ARM

White paper

Designing a System-on-Chip (SoC) with an ARM Cortex-M processor

A starter guide

Joseph Yiu

February 2017

1. Background

Since the ARM Cortex-M0 Processor was released a few years ago, the number of silicon designs based on ARM Cortex-M Processors has increased substantially. By the end of 2016, it was reported that there were over 400 Cortex-M licensees, with most of these licensees using Cortex-M processors in non-MCU products. In fact, Cortex-M processors are finding their ways into new SoC designs whose previous generations did not include a Cortex-M, let alone a digital processor core. As a result, many SoC design engineers have taken the exciting journey of integrating a Cortex-M processor core for the first time without previous experience or any knowledge of important considerations required.

This whitepaper seeks to answer some of the frequently asked questions SoC designers face when integrating an ARM Cortex-M processor for the first time.

2. What is in an ARM Cortex-M based system?

In a typical SoC with ARM Cortex-M processor, for example, an off-the-shelf microcontroller, there are the following components:

Digital system components

- An ARM Cortex-M processor
- AHB and APB bus infrastructure components
- Digital peripherals, such as I2C/I3C, SPI interface
- Optionally - system IPs such as:
    - DMA (Direct Memory Access) controller
    - Crypto engine
    - True Random Number Generator (TRNG) for communication session keys
    - Secure data storage
    - Debug authentications, etc

Memories

- Non-volatile memory (NVM), for example, Flash memory, OTP (One Time Programmable) memory or mask ROM
- Static Random Access Memory (SRAM)
- Optional boot loader memory (NVM).

Optional analogue components

- In certain applications analogue peripherals such as ADC, DAC, voltage reference, brown-out detectors, voltage regulator, etc

Wireless interface

- Some modern SoC also included on-chip wireless interface such as Bluetooth, Zigbee, etc.

System components

- Clock management functions: crystal oscillators, phase locked loop (PLL),

Other physical IP

- Standard cell libraries, clock gating and power gating cells, I/O pads

In some cases, a special peripheral interface would also require special physical IP, such as a USB I/O pad.

3. Software and Tools

Various EDA and software tools are required to start your project. ARM Cortex-M processors are delivered in generic Verilog RTL source, and can work with all major EDA synthesis tools such as the Cadence Encounter RTL Compiler, Virtuoso development environment, Synopsys Design Compiler and various FPGA design environments. In addition to simulation and synthesis tools, floor planning, timing analysis, power analysis, logic equivalent checking (LEC) tools would also be required.

For software development, there are many choices thanks to the vast ecosystem for ARM software development. In addition to the ARM Design Studio 5 (DS-5, reference 1) and Keil Microcontroller Development Kit (MDK, reference 2) from ARM, there are a number of third party software development tools (including gcc) that can be used for software compilation and system testing. A debug adaptor, such as the D-Stream or Keil ULINK*Pro* enables testing of software on target hardware, including on FPGA prototyping platforms. FPGA prototyping platforms are commonly used to allow early software development and testing on real hardware prior to the availability of the Cortex-M based SoC.

4. Overall digital system design

4.1 The ARM Cortex-M Processor

ARM Cortex-M processors are delivered as Verilog RTL source code. They are very configurable – a number of Verilog parameters are available to allow designers to select features required for each specific design. For example, the Memory Protection Unit (MPU) may not be required in some applications and can be omitted. Other configurable options include the number of interrupts and the amount of debug features included in the processor core. All Cortex-M processors are designed for easy implementation:

- Processor and bus logic use only rising edge flip-flops without internal latches or any structure that are problematic for scan tests
- No dependency to any special EDA tool or cell library

Most Cortex-M processors also include an integration kit which allows designers to carry out system level simulations out-of-the-box, supporting Mentor Modelsim/Questasim, Synopsys VCS, and Cadence NCSim. The Cortex-M3 Processor deliverable has an older style testbench for system level simulations and a separate testbench for debug integration testing. There is also support for vector capture and replay testbench in all of the Cortex-M processor deliverables. The Cortex-M7 processor deliverable also have additional test environment for cache SRAM integration check.

ARM provides example synthesis scripts for Synopsys and Cadence tools, but there is no limitation on which EDA tool is required because the Cortex-M processors are designed with generic Verilog RTL, and can be targeted to almost any semiconductor process as well as FPGA.

In order to help companies to accelerate their design investigations and product development processes, ARM introduced:

- Cortex-M0 DesignStart – a free-of-charge Cortex-M0 evaluation package to enable chip designs to create proof-of-concept systems (reference 14).
- Low cost FPGA prototyping – chip designers can use a low cost Cortex-M System Prototyping System (reference 15) to verify their system design.
- $40K Fast Track license program for Cortex-M0 – this licensable product provides the full Cortex-M0 processor in RTL source form, and include a range of system IP in the Cortex-M System Design Kit (CMSDK) to enable faster time to market (also see reference 14).
- CoreLink™ SSE Subsystem for Embedded – a pre-verified subsystem based on Cortex-M processors that designers can use as a foundation for their SoCs. Subsystems include hardware and software to help designers accelerate their SoC developments. ARM CoreLink™ SSE-100 and SSE-200 are currently available and based on the Cortex-M3 and Cortex-M33 processors respectively. The subsystems provide the fastest and lowest risk path to silicon

These offerings enable a risk-free design evaluation process and ideal project starting point for small SoC designs with the Cortex-M0 processor. For example, the Cortex-M0 processor can easily be used for smart sensors (reference 23), MEMS devices, motor controllers, and low cost microcontroller products. For other Cortex-M processors, a wide range of low cost microcontroller development boards from various microcontroller controller vendors can be used for various software evaluations and testing.

