
セクション 3. メモリ構成

ハイライト

本セクションには下記の主要項目を記載しています。

3.1	はじめに.....	3-2
3.2	制御レジスタ	3-2
3.3	PIC32MX のメモリ配置	3-13
3.4	PIC32MX のアドレスマップ	3-15
3.5	バスマトリクス	3-29
3.6	I/O ピンの制御	3-33
3.7	省電力モードとデバッグモードでの動作	3-33
3.8	サンプルコード	3-34
3.9	設計のヒント	3-35
3.10	関連アプリケーション ノート	3-36
3.11	改訂履歴	3-37

Note: ファミリ リファレンス マニュアルの本セクションは、デバイス データシートの補足を目的としています。本セクションの内容は PIC32MX ファミリの一部のデバイスには対応していません。

本書の内容がお客様のご使用になるデバイスに対応しているかどうかは、最新デバイス データシート内の「メモリ」の冒頭に記載している注意書きでご確認ください。

デバイス データシートとファミリ リファレンス マニュアルの各セクションは、マイクロチップ社のウェブサイト (<http://www.microchip.com>) からダウンロードできます。

3.1 はじめに

PIC32MX マイクロコントローラは、連続した 4 GB の仮想メモリアドレス空間を提供します。プログラムメモリ、データメモリ、SFR、コンフィグレーション レジスタを含む全てのメモリ領域は、このアドレス空間内の一意のアドレスに配置されます。オプション設定により、プログラムメモリとデータメモリをユーザメモリとカーネルメモリに分割できます。さらに、PIC32MX にデータメモリからコードを実行させる事もできます。

PIC32MX メモリ構成の主な特長は以下の通りです。

- 32 ビットのネイティブデータ幅
- ユーザモード アドレス空間とカーネルモード アドレス空間を別々に構成
- 柔軟に分割可能なプログラム フラッシュメモリ
- データ空間とプログラム空間に柔軟に分割可能なデータ RAM
- 保護コード用のブート フラッシュメモリ
- コードの暴走を阻止する信頼性の高いバス例外処理
- FMT (Fixed Mapping Translation) ユニットによる単純なメモリ割り当て
- キャッシュありとキャッシュなしのアドレス領域

3.2 制御レジスタ

以下では、ユーザモードとカーネルモードで、RAM とフラッシュメモリをデータ用とコード用に分割するために使う特殊機能レジスタ (SFR) の設定について説明します。

下記は利用可能な SFR の一覧です。

- BMXCON: コンフィグレーション レジスタ
- BMXxxxBA: メモリ分割ベースアドレス レジスタ
- BMXDRMSZ: データ RAM サイズ レジスタ
- BMXPFMSZ: プログラム フラッシュサイズ レジスタ
- BMXBOOTSZ: ブート フラッシュサイズ レジスタ

3.2.1 BMXCON レジスタ

このレジスタは DMA アクセスのキャッシュ、バスエラー例外、データ RAM 待機ステート、調停モードを有効または無効にします。

3.2.2 BMXxxxBA レジスタ

これらのレジスタは、RAM 内のカーネルおよびユーザモード データ空間とユーザモード プログラム空間の相対ベースアドレスを指定します。

3.2.3 BMXDRMSZ レジスタ

この読み出し専用レジスタは、データ RAM の容量 (byte) を定義します。

3.2.4 BMXPFMSZ レジスタ

この読み出し専用レジスタは、プログラム フラッシュメモリの容量 (byte) を定義します。

3.2.5 BMXBOOTSZ レジスタ

この読み出し専用レジスタは、ブート プログラム フラッシュメモリの容量 (byte) を定義します。表 3-1 に、メモリ構成に関連する全てのレジスタの概要を示します。この表の後に、各レジスタの詳細な説明を記載しています。

表 3-1: メモリ構成 SFR の概要

アドレス オフセット	レジスタ名	ビット レンジ	ビット 31/23/15/7	ビット 30/22/14/6	ビット 29/21/13/5	ビット 28/20/12/4	ビット 27/19/11/3	ビット 26/18/10/2	ビット 25/17/9/1	ビット 24/16/8/0	
0x2000	BMXCON ^(1,2,3)	31:24	—	—	—	—	—	BMXCHEDMA	—	—	
		23:16	—	—	—	—	BMXERRIXI	BMXERRICD	BMXERRDMA	BMXERRDS	BMXERRIS
		15:8	—	—	—	—	—	—	—	—	—
		7:0	—	BMXWSDRM	—	—	—	—	BMXARB<2:0>		
0x2010	BMXDKPBA ^(1,2,3)	31:24	—	—	—	—	—	—	—	—	
		23:16	—	—	—	—	—	—	—	—	
		15:8	BMXDKPBA<15:8>								
		7:0	BMXDKPBA<7:0>								
0x2020	BMXDUDBA ^(1,2,3)	31:24	—	—	—	—	—	—	—	—	
		23:16	—	—	—	—	—	—	—	—	
		15:8	BMXDUDBA<15:8>								
		7:0	BMXDUDBA<7:0>								
0x2030	BMXDUPBA ^(1,2,3)	31:24	—	—	—	—	—	—	—	—	
		23:16	—	—	—	—	—	—	—	—	
		15:8	BMXDUPBA<15:8>								
		7:0	BMXDUPBA<7:0>								
0x2040	BMXDRMSZ	31:24	BMXDRMSZ<31:24>								
		23:16	BMXDRMSZ<23:16>								
		15:8	BMXDRMSZ<15:8>								
		7:0	BMXDRMSZ<7:0>								
0x2050	BMXPUPBA ^(1,2,3)	31:24	—	—	—	—	—	—	—	—	
		23:16	—	—	—	—	BMXPUPBA<19:16>				
		15:8	BMXPUPBA<15:8>								
		7:0	BMXPUPBA<7:0>								
0x2060	BMXPFMSZ	31:24	BMXPFMSZ<31:24>								
		23:16	BMXPFMSZ<23:16>								
		15:8	BMXPFMSZ<15:8>								
		7:0	BMXPFMSZ<7:0>								
0x2070	BMXBOOTSZ	31:24	BMXBOOTSZ<31:24>								
		23:16	BMXBOOTSZ<23:16>								
		15:8	BMXBOOTSZ<15:8>								
		7:0	BMXBOOTSZ<7:0>								

凡例:

- = 未実装、「0」として読み出し、アドレス オフセット値は 16 進表記です。
- Note** 1: このレジスタに対応するクリアレジスタのアドレスは 0x4 バイト オフセットしています。クリアレジスタは、対応するレジスタの名前の後に「CLR」を追加した名前を持ちます (例: BMXCONCLR)。クリアレジスタの任意のビットに「1」を書き込むと、対応するレジスタの対応するビットがクリアされます。クリアレジスタからの読み出しは無視されます。
- 2: このレジスタに対応するセットレジスタのアドレスは 0x8 バイト オフセットしています。セットレジスタは、対応するレジスタの名前の後に「SET」を追加した名前を持ちます (例: BMXCONSET)。セットレジスタの任意のビットに「1」を書き込むと、対応するレジスタの対応するビットがセットされます。セットレジスタからの読み出しは無視されます。
- 3: このレジスタに対応する反転レジスタのアドレスは 0xC バイト オフセットしています。反転レジスタは、対応するレジスタの名前の後に「INV」を追加した名前を持ちます (例: BMXCONINV)。反転レジスタの任意のビットに「1」を書き込むと、対応するレジスタの対応するビットが反転します。反転レジスタからの読み出しは無視されます。

PIC32MX ファミリ リファレンス マニュアル

レジスタ 3-1: **BMXCON: バスマトリクス コンフィグレーション レジスタ** (1,2,3)

r-x	r-x	r-x	r-x	r-x	R/W-0	r-x	r-x
—	—	—	—	—	BMXCHEDMA	—	—
bit 31					bit 24		

r-x	r-x	r-x	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	BMXERRIXI	BMXERRICD	BMXERRDMA	BMXERRDS	BMXERRIS
bit 23					bit 16		

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 15					bit 8		

r-x	R/W-1	r-x	r-x	r-x	R/W-0	R/W-0	R/W-1
—	BMXWSDRM	—	—	—	BMXARB<2:0>		—
bit 7					bit 0		

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット P = 書き込み可能ビット r = 予約済みビット
 U = 未実装ビット n = POR 時のビット値 (「0」、「1」または x = ビットは未知)

- bit 31-27 **予約済み:** 「0」を書き込み; 読み出しは無視
- bit 26 **BMXCHEDMA:** DMA アクセス向け BMX PFM キャッシュ適用性ビット
 - 1 = DMA アクセスに対するプログラム フラッシュメモリ (データ) のキャッシュを有効にする (データのキャッシュを有効にするには、デバイスにキャッシュ モジュールが必要です)
 - 0 = DMA アクセスに対するプログラム フラッシュメモリ (データ) のキャッシュを無効にする (ヒット時はキャッシュから読み出しますが、ミスヒット時はキャッシュを更新しません)
- bit 25-21 **予約済み:** 「0」を書き込み; 読み出しは無視
- bit 20 **BMXERRIXI:** IXI バスエラー イネーブルビット
 - 1 = IXI 共有バスに起因する未割り当てアドレスアクセスに対するバスエラー例外を有効にする
 - 0 = IXI 共有バスに起因する未割り当てアドレスアクセスに対するバスエラー例外を無効にする
- bit 19 **BMXERRICD:** ICD デバッグユニット バスエラー イネーブルビット
 - 1 = ICD に起因する未割り当てアドレスアクセスに対するバスエラー例外を有効にする
 - 0 = ICD に起因する未割り当てアドレスアクセスに対するバスエラー例外を無効にする
- bit 18 **BMXERRDMA:** DMA バスエラービット
 - 1 = DMA に起因する未割り当てアドレスアクセスに対するバスエラー例外を有効にする
 - 0 = DMA に起因する未割り当てアドレスアクセスに対するバスエラー例外を無効にする
- bit 17 **BMXERRDS:** CPU データアクセス バスエラービット (デバッグモードでは無効)
 - 1 = CPU データアクセスに起因する未割り当てアドレスアクセスに対するバスエラー例外を有効にする
 - 0 = CPU データアクセスに起因する未割り当てアドレスアクセスに対するバスエラー例外を無効にする
- bit 16 **BMXERRIS:** CPU 命令アクセス バスエラービット (デバッグモードでは無効)
 - 1 = CPU 命令アクセスに起因する未割り当てアドレスアクセスに対するバスエラー例外を有効にする
 - 0 = CPU 命令アクセスに起因する未割り当てアドレスアクセスに対するバスエラー例外を無効にする

- Note 1:** このレジスタに対応するクリアレジスタ (BMXCONCLR) のアドレスは 0x4 バイト オフセットしています。クリアレジスタの任意のビットに「1」を書き込むと、対応するレジスタの対応するビットがクリアされます。クリアレジスタからの読み出しは無視されます。
- 2:** このレジスタに対応するセットレジスタ (BMXCONSET) のアドレスは 0x8 バイト オフセットしています。セットレジスタの任意のビットに「1」を書き込むと、対応するレジスタの対応するビットがセットされます。セットレジスタからの読み出しは無視されます。
- 3:** このレジスタに対応する反転レジスタ (BMXCONINV) のアドレスは 0xC バイト オフセットしています。反転レジスタの任意のビットに「1」を書き込むと、対応するレジスタの対応するビットが反転します。反転レジスタからの読み出しは無視されます。

レジスタ 3-1: BMXCON: バスマトリクス コンフィグレーション レジスタ^(1,2,3) (続き)

bit 15-7	予約済み: 「0」を書き込み; 読み出しは無視
bit 6	BMXWSDRM: CPU からデータ RAM への命令またはデータアクセス待機ステートビット 1 = CPU からのデータ RAM アクセスに、アドレス セットアップ用の 1 待機ステートを挿入する 0 = CPU からのデータ RAM アクセスに、アドレス セットアップ用の待機ステートを挿入しない
bit 5-3	予約済み: 「0」を書き込み; 読み出しは無視
bit 2-0	BMXARB<2:0>: バスマトリクス調停モードビット 111...011 = 予約済み (これらのモードを使った場合の挙動は不確定です) 010 = 調停モード 2 001 = 調停モード 1 (既定値) 000 = 調停モード 0

- Note 1:** このレジスタに対応するクリアレジスタ (BMXCONCLR) のアドレスは 0x4 バイト オフセットしていません。クリアレジスタの任意のビットに「1」を書き込むと、対応するレジスタの対応するビットがクリアされます。クリアレジスタからの読み出しは無視されます。
- 2:** このレジスタに対応するセットレジスタ (BMXCONSET) のアドレスは 0x8 バイト オフセットしていません。セットレジスタの任意のビットに「1」を書き込むと、対応するレジスタの対応するビットがセットされます。セットレジスタからの読み出しは無視されます。
- 3:** このレジスタに対応する反転レジスタ (BMXCONINV) のアドレスは 0xC バイト オフセットしていません。反転レジスタの任意のビットに「1」を書き込むと、対応するレジスタの対応するビットが反転します。反転レジスタからの読み出しは無視されます。

PIC32MX ファミリ リファレンス マニュアル

レジスタ 3-2: BMXDKPBA: データ RAM カーネル プログラム ベースアドレス レジスタ (1,2,3,4,5)

r-X	r-X	r-X	r-X	r-X	r-X	r-X	r-X
—	—	—	—	—	—	—	—
bit 31						bit 24	

r-X	r-X	r-X	r-X	r-X	r-X	r-X	r-X
—	—	—	—	—	—	—	—
bit 23						bit 16	

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-0	R-0
BMXDKPBA<15:8>							
bit 15						bit 8	

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
BMXDKPBA<7:0>							
bit 7						bit 0	

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット P = 書き込み可能ビット r = 予約済みビット
 U = 未実装ビット n = POR 時のビット値 (「0」、「1」または x = ビットは未知)

bit 31-16 **予約済み:** 「0」を書き込み; 読み出しは無視

bit 15-11 **BMXDKPBA<15:11>:** DRM カーネル プログラム ベースアドレス ビット
 ゼロ以外の値を設定する事により、RAM 内のカーネル プログラム空間の相対ベースアドレスを選択します。

bit 10-0 **BMXDKPBA<10:0>:** 読み出し専用ビット
 値は常に「0」です (2 KB インクリメントに固定)。

- Note 1:** このレジスタに対応するクリアレジスタ (BMXDKPBACLR) のアドレスは 0x4 バイト オフセットしています。クリアレジスタの任意のビットに「1」を書き込むと、対応するレジスタの対応するビットがクリアされます。クリアレジスタからの読み出しは無視されます。
- 2:** このレジスタに対応するセットレジスタ (BMXDKPBASET) のアドレスは 0x8 バイト オフセットしています。セットレジスタの任意のビットに「1」を書き込むと、対応するレジスタの対応するビットがセットされます。セットレジスタからの読み出しは無視されます。
- 3:** このレジスタに対応する反転レジスタ (BMXDKPBAINV) のアドレスは 0xC バイト オフセットしています。反転レジスタの任意のビットに「1」を書き込むと、対応するレジスタの対応するビットが反転します。反転レジスタからの読み出しは無視されます。
- 4:** このレジスタの値はリセット時にゼロに設定され、RAM の全領域がカーネルモード データ用に割り当てられます。
- 5:** このレジスタの値は BMXDRMSZ 以下である必要があります。

