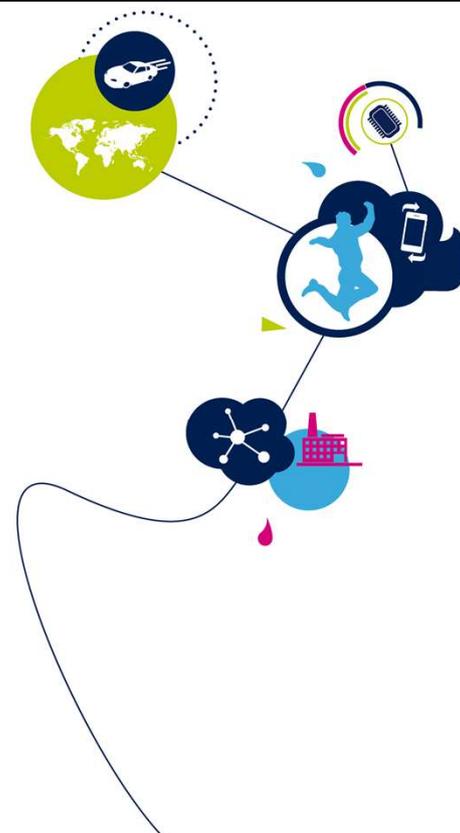


STM32G4 - DBG

デバッグとトレース

1.0版



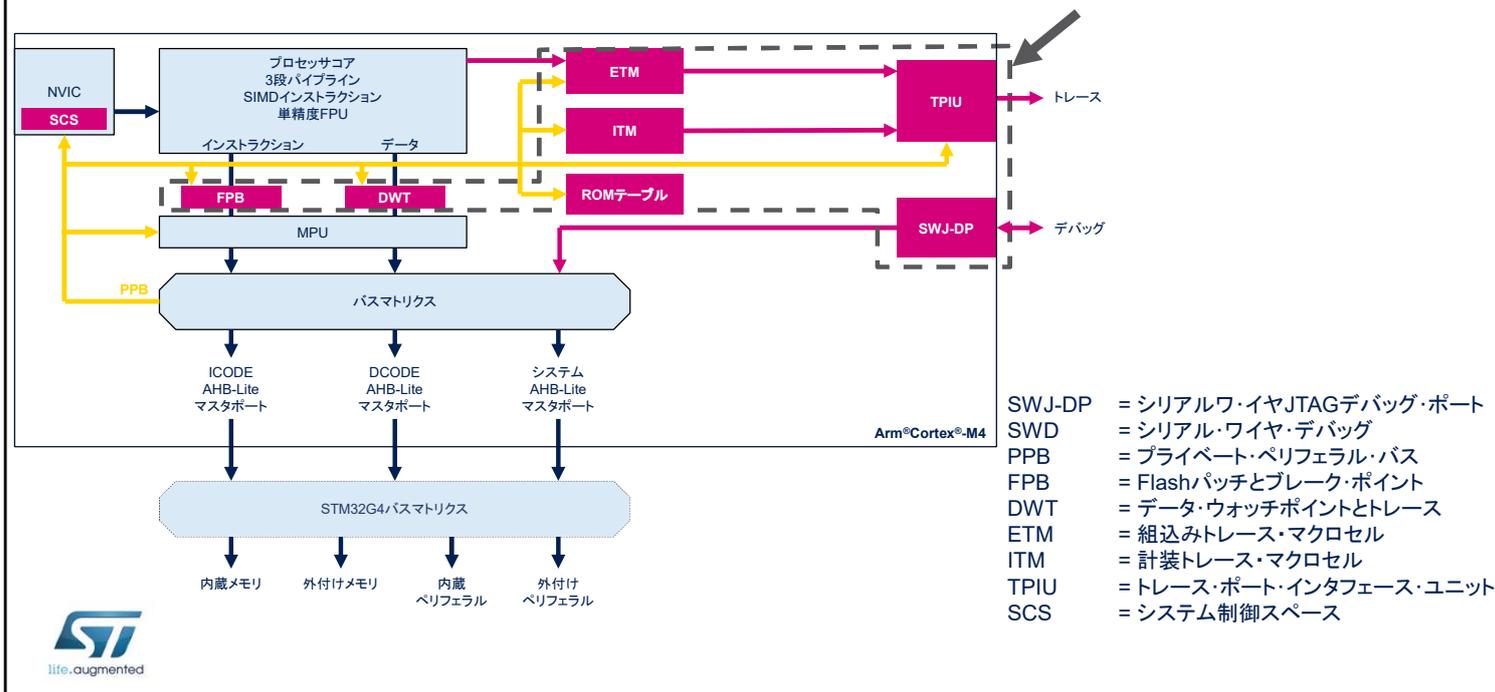
STM32デバッグとトレース用インタフェースのプレゼンテーション
によろこそ。STM32G4デバイスが備えているデバッグの機能につ
いて説明します。



