

## サイプレスはインフィニオン テクノロジーズになりました

この表紙に続く文書には「サイプレス」と表記されていますが、これは同社が最初にこの製品を開発したからです。新規および既存のお客様いずれに対しても、引き続きインフィニオンがラインアップの一部として当該製品をご提供いたします。

## 文書の内容の継続性

下記製品がインフィニオンの製品ラインアップの一部として提供されたとしても、それを理由としてこの文書に変更が加わることはありません。今後も適宜改訂は行いますが、変更があった場合は文書の履歴ページでお知らせします。

## 注文時の部品番号の継続性

インフィニオンは既存の部品番号を引き続きサポートします。ご注文の際は、データシート記載の注文部品番号をこれまで通りご利用下さい。

## Traveo II ファミリの保護ユニット設定

著者: Koji Mizumoto

関連製品ファミリ: Traveo™ II ファミリ CYT2/CYT3/CYT4 シリーズ

関連アプリケーションノート: [関連ドキュメント](#)

AN219843 は、Cypress Traveo™ II ファミリ MCU の保護ユニットの機能およびコンフィグレーションについて説明します。このドキュメントは、さまざまな操作に基づいてシステムセキュリティを強化するためのガイドとして使用できます。また、各保護ユニットの構造、アクセス属性、および使用例についても説明します。

## Contents

|                         |    |                                   |    |
|-------------------------|----|-----------------------------------|----|
| 1 はじめに .....            | 1  | 4.1 MPU 構造 .....                  | 11 |
| 2 保護ユニット .....          | 2  | 4.2 SMPU 構造 .....                 | 12 |
| 2.1 保護ユニットの配置 .....     | 2  | 4.3 PPU 構造 .....                  | 13 |
| 2.2 保護ユニットの概要 .....     | 2  | 4.4 保護ペア構造 .....                  | 13 |
| 3 動作概要 .....            | 3  | 5 保護ユニットの設定例 .....                | 15 |
| 3.1 バス転送の保護プロパティ .....  | 3  | 5.1 CPUの一部として実装される MPU の設定例 ..... | 15 |
| 3.2 属性の継承 .....         | 4  | 5.2 バスの一部として実装される MPU の設定例 .....  | 18 |
| 3.3 ユーザ/特権属性の切換え .....  | 5  | 5.3 SMPU の設定例 .....               | 18 |
| 3.4 保護コンテキスト属性の設定 ..... | 6  | 5.4 PPU の設定例 .....                | 20 |
| 3.5 バス転送の評価 .....       | 7  | 6 関連ドキュメント .....                  | 23 |
| 3.6 マスタ識別子 .....        | 9  | 改版履歴 .....                        | 24 |
| 3.7 保護違反 .....          | 10 | ワールドワイド販売と設計サポート .....            | 25 |
| 4 保護ユニット構造 .....        | 11 |                                   |    |

## 1 はじめに

このアプリケーションノートでは、Traveo II ファミリ MCU の保護ユニットについて説明します。本シリーズは、Arm® Cortex® CPU、暗号化コンポーネントとしての enhanced Secure Hardware Extension (eSHE)、CAN FD、メモリ、およびアナログとデジタルの周辺機能を 1 チップに搭載します。

CYT2 シリーズは、1 つの Arm Cortex-M4F ベースの CPU (CM4) と Cortex-M0+ ベースの CPU (CM0+) を持ちます。CYT4 シリーズは、2 つの Arm Cortex-M7 ベースの CPU (CM7) と CM0+ を持ち、CYT3 シリーズは 1 つの Arm Cortex-M7 ベースの CPU (CM7) と CM0+ を持ちます。

保護ユニットは、セキュリティシステム設計の重要な部分であり、さまざまな操作に基づいてセキュリティを強化します。保護ユニットは、特定のプロパティに従ってバス上の転送を許可または制限します。保護違反は、バス転送のアドレス領域、アクセス属性と保護ユニットのアドレス範囲、アクセス属性の不一致によって発生します。

これらのシリーズは、メモリ保護ユニット(MPU)、共有メモリ保護ユニット(SMPU)、および周辺機能保護ユニット(PPU)の 3 種類の保護ユニットを持ちます。メモリ保護は MPU および SMPU によって、周辺リソース保護は PPU によって行います。

MPU、SMPU、および PPU 保護属性の定義は、一貫したソフトウェアインタフェースを保証するため、Arm 定義(メモリ領域とアクセス属性定義の観点から)に従います。

セキュリティ要件によっては、SMPU および PPU レジスタは、システム全体の保護を実行する“セキュア”CPU によって制御される必要があります。

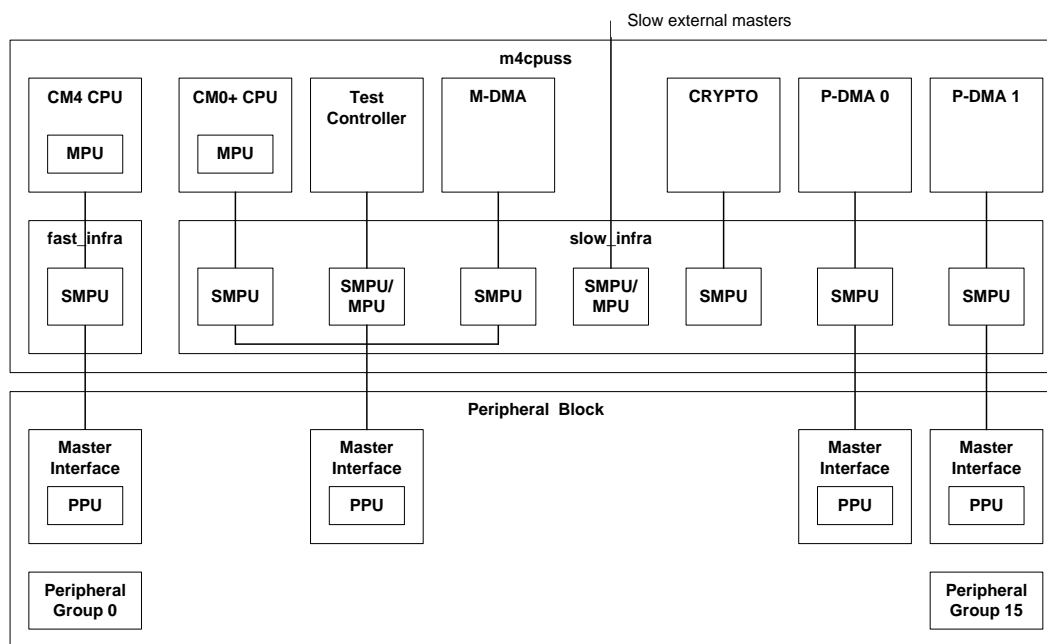
このアプリケーションノートで使用される機能と用語を理解するため、[アーキテクチャ テクニカル リファレンスマニュアル \(TRM\)](#)の Protection Unit 章を参照してください。

## 2 保護ユニット

### 2.1 保護ユニットの配置

図 1 に、CYT2B シリーズの MPU, SMPU, および PPU の配置を示します。

図 1. CYT2B シリーズの保護ユニット配置



その他のシリーズについては[アーキテクチャ TRM](#)を参照してください。

### 2.2 保護ユニットの概要

MPU は、単一のマスタに関連付けられています。以下の 2 種類の MPU があります。

- CPU の一部として実装される MPU: このタイプの MPU は Arm CPU に実装されています。
- バスの一部として実装される MPU: このタイプの MPU はテストコントローラなどのバスマスタに実装されています。

ただし、ペリフェラル DMA(P-DMA), メモリ DMA(M-DMA), および暗号コンポーネント(CRYPTO) のバスマスタには MPU はありません。

SMPU は、すべてのバスマスタによって共有されます。SMPU の単一のセットはすべての SMPU によって共有されます。

PPU は、すべてのバスマスタによって共有されます。ペリフェラルグループ内の各周辺機能へのアクセスを制御します。PPU は以下の 2 種類があります。

- Fixed PPU: 保護されるアドレス領域は固定され、ソフトウェアによる変更はできません。
- Programmable PPU: 保護されるアドレス領域はソフトウェアによってプログラムできます。

MPU と S MPU は、PPU より高い優先順位を持ちます。また、Programmable PPU は、Fixed PPU よりも高い優先順位を持っています。

保護ユニットの詳細は、[アーキテクチャ TRM](#) を参照してください。

## 3 動作概要

### 3.1 バス転送の保護プロパティ

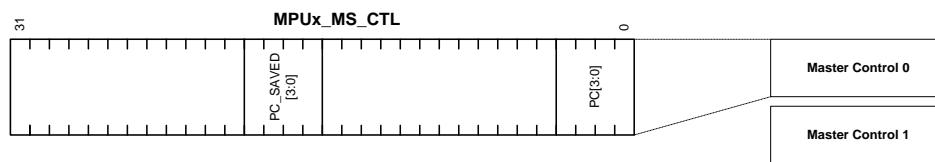
保護ユニットは、以下のバス転送のプロパティを識別します。

- アクセスするアドレス範囲
- アクセス属性
  - リード/ライト: ライトアクセスとリードアクセスを区別します。
  - 実行: データアクセスとコードアクセスを区別します。
  - ユーザ/特権: 特権アクセスとユーザアクセスを区別します。
  - セキュア/非セキュア: 非セキュアアクセスとセキュアアクセスを区別します。非セキュア属性の場合、非セキュアアクセスとセキュアアクセスの両方が許可されます。
  - 保護コンテキスト: 異なる保護コンテキストを区別します。

すべてのバスマスタが、これらのアクセス属性すべてを提供するわけではありません。保護コンテキスト属性を持つバスマスタはありません。また Arm CPU は、セキュア属性を提供しません。

バスマスタによって提供されないアクセス属性は、MPUx\_MS\_CTL および S MPU\_MSx\_CTL レジスタによって提供されます。これらのレジスタは、ブート処理中またはセキュア CPU によって設定されます。[図 2](#) に、MPUx\_MS\_CTL のレジスタ構造を示します。

図 2. MPUx\_MS\_CTL レジスタ

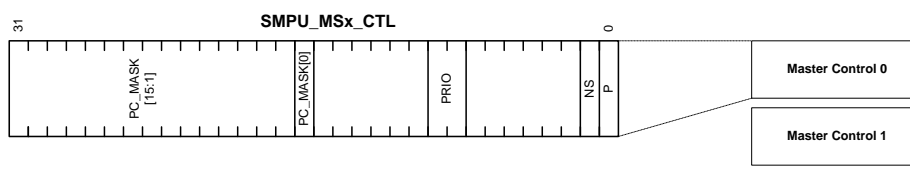


このレジスタは、マスタアクセスに保護コンテキスト属性を付与します。

- MPUx\_MS\_CTL.PC: 自身のアクセスに付与する保護コンテキスト属性を設定します。
- MPUx\_MS\_CTL.PC\_MASKED: ブートプロセスによって設定されます。このフィールドは、CM0+にのみ存在します。

