

# ARM targets automotive and industrial dominance

RECENT STATISTICS CONFIRM THAT ARM'S DOMINANCE IN WIRELESS TELEPHONY IS VIRTUALLY COMPLETE. BUT WHAT DOES THE ARCHITECTURE HOLD FOR AUTOMOTIVE AND INDUSTRIAL DESIGNERS LOOKING TO MIGRATE TO A STABLE, LOW-COST 32-BIT PLATFORM—AND WHAT'S AVAILABLE TO HELP?

**B**ack before the PC owned the personal computer market, British company Acorn sold the BBC Microcomputer—so called because the national broadcaster ran educational programmers tailored for this machine. Engineers loved the BBC Micro because its highly efficient BBC-Basic compiler freely mixed high-level code and assembly language and the system offered direct access to I/O. As a result, the machine found roles in laboratories and industrial applications that its designers could hardly have foreseen. Spurred on by success while recognising the need for a more powerful processor than an 8-bit 6502, Acorn embarked on designing its own silicon—eventually adopting the RISC

approach that researchers David Patterson at Berkeley and Bell Laboratories' David Ditzel proposed back in 1980 (Reference 1). In 1985, the ARM1 became the first commercially available RISC chip.

Ultimately, Acorn spun off its chip-design department into the fabless intellectual-property concern Advanced RISC Machines—now better known as ARM. Since its inception in 1991, this company's dominance in mobile products has come to rival the PC's desktop supremacy. According to a recent story in *Electronics Weekly*, ARM's chief executive officer Warren East said with respect to the company's market share in 3G phones: "If it's not 100 percent, it's very close to 100 percent" (Reference 2). East went on to note that despite the telecoms success, ARM's non-wireless business has grown faster than its wireless business: "Non-wireless shipments are half a billion a year and non-wireless is growing at 44%, whereas overall we're growing 30%." So what's the secret of ARM's success, and how useful are its architectures beyond the confines of mobile telephony—within automotive and industrial applications, for instance?

The key advantage that RISC promises for power-sensitive applications is low power consumption with minimal performance impact. ARM has long claimed

leadership in the MIPS-per-Watt race with, for instance, its new Cortex-A8 core achieving 600 to 800 MHz throughput for 300 to 400 mW of power consumption. By comparison, a popular DSP such as Analog Devices' Blackfin—itself a highly efficient machine—consumes around 550 mW at 600 MHz in normal operation. RISC cores also require less silicon area than traditional microcontrollers, making it possible to include more peripherals. These features allow silicon vendors to offer stand-alone ARM-based devices that span basic 32-bit microcontrollers to

complex application-specific standard products. First though, it's worth reviewing the ARM landscape to identify the core variants on offer and their relative feature sets. The company categorises its products as application cores, embedded cores, and secure cores. Application cores are those that run commercially available operating systems (OS) including Linux, Palm OS, Symbian OS, and Windows CE. These processors are intellectual-property designs for third-party integration and generally have an optional floating-point-unit to accelerate 3D graphics. The latest offerings comprise the ARM11 MPProcessor multiprocessor that can include up to four processor instances, as well as the new Cortex-A8 processor. The range also counts the ARM7 and ARM9 variants that are popular in the embedded space, with all devices in this group including a memory-management unit (MMU) to enable OS operation.

With a memory-protection unit (MPU) and an optional cryptographic coprocessor, the secure core group includes a host of security features to suit applications such as smartcards. But it's the embedded core range that's of greatest interest to most designers, where the ARM7 and the higher-performance ARM9 rule. Various bus interfaces are available, including the company's AHB (advanced high-performance bus), its open-source AMBA (advanced microcontroller bus architecture), and AMBA's latest derivative, AXI (advanced-extensible-interface). Cores may also support DSP instructions and Jazelle, ARM's Java acceleration technology. All variants support the Thumb instruction set, a 16-bit subset that compresses the commonest 32-bit ARM instructions to save on mem-

## AT A GLANCE

- \* The ARM7TDMI processor enjoys industry-standard acceptance.
- \* Vendor-specific enhancements target automotive and industrial applications.
- \* Upwardly compatible ARM processors support painless performance hikes.
- \* Silicon and 3<sup>rd</sup>-party vendors provide strong hardware and software support.
- \* ARM's new Cortex family promises to dominate the embedded space.

ory cost, expanding them at runtime with minimal performance penalty.

In deliverable terms, there's a further product subdivision into macrocells that comprise a physical layout tailored to a specific semiconductor process, and synthesizable IP (intellectual property) that takes the form of a high-level-language definition suitable for use with a cell library. Importantly from an off-the-shelf silicon user's perspective, the synthesis option allows chipmakers to implement many variations of the full feature set, with the result that not all processors include all of the same facilities. For example, chipmakers may exclude the embedded-trace macrocell (ETM) that furnishes hardware address and data comparators, address-range decoders and comparators, counters, sequencers, and external I/O ports.

to move data around its own registers and to exchange data with the ARM's registers and external memory. The coprocessor interface relies on a bus-watch/handshake technique where the coprocessor copies the instruction stream into its own pipeline and synchronously replicates the host's actions. The coprocessor can always start execution providing that it's able to revert to its original state if the handshake doesn't complete. The handshake involves three signals—\*cpi (coprocessor instruction), cpa (coprocessor absent), and cpb (coprocessor busy). When the host encounters a coprocessor instruction, it issues the \*cpi signal. If no coprocessor is present, the cpa signal remains active and the core enters an undefined instruction trap sequence. If the coprocessor accepts the instruction but is busy, it negates cpa but leaves cpb active. The core then runs a wait sequence during which it may respond to interrupts before retrying. When all three signals are low, the coprocessor is ready to execute its instruction.

