

# STM32WB – DMA & 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)

1.0 版



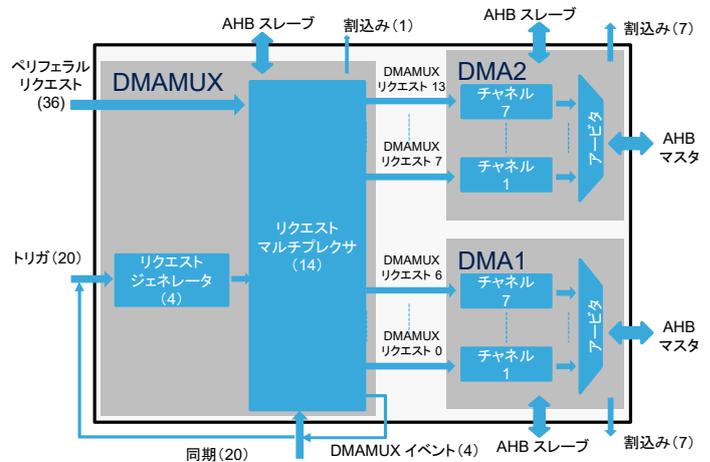
STM32WB ダイレクトメモリアクセスコントローラ (DMA) のプレゼンテーションによろこそ。  
ここでは、新しい DMA リクエストマルチプレクサ (DMAMUX) モジュールによって強化された、DMA コントローラモジュールの主な機能の説明を行います。

• それぞれ次の処理を実行可能な DMA コントローラ x2:

- 独立に設定可能でプログラム可能な同時 7 チャンネルブロック転送
- プログラム可能なチャンネルベース優先度
- AHB マスタポート経由のデータ転送

• DMA リクエストルータ (DMAMUX) x1

- プログラム可能なリクエストソース選択
  - DMA モードでペリフェラルから
  - トリガからと内部生成
- 同期モード: リクエストカウンタを用いて、同期入力から
- リクエスト連鎖: リクエストカウンタを用いて、別のリクエスト/チャンネルへの入カトリガあるいは同期であるイベントを生成



## アプリケーション側の利点

- メモリマップされた転送元からメモリマップされた転送先へのデータ転送に CPU をオフロード



アプリケーションにとっての DMA の主な利点は、あらゆるメモリマップされた転送元からあらゆるメモリマップされた転送先へのデータ転送に CPU をオフロードすることです。

### STM32WB DMA の機能:

- 2 DMA コントローラ。DMA コントローラごとに次の処理が可能:
  - ❖ プログラム可能な同時 7 チャンネルブロック転送 (チャンネルごとに独立設定可能)
  - ❖ プログラム可能なチャンネルベース優先度
  - ❖ (バスマトリックスに接続された) AHB マスタポート経由のデータ転送
- DMA リクエストルータ (DMAMUX) の機能:
  - ❖ プログラム可能なリクエストソース選択: DMA モードでペリフェラルから、あるいは、トリガ後に内部生成
  - ❖ 同期モード: DMAMUX リクエストカウンタを用いて同期入力 (ハードウェアイベント) から
  - ❖ リクエスト連鎖: DMAMUX リクエストカウンタを用いて、別のリクエスト/チャンネルへの入カトリガあるいは同期であるイベントを生成

# STM32WB DMA & DMAMUX インスタンス

3

DMAMUX の機能	DMAMUX
ペリフェラルリクエストの数	36
リクエストジェネレータの数	4
トリガの数	20
同期の数	20
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



ペリフェラルリクエストは 36 本、DMAMUX リクエストジェネレータは 4 個あります。  
トリガと同期入力は 20 本あります。  
DMA チャンネル／リクエストは 14 本あります。

