

STM32G4 - NVIC

ネスト化されたベクタ割り込みコントローラ

1.0版



STM32ネスト化されたベクタ割り込みコントローラ(NVIC)のプレゼンテーションへようこそ。このコントローラの機能についてご紹介します。

- NVICはCortex[®]-M4 CPUに内蔵:
 - 102本のマスカブル割込みチャンネル
 - 16段階のプログラム可能な優先順位レベル
 - 遅延時間の少ない例外および割込み処理
 - 電源管理制御

アプリケーション側の利点

- 動的制御可能な優先順レベルに対応
- 割込みリクエストへの高速応答
- 再配置可能なベクタテーブル



割込みコントローラはCortex-M4 CPUに属しており、プロセッサコアとの密接なカップリングを可能とします。

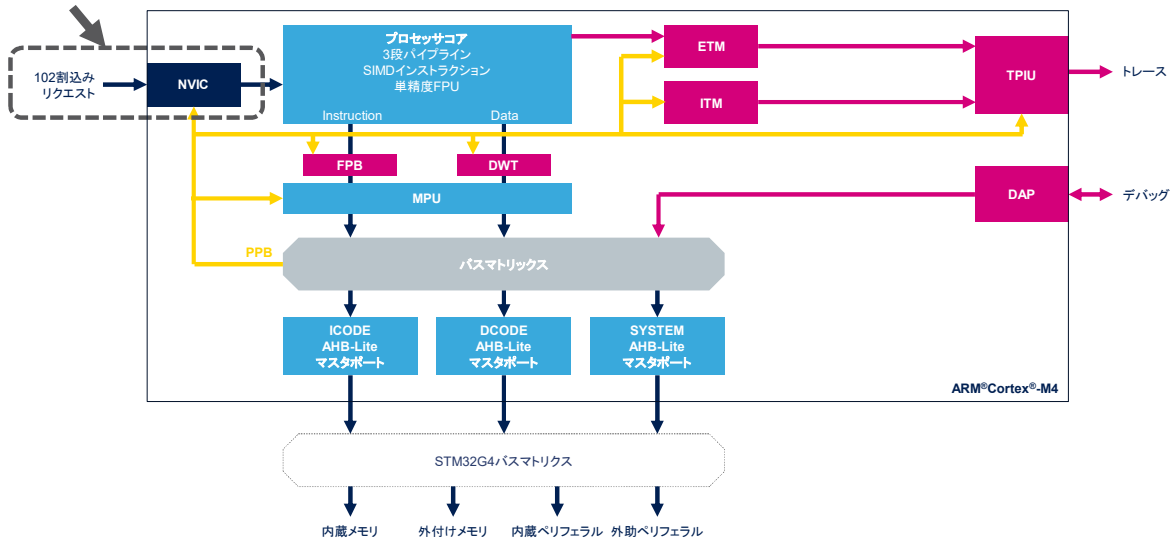
主な機能

- 102種類の割込みソース
- 16段階のプログラム可能な優先順位レベル
- 遅延時間の少ない例外および割込み処理
- 自動ネスティング
- 電源管理制御

アプリケーションは、割込みレベルの動的優先順位付け、低遅延応答によるリクエストへの高速応答、テールチェーンに加えて、ベクタテーブルの再配置による便益を受けます。

コア・アーキテクチャ概要

3



コアへの例外割込み含むすべての割込みは、NVICによって管理されます。

NVICとプロセッサコアインターフェースは密接にカップリングされており、これにより、低い割込みの遅延が保証され、遅延割込みの効率的な処理が可能になります。

NVICのコントロールレジスタとステータスレジスタへのアクセスは、内部のCortex®-M4 CPUのプライベートペリフェラルバス (PPB) を介して実行されます。

- 割込みリクエストへの高速応答
 - ロード/ストアが複数あるような長い待ち時間の命令は中断され、割込みリターン時に再開
 - 割込み操作のリターン時に、分割命令が停止され、再スタートされる
- 割込みの動的な再優先度付け
 - NVICレジスタにより、ユーザは割込み要求の優先順位を定義可能
 - このレジスタは動的に変更可能
- 割込みベクタ・テーブルの動的再配置
 - VTORレジスタを通じて、ベクタテーブルは再配置可能



NVICは割込み要求に対して高速応答が可能で、アプリケーションのイベントを迅速に処理できます。

割込みは、長い命令シーケンスの完了を待たずに処理されます。これらの手順は、割込みが戻ったときに再スタートされるか、再開されます。

各割込み要求に割り当てられた優先順位はプログラム可能であり、動的に変更できます。

例外ハンドラのアドレスを含むベクタテーブルも再配置することができ、システム設計者は割込みサービスルーチンの配置をアプリケーションのメモリレイアウトに合わせることができます。例えばベクタテーブルはRAMに再配置することができます。

