White Paper

Exploring the ARM® Cortex®-M7 Core: Providing Adaptability for the Internet of Tomorrow

Introduction

The momentum behind embedded intelligent connectivity continues to increase at a phenomenal rate. Technologies are becoming connected across multiple devices, platforms and networks, creating a web of communication that is augmenting the way we interact with the digital landscape. This so-called Internet of Things (IoT) is changing how we interact with our environment, our communities, our homes and even our bodies. It is estimated that wireless connected devices will reach over 40.9 billion by 2020.[1] To reach this astounding figure, IoT applications will grow and evolve, ranging from increasingly complex activity monitors worn on the body, to intelligent home appliances capable of optimizing energy efficiency. To support such growth and variation, there are a number of challenges that have to be faced by the embedded technology supporting the IoT.

The challenges facing embedded designers are being continuously amplified by the IoT's rapid growth rate. As this web of connectivity expands, engineers face growing and changing demands, some of which can be seen in 'Figure 1' on page two. Designers are working hard to reduce their time-to-market and thus stake their claim in the developing segments of the IoT space. On top of this, there are increasing requirements for end user expectations. As the applications within IoT are broad, end products must find the right balance of processing capability and power consumption. Of course, this is not a new challenge for hardware developers, but it is one that is particularly important in the embedded sphere. In addition, end applications have to be adaptable, having the capability to change and expand with software updates. There is also the expectation that end devices will be secure from malicious attacks, leading to greater integrity requirements to ensure that devices operate as expected.

Table of Contents

freescale™

ARM®

IoT Challenges

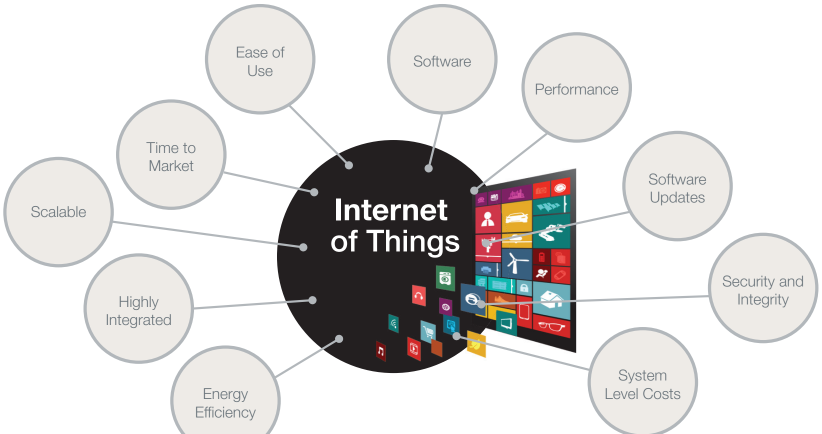
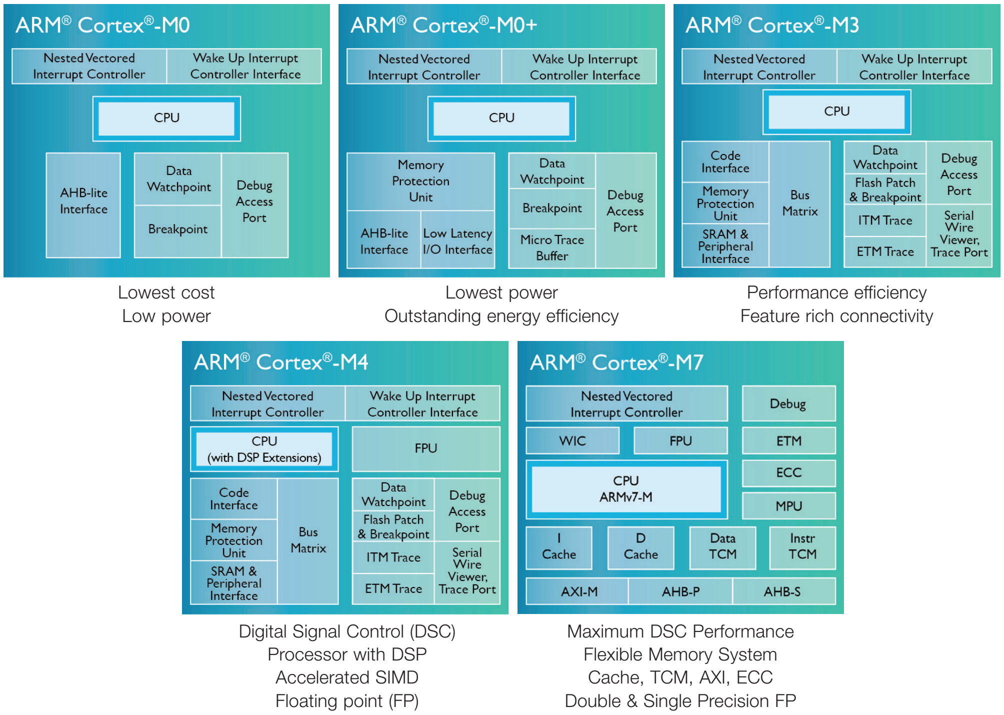