- STM32G4はデバッグのための充実したサポートを提供
 - プログラムをRAMまたはFlashメモリにダウンロード
 - メモリとレジスタの内容を確認
 - ブレーク・ポイントを挿入してプロセッサを停止
 - プログラムを実行またはシングル・ステップ実行
- ARM® CoreSight™アーキテクチャ・ベース
 - 広範囲な互換ツール
 - STリンクなど、プローブ・デバッグするための標準のシリアル・ワイヤ・デバッグ・インタフェース

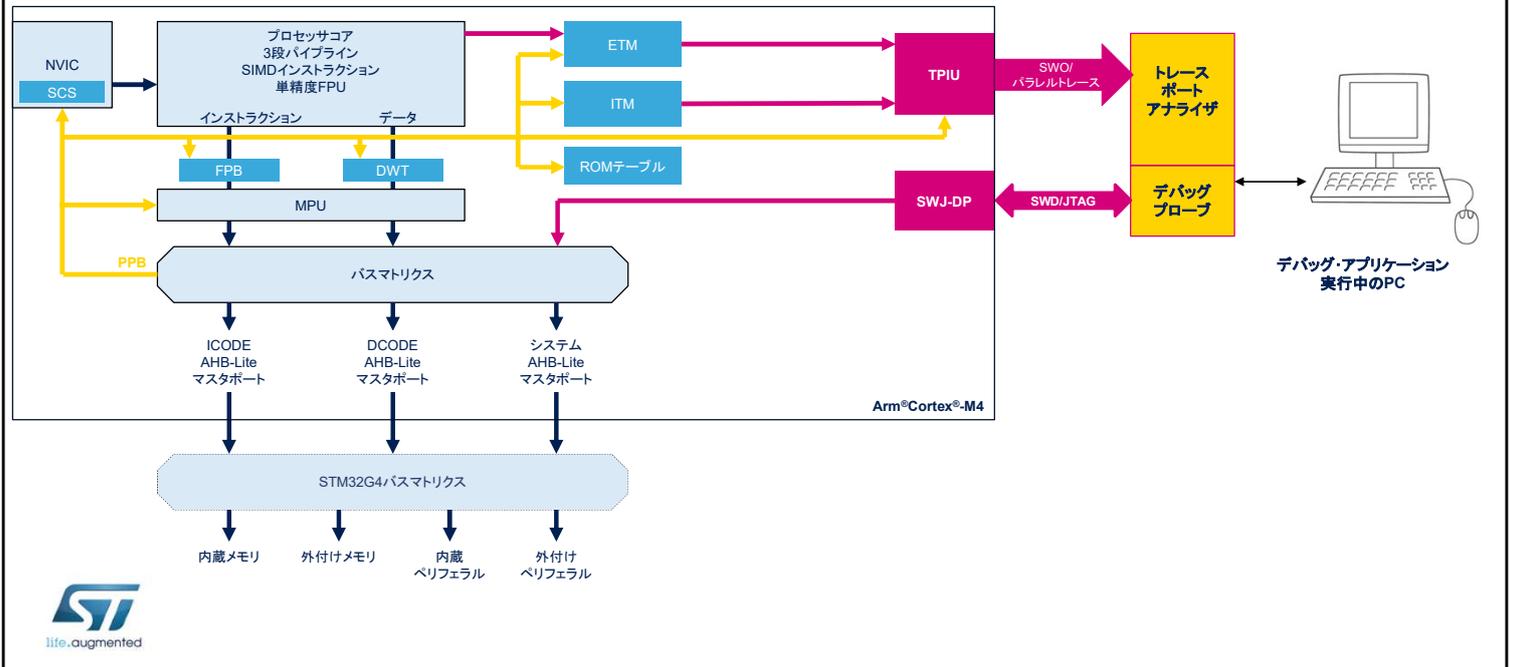
STM32G4には、STM32ファミリMCUが提供するお馴染みのデバッグ機能（Flashダウンロード、ブレークポイントデバッグ、レジスタ表示とメモリ表示）がすべて内蔵されています。デバッグとトレースのインフラストラクチャは、ほとんどのツールプロバイダによってサポートされているARM® CoreSight™規格を採用しています。

デバッグ・アーキテクチャ



デバッグプロセスに関与するすべてのユニットは、コアとSWJ-DPの両方からプライベートペリフェラルバス(PPB)を介してアクセス可能なメモリマップレジスタを持っています。デバッグは、プロセッサの実行中にメモリにマップされたリソースにアクセスできます。たとえば、プロセッサが命令を実行している間に、プライベートペリフェラルバスに接続されているFPBユニットにアクセスし、デバッグによってブレークポイントを設定できます。





