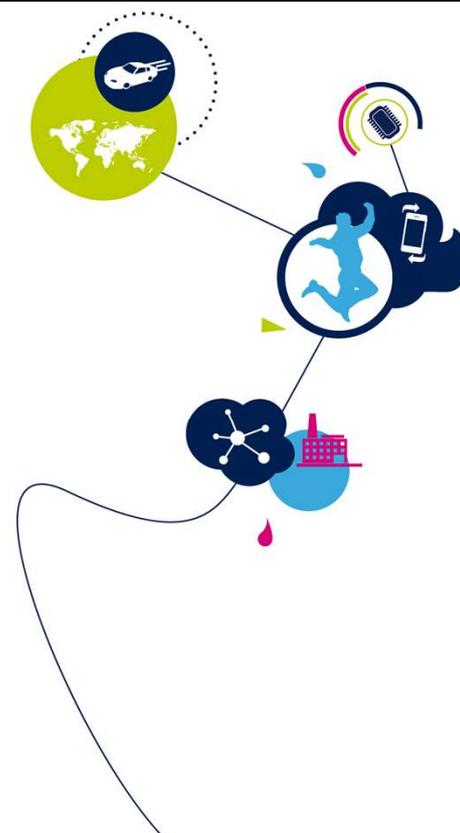
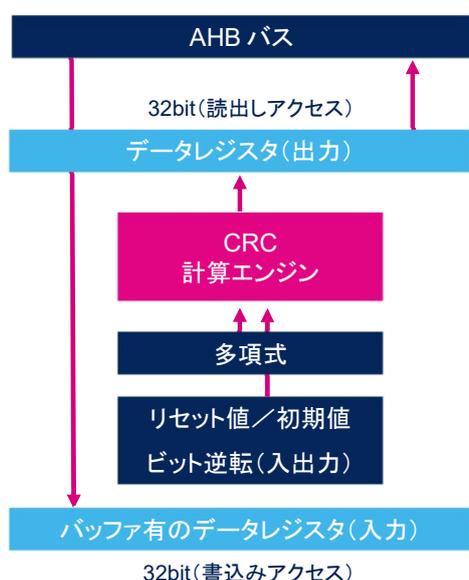


STM32MP1 - CRC

巡回冗長検査計算ユニット
1.0 版



STM32 巡回冗長検査計算ユニットのプレゼンテーションによろ
そ。ここでは、エラーの検出に使用されるコードを生成する、この
ユニットのすべての機能について説明します。



- STM32MP1 シリーズでは、2 つのインスタンス (CRC1 と CRC2) が使用できます。
- 8、16、32bit データから CRC コードを生成するために使用されます。
 - 完全に設定可能な奇数多項式
 - プログラム可能なサイズ
 - 入出力データの完全な可逆性

アプリケーション側の利点

- データ整合性の確認
- ソフトウェアのコードシグネチャの生成
- CPU または DMA による使いやすさ



CRC とは、巡回冗長検査を示します。CRC 計算ユニットは、設定可能な多項式の値とサイズを使用して、8、16、32bit データに CRC コードを生成するために使用されます。CRC ベースの技法は、データ転送やストレージの整合性を確認するために使用できます。

CRC 計算ユニットは、実行時にアプリケーションソフトウェアのシグネチャ計算にも使用できます。リンク時に生成されて、特定のメモリ領域に保存されたリファレンスシグネチャと計算されたアプリケーションソフトウェアシグネチャが比較されます。

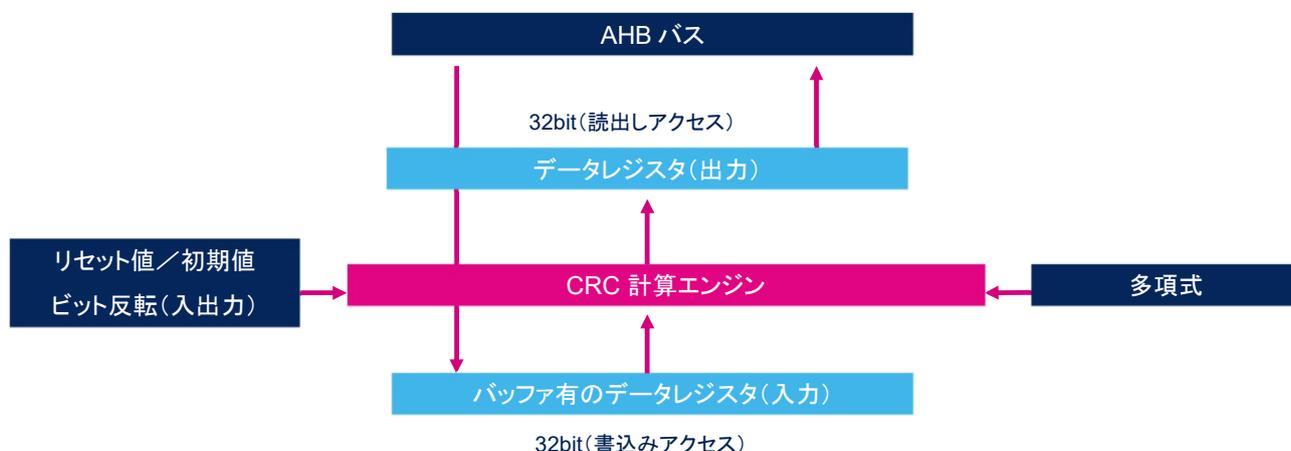
STM32 CRC 計算ユニットは完全に設定可能であるため、ソフトウェアの負荷が最小限に軽減されます。さらに、DMA コントローラは、CPU が他のタスクを実行している場合や SLEEP モードの場合に、大きなデータブロックの連続 CRC 計算に使用できます。