Figure 1: Challenges of the Growing IoT Space

ARM Cortex-M Processor Family

A leading architecture for embedded controllers is the ARM Cortex-M processor family. These CPUs provide the processing needed for vast numbers of IoT applications. This scalable processor family line spans from the ultra-low power, lowest cost solutions to the maximum performance and highly configurable ARM Cortex-M7 processor. All members of the ARM Cortex-M processor family are aligned to meet the IoT challenges, with instruction set upward compatibility, and performance efficiency to complement different IoT use cases. Running on top of this hardware is the recently announced mbed™ OS platform, and the Cortex-M Microcontroller Software Interface Standard (CMSIS), which is a vendor-independent hardware abstraction layer for the Cortex-M processor series. This software compatibility and the supporting ecosystem of developers and support means that ARM Cortex-M processors are the trusted choice for a range of embedded markets, including the IoT.

ARM Cortex-M Processor Family



Lowest cost
Low power

Lowest power
Outstanding energy efficiency

Performance efficiency
Feature rich connectivity

Digital Signal Control (DSC)
Processor with DSP
Accelerated SIMD
Floating point (FP)

Maximum DSC Performance
Flexible Memory System
Cache, TCM, AXI, ECC
Double & Single Precision FP

Figure 2: ARM Cortex-M Processor Family

The most recent member of the ARM Cortex-M processor family is the ARM Cortex-M7 processor. The new Cortex-M7 core offers capabilities that can be leveraged to support the needs of the new embedded technology supporting the expansion of the IoT. It is designed for applications that demand high-processing performance, real-time response capability and energy efficiency. At a high level, the Cortex-M7 processor includes the following key features:

- High-performance, 6-stage pipeline with dual-instruction issue, enabling it to execute up to two instructions per clock cycle
- A 64-bit AXI system bus interface
- Optional instruction cache (4 to 64KB) and data cache (4 to 64KB), with optional ECC (Error Correction Code) support for each of the cache memories
- Optional 64-bit Instruction Tightly Coupled Memory (ITCM), and optional dual 32-bit Data TCM (D{0,1}TCM), with support for customer ECC implementation for each TCM memory arrays
- Optional low latency AHB peripheral bus interface allows deterministic and fast access to peripherals in real-time applications.

The ARM Cortex-M7 processor is highly configurable, enabling chip designers to align the core with specific application tasks. This gives flexibility to engineers and developers, and provides a range of solutions that match specific challenges such as those that are present in the IoT.

This whitepaper will elaborate on the ARM Cortex-M7 processor in greater detail, discussing the architectural considerations faced when designing the core as well as information about the configuration options and why this matters in the context of the IoT. This information will be relevant for those looking to learn more about the processor itself, or for those interested in embedded system design and software development.