SWJ-DPを使用すると、外部デバッグプローブは、Cortex®-M4コアからもアクセス可能なメモリにマップされたリソースにアクセスできます。

メモリ保護ユニット(MPU)は、SWJ-DPによって開始されたリクエストをインタセプトしないことに注意してください。

SWJ-DPは、デバッグプローブとデータを交換する2つのプロトコルをサポートします。

- 2線式シリアルワイヤデバッグ(SWD)プロトコル
- または5線式JTAGプロトコル

SWJ-DPは、使用されているプロトコルを自動的に検出します。

トレース出力に関しては、TPIUは2つの可能性を提供します。

- シリアルワイヤ出力(SWO)と呼ばれる非同期の1線トレースポート
- または、クロック信号と1、2または4ビットのデータを含む同期5線トレースポート

SWOはJTAG JTDO信号で多重化されます。したがって、JTAGプロトコルと同時に使用することはできません。SWDを選択する必要があります。

柔軟なSWJ-DPとトレース・ピンの割り当て

- デバッグが必要ではない場合、すべてのデバッグ・ピンは機能用として再割当て可能

ピンアウト	PA13	PA14	PA15	PB3	PB4	PE2	PE3	PE4	PE5	PE6
SWD	SWDIO	SWCLK								
JTAG	JTMS	JTCK	JTDI	JTDO	JTRST					
SWO				TRACESWO						
トレース・ポート						TRACECK	TRACED0	TRACED1	TRACED2	TRACED2

- コントロールされていないI/Oのレベルを回避するため、デバイスはJTAG入力ピンに内部プルアップとプルダウンを設定



SWJ-DPのSTM32G4からの出力として、5本のピンが汎用I/O (GPIO)の代替機能として使用されます。

これらのピンは、すべてのパッケージで使用できます。

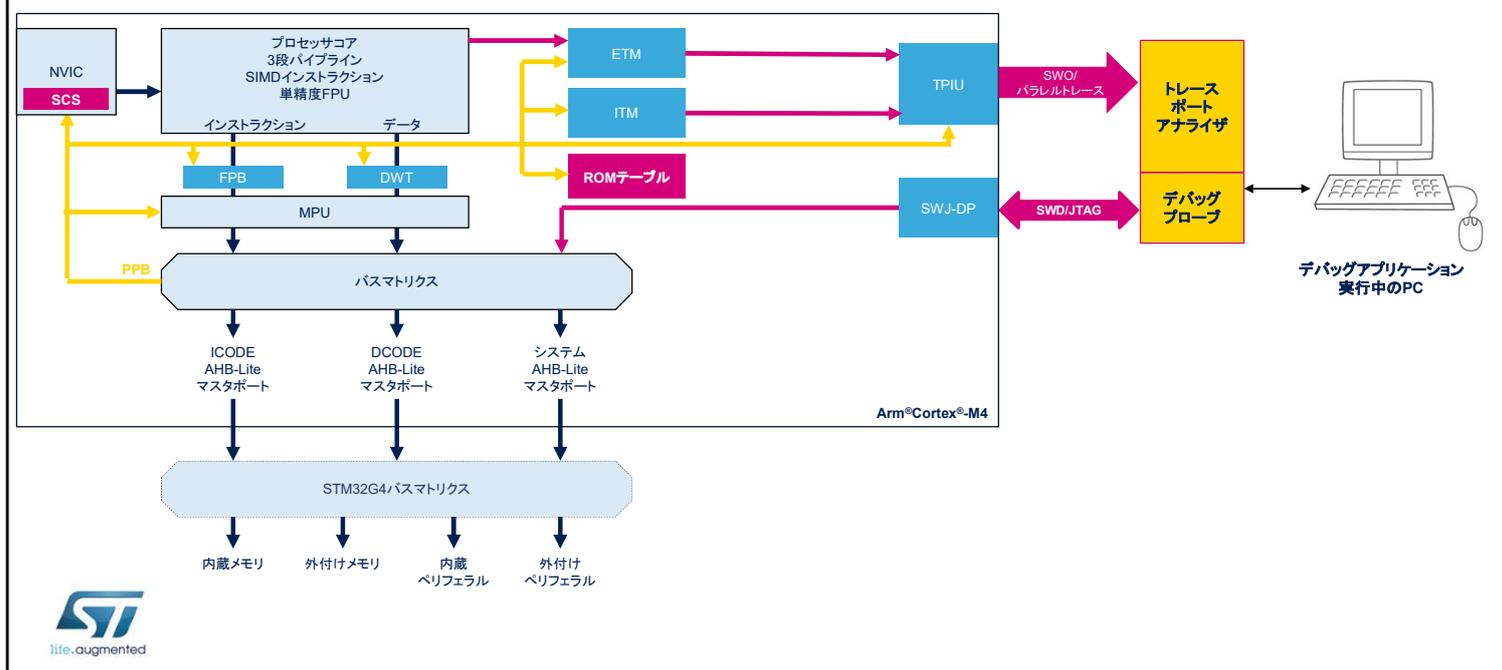
リセット解除後、SWJ-DPに使用される5つのピンはすべて、デバッグホストとして使用できる専用ピンとしてすぐに割り当てられます。

