



本ドキュメントは Cypress (サイプレス) 製品に関する情報が記載されております。本ドキュメントには、「MB」から始まるシリーズ名、品名およびオーダ型格が記載されておりますが、これらはすべて「CY」から始まるシリーズ名、品名およびオーダ型格として、新規および既存のお客様に引き続き提供してまいります。

オーダ型格の調べ方について

1. www.cypress.com/pcnにアクセスしてください。
2. SEARCH PCNS フィールドに、オーダ型格などのキーワードを入力し、「Apply」をクリックしてください。
3. 該当するタイトル(Title)をクリックしてください。
4. 「Affected Parts List」ファイルを開いてください。
当該ファイルに記載されている各種変更情報をご利用ください。

詳しいお問い合わせ先

Cypress 製品およびそのソリューションの詳細につきましては、お近くの営業所へお問い合わせください。

サイプレスについて

サイプレスは、世界で最も革新的な車載や産業機器、スマート家電、民生機器および医療機器製品向けに、最先端の組み込みシステム ソリューションを提供するリーディングカンパニーです。サイプレスのマイクロコントローラーや、アナログ IC、ワイヤレスおよび USB ベースのコネクティビティ ソリューション、高い信頼性と高性能を提供するメモリ製品は、各種機器メーカーの差異化製品の開発と早期市場参入を支援します。サイプレスは、ベストクラスのサポートと開発リソースをグローバルに提供することで、彼らが従来市場を破壊しまったく新しい製品カテゴリを歴史的なスピードで市場投入できるよう支援します。詳細はサイプレスのウェブサイト (japan.cypress.com) をご覧ください。

2 CPU モード時の注意 Traveo™ Family MB9D560 シリーズ

関連ファミリ:	シリーズ名	品種型格
	MB9D560	MB9DF564MAE/F565MAE/F566MAE MB9DF564MGE/F565MGE/F566MGE

本アプリケーションノートは、サイプレス製マイコン Traveo ファミリ MB9D560 の利用を検討されている方を対象としています。

Contents

1 はじめに.....1	3.3 推奨する方法を行わなかった場合.....6
1.1 この文書について.....1	4 ペリフェラルの使用につて.....11
1.2 開発環境.....1	4.1 推奨するペリフェラル使用方法.....11
2 メモリアクセスについて.....1	4.2 推奨する方法を行わなかった場合.....15
2.1 メモリアクセス経路.....1	5 参考文書.....20
2.2 推奨するメモリアクセス方法.....2	6 付録.....20
2.3 推奨する方法を行わなかった場合.....3	6.1 コア間で同期をとる方法.....20
3 レジスタアクセスについて.....5	7 改訂履歴.....24
3.1 ビットバンドユニット.....5	Worldwide Sales and Design Support.....25
3.2 推奨するレジスタアクセス方法.....5	

1 はじめに

1.1 この文書について

本アプリケーションノートは、サイプレス製マイコン Traveo ファミリ MB9D560 の利用を検討されている方を対象としています。

MB9D560 は CPU コアを 2 個搭載しています。そのため、ペリフェラルやメモリの使用にはデュアルコアマイコン特有の注意が必要になります。

本アプリケーションノートでは CPU コアが複数あることによるソフトウェア実装時の注意点について記載します。

1.2 開発環境

本アプリケーションノートに記載されている内容は表 1 に示す環境で開発されています。

表 1 開発環境

マイコン	MB9DF566MGB
統合開発環境	MULTI v6.1.4
評価ボード	MB2198-770-02-E0
最適化	Optimize for Speed

2 メモリアクセスについて

2.1 メモリアクセス経路

本マイコンには以下のメモリが実装されています。

- TCRAM

- TCFLASH
- EAM
- WorkFLASH

全てのコアが全てのメモリに対してアクセスすることが可能です。

TCRAM、TCFLASH はコアごとに実装されており、アクセス経路によって制限事項があります。経路ごとの制限事項は表 2 の通りです。

図 1 メモリ接続図

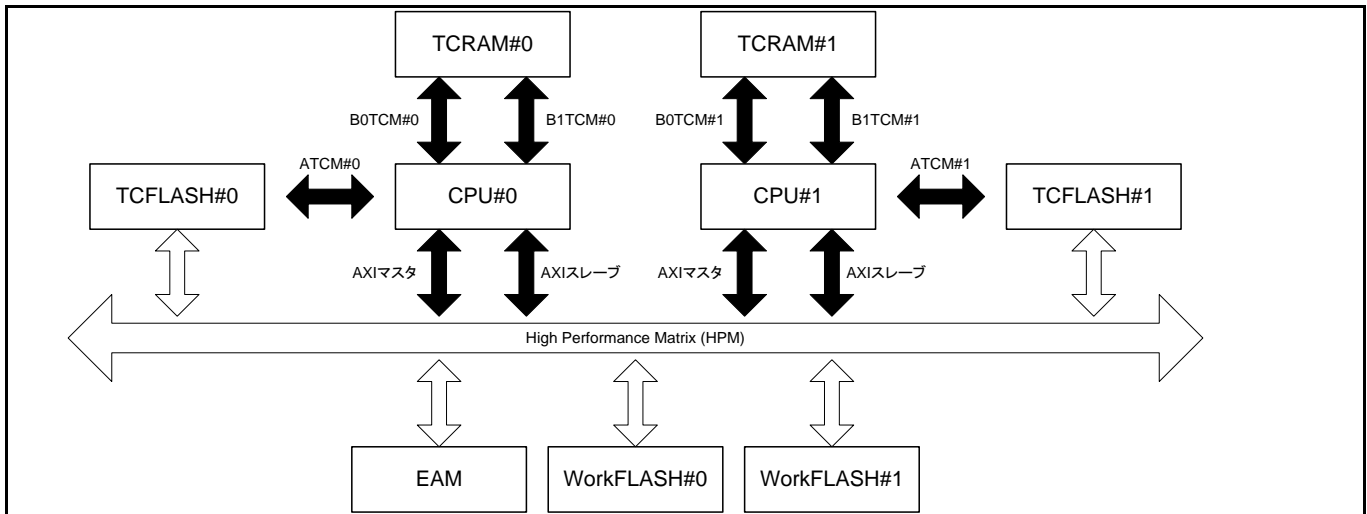


表 2 メモリアccess経路と禁止事項

アクセス先	経路	禁止事項
自コアの TCRAM	自コア-TCM-TCRAM	-
他コアの TCRAM	自コア-AXI マスター-AXI スレーブ-他コア-TCM-TCRAM	-
自コアの TCFLASH	自コア-TCM-TCFLASH	ライトアクセス
	自コア-AXI マスター-TCFLASH	リードアクセス
他コアの TCFLASH	自コア-AXI マスター-AXI スレーブ-他コア-TCM-TCFLASH	ライトアクセス
	自コア-AXI マスター-TCFLASH	リードアクセス
EAM	自コア-AXI マスター-EAM	-
WorkFLASH	自コア-AXI マスター-WorkFLASH	-

2.2 推奨するメモリアccess方法

以下のようにメモリの用途に合わせて、CPU のメモリ保護 (MPU) を使用するかメモリアccess時にコア間で同期をとるようにしてください。これによりメモリ内のデータが意図しない値に書き換わることを防ぐことが可能です。

CPU のメモリ保護 (MPU) の使用方法については ARM® Architecture Reference Manual ARM® v7-A and ARM® v7-R edition (ARM DDI 0406B)を参照してください。

コア間での同期のとりかたは「6.1 コア間で同期をとる方法」を参照してください。

- メモリに対して単一コアのみがアクセスする場合
CPU のメモリ保護 (MPU) を使用して自コアが対象外の領域に対してアクセスしないようにしてください。
- メモリに対して単一コアのみが書き込みを行う場合
CPU のメモリ保護 (MPU) を使用して自コアが対象外の領域に対して書き込みを行わないようにしてください。

- 同じメモリに対して複数コアが書き込みを行う場合
コアごとに書き込みを行うアドレスを分けることで、同じアドレスに対して複数コアが書き込みを行わないようにしてください。
CPUのメモリ保護 (MPU) を使用して自コアが対象外の領域に対して書き込みを行わないようにしてください。
- 同じアドレスに対して複数コアが書き込みを行う場合
コア間で同期をとることで、一方のコアが書き込み処理実行中に他コアが同じアドレスに対して書き込みを行わないようにしてください。

表 3 の機能を使用する場合は以下のようにしてください。

- コア間で同期を取り、全てのコアが操作対象のメモリにアクセスしていないことを確認してから機能を実行してください。
- 機能実行中は全てのコアが操作対象のメモリにアクセスすることがないようにしてください。

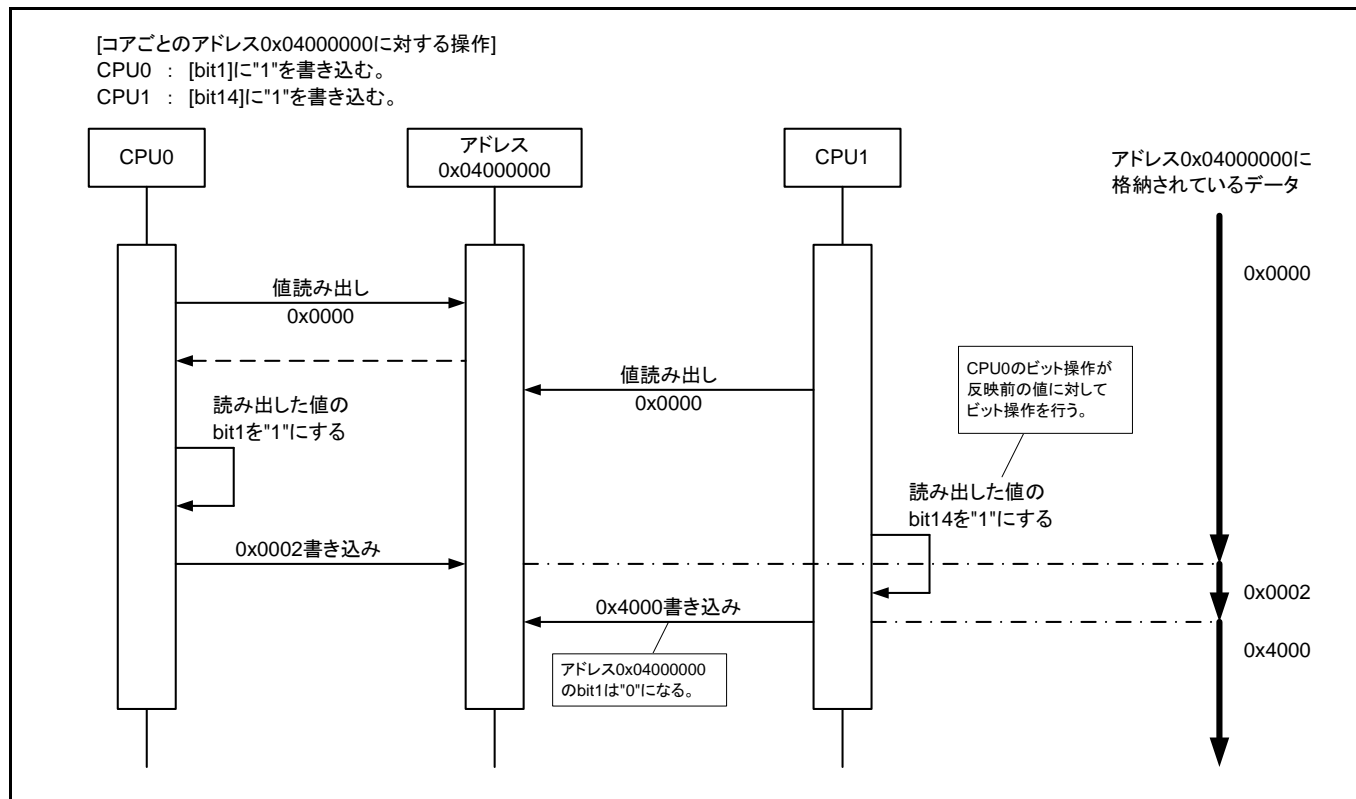
2.3 推奨する方法を行わなかった場合

「2.2 推奨するメモリアクセス方法」に記載している方法を行わなかった場合に起こりうる問題について以下に記します。

2.3.1 複数コアからリードモディファイライトを行う場合

複数のコアから同じアドレスに対してリードモディファイライトを行うと、タイミングによって操作対象ビット以外のビットの値を変更してしまう可能性があります。図 2 に例を記します。