[図 3](#) に、S MPU\_MSx\_CTL のレジスタ構造を示します。

図 3. S MPU\_MSx\_CTL レジスタ



このレジスタは、バスマスタアクセスに以下の属性を付与します。

- S MPU\_MSx\_CTL.P: 独自にユーザ/特権属性を持たないバスマスタにユーザ/特権属性を付与します。
- S MPU\_MSx\_CTL.NS: 独自にセキュア/非セキュア属性を持たないバスマスタにセキュア/非セキュア属性を付与します。

- SMPU\_MSx\_CTL.PC\_MASK:バスマスタが MPUx\_MS\_CTL.PC に設定可能な保護コンテキストを制限します。
- PC\_MASK[0]は、常に"0"です。すなわち、バスマスタは PC="0"属性を設定できません。
- SMPU\_MSx\_CTL.PRIO: バス調停時の優先順位を設定します。

しかし、すべてのバスマスタがこれらのレジスタフィールドを持つわけではありません。表 1 に、各マスタのもつレジスタフィールドを示します。

表 1. 各バスマスタに提供されるレジスタフィールド

| Register Field          | CM0+ CPU | CRYPTO Component | P-DMA 0 | P-DMA 1 | M-DMA | CM4F CPU | Test Controller |
|-------------------------|----------|------------------|---------|---------|-------|----------|-----------------|
| MPUx_MS_CTL.PC          | Yes      | —                | —       | —       | —     | Yes      | Yes             |
| MPUx_MS_CTL.PC_SAVED    | Yes      | —                | —       | —       | —     | —        | —               |
| SMPU_MSx_CTL.P          | —        | —                | —       | —       | —     | —        | Yes             |
| SMPU_MSx_CTL.NS         | Yes      | —                | —       | —       | —     | Yes      | Yes             |
| SMPU_MSx_CTL.MS_PC_MASK | Yes      | —                | —       | —       | —     | Yes      | Yes             |
| SMPU_MSx_CTL.PRIO       | Yes      | Yes              | Yes     | Yes     | Yes   | Yes      | Yes             |

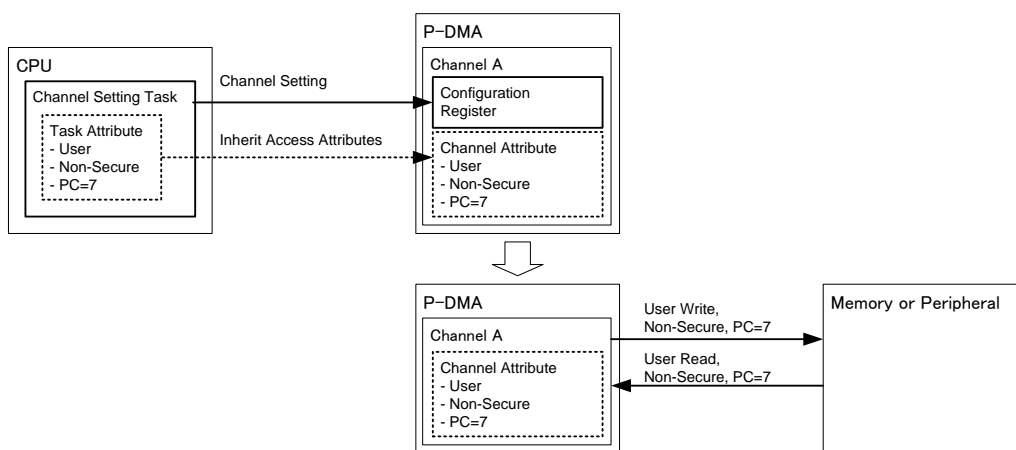
P-DMA0/1, M-DMA, および CRYPTO コンポーネントは MPU を持ちません。これらの周辺機能には、属性を設定するフィールドがありません。

各バスマスタには、関連する SMPU MS\_CTL レジスタがあります。ただし、セキュアシステムにおいて、各マスタが独自の特権、セキュリティ、調停優先順位および保護コンテキスト設定をしないように、このレジスタは通常、セキュアマスタ(CM0+)によってのみ制御できます。

## 3.2 属性の継承

P-DMA0/1, M-DMA, および CRYPTO コンポーネントはチャンネルまたはコンポーネントをプログラムしたバス転送のアクセス制御属性を継承します。継承されたアクセス属性はSMPUとPPUによって許可/制限されます。図 4 に、属性の継承と、その動作例を示します。

図 4. 属性継承の設定と動作例



### 3.3 ユーザ/特権属性の切換え

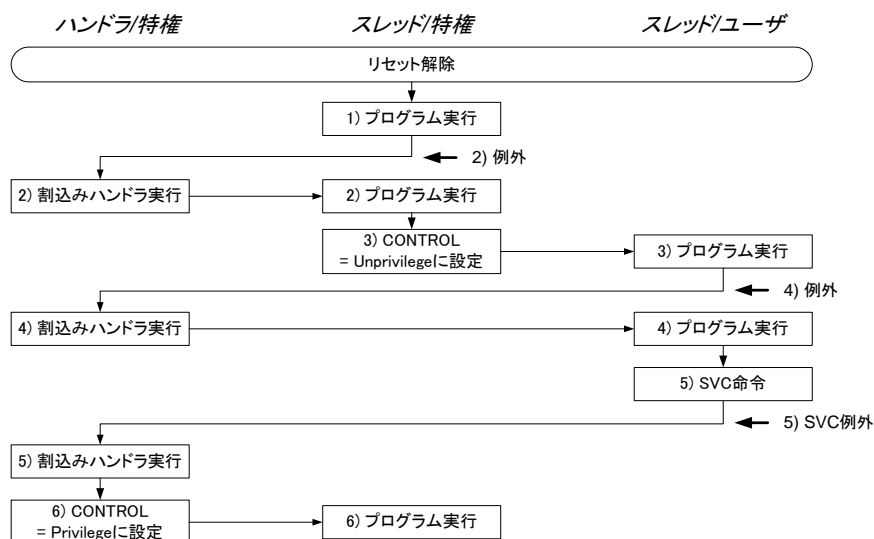
ここでは、ユーザ/特権属性をサポートする両 CPU での属性切換えについて説明します。CPU は、以下の 2 つの動作モードと 2 つの特権レベルをサポートします。

- 動作モード
  - スレッドモード: このモードは、アプリケーションソフトウェアの実行に使用されます。特権レベルまたはユーザレベルで実行可能です。
  - ハンドラモード: このモードは、例外ハンドラに使用されます。特権レベルでのみ実行可能です。
- 特権レベル
  - ユーザレベル: ソフトウェアはアクセスが制限されています。
  - 特権レベル: ソフトウェアは、すべての命令を実行でき、すべての CPU リソースにアクセスが可能です。

特権レベルは、CONTROL レジスタによって切り替えます。これは、CPU 固有のレジスタです。特権レベルからユーザレベルへの切換えは、CONTROL レジスタを使用します。ただし、CONTROL レジスタは特権レベルでのみ書換え可能なレジスタです。したがって、ユーザレベルから特権レベルに切り替える場合、常にハンドラモードを経由する必要があります。例外または割り込みが発生すると CPU はハンドラモードに遷移します。

図 5 に、SVC (スーパーバイザコール) 命令によるユーザ/特権レベル切換えの例を示します。SVC 命令は、例外を生成し、ハンドラモードへ遷移できます。

図 5. CPU のユーザ/特権レベル切換えの例



1. リセット解除後、CPU はスレッドモード/特権レベルで起動されます。
2. スレッドモード/特権レベル中に例外が発生すると、ハンドラモード/特権レベルに遷移しハンドラ処理から復帰すると、再度スレッドモード/特権レベルに遷移します。
3. スレッドモード/特権レベルでは、CONTROL レジスタによってスレッドモード/ユーザレベルへ遷移します。
4. スレッドモード/ユーザレベル中に例外が発生すると、ハンドラモード/特権レベルに遷移しハンドラ処理から復帰すると再度、スレッドモード/ユーザレベルに遷移します。
5. スレッドモード/ユーザレベルからスレッドモード/特権レベルへの切換え時は、SVC 命令を使用します。SVC 命令は、SVC 例外を発生します。
6. ハンドラモード/特権レベルに遷移後、CONTROL レジスタによって特権レベルを設定します。CPU は、ハンドラ処理から復帰後、スレッドモード/特権レベルに遷移します。

詳細は、Arm ドキュメンテーションセット(CM4, CM7, および CM0+) を参照してください。

### 3.4 保護コンテキスト属性の設定

保護コンテキスト(PC)は、セキュリティおよび安全を目的としたソフトウェア実行の分離のために使用されます。PC は、マスタによって開始されるすべてのバス転送の PC 属性として使用されます。SMPU と PPU は、PC 属性をもとにバス転送を許可または制限します。

本シリーズは、8 個の PC をサポートします。このうち、PC0, 1, 2, および 3 は特殊な PC で、これらの PC はハードウェアによって制御されます。さらに、PC0 は無制限のアクセス権を持ちます。

特定のバスマスタには、PC フィールド(MPUx\_MS\_CTL.PC and SMPU\_MSx\_CTL.PC\_MASK) が関連付けられています。

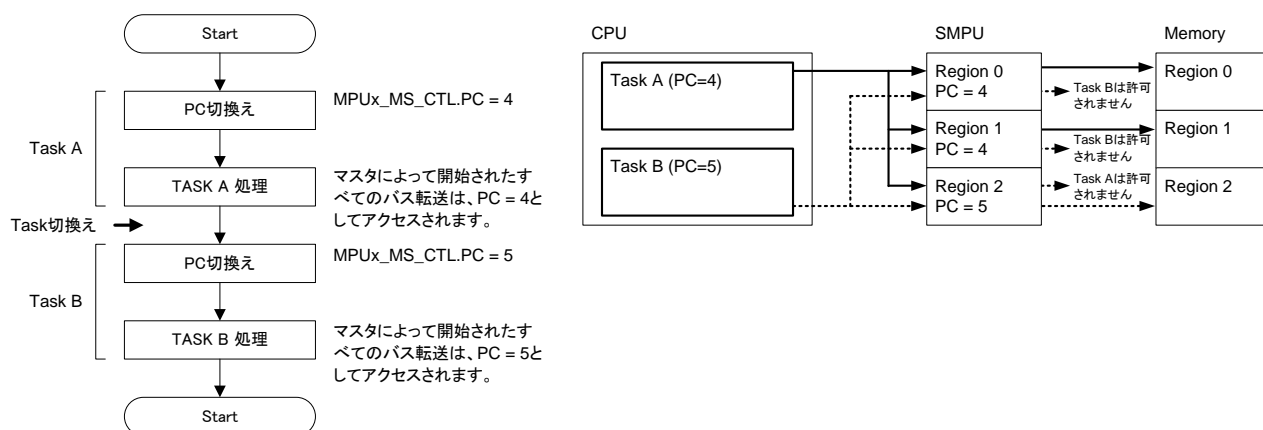
バスマスタの PC は、関連する MPUx\_MS\_CTL.PC の再プログラミングで変更できます。