## ARM7TDMI RULES EMBEDDED

From over 1 billion ARM cores shipped last year, the ARM7TDMI remains the most popular. Understanding this core's major features is the key to the range, as later products extensively build on this foundation. For instance, the ARM720 includes a coprocessor to control its cache and MMU, while ARM9 adds another two stages to the ARM7's three-stage pipeline to simplify the pipeline logic and allow higher clock speeds. This later architecture also splits ARM7's unified databus into separate data and instruction-memory busses to increase memory bandwidth. Crucially, the higher performance architectures are upwardly code-compatible with ARM7, furnishing a scaleable upgrade path that extends to multiple cores. See sidebar "ARM7TDMI Primer" for an overview of the chip's major characteristics.

One feature that sets the ARM7 apart from conventional microcontrollers is its support for coprocessors via a hardware interface and an instruction set extension mechanism. There's support for 16 logical coprocessors, each of which can access up to 16 private registers of arbitrary size. Each coprocessor uses the same load/store architecture as the ARM core, with instructions

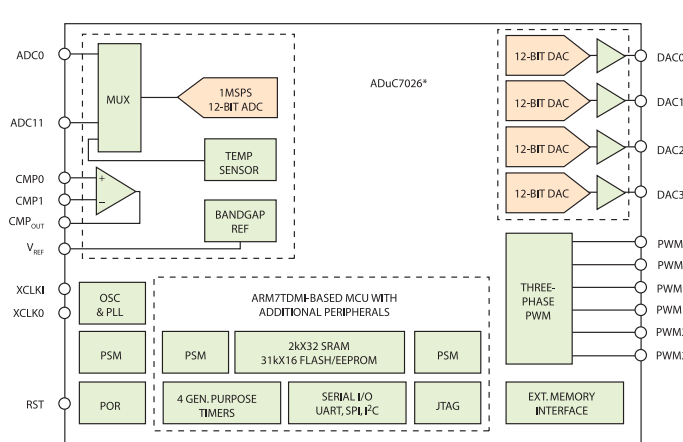


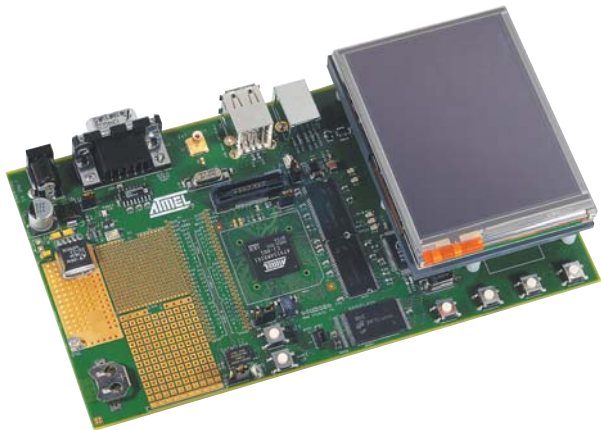
Figure 1 Precision analogue peripherals suit the AD $\mu$ C 70xx family to sensor interface applications.

to move data around its own registers and to exchange data with the ARM's registers and external memory. The coprocessor interface relies on a bus-watch/handshake technique where the coprocessor copies the instruction stream into its own pipeline and synchronously replicates the host's actions. The coprocessor can always start execution providing that it's able to revert to its original state if the handshake doesn't complete. The handshake involves three signals—\*cpi (coprocessor instruction), cpa (coprocessor absent), and cpb (coprocessor busy). When the host encounters a coprocessor instruction, it issues the \*cpi signal. If no coprocessor is present, the cpa signal remains active and the core enters an undefined instruction trap sequence. If the coprocessor accepts the instruction but is busy, it negates cpa but leaves cpb active. The core then runs a wait sequence during which it may respond to interrupts before retrying. When all three signals are low, the coprocessor is ready to execute its instruction.

This system-on-chip interface stimulates a wide range of stand-alone microcontrollers that especially suit automotive and industrial use. For example, the new mixed-signal ARM7TDMI-powered microcontrollers from Analog Devices ideally suit sensor applications. The eight-member AD $\mu$ C 70xx family shares 40 MIPS throughput, 62 kbytes of Flash, and 8 kbytes of RAM (Figure 1). Major peripherals comprise a multiple-input 12-bit ADC that runs at 1 MHz and up to four 12-bit output DACs. The smallest package measures just 6mm<sup>2</sup> and accommodates 13 general-purpose I/O pins. Large

er packages furnish up to 40 GPIO pins and a 16-bit resolution three-phase PWM module that suits medium-complexity drives such as ac-induction motors, while future derivatives may include dedicated multiply/accumulate hardware to tackle sensorless motor control. Some chip variants also offer an external memory interface, but among the common features are a precision 20ppm/ $^{\circ}$ C voltage reference, a temperature sensor accurate to  $\pm 3^{\circ}$ C, and a voltage comparator. Serial communications include a UART, SPI port, and two I<sup>2</sup>C ports. There's also a 16-element programmable-logic-array (PLA) that's useful for mopping up glue-logic functions to implement, for example, a programmable threshold-level interrupt using the comparator and a DAC channel.

