Which ARM Cortex Core Is Right for Your Application: A, R or M?

Introduction

The ARM® Cortex® series of cores encompasses a very wide range of scalable performance options offering designers a great deal of choice and the opportunity to use the best-fit core for their application without being forced into a one-size-fits-all solution. The Cortex portfolio is split broadly into three main categories:

- Cortex-A -- application processor cores for a performance-intensive systems
- Cortex-R – high-performance cores for real-time applications
- Cortex-M – microcontroller cores for a wide range of embedded applications.

Cortex-A

Cortex-A processors provide a range of solutions for devices that make use of a rich operating system such as Linux or Android and are used in a wide range of applications from low-cost handsets to smartphones, tablet computers, set-top boxes and also enterprise networking equipment. The first range of Cortex-A processors (A5, A7, A8, A9, A12, A15 and A17) is based on the ARMv7-A architecture. Each core shares a common feature set including items such as the NEON media processing engine, Trustzone for security extensions, and single- and double-precision floating point support along with support for several instruction sets (ARM, Thumb-2, Thumb, Jazelle and DSP). Together this group of processors offers design flexibility by providing the required peak performance points while delivering the desired power efficiency.

While the Cortex-A5 core is the smallest and lowest power member of the Cortex A series, it offers the possibility of multicore performance and is compatible with the larger members of the series (A9 and A15). The A5 is a natural choice for designers who have previously worked with the ARM926EJ-S or ARM1176JZ-S processors as it enables higher performance and lower silicon cost.

The Cortex-A7 is similar in power consumption and area to the Cortex-A5 but brings a performance increase in the range of 20 percent as well as full architectural compatibility with the Cortex-A15 and

Cortex-A17. The Cortex-A7 is an ideal choice for cost-sensitive smartphone and tablet implementations, and it can also be combined with a Cortex-A15 or Cortex-A17 in what ARM refers to as a "big.LITTLE" processing configuration. The big.LITTLE configuration is essentially a power optimization technology; a high-performance CPU (e.g., Cortex-A17) and an ultra-efficient CPU (e.g., Cortex-A7) are combined to provide higher sustained performance and also to enable significant overall power savings by relying on the more efficient core in cases of low to moderate performance requirements from the application, saving potentially 75 percent of CPU energy and as such extending battery life. This configuration offers a significant advantage to the developer as the performance demands of smartphones and tablets is advancing much faster than the capacity of batteries can keep pace. Design methodologies such as big.LITTLE, as part of an overall system design strategy, can significantly help reduce this battery technology gap.

Moving to the other end of the Cortex-A scale, let's consider the Cortex-A15 and Cortex-A17 cores. These are both very high-performance processors and again are available in a variety of configurations. The Cortex-A17 is the most efficient "mid-range" processor, and it squarely targets premium smartphones and tablets. The Cortex-A9 has been widely deployed in that market, but the Cortex-A17 offers an increase of more than 60percent (cycle for cycle) compared to the Cortex-A9 and achieves this performance while also improving overall power efficiency. The Cortex-A17 can be configured with up to four cores, each of which contains a fully out-of-order pipeline. As mentioned previously, the Cortex-A17 can be combined with the Cortex-A7 for an effective big.LITTLE configuration, and it can also be combined with high-end mobile graphics processors (such as the MALI from ARM), resulting in a very efficient design overall.

The Cortex-A15 is the highest performance member of this series, providing (in a mobile configuration) twice the performance you would get from a Cortex-A9. While being perfectly adequate in applications such as high-end smartphones or tablets, a multi-core Cortex-A15 processor running at 2.5 GHz opens up the possibility of using a Cortex-A processor in applications such as low-power servers or wireless infrastructure. The Cortex-A15 is the first processor from ARM to incorporate hardware support for data management and arbitration of virtualized software environments. Applications in those software environments are able to simultaneously access the system capabilities, making it possible to implement devices with virtual environments that are robust and isolated from each other.

The latest additions – the Cortex-A50 series – extend the reach of the Cortex-A series into low-power servers. These processors are built on the ARMv8 architecture and bring with them support for AArch64 – an energy-efficient 64-bit execution state that can operate alongside the existing 32-bit execution state. An obvious reason for the move to 64-bit is the support of more than 4GB of physical memory, which is already achieved on Cortex-A15 and Cortex-A7. In this case, the move to 64-bit is really about providing better support for server applications where a growing number of operating system and application implementations are using 64-bit, and the Cortex-A50 series delivers a power optimized solution for this scenario. The same is largely true for the desktop market, and support for 64-bit will enable the CortexA50 series to be more broadly adopted into this segment and will provide some level of future-proofing for the eventual migration of 64-bit operating systems into mobile applications.

Cortex-R

Moving on from Cortex-A, the Cortex-R series is the smallest ARM processor offering in terms of derivatives and possibly the least well known. The Cortex-R processors target high-performance real-time applications such as hard disk controllers (or solid state drive controllers), networking equipment and printers in the enterprise segment, consumer devices such as Blu-ray players and media players, and also automotive applications such as airbags, braking systems and engine management. The Cortex-R series is similar in some respects to a high-end microcontroller (MCU) but targets larger systems than you would typically use a standard MCU. The Cortex-R4, for example, is well suited for automotive applications. It can be clocked up to 600 MHz (delivering 2.45 DMIPS/MHz), has an 8-stage pipeline with dual-issue, pre-fetch and branch prediction and a low latency interrupt system that can interrupt multi-cycle operations to quickly serve the incoming interrupt. It can also be implemented in a dual-core configuration with the second Cortex-R4 being in a redundant lock-step configuration with logic for fault detection making it ideal for safety critical systems.