- 1 個の DMA コントローラに個別に設定可能なチャンネル
  - チャンネルは、メモリからペリフェラルまたはペリフェラルからメモリへの転送において、ペリフェラルからの DMA ハードウェアリクエストに割当て可能です。
  - あるいは、メモリ間転送におけるソフトウェアリクエストにチャンネルを割り当てます。
  - チャンネルは優先順位レベルを付けてプログラムされます。
  - チャンネルはブロックレベルでのデータ転送回数に対してプログラムされます。
- ソフトウェアは、ブロック転送完了、ハーフブロック転送完了、転送エラーなどのプログラム可能なイベントに対する個別の割込みやフラグを通じてチャンネルを制御
  - バスアクセスエラーの場合、障害のあるチャンネルは自動的に無効となります。



DMA コントローラに焦点を合わせて説明します。

DMA コントローラのチャンネルごとに個別に設定可能:

- チャンネルは、ペリフェラルからメモリまたはメモリからペリフェラルへのデータ転送において、ペリフェラルからの DMA ハードウェアリクエストに割当て可能です。
- あるいは、メモリ間データ転送におけるソフトウェアリクエストにチャンネルを割り当てます。
- チャンネルは優先順位レベル付きでプログラムされます。
- チャンネルはブロックレベルでのデータ転送回数に対してプログラムされます。

ソフトウェアは、ブロック転送完了、ハーフブロック転送完了、転送エラーなどのプログラム可能なイベントに対する個別の割込みやフラグを通じてチャンネルを制御可能です。

バスアクセスエラーの場合、障害のあるチャンネルは自動的に無効となります。

- ブロックレベルでのプログラム可能なデータ転送
  - 転送元と転送先のデータサイズは独立: 8bit、16bit、32bit
  - 転送元と転送先の開始アドレスは独立
  - 転送元と転送先のアドレスインクリメントは独立: 連続インクリメントまたは固定アドレス
  - プログラム可能なブロック内転送データ数
    - 最大 65,535 個の転送元データ
    - ハードウェアにより自動デクリメント
- サークュラバッファモード(ペリフェラルとの連続データ転送)におけるブロック転送完了時:
  - 次のプログラムされた情報がハードウェアにより自動で再ロード:
    - ブロック内で転送されるデータ量
    - 転送元と転送先の開始アドレス



チャンネルはブロックレベルでのデータ転送回数に対してプログラムされます。

- 転送元と転送先のデータサイズは独立
- 転送元と転送先の開始アドレスは独立
- 転送元と転送先のアドレスインクリメントは独立(連続インクリメントまたは固定アドレス)
- プログラム可能なブロック内転送データ数
  - ❖ 最大 65,535 個の転送元データ
  - ❖ ハードウェアにより自動デクリメント

サーキュラバッファモード(ペリフェラルとの連続データ転送)におけるブロック転送完了時:

- プログラムされたブロック内転送データ数は、ハードウェアによって自動的に再ロードされます。
- 転送元と転送先の開始アドレスも同様です。

- **メモリ間モード**
  - チャンネルがこのモードで有効(ハードウェアリクエストなし)になると、直ちにブロック転送が開始されます。
- **ペリフェラルからメモリモードおよびメモリからペリフェラルモード:**
  - 1) チャンネルが有効になり、かつ2) ペリフェラルが DMA ハードウェアリクエストを送信すると、直ちにブロック転送が開始されます。
  - DMA ハードウェアリクエストによって、DMA データ転送が(1 つだけ) 特定されます。
  - 各データが転送先に正常に転送されると、DMA ハードウェアリクエストごとに DMA による速度調整と許可が行われます。
- **全モード:**
  - データ転送ごとにチャンネルアービトレーションが再評価されます。



メモリ間モードでチャンネルが有効(ハードウェアリクエストなし)になると、直ちにブロック転送が開始されます。

その一方で、ペリフェラルからメモリモードおよびメモリからペリフェラルモードにおいて:

- 1) チャンネルが有効になり、かつ2) ペリフェラルが DMA ハードウェアリクエストを送信すると、直ちにブロック転送が開始されます。
- DMA ハードウェアリクエストによって、DMA データ転送が(1 つだけ) 特定されます。
- 各データが転送先に正常に転送されると、DMA ハードウェアリクエストごとに DMA による速度調整と許可が行われます。

