros and cons of using a 32-bit microcontroller in applications where an 8-bit device offers sufficient performance.

Some readers of Information Quarterly (IQ) probably define themselves as experts with the ARM architecture. More broadly, the expertise may extend to other 32-bit RISC architectures, but any empathy for 8-bit system designers has faded over the years. If your CPU cost and power budgets allow you to use an ARM processor, why bother keeping up with 8-bit CPU technology with all its horrible resource and performance constraints? However, give some consideration to the dilemma facing designers of low-end, microcontroller-based systems. After all, in terms of new designs, ARM-processor based products are a small fraction of the design activity in the overall embedded system industry. With the new ARM-microcontroller processor based economics introduced by Cortex-M3TM, thousands of 8-bit products have now become part of the ARM universe. And those weary 8-bit experts now have to come up to speed on 32-bit CPUs. The tent encompassing the ARM camp just got a lot bigger, and the ARM community should welcome the newcomers to the fold while recognizing how challenging it must be for these engineers to cover such a broad range of architectures.

This issue of IQ focuses on automotive and industrial control applications. While automotive applications are more likely to justify a customized device (often with an ARM core), industrial control applications utilize a range of standard products. Since most microcontroller-based designs have severe power and cost constraints, a new design project has historically started with a survey of 8-bit (and sometimes 16-bit) microcontrollers to see which vendor offers the required combination of peripherals while meeting the cost, power and performance targets. Of course, the choice of CPU is often heavily influenced by the availability of good development tools, but the CPU search was usually constrained to focus on 8-bit devices. However, with the arrival of ARM Cortex processors (ARM CortexM3) from lead-partner, Luminary Micro, designers of industrial control systems are able to expand their CPU selection process. The focus here is on the system designer, using real-world examples to highlight the design tradeoffs.

Why not 32 bits? It's all about the software.

Many 8-bit CPU experts question why low-end, microcontroller-based systems need the performance of a 32-bit CPU. The die area efficiency of Cortex-M3 allows Luminary Micro to use $1 entry-level CPU pricing and turn this question on its head to ask, "Why not 32 bits?" If there isn't a cost penalty, and the 32-bit device fits in the power budget, why not use a modern CPU architecture? Answering this question can start with a detailed examination of the system design constraints and a technical comparison to alternative 8-bit and 16-bit architectures. Many system designers go through a rigorous evaluation process each time they start a major new design, establishing the technical design constraints and giving themselves extra performance headroom to account for the inevitable demands from the software developers. However, embedded system designers have accepted the reality that software development, debugging and maintenance are the primary drivers of project cost and schedule. The key criterion is software reuse; can a project spread the software development investment across multiple products? In many cases, similar efficiencies flow from adapting off-the-shelf software to further reduce development time. Industrial control applications definitely fall into this category, since the volumes aren't exceptionally high, and many projects even integrate off-the-shelf boards with very little new hardware design of the CPU subsystem. With a few exceptions, the primary enabler of software reuse has been the move to high-level programming languages and robust development tools that support software integration and testing. Modern architectures, such as the ARM Cortex-M3, were developed to minimize the need for assembly code while optimizing the architecture to work seamlessly with powerful debugging tools. While a system designer may see an ARM-based microcontroller as overkill for a design where an 8051 can meet the proposed performance constraints, it is now all about the software when it comes to minimizing project schedules and costs.