SMPU\_MSx\_CTL.PC\_MASK は、関連するバスマスタが設定可能な PC を制限します。

例えば、SMPU\_MSx\_CTL.PC\_MASK[15:0] = "0x06" (PC1, 2 = "1")と設定した場合、設定可能な PC は PC1 と PC2 です。バスマスタは許可されていない PC(PC=0, 3, 4, 5, 6, 7) へ変更できません。

図 6 に、PC 変更フローと動作例を示します。

図 6. PC の変更フローと動作例



**Note:** 各マスタが設定可能な PC 値は SMPU\_MSx\_CTL.PC\_MASK によって制限されます。

これによって、SMPU や PPU の設定を変更せずに、PC フィールドの再プログラミングのみで、単一のバスマスタが異なる保護役割を持てます。

## 3.5 バス転送の評価

### 3.5.1 評価プロセス

保護ユニットによるバス転送の評価は、2つの独立したプロセスに分かれています。

- マッチングプロセス: このプロセスは、各保護構造に対して転送アドレスがアドレス範囲内に含まれているかを評価します。
- アクセス評価プロセス: このプロセスは、各保護構造に対してバス転送属性がアクセス制御属性と一致しているかを評価します。

以下の疑似コードは、バス転送の評価プロセスを示します。

```
match = 0;
for (i = n-1; i >= 0; i--) // n: number of protection regions
    if (Match ("transfer address", "protection context") {
        match = 1; break;
    }
}

if (match)
    AccessEvaluate ("access attributes", "protection context");
else
    "access allowed"
```

} マッチングプロセス

} アクセス評価プロセス

#### Note:

- 保護構造が一致しない場合、アクセスは許可されます。
- 複数の保護構造が一致する場合、アクセス評価のためのアクセス属性は、最高のインデックスをもつ保護構造によって評価されます。

保護ユニットは保護構造を降順で評価します。言い換えると、インデックスの高い保護構造はインデックスの低い保護構造よりも優先されます。

転送アドレスが不一致の場合、次に高いインデックスを持つ保護構造が評価されます。転送アドレスが一致の場合、バス転送はアクセス評価プロセスによって評価されます。

アクセス評価プロセスによってバス転送属性が不一致の場合、アクセス違反として検出されます。したがって、この場合、バス転送属性は次に高いインデックスをもつ保護構造によって評価されません。

### 3.5.2 PC\_MATCH 動作

SMPUはPC\_MATCHフィールドを持ちます。PC\_MATCHは、マッチングとアクセス評価プロセスを制御します。

- PC\_MATCH = 0 の場合

以下の疑似コードはPC\_MATCH = "0"の時の評価プロセスを示します。

```
match = 0;
for (i = n-1; i >= 0; i--) // n: number of protection regions
    if (Match ("transfer address") {
        match = 1; break;
    }
}

if (match)
    AccessEvaluate ("access attributes", "protection context");
else
    "access allowed"
```

PC\_MATCH = "0"の場合、保護コンテキストはアクセス評価プロセスでのみ評価されます。

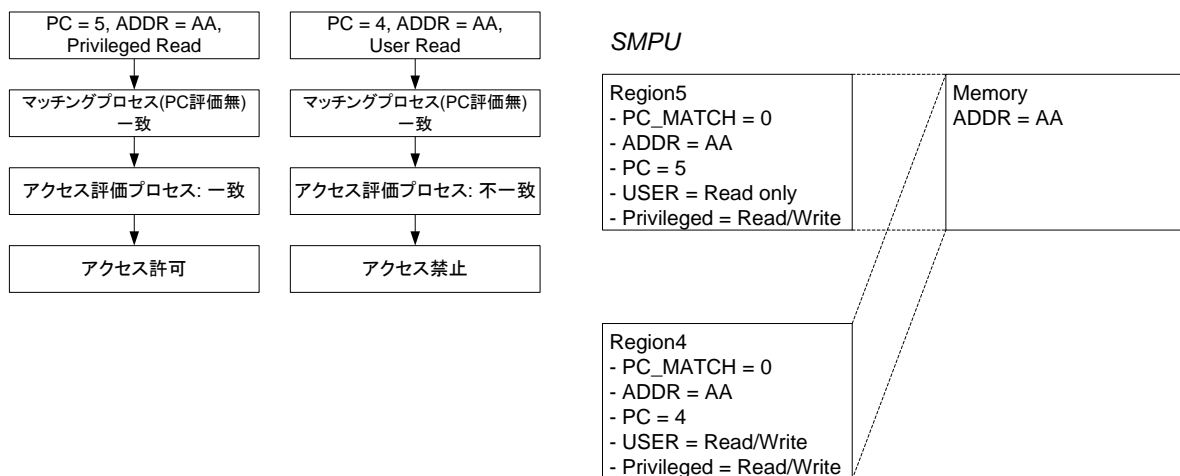
図 7 に、Region5 PC\_MATCH = "0"および Region4 PC\_MATCH = "0"時の動作例を示します。また表 2 に、各リージョンの設定を示します。



表 2. PC\_MATCH 動作リージョン設定 1

| Region  | PC_MATCH | Region Address | Protection Context | User       | Privileged |
|---------|----------|----------------|--------------------|------------|------------|
| Region4 | 0        | AA             | 4                  | Read/Write | Read/Write |
| Region5 | 0        | AA             | 5                  | Read only  | Read/Write |

図 7. PC\_MATCH 動作例 1



この場合、保護コンテキストはマッチングプロセスによって評価されません。したがって、PC = 4, PC = 5 の両アクセスとも Region5 のマッチングプロセスで「一致」します。次に、保護コンテキストは Region5 のアクセス評価プロセスによって評価され、PC = 5 のアクセスは許可されますが、PC = 4 のアクセスは許可されません。結果として、PC = 4 のアクセスは、Region4 よりも高いインデックスを持つ Region5 によって禁止されているため PC = 4 のアクセスはこのアドレスにアクセスできません。

#### ■ PC\_MATCH = 1 の場合

以下の疑似コードは PC\_MATCH = “1”の時の評価プロセスを示します。

```
match = 0;
for (i = n-1; i >= 0; i--) // n: number of protection regions
    if (Match ("transfer address", "protection context") {
        match = 1; break;
    }

if (match)
    AccessEvaluate ("access attributes", "protection context");
else
    "access allowed"
```

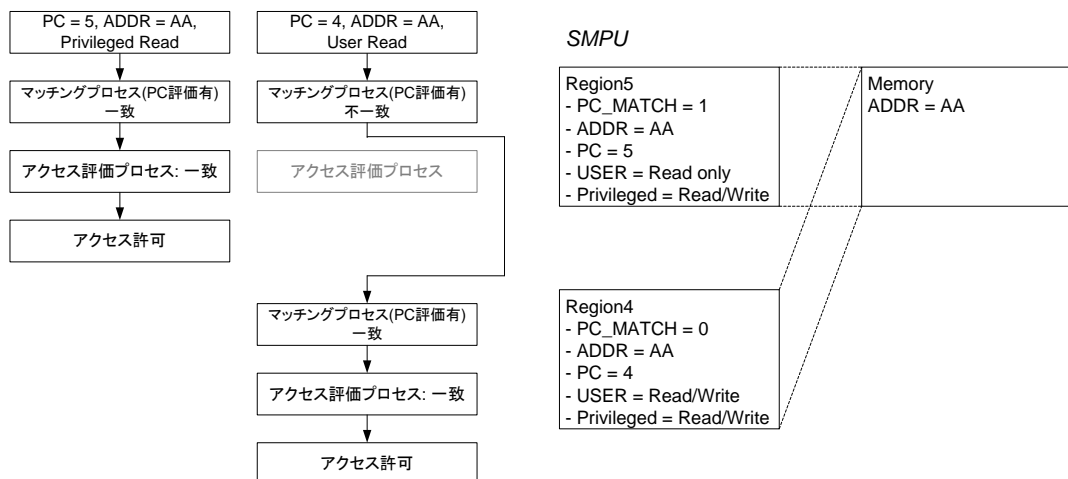
PC\_MATCH = “1”の場合、保護コンテキストはアクセス評価プロセスだけでなく、マッチングプロセスによっても評価されます。

図 8 に、Region5 PC\_MATCH = “1”および Region4 PC\_MATCH = “0”時の動作例を示します。また表 3 に、各リージョンの設定を示します。

表 3. PC\_MATCH 動作リージョン設定 2

| Region  | PC_MATCH | Region Address | Protection Context | User       | Privileged |
|---------|----------|----------------|--------------------|------------|------------|
| Region4 | 0        | AA             | 4                  | Read/Write | Read/Write |
| Region5 | 1        | AA             | 5                  | Read only  | Read/Write |

図 8. PC\_MATCH 動作例 2



この場合、保護コンテキストは、マッチングプロセスによっても評価されます。PC = 5 アクセスは、マッチングプロセスによって「一致」となり、アクセス評価プロセスによって評価されます。しかし、PC = 4 アクセスはマッチングプロセスによって「不一致」となり、次に優先順位の高いRegion4とのマッチングプロセスによって評価されます。PC = 4 は、マッチングプロセス後にアクセス評価プロセスによって評価され、その結果アクセスが許可されます。

PC\_MATCH を使用することで、保護コンテキストに応じて、同じアドレスに異なる属性を割り当てられます。

**Note:** PC\_MATCH は SMPU のみ提供されます。PPU は、すべての保護コンテキストにアクセス属性を割り当てられるためこの機能はサポートされていません。

### 3.6 マスタ識別子

各バスマスタは固有のマスタ識別子を持っています。この識別子は、保護ユニットのレジスタサフィックスや保護ユニットによるアクセス違反をしたマスタの識別に使用されます。表 4 に、各マスタのマスタ識別子を示します。