4.2 Bus interconnect and reference designs

The Cortex-M processors connect to the rest of the system via AHB (Advanced High-performance Bus), APB (Advanced Peripheral Bus), and in the case of the Cortex-M7 processor, an AXI (Advanced eXtensible Interface) interface. These all are part of the AMBA (Advanced Microcontroller Bus Architecture) specifications and are open to public access though ARM website (requires a simple registration process). These on-chip bus protocols are widely used in the chip design industry and AMBA compliant system IPs can be licensed from a number of silicon IP providers.

The addressable memory space in Cortex-M processors is 4GB (32-bit address), however, a typical SoC implementation would only require a range of several Kbytes to several hundred Kbytes of physical memory space, depending on the application.

Comparing AXI to AHB, AXI support multiple outstanding transfers and can operate at higher clock frequency, thus enabling higher performance. For most other Cortex-M processors, AHB interface are used for system buses because AHB system designs are simpler and are usually smaller and lower power.

For Cortex-M0, Cortex-M0+, Cortex-M3, Cortex-M4, Cortex-M23 and Cortex-M33 processors, memories and peripherals are connected to the Cortex-M processor via AHB protocol. The address space partitioning and memory sizes are defined by AHB address decoders at system level. Due to the simple nature of the bus protocols, AHB based systems allows easy memory map customization. Bus interfaces are 32-bit wide on these processors, but a range of bus bridges are available from ARM to convert bus widths as required.

For the Cortex-M7 processor, memory macros can be connected to the processor via Tightly Coupled Memory (TCM) interface or the 64-bit AXI (Advanced eXtensible Interface) interface. The memory space allocation accessible from AXI bus is defined in the AXI interconnect IP (e.g. CoreLink NIC-400, reference 8), which is configured using system

design tool (reference 9). In addition, the Cortex-M7 processor also support an optional peripheral bus based on AHB Lite protocol to enable easier system peripheral migration.

There are two releases of AHB protocol specifications:

- AMBA 3 AHB Lite – for Cortex-M0, Corte-M0+, Cortex-M3, Cortex-M4, Cortex-M7 (peripheral bus) processors.
- AMBA 5 AHB – for Cortex-M23, Cortex-M33 processors.

A range of ARM system IP products are available to support different processor design requirements:

Product	Bus fabric components	Included example system(s)	Additional info
Cortex-M System Design Kit (CMSDK), ref 3	AHB Lite, APB	Cortex-M0, Cortex-M0+, Cortex-M3, Cortex-M4	CM0 SDK, a subset of CMSDK for Cortex-M0/M0+ designs is also available
CoreLink SSE-100, ref 6	AHB Lite, APB	Cortex-M3	IoT subsystem. Included CG092 AHB flash cache
CoreLink SIE-200, ref 4	AHB5	-	System IP for secure IoT platforms
CoreLink SSE-200, ref 7	AHB5	Cortex-M33	Advanced IoT subsystem with security extension.
CoreLink CG092 flash cache, ref 5	AHB Lite	-	Can also be used with AHB5 systems
CoreLink NIC-400, ref 8	AXI, AHB, APB	-	Requires tooling for configuration: CoreLink Creator(ref 9)

The ARM Cortex-M System Design Kit (CMSDK, reference 3) provides a wide range of AHB and APB components including an advanced bus bridge for handling multiple clock domains, and to allow mixing of 64-bit and 32-bit AHB systems.

Example systems with baseline peripherals (for example, timers, UART, parallel I/O interface module) are also included in the CMSDK, which allows designers to simulate a prebuilt system out of the box. The example system can be modified by the chip designers. For example, peripherals can be swapped to other designs and chip designers can add additional peripherals where applicable. For each example system, simulation scripts are included to support simulators from Mentor, Cadence and Synopsys.

The CMSDK also includes example software to support system level simulations. The software framework is based on CMSIS (Cortex-M Software Interface Standard, reference 18) which is used by Cortex-M microcontroller vendors.

The CMSDK also includes verification components such as protocol checkers (modules that monitor bus activities and can flag up bus protocol violations) and bus stimulus generators.

For designs based on the Cortex-M23 and Cortex-M33 processors, system IPs for AHB5 bus protocol are required and this is available from the CoreLink SIE-200 (reference 4). Similar to the CMSDK, this product contains a range of bus infrastructure components and bus bridges. Example system for Cortex-M33 is also available from CoreLink SSE-200 (reference 7).

For high performance designs based on the Cortex-M7, normally AXI interconnection components are needed, although Cortex-M7 product bundle also included an AXI to AHB Lite bridge if a customer need a simpler migration from AHB Lite based memory system. For system integration with AXI protocol, ARM offer a range of system interconnect IP such as the CoreLink NIC-400 (reference 8). For complex high performance SoC designs, ARM recommend designers to use system IP integration tools (refence 9) such as ARM Socrates to accelerate the design process and enable early error checking.

4.3 Peripherals

There are many suppliers of peripherals based on AHB and APB protocols. Designers might also have existing peripherals from a previous design based on legacy processor architectures. In such a case, the example AHB slave and example APB slaves in the CMSDK (ref 3) and CoreLink SIE-200 (ref 4) are good references for porting these peripherals to AHB and APB. The example slaves in the CMSDK/CoreLink SIE-200 contain a simple AHB/APB bus interface module, which enables most legacy peripherals to be connected to AHB/APB without any modifications (figure 1).

