

STM32G4 – 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



STM32G4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へようこそ。

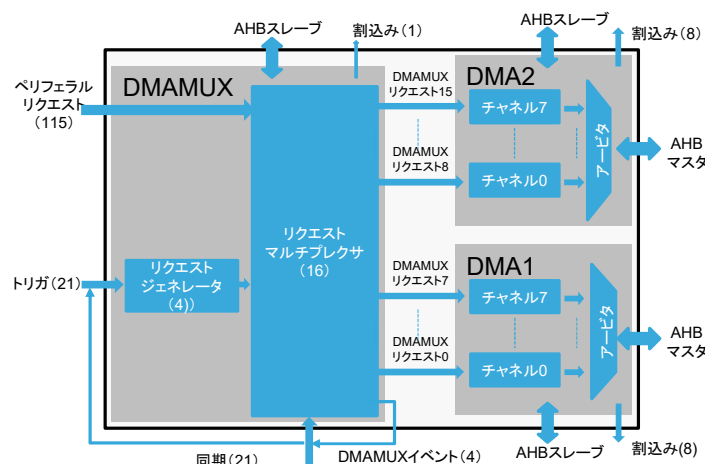
ここでは、新しいDMAリクエストマルチプレクサ (DMAMUX) モジュールによって強化された、DMAコントローラモジュールの主な機能の説明を行います。

- それぞれ次の処理が実行可能な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8チャンネル・ブロック転送
- プログラム可能なチャンネル・ベースの優先度
- 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/チャンネルへの入力トリガ、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ソースからメモリ・マップされたデスティネーションへのデータ転送
CPU負荷のオフロード

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負荷をオフロードすることです。

STM32G4 DMAの機能:

- 2DMA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8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/チャンネルへの入力トリガあるいは同期であるイベントを生成

STM32G4 DMAとDMAMUX例

3

DMAMUXの機能	DMAMUX
ペリフェラル・リクエストの数	115
リクエスト・ジェネレータの数	4
トリガの数	21
同期の数	21
出力DMAリクエストの数	16

DMAの機能	DMA1
チャンネル数	8

DMAの機能	DMA2
チャンネル数	8



ペリフェラルリクエストは115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21本あります。
DMAチャンネル/リクエストは16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
 - 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る
 - チャンネルは優先順位レベルを付けてプログラムされる
 - 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る
- 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る



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: 8ビット、16ビット、32ビット
 - 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
- サークキュラk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る
- 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る
 - 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)特定される
 - 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る
- 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る



