

RZ/A2Mグループ

HyperBus™ コントローラを使用したHyperFlash™ からのブート例

要旨

本アプリケーションノートは、RZ/A2Mの HyperBus™ コントローラを使用して、ブートモード7 (HyperFlash™ ブート2 1.8V 品) によって HyperFlash からブートを行う例について説明します。

動作確認デバイス

RZ/A2M

本アプリケーションノートを他のマイコンへ適用する場合、そのマイコンの仕様にあわせて変更し、十分評価してください。

【注】 HyperBus™/HyperFlash™/HyperRAM™ は Cypress Semiconductor Corporation.の商標です。

目次

1. 仕様	4
1.1 HyperFlashからのブート	4
1.2 使用する周辺機能	6
2. 動作確認条件	8
3. 関連アプリケーションノート	9
4. ハードウェア説明	10
4.1 ハードウェア構成例	10
4.2 仕様端子一覧	11
5. ソフトウェア説明	12
5.1 動作概要	12
5.1.1 HyperFlashブートに関する用語	12
5.1.2 サンプルコード全体の動作概要	13
5.1.3 ローダプログラムの動作概要	14
5.1.4 アプリケーションプログラム	16
5.2 サンプルコード実行時の周辺機能の設定およびメモリ配置	19
5.2.1 周辺機能の設定	19
5.2.2 メモリマップ	20
5.2.3 アプリケーションプログラムで使用する仮想アドレス空間	21
5.2.4 サンプルコードのセクション配置	23
5.3 使用割り込み一覧	26
5.4 データ型一覧	26
5.5 定数一覧	27
5.6 構造体/共用体一覧	30
5.7 変数一覧	34
5.8 関数一覧	35
5.9 関数仕様	37
5.10 ローダプログラムのフローチャート	44
5.10.1 ローダプログラム全体	44
5.10.2 R_SC_HardwareSetup関数	45
5.10.3 HyperBusコントローラの初期設定	46
5.10.4 HyperFlashの設定	47
5.11 アプリケーションプログラムのフローチャート	48
5.11.1 アプリケーションプログラム全体	48
5.11.2 R_SC_HardwareSetup関数	48
5.11.3 HyperBusコントローラの初期設定	49
5.11.4 HyperRAMの初期設定	50
5.12 HyperFlashおよびHyperRAMをアクセスするための関数のフローチャート	51
5.12.1 HyperFlashのVolatile Configuration Registerのリード関数	51
5.12.2 HyperFlashのVolatile Configuration Register (VCR)へのライト関数	52
5.12.3 HyperRAMのConfiguration Register 0 (CR0)のリード関数	53
5.12.4 HyperRAMのConfiguration Register 0 (CR0)へのライト関数	54

5.12.5 HyperRAMをアクセスする際の領域（メモリまたはレジスタ）の指定関数.....	54
6. 応用例.....	55
6.1 HyperBusコントローラの設定	55
6.2 HyperFlashのレイテンシクロックの設定	56
6.3 HyperRAMのレイテンシクロックの設定	56
7. サンプルコードの注意事項.....	57
7.1 HyperFlashおよびHyperRAMのレジスタアクセス	57
8. サンプルコード	58
9. 参考ドキュメント	58
改訂記録	59

1. 仕様

1.1 HyperFlash からのブート

RZ/A2Mは、ブートモード7の場合、HyperFlash 空間に配置された HyperFlash からブートします。図1.1 にブートモード7の動作イメージを示します。

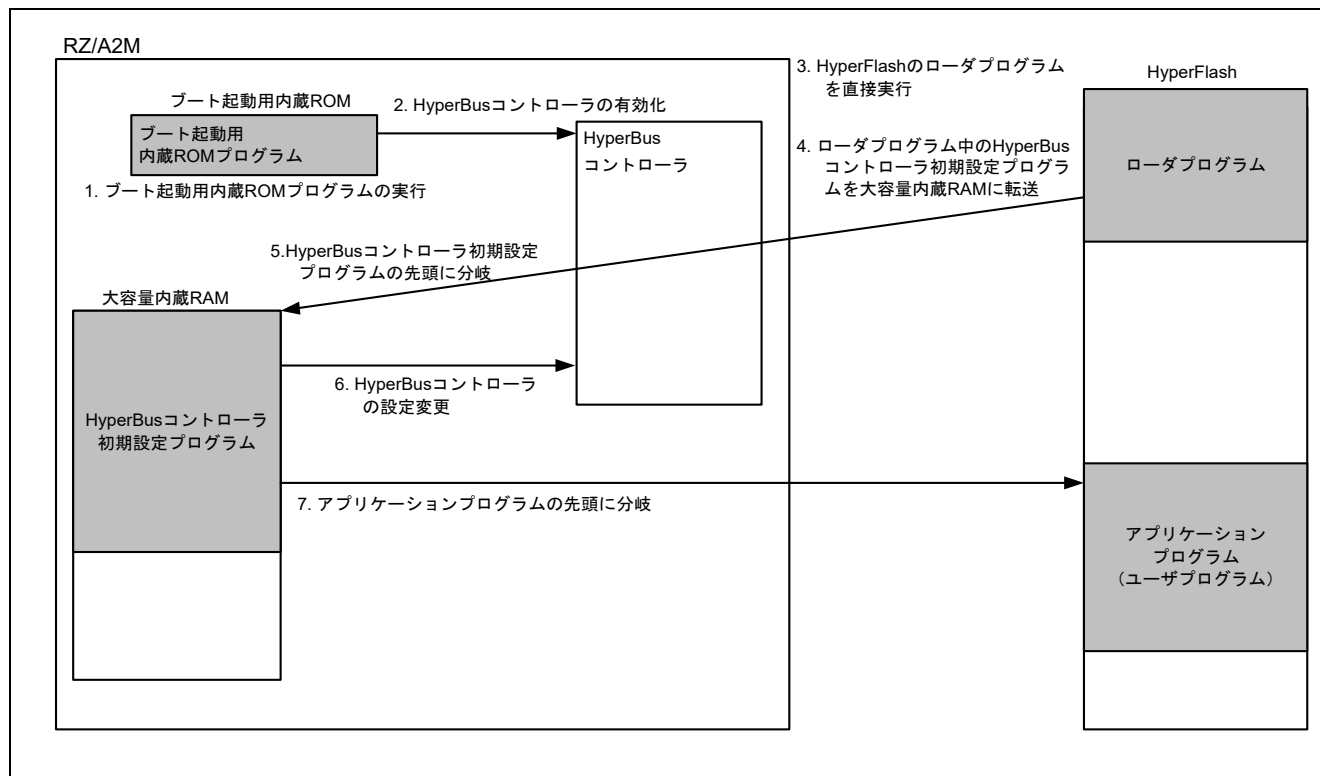


図1.1 ブートモード7の動作イメージ

ブートモード7の動作イメージについて説明します。

- 1 RZ/A2Mは、ブートモード7で起動した場合、パワーオンリセット解除後にブート起動用内蔵 ROM プログラムを実行します。
- 2 ブート起動用内蔵 ROM プログラムは、CPG レジスタ設定、HyperBus 関連端子を使用するように選択し、HyperFlash 空間に配置されたプログラムを直接実行できる状態にします。
- 3 HyperFlash のローダプログラムを実行します。
- 4 HyperFlash から HyperBus コントローラおよび HyperFlash 初期設定プログラムを大容量内蔵 RAM に転送します。
- 5 大容量内蔵 RAM に転送した HyperBus コントローラおよび HyperFlash 初期設定プログラムの先頭アドレスに分岐します。
- 6 HyperBus コントローラ初期設定プログラムにより、HyperBus コントローラおよび HyperFlash の設定を変更します。
- 7 アプリケーションプログラムの先頭アドレスに分岐します。

ブート起動用内蔵 ROM プログラムの実行直後は、RZ/A2M は HyperFlash に対して低速なアクセスを行う設定となっており、ユーザプログラムにて HyperFlash に対して高速にアクセスできる設定を行う必要があります。本アプリケーションノートでは、ブート起動用内蔵 ROM プログラムより分岐する HyperFlash 空間の先頭番地（H'3000_0000）に、お客様が使用する HyperFlash に最適な設定処理を行うローダプログラムを配置し、ローダプログラムにて HyperFlash に対して高速なアクセスができるように設定処理を行った後、お客様が作成するアプリケーションプログラム（ユーザプログラム）に分岐する方法を説明します。

1.2 使用する周辺機能

本サンプルコードでは、HyperBus コントローラの設定とともに、クロックパルス発振器、割り込みコントローラ、汎用入出力ポート、メモリ管理ユニット、1次キャッシュ（L1 キャッシュ）、および2次キャッシュ（L2 キャッシュ）の初期設定を行います。

本アプリケーションノートでは、クロックパルス発振器を CPG、割り込みコントローラを INTC、OS タイマを OSTM、FIFO 内蔵シリアルコミュニケーションインタフェースを SCIFA、汎用入出力ポートを GPIO、低消費電力モードを STB、メモリ管理ユニットを MMU とします。