図 2 同じアドレスへのリードモディファイライト例



2.3.2 コアが意図しない動作をする場合

コアが意図しない動作をした場合、コアからの意図しない書込みによってメモリ内のデータが破壊されてしまうことがあります。CPU のメモリ保護 (MPU) を使用することによって、自コアが対象外の領域に対して書込みを行うことを防ぐことができます。

2.3.3 メモリアクセス禁止条件を満たす場合

一方のコアによって表 3 の条件が満たされているときにもう一方のコアが表 3 の禁止事項を実行した場合、バスエラーとなります。図 3 に例を記します。

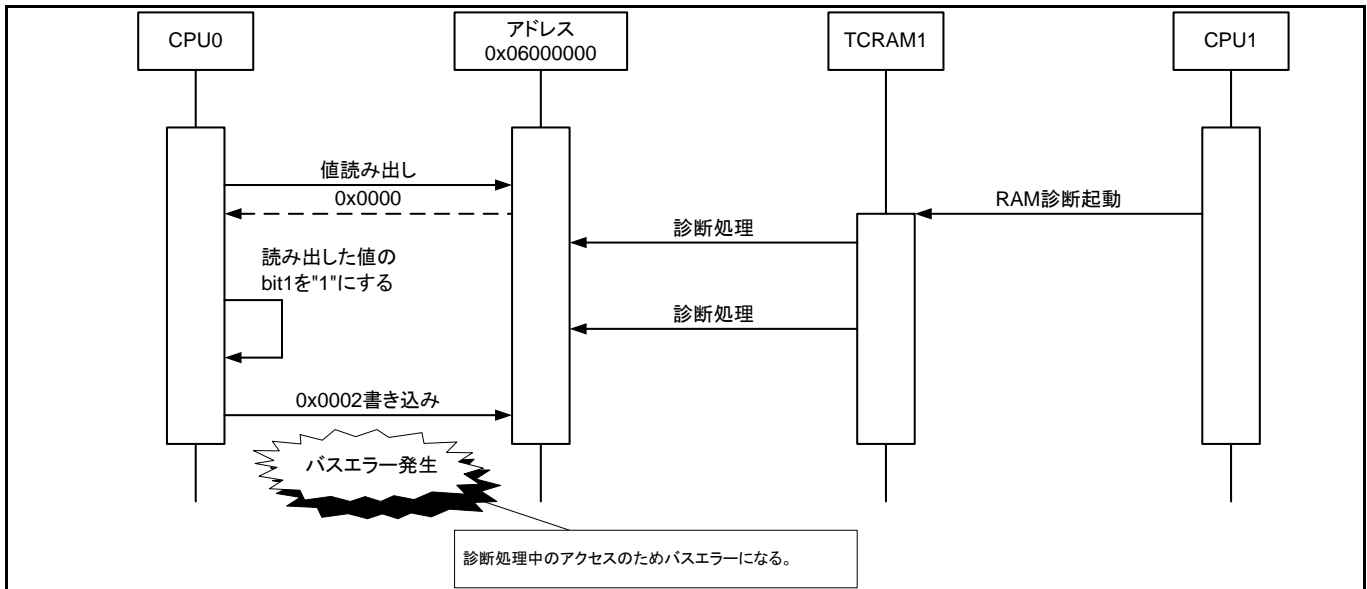
表 3 メモリアクセス禁止条件

ペリフェラル	条件	禁止事項
TCFLASH	TCFCFGn_FCFGR の WE ビットが"0"	TCFLASH への書込みまたは消去
	TCFCFGn_FCFGR の WE ビットが"1"	TCFLASH への TCM インタフェース経由の読出し
	自動アルゴリズム実行中	TCFLASH への TCM インタフェース経由の読出し
TCRAM	RAM 診断中	RAM へのアクセス
	RAM 初期化中	RAM へのアクセス
WorkFLASH	WFCFGxx_CR の WE ビットが"0"	WorkFLASH への書込み
	コマンドシーケンサがアイドル状態以外	WorkFLASH へのアクセス

n: ペリフェラルが接続されている Cortex-R5F コアの番号(n=0, 1)

xx: ユニット番号(xx=00, 01, 02, 03)

図 3 RAM アクセス中の RAM 診断実行例



3 レジスタアクセスについて

3.1 ビットバンドユニット

本マイコンではレジスタに対してビット操作を行う方法にビットバンドユニットを使用する方法があります。ビットバンドユニットによりビットバンドエイリアス領域の 1 バイトにアクセスすることで、ビットバンド領域の対応する 1 ビットを操作することができます。ビットバンドユニットによるビット操作では操作対象ビット以外のビットの値は変化しません。ビットバンド対応しているレジスタのビット操作を行う場合はビットバンドユニットを使用してください。

ビットバンドユニットの使用方法はハードウェアマニュアルを参照してください。

※提供している IO ヘッダーファイル内の定義を使用してレジスタアクセスを行った場合、ビットバンド対応しているレジスタへのアクセスはビットバンドユニットを使用したアクセスになります。

3.2 推奨するレジスタアクセス方法

レジスタへのアクセスは以下のようにしてください。

- 複数コアによるレジスタアクセスについて
コアごとにアクセスするレジスタを分け、複数コアから同じレジスタに対してアクセスすることがないようにしてください。
- ビットバンド対応しているレジスタのビット操作について
ビットバンド対応しているレジスタのビット操作を行う場合はビットバンドユニットを使用してください。
- 保護解除レジスタへの書き込みタイミングについて
保護解除レジスタ (表 7) への書き込み後は、他の処理を行うことなく保護対象レジスタへの書き込みを行ってください。
- Memory & Config Group の保護解除レジスタへの書き込みについて
Memory & Config Group には IRC が含まれているため、Memory & Config Group にあるレジスタの保護を解除する前に必ず全ての割り込みを禁止にしておいてください。

3.3 推奨する方法を行わなかった場合

「3.2 推奨するレジスタアクセス方法」に記載している方法を行わなかった場合に起こりうる問題について以下に記します。

3.3.1 ビットバンドユニットを使用しない場合

C 言語でビットバンドユニットを使用しないレジスタアクセスによるビット操作を行うコードを実装した場合、コンパイル後のアセンブラコードではリードモディファイライトの処理になります。そのため、複数コアから同じレジスタに対してビット操作を行うと、タイミングによって操作対象ビット以外のビットの値を変更してしまう可能性があります。図 4、図 5 に例を記します。

図 4 ビット操作を行うソースコード例

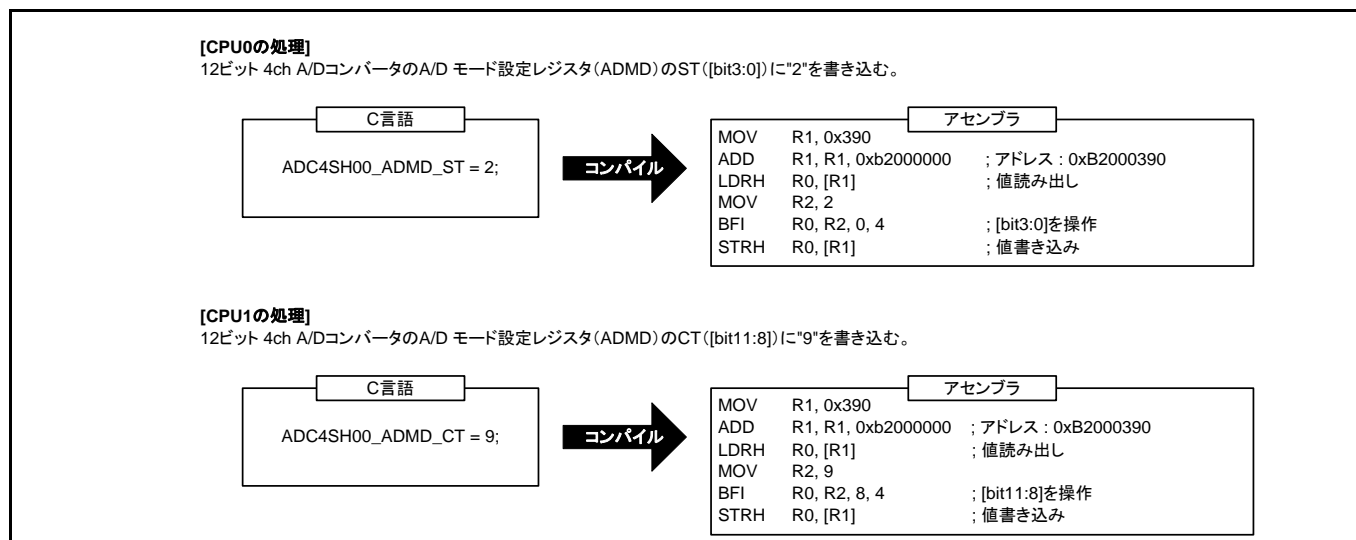
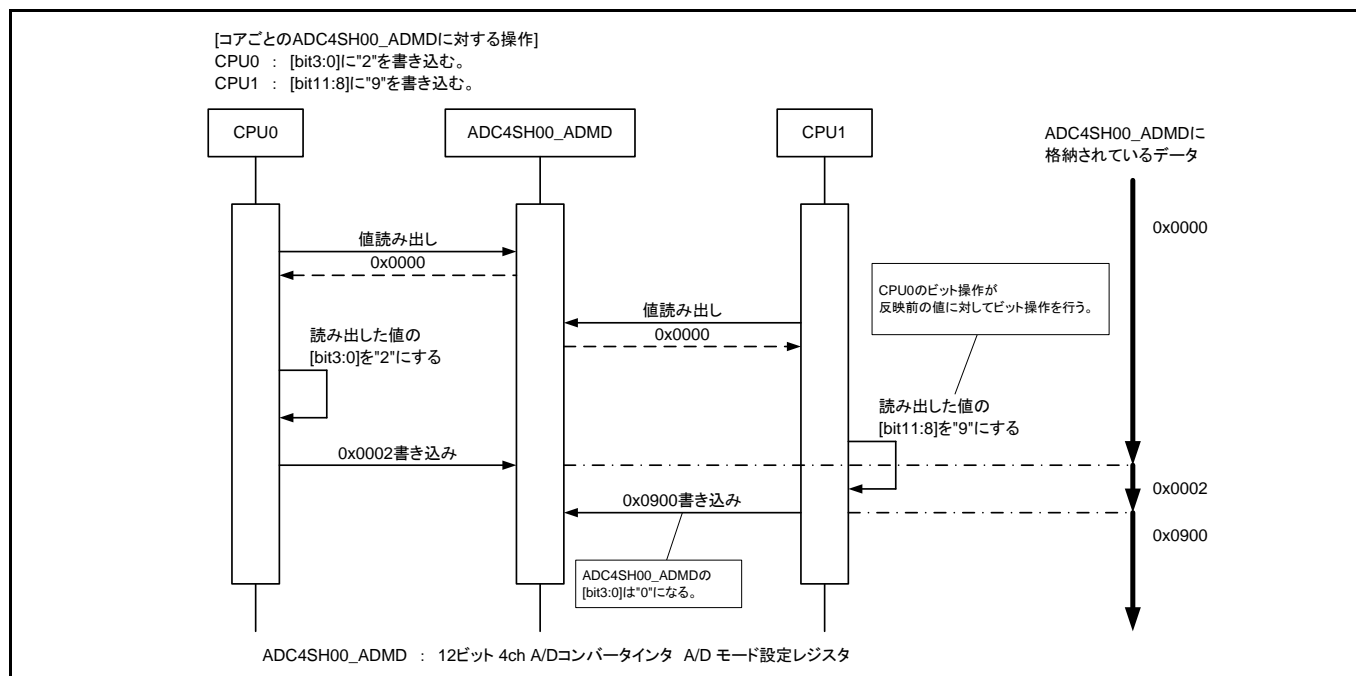