The ARM Cortex-M7 Processor

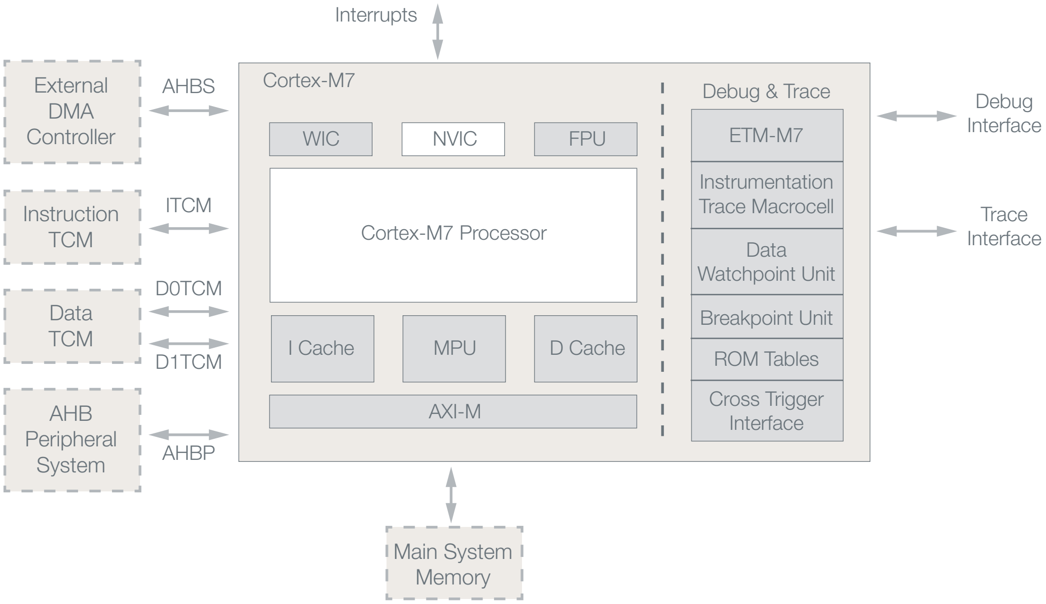


Figure 3: The ARM Cortex-M7 Processor

ARM Cortex-M7 Processor Configuration Options

The Cortex-M7 processor's microarchitecture is different from the other members of the Cortex-M processor family. With its 6-stage superscalar pipeline implementation, the Cortex-M7 microarchitecture provides a significant improvement in system performance through both the improved architecture performance (reduced cycles per instruction) and the increase in frequency of operation. To support the higher instruction and data bandwidth requirements of a superscalar design, the key memory interfaces are designed with a 64-bit width. Both the AXI system bus and the single-cycle ITCM interfaces are 64-bit, and the dual 32-bit D-TCM interface can handle two 32-bit transfers, or a 64-bit data transfer in a single cycle. The table below summarizes the buses in the microarchitecture of the Cortex-M7 processor, emphasizing the new interfaces versus previous ARM Cortex-M family devices.

ARM Cortex-M7 Bus Types and Descriptions

Bus Type	Description
New: ITCM	Instruction tightly coupled memory (TCM) – a 64-bit bus optimized for instruction accesses on the Cortex-M7 processor. TCM interface can integrate with on-chip memory arrays easily, and the direct connection path means that the transfers are not affected by traffic on the other bus interfaces.
New: D{0,1}TCM	Two 32-bit TCM interfaces that are optimized for various types of data accesses. It supports two simultaneous word data transfers at the same time, which enables the Cortex-M7 processor to achieve excellent performance in many control and DSP applications. Similar to ITCM, it enables easy integration with on-chip SRAM, and supports maximum data bandwidth.
New: AXI master (AXI-M) interface	64-bit advanced extensible interface (AXI) allows the Cortex-M7 processor to be connected to complex memory systems. The AXI bus is a split-transaction protocol and supports multiple outstanding transfers (transfer commands can be issued before the previous ones are finished). It also allows higher operation clock frequency even for complex memory systems.
AHB peripheral (AHBP) interface	The 32-bit AHB peripheral interface supports low latency transfers between the processor and peripherals. The nature of the 2-stage pipelined AHB allows easier integration with very small silicon overhead, and enables existing peripherals designed for other Cortex-M processors to be easily reused.
AHB slave port (AHBS)	The 32-bit AHB slave port allows an external bus master (e.g. DMA controller) to access to the TCM connected to the Cortex-M7 processor.
AHB debug port	The 32-bit AHB debug port allows the debug interface module (e.g., JTAG/Serial wire interface) to access all of the debug features on the Cortex-M7 processor.