表 4. マスタ識別子

| マスタ識別子 | バスマスタ            |                  |                  |
|--------|------------------|------------------|------------------|
|        | CYT2 シリーズ        | CYT4B シリーズ       | CYT4D シリーズ       |
| 0      | CM0+ CPU         | CM0+ CPU         | CM0+ CPU         |
| 1      | CRYPTO Component | CRYPTO Component | CRYPTO Component |
| 2      | P-DMA 0          | P-DMA 0          | P-DMA 0          |
| 3      | P-DMA 1          | P-DMA 1          | P-DMA 1          |
| 4      | M-DMA            | M-DMA            | M-DMA            |
| 5      | -                | SDHC             | -                |
| 9      | -                | Ethernet 1       | Ethernet 1       |
| 10     | -                | Ethernet 0       | Ethernet 0       |
| 12     | -                | -                | Audio Subsystem  |
| 13     | -                | CM7_0 CPU        | CM7_0 CPU        |
| 14     | CM4F CPU         | CM7_1 CPU        | CM7_1 CPU        |
| 15     | Test Controller  | Test Controller  | Test Controller  |

### 3.7 保護違反

CPU の一部として実装される MPU は不一致および許可違反の場合、プログラマブル優先順位 MemManage フォールトまたはハードフォールトハンドラが呼び出されます。TCM 以外のアクセスで MPU 異常が発生した場合、そのアクセスに対する AXI または AHB への転送は実行されません。MPU の詳細については、[CM4](#)、[CM7](#)、および [CM0+](#) の Arm ドキュメンテーションセットを参照してください。

バスインフラストラクチャの一部として実装される MPU と SMPU が保護違反を引き起こすバス転送を検出した場合、そのバス転送はバスエラーになります。

PPU 保護違反となる書き込み転送を検出した場合、バッファリングがイネーブル(CPUSS\_BUFF\_CTL.WRITE\_BUFF = 1)されている場合は、バスマスタはバスエラーを認識しません。これは、バスインフラストラクチャ内の AHB-Lite ブリッジが書き込み転送をバッファリングし、OK 応答をマスタに返信するためです。この場合、システムは PPU の Fault report structure によって行われます。バッファリングがディセーブル(CPUSS\_BUFF\_CTL.WRITE\_BUFF = 0)の場合、PPU 保護違反となる書き込みアクセスはバスエラーになります。PPU 保護違反となる読出しアクセスは常にバスエラーになります。

保護ユニットに違反するバス転送は、ターゲットメモリまたはペリフェラルレジスタには到達しません。

バスインフラストラクチャの一部として実装されている MPU、SMPU、および PPU によって検出された保護違反は、Fault report structure に取込まれます。Fault report structure はエラーの発生を示す割込みを生成できます。加えて、違反したバス転送に関する以下の情報が Fault report structure に通知されます。

- 違反したアドレス
- 違反した属性
  - ユーザリード/ユーザライト/ユーザ実行
  - 特権リード/特権ライト/特権実行
  - 非セキュア
  - 保護コンテキスト
- 違反したマスタ識別子
- 違反を検出した保護ユニットまたは故障の種類(\*1)

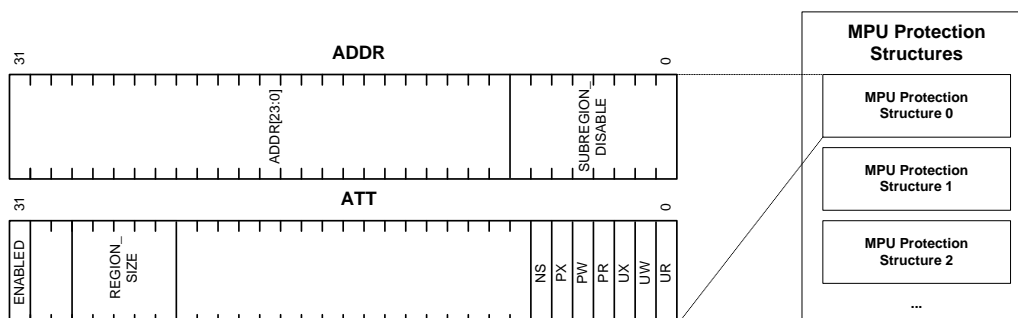
**Note:** (\*1) 検出された故障によって異なります。詳細は[レジスタ テクニカル リファレンス マニュアル \(TRM\)](#) を参照してください。

## 4 保護ユニット構造

### 4.1 MPU 構造

図 9 に、バスの一部として実装される MPU 構造について示します。CM4, CM7, および CM0+の一部として実装される MPU については、Arm ドキュメンテーションセット(CM4, CM7, および CM0+) を参照してください。

図 9. MPU 構造



MPU 保護は、マスタアクセスによって許可または制限されるプロパティを設定します。MPU は以下のプロパティを識別します。

- アドレス範囲
  - ADDR.ADDR: 領域のベースアドレスを指定します。
  - ATT.REGION\_SIZE: 領域のサイズを指定します。サイズの範囲は [256 B, 4 GB] です。
  - ADDR.SUBREGION\_DISABLE: 領域内の 8 つサブリージョンを個別に無効化します。
- アクセス属性
  - ATT.UR: ユーザリードアクセスの制御
  - ATT.UW: ユーザライトアクセスの制御
  - ATT.UX: ユーザ実行アクセスの制御
  - ATT.PR: 特権リードアクセスの制御
  - ATT.PW: 特権ライトアクセスの制御
  - ATT.PX: 特権実行アクセスの制御
  - ATT.NS: セキュアアクセスの制御
- 領域のイネーブル
  - ATT.ENABLED: 領域の有効/無効を制御

MPU は保護コンテキスト属性を持ちません。このタイプの MPU の定義は、一貫したソフトウェアインタフェースを保証するため、Arm MPU(メモリ領域とアクセス属性の定義) に従います。

領域は 8 つの等しいサイズの領域に分割できます。SUBREGION\_DISABLE によって領域内のサブリージョンは個別に有効化できます。

例えば、分割されたサブリージョンの SUBREGION\_DISABLE が 0x82(ビット 1 と 7 が"1") の場合、サブリージョン 1,7 は無効となり、サブリージョン 0, 2, 3, 4, 5, 6 が有効になります。表 5 に、開始アドレスが 0x10005400 で、範囲が 0x10005400 から 0x100055ff(512 バイト) の場合の、8 つのサブリージョンの有効/無効の状態を示します。

表 5. 各サブリージョン領域と状態

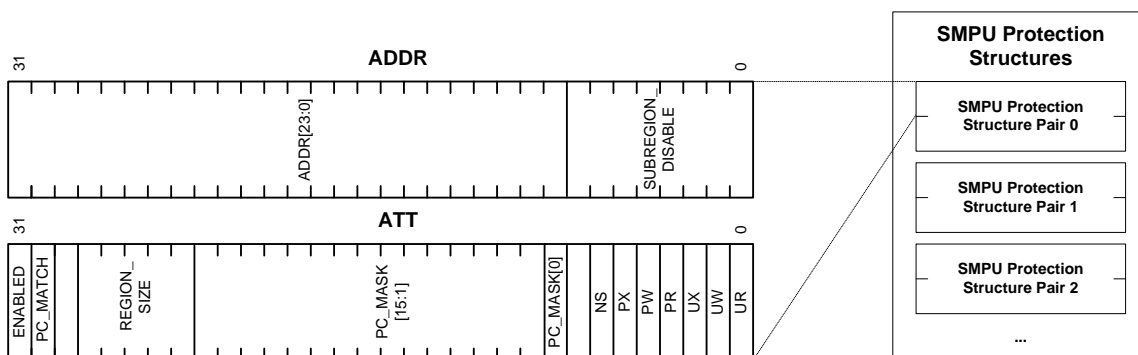
| Subregion   | Area                     | State |
|-------------|--------------------------|-------|
| Subregion 0 | 0x10005400 to 0x1000543f | 有効    |
| Subregion 1 | 0x10005440 to 0x1000547f | 無効    |
| Subregion 2 | 0x10005480 to 0x100054bf | 有効    |

| Subregion   | Area                     | State |
|-------------|--------------------------|-------|
| Subregion 3 | 0x100054c0 to 0x100054ff | 有効    |
| Subregion 4 | 0x10005500 to 0x1000553f | 有効    |
| Subregion 5 | 0x10005540 to 0x1000557f | 有効    |
| Subregion 6 | 0x10005580 to 0x100055bf | 有効    |
| Subregion 7 | 0x100055c0 to 0x100055ff | 無効    |

## 4.2 SMPU 構造

図 10 に、SMPU 構造を示します。

図 10. SMPU 構造



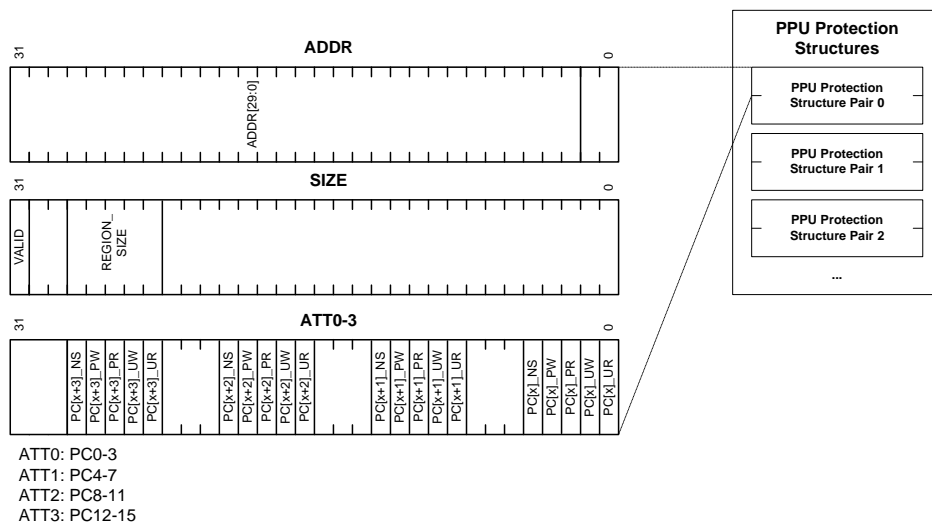
SMPU 保護構造は、マスタアクセスによって許可または制限されるプロパティを設定します。SMPU は次のプロパティを識別します。

- アドレス範囲
    - ADDR.ADDR: 領域のベースアドレスを指定します。
    - ATT.REGION\_SIZE: 領域のサイズを指定します。サイズの範囲は [256 B, 4 GB] です。
    - ADDR.SUBREGION\_DISABLE: 領域内の 8 つサブリージョンを個別に無効化します。
  - アクセス属性
    - ATT.UR: ユーザリードアクセスの制御
    - ATT.UW: ユーザライトアクセスの制御
    - ATT.UX: ユーザ実行アクセスの制御
    - ATT.PR: 特権リードアクセスの制御
    - ATT.PW: 特権ライトアクセスの制御
    - ATT.PX: 特権実行アクセスの制御
    - ATT.NS: セキュアアクセスの制御
    - ATT.PC\_MASK: 保護コンテキスト属性の制御  
The PC\_MASK[0]は、常に"1"です。言い換えれば、PC0 は常に許可されています。
    - ATT.PC\_MATCH: PC がマッチングプロセスまたはアクセス評価プロセスへの参加を制御します。詳細は [3.5.2](#) を参照してください。
  - 領域のイネーブル
    - ATT.ENABLED: 領域の有効/無効を制御
- SMPU のサブリージョン機能は MPU と同じです。