レジスタ 3-3: BMXDUDBA: データ RAM ユーザデータ ベースアドレス レジスタ (1,2,3,4,5)

r-X	r-X	r-X	r-X	r-X	r-X	r-X	r-X
—	—	—	—	—	—	—	—
bit 31						bit 24	

r-X	r-X	r-X	r-X	r-X	r-X	r-X	r-X
—	—	—	—	—	—	—	—
bit 23						bit 16	

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-0	R-0
BMXDUDBA<15:8>							
bit 15						bit 8	

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
BMXDUDBA<7:0>							
bit 7						bit 0	

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット P = 書き込み可能ビット r = 予約済みビット
 U = 未実装ビット n = POR 時のビット値 (「0」、「1」または x = ビットは未知)

- bit 31-16 **予約済み:** 「0」を書き込み; 読み出しは無視
- bit 15-11 **BMXDUDBA<15:11>:** DRM ユーザデータ ベースアドレス ビット
 ゼロ以外の値を設定する事により、RAM 内のユーザモード データ空間の相対ベースアドレスを選択します。
Note: ゼロ以外の値を設定する場合、BMXDKPBA よりも大きな値を指定する必要があります。
- bit 10-0 **BMXDUDBA<10:0>:** 読み出し専用ビット
 値は常に「0」です (2 KB インクリメントに固定)。

- Note 1:** このレジスタに対応するクリアレジスタ (BMXDUDBACLR) のアドレスは 0x4 バイト オフセットしています。クリアレジスタの任意のビットに「1」を書き込むと、対応するレジスタの対応するビットがクリアされます。クリアレジスタからの読み出しは無視されます。
- 2:** このレジスタに対応するセットレジスタ (BMXDUDBASET) のアドレスは 0x8 バイト オフセットしています。セットレジスタの任意のビットに「1」を書き込むと、対応するレジスタの対応するビットがセットされます。セットレジスタからの読み出しは無視されます。
- 3:** このレジスタに対応する反転レジスタ (BMXDUDBAINV) のアドレスは 0xC バイト オフセットしていません。反転レジスタの任意のビットに「1」を書き込むと、対応するレジスタの対応するビットが反転します。反転レジスタからの読み出しは無視されます。
- 4:** このレジスタの値はリセット時にゼロに設定され、RAM の全体がカーネルモード データ用に割り当てられます。
- 5:** このレジスタの値は、BMXDRMSZ 以下である事が必要です。

PIC32MX ファミリ リファレンス マニュアル

レジスタ 3-4: BMXDUPBA: データ RAM ユーザプログラム ベースアドレス レジスタ (1,2,3,4,5)

r-X	r-X	r-X	r-X	r-X	r-X	r-X	r-X
—	—	—	—	—	—	—	—
bit 31						bit 24	

r-X	r-X	r-X	r-X	r-X	r-X	r-X	r-X
—	—	—	—	—	—	—	—
bit 23						bit 16	

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-0	R-0
BMXDUPBA<15:8>							
bit 15						bit 8	

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
BMXDUPBA<7:0>							
bit 7						bit 0	

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット P = 書き込み可能ビット r = 予約済みビット
 U = 未実装ビット n = POR 時のビット値 (「0」、「1」または x = ビットは未知)

bit 31-16 **予約済み:** 「0」を書き込み; 読み出しは無視

bit 15-11 **BMXDUPBA<15:11>:** DRM ユーザプログラム ベースアドレスビット
 ゼロ以外の値を設定する事により、RAM 内のユーザモード プログラム空間の相対ベースアドレスを選択します。

Note: ゼロ以外の値を設定する場合、BMXDUPBA よりも大きな値を指定する必要があります。

bit 10-0 **BMXDUPBA<10:0>:** 読み出し専用ビット
 値は常に「0」です (2 KB インクリメントに固定)。

- Note 1:** このレジスタに対応するクリアレジスタ (BMXDUPBACLRL) のアドレスは 0x4 バイト オフセットしています。クリアレジスタの任意のビットに「1」を書き込むと、対応するレジスタの対応するビットがクリアされます。クリアレジスタからの読み出しは無視されます。
- 2:** このレジスタに対応するセットレジスタ (BMXDUPBASET) のアドレスは 0x8 バイト オフセットしています。セットレジスタの任意のビットに「1」を書き込むと、対応するレジスタの対応するビットがセットされます。セットレジスタからの読み出しは無視されます。
- 3:** このレジスタに対応する反転レジスタ (BMXDUPBAINV) のアドレスは 0xC バイト オフセットしています。反転レジスタの任意のビットに「1」を書き込むと、対応するレジスタの対応するビットが反転します。反転レジスタからの読み出しは無視されます。
- 4:** このレジスタの値は、リセット時にゼロに設定され、RAM の全体がカーネルモード データ用に割り当てられます。
- 5:** このレジスタの値は、BMXDRMSZ 以下である必要があります。

レジスタ 3-5: BMXDRMSZ: データ RAM サイズレジスタ

R	R	R	R	R	R	R	R
BMXDRMSZ<31:24>							
bit 31				bit 24			

R	R	R	R	R	R	R	R
BMXDRMSZ<23:16>							
bit 23				bit 16			

R	R	R	R	R	R	R	R
BMXDRMSZ<15:8>							
bit 15				bit 8			

R	R	R	R	R	R	R	R
BMXDRMSZ<7:0>							
bit 7				bit 0			

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット P = 書き込み可能ビット r = 予約済みビット
 U = 未実装ビット n = POR 時のビット値 (「0」、「1」または x = ビットは未知)

bit 31-0 **BMXDRMSZ<31:0>**: データ RAM メモリ (DRM) サイズビット

データ RAM のサイズをバイト数で表す固定値です。
 0x00002000 = デバイスは 8 KB の RAM を内蔵している
 0x00004000 = デバイスは 16 KB の RAM を内蔵している
 0x00008000 = デバイスは 32 KB の RAM を内蔵している
 0x00010000 = デバイスは 64 KB の RAM を内蔵している
 0x00020000 = デバイスは 128 KB の RAM を内蔵している

PIC32MX ファミリ リファレンス マニュアル

レジスタ 3-6: BMXPUPBA: プログラム フラッシュ (PFM) ユーザプログラム ベースアドレス レジスタ (1,2,3,4,5)

r-X	r-X	r-X	r-X	r-X	r-X	r-X	r-X
—	—	—	—	—	—	—	—
bit 31						bit 24	

r-X	r-X	r-X	r-X	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	BMXPUPBA<19:16>			
bit 23						bit 16	

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-0	R-0
BMXPUPBA<15:8>							
bit 15						bit 8	

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
BMXPUPBA<7:0>							
bit 7						bit 0	

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット P = 書き込み可能ビット r = 予約済みビット
 U = 未実装ビット n = POR 時のビット値 (「0」、「1」または x = ビットは未知)

- bit 31-20 **予約済み:** 「0」を書き込み; 読み出しは無視
- bit 19-11 **BMXPUPBA<19:11>:** プログラム フラッシュ (PFM) ユーザプログラム ベースアドレスビット
- bit 10-0 **BMXPUPBA<10:0>:** 読み出し専用ビット
 値は常に「0」です (2 KB インクリメントに固定)。

- Note 1:** このレジスタに対応するクリアレジスタ (BMXPUPBACLK) のアドレスは 0x4 バイト オフセットしています。クリアレジスタの任意のビットに「1」を書き込むと、対応するレジスタの対応するビットがクリアされます。クリアレジスタからの読み出しは無視されます。
- 2:** このレジスタに対応するセットレジスタ (BMXPUPPBASET) のアドレスは 0x8 バイト オフセットしています。セットレジスタの任意のビットに「1」を書き込むと、対応するレジスタの対応するビットがセットされます。セットレジスタからの読み出しは無視されます。
- 3:** このレジスタに対応する反転レジスタ (BMXPUPBAINV) のアドレスは 0xC バイト オフセットしています。反転レジスタの任意のビットに「1」を書き込むと、対応するレジスタの対応するビットが反転します。反転レジスタからの読み出しは無視されます。
- 4:** このレジスタの値は、リセット時にゼロに設定され、RAM の全体がカーネルモード プログラム用に割り当てられます。
- 5:** このレジスタの値は、BMXPFMSZ 以下である必要があります。

セクション 3. メモリ構成

レジスタ 3-7: **BMXPFMSZ: プログラム フラッシュ (PFM) サイズレジスタ**

R	R	R	R	R	R	R	R
BMXPFMSZ<31:24>							
bit 31				bit 24			

R	R	R	R	R	R	R	R
BMXPFMSZ<23:16>							
bit 23				bit 16			

R	R	R	R	R	R	R	R
BMXPFMSZ<15:8>							
bit 15				bit 8			

R	R	R	R	R	R	R	R
BMXPFMSZ<7:0>							
bit 7				bit 0			

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット P = 書き込み可能ビット r = 予約済みビット
 U = 未実装ビット n = POR 時のビット値 (「0」、「1」または x = ビットは未知)

bit 31-0 **BMXPFMSZ<31:0>**: プログラム フラッシュメモリ (PFM) サイズビット

PFM のサイズをバイト数で表す固定値です。
 0x00008000 = デバイスは 32 KB のフラッシュを内蔵している
 0x00010000 = デバイスは 64 KB のフラッシュを内蔵している
 0x00020000 = デバイスは 128 KB のフラッシュを内蔵している
 0x00040000 = デバイスは 256 KB のフラッシュを内蔵している
 0x00080000 = デバイスは 512 KB のフラッシュを内蔵している

PIC32MX ファミリ リファレンス マニュアル

レジスタ 3-8: BMXBOOTSZ: ブートフラッシュ (IFM) サイズレジスタ

R	R	R	R	R	R	R	R
BMXBOOTSZ<31:24>							
bit 31							bit 24

R	R	R	R	R	R	R	R
BMXBOOTSZ<23:16>							
bit 23							bit 16

R	R	R	R	R	R	R	R
BMXBOOTSZ<15:8>							
bit 15							bit 8

R	R	R	R	R	R	R	R
BMXBOOTSZ<7:0>							
bit 7							bit 0

凡例:

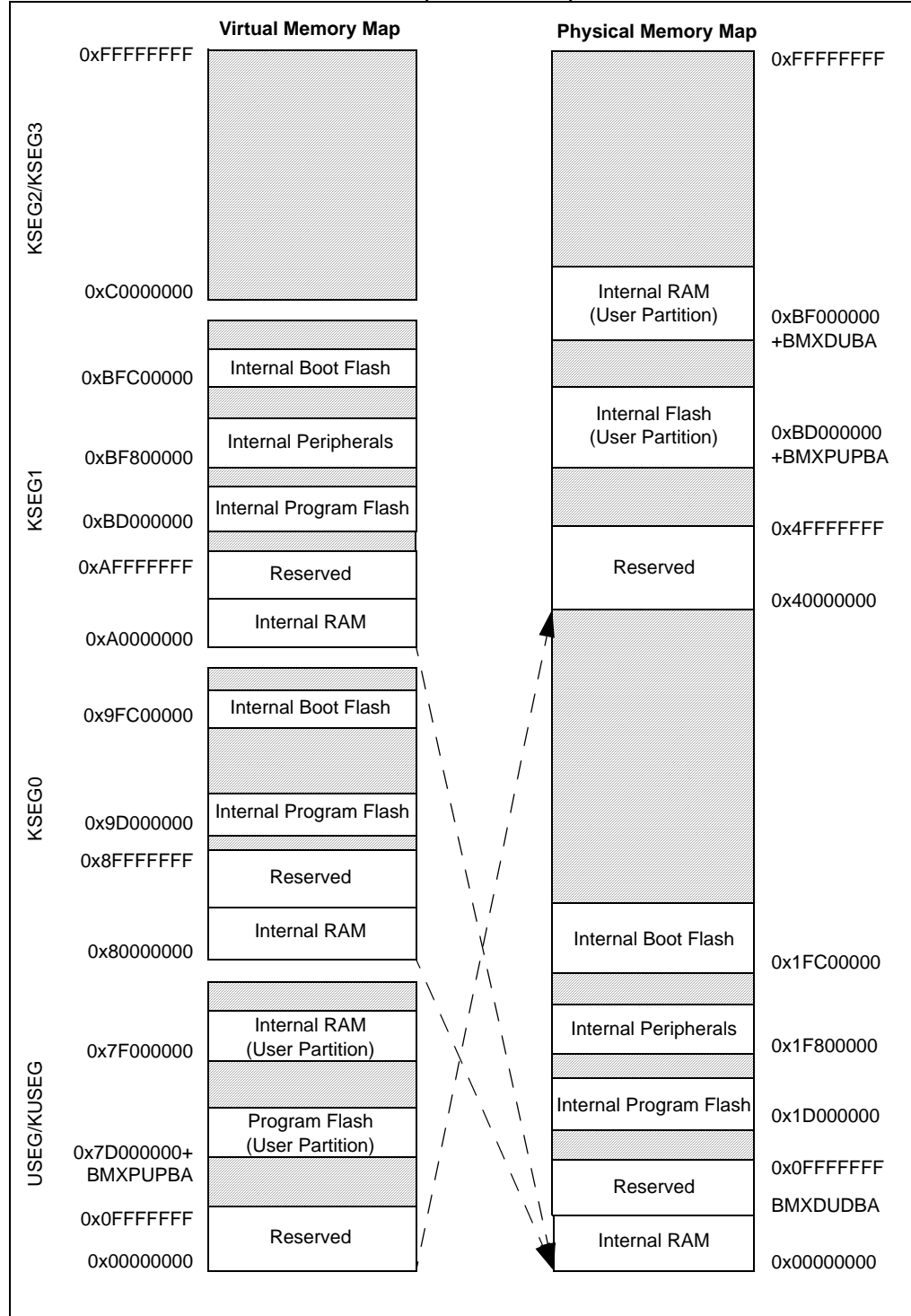
R = 読み出し可能ビット W = 書き込み可能ビット P = 書き込み可能ビット r = 予約済みビット
 U = 未実装ビット n = POR 時のビット値 (「0」、「1」または x = ビットは未知)

bit 31-0 **BMXBOOTSZ<31:0>**: ブートフラッシュメモリ (BFM) サイズビット
 ブート PFM のサイズをバイト数で表す固定値です。
 0x00003000 = デバイスは 12 KB のブートフラッシュを内蔵している