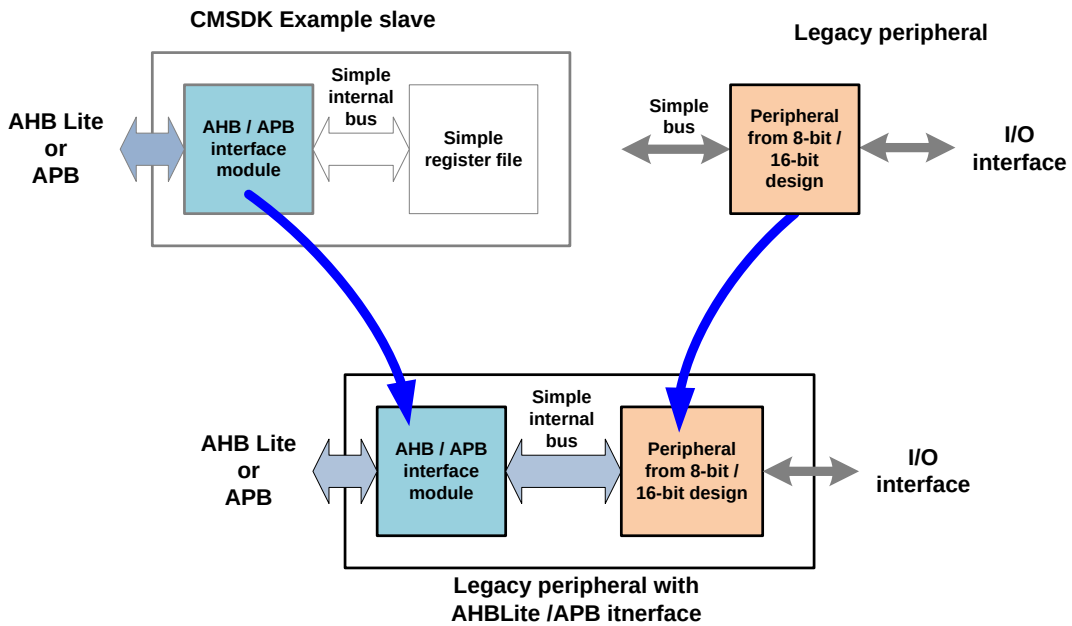

Figure 1 - Porting legacy peripherals to AHB/APB based system with example slaves from CMSDK/CoreLink SIE-200

The CMSDK also contains base line peripherals including GPIO (General Purpose Input / Output), Timers, UART and watchdog timer.

4.4 Additional system components

If additional system components, such as a DMA controller or external memory controller are required, there are a wide range of suppliers providing these solutions, including ARM. A range of DMA controllers (For example, DMA-230 micro DMA controller, and PL080, PL081 DMA controllers) for AHB based system, and various memory controllers in the PL24x product series, which support AHB on chip bus, are also available from ARM.

For system designs based on Cortex-M23 or Cortex-M33 that utilize TrustZone® security extension for system partitioning, addition system IP is required for memory partitioning (separating part of memory blocks into Secure space and Non-secure space), and peripheral partitioning (allocate each peripheral to Secure domain or Non-secure domain). The ARM CoreLink SIE-200 (ref 4) contains these memory protection controller and peripheral protection controllers (support AHB5 and APB), as well as other advanced system IP for low-power optimizations and multi-core designs.

In modern SoC designs, additional security IP might be required for:

- Cryptography acceleration
- Secure data storage
- Product life cycle security management
- Debug access permission control (debug authentication)

The ARM CryptoCell-300 family (reference 10) is one of such products to provide these security management solutions for Cortex-M (and Cortex-R) processor systems.

Increasingly we see more microcontroller products include wireless interface (such as Bluetooth, Zigbee) on chip. Such system IPs are available from a number of suppliers including ARM. ARM's Cordio product family (reference 11) supports Bluetooth (including latest Bluetooth 5), 802.15.4 (including Zigbee and Thread protocols) and NB-IoT. Wireless system integration typically involves a digital subsystem that is linked to the processor's memory system, and there are also wireless physical IPs (known as radio frequency frontend, e.g. amplifers) which are typically available as hard macros which are process dependent. The latest Cordio Bluetooth/802.15.4 wireless IP is designed to support multiple types of RF frontends in different processor nodes, and can operate with very little power thanks to sub 1 volt operation capability. When developing these systems, do not forget to look out for software solutions (e.g. protocol stack, application profiles), which are also available from ARM (reference 12).

5. On chip memories

5.1 On chip NVM memory and system integration

Various foundries have their own embedded flash solutions; licensable embedded flash memory macros are also available from a number of other companies for a range of semiconductor process nodes. For example, SONOS embedded flash from Cypress Semiconductor can now be licensed via UMC (reference 19).

Typically, on chip flash macros are available without an AHB interface:

- Some embedded flash vendors such as TSMC, provide application notes on how to interface their embedded flash with typical processor bus systems. ARM also have flash controller IP for TSMC 55ULP embedded flash.
- Some embedded flash vendors such as. Cypress Semiconductor, use a simple SRAM like interface. A simple bus adapter is required to connect the flash macro to AHB (more information in reference 19).

In many applications, the processor speed is higher than the access speed of the embedded flash. In such cases cache memory or prefetch logic could be needed to avoid stalling processor execution. In most cases cache solutions enable lower power (caches reduce the flash memory access which consume higher power), and the design can support good performance even if the flash is a lot slower than the processor. ARM CoreLink CG092 AHB flash cache (reference 5) is one of the cache solutions available, and is optimized for embedded flash integration.

Alternatively, flash prefetch logic can be used if the clock ratio between the processor and the flash is small (e.g. less than 4-to-1). Prefetch logic with 64-bit or even 128-bit embedded flash interface would be required with the flash data I/O width depend on the clock ratio. Additional details are available for ARM customers.

Some flash memory macros have various low power features. It is possible to connect the sleep mode control logic of the Cortex-M processor to the embedded flash macros to reduce sleep mode power. Alternatively, power gating header cell can be added to the power rails of the embedded flash to reduce sleep power through power gating.

If the program image is fixed and does not require updates during its lifetime, mask ROM can be used for program storage and a Via ROM compiler is available from ARM.

5.2 On chip SRAM and system integration

There are many choices for on chip SRAM. If the size requirement for SRAM is small, implementing SRAM as Single Port Register Files should be considered. For larger SRAM blocks, standard Single Port SRAM macros can be used. Both solutions are available from ARM, within the Artisan product portfolio.

