



Moving to the ARM® Cortex™-M3 from 8-Bit Applications

by Wendell Smith

8-bit CPUs ship in billions of microcontrollers each year. These low-cost chips meet specific real-time performance requirements but are limited by constraints in performance, software, and development tools. The ARM® Cortex™-M3 32-bit processor core is a modern 32-bit architecture optimized for cost-sensitive microcontroller applications. This paper introduces the ARM Cortex-M3 and discusses key considerations in moving from proprietary 8-bit architectures.

Characteristics of 8-Bit Microcontrollers

Microcontrollers surround our lives, toiling invisibly in our cars, in our appliances, on factory floors, and in our communications devices. From their first introduction 30 years ago, the use of microcontrollers has increased to billions of units every year in almost every electric device. Today, 8-bit architectures dominate the category over their 4-bit, 16-bit, and 32-bit counterparts.

What makes those 8-bit microcontrollers so useful to a wide range of applications? A couple of very important considerations account for much of the appeal:

- **Low cost.** The applications that ship 8-bit microcontrollers, almost by definition, demand a low system cost.
- **Deterministic, real-time interrupt performance.** Many applications of microcontrollers must service interrupts in a certain period of time. One of the benefits of a simple architecture is determining whether these constraints can be met. No complicated memory hierarchies, such as first- and second-level caches, exist to delay the interrupt response time.

However, limitations of 8-bit microcontrollers often provide design and development challenges:

- **Performance limitations.** Because of cost, 8-bit microcontrollers continue to be shoehorned into applications that have outgrown the devices. The resulting code optimization and tweaking—and the lack of quality C compilers—increases development time and cost. And there is no headroom left—any increase in performance requires a move to a noncompatible architecture.
- **Address space limitations.** Address space is small compared to the 4 GB available to 32-bit architectures. Many 8-bit architectures limit direct addressing to a maximum of only 64 KB.



- **Predominance of assembly language.** Applications are most commonly programmed in assembly. In some cases, performance requirements and addressing limitations encourage the use of assembly. More commonly, compiler support is limited as 8-bit instruction sets were not designed with high-level languages in mind.
- **Proprietary architectures.** No CPU architecture dominates the 8-bit microprocessor market. No single line has greater than a single-digit percentage market share, and the largest suppliers have several incompatible lines. This means that switching suppliers requires switching architectures—a non-trivial task given the challenge of porting assembly code.
- **Limited tools support.** The fragmentation of the 8-bit microcontroller market also limits the available hardware and software tools. The divisive market makes it difficult to leverage investments in tools to a different supplier and architecture.

When 8-bit microcontrollers (such as the 8051) were introduced in the 1970s, a typical integrated circuit contained 10,000 transistors. Today, with much smaller process geometries, million transistor ICs are common, and architectural trade-offs that made sense with 10,000 transistors do not make sense today.

With the Cortex-M3, ARM has enabled a transition from the constraints of legacy 8-bit architectures to a modern, efficient core optimized for these low-cost, high-volume applications.

The rest of this paper introduces the ARM Cortex-M3 processor and describes its relevance to those applications that have in the past been limited to 8-bit options.

ARM Cortex-M3 Features

The ARM Cortex-M3 processor provides a high-performance, low-cost platform that meets the needs of minimal memory implementation, reduced pin count, and low power consumption, while delivering outstanding computational performance and exceptional system response to interrupts.

Small, High-Performance Core

The ARM Cortex-M3 processor is the smallest 32-bit core designed by ARM, with just 33,000 gates for the processing core. This design reduces silicon area requirements, enabling the use of small packages or the manufacturing of devices in low-cost processes, such as 0.35 μ and 0.25 μ .

Even with this small core size, a high level of performance is maintained, as the ARM Cortex-M3 delivers 1.2 Dhrystone MIPS per MHz, a performance level greater than many larger 32-bit processors.

Thumb®-2 Instruction Set

The Thumb® instruction set is an innovative ARM technology to reduce code size. Thumb technology adds 16-bit instructions to the 32-bit RISC architecture. Use of these 16-bit instructions substantially reduces the size of compiled application code.



The Thumb-2 instruction set is binary-compatible with existing Thumb implementations but also features hardware divide and single-cycle multiply instructions—DSP-like performance features not typically found in microcontrollers. These instructions are blended with 32-bit ARM instructions to create a compact instruction set that can be efficiently supported by high-level languages, an important benefit of using a 32-bit architecture.

