

# 参考資料



life.augmented

AN4031

## アプリケーション・ノート

### STM32F2、STM32F4、および STM32F7 シリーズの DMA コントローラの使用

#### はじめに

本資料では、STM32F2、STM32F4、および STM32F7 シリーズで使用できるダイレクトメモリアクセス (DMA) コントローラの使用方法について説明します。DMA コントローラの機能、システムアーキテクチャ、マルチレイヤバスマトリックス、およびメモリシステムは、広いデータ帯域幅の提供および非常に遅延の少ない応答時間のソフトウェアの開発に貢献します。

本資料では、開発者がこれらの機能を最大限に活用して、さまざまなペリフェラルやサブシステムが適切な時間で応答できるようにするヒントとコツについても説明します。

STM32F2、STM32F4、および STM32F7 は「STM32F2/F4/F7 デバイス」、DMA コントローラは「DMA」と記されています。

本資料では、実例として、STM32F4 シリーズを選択しています。DMA の挙動は、特に指定がない限り、STM32F2、STM32F4、および STM32F7 の各シリーズで同一です。

#### 参照文献

本資料は、STM32F2/F4/F7 の各リファレンスマニュアルと合わせて読む必要があります。

- STM32F205/215 および STM32F207/217 リファレンスマニュアル (RM0033)
- STM32F405/415、STM32F407/417、STM32F427/437、および STM32F429/439 リファレンスマニュアル (RM0090)
- STM32F401xB/C および STM32F401xD/E リファレンスマニュアル (RM0368)
- STM32F410 リファレンスマニュアル (RM0401)
- STM32F411xC/E リファレンスマニュアル (RM0383)
- STM32F412 リファレンスマニュアル (RM0402)
- STM32F446xx リファレンスマニュアル (RM0390)
- STM32F469xx および STM32F479xx リファレンスマニュアル (RM0386)
- STM32F75xxx および STM32F74xxx リファレンスマニュアル (RM0385)
- STM32F76xxx および STM32F77xxx リファレンスマニュアル (RM0410)

# 参考資料

## 目次

AN4031

## 目次

|          |                                     |           |
|----------|-------------------------------------|-----------|
| <b>1</b> | <b>DMA コントローラの説明</b>                | <b>6</b>  |
| 1.1      | DMA 転送のプロパティ                        | 6         |
| 1.1.1    | DMA ストリーム / チャネル                    | 7         |
| 1.1.2    | ストリームの優先順位                          | 9         |
| 1.1.3    | 転送元 / 転送先アドレス                       | 9         |
| 1.1.4    | 転送モード                               | 9         |
| 1.1.5    | 転送サイズ                               | 10        |
| 1.1.6    | 転送元 / 転送先アドレスのインクリメント               | 10        |
| 1.1.7    | 転送元 / 転送先のデータ幅                      | 10        |
| 1.1.8    | 転送タイプ                               | 10        |
| 1.1.9    | DMA FIFO モード                        | 11        |
| 1.1.10   | 転送元 / 転送先のバーストサイズ                   | 12        |
| 1.1.11   | ダブルバッファモード                          | 13        |
| 1.1.12   | フロー制御                               | 14        |
| 1.2      | DMA 転送の設定                           | 14        |
| <b>2</b> | <b>システム性能に関する考慮事項</b>               | <b>16</b> |
| 2.1      | マルチレイヤバスマトリックス                      | 17        |
| 2.1.1    | 定義                                  | 17        |
| 2.1.2    | ラウンドロビンの優先順位方式                      | 17        |
| 2.1.3    | バスマトリックスアービトレーションと DMA 転送の遅延の最悪のケース | 19        |
| 2.2      | DMA 転送パス                            | 20        |
| 2.2.1    | デュアル DMA ポート                        | 20        |
| 2.2.2    | DMA 転送状態                            | 22        |
| 2.2.3    | DMA リクエストのアービトレーション                 | 23        |
| 2.3      | AHB – APB ブリッジ                      | 24        |
| 2.3.1    | デュアル AHB – APB ポート                  | 24        |
| 2.3.2    | AHB – APB ブリッジアービトレーション             | 24        |
| <b>3</b> | <b>DMA 遅延の予測方法</b>                  | <b>26</b> |
| 3.1      | DMA 転送時間                            | 26        |
| 3.1.1    | デフォルトの DMA 転送時間                     | 26        |
| 3.1.2    | DMA 転送時間と同時アクセス                     | 27        |
| 3.2      | 例                                   | 28        |

# 参考資料

AN4031

目次

|          |  |           |
|----------|--|-----------|
| 3.2.1    | ADC — SRAM DMA 転送 .....                              | 28        |
| 3.2.2    | SPI 全二重 DMA 転送 .....                                 | 29        |
| <b>4</b> | <b>DMA コントローラのプログラミングにおけるヒントとワーニング .....</b>         | <b>31</b> |
| 4.1      | DMA を無効にするソフトウェアシーケンス .....                          | 31        |
| 4.2      | 新しい転送を有効にする前の DMA フラグ管理 .....                        | 31        |
| 4.3      | DMA を有効にするソフトウェアシーケンス .....                          | 31        |
| 4.4      | NDTR=0 の場合のメモリ間転送 .....                              | 31        |
| 4.5      | PINC/MINC=0 の場合の DMA ペリフェラルバースト .....                | 32        |
| 4.6      | 2 回マッピングされている DMA リクエスト .....                        | 32        |
| 4.7      | 最適な DMA スループット設定 .....                               | 32        |
| 4.8      | DMA 転送の中止 .....                                      | 32        |
| 4.9      | DMA2 コントローラとシステムアーキテクチャの柔軟性の利用 .....                 | 33        |
| 4.9.1    | DMA2 AHB ポート上で転送を反転させる場合の考慮事項 .....                  | 33        |
| 4.9.2    | DMA2 AHB ポート経由の Quad-SPI 転送の反転の例に関する考慮事項 .....       | 34        |
| 4.10     | STM32F7 DMA 転送とデータのインコヒーレンシを回避するための<br>キャッシュ保持 ..... | 35        |
| <b>5</b> | <b>結論 .....</b>                                      | <b>36</b> |
| <b>6</b> | <b>改版履歴 .....</b>                                    | <b>37</b> |

# 参考資料

表の一覧

AN4031

## 表の一覧

|       |  |    |
|-------|--|----|
| 表 1.  | STM32F427/437 と STM32F429/439 の DMA1 リクエストマッピング..... | 8  |
| 表 2.  | STM32F427/437 と STM32F429/439 の DMA2 リクエストマッピング..... | 9  |
| 表 3.  | 実現可能な バースト設定 .....                                   | 13 |
| 表 4.  | ペリフェラルポートアクセス / 転送時間と使用する DMA パス .....               | 27 |
| 表 5.  | メモリポートアクセス / 転送時間 .....                              | 27 |
| 表 6.  | DMA ペリフェラル (ADC) ポートの転送の遅延.....                      | 28 |
| 表 7.  | DMA メモリ (SRAM) ポートの転送の遅延.....                        | 28 |
| 表 8.  | DMA AHB ポートの方向と転送モード設定.....                          | 33 |
| 表 9.  | コードスニペット .....                                       | 35 |
| 表 10. | 文書改版履歴.....  | 37 |
| 表 11. | 日本語版文書改版履歴.....                                      | 37 |

## 図の一覧

|       |  |    |
|-------|--|----|
| 図 1.  | DMA ブロック図 .....                                  | 7  |
| 図 2.  | チャネル選択 .....                                     | 8  |
| 図 3.  | DMA の転送元アドレスと転送先アドレスのインクリメント .....               | 10 |
| 図 4.  | FIFO の構造 .....                                   | 11 |
| 図 5.  | DMA バースト転送 .....                                 | 12 |
| 図 6.  | ダブルバッファモード .....                                 | 13 |
| 図 7.  | STM32F405/415 と STM32F407/417 のシステムアーキテクチャ ..... | 16 |
| 図 8.  | CPU と DMA1 が SRAM1 へのアクセスをリクエスト .....            | 18 |
| 図 9.  | 5 個のマスタが SRAM アクセスをリクエスト .....                   | 19 |
| 図 10. | 割込みによって発生する CPU 転送が原因の DMA 転送の遅延 .....           | 20 |
| 図 11. | DMA デュアルポート .....                                | 21 |
| 図 12. | ペリフェラルからメモリの転送の状態 .....                          | 22 |
| 図 13. | メモリからペリフェラルの転送の状態 .....                          | 23 |
| 図 14. | DMA リクエストのアービトレーション .....                        | 24 |
| 図 15. | AHB – APB1 ブリッジでの CPU と DMA1 の同時アクセスリクエスト .....  | 25 |
| 図 16. | SPI 全二重 DMA 転送時間 .....                           | 29 |
| 図 17. | メモリからペリフェラル転送モードの DMA .....                      | 34 |

## 1 DMA コントローラの説明

DMA は、AMBA アドバンストハイパフォーマンスバス (AHB) モジュールであり、3 つの AHB ポート (DMA プログラミング用のスレーブポート 1 つと、DMA が異なるスレーブモジュール間のデータ転送を開始できるマスタポート 2 つ (ペリフェラルポートとメモリポート)) を備えています。

DMA では、Cortex-Mx プロセッサが介入することなく、データ転送をバックグラウンドで実行できます。この動作が行われている間、メインプロセッサは他のタスクを実行でき、データブロック全体が処理可能になったときのみ割込みが発生します。

システム性能に大きな影響を与えずに、大量のデータを転送できます。DMA は主に、さまざまなペリフェラルモジュールが使用する中央データバッファストレージ（通常はシステム SRAM）を実装するために使用します。このソリューションは、各ペリフェラルが独自のローカルデータストレージを実装する必要のある分散ソリューションに比べて、シリコンおよび消費電力という点で費用がかかりません。

STM32F2/F4/F7 DMA コントローラは、DMA 転送と CPU の実行 / 割込みイベント検出 / サービスの両方での遅延を非常に小さくするためにマルチレイヤバスシステムを最大限に利用します。

### 1.1 DMA 転送のプロパティ

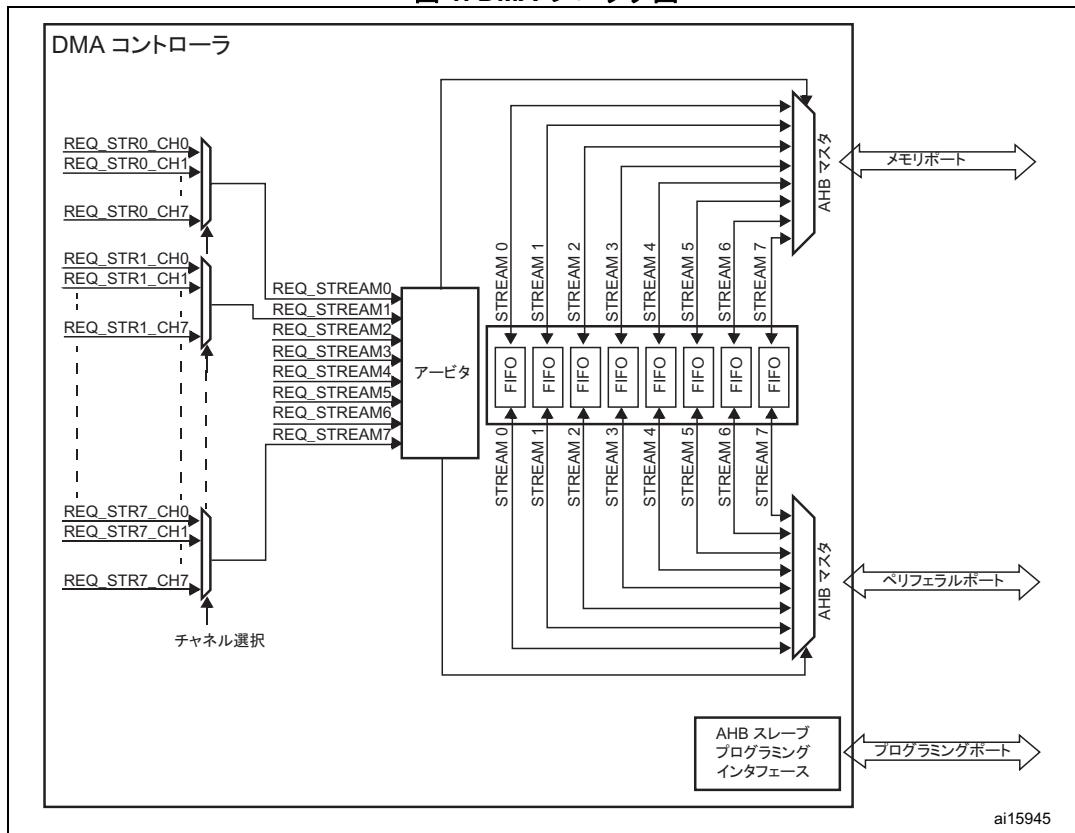
DMA 転送は、次のプロパティによって特徴付けられます。

- DMA ストリーム / チャネル
- ストリームの優先順位
- 転送元 / 転送先アドレス
- 転送モード
- 転送サイズ (DMA がフローコントローラの場合のみ)
- 転送元 / 転送先アドレスのインクリメント / ノンインクリメント
- 転送元 / 転送先のデータ幅
- 転送タイプ
- FIFO モード
- 転送元 / 転送先のバーストサイズ
- ダブルバッファモード
- フロー制御

STM32F2/F4/F7 デバイスは、2 つの DMA コントローラを内蔵しています。各 DMA に 1 つのペリフェラルポートと 1 つのメモリポートの合計 2 つのポートがあり、これらのポートは同時に動作できます。