3.3 PIC32MX のメモリ配置

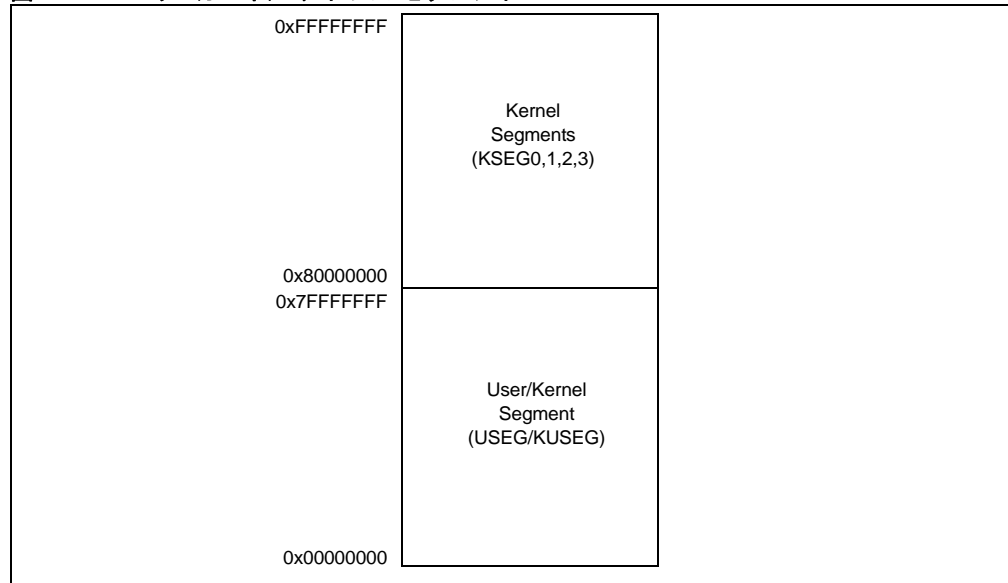
PIC32MX マイクロコントローラは、仮想アドレス空間と物理アドレス空間を実装しています。プログラムメモリ、データメモリ、周辺モジュール等を含む全てのハードウェア リソースは、それぞれ特定の物理アドレスに配置されます。仮想アドレスは、CPU による命令のフェッチと実行にだけ使われます。CPU とは別にメモリにアクセスする周辺モジュール (DMA、フラッシュコントローラ等) は、物理アドレスを使います。

図 3-1: 仮想メモリと物理メモリ間の対応 (固定割り当て)



4 GB の仮想アドレス空間は、2 つの主要領域 (ユーザ空間とカーネル空間) に分割されます。仮想アドレス空間の下位 2 GB はユーザモード セグメント (USEG/KUSEG と呼ぶ) を形成します。ユーザモード アプリケーションは、USEG セグメント内に格納して、このセグメント内で実行する必要があります。USEG セグメントは、全てのカーネルモード アプリケーションからも利用できます。ユーザモードとカーネルモードの両方から利用できるため、このセグメントには「KUSEG」という名前も付けられています。ユーザモードで動作させる場合、フラッシュメモリとデータメモリの一部を USEG/KUSEG セグメントで利用できるようにバスマトリクスを構成する必要があります。詳細は 3.4 「PIC32MX のアドレスマップ」を参照してください。

図 3-2: ユーザ/カーネルアドレスセグメント



仮想アドレス空間の上位 2 GB は、カーネル専用の空間を形成します。カーネル空間は、512 MB の 4 つのセグメント (KSEG0、KSEG1、KSEG2、KSEG3) に等分されます。カーネルモード アプリケーションだけが、カーネル空間メモリにアクセスできます。カーネル空間は、全ての周辺モジュールレジスタを格納します。従って、カーネルモード アプリケーションだけが、周辺モジュールを監視および操作できます。KSEG0 および KSEG1 セグメントだけが、実在するメモリリソースに対応します。セグメント KSEG2 は、MIPS が提供する文書の規定に従って、EJTAG プロブデバッグ向けに利用できます (EJTAG の仕様書参照)。PIC32MX は KSEG0 および KSEG1 セグメントだけを使います。ブートフラッシュメモリ (BFM)、プログラムフラッシュメモリ (PFM)、データ RAM メモリ (DRM)、周辺モジュール SFR には KSEG1 からアクセスできます。

FMT (Fixed Mapping Translation) ユニットの、これらのメモリセグメントを、対応する物理アドレス領域に変換します。図 3-1 は、PIC32MX コアが実装する仮想および物理アドレス空間の固定割り当て方式を示します。デバイスがキャッシュモジュールを実装している場合、仮想メモリセグメントにもキャッシュを適用できます。KSEG0 および USEG/KUSEG KSEG1 メモリセグメントはキャッシュ可能ですが、KSEG1 にはキャッシュを適用できない事に注意してください。

メモリセグメントの割り当ては、CPU エラーレベル (CPU ステータスレジスタの ERL ビットの状態) によって異なります。リセット、ソフトリセット、NMI のいずれかが発生すると、CPU はエラーレベルビットをセットします (ERL = 1)。このモードでは、プロセッサはカーネルモードで動作し、USEG/KUSEG セグメントは未割り当て領域として扱われ、キャッシュは適用されません。この場合のメモリマップは図 3-1 とは異なります。このモードは、TLB 方式の MMU を採用した他の MIPS プロセッサコアとの互換性を維持するために用意されています。C 起動コードで ERL ビットをゼロにクリアする事により、アプリケーションソフトウェアが起動した時に、図 3-1 に示した適正な仮想および物理メモリ間の割り当てが得られます。

セグメント KSEG0 および KSEG1 は、常に物理アドレス 0x0 に変換されます。このような変換により、CPU は 2 つの異なる仮想アドレス (KSEG0 と KSEG1) から同一の物理アドレスにアクセスできます。これにより、アプリケーションは、コードの特定部位の実行にキャッシュを適用するかどうかを選択できます。詳細はセクション 4. 「プリフェッチキャッシュモジュール」 (DS61119) を参照してください。デバイスが内蔵する周辺モジュールは、KSEG1 セグメントからのみアクセス可能です (キャッシュなしアクセス)。

3.4 PIC32MX のアドレスマップ

プログラム フラッシュメモリは、カーネル部とユーザ部に分割できます。カーネル プログラム フラッシュ空間は物理アドレス 0x1D000000 から始まり、ユーザプログラム フラッシュ空間は物理アドレス [0xBD000000 + BMXPUDBA レジスタ値] から始まります。同様に、内部 RAMも、カーネル部とユーザ部に分割できます。カーネルRAM空間は物理アドレス0x00000000 から始まり、ユーザ RAM 空間は物理アドレス [0xBF000000 + BMXDUDBA レジスタ値] から始まります。既定値では、フラッシュメモリと RAM の全領域がカーネルモード アプリケーションだけに割り当てられます。

BMXxxxBA レジスタの設定は、対象とするソフトウェア アプリケーションのメモリモデルに適合していなければならないという事に注意してください。リンクされたコードがレジスタ値に適合しない場合、プログラムは動作せず、起動時にバスエラー例外が発生する可能性があります。

Note: プログラム フラッシュメモリには、そのアドレスマップを介して書き込む事はできません。PFM アドレス領域に書き込むと、バスエラー例外が発生します。

3.4.1 仮想アドレスと物理アドレス間の変換計算

カーネルアドレス (KSEG0 または KSEG1) を物理アドレスに変換する場合、下記のように仮想アドレス値と 0x1FFFFFFF で「ビットワイズ AND」演算を実行します。

- 物理アドレス = 仮想アドレス & 0x1FFFFFFF

物理アドレスから KSEG0 仮想アドレスに変換する場合、下記のように物理アドレス値と 0x80000000 で「ビットワイズ OR」演算を実行します。

- KSEG0 仮想アドレス = 物理アドレス | 0x80000000

物理アドレスから KSEG1 仮想アドレスに変換する場合、下記のように物理アドレス値と 0xA0000000 で「ビットワイズ OR」演算を実行します。

- KSEG1 仮想アドレス = 物理アドレス | 0xA0000000

KSEG0 から KSEG1 仮想アドレスに変換する場合、下記のように KSEG0 仮想アドレス値と 0x20000000 で「ビットワイズ OR」演算を実行します。

- KSEG1 仮想アドレス = KSEG0 仮想アドレス | 0x20000000

PIC32MX ファミリ リファレンス マニュアル

表 3-2: PIC32MX アドレスマップ

メモリアドレス	メモリアドレス	仮想アドレス		物理アドレス		サイズ(バイト)
		開始アドレス	終了アドレス	開始アドレス	終了アドレス	計算
カーネルアドレス空間	ブートフラッシュ	0xBFC00000	0xBFC02FFF	0x1FC00000	0x1FC02FFF	12 KB
	周辺モジュール	0xBF800000	0xBF8FFFFFF	0x1F800000	0x1F8FFFFFF	4 KB
	KSEG1 プログラムフラッシュ ^(1,3)	0xBD000000	0xBD000000 + BMXPUPBA - 1	0x1D000000	0x1D000000 + BMXPUPBA - 1	BMXPUPBA
	KSEG1 プログラムRAM ⁽⁶⁾	0xA0000000 + BMXDKPBA	0xA0000000 + BMXDUDBA - 1	0x00000000 + BMXDKPBA	0x00000000 + BMXDUDBA - 1	BMXDUDBA - BMXDKPBA
	KSEG1 データRAM ⁽⁶⁾	0xA0000000	0xA0000000 + BMXDKPBA - 1	0x00000000	0x00000000 + BMXDKPBA - 1	BMXPUPBA
	KSEG0 プログラムフラッシュ ^(2,5)	0x9D000000	0x9D000000 + BMXPUPBA - 1	0x1D000000	0x1D000000 + BMXPUPBA - 1	BMXPUPBA
	KSEG0 プログラムRAM ⁽⁶⁾	0x80000000 + BMXDKPBA	0x80000000 + BMXDUDBA - 1	0x00000000 + BMXDKPBA	0x00000000 + BMXDUDBA - 1	BMXDUDBA - BMXDKPBA
KSEG0 データRAM ⁽⁶⁾	0x80000000	0x80000000 + BMXDKPBA - 1	0x00000000	0x00000000 + BMXDKPBA - 1	BMXDKPBA	
メモリアドレス	メモリアドレス	仮想アドレス		物理アドレス		サイズ(バイト)
		開始アドレス	終了アドレス	開始アドレス	終了アドレス	計算
ユーザアドレス空間	USEG/KSEG プログラムRAM ⁽⁶⁾	0x7F000000 + BMXDUPBA	0x7F000000 + BMXDRMSZ - 1 ⁽³⁾	0xBF000000 + BMXDUPBA	0xBF000000 + BMXDRMSZ ⁽³⁾ - 1	BMXDRMSZ ⁽³⁾ - BMXDUPBA
	USEG/KSEG データRAM ⁽⁶⁾	0x7F000000 + BMXDUDBA	0x7F000000 + BMXDUPBA - 1	0xBF000000 + BMXDUDBA	0xBF000000 + BMXDUPBA - 1	BMXDUPBA - BMXDUDBA
	USEG/KSEG プログラムフラッシュ ⁽⁶⁾	0x7D000000 + BMXPUPBA	0x7D000000 + BMXPFMSZ ⁽⁴⁾ - 1	0xBD000000 + BMXPUPBA	0xBF000000 + BMXPFMSZ ⁽⁴⁾ - 1	BMXPFMSZ ⁽⁴⁾ - BMXPUPBA

- Note**
- 1: キャッシュ非適用セグメント (KSEG1) 内のプログラムフラッシュ仮想アドレス
 - 2: キャッシュおよびプリフェッチ適用セグメント (KSEG0) 内のプログラムフラッシュ仮想アドレス
 - 3: RAM サイズは PIC32MX のデバイスバージョンごとに異なります。
 - 4: フラッシュ サイズは PIC32MX のデバイスバージョンごとに異なります。
 - 5: BMXPUPBA レジスタが「0」である場合、プログラムフラッシュの全領域がカーネルモードプログラム用に割り当てられます。これはリセット時の既定値状態です。
 - 6: BMXDUDBA、BMXDUPBA、BMXDKPBA レジスタのいずれかが「0」である場合、RAM の全領域がカーネルモードデータ用に割り当てられます。これはリセット時の既定値状態です。

3.4.2 プログラム フラッシュメモリの分割

プログラム フラッシュメモリは、ユーザモードとカーネルモードのプログラム空間に分割できます (図 3-1 参照)。

リセット後の状態では、ユーザモードパーティションは存在しません (BMXPUPBA が「0」に初期化されるため)。この場合、プログラム フラッシュメモリの全領域は、仮想アドレス KSEG1:0xBD000000 (または KSEG0:0x9D000000) から始まるカーネルモード プログラム空間に割り当てられます。ユーザモード プログラム用のパーティションを設定するには、BMXPUPBA を下記のように初期化する必要があります。

• $BMXPUPBA = BMXPFMSZ - USER_FLASH_PGM_SZ$

USER_FLASH_PGM_SZ は、ユーザモード プログラムに割り当てるパーティションのサイズです。BMXPFMSZ は、デバイスが実装するプログラム フラッシュメモリの総容量値を保持するバスマトリクス レジスタです。

例:

- PIC32MX デバイスが 512 K バイトのフラッシュメモリを実装している (すなわち BMXPFMSZ の値が 0x00080000 である) 場合を想定します。
- 20 K バイト (0x5000) のユーザ フラッシュ プログラム パーティションを作成したい場合、BMXPUPBA の値は下記のように求められます。
- $BMXPUPBA = 0x80000 - 0x5000 = 0x7B000$