The ARM Cortex-M3 processor instruction runs only Thumb-2 instructions, so these code-size savings are applicable to operating system code and interrupt service routines. This provides additional savings compared to Thumb, since the code-compression benefits of Thumb only apply to user-application code.

All this results in code density that is usually associated with 8-bit and 16-bit devices, but with the full performance of a 32-bit processor.

Support of Unaligned Data

Many 32-bit processors require data to be aligned on *natural* size boundaries. This means that 32-bit words must lie on addresses that are multiples of four. Similarly, 16-bit data must lie on addresses that are multiples of two. The practical impact is if data does not fall on these boundaries, it must be padded to align correctly, resulting in unused holes in data memory (see Figure 1 on page 3).

The ARM Cortex-M3 processor removes that limitation by directly supporting unaligned data storage, enabling compact storage of data (see Figure 2 on page 3).

Figure 1. Data Structure with Aligned Data Storage

Memory Address			
	...		
0x02800000	16-bit data	8-bit data	data padding
0x02800004	16-bit data	8-bit data	data padding
0x02800008	16-bit data	8-bit data	data padding
0x0280000C	32-bit data		
	...		

Figure 2. Data Structure with Unaligned Data Storage

Memory Address			
	...		
0x02800000	16-bit data	8-bit data	16-bit data
0x02800004	16-bit data continued	8-bit data	16-bit data
0x02800008	8-bit data	32-bit data	
0x0280000C	32-bit data continued	memory space saved	
	...		



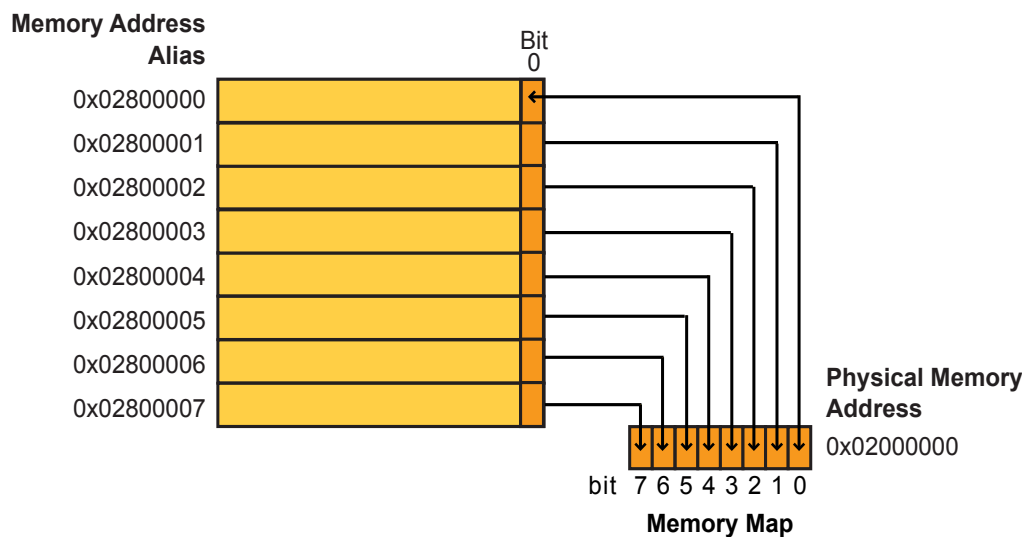
Bit-Banding

As a 32-bit processor, the ARM Cortex-M3 has a memory addressing limit of 4 GB, so, theoretically, it could support a system with gigabytes of RAM. No practical microcontroller comes close to using the full address space in a real system, so the question is how to make good use of that 32-bit addressing capability for those microcontroller systems? ARM ingeniously answered this question with *bit-banding*.

Bit-banding uses address space that aliases peripheral or SRAM address space, allowing a single bit within a word to be manipulated by a reference to a byte at an aliased address. For example, a write to address 0x02000000 modifies the SRAM 32-bit word at that location, but writing to 0x02800000 modifies only bit 0 at address 0x02000000. Figure 3 on page 4 depicts this very useful feature.

