

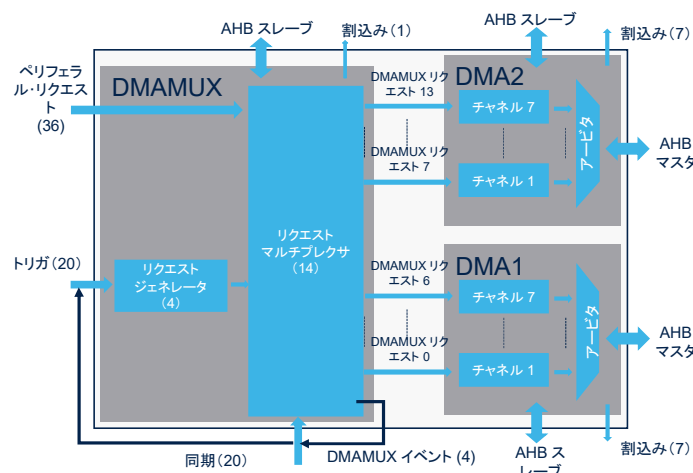
STM32WL5 – DMA および DMAMUX

ダイレクト・メモリ・アクセス (DMA) コントローラ
DMA リクエスト・マルチプレクサ (DMAMUX)

レビジョン 1.0

STM32WL5のダイレクト・メモリ・アクセス (DMA) コントローラのプレゼンテーションによろこそ。
ここでは、新たな DMA リクエスト・マルチプレクサ (DMAMUX) モジュールで強化した DMA コントローラ・モジュールの主な機能を紹介します。

- それぞれが以下の機能を提供する 2 つの DMA コントローラ
 - 7 本の同時チャンネル(個別に設定可能)によるプログラム可能なブロック転送
 - プログラム可能なチャンネルベースの優先順位
 - AHB マスタ・ポートを介したデータ転送
 - 1 つの DMA リクエスト・ルータ(DMAMUX)
 - プログラム可能なリクエスト・ソースの選択
 - DMA モードのペリフェラル
 - 外部および内部生成トリガ
 - 同期モード: リクエスト・カウンタによる同期入力
 - リクエストチェーン: 入カトリガや同期を元にした他のリクエストやチャンネル向けのイベント生成用リクエストカウンタ



アプリケーション側の利点

- メモリにマッピングされた転送元からメモリにマッピングされた転送先へのデータ転送で CPU の負荷を軽減

DMA によってアプリケーションで得られる主な利点は、メモリにマッピングしたあらゆる転送元からメモリにマッピングしたあらゆる転送先へのデータ転送で CPU の負荷を軽減できることです。

STM32WL5 DMA の特徴は次のとおりです。

- 2 つの DMA コントローラ。DMA コントローラのそれぞれで次の機能を実現します。
 - ❖ 7 つの同時チャンネルを使用した、プログラム可能なブロック転送(それぞれを別々に設定可能)
 - ❖ プログラム可能なチャンネルベースの優先順位
 - ❖ バス・マトリックスに接続された AHB マスタ・ポートを介したデータ転送
- 次の機能を提供する DMA リクエスト・ルータ(DMAMUX)
 - ❖ プログラム可能なリクエスト・ソースの選択: DMA モードのペリフェラルまたはトリガで選択して内部生成
 - ❖ 同期モード: DMAMUX リクエスト・カウンタによる同期入力によって実現
 - ❖ リクエスト・チェーン: 他のリクエスト/チャンネルに対する入カトリガ/同期となるイベントを生成する DMAMUX リクエスト・カウンタによって実現

STM32WB DMA および DMAMUX インスタンス

DMAMUX の機能	DMAMUX
ペリフェラル・リクエストの数	38
リクエスト・ジェネレータの数	4
トリガ回数	21
同期回数	21
出力 DMA リクエストの数	14

DMA の機能	DMA1
チャンネル数	7

DMA の機能	DMA2
チャンネル数	7



38 個のペリフェラル・リクエストと 4 つの DMAMUX リクエスト・ジェネレータがあります。
21 個のトリガと同期入力があります。
14 個の DMA チャンネルまたは 14 回の DMA リクエストがあります。

DMA コントローラ (1/3)

- 個別に設定可能な複数のチャンネルが 1 つの DMA コントローラに存在
 - メモリからペリフェラルへの転送またはペリフェラルからのメモリへの転送で、ペリフェラルから DMA ハードウェア・リクエストにチャンネルを割り当て可能
 - メモリ間転送ではソフトウェア・リクエストをチャンネルに割り当て
 - 優先度レベルでチャンネルをプログラム
 - ブロック・レベルで一定のデータ転送回数についてチャンネルをプログラム
- ブロック転送の完了、ハーフブロック転送の完了、転送エラーなどのプログラム可能なイベントの発生時に、独立した割込みまたはフラグを介してチャンネルをソフトウェアで制御
 - バス・アクセス・エラーの場合は障害が発生したチャンネルを自動的に無効化