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

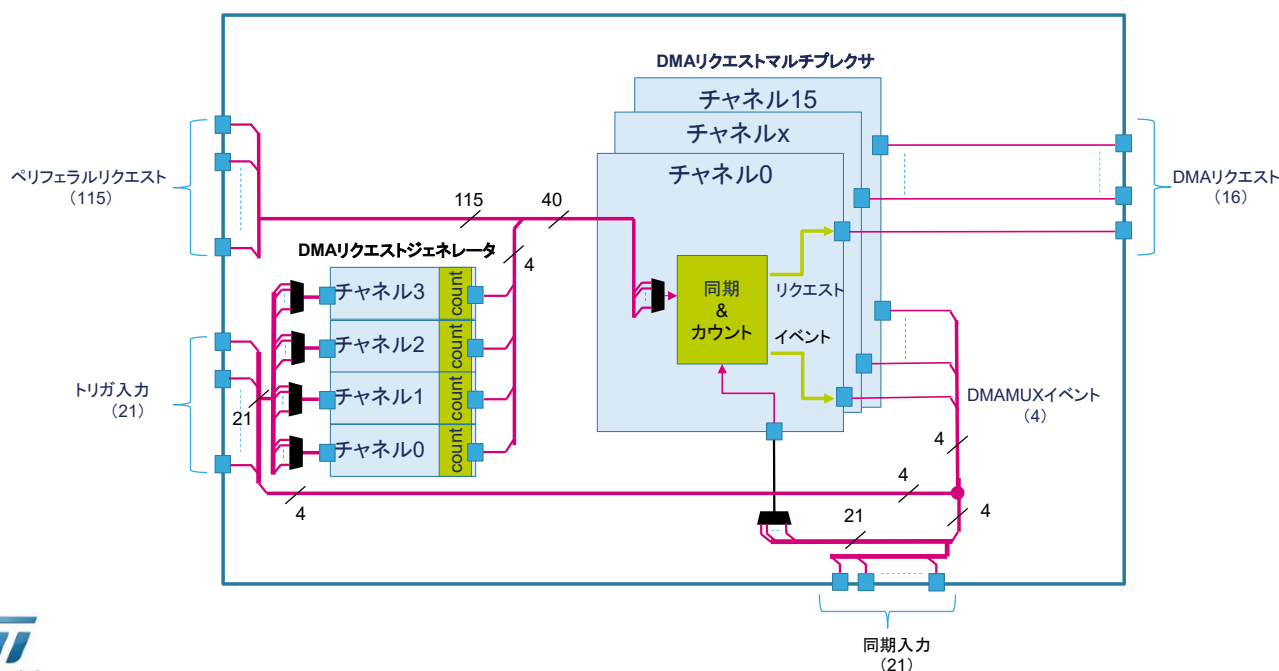


DMAチャンネルごとに、次の4種類の起こり得るイベントによってトリガされた割込みをソフトウェアに通知可能です。

- 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

8



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)

9

- 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が割り当てられる
 - DMAREQ_ID=0は、DMAリクエストが選択されていない状態に相当する
 - 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可能
 - 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出来ない



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)

10

- 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る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る
 - アンダーラン時には、DMAMUXイベントの生成が可能
 - 有効化されている場合(EGEフィールド経由)
 - (チャンネル0から3までの)4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る



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から3までの)4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・チャンネルに対して
 - (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/同期される



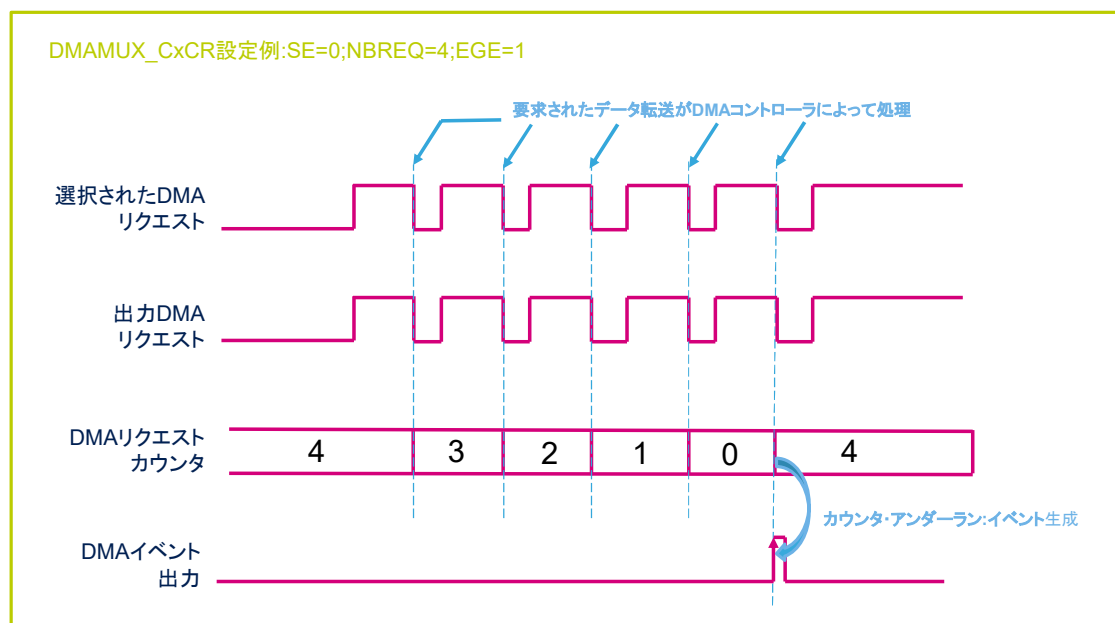
各マルチプレクサチャンネルに対して、(SEフィールド経由でプログラムされた)2種類の動作モードがあります。

- 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



リクエストマルチプレクサチャネルが無条件(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・チャンネル入力から切断
 - 同期オーバーランフラグ (DMAMUX_CSRのSOFx) が報告された
 - 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



