

**THE ONGOING EFFORT  
TO SUPPLY LOW-COST, 32-BIT  
PROCESSORS HAS THEM  
BEGINNING TO ENCROACH  
ON THE TERRITORY THAT 8-  
BIT PROCESSORS NOW OWN.**

**A** NEW BREED OF LOW-COST, 32-bit devices is targeting the healthy market share that 8- and 16-bit devices have exclusively supported. According to

Isuppli's 2004 microcontroller-unit-market forecast, 8-bit devices account for more than 40% of sales. However, the assumption behind the migration from legacy 8- and 16-bit applications to 32-bit processors is that the costs of using these architectures will reach parity as manufacturing geometries continue to shrink. Despite the fact that 32-bit cores require more gates and therefore consume more die area—typically more than four times what an 8-bit core requires—the processor core itself consumes relatively little of the die area in a device. And the percentage of die area continues to shrink with smaller manufacturing technologies, versus that of the on-chip memories and peripherals (**Figure 1**).

In fact, 0.18-micron and smaller manufacturing technologies are enabling semiconductor vendors to offer 32-bit microcontrollers whose costs are comparable with those of high-end 8-bit processors. Atmel and Philips now offer ARM7-based microcontrollers for as little as \$3 that include proprietary architectural extensions and features that directly cater to the needs of developers of high-end, 8-bit-system applications. But this move doesn't mark the beginning of the end for 8-bit microcontrollers. In fact, few 8- and 16-bit applications will migrate to larger architectures.

Illustration by Dan Guletera

New 8-bit microcontrollers continue to reach upward in features and performance—reaching 100 MIPS—and 32-bit processors keep pushing downward in performance and cost (see **sidebar** “Squeezing the 16-bit architectures”). The availability of low-cost, 32-bit microcontrollers does not change the process of evaluating processor architectures, but it does increase the number of implementation alternatives available to a design team. The availability of

<i>At a glance</i> .....	32
<i>Squeezing the 16-bit architectures</i> .....	32
<i>For more information</i> .....	34

# Reaching down: 32-bit processors aim for 8 bits

these low-cost, 32-bit processors may mark the beginning of an emerging application space—a gray area, in which choosing whether to use a smaller or a larger processor architecture may no longer be cut-and-dried.

Selecting the proper microcontroller is a critical decision that affects a design project's success. The goal is to select the microcontroller architecture that best meets the design specification and minimizes the overall system cost, including the design and resource requirements for current and future design efforts using the processor. Although price and performance are primary considerations, other factors, such as whether the architecture includes the optimum mix of memory and peripherals and whether it's appropriate for low-power applications, influence design teams. Designers also need to determine whether the processor and accompanying development tools reduce the risk of not completing the design or getting it to market on schedule, and whether the device family can scale up and down in price, features, and performance, enabling the team to quickly respond to market changes.

#### WHY MIGRATE?

Microcontrollers with 8-bit cores successfully support applications requiring motor-control functions, simple sensors, and switches. They also reside in low-end consumer products and enable the replacement of mechanical instrumentation. The nearly two dozen 8-bit-proces-

#### AT A GLANCE

- ▶ Lower priced, 32-bit processors are beginning to target 8- and 16-bit architectures.
- ▶ The standby-power dissipation of 32-bit processors is significantly worse than that of 8- and 16-bit architectures.
- ▶ Hybrid 8- and 32-bit implementations can enable an incremental migration effort.
- ▶ Low-cost, 32-bit processors may open application spaces rather than replace 8- and 16-bit applications.

sor vendors are still consistently and regularly releasing new devices and families, and, for many applications, an 8-bit architecture is sufficient. Yet forces are still driving 32-bit architectures to push down into the high-end, 8- and 16-bit processor markets.

Rising computational performance and larger memory requirements are the primary drivers pushing 8-bit designs toward wider architectures. The need for more efficient motor control to meet energy standards is pushing up the computational requirements for some motor-control applications. The increasing use of encryption is also driving up the computational requirements. Smart devices that support a connection to a network and require a network stack are increasing memory requirements. To maintain flexibility, some applications are imple-

menting functions such as audio decoding, image processing, and soft modems as software rather than as external hardware with an 8- or a 16-bit processor. Applications supporting multiple functions may need to adopt an RTOS to manage the complexity, which also drives up a system's memory requirements.

The data width of 8-bit architectures limits the amount of real-time computation these devices can perform. An 8-bit processor would have to operate faster to process wider data types, whereas 16- and 32-bit processors use a register and bus structure that more naturally handles those wider data sizes. Some 32-bit processors can perform SIMD (single-instruction, multiple-data) arithmetic operations that act on multiple 8-bit data at the same time. For 8-bit designs to support the expanding computational loads for these applications, external hardware extensions would have to handle the heavy lifting.