DMA コントローラに注目します。

DMA コントローラの各チャンネルは互いに別々に設定できます。

- ペリフェラルからメモリへの転送またはメモリからペリフェラルへの転送で、ペリフェラルから DMA ハードウェア・リクエストにチャンネルを割り当てることができます。
- メモリ間転送ではソフトウェア・リクエストにチャンネルが割り当てられます。
- 優先度レベルでチャンネルがプログラムされます。
- ブロック・レベルで一定のデータ転送回数についてチャンネルがプログラムされます。

ブロック転送の完了、ハーフブロック転送の完了、転送エラーなどのプログラム可能なイベントの発生時に、独立した割込みまたはフラグを介してチャンネルをソフトウェアで制御できます。

バス・アクセス・エラーの場合は、障害が発生したチャンネルが自動的に無効化されます。

- **ブロック・レベルのプログラム可能なデータ転送**
 - 転送元と転送先で別々のデータ・サイズ: 8 ビット、16 ビット、または 32 ビット
 - 転送元と転送先で別々の開始アドレス
 - 転送元と転送先で別々のアドレス・インクリメント: 連続インクリメントまたは固定アドレス
 - プログラム可能な、1 ブロックあたりの転送データ量
 - 転送元最大データ 65,535
 - ハードウェアによる自動デクリメント
- **ブロック転送が完了したときのサーキュラ・バッファ・モード(ペリフェラルとの間でデータを連続的に転送)**
 - 以下のプログラムされた情報をハードウェアで自動的にリロード
 - 1 ブロックあたりの転送データ量
 - 転送元および転送先の開始アドレス



一定のデータ転送回数について、ブロック・レベルでチャンネルが以下の値でプログラムされます。

- 転送元と転送先で別々のデータ・サイズ
- 転送元と転送先で別々の開始アドレス
- 転送元と転送先で別々のアドレス・インクリメント: (連続インクリメントまたは固定アドレス)
- プログラム可能な、1 ブロックあたりの転送データ量
 - ❖ 転送元最大データ 65,535
 - ❖ ハードウェアによる自動デクリメント

ブロック転送が完了したときのサーキュラ・バッファ・モード(ペリフェラルとの間でデータを連続的に転送)

- 1 ブロックで転送するようにプログラムしたデータ量が、ハードウェアによって自動的にリロードされます。
- 転送元と転送先の開始アドレスも同様

- **メモリ間モード**
 - このモードでは(ハードウェア・リクエストなしで)、チャンネルが有効になると直ちにブロック転送を開始
- **ペリフェラルからメモリへの転送モードと、メモリからペリフェラルへの転送モード**
 - チャンネルが有効になったとき、およびペリフェラルから DMA ハードウェア・リクエストが送信されたときの両方で、直ちにブロック転送開始
 - DMA ハードウェア・リクエストでは、単一の DMA データ転送を識別可能
 - 各データが正常に転送先に転送される場合、各 DMA ハードウェア・リクエストは DMA で調整され生成
- **あらゆるモードでの動作**
 - すべてのデータ転送間でチャンネル・アービトレーションを再評価

メモリ間モードでは、チャンネルが有効になると直ちにブロック転送が始まります(ハードウェア・リクエストはなし)。

ペリフェラルからメモリへの転送モードと、メモリからペリフェラルへの転送モードでは次のようになります。

- チャンネルが有効になったとき、およびペリフェラルから DMA ハードウェア・リクエストが送信されたときの両方で、直ちにブロック転送開始
- DMA ハードウェア・リクエストでは、単一の DMA データ転送を特定
- 各データが正常に転送先に転送される場合、各 DMA ハードウェア・リクエストは DMA で調整され生成