ただし、STM32G4 MCUは、SWJ-DPポートの一部またはすべてを無効にする可能性があるため、接続を切断したままにできるが、デバッグ接続を失うことなくGPIOとして使用できないNJTRSTを除き、GPIOの使用のために関連ピンを解放する可能性があります。

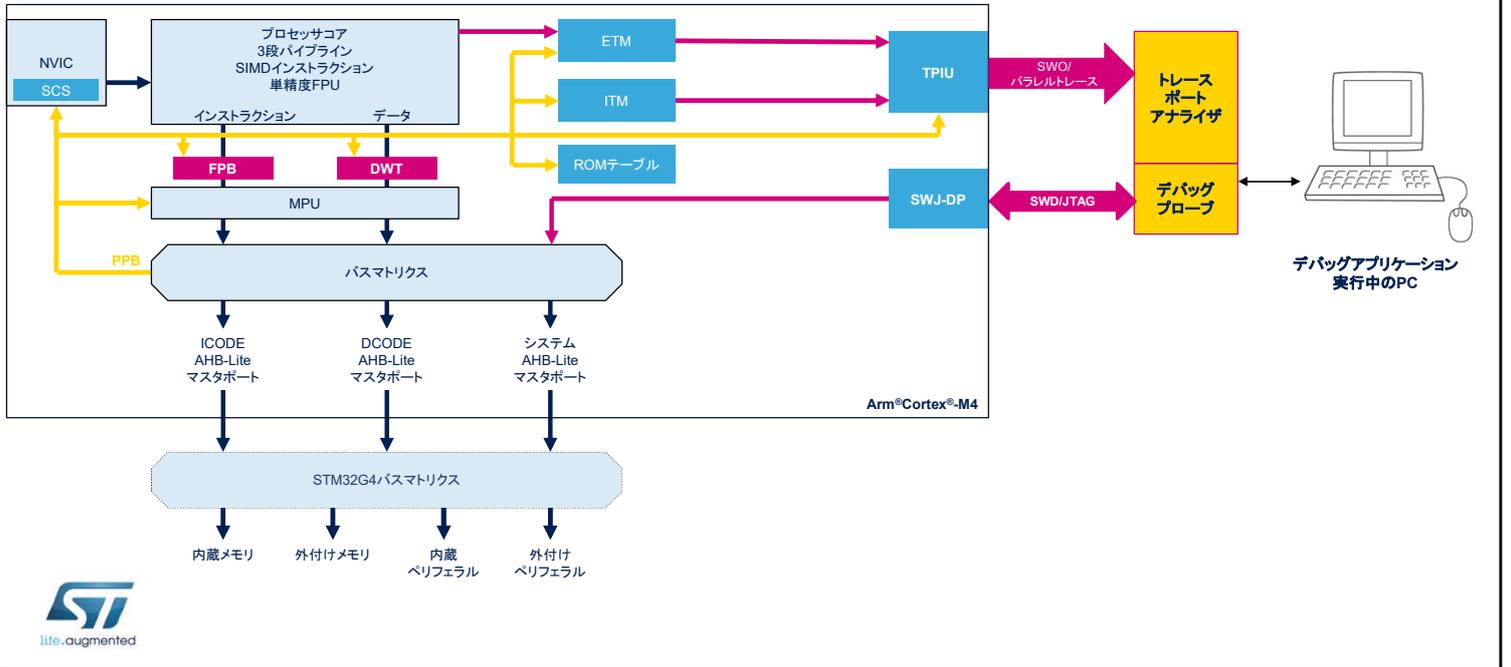
注意: デバッグホストによって明示的にプログラムされている場合を除き、パラレルトレースポートは割り当てられません。