どのモードでも、データ転送ごとにチャンネルアービトレーションが再評価されます。

## • チャンネルごとの割込みイベント

割込みイベント	説明
1/2 転送	データブロックの半分が転送されたときにセット
転送完了	ブロック転送が完了したときにセット
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、転送完了、または転送エラーのイベントが発生したときに常にセット

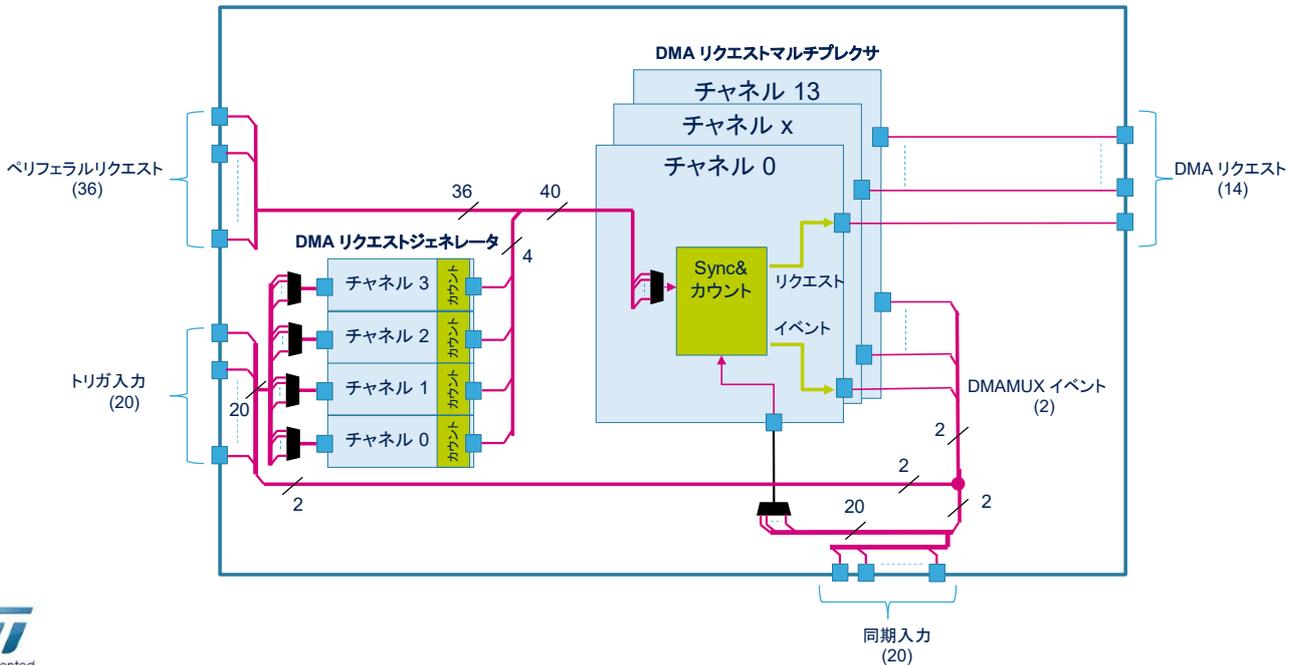


life.augmented

DMA チャンネルごとに、次の 4 種類の起こり得るイベントによってトリガされた割込みをソフトウェアに通知可能です。

- ハーフブロック転送完了
- ブロック転送完了
- 転送エラー
- 上記 3 イベントのいずれか(グローバル)

# DMAMUX ブロック図



DMAMUX には、リクエストマルチプレクサとリクエストジェネレータの 2 つの主なサブブロックがあります。

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 コントローラへの DMA リクエストをルーティングすることができます。

