WHITE PAPER

An Introduction to the ARM Cortex-M3 Processor

Shyam Sadasivan **October 2006**

1. Introduction

System-on-chip solutions based on ARM embedded processors address many different market segments including enterprise applications, automotive systems, home networking and wireless technologies. The ARM Cortex™ family of processors provides a standard architecture to address the broad performance spectrum required by these diverse technologies. The ARM Cortex family includes processors based on the three distinct profiles of the ARMv7 architecture; the A profile for sophisticated, high-end applications running open and complex operating systems; the R profile for real-time systems; and the M profile optimized for cost-sensitive and microcontroller applications. The Cortex-M3 processor is the first ARM processor based on the ARMv7-M architecture and has been specifically designed to achieve high system performance in power- and cost-sensitive embedded applications, such as microcontrollers, automotive body systems, industrial control systems and wireless networking, while significantly simplifying programmability to make the ARM architecture an option for even the simplest applications.

1.1 Higher performance through better efficiency

In order to achieve higher performance, processors can either work hard or work smart. Pushing higher clock frequencies may increase performance but is also accompanied by higher power consumption and design complexity. On the other hand, higher compute efficiency at slower clock speeds results in simpler and lower power designs that can perform the same tasks. At the heart of the Cortex-M3 processor is an advanced 3-stage pipeline core, based on the Harvard architecture, that incorporates many new powerful features such as branch speculation, single cycle multiply and hardware divide to deliver an exceptional Dhrystone benchmark performance of 1.25 DMIPS/MHz. The Cortex-M3 processor also implements the new Thumb®-2 instruction set architecture, helping it to be 70% more efficient per MHz than an ARM7TDMI-S® processor executing Thumb instructions, and 35% more efficient than the ARM7TDMI-S processor executing ARM instructions, for the Dhrystone benchmark.

1.2 Ease of use for quick and efficient application development

Reducing time-to-market and lowering development costs are critical criteria in the choice of microcontrollers, and the ability to quickly and easily develop software is key to these requirements. The Cortex-M3 processor has been designed to be fast and easy to program, with the users not required to write any assembler code or have deep knowledge of the architecture to create simple applications. The processor has a simplified stack-based programmer's model which still maintains compatibility with the traditional ARM architecture but is analogous to the systems employed by legacy 8- and 16-bit architectures, making the transition to 32-bit easier. Additionally a hardware-based interrupt scheme means that writing interrupt service routines (handlers) becomes trivial, and that start-up code is now significantly simplified as no assembler code register manipulation is required.

Key new features in the underlying Thumb-2 Instruction Set Architecture (ISA) implement C code more naturally, with native bitfield manipulation, hardware division and If/Then instructions. Further, from a development perspective, Thumb-2 instructions speed up development and simplify long term maintenance and support of compiled objects through automatic optimization for both performance and code density, without the need for complex interworking between code compiled for ARM or Thumb modes. The effect of this is that users can maintain their code in C and not have to create libraries of pre-compiled object code, allowing for far greater code reuse.

1.3 Reduced costs and lower power for sensitive markets

A constant barrier to the adoption of higher performance microcontrollers has always been cost. Advanced manufacturing technologies are expensive and therefore smaller silicon area requirements can reduce costs significantly. The Cortex-M3 processor reduces system area by implementing the smallest ARM core to date, with just 33,000 gates in the central core (0.18um G) and by efficiently incorporating tightly coupled system components in the processor. Memory requirements are minimized by implementing unaligned data storage, atomic bit manipulation and the Thumb-2 instruction set that reduces instruction memory requirements for the Dhrystone benchmark by more than 25% compared to ARM instructions.

In order to address the increasing need for energy conservation in markets like white goods and wireless networking, the Cortex-M3 processor supports extensive clock gating and integrated sleep modes. Enabled by these features, the processor delivers a power consumption of just 4.5mW and a silicon footprint of 0.30mm^2 when implemented at a target frequency of 50MHz on the TSMC 0.13G process using ARM Metro™ standard cells.

1.4 Integrated debug and trace for faster time to market

Embedded systems typically have no graphical user interface making software debug a special challenge for programmers. In-circuit Emulator (ICE) units have traditionally been used as plug-in devices to provide a window into the system through a familiar PC interface. As systems get smaller and more complex, physically attaching such debug units is no longer a viable solution. The Cortex-M3 processor implements debug technology in the hardware itself with several integrated components that facilitate quicker debug with trace & profiling, breakpoints, watchpoints and code patching, significantly reducing time to market. Additionally, the processor provides a high level of visibility into the system through a traditional JTAG port or the 2-pin Serial Wire Debug (SWD) port that is suitable for devices in low pin-count packages.

1.5 Migration from the ARM7™ processor family for better performance and power efficiency

Over the last decade, the ARM7 family of processors has been widely adopted for many applications. The Cortex-M3 processor builds on this success to present the logical migration path for ARM7 processor-based systems. The central core offers higher efficiency; a simpler programming model and excellent deterministic interrupt behaviour, whilst the integrated peripherals offer enhanced performance at low cost.

Table 1. ARM7TDMI-S and Cortex-M3 comparison (100MHz frequency on TSMC 0.18G)

Features	ARM7TDMI-S	Cortex-M3
Architecture	ARMv4T (von Neumann)	ARMv7-M (Harvard)
ISA Support	Thumb / ARM	Thumb / Thumb-2
Pipeline	3-Stage	3-Stage + branch speculation
Interrupts	FIQ / IRQ	NMI + 1 to 240 Physical Interrupts
Interrupt Latency	24-42 Cycles	12 Cycles
Sleep Modes	None	Integrated
Memory Protection	None	8 region Memory Protection Unit
Dhrystone	0.95 DMIPS/MHz (ARM mode)	1.25 DMIPS/MHz
Power Consumption	0.28mW/MHz	0.19mW/MHz
Area	0.62mm2 (Core Only)	0.86mm2 (Core & Peripherals)*

* Does not include optional system peripherals (MPU & ETM) or integration level components

Figure 1. Relative performance for ARM7TDMI-S (ARM) and Cortex-M3 (Thumb-2)

