Beyond the Hype: MIPS® - the Processor for MCUs

Document Number: MD00795
Revision 01.01
September 10, 2010

1 Introduction

MIPS Technologies is well known for developing and licensing high-performance processor cores as well as 32- and 64-bit architectures. A market leader in the Digital Home and Networking sectors, MIPS has adapted its industry-standard MIPS32® architecture to address the requirements of 32-bit microcontroller (MCU) product development, offering a higher-performance, more feature-rich and lower-power solution than that offered by competing cores based on the ARM® architecture.

MCUs are used in a wide and very diverse set of market applications including industrial, office automation, automotive, consumer electronic systems and leading-edge technologies such as wireless communications. The use of MCUs in these types of applications is placing increasing demands on embedded processor cores to provide higher levels of performance efficiency, improved real-time response, lower power and a broad ecosystem of support. These demands stem from a variety of new challenges, including the requirement to run more complex RTOS-controlled software, the integration of higher-speed communications interfaces and more sophisticated deterministic interfaces.

32-bit MCUs are stepping in to provide the solution for next-generation applications. Semico Research Corp. forecasts an 18% compound annual growth rate (CAGR) increase in 32-bit MCU product shipments over the next several years, reaching a 2,573 million unit volume in 2014.

Picking the right processor architecture is a key decision criterion to achieving performance, cost and time-to-market objectives in a MCU product. This paper outlines the design features that are implemented in MIPS® processor cores that contribute to its industry-leading performance. Additionally, we'll compare and contrast MCU design solutions based on the MIPS and ARM architectures– the two most popular embedded processor IP architectures. Our analysis demonstrates that MIPS offers higher-performance, lower-power solutions with more advanced features and superior development support.

2 The MIPS® Architecture

The MIPS® architecture was developed in the early 1980s at Stanford University based on an elegant load/store RISC (reduced instruction set computing) technology. RISC technology achieves faster execution and higher performance compared to previous CISC (complex instruction set computing) architectures by implementing a simple, yet comprehensive instruction set and the use of deep instruction pipelines. In contrast, the ARM architecture is based on a hybrid RISC/CISC architecture that is complex in design and limited in its ability to reach a high level of performance.

The MIPS architecture has evolved over time since the introduction of the first MIPS processor (the R2000) in 1985. The Instruction Set Architecture (ISA) has been extended and its performance improved through several revisions. Current versions include both 32- and 64-bit implementations of the architecture, MIPS32® and MIPS64® respectively. In addition to developing a range of 32-bit processor cores based on MIPS32, MIPS licenses both the MIPS32 and MIPS64 architectures. Architecture licensees, including Broadcom, Cavium Networks, LSI Logic, NetLogic Microsystems, Renesas Electronics, Sony, Toshiba and the Institute of Computing Technology of the Chinese Academy of Sciences, are actively shipping MIPS-Based™ products for digital home, networking, microcontroller and other applications. Combined, MIPS licensees are now shipping over 500M units annually, with over 2 billion MIPS-Based SoCs shipped to-date.

Figure 1 MIPS Technologies Architectures and ASEs

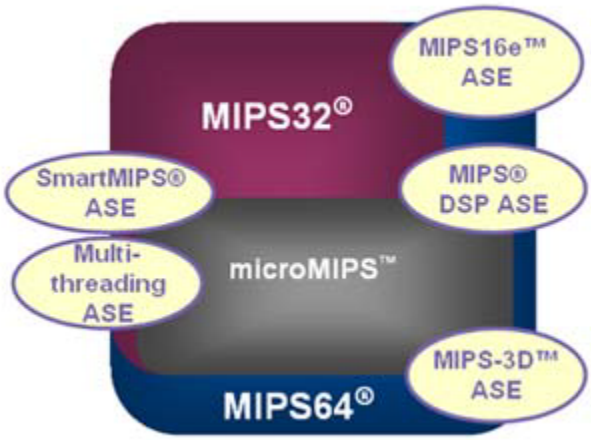


Figure 1 highlights the components that make up the family of available MIPS architectures. The standard MIPS32/64 architectures can be extended through optional Application Specific Extensions (ASEs) including MIPS16e®, SmartMIPS®, DSP, 3D, and Multi-threading. These ASEs are individually designed to provide enhancements for specific applications. The DSP ASE, for example, provides hardware and software enhancements to accelerate signal processing functions in MIPS processor core designs. Similarly, MIPS16e is a set of instructions comprised of 'most often' used MIPS32 instructions decoded into their corresponding 16-bit equivalents. MIPS16e compresses the application code into a smaller memory footprint compared to MIPS32, while maintaining a high level of performance by reducing the memory bandwidth and execution time. Each of the ASEs shown in Figure 1 has a positive effect in increasing the application-specific performance of the target processor core.

MIPS Technologies recently introduced microMIPS™, a complete, self-contained instruction set architecture (ISA) that contains both 16- and 32-bit instructions designed to maximize software code density and execution throughput. microMIPS reduces code size by at least 30% and executes at virtually the same level of performance as MIPS32. microMIPS is incorporated in the MIPS32 M14K™ and M14Kc™ processor cores that were developed for integration in MCU and embedded controller SoC designs.

