

**Figure 1:**  
Block diagram of the Cortex-M0+ processor

## Overview

The Cortex-M0+ processor builds on the very successful Cortex-M0 processor, retaining full instruction set and tool compatibility, while further reducing energy consumption and increasing performance.

The exceptionally small silicon area, low power and minimal code footprint of Cortex-M0+ enables developers to achieve 32-bit performance at an 8-bit price point, bypassing the step to 16-bit devices. The Cortex-M0+ processor comes with a wide selection of options to provide flexible development.

## Features

Feature	Description
Architecture	Armv6-M
Pipeline	2-stage
Bus Interface	AMBA AHB-Lite (Von Neumann bus architecture)
ISA Support	Thumb/Thumb-2 subset
Memory Protection	Optional Memory Protection Unit (MPU) with up to eight regions
Interrupts	Non-Maskable Interrupt (NMI) and up to 32 physical interrupts
Wake-up Interrupt Controller (WIC)	Optional for waking up the processor from state retention power gating or when all clocks are stopped
Sleep Modes	Integrated Wait For Interrupt (WFI) and Wait For Event (WFE) instructions and Sleep On Exit capability Sleep and Deep Sleep signals Optional retention mode with <a href="#">Arm Power Management Kit</a>
Enhanced Instructions	Hardware single-cycle (32x32) multiply
Debug	Optional JTAG and Serial Wire Debug ports Up to four breakpoints and two watchpoints
Trace	Optional Micro Trace Buffer (MTB)

## About the Processor

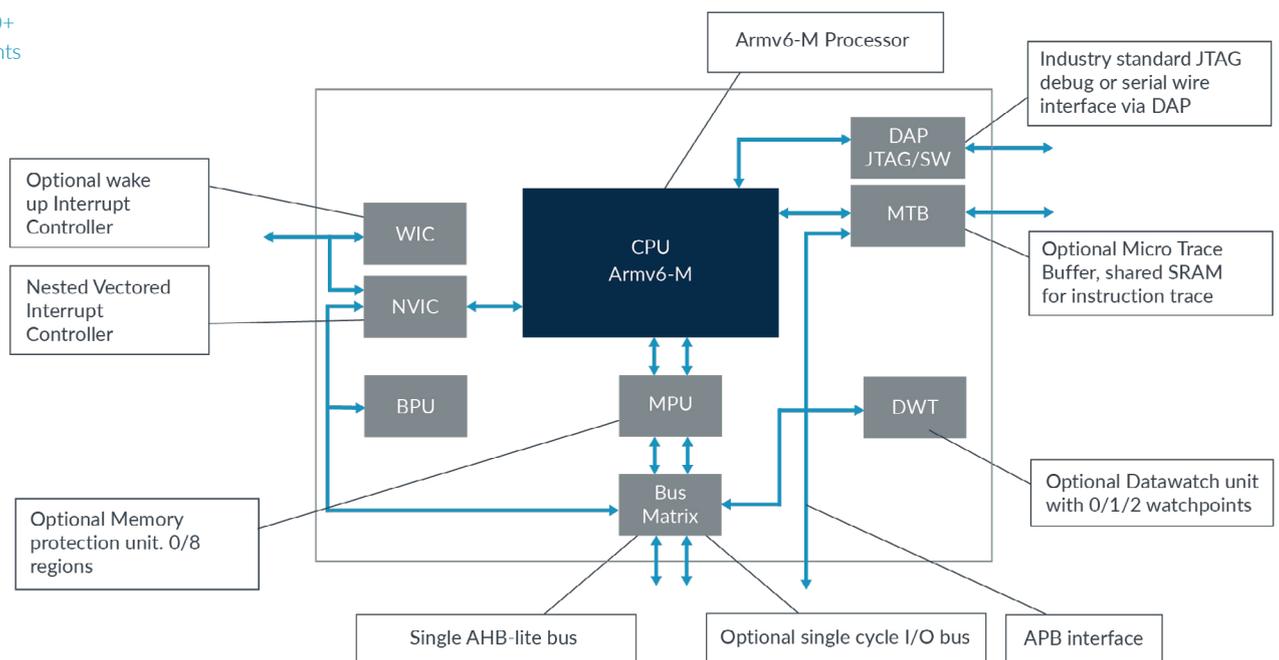
The Cortex-M0+ processor is a configurable, multistage, 32-bit RISC processor. It has an AMBA AHB-Lite interface and includes a Nested Vectored Interrupt Controller (NVIC) component. It also has optional hardware debug, single-cycle I/O interfacing, and memory-protection functionality. The processor can execute Thumb code and is compatible with other Cortex-M profile processors.