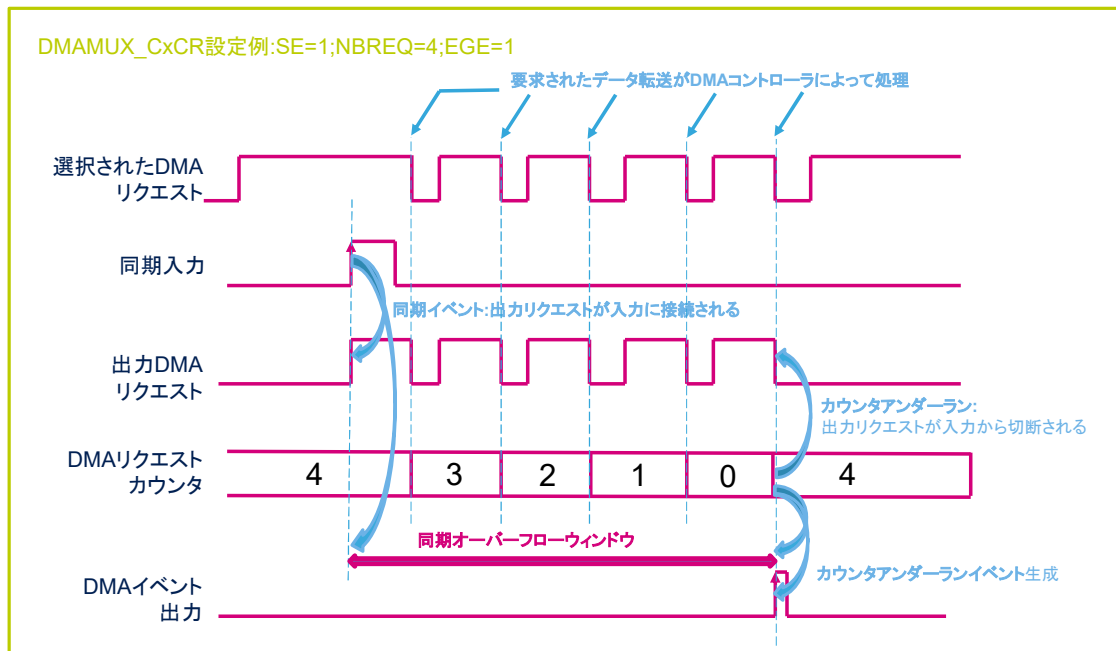
同期モードでは、追加で:

- 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ンネル入力から切断
- 最後に、同期オーバーランフラグ (DMAMUX_CSRのSOFx) 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、その動作は以下のようになります。

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。

- 各チャネルに対して
 - DMAリクエストがトリガイイベントの後に生成可能。DMAMUXリクエスト・マルチプレクサ・チャネルの入力として選択される。(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る
 - 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る
 - 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る
 - リクエスト・ジェネレータは、リクエストの生成を停止
 - トリガ・オーバーラン・フラグ(DMAMUX_RGSRのOFx)が報告される
 - カウンタアンダーランの前に新しいトリガイ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



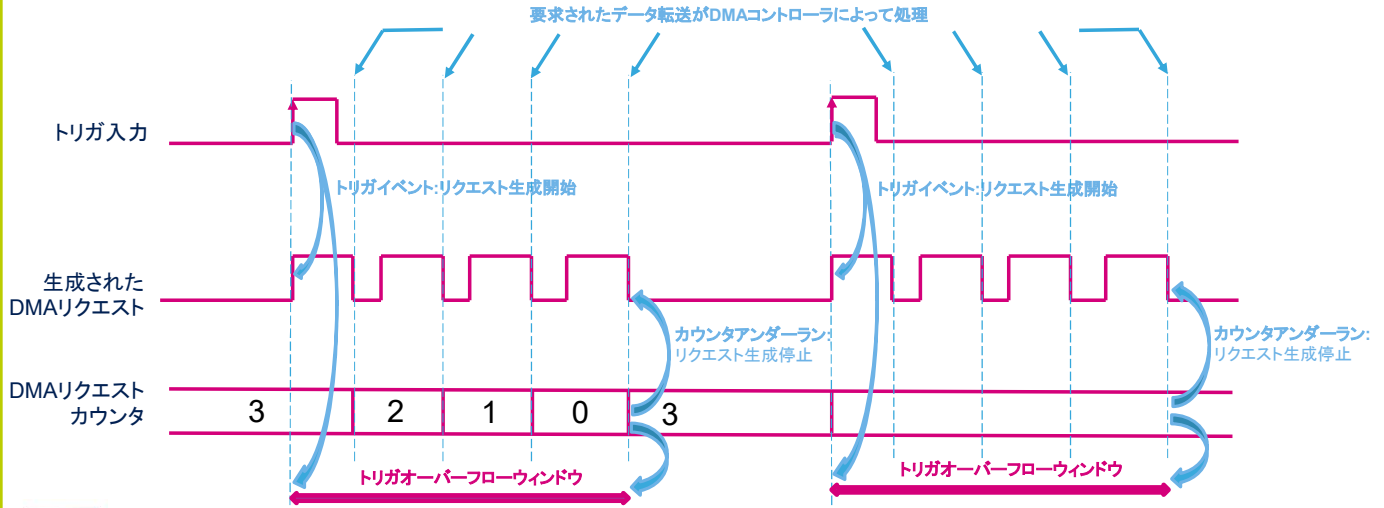
各リクエストジェネレータチャネルに対して:

- DMAリクエストがトリガイ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(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

16

DMAMUX_RGxCR設定例:GE=0;GPOL=1;GNBREQ=3



トリガイイベント発生時に、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

17

DMAREQ_ID	リソース	DMAREQ_ID	リソース	DMAREQ_ID	リソース
1	dmamux_req_gen0	44	TIM1_CH3	87	TIM20_CH2
2	dmamux_req_gen1	45	TIM1_CH4	88	TIM20_CH3
3	dmamux_req_gen2	46	TIM1_UP	89	TIM20_CH4
4	dmamux_req_gen3	47	TIM1_TRIG	90	TIM20_UP
5	ADC1	48	TIM1_COM	91	AES_IN
6	DAC1_CH1	49	TIM8_CH1	92	AES_OUT
7	DAC1_CH2	50	TIM8_CH2	93	TIM20_TRIG
8	TIM6_UP	51	TIM8_CH3	94	TIM20_COM
9	TIM7_UP	52	TIM8_CH4	95	HRTIM_MASTER
10	SPI1_RX	53	TIM8_UP	96	HRTIM_TIMA
11	SPI1_TX	54	TIM8_TRIG	97	HRTIM_TIMB
12	SPI2_RX	55	TIM8_COM	98	HRTIM_TIMC
13	SPI2_TX	56	TIM2_CH1	99	HRTIM_TMD
14	SPI3_RX	57	TIM2_CH2	100	HRTIM_TIME
15	SPI3_TX	58	TIM2_CH3	101	HRTIM_TIMF
16	I2C1_RX	59	TIM2_CH4	102	DAC3_CH1
17	I2C1_TX	60	TIM2_UP	103	DAC3_CH2
18	I2C2_RX	61	TIM3_CH1	104	DAC4_CH1
19	I2C2_TX	62	TIM3_CH2	105	DAC4_CH2
20	I2C3_RX	63	TIM3_CH3	106	SPI4_RX
21	I2C3_TX	64	TIM3_CH4	107	SPI4_TX
22	I2C4_RX	65	TIM3_UP	108	SAI1_A
23	I2C4_TX	66	TIM3_TRIG	109	SAI1_B
24	USART1_RX	67	TIM4_CH1	110	FMAC_Read
25	USART1_TX	68	TIM4_CH2	111	FMAC_Write
26	USART2_RX	69	TIM4_CH3	112	Cordic_Read
27	USART2_TX	70	TIM4_CH4	113	Cordic_Write
28	USART3_RX	71	TIM4_UP	114	USBPD_RX
29	USART3_TX	72	TIM5_CH1	115	USBPD_TX
30	UART4_RX	73	TIM5_CH2		
31	UART4_TX	74	TIM5_CH3		
32	UART5_RX	75	TIM5_CH4		
33	UART5_TX	76	TIM5_UP		
34	LPUART1_RX	77	TIM5_TRIG		
35	LPUART1_TX	78	TIM15_CH1		
36	ADC2	79	TIM15_UP		
37	ADC3	80	TIM15_TRIG		
38	ADC4	81	TIM15_COM		
39	ADC5	82	TIM16_CH1		
40	QUADSPI	83	TIM16_UP		
41	DAC2_CH1	84	TIM17_CH1		
42	TIM1_CH1	85	TIM17_UP		
43	TIM1_CH2	86	TIM20_CH1		



この表には、STM32G4のあらゆるチャネルに対する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ネルに対するDMAREQ_ID (DMAMUX_CxCRL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つのチャネルに配置することはできません。

DMAMUXトリガ&同期入力

18

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	dmamux_evt2
7	EXTIライン7	19	dmamux_evt3
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、あらゆるチャネルに対するトリガ入力と同期入力のSTM32G4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_RGxCRL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ネルx(DMAMUX_CxCRL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

20

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
低電力SLEEP	有効 DMA/DMAMUX割込みがCPUをウェイクアップ可能
STOP0/STOP1	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 DMA/DMAMUXは、STANDBYモード終了後に再初期化する必要がある
SHUTDOWN	パワーダウン状態 DMA/DMAMUXは、SHUTDOWNモード終了後に再初期化する必要がある



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。