### 4.3 PPU 構造

図 11 に、PPU 構造を示します。

図 11. PPU 構造



PPU 保護構造は、マスタアクセスによって許可または制限されるプロパティを設定します。また、すべての保護コンテキストに対して独立して属性を設定できます。PPU は次のプロパティを識別します。

#### ■ アドレス範囲

- ADDR.ADDR: 領域のベースアドレスを指定します。
- SIZE.REGION\_SIZE: 領域のサイズを指定します。サイズの範囲は [4 B, 2 GB] です。

Fixed PPU 構造は、固定されたアドレス領域を持ちます。

#### ■ アクセス属性

- ATT.PCx\_UR: PCx のユーザーリードアクセスの制御
- ATT.PCx\_UW: PCx のユーザーライトアクセスの制御
- ATT.PCx\_PR: PCx の特権リードアクセスの制御
- ATT.PCx\_PW: PCx の特権ライトアクセスの制御
- ATT.PCx\_NS: PCx のセキュアアクセスの制御

#### ■ 領域のイネーブル

- SIZE.VALID: 領域の有効/無効を制御

**Note:** 本シリーズでは、保護コンテキストは 0~7 まですをサポートします。したがって、ATT0, 1 レジスタのみ存在します。

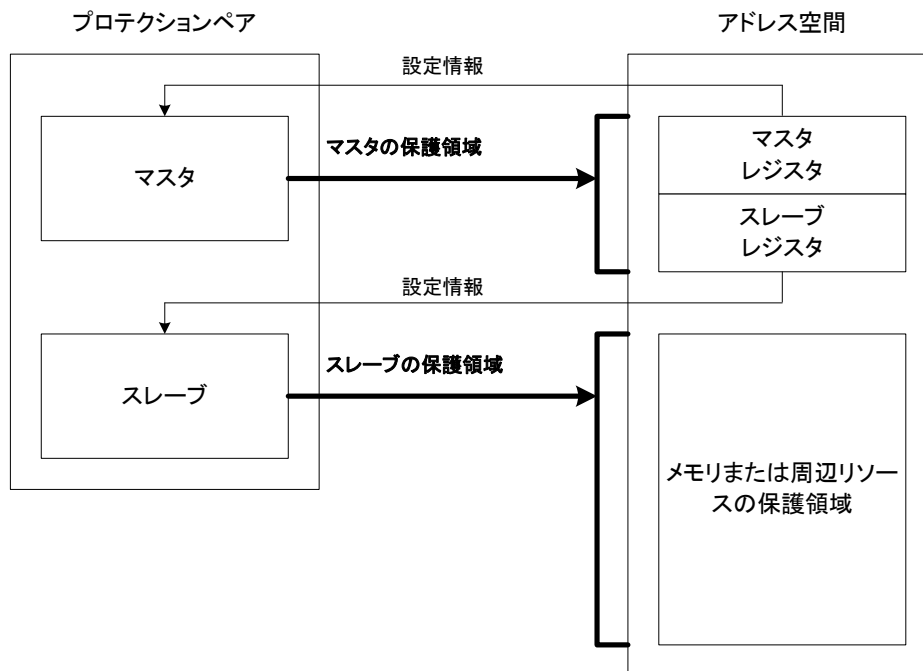
### 4.4 保護ペア構造

保護ユニットのレジスタは、他の周辺機能と同じレジスタです。さらに、保護構造のレジスタは、周辺機能のレジスタと同様に他の保護構造のアドレス範囲に含められます。したがって保護構造は、保護構造によって保護できます。

保護構造を保護する保護構造をマスタと呼び、マスタによって保護される保護構造をスレーブと呼びます。スレーブは周辺機能を保護します。

スレーブとマスタの保護構造を保護ペアと呼びます。SMPU と PPU には、この保護ペアがあります。図 12 に、保護ペアの構造を示します。

図 12. 保護ペア構造



スレーブを保護するマスタには以下の特長があります。

- アドレス範囲
  - ADDR.ADDR: 読出し専用です。固定されたアドレス領域です。
  - ATT.REGION\_SIZE: 読出し専用です。固定されたアドレス領域です。
  - ADDR.SUBREGION\_DISABLE: 読出し専用です。固定されたアドレス領域です。
- アクセス属性
  - ATT.UR: “1” 固定です。ユーザリードは常に許可されています。
  - ATT.UW: ユーザライトアクセスの制御
  - ATT.UX: “0” 固定です。ユーザ実行アクセスは許可されません。
  - ATT.PR: “1” 固定です。特権リードは常に許可されています。
  - ATT.PW: 特権ライトアクセスの制御
  - ATT.PX: “0” 固定です。特権実行アクセスは許可されません。
  - ATT.NS: セキュアアクセスの制御

上記は SMPU マスタの例です。

マスタの保護領域は固定されています。読出しアクセスは常に許可されています。実行アクセスは許可されません。SMPU の場合、有効/無効に設定できますが、PPU では無効にできません。

## 5 保護ユニットの設定例

保護ユニットの使用例を、使用仮定に従って説明します。

**Note:** ここで示すアドレスや周辺チャネル番号は CYT2B シリーズのものです。実際のアドレスと周辺チャネル番号は、[テクニカルリファレンスマニュアル](#)を参照してください。

### 5.1 CPUの一部として実装される MPU の設定例

ここでは、オペレーティングシステム(OS)が使用する領域を、タスクアクセスから保護する方法および MPU の設定例を示します。

#### 5.1.1 使用例

ここでは、CM4 で実行される OS とタスクソフトウェアの分離を示します。OS は特権レベル、タスクはユーザレベルでアクセスします。MPU は、ユーザ/特権、リード/ライトおよび実行アクセスを区別します。表 6 に、MPU のアクセス制限を示します。

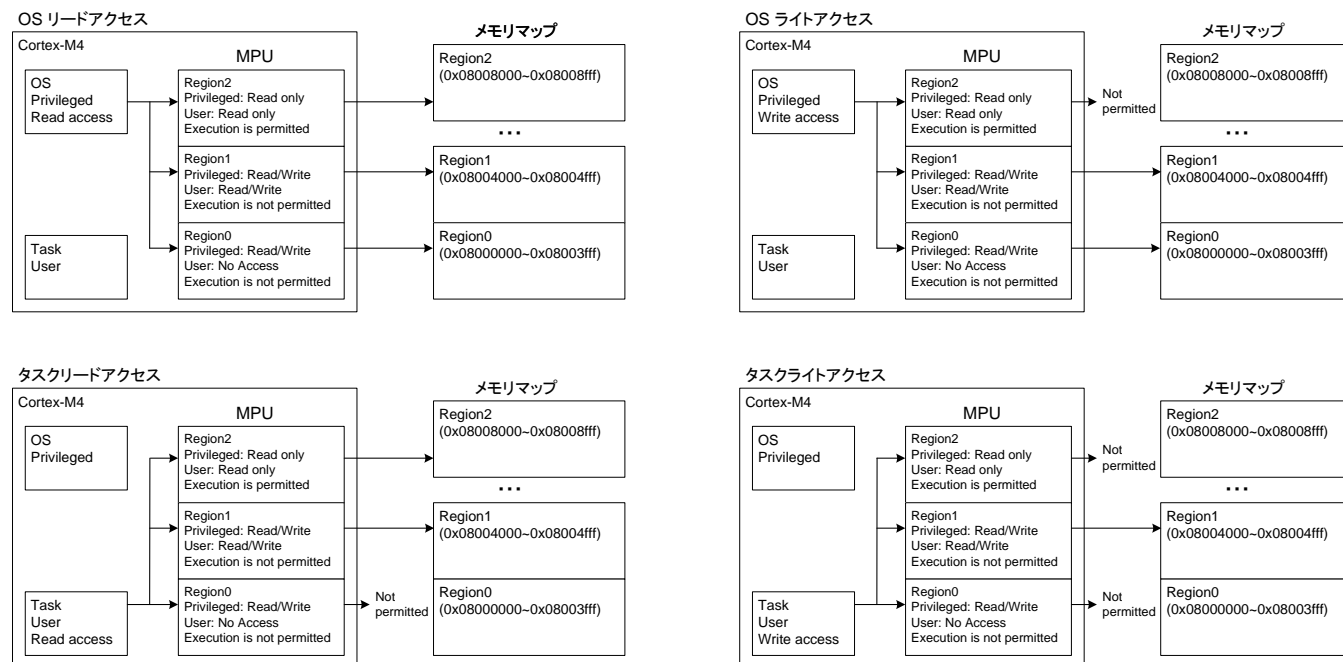
表 6. CPU の一部として実装される MPU のアクセス制限例

| Region                                      | Attribute                                |
|---|--|
| Region0<br>0x08000000 to 0x08003fff (16 KB) | 特権: リード/ライト<br>ユーザ: アクセス不可<br>実行アクセスは禁止  |
| Region1<br>0x08004000 to 0x08004fff (4 KB)  | 特権: リード/ライト<br>ユーザ: リード/ライト<br>実行アクセスは禁止 |
| Region2<br>0x08008000 to 0x08008fff (4 KB)  | 特権: リード専用<br>ユーザ: リード専用<br>実行アクセスは許可     |

#### 5.1.2 動作概要

図 13 に、OS およびタスクのアクセス制限を示します。

図 13. OS/タスクアクセスの動作





Region0 は、OS からのみリード/ライトが許可され、タスクアクセスは禁止されます。タスクから分離された領域です。

Region1 は、OS/タスクいずれもアクセスが許可されます。両ソフトウェアの共有領域として使用できます。

Region2 は、OS/タスクいずれのソフトウェアもリードアクセスは許可されますが、ライトアクセスは禁止されます。この領域はフェッチされた命令の実行が許可されているためコード領域として使用できます。

