Arm Cortex-M33 Processor Datasheet

Datasheet

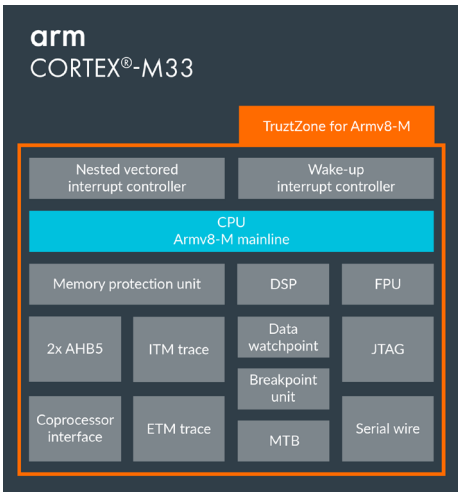

Figure 1: Block diagram of the Cortex-M33 processor

Overview

The Arm Cortex-M33 processor is the first Armv8-M processor designed to address embedded and IoT markets especially those that require efficient security or digital signal control. Armv8-M introduces TrustZone which forms the foundation of security for embedded and IoT applications. The processor has specific features to increase processing at the endpoint such as a 20% increase in performance over the Cortex-M4, a Digital Signal Processing (DSP) extension, a Floating-point Unit (FPU), a coprocessor interface to offload compute intensive operations, and Arm Custom Instructions (ACI) to speed up specific operations.

Features

Feature	Description
Pipeline	3-stage
Software security	Optional TrustZone for Armv8-M, with optional security attribution unit (SAU) of up to 8 regions Stack limit checking
DSP extension	Optional DSP/SIMD instructions Single-cycle 16/32-bit MAC Single-cycle dual 16-bit MAC 8/16-bit SIMD arithmetic
FPU	Optional single-precision FPU IEEE 754 compliant
Coprocessor interface	Optional dedicated coprocessor bus interface for up to 8 coprocessor units for custom compute
Arm Custom Instructions	Custom data path to add custom instructions
Memory protection	Optional Memory Protection Unit (MPU) with up to 16 regions per security state
Interrupts	Non-maskable Interrupt (NMI) and up to 480 physical interrupts with 8 to 256 priority levels
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 event (WFE) and wait for interrupt (WFI) instructions with Sleep On Exit functionality
Debug	Optional JTAG and Serial Wire Debug ports Up to 8 Breakpoints and 4 Watchpoints
Trace	Optional Embedded Trace Macrocell (ETM), Micro Trace Buffer (MTB), Data Watchpoint and Trace (DWT) and Instrumentation Trace Macrocell (ITM)

About the Processor

The Cortex-M33 processor has a low gate count, it is very energy-efficient and is intended for microcontroller and deeply embedded applications. The processor is based on the Armv8-M architecture and is suited for applications where security is an important consideration.

The interfaces that the processor supports include:

- Code AHB (C-AHB) interface
- System AHB (S-AHB) interface
- External PPB (EPPB) APB interface
- Debug AHB (D-AHB) interface

The processor has optional:

- Arm TrustZone technology, using the Armv8-M security extension supporting Secure and Non-secure states
- MPUs which you can configure to protect regions of memory
- Floating-point arithmetic functionality with support for single-precision arithmetic
- Support for ETM and MTB trace
- Arm Custom Instructions

The processor is highly configurable and is intended for a wide range of high-performance, deeply embedded applications that require fast interrupt response features.

