Introducing the Cortex-M0+ processor: The Ultimate in Low Power

The ARM Cortex-M0+ processor has just been announced. In this article we will introduce this new processor and explain how it can bring additional advantages to your embedded products.

The Cortex-M0+ processor builds on the successful Cortex-M0 processor, which was released three years ago. The Cortex-M0 processor provided excellent code density and best in class energy efficiency in about the same silicon area as 8-bit and 16-bit processors. Since the release, the Cortex-M0 processor is the fastest ever licensed ARM processor core, passing 50 licensees by the end of 2011. Since then our design teams have continued to work hard and work with our Partners closely to see what can be improved. From that work the concept of the "Flycatcher" (project code name for Cortex-M0+) was formed.

Our Partners had highlighted several criteria in demanding applications that cannot be addressed by existing processor solutions. One of the top requirements was to achieve even lower power and much greater energy efficiency. In order to satisfy these requirements the Cortex-M0+ processor was completely redesigned from the ground up while keeping complete instruction set and debug compatibility. For the first time, ARM has produced a processor design with a two stage pipeline, and used the opportunity to improve the performance while maintaining a very similar maximum frequency. The overall result is very encouraging. When compared to the existing Cortex-M0 processor, the Cortex-M0+ processor consumes only two thirds of the dynamic power in our power analysis test, when running Dhrystone loops.

	Cortex-M0+	**Cortex-M0**
Static power	0.95uW	0.96 uW
Dynamic power	11.21uW/MHz	16.36uW/MHz

Table 1: Power characteristics based on TSMC 90LP processor at 50MHz operation in minimum configurations

The lower power consumption of the processor is certainly important, but how about system level power consumption? By moving to a two stage pipeline design, the branch shadow of the processor is reduced. As a result, the number of accesses to Flash memory is cut. Flash memory power often contributes the majority of the power consumed in a microcontroller so any reduction in Flash accesses has then a very direct effect on the overall power.

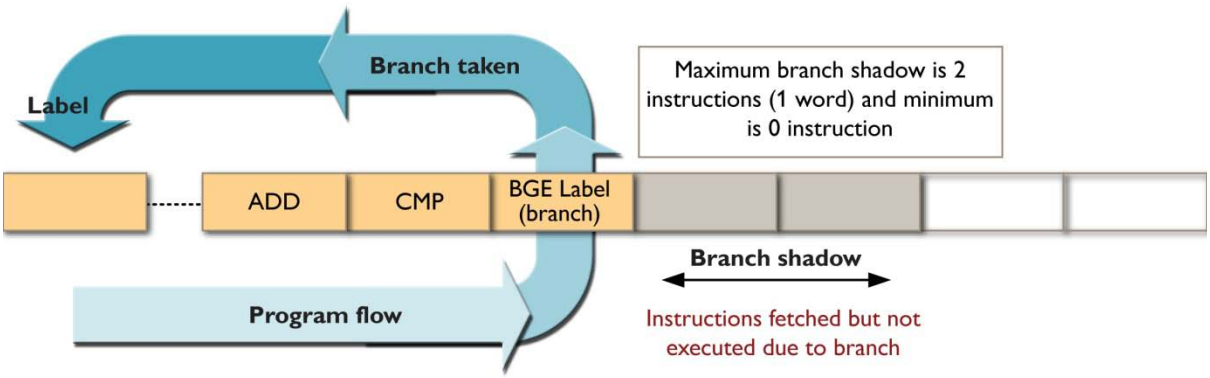
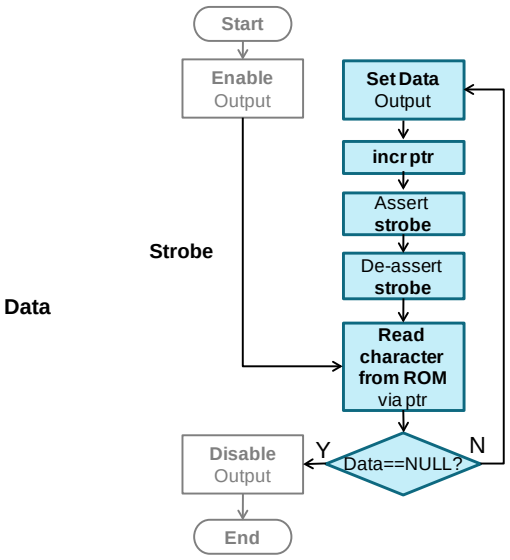