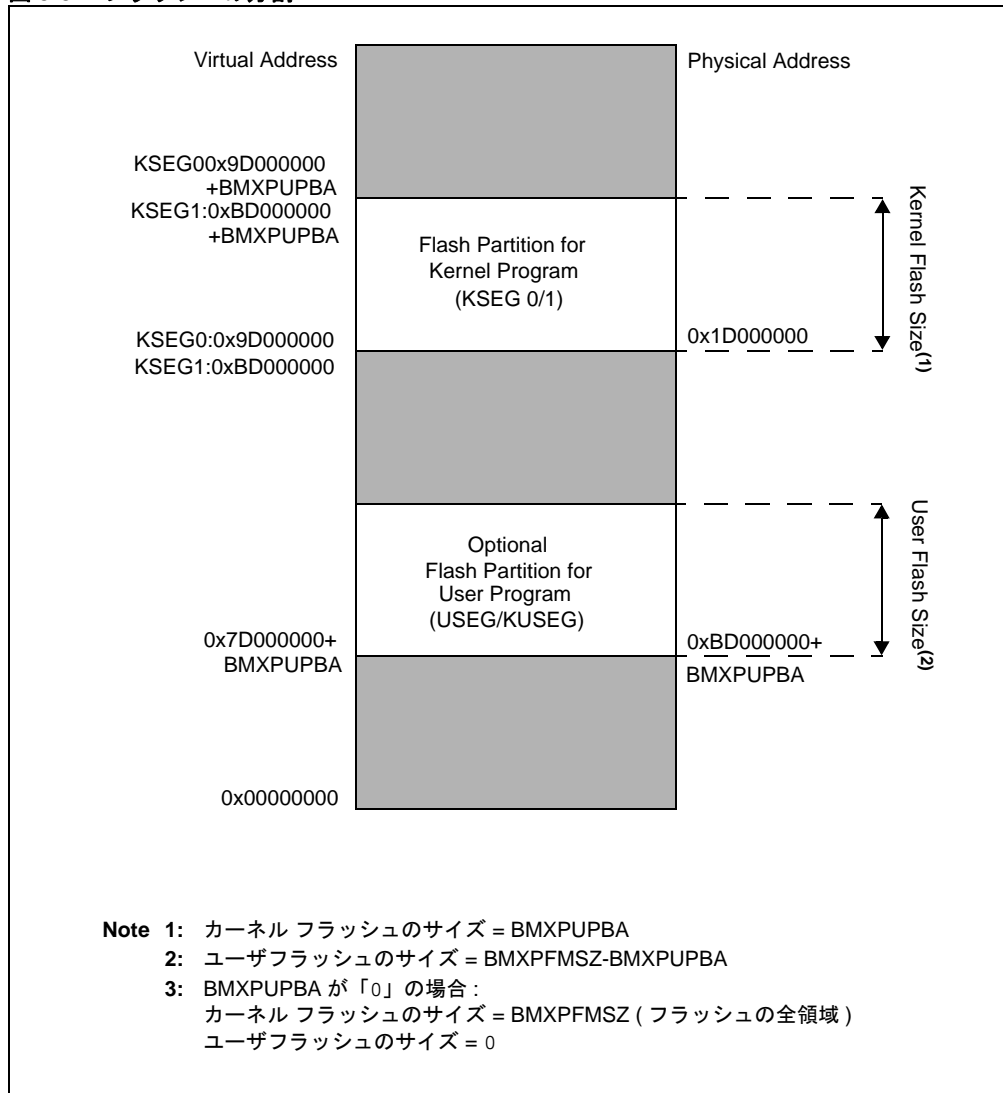
ユーザフラッシュのサイズは 20 K バイトであるため、残りのカーネルフラッシュのサイズは下記のように求められます。

$512 \text{ K バイト} - 20 \text{ K バイト} = 492 \text{ K バイト}$

ユーザフラッシュ パーティションは、仮想アドレス 0x7D07B000 から始まり 0x7D07FFFF で終わります。

カーネルモード パーティションは常に KSEG1:0xBD000000 または KSEG0:0x9D000000 から始まります。上記の例の場合、カーネルパーティションは 0xBD000000 から始まり 0xBD07AFFF で終わります (サイズは 492 K バイト)。

図 3-3: フラッシュの分割



3.4.3 RAM の分割

RAM メモリは下記の 4 つのパーティションに分割できます。

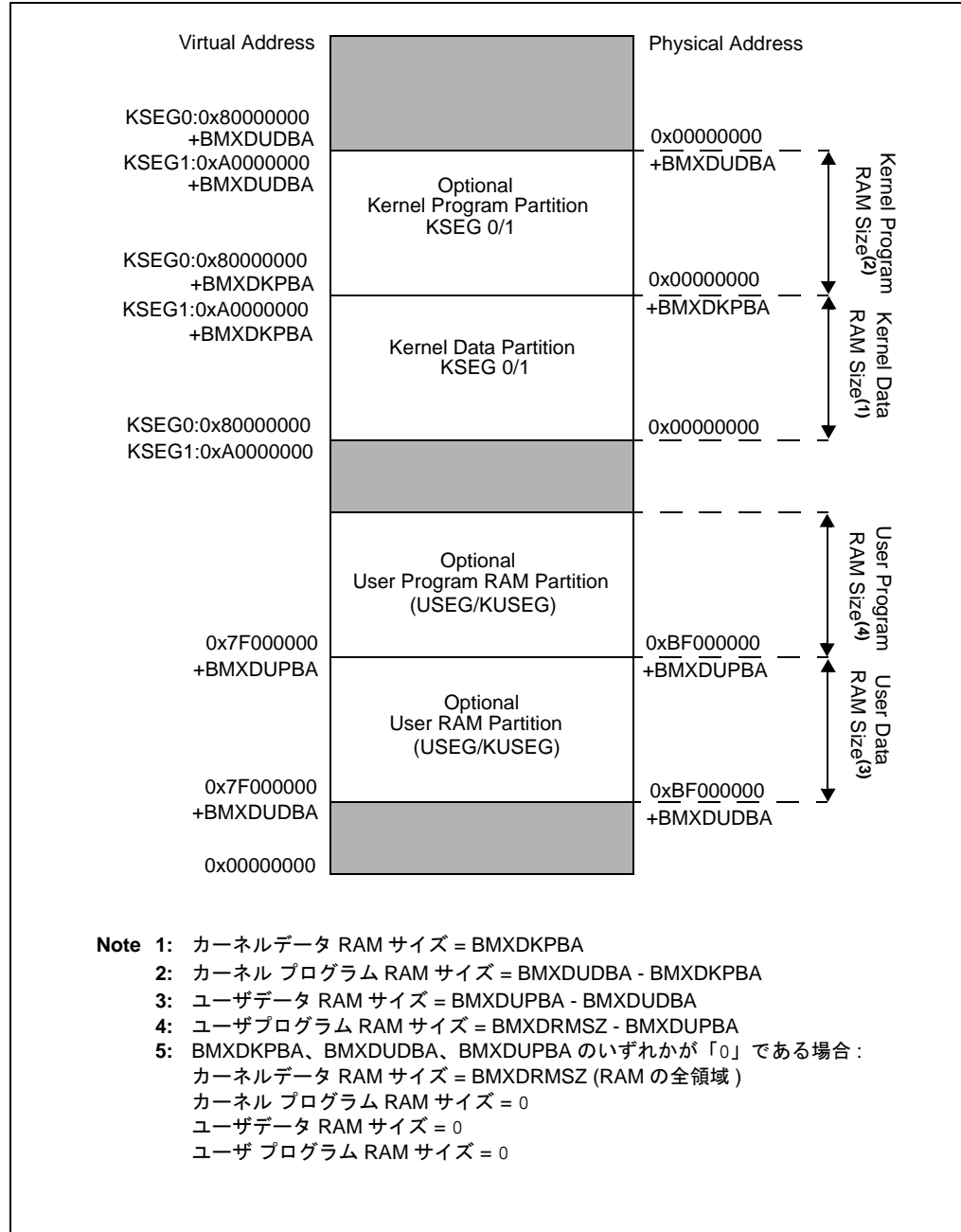
- カーネルデータ
- カーネルプログラム
- ユーザデータ
- ユーザプログラム

データ RAM から実行するには、カーネルプログラムまたはユーザプログラムパーティションを定義する必要があります。パワーオンリセット (POR) が発生すると、データ RAM の全領域はカーネルデータパーティションに割り当てられます。このパーティションは、常にデータ RAM のベースアドレスから始まります。詳細は図 3-4 を参照してください。

Note 1: RAM を適切に分割するには、下記の全てのレジスタを設定する必要があります:
:BMXDKPBA、BMXDUDBA、BMXDUPBA

Note 2: 利用可能な RAM の容量は、BMXDRMSZ レジスタで定義されています。

図 3-4: RAM の分割



3.4.3.1 カーネルデータ RAM パーティション

カーネルデータ RAM パーティションは、仮想アドレス KSEG0:0x80000000、KSEG1:0xA0000000 に配置されます。このパーティションは常に有効であり、無効にはできません。BMXDKPBA、BMXDUDBA、BMXDUPBA レジスタのいずれかが「0」である場合、RAM の全領域がカーネルデータ RAM に割り当てられる事に注意してください。この場合、カーネルデータ RAM パーティションのサイズは BMXDRMSZ レジスタの値によって決まります (図 3-5 参照)。これ以外の場合、カーネルデータ RAM パーティションのサイズは、BMXDKPBA レジスタの値によって決まります (図 3-6 参照)。全てのリセット時に BMXDKPBA、BMXDUDBA、BMXDUPBA レジスタは既定値のゼロに設定されるため、リセットが発生すると、利用可能な RAM の全領域がカーネルデータ RAM パーティションに割り当てられます。

図 3-5: BMXDKPBA、BMXDUDBA、BMXDUPBA のいずれかが「0」の場合の RAM の分割

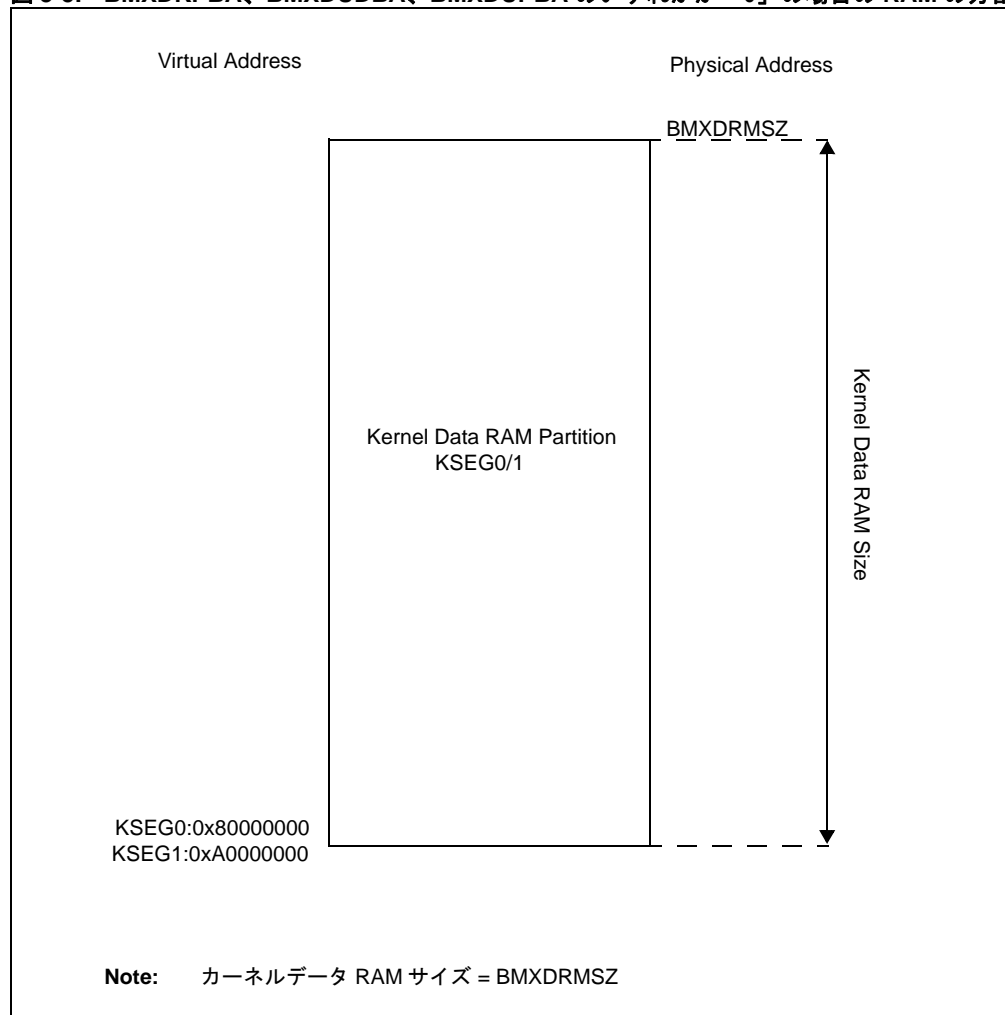
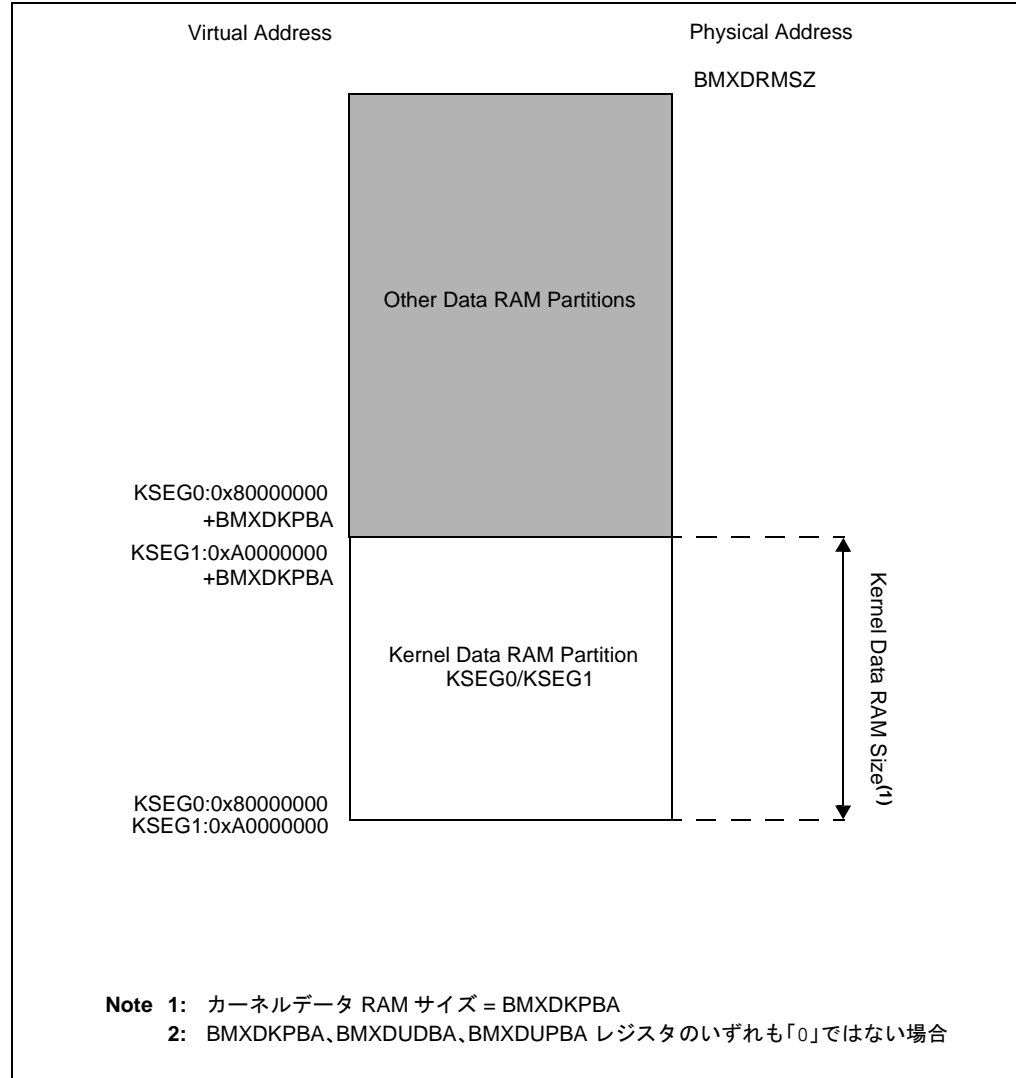