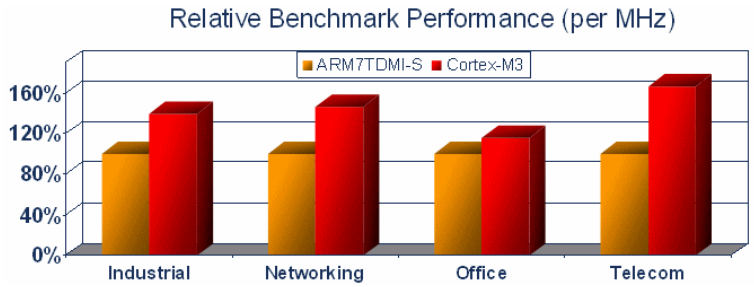
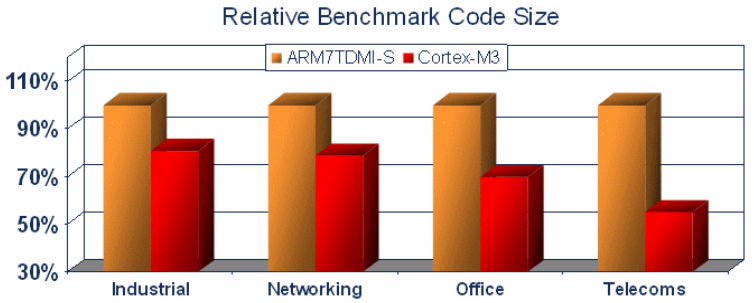


Figure 2. Relative code size for ARM7TDMI-S (ARM) and Cortex-M3 (Thumb-2)



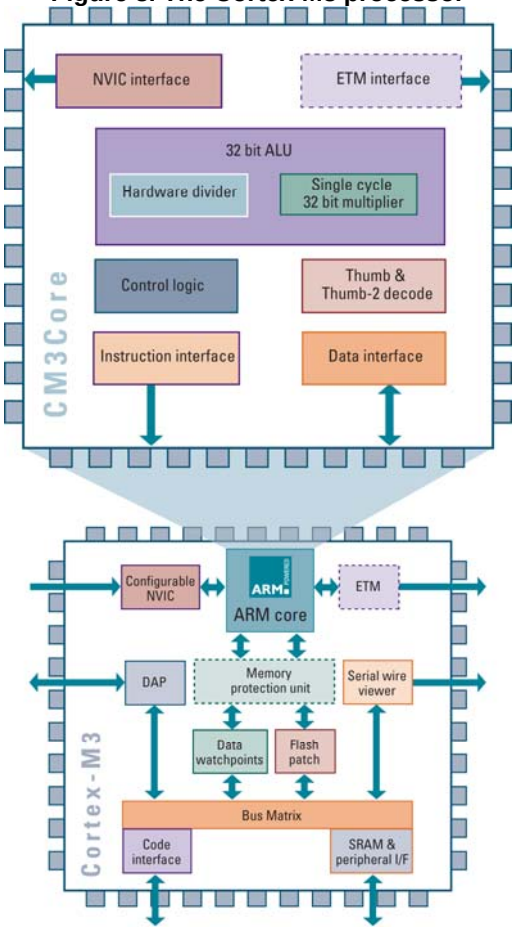
2. Cortex-M3 processor architecture and features

The Cortex-M3 processor, based on the ARMv7-M architecture, has a hierarchical structure. It integrates the central processor core, called the CM3Core, with advanced system peripherals to enable integrated capabilities like interrupt control, memory protection and system debug and trace. These peripherals are highly configurable to allow the Cortex-M3 processor to address a wide range of applications and be more closely aligned with the system requirements. The Cortex-M3 core and the integrated components (Figure 3) have been specifically designed to meet the requirements of minimal memory implementation, reduced pin count and low power consumption.

2.1 The Cortex-M3 Core

The central Cortex-M3 core is based on the Harvard architecture characterized by separate buses for instructions and data (Figure 3). The processor differs from the von Neumann architecture based ARM7 family of processors which use the same signal buses and memory for both instructions and data. By being able to read both an instruction and data from memory at the same time, the Cortex-M3 processor can perform many operations in parallel, speeding application execution.

Figure 3. The Cortex-M3 processor



The core pipeline has 3 stages: Instruction Fetch, Instruction Decode and Instruction Execute. When a branch instruction is encountered, the decode stage also includes a speculative instruction fetch that could lead to faster execution. The processor fetches the branch destination instruction during the decode stage itself. Later, during the execute stage, the branch is resolved and it is known which instruction is to be executed next. If the branch is not to be taken, the next sequential instruction is already available. If the branch is to be taken, the branch instruction is made available at the same time as the decision is made, restricting idle time to just one cycle.

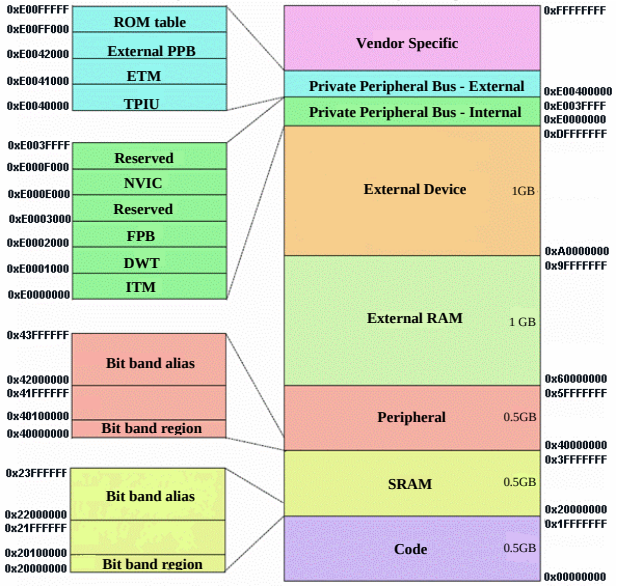
The Cortex-M3 core contains a decoder for traditional Thumb and new Thumb-2 instructions, an advanced ALU with support for hardware multiply and divide, control logic, and interfaces to the other components of the processor.

The Cortex-M3 processor is a 32-bit processor, with a 32-bit wide data path, register bank and memory interface. There are 13 general-purpose registers, two stack pointers, a link register, a program counter and a number of special registers including a program status register.

The Cortex-M3 processor supports two operating modes, Thread and Handler and two levels of access for the code, privileged and unprivileged, enabling the implementation of complex and open systems without sacrificing the security of the application. Unprivileged code execution limits or excludes access to some resources like certain instructions and specific memory locations. The Thread mode is the typical operating mode and supports both privileged and unprivileged code. The Handler mode is entered when an exception occurs and all code is privileged during this mode. In addition, all operation is categorized under two operating states, Thumb for normal execution and Debug for debug activities.

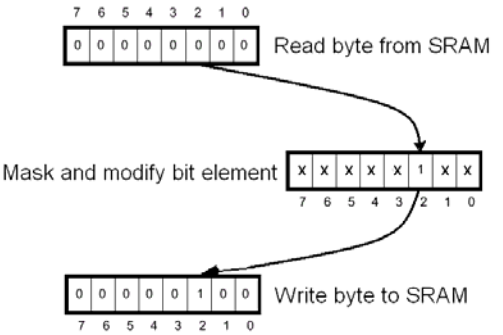
The Cortex-M3 processor is a memory mapped system with a simple, fixed memory map for up to 4 gigabytes of addressable memory space with predefined, dedicated addresses for code (code space), SRAM(memory space), external memories/devices and internal/external peripherals. There is also a special region to provide for vendor specific addressability.