Through a combination of advanced design techniques implemented in the MIPS32/64 architectures, and leading-edge design features incorporated in its processor cores, MIPS ISA standard software platform outperforms competitive solutions and offers additional flexibility and scope for continued improvements.

2.1 MIPS Architecture Performance

All MIPS processor cores, from high-end multi-core solutions to compact footprint, smaller pipeline stage cores, are designed from the same high performance MIPS32 base architecture.

The majority of the performance improvements of MIPS cores result from enhancements made to the core execution unit, increasing the processor's maximum operating clock frequency by implementing longer pipeline stages, superscalar and multi-threading microarchitectures. Additional performance is achieved from the inclusion of design features such as high-speed memory interfaces, highly-efficient cache controllers, memory management unit, support of a large set of registers and accelerators for floating point support in the standard architecture.

The MIPS32 architecture provides as standard 32 General Purpose Registers (GPRs), with each register 32-bits wide. Additional sets of 32 register files are available as a build-time option that can be used as additional data storage or as ‘shadow registers’ that can be assigned to a dedicated vectored interrupt controller logic, significantly reducing interrupt latency and context switching times over conventional hardware/software means.

Signal processing performance in the MIPS32 architecture is improved through availability of a hardware Multiply Divide Unit (MDU), with software support from a number of signed/unsigned multiply, divide and multiply-accumulate (MAC) instructions. The MIPS architecture employs a separate pipeline for the MDU that operates in parallel with the integer pipeline.

2.2 Comparing MIPS & ARM Performance Features

The fundamentals of RISC technology, along with the extendible hardware and software design in the MIPS architecture, combine to offer a higher-performance, lower-power, more compact solution compared to similar solutions from ARM. MIPS Technologies’ heritage comes from designing high- performance workstations and servers, while ARM started out developing basic cores for low-end mobile systems. MIPS leveraged its high-performance experience and design, migrating to the mainstream embedded system market. ARM has carried forward aspects of its legacy architecture that limit the level of performance it is able to achieve and put it at a disadvantage relative to MIPS.

MIPS32 4K® processor cores, which include the MIPS32 M4K® core, outperform and execute applications faster than the equivalent ARM Cortex™-M series cores. This is in part due to a more efficient MIPS ISA and optimized software tools, but mainly because the MIPS architecture has superior features that are designed in for higher rates of performance and application efficiency, including features to accelerate functions typically implemented in a microcontroller design. For instance:

- MIPS cores include 32 GPRs versus ARM’s 16 GPRs. This translates to higher performance due to less register spillage.
- MIPS cores include shadow register sets, whereas ARM cores do not. Use of shadow registers accelerates the interrupt processing save/restore functions, resulting in fewer cycles being used in context switching and interrupts latency.
- The MIPS architecture primarily executes single-operation instructions, whereas ARM instructions perform multiple operations prior to writing to a GPR (e.g. shift operand, arithmetic, check condition bit and others. This makes it easier for MIPS to achieve higher clock frequencies.
- The MIPS architecture operates with simpler memory addressing modes compared to ARM, making it easier to achieve higher clock operating frequencies.
- The MIPS architecture has less predicated execution, which minimizes reduces logic complexity and enables MIPS cores to achieve higher frequencies.
- Because of the 5-stage pipeline structure the M4K and M14K do not need to predict the direction of the branch. ARM cores employ complex branch prediction and branch speculation logic.
- The MIPS architecture implements branches with delay slots, whereas the ARM architecture does not, translating to higher efficiency in shorter stage pipeline designs with MIPS.
- MIPS provides both 32- and 64-bit architectures, offering backward compatibility and even higher performance with MIPS64. ARM has only 32-bit architecture, not all versions are backward-compatible.

3 A Processor Core designed for high performance MCUs

In 2002, MIPS Technologies introduced the M4K core, a high performance, synthesizable processor core optimized for use in MCU and small footprint embedded controller designs. Licensed by nearly 30 companies, and as part of the 4K family of cores that has > 120 licensees, the M4K has been designed into a wide range of applications as a controller in mobile phone, DTV, cable modem, GPS, and digital camera systems. Additionally, the M4K core is implemented as a standard microcontroller in Microchip Technology's 32-bit PIC32 family of MCU products.

The M4K core is designed with a set of features that combine to provide best-in-class performance, significantly superior to that offered by the ARM Cortex-M series processors.

3.1 M4K® Execution Pipeline

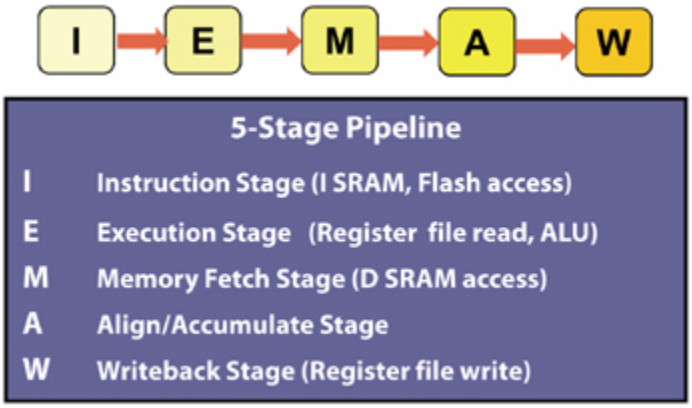
The M4K core achieves a performance of 1.5 DMIPS/MHz while the Cortex-M3 achieves approximately 20% less performance at 1.25 DMIPS/MHz, as listed on the ARM website. (Performance of the ARM Cortex-M0 is even lower at 0.9 DMIPS/MHz, 40% less than the MIPS32 M4K core. The Cortex-M0 also has numerous other limitations which we'll come to later.) To put it another way, the Cortex-M3 would need to use a 20% higher clock frequency just to keep up with the performance of the M4K core, and would suffer additional power consumption as a consequence.