The size of the address register limits the addressable memory of an 8- or a 16-bit architecture. Using an 8- or a 16-bit processor becomes less reasonable as memory requirements exceed 64 kbytes. Some 8-bit architectures support addressable memory greater than 64 kbytes, even up to 1 Mbyte, but exceeding the 64-kbyte address space increases complexity because it requires handling an additional layer of pointers or paging registers. Network stacks for 8-bit processors are available for licensing, but they must eliminate support for some func-

## SQUEEZING THE 16-BIT ARCHITECTURES

According to Isuppli's 2004 microcontroller-unit-market forecast, the 16-bit microcontroller market is worth \$4 billion in units shipped and growing; it is larger than the current 32-bit embedded-system-microcontroller market. The 16-bit market exists in spite of price overlaps with high-end, 8-bit and low-end, 32-bit architectures. Marketing material suggests that 32-bit processors are better choices than 16-bit processors for applications with lower processing requirements because, despite their comparable prices, 32-bit processors

offer better performance.

Obviously, more goes into choosing these devices than price and performance determination; other factors, including lower power consumption, device size, the integrated peripheral set, and development tools, also matter.

The 16-bit-microcontroller market lacks a standard architecture that dominates the device space; it comprises proprietary architectures. In stark contrast, many processor vendors offer and support the 8051 and ARM architectures, making them de facto standard

architectures for 8- and 32-bit microcontrollers, respectively. Wide vendor support for these architectures provides a critical mass that reduces the risk for RTOS and development-tool providers for these architectures; it, in effect, encourages more development support for these architectures and device offerings from more companies.

Whereas a semiconductor company offering an 8051 or ARM-based controller can leverage the third-party development tools for those architectures, vendors of 16-bit proces-

sors must specify and provide the development tools necessary to win a design-in. The size of the 16-bit market points up the fact that these devices reside in a sweet spot to best meet the composite requirements for price, performance, power, size, packaging, features, and development tools for applications currently using them. The specifics of an application's sweet spot also apply when examining 8-bit processors, but, for similar reasons, the larger processor architectures have not usurped the 8-bit microcontroller market.





bit platforms often include an operating system, which adds to a project's development cost. The operating system helps reduce the complexity of developing for the target processor by abstracting and managing the subsystems, including the DMA controllers and the MMU.

### ENCOURAGING THE MIGRATION

The basic ARM7 core is not ideally suited to satisfy all of the features that are important to 8-bit-application developers. As a result, Atmel and Philips added proprietary extensions, such as atomic bit-manipulation operations, to the basic architecture of their \$3 devices. Both companies chose to omit a cache and instead implement a memory subsystem that supports single-cycle execution running from the flash memory to facilitate deterministic operation, a key concern for many control systems. These devices also include power-on-reset and brown-out-detection circuits that are common to 8-bit offerings. Atmel extended the ARM7's basic interrupt facilities to support vectored priority-interrupt handling and to minimize the number of instruction cycles the system requires to handle interrupts.

Mike Skrtic, tools and applications manager at IAR Systems, says, "The ARM-processor core is the 8051 of the 32-bit world, except that it has an upgrade path." The core is available as an off-the-shelf part from many suppliers, which have based a rich third-party ecosystem of development tools and services on these devices.

Jon Ward, president of Keil Software USA, points out, "The basic ARM development tools support the core. For each new ARM device, the development tools need to incorporate vendor-specific start-up code, peripheral, interrupt controller, and Thumb-mode switching libraries." Although the architecture extensions that Atmel and Philips made may be a move in the right direction for encouraging 8-bit-system developers to migrate to a 32-bit platform, they are proprietary extensions that threaten to weaken ARM's position as a portable architecture in the 32-bit market.

The recently announced ARM Cortex M3 processor addresses some of these proprietary-extension issues by standardizing some of the features for low-end, 32-bit processors that may encourage developers to migrate from 8- and

16-bit platforms. With 33,000 gates, the M3 is the smallest and lowest power ARM core; with system peripherals, the core gate count rises to 60,000. In contrast, an embedded 8051 implementation requires approximately 12,000 gates. The M3 core has no caches, and the memory subsystem matches the processor performance with flash-memory performance. The M3 supports atomic bit manipulation of data and peripherals by designating an address range for use with bit operations.