Table 1: ARM Cortex-M7 Bus Types and Descriptions

The AXI master interface is a crucial feature in supporting the memory expandability needed for many IoT applications. As new usage models are built from data that is continually gathered and analyzed, having the capability to leverage external memories to add functionality is a key feature. In addition to the AXI master interface, the TCM interfaces provide the optimal single-cycle interface for performing the real-time operations needed for control. The high-performance memory and bus interfaces are critical to supporting the greater than 5 CoreMarks®/MHz performance level of the processor.

There are a number of key areas to consider when choosing which buses to use in an SoC and how they will be utilized, including:

- Which peripherals require connection to the AHB peripheral bus on the Cortex-M7 processor for low access latency?
- Which peripherals need to be accessed by the DMA controller?
- What forms of access control and memory protection are needed?

A very simple design, for example, can have the memory system attached to the TCM interface, and the peripherals attached to the AHB peripheral interface as shown in the diagram below. This configuration allows an SoC to take advantage of the scalable performance of the ARM Cortex-M7 core and still meet the challenges associated with cost and size. For example, this particular setup would enable a control edge node that needs the real-time performance supported by the SRAM connected to the TCM interfaces.

Minimal Microcontroller

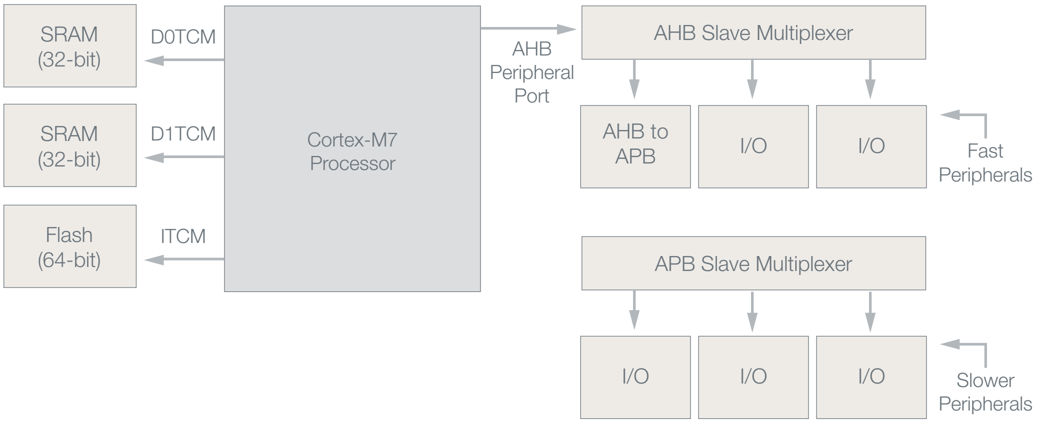


Figure 4: Minimal Microcontroller

Another configuration option is to connect the embedded memory and/or external memory to the AXI interface and use cache memories to enable better performance. Most microcontroller applications contain many small control loops and therefore have firmware execution with a very low number of cache misses. With a cache-based design, the system may have less determinism when executing programs from the AXI bus system. However, the exception vector table and the interrupt handlers can be placed into SRAM connected to the ITCM interface, thus enabling deterministic behaviors when executing interrupt handlers.