This bit-banding operation greatly simplifies bit manipulations. Instead of reading a word, AND'ing in the appropriate bit, and then writing the word back out, bit-banding accomplishes this with a single store instruction. But that single instruction has another benefit: it is an *atomic* operation, executed as a single operation. With the prior method of bit manipulations with sequential instructions doing the read-modify-write operations, an interrupt could occur which, potentially, could change a bit at that memory location. After the interrupt returns, the store could be writing corrupt data back to that memory location. The atomic nature of bit-band operations eliminates that problem.

Figure 3. Bit Banding



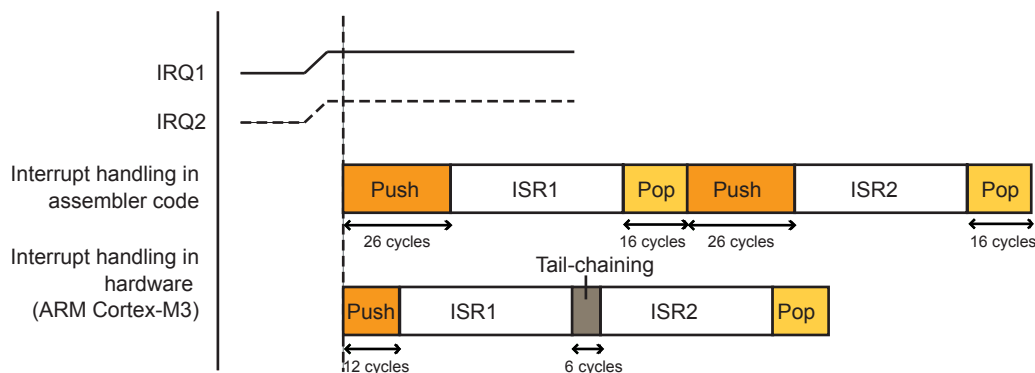


Deterministic Interrupt Handling

Many applications currently dominated by 8- and 16-bit processors, such as motor control, demand very deterministic behavior. The ARM Cortex-M3 processor has architectural optimizations, contained in the integrated Nested Vectored Interrupt Controller (NVIC), for reducing interrupt latency for those microcontroller applications. These features include:

- **Hardware interrupt handling.** The processor hardware saves registers onto the processor stack as part of servicing.
- **Preemption support.** An exception with higher priority than the current exception can be immediately handled, rather than delaying until the current interrupt is completed.
- **Tail-chaining.** The ARM Cortex-M3 processor checks pending interrupts before exiting an interrupt service routine, avoiding unnecessary pop and push operations (see Figure 4 on page 5). This capability guarantees no more than six cycles between successive back-to-back interrupts.
- **Late arrival preemption.** The ARM Cortex-M3 processor can switch to a higher priority interrupt during exception entry. The effect of this is interrupt performance that exceeds many 8-bit processors.

