

Technical Review of the Arm Cortex-M3 Processor and LPC1768 microcontroller by Dan Coleman

Arm Cortex-M3 processor introduction

The Arm Cortex-M3 is a next generation core that offers system enhancements such as an enhanced debug features and a higher level of support block integration. [1] The M3 is specifically developed to enable partners to develop high-performance, low-cost platforms. It was designed to be fully programmed in C, and is used in about 75% of new 32-bit systems in the IOT space in 2016 [2]

It incorporates a 3-stage pipeline and uses a Armv7-M Harvard Architecture. [3] This, unlike von Neumann architecture, means that the CPU can both perform a data memory access and read instructions at the same time, even without a cache. It also has a third bus for peripherals. It is a 32-bit microprocessor meaning it has a 32-bit data path, register bank and memory interfaces.

It supports the **Little endian** memory system where the least significant byte (the "little end") of the data is placed at the byte with the lowest address. The rest of the data is placed in order in the next three bytes in memory. It also supports the **Big endian** memory system where the most significant byte (the "big end") of the data is placed at the byte with the lowest address. The rest of the data is placed in order in the next three bytes in memory. It has 16 normal 32-bit registers and 5 special purpose registers the Program Status Register(PSR), Priority Mask Register (PRIMASK), Fault Mask Register (FAULTMASK), Base Priority Register(BASEPRI), and Control Register (CONTROL). [4]

Security

The Arm Cortex-M3 also includes an **optional Memory Protection Unit (MPU)** which supports 8 regions with sub regions and background region. [1] [3] MPU's are a trimmed down version of a memory management unit that provide only the memory protection. This allows software to access only specific areas of memory that it needs to operate in, avoiding any accidental or unexpected overwrites of critical data. The MPU implements Zero Latency Memory Protection, which as previously has 8 register- stored regions. The same regions are used for instructions and sending data. There is no address translations or page tables and the minimum region size is 32 Bytes. [5]

Interrupt Capability

Another key feature is the built in **Nested Vectored Interrupt Controller (NVIC)** which is tightly coupled with the processor core. The Cortex has 1-240 prioritizable interrupts supported. [5] This means there is no need for a standalone external interrupt controller, as all interrupt handling is taken care of by the NVIC, this means the interrupts are managed manually by the processor. [3] The NVIC also handles the multiple sleep modes supported by the processor. These include Sleep Now (Where it waits for event instructions or an interrupt), Sleep On Exit (where it will sleep on return from the last Interrupt Service Routine (ISR)), or Deep Sleep (long duration sleep used to conserve power). [3] Finally, it has an optional Retention Mode with Arm Power Management. Traditionally ISRs require an assembler wrapper to handle stack manipulation before and after the main C-based routine,

however in the Cortex M3 it is handled by the hardware. [6] The M3 two modes Thread and Handler are the two operation modes. They determine if the M3 runs a normal program or runs an exception handler like an interrupt handler or a system exception handler.

Instruction Set

The Instruction is the complete set of all the instruction in machine code and can be recognized and executed by a Central Processing Unit (CPU). [6] The processor uses the **Thumb-2 instruction set**. While ARM instructions are at a fixed 32 bits and the original Thumb are at a fixed 16 bits, Thumb-2 instructions are allowed be either 16 or 32 bit. [4]It is said that on average the mix of 32 and 16-bit instruction yields a better code density then when compared to 8, 16 or 32 bit fixed architectures. [3] The thumb-2 instruction set gives an approximately 25% improvement in performance over Thumb and 26% improvement over ARM. [5] This leads to reduced memory requirements and allows us to maximize the use of the Flash memory given. [3] Developers are not required to write any assembler code or have a deep knowledge of the processor and its register set. [2]

Bus Width

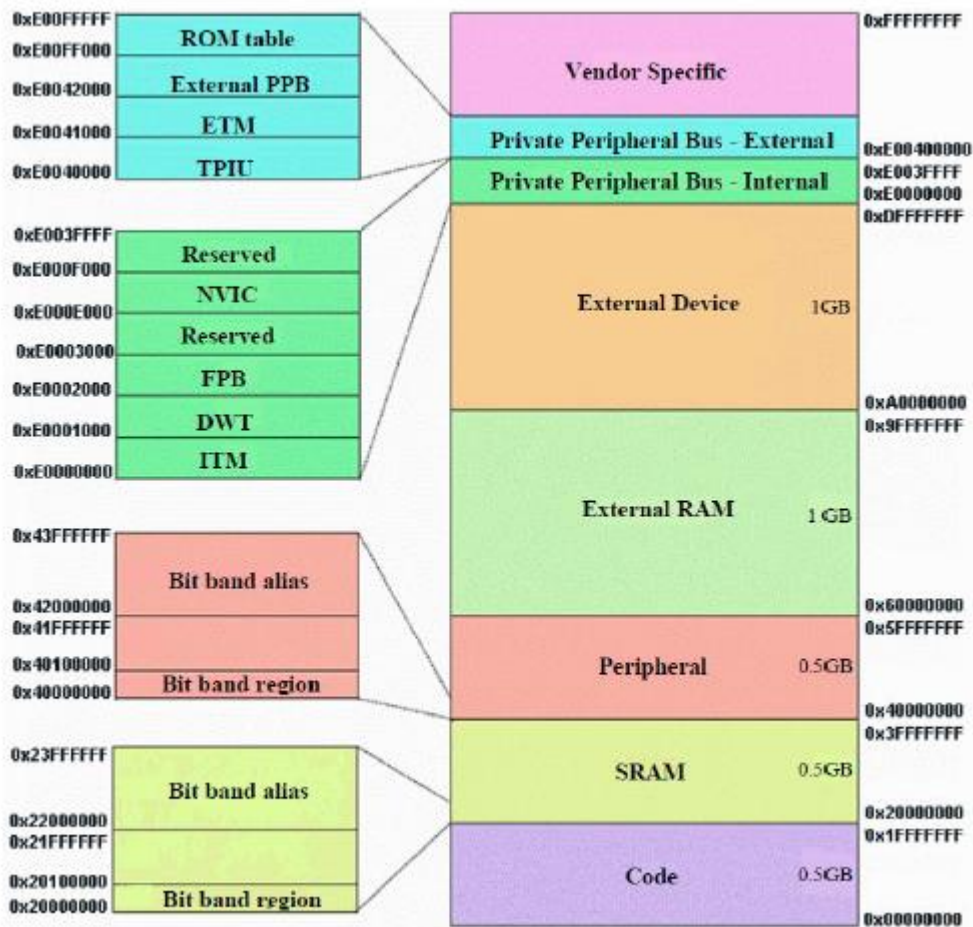
The bus width is the largest amount of bits allowed to be passed through the bus at once, as mentioned above the Cortex-M3 supports Thumb-2 which allows instructions to be up to 32-bits.