図 3-6: カーネルデータ RAM の分割



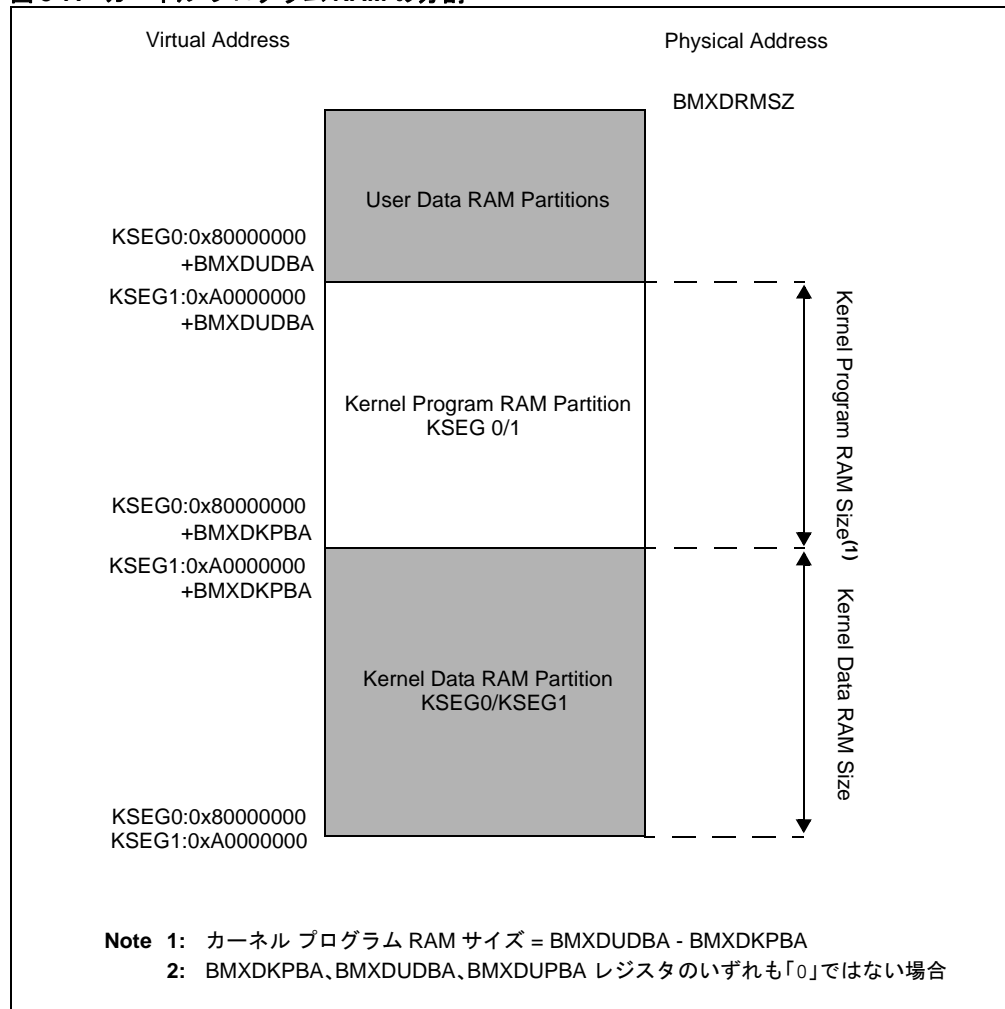
3.4.3.2 カーネル プログラム RAM パーティション

カーネルモードでデータ RAM からコードを実行する必要がある場合、カーネル プログラム RAM パーティションが必要です。

このパーティションは[KSEG0:0x80000000 + BMXDKPBA] (KSEG1:0xA0000000 + BMXDKPBA) から始まり、そのサイズは [BMXDUDBA - BMXDKPBA] です。図 3-7 を参照してください。

リセット時に BMXDKPBA および BMXDUDBA レジスタは既定値のゼロに設定されるため、リセット後の状態ではカーネル プログラム RAM パーティションは存在しません。

図 3-7: カーネル プログラム RAM の分割

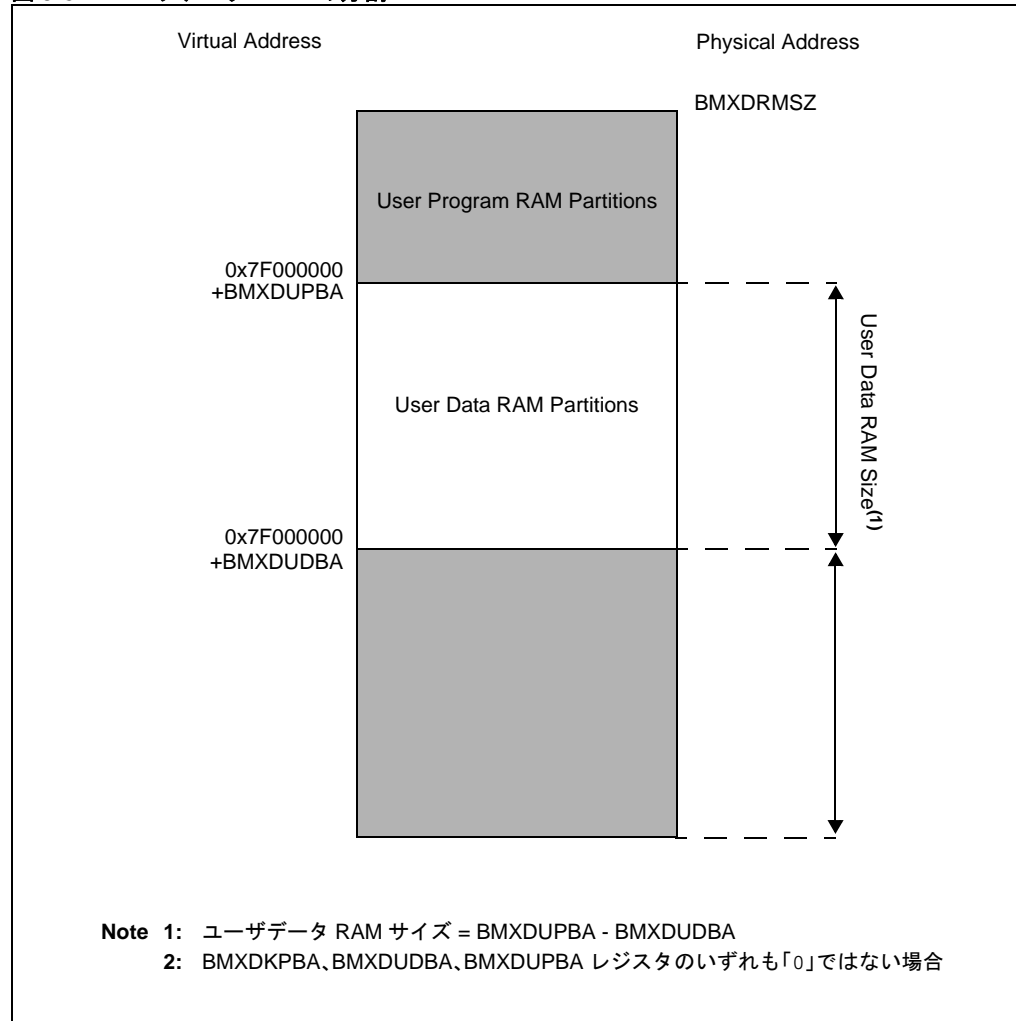


3.4.3.3 ユーザデータ RAM パーティション

ユーザモードアプリケーションは、RAM 内にユーザモード データ パーティションを必要とします。このパーティションはアドレス $[0x7F000000 + BMXDUDBA]$ から始まり、サイズは $[BMXDUPBA - BMXDUDBA]$ です (図 3-8 参照)。

リセット時に BMXDUDBA および BMXDUPBA レジスタは既定値のゼロに設定されるため、リセット後の状態ではユーザデータ RAM パーティションは存在しません。

図 3-8: ユーザデータ RAM の分割

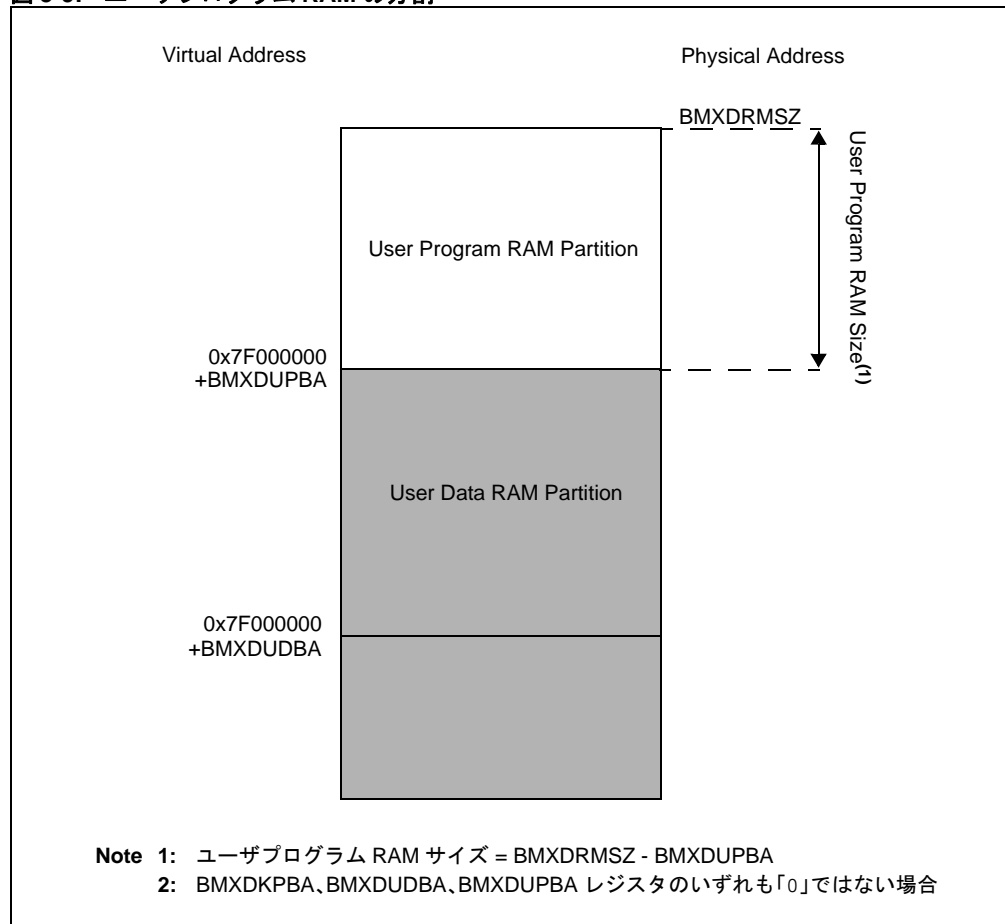


3.4.3.4 ユーザプログラム RAM パーティション

ユーザモードでデータ RAM からコードを実行する必要がある場合、データ RAM 内にユーザプログラム パーティションが必要です。このパーティションはアドレス $[0x7F000000 + BMXDUPBA]$ から始まり、サイズは $[BMXDRMSZ - BMXDUPBA]$ です。図 3-9 を参照してください。

リセット時に $BMXDUPBA$ レジスタは既定値のゼロに設定されるため、リセット後の状態ではユーザプログラム RAM パーティションは存在しません。

図 3-9: ユーザプログラム RAM の分割



3.4.3.5 RAM 分割の例

以下では、RAM の分割方法に関して、下記の実例を挙げて説明します。

- RAM 全体をカーネルデータとして使う
- RAM をカーネルデータとカーネル プログラムに分割する
- RAM をカーネルデータとユーザデータに分割する
- RAM をカーネルデータ、カーネル プログラム、ユーザデータに分割する
- RAM をカーネルデータ、カーネル プログラム、ユーザデータ、ユーザ プログラムに分割する

例 1. RAM 全体をカーネルデータとして使う

リセット後は、RAM の全領域がカーネルデータ RAM に割り当てられます。この場合、一切のプログラミングは不要です。BMXDKPBA、BMXDUDBA、BMXDUPBA レジスタのいずれかを「0」に設定すると、RAM 空間の全体がカーネルデータパーティションに割り当てられます (図 3-5 参照)。

例 2. RAM をカーネルデータとカーネル プログラムに分割する

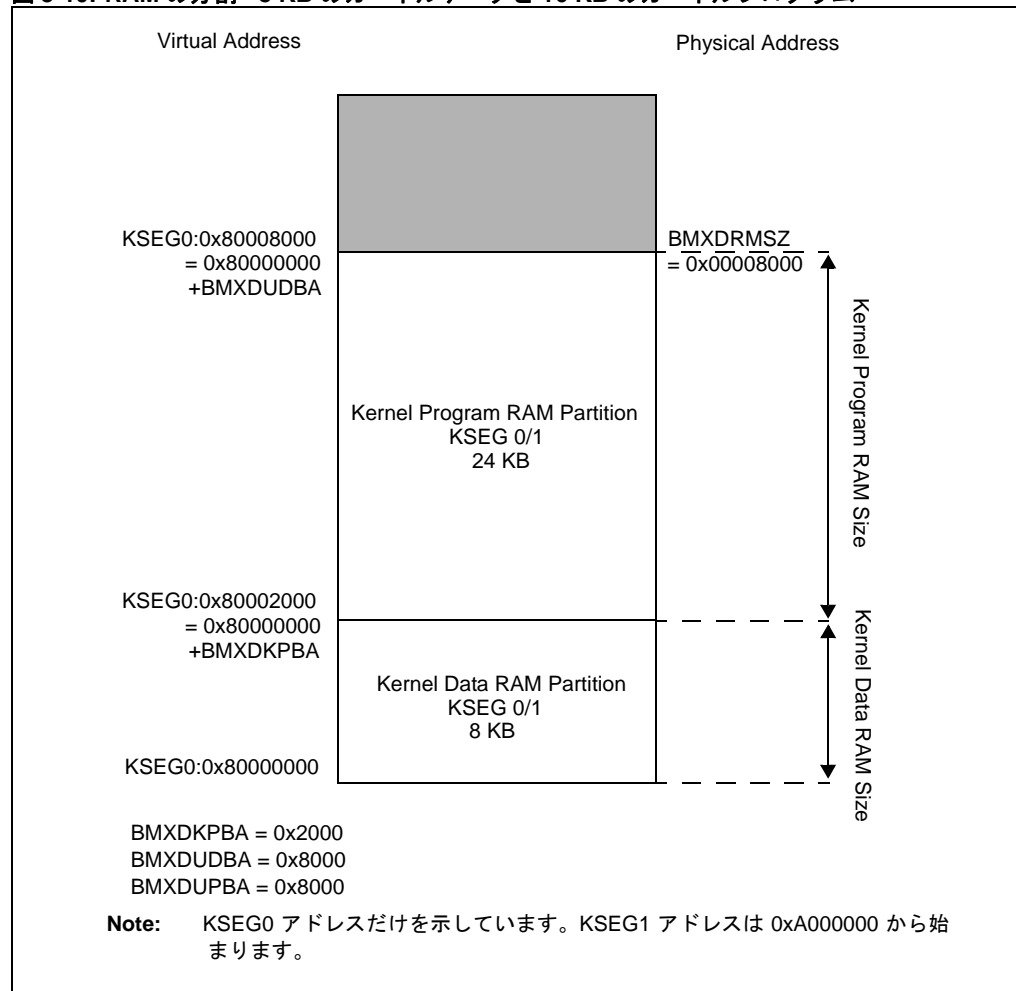
この例では、PIC32MX が 32 KB の RAM を実装していると仮定し、これを 8 KB のカーネルデータ RAM と 24 KB のカーネル プログラム RAM に分割します。この例の場合、ユーザデータ RAM とユーザプログラム RAM のサイズは「0」となります。