優先順位処理

5

- Cortex®-M CPU例外管理に関しては、値が小さいほど優先順位が高い

例外ソース	優先順位レベル	
リセット	-3	ハードコードされた 固定優先順位
ノンマスクابل割込み(NMI)	-2	
ハードフォルト	-1	
その他の例外: - ペリフェラル割込み - ソフトウェア例外	0~15までのプログラム可能な値	



ソフトウェアは、それぞれの割込みはもとより、リセットを除くすべての例外ソース、ノンマスクابل割込み、ハードフォルトに対する優先順位レベルの割当てを管理します。

スーパーバイザコール命令の実行と同時にペリフェラル割込みがリクエストされた場合には必ず、これらのハードウェアとソフトウェアの例外の相対的優先順位によって、どれが最初に処理されるかが決定されます。

STM32G4では、ノンマスクابل割込み(NMI)はSRAMパリティエラー、フラッシュダブルECCエラーまたはクロック障害のいずれかの原因により発生します。

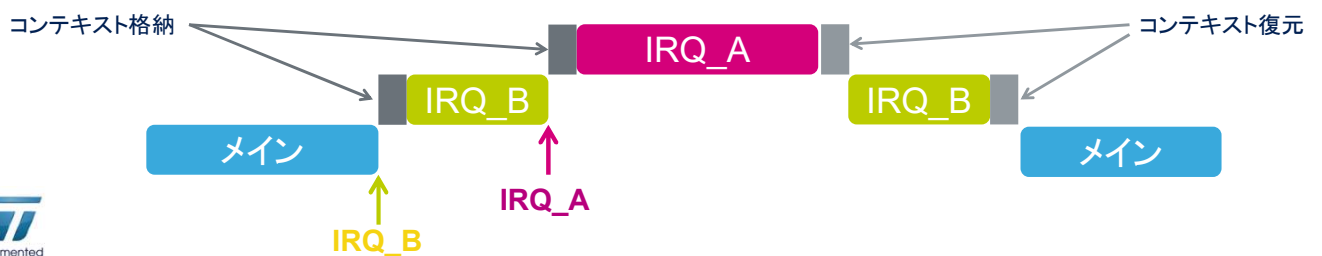
102種類のペリフェラル割込みリクエストのいずれかの優先順位は、Cortex®-M4 NVICレジスタにある専用の優先順位フィールドにプログラム可能です。

テールチェーンとネスティング

- テールチェーンとネスティングのメカニズムを説明するため、以下のペリフェラル割り込みソースについて検討

割り込みソース	優先順位レベル
IRQ_A	0
IRQ_B	1

- プリエンプションと割り込みネスティング



ネスト化されたベクタ割り込みコントローラによって、例外の効率的処理のためのいくつかの機能が提供されます。割り込みの処理中にそれよりも優先順位の高い新規のリクエストが到着した場合、新しい例外が現在の例外をプリエンプトできます。この処理は、ネスト化された例外処理と呼ばれます。優先順位が高い方の例外が処理された後に、その前の例外ハンドラが実行を再開します。Cortex®-M4の中に存在するマイクロコードによって、自動的にコンテキストが現在のスタックにプッシュされ、割り込みからの復帰時に復元されます。

例外の開始と復帰(1/2)

7

- テールチェーン

- 例外ハンドラの完了時に割込みがペンディングされている場合、コンテキスト格納がスキップされて、前のハンドラが完了すると、制御は新しい例外ハンドラに直ちに転送される



ある割込みハンドラの実行中に優先順位がそれ以下か同一の割込みリクエストが立つと、そのリクエストはペンディング扱いとなります。現在の割込みハンドラが終了した時点で、割込みの遅延低減のために、コンテキストの保存と復元のプロセスはスキップされて、制御は新しい例外ハンドラに直接転送されます。

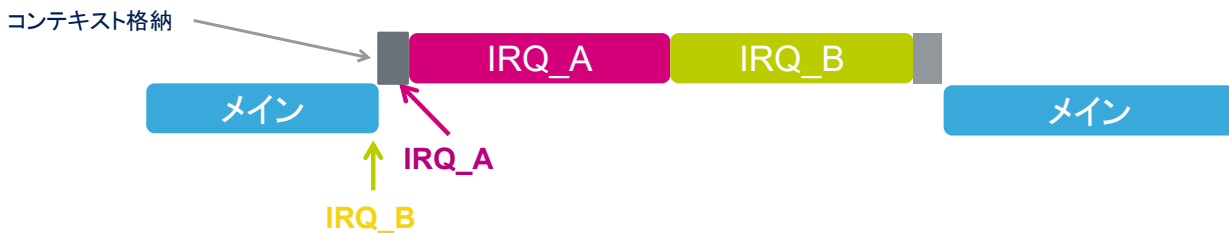
したがって、連続した優先順位が低い(優先順位値が大きい)割込みは、数クロックサイクルという非常に短い遅延で連鎖します。

例外の開始と復帰(2/2)