Block Diagram

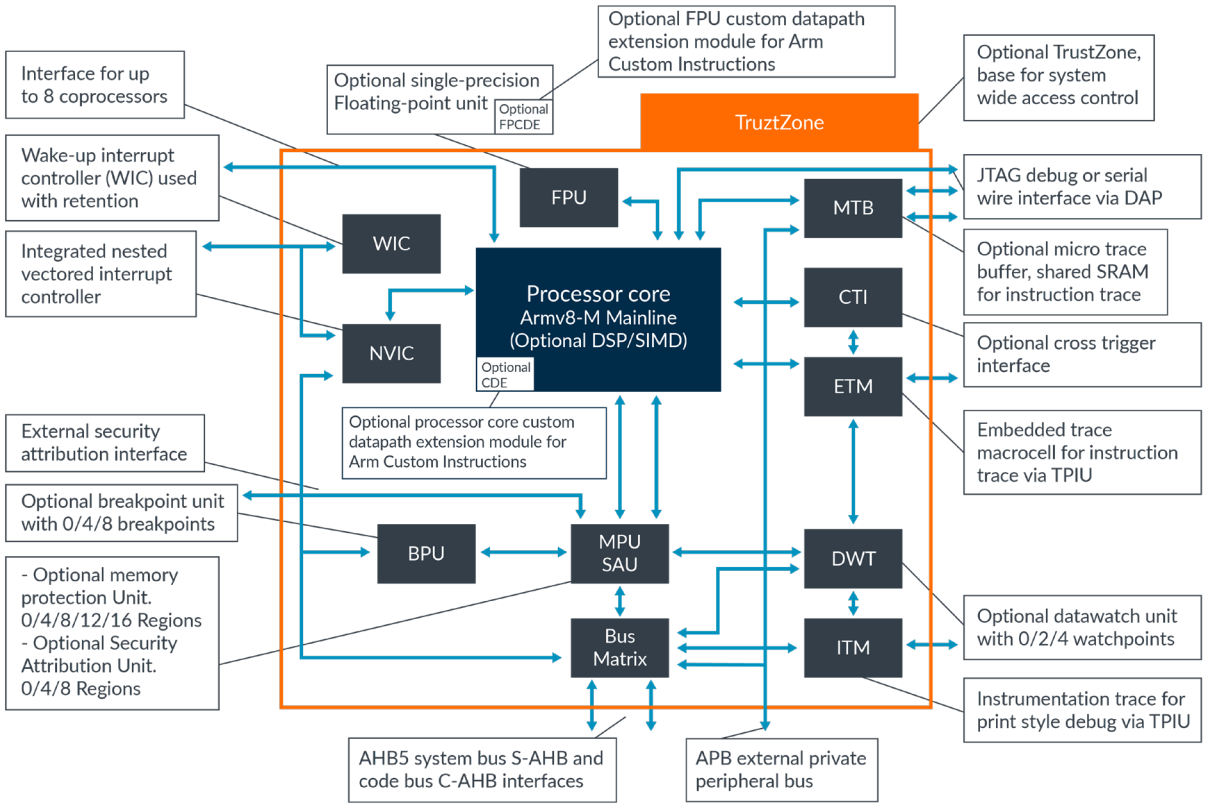

Figure 2: Cortex-M33 processor components

Cortex-M33 Components

Processor Core

The processor core provides:

- Limited dual-issue of common 16-bit instruction pairs
- Single cycle 32×32-bit multiplier
- Integer divide unit with support for operand-dependent early termination
- Support for interrupted continuable load and store multiple operations
- Load and store operations that both support precise bus errors
- To support Arm Custom Instruction, the processor core includes an optional CDE module. This module is used to execute user-defined instructions that work on general-purpose registers

Security Attribution and Memory Protection

The Cortex-M33 processor supports the Armv8-M Protected Memory System Architecture (PMSA) that provides programmable support for memory protection using a number of software controllable regions.

Memory regions can be programmed to generate faults when accessed inappropriately by unprivileged software reducing the scope of incorrectly written application code. The architecture includes fault status registers to allow an exception handler to determine the source of the fault and to apply corrective action or notify the system.

The Cortex-M33 processor also includes optional support for defining memory regions as Secure or Non- secure, as defined in the Armv8-M security extension, and protecting the regions from accesses with an inappropriate level of security.