Figure 1: Branch shadow

System level features

The Cortex-M0+ processor supports a new I/O interface which allows single cycle accesses and so enables faster I/O port operations. The processor's I/O interface is a generic 32-bit interface to which microcontroller vendors then add their own I/O port peripherals. With this I/O interface the Cortex-M0+ processor can perform peripheral accesses faster than any of the popular microcontrollers.




```
Instruction                    Cycle  Size
Loop
    STR      r4,[r1,#0]    ;1     2
    ADDS     r0,r0,#1      ;1     2
    STR      r5,[r2,#4]    ;1     2
    STR      r6,[r2,#4]    ;1     2
; Loop entry point
    LDRB     r4,[r0,#0]    ;2     2
    CMP      r4,#0         ;1     2
    BNE      Loop          ;2     2
```

MCU	Loop cycles	Loop size (bytes)
MSP430	12	20
78K	10	14
8051 asm	15	12
Cortex-M0+	9	14

Figure 2: I/O interface feature allows faster I/O operations

Since the I/O interface is part of the system's memory map, the I/O register on this interface can be accessed with normal pointers in C and does not require specific C language extension features such as special data types.

Because accesses to the AMBA® AHB-Lite™ and the single cycle I/O interface can be made concurrently, the Cortex-M0+ processor can fetch the next instructions while accessing the I/Os. This enables single cycle I/O accesses to be sustained for as long as needed.

Power-sensitive applications will also benefit from this improvement, by either running at lower speed for the same I/O toggling frequency, or by completing the I/O control more quickly, then going faster into a sleep mode.

The Cortex-M0+ processor also includes many useful features from the Cortex-M3 and Cortex-M4 processors previously not available in the Cortex-M0 processor. For example, it supports privileged and unprivileged execution levels, and a Memory Protection Unit (MPU) which is similar to that in the Cortex-M3 and Cortex-M4 processors. The MPU is a programmable component with 8 programmable regions, and can be used by an OS to create access permission rules for various application tasks dynamically during run time. By using this mechanism, the design can prevent an application task from corrupting memory space used by other tasks and the OS kernel.

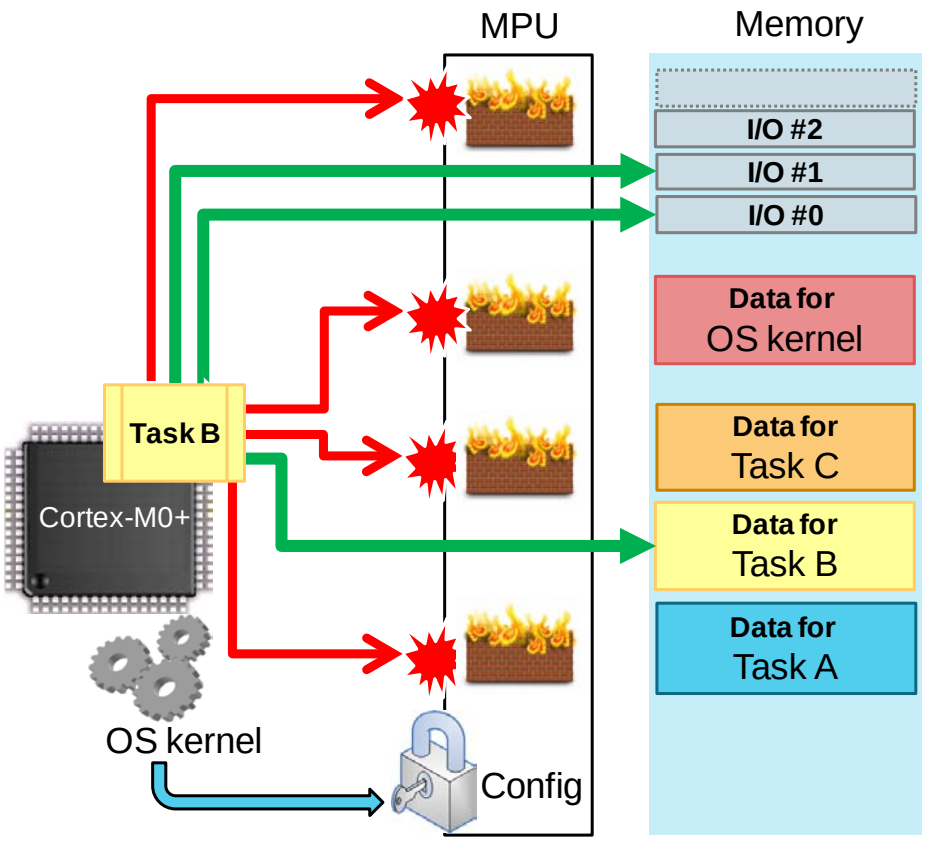


Figure 3: Memory Protection Unit (MPU)

In addition, the Cortex-M0+ processor supports vector table relocation, just like the Cortex-M3 and Cortex-M4 processors. This allows easy reassignment of exception vectors at run time. All these features are configurable options for silicon designers. In its minimum configuration the silicon area of the Cortex-M0+ processor is the same as the Cortex-M0 processor.