Donal Killackey, product marketing manager for precision analogue microcontrollers, says that Analog Devices' choice of the ARM core to partner its existing range of 8052-based devices was natural: "We already had an ARM license for the telecoms products and were working on an instrumentation project for fibre transceivers that needed big compute power in a tiny form factor. Coincidentally, our 8052-based customers were asking for options with more processing power, with many expressing a preference for an ARM7 family." As a result, Killackey says, the company adopted a two-prong approach, offering both custom and generic parts to automotive and industrial OEMs. He foresees new chip variants becoming available with more Flash and interfaces such as CAN (controller-area-network) and LIN (local-area-interconnect). Available now is the \$249 QuickStart Plus development kit that comes complete with an evaluation board, code-limited versions of IAR's Embedded Workbench and Keil's  $\mu$ Vision3 environment, the PLA programming tool, a Windows serial downloader, a power supply and documentation. The kit includes Analog Devices' own nonintrusive JTAG debugger, which—Killackey notes—is RDI (remote-debugger-interface)-compliant for compatibility with a wide variety of debug environments.



**Figure 2** Atmel's AT91SAM9261 development board showcases the ARM926 core for mobile applications.

Renowned for its AVR microcontrollers and its 8051 range, Atmel is another enthusiastic ARM proponent. Peter Bishop, Atmel's communications manager at its Rousset, France facility, says that the company took this route for its mass-market 32-bit products because "ARM is a rock-solid standard in 32 bits." He notes that the ARM7 especially suits application as a connections engine, providing the bridge between different physical layers and protocols. Atmel offers two subfamilies, with its XC variant offering encryption blocks that include AES (advanced-encryption-standard) and triple DES (data-encryption-standard) cryptographic engines. Bishop notes that encryption is no longer limited to areas such as financial transactions: "Secure Web access is now a prerequisite for industrial Ethernet ports. You don't want some hacker wreaking havoc with industrial processes," he warns. Atmel's entry-level ARM7 device is the SAM7S series, which offers 32 to 256 kbytes of Flash together with 8 to 64 kbytes of SRAM; peripherals include a USB 2.0 device. Bishop says that the 7SE variant will also have an external bus interface to augment on-chip memory. Other derivatives include the 7A that features CAN interfaces, the 7L that targets ultra-low-power applications such as environmental sensors, and the new 7X that boasts CAN, Ethernet, and USB interfaces on its 50-MIPs silicon.

For example, the AT91SAM7X128 carries 128 kbytes of Flash and 32 kbytes of SRAM. Its 100-lead package is also home to two UARTs and two SPI ports,

encryption engines. Bishop notes that these designs feature Atmel's multilevel vectored-interrupt-controller that circumvents the native core's two-level interrupt restriction: "The ARM7 is just fine as an execution engine, but its lack of a prioritised interrupt controller often requires software implementations that choke performance." Atmel incorporates a bit-set/clear model for all of its peripherals, which facilitates atomic I/O operations within a single non-interruptible instruction cycle—dispensing with the need to mask and re-enable interrupts around a bit-manipulation routine.

Bishop points to Atmel's long-standing relationship with vendors such as IAR for development support. At the top end, the Integrity RTOS from Green Hills tackles critical applications, but developers can get started with products such as the AT91SAM7X-EK evaluation board that's available for \$249 (also budget \$129 for the AT91SAM-ICE debugger).

Atmel's latest ARM-based product is the AT91SAM9261, which embeds the 200 MIPs ARM926 core. Designed for very low power consumption in battery-powered wireless devices, the device includes an LCD controller that suits black-and-white and colour displays of up to 2048×2048 pixels. A development board showcases these features (**Figure 2**). To further minimise system-power consumption and also reduce the bill-of-materials, it's possible to configure the chip's 160 kbytes of SRAM as a frame buffer. The chip is fast enough to run security algorithms in software, while its hardware comprises optimisations for real-time use,

an 8-channel 10-bit 384k-samples/sec ADC, a PWM-capable counter/timer block, and an 18-channel peripheral DMA controller. The latter device constrains a 10-Mbps stream to consuming only about 4% processor overhead whereas a conventional design exhausts its bandwidth at this point. The similar 7X256 houses 256 kbytes of Flash and 64 kbytes of SRAM, while the 7XC versions add

such as Atmel's tightly-coupled memory architecture that permits users to directly connect external SRAM to the CPU with no latency concerns, as well as a prioritised interrupt controller and DMA subsystem. The chip also carries a USB host controller within a 217-pin ball-grid-array package for less than \$10 in production volumes.

Mike Olivarez, principal staff scientist at Freescale's wireless and mobile systems group, says that from its initial wireless communications background, Freescale is positioning its ARM core-based i.MX application processors ever more toward automotive, industrial, and medical applications: "Anything that has batteries, buttons, or displays is a perfect fit for the i.MX family." He notes that Freescale is an ARM architectural licensee and lead partner, which enables the company to make changes to the ARM architecture while maintaining complete software compatibility. To provide the compute horsepower that multimedia applications need, Freescale's i.MX family embodies ARM920, 926, and 1136 cores. Olivarez points to devices such as the new i.MX31 as enabling a range of ultra-portable computing devices: "Having a hard-disk controller on-chip as well as the video interfaces opens up non-traditional applications, such as embedded security monitoring." Key features couple the MMU that most OS demand with multimedia-specific hardware including a CCIR-656 video interface, MPEG4 hardware video acceleration, a 2- and 3-D graphics engine, and a versatile LCD controller. There's also an array of serial interfaces including wireless support with a SDIO (secure digital input/output) interface to support trusted-content strategies. An internal crossbar switch that overcomes bus-bandwidth limitations and a vectored-interrupt-controller to handle real-time events help build a fully parallel processing system, Olivarez says.