Floating-point Unit

The FPU provides:

- Instructions for single-precision (C programming language float type) data-processing operations
- Instructions for double-precision (C double type) load and store operations
- Combined multiply-add instructions for increased precision (Fused MAC)
- Hardware support for conversion, addition, subtraction, multiplication with optional accumulate, division, and square-root
- Hardware support for denormals and all IEEE Standard 754-2008 rounding modes
- 32 32-bit single-precision registers or 16 64-bit double-precision registers
- Lazy floating-point context save. Automated stacking of floating-point state is delayed until the ISR attempts to execute a floating-point instruction. This reduces the latency to enter the ISR and removes floating-point context save for ISRs that do not use floating-point

- To support Arm Custom Instructions, the FPU includes an optional floating-point CDE module. This module is used to execute user-defined instructions that work on floating-point registers. If the optional FPU is not present, then the optional floating-point CDE module is not present either

Nested Vectored Interrupt Controller

The Nested Vectored Interrupt Controller (NVIC) is closely integrated with the core to achieve low-latency interrupt processing.

Functions of the NVIC include:

- External interrupts, configurable from 1 to 480 using a contiguous or non-contiguous mapping. This is configured at implementation
- Configurable levels of interrupt priority from 8 to 256. This is configured at implementation. Dynamic reprioritization of interrupts
- Priority grouping. This enables selection of preempting interrupt levels and non-preempting interrupt levels
- Support for tail-chaining and late arrival of interrupts. This enables back-to-back interrupt processing without the overhead of state saving and restoration between interrupts
- Optional support for the Armv8-M security extension. Secure interrupts can be prioritized above any Non-secure interrupt

Coprocessor Interface

The Cortex-M33 processor supports an external coprocessor interface which allows the integration of tightly coupled accelerator hardware with the processor. The programmers model allows software to communicate with the hardware using architectural coprocessor instructions.

The external coprocessor interface:

- Supports up to eight separate coprocessors, CP0-CP7, depending on your implementation. The remaining coprocessor numbers, C8-C15, are reserved. CP10 and CP11 are always reserved for hardware floating-point. For more information, see the Armv8-M Architecture Reference Manual
- Supports low-latency data transfer from the processor to and from the accelerator components
- Has a sustained bandwidth up to twice of the processor memory interface

Cross Trigger Interface Unit

The optional CTI enables the debug logic, micro trace buffer (MTB), and ETM to interact with each other and with other CoreSight components.

ETM

The optional Embedded Trace Macrocell provides instruction-only capabilities when configured. See the Arm CoreSight ETM-M33 Technical Reference Manual for more information.

Micro Trace Buffer

The MTB provides a simple low-cost execution trace solution for the Cortex-M33 processor.

Trace is written to an SRAM interface, and can be extracted using a dedicated AHB slave interface (M- AHB) on the processor. The MTB can be controlled by memory mapped registers in the PPB region or by events generated by the DWT or through the CTI. See the Arm CoreSight MTB-M33 Technical Reference Manual for more information.

Debug and Trace

Debug and trace components include a configurable Breakpoint Unit (BPU) for implementing breakpoints, and a configurable DWT unit for implementing watchpoints, data tracing, and system profiling.

Other debug and trace components include optional ITM for support of printf() style debugging, using instrumentation trace. Interfaces are suitable for:

- Passing on-chip data through a Trace Port Interface Unit (TPIU) to a Trace Port Analyzer (TPA), including Serial Wire Output (SWO) mode
- A ROM table to allow debuggers to determine which components are implemented in the Cortex-M33 processor
- Debugger access to all memory and registers in the system, including access to memory-mapped devices, access to internal core registers when the core is halted, and access to debug control registers even when reset is asserted

Interfaces

The processor has various external interfaces.