図 5 同じレジスタへのビット操作例



3.3.2 キーコード入力レジスタに書き込む場合

キーコード入力レジスタ（表 4）を動作させるにはキーコードを決められた順番で連続的に書き込む必要があります。複数コアから同じキーコード入力レジスタに対してキーコードの書き込みを行うと、タイミングによってキーコードの書き込みが意図しない順番になる可能性があります。図 6 に例を記します。

以下のレジスタはキーコード機能を搭載していない型格ではキーコードの書き込みは不要です。キーコード機能を搭載していない型格では以下のレジスタに対する書き込みは無効となります。型格ごとのキーコード機能の有無はデータシートの型格オプションを参照してください。

- GPIO_KEYCDR
- PPC_KEYCDR
- KEYCDR
- DACxx_KEYCDR (xx=00, 01)

表 4 キーコード入力レジスタ

レジスタ略称	レジスタ名	機能
GPIO_KEYCDR *1	GPIO キーコードレジスタ	対象レジスタの誤書き込み保護解除
PPC_KEYCDR *1	PPC キーコードレジスタ	対象レジスタの誤書き込み保護解除
LAMERT	LIN アシストモード障害試験レジスタ	疑似障害設定の有効/無効
CIF1	制御レジスタ	FlexRay マクロに対してリセットを出力
LCK	ロックレジスタ	READY コマンドの書き込み許可
KEYCDR *1	キーコードレジスタ	対象レジスタの誤書き込み保護解除
WFGnn_DTMNS	16 ビットデッドタイムマイナス制御レジスタ	デッドタイム機能のマイナス制御に対する設定反映
DACxx_KEYCDR *1	キーコードレジスタ	対象レジスタの誤書き込み保護解除
MVACC	演算制御指示レジスタ	定数パラメータ保護対象レジスタの保護解除

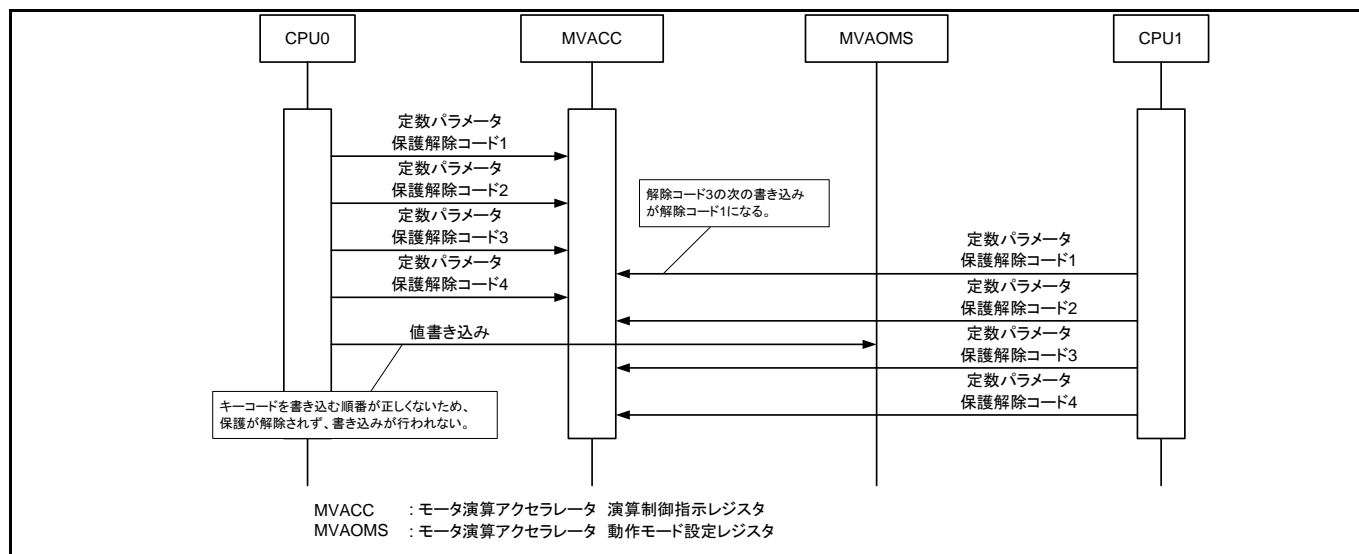
nn:チャンネル番号(xx=00~03)

xx:チャンネル番号(xx=00, 01)

*1:キーコード機能搭載型格のみキーコードの書き込みが必要。

※コアごとにレジスタが分かれているレジスタは他コアからアクセスするとバスエラーとなるため表 4 には記載していません。

図 6 キーコード入力レジスタへの書き込み例



3.3.3 ロック解除レジスタに書き込む場合

ロック解除レジスタ（表 5）が保護対象としているレジスタに対して書き込みを行う場合、ロック解除後に書き込みを行う必要があります。複数コアから同じロック解除レジスタに対してロック解除値およびロック値の書き込みを行うと、タイミングによって保護対象レジスタへの書き込みが行われない可能性があります。図 7 に例を記します。

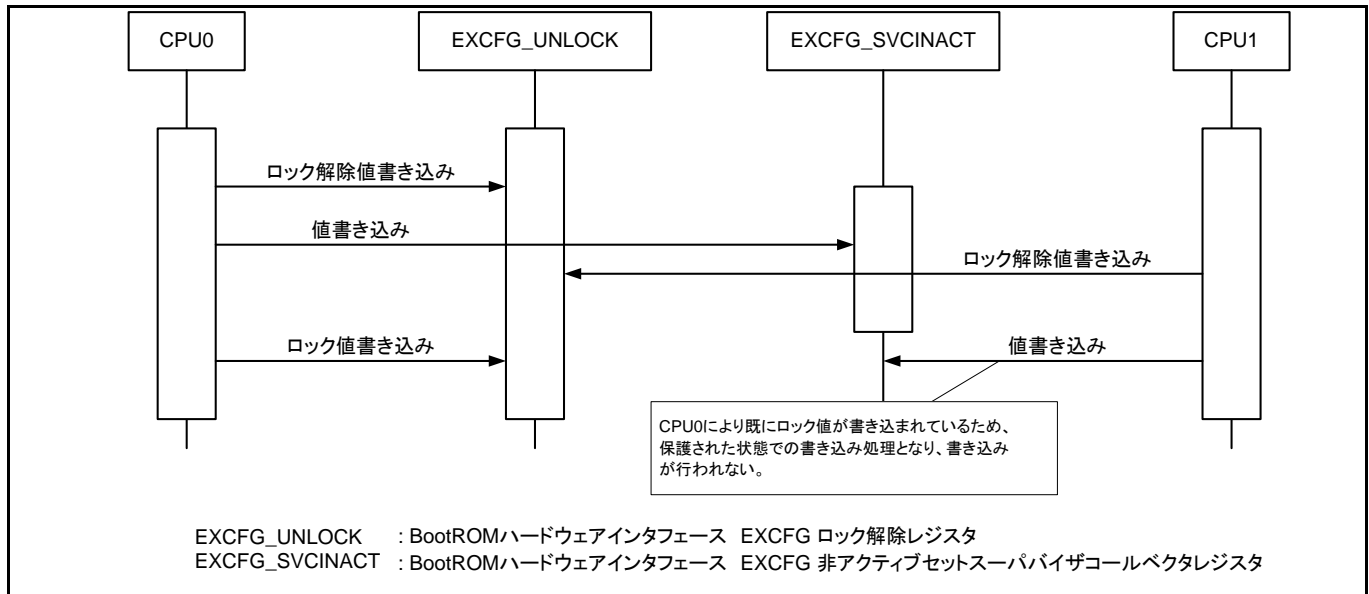
表 5 ロック解除レジスタ

レジスタ略称	レジスタ名	機能
EXCFG_UNLOCK	EXCFG ロック解除レジスタ	BootROM ハードウェアインタフェースの各レジスタを保護
NMID_UNLOCK	NMID ロック解除レジスタ	NMI 分配ユニットの各レジスタを保護
MPUHm_UNLOCK	MPU AHB ロック解除レジスタ	MPU AHB モジュールの各レジスタを保護
MVACC	演算制御指示レジスタ	定数パラメータ保護対象レジスタを保護

m:モジュールのインスタンス"m"であることを示す(m=0)

※コアごとにレジスタが分かれているレジスタは他コアからアクセスするとバスエラーとなるため表 5 には記載していません。

図 7 ロック解除レジスタへの書き込み例



3.3.4 連続書き込み禁止レジスタに書き込む場合

複数コアから連続書き込み禁止レジスタ（表 6）に対して書き込みを行うと、タイミングによって連続書き込み禁止レジスタへ連続で書き込みが行われる可能性があります。図 8 に例を記します。

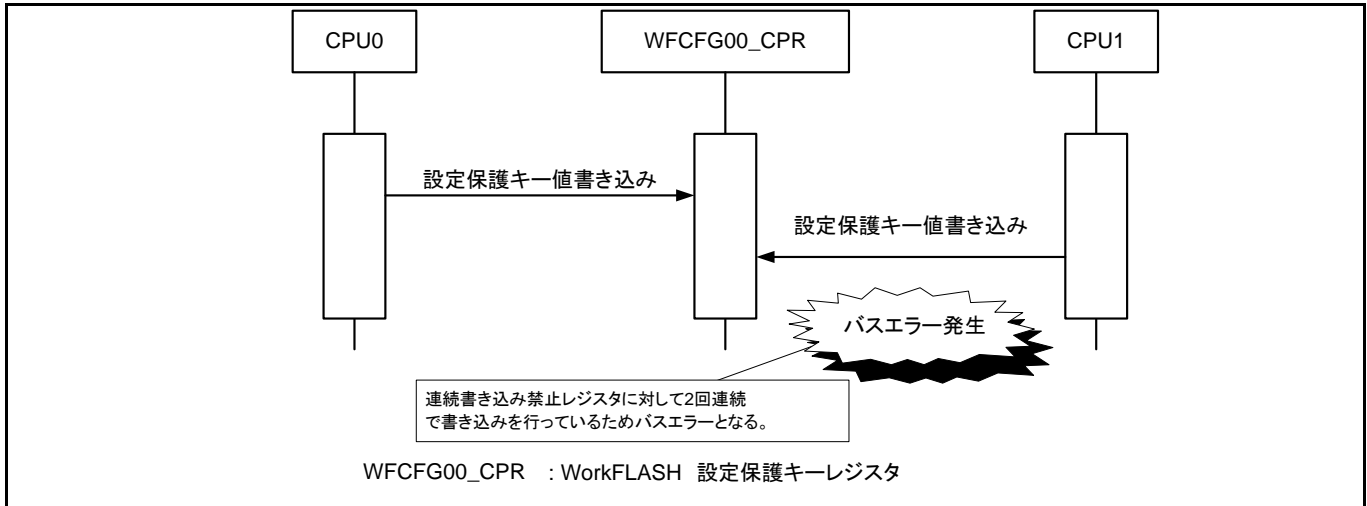
表 6 連続書き込み禁止レジスタ

レジスタ略称	レジスタ名	機能
WFCFGxx_CPR	WorkFLASH 設定保護キーレジスタ	対象レジスタへの設定値書き込み保護解除

xx:ユニット番号(xx=00, 01, 02, 03)