カーネルデータ RAM パーティションは常に必要である事に注意してください。詳細は図 3-10 を参照してください。

各レジスタの値は下記の通りです。

- BMXDRMSZ = 0x00008000 (読み出し専用値)
- BMXDKPBA = 0x00002000 (8 KB のカーネルデータ)
- BMXDUDBA = 0x00008000 (24 KB のカーネル プログラム)
- BMXDUPBA = 0x00008000 (ユーザデータのサイズ = 0、ユーザプログラムのサイズ = 0)

図 3-10: RAM の分割 - 8 KB のカーネルデータと 16 KB のカーネルプログラム

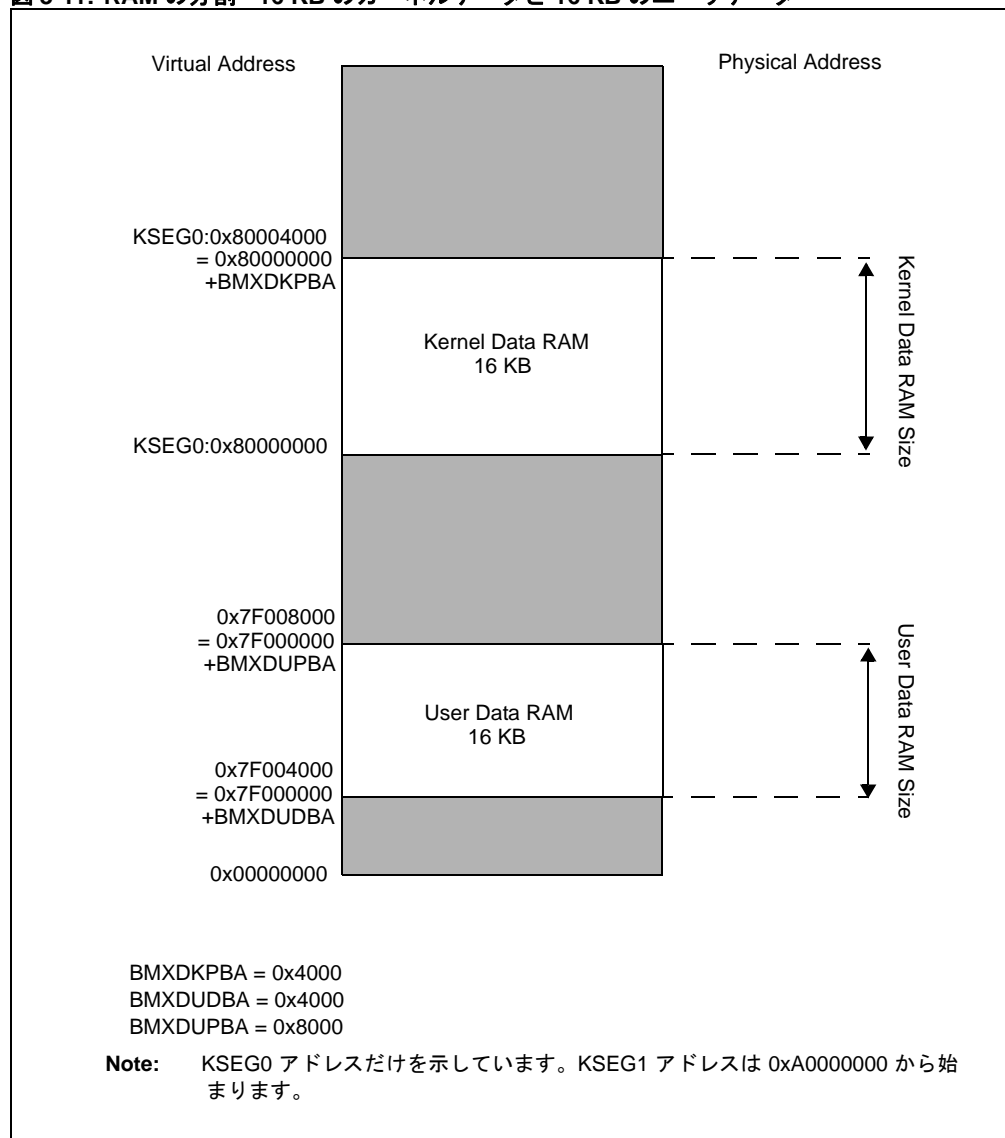


例 3. RAM をカーネルデータとユーザデータに分割する

この例では、PIC32MX が 32 KB の RAM を実装していると仮定し、これを 16 KB のカーネルデータ RAM と 16 KB のユーザデータ RAM に分割します。この例の場合、カーネル プログラム RAM とユーザプログラム RAM のサイズは「0」となります。詳細は図 3-11 を参照してください。各レジスタの値は下記の通りです。

- BMXDRMSZ = 0x00008000 (読み出し専用値)
- BMXDKPBA = 0x00004000 (16 KB のカーネルデータ)
- BMXDUDBA = 0x00004000 (カーネル プログラムのサイズは 0)
- BMXDUPBA = 0x00008000 (ユーザデータのサイズ = 16 KB、ユーザプログラムのサイズ = 0)

図 3-11: RAM の分割 - 16 KB のカーネルデータと 16 KB のユーザデータ



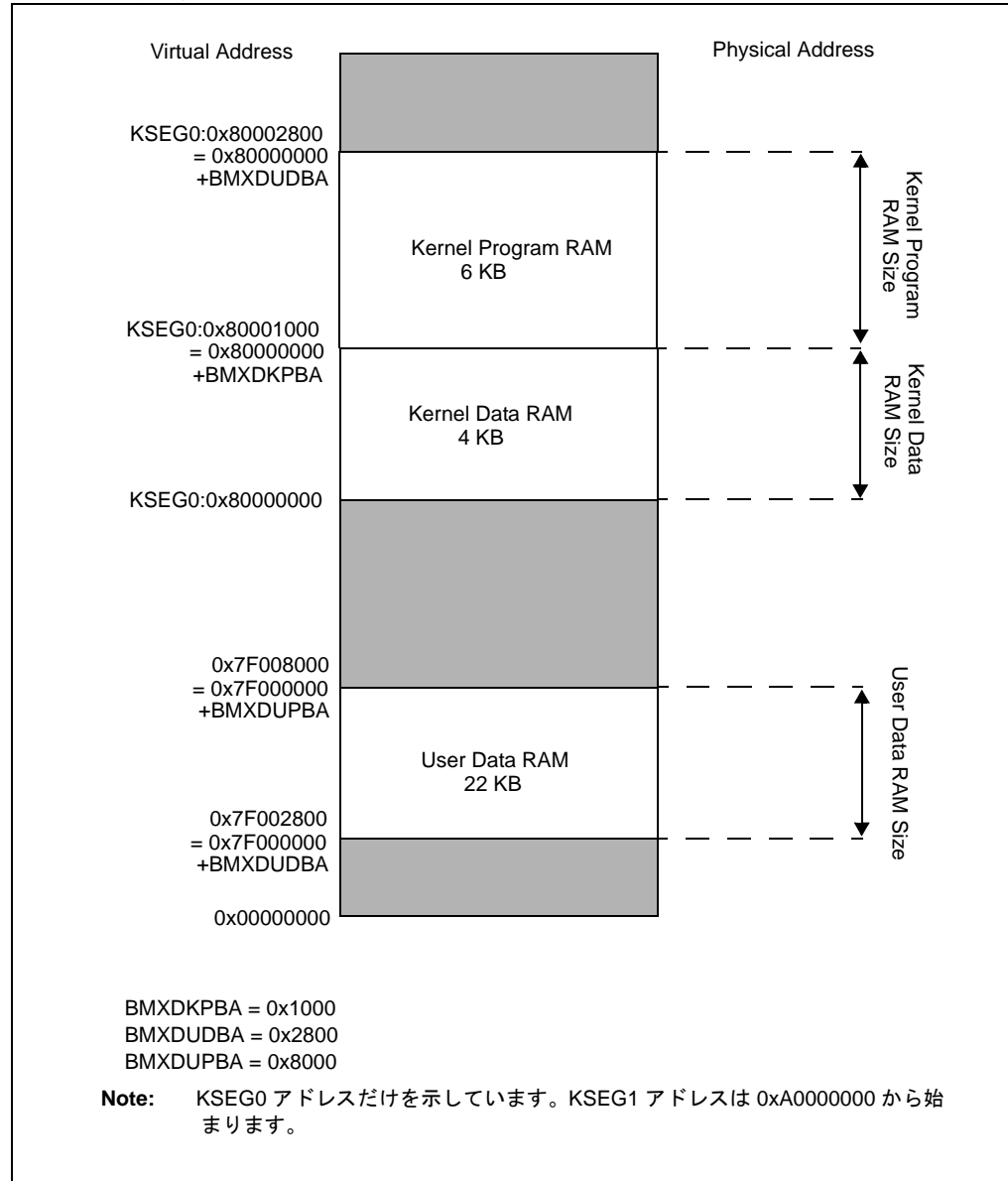
例 4. RAM をカーネルデータ、カーネル プログラム、ユーザデータに分割する

この例では、PIC32MX が 32 KB の RAM を実装していると仮定し、これを 4 KB のカーネルデータ RAM、6 KB のカーネル プログラム RAM、22 KB のユーザデータ RAM に分割します。この例の場合、ユーザプログラム RAM のサイズは「0」となります。詳細は図 3-12 を参照してください。

各レジスタの値は下記の通りです。

- BMXDRMSZ = 0x00008000 (読み出し専用値)
- BMXDKPBA = 0x00001000 (4 KB のカーネルデータ)
- BMXDUDBA = 0x00002800 (6 KB のカーネルプログラム)
- BMXDUPBA = 0x00008000 (ユーザデータのサイズ = 22 KB、ユーザプログラムのサイズ = 0)

図 3-12: RAM の分割 - 4 KB のカーネルデータ、6 KB のカーネルプログラム、22 KB のユーザデータ

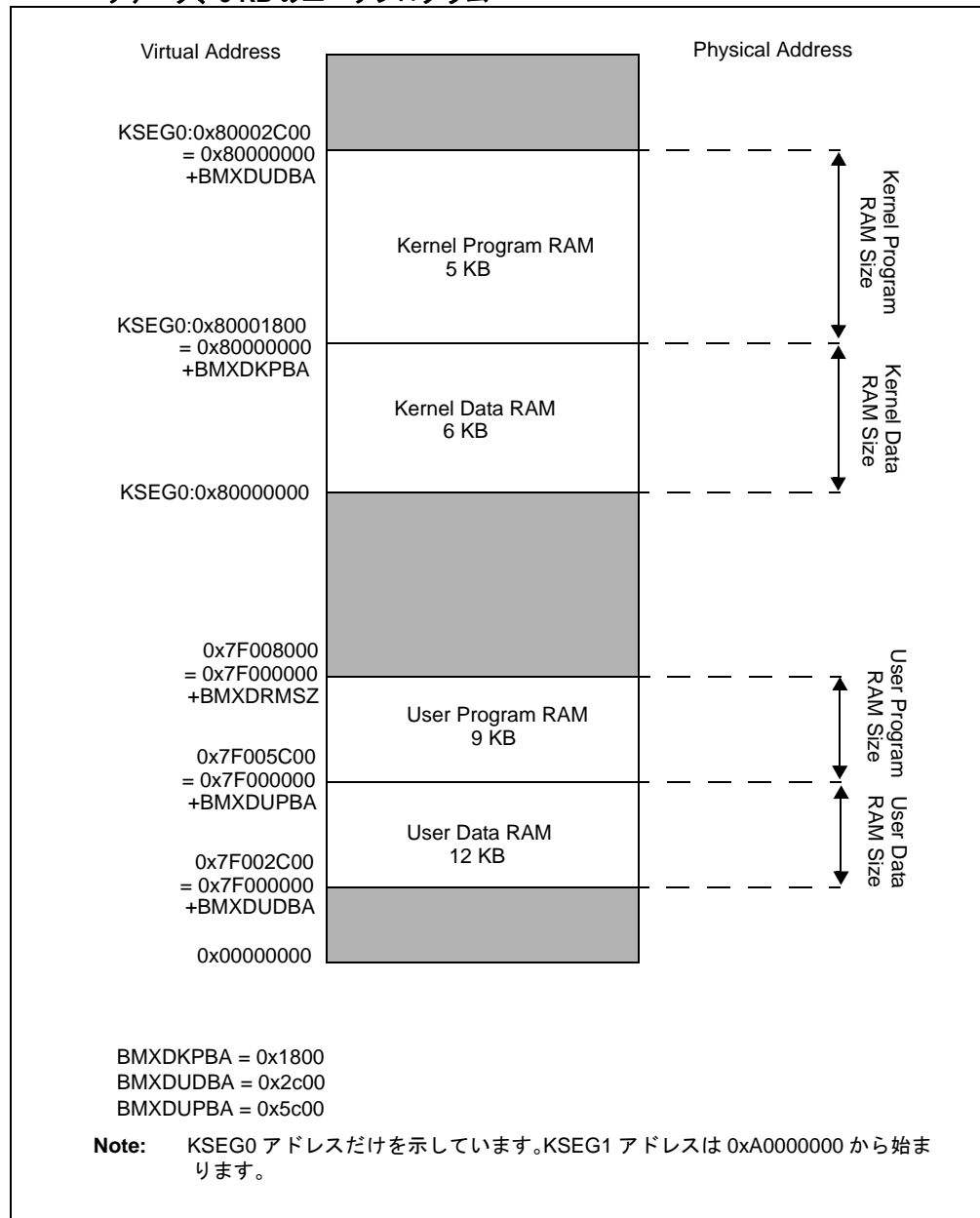


例5. RAMをカーネルデータ、カーネル プログラム、ユーザデータ、ユーザ プログラムに分割する
 この例では、PIC32MX が 32 KB の RAM を実装していると仮定し、これを 6 KB のカーネルデータ RAM、5 KB のカーネル プログラム RAM、12 KB のユーザデータ RAM、9 KB のユーザ プログラム RAM に分割します。詳細は図 3-13 を参照してください。