表1.1に使用する周辺機能と用途を、図1.2にサンプルコード実行時の動作環境を示します。

表1.1 使用する周辺機能と用途

周辺機能	用途
HyperBus コントローラ	メモリマップドリードモードで、CPU が HyperFlash 空間に接続された HyperFlash から、直接リードするための信号を生成。
クロックパルス発振器（CPG）	RZ/A2Mの動作周波数の生成
割り込みコントローラ（INTC）	OSTM チャンネル 0 および OSTM チャンネル 2、SCIFA チャンネル 4 の割り込み制御に使用
OS タイマ（OSTM）	OSTM チャンネル 0 およびチャンネル 2 を使用 <ul style="list-style-type: none">OSTM チャンネル 0 LED 点灯および消灯の周期を生成OSTM チャンネル 2 OS Abstraction Layer による時間管理に使用
FIFO 内蔵シリアルコミュニケーションインタフェース（SCIFA）	SCIFA チャンネル 4 を用いて、ホスト PC との通信用として使用
汎用入出力ポート（GPIO）	SCIFA チャンネル 4 の兼用端子の切り替えに使用、LED の点灯および消灯のための端子制御に使用
低消費電力モード（STB）	RZ/A2Mの周辺 IO のモジュールスタンバイを解除するために使用、保持用内蔵 RAM をライト許可するために使用
メモリ管理ユニット（MMU）、 L1 キャッシュ、L2 キャッシュ	RZ/A2Mの外部アドレス空間において、L1 キャッシュの有効領域の指定やメモリタイプの指定などの変換テーブルを生成。L1 キャッシュおよび L2 キャッシュを有効に設定

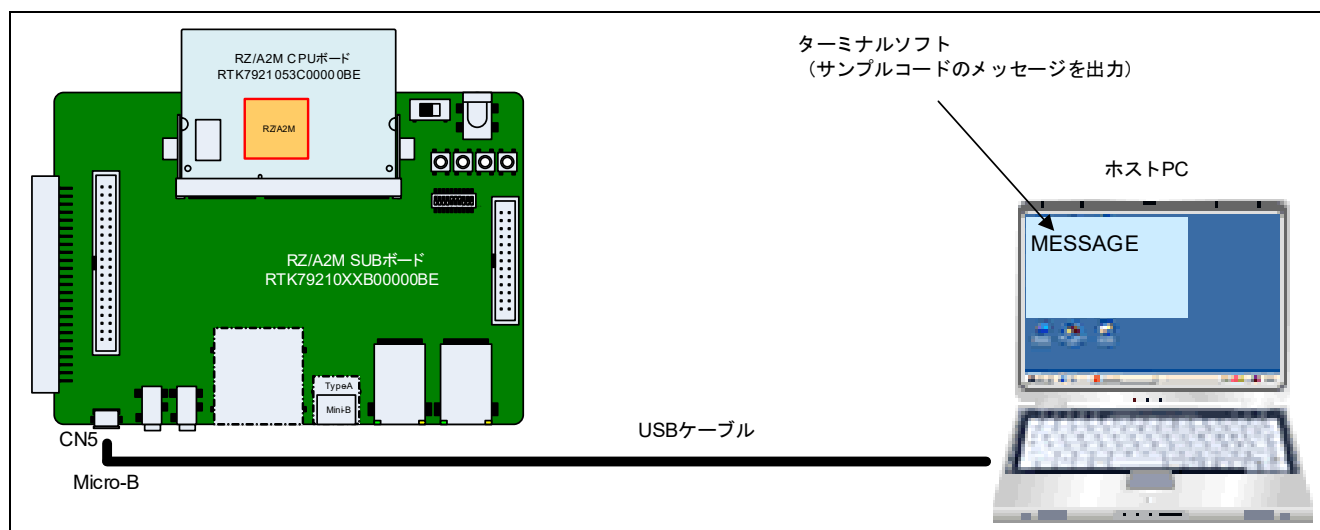


図1.2 動作環境

2. 動作確認条件

本アプリケーションノートのサンプルコードは、下記の条件で動作を確認しています。

表2.1 動作確認条件(1/2)

項目	内容
使用 MCU	RZ/A2M
動作周波数（注）	CPU クロック（I ϕ ）：528MHz 画像処理クロック（G ϕ ）：264MHz 内部バスクロック（B ϕ ）：132MHz 周辺クロック 1（P1 ϕ ）：66MHz 周辺クロック 0（P0 ϕ ）：33MHz HM_CK/HM_CK#：132MHz CKIO：132MHz
動作電圧	電源電圧（I/O）：3.3V 電源電圧（PVcc_HO）：1.8V 電源電圧（内部）：1.2V
統合開発環境	e2 studio V7.6.0
C コンパイラ	GNU Arm Embedded Toolchain 6-2017-q2-update コンパイラオプション（ディレクトリパスの追加は除く） Release コンフィグレーション： <ul style="list-style-type: none"> -mcpu=cortex-a9 -march=armv7-a -marm -mlittle-endian -mfloat-abi=hard -mfpu=neon -mno-unaligned-access -Os -ffunction-sections -fdata-sections -Wunused -Wuninitialized -Wall -Wextra -Wmissing-declarations -Wconversion -Wpointer-arith -Wpadded -Wshadow -Wlogical-op -Waggregate-return -Wfloat-equal -Wnull-dereference -Wmaybe-uninitialized -Wstack-usage=100 -fabi-version=0 Hardware Debug コンフィグレーション： <ul style="list-style-type: none"> -mcpu=cortex-a9 -march=armv7-a -marm -mlittle-endian -mfloat-abi=hard -mfpu=neon -mno-unaligned-access -Og -ffunction-sections -fdata-sections -Wunused -Wuninitialized -Wall -Wextra -Wmissing-declarations -Wconversion -Wpointer-arith -Wpadded -Wshadow -Wlogical-op -Waggregate-return -Wfloat-equal -Wnull-dereference -Wmaybe-uninitialized -g3 -Wstack-usage=100 -fabi-version=0

【注】 クロックモード 1（EXTAL 端子からの 24MHz のクロック入力）で使用時の動作周波数です。

表2.2 動作確認条件(2/2)

項目	内容
動作モード	ブートモード 7 (HyperFlash ブート 2 1.8V 品)
ターミナルソフトの通信設定	<ul style="list-style-type: none">通信速度 : 115200bpsデータ長 : 8 ビットパリティ : なしストップビット長 : 1 ビットフロー制御 : なし
使用ボード	RZ/A2M CPUボード RTK7921053C00000BE RZ/A2M SUBボード RTK79210XXB00000BE
使用デバイス (ボード上で使用する機能)	<ul style="list-style-type: none">HyperBus MCP (HyperFlash:64M バイト、HyperRAM:8M バイト) メーカー名 : Cypress社、型名 : S71KS512SC0BHV000RL78/G1C (USB 通信とシリアル通信を変換し、ホスト PC との通信に使用)LED1

3. 関連アプリケーションノート

本アプリケーションノートに関連するアプリケーションノートを以下に示します。併せて参照してください。

- RZ/A2M グループ 初期設定例 (R01AN4321)

4. ハードウェア説明

4.1 ハードウェア構成例

本アプリケーションノートで紹介する HyperFlash ブート例は、ブートモード 7 を使用して、HyperFlash 空間に接続された HyperFlash に格納されたプログラムにより処理を行います。図4.1にブートモード7で HyperFlashからブートする場合の接続例を示します。