JTAGピンには内部プルアップとプルダウンがあり、デフォルトはアクティブです。

- プルアップオンNJTRST、JTDI、JTMS/SWDIO
- プルダウンオンTCK/SWCLK



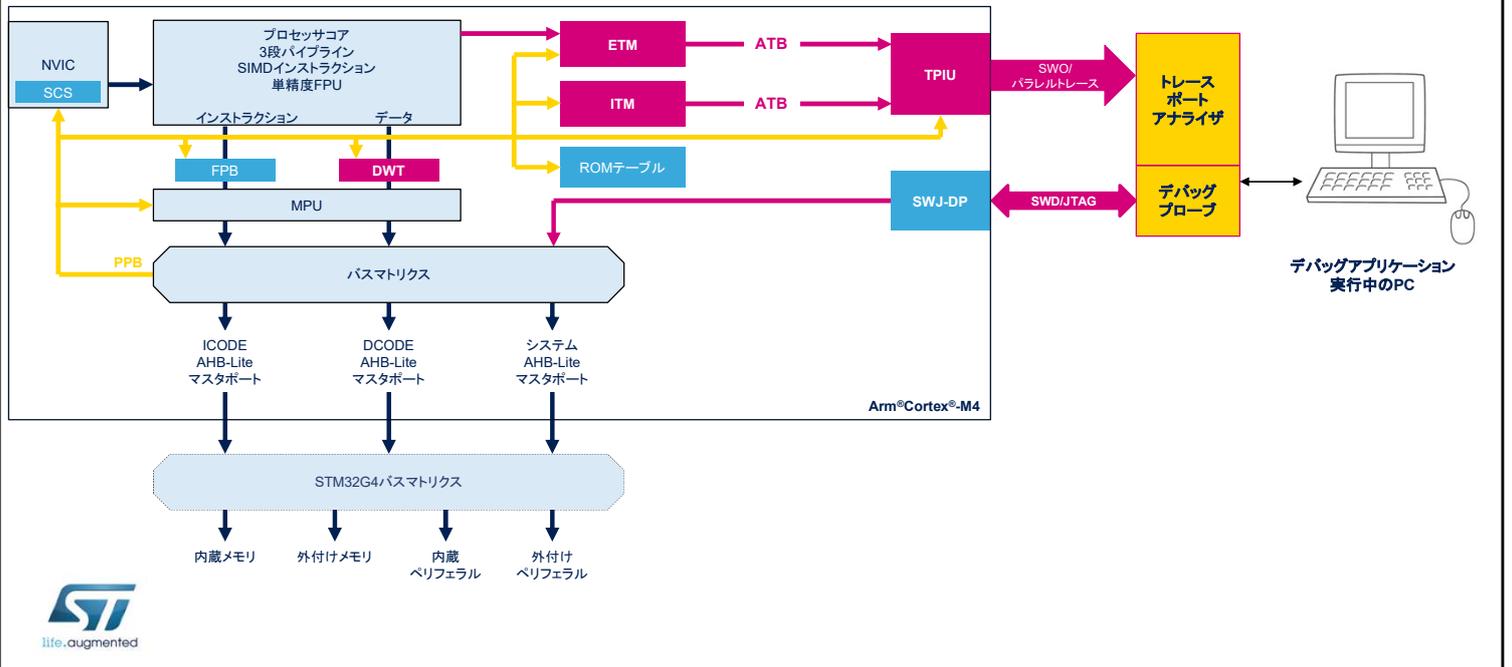
ROMテーブルには、コアから見る事が出来る各デバッグコンポーネントのベースアドレスへのポインターが含まれています。一部のデバッグツールでは、ターゲットのCoreSight™インフラストラクチャのトポロジを自動的に検出するために使用されます。SWJ-DPにはROMテーブルを指す読み取り専用レジスタが含まれています。SCS(システム制御スペース)は、CPUの参照とリビジョンを示すCPUIDレジスタを含むレジスタのブロックであり、デバッガがHALTモードへの入り口と終了を制御するために使用します。



侵略性デバッグは、デバッグイベントが発生したときにプロセッサを停止します。

2つのユニットは、侵略性デバッグに関与しています:

- フラッシュパッチとブレイクポイント (FPB)
- データウォッチポイントとトレース (DWT)



非侵襲性デバッグは、プロセッサのパフォーマンスに影響を与えることなく、プロセッサで何が起こるかをトレースすることを目的としています。これはリアルタイムトレースに基づいています。

ETMおよびITMモジュールは、トレースパケットを合成し、AMBAトレースバス(ATB)を介して発行する機能を備えています。

TPIUはトレースパケットを受信し、外部インタフェースを介して出力するため、トレースポートアナライザ(TPA)はパケットをキャプチャできます。

通常、トレースポートアナライザはデバッグプローブと同じデバイスに含まれています。

TPAにトレースパケットを転送するために2つのインタフェースがサポートされています。

- 非同期式シングルワイヤ出力(SWO)
- または同期式パラレルトレースポート

TPIUで競合が検出されると、オーバフロー状態が発生します。これはトレースポートの帯域幅が、すべてのトレースパケットをエクスポートするのに十分な大きさではないことを意味します。

トレースパケットはITMモジュールとETMモジュールでタイムスタンプ付きであることに注意してください。

Flashパッチとブレークポイント(FPB)

- FPBはハードウェアのブレークポイントを設定可能
 - これは、不揮発性メモリにブレークポイントを設定できる6つのコンパレータが含まれる
 - 2つのコンパレータがデータ・アクセスをモニタリング
- FPBは、STM32G4のFlashメモリの0~0x1FFFFFFFの範囲内のアドレスコンペアのみをサポート
 - アドレス0x20000000以上では、コードはRAMから実行しDWTを使用してデータ・アクセスを停止する場合に、ソフトウェア・ブレークポイントが使用可能
- これらの8つのコンパレータのどれでも、CPUを停止する代わりにインストラクションの変更またはデータを出力するように設定可能



Flashパッチとブレークポイントユニット(FPB)では、ハードウェアのブレークポイントを設定できます。

これは、不揮発性メモリにブレークポイントを設定する6つのコンパレータが含まれています。一致が検出されると、命令フェッチアドレスを監視し、ブレークポイント命令を返します。ブレークポイント命令が実行されると、プロセッサはデバッグモードで停止します。

