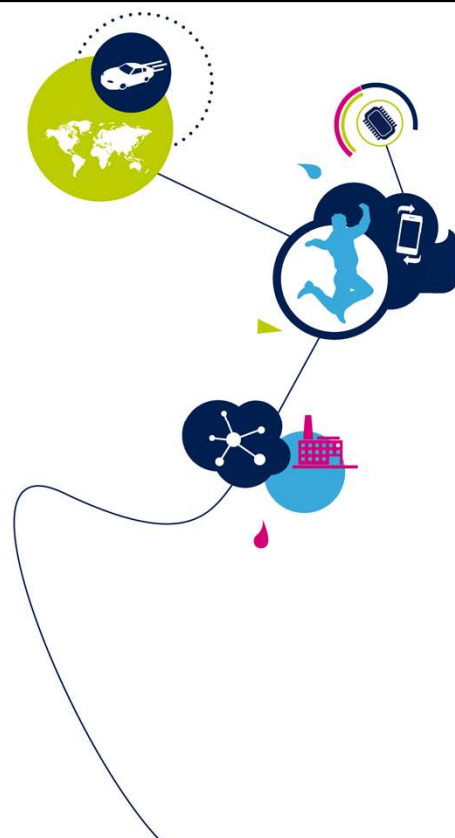


# STM32MP1-M4 コア

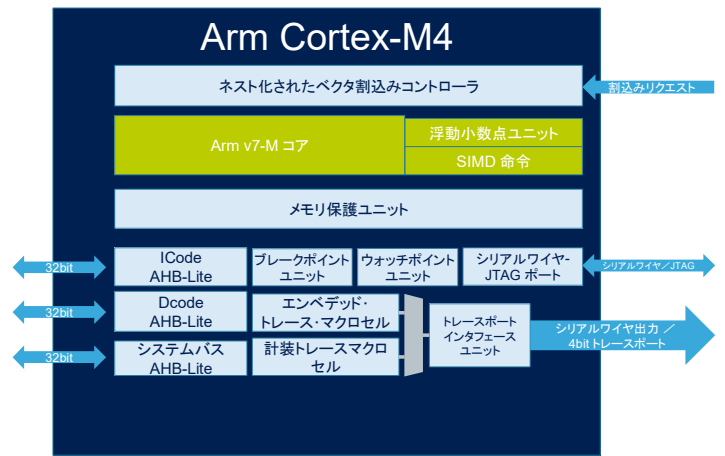
Arm® Cortex®-M4 コア  
1.0 版



こんにちは、STM32MP1 マイクロプロセッサシリーズに組み込まれている Arm Cortex-M4 コアのプレゼンテーションへようこそ。

# Cortex-M4 プロセッサの概要

- Arm v7-M アーキテクチャ
- ハーバードアーキテクチャ、3 段パイプライン
  - 浮動小数点ユニット(FPU)
  - シングルインストラクションマルチプルデータ命令
- メモリ保護ユニット(MPU)
- 統合されたネスト化されたベクタ割り込みコントローラ(NVIC)
- 侵入型および非侵入型デバッグ
- 最大 703Coremark @209MHz



省電力型	高性能な信号処理
MCU+DSP 要件に対応する最も省電力型の 32bit 組込みプロセッサ	SIMD 命令によって加速されたデータ信号処理



Cortex-M4 コアは、32bit RISC コアの Arm Cortex-M グループの一部です。Arm v7-M アーキテクチャを実装し、3 段パイプラインを備えています。スカラー整数命令に加えて、DSP アルゴリズムのパフォーマンスを向上させるのに役立つ単精度浮動小数点ユニットと SIMD 整数命令もサポートしています。

209MHz で動作するとき、最大 703Coremark を提供します。

Cortex-M4 には 3 つの AHB-Lite マスタポートがあり、命令とデータの同時トランザクションが可能になります。

STM32MP1 ペリフェラルからの割り込みは、ネスト化されたベクタ割り込みコントローラ (NVIC) で処理されます。

メモリ保護ユニット(MPU)は、コアによって開始された命令およびデータの要求に対して、属性とアクセス許可の割当てを担当します。

多くのデバッグユニットが実装されています。

シリアルワイヤまたは JTAG デバッグポート (SWJ-DP) と外部デバッグプローブ間の通信には、シリアルワイヤまたは JTAG の 2 つのプロトコルを使用できます。

侵入型デバッグは、ブレークポイントユニットとウォッチポイントユニットによって実行されます。

非侵入型デバッグでは、Cortex-M4 は 2 つのリアルタイムトレース機能をサポートしています。すなわち、組込みトレースマクロセル (ETM) と計装トレースマクロセル (ITM) です。トレースパケットは、トレースポートインタフェースユニット (TPIU) を介して外部トレースポートアナライザに出力されます。

