

# STM32WL5 - EXTI

拡張割込み/イベントコントローラ

レビジョン 1.0

**STM32WL5 拡張割込み/イベントコントローラ(EXTI)のプレゼンテーションへようこそ。**

- 47 件のイベント/割込みライン
  - 23 件の設定可能なイベント
  - 24 件のダイレクトイベント
- 独立マスクと設定
- STM32WL5 の両方の CPU に対する独立したウェイクアップ

### アプリケーション側の利点

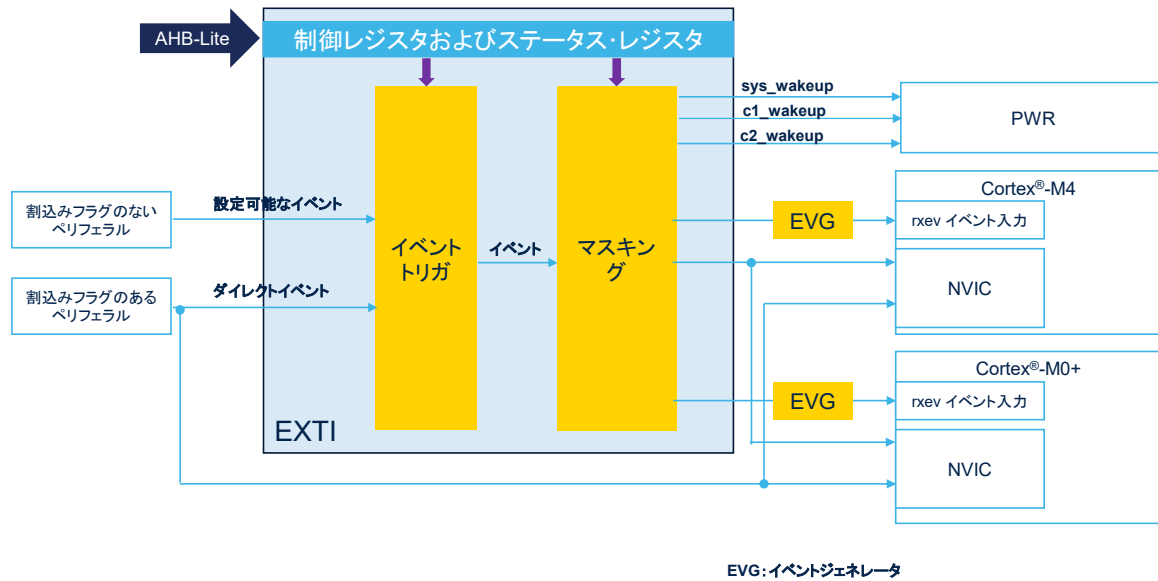
- 外部と内部のウェイクアップイベントと割込みを管理
- 設定可能なイベントのペンディングフラグを提供



拡張割込み/イベントコントローラ(EXTI)は、47 件の独立したイベントを提供します。これらは設定可能なイベントとダイレクトイベントの 2 つのカテゴリに分類されます。

アプリケーションは、外部通信またはリクエストによって各 CPU を別々にウェイクアップする STM32WL5 の機能を利用して、低消費電力モードをよりスマートに使用できます。