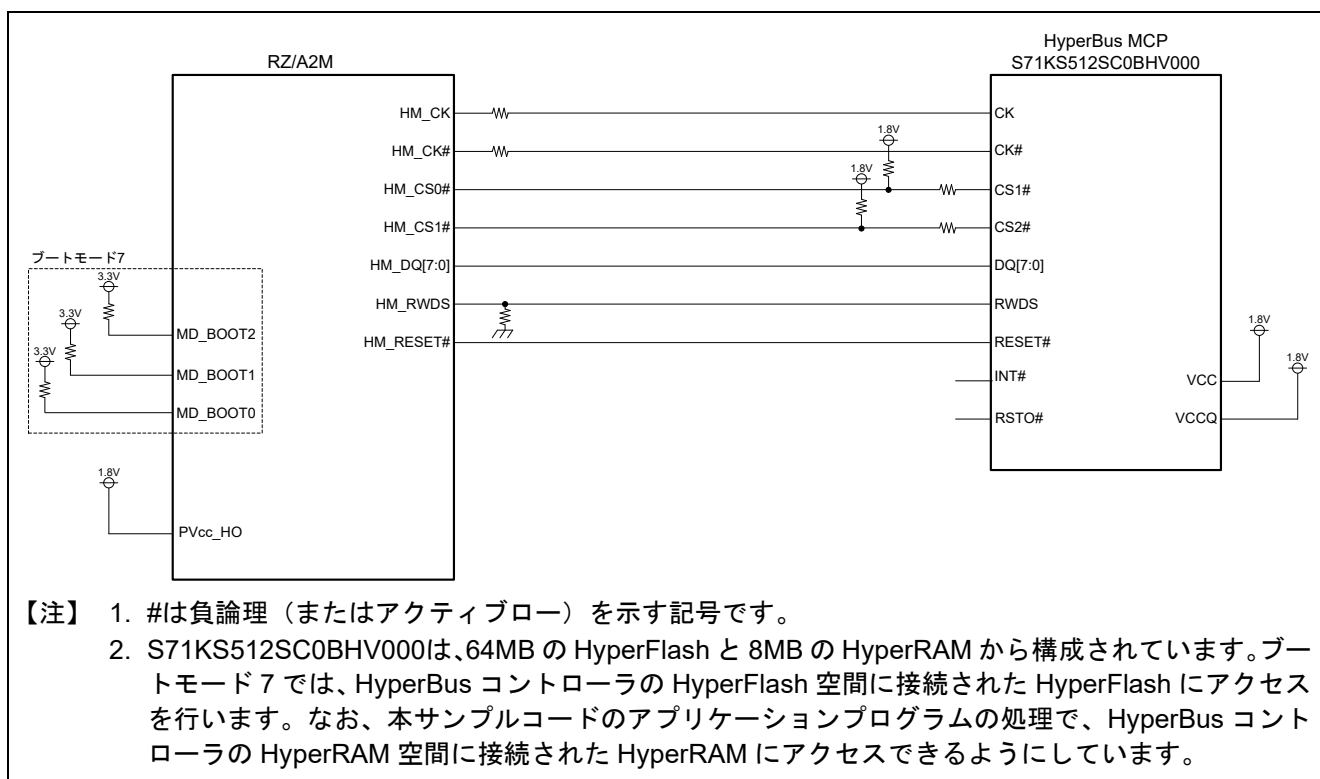


図4.1 ブートモード7でHyperFlashからブートする場合の接続例

4.2 仕様端子一覧

表4.1に仕様端子と機能を示します。

表4.1 仕様端子と機能

端子名	入出力	内容
MD_BOOT2	入力	ブートモードの選択（ブートモード7に設定） MD_BOOT2:"H"、MD_BOOT1:"H"、MD_BOOT0:"H"
MD_BOOT1	入力	
MD_BOOT0	入力	
HM_CK	出力	HyperBus コントローラの差動クロック
HM_CK#	出力	HyperBus コントローラの差動クロック
HM_CS0#	出力	HyperBus コントローラのチップセレクト0（HyperFlash 用）
HM_CS1#	出力	HyperBus コントローラのチップセレクト1（HyperRAM 用）
HM_DQ[7:0]	入出力	HyperBus コントローラのデータ
HM_RWDS	入出力	HyperBus コントローラのリードライトデータマスク
HM_RESET#	出力	HyperBus コントローラからのリセット
P6_0	出力	LED の点灯および消灯
RxD4(P9_1)	入力	シリアル受信データ信号
TxD4(P9_0)	出力	シリアル送信データ信号

【注】 #は負論理（またはアクティブロー）を示す信号です。

5. ソフトウェア説明

5.1 動作概要

ここでは、本アプリケーションノートのサンプルコードの動作概要について説明します。

5.1.1 HyperFlash ブートに関する用語

表5.1に本アプリケーションノートで説明するHyperFlashブート動作に関する用語を示します。

表5.1 HyperFlash ブート動作に関する用語

用語	説明
ブート起動用内蔵 ROM プログラム	ブート起動用内蔵 ROM プログラムは、ブートモード 7 (HyperFlash ブート 2 1.8V 品) で起動した場合に、HyperFlash 空間に接続された HyperFlash に格納されているプログラムを直接実行するための設定を行うプログラムです。 RZ/A2Mはブート起動用内蔵 ROM プログラムの実行完了後、HyperFlash 空間の先頭アドレスである H'3000_0000 番地に分岐します。RZ/A2Mの内蔵 ROM に格納されているプログラムのため、お客様が作成する必要はありません。
ローダプログラム	ローダプログラムは、ブート起動用内蔵 ROM プログラムの処理完了後に実行するプログラムです。 ローダプログラムは、お客様が使用する HyperFlash に合わせて、HyperBus コントローラおよび HyperFlash のレジスタ設定処理を行い、アプリケーションプログラムの先頭アドレスへ分岐する処理を行います。 ローダプログラムは、本アプリケーションノートを参考に、使用する HyperFlash の仕様に合わせてお客様が作成してください。なお、サンプルコードでは、Cypress社製 HyperBus MCP (S71KS512SC0BHV000) を使用する場合に最適な設定を行っています。
アプリケーションプログラム (ユーザプログラム)	アプリケーションプログラムは、お客様がシステムに合わせて作成するプログラムです。

5.1.2 サンプルコード全体の動作概要

サンプルコードは、ブート起動用内蔵 ROM プログラムの処理完了後に実行するローダプログラムとアプリケーションプログラムで構成されています。

1 ローダプログラム（プロジェクト名：rza2m_hyperflash_boot_loader_gcc）

ローダプログラムは、使用する HyperFlash（Cypress社製 HyperBus MCP（S71KS512SC0BHV000））に最適な設定を行います。ローダプログラムはブート起動用内蔵 ROM プログラムより分岐する HyperFlash 空間の先頭番地（H'3000_0000）に配置し、ブート起動用内蔵 ROM プログラムから実行できるようにしています。ローダプログラム実行後、アプリケーションプログラムに含まれているシグネチャのチェックを行い、アプリケーションプログラムの先頭番地に分岐します。

アプリケーションプログラムの先頭アドレスは、「linker_script.ld」のリンクスクリプトのシンボル定義 `"__application_base_address"`により指定しています。

2 アプリケーションプログラム（プロジェクト名：rza2m_hyperflash_boot_sample_osless_gcc）

アプリケーションプログラムは、ローダプログラムにて HyperFlash に最適な設定後に実行するプログラムです。アプリケーションプログラムの"VECTOR_TABLE"のセクションが、

`"__application_base_address"`で指定したアドレスと一致するように配置アドレスを変更してください。

サンプルコードでは、アプリケーションプログラムを H'3004_0000 番地に配置しています。

図5.1に本アプリケーションノートのサンプルコードの動作概要を示します。

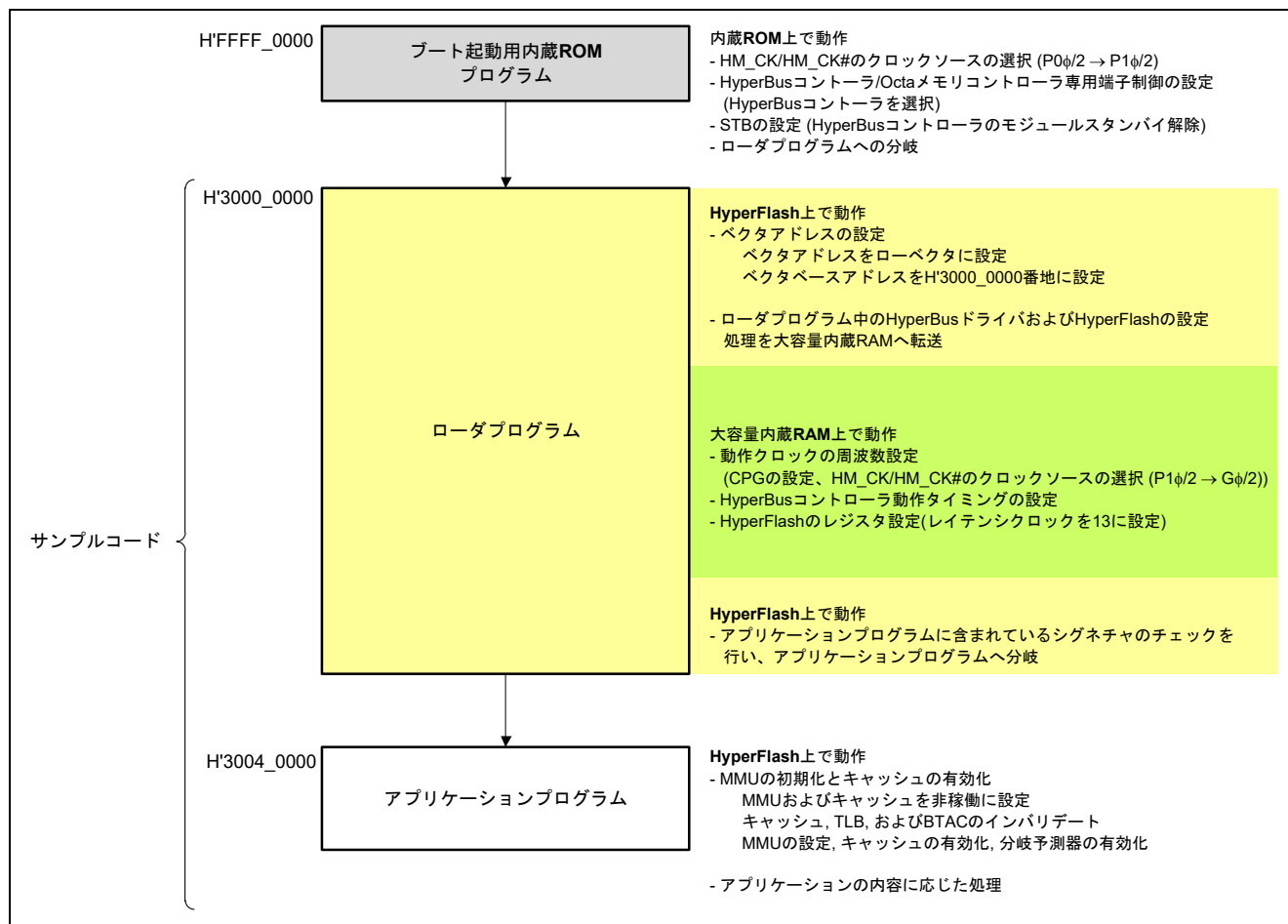


図5.1 サンプルコードの動作概要

5.1.3 ローダプログラムの動作概要

ローダプログラムは、ブート起動用内蔵 ROM プログラムの処理完了後に実行されるプログラムです。ローダプログラムは、ブート起動用内蔵 ROM プログラムより分岐する HyperFlash 空間の先頭番地 (H'3000_0000) に配置してください。