※コアごとにレジスタが分かれているレジスタは他コアからアクセスするとバスエラーとなるため表 6 には記載していません。

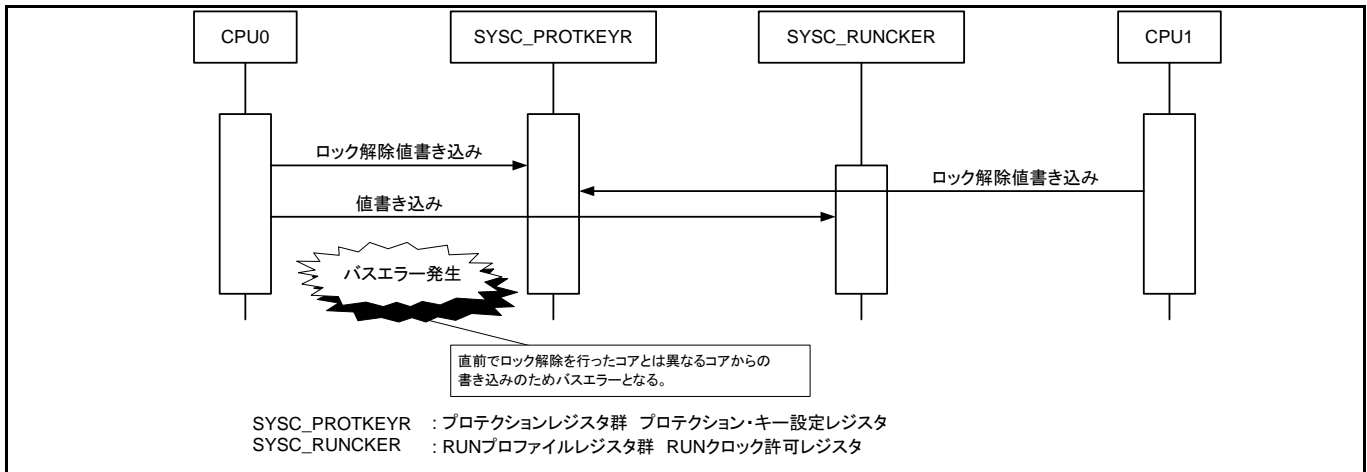
図 8 連続書き込み禁止レジスタへの書き込み例



3.3.5 プロテクション・キー設定レジスタに書き込む場合

複数コアからプロテクション・キー設定レジスタ (SYSC_PROTKEYR) に対して書き込みを行うと、タイミングによってロック解除値を書き込んだコアとプロテクション対象レジスタへ書き込みを行ったコアが一致しない可能性があります。図 9 に例を記します。

図 9 プロテクション・キー設定レジスタへの書き込み例



3.3.6 保護解除レジスタを使用する場合

ロック値を書き込むことで保護対象レジスタをロックするレジスタとは異なり、表 7 の保護解除レジスタは以下のいずれかの処理が行われると保護対象レジスタへの保護を開始します。

- 保護対象レジスタへの書き込み
- 保護対象レジスタと同じグループのレジスタへの書き込み

他コアからのアクセスによっても保護を開始するものがあり、保護された状態で保護対象レジスタへの書き込みを行うとバスエラーとなります。

特に Memory & Config Group には IRC が含まれているため、割り込みが発生することによって保護を解除していたレジスタが再度保護されてしまいます。そのため、Memory & Config Group に属するレジスタの保護を解除する場合は、必ず事前に全ての割り込みを禁止にしておいてください。

表 7 保護対象レジスタ書き込みブロック開始条件

ペリフェラル名 [グループ名]	保護解除レジスタ名 <略称>	書き込み処理ごとの保護開始有無			
		自コア Tw	自コア Gw	他コア Tw	他コア Gw
System Controller (SYSC) [MCU Config Group]	プロテクション・キー設定 レジスタ <SYSC_PROTKEYR>	保護開始	保護開始	バスエラー	保護開始
TCFLASH [Memory & Config Group]	TCFLASH0/1/2/3 設定保護 キーレジスタ <TCFCFGn_FCPROTKEY>	保護開始	保護開始	バスエラー	バスエラー *1
WorkFLASH [Memory & Config Group]	WorkFLASH00/01/02/03 設定保護キーレジスタ <WFCFGxx_CPR>	保護開始	保護開始	バスエラー	保護開始*1
ハードウェア ウォッチドッグ [MCU Config Group]	ハードウェアウォッチドッグ プロテクションレジスタ <HWDG_PROT>	保護開始	保護開始*2	バスエラー	バスエラー *1
ソフトウェア ウォッチドッグ [MCU Config Group]	ソフトウェアウォッチドッグ プロテクションレジスタ <SWDGn_PROT>	保護開始	保護開始*2	バスエラー	バスエラー *1

n: ペリフェラルが接続されている Cortex-R5F コアの番号(n=0, 1)

xx: ユニット番号(xx=00, 01, 02, 03)

自コア Tw: 自コアから保護対象レジスタへ書き込み。

自コア Gw: 自コアから保護対象レジスタと同グループのレジスタ*3 へ書き込み。

他コア Tw: 他コアから保護対象レジスタへ書き込み。

他コア Gw: 他コアから保護対象レジスタと同グループのレジスタ*3 へ書き込み。

*1: 同グループの異なるペリフェラルのレジスタへの書き込みは影響なし。

*2: 同ペリフェラルの保護対象外レジスタへの書き込みは影響なし。

*3: 同ペリフェラルの保護対象外レジスタへの書き込みも含む。

4 ペリフェラルの使用につて

4.1 推奨するペリフェラル使用方法

ペリフェラルを使用する際は以下のようにしてください。

- アクセスするチャンネルについて
コアごとにアクセスするチャンネルを分けてください。表 8 に例を記します。
- 例外ハンドラ開始アドレスに設定するアドレスについて
例外ハンドラ開始アドレスの設定は全てのコアで共通の設定となるため、以下の点に注意してください。
 - 設定は片方のコアでのみ行ってください。
 - 設定した内容は全てのコアに対して反映されます。
 - 全てのコアにおいて、例外発生時に遷移しても問題ないアドレスを設定してください。
 図 10、図 11 に設定例を記します。
- CAN の設定について
コアごとに CAN の設定を行う場合、CAN プリスケアラの設定が完了したことを確認してから CAN の設定を行ってください。図 12 に例を記します。
- ソフトウェアリセットの実施タイミングについて
ソフトウェアリセットは以下のシーケンスが実行中でないことを確認してから行ってください。図 12 に例を記します。
 - ウォッチドッグレジスタライト保護シーケンス
 - ウォッチドッグカウンタクリア保護トリガシーケンス
- ソフトウェアリセットビットセット後の処理について
CPU 動作モードに関わらず、ソフトウェアリセットビットをセットした後はウォッチドッグにおける全てのシーケンスを実行しないでください。
ソフトウェアリセットビットセット後、リセットが発行されるまでには時間がかかります。ソフトウェアリセットビットをセットしてからリセットが発行されるまでの時間は不定のため、支障がない限りはリセットが発行されるまで処理を行わず、無限ループでリセット発行を待つようにしてください。図 13 に例を記します。

表 8 ペリフェラル使用例

コア	使用するペリフェラル
CPU0	WorkFLASH#0
	CAN ch0
	32 ビットフリーランタイム ch0
	32 ビットフリーランタイム ch2
	12 ビット A/D コンバータ ch0
	12 ビット A/D コンバータ ch2
CPU1	WorkFLASH#1
	CAN ch1
	32 ビットフリーランタイム ch1
	32 ビットフリーランタイム ch3
	12 ビット A/D コンバータ ch1
	12 ビット A/D コンバータ ch3

図 10 例外ハンドラ開始アドレス設定例 (CPU0)

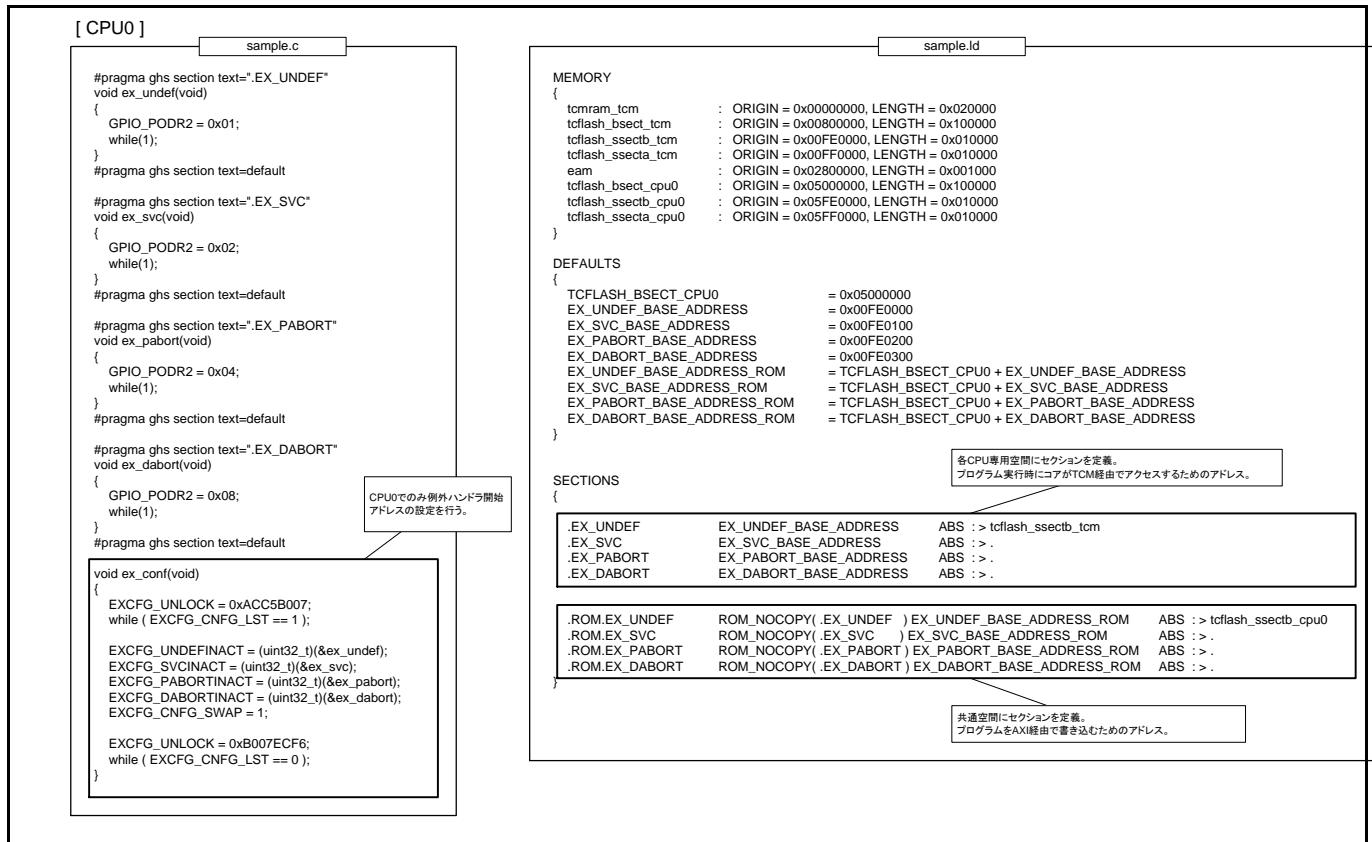


図 11 例外ハンドラ開始アドレス設定例 (CPU1)