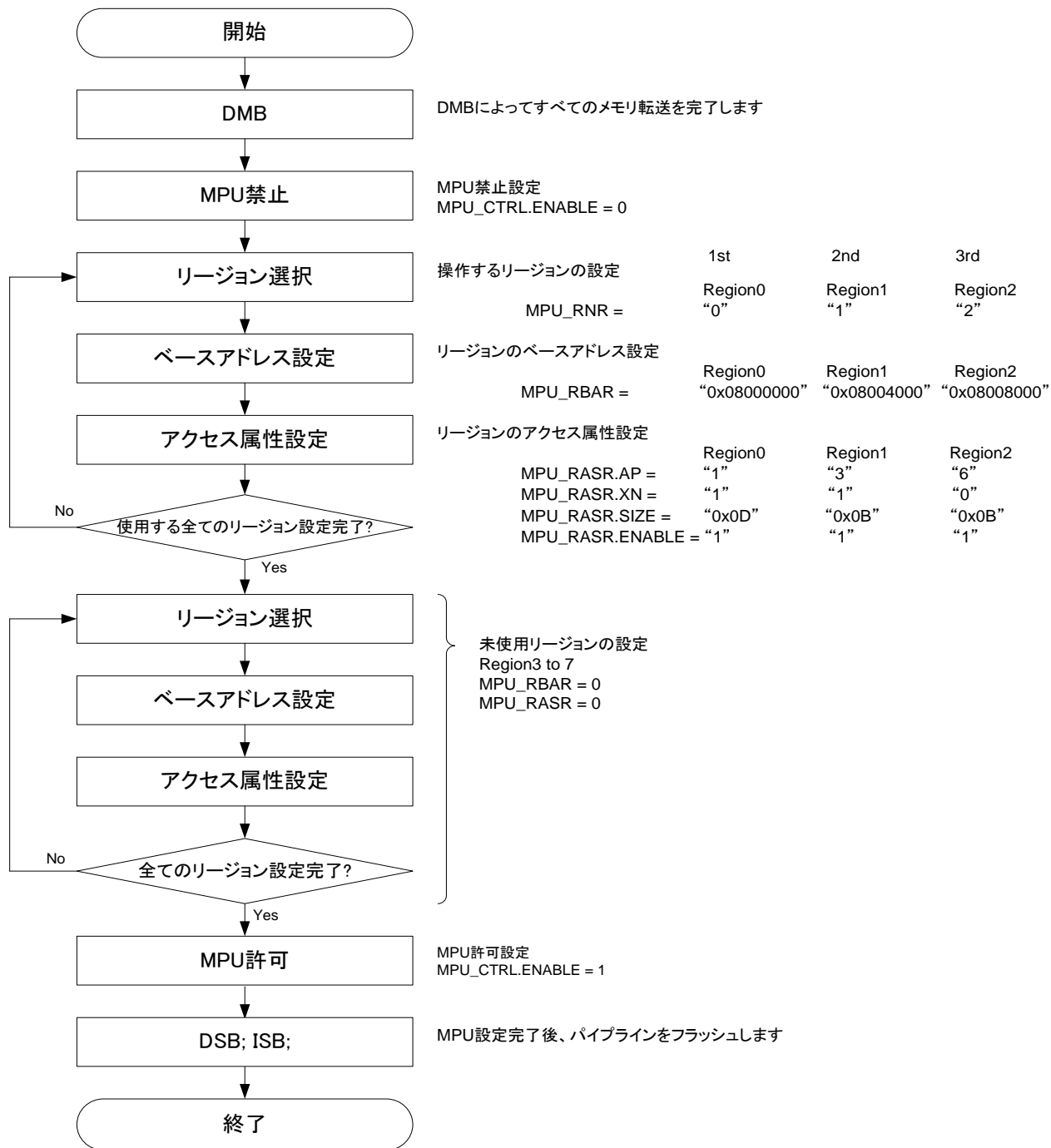
CPU の一部として実装される MPU が有効の時、未設定領域へのアクセスはアクセス違反の要因となります。意図しないアクセス違反を防止するため、このタイプの MPU はオーバラッピングをサポートしています。これにより、大きい番号を持つ両機の優先順位が最も高く、小さい番号 (Region 0) の優先順位が最も低くなります。Region 0 とオーバラップする領域は、Region 0 よりも高い優先順位を持ちます。つまり、Region 0 はバックグラウンド領域 (全領域の設定) として使用できます。

MPU の詳細については、[CM4](#)、[CM7](#)、および [CM0+](#) の Arm ドキュメンテーションセットを参照してください。

### 5.1.3 設定手順

このタイプの MPU は、CPU 固有のレジスタによって設定されます。設定中は MPU を無効にする必要があり、特権レベルでのアクセスが必要です。[図 14](#) に、MPU の設定手順例を示します。

図 14. CPU の一部として実装される MPU の設定手順例



MPU は、設定中は無効にする必要があります。次に、MPU\_RNR レジスタによって設定するリージョンを指定します。MPU\_RNR を設定後、MPU\_RBAR, MPU\_RASR レジスタによってベースアドレス、サイズ、アクセス属性を設定します。各リージョンに対して、この設定を繰り返します。また、未使用リージョンの MPU\_RBAR および MPU\_RASR レジスタは"0"に設定します。最後に MPU 許可を設定します。

また、MPU\_RBAR レジスタにより設定する領域番号の、直接指定もできます。MPU\_RASR を使用して、Normal memory, Strongly-ordered および Device といったメモリ属性も設定できます。

詳細は、Arm ドキュメンテーションセット(CM4, CM7, および CM0+) を参照してください。

## 5.2 バスの一部として実装される MPU の設定例

この MPU は、MPU.ADDR および MPU.ATT レジスタによって設定され、テストコントローラによって使用されます。ただし、通常この MPU はセキュリティ要件に応じてセキュア CPU である CM0+により設定されます。

この MPU はブートプロセスで設定されます。

## 5.3 SMPU の設定例

SMPU はメモリ保護機能を提供し、すべてのバスマスタによって共有され、すべてのバスマスタは各リージョンに同じ制限を受けます。

SMPU はマスタ/スレーブによる保護ペア構造を持っています。したがって、スレーブへの設定はマスタの設定によって制限されます。

### 5.3.1 使用例

SMPU は、ユーザ/特権、セキュア/非セキュアおよび保護コンテキストを区別します。

- CM0+アクセスは PC6 を使用します。
- CM4 アクセスは PC7 を使用します。
- 各マスタは PC4 および特権による書込みが必要です。

表 7 に、使用例での SMPU のアクセス制限の例を示します。

表 7. SMPU のアクセス制限例

| Region                                       | Privileged                               | User                                     | Secure     | Allowed Protection Context | PC_MATCH          | Resources |
|--|--|--|------------|----------------------------|-------------------|-----------|
| Region0<br>0x08000000 to 0x08007fff (32 KB)  | Read/Write<br>Execution is permitted     | No Access<br>Execution is not permitted  | Secure     | PC = 6                     | Access evaluation | SRAM0     |
| Region1<br>0x08008000 to 0x0800ffff (32 KB)  | Read/Write<br>Execution is permitted     | Read/Write<br>Execution is not permitted | Non-secure | PC = 7                     | Access evaluation | SRAM0     |
| Region2<br>0x08010000 to 0x08013fff (16 KB)  | Read/Write<br>Execution is not permitted | Read/Write<br>Execution is not permitted | Non-secure | PC = 6                     | Access evaluation | SRAM1     |
| Region3<br>0x08010000 to 0x08013fff (16 KB)  | Read/Write<br>Execution is not permitted | Read only<br>Execution is not permitted  | Non-secure | PC = 7                     | Matching          | SRAM1     |
| Region4<br>0x08014000 to 0x08017fff (16 KB)  | Read/Write<br>Execution is not permitted | Read/Write<br>Execution is not permitted | Non-secure | PC = 6,7                   | Access evaluation | SRAM1     |
| Region5<br>0x10000000 to 0x1001ffff (128 KB) | Read/Write<br>Execution is permitted     | Read/Write<br>Execution is permitted     | Secure     | PC = 6                     | Access evaluation | FLASH     |
| Region6<br>0x10040000 to 0x1007ffff (256 KB) | Read/Write<br>Execution is permitted     | Read/Write<br>Execution is permitted     | Non-secure | PC = 7                     | Access evaluation | FLASH     |

Region0 は、CM0+の専用 RAM 領域です。セキュアおよび特権アクセスが必要です。

Region1 は、CM4 の専用 RAM 領域です。セキュアアクセスが必要です。ユーザ/特権レベルでアクセスできます。

Region2 および Region3 は、特別な領域です。CM0+はユーザ/特権レベルいずれもアクセスできますが、CM4 は特権レベルではアクセス可能、ユーザレベルでは読み出し専用です。PC\_MATCH を使用して同じ領域に PC ごとに属性を変更できます。

Region4 は CM0+と CM4 の共有 RAM 領域です。

Region5 は CM0+専用の flash 領域、Region6 は CM4 専用の Flash 領域です。

### 5.3.2 使用想定での SMPU 設定値

表 8 に、上記の使用想定を実現するための SMPU レジスタ (SMPU\_STRUCTx\_ADD0 および SMPU\_STRUCTx\_ATT0) の設定値を示します。レジスタの詳細は、4.2 を参照してください。

SMPU は、マスタとスレーブの保護ペア構造を持ちます。これらの設定はスレーブに対して行われます。

表 9 に、使用例でのマスタ設定の例を示します。

**Note:** 'x'は、リージョン番号を示します。このシリーズでは、最大 16 のリージョンを設定できます。

表 8. SMPU スレーブ設定値

| Region  | SMPU_STRUCTx_ADD0 |            | SMPU_STRUCTx_ATT0 |    |    |    |    |    |    |                     |          |             |
|---------|-------------------|------------|-------------------|----|----|----|----|----|----|---------------------|----------|-------------|
|         | SUBREGION_DISABLE | ADDR       | UR                | UW | UX | PR | PW | PX | NS | PC_MASK [15:0] (*1) | PC_MATCH | REGION_SIZE |
| Region0 | 0x00              | 0x08000000 | 0                 | 0  | 0  | 1  | 1  | 1  | 0  | 0x0041              | 0        | 14          |
| Region1 | 0x00              | 0x08008000 | 1                 | 1  | 0  | 1  | 1  | 1  | 1  | 0x0081              | 0        | 14          |
| Region2 | 0x00              | 0x08010000 | 1                 | 1  | 0  | 1  | 1  | 0  | 1  | 0x0041              | 0        | 13          |
| Region3 | 0x00              | 0x08010000 | 1                 | 0  | 0  | 1  | 1  | 0  | 1  | 0x0081              | 1        | 13          |
| Region4 | 0x00              | 0x08014000 | 1                 | 1  | 0  | 1  | 1  | 0  | 1  | 0x00c1              | 0        | 13          |
| Region5 | 0x00              | 0x10000000 | 1                 | 1  | 1  | 1  | 1  | 1  | 0  | 0x0041              | 0        | 16          |
| Region6 | 0x00              | 0x10040000 | 1                 | 1  | 1  | 1  | 1  | 1  | 1  | 0x0081              | 0        | 17          |