Figure 4. The memory map



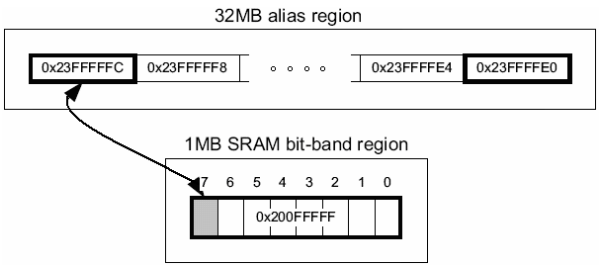
The Cortex-M3 processor enables direct access to single bits of data in simple systems by implementing a technique called bit-banding (Figure 5). The memory map includes two 1MB bit-band regions in the SRAM and peripheral space that map on to 32MB of alias regions. Load/store operations on an address in the alias region directly get translated to an operation on the bit aliased by that address. Writing to an address in the alias region with the least-significant bit set writes a 1 to the bit-band bit and writing with the least-significant bit cleared writes a 0 to the bit. Reading the aliased address directly returns the value in the appropriate bit-band bit. Additionally, this operation is atomic and cannot be interrupted by other bus activities.

Figure 5. Comparison of traditional bit manipulation with Cortex-M3 bit-banding




```
LDR R0,=0x200FFFFF ; Setup address
MOV R2, #0x4       ; Setup data
LDR R1, [R0]       ; Read
ORR R1, R2         ; Modify bit
STR R1, [R0]       ; Write back result
```

Traditional bit manipulation method




```
LDR R0,=0x23FFFFFC ; Setup address
MOV R1, #0x1       ; Setup data
STR R1, [R0]       ; Write
```

Direct, single cycle access with bit banding

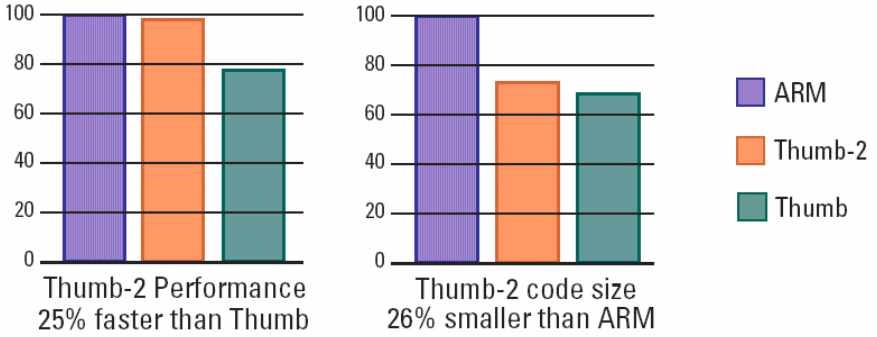
Traditional ARM7 processor-based systems support only aligned data access, allowing data to be stored and accessed only along aligned word boundaries. The Cortex-M3 processor implements unaligned data access that enables unaligned data transfers in a single core access. When unaligned transfers are used, they are converted into multiple aligned transfers and remain transparent to application programmers.

In addition the Cortex-M3 processor supports 32-bit multiply operations in a single cycle and also supports signed and unsigned divide operations with the SDIV and UDIV instructions that take between 2 and 12 cycles depending upon the size of the operands. The division operation is completed faster if the dividend and the divisor are closer in size. These improvements in the mathematical capabilities make the Cortex-M3 processor ideal for many numerically intensive applications such as sensor reading and scaling or hardware-in-the-loop simulation systems.

2.2 Thumb-2 Instruction Set Architecture

The ARMv7-M is the microcontroller profile of the ARMV7 architecture and is different from earlier ARM architectures in that it supports Thumb-2 instructions alone. Thumb-2 technology is a blend of 16 and 32-bit instructions that delivers the performance of 32-bit ARM instructions, matches the code density of and is backwards compatible with, the original 16-bit Thumb instruction set. Figure 6 shows indicative Dhrystone benchmark results that illustrate that Thumb-2 technology indeed achieves this objective.

Figure 6. Relative Dhrystone performance and code size for ARM, Thumb and Thumb-2



In an ARM7 processor-based system, switching the processor core between the Thumb state for code density and ARM state for high performance would be necessary for certain applications. However, the Cortex-M3 processor does not need to interwork instructions since both 16-bit and 32-bit instructions co-exist in the same mode, enabling higher code density and performance with far less complexity. As theThumb-2 instructions are a superset of 16-bit Thumb instructions, the Cortex-M3 processor can execute any previously written Thumb code. By implementing Thumb-2 instructions, the Cortex-M3 processor also delivers compatibility with other members of the ARM Cortex processor family.

The Thumb-2 instruction set includes instructions that make it easier to write compact code for many different applications. The BFI and BFC instructions are bit-field instructions that are useful in applications like network packet processing. The SBFX and UBFX instructions improve the ability to insert or extract a number of bits to or from a register, a capability particularly useful in automotive applications. The RBIT instruction reverses the bits in a word and is useful in DSP algorithms such as DFT. The table branch instructions TBB and TBH enable a balance of code compaction and high performance. Thumb-2 instructions also introduce a new If-Then construct that predicates the conditional execution of up to four subsequent instructions.

2.3 The Nested Vectored Interrupt Controller (NVIC)

The highly configurable NVIC is an integral part of the Cortex-M3 processor and provides the processor's outstanding interrupt handling abilities. In its standard implementation it supplies a Non-Maskable Interrupt (NMI) and 32 general purpose physical interrupts with 8 levels of pre-emption priority. It can be configured to anywhere between 1 and 240 physical interrupts with up to 256 levels of priority though simple synthesis choices.

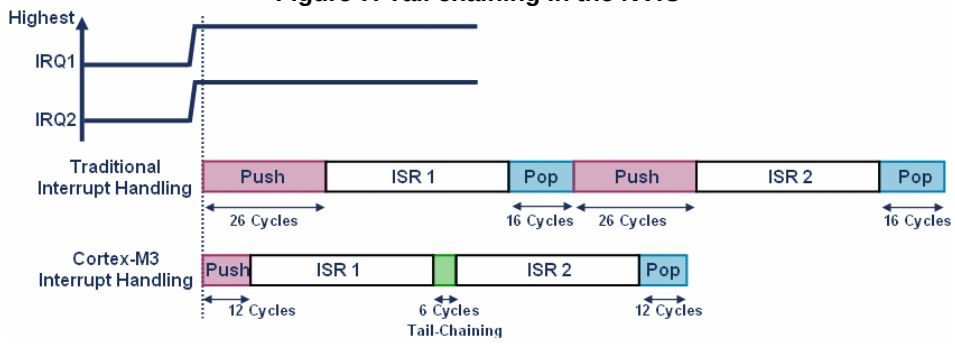
The Cortex-M3 processor uses a re-locatable vector table that contains the address of the function to be executed for a particular interrupt handler. On accepting an interrupt, the processor fetches the address from the vector table through the instruction bus interface. The vector table is located at address zero at reset, but can be relocated by programming a control register.

