

STM32WL5 - NVIC

ネスト化された割込みコントローラ

レビジョン 1.0

STM32 のネスト化された割込みコントローラ (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説明します。

- Cortex®-M4 CPU1 に内蔵された NVIC
 - 遅延時間の少ない例外と割込みを処理
 - 62 のマスカブル割込みチャンネル
 - 16 のプログラム可能な優先順位レベル
 - 割込みのネスティング
 - 電源管理制御
- Cortex®-M0+ CPU2 に内蔵された NVIC
 - 遅延時間の少ない例外と割込みを処理
 - 32 のマスカブル割込みチャンネル
 - 4 つのプログラム可能な優先順位レベル
 - 割込みのネスティング
 - 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位レベルをサポート
- 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 CPU に内蔵され、プロセッサのコアと緊密に連携動作します。

Cortex®-M4 CPU1 の主な機能:

- **62 の割込みソース**
- **16 のプログラム可能な優先順位レベル**

Cortex®-M0+ CPU2 の主な機能:

- **32 の割込みソース**
- **4 つのプログラム可能な優先順位レベル**

両方の CPU の NVIC で実現できる機能:

- 遅延時間の少ない例外と割込みを処理
- 自動ネスティング
- 電源管理制御

アプリケーションで得られる利点として、割込みレベルの動的な優先順位付け、低遅延応答とテールチェーンによるリクエストへの高速応答、ベクタテーブルの再配置があります。

- 割り込みリクエストへの高速応答
 - NVIC の割り込みリクエスト・エントリ数は、Cortex®-M4 CPU1 では 62、Cortex®-M0+ CPU2 では 32 ただし、STM32WL5 に実装されている**割り込みの数は 62 より多い**
 - 割り込みイベントの数が NVIC エントリの数より多いので、NVIC ベクタによっては複数の割り込みに接続されるものがある
 - ユーザソフトウェアでは、ペリフェラル割り込みレジスタを読み出すことによって、割り込みをリクエストしたペリフェラルを特定できる
 - 2 つの CPU 間で柔軟に割り込みを配分できるように、複数の割り込みに接続されている NVIC ベクタは SYSCFG でプリマスキングされる(詳細については SYSCFG モジュールを参照)
- 割り込みの動的な再優先順位付け
- 割り込みベクタテーブルの動的な再配置



ネスト化された割り込みコントローラは割り込みリクエストに対して高速に応答するので、アプリケーションは速やかに受信イベントを処理できます。

STM32WL5 には、NVIC エントリ数より多くの割り込みが実装されています。したがって、割り込みによっては、ネスト化された割り込みコントローラの同一ベクタに結び付けられているものがあります。ソフトウェアでは、ペリフェラル割り込みレジスタを読み出すことによって、割り込みをリクエストしたペリフェラルを特定できます。ネスト化された割り込みコントローラの同一ベクタに結び付けられている複数の割り込みは、SYSCFG でプリマスキングすることで、Cortex-M4 CPU と Cortex-M0+ CPU との間で柔軟に配分できます。

各割り込みリクエストに割り当てられている優先順位はプログラム可能であり、動的に変更できます。

割り込みベクタテーブルは再配置することもできるので、アプリケーションのメモリレイアウトに合わせて、割り込みサービスルーチンの配置を調整できます。たとえば、ベクタテーブルを RAM に再配置できます。

優先順位の処理

- Cortex®-M CPUの例外管理では、優先順位の値が小さいほど優先順位が高くなる

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	
ハードフォールト	-1	
他の例外: <ul style="list-style-type: none">• ソフトウェア割込み• ペリフェラル割込み	Cortex-M4 では 0 ~ 15 のレベルをプログラム可能 Cortex-M0+ では 0 ~ 3 のレベルをプログラム可能	



ソフトウェアでは、各割込みに優先順位レベルを割り当てるほか、リセット、ノンマスクابل割込み、およびハードフォールトを除くすべての例外ソースに優先順位レベルを割り当てます。

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は、これらのハードウェアとソフトウェアの例外の相対的な優先順位によって、どれが最初に処理されるかが決まります。

STM32WL5 マイクロコントローラでは、SRAM2 パリティエラー、Flash ダブル ECC エラー、またはクロック障害に起因してノンマスクابل割込み(NMI)が発生します。