- **Code and System AHB Interfaces -** Harvard AHB bus architecture supporting exclusive transactions and security state.
- **System AHB Interface -** The S-AHB interface is used for any instruction fetch and data access to the memory-mapped SRAM, peripheral, external RAM and external device, or vendor_SYS regions of the Armv8-M memory map.
- **Code AHB Interface -** The C-AHB interface is used for any instruction fetch and data access to the code region of the Armv8-M memory map.
- **External Private Peripheral Bus -** The EPPB APB interface enables access to CoreSight-compatible debug and trace components in a system connected to the processor.
- **Secure Attribution Interface -** The processor has an interface that connects to an external Implementation Defined Attribution Unit (IDAU), which enables your system to set security attributes based on address.

- **ATB Interfaces -** The ATB interfaces output trace data for debugging. The ATB interfaces are compatible with the CoreSight architecture. See the *Arm CoreSight Architecture Specification v2.0* for more information. The instruction ATB interface is used by the optional ETM, and the instrumentation ATB interface is used by the optional *ITM*.
- **Micro Trace Buffer Interfaces -** The MTB AHB slave interface and SRAM interface are for the optional CoreSight Micro Trace Buffer.
- **Coprocessor Interface -** The coprocessor interface is designed for closely coupled external accelerator hardware.
- **Debug AHB Interface -** The D-AHB slave interface allows a debugger access to registers, memory, and peripherals. The D-AHB interface provides debug access to the processor and the complete memory map.
- **Cross Trigger Interface -** The processor includes an optional CTI Unit that has an interface that is suitable for connection to external CoreSight components using a Cross Trigger Matrix (CTM).
- **Power Control Interface -** The processor optionally supports a number of internal power domains which can be enabled and disabled using Q-channel interfaces connected to a Power Management Unit (PMU) in the system.

Arm Custom instructions

Arm introduces two sets of three classes of instruction extension in the coprocessor instruction space:

- Three classes operate on the general-purpose register file, including the condition code flags NZCV
- Three classes operate on the floating-point/SIMD register file only

The three classes are defined by the following instruction patterns:

- The destination register or the destination register pair of an instruction might be read, as well as written (non-accumulator and accumulator variants)
- The operation code can be split between a true operation code in the custom datapath and an immediate value used in the custom datapath
- Immediate consequences of the above are:
  - No operations on the floating-point registers can set condition codes
  - There are no operations using registers from both register files
- Operations on the general-purpose register file operate on 32-bit registers, or a dual-register consisting of a 64-bit value constructed from an even-numbered, general-purpose register and its immediately following odd pair