Similarly, as detailed in Section 4 "Performance Benchmark", the M4K core achieves a 2.297 CM/MHz result when running the CoreMark benchmark, 20-30% higher than an equivalent Cortex-M3 based solution. MIPS sees increasing acceptance of the CoreMark benchmark as a more accurate measurement of CPU performance, as compared to Dhrystone DMIPS.

The M4K execution unit employs a 5-stage pipeline micro-architecture, as shown in Figure 2, while the Cortex-M3 core execution is built around a 3-stage pipeline architecture. The deeper pipeline of the M4K core enables it to operate at a higher maximum clock frequency, processing more instructions per second, which results in a higher performance and execution efficiency compared to the Cortex-M3.

In the M4K core, all ALU and shift operations complete in a single cycle. Bypass logic is included in the pipeline, providing fast access to data for use by the next instruction before all pipeline stages complete. Performance improves as a result of reduced cycles needed to execute the specific task.

Figure 2 M4K® Core 5-stage pipeline



3.2 System Co-Processor (CP0)

The System Co-Processor (CP0) is unique to the MIPS architecture, and can be found in the M4K core. The CP0 adds to the core's high performance, operating as an auxiliary execution unit to off-load the management of some of the core resources, including exception handling and memory management.

3.3 GPRs and Shadow Registers

The M4K core has a build-time option to increase the number of GPRs to a maximum of 16 sets, each set being a full complement of 32 registers. These GPRs store parameters and operands on-chip, thereby reducing the overhead of memory transfers, and freeing up instruction cycles. This has a positive effect on increasing the computational throughput.

As mentioned previously, use of the GPRs as shadow registers can improve system performance by reducing the overhead in servicing interrupts, a common event in MCU systems.

When processing an interrupt or exception, the M4K core will determine which shadow set is to be used, establishing it as the active set of GPRs, allowing the interrupt vector to continue execution. This process completely eliminates the need for a context save or restore cycle, since the specified interrupt service routine is the sole owner of the shadow register that is currently active. Not only does this mean that no time is wasted before the interrupt or exception code can begin actual implementation, but it also means that the content of the registers has been preserved since the last exception or interrupt event was active. This saves time on retrieving specific values from the SRAM space.

3.4 MDU

The high-performance implementation of the MDU in the M4K core completes one 32x16 bit multiply (or MAC instruction) in a single cycle. 32x32 multiply/MAC operations execute in 2 cycles.

The MDU has its own dedicated functional unit which operates independently to the core execution pipeline. Any multiply/divide instruction is directed to the MDU, allowing other instructions requiring ALU, load/store and shift operations, for example, to be processed by the core pipeline in parallel. The MDU in the M4K core offers advantages in accelerating signal processing operations, such as FFT, FIR and IIR filter calculations that are typically executed by microcontrollers in industrial and networking type applications. As an example of the capabilities of the M4K to accelerate DSP-type functions, a PIC32 executes a 256 point 16-bit radix-4 FFT in 22K cycles, 283 microseconds @80MHz, 14% fewer cycles than the STM32, a Cortex-M3 based microcontroller.

3.5 SRAM Interface

How fast code and data is accessed has a measurable effect on the processor's performance. Designers make great efforts to design a memory interface to maximize the available bandwidth and minimize delays, the goal being to achieve 0-wait state data transfers. The MIPS architecture incorporates a flexible memory bus structure that enables code to be executed from accelerated flash or from high performance on-chip SRAM. The M4K core incorporates a high-speed, low-latency SRAM interface to both instruction and data memory. The interface supports both single-cycle and multi-cycle memory access. The M4K SRAM interface can be figured in either Dual-mode or Unified-mode. Dual-mode provides the highest performance, with independent buses and control for data (D-SRAM) and instructions (I-SRAM). Dual Mode allows simultaneous transactions to occur on the I-SRAM and D-SRAM interfaces, removing any stalls that might appear on a common bus interface from slowing program execution.

The I-SRAM interface has the ability to redirect signal inputs, allowing D-SRAM read cycles to be redirected to the I-side if required. This allows implementation of the modified Harvard architecture, which is a normal feature of MCU-based systems, allowing non-volatile data to be stored in program memory.

The SRAM interface provides the ability to abort an instruction transaction anywhere in the M4K core's 5-stage pipeline. This allows immediate response from the external system controller to external events such as interrupt requests or requests through the EJTAG debug interface. Quick response to external interrupt events is crucial when dealing with the highly deterministic nature of typical microcontroller applications.

The SRAM interface provides the ability to abort a long latency transaction. Quick response to external interrupt events is crucial when dealing with the highly deterministic nature of typical microcontroller applications.