To reduce gate count and enhance system flexibility the Cortex-M3 has migrated from the banked shadow register exception model of the ARM7 processor to a stack based exception model. When an exception takes place, the Program Counter, Program Status Register, Link Register and the R0-R3,R12 general purpose registers are pushed on to the stack. The data bus stacks the registers whilst the instruction bus identifies the exception vector from the vector table and fetches the first instruction of the exception code. Once the stacking and instruction fetch are completed, the interrupt service routine or fault handler is executed, followed by the automatic restoration of the registers to enable the interrupted program to resume normal execution. By handling the stack operations in hardware, the Cortex-M3 processor removes the need to write assembler wrappers that are required to perform stack manipulation for traditional C-based interrupt service routines, making application development significantly easier.

The NVIC supports nesting (stacking) of interrupts, allowing an interrupt to be serviced earlier by exerting higher priority. It also supports dynamic reprioritisation of interrupts. Priority levels can be changed by software during run time. Interrupts that are being serviced are blocked from further activation until the interrupt service routine is completed, so their priority can be changed without risk of accidental re-entry.

In the case of back-to-back interrupts, traditional systems would repeat the complete state save and restore cycle twice, resulting in higher latency. The Cortex-M3 processor simplifies moving between active and pending interrupts by implementing tail-chaining technology in the NVIC hardware. Tail-chaining achieves much lower latency by replacing serial stack pop and push actions that normally take over 30 clock cycles with a simple 6 cycle instruction fetch. The processor state is automatically saved on interrupt entry, and restored on interrupt exit, in fewer cycles than a software implementation, significantly enhancing performance in sub-100MHz systems.

Figure 7. Tail chaining in the NVIC



The NVIC also implements the power-management scheme of the Cortex-M3 processor that supports integrated sleep modes. The Sleep Now mode is invoked by either the Wait For Interrupt (WFI) or the Wait For Event (WFE) instructions that immediately puts the core into low-power state pending an exception. The Sleep On Exit mode puts the system into low-power mode as soon as it exits the lowest priority interrupt-service routine. The core stays in sleep state until another exception is encountered. Since only an interrupt can exit this mode, the system state is not restored. The SLEEPDEEP bit of the system control register, if set; can be used to clock gate the core and other system components for optimal power savings.

The NVIC also integrates a System Tick (SysTick) timer, which is a 24-bit count-down timer that can be used to generate interrupts at regular time intervals, proving an ideal heartbeat to drive a Real Time OS or other scheduled tasks.

2.4 The Memory Protection Unit (MPU)

The MPU is an optional component of the Cortex-M3 processor that can improve the reliability of an embedded system by protecting critical data used by the operating system from user applications, separating processing tasks by disallowing access to each other's data, disabling access to memory regions, allowing memory regions to be defined as read-only and detecting unexpected memory accesses that could potentially break the system.

The MPU enables the application to be broken down into a set of processes. Each process owns specific memory (code, data, stack, and heap) and devices, as well as having access to shared memory and devices. The MPU also enforces user and privilege access rules. This includes executing code from the right privilege level as well as enforcing ownership of memory and devices by privileged and user code.

The MPU separates the memory into distinct regions and implements protection by preventing disallowed accesses. The MPU supports up to 8 regions each of which can be divided into 8 sub-regions. The region sizes supported start from 32 bytes and increase by factors of 2 to all of the addressable memory of 4 gigabytes. Each region is associated with a region number that is an index starting at 0, used to address the region. It is also possible to define a default background memory map for privileged accesses. Accesses to memory locations that are not defined in the MPU regions, or not permitted by the region setting, will cause the Memory Management Fault exception to take place.

The protection for the regions is implemented with rules that are based on the type of transaction (read, write or execute) and privilege of code performing the access. Each region includes a set of bits which affect what kinds of accesses are permitted, as well as bits that affect what kind of bus action is allowed. The MPU also supports overlapping regions, which are regions that cover the same address. Since sizes are in multiples of 2, overlap means that one may be fully enclosed within another. It is therefore possible to have multiple regions enclosed by a single region and it is also possible to have nested overlapping. In the case of address lookups to locations within overlapping regions, the region with the highest region number is returned.

2.5 Debug and Trace

The debug access into a Cortex-M3 processor based system is through the Debug Access Port (DAP) that can be implemented as either a Serial Wire Debug Port (SW-DP) for a two-pin (clock and data) Interface or a Serial Wire JTAG Debug Port (SWJ-DP) that enables either JTAG or SW protocol to be used. The SWJ-DP defaults to JTAG mode on power reset and can be made to switch protocols with a specific control sequence provided by the external debug hardware.

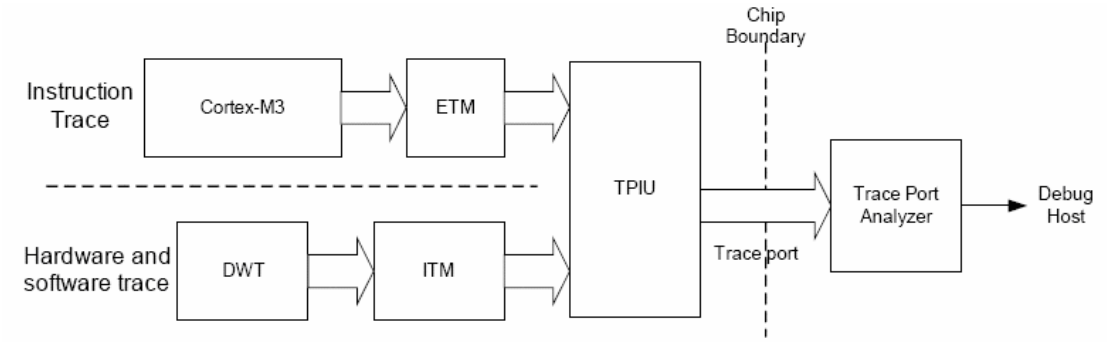
Debug actions can be triggered by various events like breakpoints, watchpoints, fault conditions, or external debug requests. When a debug event takes place, the Cortex-M3 processor can either enter the halt mode or the debug monitor mode. During the halt mode, the processor stops program execution completely. The halt mode supports single step operations. Interrupts can be pended, and can get invoked during single stepping, or be masked so external interrupts get ignored during stepping. During the debug monitor mode, the processor executes an exception handler to carry out the debug tasks, while still allowing higher priority exceptions to take place. This mode also supports single stepping.