## EXTI のブロック図



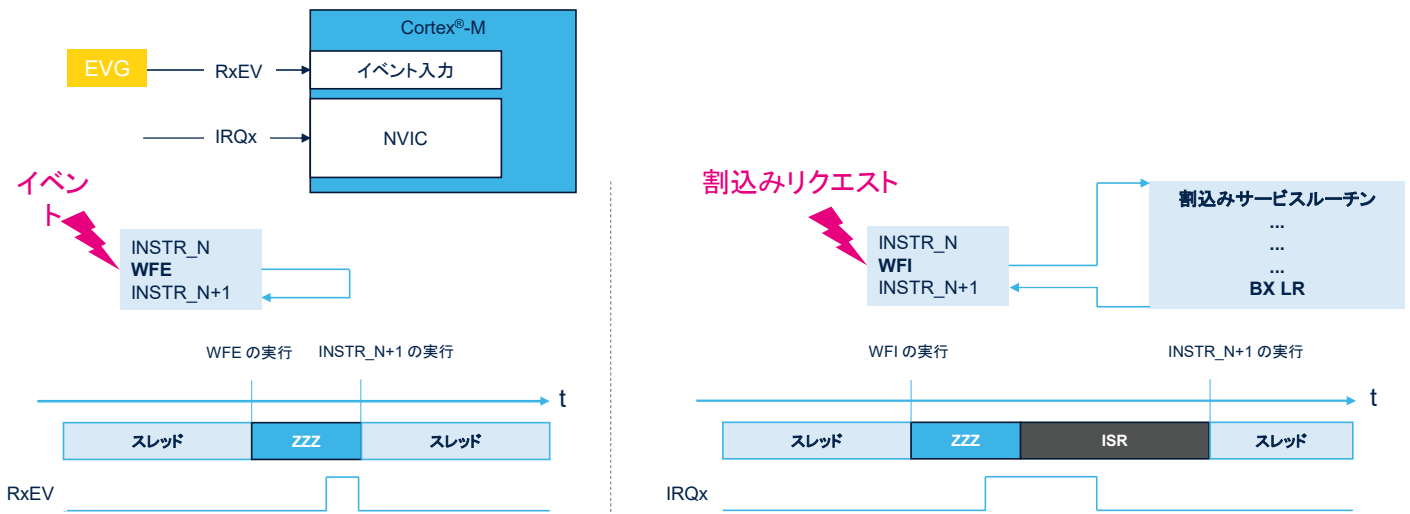
拡張割り込み/イベントコントローラのブロック図を示します。割り込み機能はないけれどもパルスの生成が可能なペリフェラルでは、設定可能なイベントが生成されます。EXTI コントローラは、割り込みの検出、マスキング、およびソフトウェアトリガを提供します。割り込みリクエストに対応したペリフェラルでは、ダイレクトイベントが生成されます。この場合、EXTI コントローラを使用して、CPU に対するイベントが生成され、システムウェイクアップがリクエストされます。拡張割り込みとイベントコントローラは、設定可能なイベント 1 つごとに、両方の CPU に対して割り込みを 1 つずつ発行します。CPU のウェイクアップは個別に発行されるので、両方のプロセッサを別々にウェイクアップできます。

- STOP モードからのウェイクアップ、割込み、およびイベントの生成
  - 独立した割込みとイベントマスク
- 設定可能なイベント
  - アクティブエッジの選択
  - 単一のペンディングフラグ
  - ソフトウェアからトリガ可能
  - リンク先:
    - GPIO、PVD、PVM、COMP、CPU SEV、および Sub-GHz 無線ビジー
- ダイレクトイベント
  - 関連ペリフェラルから提供されるステータスフラグ
  - リンク先:
    - RTC、TAMP、I2C、USART、LPUART、LPTIM、Sub-GHz 無線、IPCC HSEM、Flash、および DEBUG



拡張割込みとイベントコントローラは、割込みとイベントを生成できるほか、STOP モードからプロセッサをウェイクアップできます。設定可能なイベントは、GPIO、PVD、PVM、コンパレータ COMP、CPU 送信イベント、および Sub-GHz 無線ビジーからの外部割込みにリンクしています。ダイレクトイベントは、RTC、TAMP、I2C、USARTS、LPUART、LPTIM、**Sub-GHz 無線**、IPCC、HSEM、Flash、および DEBUG にリンクしています。