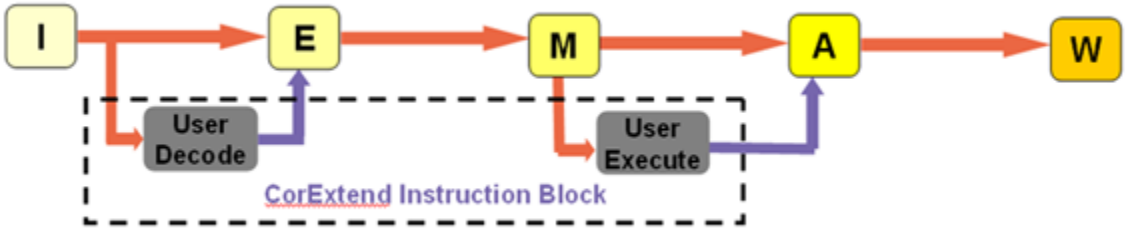
The M4K core SRAM interface provides a high speed, easy-to-use and highly-configurable memory interface into the M4K core, where most transactions are completed within a single clock cycle. It contains no additional overhead in protocol or signals to deal with anything but instruction and data memory, enabling chip designers to extract maximum performance from the M4K core.

The Cortex-M3 does not have such comprehensive memory control features, and consequently suffers in performance relative to the M4K core.

3.6 CorExtend®

CorExtend is another unique feature of the MIPS32 architecture, offering the developer product differentiation and customization capabilities. It is a build-time option that extends the core instruction set through a combination of User Defined Instructions (UDI) and custom hardware. CorExtend enables designers to add functionality to the core to accelerate application-specific functions that are bottlenecks in the target application, improving overall system performance. Possible uses for CorExtend in a typical MCU environment could include the design of dedicated graphics controller, TCP/IP accelerator, custom security/encryption logic, wireless baseband control or other real-time control interfaces. CorExtend works in conjunction with the core pipeline, as shown in Figure 3. The functionality of CorExtend is fully compatible with MIPS32 and is supported by all leading MIPS-compatible development tools.

Figure 3 CorExtend® pipeline structure



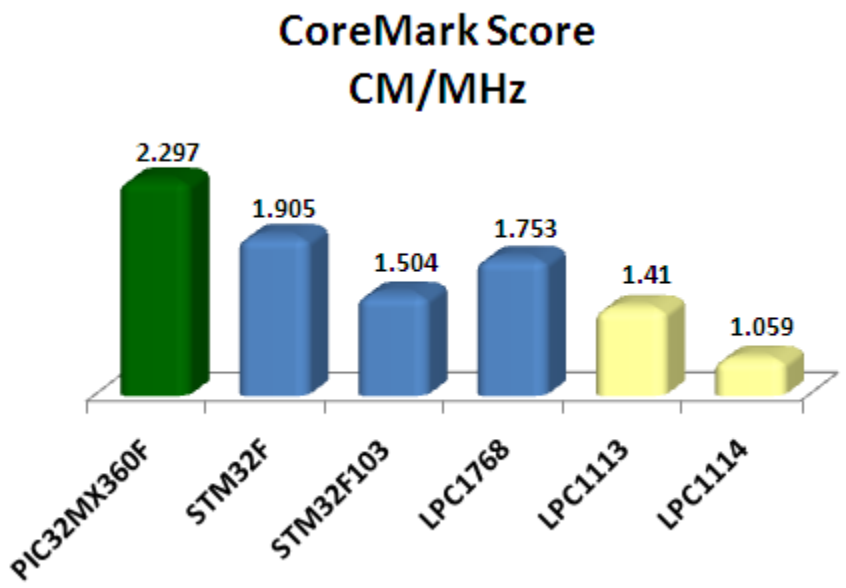
4 Performance Benchmark

CoreMark™ is an open source benchmark from EEMBC® that is designed specifically to test the performance of a processor core. CoreMark's architecture isolates the processor core from any system dependencies, including influences from the memory subsystem, and 'optimizing tricks' the compiler could play. CoreMark tests the performance of a processor's pipeline and of commonly used functions, including read/write, integer and control operations. As such, it provides a less synthetic, closer to real world representation of a processor core's performance and functionality than other benchmarks.

Referring to the CoreMark website: 'the workload is actually comprised of several commonly-used algorithms that include matrix manipulation (to allow for the use of MAC and common math operations), linked list manipulation (to exercise the common use of pointers), state machine operation (common use of data dependent branches), and Cyclic Redundancy Check (CRC is a very common function used in embedded).'

Figure 4 compares data taken from the CoreMark website for MCUs based on MIPS M4K and ARM Cortex-M3 and –M0.

Figure 4 M4K® PIC32 CoreMark results, comparison against ST and NXP Cortex-M devices



The M4K-based PIC32, executing with 2 wait state memory, achieves a higher CoreMark/MHz than competitive Cortex-M3 and Cortex-M0 based devices operating with 0 wait state memory. Fig 4 shows a PIC32 outperforms Cortex-M3 STM and NXP solutions (in blue) at equivalent clock frequencies by 20-50%, Cortex-M0 NXP solutions (in yellow) by over 63%.