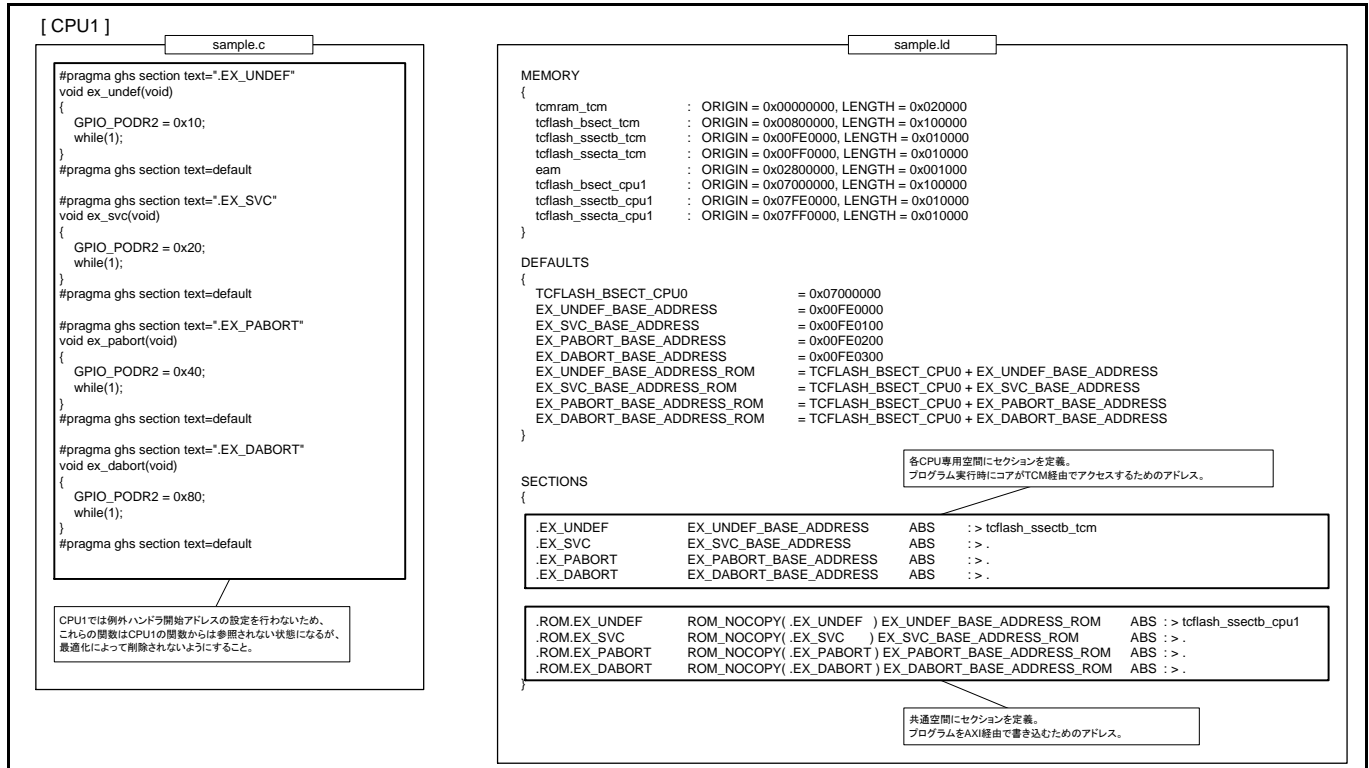


図 12 推奨する CAN 設定例

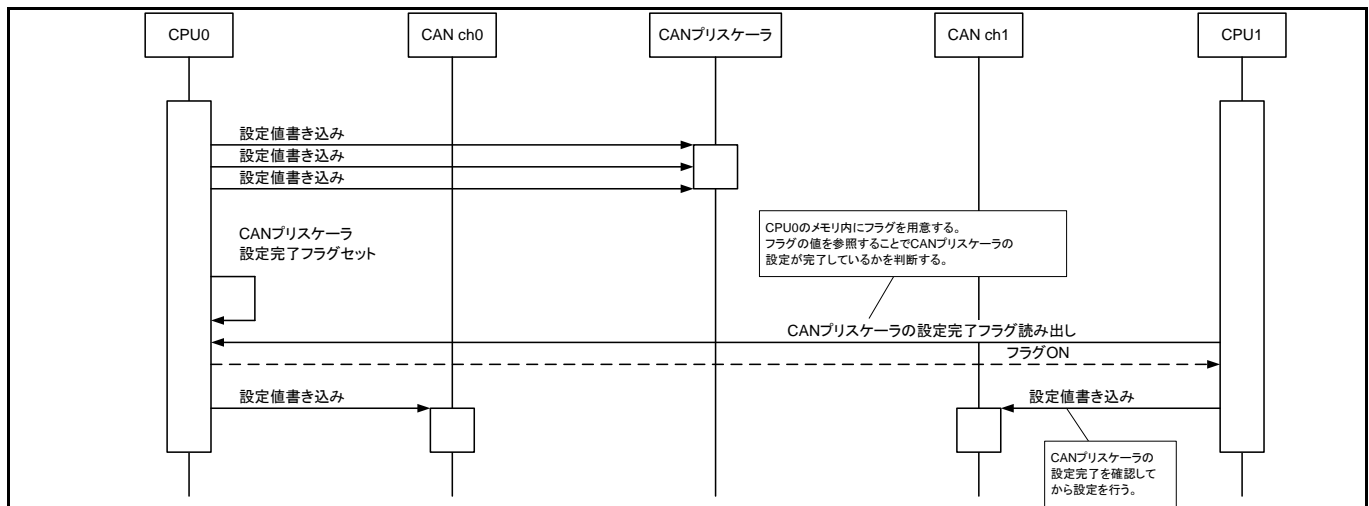
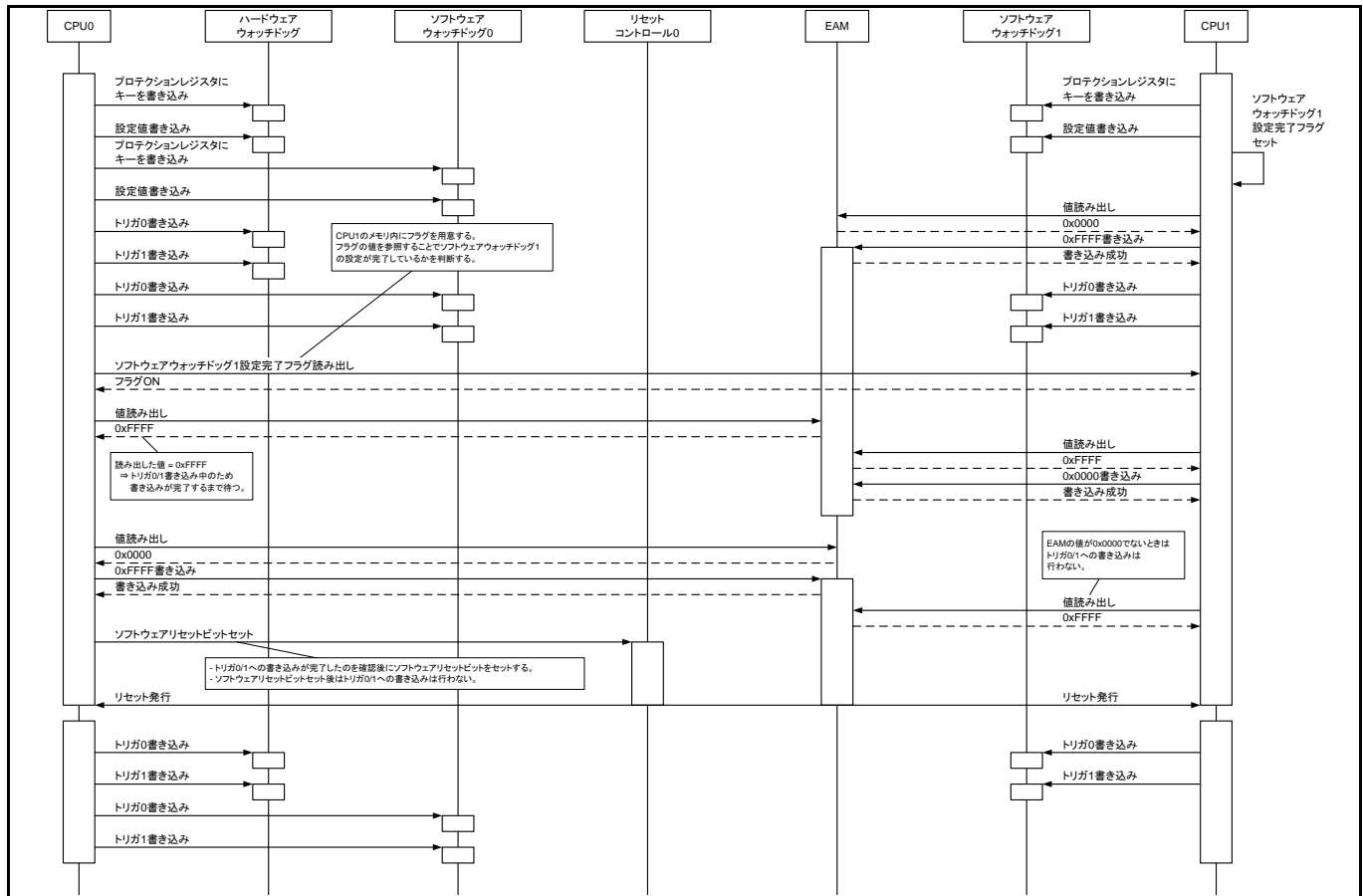


図 13 推奨するソフトウェアリセット発行例



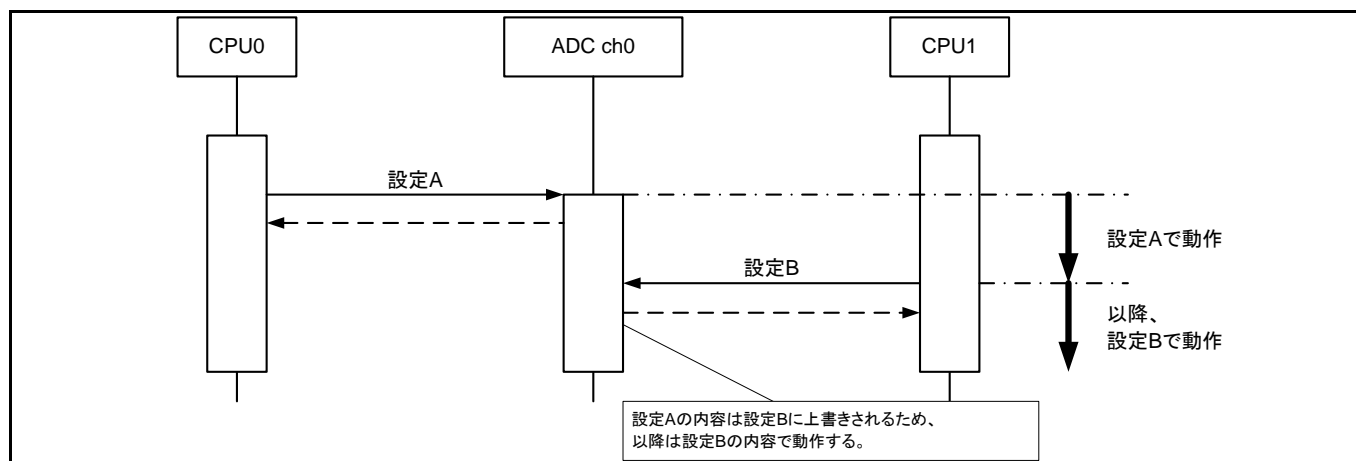
4.2 推奨する方法を行わなかった場合

「4.1 推奨するペリフェラル使用方法」に記載している方法を行わなかった場合に起こりうる問題について以下に記します。

4.2.1 ペリフェラルを設定する場合

複数のコアから同じペリフェラルの同じチャンネルに対して異なる設定を行った場合、後から行われた設定によって上書きされるため、ペリフェラルは最後に設定された内容で動作します。図 14 に例を記します。