2つのコンパレータは、不揮発性メモリに存在する特定の定数がアクセスされたときにプロセッサを停止するために使用されるデータアクセスを監視します。

FPBは、STM32G4のFlashメモリに対応する0~0x1FFFFFFFの範囲内のアドレスコンペアのみをサポートします。

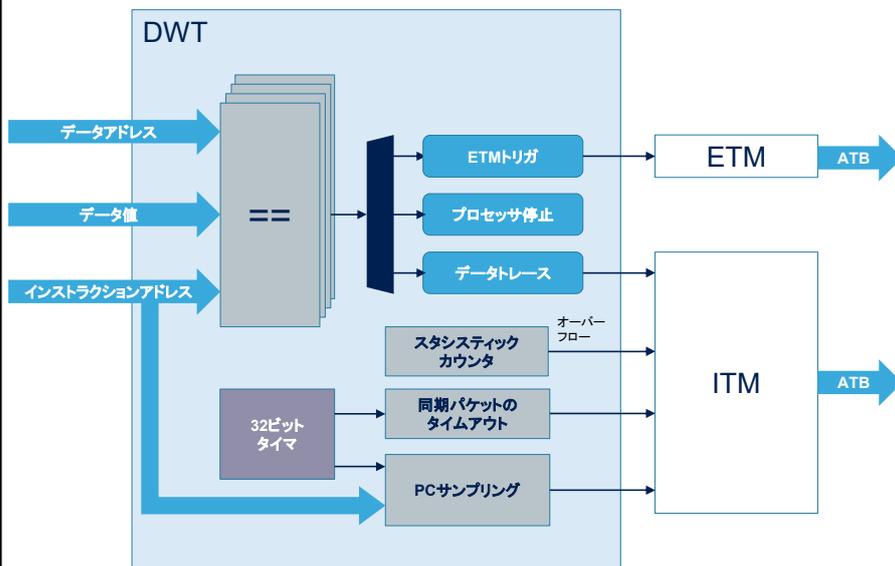
アドレス0x20000000以上では、RAMからコードを実行し、DWTを使用してデータアクセスを停止する場合に、ソフトウェアブレークポイントを使用できます。

ソフトウェアブレークポイントを使用すると、デバッガは、ユーザーが停止する命令をBKPTという専用の命令に置き換えます。

これらの8つのコンパレータのどれでも、CPUを停止する代わりにインストラクションの変更、または定数データを出力するように設定できます。

データ・ウォッチポイントとトレース・ユニット

10



- DWTは自律的にデータ・ウォッチポイントを設定、およびトリガするために使用可能
- ETMトリガ信号をアサートするようにコンパレータを設定できる
- DWTモジュールはITMと連携し、さまざまなリアルタイム・トレース機能をサポート:
 - データ・トレース
 - スタティスティック・カウンタ・オーバーフロー
 - 定期的なPCサンプリング



DWTは、単にデータウォッチポイントを実装するよりも多くの機能をサポートします。

図の左側にある4つのコンパレータは、データアドレスまたは命令アドレスの一致条件を検出します。

コンパレータ1は、データ値の一致条件を検出するように構成することもできます。

一致条件が発生した場合、次のアクションがサポートされます。

- コアを停止、これはウォッチポイント機能
- ETMへのトリガ信号のアサート
- データアクセスを報告し、それをITMに渡してエクスポートできるようにトレースパケットを生成します。トレース情報を補完するために、現在のデータとPCの値をキャプチャすることもできます。

複雑な条件、混合アドレスとデータ比較はプログラム可能です。

比較ロジックに加えて、DWTには8ビットスタティスティックカウンタが含まれています。それらのいずれかにオーバーフロー条件が発生すると、DWTは、このオーバーフローをデバッガに報告するトレースパケットを発行するようにITMに通知します。

DWTは、トレースポートアナライザとの同期を維持するために必須の同期パケットの定期的な生成をトリガも担っています。

これは32ビットタイマによって実現され、トレースパケットに格納するためにITMに渡されるPCの値の定期的なサンプリングをトリガするためにも使用できます。

• ウォッチポイントの機能

- DWTは命令アドレスとデータ・アドレスを比較できる4つのコンパレータを提供
 - 選択可能な方向(読取り、書込み、またはその両方)で特定のデータにアクセスしようとする試みを検出するのに非常に便利
 - 命令アドレスの比較では、ウォッチポイント・イベントはフェッチ時に取られる

• 統計カウンタ

- DWTは、プロセッサのシステム・プロファイリングを提供
- カウンタが含まれている:
 - クロックサイクル数、フォールドされた命令数、ロード・ストア・ユニット(LSU)の動作数、スリープ・サイクル数、命令当たりのサイクル数、例外マイクロコード実行数