The Register File memory macros provide higher density, but with larger timing delay compared to standard SRAM macros, as shown below (figure 2):

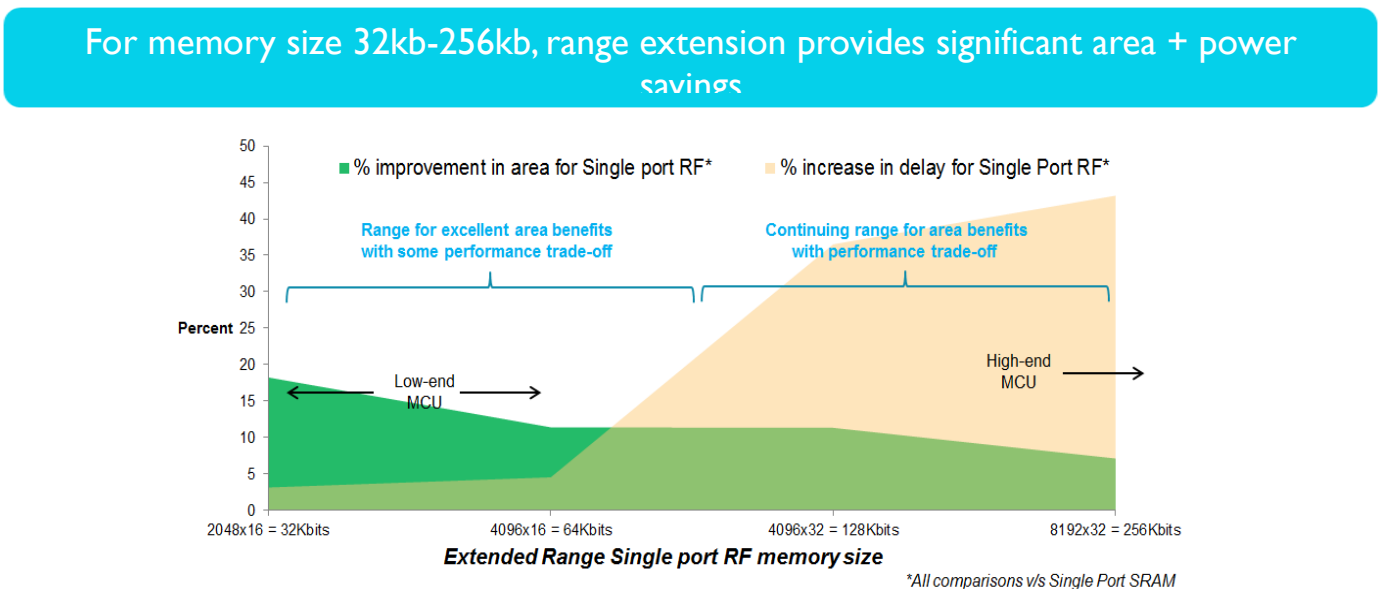

Figure 2 – Single port Register File provides high density memory

If timing of the SRAM is not critical, the single port register file memory solution can provide higher density for memory sizes up to 32Kbytes.

Typically, the Artisan SRAM macros are generated using a memory compiler. A number of optimization points are available so that optimized SRAM macros can be generated for different SRAM sizes. In addition, a range of low power modes are also available in Artisan SRAM designs (figure 3). For example, if a device remains in sleep mode during long periods of time, retention modes could be ideal. The power mode control signals can be connected to the sleep mode logic of the Cortex-M processor to fully utilize the low power capabilities of these SRAMs.

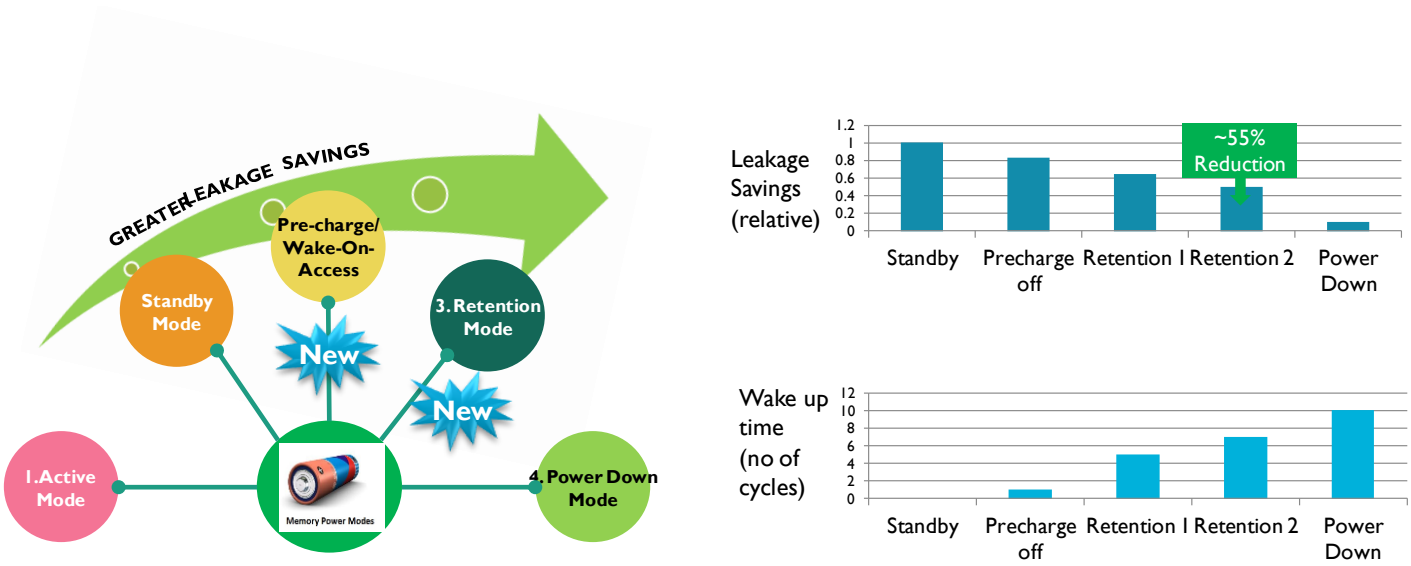

Figure 3 – Artisan SRAM macros support various low power modes

A bus wrapper is likely to be required when connecting on-chip SRAM macros to the AHB system. The Cortex-M System Design Kit (CMSDK) and CoreLink SIE-200 include an AHB to SRAM bridge, which can work with most on-chip synchronous SRAM macro with zero wait state configuration.