# Arm Cortex-M の互換性

3

- すべてのアプリケーションにわたってシームレスなアーキテクチャ

Cortex-M0 & M0+	Cortex-M3	Cortex-M4	Cortex-M7
超低電力	Arm がリリースした最初の Cortex-M CPU	高性能	

バイナリ互換およびツール互換

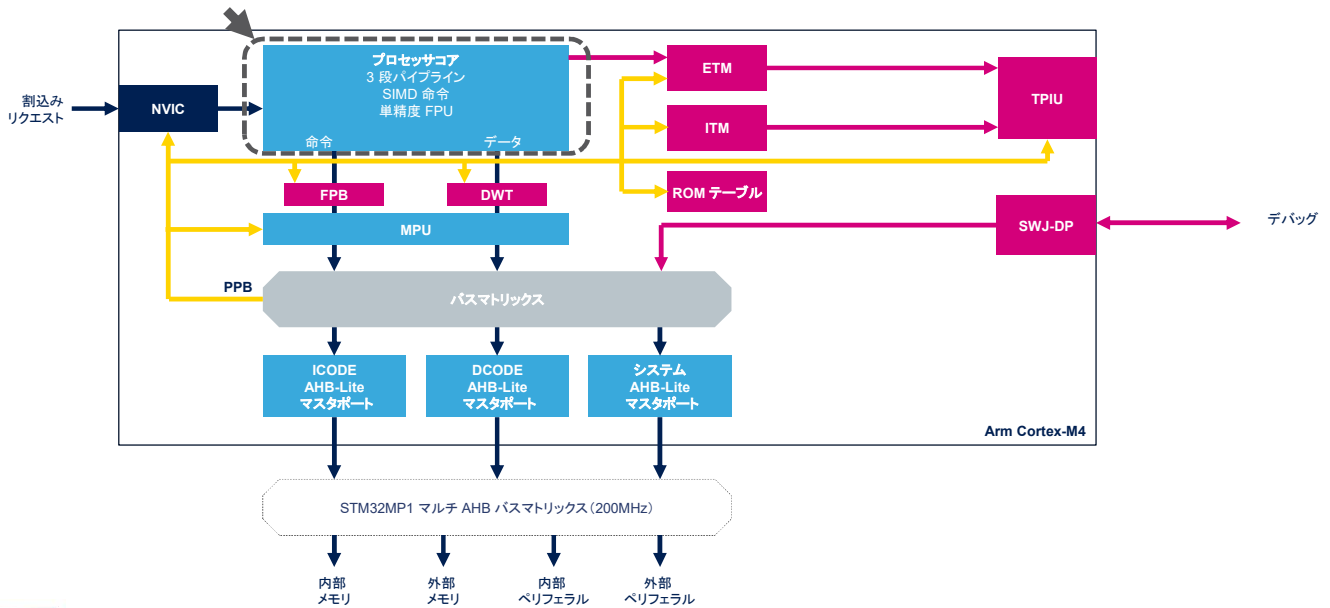
最高性能



STM32MP1 マイクロプロセッサは、32bit プロセッサアーキテクチャの強力なパフォーマンスと特に高レベルの決定論的処理の恩恵を受けるために、Arm Cortex-M4 コアを内蔵しています。すべての Cortex-M CPU は 32bit アーキテクチャです。Cortex-M3 は Arm がリリースした最初の Cortex-M CPU です。その後、Arm は、互換性を維持しながら高性能と低消費電力の2つの製品ラインを区別することにしました。Cortex-M4 は高性能製品ラインに属します。