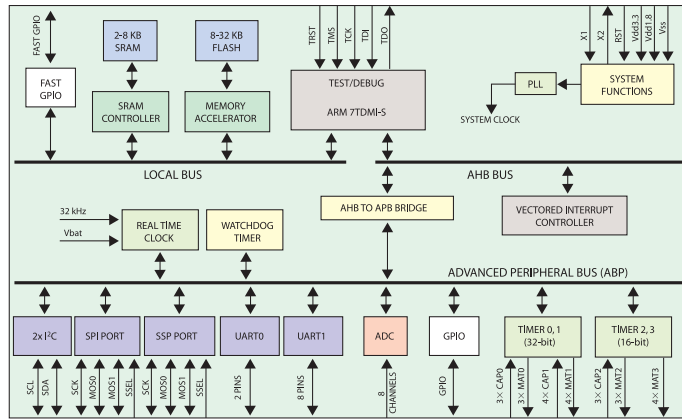
Over at its wireless and mobile systems group's developer relations team, Fred Stotz views Freescale's success in building third-party communities as enabling a range of toolchain support that customers find irresistible. To help proliferate i.MX processors, Stotz worked with partners to develop entry-level tools such as the i.MXL and i.MX21\_litekits that retail for a recommended \$499. These kits comprise a 6.35×3.8-cm standalone board developed by Cogent Computer Systems that carries an i.MXL or i.MX21 processor,

64 Mbytes of SDRAM, 8 Mbytes of Flash, and a USB port. You also get Cogent's expansion board that adds peripherals including a touchscreen LCD, Ethernet controller, audio CODEC, and USB host and device ports. Running on a Windows or Linux host via a TFTP (trivial-file-transfer-protocol) USB link, software support comes from the GNU-X toolchain and GX-Linux ports by Microcross that include Ethernet, serial,

and video-interface drivers as well as the normal utilities and libraries. The debugger is Visual GDB, a variant of the GNU debugger. Stotz adds that Linux board support packages are available from Freescale's website for these kits as well as for the ADS (application-development system) versions that feature the ubiquitous Metrowerks CodeWarrior IDE and retail for \$1,888.

Ata Khan, director of product innovation for microcontrollers at Philips Semiconductors, reflects that the company's involvement with ARM started about 10 years ago for cellphone applications. Today, the company employs ARM cores within its own products, and has publicly offered its ARM7-based LPC2000 family for about five years. The LPC2000 series now comprises some twenty variants that offer from 32 to 256 kbytes of Flash as well as external memory options; 8 to 64k SRAM; various timers that include watchdog and capture/compare facilities; a dedicated PWM unit; serial interfaces including CAN, I<sup>2</sup>C, and SPI; a multi-channel 10-bit ADC; and from 32 to 112 GPIO lines. Packages range from 28-pin PLCC to 144-pin LQFP. Growing application areas span low-end DSP and control to protocol converters and PMBus ports.

Announced last September, the latest family members are the LPC2101/2/3. With pricing set at just \$1.47/10,000 for the 8-kbyte Flash and 2-kbyte SRAM-equipped LPC2101, this ARM7TDMI machine sets a new low price point. The LPC2102 packs 16 kbytes of Flash and 4 kbytes of SRAM for \$1.85/10,000, while the LPC2103 provides 32 kbytes of Flash



**Figure 3** The LPC210x series from Philips sets a new low cost point for complex ARM7 microcontrollers.

and 8 kbytes of SRAM for \$2.20/10,000. The 7-mm<sup>2</sup> outline houses four timers with capture/compare for PWM support, a real-time clock and a watchdog timer, two 16C550-style UARTs, two I<sup>2</sup>C and two SPI ports, and an 8-channel 10-bit ADC (Figure 3). All of these resources reside on the advanced peripheral bus and communicate with the core via a bridge onto ARM's advanced high-performance bus. Notice too the vectored interrupt controller that augments the core's limited interrupt-handling hardware.

Khan points to the 32 fast 5V-tolerant GPIO ports that sit on the core's local bus: "As it's a RISC machine, there's no intrinsic read-modify-write capability, so we added atomic bit manipulation and moved the GPIO to the CPU bus. These measures ensure that bit manipulations complete in a single instruction cycle, so they are deterministic and non-interruptible—and capable of toggling at speeds up to 17.5 MHz." Khan also notes that the chip's 128-bit wide Flash provides some 280 Mbytes/sec of memory bandwidth that allows the core to run at its full 70 MHz, assisting deterministic behaviour when running from on-chip memory. This Flash architecture employs a two-transistor construction that's a trade-off between size and robustness and also facilitates building EEPROM. The Flash uses prefetch and branch buffers to virtually equal SRAM performance, and includes an 8-bit error-correction-code field that detects and corrects all single-bit errors, a feature that automotive applications increasingly demand. Four power-management modes progressively shut the chip down to its 5- $\mu$ A sleep mode.