6. Analogue components

Although ARM does not provide analogue IP, a number of companies provide analogue IP that have an integrated digital interface, which can easily be connected to an ARM based system. For example, ADC and DAC macros from Cadence have digital interfaces that can be hooked up easily with an APB interface module. Reference designs for such arrangement are available, for example, Cadence has prepared a mixed signal design demonstration (reference 20 & 21) based on the Cortex-M0 Processor, Cortex-M System Design Kit, their Analogue IP and their Virtuoso design environment (figure 4). Cadence Virtuoso design environment enable the co-simulation of the digital and the analogue systems of the SoC in a single environment.

Figure 4 – System Simulation: Virtuoso Analog Design Environment + SimVision

For more information on this demo, please contact Cadence and ARM marketing teams directly.

7. Physical IP / cell libraries

A wide range of Artisan physical IP is available from the ARM Physical Design Group.

- Standard cell products and add-on kits for various foundries and process nodes
- A range of logic library architectures for different performance and density requirements within each node
- Multiple library selections for different voltage thresholds and channel lengths (e.g. standard Vt for normal design, high Vt for low power, low Vt for faster speed, long channel for improved leakage reduction) available at most nodes

For example, if high performance is needed, the SC9 or SC12 (larger but faster) libraries can be used. And for low-cost designs, the SC6 or SC8 libraries can be used for higher density (figure 5).

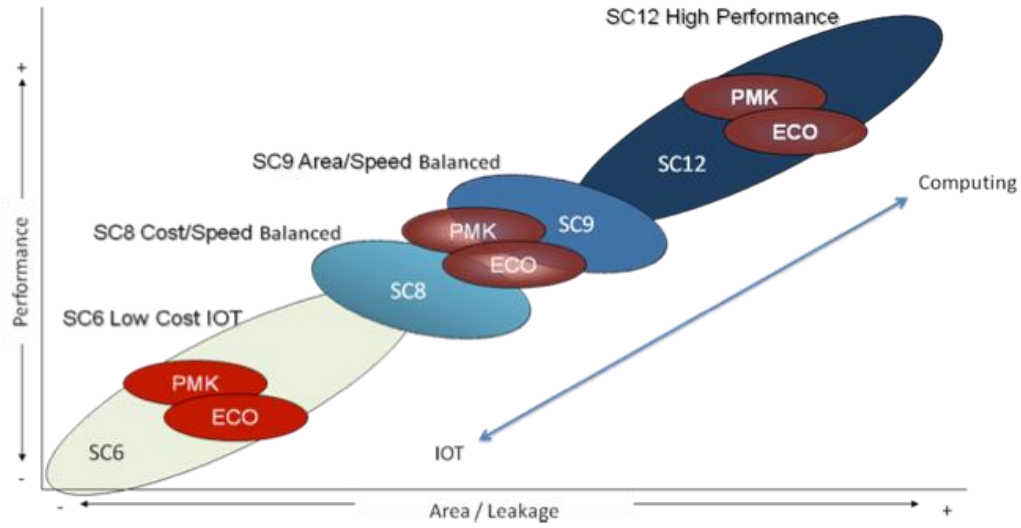

Figure 5 – A wide range of logic cell libraries can be available in a single process node

Often, each of the libraries also provides a Power Management Kit (PMK) and ECO kit. The PMK provides a range of special cells for power optimizations:

- Power gating (header cells, isolation cells)
  - Reduce leakage current in sleep modes
  - Can be used with state-retention power gating, SRPG
- Dynamic voltage and frequency scaling, DVFS (level shifters)
  - Reduce dynamic power by lowering voltage
  - Or just use multiple voltage domains
- SRPG (State retention cells: DFF, scan DFF, latch)
  - Power down without losing states
  - Fast wake up time in hundreds of ns
- Supporting cells (e.g. buffers, inverters)
  - Keep control signals always-on in power-gated blocks
  - No special well spacing required within blocks, available for wide range of availability from 180nm down to 16nm

In 2016 ARM also announced new Processor Optimization Pack (POP) IP for embedded processors. It helps accelerate the implementation of IoT SoCs by offering physical IP as well as reference designs and know-how to develop a design that has optimal performance with minimum area. The Artisan IoT POP technology includes implementation solutions that takes into account the power budget that is needed not just for the CPU, but the rest of the system, including the Always on Subsystem that require a minimum amount of leakage power. The ultra-low power nature of the process technology helps to address the wide variety of energy or power needs for the variety of IoT applications that exist. The first product of this kind is an IoT POP IP for Cortex-M33 processor.

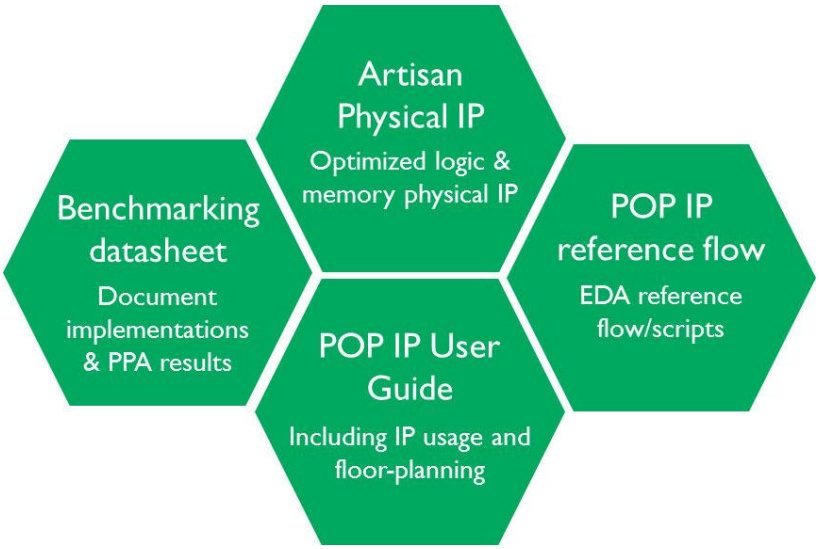

Figure 6 – ARM Artisan IoT POP IP

The key benefits of the IoT POP IP include:

- Innovative logic and memory architecture features that minimizes area, leakage and dynamic power while optimizing performance
- Silicon proven physical IP that works seamlessly with latest ARMv8-M processors and system IP
- Power implementation solutions for various IoT applications that can maximize battery life

For ultra-low power designs, one should also consider the Thick Gate Oxide (TGO) libraries to reduce leakage current, especially in always-on logic domains that don't require high performance (figure 7). TGO libraries can also interface directly with higher voltage batteries.

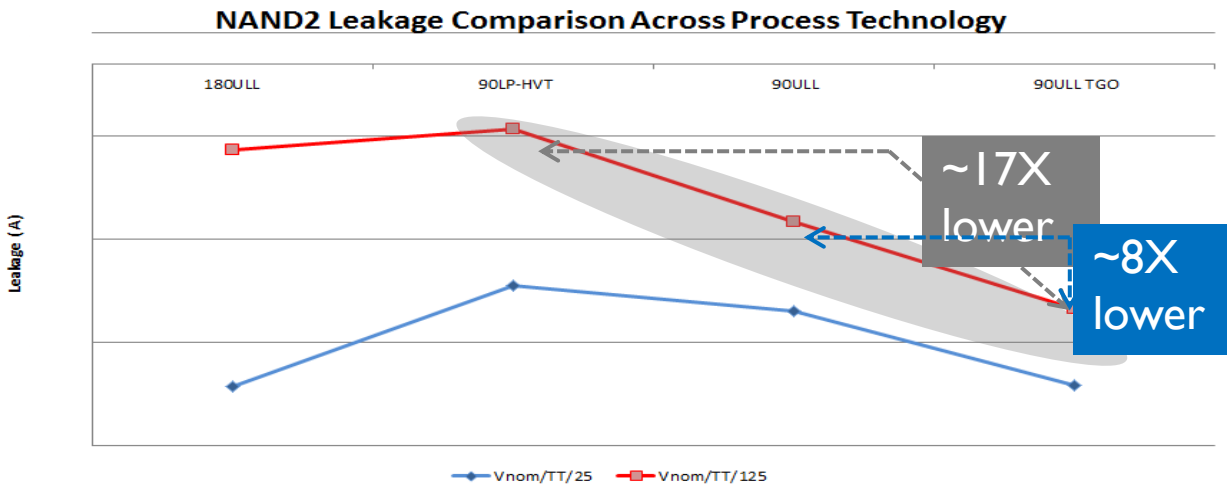

Figure 7 – TGO device provides significantly lower leakage than standard device

Front end views of many of the popular logic cell libraries are available on the ARM website through the DesignStart program. More information can be found at http://designstart.arm.com.

8. FPGA prototyping solutions

FPGA is one of the commonly used methods for system level testing and prototyping. It also enables earlier software development. Since the Verilog sources of the Cortex-M processors are very generic, they can be prototyped in a range of FPGA devices. ARM has developed an FPGA board specifically for prototyping of Cortex-M processor systems, called Cortex-M Prototyping System+ (reference 15, figure 8)

Figure 8 – Cortex-M Prototyping System, an FPGA designed for Cortex-M system development

The Cortex-M Prototyping System+ provides a very flexible memory system for emulating the memory system of your SoC designs, and includes a wide range of peripheral interface, expansion ports, and all the debug connectors you might need. The FPGA images of ready-to-use systems are included, as well as an example projects, including many basic AHB and APB infrastructure components and a range of peripherals.

To enable secure IP distribution, the board supports FPGA image encryption, so an encrypted image of a design can be shared with customers securely. One of the example FPGA projects included an encrypted partial image for the Cortex-M0 processor, which enables chip designers to add their own system IP design around it.

The Cortex-M Prototyping System is sold for US$995.

9. A Quick Glance of the Design Process

9.1 Decide which ARM Cortex-M Processor to use

Currently, seven Cortex-M processors have been released and are suitable for ASIC/SoC designs (see Figure 9):