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り保証されます。

各チャンネルは、DMAMUX 同期入力のイベントで無条件に、または同期的に一意の DMA リクエストを選択します。

DMAMUX は、入カトリガ信号のプログラム可能なイベントからの 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 コントローラに対するリクエストと、同期入力としてもトリガ入力としても使用可能な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) のことを、DMA リクエストマルチプレクサチャンネル (0~13) と混同しないようにしてください。

# 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に対して
  - 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 フィールド経由)
      - DMA モードで動作している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 = 0 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、それがルーティングされる DMA コントローラが設定された後に、DMA チャンネルが有効になることができます。
    - 同じ DMA リクエスト入力を 2 つの異なるチャンネルに設定することはできません。



マルチプレクサチャンネルごとに設定レジスタがあります。

## DMAMUX\_CxCR

- プログラム可能な入力リクエスト選択 (DMAREQ\_ID フィールド経由)
  - ❖ DMA モードで動作している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、それがルーティングされる DMA コントローラが設定された後に、DMA チャンネルの有効化が可能となります。同じ DMA リクエスト入力を 2 つの異なるチャンネルに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に対して
  - 内蔵 DMA リクエストカウンタ
    - プログラム可能 (NBREQ フィールド経由)
  - 処理された DMA リクエストによって、プログラムされた DMA リクエストカウンタがデクリメントされます。アンダーラン時には:
    - NBREQ フィールドにプログラムされた値に DMA リクエストカウンタが自動的に再ロードされます。
  - アンダーラン時には、DMAMUX イベントの生成が可能です。
    - 有効化されている場合 (EGE フィールド経由)
  - (チャンネル 0 と 1 からの) 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。
    - これによって、同期やトリガを通じて、リクエストが別の異なる DMA チャンネルに連鎖できます。



各マルチプレクサチャンネルに対して:

- 内蔵 DMA リクエストカウンタが (NBREQ フィールド経由で) プログラム可能です。
- 処理された DMA リクエストによって、プログラムされた DMA リクエストカウンタがデクリメントされます。アンダーラン時には、NBREQ フィールドにプログラムされた値に DMA リクエストカウンタが自動的に再ロードされます。
- アンダーラン時には、有効化されている場合 (EGE フィールド経由)、DMAMUX イベントの生成が可能です。
- (チャンネル 0 と 1 からの) 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。これによって、同期やトリガを通じて、リクエストが別の異なる DMA チャンネルに連鎖できます。

# DMAMUX リクエストマルチプレクサ (3/6)

11

- 各マルチプレクサチャネルに対して
  - (SE フィールド経由でプログラムされた)2 種類の動作モード
    - 無条件モード: 入力リクエストがそのまま出力
    - 同期モード: いくつかのリクエストがグループ化されて遅延/同期されます。



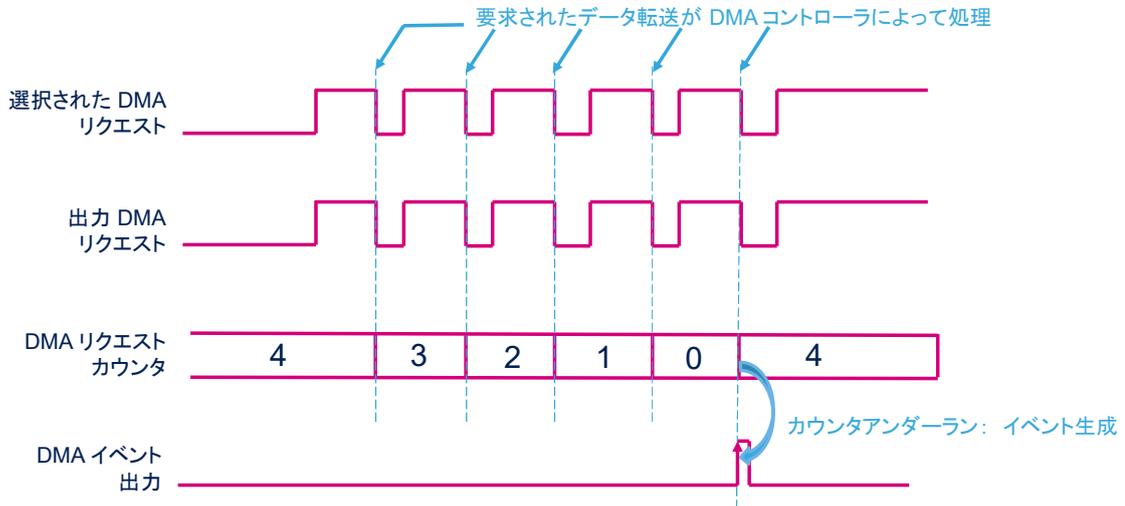
各マルチプレクサチャネルに対して、(SE フィールド経由でプログラムされた)2 種類の動作モードがあります。