ブート起動用内蔵 ROM プログラムは、HyperFlash にアクセスするために HyperBus コントローラを設定します。設定により、RZ/A2Mは接続された HyperFlash に対してリードが可能となり、HyperFlash 空間に配置されたプログラムを直接実行することが可能な状態となります。RZ/A2M は一般的な HyperFlash に対してアクセスできる設定となっており、ローダプログラムにてお客様が使用する HyperFlash に最適にアクセスするための処理（クロック周波数の変更、HyperFlash の仕様に合わせたレイテンシ変更処理など）を行う必要があります。

ローダプログラムの HyperBus コントローラ設定処理および HyperFlash のレジスタ変更処理は、HyperFlash 空間に配置されたプログラムで設定することはできないため、その処理を大容量内蔵 RAM に転送して実行しています。

ブート起動用内蔵 ROM プログラムおよびローダプログラム実行後の設定については、表5.2を参照してください。

表5.2にブート起動用内蔵 ROM プログラムおよびローダプログラムの設定内容を示します。

ローダプログラムにて、表5.2に示す設定を行った後、アプリケーションプログラムの先頭番地に分岐します。サンプルコードでは、アプリケーションプログラムを H'3004_0000 番地に配置しています。

表5.2 ブート起動用内蔵 ROM プログラムおよびローダプログラムの設定内容

	項目	ブート起動用内蔵 ROM プログラム実行後	ローダプログラム実行後
CPG 設定	動作クロックの設定 クロックモード 1 で EXTAL から 24MHz 入力時	I ϕ =132[MHz] G ϕ =264[MHz] B ϕ =132[MHz] P1 ϕ =66[MHz] P0 ϕ =33[MHz]	I ϕ =528[MHz] G ϕ =264[MHz] B ϕ =132[MHz] P1 ϕ =66[MHz] P0 ϕ =33[MHz]
	HYPCLK の選択 SCLKSEL.HYMC[R1:0]	P1 ϕ を選択 B'01	G ϕ を選択 B'11
HyperBus コントローラ設定	最大時間設定イネーブル : MCR0.MAXEN	HM_CS0#信号の Low 時間を設定しない 0	
	最大時間設定 : MCR0.MAXLEN[8:0]	MCR0.MAXEN が"0"のため、最大時間設定は無効 -	
	リードチップセレクト タイミング設定 : MTR0.RCSHI[3:0]	HM_CS0#信号がネゲートされてから次のリードアクセス開始までの時間 B'0000 (1.5 クロックサイクル)	
	ライトチップセレクト タイミング設定 : MTR0.WCSHI[3:0]	HM_CS0#信号がネゲートされてから次のライトアクセス開始までの時間 B'0000 (1.5 クロックサイクル)	
	リードチップセレクト セットアップタイミング設定 : MTR0.RCSS[3:0]	HM_CS0#信号がアサートされてから次のリードアクセス開始までの時間 B'0000 (1 クロックサイクル)	
	ライトチップセレクト セットアップタイミング設定 : MTR0.WCSS[3:0]	HM_CS0#信号がアサートされてから次のライトアクセス開始までの時間 B'0000 (1 クロックサイクル)	
	リードチップセレクト ホールドタイミング設定 : MTR0.RCSH[3:0]	リードアクセス終了から HM_CS0#信号がネゲートされるまでの時間 B'0000 (1 クロックサイクル)	
	ライトチップセレクト ホールドタイミング設定 : MTR0.RCSH[3:0]	ライトアクセス終了から HM_CS0#信号がネゲートされるまでの時間 B'0000 (1 クロックサイクル)	
HyperFlash レジスタ設定	Volatile Configuration Register	16 クロック読み出しレイテンシ xVCR[7:4] = B'1011	13 クロック読み出しレイテンシ xVCR[7:4] = B'1000
その他	CP15 システムコントロール レジスタ ベクタビット	1 (ハイベクタ)	0 (ローベクタ)
	CP15 ベクタベース アドレスレジスタ (VBAR) (注)	-	H'3000_0000

【注】 CP15 システムコントロールレジスタの V ビットによってハイベクタ (H'FFFF_0000) またはローベクタ (H'0000_0000) にベクタベースアドレスを配置することができ、ローベクタに設定した場合は VBAR によりベクタベースアドレスを設定することが可能です。

5.1.4 アプリケーションプログラム

(1) アプリケーションプログラムの動作

本アプリケーションプログラムでは、RZ/A2Mに接続されたHyperRAMにアクセスする例として、HyperRAM空間のHyperBusコントローラおよびHyperRAMのレジスタ設定を行い、HyperRAM空間に配置した領域に対してライトおよびリードの処理を行います。

リセット解除後に、ブート起動用内蔵ROMプログラム、ローダプログラムの順にプログラムが実行され、H'3004_0000番地に配置されているアプリケーションプログラムのスタートアップ処理に分岐します。

スタートアップ処理では、スタックポインタの設定を行った後、HyperRAMにアクセスできるように、表5.3に示す内容でHyperBusコントローラおよびHyperRAMのレジスタの設定を行います（注）。MMU、FPUの設定を行い、セクションの初期化を行った後、resetprg関数に分岐します。

resetprg関数では、RTCとUSB未使用チャンネルの初期化処理を実行後、L1キャッシュおよびL2キャッシュの有効化とINTCの初期化を行い、割り込み処理高速化のためにVBARに大容量内蔵RAMのアドレスを設定し、IRQ割り込みおよびFIQ割り込みを許可にし、main関数をコールしています。

また本アプリケーションプログラムのmain関数では、CPG、OSTMチャンネル0、SCIFAチャンネル4、GPIOの初期設定処理を行います。これら初期化処理により、シリアルインタフェースに接続されたホストPC上のターミナルに文字列（起動メッセージ）を出力し、OSTMのチャンネル0をインターバルタイマモードに設定してタイマを起動します。500msの周期でOSTMチャンネル0の割り込みを発生させ、CPUボード上のLEDを割り込み処理により500msごとに点灯および消灯を繰り返す処理を行います。

アプリケーションプログラムで実施している初期設定の詳細は、「RZ/A2Mグループ 初期設定例」のアプリケーションノートを参照してください。

また、本アプリケーションプログラムのmain関数では、HyperRAMキャッシュ無効領域へのライトを行います。

【注】 HyperRAM空間のHyperBusコントローラの設定は、HyperRAMをワークRAM領域として使用できるようにセクション初期化の前に行い、またHyperRAMのレジスタへのアクセスはMMUをイネーブルに設定する前に行っています（HyperRAMのレジスタへのアクセスは、メモリ属性がストロングリーオーダ属性、またはデバイス属性のHyperRAMの領域に対して実施する必要があります）。