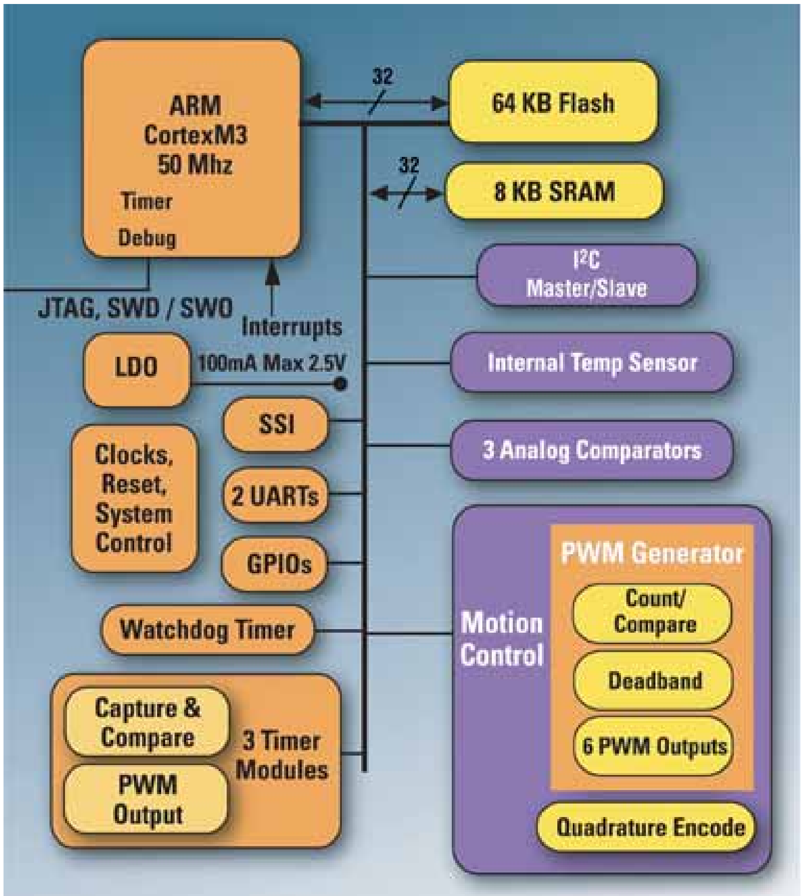

Figure 1: The Stellaris LM3S801 is one of the newest Cortex-M3 devices from Luminary Micro, bumping up the clock rate and adding more motor control features.

A closer look at industrial controller design constraints

Having admitted that the software environment is the first-order driver for CPU selection in embedded systems, we can dive into more of the glorious details about hardware issues that impact the CPU selection process. Selecting a microcontroller requires an evaluation of the constraints for power consumption (both active and standby), onboard memory requirements, and system interfaces to both general-purpose and specialized peripherals. As it turns out, the CPU core is a very small percentage of modern microcontrollers, since a fully-configured Cortex-M3 is only about 60K gates. Integrated peripherals and memory take up most of the die area, while the package pins are dominated by general-purpose I/O. Figure 1 illustrates the LM3S801, one of the products in a recent wave of new announcements from Luminary Micro, expanding its portfolio to 19 MCUs. All of the Stellaris devices conform to this basic architecture, built on the Cortex-M3 and differing in memory size, peripherals, and I/O. With the exception of the label on the CPU core, these block diagrams looks similar to a multitude of 8-bit and 16-bit microcontrollers. To a system designer, this isn't a RISC microprocessor; it's a microcontroller, and the key questions are based on how well the configuration of memory, peripherals and I/O match the system needs. The incredible diversity of system configurations creates a competitive environment where microcontroller vendors need a huge portfolio of devices to choose from. This is why Luminary Micro has been forced to rapidly introduce new Cortex-M3 products that span a range of price, performance and configuration.

Extending Cortex-M3 economics to lower memory costs

Many system designers have struggled with the popular misconception that CISC CPUs require less program memory storage than RISC. This notion dates back to the RISC-versus-CISC wars during the 1980s, since 32-bit RISC compilers generated a larger code segment than 32-bit CISC compilers. However, an 8-bit CISC machine like the 8051 is actually a very inefficient architecture for modern high-level software. Moreover, the Cortex-M3 uses the Thumb2 instruction set which includes both 16- and 32-bit instructions that are optimized for high-level languages and mixed data types. The results are dramatic, since ARM reports that Cortex-M3 requires about 1/4th the code space of an 8051. This code size savings becomes very important to system designers who deal with the fact that memory requirements tend to double over the initial project estimates. Since onboard memory size is one of the biggest drivers of chip cost, once again the Cortex-M3 challenges the perceived economic benefit of 8-bit devices.

High performance can save power?

Power is the one area where 8-bit architectures would seem to have an uncontested advantage. After all, a 32-bit architecture has more transistors burning leakage power in standby mode. In active mode, the wider buses on a 32-bit device should burn more power. However, once again, the CPU core is a very small part of the overall microcontroller device. Estimates from ARM show that the Cortex-M3 core burns about .19mW per megahertz (.18 micron), while a complete 8051-based microcontroller (Atmel AT89LP) is advertised as low power at 3mW per megahertz. Obviously, the CPU core is a very small contributor to total active power consumption when peripherals, memory and I/O are factored in. Still, how can a 32-bit device actually save power in a system? For most systems with power constraints, the real issue is average power consumption, whether the constraint is battery life or just energy costs. Most low-power systems stay in standby/sleep mode as much as possible until a system event brings them out of standby to execute a task, then go back to sleep. A common example is a device which sleeps until receiving data from a sensor. The power consumption in standby/sleep mode is infinitesimal when compared to active mode, so average power is based on the