For some silicon designers, more good news is that they can very easily use the Cortex-M0+ processor with 16-bit Flash memories with minimal impact to the performance or the system design. The Cortex-M0+ processor can be configured to generate instruction fetches as 16-bit rather than 32-bit accesses. Because most of the instructions in the ARMv6-M architecture are 16-bit, the system can still run with very good performance.

This focus on maintaining a very low silicon area allows our Partners to conceive size optimized solutions that address consumer and safety markets such as small sensors/actuators in factory automation, automotive and medical.

Debug

There are also new features in the debug support. Beside the debug features in the existing Cortex-M0 processor, the Cortex-M0+ processor supports a Micro Trace Buffer (MTB) which provides simple instruction trace. The programmer allocates a small part of the system SRAM as a trace buffer and the MTB stores instruction flow information to the reserved SRAM as a circular buffer. After the processor is halted, for example due to a breakpoint, the debugger can then retrieve the trace information via the widely adopted Serial Wire Debug (SWD) connection – this needs only two pins – and then reconstruct the recent execution history. The MTB can also support "one-shot" triggering.

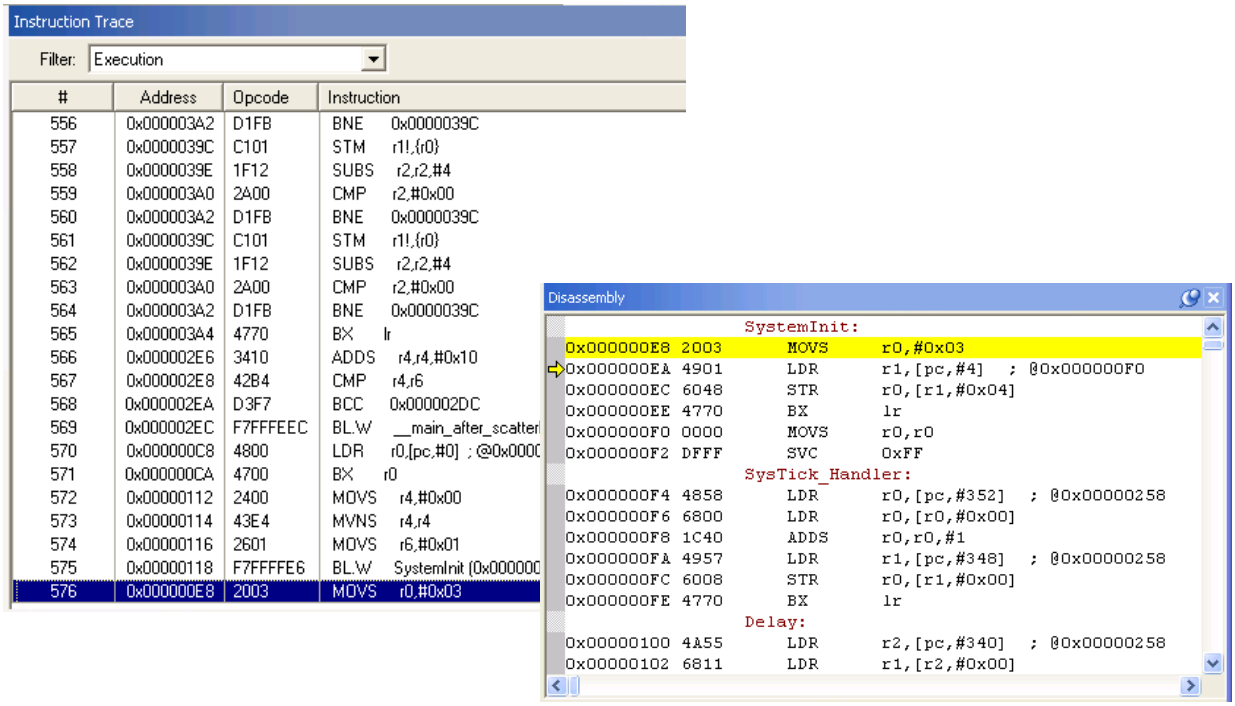


Figure 4: Instruction Trace support via MTB, already fully supported in Keil MDK-ARM

The Serial Wire Debug interface in the Cortex-M0+ processor can optionally include a feature called Multi-drop Serial-Wire. This allows multiple Multi-drop Serial-Wire capable devices to share a single debug connection, of particular benefit when building many-core SoC systems.

Using Cortex-M0+ processor-based microcontrollers

For software developers, the great news is that you can reuse all your existing software for Cortex-M0 processor-based products. You can also use the same compiler suite, the same IDE and the same debug adaptor. The instruction set of the Cortex-M0+ processor is identical to that of Cortex-M0 processor, and supports all features included in the Cortex-M0 processor. The Cortex-M0+ processor also supports the sleep mode features as in Cortex-M0 processor, and provides excellent interrupt processing capability.