表5.3にアプリケーションプログラムのHyperBusコントローラおよびHyperRAMのレジスタの設定内容を示します。

表5.3 HyperBus コントローラおよび HyperRAM のレジスタの設定内容

	項目	アプリケーションプログラム実行後
HyperBus コントローラ設定	最大時間設定イネーブル : MCR1.MAXEN	HM_CS1#信号の Low 時間を設定する 0
	最大時間設定 : MCR1.MAXLEN[8:0]	MCR1.MAXEN が"0"のため、最大時間設定は無効 -
	アクセス対象 : MCR1.CRT	リードライト時のアクセス対象 0 (メモリベース)
	デバイスタイプ : MCR1.DEVTYPE	1 (HyperRAM)
	リードチップセレクト タイミング設定 : MTR1.RCSHI[3:0]	HM_CS1#信号がネゲートされてから次のリードアクセス開始までの時間 B'0000 (1.5 クロックサイクル)
	ライトチップセレクト タイミング設定 : MTR1.WCSHI[3:0]	HM_CS1#信号がネゲートされてから次のライトアクセス開始までの時間 B'0000 (1.5 クロックサイクル)
	リードチップセレクト セットアップタイミング設定 : MTR1.RCSS[3:0]	HM_CS1#信号がアサートされてから次のリードアクセス開始までの時間 B'0000 (1 クロックサイクル)
	ライトチップセレクト セットアップタイミング設定 : MTR1.WCSS[3:0]	HM_CS1#信号がアサートされてから次のライトアクセス開始までの時間 B'0000 (1 クロックサイクル)
	リードチップセレクト ホールドタイミング設定 : MTR1.RCSH[3:0]	リードアクセス終了から HM_CS1#信号がネゲートされるまでの時間 B'0000 (1 クロックサイクル)
	ライトチップセレクト ホールドタイミング設定 : MTR1.RCSH[3:0]	ライトアクセス終了から HM_CS1#信号がネゲートされるまでの時間 B'0000 (1 クロックサイクル)
	レイテンシ設定 : MTR1.LTCY[3:0]	B'0000 (5 クロックサイクル)
HyperRAM レジスタ設定	Configuration Register 0	CR0[7:4] = B'0000 (5 クロックレイテンシ)

(2) アプリケーションプログラム作成時の注意事項

アプリケーションプログラムは、ローダプログラムから分岐するアドレスに配置してください。なお、アプリケーションプログラムは、ローダプログラムとは異なる HyperFlash のセクタに配置してください。

RZ/A2M CPUボードに搭載されているCypress社製 HyperBus MCP (S71KS512SC0BHV000) の HyperFlash のセクタサイズは 256KB です。サンプルコードでは、アプリケーションプログラムをセクタ 01 の H'3004_0000 番地に配置しています。

図5.2にサンプルコードのプログラム配置を示します。

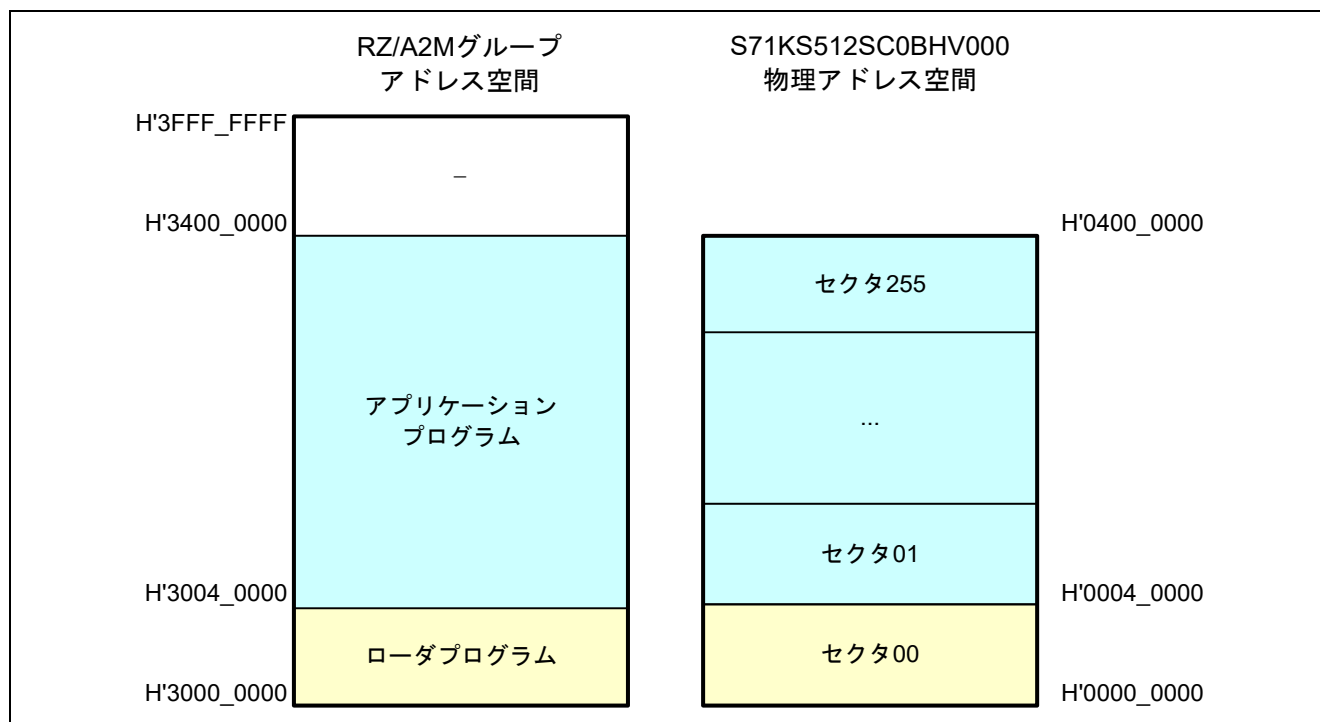


図5.2 サンプルコードのプログラム配置

アプリケーションプログラムの先頭アドレスは、以下の内容を変更することにより、アドレス配置の変更が可能です。

- ローダプログラムのプロジェクト
アプリケーションプログラムの先頭アドレスへの分岐は、ローダプログラム (reset_handler.asm) にて行っています。「linker_script.ld」のリンクスクリプトのシンボル定義"__application_base_address"により、分岐先を指定してください。
- アプリケーションプログラムのプロジェクト
アプリケーションプログラムの"VECTOR_TABLE"のセクションが、"__application_base_address"で指定したアドレスと一致するように配置アドレスを変更してください。

5.2 サンプルコード実行時の周辺機能の設定およびメモリ配置

5.2.1 周辺機能の設定

表5.4にサンプルコード実行時の周辺機能の設定内容を示します。

表5.4 周辺機能の設定内容

モジュール	設定内容
CPG	<p>CPU クロック : PLL 回路の周波数に対して×1/2 倍に設定 内部バスクロック : PLL 回路の周波数に対して×1/8 倍に設定 周辺クロック 1 (P1φ) : PLL 回路の周波数に対して×1/16 倍に設定</p> <p>クロックモード 1 (分周器 1 : ×1/2 倍、PLL 回路 : ×88 倍) で、入力クロックが 24MHz の場合に以下の周波数となるように設定</p> <ul style="list-style-type: none"> • CPU クロック (Iφ) : 528MHz • 画像処理クロック (Gφ) : 264MHz • 内部バスクロック (Bφ) : 132MHz • 周辺クロック 1 (P1φ) : 66MHz • 周辺クロック 0 (P0φ) : 33MHz • HM_CK/HM_CK# : 132MHz (Gφ 選択時) • CKIO クロック : 132MHz (Bφ 選択時)
HyperBus コントローラ	CPU が HyperFlash 空間に接続された HyperFlash および HyperRAM 空間に接続された HyperRAM から、直接リードする信号を生成するための設定
STB	保持用内蔵 RAM へのライト許可および周辺機能へのクロック供給 STBCR3, STBCR4, STBCR9 で OSTM0, OSTM2, SCIFA4, HyperBus コントローラにクロックを供給
GPIO	<p>PORT6、PORT9 の兼用端子機能を設定</p> <ul style="list-style-type: none"> • P6_0 : LED の点灯および消灯 • P9_1 : RxD4、P9_0 : TxD4
OSTM	<p>チャンネル 0、チャンネル 2 をインターバルタイマモードに設定</p> <ul style="list-style-type: none"> • チャンネル 0 P1φ=66MHz の時に 500ms ごとに割り込み要求を発生するようにタイマカウントを設定。LED 点灯および消灯処理で使用。 • チャンネル 2 P1φ=66MHz の時に 1ms ごとに割り込み要求を発生するようにタイマカウントを設定。OS Abstraction Layer による時間管理に使用。
INTC	INTC の初期設定および OSTM チャンネル 0 割り込み (割り込み ID が 88) ハンドラの登録、OSTM チャンネル 2 割り込み (割り込み ID が 90) ハンドラの登録、SCIFA チャンネル 4 割り込み (割り込み ID が 322, 323) ハンドラの登録と実行
SCIFA	<p>チャンネル 4 を調歩同期式モードに設定</p> <ul style="list-style-type: none"> • データ長 : 8 ビット • ストップビット長 : 1 ビット • パリティ : なし • データ転送方向 : LSB ファースト転送 <p>P1φ=66MHz の時に、クロックソースを分周なし、ボーレートジェネレータは倍速モード、ビットレートの 8 倍の基本クロックで動作するように設定。ビットレートが 115200bps となるように、ビットレート値に 71 を設定 (誤差 : -0.53%)</p>

5.2.2 メモリマップ

図5.3にRZ/A2Mグループのアドレス空間とRZ/A2M CPUボードのメモリマップを示します。

サンプルコードでは、ROM 領域を使用するコードおよびデータを HyperFlash 空間に接続した HyperFlash に配置し、RAM 領域を使用するコードおよびデータを大容量内蔵 RAM に配置するようにしています。