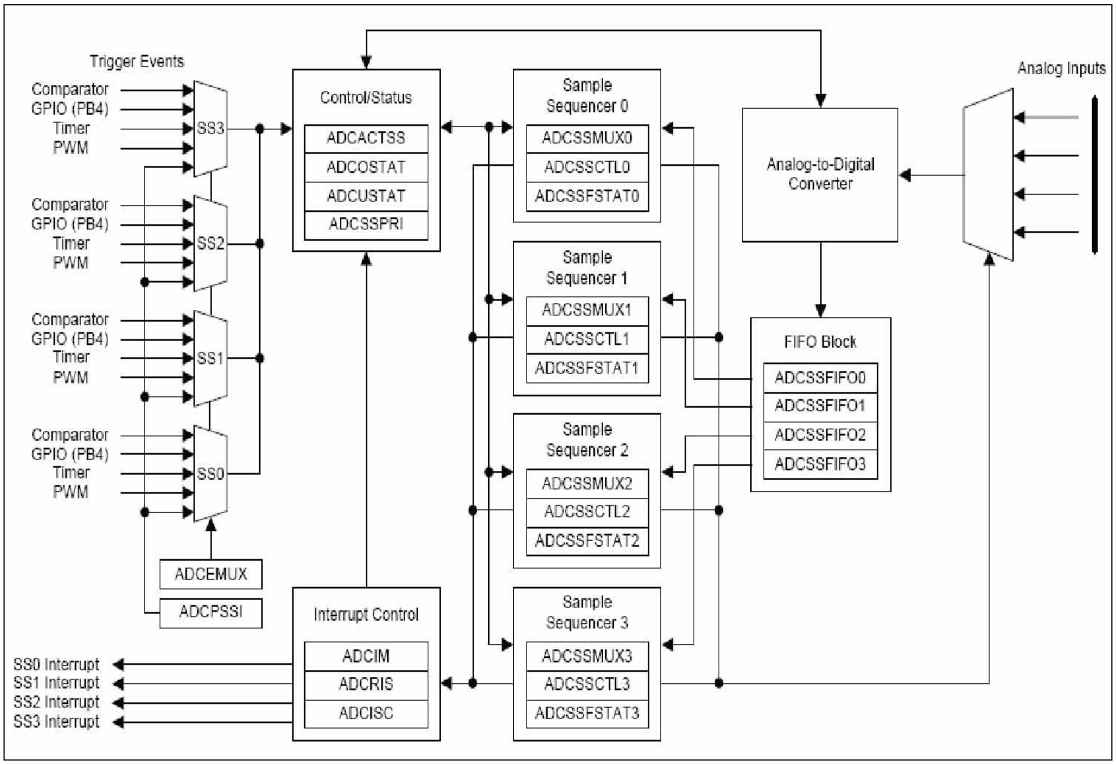

Figure 2: Luminary Micro's multi-channel ADC utilizes programmable sequencers with configurable trigger events to maximize flexibility and offload the CPU.

duty cycle time spent in active mode. Since a 32-bit Cortex-M3 device can execute tasks many-times faster than an 8-bit device, the shorter time spent in active mode results in a corresponding reduction in average power. A 32-bit device that, at first, seemed like performance overkill, actually may be the most efficient and cost-effective solution.

Industrial control designers have been choosing Cortex-M3

Luminary Micro has reported several design wins from industrial control customers that have expanded the CPU selection process to include the 32-bit Cortex-M3. Each of these 3 example products had previously been based on 8-bit microcontrollers, but Luminary Micro's pricing strategy removed cost as an obstacle to adopting a 32-bit architecture for the new design. In many cases, the customers were easily sold on the overall advantages of ARM processors and the extensive ARM software ecosystem. Cost parity with 8-bit devices allowed system designers to evaluate Luminary's Cortex-M3 devices with the traditional microcontroller criteria for features and performance.

Example #1: Ultrasonic Water Flow Meter

The customer's previous designs used Microchip and Atmel 8-bit microcontrollers, but code reuse was a big issue that led to consideration of the ARM architecture. The design required a large number of general-purpose I/O (GPIO) and 3 analog-to-digital (ADC) conversion channels. While Luminary Micro's lowest-cost devices do not integrate an ADC, the 3xx family added a unique multi-channel ADC on-chip. Since two of the example designs use this ADC, it's worth highlighting in Figure 2, illustrating the programmable sequencers that offload the CPU from the real-time task of dealing with multiple channels at variable sample rates. The sequencers can be controlled by numerous trigger events and can dedicate one of the sample channels to an on-chip temperature sensor for device calibration purposes. The flexible ADC was one of the technical factors that ultimately led the customer to choose the Stellaris LM3S301.