DWTは、現在のデータまたは命令アドレスと、デバッガによってプログラムされたコンパレータの内容との一致によって発生するウォッチポイントイベントをトリガします。

アドレスの照合については、アドレスの範囲に照合出来るように、マスクを使用できます。

照合が成立すると、コンパレータはPC値またはアクセスされたデータアドレスに対してウォッチポイントデバッグイベントを生成します。

データ値との一致は、アドレス一致と組み合わせることが出来ます。

FPBは、命令が実行ユニットに入ろうとしたときにブレークポイントイベントが発生するため、命令ブレークポイントの実装に適しています。したがって、フェッチされた命令が、分岐によって破棄された場合、ウォッチポイントイベントが発生する間、ブレークポイントイベントは発生しません。

R0などのコアレジスタを読み書きするには、デバッガはコアを停止する必要があります。

DWTには、プロセッサのシステムプロファイリングを容易にする統計カウンタが含まれています。

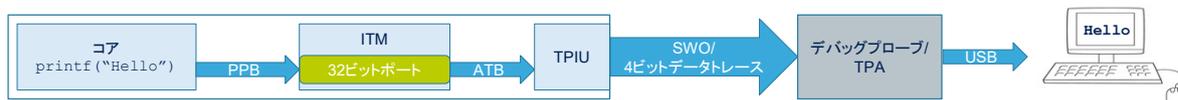
たとえば、カウンタは、例外の取得と終了を担当するマイクロコードがアクティブになっているクロック数をカウントします。

計装トレース・マクロセル (ITM)

12

- ITMトレース・パケットの4つのソースは:

- ソフトウェア・トレース
 - 32個のスティムラス・レジスタのいずれかへのソフトウェア書込み



- ハードウェア・トレース
 - DWTはパケットを生成し、ITMはパケットを出力
- ローカル・タイムスタンプ
 - ITMトレース・パケットのタイミング情報を提供
- グローバル・タイムスタンプ
 - タイムスタンプ生成コンポーネントから取得されるシステム全体の64ビットカウント値を使用して生成



2種類のトレース情報がITMによって処理されます。

- DWTモジュールで説明されているハードウェア・トレース
- ソフトウェアトレースは、デバッガウィンドウに表示されるデータをソフトウェアが送信できるようにします。

32のITM32ビットポートのいずれかでソフトウェアによって書き込まれたデータは、トレースパケットのペイロードでホストデバッガに転送されます。

printf関数は、コードインストルメンテーションを容易にするためにITMポートにリダイレクトできます。

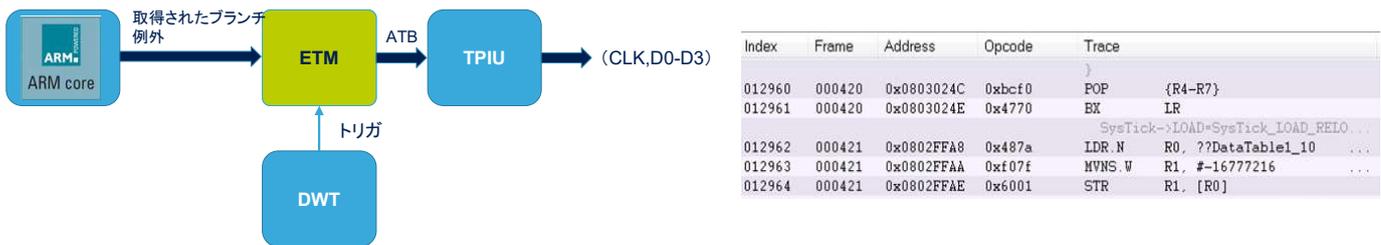
ITMユニットでは、2つの連続したローカルタイムスタンプパケットとグローバルタイムスタンプの相対時間を示すローカルタイムスタンプという2種類のタイムスタンプがサポートされています。

グローバルタイムスタンプは、ETMユニットでも使用される専用の48ビットから来ています。

組み込みトレース・マクロセル(ETM)

13

- ETMは、プロセッサの命令トレースを提供するリアルタイム・トレース・モジュール
 - ETMは、クロックと最大4つのデータ出力TRACECLK、および4つのデータ出力TRACEDATA(3:0)からなるTPIUで 사용되는



SWOは十分な帯域幅を提供しないため、ETMは4ビットのデータトレースポートを必要とします。

ITMとは異なり、コードインストルメンテーションは必要ありません。

取得されたブランチと例外イベントは、ETMユニットに対してCortex®-M4コアによって通知され、ATBポートに出力されるトレースパケットに変換されます。

トレースがキャプチャされると、デバッガはトレースを解凍して、実行されたコードの完全な逆アセンブリ(シンボル付き)を提供します。デバッガはまた、これを元の高レベルソースコードにリンクして、ターゲットでコードがどのように実行されたかを視覚化します。