# Cortex®-m のイベントと割り込みの比較



- イベント生成可能なペリフェラル:
  - GPIO EXTI、RTC、TAMP、COMP、および CPU SEV



Cortex®-M は、低消費電力状態に移行する次の 2 種類の方法をサポートしています。

1. WFE (Wait For Event) 命令の実行
2. WFI (Wait For Interrupt) 命令の実行

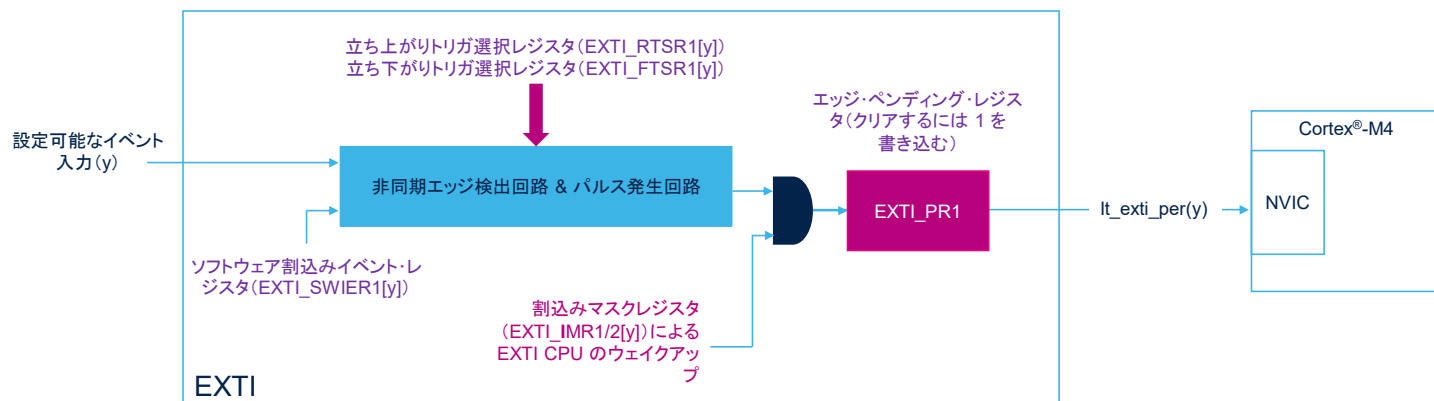
左側のシーケンスに示すように、WFE の場合、ウェイクアップイベント後に実行される最初の命令は、次に続く命令 INSTR\_N+1 です。

WFI を実装することにより、有効な割り込みリクエストを受け取ったプロセッサは、割り込みサービスルーチンにジャンプします。

割り込みリクエストは WFE の終了条件ですが、RXEV で受け取ったイベントは WFI の終了条件ではありません。

**STM32WL5** シリーズでは、このスライドに挙げているペリフェラルでのみイベントを生成できます。

- 設定可能イベントを割込みリクエストとして使用



この図は、設定可能なイベントのアクティブエッジを割込みリクエストに変換できるようにするさまざまなステージについて説明しています。

最初のステージは、2つのレジスタである EXTI\_RTSTR1 と EXTI\_FTSTR1 で設定する非同期エッジ検出回路です。どちらのエッジも、場合によっては両方とも、選択できます。

EXTI\_SWIER レジスタの該当ビットをセットすることにより、設定可能なイベントをソフトウェアでエミュレートできます。このビットは、ハードウェアによって自動的にクリアされます。

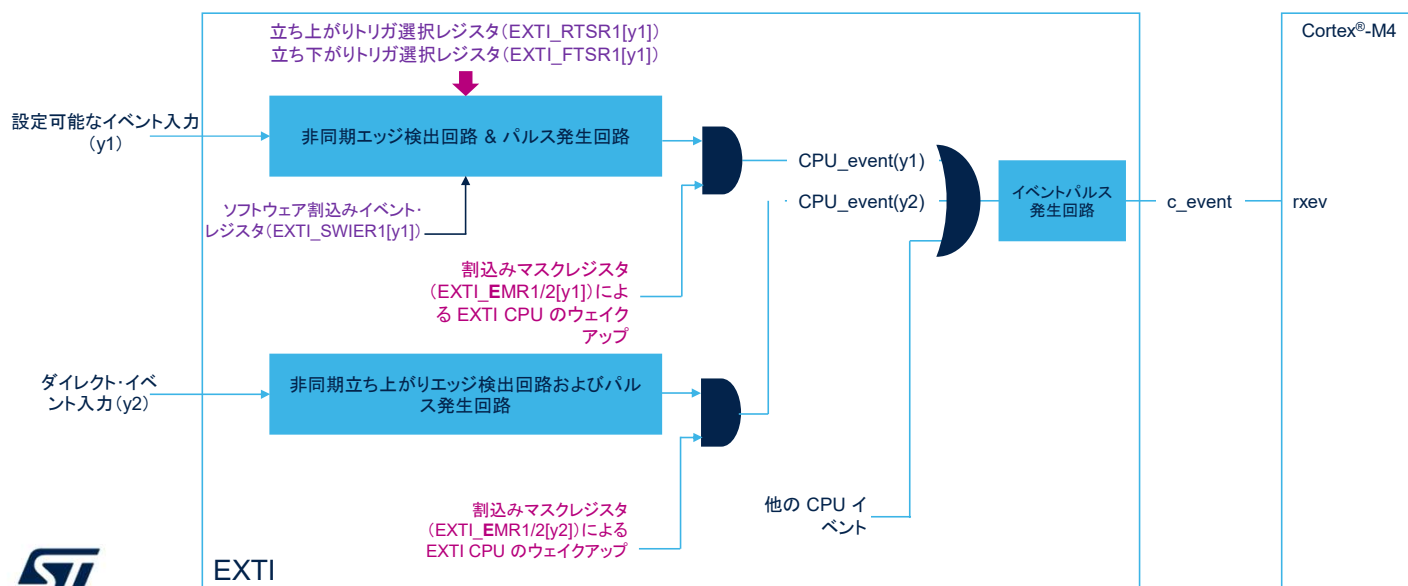
AND ゲートを使用して、CPU NVIC に対する割込みの生成をマスクまたは有効にすることができます。