どのモードでも、すべてのデータ転送間でチャンネル・アービトレーションが再評価されます。

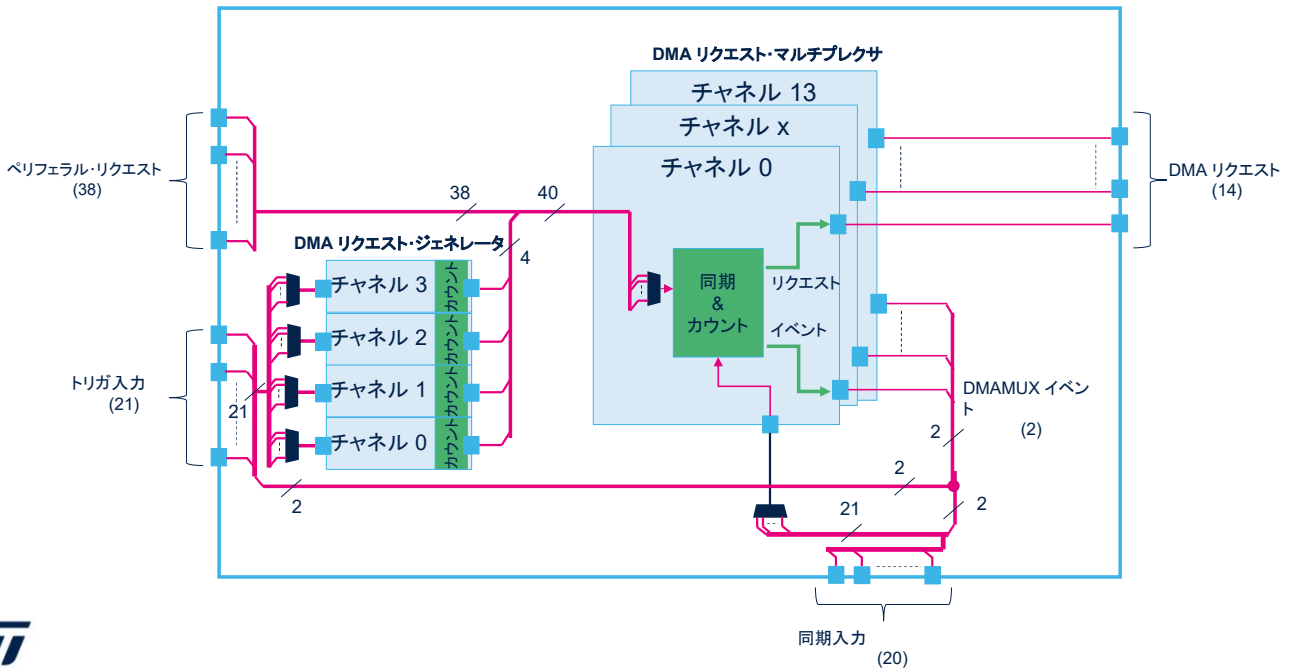
- 各チャネルの割込みイベント

割込みイベント	説明
1/2 転送	ブロックのデータが半分転送されるとセット
転送完了	ブロック転送完了でセット
転送エラー	データ転送中にエラーが発生したときにセット
グローバル割込み	1/2 転送の完了、転送の完了、または転送エラーのイベントが発生するとセット

各 DMA チャネルは、以下の 4 種類のイベントでトリガされる割込みをソフトウェアに通知できます。

- ハーフブロック転送の完了
- ブロック転送の完了
- 転送エラー
- 上記の 3 種類の任意の組み合わせ(グローバル)

DMAMUX のブロック図



DMAMUX には、2 つの主なサブブロックとして、リクエスト・マルチプレクサとリクエスト・ジェネレータがあります。

DMAMUX リクエスト・マルチプレクサにより、ペリフェラルから DMA コントローラに DMA リクエストをルーティングできます。このルーティング機能は、プログラム可能なマルチチャンネル DMA リクエスト・マルチプレクサによって確保されます。

各チャンネルは、無条件に、またはイベントと同期して、DMAMUX 同期入力から一意の DMA リクエストを選択します。

DMAMUX は、入カトリガ信号によるプログラム可能イベントから DMA リクエスト・ジェネレータとしても使用できます。

DMA リクエスト・マルチプレクサ・チャンネルは、DMA コントローラへのリクエストと、同期入力およびトリガ入力として使用できるイベントの両方を生成します。

DMA リクエスト・ジェネレータ・チャンネル (0 ~ 3) と DMA リクエスト・マルチプレクサ・チャンネル (0 ~ 13) を混同しないでください。

DMAMUX リクエスト・マルチプレクサ (1/6)