## Processor features

- + The Armv6-M Thumb® instruction set with Thumb-2 technology
- + Optionally, an Armv6-M compliant 24-bit SysTick timer
- + A 32-bit hardware multiplier. This can be the standard single-cycle multiplier, or a 32-cycle multiplier that has a lower area and performance implementation
- + Support for either little-endian or byte invariant big-endian data accesses
- + The ability to have deterministic, fixed-latency, interrupt handling
- + Load/store multiple and multicycle multiply instructions that can be abandoned and restarted to facilitate rapid interrupt handling
- + Optionally, Unprivileged/Privileged support for improved system integrity
- + Armv6-M C Application Binary Interface (C-ABI) compliant exception model, enabling the use of pure C functions as interrupt handlers
- + Low power sleep-mode entry using **WFI** and **WFE** instructions, or the return from interrupt sleep-on-exit feature

## Block Diagram

Figure 2: Cortex-M0+ processor components



---

# Cortex-M0+ Components

## NVIC features

- + Up to 32 external interrupt inputs, each with four levels of priority
- + Dedicated NMI input
- + Support for both level-sensitive and pulse-sensitive interrupt lines
- + Optional WIC providing ultra-low power sleep mode support
- + Optional relocation of the vector table

## Optional debug support

- + Zero to four hardware breakpoints
- + Zero to two watchpoints
- + Program Counter Sampling Register (PCSR) for non-intrusive code profiling, if at least one hardware data watchpoint is implemented
- + Single step and vector catch capabilities
- + Support for unlimited software breakpoints using BKPT instruction
- + Non-intrusive access to core peripherals and zero-wait state system slaves through a compact bus matrix. A debugger can access these devices, including memory, even when the processor is running
- + Full access to core registers when the processor is halted
- + Optional, low gate-count CoreSight compliant debug access through a Debug Access Port (DAP) supporting either Serial Wire or JTAG debug connections

## Bus interfaces

- + Single 32-bit AMBA-3 AHB-Lite system interface that provides simple integration to all system peripherals and memory
- + Optional single 32-bit single-cycle I/O port
- + Optional single 32-bit slave port that supports the DAP
- + Optional MPU:
  - Eight user configurable memory regions
  - Eight sub-region disables per region
  - Execute Never (XN) support
  - Default memory map support

## Optional Memory Protection Unit (MPU):

- + Eight user configurable memory regions
- + Eight sub-region disables per region
- + Execute Never (XN) support
- + Default memory map support

---

# Interfaces

## AHB-Lite interface

Transactions on the AHB-Lite interface are always marked as non-sequential.

Processor accesses and debug accesses share the external interface to external AHB peripherals. The processor accesses take priority over debug accesses.

Any vendor specific components can populate this bus.

**Note:** Instructions are only fetched using the AHB-Lite interface. To optimize performance, the Cortex-M0+ processor fetches ahead of the instruction it is executing. To minimize power consumption, the fetch ahead is limited to a maximum of 32 bits.

## Single-cycle I/O Port

The processor optionally implements a single-cycle I/O port that provides very high-speed access to tightly coupled peripherals, such as general-purpose-I/O (GPIO). The port is accessible both by loads and stores, from the processor and from the debugger. Code cannot be executed from the I/O port.

## Debug Access Port

The processor is implemented with either a low gate count DAP or a full CoreSight DAP.

The low gate count DAP provides a Serial Wire or JTAG debug port and connects to the processor slave port to provide full system-level debug access.