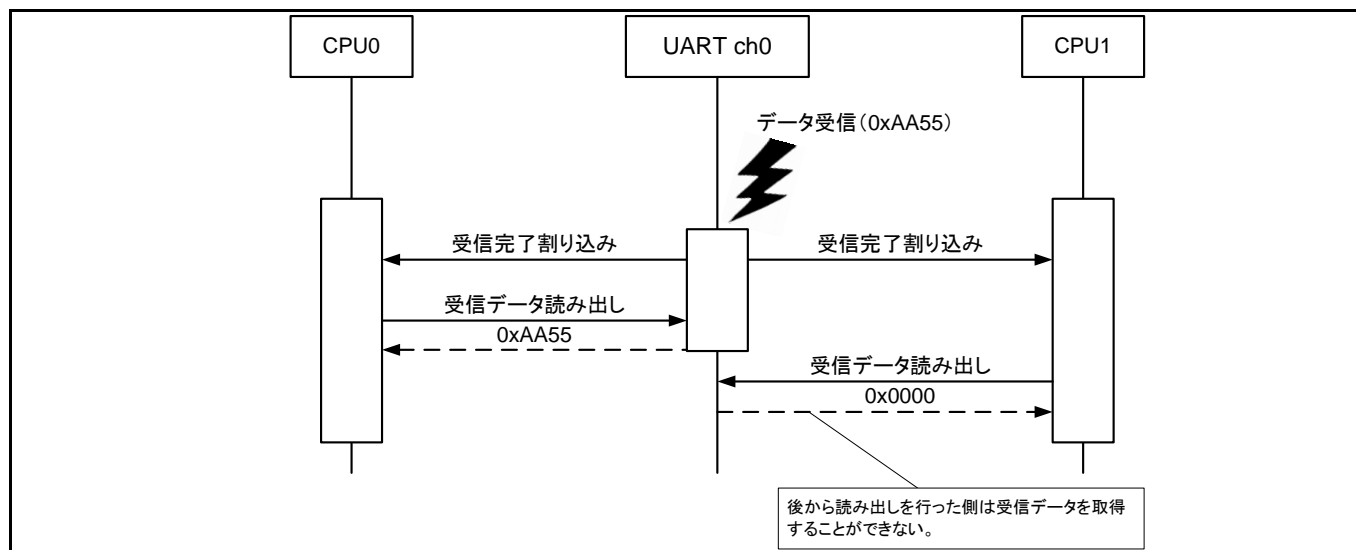
図 14 複数コアからの設定例



4.2.2 ペリフェラルから読み出しを行う場合

複数コアから同じペリフェラルの同じチャンネルに対してデータの読み出しを行った場合、読み出しを行った順番によって読み出される値が異なる場合があります。これにより読み出しが行われるタイミングによって読み出し結果が異なることになります。図 15 に例を記します。

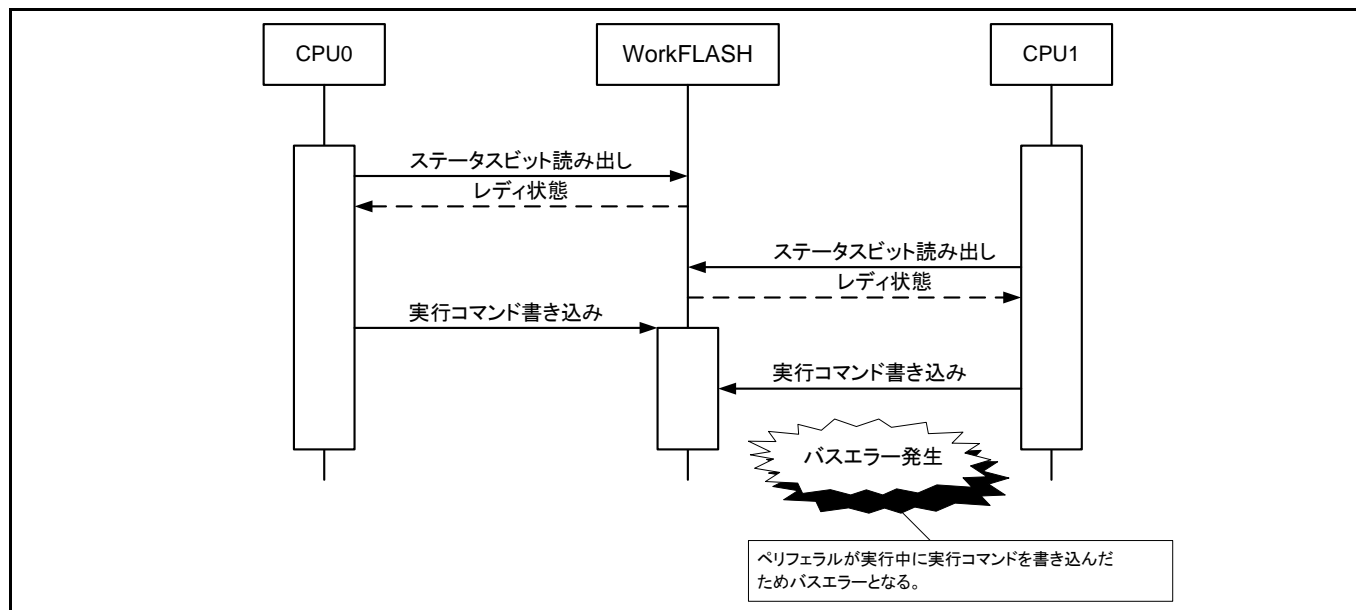
図 15 複数コアからのデータ読み出し例



4.2.3 同じペリフェラルを使用する場合

複数コアから同じペリフェラルの同じチャネルを使用した場合、タイミングによってペリフェラルの状態確認後にペリフェラルが使用できない状態に遷移している可能性があります。図 16 に例を記します。

図 16 他コアにより使用不可能になる例



4.2.4 例外ハンドラ開始アドレスを設定する場合

表 9 のレジスタにて例外ハンドラの開始アドレスを設定することが可能です。ただし、全てのコアで共通の設定となるため、アドレスを設定したコアと例外が発生したコアに関わらず、設定したアドレスにジャンプします。

例) CPU0 で例外ハンドラの開始アドレスを設定したときに、CPU1 で例外が発生した場合の処理の流れ

1. CPU0 にて例外ハンドラの開始アドレスにアドレス"0x00FE0000"を設定
2. 1.の処理完了後、CPU1 にて定常処理実行中に例外発生
3. CPU1 の処理がアドレス"0x00FE0000"にジャンプする (CPU0 では例外が発生していないので、ジャンプせずに定常処理を続行する)

ジャンプ先のアドレスが例外ハンドラの処理の先頭でなかった場合、誤動作を起こす原因となります。

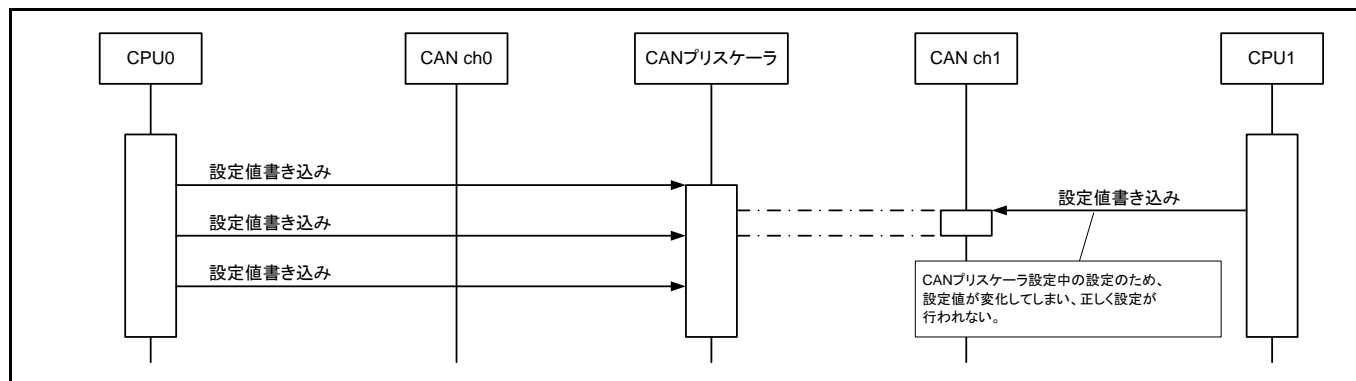
表 9 例外ハンドラ開始アドレス設定レジスタ

レジスタ略称	レジスタ名
EXCFG_UNDEFINACT	EXCFG 非アクティブセット 未定義命令ベクタレジスタ
EXCFG_SVCINACT	EXCFG 非アクティブセット スーパーバイザコールベクタレジスタ
EXCFG_PABORTINACT	EXCFG 非アクティブセット プリフェッチアポートベクタレジスタ
EXCFG_DABORTINACT	EXCFG 非アクティブセット データアポートベクタレジスタ
EXCFG_UNDEFACT	EXCFG アクティブセット 未定義命令ベクタレジスタ
EXCFG_SVCACT	EXCFG アクティブセット スーパーバイザコールベクタレジスタ
EXCFG_PABORTACT	EXCFG アクティブセット プリフェッチアポートベクタレジスタ
EXCFG_DABORTACT	EXCFG アクティブセット データアポートベクタレジスタ

4.2.5 CAN を設定する場合

ハードウェアの構成上、CAN プリスケーラの設定中は CAN の設定を行うことができません。CAN プリスケーラの設定中に CAN の設定を行うと設定値が意図しない値に変化してしまいます。図 17 に例を記します。

図 17 CAN プリスケーラ設定中に CAN 設定実施例



4.2.6 ソフトウェアリセットを行う場合

ハードウェアウォッチドッグ、ソフトウェアウォッチドッグともにソフトリセット（ソフトウェアリセット）ではプロテクションレジスタおよび、トリガ 0/1 レジスタはクリアされずリセット後もリセット時のシーケンスが維持されます。そのため表 10 にあるシーケンス実行中にソフトウェアリセットが発行されると、リセット前後でシーケンス違反が起こります。

図 18、図 19 にウォッチドッグレジスタライト保護シーケンス実行中にソフトウェアリセットを行った例を記します。

図 20、図 21 にウォッチドッグカウンタクリア保護トリガシーケンス実行中にソフトウェアリセットを行った例を記します。

図 22、図 23 にソフトウェアリセットビットセット後にウォッチドッグカウンタクリア保護トリガシーケンスを実行した例を記します。

表 10 ウォッチドッグに関するシーケンス

シーケンス	用途
ウォッチドッグレジスタライト保護シーケンス	ウォッチドッグの設定を行うレジスタへの書き込み保護解除
ウォッチドッグカウンタクリア保護トリガシーケンス	ウォッチドッグカウンタのクリアを行う