Khan stresses the importance of the chip's debug abilities and his company's toolchain partners. Entry-level development kits are available from suppliers such as IAR and Keil starting at around \$99, with tools working up to the \$10,000 level for highly optimising compilers that support the very large code sets that, for example, a multi-user Linux development project demands. Because the embedded-trace-module takes 10 pins for

a full-speed implementation, Khan suggests starting off with a high-end LPC part that makes the module available to the outside world, then migrating to a part that optimally fits the end-user application. For hands-on development guidance, LPC designers can freely download a book written by Trevor Martin—an engineer at development specialist Hitex, whose website offers a variety of ARM development kits (Reference 3).

Having just announced USB peripherals for the LPC series, Khan advises that Ethernet will shortly be available. He also sees Philips moving on to ARM9 architectures and shrinking the process geometry to 90 nm within the next year: "Although 90 nm decreases part cost, the negative is a big rise in on-chip leakage currents. But if you constrain speed, you can also constrain leakage." For this reason, Khan expects the company to use different cell libraries to provide alternative power-versus-speed choices to complement its multilevel power-domain hardware: "To suit applications such as wireless sensor nodes, we'd really like to get sleep-mode power consumption down below 1 $\mu$ A."

With an ongoing partnership set up of more than 12 years, Sharp Microelectronics was ARM's third licensee and enjoys extensive experience in combining ARM cores with custom graphics engines. Gunter Wagschal, product-marketing manager for Sharp's BlueStreak products, points to a range that now spans ARM7TDMI, ARM720, and ARM9: "As an LCD manufacturer, we benefit from a scaleable range of computation engines to handle graphical interfaces

from black-and-white displays in white goods to complex man-machine interfaces in industrial automation.” For minimum cost, the ARM7TDMI derivatives employ a 16-bit memory interface and exclude the embedded-trace-module that Wagschal notes is only necessary during early development: “Companies such as Lauterbach offer very good trace tools that work out much less expensive over long production runs than including the ETM,” he says. The LH754xx family combines graphical controllers with VGA, XGA, and colour or black-and-white interfaces with 32-kbyte SRAM, an 8-channel 10-bit ADC, three 16-bit timers with capture/compare facilities, a universal serial interface, and a four-channel DMA controller. The chips run at up to 90 MHz from a 3.3-V supply, with derivatives furnishing up to 76 5V-tolerant I/O pins and a CAN controller within a 144-pin outline. There’s also a range of complementary ARM720T products that include MMUs and SDRAM interfaces to suit OS applications.

Sharp’s latest additions are the 250-MHz LH7A400 and its 266-MHz cousin, the LH7A404. Both devices employ the ARM9 core coupled with a colour XGA graphics engine to suit high-end graphical-user-interface applications, with 16 kbytes of both cache and instruction memory complementing 80 kbytes of SRAM. Other common features include an MMU, SDRAM interface, serial ports including SPI and USB capabilities, an infrared interface port, an interface that supports smart cards and memory cards, and the JTAG port. Wagschal observes that such applications almost invariably run a complex OS such as Linux to serve the application’s needs. He stresses Sharp’s commitment to customer support, which is crucial win and retain business: “We’ve done a huge amount of work internally and with third-parties to, for example, port Linux onto our high-end products. We’re now running the latest Linux kernel, version 2.6.12—but probably most importantly for the majority of our customers, we provide full support for our peripherals in our board support packages.” Usefully, Sharp’s driver support takes the form of C-source code rather than object code—and it’s freely available under the BlueStreak software library website, where engineers can access a range of resources including Metrowerks’ open-source Linux for the LH7A400.

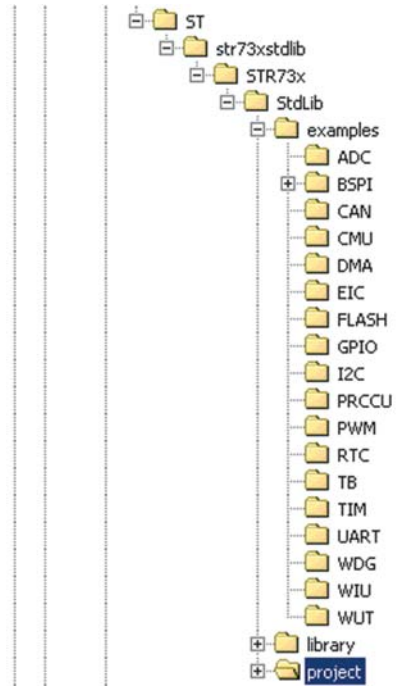


Figure 4 Free source-code device drivers speed STR730 development.

Sharp offers several reference designs for media players, with third-party development-board support coming from Logic Product Development.

Stephanie Ordan, product line manager for the STR730 series at STMicroelectronics, says that her company first became involved with ARM some five years ago with the intention of pursuing automotive applications: “From a platform viewpoint, the ARM architecture is upwardly compatible from ARM7 through the ARM9 and 11, which is a huge advantage for developers looking for a migration path within a familiar environment. We will be moving to ARM9 during 2006”. ST has recently taken advantage of its CAN-bus-focused ARM silicon from its STR710/720 series to tackle the industrial market. Released only this October, the company promotes its new STR730 range as the first ARM7 devices to tackle industry’s needs. Key features include a 5V power supply, –40 to +105°C operation, an enhanced interrupt controller, and full-speed execution from Flash to ensure deterministic response from the 36-MHz, 32-MIPS ARM7TDMI core.