Microcontroller With External Memory

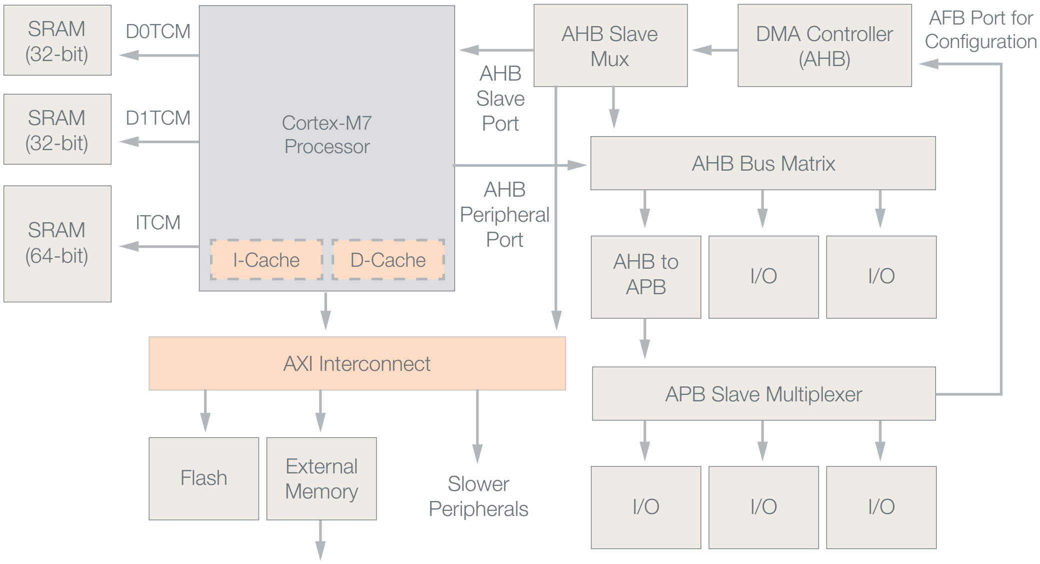


Figure 5: Microcontroller With External Memory

The memory expandability, performance and efficiency benefits of the AXI interface and caches are key to meeting application needs. These types of configuration offer a number of advantages aligned to IoT applications, such as support for over-the-air firmware updates and data storage needs that leverage large external memories. However not every option is needed for all application cases, and so the challenges around cost, size and power must be considered.

The memory system is designed to offer a variety of different configurability options. There are multiple areas and factors to consider including:

- Execution from AXI or TCM interface
- Cache sizes (if using AXI)
- Method for embedded flash access accelerations and width of the flash memory
- Optional ECC support

The decisions can be affected by many different factors such as the read access speed of the embedded flash, the clock speed requirement, and the typical size of the targeted applications and its program flow behaviors.

If the embedded flash memory access speed is close to the required processor speed, the embedded flash can be connected to the ITCM interface with some flash access acceleration. In other cases, using AXI with caches would be more suitable.

If the application needs to execute programs from an external memory controller, then normally the memory controller would be connected to the AXI interface and the support for instruction cache and data cache would be required. In some cases, the application might only need to use the external memory for data storage. In such cases the instruction cache is not needed.

Choosing cache size depends heavily on the attributes of the application code. When running program code from embedded flash, both instruction cache and data cache are leveraged, as the program image typically contains literal data, look up tables or read-only constants, along with instructions.

Applications typically have more instruction words than data/constants inside the program image. As the program size gets bigger, the cache requirement also increases, and it is not uncommon to have instruction cache larger than data cache.

Conversely some applications might have small control or DSP loops, and may have a large amount of data for coefficients for the calculations. In these cases, the performance of the system could benefit from a larger D-cache rather than a larger I-cache.

Of course, when optimizing for performance, maximizing the cache to ensure the lowest latency for larger code and data sizes is desired. By running a large cache memory at the same speed as the processor, however, the cache look up may consume substantial amounts of power depending on other factors. Additionally, cache miss ratio curves for most applications approach zero as the size increases, meaning further increases of the cache size do not increase performance. Fortunately, the configurability of the ARM Cortex-M7 core allows SoC architects to integrate a range of cache sizes, going from no cache, up to 64KB of instruction and 64KB of data cache. With this flexibility, designers can tune the SoC to meet the needs of the targeted application.