図 18 ハードウェアウォッチドッグレジスタライト保護シーケンス違反例

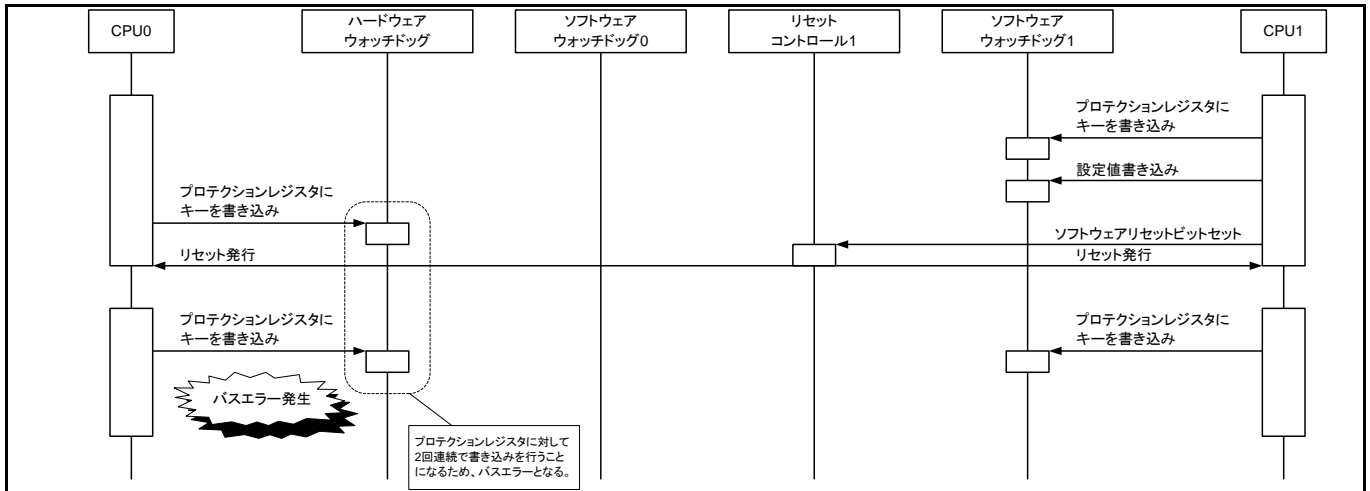


図 19 ソフトウェアウォッチドッグレジスタライト保護シーケンス違反例

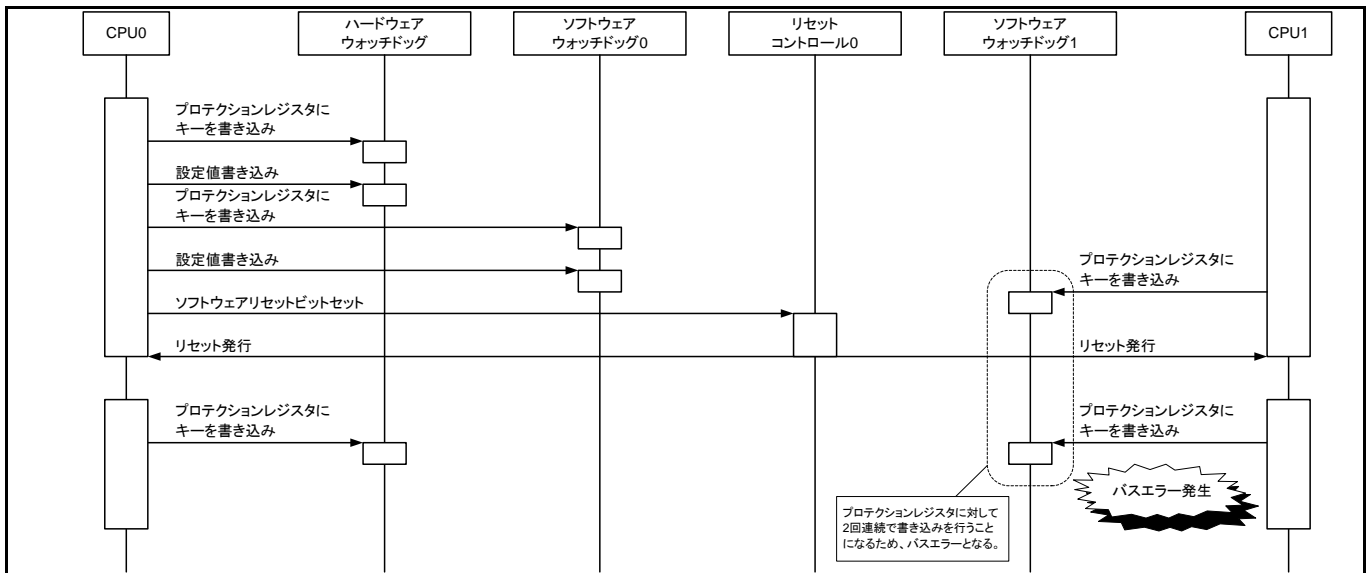


図 20 ハードウェアウォッチドッグカウンタクリア保護トリガシーケンス違反例

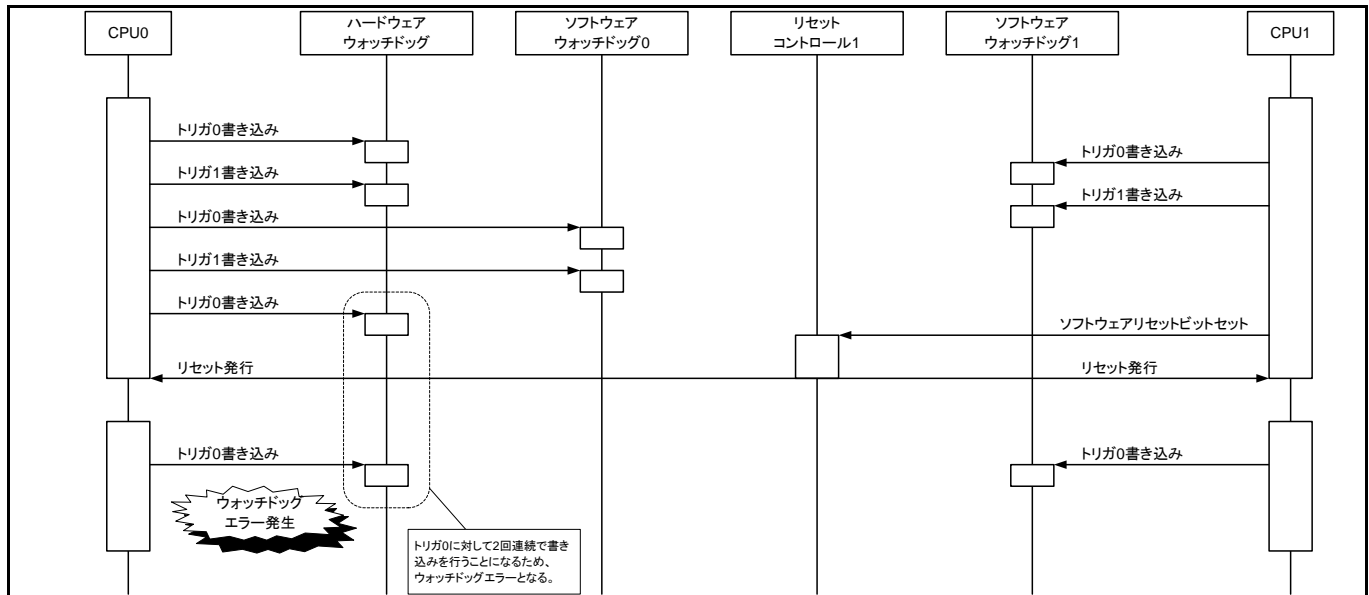
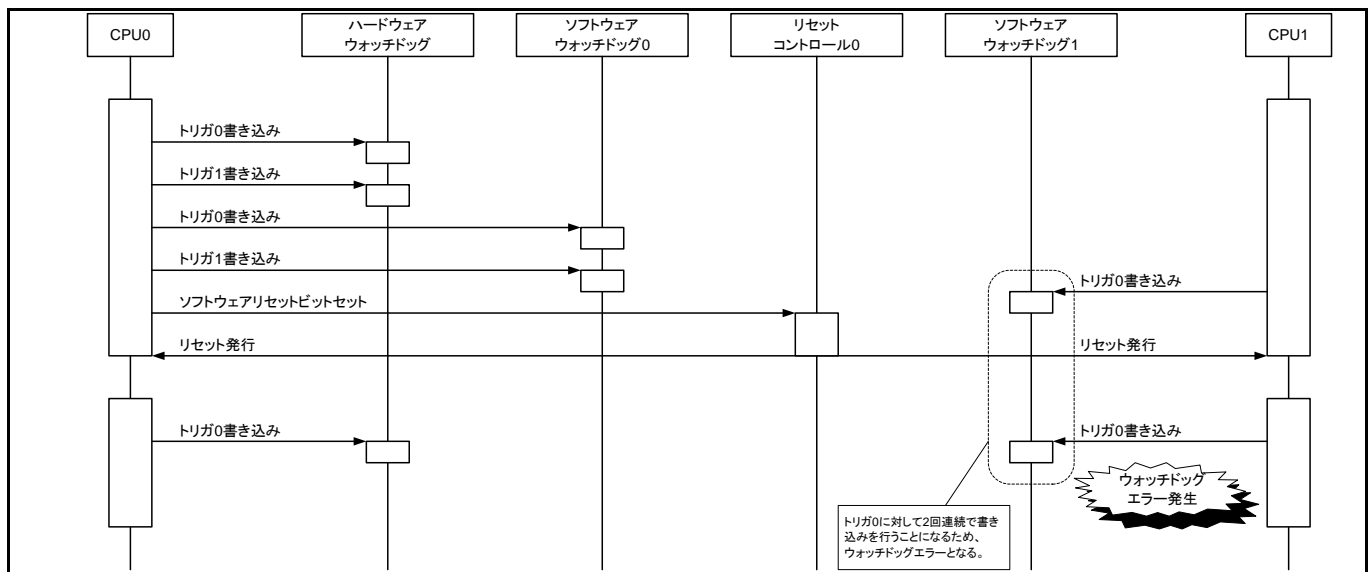


図 21 ソフトウェアウォッチドッグカウンタクリア保護トリガシーケンス違反例



```

sequenceDiagram
    participant CPU0
    participant HW_WD as ハードウェア ウォッチドッグ
    participant SW_WD0 as ソフトウェア ウォッチドッグ0
    participant Reset_Ctrl1 as リセット コントロール1
    participant SW_WD1 as ソフトウェア ウォッチドッグ1
    participant CPU1

    CPU0->>HW_WD: トリガ0書き込み
    HW_WD->>CPU0: リセット発行
    CPU0->>HW_WD: トリガ0書き込み
    HW_WD->>SW_WD0: トリガ0書き込み
    SW_WD0->>Reset_Ctrl1: リセットビットセット
    Reset_Ctrl1->>CPU1: リセット発行
    CPU1->>SW_WD1: トリガ0書き込み
    SW_WD1->>CPU1: リセット発行
    
```

The diagram illustrates the reset process between CPU0 and CPU1. CPU0 triggers a reset of the hardware watchdog, which then triggers a reset of the software watchdog. CPU1 also triggers a reset of the software watchdog. The diagram shows the flow of reset signals and the resulting watchdog errors.

トリガ0に対して2回連続で書き込みを行うことになるため、ウォッチドッグエラーとなる。

[1]. 32 ビット・マイクロコントローラ MB9D560 シリーズハードウェアマニュアル
[2]. 32-bit Microcontroller MB9D560 シリーズ DATA SHEET

6.1 コア間で同期をとる方法

1. メモリまたはレジスタをフラグとして使用する (図 24)
2. EAM をセマフォとして使用する (図 25)
3. コア間通信を使用する (図 26)