- 各マルチプレクサ・チャンネルの特性
 - 設定レジスタ: DMAMUX_CxCR
 - プログラム可能な入力リクエスト選択 (DMAREQ_ID フィールドを使用)
 - DMA モードで動作している各ペリフェラルからのリクエストごとに DMAREQ_ID を割り当て
 - DMAREQ_ID = 0 は、DMA リクエストが選択されていない状態に相当
 - このチャンネルとそのルーティング先となる DMA コントローラ・チャンネルを設定すると、その DMA チャンネルを有効化可能
 - 同じ DMA リクエスト入力で 2 つの チャンネルを設定することは不可



life.augmented

9

マルチプレクサ・チャンネルごとに、次の機能を持つ設定レジスタ DMAMUX_CxCR があります。

- プログラム可能な入力リクエスト選択 (DMAREQ_ID フィールドを使用)。
 - ❖ DMA モードで動作している各ペリフェラルからのリクエストごとに DMAREQ_ID が割り当てられます。
 - ❖ DMAREQ_ID = 0x00 は、DMA リクエストが選択されていない状態に相当します。
- このチャンネルとそのルーティング先となる DMA コントローラ・チャンネルを設定すると、その DMA チャンネルを有効化できます。同じ DMA リクエスト入力で 2 つの チャンネルを設定することはできません。

DMAMUX リクエスト・マルチプレクサ (2/6)

- 各マルチプレクサ・チャンネルの特性
 - 内蔵 DMA リクエスト・カウンタ
 - プログラム可能 (NBREQ フィールドを使用)
 - DMA リクエストが処理されると、プログラムした DMA リクエスト・カウンタがデクリメント
アンダーランの際の処理
 - NBREQ フィールドにプログラムされた値で自動的に DMA リクエスト・カウンタをリロード
 - カウンタがアンダーランすると DMAMUX イベントを生成可能
 - EGE フィールドで有効にしている場合
 - チャンネル 0 と 1 で発生した 2 つの DMAMUX イベントがループバックされ、トリガ入力および同期入力として DMAMUX に接続
 - これにより、同期またはトリガを介して別の DMA チャンネルとのリクエスト・チェーンが可能



各マルチプレクサ・チャンネルの特性は次のとおりです。

- 内蔵 DMA リクエスト・カウンタは NBREQ フィールドを使用してプログラムできます。
- DMA リクエストが処理されると、プログラムした DMA リクエスト・カウンタがデクリメントされます。カウンタがアンダーランすると、NBREQ フィールドにプログラムした値で自動的に DMA リクエスト・カウンタがリロードされます。
- カウンタがアンダーランすると、EGE フィールドで DMAMUX イベントを有効にしていれば、そのイベントを生成できます。
- チャンネル 0 と 1 で発生した 2 つの DMAMUX イベントがループバックされ、トリガ入力および同期入力として DMAMUX に接続されます。これにより、同期またはトリガを介して別の DMA チャンネルとのリクエスト・チェーンを形成できます。

DMAMUX リクエスト・マルチプレクサ (3/6)

- 各マルチプレクサ・チャンネルの特性
 - 2つの動作モード(DMAMUX の SE フィールドでプログラム)
 - 無条件モード: 入力リクエストをそのまま出力
 - 同期モード: 複数のリクエストをグループ化して遅延や同期を適用
 - DMA がセキュアであればその関連チャンネルもセキュア(DMA SECM フィールドでプログラム)
 - セキュアな CM0+ のみでアクセス可能であればセキュア
 - CM4 と CM0+ の両方でアクセス可能であれば非セキュア



マルチプレクサ・チャンネルごとに 2 つの動作モードがあります (DMAMUX の SE フィールドでプログラムします)。

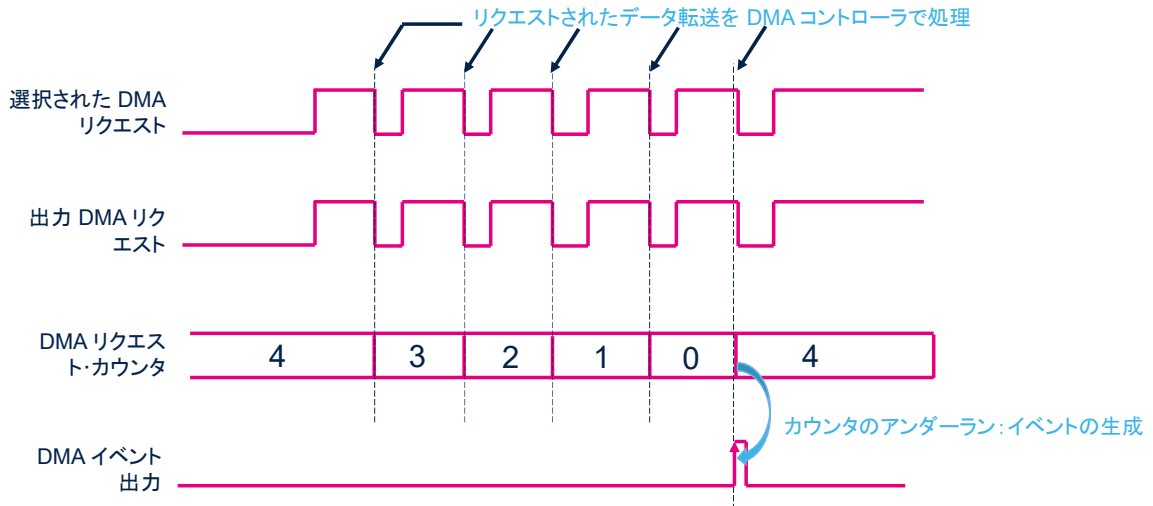
- 無条件モード: 入力リクエストをそのまま出力します。
- 同期モード: 複数のリクエストをグループ化して遅延や同期を適用します。