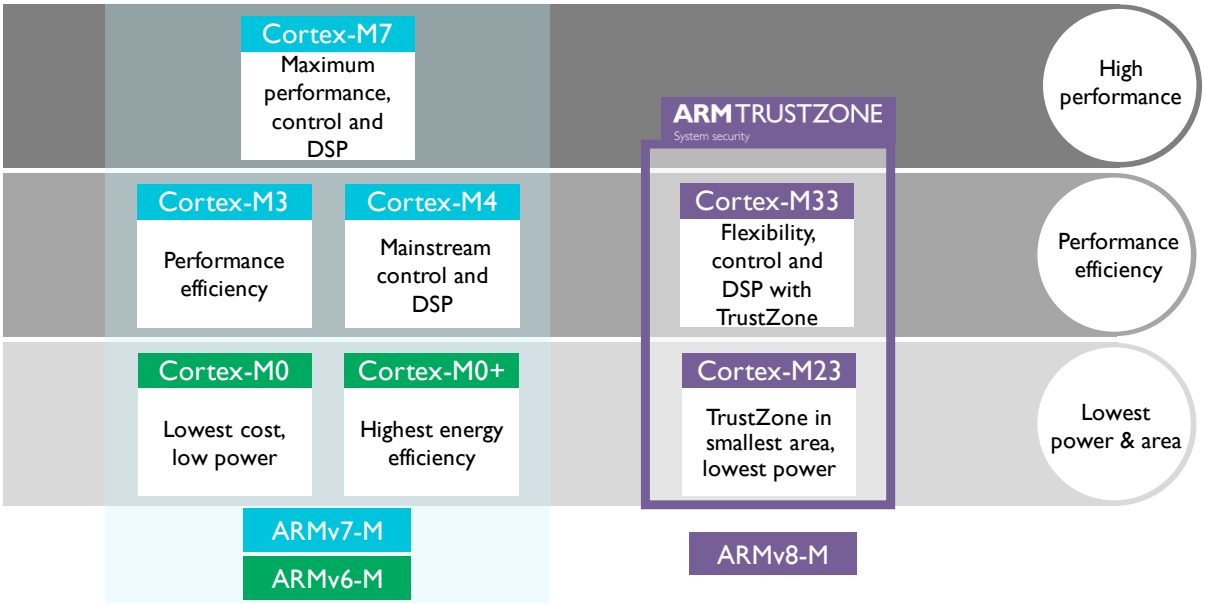

Figure 9 - the ARM Cortex-M processor family

- Cortex-M0 processor: The smallest ARM Processor – ideal for low cost microcontrollers for general data handling and I/O control tasks
- Cortex-M0+ processor: The most energy efficient ARM embedded processor for low power design. Same instruction set as Cortex-M0 processor, suitable for general data handling, I/O control tasks and IoT applications, plus many additional features.
- Cortex-M3 processor: The most popular processor for high performance microcontrollers with outstanding energy efficiency and advanced debug support.
- Cortex-M4 processor: Combining all the functions of the Cortex-M3 processor, with additional instructions to support DSP operations as well as an optional floating point unit.
- Cortex-M7 processor: The highest performance core with flexible memory interface options as well as support for safety-critical features. DSP performance is up to twice that of Cortex-M4, while maintaining upward software compatibility.
- Cortex-M23 processor: similar to the Cortex-M0+ processor, but supporting a newer architecture version called ARMv8-M, which adds TrustZone® security extension and several additional instructions. Many system level features including NVIC, MPU and debug/trace has also been enhanced.
- Cortex-M33 processor: similar to the Cortex-M3/M4 processor but also supporting ARMv8-M architecture including TrustZone security extension. This design has very high level of configurability and enhanced system level features.

Overview of the Cortex-M processors and feature comparison can be found in ARM web page:
https://developer.arm.com/products/processors/cortex-m

Information about ARMv-8M architecture and TrustZone security extension can be found in:
https://community.arm.com/processors/b/blog/posts/whitepaper---armv8-m-architecture-technical-overview

The choice of Cortex-M processor depends on that target application requirements. To help with the selection, one can easily benchmark an application code with a low-cost microcontroller board.

9.2 Defining power management strategy

The Cortex-M processors support multiple clock domains, multiple power domains and also support advanced low power techniques like State Retention Power Gating (SRPG). The processors architecturally provide two sleep modes, but this can be further expanded by adding device specific power control registers.

Power management in the system level is even more important. Clock gating control signal is available from AHB to APB bridges in the CMSDK, and CoreLink SIE-200 have additional multiple power domain supports. The IoT subsystem products including CoreLink SSE-200 have reference power management design. The latest ARM CPU and System IP make use of the ARM AMBA Low Power Interface Specification (LPI) (Reference 24) that supports advanced power management schemes. CoreLink SIE-200 has an implementation of system level power management that makes use of the LPI interfaces on the Cortex-M33 processor and CoreLink SIE-200 System IP.

In addition, memory macros might also have sleep mode and state retention mode support. For NVM, if there is no sleep mode support, one can possibly use power gating to switch off NVM during sleep operations. Header cells for power gating are available in the Artisan PMK products.

For Ultra Low Power design, you might need to use State Retention Power Gating (SRPG), and again the PMK products would be necessary. ARM also provides example power intend description files like CPF (Common Power Format) and UPF (Unified Power Format) for the Cortex-M Processors.

If performance requirement is low, some of the digital logic in always-on power domain could use TGO library for lower leakage current.

9.3 System Level integration

There is certainly a wide range of system integration tasks involved. The example systems in the CMSDK (for Cortex-M0, M0+, M3, M4 processors) can be a good starting point. These example systems can be easily modified by system designers. For designs using Cortex-M3 and Cortex-M33 processors, ready-to-use systems including CoreLink SSE-100 (for Cortex-M3) and CoreLink SSE-200 (for Cortex-M33) are available and are silicon proven. For early prototyping, example FPGA projects for the Cortex-M Prototyping System+ (reference 15) are also available for ARM customers.

For those designers who are new to Cortex-M system design, a range of training course are available from ARM (reference 16) and ARM approved training partners (reference 17).

9.4 Verification

The verification approaches and processes are often project dependent. Typically this requires a range of system level simulations, FPGA prototyping, and sometimes formal verifications. Please note there are also some verification tasks that must be done prior to tape out (see Signoff requirements).

The integration kit in the Cortex-M processor deliverable and the CMSDK provides infrastructure of the debug and trace connectivity. The CMSDK and CoreLink SIE-200 also included a range of verification components for testing bus protocol compliance, and for bus stimulus generation.

If using multiple power domains, you should also carry out verification of power intend (verification of CPF/UPF) and should carry out RTL simulations with power gating support.

9.5 Other design tasks

A number of other design tasks are also required. For example, on the DFT (Design For Testing) side you need to add support logic for scan tests, boundary tests (if needed), and memory BIST. ATPG (Automatic Test Pattern Generation) and test simulations are also often required. These are generic design tasks for every chip design and are not ARM platform specific.

A number of clock gating bypass and reset bypass control signals are provided in the ARM Cortex-M processors. These signals need to be connected properly during DFT integration. In addition, the processors have been tested with ATPG simulations as a part of the ARM internal design flow, and reached high level of test coverage.