8

• 後着

- 例外の状態を保存している途中で、それより優先順位の高い例外が発生した場合、プロセッサは優先順位の高い例外の処理に直ちに切り替える



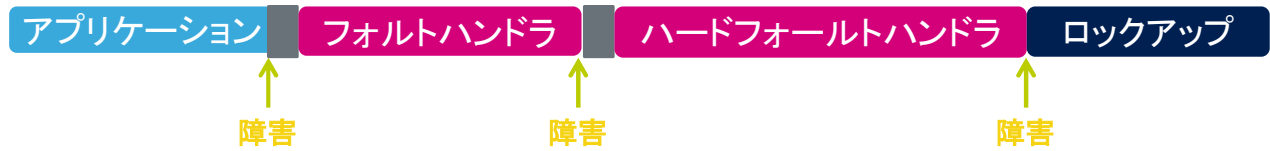
• 復帰

- 例外ハンドラが完了し、それ以外の例外がペンディングになっていない場合、プロセッサはスタックをポップして、割り込み発生前のプログラム状態を復元



割り込みが到着すると、プロセッサは、割り込みハンドラを実行する前にプログラムをまず保存します。優先順位がより高い割り込みが到着したときに、プロセッサがコンテキスト保存処理を実行中であった場合、プロセッサは、プログラムコンテキストの保存が完了すると、優先順位の高い方の割り込みの処理に直接切り替えます。その後、IRQ_B割り込みサービスルーチンを実行する前に、テールチェーンが用いられます。すべての例外ハンドラの実行が終わって、それ以外の例外がペンディングになっていない場合、プロセッサは、前のコンテキストをスタックから復元して、通常のアプリケーション実行に復帰します。

- Cortex®-M4は障害エスカレーション・メカニズムを実装



- 障害が発生すると、プロセッサはハードフォルトにエスカレートするが、
 - 障害が有効になっていない
 - ハンドラの実行に必要な優先順位が低い
 - 障害ハンドラが同じ障害を検出
- ハードフォルトまたはNMIハンドラの実行中に障害が発生すると、プロセッサはロックアップ状態になる



バス障害ハンドラ、メモリマネージメント障害ハンドラ、または用法エラーハンドラの実行中に障害が発生すると、プロセッサはこのイベントをハードフォルトハンドラにエスカレートします。バス障害、メモリ管理障害、用法エラーは明示的に有効にする必要があります。デフォルトではすべての障害がハードフォルトハンドラにつながります。

ハードフォルトハンドラの実行中に障害が発生すると、プロセッサはロックアップという状態になります。

ロックアップ出力をアサートし、回復不能な重大エラーが発生したことを示しています。

STM32G4マイクロコントローラでは、このLOCKUP出力はタイマブレーク入力に内部的に接続されており、そのためモータなどのパワーエレクトロニクスの制御において安全な状態で停止することができます。

- ソフトウェアが正しく整列されたレジスタ・アクセスを使用するようにする
- 割込みは、無効であってもペンディング状態となることができる
 - 割込みを無効にすると、プロセッサが割込みを取得しなくなるだけ
- ベクタテーブルを再配置する前に、すべての有効な割込みに対して新しいエントリが正しく設定されていることを確認する
 - この対象には、フォルト・ハンドラとNMIが含まれる
 - ベクタテーブルを再配置するため、VTORレジスタをプログラムする前にこれを行う



ネスト化されたベクタ割込みコントローラレジスタを評価する際には、自分のコードが正しく整列されたレジスタアクセスを使用していることを確認します。整列されていないアクセスには、ネスト化されたベクタ割込みコントローラレジスタも、Cortex®-M4の中にあるすべてのメモリマップされたレジスタも対応していません。

転送元がサービスを要求した場合、割込みがペンディングとなります。割込みを無効にすると、プロセッサが割込みを取得しなくなるだけです。割込みベクタを有効とする前に、関係する割込みフラグがクリアされていることを確認します。

VTORレジスタを用いてベクタテーブルを再配置する前に、フォルトハンドラ、ノンマスクابل割込み、すべての有効な割込みが新しい位置で正しく設定されていることを確認します。

- タイマにリンクされた以下のペリフェラルに関するトレーニング教材を参照してください:
 - タイマ
 - 一部のタイマのブレーク入力は、Cortex®-M4 LOCKUP出力に接続
 - Cortex®-M4
 - CPUは、ソフトウェア例外とハードウェア例外両方の処理に用いられる例外メカニズムを実装



ネスト化されたベクタ割り込みコントローラは、タイマモジュールとCortex-M4 CPUにリンクされています。関連するプレゼンテーションを参照してください。

- 詳細については、以下の関連資料を参照してください。:
 - PM0214 Cortex®-M4プログラミング・マニュアル
 - STM32G4リファレンス・マニュアル



詳細については、Cortex®-M4コアのプログラミングマニュアルおよびSTM32G4のリファレンスマニュアルも参照してください。