Instruction	Assembly	Inputs	Outputs
General-purpose registers and NZCV flags			
`CX1{A}`	`CX1{A} Pn,{Rd,}Rd,#imm`	Immediate and **1x 32-bit** GPR/NZCV {same as output}	**1x 32-bit** GPR or NZCV
`CX2{A}`	`CX2{A} Pn,{Rd,}Rd,Rn,#imm`	Immediate and **2x 32-bit** GPR/NZCV {one same as output}	**1x 32-bit** GPR or NZCV
`CX3{A}`	`CX3{A} Pn,{Rd,}Rd,Rn,Rm,#imm`	Immediate and **3x 32-bit** GPR/NZCV {one same as output}	**1x 32-bit** GPR or NZCV
`CX1D{A}`	`CX1D{A} Pn,{Rd,}Rd,#imm`	Immediate and **1x 32-bit** GPR/NZCV {two same as output}	**2x 32-bit** GPR
`CX2D{A}`	`CX2D{A} Pn,{Rd,}Rd,Rn,#imm`	Immediate and **2x 32-bit** GPR/NZCV {two same as output}	**2x 32-bit** GPR
`CX3D{A}`	`CX3D{A} Pn,{Rd,}Rd,Rn,Rm,#imm`	Immediate and **3x 32-bit** GPR/NZCV {two same as output}	**2x 32-bit** GPR
Vector registers			
`VCX1{A}.F`	`VCX1{A}.F Pn,{Sd,}Sd,#imm`	Immediate and **1x 32-bit** fp32 register {same as output}	**1x 32-bit** fp32 register
`VCX2{A}.F`	`VCX2{A}.F Pn,{Sd,}Sd,Sd,#imm`	Immediate and **2x 32-bit** fp32 register {one same as output}	**1x 32-bit** fp32 register
`VCX3{A}.F`	`VCX3{A}.F Pn,{Sd,}Sd,Sn,Sm,#imm`	Immediate and **3x 32-bit** fp32 register {one same as output}	**1x 32-bit** fp32 register
`VCX1{A}.D`	`VCX1{A}.D Pn,{Dd,}Dd,#imm`	Immediate and **1x 64-bit** fp64 register {same as output}	**1x 64-bit** fp64 register
`VCX2{A}.D`	`VCX2{A}.D Pn,{Dd,}Dd,Dn,#imm`	Immediate and **2x 64-bit** fp64 register {one same as output}	**1x 64-bit** fp64 register
`VCX3{A}.D`	`VCX3{A}.D Pn,{Dd,}Dd,Dn,Dm,#imm`	Immediate and **3x 64-bit** fp64 register {one same as output}	**1x 64-bit** fp64 register
`VCX1{A}.Q`	`VCX1{A}.Q Pn,{Qd,}Qd,#imm`	Immediate and **1x 128-bit** vector register {same as output}	**1x 128-bit** vector register
`VCX2{A}.Q`	`VCX2{A}.Q Pn,{Qd,}Qd,Qn,#imm`	Immediate and **2x 128-bit** vector register {one same as output}	**1x 128-bit** vector register
`VCX3{A}.Q`	`VCX3{A}.Q Pn,{Qd,}Qd,Qn,Qm,#imm`	Immediate and **3x 128-bit** vector register {one same as output}	**1x 128-bit** vector register

Pn	Coprocessor number	Sd	Destination register (32-bit vector register)
A	Accumulate	Sn, Sm	Input registers (32-bit vector registers)
Rd	Destination register	Dd	Destination register (64-bit vector register)
Rn, Rm	Input registers	Dn, Dm	Input registers (64-bit vector registers)
{ }	Optional	Qd	Destination register (128-bit vector)
#imm	Immediate value	Qn, Qm	Input registers (128-bit vector registers)

Cortex-M33 Pipeline

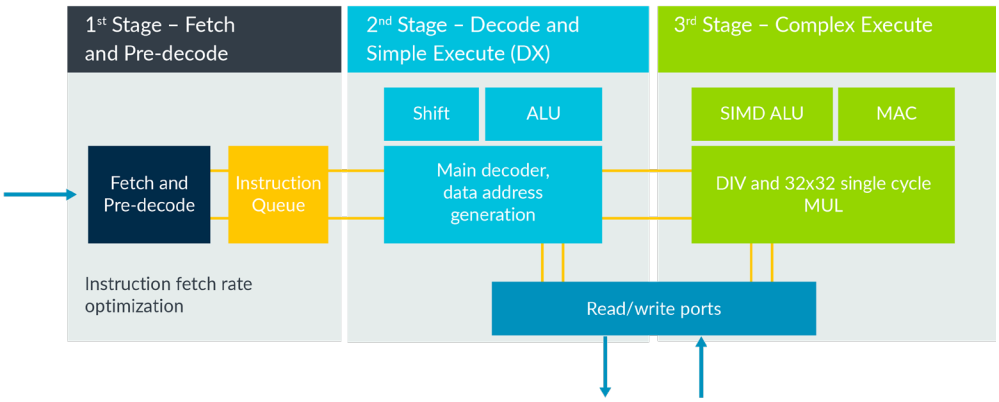

Figure 3: Cortex-M33 processor pipeline

- 3-stage pipeline
- Early termination of operation completing in DX
- Limited dual issue capability
- Five x 16-bit instruction queue
- Full Harvard memory interface

Corstone Reference Design

Corstone provides an ideal starting point for any SoC design, with the lowest risk and development cost. It includes various system IP components and a reference design integrating the processor, security and system IP, as well as a range of software and development tools.