STM32MP1 シリーズでは、2 つのインスタンス (CRC1 と CRC2) が使用できます。

- 完全にプログラム可能な奇数多項式：
 - デフォルトでは、CRC-32 (Ethernet) 多項式 0x04C11DB7 を使用します。
 - 多項式は $x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$ で表されます。
- プログラム可能な多項式サイズ (7、8、16、32bit)
- プログラム可能な初期値 (デフォルト = 0xFFFFFFFF)
- 入出力データのさまざまなエンディアン方式



STM32 に組み込まれている CRC 計算ユニットは、7、8、16、32bit でサイズをプログラムできる完全にプログラム可能な奇数値の多項式を備えています。非常に柔軟に実行時の CRC コードを生成できるように、初期値も完全にプログラム可能です。ペリフェラルは、さまざまな通信プロトコルをサポートした入出力データのビッグエンディアン形式やリトルエンディアン形式をサポートするように設定できます。



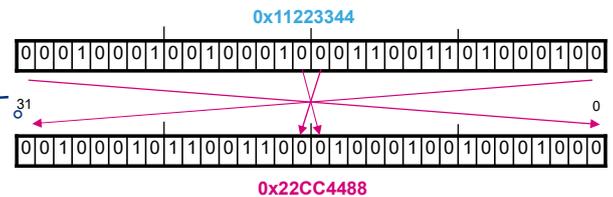
CRC 計算ユニットは、最適なパフォーマンスを発揮できるように AHB バスインタフェースに接続されます。

これには、CPU や DMA コントローラによる書込みと読出し両方に使用される単一の 32bit レジスタが含まれています。

初期値、入出力のビット反転、多項式係数は、新しい CRC 計算を開始する前の初期化フェーズ中にソフトウェアで設定されます。CRC 計算エンジンでは、前の CRC 計算の結果が保持され、次の計算の初期値になり、データブロックの新しい CRC 計算が可能になります。

軽減されるソフトウェアの負荷

- 入力データは 8、16 または 32bit で反転できます。
 - CRC_CR レジスタの REV_IN[1:0] の値に依存します。
 - たとえば、入力データ 0x1A2B3C4D は、CRC 計算では以下のように使用されます。
 - 0x58D43CB2 (バイト単位でビットが反転)
 - 0xD458B23C (ハーフワード単位でビットが反転)
 - 0xB23CD458 (フルワードでビットが反転)
- 出力データは、32bit で反転できます。
 - CRC_CR レジスタの REV_OUT ビットをセットします
 - 次は出力データ 0x11223344 の例です。



入出力フォーマットはソフトウェアの負荷を軽減するために完全に設定可能です。これによって、アプリケーションは異なるエンディアン方式を管理できます。

入力データは、CRC_CR レジスタの REV_IN[1:0] ビットに応じて、8、16、または 32bit で反転できます。たとえば、入力データ 0x1A2B3C4D は、CRC 計算では以下のように使用されます。

- 0x58D43CB2 (バイト単位でビットが反転)
- 0xD458B23C (ハーフワード単位でビットが反転)
- 0xB23CD458 (フルワードでビットが反転)

また、出力データも、CRC_CR レジスタの REV_OUT ビットをセットすることによって逆にすることができます。操作はビットレベルで行われます。たとえば、出力データ 0x11223344 は 0x22CC4488 に変換されます。

- シンプルなファームウェア操作の入出力 32bit データレジスタ 1 個
- AHB バスを停止せずに連続で書き込むための入力バッファ
- 入力データ幅に依存する CRC 計算の時間:
 - 32bit ブロックの場合は 4AHB クロックサイクル
 - 16bit ブロックの場合は 2AHB クロックサイクル
 - 8bit ブロックの場合は 1AHB クロックサイクル
- CPU が SLEEP モード時は DMA で書込み可能
 - 低電力動作(連続した計算)で、時間の影響を受けるタスクでは CPU を解放



CRC データレジスタには、アクティブな CRC 計算によるウェイトステートを待つことなく、CPU や DMA がすぐに別のデータワードを書き込める入力バッファがあります。

CRC データレジスタは、ワード、右詰め of ハーフワード、右詰め of バイトによってアクセスできます。

計算の時間はデータ幅に依存します。

- 32bit ブロックの場合は 4AHB クロックサイクル
- 16bit ブロックの場合は 2AHB クロックサイクル
- 8bit ブロックの場合は 1AHB クロックサイクル

DMA コントローラは CRC エンジンへの書込みに使用できます。これによって、他のタスク用に CPU の負荷を軽減し、CPU をコードやデータの整合性を確認するための連続した計算に使用できます。

モード	説明
RUN	アクティブ
SLEEP	アクティブ。DMA コントローラは CRC 計算に使用できます。
STOP	停止。ペリフェラルレジスタの内容は保たれます。
STANDBY	パワーダウン状態です。CRC ユニットの再初期化は、STANDBY モード終了後に再初期化する必要があります。

このスライドでは、CRC 計算ユニットを使用できる電力モードについてまとめています。CRC 計算ユニットでは、動作にメインの AHB クロックが必要です。