5 Low power and compact design

The majority of the power consumption and area of a typical SoC comes from the memory, peripherals and control logic that sit outside of the processor core. However, in addition to the best performance efficiency, MIPS Technologies is mindful that in microcontroller design, silicon cost and power consumption are key criteria for successful products. MIPS incorporates specific features into the processor core design for minimum area and power consumption, some of which will be described in this section. As a result, MIPS Technologies adds to its leadership in performance by offering ‘best-in- class’ area and power solutions, as shown when comparing the M4K with the Cortex-M3 in the following sections.

5.1 Features for low power

The high performance of the M4K micro-engine allows applications to run at a lower clock frequency than would otherwise be the case. Power is directly proportional to frequency, in that reducing frequency reduces power. We have seen earlier that the M4K core has at least 20% more processing performance capability than the Cortex-M3, which translates to better power efficiency (DMIPS/mW) – which means less power is required for the same job. The high performance of the M4K core helps to reduce power consumption since it can complete tasks faster, thus spending more time in a lower power (idle) state.

The M4K core is a synthesizable design that is portable across low power processes and libraries. The core is a static design, allowing on-the-fly clock changes (to a lower frequency if required). It even supports the clock being stopped, which would reduce power consumption to an absolute minimum uW level (power consumption in this instance would be determined primarily by the process leakage current).

The M4K core provides several power management features, controlling active power through the use of fine-grain clock gating and support of power-down modes. The majority of power consumed by the M4K core is through the clocking logic and registers. The M4K core employs extensive clock gating control throughout the core, in effect providing a mechanism to turn off select areas of the core when not in use. The M4K core also provides mechanisms to enter into low-power and sleep modes through the use of internal registers and control by a specific WAIT instruction. When the WAIT instruction is executed, the internal clock is suspended and the pipeline is frozen. Any interrupt, or a reset, will cause the core to exit sleep mode and resume normal operation.

Efficient implementation of these power management features, along with low-power fabrication tools, can contribute to a significant reduction in active power consumption.

Table 1 compares the power consumption of the M4K and Cortex-M3 at 180nm and 90nm for both speed and area optimized configurations. The operating frequency of the M4K was reduced to match the maximum clock speed of the Cortex-M3 and its power consumption measured at that frequency.

	TSMC 180nm G					TSMC 90nm G				
	M4K			Cortex-M3		M4K			Cortex-M3	
Optimization	**Speed**		**Area**	**Speed**	**Area**	**Speed**		**Area**	**Speed**	**Area**
Frequency (MHz)	180	100	50	100	50	360	275	50	275	50
Power (mW/MHz)	0.37	0.21	0.17	0.33	0.2	0.06	0.03	0.02	TBD	0.1
Power Efficiency (DMIPS/mw)	4.05	7.14	8.82	3.75	6.25	25	50	75	TBD	12.5
Libraries	TSMC 7T SVT			ARM SC7		TSMC 9T SVT		TSMC 7T SVT	ARM SC9	

Table 1 Speed/power comparison of M4K® and Cortex–M3

Table 1 highlights, at the same clock frequency:

- In 180nm, with a speed-optimized configuration, the M4K consumes only **65%** of the power of the Cortex-M3. With an area-optimized 50MHz target clock speed, the M4K operates with **15%** less of the power required by the Cortex-M3.
- In a 90nm area-optimized 50MHz configuration, the M4K only consumes **20%** of the power of an equivalent Cortex-M3. Unfortunately, the ARM website does not provide 90nm power numbers, but, by extrapolation, one can calculate the significant power savings that the M4K achieves.
- The M4K wins over the Cortex-M3 in Power Efficiency (approximately 2x in 180nm, and 6x in 90nm)—no surprise when taking into account the higher performance, lower power profile of the M4K.

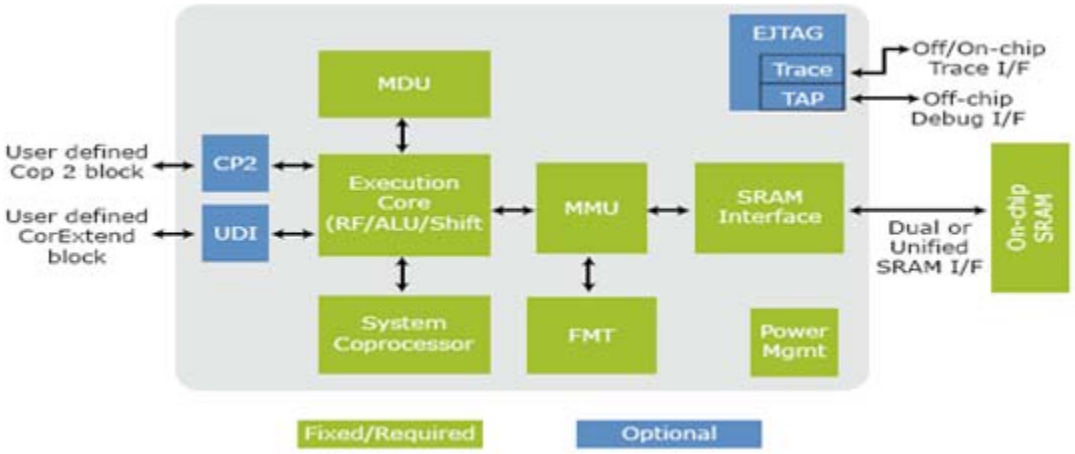
The M4K core shows similar low power characteristics in 130nm. Cortex-M3 data is not available at 130nm for comparison, so below we have indicated only the M4K power data at 130nm:

- Speed-optimized configuration @ 216MHz max frequency consumes 0.17mW/MHz power
- Area-optimized configuration @100MHz frequency consumes 0.06mW/MHz power

5.2 Features for small size

The M4K core is a function-efficient, highly-configurable and flexible processor core. Figure 5 shows the optional blocks in the M4K core, including Debug/Trace (ETAG), COP2 co-processor interface and CorExtend extensions. The MIPS16e ASE instruction decoder is optional.