# コアアーキテクチャの概要

4

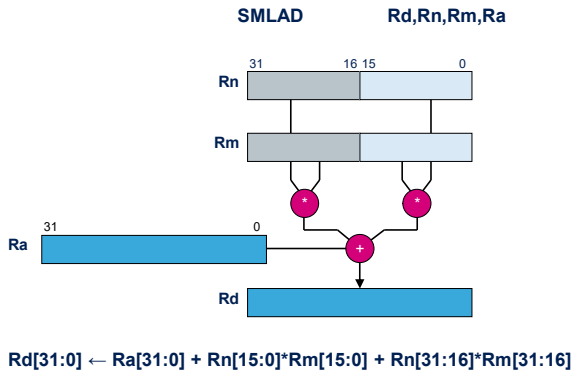


プロセッサコアは、命令フェッチとデータロード／ストアの同時ランザクションをサポートするために、ハーバードアーキテクチャを実装しています。

命令パイプラインには、フェッチ、デコード、実行の3つの段階があります。ターゲット命令を早期にフェッチすることにより、条件分岐の実行が加速されます。

## • SIMD 命令

- ~100命令
- 1サイクルで複数の動作を実行
- 例:デュアル符号付き積和



$$Rd[31:0] \leftarrow Ra[31:0] + Rn[15:0] * Rm[15:0] + Rn[31:16] * Rm[31:16]$$

## • 浮動小数点命令

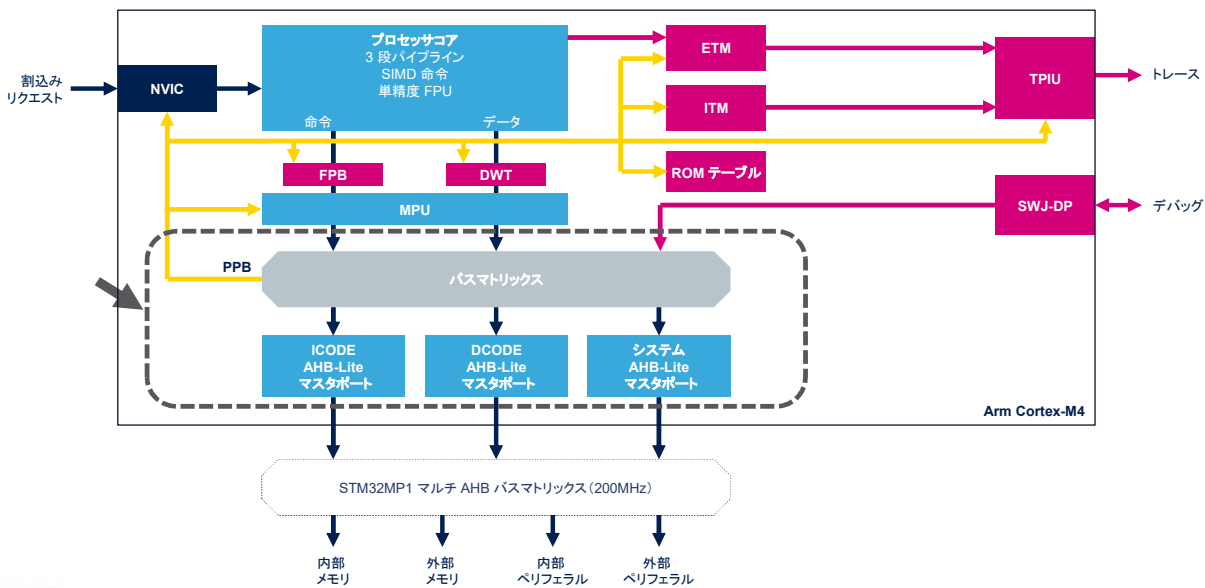
- Cortex-M4 FPU
- IEEE 754 規格準拠
- 単精度浮動小数点演算

動作	サイクルカウント
加算／減算	1
除算	14
積算	1
積和	3
平方根	14



SIMD 技法は、パックされたデータで動作します。たとえば、A/D コンバータで取得した 2 つの 12bit サンプルは、同じ 32bit レジスタの 2 つのハーフワードに格納できます。このスライドで説明されている例では、2 組のサンプルが乗算され、転送先レジスタに累積されます。データ信号処理は積和に基づいているため、SIMD 命令によって、通常のスカラー固定小数点命令に関して性能が向上します。

STM32MP1 マイクロプロセッサに組み込まれた Cortex-M4 コアは、IEEE754 規格に準拠したオプションの単精度浮動小数点ユニットを実装しています。加算、減算、乗算命令の実行には 1 クロックを要し、積和命令には 3 クロック、除算と平方根命令には 14 クロックを要します。



Cortex-M4 にはキャッシュや内部 RAM がありません。したがって、命令フェッチトランザクションとデータアクセスは、内部バスマトリックスに導かれます。このバスマトリックスで、アドレスとアクセスタイプ、命令、またはデータに応じて、出力 AHB-Lite マスタポートが選択されます。一度に 3 つの AHB トランザクションを進行できます (たとえば次のように)。