マルチプレクサ・チャンネルごとに、関連する DMA チャンネルと同時にセキュリティが有効になります (DMA の SECM フィールドでプログラムします)。

- セキュアな CM0+ のみでアクセス可能であればセキュアです。
- CM4 と CM0+ の両方でアクセス可能であれば非セキュアです。

DMAMUX リクエスト・マルチプレクサ (4/6) 無条件モード

- DMAMUX_CxCR の設定例: SE=0、NBREQ=4、EGE=1



リクエスト・マルチプレクサ・チャンネルを無条件に設定すると (SE=0)、DMA コントローラによる調整に従い、DMA リクエストがそのまま送信されます。

DMA コントローラでデータ転送が処理されると、DMA リクエストがネゲートされ、内蔵 DMA リクエスト・カウンタがデクリメントされます。

EGE フィールドでイベントを有効にしておくと、カウンタがアンダーランしたときにそのイベントを生成できます。

DMAMUX リクエスト・マルチプレクサ (5/6)

同期モード

- 同期モードでは以下の処理を追加
 - リクエストの条件設定
 - プログラム可能な同期入力選択 (SYNC_ID フィールドを使用)
 - プログラム可能な同期イベント: なし、立ち上がりエッジ、立ち下がりエッジ、両方のエッジ (SPOL フィールドを使用)
 - イベント生成にも使用できる単一の内蔵リクエスト・カウンタ (NBREQ フィールドを使用)
 - 同期イベント後の動作
 - 出力された DMA リクエストをペンディング中の入力リクエストに接続
 - カウンタがアンダーランしたときの動作
 - DMA 出力リクエストをマルチプレクサ・チャンネル入力から切り離し
 - 同期オーバーラン・フラグ (DMAMUX_CSR の SOFx) を報告
 - カウンタがアンダーランする前に新たな同期イベントが発生した場合
 - SOIE フィールドで割込みが有効になっていれば、その割込みを生成