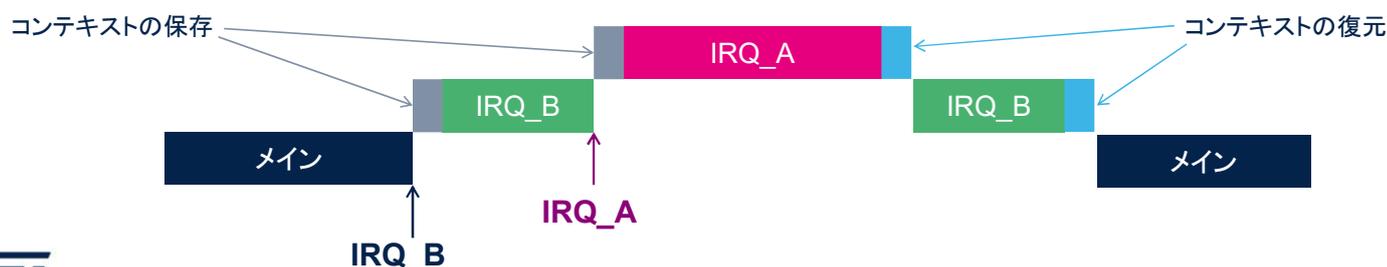
ネスト化された割込みコントローラレジスタにある専用の優先順位フィールドで、ソフトウェアとペリフェラルの割込みリクエストの優先順位をプログラムできます。

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に、次のペリフェラル割り込みソースを想定

割り込みソース	優先順位レベル
IRQ_A	0
IRQ_B	1

- 優先順位の横取りと割り込みネスティング



ネスト化された割り込みコントローラは、例外を効率的に処理するためのさまざまな機能を提供します。

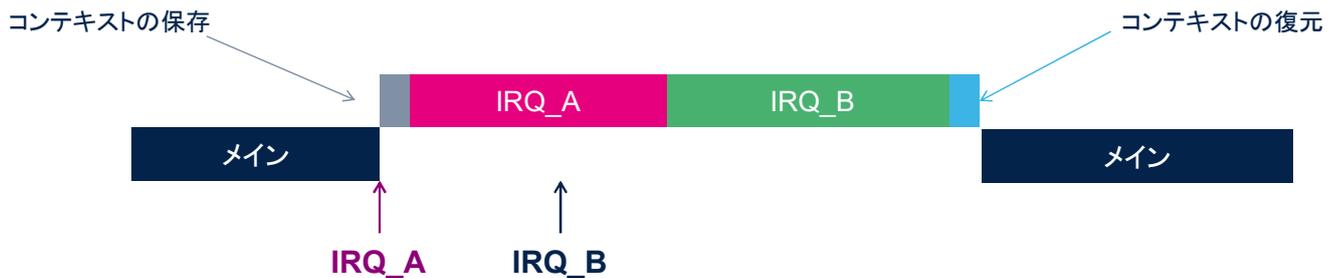
割り込みの処理中にそれより優先順位の高い新しいリクエストを受け取ると、その新しい例外が処理中の例外に代わって優先権を獲得します。これは、ネスト化された例外処理と呼ばれます。優先順位の高い例外が処理された後で、その前に処理中だった例外ハンドラの実行が再開されます。

Cortex[®]-M4 に置かれたマイクロコードによってコンテキストが自動的に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

• テールチェーン

- ある例外ハンドラが完了した時点で保留状態の割込みが存在する場合は、コンテキストの保存がスキップされ、そのハンドラが完了した直後に制御は新しい例外ハンドラに移る



割込みハンドラの実行中にそれ以下の優先順位の割込みリクエストが発生した場合、そのリクエストは保留状態になります。現在実行中の割込みハンドラが完了した後、割込みの遅延を短くするために、コンテキストの保存と復元のプロセスはスキップされ、直ちに新しい例外ハンドラに制御が直接移ります。したがって、実行中の割込みより優先順位が低い(優先順位値が大きい)割込みが連続して発生した場合、それらは数クロックサイクルのきわめて短い遅延で連続的に処理されます。