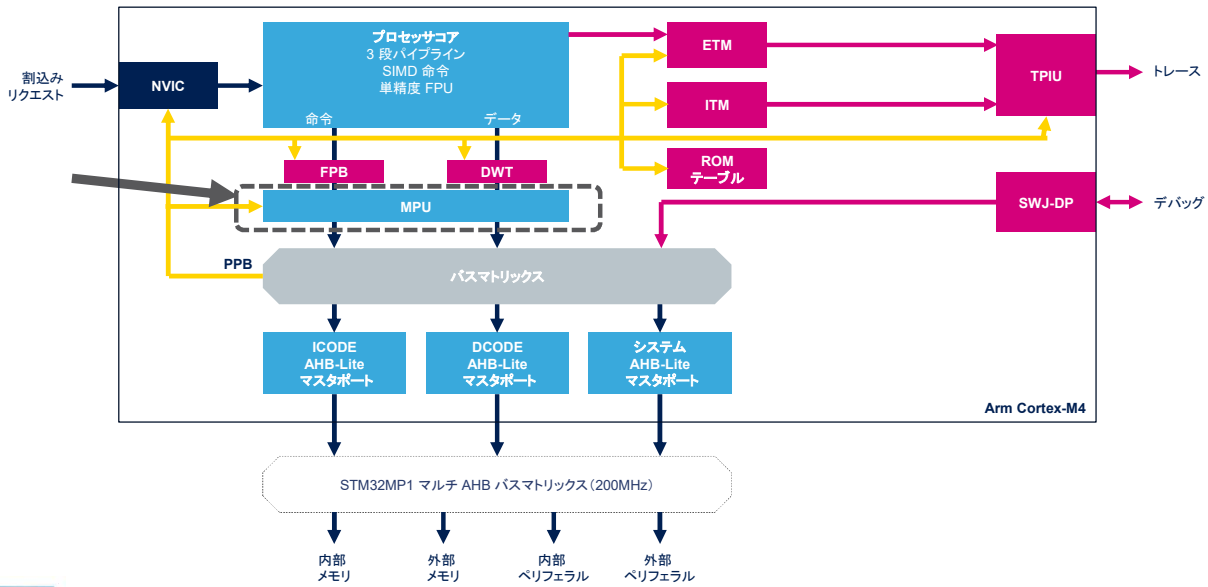
- ICODE マスタポートを使用して Flash メモリから命令をアクセス
- DCODE マスタポートを使用して Flash メモリから定数データをアクセス
- SYSTEM マスタポートを使用して SRAM をアクセス

Cortex-M4 のバスマトリックスは STM32MP1 マルチ AHB バスマトリックスに接続されており、CPU がメモリとペリフェラルにアクセスできるようにしています。トランザクションは AHB-Lite でパイプライン化されるため、最適なスループットは、クロックあたり 32bit のデータまたは命令であり、最小 2クロックの遅延があります。

Cortex-M4 のバスマトリックスの出力の 1 つは、CPU の内部にあるプライベートペリフェラルバス (PPB) です。これは、NVIC、MPU、およびデバッグユニットに存在するメモリマップドレジスタにアクセスするために使用されます。

# コアアーキテクチャの概要

7



Cortex-M4 コアでは、設定されたアクセス許可に従ってメモリ保護ユニット(MPU)を使用してアドレス範囲を保護します。MPU を有効にすると、プロセッサコアによって開始されたアクセスがインターセプトされます。

- MPU の属性設定によりアクセス許可を定義
- 8 つの独立したメモリ領域
  - アクセス許可の制御
    - コードを実行可能？
    - データ書込み可能？
    - 非特権モードアクセス？
  - 属性の割当て
    - ノーマル(メモリ)
    - デバイス(ペリフェラル)
    - ストロングリオーダ(ペリフェラル)



STM32MP1 マイクロプロセッサのメモリ保護ユニット(MPU)は、8 つの独立したメモリ領域をサポートしており、以下の独立したアクセス許可を設定できます。

- アクセス許可: 特権モードまたは非特権モードでの読取り／書込みの許可または禁止
- 実行許可: 実行可能領域または命令フェッチが禁止されている領域

MPU は、通常、デバイス、Strongly-ordered と呼ばれる属性を領域へ割り当てることも担当しています。

通常属性は、メモリをマッピングするために使用されます。

デバイス属性とストロングリオーダ属性は、ペリフェラルをマッピングするために使用されます。両者の違いは、ペリフェラルのアクセス中にデータをバッファリングする機能です。デバイス属性では書込みをポスティングできますが、ストロングリオーダ領域へのストアでは、ターゲットペリフェラルから応答を受信するまでパイプラインが停止します。



- 詳細については、以下の関連資料を参照してください。
  - STM32MP1 シリーズ Cortex-M4 プロセッサのプログラミングマニュアル
  - Managing memory protection unit (MPU) in STM32 MCUs (AN4838)
  - 次のリンク先の Arm ウェブサイト:
    - <http://www.arm.com/products/processors/cortex-m/cortex-m4-processor.php>



life.augmented

NVIC およびデバッグユニットは別のプレゼンテーションで説明しています。

詳細については、[www.st.com](http://www.st.com) ウェブサイトにあるこれらのアプリケーションノートおよび Cortex-M4 プログラミングマニュアルを参照してください。

また、Arm ウェブサイトにアクセスすれば、Cortex-M4 コアの詳しい情報を見ることができます。