Therefore, the bus width is 32-bits.

Memory Maps

The Cortex-M3 has a predefined 4GB memory map. This allows the built-in peripherals, such as the interrupt controller and the debug components, to be accessed by simple memory access instructions.

The Internal bus infrastructure is optimized for this memory usage shown in the below image but can be used differently. [2]



LCP1768

Peripheral Support

The LPC1768 is a microcontroller that is based on the ARM Cortex-M3 processor and is used for embedded applications featuring a high level of integration and low power consumption. The LPC1768 operates at a CPU frequency of up to 100 MHz. It includes 512kB of on flash memory for programming with a flash memory accelerator that allows high-speed 120MGz operations with no wait states. [1] It also comes equipped with 64KB of SRAM (Static Random-Access Memory), up to 32 kB of SRAM on the CPU with local code/data bus for high-performance, with up to two 16 kB SRAM blocks with separate access paths for higher throughput. These SRAM blocks may be used for Ethernet, USB, and DMA memory, as well as for general purpose instruction and data storage.

It also features In-System Programming (ISP) and In-Application Programming (IAP), Ethernet MAC with RMII interface and dedicated DMA controller, USB 2.0 full-speed device/Host/OTG controller, it also has 70 general I/O pins with configurable pull-up/down resistors, a 12-bit ADC and 10-bit DAC. It has Four general purpose timers/counters, with eight capture inputs and ten compare outputs. Each timer block has an external count input. Specific timer events can be selected to generate DMA requests. It also features One motor control Pulse Width Modulator (PWM) with support for three-phase motor control, one standard PWM/timer block with external count input and a Cortex M-3 system tick timer, also including an external clock input option which brings us to the clocks. [7]

Clocks

The LPC1768 has three independent oscillators. An Oscillator is an electronic circuit that produces a periodic vibrating (oscillating) electronic signal, typically either a square wave or a sine wave (~) Oscillators are used in ADC (Analog to Digital Conversion) and DAC (Digital to Analog Conversion). The main Oscillator is the first it is charge of the USB PLL and works with the Internal RC Oscillator on the System Clock select to control the Main PLL. The Main PLL controls the USB and CPU Clock Dividers. The CPU Clock Divider is what gives power to the Arm Cortex-M3, along with the Ethernet Block, DMA, NVIC GPO. The Internal RC Oscillator also controls the Watchdog timer which automatically generates a system reset of the microcontroller within a reasonable time if it enters an error state. On the LCP1768 there is a single state watchdog timer, but multistage timers allow timed interrupts. Finally, the RTC Oscillator also assists in the System clock select and the watchdog timer, but also controls the Real-Time Clock which is what RTC stands for, it's purpose is like a watch - it runs on a battery and keeps time for you even when there is a power outage. [2]

Power Requirements

Which brings us to the power requirements. Which are as follows: [2]

Type	Draw(mA)	Consumption (W)	% Normal
Normal mode (while{ 1 }[]loop)	138.5	0.69	100
128 MHz, all Peripherals on (PHY on, ETH in)	220.6	1.003W	159
Sleep	127.0	0.635	91.6
Deep Sleep	96.5	0.4825	69.7
Power Down	96.3	0.4815	69.5
Deep Power Down (PHY off)	57.7	0.2885	41.6

Note: I put in Consumption for All Peripherals on as was in the referenced notes although I believe it may be a typo and that 1.103 may have been the actual figure.

References

- [1] https://www.nxp.com/docs/en/data-sheet/LPC1769_68_67_66_65_64_63.pdf
- [2] Anatomy of a Contemporary Platform Lecture notes.
- [3] <https://developer.arm.com/products/processors/cortex-m/cortex-m3>
- [4] <https://www.quora.com/What-are-the-special-purpose-registers-in-ARM>
- [5] https://www.arm.com/files/pdf/CortexM3_Uni_Intro.pdf
- [6] “The Definitive Guide to the ARM® Cortex-M3” – 2nd Edition by Joseph Yiu
- [7] <https://www.nxp.com/docs/en/user-guide/UM10360.pdf>