図 1 に DMA ブロック図を示します。

図 1. DMA ブロック図



以降のサブセクションでは、DMA 転送の各プロパティについて詳しく説明します。

## 1.1.1 DMA ストリーム / チャネル

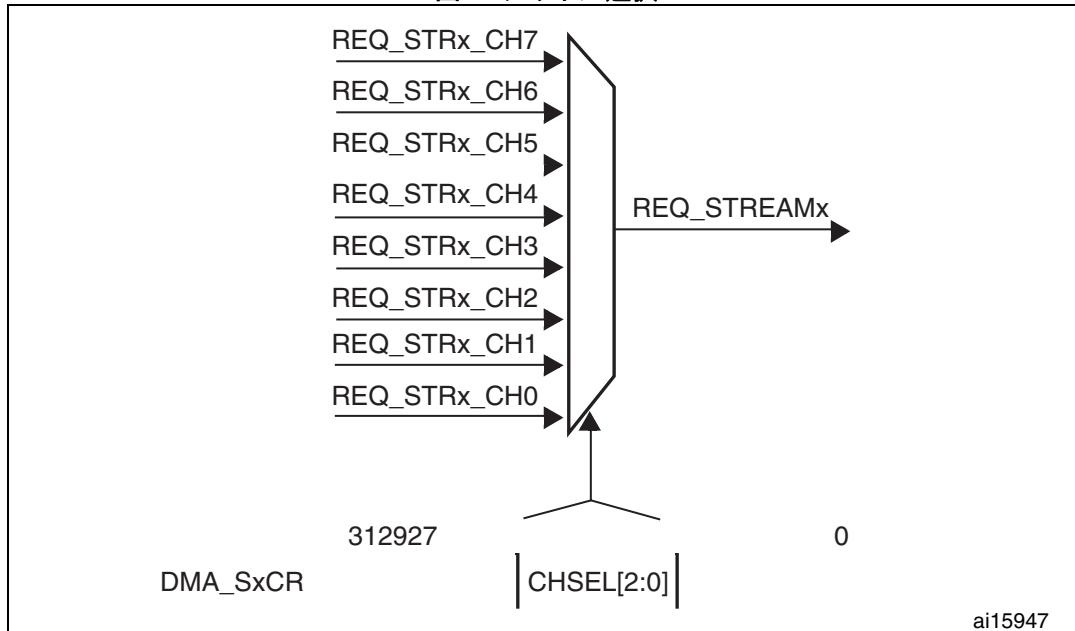
STM32F2/F4/F7 デバイスは 2 つの DMA コントローラを内蔵し、それぞれが 1 つ以上のペリフェラルからのメモリアクセスリクエストの管理に特化した、最大で合計 16 のストリーム（コントローラあたり 8 ストリーム）を提供します。

各ストリームには、最大で合計 8 つの選択可能なチャネル（リクエスト）があります。この選択は、ソフトウェアによって設定可能であり、これによっていくつかのペリフェラルが DMA リクエストを発生することができるようになります。

図 2 は、専用ストリームのチャネル選択を示しています。

# 参考資料

図 2. チャネル選択



注：  
1つのストリームで同時にアクティブになることはできるのは1つのチャネル / リクエストのみです。  
複数の有効な DMA ストリームが同一のペリフェラルリクエストを処理することはできません。

表 1 および 表 2 は、STM32F427/437 および STM32F429/439 の DMA1/DMA2 リクエストマッピングを示しています。これらの表は、使用可能な DMA ストリーム / チャネルの設定とペリフェラルリクエストの対応を示しています。

表 1. STM32F427/437 と STM32F429/439 の DMA1 リクエストマッピング

| ペリフェラル<br>リクエスト | ストリーム<br>0          | ストリーム<br>1            | ストリーム<br>2          | ストリーム<br>3            | ストリーム<br>4            | ストリーム<br>5  | ストリーム<br>6                      | ストリーム<br>7          |
|-----------------|---------------------|-----------------------|---------------------|-----------------------|-----------------------|-------------|---------------------------------|---------------------|
| チャネル 0          | SPI3_RX             | -                     | SPI3_RX             | SPI2_RX               | SPI2_TX               | SPI3_TX     | -                               | SPI3_TX             |
| チャネル 1          | I2C1_RX             | -                     | TIM7_UP             |                       | TIM7_UP               | I2C1_RX     | I2C1_TX                         | I2C1_TX             |
| チャネル 2          | TIM4_CH1            | -                     | I2S3_EXT_RX         | TIM4_CH2              | I2S2_EXT_TX           | I2S3_EXT_TX | TIM4_UP                         | TIM4_CH3            |
| チャネル 3          | I2S3_EXT_RX         | TIM2_UP<br>TIM2_CH3   | I2C3_RX             | I2S2_EXT_RX           | I2C3_TX               | TIM2_CH1    | TIM2_CH2<br>TIM2_UP<br>TIM2_CH4 | TIM2_UP<br>TIM2_CH4 |
| チャネル 4          | UART5_RX            | USART3_RX             | UART4_RX            | USART3_TX             | UART4_TX              | USART2_RX   | USART2_TX                       | UART5_TX            |
| チャネル 5          | UART8_TX            | UART7_TX              | TIM3_CH4<br>TIM3_UP | UART7_RX              | TIM3_CH1<br>TIM3_TRIG | TIM3_CH2    | UART8_RX                        | TIM3_CH3            |
| チャネル 6          | TIM5_CH3<br>TIM5_UP | TIM5_CH4<br>TIM5_TRIG | TIM5_CH1            | TIM5_CH4<br>TIM5_TRIG | TIM5_CH2              | -           | TIM5_UP                         | -                   |
| チャネル 7          | -                   | TIM6_UP               | I2C2_RX             | I2C2_RX               | USART3_TX             | DAC1        | DAC2                            | I2C2_TX             |

# 参考資料

AN4031

DMA コントローラの説明

表 2. STM32F427/437 と STM32F429/439 の DMA2 リクエストマッピング

| ペリフェラル<br>リクエスト | ストリーム<br>0 | ストリーム<br>1 | ストリーム<br>2                       | ストリーム<br>3 | ストリーム<br>4                        | ストリーム<br>5 | ストリーム<br>6                       | ストリーム<br>7                        |
|-----------------|------------|------------|----------------------------------|------------|-----------------------------------|------------|----------------------------------|-----------------------------------|
| チャネル 0          | ADC1       | SAI1_A     | TIM8_CH1<br>TIM8_CH2<br>TIM8_CH3 | SAI1_A     | ADC1                              | SAI1_B     | TIM1_CH1<br>TIM1_CH2<br>TIM1_CH3 | -                                 |
| チャネル 1          | -          | DCMI       | ADC2                             | ADC2       | SAI1_B                            | SPI6_TX    | SPI6_RX                          | DCMI                              |
| チャネル 2          | ADC3       | ADC3       | -                                | SPI5_RX    | SPI5_TX                           | CRYP_OUT   | CRYP_IN                          | HASH_IN                           |
| チャネル 3          | SPI1_RX    | -          | SPI1_RX                          | SPI1_TX    | -                                 | SPI1_TX    | -                                | -                                 |
| チャネル 4          | SPI4_RX    | SPI4_TX    | USART1_RX                        | SDIO       | -                                 | USART1_RX  | SDIO                             | USART1_TX                         |
| チャネル 5          | -          | USART6_RX  | USART6_RX                        | SPI4_RX    | SPI4_TX                           | -          | USART6_TX                        | USART6_TX                         |
| チャネル 6          | TIM1_TRIG  | TIM1_CH1   | TIM1_CH2                         | TIM1_CH1   | TIM1_CH4<br>TIM1_TRIG<br>TIM1_COM | TIM1_UP    | TIM1_CH3                         | -                                 |
| チャネル 7          | -          | TIM8_UP    | TIM8_CH1                         | TIM8_CH2   | TIM8_CH3                          | SPI5_RX    | SPI5_TX                          | TIM8_CH4<br>TIM8_TRIG<br>TIM8_COM |

STM32F2/F4/F7 DMA リクエストマッピングは、ソフトウェアアプリケーションが柔軟に関連付けられているペリフェラルリクエストの各 DMA リクエストをマッピングできるように、またほとんどの使用例が対応する DMA ストリームおよびチャネルの多重化によってカバーされるように設計されています。使用するマイクロコントローラに対応するリファレンスマニュアル（セクション : 参照文献を参照）の DMA1/DMA2 リクエストマッピングの表を参照してください。

## 1.1.2 ストリームの優先順位

各 DMA ポートは、他の DMA ストリーム間の優先順位を操作するためのアービタを内蔵しています。ストリームの優先順位は、ソフトウェアで設定可能です（ソフトウェアレベルは 4 つあります）。2 つ以上の DMA ストリームが同じソフトウェア優先順位レベルである場合、ハードウェア優先順位が使用されます（ストリーム 0 はストリーム 1 より優先され、以下同様となります）。

## 1.1.3 転送元 / 転送先アドレス

DMA 転送は、転送元アドレスと転送先アドレスによって定義されます。転送元と転送先はどちらも AHB または APB のメモリ範囲内とし、転送サイズで整列されている必要があります。

## 1.1.4 転送モード

DMA は、次の 3 つの転送モードを実行できます。

- ペリフェラルからメモリ
- メモリからペリフェラル
- メモリ間（この転送を実行できるのは DMA2 のみです。このモードでは サーキュラモードとダイレクトモードは使用できません）

### 1.1.5 転送サイズ

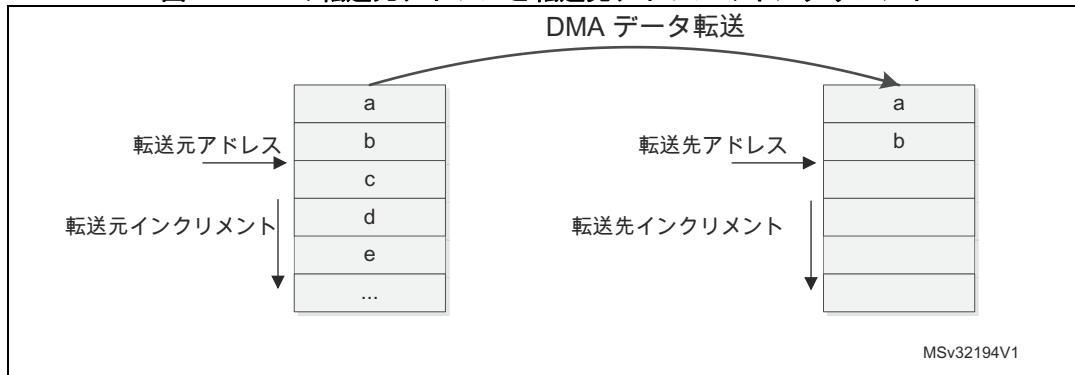
転送サイズ値は、DMA がフローコントローラである場合のみ定義する必要があります。実際には、この値は、転送元から転送先に転送するデータの量を定義します。

転送サイズは、DMA\_SxNDTR レジスタの値とペリフェラル側のデータ幅によって定義されます。受信したリクエスト（バーストまたはシングル）に応じて、転送サイズの値は転送されたデータの量だけ減ります。

### 1.1.6 転送元 / 転送先アドレスのインクリメント

各データ転送の後に、転送元または転送先、あるいはその両方のアドレスを自動的にインクリメントするように DMA を設定できます。

図 3. DMA の転送元アドレスと転送先アドレスのインクリメント



### 1.1.7 転送元 / 転送先のデータ幅

転送元と転送先のデータ幅は、次のいずれかとして定義できます。

- バイト (8 ビット)
- ハーフワード (16 ビット)
- ワード (32 ビット)

### 1.1.8 転送タイプ

- サーキュラモード：サーキュラモードを使用すると、サーキュラバッファや連続したデータフレームを処理できます (DMA\_SxNDTR レジスタには事前にプログラムされた値が自動的に再ロードされます)。
- 通常モード：DMA\_SxNDTR レジスタがゼロに達すると、ストリームは無効になります (その場合、DMA\_SxCR レジスタの EN ビットは 0 に等しくなります)。

# 参考資料

AN4031

DMA コントローラの説明

## 1.1.9 DMA FIFO モード

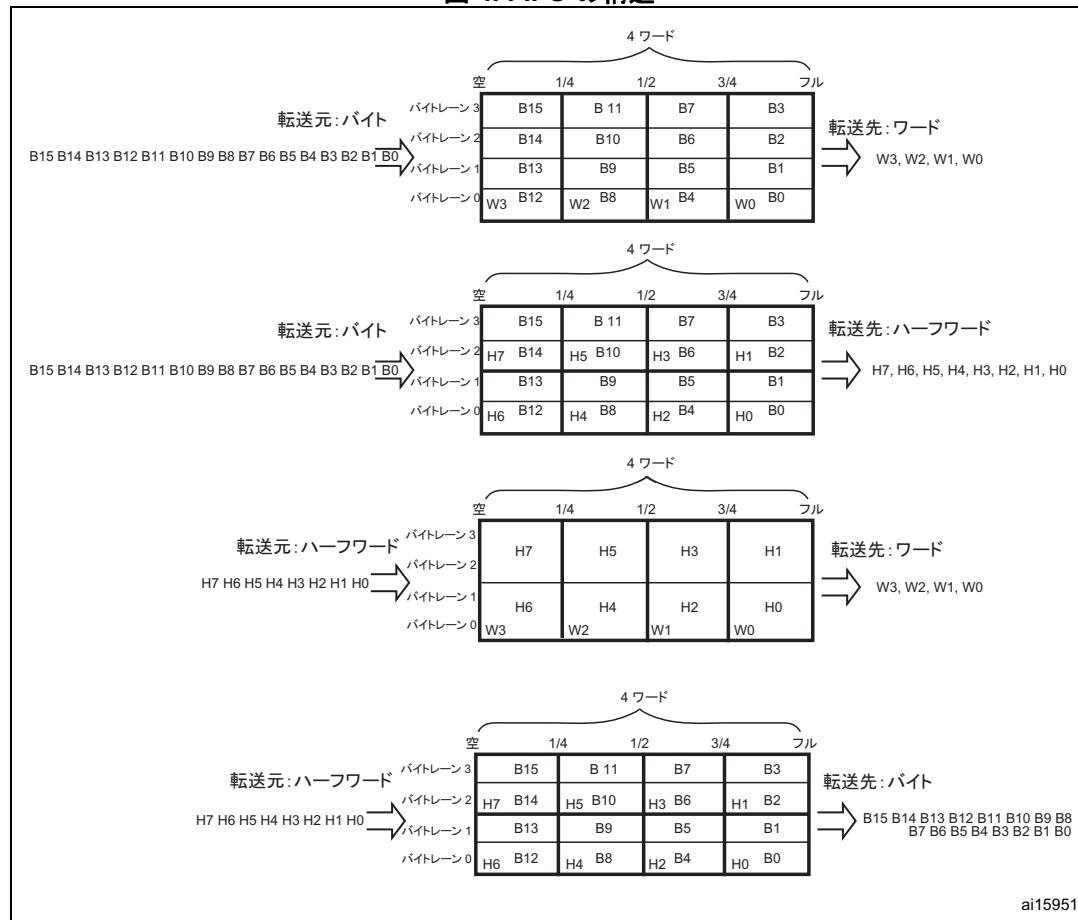
各ストリームには独立した 4 ワード (4 \* 32 ビット) の FIFO があり、閾値レベルはソフトウェアで 1/4、1/2、3/4、またはフルに設定できます。FIFO は転送元からのデータを転送先へ送信する前に一時的に格納するために使用します。