Corstone features include:

- Implementation of an Arm-defined subsystem architecture
- Integration of the main components
- Extensively verified
- Broad software roadmap
- Build your SoC on top of it
- Configurable and modifiable
- Tailor it to specific needs
- Accelerates PSA Certified
- Silicon-proven

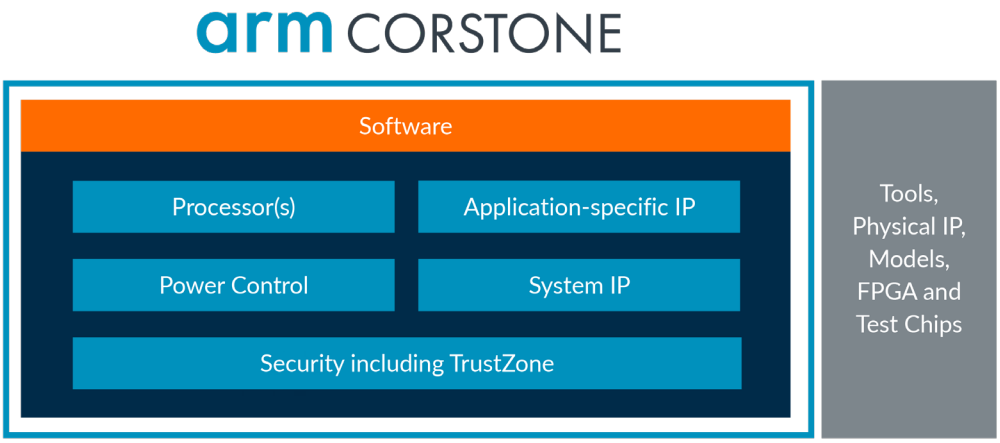

Figure 4: Corstone reference design diagram

Processor Configuration Options

The Cortex-M33 processor has configurable options that you can set during the implementation and integration stages to match your functional requirements.

Feature	Options
Floating-point	No floating-point
	Single-precision floating-point only
DSP extension	No Armv8-M DSP Extension
	Armv8-M DSP extension supported, including the following instruction classes
TrustZone	No TrustZone for Armv8-M
	TrustZone for Armv8-M security extension
Non-secure protected memory regions	0 region, 4 regions, 8 regions, 12 regions, or 16 regions
Secure protected memory regions	0 region, 4 regions, 8 regions, 12 regions, or 16 regions when TrustZone is included
Security Attribution Unit (SAU)	0 region, 4 regions, or 8 regions when TrustZone is included
Interrupts	1-480 interrupts To support non-contiguous mapping, you can remove individual interrupts
Number of bits of interrupt priority	Between three and eight bits of interrupt priority, between 8 and 256 levels of priority implemented
Debug watchpoints and breakpoints	Minimal debug No Halting debug or memory and peripheral access
	Reduced set Two data watchpoint comparators and four breakpoint comparators
	Full set Four data watchpoint comparators and eight breakpoint comparators
ITM and DWT functionality	No ITM or DWT trace
	Complete ITM and DWT trace
ETM	No ETM support
	ETM instruction execution trace
MTB	No MTB support
	MTB instruction trace
CTI	No CTI
	CTI included
WIC	No WIC controller
	WIC controller included
External coprocessor interface	No support for coprocessor hardware
	Support for coprocessor hardware
Arm Custom Instructions with Custom Datapath Extension (CDE) modules on a coprocessor basis	The coprocessor executes instructions and the CDE modules are not used
	The CDE module executes instructions and the coprocessor is bypassed