Aside from the architecture options, there are many other features on the Cortex-M7 processor that are configurable. For example, the floating-point unit (FPU) feature of the SoC can be configured so as to not have an FPU at all, to have an FPU with IEEE®-754 single-precision floating-point operations, or to have an FPU with both IEEE-754 single-precision and double-precision floating-point operations.

Other configuration features include:

- Number of interrupts and number of priority levels in the NVIC
- Memory Protection Unit (MPU) configuration
- Debug and trace features
- Functional safety related features (ECC, dual-core lock-step)

The hardware acceleration for floating-point operations offers several benefits. Obviously the performance of floating point operations is accelerated with the availability of a hardware floating point unit. In addition, memory space is optimized as hardware support reduces the number and associated size of software libraries to perform floating-point operations. Reducing processing time and memory footprint ultimately improves the energy efficiency of the application and smoothes the path to performing functions that traditionally need more complex embedded systems. The benefit can be significant to energy efficiency as floating-point operations for DSP filters can be up to 20x faster. Having both single and double precision floating-point capability options further improves the scalability of the new processor.

As technology continues to be expanded by the IoT trends, there is the growing need to meet the challenge of security and integrity of embedded applications. In addition to fault exception features and a memory protection unit as in other Cortex-M processors, the Cortex-M7 processor also includes optional Error Correction Code (ECC) support for TCM memories and caches. This enables automatic on-the-fly correction of single bit errors in the memory, and detection of double bit errors.

In addition, the Cortex-M7 processor supports a dual-core lock-step configuration option. In this configuration, the core logic is instantiated twice, but the cache and TCM memory arrays are shared. This is because they can be protected by ECC, which has a significantly lower silicon area overhead (see figure below), enabling a highly robust, fault tolerant system design. This enables a highly robust, fault tolerant system design.