図 24 フラグを使用した同期例

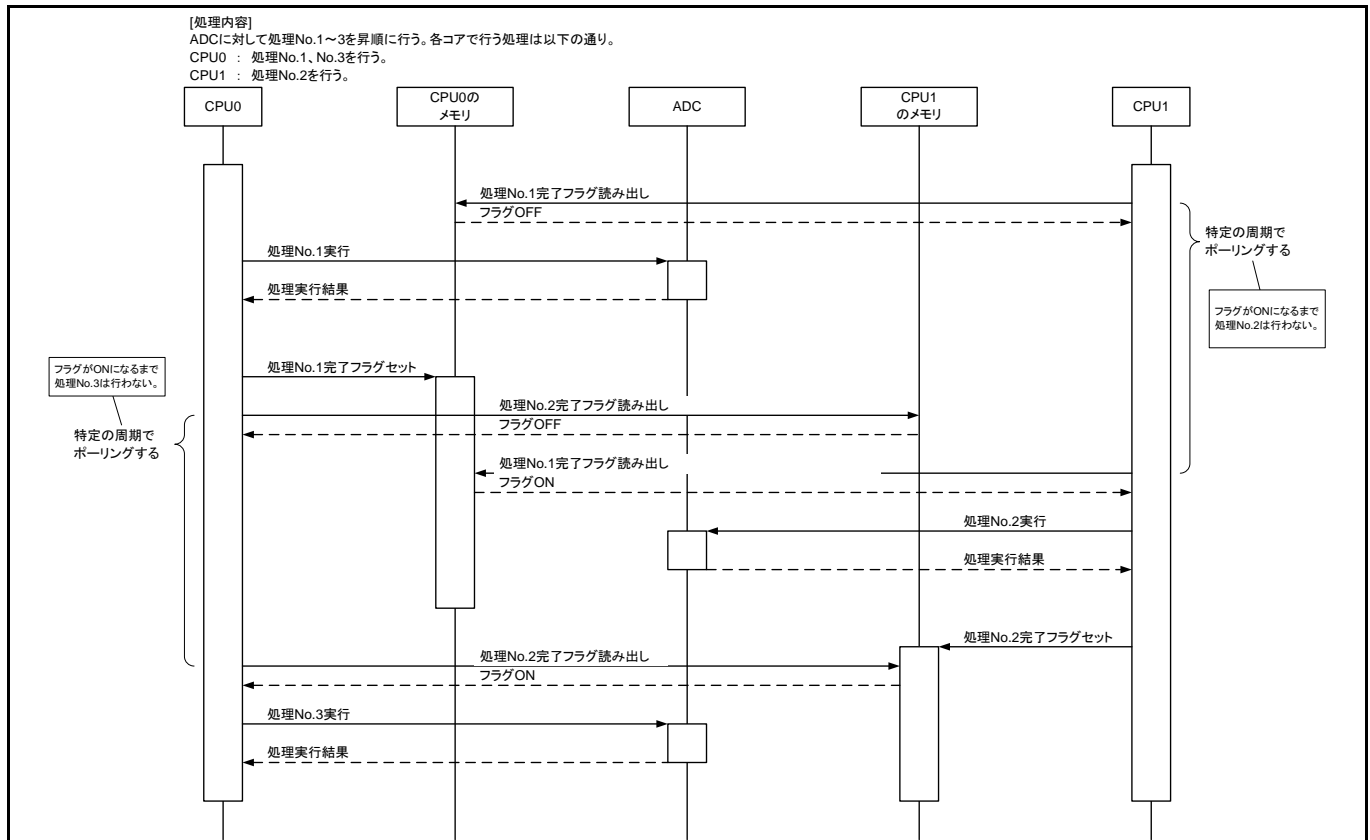


図 25 EAM をセマフォとして使用した同期例

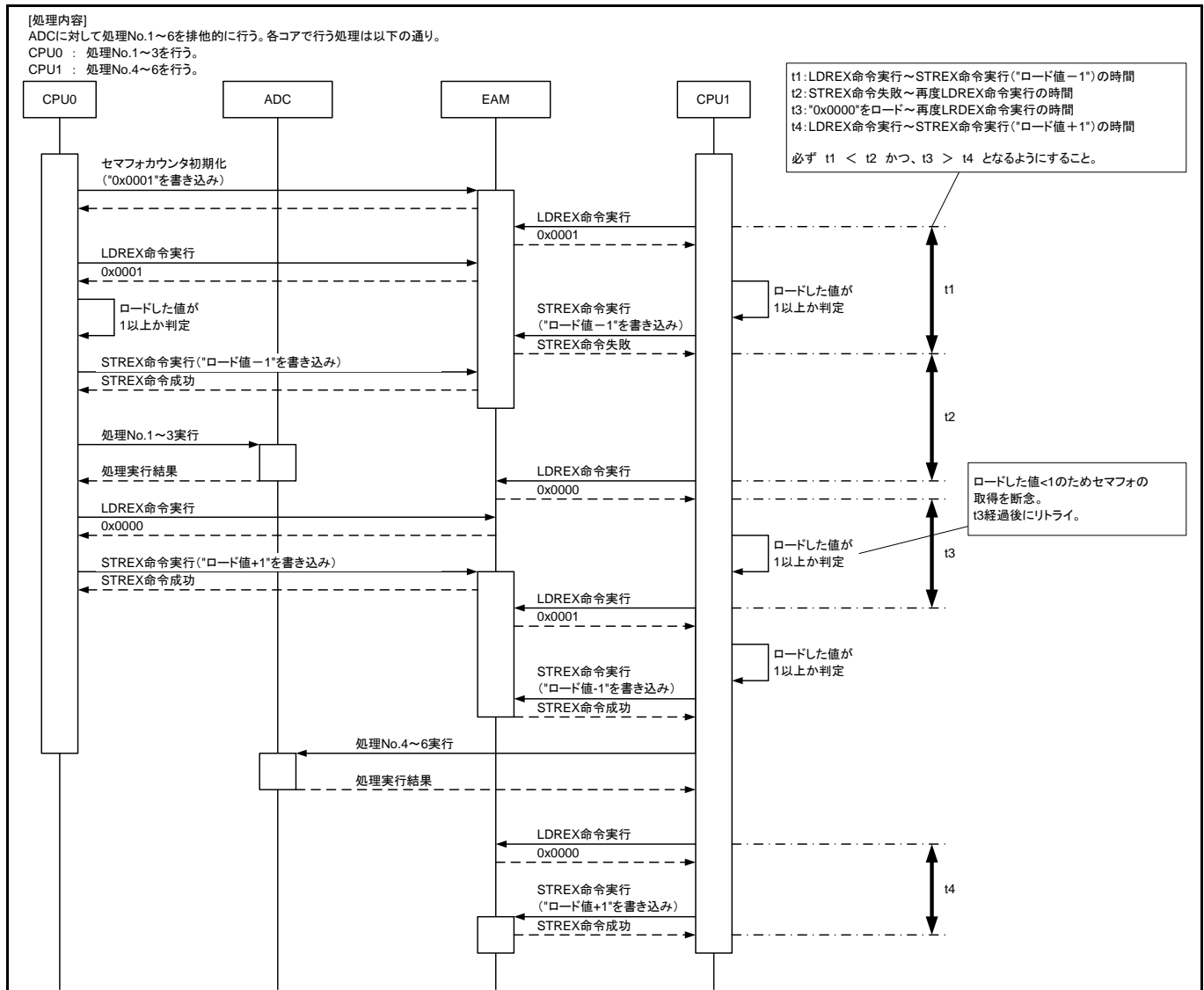
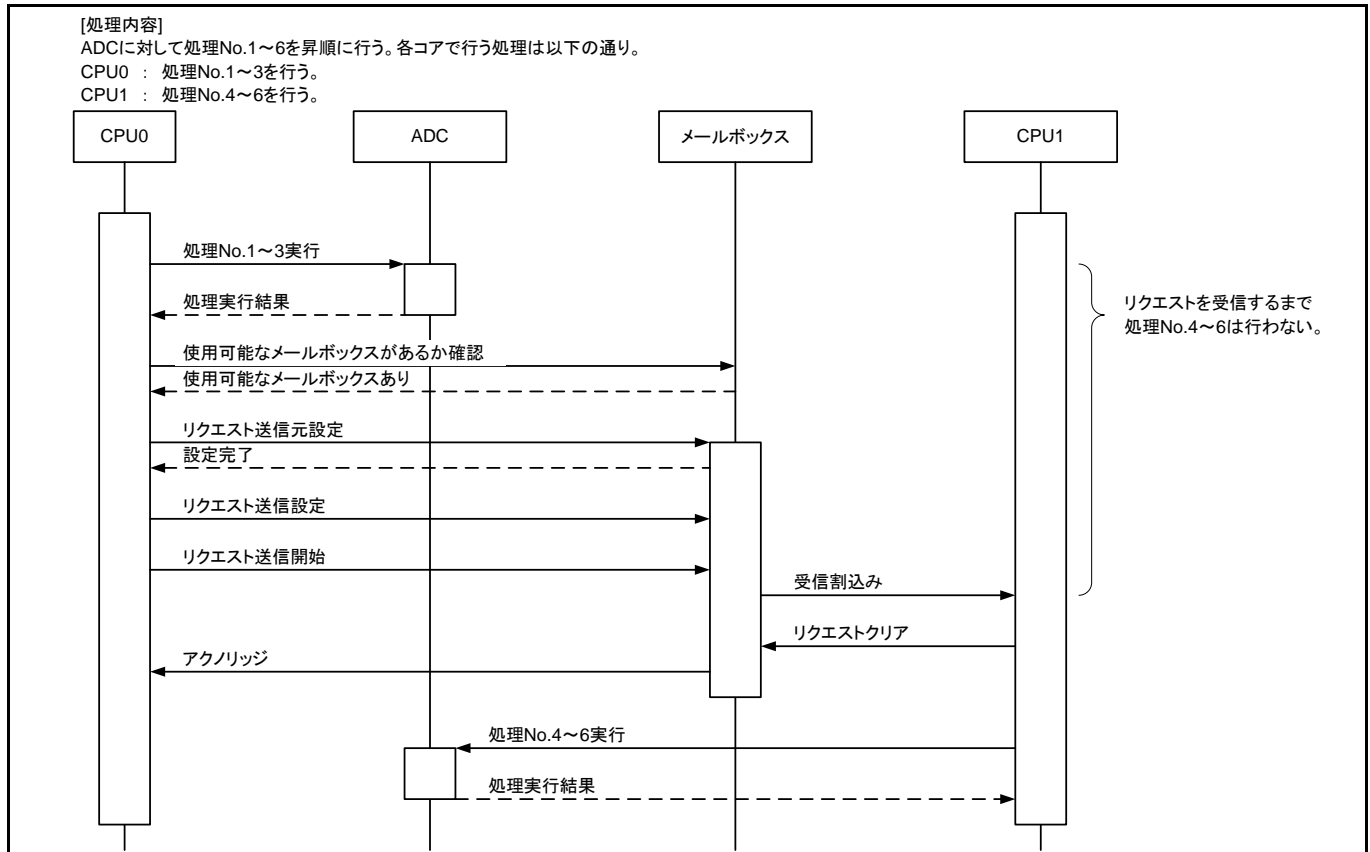


図 26 コア間通信を使用した同期例



7 改訂履歴

文書名: AN204448 - 2 CPU モード時の注意 Traveo™ Family MB9D560 シリーズ

文書番号: 002-04449

版	ECN 番号	変更者	発行日	変更内容
**	-	KHAS	07/16/2015	スパンションアプリケーションノート MB9D560_AN708-00007-1v0-J をサイプレスとして登録したものです。
*A	5587660	KHAS	01/16/2015	これは英語版の 002-04448 Rev. *A を翻訳した日本語版です。

Worldwide Sales and Design Support

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	cypress.com/arm
車載用	cypress.com/automotive
クロック&パッファ	cypress.com/clocks
インターフェース	cypress.com/interface
IoT (モノのインターネット)	cypress.com/iot
メモリ	cypress.com/memory
マイクロコントローラ	cypress.com/mcu
PSoC	cypress.com/psoc
電源用 IC	cypress.com/pmic
タッチ センシング	cypress.com/touch
USB コントローラー	cypress.com/usb
ワイヤレス/RF	cypress.com/wireless

PSoC® ソリューション

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP

サイプレス開発者コミュニティ

[フォーラム](#) | [WICED IOT Forums](#) | [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

All other trademarks or registered trademarks referenced herein are the property of their respective owners.

 <p>CYPRESS Embedded in Tomorrow™</p>	Cypress Semiconductor		Phone : 408-943-2600
	198 Champion Court		Fax : 408-943-4730
	San Jose, CA 95134-1709		Website : www.cypress.com

© Cypress Semiconductor Corporation, 2015-2017. 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又はファームウェア（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、また、本段落で特に記載されているものを除き、Cypress 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ていない。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、あなたが Cypress との間で別途本ソフトウェアの使用法を定める書面による合意をしていない場合、Cypress は、あなたに対して、（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、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）エンドユーザーに対して、バイナリーコード形式で本ソフトウェアを外部に配布すること、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）に抵触する 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身専属的ライセンス（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に関しても、明示又は黙示をとわず、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。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又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し、プログラムし、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用いるため、又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こととなるその他の使用（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、装置又はシステムのその構成部分の不具合が、その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、機器又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とわず一切の責任を負わず、かつ、あなたは Cypress をそれら一切から免除するものとし、本書により免除する。あなたは、Cypress 製品の本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 Cypress を免責補償する。

Cypress、Cypress のロゴ、Spansion、Spansion のロゴ及びこれらの組み合わせ、WICED、PSoC、CapsSense、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 の商標のより完全なリストは、cypress.com を参照のこと。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。