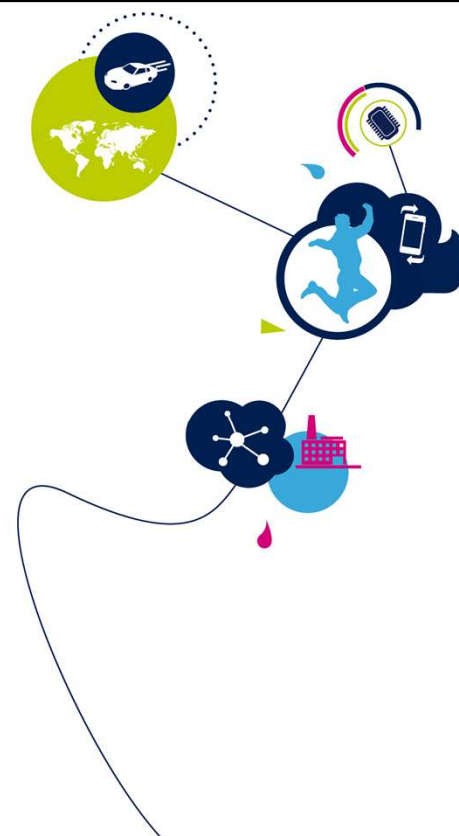


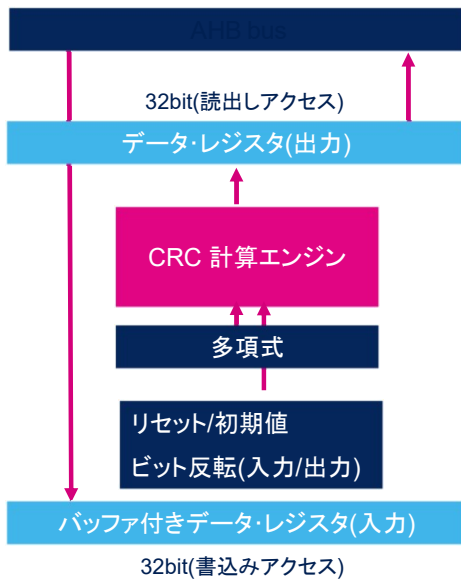
STM32G4 - CRC

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、エラー検出に使用されているコードを生成する
CRCユニットの機能のすべてをカバーしています。



- 8、16、32bitデータからのCRCコード生成に使用
 - 完全に設定可能な奇数多項式
 - プログラム可能なサイズ
 - 入出力データは完全可逆

アプリケーション側の利点

- データ整合性を検証
- ソフトウェア・コード・シグネチャを生成
- CPUとDMAのどちらでも簡単に使用可能

CRC とは、巡回冗長検査のことです。CRC 計算ユニットは、設定可能な多項式値とサイズを用いて 8、16、32bit データのCRC コードを生成するために使用されます。CRC に基づくテクニックは、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 計算は、実行時の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ソフトウェアシグネチャが比較されます。

STM32 のCRC 計算ユニットは完全に設定可能ですので、ソフトウェアのオーバーヘッドは最小限に抑えられます。さらに、DMA コントローラは、CPU が別のタスクを実行しているか、SLEEP モードにある間に、大きなデータブロックに対する連続 CRC 計算のために使用できます。

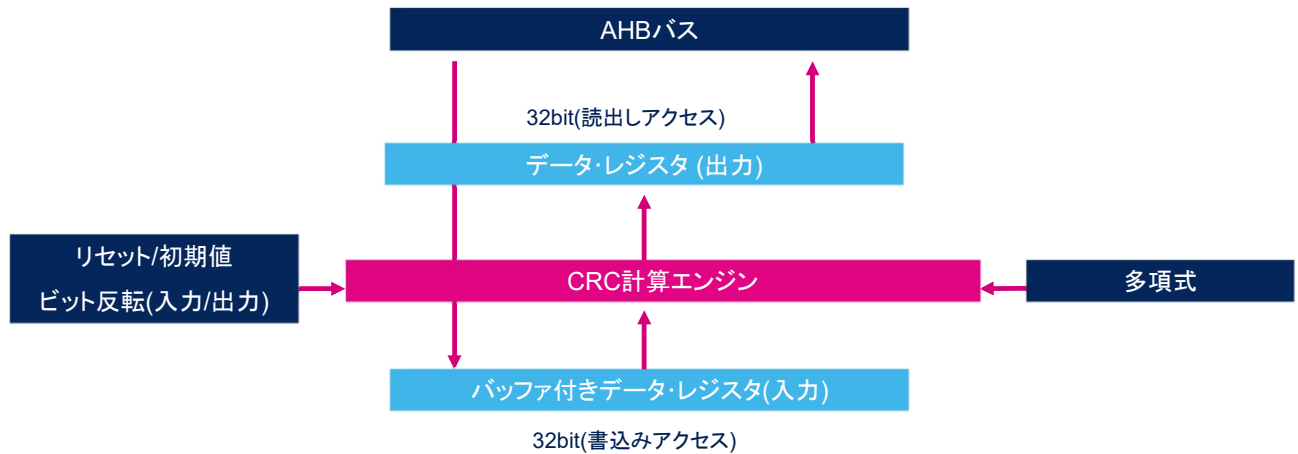
- 完全にプログラム可能な奇数多項式:
 - デフォルトでは、CRC-32 (Ethernet) 多項式 (0x04C11DB7) を使用
 - これは、 $x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$ という多項式を表す
- プログラム可能な多項式サイズ (7、8、16、32bit)
- プログラム可能な初期値 (デフォルトは 0xFFFFFFFF)
- 入出力データ用各種エンディアン・スキーム