ARM Cortex-M7 Processor Dual-Core Lock-Step Configuration

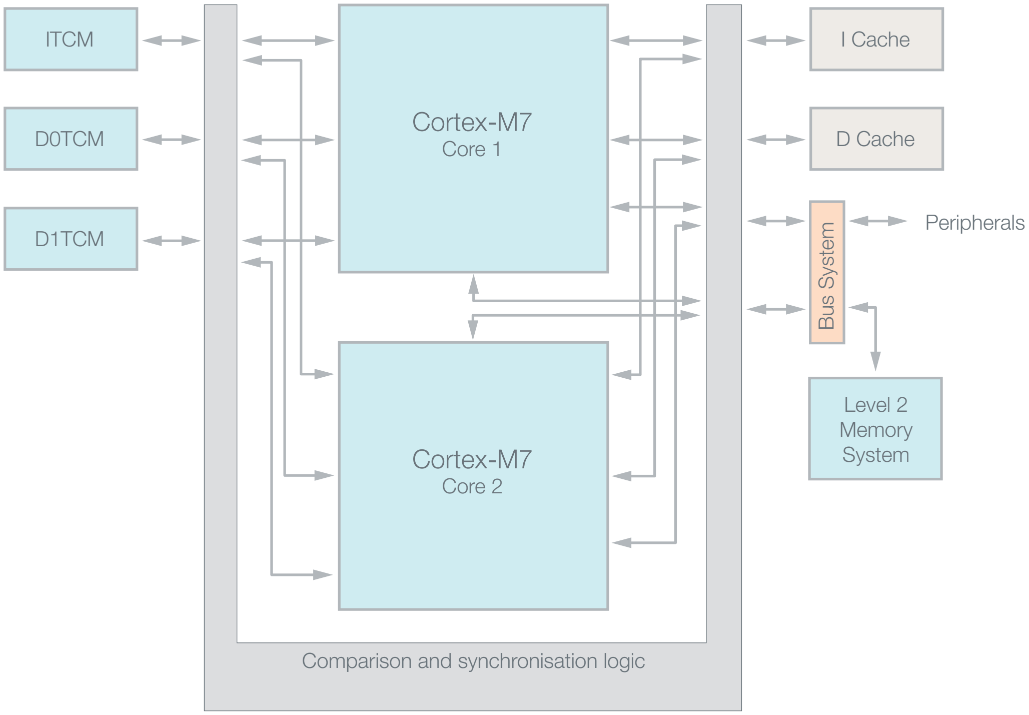


Figure 6: ARM Cortex-M7 Processor Dual-Core Lock-Step Configuration

Implementation of the Kinetis KV5x MCU Family from Freescale Semiconductor

One example of the implementation choices of the Cortex-M7 processor is the newly-announced Kinetis KV5x MCU family from Freescale Semiconductor. The Kinetis KV5x MCU family is the newest member of the Kinetis V series, a scalable line of MCUs targeting motor control and digital power conversion applications. Kinetis V series MCUs couple the ARM Cortex-M processor cores with the peripherals and enablement needed to provide a path for all types of embedded innovators to quickly produce control solutions. Within the Kinetis V series, there are SoCs that use the Cortex-M0+, Cortex-M4 and now the ARM Cortex-M7 core. This addition to the Kinetis V series is designed to meet the challenge of supporting both real-time control and Ethernet connectivity with a single SoC. The key features of this device are a 10/100 Ethernet MAC supporting IEEE-1588, four 5 Megasample per second analog to digital converters, multiple PWM control blocks with picoseconds resolution, and hardware acceleration for cryptographic operations.

Kinetis KV5x MCU Family Block Diagram

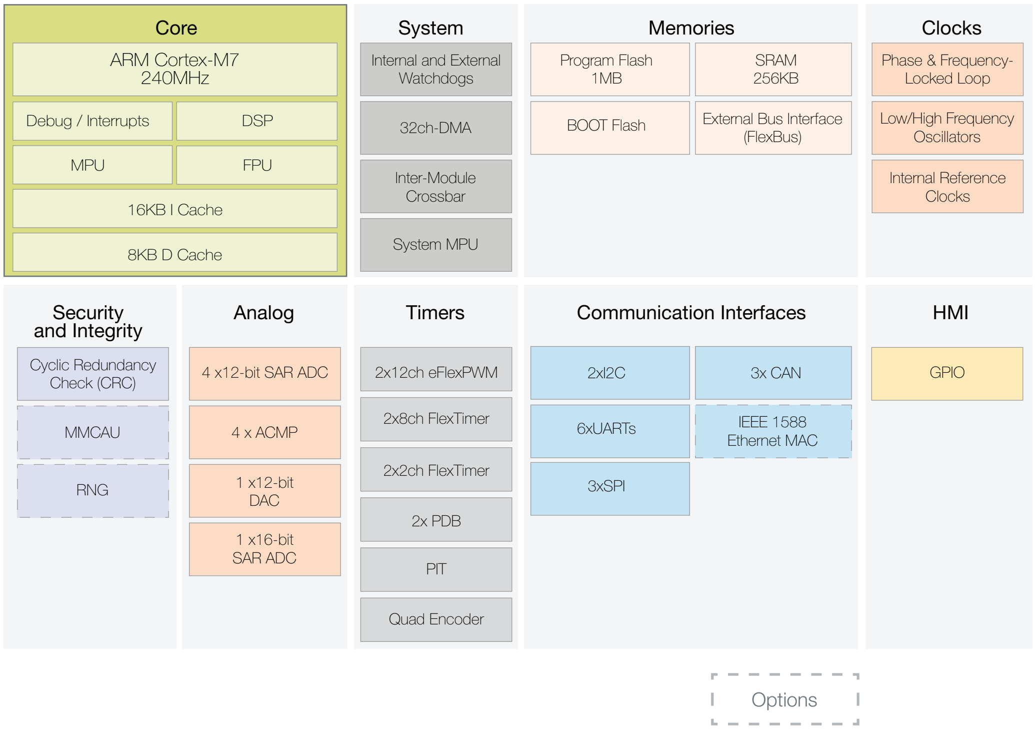


Figure 7: Kinetis KV5x MCU Family Block Diagram

Some of the configuration options selected for the ARM Cortex-M7 processor in this SoC include the integration of 16KB of instruction cache and 8KB of data cache. This SoC utilizes the 64-bit AXI bus as an access port to the embedded flash memory. The instruction and data caches ensure that the control software that resides in the embedded memory is accelerated to support the performance levels needed for the connected industrial control use case. In addition to the caches, the Kinetis KV5x MCU family integrates 64KB of SRAM connected to the ITCM interface and 128KB of SRAM connected to the DTCM interface. This provides the processor-local storage needed to support the real-time control operations with the lowest latency memory.