The four-member STR730 range currently offers 64 to 256 kbytes of Flash with 16 kbytes of SRAM in packages that range from 100-pin TQFP to a 144-pin

## ARM7TDMI PRIMER

ARM's most popular product, the TDMI label signifies ARM7 features that comprise Thumb instruction, debugger, multiplier, and ICE (in-circuit emulator) support (Figure Aa). Notice the embedded realtime debug unit, a test-access-port (TAP) that supports multiple JTAG scan chains, and an interface for connection to ARM's embedded-trace-macrocell (ETM). These facilities enable a vast range of third-party development support

environments. The underlying architecture is ARM's version 4T, a 32-bit integer arithmetic processor that supports mixed-mode 16- and 32-bit instructions within a von Neumann-style load/store machine that processes register-based or immediate values, storing the result in a target register (Figure Ab). At the machine's heart lies a block of thirty-one general-purpose 32-bit registers and six status registers that connect to other hardware via five inde-

pendent busses—the programme counter (PC), arithmetic-logic-unit (ALU), increment, and the A and B busses—maximising parallel operation during instruction execution. Externally, the 32-bit unified bus interface carries both instructions and data to support 8-, 16-, and 32-bit data types within a linear 4-Gbyte addressing range. All I/O is memory-mapped into this space, possibly with external DMA hardware handling high-bandwidth channels.

The instruction-decode logic's three-stage pipeline feeds the ALU, which couples to a 32-bit barrel shifter and a multiplier that supports 32-bit multiplies and multiply-accumulate operations, yielding a 64-bit result. In normal programme flow, the pipeline implements an instruction-fetch, decode, and execution sequence on three independent data-processing instructions. The system completes one such instruction per input clock cycle with three-cycle latency:

**fetch**—control logic fetches an instruction from memory and places it into the pipeline.  
**decode**—decodes the instruction and generates datapath control signals for the execution phase.  
**execute**—the ALU processes the operands on the A and B busses and writes the result back to a register.

While most microcontrollers reserve conditional execution for branches, every ARM instruction

executes conditionally, depending on the state of the condition code field in the top four bits of the status register. Data-processing instructions require two operands that are either the contents of two registers, or that of one register and an immediate value. For example, the assembly-language statement `ADD r3, r1, r2` adds the contents of r1 on the A bus and r2 on the B bus and deposits the result in r3. The internal data paths are in use during every cycle, with the supervisory logic generating a two-phase non-overlapping clock to control operations and facilitate transfers. The operand on the B bus also passes through the barrel shifter, so a simultaneous shift operation is also possible. To avoid slowing the ALU, the shift logic employs a  $32 \times 32$ -crossbar switch matrix to steer inputs to the requisite outputs.

Data-transfer instructions complicate flow due to the need to calculate the target address prior to the data transfer, but the pipeline can still access memory on every cycle. Memory access time is a crucial performance issue that's especially relevant when executing from Flash, which typically has a maximum operating frequency of 30 to 35 MHz. The sequence that computes a target memory location for data storage mostly follows the data-processing model, combining a base address value from one register with another value

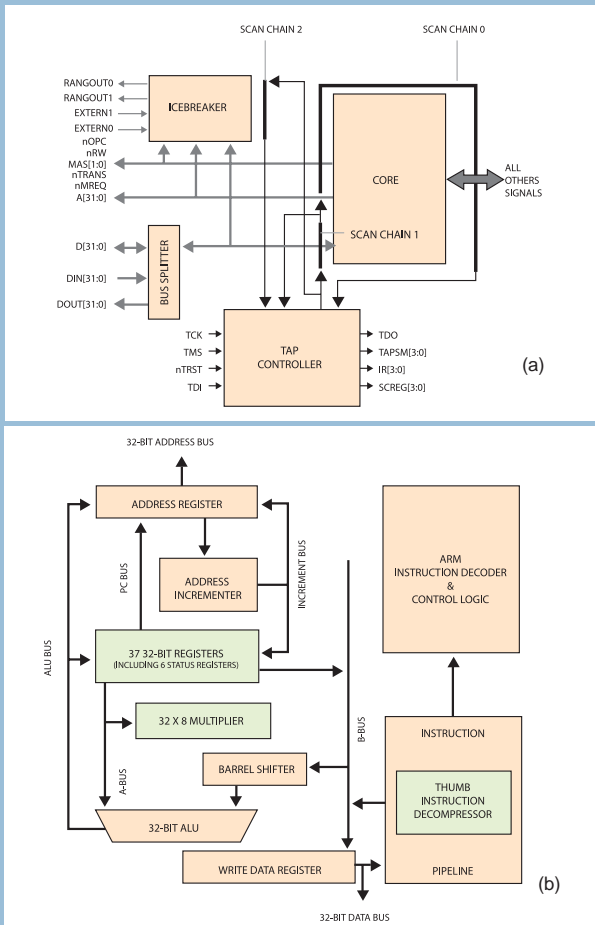


Figure A The ARM7TDMI's debug facilities include embedded in-circuit emulation (a). Its core includes a multiplier and barrel shifter built around a 37-entry register file (b).

that's either immediate or register-based. But the barrel shifter plays no part here, and the target is the address register. The address update occurs on the next cycle, during which the ALU can take advantage of an otherwise idle period to update the base register with an auto-indexed value, maximising datapath use. The new target address value is available at the start of the third cycle, so the address register effectively operates as a single-stage pipeline between the processor and external memory. Load instructions similarly require three cycles to access the data, move it into the data input register, and finally to the register file.

Branch instructions are a special case of target address computation that reads programme counter (PC) register 15 (r15)—that is, the programmer's view of the address register—and adds it to a 24-bit immediate-value field that's contained within the branch instruction.