DMA FIFO は、ソフトウェアで有効または無効にできます。無効な場合は、ダイレクトモードが使用されます。DMA FIFO が有効な場合、データのパッキング / アンパッキングまたはバーストモード、あるいはその両方を使用できます。設定されている DMA FIFO 閾値は、DMA メモリポートのリクエスト時間を定義します。

STM32F2/F4/F7 に実装されている DMA FIFO は、次のように役立ちます。

- SRAM アクセスを減らして、同時並行性を増やすために他のマスタがバスマトリックスにアクセスする時間を増やします。
- ソフトウェアが、転送帯域幅を最適化するバーストトランザクションを実行できます。
- データをパッキング / アンパッキングして、余分な DMA アクセスを行わずに、転送元と転送先のデータ幅を適応させることができます。

図 4. FIFO の構造



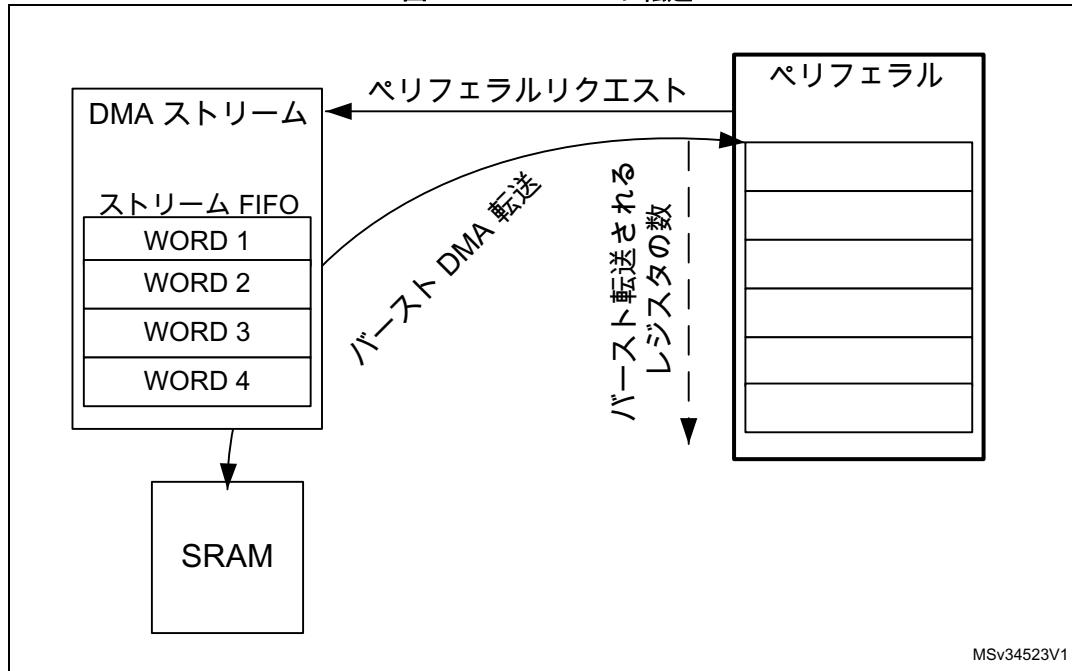
ai15951



### 1.1.10 転送元 / 転送先のバーストサイズ

バースト転送は、実装されている DMA FIFO によって保証されます。

図 5. DMA バースト転送



MSv34523V1

ペリフェラルからのバーストリクエストに応じて、DMA はバーストサイズ（4/8/16 × データユニット）でプログラムされているデータユニット数（データユニットはワード、ハーフワード、またはバイト）の読み出し / 書込みを行います。DMA のペリフェラルポートのバーストサイズは、ペリフェラルのニーズ / 能力に従ってセットする必要があります。

メモリポートの DMA バーストサイズと FIFO 閾値設定は一致する必要があります。これにより、メモリポートでバースト転送が開始されるときに、DMA ストリームの FIFO に十分なデータを用意できます。[表 3](#) に、メモリバーストサイズ、FIFO 閾値設定、およびデータサイズの実現可能な組み合わせを示します。

データの一貫性を確保するため、バーストを形成する転送の各グループは分割できません。AHB 転送はロックされ、AHB バスマトリックスのアビタは、バースト転送シーケンス中の DMA マスタのアクセス権を削除しません。

# 参考資料

表 3. 実現可能な バースト設定

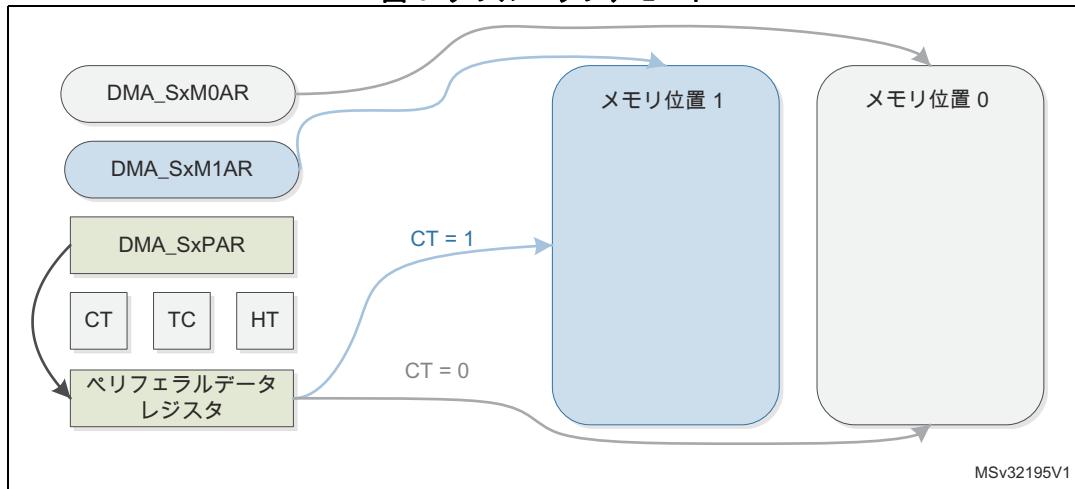
| MSIZE  | FIFO レベル | MBURST = INCR4   | MBURST = INCR8   | MBURST = INCR16 |
|--------|----------|------------------|------------------|-----------------|
| バイト    | 1/4      | 4 バイトの 1 バースト    | 設定禁止             | 設定禁止            |
|        | 1/2      | 4 バイトの 2 バースト    | 8 バイトの 1 バースト    |                 |
|        | 3/4      | 4 バイトの 3 バースト    | 設定禁止             |                 |
|        | フル       | 4 バイトの 4 バースト    | 8 バイトの 2 バースト    | 16 バイトの 1 バースト  |
| ハーフワード | 1/4      | 設定禁止             | 設定禁止             | 設定禁止            |
|        | 1/2      | 4 ハーフワードの 1 バースト |                  |                 |
|        | 3/4      | 設定禁止             |                  |                 |
|        | フル       | 4 ハーフワードの 2 バースト | 8 ハーフワードの 1 バースト |                 |
| ワード    | 1/4      | 設定禁止             | 設定禁止             | 設定禁止            |
|        | 1/2      |                  |                  |                 |
|        | 3/4      |                  |                  |                 |
|        | フル       | 4 ワードの 1 バースト    |                  |                 |

## 1.1.11 ダブルバッファモード

ダブルバッファストリームは通常の（シングルバッファ）ストリームとして動作しますが、2つのメモリポインタを持っているところが異なります。ダブルバッファモードが有効になっていると、サイキュラモードが自動的に有効になり、トランザクションが終わる（DMA\_SxNDTR レジスタが 0 に達する）たびにメモリポインタがスワップされます。

これにより、DMA 転送によってある 1 つのメモリ領域が書き込み/使用中である間に、ソフトウェアは別のメモリ領域を処理することができます。

図 6. ダブルバッファモード



ダブルバッファモードでは、ストリームが有効なとき、いつでも AHB メモリポートのベースアドレス (DMA\_SxM0AR または DMA\_SxM1AR) を更新することができます。

- DMA\_SxCR レジスタの CT (現在のターゲット) ビットが 0 に等しい場合、現在の DMA メモリターゲットはメモリ位置 0 であり、したがってベースアドレスのメモリ位置 1 (DMA\_SxM1AR) は更新可能です。
- DMA\_SxCR レジスタの CT ビットが 1 に等しい場合、現在の DMA メモリターゲットはメモリ位置 1 であり、したがってベースアドレスのメモリ位置 0 (DMA\_SxM0AR) は更新可能です。

### 1.1.12 フロー制御

フローコントローラは、データ転送長を制御するユニットであり、DMA 転送の停止を処理します。

フローコントローラは DMA またはペリフェラルのどちらかとなります。

- DMA がフローコントローラの場合 :

この場合、関連付けられている DMA ストリームを有効にする前に、DMA\_SxNDTR レジスタに転送サイズ値を定義する必要があります。DMA リクエストを処理する際、転送サイズ値は、(リクエストのタイプがバーストまたはシングルのどちらであるかに応じて) 転送されたデータの量だけ減少します。

転送サイズ値が 0 に達すると、DMA 転送は完了し、DMA ストリームは無効になります。

- ペリフェラルがフローコントローラの場合 :

この場合、転送されるデータ項目数が不明です。最後のデータが転送されるとき、ペリフェラルがハードウェアによって DMA コントローラに知らせます。SD/MMC と JPEG の各ペリフェラルだけがこのモードをサポートします。

## 1.2 DMA 転送の設定

DMA ストリーム x (x はストリーム番号) を設定するには、次の手順を実行する必要があります。

1. ストリームが有効になっている場合、DMA\_SxCR レジスタの EN ビットをリセットすることによって無効にし、その後、進行中のストリーム動作がないことを確認するため、このビットを読み出します。このビットに 0 を書き込んでもすぐに効果はありません。それは現在行われているすべての転送が完了した後、実際に 0 が書き込まれるからです。EN ビットが 0 として読み出される場合、ストリームが設定可能な状態であることを意味します。したがって、ストリームの設定を開始する前に EN ビットがクリアされるのを待つ必要があります。以前のデータブロックの DMA 転送からステータスレジスタ (DMA\_LISR と DMA\_HISR) に設定されたすべてのストリーム専用ビットは、ストリームを再度有効にする前にクリアする必要があります。
2. DMA\_SxPAR レジスタにペリフェラルポートレジスタアドレスを設定します。ペリフェラルイベントの後、データはこのアドレスとペリフェラルポートの間で転送されます。
3. DMA\_SxMA0R レジスタ (およびダブルバッファモードの場合は DMA\_SxMA1R レジスタ) にメモリアドレスを設定します。ペリフェラルイベント後は、このメモリに対してデータを読み書きします。
4. 転送されるデータ項目の合計数を DMA\_SxNDTR レジスタで設定します。各ペリフェラルイベントまたはバーストの各ビートの後、この値はデクリメントされます。
5. DMA\_SxCR レジスタの CHSEL[2:0] を使用して DMA チャネル (リクエスト) を選択します。
6. ペリフェラルをフローコントローラとして使用することを意図し、かつこの機能をサポートしている場合、DMA\_SxCR レジスタの PFCTRL ビットをセットします。
7. DMA\_SxCR レジスタの PL[1:0] ビットを使用して、ストリームの優先順位を設定します。
8. FIFO の使用方法を設定します (有効または無効にする、送信と受信における閾値)。

# 参考資料

9. DMA\_SxCR レジスタで、データ転送方向、ペリフェラルとメモリのインクリメントモード / 固定モード、シングルトランザクションまたはバーストトランザクション、ペリフェラルとメモリのデータ幅、サーキュラモード、ダブルバッファモード、ハーフ転送やフル転送後の割り込み、およびエラーを設定します。
10. DMA\_SxCR レジスタの EN ビットをセットすることにより、ストリームを有効にします。ストリームが有効になると、そのストリームに接続されているペリフェラルからの DMA リクエストを処理できます。