	RZ/A2Mグループの アドレス空間	RZ/A2M CPUボード メモリマップ
H'FFFF FFFF	内蔵IO領域 および 予約領域 (2044MB)	内蔵IO領域 および 予約領域 (2044MB)
H'8040 0000	大容量内蔵RAM (4MB)	大容量内蔵RAM (4MB)
H'8000 0000	予約領域 (256MB)	予約領域 (256MB)
H'7000 0000	OctaRAM™空間 (256MB)	-
H'6100 0000	OctaFlash™空間 (256MB)	-
H'6000 0000	HyperRAM™空間 (256MB)	-
H'5400 0000	HyperFlash™空間 (256MB)	HyperRAM™ (8MB)
H'5000 0000		-
H'4080 0000		HyperFlash™ (64MB)
H'4000 0000		-
H'3400 0000	SPIマルチI/Oバス 空間 (256MB)	シリアルフラッシュ メモリ (64MB)
H'3000 0000	内蔵IO領域および 予約領域 (128MB)	内蔵IO領域および 予約領域 (128MB)
H'2400 0000	CS5空間 (64MB)	-
H'2000 0000	CS4空間 (64MB)	-
H'1800 0000	CS3空間 (64MB)	-
H'1400 0000	CS2空間 (64MB)	-
H'1000 0000	CS1空間 (64MB)	-
H'0C00 0000	CS0空間 (64MB)	-
H'0800 0000		
H'0400 0000		
H'0000 0000		

図5.3 メモリマップ

5.2.3 アプリケーションプログラムで使用する仮想アドレス空間

アプリケーションプログラムでは、MMU を有効に設定して、図5.4および図5.5に示すように CPU がアクセス可能な仮想アドレス空間を準備しています。なお、ローダプログラムは MMU が無効の状態で作動します。

RZ/A2Mグループの 物理アドレス空間		サンプルコードの 仮想アドレス空間	
H'803F FFFF	大容量内蔵RAM (4MB)	H'803F FFFF	大容量内蔵RAM キャッシュ有効領域 (4MB)
H'8000 0000		H'8000 0000	
	予約領域 (256MB)		SPIマルチI/Oバス空間 キャッシュ無効領域 (256MB)
H'7000 0000		H'7000 0000	
	OctaRAM空間 (256MB)		OctaRAM キャッシュ有効領域 (256MB)
H'6000 0000		H'6000 0000	
	OctaFlash空間 (256MB)		OctaFlash キャッシュ有効領域 (256MB)
H'5000 0000		H'5000 0000	
	HyperRAM空間 (256MB)		HyperRAM キャッシュ有効領域 (256MB)
H'4000 0000		H'4000 0000	
	HyperFlash空間 (256MB)		HyperFlash キャッシュ有効領域 (256MB)
H'3000 0000		H'3000 0000	
	SPIマルチI/Oバス 空間 (256MB)		SPIマルチI/Oバス空間 キャッシュ有効領域 (256MB)
H'2000 0000		H'2000 0000	
H'1F00 0000	内蔵I/O領域 (16MB)	H'1F00 0000	内蔵I/O領域 (16MB)
H'1800 0000	予約領域 (112MB)	H'1800 0000	予約領域 (112MB)
	CS5空間 (64MB)		CS5空間 (64MB) (未使用)
H'1400 0000		H'1400 0000	
	CS4空間 (64MB)		CS4空間 (64MB) (未使用)
H'1000 0000		H'1000 0000	
	CS3空間 (64MB)		CS3空間キャッシュ 有効領域 (64MB)
H'0C00 0000		H'0C00 0000	
	CS2空間 (64MB)		CS2空間 (64MB) (未使用)
H'0800 0000		H'0800 0000	
	CS1空間 (64MB)		CS1空間 (64MB) (未使用)
H'0400 0000		H'0400 0000	
	CS0空間 (64MB)		CS0空間 (64MB) (未使用)
H'0000 0000		H'0000 0000	

図5.4 アプリケーションプログラムで使用する仮想アドレス空間 (1/2)

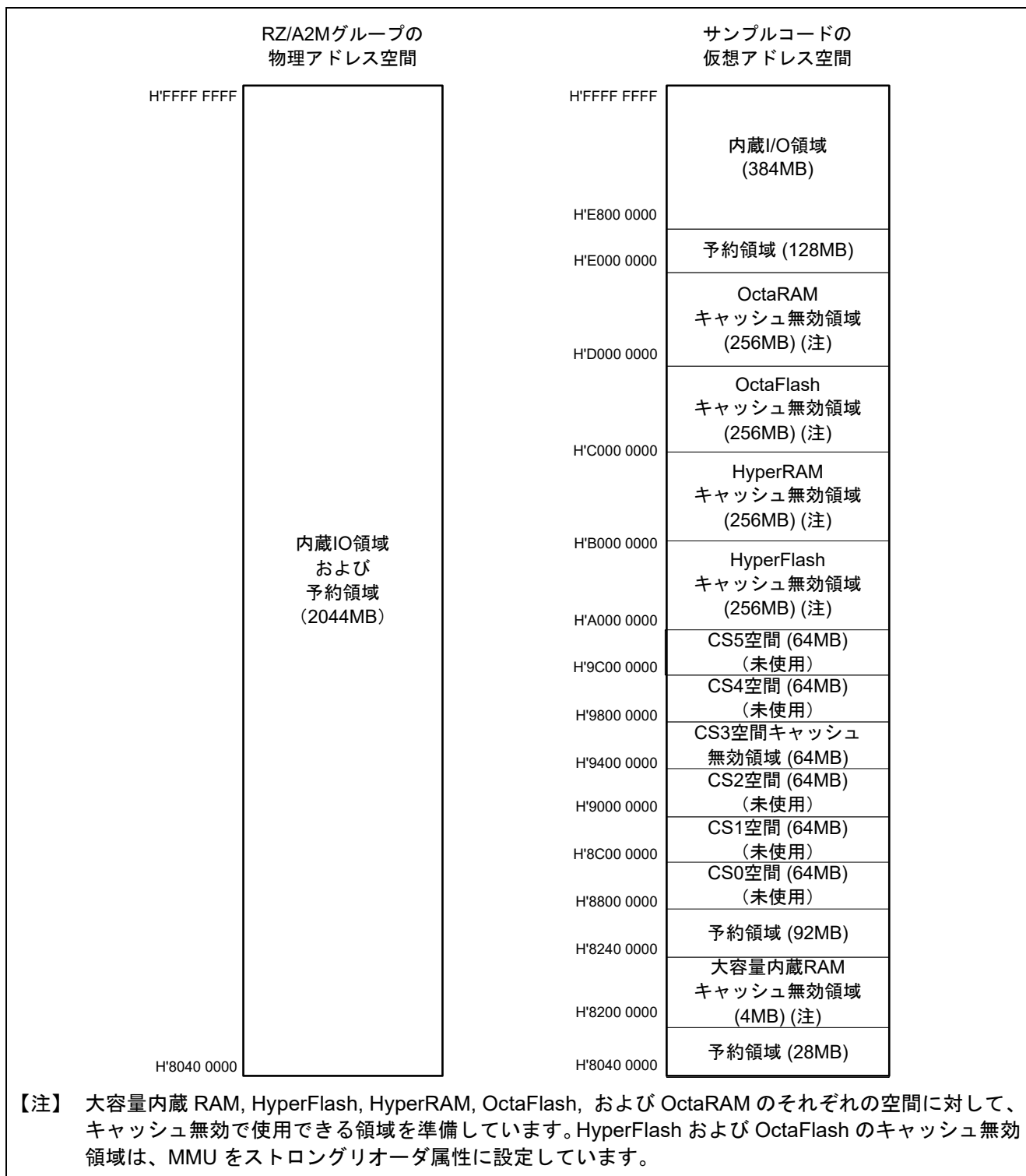


図5.5 アプリケーションプログラムで使用する仮想アドレス空間 (2/2)

5.2.4 サンプルコードのセクション配置

表5.5にローダプログラムで使用するセクション名とオブジェクト名一覧を、表5.6および表5.7にアプリケーションプログラムで使用するセクション名とオブジェクト名一覧を示します。

表5.5 ローダプログラムで使用するセクション名とオブジェクト名一覧

出力セクション名	入力セクション名 入力オブジェクト名	内容	ロード 領域	実行 領域
LOAD_MODULE1	VECTOR_TABLE	例外処理ベクタテーブル	FLASH	FLASH
LOAD_MODULE2	*r_cpg/*.o (.text .rodata)	CPG の設定処理	FLASH	LRAM
	*rza_io_regrw.o (.text .rodata)	IO レジスタアクセス処理		
	r_hyperbus/.o (.text .rodata)	HyperBus コントローラの設定処理		
	hwsetup.o (.text .rodata)	HardwareSetup の設定処理		
	* (.data)	デフォルトの初期値ありデータ領域		
LOAD_MODULE3	RESET_HANDLER	リセット処理	FLASH	FLASH
	INIT_SECTION */sections.o	セクション初期化処理		
	* (.text)	デフォルトのコード領域		
	* (.rodata)	デフォルトの定数データ領域		
.data.memclk_setup	*r_memclk_setup.o (.text .rodata .data)	メモリクロックの設定処理	FLASH	LRAM
	*r_*_memclk_setup.o (.text .rodata .data)	各ドライバ用メモリクロックの設定処理		
.bss.memclk_setup	*r_memclk_setup.o (.bss COMMON)	メモリクロックの設定処理の初期値なしデータ領域	-	LRAM
	*r_*_memclk_setup.o (.bss COMMON)	各ドライバ用メモリクロックの設定処理の初期値なしデータ領域		
.stack	なし	SVC モードのスタック領域	-	LRAM
.bss	* (.bss .bss.*) * (COMMON)	デフォルトの初期値なしデータ領域	-	LRAM
.heap	なし	ヒープ領域	-	LRAM