The Flash Patch and Breakpoint (FPB) unit implements six program breakpoints and two literal data fetch breakpoints, or alternatively patches instruction or literal data from code memory space to system memory space. The unit contains six instruction comparators for matching against instruction fetches from code space. Each comparator can be enabled to either remap code to an area in system space, or implement a hardware breakpoint, by returning a breakpoint instruction to the processor. It also contains two literal comparators, for matching against literal loads from code space and remapping to an area in system space.

The Data Watchpoint and Trace (DWT) unit contains four comparators, each of which can be configured as hardware watchpoints. When used in this configuration, the comparators can be programmed to compare either the data address or program counter. The DWT comparators can also be configured to trigger PC sampler events, data address sampler events and to cause the Embedded Trace Macrocell (ETM) to emit trigger packets in the instruction trace stream

The ETM is an optional component designed to support instruction trace alone to ensure that the reconstruction of program execution is possible with minimal area impact. The ETM enables high-performance real-time trace of instruction execution and data transfers by compressing trace information from the processor core to minimize bandwidth requirements.

Figure 8. The Cortex-M3 Trace System



The Cortex-M3 processor implements data trace with the DWT and the Instrumentation Trace Macrocell (ITM). The DWT provides instruction execution statistics and can generate watchpoint events to invoke debug or trigger the ETM on specific system events. The ITM is an application driven trace source that supports printf style debugging to trace OS and application events. It accepts hardware trace packets from the DWT and software trace stimuli from the processor core and emits diagnostic system information with timestamps. The Trace Port Interface Unit (TPIU) accepts tracing information from the ETM and ITM; and then merges, formats and emits it via the Serial Wire Viewer (SWV) to external trace analyzer units. The SWV allows simple and cost effective profiling of system events by exporting streams of data through a single pin. Manchester encoded and UART are supported formats for the SWV.

2.6 The bus matrix and interfaces

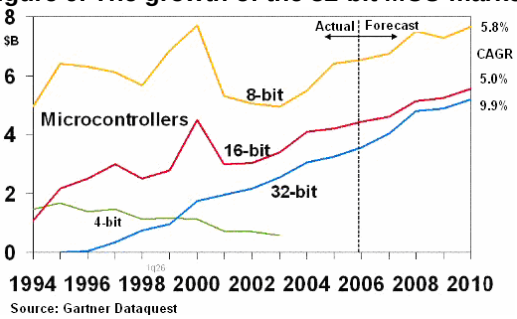
The Cortex-M3 processor bus matrix connects the processor and debug interface to the external buses; the 32-bit AMBA® AHB-Lite based ICode, DCode and System interfaces and the 32-bit AMBA APB™ based Private Peripheral Bus (PPB). The bus matrix also implements unaligned data accesses and bit banding.

The 32-bit ICode interface fetches instructions from the code space and can be accessed only by the CM3Core. All fetches are one word wide, with the number of instructions fetched per word depending upon the type of code implemented and its alignment in memory. The 32-bit DCode interface accesses data from the code memory space and can be accessed by the CM3Core and the DAP. The 32-bit System interface fetches instructions and accesses data within the system memory space and like the DCode bus, can be accessed by the CM3Core and the DAP. The PPB enables access to components outside of the Cortex-M3 processor system.

3. 32-bit performance at 8-bit cost for next generation MCUs

A microcontroller unit (MCU) is a highly integrated single chip that encapsulates a processor, non-volatile memory for programs, volatile memory for data, I/O ports, timers and typically multiple peripherals. These MCUs target an enormous range of applications ranging from low-end general purpose applications to high-performance digital signal control. The biggest challenge in this extremely competitive market is to keep costs low, to enable wide market adoption, while delivering a product with a wide enough array of performance and peripherals to satisfy a broad set of users.

Figure 9. The growth of the 32-bit MCU market



The MCU market has long been dominated by 8 and 16 bit architectures that offer very low cost solutions; for example 8-bit MCUs are available for as little as 40 cents. As next-generation applications demand higher performance and increased functionality, the market is accelerating the migration to 32-bit architectures that now offer significantly higher performance and greater ease-of-use at comparable system costs.

The processor at the heart of the device is in many cases one of the most important product differentiating factors for MCUs since many implement industry standard interfaces and peripherals. The ARM7 and ARM9™ processor families have had immense success in MCUs and with the introduction of the Cortex-M3 processor there are many more compelling reasons to make the switch to the ARM 32-bit architecture.

The need for greater performance is one of the main motivations to move to a 32-bit architecture, but performance is a very relative term. High-end media processing, for example, requires very high system clock frequencies to meet challenging bandwidth requirements. Performance in a MCU, on the other hand, is gauged through features like fast instruction execution from flash memory, efficient bit manipulation, inexpensive debug and trace technology, and very importantly, a robust interrupt structure vital for real-time applications. The Cortex-M3 processor offers a low cost, high performance solution to MCU designers, with a combination of architectural features and integrated peripheral components that simplify development and improve time to market.

3.1 Efficient memory usage for lower costs

Since memory dominates the bill of materials in virtually all embedded designs, efficient memory usage for both code and data is essential to reduce costs. Thumb-2 technology and the bus interfaces in the Cortex-M3 processor allow for higher code density and efficient instruction fetches while bit handling techniques enable better data manipulation.

Thumb-2 instructions include a mix of the original 16-bit Thumb instructions and new 32-bit instructions that provide equivalent functionality to the 32-Bit ARM instruction set. These instructions are fetched by the ICode interface in the Cortex-M3 processor that performs word-wide instruction fetches. The 16-bit instructions are halfword aligned in memory and are fetched two at a time. The 32-bit instructions, if word aligned, are completely fetched in a single cycle. If they are halfword aligned, they are fetched with a maximum latency of two cycles, presuming zero wait state memory. This allows for flexible usage of instructions with low instruction fetch latency.

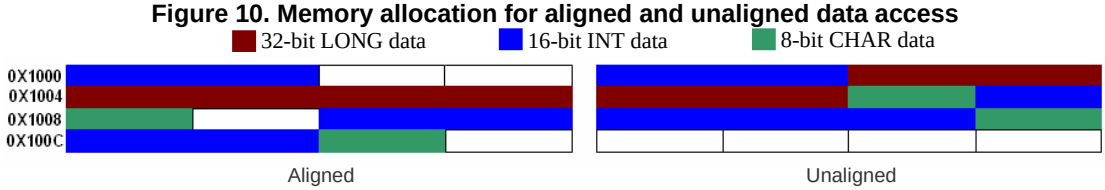
Embedded flash memory is a popular technology in many embedded applications. However, there has been a growing gap between flash and processor speeds, with flash typically operating at around 30-40MHz. In order to accommodate this gap, systems need to enable highly efficient

instruction and data retrieval while accessing flash. Flash memory interfaces typically fetch more than one 16-bit value at a time either immediately with the instruction or through a literal pool fetch. The MOVW and MOVT instructions embed 16-bit constants more effectively; by using these instructions, the data is available immediately, with no need to execute a load operation as required by literal pools.