Figure 5 M4K® core block diagram



To reduce gate count, the M4K core provides a comprehensive set of build time configuration options. Configurable options include enabling/disabling debug features, setting the number and type of debug/trace breakpoints, enabling fast or slow MDU, setting the number of GPR files and enabling either unified or separate data and address SRAM interface.

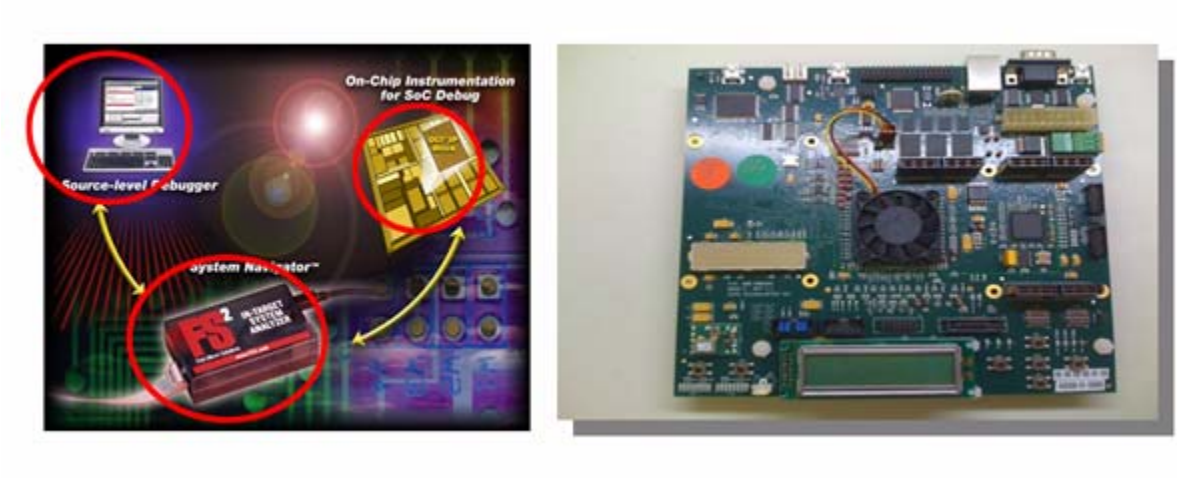
Configuration options are used to synthesize an implementation that can be either speed- or area- optimized to satisfy the required application target frequency while generating minimal size and gate count. Core area is dependant on process, cell libraries and the performance required by the application. A meaningful comparison of the design efficiencies of the M4K core and Cortex-M3 in reducing core area should take all of these factors into account. The M4K core, however, can be built in as a little as 33K gates, which in spite of it having more features, is still smaller than the Cortex-M3.

6 Ecosystem

The SoC development environment, including hardware/software debug tools and other third party solutions, is an important consideration in processor selection.

MIPS Technologies has a dedicated team that provides a range of hardware and software development tools that a designer would need to successfully integrate and test a MIPS processor core in an SoC, and validate its operation in the target system application. These tools include the System Navigator™ EJTAG debug probe, a GNU software tool-chain that supports both RTOS and Linux targets, Cycle Accurate and Instruction Accurate simulators for software profiling and co-simulation, an FPGA-based development/evaluation board and Navigator Integrated Component Suite (ICS), and an Eclipse-based development environment that is fully integrated with other MIPS tools.

Figure 6 System Navigator™ debug probe and SEAD3 development board



In addition, MIPS Technologies has established a MIPS Alliance Program (MAP) to support its broad third-party ecosystem. MAP is a community of over one hundred partners who together provide hundreds of solutions that support the MIPS architecture and processor cores. This ecosystem is a resource for designers to reduce time-to-market with MIPS-compatible tools and solutions targeting a wide range of market applications, including MCUs.