The sequence again takes three clock cycles to complete. First, the barrel shifter moves the 24-bit value two positions to the left to generate a word-aligned offset that the ALU adds to the PC's value to form the instruction-fetch address. The address register outputs this value in the next cycle, during which control logic copies the return address—if required—to the link register, r14. Lastly, a third cycle refills the flushed pipeline and also adjusts the address pointer value in the link register to point to the instruction that immediately follows the branch. This adjustment compensates for r15 pointing at the PC's value +8—whereas the return address for the next instruction actually sits at PC +4—and results from the machine's 3-stage pipeline (in normal operation, the PC runs two steps ahead of the currently executing instruction). It's part of the compiler writer's job to

track r15's behaviour and to present the programmer with unambiguous PC information.

The register file that holds the processor's state information has one read and two write ports that permit access to any register, together with a read/write port to access PC register r15. This file forms the programmer's model, with the number of available registers depending on the chip's instruction set mode—32-bit ARM or 16-bit Thumb—and the processor's operating mode. Software commands, interrupts, and exception processing can all cause mode changes. For example, the processor's normal execution state is user mode, which provides direct access to fifteen general-purpose registers, the r15 PC register, and a status register. These same registers are also available in system mode, a privileged version of user mode that's reserved for use by an OS. The supervisor mode that furnishes

protected-mode operation also suits OS use.

Operations apart from user and system mode have associated banked registers that hold the return-from-subroutine address and a stack pointer. The fast-interrupt (FIQ) mode that supports rapid data transfers and channel processes has five banked registers to speed exception handling, while the normal interrupt request (IRQ) mechanism is a general-purpose interrupt handler. The processor enters abort mode if a memory access aborts, while the undefined mode results from attempting to execute an undefined instruction. These facilities allow graceful recovery from interrupts, traps, and supervisory calls/software-interrupts—together treated as exceptions—by storing state information, entering an appropriate exception handler via a vector table in low memory, and returning to the user programme when the exception handler's actions complete.

TQFP. A 144-ball BGA version shrinks the outline to just 10 mm<sup>2</sup>. Peripherals can include as many as twelve serial communications interfaces—three CAN 2.0B channels, two I<sup>2</sup>C ports, three SSI ports, and four UARTs—as well as a 16-level, 64-vector nested interrupt controller; ten 16-bit timers with capture/compare; six 16-bit PWM modules; three general-purpose 16-bit timers with 8-bit prescalers; sixteen channels of 3 μsec ADC; four four-channel DMA controllers; a watchdog timer and a real-time clock; and up to 112 I/O lines. The chip also implements a five-level power-minimisation strategy via two embedded voltage regulators. Toolchain support includes starter kits from IAR and Raisonance, as well as ST's own STR730-EVAL/WS evaluation board. Crucially,

developers can freely access source-code device drivers for all the STR730's peripherals from the company's website (Figure 4). There's even a free uClinux port that's designed for the STR710-EVAL board. The STR730 is sampling now, with prices that span \$4.53/10,000 for the CAN-less 100-pin, 64-kbyte Flash STR736FV0 to \$8.99/10,000 for the 256-kbyte STR730FZ2 that carries three CAN ports and 112 GPIOs within its 144-lead TQFP.

Speaking for Texas Instruments, Matthias Poppel, marketing manager for TI's advanced embedded control division, reflects that it doesn't make a lot of sense for TI Automotive to keep on developing their own cores when very capable industry-standard designs are readily available: "TI is ARM's biggest single customer, and

we have a very tightly coupled relationship." Acknowledging TI's leadership in DSPs, Poppel points to platforms such as TI's OMAP (open-multimedia-applications platform) that combines ARM cores including the 9 and 11 variants with DSPs from the TMS320C family. Principally intended for automotive use, the ARM7TDMI-based TMS470 series runs at the sub-100-MHz level rather than the headline-grabbing GHz rates of its DSP brethren. Poppel explains: "Automotive designers don't want cores running at very high speeds as this not only adds to the RF noise field—possibly disturbing audio and video signals—but could also potentially impact safety-critical subsystems."

Performance is nonetheless a big issue, so TI has developed efficient peripherals

that can operate for the most part autonomously. Examples include its high-end timer module that includes intelligence to free the CPU from interrupts, and an ADC module with internal buffers that minimise data movement. For development support, Poppel references TI's Code Composer suite for the TMS470, which comprises a 470-specific compiler and linker but maintains the same graphical user interface as other versions of this popular environment. The company is also developing a peripheral configuration tool that will allow users to set hardware without having to understand the registers that underpin each block—which, as anyone who has tried will know, can be a hugely time-consuming operation. For the future, Poppel confirms that TI will be developing silicon based on ARM's new Cortex architecture, probably sampling as soon as 2007.

## ARM SEALS NEW DEALS

As we were closing for press, Actel announced that it now supplies a version of the ARM7TDMI-S (the -S signifies the synthesis version) to suit its ProASIC3 Flash-based FPGAs. Dennis Kish, Actel's vice-president of marketing, said: "Up until now there hasn't been a synthesisable ARM solution in the FPGA space... people wanting to put a processor in an FPGA didn't have many choices—and especially not a popular choice like the ARM7, which is arguably the most used and well-known 32-bit core." While noting that FPGA design starts outnumber those of ASICs by about 40:1, Actel reckons that only around 15% of the annual 80,000 FPGA design starts embed a microcontroller.