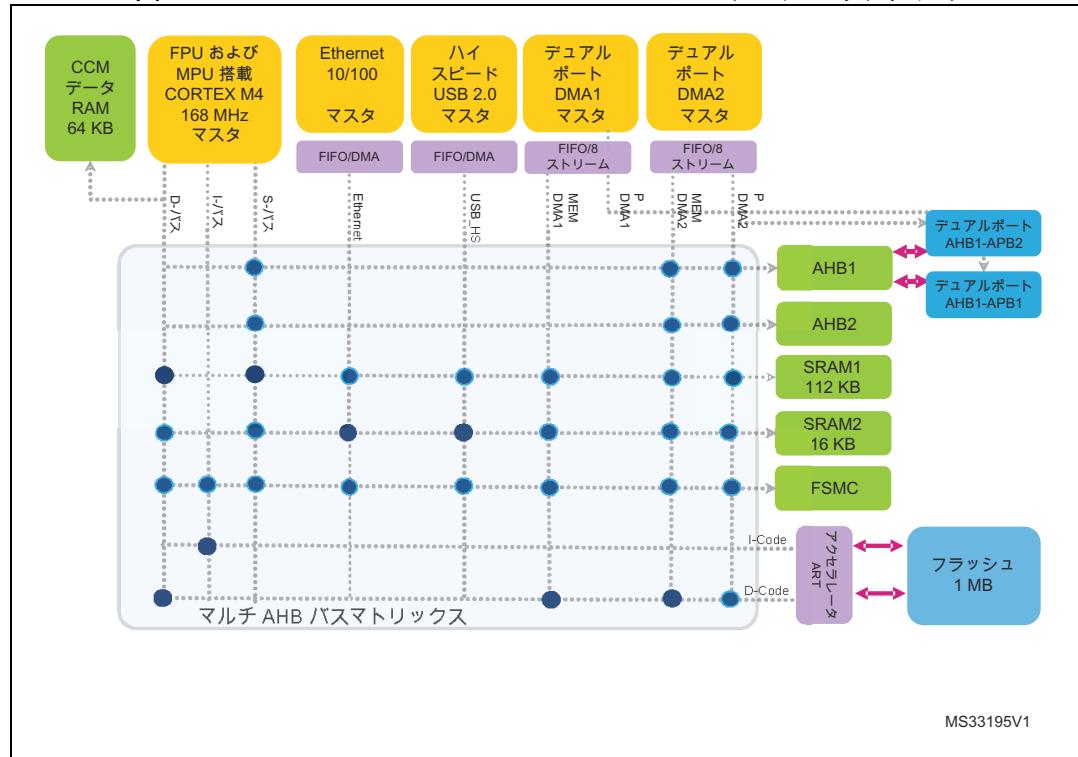
## 2 システム性能に関する考慮事項

STM32F2/F4/F7 デバイスは、マルチマスタ / マルチスレーブアーキテクチャを組み込んでいます。

- 複数のマスター :
  - Cortex®-Mx コアの AHB バス
  - DMA1 メモリバス
  - DMA2 メモリバス
  - DMA2 ペリフェラルバス
  - Ethernet DMA バス
  - USB ハイスピード DMA バス
  - Chrom-ART アクセラレータバス
  - LCD-TFT バス
- 複数スレーブ :
  - マルチレイヤバスマトリックスに接続されている内蔵フラッシュインターフェース
  - メイン内蔵 SRAM1 と補助内蔵 SRAM (SRAM2、SRAM3 : デバイスで利用可能な場合)
  - AHB-APB ブリッジおよび APB ペリフェラルを含む AHB1 ペリフェラル
  - AHB2 ペリフェラル
  - AHB3 ペリフェラル (FMC、Quad-SPI ペリフェラル : 製品ファミリで利用可能な場合)

マスターとスレーブはマルチレイヤバスマトリックス経由で接続され、複数のハイスピードペリフェラルが同時動作中であっても、同時アクセスと効率的な動作が保証されます。次の図に、STM32F405/415 ファミリと STM32F407/417 ファミリの場合のこのアーキテクチャを示します。

図 7. STM32F405/415 と STM32F407/417 のシステムアーキテクチャ



## 2.1 マルチレイヤバスマトリックス

マスタは、マルチレイヤバスマトリックスによって、異なるスレーブモジュールをアドレス指定している限り、データ転送を同時に実行できます。Cortex-Mx アーキテクチャとデュアル AHB ポート DMA に加え、この構造によってデータ転送の並列性が強化されたことにより、実行時間が短縮され、DMA の効率と消費電力が最適化されています。

### 2.1.1 定義

- AHB マスタ：バスマスタは、読み出しおよび書き込み動作を開始できます。定義されている期間にバスの所有権を獲得できるのは 1 つのマスタだけです。
- AHB スレーブ：バススレーブは、マスタの読み出しありは書き込み動作に対応します。バススレーブは、成功、失敗、またはウェイトのいずれかの状態を表す信号をマスタに返します。
- AHB アービタ：バスアービタは、一度に確実に 1 つのマスタだけが読み出し動作または書き込み動作を開始できるようにします。
- AHB バスマトリックス：各レイヤの専用 AHB アービタを使用して AHB マスタと AHB スレーブを相互接続するマルチレイヤ AHB バスマトリックス。アービトレーションには、ラウンドロビンアルゴリズムを使用します。

### 2.1.2 ラウンドロビンの優先順位方式

各マスタが任意のスレーブに非常に小さな遅延でアクセスできるようにするために、ラウンドロビン優先順位方式がバスマトリックスレベルで実装されています。

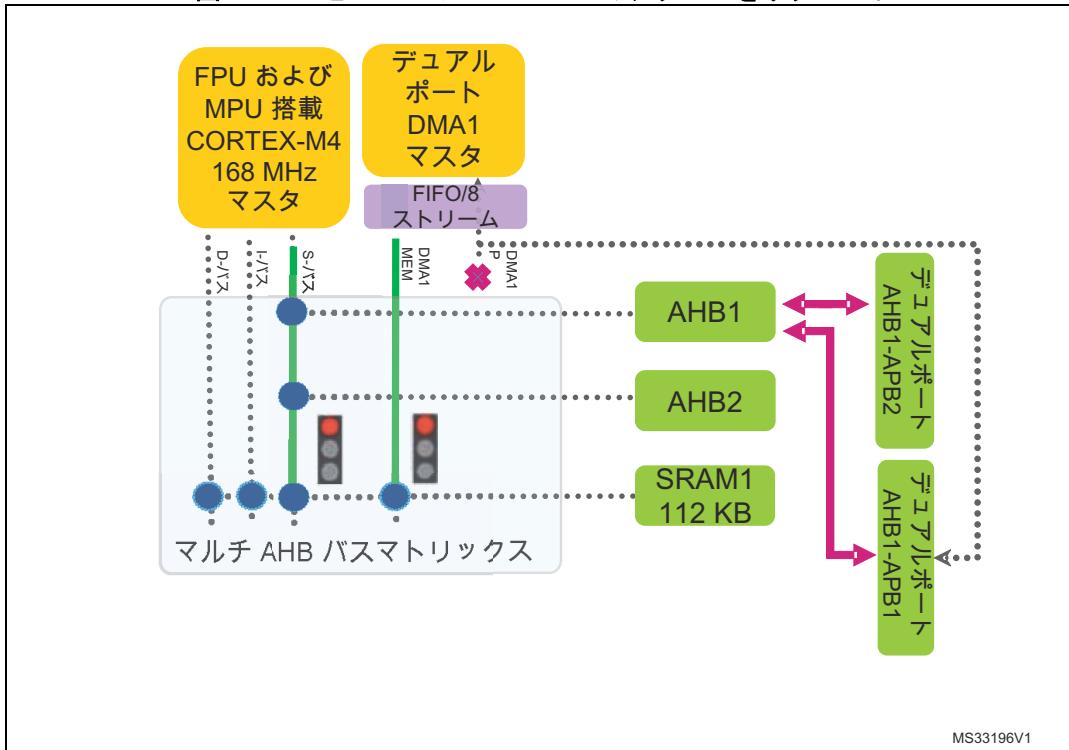
- ラウンドロビンアービトレーションポリシーによって、バス帯域幅を公平に分配できます。
- 最大遅延が制限されています。
- ラウンドロビンのクォンタムは 1 回の転送です。

複数の AHB マスタが同一の AHB スレーブに同時にアクセスしようとしたときは、バスマトリックスアービタが介入してアクセス競合を解決します。

次の例（図 8）では、CPU と DMA1 の両方がデータを読み出すために SRAM1 にアクセスしようとしています。

# 参考資料

図 8. CPU と DMA1 が SRAM1 へのアクセスをリクエスト

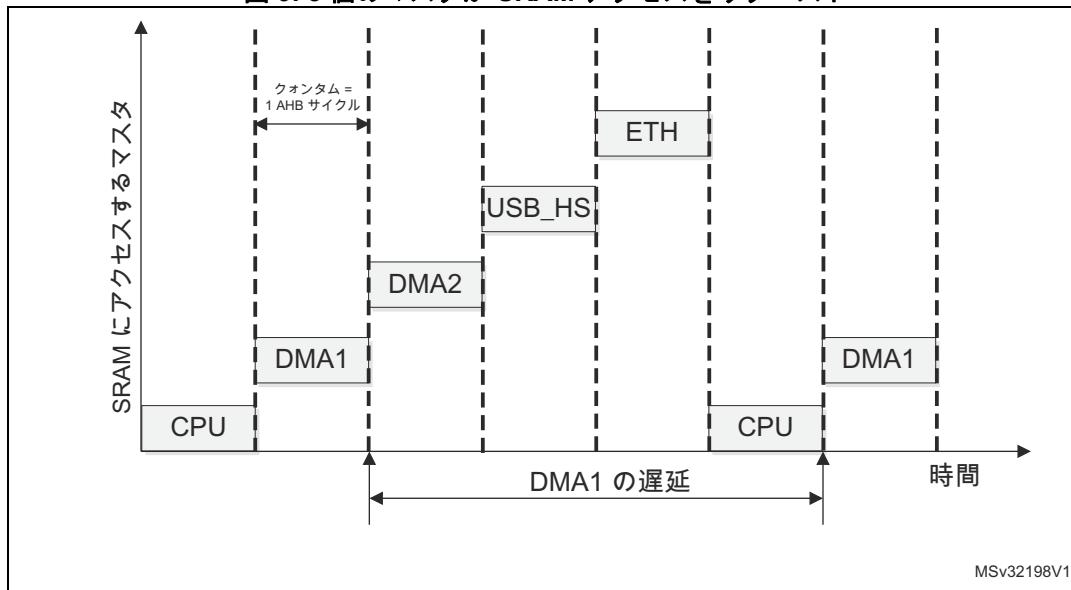


MS33196V1

上記の例のように、バスアクセスが同時に発生した場合は、バスマトリックスアービトレーションが必要です。この場合、問題を解決するためにラウンドロビンポリシーが適用されます。すなわち、バスを獲得した最後のマスターが CPU だった場合は、次のアクセスで DMA1 がバスを獲得して、先に SRAM1 にアクセスします。その後に、CPU が SRAM1 にアクセスする権利を獲得します。

これは、特定のマスターにおける転送の遅延は、保留されている他のマスターの同一の AHB スレーブへのアクセスリクエストの数に依存することを示しています。次の例（図 9）では、5 つのマスターが同時に SRAM1 にアクセスしようとしています。

図 9. 5 個のマスタが SRAM アクセスをリクエスト



DMA1 がバスマトリックスを再び獲得して（たとえば）SRAM1 にアクセスするまでの遅延は、他のマスタによるすべての保留中のリクエストの実行時間に等しくなります。

## 2.1.3 バスマトリックスアービトレーションと DMA 転送の遅延の最悪のケース

1 回のトランザクションで DMA マスターがポートで生じる遅延は、他のマスターの転送のタイプと長さに応じて決まります。

たとえば、前述の DMA1 と CPU が同時に SRAM にアクセスする例（図 8）では、DMA 転送の遅延は、CPU のトランザクションの長さに応じて変わります。

バスアクセスが先に CPU に付与されて、CPU が単一の読み書きを実行していない場合、DMA が SRAM へのアクセスを獲得するまでのウェイト時間は、単一のデータ読み書きを実行する 1 AHB サイクルから N AHB サイクル（N は CPU トランザクションのデータワード数）まで増える可能性があります。

CPU は AHB バスをロックして所有権を維持し、多重読み書き動作と割込み入力で発生する遅延を減らします。これにより、ファームウェアの応答性は高くなりますが、DMA トランザクションに遅延が発生する可能性があります。

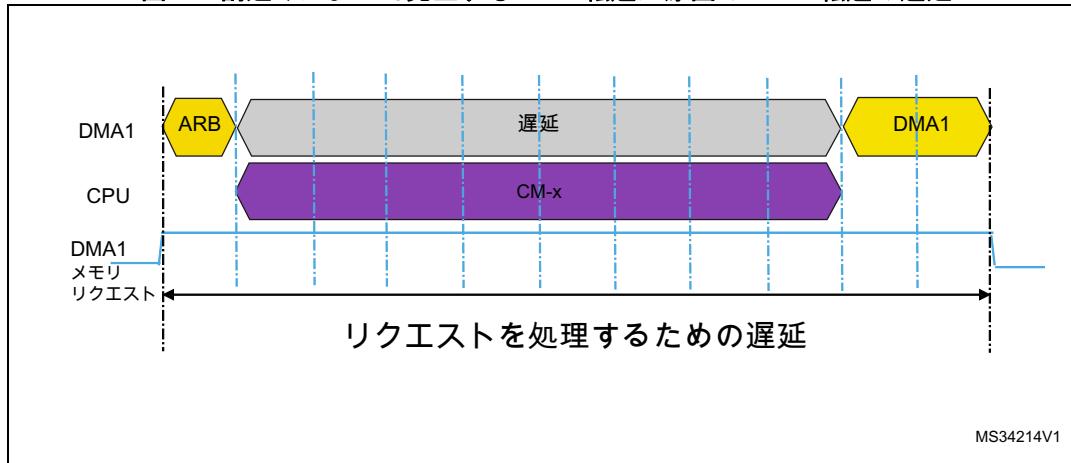
DMA1 による SRAM アクセスが CPU と同時実行される場合の遅延は、転送のタイプによって決まります。