To enable efficient use of available on-chip RAM, many applications use semaphores (i.e. single-bits of data) packed into a byte. Traditional methods for atomic manipulation of semaphores read the byte, mask/modify the bit and then write back the byte. The Cortex-M3 processor simplifies this task by using bit-banding. Up to 1MB of the memory address region is aliased to 32MB of bit-specific locations. If a read or write is performed on an address within the 32MB region, it automatically performs a bit-set or bit-clear on the aliased address. This scheme simplifies a multiple operation task into a single core write operation.

Another technique implemented by the Cortex-M3 processor that provides for efficient SRAM utilization is unaligned data access. Figure 10 shows the memory allocation for aligned and unaligned data access. Each box represents a byte of data with various data types like Long, Char and Integer requiring different byte lengths. The white boxes represent empty, unused memory spaces. By packing together the unused memory into a continuous space, the amount of SRAM required by the application is potentially reduced.

Figure 10. Memory allocation for aligned and unaligned data access



3.2 Low-cost debug and trace

Traditional system debug is performed through the popular JTAG interface that typically requires 5 pins; this can be a deterrent to low-cost systems with as few as 10 pins in total. The Cortex-M3 processor implements debug through the Debug Access Port(DAP), which in addition to JTAG, supports Serial Wire Debug technology that enables debug through just 2 pins while maintaining the same performance as JTAG. By including the optional ETM in the system, designers can avail of excellent instruction trace capabilities with minimal cost impact.

For mass volume ROM-only microcontrollers the Flash Patch technology allows debug actions to be performed on read-only code through independent remapping of ROM code and data to a region of SRAM. This offers device and system engineers the ability to patch errors in code at run-time, potentially eliminating the need for costly respins.

3.3 Low latency interrupt handling scheme

Typical MCU systems are interrupt intensive but do not have the luxury to include expensive, intelligent peripherals to service them independently. In such a scenario, the processor itself needs to sacrifice performance to service the interrupts with an efficient interrupt response and handling scheme. The integrated NVIC in the Cortex-M3 processor implements the interrupt handling in hardware and facilitates exceptionally low latency so that there is minimal impact on the processor performance. The tight integration of the NVIC with the processing core enables faster execution of Interrupt Service Routines (ISR), reducing the number of cycles typically taken to enter an interrupt by up to 70%. This is accomplished through the use of hardware stacking of registers, plus the ability to exit and restart load-store multiple executions.

3.4 Delivering in the field

The ARM architecture has become the de-facto standard for 32-bit MCUs, and is often called the 8051 of the next-generation. The Cortex-M3 processor, with its outstanding balance of high performance and low cost, often comparable to high-end 8-bit MCUs, is now extending the range of applications further enabling more users to migrate to 32-bit systems. Initial microcontroller devices from lead partners already combine the Cortex-M3 processor with flash, SRAM and multiple peripherals to provide a very competitive offering at the price of just $1.

4. Reliable & secure automotive and industrial control

Automotive systems of today are extremely complex, with over a hundred microcontrollers in some high end automobiles. These systems are characterised by a wide range of applications from high performance navigation systems to cost sensitive functions like electric window controls and oil pressure sensors. Reliable performance is paramount to such highly complex interoperable systems. Industrial control processes, in addition to reliability, need to consider safety issues for critical control operations. The Cortex-M3 processor has been designed to specifically address these requirements.

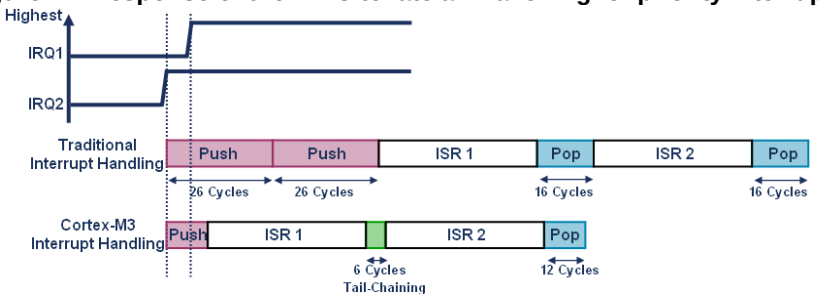
4.1 Deterministic interrupt handling for predictable response in automobiles

Automotive systems are highly interrupt driven, with routine actions like applying brakes and taking sharp turns being instances of interrupts. Fast and predictable response to such interrupts for the implementation of technology like anti-lock braking systems (ABS) and automatic stability control is critical for safe operation. These systems are especially complicated since many different such sub-systems could deploy at any time.

Tail-chaining technology in the NVIC supports interrupts that occur back-to-back, but there could be cases where an interrupt of higher priority could also occur during the stacking (Push) or state restore (Pop) stages of the interrupt being serviced. In traditional ARM7 processor based systems, these stages need to complete before the pending interrupt can take over. The Cortex-M3 NVIC, on the other hand, provides deterministic response to these possibilities with support for late arrival and pre-emption.

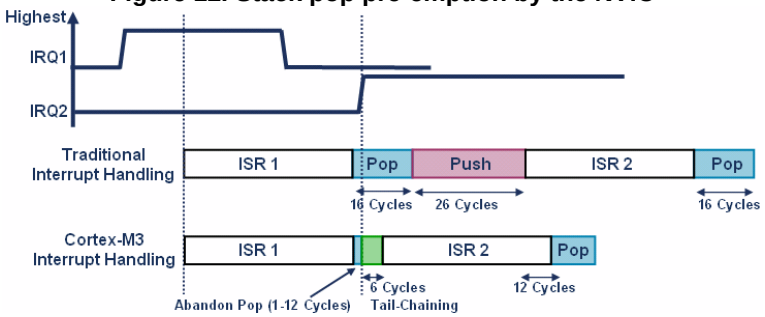
In case of the late arrival of a higher priority interrupt during the execution of the stack Push for a previous interrupt, the NVIC immediately fetches a new vector address to service the pending interrupt, as shown in Figure 11.

Figure 11. Response of the NVIC to late arrival of higher priority interrupts



Similarly, the NVIC abandons a stack Pop if an exception arrives and services the new interrupt immediately. By pre-empting and switching to the second interrupt without completing the state restore and save, the NVIC achieves lower latency in a deterministic manner.