Networking and data storage applications are well served by the Cortex-R5, which extends the feature set offered by the Cortex-R4 to offer increased efficiency and reliability and enhance error management in dependable real-time systems. One such system-level feature is the low latency peripheral port (LLPP) to enable fast peripheral reads and writes (instead of having to perform a read-modify-write on the entire port). The Cortex-R5 can also be implemented as a "lock-step" dual-core system with the processors running independently, each executing its own programs with its own bus interfaces, and interrupts. This dual-core implementation makes it possible to build very powerful, flexible systems with real-time responses.

The Cortex-R7 significantly extends the performance reach of the series, with clock speeds in excess of 1 GHz and a performance of 3.77 DMIPS/MHz. The 11-stage pipeline on the Cortex-R7 now adds out-of-order execution along with improved branch prediction. There are several options for multi-core implementations as well: lock-step, symmetric multi-processing and asymmetric multi-processing. The Cortex-R7 also has a fully integrated generic interrupt controller (GIC) supporting complex priority-based interrupt handling. It is worth noting, however, that despite its high-performance levels, the Cortex-R7 is it not suitable for running rich operating systems (such as Linux and Android), which remains the domain of the Cortex-A series.

Cortex-M

Finally we come to the Cortex-M series, designed specifically to target the already very crowded MCU market. The Cortex-M series is built on the ARMv7-M architecture (used for Cortex-M3 and Cortex-M4), and the smaller Cortex-M0+ is built on the ARMv6-M architecture. The first Cortex-M processor was released in 2004, and it quickly gained popularity when a few mainstream MCU vendors picked up the core and started producing MCU devices. It is safe to say that the Cortex-M has become for the 32-bit world what the 8051 is for the 8-bit – an industry-standard core supplied by many vendors, each of which dip the core in their

own special sauce to provide differentiation in the market. The Cortex-M series can be implemented as a soft core in an FPGA, for example, but it is much more common to find them implemented as MCU with integrated memories, clocks and peripherals. Some are optimized for energy efficiency, some for high performance and some are tailored to a specific market segment such as smart metering.

The Cortex-M3 and Cortex-M4 are very similar cores. Each offers a performance of 1.25 DMIPS/MHz with a 3-stage pipeline, multiple 32-bit busses, clock speeds up to 200 MHz and very efficient debug options. The significant difference is the Cortex-M4 core's capability for DSP. The Cortex-M3 and Cortex-M4 share the same architecture and instruction set (Thumb-2). However, the Cortex-M4 adds a range of saturating and SIMD instructions specifically optimized to handle DSP algorithms. For example, consider the case of a 512 point FFT running every 0.5 second on equivalent off-the-shelf Cortex-M3 and Cortex-M4 MCUs. For comparison, the Cortex-M3 would consume around three times the power that a Cortex-M4 would need for the same job. There is also the option to get a single precision floating point unit (FPU) on a Cortex-M4. If your application requires floating point math, you will get this done considerably faster on a Cortex-M4 than you will on a Cortex-M3. That said, for an application that is not using the DSP or FPU capabilities of the Cortex-M4, you will see the same level of performance and power consumption on a Cortex-M3. In other words, if you need DSP functionality, go with a Cortex-M4. Otherwise, the Cortex-M3 will do the job.

For applications that are particularly cost sensitive or are migrating from 8-bit to 32-bit, the smallest member of the Cortex-M series might be the best choice. The Cortex-M0+ performance sits a little below that of the Cortex-M3 and Cortex-M4 at 0.95 DMIPS/MHz but is still compatible with its bigger brothers. The Cortex-M0+ uses a subset of the Thumb-2 instruction set, and those instructions are predominantly 16-bit operands (although all data operations are 32-bit), which lend themselves nicely to the 2-stage pipeline that the Cortex-M0+ offers. This brings some overall power saving to the system through reduced branch shadow, and the pipeline will in most cases hold the next four instructions. The Cortex-M0+ also has a dedicated bus for single-cycle GPIO, meaning you can implement certain interfaces with bit-bashed GPIO like you would on an 8-bit MCU but with the performance of a 32-bit core to process the data.

Another key difference on the Cortex-M0+ is the addition of the micro trace buffer (MTB). This peripherals allows you to dedicate some of the on-chip RAM to store program branches while in debug.– These branches can then be passed back up to the integrated development environment (IDE), and the program flow can be reconstructed. This capability provides a rudimentary form of instruction trace and compensates for not having the extended trace macrocell (ETM) found on the Cortex-M3 and Cortex-M4. The level of debug information you can extract from a Cortex-M0+ is significantly higher than that which you can get from an 8-bit MCU, meaning those hard to solve bugs just got easier to fix.

Conclusion

In summary, the Cortex processor family offers many options regardless of the performance level you need for your application. With a little bit of thought and investigation, you will be able to find the right processor that suits your application needs, whether it's for a high-end tablet or an ultra-low-cost wireless sensor node for the Internet of Things.

Making Electronics Smart, Connected, and Energy Friendly

Silicon Labs (NASDAQ: SLAB) is a leading provider of silicon, software and system solutions for the Internet of Things, Internet infrastructure, industrial control, consumer and automotive markets. We solve the electronics industry's toughest problems, providing customers with significant advantages in performance, energy savings, connectivity and design simplicity. Backed by our world-class engineering teams with unsurpassed software and mixed-signal design expertise, Silicon Labs empowers developers with the tools and technologies they need to advance quickly and easily from initial idea to final product.

Learn more about Silicon Labs' ARM Microcontroller solutions at www.silabs.com/32bit-mcu