各レジスタの値は下記の通りです。

- BMXDRMSZ = 0x00008000 (読み出し専用値)
- BMXDKPBA = 0x00001800 (6 KB のカーネルデータ)
- BMXDUDBA = 0x00002C00 (5 KB のカーネル プログラム)
- BMXDUPBA = 0x00005C00 (ユーザデータのサイズ = 12 KB、ユーザプログラムのサイズ = 9 KB)

図 3-13: RAM の分割 - 6 KB のカーネルデータ、5 KB のカーネル プログラム、12 KB のユーザデータ、9 KB のユーザプログラム



3.5 バスマトリクス

プロセッサは、2つの動作モード（カーネルモードとユーザモード）をサポートします。バスマトリクスは、これらの各モードに対するメモリの割り当てを制御します。また、アドレス空間の特定領域に対するアクセスのタイプ（プログラムまたはデータ）も制御します。

バスマトリクスは、マスタデバイス（一般的にイニシエータと呼ぶ）をスレーブデバイス（一般的にターゲットと呼ぶ）に接続します。PIC32MX 製品ファミリーは、メインバス構造上に最大5個のイニシエータと3個のターゲット（フラッシュ、RAM等）を持つことができます。

5個のイニシエータのうち、CPU 命令バス（CPU IS）、CPU データバス（CPU DS）、インサーキット デバッグ（ICD）、DMA コントローラ（DMA）はイニシエータの既定値設定であり、常に存在します。PIC32MX は、将来のイニシエータの追加に備えて、イニシエータ拡張インターフェイス（IXI）も実装しています。

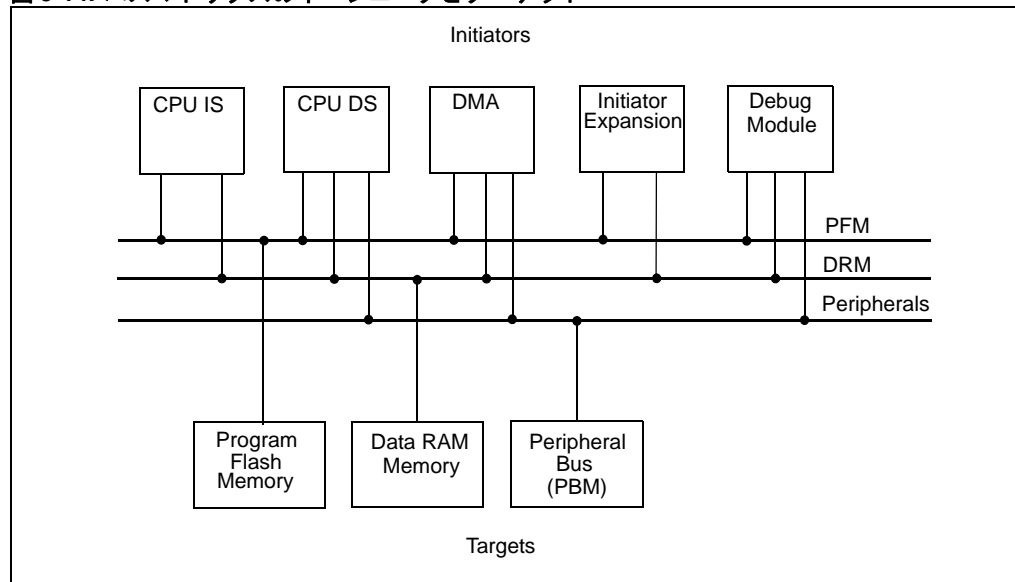
バスマトリクスは、ターゲットに割り当てられるアドレスの一般的レンジをデコードします。ターゲット（メモリまたは周辺モジュール）によっては、その機能に応じて追加のアドレスを提供する場合があります。

表 3-3 に、イニシエータからアクセス可能なターゲットの一覧を示します。

表 3-3: イニシエータ アクセスマップ

イニシエータ	ターゲット		
	フラッシュ	RAM	周辺モジュールバス
CPU IS	Y	Y	N
CPU DS	Y	Y	Y
DMA	Y	Y	Y
IXI	Y	Y	N
ICD	Y	Y	Y

図 3-14: バスマトリクスのイニシエータとターゲット



3.5.1 イニシエータ調停モード

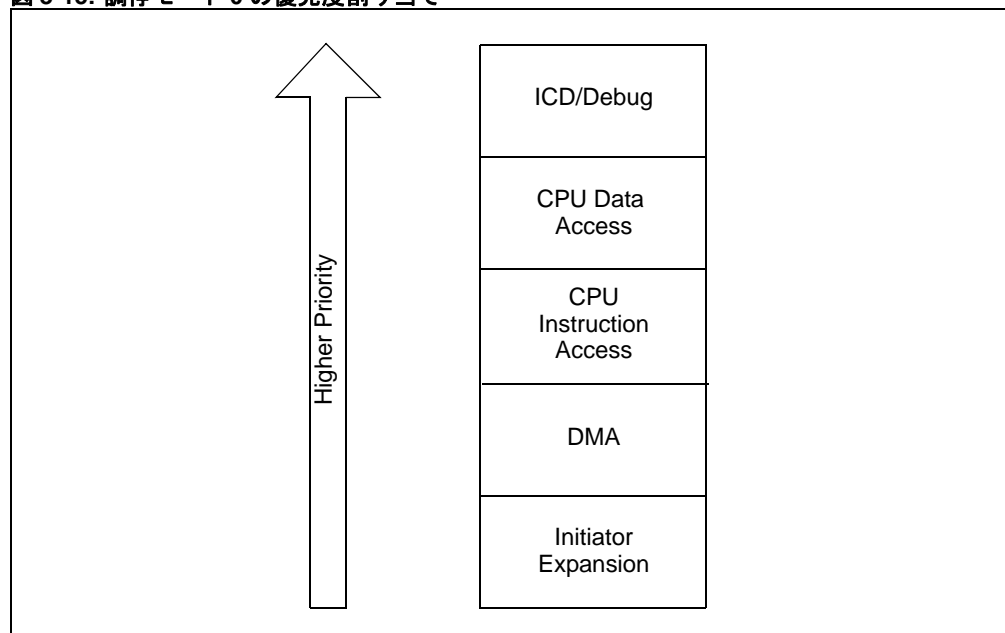
複数のイニシエータが同一ターゲットにアクセスを試みる可能性があるため、ターゲットへのアクセスを制御する調停処理が必要です。このため、各種の調停モードを使って、全てのイニシエータに優先度を割り当てます。調停では、常に優先度がより高いイニシエータがターゲットへのアクセス権を獲得します。

3.5.1.1 調停モード 0

調停モード 0 の優先度の割り当ては、図 3-15 のように固定されています。CPU データおよび CPU 命令アクセスには、DMA アクセスよりも高い優先度が割り当てられます。このモードは DMA を阻害する可能性があるため、DMA を使わない場合に選択できます。

各イニシエータには、図 3-15 に示す優先度が固定的に割り当てられます。調停モード 0 を選択するには、レジスタフィールド BMXARB (BMXCON<2:0>) に「0」を書き込む必要があります。

図 3-15: 調停モード 0 の優先度割り当て

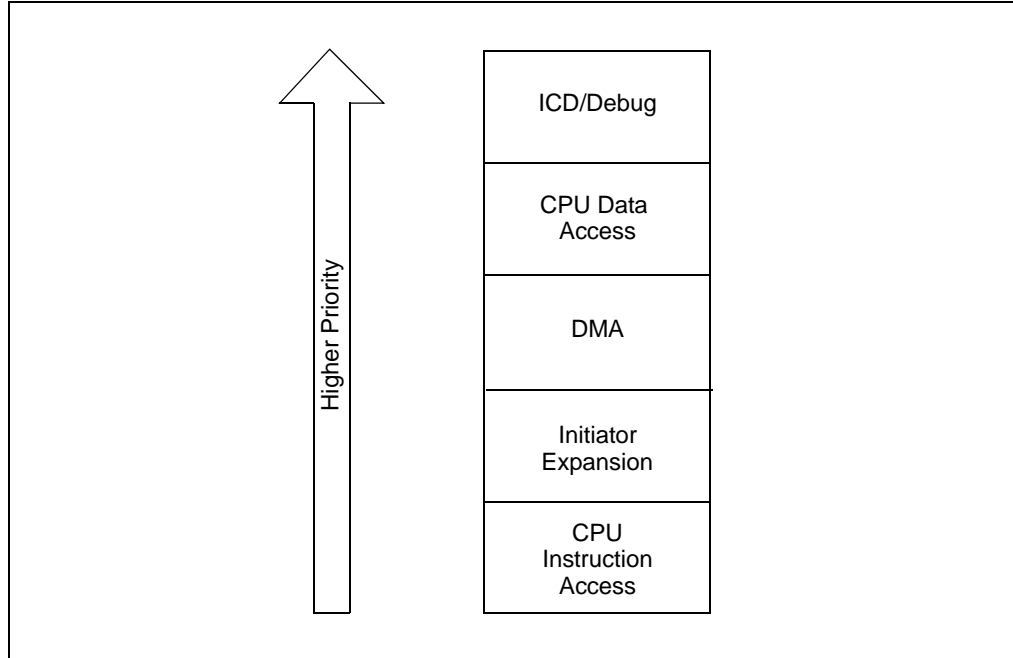


3.5.1.2 調停モード 1

モード 0 と同様に、調停モード 1 の優先度割り当ても固定されていますが、CPU IS に最低優先度が割り当てられます。図 3-16 に、モード 1 の優先度割り当てを示します。調停モード 1 は既定値モードです。

調停モード 1 を選択するには、レジスタフィールド BMXARB (BMXCON<2:1>) に「1」を書き込む必要があります。

図 3-16: 調停モード 1 の優先度割り当て

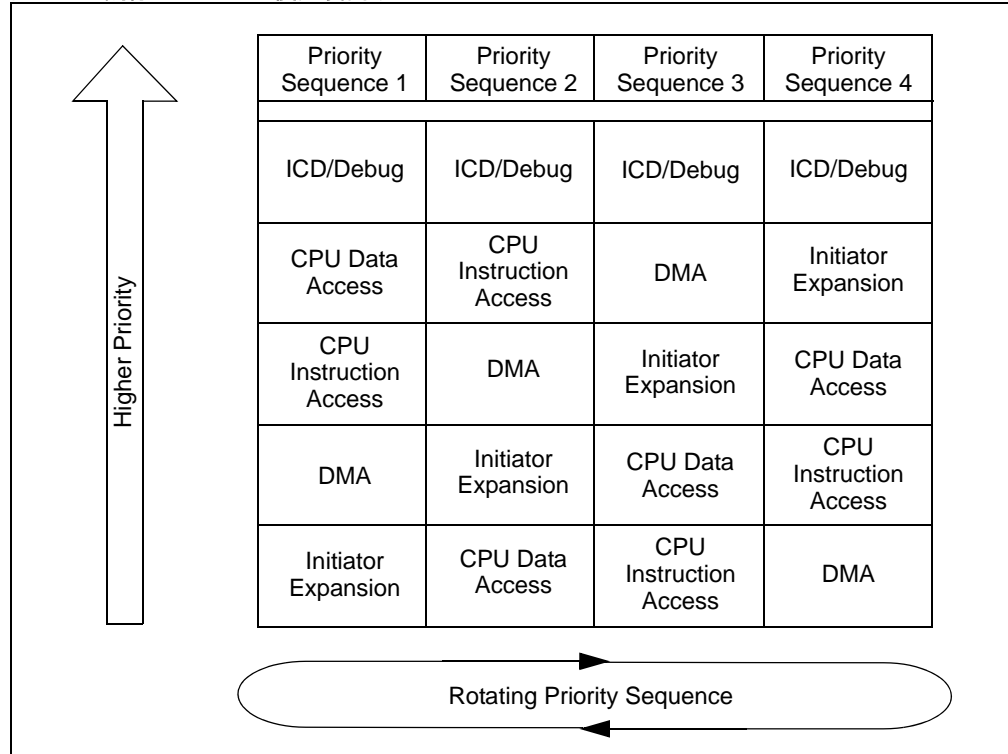


3.5.1.3 調停モード 2

調停モード 2 では、全てのイニシエータの優先度割り当てをローテートできます。つまり、優先度割り当ては 1 つに固定されるのではなく、最優先にするイニシエータを循環式に変更できます。この調停モードでは、優先度を循環式に変更する際に下記の例外を適用します。

1. 常に CPU データ アクセスを CPU 命令アクセスよりも優先する
2. ICD を常に最優先にする
3. CPU が例外 (EXL = 1) またはエラー (ERL = 1) 処理中である時、調停モードは一時的にモード 0 に戻る

図 3-17: 調停モード 2 の優先度割り当て



CPU データアクセスが保留中である場合、優先度割り当てをローテートしても優先度シーケンス 2 は選択されない事に注意が必要です。この場合、保留中のデータアクセスが完了した後にシーケンス 2 が選択されます。

調停モード 2 を選択するには、レジスタフィールド BMXARB (BMXCON<2:2>) に「2」を書き込む必要があります。

3.5.2 バスエラー例外

バスマトリクスは、下記の場合にバスエラー例外を生成します。

- 未実装メモリへのアクセスが発生した場合
- 不正なターゲットへのアクセスが発生した場合
- プログラム フラッシュメモリへの書き込みが発生した場合

BMXCON レジスタの BMXERRxxx ビットをクリアする事により、バスエラー例外を一時的に無効にできますが、このような操作は推奨しません。

バスマトリクスは、デバッグモード時に、CPU IS と CPU DS からのアクセスに対するバスエラー例外を無効にします。

3.5.3 厳密なブレークポイントのサポート

PIC32MX は、データ RAM アクセスに 1 待機ステートを挿入する事により、ブレークポイントにおける厳密なブレークをサポートします。この方法により、ブレークポイントアドレスの命令の直前で、CPU の実行を正確に停止できます。これは、ストア命令にブレークポイントを設定した場合に便利です。待機ステートを使わなくても、ブレークはストア命令位置で発生しますが、DRM 位置はストア値により更新されます。待機ステートを有効にした場合、DRM はストア値により更新されません。

3.6 I/O ピンの制御

このモジュールに関連するピンはありません。

3.7 省電力モードとデバッグモードでの動作

3.7.1 電源投入時またはブラウンアウトリセット (BOR) 時のメモリ動作

- データ RAM の内容は不確定
- BMXxxxBA レジスタは「0」にリセットされる
- CPU はカーネルモードに切り換わる

3.7.2 リセット時のメモリ動作

- データ RAM の内容は保持される。ただし、デバイスがコード保護されている場合、RAM の内容はクリアされる
- BMX ベースアドレス レジスタ (BMXxxxBA) は「0」に設定される
- CPU はカーネルモードに切り換わる