Example #2: Motor Control - Electronic Variable Speed Drives

Previous designs used 8-bit devices from Freescale and Cypress, but this customer viewed the highly-integrated, higher-performance Cortex-M3 devices from Luminary Micro as a way to reduce cost. The application uses 4 ADC channels and pulse-width-modulation (PWM) outputs to control motors. The 8-bit processors were running into performance limitations when dealing with the large number of multiply-accumulate operations in the motion control algorithm. Code reusability was also a consideration, but not as important as with the other design win examples. The need for integrated ADC led this customer to the Stellaris LM3S315.

Example #3: Unit Counter for Assembly Line

Past designs used 8-bit processors from TI and Microchip, but once again the strength of the ARM camp led the customer to the conclusion that code reuse would be improved with a move to ARM. The design requires lots of GPIOs and 2 PWM outputs to drive a speaker. The fast interrupt response feature of Cortex-M3 was also important, since the application used a GPIO input to signal when an item on the assembly line crossed an infrared beam. This design now uses the Stellaris LM3S310.

No application is cooler than a robot car

Since the current Cortex-M3 customers aren't ready to show off their products, it's worth highlighting an application that Luminary Micro put together for its product launch. Figure 3 on page 32 shows a picture of the autonomous vehicle that the engineers at Luminary Micro designed, built and programmed – all in C code. The original car used a 12 MHz LM3S102 controller card with software control of brushed DC motors using PWM outputs. The motors control the wheels and allow the car to turn when it hits an obstacle. The car actually "sees" which way to go by reading an optical sensor, even turning on front headlights if a photo sensor notes that it has gotten dark. When turning, the car lights up the rear lights to indicate the type of turn. The software algorithm includes some randomization elements to keep the car exploring new terrain.

Cortex-M3 microcontroller economics gaining momentum

This has been a brief look at how the Cortex-M3 is impacting the microcontroller industry, and the conclusion should highlight the obvious gains that ARM has been making in the fight to convert 8-bit designs. Luminary Micro has been in the enviable position of lead partner for ARM during the development and launch of the Cortex-M3 core, and the company has been moving quickly to take advantage of this opportunity to capture customer mindshare. For readers who look to the Luminary Micro website for more details on the 19 announced Cortex-M3 products, be sure to check out the video demo of the robot car. The engineers continue to tinker with this neat toy, and the car now features an LM3S315 with support for ADC channels and even a Zigbee radio to communicate back to a PC. The website also features a link to a recent Microprocessor Report article by this author, providing a more in-depth analysis of the Cortex-M3 and Stellaris architectures.

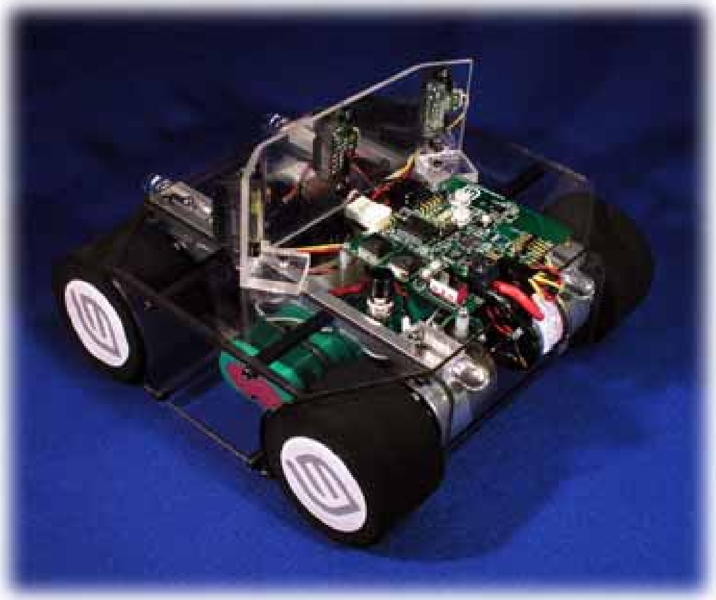

Figure 3: Luminary Micro's Cortex-M3 robot car was developed to showcase new products and give the engineers something fun to play with.

As 8-bit system designers begin to recognize that the CPU landscape has changed forever to make 32-bit CPUs a viable option for low-end designs, the ARM camp will change as well. While the existing consumer, telecom and automotive markets will dominate mindshare through high-volume production, the huge number of design starts for low-end systems will have a dramatic effect on the ARM ecosystem. Tools vendors will certainly welcome the expanding customer base, but nimble new companies like Luminary Micro will continue to thrive in such a diverse market. Even as the larger players introduce their own Cortex-M3 products, the ever-expanding number of applications will allow the entire ARM camp to grow.