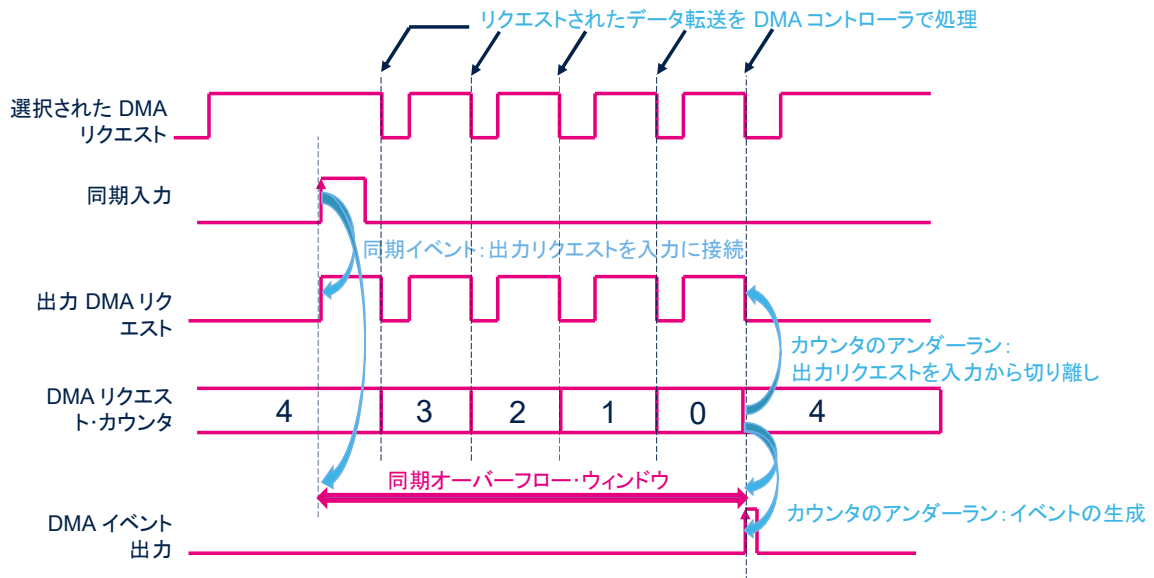


同期モードでは以下の処理も実行されます。

- リクエストの条件設定:
 - ❖ プログラム可能な同期入力選択 (SYNC_ID フィールドを使用)
 - ❖ プログラム可能な同期イベント: なし、立ち上がりエッジ、立ち下がりエッジ、両方のエッジ (SPOL フィールドを使用)
 - ❖ イベント生成にも使用できる単一の内蔵リクエスト・カウンタ (NBREQ フィールドを使用)
- 同期イベント後の動作:
 - ❖ 出力された DMA リクエストをペンディング中の入力リクエストに接続
- カウンタがアンダーランしたときの動作:
 - ❖ DMA 出力リクエストをマルチプレクサ・チャンネル入力から切り離し
- 最後に、同期オーバーラン・フラグ (DMAMUX_CSR の SOFx) を報告
 - ❖ カウンタがアンダーランする前に新たな同期イベントが発生した場合
 - ❖ SOIE フィールドで割込みが有効になっていれば、その割込みを生成

DMAMUX リクエスト・マルチプレクサ (6/6) 同期モード

- DMAMUX_CxCR の設定例: SE=1、NBREQ=4、EGE=1



DMAMUX チャンネルが同期モードに設定されている場合、その動作は以下ようになります。

リクエスト・マルチプレクサ入力(ペリフェラルからの DMA リクエスト)をペンディングにすることができ、同期イベントを受信するまで、この入力は DMAMUX リクエスト・マルチプレクサ出力に転送されなくなります。

同期イベントを受信すると、リクエスト・マルチプレクサによってその入力と出力が接続され、すべてのペリフェラル・リクエストが転送されます。

DMA リクエストが転送されて許可されるたびに、プログラムで定義した値でリクエスト・マルチプレクサ・カウンタがデクリメントされます。

このカウンタがゼロに達すると、DMA コントローラとペリフェラルの間の接続は切断され、新しい同期イベントを待ちます。

カウンタがアンダーランするたびに、2 番目の DMAMUX リクエスト・マルチプレクサ・チャンネルを同期またはトリガするオプションのイベントをリクエスト・マルチプレクサで生成できます。

CPU の介入なくシステムを STOP モードに戻す一部の低消費電力シナリオで、この同じイベントを使用できます。

同期モードを使用して、たとえばタイマにデータ転送が自動的に同期する設定や、同期入力としてマッピングされるペリフェラル・イベントからの転送に対する条件設定などができます。

また、プログラムした回数の DMA リクエストが 2 つの同期イベントの間で完了しなかった場合に、同期オーバーフローをソフトウェアに通知できます。

DMAMUX リクエスト・ジェネレータ(1/2)

• チャンネルごとの動作

- トリガ・イベントに続き、DMA リクエストを生成可能、それを DMAMUX リクエスト・マルチプレクサ・チャンネルの入力として選択可能(DMAMUX_CxCR の DMAREQ_ID フィールドで指定)
- GE フィールドによって有効になっていれば、設定レジスタ DMAMUX_RGxCR の設定によって、以下でリクエストを生成
 - プログラム可能なトリガ入力選択(SIG_ID フィールドを使用)
 - プログラム可能なトリガ・イベント:なし、立ち上がりエッジ、立ち下がりエッジ、両方のエッジ(GPOL フィールドを使用)
 - 内蔵リクエスト・カウンタ(GNBREQ フィールドを使用)
- DMA リクエストが処理されると、プログラムされたリクエスト・カウンタをデクリメント
- アンダーランの際の処理
 - GNBREQ フィールドにプログラムされた値で自動的にリクエスト・カウンタをリロード
 - リクエスト・ジェネレータでのリクエスト生成を停止
- トリガのオーバーラン・フラグ(DMAMUX_RGSR の OFx)を報告
 - カウンタがアンダーランする前に新たなトリガ・イベントが発生した場合
 - OIE フィールドで割込みが有効になっていれば、その割込みを生成