The MIPS ecosystem has developed into a rich and diverse support infrastructure that includes partners providing OS, RTOS, hardware debug tools, software development products, application software, EDA tools, physical IP, and other application-specific related products. For MCU development, examples of support from the ecosystem are available from numerous vendors. Some MIPS Alliance Partners that provide MCU support are:

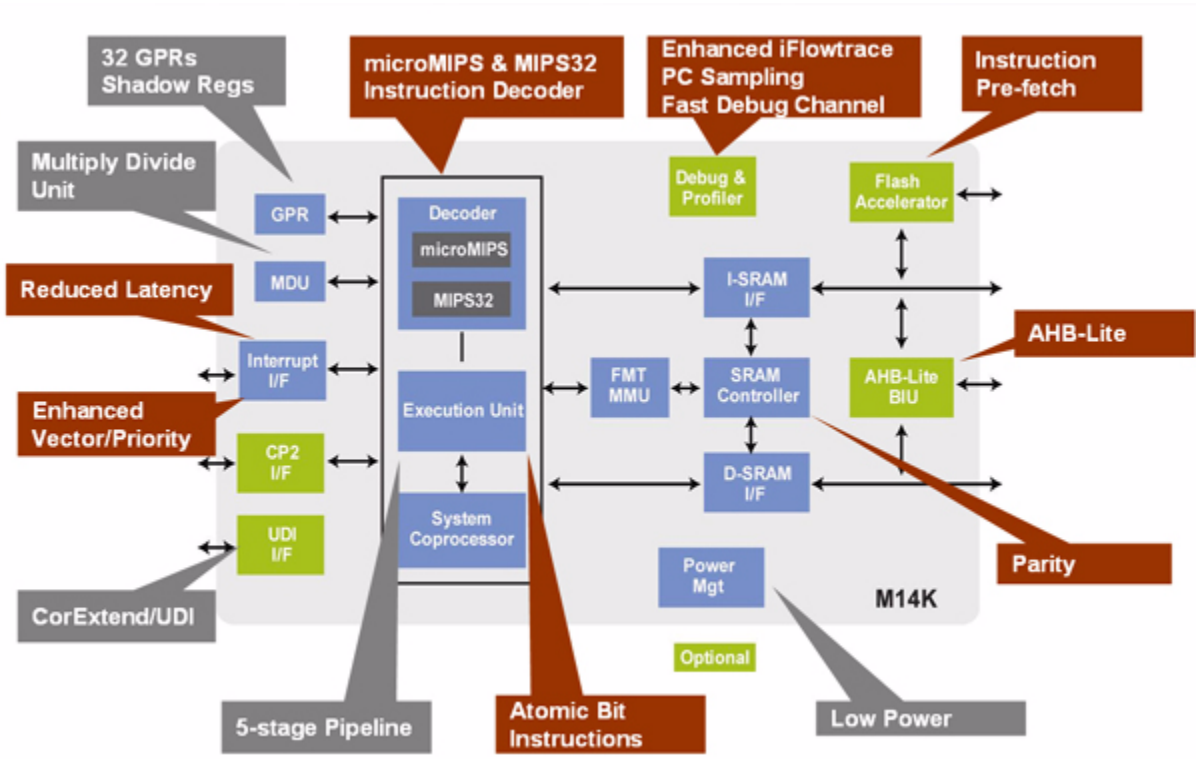
- RTOS vendors such as Express Logic, Mentor Graphics, Micrium, Segger, Green Hills Software, Wind River, CMX, FreeRTOS
- Debug probes and emulators from Ashling, Lauterbach, Macraigor, Corelis
- Software development tools from CodeSourcery, Green Hills, Mentor Graphics, Hi-Tech
- Simulation models from Carbon, Imperas
- SoC IP from Sonics, Dolphin, Denali
- EDA/ESL tools from Synopsys, Cadence, Magma

7 MIPS32 M14K™ Core

MIPS Technologies' innovations for microcontrollers and other high performance, compact footprint, cost-sensitive embedded applications continue with the recent introduction of two new processor cores in the MIPS32 4K core family. The MIPS32 M14K and M14Kc cores are implemented as a dual decoder design, including MIPS32 and microMIPS instruction decoders. These cores are the first MIPS32- compatible cores to incorporate the new microMIPS code compression ISA. microMIPS delivers an uncompromised level of performance with a high degree of code density: maintaining the performance of MIPS32 while reducing code size by at least 30%.

The M14K core is designed using the same high-performance 5-stage pipeline architecture found in the M4K core. A superset of the M4K core, the M14K processor core retains all of its capabilities with additional features to reduce interrupt latency, accelerate access to flash code and enhance interrupt handling. In addition, the M14K core provides a comprehensive set of advanced debug/profiling features and a standard AHB interface.

Figure 7 M14K™ core block diagram



The M14K core has all of the same advantages that the M4K core has over the Cortex-M3: higher performance, lower power, smaller size, more configurability and flexibility. In addition, the M14K core provides additional benefits over the Cortex-M series as shown in Table 2.

Feature	MIPS M14K	ARM Cortex-M3	ARM Cortex-M0
Architecture	Harvard	Harvard	Von Neumann
Pipeline stages	5	3	3
ISA	MIPS32 microMIPS	Thumb-2	Thumb Thumb-2 (subset)
Legacy 32-bit decoder	Y - MIPS32	N	N
Total instructions	300+	155	56
DMIPS performance	1.5 DMIPS/MHz	1.25 DMIPS/MHz	0.9 DMIPS/MHz
CoreMark performance	2.36	1.76	1.6
GPRs	32	16	13
GPR sets (max)	16	1	1
Interrupt control	Y - int & ext	Y - int NMC	32
Priority levels	8	4	4
Interrupt latency	10 cycles	16 cycles	16 cycles
Tailchaining	Y	Y	Y
Atomic bit instructions	Y	Y	N
Instruction-only trace	Y	N	N
PC sampling	Y	N	N
Performance counter	Y	N	N
Fast debug channel	Y	N	N
Multiply-divide unit	Y	Y	Multiply only
Local code RAM (max)	4 GB	1 GB	None
Local data RAM (max)	4 GB	1 GB	None
Parity	Optional	N	N
Fast SRAM interface	Y	N	N
Flash memory prefetch	Y	N	N
MMU	Y- FMT	Optional	Optional
External interface	AHB-Lite	AHB-Lite	AHB-Lite
Co-processor interface	Y	N	N
Custom instruction support	Y	N	N