Kinetis KV5x MCU Family Implementation

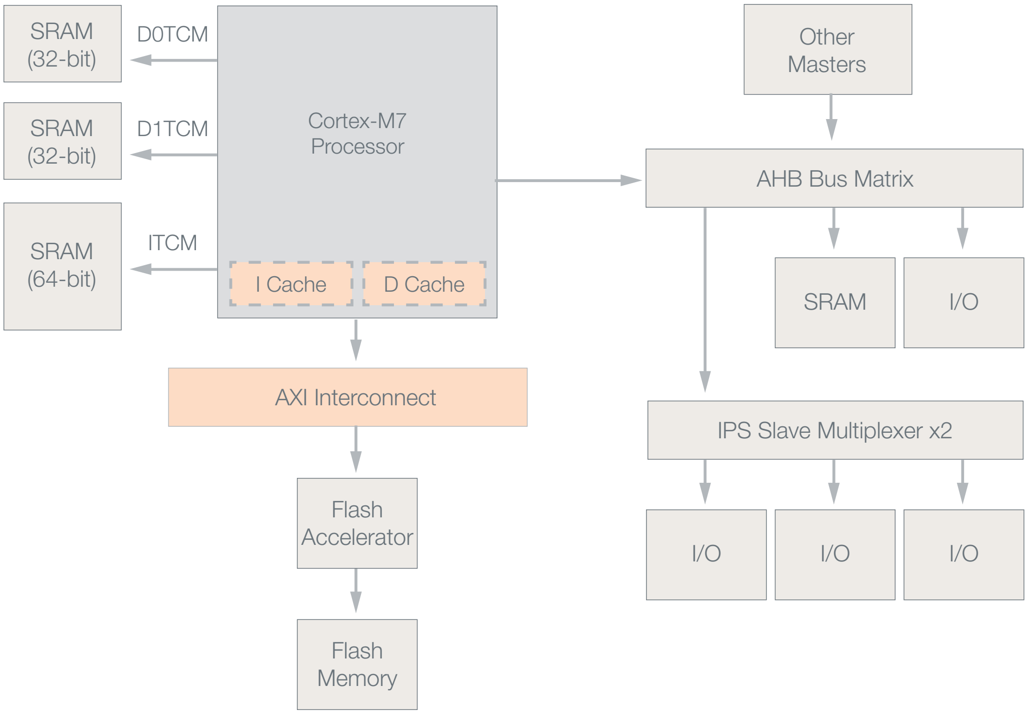


Figure 8: Kinetis KV5x MCU Family Implementation

The Kinetis KV5x MCU configuration is just one example of how an SoC can be built to align for a particular application focused on connected control. As time progresses and the number and variation of connected applications grow, there are certain to be adaptations of SoC designs that leverage a wider range of configurations of the ARM Cortex-M7 processor. The increase to performance levels that will be more than two times the performance of comparable Cortex-M4 solutions is sure to lead to embedded innovations. With its scalability, performance and expandability features, the adaptable ARM Cortex-M7 core will be playing a key role in supporting the Internet of Tomorrow.

Contributors

This whitepaper was collaboration among experts from Freescale and ARM. Authors include:

- Donnie Garcia, Kinetis MCU Product Marketer, Freescale
- Joseph Yiu, Senior Embedded Technology Manager, ARM

References

1. The IoT effect: 40.9 billion wireless connected devices by 2020 | *First Post* | August 2014

**For more information about Freescale and ARM,
visit freescale.com and arm.com**