The M3's interrupt handling supports interrupt responses within six to 12 instructions cycles, down from the nine- to 30-clock-cycle response times of other ARM cores. Vectored interrupt addresses pass directly to the processor core when an interrupt occurs, enabling early processing of the interrupt while the pipeline clears. The six-clock-cycle response time is a best-case condition when the processor is tail-chaining or handling the transition from a higher priority interrupt back to a lower priority interrupt; the processor can switch to the second interrupt without restoring and saving the state for the interrupt task. Despite this shorter interrupt and response, 8-bit devices, such as from Microchip, can boast interrupt responses within three clock cycles.

To deliver the best memory usage and highest code density of any ARM core, the M3 does not support the full 32-bit ARM instructions set and executes only Thumb2 instructions. The Thumb2 instructions are parts of future ARM cores, providing an upgrade path from M3-based processors to the other ARM cores. To enable designers to develop devices with fewer pins, the M3 reduces the number of pins necessary to support on-chip debugging from five to one by replacing the current multipin JTAG port with a single-wire debugging technology. ARM's embedded trace module remains an optional subsystem that the M3 core supports. ARM is betting that incorporating these architectural changes into the Cortex M3 core will encourage and accelerate the migration from legacy 8- and 16-bit architectures to its 32-bit platform.

### PORTING PAIN

Everything has its price, and the cost of porting a legacy design to a new platform includes engineering effort and lost time-

*(continued on pg 38)*



to-market opportunities. If designers need to switch to another processor architecture, they should be able to explicitly explain why—for example, whether the added scope of the project is more peripheral-intensive or computationally intensive. Jack Berlien, director of strategic marketing at TAOS, says, “Familiarity with a target architecture and maximizing code reuse are key barriers to resolve when you are considering migrating to a different processor architecture.” If a higher end microcontroller within the same family can accommodate the new scope, using it will positively impact the design team’s ability to quickly market the next design generation.

Porting assembly code can be a significant undertaking. For that matter, porting C code can be an equally daunting task. Even with strict ANSI C code, despite the fact that a C compiler may be available for an 8-bit microcontroller, designers of the C-language specification did not develop it with 8-bit architectures in mind. For example, although the target processor should define an integer, compilers for 8-bit architectures may not consistently specify how the processor should treat integers versus chars and may consider them identical. Target-specific functions and peripheral-access code are additional sources of porting effort. Each component that does not port perfectly to the new target represents a new source of error and a possibly delayed project schedule.

Even when a design project requires 32-bit arithmetic, the cost of migrating legacy code may encourage designers to find ways to avoid migrating the whole implementation to a 32-bit processor. One way that design teams can avoid migrating their designs and legacy code is by incorporating programmable logic or a separate 32-bit processor into the design to handle only the 32-bit arithmetic or higher performance functions. This approach can facilitate a shorter time to market by preserving the legacy design in the product and still incorporating value-added functions. Such a hybrid implementation can allow a design team to incrementally migrate from an older architecture and minimize the risk of delaying the product’s release.

Similarly, for applications experiencing an increase in processing requirements that must maintain lower power operations, design teams may consider combining 8- and 32-bit processors in the same implementation. Roger Bryner, a technical marketing engineer at Xilinx, observes, “We are seeing customers combining [an] 8-bit core with our 32-bit core on the same device to offload the peripheral handling, interrupt handling, and other housekeeping functions from the larger core.” Offloading the housekeeping functions from a 32-bit processor to an 8-bit microcontroller enables the 32-bit processor to perform its higher performance processing and more frequently remain in standby mode, or it can operate at a lower clock rate to save power.

The decreasing cost of 32-bit microcontrollers will open up the number of applications that can take advantage of them. But whether these lower priced processors will supplant 8- and 16-bit designs remains to be seen. Today’s low-cost 32-bit processors are bridge products; they attempt to simplify an 8-bit developer’s transition to a 32-bit architecture. As the price drops further, the usefulness of the peripheral mix and on-chip memory becomes a growing concern. Some ARM licensees are unofficially planning \$1 32-bit devices by 2007, presumably using even smaller process geometries. But we’ve yet to determine whether these devices will support 5V I/O.

In situations in which 8- and 16-bit applications are running out of performance headroom, the switch to 32-bit devices may be a reasonable expectation, assuming that 8-bit vendors avoid modifying their products to continuously meet those changing requirements. Continuing frequent announcements by processor vendors for new 8-bit device families suggest that such a situation is unlikely. However, low-cost, 32-bit processors may see their largest adoption from a new set of applications that require the processing bandwidth of the 32-bit architectures but that designers could not achieve until the devices reached a lower price. □



You can reach Technical Editor Robert Cravotta at 1-661-296-5096, fax 1-661-296-1087, e-mail rcravotta@edn.com.

### TALK TO US

Post comments via *TalkBack* at the online version of this article at [www.edn.com](http://www.edn.com).