【注】 表中のロード領域および実行領域において、FLASHはHyperFlashの領域を、LRAMは大容量内蔵RAMの領域を表します。

表5.6 アプリケーションプログラムで使用するセクション名とオブジェクト名一覧(1/2)

出力セクション名	入力セクション名 入力オブジェクト名	内容	ロード 領域	実行 領域
LOAD_MODULE1	VECTOR_TABLE	例外処理ベクタテーブル	FLASH	FLASH
LOAD_MODULE2	*r_cpg/*.o (.text .rodata .data)	CPG の設定処理	FLASH	LRAM
	*rza_io_regrw.o (.text .rodata .data)	IO レジスタアクセス処理		
	r_hyperbus/.o (.text .rodata .data)	HyperBus コントローラの設定処理		
	hwsetup.o (.text .rodata .data)	HardwareSetup の設定処理		
LOAD_MODULE3	*r_cpg/*.o (.bss)	CPG の設定処理の初期値なしデータ領域	-	LRAM
	*rza_io_regrw.o (.bss)	IO レジスタアクセス処理用の初期値なしデータ領域		
LOAD_MODULE4	RESET_HANDLER	リセット処理	FLASH	FLASH
	INIT_SECTION */sections.o	セクション初期化処理		
.data.memclk_setup	*r_memclk_setup.o (.text .rodata .data)	メモリクロックの設定処理	FLASH	LRAM
	*r_*_memclk_setup.o (.text .rodata .data)	各ドライバ用メモリクロックの設定処理		
.bss.memclk_setup	*r_memclk_setup.o (.bss COMMON)	メモリクロックの設定処理の初期値なしデータ領域	-	LRAM
	*r_*_memclk_setup.o (.bss COMMON)	各ドライバ用メモリクロックの設定処理の初期値なしデータ領域		

表5.7 アプリケーションプログラムで使用するセクション名とオブジェクト名一覧(2/2)

出力セクション名	入力セクション名 入力オブジェクト名	内容	ロード 領域	実行 領域
.data	VECTOR_MIRROR_TABLE	例外処理ベクタテーブル	FLASH	LRAM
	/r_intc_.o (.text .rodata .data)	INTC ドライバ処理用コード領域		
	IRQ_FIQ_HANDLER	IRQ/FIQ ハンドラ処理		
.bss	なし	なし	–	LRAM
.uncached_RAM	*/r_cache_*.o (.bss)	L1 および L2 キャッシュ設定処理 用初期値なしデータ領域 (注 2)	–	LRAM
	UNCACHED_BSS	初期値なしデータ領域 (キャッ シュ無効設定)		
.uncached_RAM2	*/r_cache_*.o (.text .rodata .data)	L1 および L2 キャッシュ設定処理 (注 2)	FLASH	LRAM
	UNCACHED_DATA	初期値ありデータ領域 (キャッ シュ無効設定)		
.mmu_page_table	なし	MMU 変換テーブル領域	–	LRAM
.stack	なし	システムモードのスタック領域	–	LRAM
		IRQ モードのスタック領域		
		FIQ モードのスタック領域		
		SVC モードのスタック領域		
		アボート(ABT)モードの スタック領域		
.text2	*(.text .text.*)	デフォルトのコード領域	FLASH	FLASH
	(.rodata .rodata.)	デフォルトの定数データ領域		
.data2	*(.data .data.*)	デフォルトの初期値ありデータ領 域	FLASH	LRAM
.bss2	*(.bss .bss.*)	デフォルトの初期値なしデータ領 域	–	LRAM
	*(COMMON)			
.heap	なし	ヒープ領域	–	LRAM

- 【注】 1. 表中のロード領域および実行領域において、FLASH は HyperFlash の領域を、LRAM は大容量内蔵 RAM の領域を表します。
2. このセクションは、キャッシュ無効領域に配置する必要があります。

5.3 使用割り込み一覧

ローダプログラムでは割り込みは使用していません。

アプリケーションプログラムで使用する割り込みについては、「RZ/A2M グループ 初期設定例」のアプリケーションノートをご参照ください。

5.4 データ型一覧

表5.8にサンプルコードで使用するデータ型一覧を示します。

表5.8 サンプルコードで使用するデータ型一覧

シンボル	内容
char_t	8 ビット文字
bool_t	論理型。値は true (1)、false (0)
int_t	高速な整数、符号あり、本サンプルコードでは 32 ビット整数。
int8_t	8 ビット整数、符号あり (標準ライブラリ stdint.h にて定義)
int16_t	16 ビット整数、符号あり (標準ライブラリ stdint.h にて定義)
int32_t	32 ビット整数、符号あり (標準ライブラリ stdint.h にて定義)
int64_t	64 ビット整数、符号あり (標準ライブラリ stdint.h にて定義)
uint8_t	8 ビット整数、符号なし (標準ライブラリ stdint.h にて定義)
uint16_t	16 ビット整数、符号なし (標準ライブラリ stdint.h にて定義)
uint32_t	32 ビット整数、符号なし (標準ライブラリ stdint.h にて定義)
uint64_t	64 ビット整数、符号なし (標準ライブラリ stdint.h にて定義)
float32_t	32 ビット浮動小数
float64_t	64 ビット浮動小数
float128_t	128 ビット浮動小数

5.5 定数一覧

表5.9～表5.11にサンプルコードで使用する定数を示します。

アプリケーションプログラムで使用する定数については、「RZ/A2M グループ 初期設定例」のアプリケーションノートをご参照ください。

表5.9 ローダプログラムで使用する定数(1/3)

定数名	設定値	内容
DRV_SUCCESS	(0)	正常終了
DRV_ERROR	(-1)	エラー終了
HYPERBUS_CS0_AREA	(0)	制御対象とするメモリ空間(CS0/CS1)を指定します CS0 空間の制御を指定します
HYPERBUS_CS1_AREA	(1)	CS1 空間の制御を指定します
HYPERBUS_MEMORY_SPACE	(0)	CS1 空間に接続された HyperRAM をアクセスする際の領域 (メモリまたはレジスタ) を指定します メモリ領域へのアクセスを設定します
HYPERBUS_REGISTER_SPACE	(1)	レジスタ領域へのアクセスを設定します
HYPERBUS_NO_INIT	(0)	HyperBus コントローラおよび接続しているデバイスの初期化 を実行するプログラムの指定を行います 初期化は行いません
HYPERBUS_INIT_AT_LOADER	(1)	ローダプログラムで初期化を行います
HYPERBUS_INIT_AT_APP	(2)	アプリケーションプログラムで初期化を行います
HYPERBUS_MAXEN_OFF	(0)	HM_CSn#信号の Low 時間を MCR0 または MCR1 の MAXLEN[8:0]ビットでの指定を無効にします
HYPERBUS_MAXEN_ON	(1)	HM_CSn#信号の Low 時間設定を MCR0 または MCR1 の MAXLEN[8:0]ビットで指定します
HYPERBUS_CSHI_1P5_CYCLE	(0)	HM_CSn#信号がネゲートされてから次のアクセス開始ま での最小待ち時間を指定します 1.5 クロックサイクル
HYPERBUS_CSHI_2P5_CYCLE	(1)	2.5 クロックサイクル
HYPERBUS_CSHI_3P5_CYCLE	(2)	3.5 クロックサイクル
HYPERBUS_CSHI_4P5_CYCLE	(3)	4.5 クロックサイクル
HYPERBUS_CSHI_5P5_CYCLE	(4)	5.5 クロックサイクル
HYPERBUS_CSHI_6P5_CYCLE	(5)	6.5 クロックサイクル
HYPERBUS_CSHI_7P5_CYCLE	(6)	7.5 クロックサイクル
HYPERBUS_CSHI_8P5_CYCLE	(7)	8.5 クロックサイクル
HYPERBUS_CSHI_9P5_CYCLE	(8)	9.5 クロックサイクル
HYPERBUS_CSHI_10P5_CYCLE	(9)	10.5 クロックサイクル
HYPERBUS_CSHI_11P5_CYCLE	(10)	11.5 クロックサイクル
HYPERBUS_CSHI_12P5_CYCLE	(11)	12.5 クロックサイクル
HYPERBUS_CSHI_13P5_CYCLE	(12)	13.5 クロックサイクル
HYPERBUS_CSHI_14P5_CYCLE	(13)	14.5 クロックサイクル
HYPERBUS_CSHI_15P5_CYCLE	(14)	15.5 クロックサイクル
HYPERBUS_CSHI_16P5_CYCLE	(15)	16.5 クロックサイクル