Table 2 Feature comparison of M14K™, Cortex–M3, and Cortex-M0

A brief mention of Cortex-M0: The Cortex-M0 implements a version of the ARM architecture, called ARMv6, previous to that implemented by the Cortex-M3. It is, in essence, an ARM7 architecture with a few Cortex-M3 features thrown in. It implements a 3-stage pipeline, which at 0.9 DMIPS/MHz has even less performance than the Cortex-M3.

The Cortex-M0 executes both Thumb and Thumb-2 instructions, totaling 56 instructions—only six of which are 32-bit instructions. A majority of the code written for the Cortex-M3 will not run on the Cortex-M0 without modification.

The Cortex-M0 reverts to a von Neumann architecture, the same as that implemented in ARM7. Cortex-M0 does not support local memory; instead it accesses code and data from main memory over the AHB bus, which slows down performance significantly as a consequence of the additional wait states required before the data transfer is completed.

Fully loaded, the Cortex-M0 is about 24K gates. Yet in spite of this size, the Cortex-M0 lacks many of the features and performance that are standard in an area optimized configured M4K or M14K core of approximately 33K gates. What is lost in performance and functionality in the Cortex-M0 is not worth the area savings.

8 Conclusions

MIPS Technologies is recognized for delivering high-performance, application efficient products for the digital home and networking market sectors. The standard MIPS architecture has been specifically enhanced to address the technical challenges facing the microcontroller designer, providing advantages over the ARM Cortex-M series products in terms of superior performance, lower power, more advanced features.

The efficiency and configurability of the M4K and M14K cores present designers of MCUs and embedded controllers with a ‘3-for-1’ package: a single M4K/M14K core has more performance and functionality than either the Cortex-M3, -M0 or -M1, offering a single replacement for all 3 cores.

A growing number of today’s MCU applications require high performance, low power and real-time response that are best served by a 32-bit processor architecture.

The following summary provides the substance beyond the hype, and key considerations for choosing a MIPS processor core:

- Performance
    - The MIPS M4K and M14K cores implement a 5-stage pipeline architecture achieving 1.5 DMIPS/MHz performance. ARM Cortex-M3 and -M0 are 3-stage pipeline designs, with 1.25 and 0.9 DMIPS/MHz performance respectively – a 20% and 60% reduction in performance compared with the M4K/M14K.
    - The M4K and M14K cores reach a maximum clock frequency that is 20% higher compared to an equivalently configured Cortex-M3, at 180 and 90nm.
    - The results of CoreMark benchmark give the M4K PIC32, with a 2.297 CM/MHz score, a 20% performance advantage over the Cortex-M3 STM32F executing at similar clock frequency.
    - The M14K interrupt latency is 10 cycles, as opposed to 12 cycles for the Cortex-M3. The M14K core requires 30% fewer cycles to service back-back interrupts than Cortex-M3.
    - The PIC32 and M14K core implement a pre-fetch buffer to reduce access times to flash memory and a fast SRAM interface which speeds up execution times compared to Cortex-M3.
    - The DSP performance of the PIC32 is 14% higher than the STM32F executing commonly used signal processing FFT algorithms.

- Low Power & Power Efficiency
    - The M4K offers significant power savings and higher performance efficiency over the Cortex-M3 across MCU target process nodes. At 180nm, M4K consumes up to 65% less power and achieves a 2x higher Power Efficiency compared to a Cortex-M3 operating at the same clock frequency.
- MIPS Ecosystem: The broad range of offerings from MIPS and its partners who provide hardware and software development tools, compatibility with leading RTOSes, middleware and support for leading EDA tools, enable designers to reduce development time and accelerate time to market.
- Proven technology, reduced risk: the MIPS32 and MIPS64 architectures have been successfully designed into billions of SoCs in a wide range of high performance applications. MIPS is the market leader in the Digital Home (DTV, STB), Broadband Access and Wireless Networking (WLAN, WiMAX).

A new generation of embedded applications is emerging, driven by the increasing performance capability of 32-bit CPU architectures. MIPS Technologies' high performance and performance/power-efficiency leadership is perfectly suited to drive this new generation of products.

9 References

- MIPS Technologies www mips.com
- ARM www.arm.com
- EEMBC CoreMark www.coremark.org
- Microprocessor Report www.mdronline.com
- Berkeley Design Technology Inc. www.BDTI.com
- 'See MIPS Run' by Dominic Sweetman. ISBN 13; 978-0120884216
- 'Exploring the PIC32' by Lucio Di Jasio. ISBN 13: 978-0750687096

. Ŝë2yŘ ŭKŜ I ëLJŜY a l t {† π ŭKŜ t ℕ2ŐŜăă2ℕ ₮2ℕ a / ¦ ăΣ wŜØℵăℵ2y℣ лмΦлм