- 割込みによって発生する CPU 転送（コンテキストの保存）: 8 AHB サイクル
- キャッシュコントローラによって実行される CPU 転送（256 ビットキャッシュラインファイル / エビクション）: 8 AHB サイクル<sup>(a)</sup>
- LDM/STM 命令によって発生する CPU 転送: 14 AHB サイクル<sup>(b)</sup>
  - 最大 14 個のレジスタとメモリの間の転送

a. STM32F7xx デバイスのみ

b. LDM/STM 命令によって発生する転送が原因の遅延は、多重読み書き命令を分割して単一読み書き命令にするようにコンパイラを設定することで減らすことができます。

図 10. 割込みによって発生する CPU 転送が原因の DMA 転送の遅延



上図は、割込み入力による CPU のマルチサイクル転送が原因で遅延する DMA 転送の例について詳しく示しています。DMA メモリポートが、メモリアクセスを実行するようにトリガれます。アビトレーションの後、AHB バスは DMA1 メモリポートではなく CPU に付与されます。DMA リクエストを処理するための追加遅延が見られます。この遅延は、割込みによって発生する CPU 転送の 8 AHB サイクルです。

他のマスタ（DMA2、USB\_HS、Ethernet など）でも、トランザクション長が 1 データユニットではない同一のスレーブを同時に処理した場合、同じ挙動が見られます。

バスマトリックスでの DMA アクセス性能を向上させるには、バス競合を避けることを推奨します。

## 2.2 DMA 転送バス

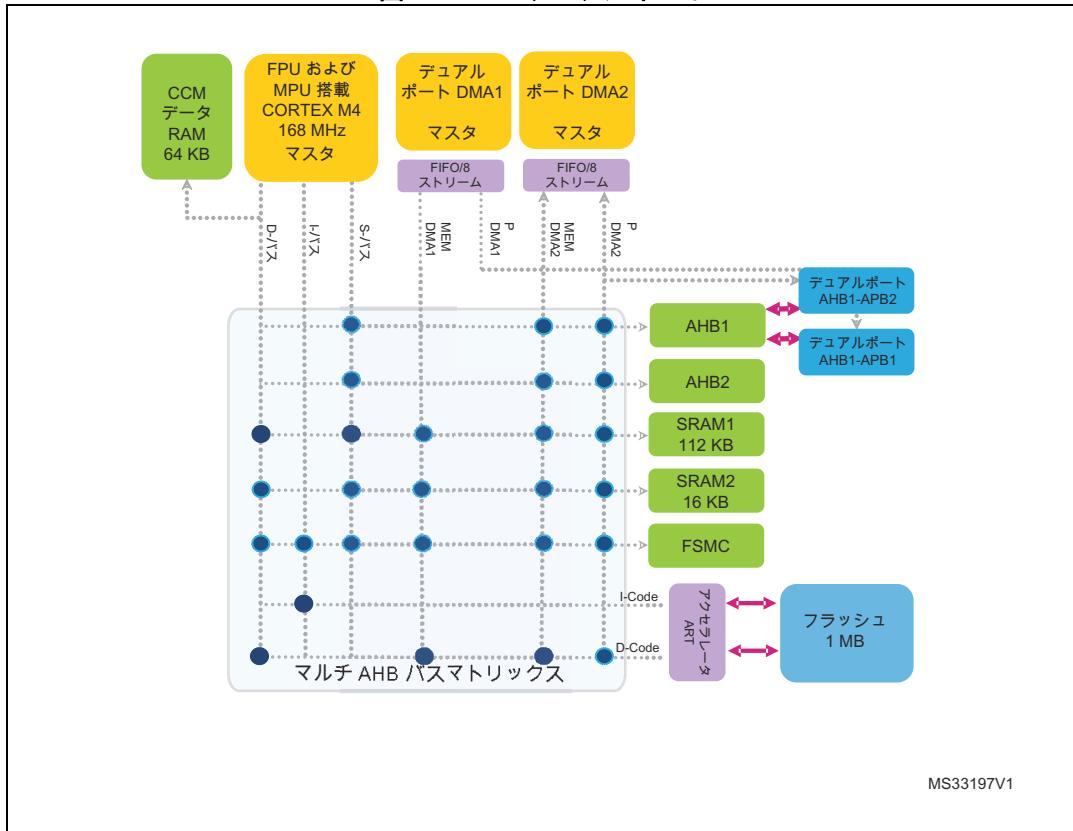
### 2.2.1 デュアル DMA ポート

STM32F2/F4/F7 デバイスは、2 つの DMA を内蔵しています。各 DMA には、メモリポートとペリフェラルポートの 2 つのポートがあり、これらは、DMA レベルだけでなく、外部バスマトリックスおよび専用の DMA パスを使用して他のシステムマスタと同時に動作することもできます。

この同時動作によって、DMA 効率が最適化され、応答時間（リクエストからデータ転送までのウェイト時間）が短縮されます。

# 参考資料

図 11. DMA デュアルポート



DMA2 の場合 :

- MEM (メモリポート) : バスマトリックス経由で AHB1、AHB2、AHB3 (外部メモリコントローラ FSMC)、SRAM、およびフラッシュメモリにアクセスできます。
- Periph (ペリフェラルポート) :
  - バスマトリックス経由で AHB1、AHB2、AHB3 (外部メモリコントローラ FSMC)、SRAM、およびフラッシュメモリにアクセスできます。
  - ダイレクトパス経由 (バスマトリックスを通過しない) で AHB-APB2 ブリッジにアクセスできます。

DMA1 の場合 :

- MEM (メモリポート) : バスマトリックス経由で AHB3 (外部メモリコントローラ FSMC)、SRAM、およびフラッシュメモリにアクセスできます。
- Periph (ペリフェラルポート) : ダイレクトパス経由 (バスマトリックスを通過しない) で AHB-APB1 ブリッジにのみアクセスできます。

### 2.2.2 DMA 転送状態

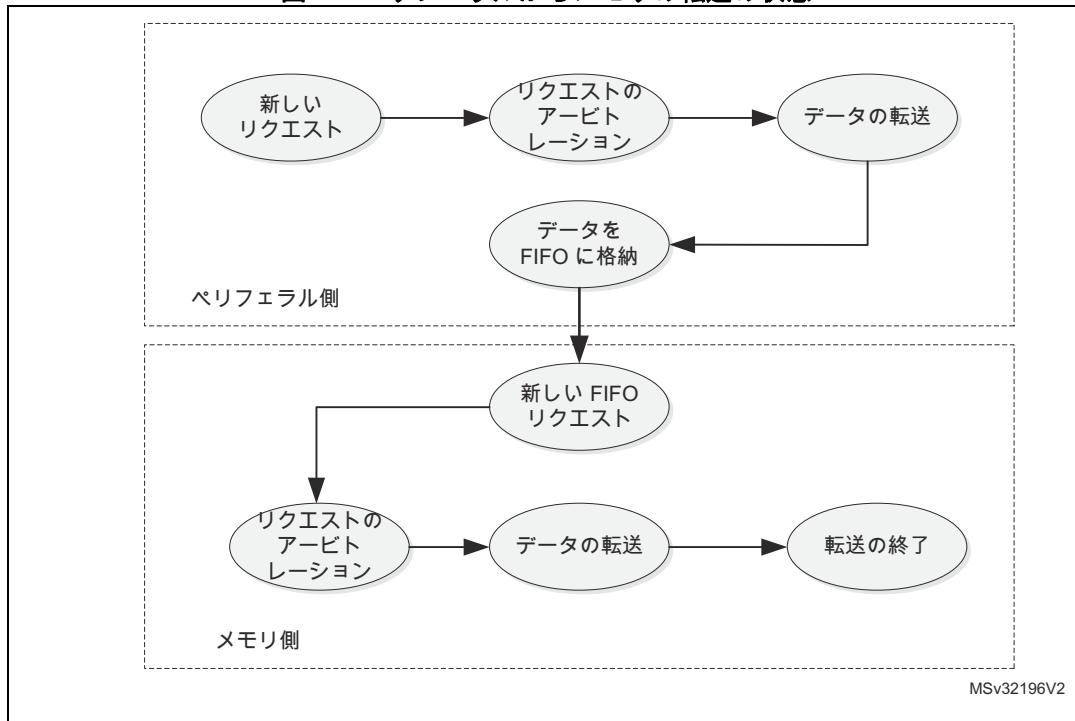
このセクションでは、ペリフェラルポートレベルとメモリポートレベルでの DMA 転送ステップについて説明します。

- ペリフェラルからメモリの転送の場合 :

この転送モードでは、DMA は転送を実行するために 2 つのバスアクセスを必要とします。

- ペリフェラルのリクエストによってトリガされるペリフェラルポート経由のアクセス。
- メモリポート経由のアクセス。FIFO 閾値 (FIFO モード使用時) によるトリガまたはペリフェラル読出し (ダイレクトモード使用時) の直後のトリガのどちらかが可能です。

図 12. ペリフェラルからメモリの転送の状態

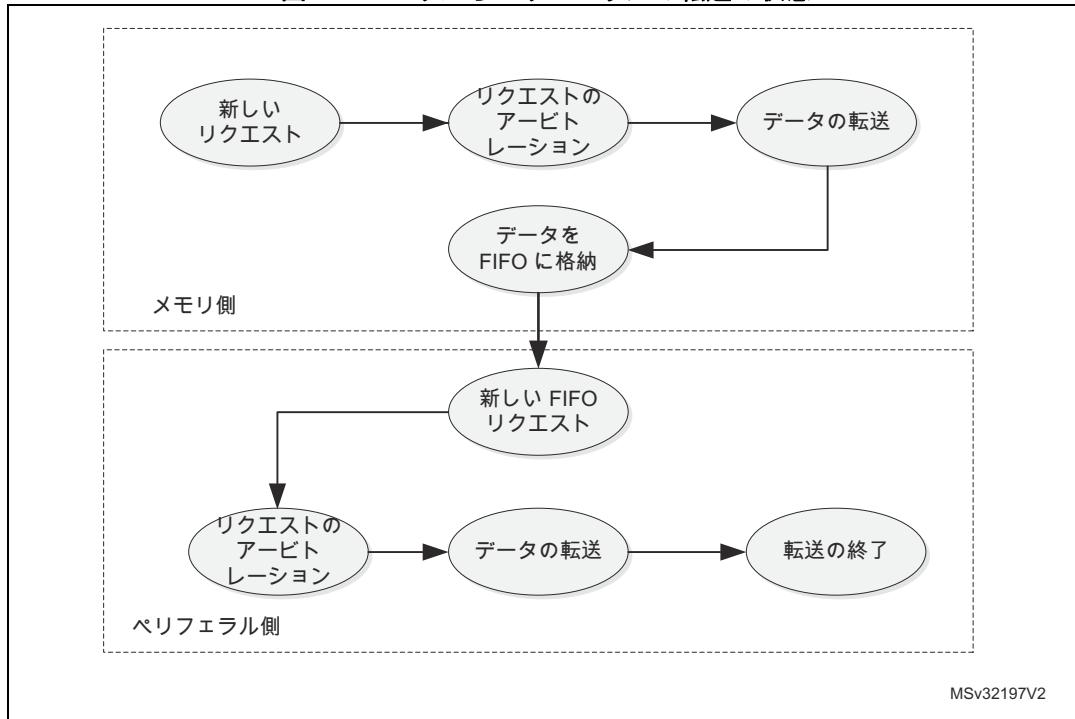


- メモリからペリフェラルの転送の場合 :

この転送モードでは、DMA は転送を実行するために 2 つのバスアクセスを必要とします。

- DMA は、ペリフェラルのアクセスを予測して、メモリからデータを読み出し、FIFO に格納することで、DMA ペリフェラルリクエストがトリガされるとすぐにデータ転送が行われます。
- ペリフェラルリクエストがトリガされると、DMA ペリフェラルポート上で転送が生成されます。

図 13. メモリからペリフェラルの転送の状態



## 2.2.3 DMA リクエストのアービットレーション

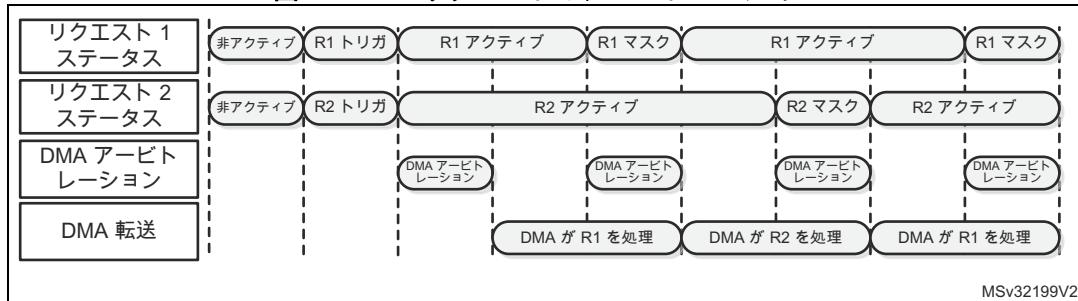
セクション 1.1.2 : ストリームの優先順位で説明しているように、STM32F2/F4/F7 DMA が内蔵するアービタは、2 つの AHB マスタポート（メモリポートおよびペリフェラルポート）のそれぞれについて、その優先順位に基づいて 8 つの DMA ストリームリクエストを管理し、ペリフェラル / メモリアクセスシーケンスを起動します。