例外の開始と復帰

• 後着

- 処理した例外の状態を保存しているときに、その例外よりも優先順位の高い例外が発生した場合、優先順位の高い例外の処理に実行が直ちに切り替えられる



• 復帰

- 例外ハンドラが完了したとき、保留状態の例外がなければスタックの内容がポップされ、割り込みが発生する前のプログラム状態が復元される



割り込みを受け取ったプロセッサは、現在のプログラムのコンテキストを保存したうえで、割り込みハンドラを実行します。このコンテキスト保存操作を実行しているときに、現在の割り込みより優先順位の高い割り込みを受け取った場合、プロセッサはプログラムのコンテキスト保存を完了した後、その優先順位の高い割り込みの処理に実行を直接切り替えます。IRQ_B の割り込みサービスルーチンを実行する前に、テールチェーンが使用されます。

すべての例外ハンドラを実行し、保留中の割り込みがなくなると、プロセッサは、スタックから前のコンテキストを復元し、通常のアプリケーション実行に復帰します。

- 必ず正しい配列のレジスタアクセスをソフトウェアで使用する
- 無効な割込みであっても保留状態になることがある
 - 割込みを無効にしても、プロセッサがその割込みを受け取らないだけである
- 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
 - このような割込みとしてフォールト・ハンドラと NMI がある
 - ベクタテーブルを再配置するために VTOR レジスタをプログラミングする前に、これを確認する

ネスト化された割込みコントローラレジスタにアクセスする場合、正しい配列のレジスタアクセスをコードで使用していることを確認します。ネスト化された割込みコントローラレジスタだけでなく、Cortex®-M4 でメモリにマップされたどのレジスタでも、正しく配列されていないアクセスはサポートされていません。

割込みの発生元がサービスを要求した時点で、その割込みは保留状態になります。割込みを無効にしても、プロセッサがその割込みを受け取らないだけです。割込みベクタを有効にする前に、関連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 レジスタを使用してベクタテーブルを再配置する前に、フォールトハンドラ、ノンマスクابل割込み、すべての有効な割込みを新しい配置先で正しく設定していることを確認します。

デュアルコアによる割込み共有

- NVIC 割込みベクタによっては複数のペリフェラル割込みに接続されるものがある
- これらの割込みには、個別に制御できるように、SYSCFG ブロックでプリマスキングが適用される
- 関連する Cortex-M4 ペリフェラル：
 - EXTI、PVM3、PVD、RTC からの GPIO EXTI マルチプレクス信号
- 関連する Cortex-M0+ ペリフェラル：
 - EXTI、PVM3、PVD、RTC、RCC、FLASH インタフェース、PKA、AES、COMP、ADC、DAC、DMA、DMAMUX からの GPIO EXTI マルチプレクス信号



life.augmented

Cortex®-M4 コアと Cortex®-M0+ コアを搭載するデュアルコア STM32WL5 マイクロコントローラでは、ペリフェラル割込みが両方のコアに接続されます。不要な割込みが発生しないように、ネスト化された割込みコントローラの単一のエントリにマップされている複数の割込みを、システム設定コントローラ(SYSCFG)でプリマスキングできます。SYSCFG の割込みマスケジスタにより、目的の CPU (Cortex®-M4 または Cortex®-M0+) にのみ確実に割込み発生元が転送されます。スライドでは、CPU ごとに関連するペリフェラルがリストされています。

- NVIC にリンクしている次のペリフェラルのトレーニングを参照
 - SYSCFG
 - 共有されている割込みリクエスト信号に NVIC で接続されている複数の割込み発生元をプリマスキング
 - Cortex[®]-M4
 - CPU はソフトウェア例外とハードウェア例外の両方の処理に使用する例外メカニズムを実装
 - Cortex[®]-M0+
 - CPU はソフトウェア例外とハードウェア例外の両方の処理に使用する例外メカニズムを実装

ネスト化された割込みコントローラは、SYSCFG モジュール、Cortex-M4 CPU1、Cortex-M0+ CPU2 にリンクしています。詳細については、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、次の文書を参照：
 - STM32F3、F4、L4、および L4+ シリーズ向けプログラミングマニュアル (PM0214)
 - STM32WL5 リファレンスマニュアル

詳細については、STM32F3、F4、L4、および L4+ シリーズのプログラミングマニュアルおよび **STM32WL5** のリファレンスマニュアルも参照してください。