**Note:** (\*1) PC\_MASK[0]は常に"1"です。

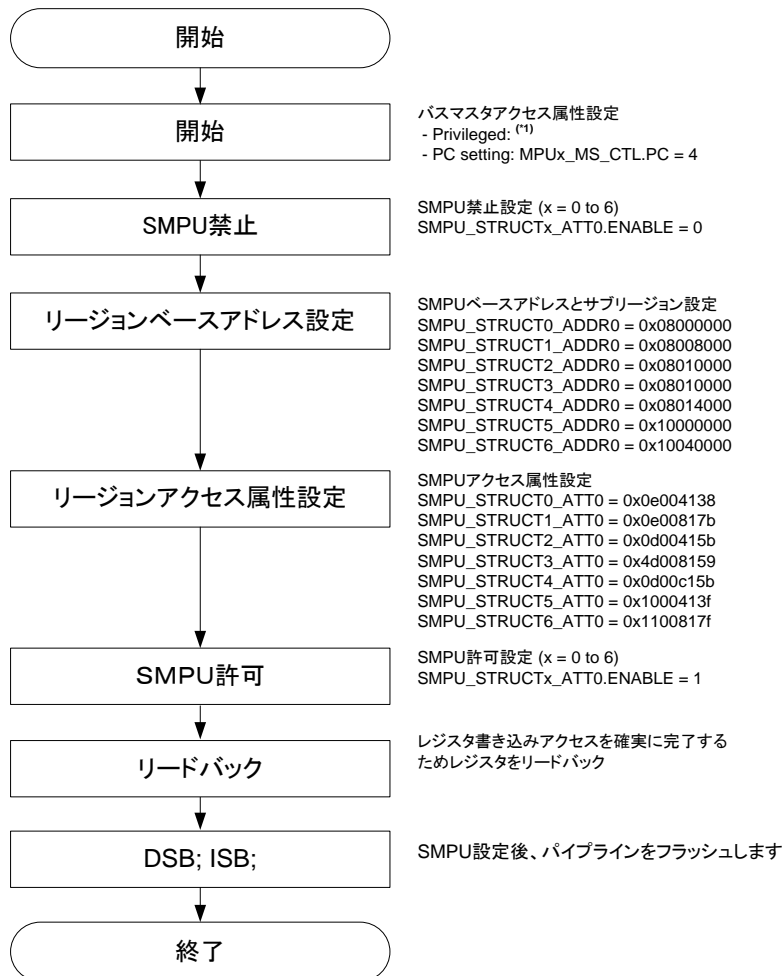
表 9. SMPU マスタ設定値

| SMPU_STRUCTx_ATT1 (x=0 to 6) |    |          |          |    |          |    |         |          |             |
|------------------------------|----|----------|----------|----|----------|----|---------|----------|-------------|
| UR                           | UW | UX       | PR       | PW | PX       | NS | PC_MASK | PC_MATCH | REGION_SIZE |
| 1(Fixed)                     | 0  | 0(Fixed) | 1(Fixed) | 1  | 0(Fixed) | 1  | 0x0011  | 0        | 7(Fixed)    |

### 5.3.3 SMPU の設定手順

図 15 に、設定手順例を示します。

図 15. SMPU の設定手順例



設定値については表 8 を参照してください。

スレーブ(SMPU\_STRUCTx\_ADDR0 and SMPU\_STRUCTx\_ATT0)を設定する場合のアクセス属性は、マスタ(SMPU\_STRUCTx\_ADDR1 and SMPU\_STRUCTx\_ATT1)の設定によって許可されます。

SMPU の設定後、レジスタのライトアクセスを確実に完了させるためレジスタをリードバックする必要があります。

**Note:** \*1: 特権属性への変更は、3.3 章を参照してください。

## 5.4 PPU の設定例

PPU は、周辺機能への保護機能を提供し、すべてのバスマスタによって共有されます。

一般的に、周辺機能のレジスタアドレスは固定されています。したがって、PPU には固定保護アドレス範囲をもつ Fixed PPU とプログラム可能な保護アドレス範囲を持つ Programmable PPU の 2 種類があります。通常、プログラム可能な PPU 保護領域であるベースアドレスとサイズは、ブートプロセスの PC0 によって設定されます。

ここでは、Fixed PPU の設定方法について説明します。Fixed PPU は、保護されたアドレス領域が固定されていることを除いて Programmable PPU と同じです。

PPU には、マスタ/スレーブの保護ペア構造を持ちます。したがって、スレーブの属性設定はマスタの設定によって制限されています。

#### 5.4.1 使用例

PPU は、ユーザ/特権、セキュア/非セキュアおよび保護コンテキストを区別します。

- CM0+アクセスは PC6 を使用します。
- CM4 アクセスは PC7 を使用します。

表 10 に、使用例での PPU のアクセス制限の例を示します。

表 10. PPU のアクセス制限例

| Setting PPU | Specify Protected Area             | Privileged | User       | Secure     | Protection Context |
|-------------|------------------------------------|------------|------------|------------|--------------------|
| PPU_FX0     | PPU_FX0_SL_ADDR<br>PPU_FX0_SL_SIZE | Read/Write | No Access  | Secure     | PC = 6             |
|             |                                    | No Access  | No Access  |            | PC = 7             |
| PPU_FX1     | PPU_FX1_SL_ADDR<br>PPU_FX1_SL_SIZE | No Access  | No Access  | Non-secure | PC = 6             |
|             |                                    | Read/Write | Read/Write |            | PC = 7             |
| PPU_FX2     | PPU_FX2_SL_ADDR<br>PPU_FX2_SL_SIZE | Read/Write | Read/Write | Non-secure | PC = 6             |
|             |                                    | Read-only  | No Access  |            | PC = 7             |
| PPU_FX3     | PPU_FX3_SL_ADDR<br>PPU_FX3_SL_SIZE | Read/Write | Read-only  | Non-secure | PC = 6             |
|             |                                    | Read/Write | Read-only  |            | PC = 7             |

PPU\_FX0 は、CM0+からの特権レベルでのアクセスのみが許可されます。さらに、アクセスにはセキュア属性が必要です。CM4 のアプリケーションソフトウェアは、この領域にはアクセスできません。

PPU\_FX1 は、CM4 からのアクセスのみ許可されます。この領域はアプリケーションソフトウェア専用の領域です。

PPU\_FX2 は、CM0+がアクセスできる領域です。CM4 は特権レベルでのアクセスによる読出しのみ許可されます。

PPU\_FX3 は、CM0+と CM4 の両方が特権レベルでのみアクセスできる領域です。

**Note:** 保護領域は固定されています。実際に保護するアドレスとペリフェラルチャネル番号については[テクニカルリファレンスマニュアル \(TRM\)](#) を参照してください。

#### 5.4.2 使用想定での PPU 設定値

PPU は、保護コンテキストごとに異なる設定を付与できます。PPU\_FXx\_SL\_ATT1 レジスタは、使用想定で使用する PC6, 7 の属性設定に使用します。レジスタの詳細は [4.3](#) を参照してください。

表 11 に、上記の使用想定を実現する PPU レジスタの設定値を示します。

表 11. PPU 設定値

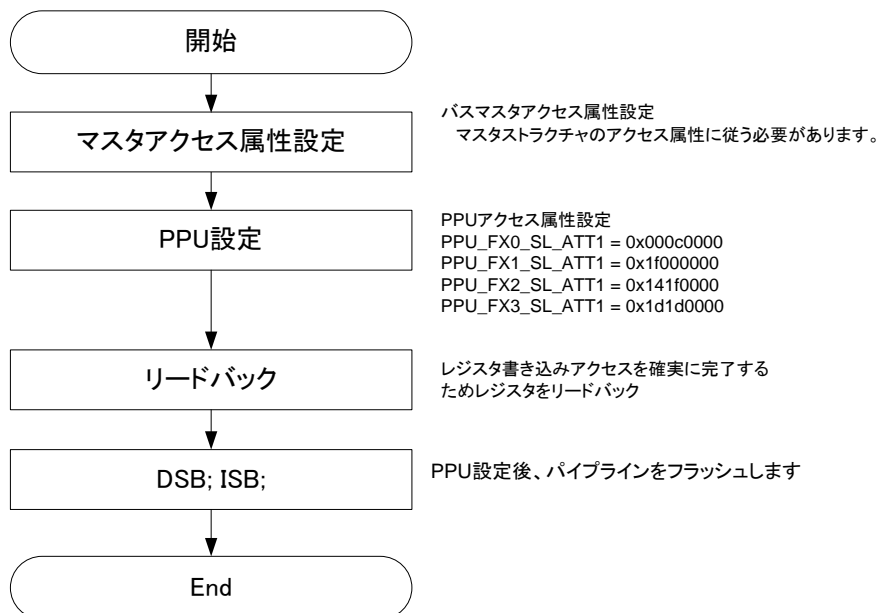
| PPU     | PPU_FXx_SL_ATT1 |    |    |    |    |     |    |    |    |    |     |    |    |    |    |     |    |    |    |    |
|---------|-----------------|----|----|----|----|-----|----|----|----|----|-----|----|----|----|----|-----|----|----|----|----|
|         | PC4             |    |    |    |    | PC5 |    |    |    |    | PC6 |    |    |    |    | PC7 |    |    |    |    |
|         | UR              | UW | PR | PW | NS | UR  | UW | PR | PW | NS | UR  | UW | PR | PW | NS | UR  | UW | PR | PW | NS |
| PPU_FX0 | 0               | 0  | 0  | 0  | 0  | 0   | 0  | 0  | 0  | 0  | 0   | 0  | 1  | 1  | 0  | 0   | 0  | 0  | 0  | 0  |
| PPU_FX1 | 0               | 0  | 0  | 0  | 0  | 0   | 0  | 0  | 0  | 0  | 0   | 0  | 0  | 0  | 0  | 1   | 1  | 1  | 1  | 1  |
| PPU_FX2 | 0               | 0  | 0  | 0  | 0  | 0   | 0  | 0  | 0  | 0  | 1   | 1  | 1  | 1  | 1  | 0   | 0  | 1  | 0  | 1  |
| PPU_FX3 | 0               | 0  | 0  | 0  | 0  | 0   | 0  | 0  | 0  | 0  | 1   | 0  | 1  | 1  | 1  | 1   | 0  | 1  | 1  | 1  |