The deal with ARM is significant because customers don't have to negotiate with the intellectual-property supplier, which always zealously guarded its IP—the subtext being that ARM trusts Actel's Flash devices to protect their contents. Further protection comes from the deliverable's firm core format rather than Verilog or VHDL code, an approach that also guarantees the core's performance. It appears to users as a black box during the synthesis phase, when it's possible to tweak subsystems including the bus interface, peripherals, and I/O. Actel's new Windows-based CoreConsole tool permits users to set the system specification, with hardware development via its Libero integrated development environment

and an Actel-specific version of ARM's RealView software tools. According to Kish, the core consumes about 6,000 ProASIC tiles or roughly 250,000 system gates. Actel will provide the core free to customers buying ARM-ready M7 ProASIC3 devices and expects to offer the core for other families. Its M7A3P250, M7A3PE600, and M7A3P1000 devices are sampling now.

Other breaking news is ARM's acquisition of toolmaker Keil, which has over 100,000 developers using its tools. ARM's Warren East said that his company identified the MCU market as a critical growth area for its future business: "As the MCU applications shift from 8/16-bit to 32-bit solutions, the combination of the ARM Cortex-M3 processor, which was specifically designed for microcontroller applications, our RealView high-performance compiler, and Keil's complementary MCU tools for ARM, will enable new generations of ARM MCU solutions." Notice here the reference to ARM's positioning the Cortex specifically for microcontroller makers. Haydn Povey, MCU product manager at ARM, expands: "Cortex represents version 7 of the ARM architecture and it's the first time that we haven't followed the incremental development path of its predecessors." Povey explains that the company took the decision to diverge from its traditional path as a result of customer inputs, specifically from the automotive and industrial areas.

Cortex variants include the A family such as the newly announced Cortex-A8 that targets mobile devices, and the M family that includes the Cortex-M3 that's intended for deeply embedded microcontroller applications. Common new features include the version 2 of the Thumb instruction set, which Povey asserts redresses the typical 25% performance loss from native 32-bit mode while still saving around 25–30% of Flash space. The M variant implements a highly deterministic system that includes its own nested vectored-interrupt-controller, which—because it is generic ARM and not a vendor-specific implementation—avoids vendor-to-ARM interrupt handovers, cumbersome assembly language routines, and compiler-specific code generation that complicates porting between different realtime OS.

Other key architectural differences include the M3's moving away from ARM7's multimode model with its

shadow registers in favour of a stack-based model. This helps shrink relative silicon size by over 30%. A new scheme permits bit manipulation via a 32-bit virtual word that aliases a single bit into memory or I/O, also saving memory space. Noting that the M3 is at least 50% more efficient in terms of work done per mW than the ARM7TDMI, Povey re-iterates that power consumption is now crucial almost everywhere: "Comparatively speaking, the M3 offers about twice the ARM7's performance and a quarter the power consumption, while occupying around one-third the silicon area and half the memory footprint." Watch out for more on 32-bit development tools and multiple-core chips within upcoming editions of EDNE.EDN

## REFERENCES

1. Patterson, D. A. and D. R. Ditzel, "The case for the reduced instruction set computer," ACM SIGARCH *Computer Architecture News*, Vol.8, No.6, October 15, 1980, pg 25-33.
2. Ref. 2. Manners, David, "ARM claims 3G mobile ownership," *Electronics Weekly*, July 28th 2005, [www.electronicsworld.com](http://www.electronicsworld.com).
3. Ref. 3. Martin, Trevor, "The insider's guide to the Philips ARM7-based microcontrollers," Hitex UK Ltd., Feb. 2005, ISBN 0954998804, [www.hitex.co.uk](http://www.hitex.co.uk).

## AUTHOR'S BIOGRAPHY

You can reach Contributing Technical Editor David Marsh at [forncett@btinternet.com](mailto:forncett@btinternet.com).

## FOR MORE INFORMATION

<b>Actel</b> <a href="http://www.actel.com">www.actel.com</a>	<b>Logic Product Development</b> <a href="http://www.logicpd.com">www.logicpd.com</a>
<b>Analog Devices</b> <a href="http://www.analog.com">www.analog.com</a>	<b>Microcross</b> <a href="http://www.microcross.com">www.microcross.com</a>
<b>ARM</b> <a href="http://www.arm.com">www.arm.com</a>	<b>Philips Semiconductors</b> <a href="http://www.semiconductors.philips.com">www.semiconductors.philips.com</a>
<b>Atmel</b> <a href="http://www.atmel.com">www.atmel.com</a>	<b>Raisonance</b> <a href="http://www.raisonance.com">www.raisonance.com</a>
<b>Freescal Semiconductor</b> <a href="http://www.freescal.com">www.freescal.com</a>	<b>Sharp Microelectronics</b> <a href="http://www.sharp-sme.com">www.sharp-sme.com</a>
<b>Green Hills Software</b> <a href="http://www.ghs.com">www.ghs.com</a>	<b>STMicroelectronics</b> <a href="http://www.st.com">www.st.com</a>
<b>IAR Systems</b> <a href="http://www.iar.com">www.iar.com</a>	<b>Texas Instruments</b> <a href="http://www.ti.com">www.ti.com</a>
<b>Keil Software</b> <a href="http://www.keil.com">www.keil.com</a>	
<b>Lauterbach</b> <a href="http://www.lauterbach.com">www.lauterbach.com</a>	