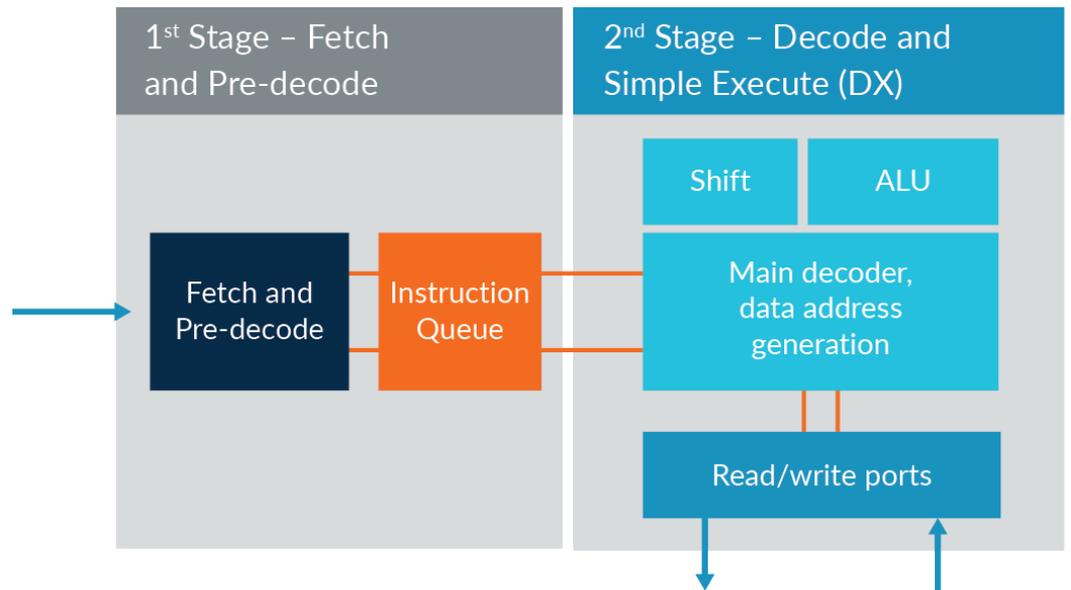
The full CoreSight DAP system enables the processor to provide full multiprocessor debug with simultaneous halt and release cross-triggering capabilities.

## Execution Trace Interface

The processor optionally implements an interface for the MTB execution trace component.

## Cortex-M0+ Pipeline

Figure 3: Cortex-M0+ processor pipeline



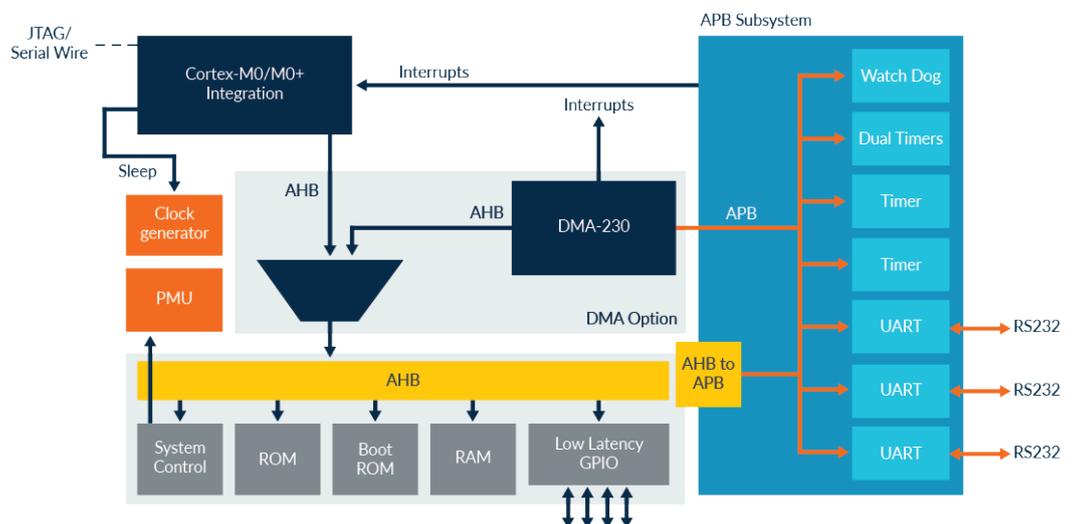
## Corstone-101

Corstone-101 is a licensable package that includes many useful components including the Cortex-M System Design Kit (CMSDK) which provides all the fundamental system elements to design an Soc around Arm Cortex-M0+.