複数の DMA リクエストがアクティブな場合、DMA はアクティブなリクエスト間で内部的にアービットレーションを実行し、最初に処理するリクエストを決定します。

次の図は、DMA ストリーム「リクエスト 1」と DMA ストリーム「リクエスト 2」によって同時にトリガされる 2 つのサーキュラ DMA リクエストを示します（リクエスト 1 とリクエスト 2 は任意の DMA ペリフェラルリクエスト）。次の AHB クロックサイクルで、DMA アービタはアクティブな保留中のリクエストをチェックして、最も高い優先順位を持つ「リクエスト 1」ストリームにアクセス権を付与します。

次のアービットレーションサイクルは、「リクエスト 1」ストリームの最後のデータサイクル中に発生します。そのとき、「リクエスト 1」はマスクされていて、アービタからは「リクエスト 2」だけがアクティブに見えます。したがって、このときはアクセスは「リクエスト 2」に予約され、以下同様です。

図 14. DMA リクエストのアービトレーション



一般的な推奨事項 :

- ハイスピード / 広帯域ペリフェラルには、最も高い DMA 優先順位を設定する必要があります。これにより、それらのペリフェラルで最大データ遅延が順守され、オーバーラン / アンダーラン状態が回避されます。
- 帯域幅要件が等しい場合、スレーブモード（データ転送速度の制御なし）で動作するペリフェラルには、マスタモード（データフローを制御可能）で動作するペリフェラルに比べて高い優先順位を割り当てる 것을 推奨します。
- マルチレイヤバスマトリックス構造に基づいて 2 つの DMA が並列に動作できるので、可能であれば、ハイスピードペリフェラルのリクエストを 2 つの DMA の間で調整できます。

## 2.3 AHB—APB ブリッジ

STM32F2/F4/F7 デバイスは、APB1 と APB2 の 2 つの AHB—APB ブリッジを内蔵しており、それらにペリフェラルが接続されます。

### 2.3.1 デュアル AHB—APB ポート

AHB—APB ブリッジは、次の 2 つの異なるパスでアクセスできるデュアルポートアーキテクチャです。

- ダイレクトパス（バスマトリックスを通過しない）。DMA1 から APB1 に、または DMA2 から APB2 に生成できます。このパスでアクセスする場合、バスマトリックスアービタによってペナルティを科せられることはできません。
- 共通パス（バスマトリックスを通過する）。CPU から、または DMA2 から生成できます。バスを獲得するために、バスマトリックスアービトレーションが必要です。

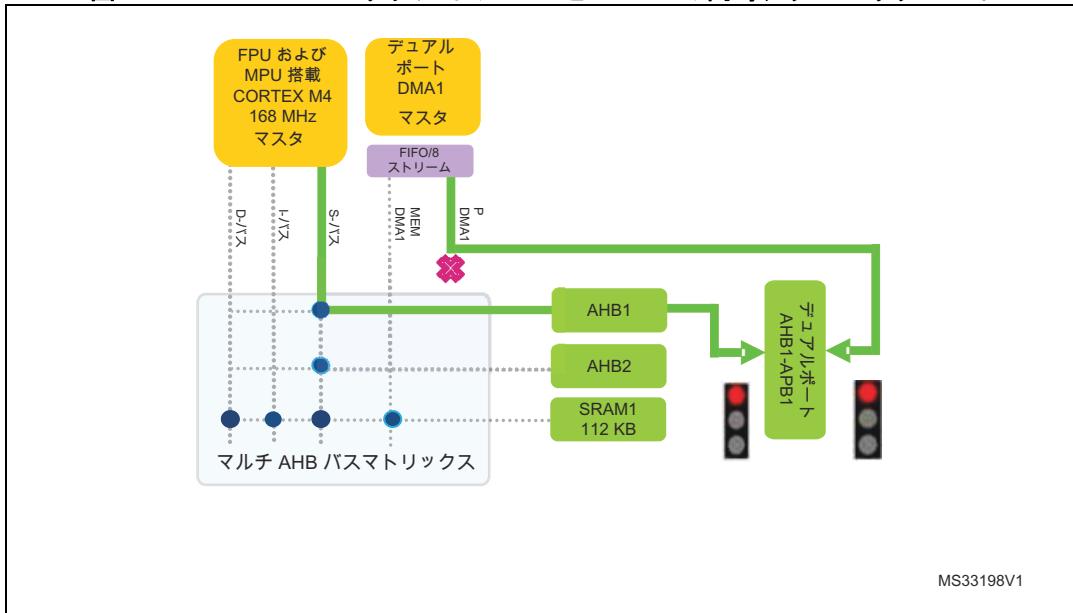
### 2.3.2 AHB—APB ブリッジアービトレーション

上記の製品には DMA のダイレクトパスが実装されているので、同時アクセスリクエストを解決するために、AHB—APB ブリッジレベルでアービタが実装されています。

次の図は、CPU（バスマトリックス経由でアクセス）と DMA1（ダイレクトパス経由でアクセス）によって AHB—APB1 ブリッジで生成される同時アクセスリクエストを示しています。

# 参考資料

図 15. AHB-APB1 ブリッジでの CPU と DMA1 の同時アクセスリクエスト



バスアクセスを付与するために、AHB-APB ブリッジは、ラウンドロビンポリシーを適用します。

- ラウンドロビンのクォンタムは 1 回の APB 転送です。
- DMA ペリフェラルポートの最大遅延は制限されています（1 回の APB 転送）。

APB1 バスと APB2 バスへの同時アクセスが発生する可能性があるのは、CPU と DMA のみです。

- APB1 では、CPU、DMA1、または DMA2、あるいはそれらの組み合わせが同時アクセスをリクエストした場合に同時アクセスが発生する可能性があります。
- APB2 では、CPU と DMA2 が同時アクセスをリクエストした場合に同時アクセスが発生する可能性があります。

## 3 DMA 遅延の予測方法

マイクロコントローラベースのファームウェアアプリケーションを設計する場合、ユーザは、アンダーラン / オーバーランが発生しないようにする必要があります。そのためには、内部システムがアプリケーションに必要な総データ帯域幅を維持できるかどうかをチェックするために、各転送の正確な DMA 遅延を知る必要があります。

### 3.1 DMA 転送時間

#### 3.1.1 デフォルトの DMA 転送時間

セクション 2.2.2 で説明しているように、ペリフェラルからメモリへの DMA 転送を実行するには、次の 2 つのバスアクセスが必要です。

- ペリフェラルのリクエストによってトリガされるペリフェラルポート経由のアクセス。次の処理が必要です。
  - DMA ペリフェラルポートのリクエストのアビトレーション
  - ペリフェラルアドレスの計算
  - ペリフェラルから DMA FIFO (DMA 転送元) へのデータ読出し
- メモリポート経由のアクセス。FIFO 閾値 (FIFO モード使用時) によるトリガまたはペリフェラル読出し (ダイレクトモード使用時) の直後のトリガのどちらかが可能です。次の処理が必要です。
  - DMA メモリポートのリクエストのアビトレーション
  - メモリアドレスの計算
  - ロードされたデータの SRAM (DMA 転送先) への書き込み

メモリからペリフェラルにデータを転送する場合、セクション 2.2.2 で説明しているように、次の 2 つのアクセスも必要です。

- 1 つめのアクセス : DMA は、ペリフェラルのアクセスを予測して、メモリからデータを読み出し、FIFO に格納することで、DMA ペリフェラルリクエストがトリガされるとすぐにデータ転送が行われます。この動作には、次の処理が必要です。
  - DMA メモリポートのリクエストのアビトレーション
  - メモリアドレスの計算
  - メモリから DMA FIFO (DMA 転送元) へのデータ読出し
- 2 つめのアクセス : ペリフェラルリクエストがトリガされると、DMA ペリフェラルポート上で転送が生成されます。この動作には、次の処理が必要です。
  - DMA ペリフェラルポートのリクエストのアビトレーション
  - ペリフェラルアドレスの計算
  - ロードされたデータのペリフェラルアドレス (DMA 転送先) への書き込み

原則として、DMA ストリーム  $T_S$  による総転送時間は、次の式に等しくなります。

$$T_S = T_{SP} \text{ (ペリフェラルアクセス / 転送時間)} + T_{SM} \text{ (メモリアクセス / 転送時間)}$$

ここで :

$$T_{SP} \text{ は、DMA ペリフェラルポートアクセスと転送の合計時間であり、次の式に等しくなります。}$$
$$T_{SP} = t_{PA} + t_{PAC} + t_{BMA} + t_{EDT} + t_{BS}$$

# 参考資料

ここで、

表 4. ペリフェラルポートアクセス / 転送時間と使用する DMA パス

| 説明   | バスマトリックス経由                    |                | DMA の<br>ダイレクトパス |
|--|-------------------------------|----------------|------------------|
|  | AHB<br>ペリフェラルへ                | APB<br>ペリフェラルへ |                  |
| $t_{PA}$ : DMA ペリフェラルポートのアービトレーション                           | 1 AHB サイクル                    | 1 AHB サイクル     | 1 AHB サイクル       |
| $t_{PAC}$ : ペリフェラルアドレスの計算                                    | 1 AHB サイクル                    | 1 AHB サイクル     | 1 AHB サイクル       |
| $t_{BMA}$ : バスマトリックスのアービトレーション<br>(同時実行なしの場合) <sup>(1)</sup> | 1 AHB サイクル                    | 1 AHB サイクル     | N/A              |
| $t_{EDT}$ : 有効データ転送  | 1 AHB サイクル <sup>(2) (3)</sup> | 2 APB サイクル     | 2 APB サイクル       |
| $t_{BS}$ : バス同期  | N/A                           | 1 AHB サイクル     | 1 AHB サイクル       |

1. STM32F401/STM32F410/STM32F411/STM32F412 ファミリでは  $t_{BMA}$  はゼロに等しくなります。

2. FMC の場合、使用する外部メモリに応じてサイクルが追加される可能性があります。外部メモリのタイミングに応じて AHB サイクルが追加されます。

3. パーストの場合、有効データ転送時間はパースト長 ( $INC4 \cdot t_{EDT} = 4 \text{ AHB サイクル}$ ) によって決まります。

- $T_{SM}$  は、DMA メモリポートアクセスと転送の合計時間であり、次の式に等しくなります。  
$$T_{SM} = t_{MA} + t_{MAC} + t_{BMA} + t_{SRAM}$$

ここで、

表 5. メモリポートアクセス / 転送時間

| 説明   | 遅延                        |
|--|---------------------------|
| $t_{MA}$ : DMA メモリポートのアービトレーション                          | 1 AHB サイクル                |
| $t_{MAC}$ : メモリアドレスの計算                                   | 1 AHB サイクル                |
| $t_{BMA}$ : バスマトリックスのアービトレーション(同時実行なしの場合) <sup>(1)</sup> | 1 AHB サイクル <sup>(2)</sup> |
| $t_{SRAM}$ : SRAM の読み出し / 書込みアクセス                        | 1 AHB サイクル                |

1. STM32F401/STM32F410/STM32F411/STM32F412 ファミリでは  $t_{BMA}$  はゼロに等しくなります。

2. 連続 SRAM アクセス (途中で他のマスタが同一の SRAM にアクセスしない) の場合、 $t_{BMA} = 0$  サイクルです。

## 3.1.2 DMA 転送時間と同時アクセス

複数のマスタが同一のスレーブに同時にアクセスしようとしている場合は、[セクション 3.1.1](#) で説明している DMA サービス時間に、さらに遅延が加わる可能性があります。

ペリフェラルとメモリの最悪ケースのアクセス / 転送時間に関して、次の要因が、DMA ストリームサービスの総遅延時間に影響を与えます。

- 複数のマスタが同一の AHB 転送先に同時にアクセスしている場合、DMA の遅延に影響を与えます。DMA 転送は、[セクション 2.1.2](#) で説明しているように、バスマトリックスアービタがアクセス権を DMA に付与するまで開始できません。
- 複数のマスタ (DMA と CPU) が同一の AHB-APB ブリッジにアクセスしている場合、[セクション 2.3.2](#) で説明しているように、AHB-APB ブリッジアービトレーションにより DMA 転送時間に遅延が発生します。

### 3.2 例

#### 3.2.1 ADC—SRAM DMA 転送

この例は、STM32F2、STM32F405/415、STM32F407/417、STM32F427/437、および STM32F429/439 の各製品ファミリに該当します。

ADC は、連続トリプルインターブモードに設定されています。このモードの ADC は、最大 ADC 速度 (36 MHz) で 1 つのアナログ入力チャネルを連続的に変換します。ADC プリスケーラは 2 に設定され、サンプリング時間は 1.5 サイクルに設定され、インターブモードの 2 つの連続する ADC サンプルの間の遅延は 5 サイクルに設定されています。

DMA2 ストリーム 0 は、ADC 変換値を SRAM バッファに転送します。DMA2 はダイレクトパス経由で ADC にアクセスしますが、DMA はバスマトリックス経由で SRAM にアクセスします。