**Note:** PPU は、PX および UX 属性をサポートしません。したがって、周辺機能アドレスからの実行は許可されません。(PC0 はアクセス制限がないためこの実行制限はありません)

### 5.4.3 PPU の設定手順

図 16 に、PPU の設定手順例を示します。

図 16. PPU の設定手順例



設定値については、表 11 を参照してください。

スレーブ設定(PPU\_FXx\_SL\_ATT0)のアクセス属性は、マスタの設定属性 (PPU\_FXx\_MS\_ATT0) によって許可されます。

SMPU の設定後、レジスタのライトアクセスを確実に完了させるためレジスタをリードバックする必要があります。

## 6 用語集

| 用語     | 説明   |
|--------|--|
| MPU    | Memory Protection Unit (メモリ保護ユニット)   |
| SMPU   | Shared Memory Protection Unit (共有メモリ保護ユニット)  |
| PPU    | Peripheral Protection Unit (周辺機能保護ユニット)  |
| PC     | Protection Contexts (保護コンテキスト)。詳細はアーキテクチャ TRM の Direct Memory Access “Protection Context”章を参照してください。 |
| DMB    | Data memory barrier。この命令は Thumb 命令セットです。   |
| DSB    | Data Synchronization Barrier。この命令は Thumb 命令セットです。  |
| ISB    | Instruction Synchronization Barrier。この命令は Thumb 命令セットです。   |
| P-DMA  | Peripheral DMA。詳細はアーキテクチャ TRM の Direct Memory Access 章を参照してください。                                     |
| M-DMA  | Memory DMA。詳細はアーキテクチャ TRM の Direct Memory Access 章を参照してください。   |
| CRYPTO | Cryptography Component (暗号コンポーネント)。詳細はアーキテクチャ TRM の Cryptography Block 章を参照してください。                   |

## 7 関連ドキュメント

以下は、Traveo II ファミリシリーズのデータシートとテクニカルリファレンスマニュアルです。これらの資料を入手する場合は[テクニカルサポート](#)にお問い合わせください。

- デバイスデータシート
  - CYT2B7 Datasheet 32-Bit Arm® Cortex®-M4F Microcontroller Traveo™ II Family
  - CYT2B9 Datasheet 32-Bit Arm® Cortex®-M4F Microcontroller Traveo™ II Family
  - CYT4BF Datasheet 32-Bit Arm® Cortex®-M7 Microcontroller Traveo™ II Family
  - CYT4DN Datasheet 32-Bit Arm® Cortex®-M7 Microcontroller Traveo™ II Family
  - CYT3BB/4BB Datasheet 32-Bit Arm® Cortex®-M7 Microcontroller Traveo™ II Family
- Body Controller Entry ファミリ
  - Traveo™ II Automotive Body Controller Entry Family Architecture Technical Reference Manual (TRM)
  - Traveo™ II Automotive Body Controller Entry Registers Technical Reference Manual (TRM) for CYT2B7
  - Traveo™ II Automotive Body Controller Entry Registers Technical Reference Manual (TRM) for CYT2B9
- Body Controller High ファミリ
  - Traveo™ II Automotive Body Controller High Family Architecture Technical Reference Manual (TRM)
  - Traveo™ II Automotive Body Controller High Registers Technical Reference Manual (TRM) for CYT4BF
  - Traveo™ II Automotive Body Controller High Registers Technical Reference Manual (TRM) for CYT3BB/4BB
- Cluster 2D ファミリ
  - Traveo™ II Automotive Cluster 2D Family Architecture Technical Reference Manual (TRM)
  - Traveo™ II Automotive Cluster 2D Registers Technical Reference Manual (TRM)



## 改版履歴

文書名: AN219843 - Traveo II ファミリの保護ユニット設定

文書番号: 002-23843

| 版  | ECN     | 発行日        | 変更内容   |
|----|---------|------------|--|
| ** | 6197450 | 06/05/2018 | これは英語版 002-19843 Rev. **を翻訳した日本語版 002-23843 Rev. **です。   |
| *A | 6397877 | 12/05/2018 | これは英語版 002-19843 Rev. *A を翻訳した日本語版 002-23843 Rev. *A です。 |
| *B | 6502503 | 03/06/2019 | これは英語版 002-19843 Rev. *B を翻訳した日本語版 002-23843 Rev. *B です。 |
| *C | 6652951 | 08/20/2019 | これは英語版 002-19843 Rev. *C を翻訳した日本語版 002-23843 Rev. *C です。 |
| +D | 6853176 | 04/15/2020 | これは英語版 002-19843 Rev. *D を翻訳した日本語版 002-23843 Rev. *D です。 |

## ワールドワイド販売と設計サポート

サイプレスは、事業所、ソリューション センター、メーカー代理店、および販売代理店の世界的なネットワークを保持しています。お客様の最寄りのオフィスについては、[サイプレスのロケーション ページ](#)をご覧ください。

### 製品

|                               |  |
|-------------------------------|--|
| Arm® Cortex® Microcontrollers | <a href="http://cypress.com/arm">cypress.com/arm</a>               |
| 車載用                           | <a href="http://cypress.com/automotive">cypress.com/automotive</a> |
| クロック&バッファ                     | <a href="http://cypress.com/clocks">cypress.com/clocks</a>         |
| インターフェース                      | <a href="http://cypress.com/interface">cypress.com/interface</a>   |
| IoT(モノのインターネット)               | <a href="http://cypress.com/iot">cypress.com/iot</a>               |
| メモリ                           | <a href="http://cypress.com/memory">cypress.com/memory</a>         |
| マイクロコントローラ                    | <a href="http://cypress.com/mcu">cypress.com/mcu</a>               |
| PSoC                          | <a href="http://cypress.com/psoc">cypress.com/psoc</a>             |
| 電源用 IC                        | <a href="http://cypress.com/pmuc">cypress.com/pmuc</a>             |
| タッチ センシング                     | <a href="http://cypress.com/touch">cypress.com/touch</a>           |
| USB コントローラー                   | <a href="http://cypress.com/usb">cypress.com/usb</a>               |
| ワイヤレス                         | <a href="http://cypress.com/wireless">cypress.com/wireless</a>     |

### PSoC® ソリューション

[PSoC 1](#) | [PSoC 3](#) | [PSoC 4](#) | [PSoC 5LP](#) | [PSoC 6 MCU](#)

### サイプレス開発者コミュニティ

[コミュニティ](#) | [サンプルコード](#) | [Projects](#) | [ビデオ](#) | [ブログ](#) | [トレーニング](#) | [Components](#)

### テクニカルサポート

[cypress.com/support](http://cypress.com/support)

Arm and Cortex are registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere.



© Cypress Semiconductor Corporation, 2018-2020. 本書面は、Cypress Semiconductor Corporation 及び Spansion LLC を含むその子会社 (以下「Cypress」という。) に帰属する財産である。本書面 (本書面に含まれ又は言及されているあらゆるソフトウェア若しくはファームウェア (以下「本ソフトウェア」という。)) を含む) は、アメリカ合衆国及び世界のその他の国における知的財産法令及び条約に基づき Cypress が所有する。Cypress はこれらの法令及び条約に基づく全ての権利を留保し、本段落で特に記載されているものを除き、その特許権、著作権、商標権又はその他の知的財産権のライセンスを一切許諾しない。本ソフトウェアにライセンス契約書が伴っておらず、かつ Cypress との間で別途本ソフトウェアの使用方法を定める書面による合意がない場合、Cypress は、(1) 本ソフトウェアの著作権に基づき、(a) ソースコード形式で提供されている本ソフトウェアについて、Cypress ハードウェア製品と共に用いるためにのみ、かつ組織内部でのみ、本ソフトウェアの修正及び複製を行うこと、並びに (b) Cypress のハードウェア製品ユニットに用いるためにのみ、(直接又は再販売者及び販売代理店を介して間接のいずれかで) 本ソフトウェアをバイナリーコード形式で外部エンドユーザーに配布すること、並びに (2) 本ソフトウェア (Cypress により提供され、修正がなされていないもの) が抵触する Cypress の特許権のクレームに基づき、Cypress ハードウェア製品と共に用いるためにのみ、本ソフトウェアの作成、利用、配布及び輸入を行うことについての非独占的で譲渡不能な一専属的ライセンス (サブライセンスの権利を除く) を付与する。本ソフトウェアのその他の使用、複製、修正、変換又はコンパイルを禁止する。

**適用される法律により許される範囲内で、Cypress は、本書面又はいかなる本ソフトウェア若しくはこれに伴うハードウェアに関しても、明示又は黙示を問わず、いかなる保証 (商品性及び特定の目的への適合性の黙示の保証を含むがこれらに限られない) も行わない。**いかなるコンピューティングデバイスも絶対に安全ということはない。従って、Cypress のハードウェアまたはソフトウェア製品に講じられたセキュリティ対策にもかかわらず、Cypress は、Cypress 製品への権限のないアクセスまたは使用といったセキュリティ違反から生じる一切の責任を負わない。加えて、本書面に記載された製品には、エラーと呼ばれる設計上の欠陥またはエラーが含まれている可能性があり、公表された仕様とは異なる動作をする場合がある。適用される法律により許される範囲内で、Cypress は、別途通知することなく、本書面を変更する権利を留保する。Cypress は、本書面に記載のある、いかなる製品若しくは回路の適用又は使用から生じる一切の責任を負わない。本書面で提供されたあらゆる情報 (あらゆるサンプルデザイン情報又はプログラムコードを含む) は、参照目的のためのみに提供されたものである。この情報で構成するあらゆるアプリケーション及びその結果としてのあらゆる製品の機能性及び安全性を適切に設計、プログラム、かつテストすることは、本書面のユーザーの責任において行われるものとする。Cypress 製品は、兵器、兵器システム、原子力施設、生命維持装置若しくは生命維持システム、蘇生用の設備及び外科的移植を含むその他の医療機器若しくは医療システム、汚染管理若しくは有害物質管理の運用のために設計され若しくは意図されたシステムの重要な構成部分としての使用、又は装置若しくはシステムの不具合が人身傷害、死亡若しくは物的損害を生じさせるようなその他の使用 (以下「本目的外使用」という。) のためには設計、意図又は承認されていない。重要な構成部分とは、その不具合が装置若しくはシステムの不具合を生じさせるか又はその安全性若しくは実効性に影響すると合理的に予想できるような装置若しくはシステムのあらゆる構成部分をいう。Cypress 製品のあらゆる本目的外使用から生じ、若しくは本目的外使用に関連するいかなる請求、損害又はその他の責任についても、Cypress はその全部又は一部を問わず一切の責任を負わず、かつ Cypress はそれら一切から本書により免除される。Cypress は Cypress 製品の本来目的外使用から生じ又は本目的外使用に関連するあらゆる請求、費用、損害及びその他の責任 (人身傷害又は死亡に基づく請求を含む) から免責補償される。

Cypress, Cypress のロゴ, Spansion, Spansion のロゴ及びこれらの組み合わせ, WICED, PSoC, CapSense, EZ-USB, F-RAM, 及び Traveo は、米国及びその他の国における Cypress の商標又は登録商標である。Cypress のより完全な商標のリストは、[cypress.com](http://cypress.com) を参照すること。その他の名称及びブランドは、それぞれの権利者の財産として権利主張がなされている可能性がある。