Instruction Set

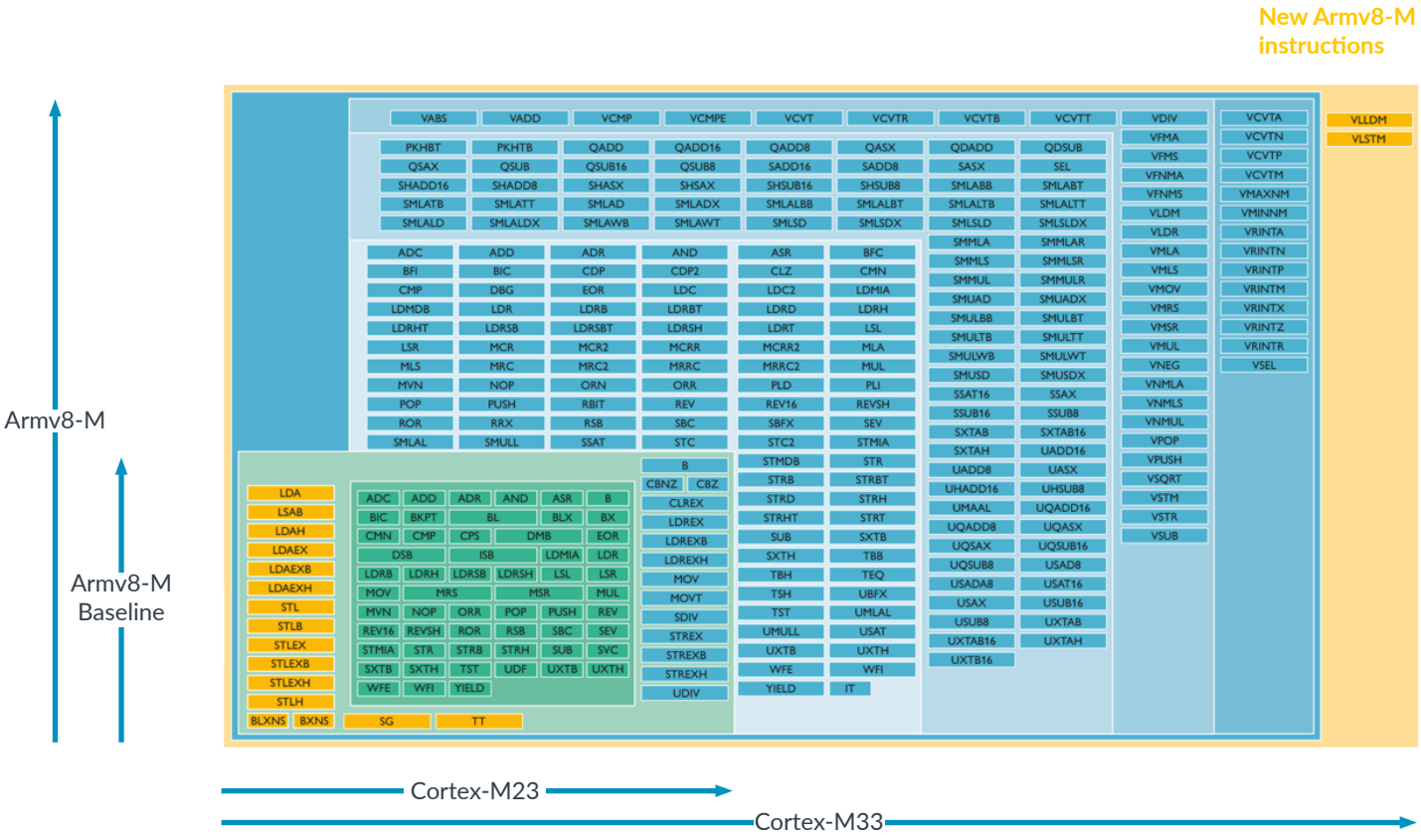

Figure 5: Instruction set

Power, Performance and Area

DMIPS	CoreMark/MHz
1.50	4.02

Configuration	40ULP Arm SC9 ELVT C40 SSG 0.99V, -40°C		40LP Arm SC9 RVT C50 SS 0.99V, -40°C		28HT Arm SC7MC PP140Z SVT HD C40 SSG 0.81V 0°C		16FFC Arm SC7P5MC PP96P SVT C16 SSGNP 0.72V 0°C	
	Area mm^2	Power µW/MHz	Area mm^2	Power µW/MHz	Area mm^2	Power µW/MHz	Area mm^2	Power µW/MHz
Minimum Configuration*	0.026	12.6	0.028	12.0	0.011	5.6	0.008	3.9
Feature Rich with TrustZone**	0.093	17.1	0.097	17.4	0.039	7.9	0.026	5.4

Max Freq	40LP Arm 40LP SC12 RVT C50 SS 0.99V, -40°C	28HT Arm CLN28HT SC9MC PP140Z SVT C30 HD SSG 0.81V 0°C	16FFC Arm SC7P5MC PP96P SVT C16 SSGNP 0.72V 0°C
Feature Rich Configuration with TrustZone**	235 MHz (register to I/O)	614 MHz (register to I/O)	1028 MHz (register to I/O)

*FPU 0; DSP 0; SECEXT 0; CPIF 0; MPU_NS 0; MPU_S 0; SAU 0; NUMIRQ 1; IRQLVL 3; IRQLATENCY 4294967295; IRQDIS 0; DBGLVL 0; ITM 0; ETM 0; MTB 0; MTBAWIDTH 0; WIC 0; WICLINES 0; CTI 0; RAR 0;

**FPU 1; DSP 1; SECEXT 1; CPIF 0; MPU_NS 8; MPU_S 0; SAU 0; NUMIRQ 32; IRQLVL 3; IRQLATENCY 4294967295; IRQDIS 0; DBGLVL 2; ITM 1; ETM 1; MTB 0; MTBAWIDTH 0; WIC 1; WICLINES 35; CTI 0; RAR 0;

Additional Technical documents

1. Cortex-M33 Technical Reference Manual - TRM

2. Cortex-M33 Integration and Implementation Manual – available as part of the Bill of Materials