Figure 4. ARM Cortex-M3 Tail-Chaining

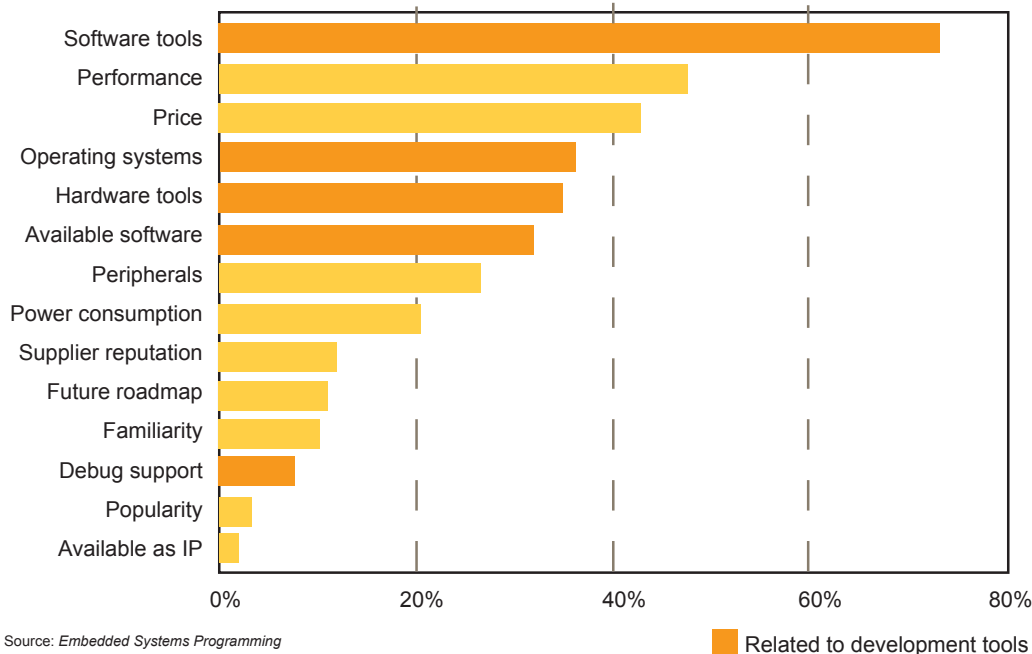


Development Tools

As the competitive pressures increase and product life cycles shorten, more importance is being placed on development tools as a key contributor to development success. A recent survey by *Embedded Systems Programming* illustrates the importance of this. They asked engineers what was important to them when choosing a microprocessor. The perennial favorites—performance and price—placed and showed as they came in second and third in the balloting. The top pick? Software tools! And many other highly rated factors are related to software and hardware development tools (see Figure 5).



Figure 5. Embedded Systems Programming Magazine’s Survey of Factors in Choosing a Processor



Companies can make significant investment in tools—not just in the purchase of tools, but also in the accumulated knowledge and expertise in the use of those tools.

Over 140 companies are licensees of ARM, shipping over 1.3 billion units with ARM processors in 2003 (see Figure 6). These volumes are supported by what is widely considered to be the broadest range of tools in the industry. In addition to the tools that ARM provides directly, numerous third-party tools are available. Besides the benefit of the broad support enabled by the widespread success of the ARM cores, tools from multiple suppliers support the ARM processors. Unlike with 8-bit processors, making the decision to move up in performance does not mean purchasing new tools with learning curves.



Figure 6. ARM Development Tools Support



But tool considerations go beyond debug tools. Most applications that run on 8-bit processors are written in assembly. (Often, performance and address-space constraints force the use of assembly.) In addition, the limitations of 8-bit instruction sets have restricted the use of C/C++ compilers. The ARM Cortex-M3 processor has been optimized for implementation by compilers, so that C/C++ compilers can be used with their obvious benefits for development time, debugging, and portability.

The widespread use of the ARM cores has spawned an extensive user community. Open software such as GNU compilers, ports to open source real-time operating systems, and user-contributed software libraries are widely available.

Summary

Users of low-cost microcontrollers no longer need to be limited to the constraints of 8-bit CPUs. The ARM Cortex-M3 processor brings with it these benefits:

- 32-bit performance at 8-/16-bit costs for cost-sensitive applications
- Small code size for memory-limited applications
- Deterministic interrupt response for timing critical microcontroller applications
- Support of high-level languages for shorter development time
- Strong development tools support



- Protection of investment enabled by the broad tools support across the ARM family of processors

With the ARM Cortex-M3, the benefits of a modern, standard 32-bit architecture are available to all microcontroller applications.

Company Information

Luminary Micro, Inc. designs, markets, and sells ARM Cortex-M3-based microcontrollers (MCUs). Austin, Texas-based Luminary Micro is the lead partner for the Cortex-M3 processor, delivering the world's first silicon implementation of the Cortex-M3 processor. Luminary Micro's introduction of the Stellaris® family of products provides 32-bit performance for the same price as current 8- and 16-bit microcontroller designs. With entry-level pricing at \$1.00 for an ARM technology-based MCU, Luminary Micro's Stellaris product line allows for standardization that eliminates future architectural upgrades or software tool changes.

Luminary Micro, Inc.
108 Wild Basin, Suite 350
Austin, TX 78746
Main: +1-512-279-8800
Fax: +1-512-279-8879
<http://www.luminarymicro.com>

Copyright © 2006-2007 Luminary Micro, Inc. All rights reserved. Stellaris is a registered trademark and Luminary Micro and the Luminary Micro logo are trademarks of Luminary Micro, Inc. or its subsidiaries in the United States and other countries. ARM and Thumb are registered trademarks, and Cortex is a trademark of ARM Limited. Other names and brands may be claimed as the property of others.