STM32 に搭載されている CRC 計算ユニットは、サイズが 7、8、16、32bit にプログラム可能で、完全にプログラム可能な奇数多項式を特徴としています。

初期値も完全にプログラム可能であり、非常に柔軟な実行時の CRC コード生成が可能となっています。

ペリフェラルはビッグエンディアンまたはリトルエンディアン形式の入出力データに対応するように設定可能であり、各種の通信プロトコルに対応しています。



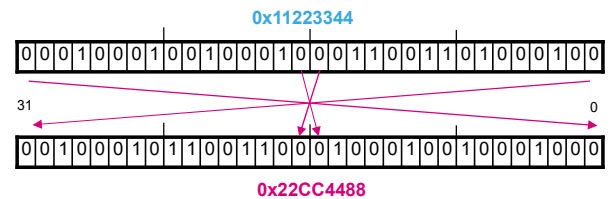
CRC 計算ユニットは、最適な性能を出すためにAHB バスインタフェースに接続されています。

32bit レジスタが 1 個含まれており、CPU と DMA コントローラのいずれかによる書込みと読出しの両方のために使用されます。

初期値、入出力ビット反転、および、多項式係数は、新たにCRC 計算を始める前の初期化フェーズ中に、ソフトウェアによって設定されます。CRC 計算エンジンは過去の CRC 計算結果をすべて保持しており、これが次回の計算の初期値となりますので、データブロックの新たな CRC 計算が可能となります。

ソフトウェア・オーバーヘッドを削減

- 入力データは、8、16、32bit単位で反転可能
 - CRC_CR レジスタの REV_IN[1:0] 値に依存
 - たとえば、入力データが0x1A2B3C4Dの場合、CRC計算では以下のように使用される：
 - 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単位で反転可能
 - CRC_CRレジスタの REV_OUT ビットを設定
 - 出力データ 0x11223344 の例を右図に示す



入出力フォーマットは、ソフトウェアオーバーヘッドの削減のために完全に設定可能です。そのため、アプリケーションは各種エンディアンスキームの管理が可能です。

CRC_CR レジスタの REV_IN[1:0] ビットによって、8、16、32bit 単位で入力データの反転と実行が可能です。

たとえば、入力データが0x1A2B3C4Dの場合、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反転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。

- 単一の入出力 32bitデータレジスタによってファームウェア操作が簡単
- AHB バスがストールしない連続書込み用入力バッファ
- CRC 計算時間は入力データ幅に依存:
 - 32bitブロックの場合、4AHBクロック・サイクル
 - 16bitブロックの場合、2AHBクロック・サイクル
 - 8bitブロックの場合、1AHBクロック・サイクル
- 書込みはCPU がSLEEPモードの間にDMAにより実行可能
 - 動作は低消費電力(連続計算)で、CPUは時間要求の厳しいタスクのために開放



life.augmented

CRC データレジスタには入力バッファが含まれており、実行中のCRC 計算によるウェイトステートを待つことなく、CPU またはDMA ペリフェラルによる 2 番目のデータの書込みが直ちに可能です。

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:

- 32ビットブロックの場合、4AHB クロックサイクル
- 16ビットブロックの場合、2AHB クロックサイクル
- 8ビットブロックの場合、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、CPU が別のタスクのためにオフロードされ、コードまたはデータの整合性の連続計算に用いることが可能となります。

モード	説明
RUN	有効
低電力 RUN	有効
SLEEP	有効 DMA コントローラは CRC 計算に使用可能
低電力 SLEEP	有効 DMA コントローラは CRC 計算に使用可能
STOP	STOP 0、STOP 1、STOP 2 モードで停止 ペリフェラル・レジスタの内容は保たれる
STANDBY	パワーダウン状態 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る
SHUTDOWN	パワーダウン状態 CRC ユニットの再初期化は、SHUTDOWN モード終了後に再初期化する必要がある



このスライドには、CRC 計算ユニットを使用可能な電源モードの概要が示されています。

CRC 計算ユニットは、動作にメイン AHB クロックが必要です。