- 無条件モード: 入力リクエストがそのまま出力されます。
- 同期モード: いくつかのリクエストがグループ化されて遅延/同期されます。

# DMAMUX リクエストマルチプレクサ (4/6) 無条件モード

12

- 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が無条件 (SE=0) に設定されている場合、DMA リクエストはそのまま、DMA コントローラによって速度調整された通りに送信されます。  
DMA コントローラがデータ転送を完了すると、DMA リクエストがデアサートされて、内蔵 DMA リクエストカウンタがデクリメントされます。  
カウンタのアンダーラン時には、EGE フィールド経由で有効化されている場合、イベントの生成が可能です。

# DMAMUX リクエストマルチプレクサ (5/6)

## 同期モード

13

- 同期モードでは、追加で
  - リクエストが次のもので条件付けされます。
    - プログラム可能な同期入力選択 (SYNC\_ID フィールド経由)
    - 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 / 両エッジ (SPOL フィールド経由)
    - イベント生成にも使用可能な単一の内蔵リクエストカウンタ (NBREQ フィールド経由)
  - 同期イベント後
    - 出力 DMA リクエストは保留中の入力リクエストに接続
  - カウンタアンダーラン時
    - DMA 出力リクエストはマルチプレクサチャネル入力から切断
  - 同期オーバーランフラグ (DMAMUX\_CSR の SOF<sub>x</sub>) が報告されます。
    - カウンタアンダーランの前に新しい同期イベントが発生した場合
    - (SOIE フィールド経由で) 有効になっている場合に割込みを生成



同期モードでは、追加で:

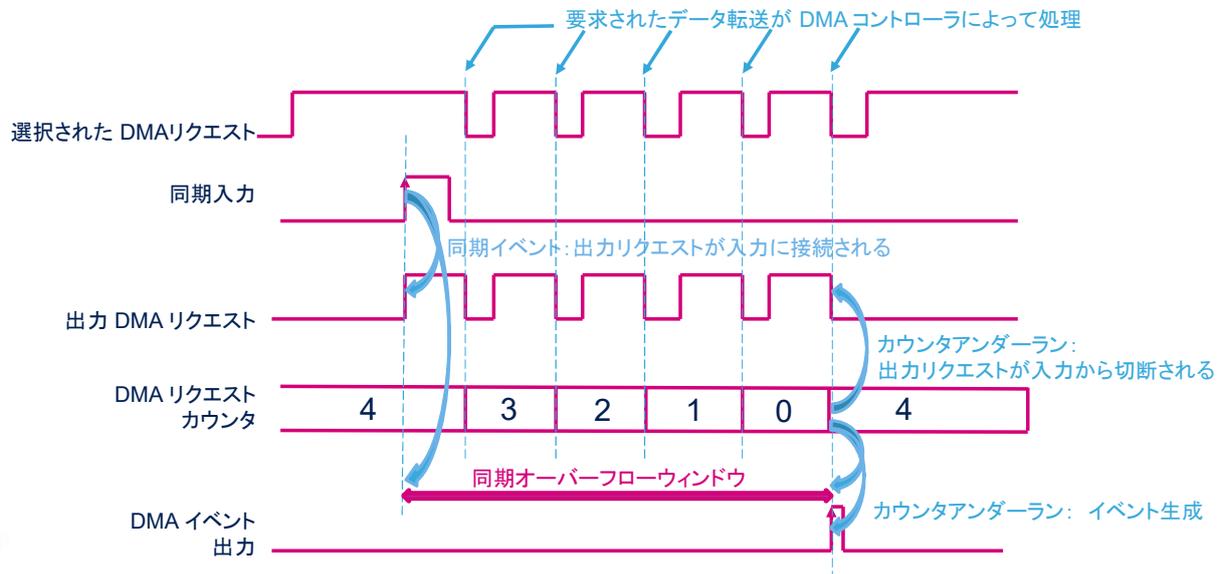
- リクエストが条件付けされる:
  - ❖ プログラム可能な同期入力選択 (SYNC\_ID フィールド経由)
  - ❖ プログラム可能な同期イベント: なし / 立ち上がりエッジ / 立ち下がりエッジ / 両エッジ (SPOL フィールド経由)
  - ❖ イベント生成にも使用可能な単一の内蔵リクエストカウンタ (NBREQ フィールド経由)
- 同期イベント後:
  - ❖ 出力 DMA リクエストは保留中の入力リクエストに接続
- カウンタアンダーラン時:
  - ❖ DMA 出力リクエストはマルチプレクサチャネル入力から切断
- 最後に、同期オーバーランフラグ (DMAMUX\_CSR の SOF<sub>x</sub>) が報告される:
  - ❖ カウンタアンダーランの前に新しい同期イベントが発生した場合
  - ❖ (SOIE フィールド経由で) 有効になっている場合に割込みを生成

# DMAMUX リクエストマルチプレクサ (6/6)

## 同期モード

14

- 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)は保留とすることが可能であり、同期イベントを受け取るまでは DMAMUX リクエストマルチプレクサ出力に転送されません。

その後、リクエストマルチプレクサはその入力と出力を接続し、すべてのペリフェラルリクエストが転送されます。

転送と許可が行われた DMA リクエストごとに、(規定されたプログラム値に設定されている)リクエストマルチプレクサカウンタをデクリメントします。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、リクエストマルチプレクサがオプションのイベントを生成して、2 番目の DMAMUX リクエストマルチプレクサチャンネルの同期やトリガを行うことができます。

同じイベントが、CPU の介入なくシステムを STOP モードに戻す一部の低電力シナリオで使用することができます。

同期モードは、たとえば、データ転送をタイマに自動同期させたり、同期入力としてマッピングされているあらゆるペリフェラルイベントからの転送を条件づけたりするために使用することができます。

加えて、同期オーバーフローによって、プログラムされている数の DMA リクエストが 2 つの同期イベントの間に完了していない場合に、ソフトウェアへの通知を行えます。

- 各チャンネルに対して
  - DMA リクエストがトリガイベントの後に生成可能です。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されます (DMAMUX\_CxCR の DMAREQ\_ID フィールド経由)。
  - (GE フィールドによって)有効化されている場合、設定レジスタ DMAMUX\_RGxCR 経由で次のものによって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 フィールド経由)
    - 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 / 両エッジ (GPOL フィールド経由)
    - 内蔵リクエストカウンタが (GNBREQ フィールド経由)
  - 処理された DMA リクエストによって、プログラムされたリクエストカウンタがデクリメントされます。
- アンダーラン時には:
  - GNBREQ フィールドにプログラムされた値にリクエストカウンタが自動的に再ロードされます。
  - リクエストジェネレータは、リクエストの生成を停止します。
- トリガオーバーランフラグ (DMAMUX\_RGSR の OFx) が報告されます。
  - カウンタアンダーランの前に新しいトリガイベントが発生した場合
  - (OIE フィールド経由で)有効になっている場合に割込みを生成