**表 6. DMA ペリフェラル (ADC) ポートの転送の遅延**

| AHB/APB2 周波数                       | $F_{AHB} = 72 \text{ MHz}$ /<br>$F_{APB2} = 72 \text{ MHz}$<br>AHB/APB 比 = 1 | $F_{AHB} = 144 \text{ MHz}$ /<br>$F_{APB2} = 72 \text{ MHz}$<br>AHB/APB 比 = 2 |
|------------------------------------|--|---|
| $t_{PA}$ : DMA ペリフェラルポートのアービトレーション | 1 AHB サイクル   | 1 AHB サイクル  |
| $t_{PAC}$ : ペリフェラルアドレスの計算          | 1 AHB サイクル   | 1 AHB サイクル  |
| $t_{BMA}$ : バスマトリックスのアービトレーション     | N/A <sup>(1)</sup>   | N/A <sup>(1)</sup>  |
| $t_{EDT}$ : 有効データ転送                | 2 AHB サイクル   | 4 AHB サイクル  |
| $t_{BS}$ : バス同期                    | 1 AHB サイクル   | 1 AHB サイクル  |
| $T_{SP}$ : ペリフェラルポートの総 DMA 転送時間    | 5 AHB サイクル   | 7 AHB サイクル  |

1. DMA2 はダイレクトパス経由で ADC にアクセスし、バスマトリックスアービトレーションは実行されません。

**表 7. DMA メモリ (SRAM) ポートの転送の遅延**

| CPU/APB2 周波数                    | $FAHB = 72 \text{ MHz}$ /<br>$FAPB2 = 72 \text{ MHz}$<br>AHB/APB 比 = 1 | $FAHB = 144 \text{ MHz}$ /<br>$FAPB2 = 72 \text{ MHz}$<br>AHB/APB 比 = 2 |
|---------------------------------|--|---|
| $t_{MA}$ : DMA メモリポートのアービトレーション | 1 AHB サイクル   | 1 AHB サイクル  |
| $t_{MAC}$ : メモリアドレスの計算          | 1 AHB サイクル   | 1 AHB サイクル  |
| $t_{BMA}$ : バスマトリックスのアービトレーション  | 1 AHB サイクル <sup>(1)</sup>  | 1 AHB サイクル <sup>(1)</sup>   |
| $t_{SRAM}$ : SRAM の書き込みアクセス     | 1 AHB サイクル   | 1 AHB サイクル  |
| $T_{SM}$ : メモリポートの総 DMA 転送時間    | 4 AHB サイクル   | 4 AHB サイクル  |

1. DMA が SRAM に複数回アクセスする場合、途中で他のマスターが SRAM にアクセスしなければ、バスマトリックスアービトレーションは 0 サイクルになります。

この例では、SRAM への ADC 値の書き込みの ADC DMA トリガ (ADC EOC) による総 DMA 遅延は、AHB/APB プリスケーラが 1 の場合は 9 AHB サイクル、AHB/APB プリスケーラが 2 の場合は 11 AHB サイクルになります。

注： FIFO を使用している場合、ユーザが設定した FIFO レベルに達すると、DMA メモリポートアクセスが起動されます。

## 3.2.2 SPI 全二重 DMA 転送

この例は、STM32F2、STM32F405/415、STM32F407/417、STM32F427/437、および STM32F429/439 の各製品ファミリに該当し、SPI1 ペリフェラルに基づいています。

次の 2 つの DMA リクエストが設定されています。

- DMA2\_Stream2 (SPI1\_RX) : このストリームは、SPI1 で受信したデータを速やかに処理して SPI1\_DR レジスタから SRAM バッファに転送するために、最も高い優先順位が設定されています。
- DMA2\_Stream3 (SPI1\_TX) : このストリームは、SRAM バッファから SPI1\_DR レジスタにデータを転送します。

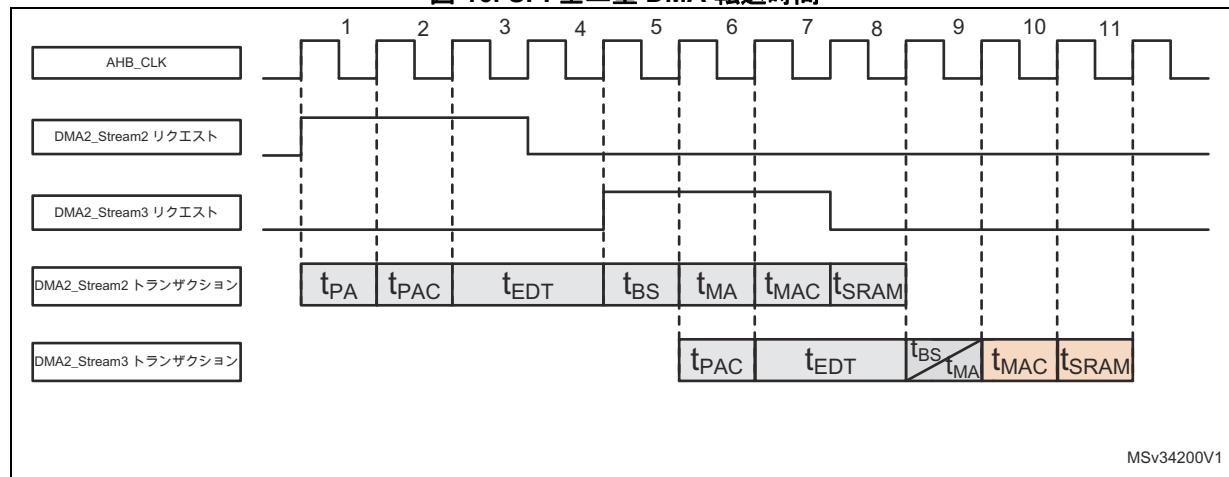
AHB 周波数は APB2 周波数 (84 MHz) に等しく、SPI1 は最大速度 (42 MHz) で動作するように設定されています。DMA2\_Stream2 (SPI1\_RX) は、DMA2\_Stream3 (SPI1\_TX) より前にトリガされます。DMA2\_Stream3 (SPI1\_TX) は、2 AHB サイクル後にトリガされます。

この設定で、CPU は無期限に I2C1\_DR レジスタをポーリングします。I2C1 ペリフェラルが APB1 でマッピングされ、SPI1 ペリフェラルが APB2 でマッピングされているので、システムパスは次のようにになります。

- DMA2 が APB2 にアクセスするためのダイレクトパス (バスマトリックスを経由しない)
- CPU がバスマトリックス経由で APB1 にアクセス

目的は、CPU による APB1 のポーリングが、DMA のタイミングに影響を与えないことを示すことです。次の図は、転送モードと受信モードの DMA タイミング、および各動作の時間のスケジューリングをまとめています。

図 16. SPI 全二重 DMA 転送時間



この図は、次の結論を示しています。

- CPU による APB1 のポーリングは、APB2 での DMA 転送遅延に影響を及ぼしません。
- DMA2\_Stream2 (SPI1\_RX) のトランザクションの場合、8 番目の AHB クロックサイクルでバスマトリックスアービトレーションは行われません。これは、SRAM にアクセスした最後のマスターが DMA2 である（したがって再アービトレーションは必要ない）と考えられるためです。
- DMA2\_Stream3 (SPI1\_TX) のトランザクションの場合、このストリームは SRAM からの読み出しを予測してそれを FIFO に書き込みます。その後、トリガされると DMA ペリフェラルポート（転送先は SPI1）が動作を開始します。
- DMA2\_Stream3 の場合、DMA2\_Stream2 バス同期サイクル中に DMA ペリフェラルアービトレーションフェーズ (1 AHB サイクル) が実行されます。

# 参考資料

この最適化は、現在の DMA リクエストトランザクションが終わる前にDMA リクエストがトリガされた場合は常に、このように実行されます。

## 4 DMA コントローラのプログラミングにおけるヒントとワーニング

### 4.1 DMA を無効にするソフトウェアシーケンス

DMA ストリームリクエストに接続されているペリフェラルをオフするには、次のことを実行する必要があります。

1. ペリフェラルの接続先の DMA ストリームをオフします。
2. DMA\_SxCR レジスタの EN ビットがリセットされる (0 になる) まで待ちます。

その後はじめてペリフェラルを安全に無効にすることができます。ペリフェラル側からの保留中のリクエストがクリアされるようにするには、ペリフェラル制御レジスタの DMA リクエストのイネーブルビットをリセットする (0 にする) 必要があります。

注： どちらの場合も、ストリームが無効になったことによる転送の終了を示すため、転送完了割込みフラグ (DMA\_LISR または DMA\_HISR の TCIF) がセットされます。

### 4.2 新しい転送を有効にする前の DMA フラグ管理

新しい転送を有効にする前に、ユーザは、DMA\_LISR または DMA\_HISR の転送完了割込みフラグ (TCIF) がクリアされていることを確認する必要があります。

一般的には、新しい転送を開始する前に、DMA\_LIFCR レジスタと DMA\_HIFCR レジスタのすべてのフラグをクリアすることを推奨します。

### 4.3 DMA を有効にするソフトウェアシーケンス

次のソフトウェアシーケンスは、DMA を有効にするときに適用します。

1. 適切な DMA ストリームを設定します。
2. 使用する DMA ストリームを有効にします (DMA\_SxCR レジスタの EN ビットをセット)。
3. 使用するペリフェラルを有効にします。ペリフェラル制御レジスタの DMA リクエストのイネーブルビットをセットする (1 にする) 必要があります。

注： 使用するペリフェラルを、対応する DMA ストリームより先に有効にした場合、DMA が最初に必要なデータをペリフェラルに提供する (メモリからペリフェラル転送の場合) 準備ができていないことにより「FEIF」(FIFO エラー割込みフラグ) がセットされる可能性があります。

### 4.4 NDTR=0 の場合のメモリ間転送

通常モードでメモリ間転送を実行するように DMA ストリームが設定されている場合、NDTR が 0 に到達すると、転送完了がセットされます。そのとき、ユーザがこのストリームのイネーブルビット (DMA\_SxCR の EN ビット) をセットしている場合は、メモリ間転送が自動的に最後の NDTR 値を使用して再トリガされます。

## 4.5 PINC/MINC=0 の場合の DMA ペリフェラルバースト

ペリフェラルアドレスインクリメント (PINC) またはメモリアドレスインクリメント (MINC) を無効にして、DMA バースト機能を使用すると、バースト対応 (FIFO 内蔵) の内蔵または外部 (FSMC) ペリフェラルに対処できます。このモードは、この DMA ストリームのトランザクション中に他の DMA ストリームが割り込むことはできないようにします。

## 4.6 2 回マッピングされている DMA リクエスト

ユーザが、同一のペリフェラルを処理するように 2 つ（以上）の DMA ストリームを設定する場合、新しい DMA ストリームを有効にする前に現在の DMA ストリームが完全に無効になっていることを、ソフトウェアで (DMA\_SxCR レジスタの EN ビットをポーリングすることによって) 確認する必要があります。

## 4.7 最適な DMA スループット設定

STM32F4xx を AHB 周波数を下げて使用していて、DMA がハイスピードペリフェラルを処理している場合、スタックヒープは、SRAM ではなく、CCM (CPU が D-バス経由で直接アドレス指定可能) に配置することを推奨します。これにより、SRAM メモリにアクセスする CPU と DMA の同時並行性が高まります。

## 4.8 DMA 転送の中斷

DMA 転送は、いつでも中斷して再開することもできれば、DMA 転送が終了する前に完全に無効化することもできます。

これには、2 つのケースがあります。

- ストリームは、転送を無効にし、後で停止したところから再開することはできません。ストリームを無効化するために DMA\_SxCR レジスタの EN ビットをクリアして EN ビットがリセットされるまで待つこと以外、特に行うことはありません。その結果、以下のようになります。
  - ソフトウェアによって、ストリームが中斷される前に転送されたデータ項目の数を判断できるように、DMA\_SxNDTR レジスタに、ストリームが停止した時点で残っているデータ項目の数を保持しています。
- ストリームは転送を中斷し、ストリームを再度有効にすることで後から転送を再開することができます。転送が停止したところから再開するために、ソフトウェアはストリームを無効 (EN ビットを 0) にした後に DMA\_SxNDTR レジスタを読み出して、すでに収集したデータ項目数を確認する必要があります。その後、
  - アドレスポインタを調整するために、ペリフェラルやメモリのアドレスを更新する必要があります。
  - SxNDTR レジスタは、転送すべきデータ項目の残数 (ストリームを無効にしたとき読み出される値) で更新する必要があります。
  - ストリームは、その後再度有効化して転送が停止したところから再開することができます。

注： どちらの場合も、ストリームが中斷されたことによる転送の終了を示すため、転送完了割り込みフラグ (DMA\_LISR または DMA\_HISR の TCIF) がセットされます。

## 4.9 DMA2 コントローラとシステムアーキテクチャの柔軟性の利用

このセクションでは、STM32 アーキテクチャと DMA コントローラによって提供される柔軟性を利用する方法を示します。この柔軟性を説明するために、DMA2 AHB のペリフェラルポートとメモリポートを反転させて、ペリフェラルのデータ転送を適切に管理し続ける方法を示します。これを達成して、通常の DMA2 動作の制御を引き継ぐには、DMA2 のワーキングモデルを見直す必要があります。

両方の DMA2 ポートが AHB バスマトリックスに接続され、AHB スレーブと対称接続されているので、このアーキテクチャは、ソフトウェア設定に応じて、ペリフェラルポートおよびメモリポートを経由してどちらかの方向にトラフィックを流すことができます。

### 4.9.1 DMA2 AHB ポート上で転送を反転させる場合の考慮事項

ソフトウェアには、ニーズに応じて DMA2 ストリームの転送モードを設定する柔軟性があります。この設定に応じて、DMA2 AHB ポートの一方は読出し方向で、もう一方は書込み方向で、プログラムされます。

表 8 に、DMA AHB ポートの方向と転送モード設定を示します。

表 8. DMA AHB ポートの方向と転送モード設定

| 転送モード       | DMA2 AHB メモリポート | DMA2 AHB ペリフェラルポート |
|-------------|-----------------|--------------------|
| メモリからペリフェラル | 読出し方向           | 書込み方向              |
| ペリフェラルからメモリ | 書込み方向           | 読出し方向              |
| メモリ間        | 書込み方向           | 読出し方向              |