Figure 12. Stack pop pre-emption by the NVIC



The Non-Maskable Interrupt (NMI) further enhances determinism by allowing the critical interrupt source to be made non-maskable; a particularly important feature in systems with a watchdog timer that needs to be reliably serviced at particular time intervals.

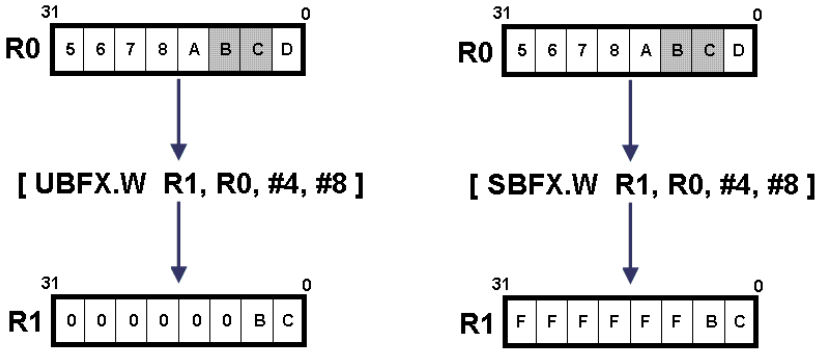
4.2 Fine grain memory protection for reliable software integration

The re-use of application software, based on automotive standard API is a very important requirement for next-generation automotive systems. In order to enable quicker time-to-market, automotive system designers want to be able to select software models from a variety of vendors and re-use it for various vehicles. In order to enable this, it is essential to have a mechanism to allow each module to be isolated so the risk of interference is minimal. The MPU in the Cortex-M3 processor provides regions with a fine granularity of as little as 32 bytes for efficient segregation of individual tasks. This allows the system to separate many different small software routines that would otherwise have to share protection schemes.

4.3 Faster bit extraction for efficient I/O data processing

Automotive and industrial applications are characterized by the need to process large volumes of general-purpose I/O data. Interfaces to sensors or similar external devices are often read in as 8 or 16-bit words, which then have to be manipulated to extract bit-level data. These manipulations are typically done by micro-code that takes several cycles to complete. The Thumb-2 UBFX (zero extend) and SBFX (sign extend) bit-field instructions enable this bit level extraction in a single cycle.

Figure 13. Bit extraction with the UBFX and SBFX instructions



4.4 Secure operation for a safer industrial environment

Industrial processes typically employ real time operating systems to implement control in a protected environment. There could be a number of different applications running simultaneously and a protection scheme is necessary to determine security levels for the applications.

The MPU enforces privilege and access rules for safe operation. It implements security privilege levels by separating code, data and stack on a task-by-task basis and also implements access restrictions by controlling what memory locations can be read, written, and executed. It can also detect unexpected, harmful memory access events like stack corruption, and allow safe and clean handling of these issues to avoid catastrophic failure.

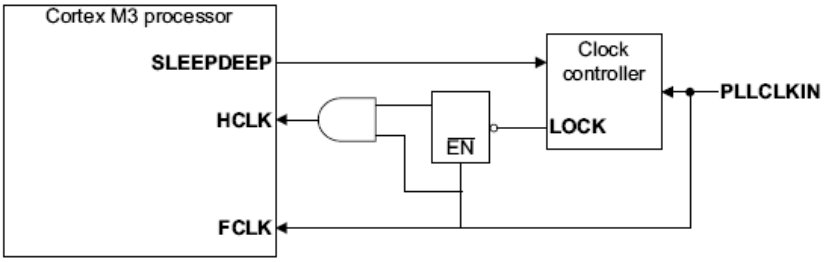
5. Lower power consumption for wireless networks

Wireless networks are ubiquitous in today's increasingly connected world. Various wireless standards like Wi-Fi, WiMAX, Bluetooth and Zigbee drive the technology that enables connectivity over the internet and with other similarly enabled devices. The common factor between all of these technologies is the increasing need for lower power consumption. For example, ZigBee based wireless devices are expected to last more than ten years when powered by a pair of AA batteries.

5.1 Clock gating and integrated sleep modes for lower power

Clock gating is a technique that shuts down the power to parts of the system, effectively disabling them. The Cortex-M3 processor extensively uses gated clocks to disable unused functionality, and disables inputs to unused functional blocks, so that only actively used logic consumes any dynamic power.

Figure 14. Power control in Deep Sleep mode



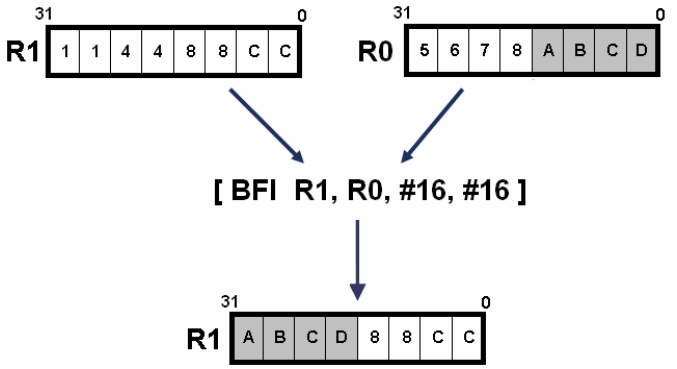
In addition to the broad implementation of clock gating in the processor design the Cortex-M3 processor also delivers a number of sleep modes to put the processor into low power statues, effectively clock gating the entire processor with the exception of sections of the interrupt controller. The sleep modes can be implemented through the WFI (wait For Interrupt) and WFE (Wait for Event) instructions. Additionally the Cortex-M3 processor enables wider system level power reduction through the use of the SLEEPDEEP signal. This signal is asserted when in Sleep-now or Sleep-on-exit modes when the SLEEPDEEP bit of the System Control Register is set. This signal is routed to the clock manager and can be used to gate the processor and system components including the Phase Locked Loop (PLL) to achieve greater power savings. Figure 14 shows an example of how to reduce power consumption by stopping the clock controller with SLEEPDEEP in the low-power state. When exiting low-power state, the LOCK signal indicates that the PLL is stable, and it is safe to enable the Cortex-M3 clock, ensuring that the processor is not re-started until the clocks are stable. To detect interrupts, the processor must receive the free-running FCLK in the low-power state.

5.2 Working smarter to sleep longer

Typically, higher power consumption is due to the processor working at faster clock speeds to meet challenging performance requirements. The processor can also choose to lower power consumption by working smarter at lower clock frequencies. The Cortex-M3 processor works at an outstanding efficiency of 1.25 DMIPS/MHz, enabling the system to complete compute intensive tasks faster and hence allowing it to spend more time in low power sleep modes.