3.7.3 ディープスリープまたはアイドルモードからの復帰時のメモリ動作

- RAM の内容は保持される
- BMX ベースアドレス レジスタ (BMXxxxBA) の内容は変更されない
- CPU モードは変更されない

3.8 サンプルコード

例 3-1: プログラム フラッシュに 12K のユーザモードパーティションを作成する

```
BMXPUPBA = BMXPFMSZ - (12*1024); // User Mode Flash 12K,  
// Kernel Mode Flash 500K (512K-12K)
```

例 3-2: 16K のカーネルモードデータ RAM パーティションを作成する ; カーネル プログラム用の RAM をリセットする

```
BMXDKPBA = 16*1024;  
BMXDUDBA = BMXDRMSZ;  
BMXDUPBA = BMXDRMSZ;
```

例 3-3 は、RAM 内に下記のパーティションを作成するために使えます。

- カーネルモード データ = 12K
- カーネルモード プログラム = 6K
- ユーザモード データ = 8K
- ユーザモード プログラム = 6K

例 3-3: RAM パーティションの作成

```
BMXDKPBA = 12*1024; // Kernel Data Partition of 12K.  
// Start offset of Kernel Program Partition  
BMXDUDBA = BMXDKPBA + (6*1024); // Kernel Program Partition of 6K  
// Start offset of User Data Partition  
BMXDUPBA = BMXDUDBA + (8*1024); // User Data Partition of 8K  
// Start offset of User Program Partition.  
// This partition will go up to the size of  
// RAM (32K).So the partition size will be  
// 6K (32K - 8K - 6K - 12K)
```

3.9 設計のヒント

質問 1: リセット時に CPU はどのモードで動作しますか

回答: CPU はカーネルモードで動作を始めます。この際、RAM の全領域は、KSEG0 および KSEG1 内のカーネルデータ セグメントに割り当てられます。フラッシュメモリは、KSEG0 および KSEG1 内のカーネル プログラム セグメントに割り当てられません。ERL = 1 である場合、ゼロにリセットする必要があります (通常は C 起動コードでリセット)。

質問 2: BMX レジスタは初期化する必要がありますか

回答: 一般的に、その必要はありません。BMX レジスタを既定値から変更しない場合、カーネルモード アプリケーションに最大限の RAM およびフラッシュメモリを割り当てる事ができます。RAM からコードを実行したい場合やユーザモードパーティションを設定したい場合は、BMX レジスタの設定を変更する必要があります。

質問 3: CPU リセットベクタ アドレスを教えてください

回答: CPU リセットアドレスは 0xBFC00000 です。

質問 4: バスエラー例外とは何ですか

回答: バスエラー例外は、CPU が未実装アドレスにアクセスを試みた時に発生します。また、RAM プログラムパーティションが未定義であるにもかかわらず CPU が RAM からの実行を試みた場合にも、バスエラー例外が発生します。

3.10 関連アプリケーションノート

本セクションに関連するアプリケーションノートの一覧を下に記載します。一部のアプリケーションノートは PIC32MX デバイスファミリ向けではありません。ただし、概念は共通しており、変更が必要であったり制限事項が存在するものの利用が可能です。PIC32MX ファミリのメモリ構成に関連する最新のアプリケーションノートは以下の通りです。

タイトル	アプリケーションノート番号
現在、関連するアプリケーションノートはありません。	N/A

Note: PIC32MX ファミリ関連のアプリケーションノートとサンプルコードは、弊社ウェブサイト (www.microchip.com) でご覧になれます。

3.11 改訂履歴

リビジョン A (2007 年 8 月)

本書の初版

リビジョン B (2007 年 10 月)

機密扱いのステータスを解除して内容を更新

リビジョン C (2008 年 4 月)

ステータスを「Preliminary」に変更 ; U-0 を r-x に変更

リビジョン D (2008 年 6 月)

予約済みビットを「Maintain as」から「Write」に変更

リビジョン E (2009 年 7 月)

このリビジョンでの変更内容は以下の通りです。

- 文章および体裁の変更等、本書全体の細部を修正
- 下記の箇所にクリア、セット、反転レジスタに関する Note 1、2、3 を追加
 - 表 3-1: メモリ構成 SFR の概要
 - レジスタ 3-1: BMXCON: バスマトリクス コンフィグレーション レジスタ
 - レジスタ 3-2: BMXDKPBA: データ RAM カーネル プログラム ベースアドレス レジスタ
 - レジスタ 3-3: BMXDUDBA: データ RAM ユーザデータ ベースアドレス レジスタ
 - レジスタ 3-4: BMXDUPBA: データ RAM ユーザ プログラム ベースアドレス レジスタ
 - レジスタ 3-6: BMXPUPBA: プログラム フラッシュ (PFM) ユーザプログラム ベースアドレス レジスタ
- クリア、セット、反転レジスタに関する説明を全て削除
- ビット値定義 (0x0001000) を BMXDRMSZ: データ RAM サイズレジスタに追加 (レジスタ 3-5 参照)

リビジョン F (2012 年 7 月)

このリビジョンでの変更内容は以下の通りです。

- 文章および体裁の変更等、本書全体の細部を修正
- 下記のレジスタの説明に Note 4 と 5 を追加:
 - BMXDKPBA: データ RAM カーネル プログラム ベースアドレス レジスタ (レジスタ 3-2 参照)
 - BMXDUDBA: データ RAM ユーザデータ ベースアドレス レジスタ (レジスタ 3-3 参照)
 - BMXDUPBA: データ RAM ユーザ プログラム ベースアドレス レジスタ (レジスタ 3-4 参照)
 - BMXPUPBA: プログラム フラッシュ (PFM) ユーザプログラム ベースアドレス レジスタ (レジスタ 3-6 参照)
- PIC32MX アドレスマップを更新 (表 3-2 参照)

NOTE:

マイクロチップ社製デバイスのコード保護機能に関して次の点にご注意ください。

- マイクロチップ社製品は、該当するマイクロチップ社データシートに記載の仕様を満たしています。
- マイクロチップ社では、通常の条件ならびに仕様に従って使用した場合、マイクロチップ社製品のセキュリティレベルは、現在市場に流通している同種製品の中でも最も高度であると考えています。
- しかし、コード保護機能を解除するための不正かつ違法な方法が存在する事もまた事実です。弊社の理解ではこうした手法は、マイクロチップ社データシートにある動作仕様書以外の方法でマイクロチップ社製品を使用する事になります。このような行為は知的所有権の侵害に該当する可能性が非常に高いと言えます。
- マイクロチップ社は、コードの保全性に懸念を抱くお客様と連携し、対応策に取り組んでいきます。
- マイクロチップ社を含む全ての半導体メーカーで、自社のコードのセキュリティを完全に保証できる企業はありません。コード保護機能とは、マイクロチップ社が製品を「解読不能」として保証するものではありません。

コード保護機能は常に進歩しています。マイクロチップ社では、常に製品のコード保護機能の改善に取り組んでいます。マイクロチップ社のコード保護機能の侵害は、デジタル ミレニアム著作権法に違反します。そのような行為によってソフトウェアまたはその他の著作物に不正なアクセスを受けた場合は、デジタル ミレニアム著作権法の定めるところにより損害賠償訴訟を起こす権利があります。

本書に記載されているデバイス アプリケーション等に関する情報は、ユーザの便宜のためにのみ提供されているものであり、更新によって無効とされる事があります。お客様のアプリケーションが仕様を満たす事を保証する責任は、お客様にあります。マイクロチップ社は、明示的、暗黙的、書面、口頭、法定のいずれであるかを問わず、本書に記載されている情報に関して、状態、品質、性能、品性、特定目的への適合性をはじめとする、いかなる類の表明も保証も行いません。マイクロチップ社は、本書の情報およびその使用に起因する一切の責任を否認します。マイクロチップ社の明示的な書面による承認なしに、生命維持装置あるいは生命安全用途にマイクロチップ社の製品を使用する事は全て購入者のリスクとし、また購入者はこれによって発生したあらゆる損害、クレーム、訴訟、費用に関して、マイクロチップ社は擁護され、免責され、損害うけない事に同意するものとします。暗黙的あるいは明示的を問わず、マイクロチップ社が知的財産権を保有しているライセンスは一切譲渡されません。

商標

マイクロチップ社の名称と Microchip ロゴ、dsPIC、KEELOQ、KEELOQ ロゴ、MPLAB、PIC、PICmicro、PICSTART、rfPIC、UNI/O は、米国およびその他の国におけるマイクロチップ・テクノロジー社の登録商標です。

FilterLab、Hampshire、HI-TECH C、Linear Active Thermistor、MXDEV、MXLAB、SEEVAL、Embedded Control Solutions Company は、米国におけるマイクロチップ・テクノロジー社の登録商標です。

Analog-for-the-Digital Age、Application Maestro、chipKIT、chipKIT logo、CodeGuard、dsPICDEM、dsPICDEM.net、dsPICworks、dsSPEAK、ECAN、ECONOMONITOR、FanSense、HI-TIDE、In-Circuit Serial Programming、ICSP、Mindī、MiWi、MPASM、MPLAB Certified ロゴ、MPLIB、MPLINK、mTouch、Omniscient Code Generation、PICC、PICC-18、PICDEM、PICDEM.net、PICkit、PICtail、REAL ICE、rLAB、Select Mode、Total Endurance、TSHARC、UniWinDriver、WiperLock、ZENA は、米国およびその他の国におけるマイクロチップ・テクノロジー社の登録商標です。

SQTP は、米国におけるマイクロチップ・テクノロジー社のサービスマークです。

その他、本書に記載されている商標は各社に帰属します。

© 2011, Microchip Technology Incorporated, All Rights Reserved.

ISBN: 9781-62076-175-5

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
== ISO/TS 16949:2009 ==

マイクロチップ社では、Chandler および Tempe (アリゾナ州)、Gresham (オレゴン州) の本部、設計部およびウェハー製造工場そしてカリフォルニア州とイダのデザインセンターが ISO/TS-16949:2009 認証を取得しています。マイクロチップ社の品質システム プロセスおよび手順は、PIC® MCU および dsPIC® DSC、KEELOQ® コードホッピングデバイス、シリアル EEPROM、マイクロペリフェラル、不揮発性メモリ、アナログ製品に採用されています。さらに、開発システムの設計と製造に関するマイクロチップ社の品質システムは ISO 9001:2000 認証を取得しています。



MICROCHIP

各国の営業所とサービス

北米

本社
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel:480-792-7200
Fax:480-792-7277
技術サポート：
[http://www.microchip.com/
support](http://www.microchip.com/support)
URL:
www.microchip.com

アトランタ
Duluth, GA
Tel:678-957-9614
Fax:678-957-1455

ボストン
Westborough, MA
Tel:774-760-0087
Fax:774-760-0088

シカゴ
Itasca, IL
Tel:630-285-0071
Fax:630-285-0075

クリーブランド
Independence, OH
Tel:216-447-0464
Fax:216-447-0643

ダラス
Addison, TX
Tel:972-818-7423
Fax:972-818-2924

デトロイト
Farmington Hills, MI
Tel:248-538-2250
Fax:248-538-2260

インディアナポリス
Noblesville, IN
Tel:317-773-8323
Fax:317-773-5453

ロサンゼルス
Mission Viejo, CA
Tel:949-462-9523
Fax:949-462-9608

サンタクララ
Santa Clara, CA
Tel:408-961-6444
Fax:408-961-6445

トロント
Mississauga, Ontario,
Canada
Tel:905-673-0699
Fax:905-673-6509

アジア/太平洋

アジア太平洋支社
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon
Hong Kong
Tel:852-2401-1200
Fax:852-2401-3431

オーストラリア - シドニー
Tel:61-2-9868-6733
Fax:61-2-9868-6755

中国 - 北京
Tel:86-10-8569-7000
Fax:86-10-8528-2104

中国 - 成都
Tel:86-28-8665-5511
Fax:86-28-8665-7889

中国 - 重慶
Tel:86-23-8980-9588
Fax:86-23-8980-9500

中国 - 杭州
Tel:86-571-2819-3187
Fax:86-571-2819-3189

中国 - 香港 SAR
Tel:852-2401-1200
Fax:852-2401-3431

中国 - 南京
Tel:86-25-8473-2460
Fax:86-25-8473-2470

中国 - 青島
Tel:86-532-8502-7355
Fax:86-532-8502-7205

中国 - 上海
Tel:86-21-5407-5533
Fax:86-21-5407-5066

中国 - 瀋陽
Tel:86-24-2334-2829
Fax:86-24-2334-2393

中国 - 深圳
Tel:86-755-8203-2660
Fax:86-755-8203-1760

中国 - 武漢
Tel:86-27-5980-5300
Fax:86-27-5980-5118

中国 - 西安
Tel:86-29-8833-7252
Fax:86-29-8833-7256

中国 - 厦門
Tel:86-592-2388138
Fax:86-592-2388130

中国 - 珠海
Tel:86-756-3210040
Fax:86-756-3210049

アジア/太平洋

インド - バンガロール
Tel:91-80-3090-4444
Fax:91-80-3090-4123

インド - ニューデリー
Tel:91-11-4160-8631
Fax:91-11-4160-8632

インド - プネ
Tel:91-20-2566-1512
Fax:91-20-2566-1513

日本 - 大阪
Tel:81-66-152-7160
Fax:81-66-152-9310

日本 - 横浜
Tel:81-45-471-6166
Fax:81-45-471-6122

韓国 - 大邱
Tel:82-53-744-4301
Fax:82-53-744-4302

韓国 - ソウル
Tel:82-2-554-7200
Fax:82-2-558-5932 または
82-2-558-5934

マレーシア - クアラルンプール
Tel:60-3-6201-9857
Fax:60-3-6201-9859

マレーシア - ペナン
Tel:60-4-227-8870
Fax:60-4-227-4068

フィリピン - マニラ
Tel:63-2-634-9065
Fax:63-2-634-9069

シンガポール
Tel:65-6334-8870
Fax:65-6334-8850

台湾 - 新竹
Tel:886-3-5778-366
Fax:886-3-5770-955

台湾 - 高雄
Tel:886-7-536-4818
Fax:886-7-330-9305

台湾 - 台北
Tel:886-2-2500-6610
Fax:886-2-2508-0102

タイ - バンコク
Tel:66-2-694-1351
Fax:66-2-694-1350

ヨーロッパ

オーストリア - ヴェルス
Tel:43-7242-2244-39
Fax:43-7242-2244-393

デンマーク - コペンハーゲン
Tel:45-4450-2828
Fax:45-4485-2829

フランス - パリ
Tel:33-1-69-53-63-20
Fax:33-1-69-30-90-79

ドイツ - ミュンヘン
Tel:49-89-627-144-0
Fax:49-89-627-144-44

イタリア - ミラノ
Tel:39-0331-742611
Fax:39-0331-466781

オランダ - ドリュエーン
Tel:31-416-690399
Fax:31-416-690340

スペイン - マドリッド
Tel:34-91-708-08-90
Fax:34-91-708-08-91

イギリス - ウォーキングム
Tel:44-118-921-5869
Fax:44-118-921-5820