最後に、NVIC に対して割込みが生成されると、EXTI\_PR1 レジスタにフラグがセットされます。このフラグによって、割込みが発生した原因をソフトウェアで判定できます。

このフラグは割込みサービスルーチンによってクリアされることが想定されています。

# CPU イベントの生成

- 設定可能なイベントとダイレクトイベントを CPU イベントリクエストとして使用



この図は、設定可能なイベントのアクティブエッジを CPU イベントに変換できるようにするさまざまなステージについて説明しています。

設定可能なイベントとダイレクトイベントのどちらも、CPU の rxev 入力に送られるイベントを発行するように設定できます。

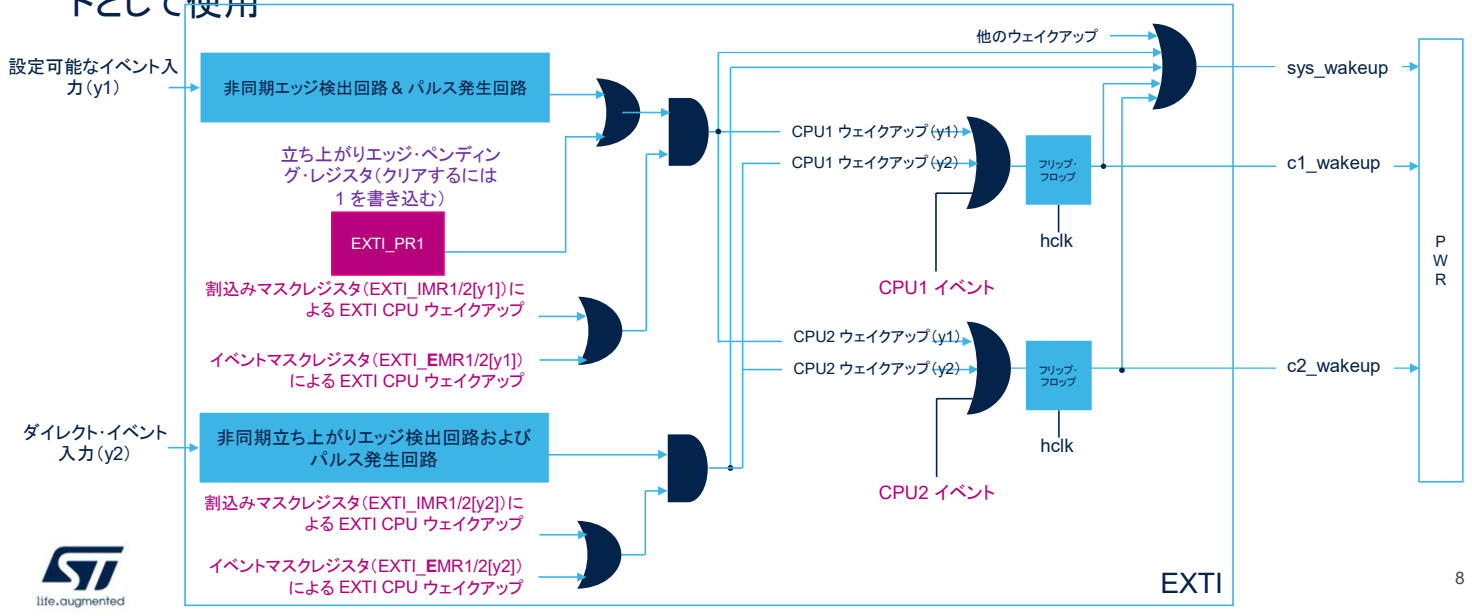
割込みリクエストとは異なり、CPU には固有のイベント入力があるので、すべてのイベントリクエストがまとめて論理和演算され、イベントパルス発生回路に入力されます。