ここでトラフィックフローに注目すると、セクション 2.2.2 : DMA 転送状態に示すように、ペリフェラルポート上の転送はペリフェラルリクエストによってトリガされ、メモリポート上の転送は FIFO 閾値によって（FIFO モード使用時）トリガされるか、またはペリフェラル読出しの直後に（ダイレクトモード使用時）トリガされます。

DMA2 メモリポートでペリフェラルを管理する場合、プリトリガ転送、バッファ転送、および最終データ管理に注意する必要があります。

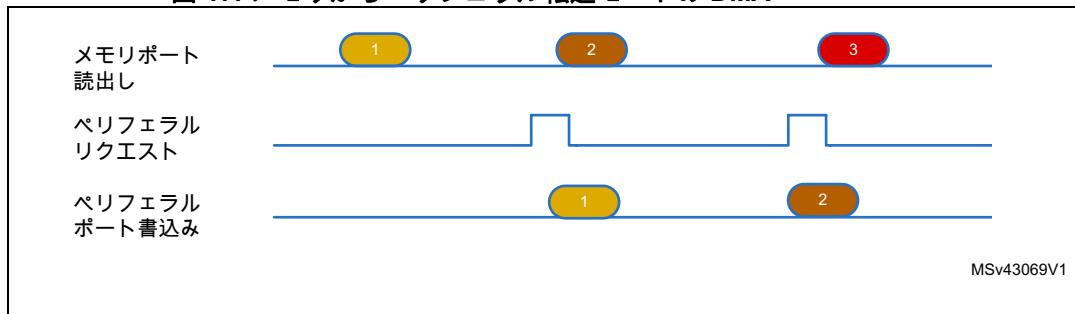
#### プリトリガ転送 :

セクション 2.2.2 : DMA 転送状態で説明しているように、メモリからペリフェラル転送モード（メモリポート経由のデータ読出し）が設定されている場合、DMA はペリフェラルのアクセスを予測して、DMA ストリームが有効になるとすぐにデータを読み出します。1 つのデータがダイレクトモードで、DMA FIFO が有効であれば最大  $4 \times 32$  ビットワードまでバッファされます。

DMA2 メモリポート経由でのペリフェラル読出しを管理する場合、ソフトウェアでは、最初の DMA アクセスの有効性を保証するため、DMA を有効にする前にペリフェラルを有効にしておく必要があります。

図 17 は、メモリポートおよびペリフェラルポートを経由した DMA アクセスとペリフェラルのトリガを示しています。

図 17. メモリからペリフェラル転送モードの DMA



## 最終データ読出しの管理

DMA コントローラにはストリームあたり 1 つの  $4 \times 32$  ビットワード FIFO が搭載され、AHB ポート間でデータをバッファするため使用されます。DMA メモリポート経由のペリフェラル読出しを管理する場合、ソフトウェアでは、4 ワード余分にペリフェラルから読み出すようにする必要があります。これは、DMA FIFO から最後の有効なデータが確実に転送されるようにするためです。

## DMA ダイレクトモードが無効な場合のバッファ転送 :

インダイレクトモード (FIFO モードが有効) でメモリポートからペリフェラルにデータを書き込む場合、ソフトウェアでは、このポート経由のアクセスがプログラムされている FIFO 閾値によってトリガされることに注意する必要があります。閾値に到達すると、データは FIFO から転送先にメモリポート経由で転送されます。

レジスタに書き込む場合 (たとえば FIFO を持たない GPIO)、DMA FIFO からのデータは連続的に転送先に書き込まれます。

最後に、ペリフェラルの管理をペリフェラルポートからメモリポートにスワップする場合、ソフトウェアでは、転送サイズとアドレスのインクリメントの設定を再検討する必要があります。

**セクション 1.1.5 : 転送サイズ**で説明しているように、転送サイズは、ペリフェラル側の転送幅 (バイト、ハーフワード、ワード) および転送するデータ項目数 (DMA\_SxNTDR レジスタにプログラムされる値) によって定義されます。新しい DMA 設定に従って、ポートを反転する際に、DMA\_SxNTDR レジスタにプログラムされている値を調整する必要がある場合があります。

## 4.9.2 DMA2 AHB ポート経由の Quad-SPI 転送の反転の例に関する考慮事項

この例で、DMA\_S7M0AR は Quad-SPI データレジスタアドレスでプログラムされ、DMA\_S7PAR はデータバッファアドレス (SRAM のバッファなど) でプログラムされています。Quad-SPI に書き込む場合は、DMA\_S7CR レジスタで DMA2 ストリーム方向をペリフェラルからメモリ転送モードに設定する必要があります。Quad-SPI から読み出す場合は、DMA2 ストリーム方向をメモリからペリフェラル転送モードに設定する必要があります。

DMA FIFO から最後のデータが RAM メモリに転送されるようにするには、読出し操作に余分な 4 ワード (32 ビット) が必要です。

# 参考資料

表 9. コードスニペット

| 書き込み動作  | 読み出し動作  |
|---|---|
| <pre>/* M0AR を QUADSPI データレジスタアドレスでプログラムします */ DMA2_Stream7-&gt;M0AR = (uint32_t)&amp;QUADSPI-&gt;DR; /* PAR をバッファアドレスでプログラムします */ DMA2_Stream7-&gt;PAR = (uint32_t)&amp;u32Buffer[0]; /* 転送するデータ項目の数を書き込みます */ DMA2_Stream7-&gt;NDTR = 0x100; /* DMA の設定 : MSIZE=PSIZE=0x02 (ワード)、 CHSEL=0x03 (QUADSPI)、PINC=1、DIR=0x00 */ DMA2_Stream7-&gt;CR = DMA_SxCR_PSIZE_1   DMA_SxCR_MSIZE_1   3ul&lt;&lt;25   DMA_SxCR_PINC; DMA_SxCR_DIR_0;  /* DMA リクエストの生成を有効にします */ QUADSPI-&gt;CR  = QUADSPI_CR_DMAEN;  /* DLR レジスタを書き込みます */ QUADSPI-&gt;DLR = (0x100* 4)-1;  /* QUADSPI CCR に書き込みます */ QUADSPI-&gt;CCR = QUADSPI_CCR_IMODE_0  QUADSPI_CCR_ADMODE_0  QUADSPI_CCR_DMODE  QUADSPI_CCR_ADSIZE  QUAD_IN_FAST_PROG_CMD;  /* AR レジスタを書き込みます */ QUADSPI-&gt;AR = 0x00ul;  /* EN ビットをセットして、選択されている DMA2_Stream7 を有効にします */ DMA2_Stream7-&gt;CR  = (uint32_t)DMA_SxCR_EN;  /* 転送終了を待ちます */ while((QUADSPI-&gt;SR &amp; QUADSPI_SR_TCF) != QUADSPI_SR_TCF);</pre> | <pre>/* M0AR を QUADSPI データレジスタアドレスでプログラムします */ DMA2_Stream7-&gt;M0AR = (uint32_t)&amp;QUADSPI-&gt;DR; /* PAR をバッファアドレスでプログラムします */ DMA2_Stream7-&gt;PAR = (uint32_t)&amp;u32Buffer[0]; /* 転送するデータ項目の数を書き込みます */ DMA2_Stream7-&gt;NDTR = 0x100; /* DMA の設定 : MSIZE=PSIZE=0x02 (ワード)、 CHSEL=0x03 (QUADSPI)、PINC=1、DIR=0x01 */ DMA2_Stream7-&gt;CR = DMA_SxCR_PSIZE_1   DMA_SxCR_MSIZE_1   3ul&lt;&lt;25   DMA_SxCR_PINC  DMA_SxCR_DIR_0;  /* DMA リクエストの生成を有効にします */ QUADSPI-&gt;CR  = QUADSPI_CR_DMAEN;  /* DLR レジスタを書き込みます */ QUADSPI-&gt;DLR = ((0x100+4)* 4)-1;  /* QUADSPI CCR に書き込みます */ QUADSPI-&gt;CCR = QUADSPI_CCR_IMODE_0  QUADSPI_CCR_ADMODE_0  QUADSPI_CCR_DMODE  QUADSPI_CCR_ADSIZE  QUADSPI_CCR_FMODE_0 QUAD_OUT_FAST_READ_CMD;  /* AR レジスタを書き込みます */ QUADSPI-&gt;AR = 0x00ul;  /* EN ビットをセットして、選択されている DMA2_Stream7 を有効にします */ DMA2_Stream7-&gt;CR  = (uint32_t)DMA_SxCR_EN;  /* 転送終了を待ちます */ while((DMA2_Stream7-&gt;CR &amp; DMA_SxCR_EN) == DMA_SxCR_EN);</pre> |

注 :

DMA2 が AHB 転送と APB2 転送を並行して管理している場合のデータ破損の制限（影響を受ける STM32F2/F4 MCU を特定するには製品エラッターシートを参照）は、このセクションで説明したように、DMA2 のペリフェラルポートとメモリポートをスワップすることによって克服できます。

## 4.10 STM32F7 DMA 転送とデータのインコヒーレンシを回避するためのキャッシュ保持

ソフトウェアが DMA 転送元 / 転送先バッファとしてキャッシュ可能なメモリ領域を使用している場合、DMA の動作を開始する前に、すべてのデータがサブシステムメモリにコミットされるようするため、キャッシュクリーンをトリガする必要があります。DMA 転送の完了後にペリフェラルからデータを読み出す場合、ソフトウェアは、更新されたメモリ領域を読み出す前にキャッシュの無効化を実行する必要があります。

DMA バッファにはキャッシュ不能な領域を使用することを推奨します。ソフトウェアは、MPU を使用して、キャッシュ不能なメモリブロックを CPU と DMA の間の共有メモリとして使用するようにセットアップできます。

## 5 結論

DMA コントローラは、次のような特徴によって、ほとんどの組込み用途に対応するように設計されています。

- 16 ストリーム x 16 チャネル（DMA ごとに 8 チャネル）の中で適切な組み合わせを選択する柔軟性をファームウェアに与えます。
- デュアル AHB ポートアーキテクチャによって DMA 転送の総遅延時間を短縮します。また、APB ブリッジへのダイレクトパスによって、DMA がロースピード APB ペリフェラルを処理しているときに CPU が AHB1 アクセスで停止するのを回避します。
- DMA の FIFO 実装は、転送先と転送元で異なるデータサイズを設定する柔軟性をファームウェアに与え、インクリメンタルバースト転送モードを使用する際の転送をスピードアップします。

# 参考資料

AN4031

改版履歴

## 6 改版履歴

表 10. 文書改版履歴

| 日付              | 版 | 変更内容   |
|-----------------|---|--|
| 2014 年 2 月 4 日  | 1 | 初版発行   |
| 2015 年 8 月 6 日  | 2 | <p>セクション : 参照文献を追加。</p> <p>ドキュメント範囲を STM32F2 および STM32F4 シリーズに拡張し、リファレンスマニュアルのリストを更新。</p> <p>表 1: STM32F427/437 と STM32F429/439 の DMA1 リクエストマッピングおよび表 2: STM32F427/437 と STM32F429/439 の DMA2 リクエストマッピングの注 1 を削除。STM32F401 ラインの DMA1 リクエストマッピングおよび STM32F401 ラインの DMA2 リクエストマッピングの 2 つの表を削除。</p> <p>セクション 4.1: DMA を無効にするソフトウェアシーケンスおよびセクション 4.3: DMA を有効にするソフトウェアシーケンスを更新。</p>  |
| 2016 年 6 月 23 日 | 3 | <p>追加 :</p> <ul style="list-style-type: none"><li>- セクション 4.9 : DMA2 コントローラとシステムアーキテクチャの柔軟性の利用</li><li>- セクション 4.10: STM32F7 DMA 転送とデータのインコヒーレンシを回避するためのキャッシュ保持</li><li>- 更新 :</li><li>- はじめに</li><li>- セクション 2 : システム性能に関する考慮事項</li><li>- セクション 2.1.3 : バスマトリックスアービトレーションと DMA 転送の遅延の最悪のケース</li><li>- 図 14 : DMA リクエストのアービトレーション</li></ul> <p>削除 :</p> <ul style="list-style-type: none"><li>- 表 1: 対象とする製品</li></ul> |

表 11. 日本語版文書改版履歴

| 日付         | 版 | 変更内容      |
|------------|---|-----------|
| 2018 年 6 月 | 1 | 日本語版 初版発行 |



## 重要なお知らせ（よくお読み下さい）

STMicroelectronics NV およびその子会社（以下、ST）は、ST製品及び本書の内容をいつでも予告なく変更、修正、改善、改定及び改良する権利を留保します。購入される方は、発注前にST製品に関する最新の関連情報を必ず入手してください。ST 製品は、注文請書発行時点での有効なST の販売条件に従って販売されます。

ST製品の選択並びに使用については購入される方が全ての責任を負うものとします。購入される方の製品上の操作や設計に関してST は一切の責任を負いません。

明示又は默示を問わず、STは本書においていかなる知的財産権の実施権も許諾致しません。

本書で説明されている情報とは異なる条件でST 製品が再販された場合、その製品についてSTが与えたいかなる保証も無効となります。

ST およびST ロゴはSTMicroelectronics の商標です。その他の製品またはサービスの名称は、それぞれの所有者に帰属します。

本書の情報は本書の以前のバージョンで提供された全ての情報に優先し、これに代わるものです。

この資料は、STMicroelectronics NV 並びにその子会社(以下ST)が英文で記述した資料（以下、「正規英語版資料」）を、皆様のご理解の一助として頂くためにSTマイクロエレクトロニクス(株)が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST及びSTマイクロエレクトロニクス(株)は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにも関わらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や障害などにつきましては如何なる責任も負いません。

© 2018 STMicroelectronics - All rights reserved