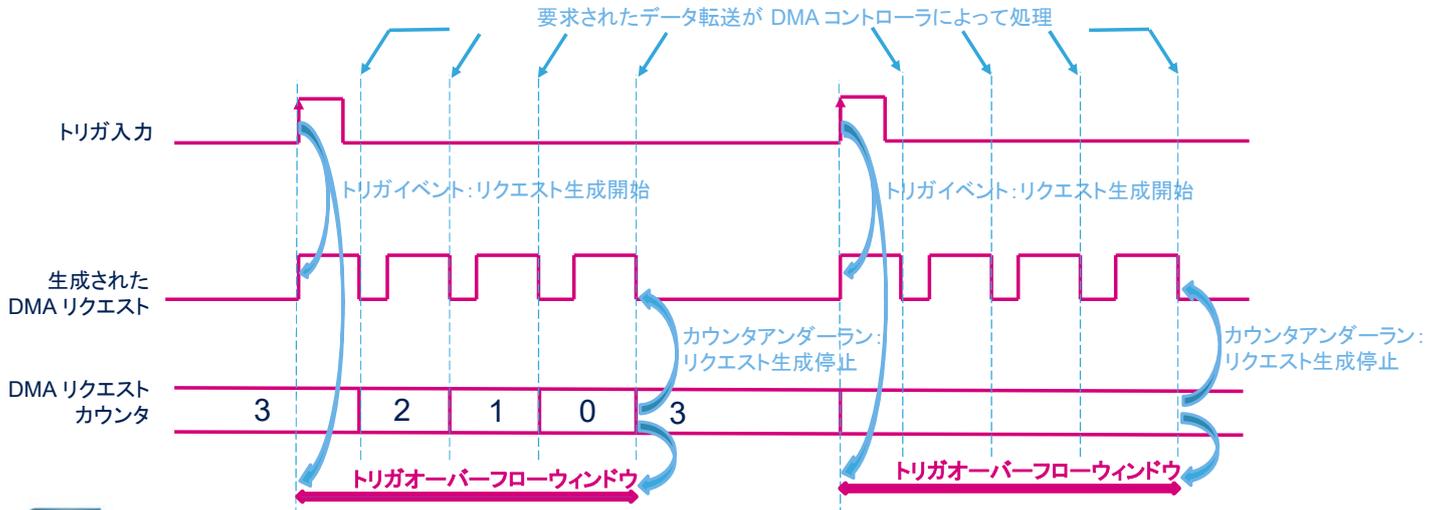
各リクエストジェネレータチャンネルに対して:

DMA リクエストがトリガイベントの後に生成可能であり、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されます (DMAMUX\_CxCR の DMAREQ\_ID フィールド経由)。

- (GE フィールドによって)有効化されている場合、設定レジスタ DMAMUX\_RGxCR 経由で次のものによって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 フィールド経由)
  - ❖ 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 / 両エッジ (GPOL フィールド経由)
  - ❖ 内蔵リクエストカウンタが (GNBREQ フィールド経由)
- 処理された DMA リクエストによって、プログラムされたリクエストカウンタがデクリメントされます。
- アンダーラン時には:
  - ❖ GNBREQ フィールドにプログラムされた値にリクエストカウンタが自動的に再ロードされます。
  - ❖ リクエストジェネレータは、リクエストの生成を停止します。
- トリガオーバーランフラグ (DMAMUX\_RGSR の OFx) が報告されません。
  - ❖ カウンタアンダーランの前に新しいトリガイベントが発生した場合
  - ❖ (OIE フィールド経由で)有効になっている場合に割込みを生成

# 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



トリガイベント発生時に、プログラムされた数の DMA リクエスト (GNBREQ+1) が生成されます。  
GNBREQ+1 回のリクエストの前にトリガイベントが 2 回発生し、データ転送が完了した場合、トリガオーバーフローが発生することがあります。