ETMはCortex®-M4によって実行されるコードをプロファイリングし、複雑なソフトウェアバグを調査するのに役立ちます。

- 「MCUデバッグ」ブロックはデバイス固有のデバッグ機能を有効化
 - デバイスID
 - デバイスIDコード・レジスタを読み取るための標準の場所
 - 低電力モードのエミュレーション
 - デバイスが低電力モード(SLEEP、STOP、STANDBY)になったときに、デバッグ・アクセスが引き続き可能となるように電力とクロックを維持
 - デバッグモード時のペリフェラルクロックの「停止」
 - プロセッサが停止している間は、RTC、TIM、LPTIM、およびウォッチドッグ(IWDG、WWDG)タイマのカウンタ、ならびにSMBUSおよびFDCCANタイムアウト・カウンタを停止
 - トレース・ピンの割当ての制御



MCUデバッグコンポーネント以下を提供します。

- MCU ID
- 低電力モード
- ブレークポイント時のタイマ、ウォッチドッグ、I2C、およびbxCANのクロック制御
- トレースピンの割り当ての制御

DBG_IDCODEレジスタは、STM32標準フォーマットのデバイスIDおよびリビジョンコードが格納されています。この情報は、SWJ-DPまたはユーザーソフトウェアからアクセスできます。

低電力モードのエミュレーションは、低電力モードに入ってもデバッグとの接続が失われないことを意味します。この機能により、低電力に入力されるコマンド(WFI/WFEなど)をwhile()ループに置き換える必要がなくなります。終了時に、デバイスはエミュレーションがアクティブでなかった場合と同じ状態になります(低電力モードエミュレーション中にデバッグが行った変更を除く)。

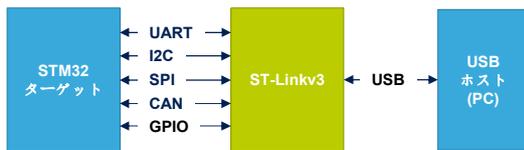
ペリフェラルクロックのフリーズは、デバッグを用いてウォッチドッグを再設定をする必要が無く、デバッグ中にウォッチドッグタイムアウトがデバイスをリセットするのを防ぐのに特に便利です。また、タイマ値の検査と、対応する割り込みを「通常の」操作が再開するまでサスペンドすることが可能です。

DBGMCU_CRLレジスタには、トレースポートを構成する制御フィールド(非同期SWOモードまたはTRACEDATAサイズが1、2、または4の同期モード)を設定するフィールドが含まれています。

- パフォーマンス向上

	ST-Link V2 (KHz)	ST-Link V3 (KHz)
SWD	4000	24000
JTAG	9000	21333

- ブリッジ機能の概要



- STLINK-V3SETは複数のプロトコルのSTM32ターゲットとの通信を可能にする独自のUSB インタフェースを提供:SPI、I²C、CAN、UART、GPIO
- このインタフェースは、ターゲット・ブートローダと通信するために使用可能



ST-LinkはSTによって開発されたデバッガで、SWおよびJTAGデバッグプロトコルとSWOTレースポートをサポートします。

ST-Linkの第3世代は、SWDおよびJTAGの早い通信速度をサポートしています。

ST-LinkV3は、UART、I2C、SPI、CAN、またはGPIOを使用してターゲット上のデータを送受信するために使用できます。同時にアクティブにできるのは、これらのインタフェースの1つだけです。たとえば、STM32G4のUARTは、ST-LinkV3経由でホストPCに接続しテスト可能です。キャラクタストリームは、USB経由で透過的に転送されます。

さらに、ST-LinkV3では、ホストPCからのUARTまたはCANリンクを使用してユーザイメージのダウンロードがサポートされ、STM32G4のリセットデアサーションでブーストラップモードが選択されます。

- STM32CubeProgrammerは、既存のプログラミング・ソフトウェア・ツールを1つのソリューションに統合します
- コマンドラインおよびグラフィカル・ユーザ・インタフェースで利用可能
- 主な特長
 - マルチプラットフォーム (Windows®、Linux®、macOS®)
 - デバッグ・インタフェースおよびブートローダ・インタフェースのサポート
 - JTAG/SWD (ST-LinkV2、V3)
 - ブートローダ・インタフェース (UART、USBデバイス・ファームウェア・アップグレード・クラス)
 - ST-LinkV3ブリッジ
 - STM32内蔵/外付けFlashの消去、プログラム (セキュア、またはノン・セキュア)
 - オプション・バイト・プログラミング



STM32CubeProgrammerはSTM32のマイクロコントローラをプログラミングするためのオールインワンマルチOSソフトウェアツールです。

STM32CubeProgrammerはGUI(グラフィカルユーザインタフェース)およびCLI(コマンドラインインタフェース)バージョンで提供されます。

デバッグインタフェース(JTAGとSWD)とブートローダインタフェース(UARTとUSB)の両方を使用して、デバイスメモリの読取り、書込み、およびベリファイを行うための、簡単で効率的な環境を提供します。

追加された機能は、:

- マルチOSサポート:Windows®、Linux®、macOS®
- デバッグインタフェースおよびブートローダインタフェースのサポート
- ST-LinkV3ブリッジ

STM32CubeProgrammerは、USBデバイスにファームウェアイメージをダウンロードするように設計されたデバイスファームウェアアップグレード(DFU)USBクラスをサポートしています。