### Features include:

- + A selection of AMBA AHB and APB infrastructure components
- + Essential peripherals such as GPIO, timers, watchdog, and UART
- + Example systems for Cortex-M0, Cortex-M0+, Cortex-M3, and Cortex-M4 processors
- + Compilation and simulation scripts for the Verilog environment
- + Software driver and example programs

Figure 4: Example System for Cortex-M0+



# Processor Configuration Options

The Cortex-M0 processor has configurable options that can be set during the implementation and integration stages to match the functional requirements.

Feature	Options
Interrupts	External interrupts 0-32
Data Endianness	Little-endian or big-endian
SysTick Timer	Present or absent
Number of Watchpoint Comparators	0, 1, 2
Number of breakpoint comparators	0, 1, 2, 3, 4
Halting Debug Support	Present or absent
Multiplier	Fast (one cycle) or slow (32 cycles)
Single-cycle I/O Port	Present or absent
Wake-up interrupt controller	Supported or not supported
Vector Table Offset Register	Present or absent
Unprivileged/Privileged support	Present or absent
Memory Protection Unit	Not present or 8-region
Reset all Registers	Present or absent
Instruction Fetch Width	16-bit only or mostly 32-bit

# Instruction Set

Figure 5: Instruction set



## Power, Performance and Area

DMIPS	CoreMark/MHz
0.95	2.39

Configuration	90LP Arm SC7 RVT SS 1.08V, 125°C		40LP Arm SC9 RVT C50 SS 0.99V, 125°C	
	Area mm <sup>2</sup>	Power $\mu$ W/MHz	Area mm <sup>2</sup>	Power $\mu$ W/MHz
Minimum Configuration*	0.0275	9.36	0.0066	3.8
Typical**	0.0576	13.04	0.0141	5.5

Max Freq	90LP Arm SC7 RVT SS 1.08V, 125°C	40LP Arm SC9 RVT C50 SS 0.99V, 125°C
Typical**	567MHz	297MHz

\* 1 IRQ, small multiplier, no debug, no WIC, 2 WIC lines 0 breakpoints, 0 watchpoints

\*\* 32 IRQ, fast multiplier, Debug, SysTick timer & WIC present, 34 WIC lines 4 breakpoints, 2 watchpoints

## Additional Technical Documents

1. Cortex-M0+ Technical Reference Manual - [TRM](#)
2. Cortex-M0+ Integration and Implementation Manual – available as part of the Bill of Materials
3. Armv6-M Architecture Reference Manual - [ARM](#)
4. CoreSight MTB-M0+ Technical Reference Manual - [MTB](#)

## Glossary of Terms

AHB-Lite	Advanced High-performance Bus Lite
BPU	Breakpoint Unit
C-ABI	C Application Binary Interface
CTI	Cross Trigger Interface Unit
DWT	Data Watchpoint and Trace
JTAG	Joint Test Action Group
MPU	Memory Protection Unit
MTB	Micro Trace Buffer
NMI	Non-Maskable Interrupt
NVIC	Nested Vectored Interrupt Controller
SWO	Serial Wire Output
WFE	Wait for event
WFI	Wait for interrupt
WIC	Wake-up interrupt controller
WIC	Wake-up Interrupt Controller

### Contact details

**UK**  
Salesinfo-eu@Arm.com

**Europe**  
Salesinfo-eu@Arm.com

**Japan**  
Salesinfo-eu@Arm.com

**Taiwan**  
Salesinfo-eu@Arm.com

**China**  
Salesinfo-eu@Arm.com

**USA**  
Salesinfo-us@Arm.com

**Asia Pacific**  
Salesinfo-us@Arm.com

**Korea**  
Salesinfo-us@Arm.com

**Israel**  
Salesinfo-us@Arm.com

**India**  
Salesinfo-us@Arm.com



All brand names or product names are the property of their respective holders. Neither the whole nor any part of the information contained in, or the product described in, this document may be adapted or reproduced in any material form except with the prior written permission of the copyright holder. The product described in this document is subject to continuous developments and improvements. All particulars of the product and its use contained in this document are given in good faith. All warranties implied or expressed, including but not limited to implied warranties of satisfactory quality or fitness for purpose are excluded. This document is intended only to provide information to the reader about the product. To the extent permitted by local laws Arm shall not be liable for any loss or damage arising from the use of any information in this document or any error or omission in such information.