# 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	TIM1_CH2
2	dmamux_req_gen1	23	TIM1_CH3
3	dmamux_req_gen2	24	TIM1_CH4
4	dmamux_req_gen3	25	TIM1_UP
5	ADC1	26	TIM1_TRIG
6	SPI1_RX	27	TIM1_COM
7	SPI1_TX	28	TIM2_CH1
8	SPI2_RX	29	TIM2_CH2
9	SPI2_TX	30	TIM2_CH3
10	I2C1_RX	31	TIM2_CH4
11	I2C1_TX	32	TIM2_UP
12	I2C3_RX	33	TIM16_CH1
13	I2C3_TX	34	TIM16_UP
14	USART1_RX	35	TIM17_CH1
15	USART1_TX	36	TIM17_UP
16	LPUART1_RX	37	AES1_IN
17	LPUART1_TX	38	AES1_OUT
18	SAI1_A	39	AES2_IN
19	SAI1_B	40	AES2_OUT
20	QUADSPI		
21	TIM1_CH1		



この表には、STM32WBの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マッピングが示されています。

リクエスト入力の割当ては、あらゆる DMAMUX リクエストマルチプレクサチャンネルに対する DMAREQ\_ID (DMAMUX\_CxCR レジスタ)によりプログラムされます。

同じリクエスト入力は、異なる 2 つのチャンネルに配置することはできません。

# DMAMUX トリガ & 複数同期

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		
9	EXTI ライン 9		
10	EXTI ライン 10		
11	EXTI ライン 11		



この表には、あらゆるチャンネルに対するトリガ入力と同期入力の STM32WB のマッピングが示されています。

トリガ入力の割当ては、あらゆる DMAMUX リクエストジェネレータ x (DMAMUX\_RGxCR レジスタ) の SIG\_ID フィールドによりプログラムされます。

同期入力の割当ては、あらゆる DMAMUX リクエストマルチプレクサチャンネル x (DMAMUX\_CxCR レジスタ) の SYNC\_ID フィールドによりプログラムされます。

- 各リクエストジェネレータチャンネルに対する割込みイベント

割込みイベント	説明
リクエストジェネレータトリガオーバーラン	トリガ入力オーバーランが検出されたときにセット (その前に、DMAMUX リクエストジェネレータにより生成されたプログラムされた数の DMA リクエストが完了)

- 各リクエストマルチプレクサチャンネルに対する割込みイベント

割込みイベント	説明
リクエストマルチプレクサ同期オーバーラン	同期入力オーバーランが検出されたときにセット (その前に、プログラムされた数の送信された DMA リクエストまたは生成された DMAMUX イベントが完了)



DMAMUX リクエストジェネレータチャンネルごとに、トリガオーバーランをソフトウェアに通知可能です。  
DMAMUX リクエストマルチプレクサチャンネルごとに、同期オーバーランをソフトウェアに通知可能です。

# 低電力モードでの DMA/DMAMUX

モード	説明
RUN	有効。
低電力 RUN	有効。
SLEEP	有効DMA/DMAMUX 割込みが CPU をウェイクアップ可能です。
低電力 SLEEP	有効。DMA/DMAMUX 割込みが CPU をウェイクアップ可能です。
STOP0/STOP1/STOP2	クロックオフ & 停止。DMA/DMAMUX レジスタは保持
STANDBY	パワーダウン状態です。DMA/DMAMUX は、STANBY モード終了後に再初期化する必要があります。
SHUTDOWN	パワーダウン状態です。DMA/DMAMUX は SHUTDOWN モード終了後に再初期化する必要があります。



この表には、電源モードに対応する DMA コントローラ と DMAMUX の状態が示されています。  
SLEEP モードと低電力 SLEEP モードでは、DMA コントローラ と DMAMUX は有効のままであり、たとえば、UART や I2C で受信した文字をメモリに転送した後に CPU をウェイクアップするためなどに使用できます。