イベント生成のマスクに使用されるレジスタは EXTI\_EMR であり、割込み生成のマスクに使用されるレジスタ EXTI\_IMR とは異なります。

デュアルコアの STM32WL5 マイクロコントローラには、その CPU ごとに独立したイベント生成ロジックが搭載されています。

# ウェイクアップイベントの生成

- 設定可能なイベントとダイレクトイベントをコアとシステムに対するウェイクアップリクエストとして使用



EXT� ブロックによって生成される CPU ウェイクアップ信号は、PWR ブロックに接続され、システムと CPU サブシステムバスクロックのウェイクアップに使用されます。

設定可能なイベントとダイレクトイベントはどちらも、ウェイクアップをリクエストできます。

非同期エッジ検出回路でアクティブエッジが検出されるとウェイクアップが発生します。

その結果、ウェイクアップのソースが設定可能なイベントであれば、ソフトウェアによって EXTI\_PR1 レジスタのフラグがクリアされてウェイクアップリクエストが無効になることが想定されます。ダイレクトイベントの場合、このフラグはペリフェラル・ユニットにあります。

これらのフラグによって、ソフトウェアでウェイクアップの原因を把握できます。

割込みまたはイベントのどちらかの生成が有効になると、ウェイクアップ指示がアサートされます。EXT�\_IMR レジスタと EXTI\_EMR レジスタの接続先である OR ゲートの動作を確認してください。

すべての CPU ウェイクアップ信号の論理和を計算し、その結果とイベントリクエストとの論理和を計算します。

sys\_wakeup は非同期であり、クロックをウェイクアップします。hclk が動作していれば、同期 c1\_wakeup と同期 c2\_wakeup が生成されて、それぞれの CPU をウェイクアップします。



## ダイレクトイベントによる論理 CPU ウェイクアップのトリガ

- ダイレクトイベントには関連する EXTI 割込みがない
  - EXTI はシステムクロックと CPU サブシステムのクロックのみをウェイクアップし、CPU ウェイクアップイベントを生成できる
  - ダイレクトウェイクアップイベントに関連するペリフェラル同期割込みによって CPU がウェイクアップする
- EXTI のダイレクトイベントは CPU イベントを生成できる



life.augmented

9

ダイレクトイベントは、EXTI コントローラを通じて CPU イベントを生成し、システムウェイクアップをトリガできます。

ダイレクトイベントのアクティブエッジは、立ち上がりエッジです。

ダイレクトイベントは、NVIC への専用ラインがあることから、EXTI コントローラによる割込みリクエストのアサートを必要としません。

それ以外の場合、これまでのスライドで説明したものと同一回路が実装されます。ダイレクトイベントは、イベント生成向けと割り込み生成向けに別々にマスクできます。EXTI コントローラの割込みマスクは、ウェイクアップマスクとしてのみ使用されます。

- タイマ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
  - NVIC
    - CPU 割込みリクエストを処理
  - 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

拡張割込みとイベントコントローラは、Cortex-M4 CPU と **Cortex-M0+ CPU**のネスト化されたベクタ割込みコントローラ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 MCU 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** シリーズのリファレンスマニュアルを参照してください。