On the implementation side, detailed floor planning, timing and signal integrity analysis and power analysis would need to be carried out.

9.6 Signoff requirements

The ARM document listed a number of verification tasks as signoff requirements. Typically this included Static Timing Analysis (STA), gate level simulations of booting and Logic Equivalent Checking (LEC). Please refer to product documentation for details.

10. Design Services

While not exactly push button flow, designing an ARM based SoC is not that difficult. Many ready-to-use system and peripheral IPs are designed using AMBA bus protocols, so they can be integrate with ARM Cortex-M Processors easily.

With the help of IP reuses and reference designs such as the CMSDK, for experienced companies, the development cycle from starting to engineering sample can be less than 6 months. For companies that are less experienced in chip designs, or even OEM / ODM that want to create a SoC for their particular applications, design services are available from various companies like design houses, consultancies and silicon aggregator. A list of ARM approved design partners can be found on ARM webpage:

https://www.arm.com/support/arm-approved-program/index.php

Thanks to wide range of companies working on ARM platforms, there are also large numbers of chip design engineers that are experienced with ARM processors. A webinar video on ARM website (ref 22): "So you think developing an ARM-based IoT chips needs to be complex or expensive? Think again." Provides a range of information about the Cortex-M0 DesignStart and useful information about starting an SoC project.

11. Summary

A range of IP products are needed in the development of SoC products based on the ARM Cortex-M processors. While many of the required IP are available from ARM, including the Cortex-M and Artisan families of products, there is also a wide range of IP products available from various third-party IP suppliers. The Cortex-M processors, peripheral IPs from various sources, together with system IP such as the Cortex-M System Design Kit (CMSDK), CoreLink SIE-200, CoreLink SSE subsystems, allows chip designers to create their Cortex-M based products quickly and with high quality.

References and additional information

1	Details of the ARM Development Studio 5 (DS 5) can be found on ARM web site: www.arm.com/ds5
2	Details of the Keil Microcontroller Development Kit (MDK) can be found on Keil web site: www.keil.com
3	Cortex-M System Design Kit http://www.arm.com/products/processors/cortex-m/cortex-m-system-design-kit.php
4	CoreLink SIE-200 – System IP for AHB5 designs https://www.arm.com/products/system-ip/interconnect/corelink-sie-200.php
5	CoreLink AHB Flash Cache – cache memory for flash memory http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.ddi0569a/index.html
6	CoreLink SSE-100 – Pre-verified IoT subsystem for Cortex-M3 https://www.arm.com/products/internet-of-things-solutions/corelink-sse-100-subsystem.php
7	CoreLink SSE-200 – Pre-verified IoT subsystem for Cortex-M33 https://www.arm.com/products/internet-of-things-solutions/corelink-sse-200-subsystem.php
8	CoreLink NIC-400 and other AXI based system IP https://www.arm.com/products/system-ip/interconnect/corelink-nic-family.php
9	ARM System IP tooling solutions https://www.arm.com/products/system-ip/system-ip-tooling
10	ARM CryptoCell-300 – system IP for security management https://www.arm.com/products/security-on-arm/trustzone-cryptocell
11	ARM Cordio wireless IP https://www.arm.com/products/wireless-ip
12	ARM wireless software solutions https://www.arm.com/products/wireless-ip/Software
13	ARM Artisan IoT POP IP for Cortex-M33 https://www.arm.com/products/physical-ip/pop-ip/iot-pop-ip.php
14	ARM DesignStart home page (include Cortex-M0 DesignStart) https://www.arm.com/products/designstart/index.php
15	Cortex-M Prototyping System http://www.arm.com/mps
16	ARM Training https://www.arm.com/support/training/index.php
17	ARM Approved Program (Design Partners and Training Partners) https://www.arm.com/support/arm-approved-program/index.php
18	Details of the Cortex-M Software Interface Standard (CMSIS) can be found on ARM web site: www.arm.com/cmsis
19	Cypress 55nm SONOS embedded flash is licensable via UMC (http://www.cypress.com/?rID=97512) Information from Cypress: The macro provides generic "SRAM like" interfaces for reading and programming. Soft bus specific adaptors are required for bus protocols such as AHB, AXI, APB. The macro sizes that can be made available range from 128KB to 8MB. The data bus width options can be X32, X64 and/or X128. The macro speed will be 20ns typical (25ns worst case) It is easy to design SONOS embedded flash: a. All support circuitry is included in the macro: Examples: Analog components such as pumps, bias circuits, DAC's, temp compensated references and digital logic for program and erase functionality. The flash macro only requires a single power supply. b. Generic Flash macro interfaces allow for an easy integration with AXI, AHB & APB bus interfaces.

In addition, there is a range of other benefits:
a. No ECC needed. The program/erase bit distributions is normal with no tail bits.
b. Adding SONOS preserves existing design IP and device models.
c. SONOS is only a 3-4 mask adder to the baseline CMOS process.

20	Cadence Mixed-signal/Low-Power Flow for Embedded ARM Cortex-M0 Designs http://www.arm.com/files/event/3_2014_PDG_Seminar_Cadence.pdf
21	Creating Mixed-Signal Devices for Smart Analogue & IoT Applications http://www.chinawebinar.com/zh_CN/STATIC/SITE/Mixed_Signal_Design_How_to-Approved_PPT.pdf
22	"So you think developing an ARM-based IoT chips needs to be complex or expensive? Think again." https://community.arm.com/company/b/blog/posts/so-you-think-developing-an-arm--based-iot-chip-needs-to-be-complex-or-expensive-think-again--webinar
23	Smart Sensors - Why Sensors need Intelligence? https://community.arm.com/processors/b/blog/posts/smart-sensors---why-sensors-need-intelligence
24	AMBA Low-power interface specification https://silver.arm.com/download/download.tm?pv=3552403&p=1344111

Trademarks