Bit manipulation techniques are essential for efficient packet processing tasks in networking applications. For example, bit-field modification instructions provide low level access to packed data structures, such as those found in network packet headers. The bit-manipulation instructions in the Cortex-M3 processor like BFI and BFC reduce data-bus activity and the need to unpack data structures, thus speeding up network processing.

Figure 15. Insertion of any number of adjacent bits using BFI



5.3 Real power savings

The Cortex-M3 processor, configured without its optional components, delivers a power consumption of just 0.24mW/MHz, when implemented for a target frequency of 100MHz on the TSMC 0.18G process with ARM SAGE-X standard cell libraries. Not only does the Cortex-M3 processor reduce power consumption by over 30% as compared to the ARM7TDMI-S processor running at the same frequency, it achieves twice the performance.

Table 2. Performance and power comparison for 100MHz implementation on TSMC 0.18G

	CM3Core	Cortex-M3	ARM7TDMI-S (ARM)	ARM7TDMI-S (Thumb)
mW/MHz	0.19	0.24	0.28	0.28
DMIPS/MHz	1.25	1.25	0.93	0.74
DMIPS/mW	6.57	5.21	3.32	2.64

6. Faster time-to-market

Every embedded system designer faces the challenge of fast turnaround, the problem being compounded in many cases by design complexities and the unavailability of quality software tools. The reuse of design and software from earlier systems is a widely adopted technique used to meet this challenge. The Cortex-M3 processor, supported by integrated system components and a broad ecosystem of software tools, provides a complete solution that facilitates faster time-to-market for both new systems and those migrating to the ARMv7 architecture.

6.1 Simple, configurable hardware design and debug

The Cortex-M3 processor is highly configurable and can be adapted to system requirements quickly and efficiently. Designers can create complex systems faster by including the optional MPU and ETM in the system or by removing the DWT and FPB during sythesis. In simple control systems the NVIC can be configured to just 1 interrupt, while in interrupt intensive systems like automotive applications, the NVIC can be configured to support up to 240 physical interrupts with up to 256 levels of priorities. In systems demanding safe operation of many different processes, the MPU can be included to enforce process separation and use of privileged access modes.

The increasingly time consuming validation of applications can make on-chip debug and trace invaluable to on-time delivery of products. The integrated debug capabilities of the Cortex-M3 processor allow for faster verification, removing the need for hard to use ICE units. The system can be viewed through either a JTAG port or a 2-pin Serial Wire Debug port. The optional ETM provides excellent instruction trace capabilities while the FPB and DWT provide the capability to use breakpoints and hardware watchpoints for debug.

6.2 Easy application development

The Cortex-M3 processor includes many features that enable faster software development, with developers not required to write any assembler code or have a deep knowledge of the processor and its register set. The Cortex-M3 processor has a simplified stack-based programmer's model which is simpler to understand and a vector based interrupt scheme to handle interrupt and exceptions. Additionally the Thumb-2 instructions enable the writing of efficient code without the interworking required by ARM/Thumb instructions. The simple and linear 4 gigabytes of address space contains no data pages or code pages. The architecture is very compiler friendly with a flexible register scheme that allows single-cycle multiply operation between any of the registers and the use of any register as a pointer to data structures/arrays.

Traditionally ISRs require an assembler wrapper to handle stack manipulation before and after the main C-based routine is called and during start-up boot code. The Cortex-M3 processor handles all stack manipulation in hardware, removing the need for the assembler and enabling the developer to program just in C and without having to learn exactly how the processor and all of the register banks operate. Additionally, in many operating systems, a hardware timer is used to generate interrupts so that the OS can perform task management. This is especially necessary to ensure that multiple tasks can run on the system and no single task takes up all the resources. The SysTick timer enables Cortex-M3 processor-based systems with this capability and makes OS porting between Cortex-M3 processor-based MCU devices much easier by removing the need to make changes to the system timer code of the OS.

The Cortex-M3 processor is fully supported by the RealView® DEVELOP and CREATE families of tools. RealView SoC Designer technology enables system hardware prototyping on Cortex-M3 processor-based systems and permits early software development and debug. The use of cycle accurate models for the processor core and peripherals permits the development of device drivers in parallel with hardware design. This facilitates faster time-to-market by allowing software development to begin long before the hardware has been designed completely.

The Thumb-2 ISA enables a simplified compilation flow within the RealView Development Suite (RVDS) removing the need for profiling, compilation and interworking of individual ARM and Thumb routines. Applications written earlier in Thumb will run on the Cortex-M3 processor without modification, since the Thumb-2 ISA includes all Thumb instructions. Additionally, with the ARM Unified Assembler framework, ARM assembly code can be easily ported to Thumb-2 instructions. If the source code has been written in a high level language like C, it can recompiled to Thumb-2 code with RealView Compilation Tools or third party tools like the GNU compiler.

The RealView Microcontroller Developer Kit provides a complete software development environment for all microcontrollers based on ARM processors, including the Cortex-M3 processor. Third partly tools include compilers, emulators and debuggers from IAR, Green Hills and Lauterbach, to name a few.

7. Summary

The Cortex-M3 processor is the first ARM processor-based on the ARMv7-M architecture. The central Cortex-M3 core, based on a 3-stage pipeline Harvard architecture, incorporates features like branch speculation, single cycle multiply and hardware divide to deliver an outstanding efficiency of 1.25 DMIPS/MHz. The combination of the Thumb-2 instruction set with features like unaligned data

storage and atomic bit manipulation delivers 32-bit performance at the memory costs expected of 8 to 16-bit devices.

Flexible configuration of integrated hardware components, quick system debug and easy software programming allow Cortex-M3 processor-based designs to go to market faster. In order to enable reliable operation in interrupt intensive automotive applications, the integrated Nested Vectored Interrupt Controller (NVIC) offers deterministic, low latency interrupt handling through tail-chaining technology and can be configured for up to 240 interrupts. For safe operation in industrial control applications, the optional Memory Protection Unit (MPU) enables secure operation through privileged access modes and the separation of processes in an application. The Flash Patch and Breakpoint (FPB) unit, Data Watchpoint and Trace (DWT) unit, Instrumentation Trace Macrocell (ITM) and the optional Embedded Trace Macrocell (ETM™) offer low-cost debug and trace capabilities for deeply embedded devices. Extensive clock gating technology and integrated sleep modes enable low power wireless designs.

The Cortex-M3 processor has been specifically designed to deliver high performance in cost and power sensitive applications. The combination of reduced core size, excellent interrupt latency, integrated system components, flexible configuration, easy high-level programming and a strong software ecosystem make the Cortex-M3 processor a compelling solution for a wide range of systems ranging from complex system-on-chips to low-end microcontrollers.