15

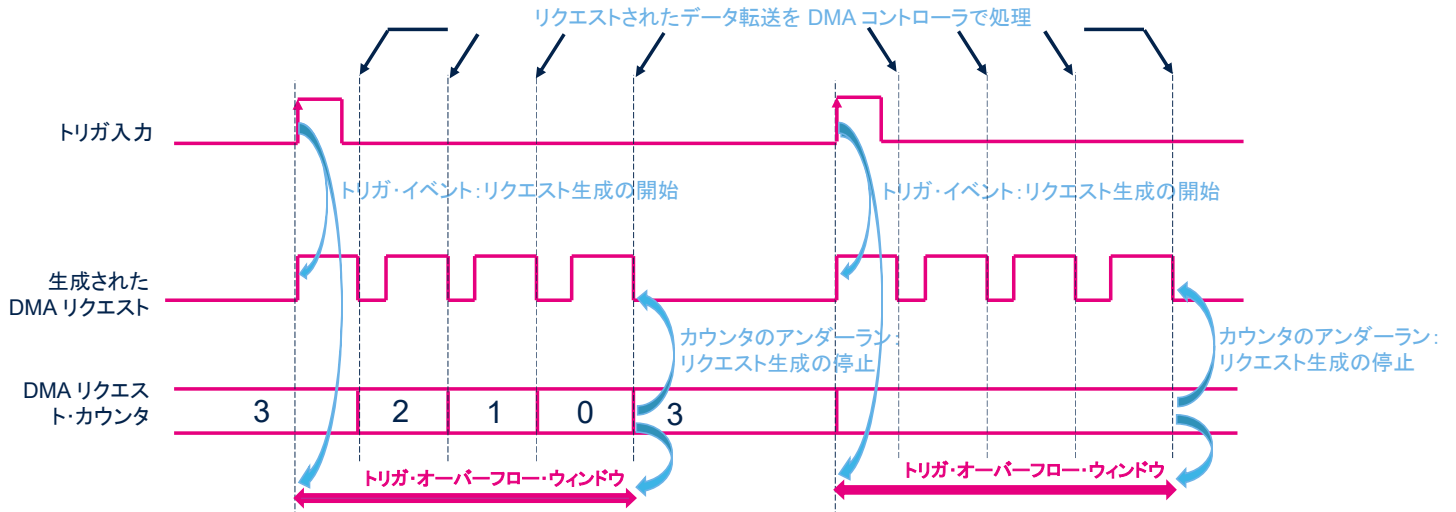
リクエスト・ジェネレータ・チャンネルごとの動作:

トリガ・イベントに続いて DMA リクエストを生成し、DMAMUX リクエスト・マルチプレクサ・チャンネルの入力として選択できます(DMAMUX_CxCR の DMAREQ_ID フィールドで指定)。

- GE フィールドによって有効になっていれば、設定レジスタ DMAMUX_RGxCR の設定によって、以下でリクエストが生成:されます。
 - ❖ プログラム可能なトリガ入力選択(SIG_ID フィールドを使用)
 - ❖ プログラム可能なトリガ・イベント:なし、立ち上がりエッジ、立ち下がりエッジ、両方のエッジ(GPOL フィールドを使用)
 - ❖ 内蔵リクエスト・カウンタ(GNBREQ フィールドを使用)
- DMA リクエストが処理されると、プログラムしたリクエスト・カウンタがデクリメントされます。
- アンダーランの際の処理
 - ❖ GNBREQ フィールドにプログラムされた値で自動的にリクエスト・カウンタをリロード
 - ❖ リクエスト・ジェネレータでのリクエスト生成を停止
- トリガのオーバーラン・フラグ(DMAMUX_RGSR の OFx)が報告:されます。
 - ❖ カウンタがアンダーランする前に新たなトリガ・イベントが発生した場合
 - ❖ OIE フィールドで割込みが有効になっていれば、その割込みを生成

DMAMUX リクエスト・ジェネレータ(2/2)

- DMAMUX_RGxCR の設定例: GE=0、GPOL=1、GNBREQ=3



トリガ・イベントでは、プログラムした回数 (GNBREQ+1) の DMA リクエストが生成されます。
GNBREQ+1 回のリクエストとデータ転送の完了の前に 2 回のトリガ・イベントが発生すると、トリガ・オーバーフローになることがあります。