3. Armv8-M Architecture Reference Manual - Arm

4. CoreSight ETM-M33 Technical Reference Manual - ETM

5. CoreSight MTB-M33 Technical Reference Manual - MTB

Glossary of Terms

ACI	Arm Custom Instructions
AHB	Advanced High-performance Bus
APB	Advanced Peripheral Bus
ATB	Advanced Trace Bus
BPU	Breakpoint Unit
C-AHB	Code AHB
CDE	Custom Datapath Extension
CTI	Cross Trigger Interface
CTM	Cross Trigger Matrix
D-AHB	Debug AHB
DSP	Digital Signal Processing
DWT	Data Watchpoint and Trace
EPPB	External Private Peripheral Bus
ETM	Instruction TraceEmbedded Trace Macrocell
FPU	Floating-point Unit
IDAU	Implementation Defined Attribution Unit
IEEE	Institute of Electrical and Electronics Engineers
ISR	Interrupt Service Routine
ITM	Instrumentation Trace Macrocell
JTAG	Joint Test Action Group
MAC	Multiply and Accumulate
MPU	Memory Protection Unit
MTB	Micro Trace Buffer
NMI	Non-maskable Interrupt
NVIC	Nested Vectored Interrupt Controller
PMSA	Protected Memory System Architecture
PMU	Power Management Unit
PPB	Private Peripheral Bus
PSA	Platform Security Architecture
RAM	Rand Access Memory
ROM	Read Only Memory
S-AHB	System AHB

SAU	Security Attribution Unit
SIMD	Single Instruction, Multiple Data
SRAM	Static RAM
SWO	Serial Wire Output
TPA	Trace Port Analyzer
TPIU	Trace Port Interface Unit
WFE	Wait for Event
WFI	Wait for Interrupt
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

arm