DMAMUX リクエスト・マルチプレクサ入力

DMAREQ_ID	リソース	DMAREQ_ID	リソース
1	dmamux_req_gen0	22	LPUART1_TX
2	dmamux_req_gen1	23	TIM1_CH1
3	dmamux_req_gen2	24	TIM1_CH2
4	dmamux_req_gen3	25	TIM1_CH3
5	ADC1	26	TIM1_CH4
6	DAC_OUT1	27	TIM1_UP
7	SPI1_RX	28	TIM1_TRIG
8	SPI1_TX	29	TIM1_COM
9	SPI2_RX	30	TIM2_CH1
10	SPI2_TX	31	TIM2_CH2
11	I2C1_RX	32	TIM2_CH3
12	I2C1_TX	33	TIM2_CH4
13	I2C2_RX	34	TIM2_UP
14	I2C2_TX	35	TIM16_CH1
15	I2C3_RX	36	TIM16_UP
16	I2C3_TX	37	TIM17_CH1
17	USART1_RX	38	TIM17_UP
18	USART1_TX	39	AES1_IN
19	USART2_RX	40	AES1_OUT
20	USART2_TX	41	SUBGHZSPI_RX
21	LPUART1_RX	42	SUBGHZSPI_TX

この表は、あらゆるチャンネルに対する DMAMUX リクエスト・マルチプレクサ入力の STM32WL5 マッピングを示しています。リクエスト入力の割り当ては、どの DMAMUX リクエスト・マルチプレクサ・チャンネルでも DMAREQ_ID でプログラムします (DMAMUX_CxCR レジスタ)。同じリクエスト入力を 2 つの異なるチャンネルにマッピングしないでください。

DMAMUX トリガと同期入力

SIG_ID SYNC_ID	リソース	SIG_ID SYNC_ID	リソース
0	EXTI ライン 0	12	EXTI ライン 12
1	EXTI ライン 1	13	EXTI ライン 13
2	EXTI ライン 2	14	EXTI ライン 14
3	EXTI ライン 3	15	EXTI ライン 15
4	EXTI ライン 4	16	dmamux_evt0
5	EXTI ライン 5	17	dmamux_evt1
6	EXTI ライン 6	18	LPTIM1_OUT
7	EXTI ライン 7	19	LPTIM2_OUT
8	EXTI ライン 8	20	LPTIM3_OUT
9	EXTI ライン 9		
10	EXTI ライン 10		
11	EXTI ライン 11		

この表は、あらゆるチャンネルに対するトリガ入力および同期入力の STM32WL5 マッピングを示しています。

DMAMUX リクエスト・ジェネレータのどの x チャンネルでも、SIG_ID フィールドでトリガ入力の割り当てをプログラムします (DMAMUX_RGxCR レジスタ)。

DMAMUX リクエスト・マルチプレクサ・チャンネルのどの x チャンネルでも、SYNC_ID フィールドで同期入力の割り当てをプログラムします (DMAMUX_CxCR レジスタ)。

DMAMUX 割込み

- リクエスト・ジェネレータ・チャンネルごとの割込みイベント

割込みイベント	説明
リクエスト・ジェネレータのトリガ・オーバーラン	トリガ入力のオーバーランを検出したときにセット (その前に、プログラムした回数の DMA リクエストが DMAMUX リクエスト・ジェネレータで生成されて完了済み)

- リクエスト・マルチプレクサ・チャンネルごとの割込みイベント

割込みイベント	説明
リクエスト・マルチプレクサの同期オーバーラン	同期入力のオーバーランを検出したときにセット (その前に、プログラムした回数で送信された DMA リクエストまたは生成された DMAMUX イベントが完了済み)



DMAMUX リクエスト・ジェネレータの各チャンネルからソフトウェアにトリガのオーバーランを通知できます。
DMAMUX リクエスト・マルチプレクサの各チャンネルからソフトウェアに同期のオーバーランを通知できます。

低消費電力モードの DMA/DMAMUX

モード	説明
RUN	アクティブ
低消費電力 RUN	アクティブ
SLEEP	アクティブ、DMA/DMAMUX 割込みで CPU をウェイクアップ可能
低消費電力 SLEEP	アクティブ、DMA/DMAMUX 割込みで CPU をウェイクアップ可能
STOP 0/STOP 1	クロックオフおよび停止、DMA/DMAMUX レジスタ保持
STOP 2	パワーダウン、STANBY モード終了後に DMA/DMAMUX の再初期化が必要
STANDBY	パワーダウン、STANBY モード終了後に DMA/DMAMUX の再初期化が必要
SHUTDOWN	パワーダウン、SHUTDOWN モード終了後に DMA/DMAMUX の再初期化が必要



この表は、電力モードに応じた DMA コントローラと DMAMUX の状態を示しています。SLEEP モードと低消費電力 SLEEP モードの DMA コントローラと DMAMUX はアクティブな状態を維持しているため、たとえば UART や I2C で受信したキャラクタのメモリへの転送や、CPU のウェイクアップなどに使用できます。