表5.10 ローダプログラムで使用する定数(2/3)

定数名	設定値	内容
		HM_CSn#信号のアサートから HM_CSn#のセットアップ時間を指定します
HYPERBUS_CSS_1_CYCLE	(0)	1クロックサイクル
HYPERBUS_CSS_2_CYCLE	(1)	2クロックサイクル
HYPERBUS_CSS_3_CYCLE	(2)	3クロックサイクル
HYPERBUS_CSS_4_CYCLE	(3)	4クロックサイクル
HYPERBUS_CSS_5_CYCLE	(4)	5クロックサイクル
HYPERBUS_CSS_6_CYCLE	(5)	6クロックサイクル
HYPERBUS_CSS_7_CYCLE	(6)	7クロックサイクル
HYPERBUS_CSS_8_CYCLE	(7)	8クロックサイクル
HYPERBUS_CSS_9_CYCLE	(8)	9クロックサイクル
HYPERBUS_CSS_10_CYCLE	(9)	10クロックサイクル
HYPERBUS_CSS_11_CYCLE	(10)	11クロックサイクル
HYPERBUS_CSS_12_CYCLE	(11)	12クロックサイクル
HYPERBUS_CSS_13_CYCLE	(12)	13クロックサイクル
HYPERBUS_CSS_14_CYCLE	(13)	14クロックサイクル
HYPERBUS_CSS_15_CYCLE	(14)	15クロックサイクル
HYPERBUS_CSS_16_CYCLE	(15)	16クロックサイクル

表5.11 ローダプログラムで使用する定数(3/3)

定数名	設定値	内容
		アクセス終了から HM_CSn#のネゲートのホールド時間を指定します
HYPERBUS_CSH_1_CYCLE	(0)	1クロックサイクル
HYPERBUS_CSH_2_CYCLE	(1)	2クロックサイクル
HYPERBUS_CSH_3_CYCLE	(2)	3クロックサイクル
HYPERBUS_CSH_4_CYCLE	(3)	4クロックサイクル
HYPERBUS_CSH_5_CYCLE	(4)	5クロックサイクル
HYPERBUS_CSH_6_CYCLE	(5)	6クロックサイクル
HYPERBUS_CSH_7_CYCLE	(6)	7クロックサイクル
HYPERBUS_CSH_8_CYCLE	(7)	8クロックサイクル
HYPERBUS_CSH_9_CYCLE	(8)	9クロックサイクル
HYPERBUS_CSH_10_CYCLE	(9)	10クロックサイクル
HYPERBUS_CSH_11_CYCLE	(10)	11クロックサイクル
HYPERBUS_CSH_12_CYCLE	(11)	12クロックサイクル
HYPERBUS_CSH_13_CYCLE	(12)	13クロックサイクル
HYPERBUS_CSH_14_CYCLE	(13)	14クロックサイクル
HYPERBUS_CSH_15_CYCLE	(14)	15クロックサイクル
HYPERBUS_CSH_16_CYCLE	(15)	16クロックサイクル
		HyperFlash 空間アクセス時のリードおよび HyperRAM 空間アクセス時のリードライトレイテンシを指定します
HYPERBUS_LTCY_5_CYCLE	(0)	5クロックレイテンシ
HYPERBUS_LTCY_6_CYCLE	(1)	6クロックレイテンシ
HYPERBUS_LTCY_7_CYCLE	(2)	7クロックレイテンシ
HYPERBUS_LTCY_8_CYCLE	(3)	8クロックレイテンシ
HYPERBUS_LTCY_9_CYCLE	(4)	9クロックレイテンシ
HYPERBUS_LTCY_10_CYCLE	(5)	10クロックレイテンシ
HYPERBUS_LTCY_11_CYCLE	(6)	11クロックレイテンシ
HYPERBUS_LTCY_12_CYCLE	(7)	12クロックレイテンシ
HYPERBUS_LTCY_13_CYCLE	(8)	13クロックレイテンシ
HYPERBUS_LTCY_14_CYCLE	(9)	14クロックレイテンシ
HYPERBUS_LTCY_15_CYCLE	(10)	15クロックレイテンシ
HYPERBUS_LTCY_16_CYCLE	(11)	16クロックレイテンシ

【注】 表中の HM_CSn#は、HM_CS0#および HM_CS1#を示しています。

5.6 構造体/共用体一覧

表5.12および表5.13にローダプログラムで使用する HyperBus コントローラ初期設定構造体を示します。

表5.12 HyperBus コントローラ初期設定構造体 (st_hyperbus_cfg_t) (1/2)

空間	メンバ名	内容
CS0	e_hyperbus_init_control init_flag0	CS0 空間の HyperBus コントローラおよび HyperFlash の初期化方法を指定します。 HYPERBUS_NO_INIT : 初期化しない HYPERBUS_INIT_AT_LOADER : ロードプロジェクトで初期化 HYPERBUS_INIT_AT_APP : アプリケーションプロジェクトで初期化
	e_hyperbus_maxen_t maxen0	HM_CS0#信号の Low 時間設定を maxlen0 で制御するかを設定します。 HYPERBUS_MAXEN_OFF : 設定しない HYPERBUS_MAXEN_ON : 設定する
	uint16_t maxlen0	HM_CS0#信号の最大リードライト処理時間を設定します。 0 : 1 クロックサイクル ... 511 : 512 クロックサイクル
	e_hyperbus_cshi_t rcshi0	HM_CS0#信号がネゲートされてから次のリードアクセス開始までの時間を設定します。 HYPERBUS_CSHI_1P5_CYCLE : 1.5 クロックサイクル ... HYPERBUS_CSHI_16P5_CYCLE : 16.5 クロックサイクル
	e_hyperbus_cshi_t wcshi0	HM_CS0#信号がネゲートされてから次のライトアクセス開始までの時間を設定します。 HYPERBUS_CSHI_1P5_CYCLE : 1.5 クロックサイクル ... HYPERBUS_CSHI_16P5_CYCLE : 16.5 クロックサイクル

表5.13 HyperBus コントローラ初期設定構造体 (st_hyperbus_cfg_t) (2/2)

空間	メンバ名	内容
CS0	e_hyperbus_css_t rcss0	HM_CS0#信号がアサートされてから次のリードアクセス開始までの時間を設定します。 HYPERBUS_CSS_1_CYCLE : 1 クロックサイクル ... HYPERBUS_CSS_16_CYCLE : 16 クロックサイクル
	e_hyperbus_css_t wcss0	HM_CS0#信号がアサートされてから次のライトアクセス開始までの時間を設定します。 HYPERBUS_CSS_1_CYCLE : 1 クロックサイクル ... HYPERBUS_CSS_16_CYCLE : 16 クロックサイクル
	e_hyperbus_csh_t rcsh0	リードアクセス終了から HM_CS0#信号がネゲートされるまでの時間を設定します。 HYPERBUS_CSH_1_CYCLE : 1 クロックサイクル ... HYPERBUS_CSH_16_CYCLE : 16 クロックサイクル
	e_hyperbus_csh_t wcsh0	ライトアクセス終了から HM_CS0#信号がネゲートされるまでの時間を設定します。 HYPERBUS_CSH_1_CYCLE : 1 クロックサイクル ... HYPERBUS_CSH_16_CYCLE : 16 クロックサイクル
	e_hyperbus_ltcy_t operate_ltcy0	動作時のリードレイテンシを設定します。 HYPERBUS_LTCY_5_CYCLE : 5 クロックレイテンシ ... HYPERBUS_LTCY_6_CYCLE : 16 クロックレイテンシ

表5.14および表5.15にアプリケーションプログラムで使用する HyperBus コントローラ初期設定構造体を示します。

表5.14 HyperBus コントローラ初期設定構造体 (st_hyperbus_cfg_t) (1/2)

空間	メンバ名	内容
CS1	e_hyperbus_init_control init_flag1	CS1 空間の HyperBus コントローラおよび HyperRAM の初期化方法を指定します。 HYPERBUS_NO_INIT : 初期化しない HYPERBUS_INIT_AT_LOADER : ロードプロジェクトで初期化 HYPERBUS_INIT_AT_APP : アプリケーションプロジェクトで初期化
	e_hyperbus_maxen_t maxen1	HM_CS1#信号の Low 時間設定を maxen1 で制御するかを設定します。 HYPERBUS_MAXEN_OFF : 設定しない HYPERBUS_MAXEN_ON : 設定する
	uint16_t maxlen1	HM_CS1#信号の最大リードライト処理時間を設定します。 0 : 1 クロックサイクル ... 511 : 512 クロックサイクル
	e_hyperbus_cshi_t rcshi1	HM_CS1#信号がネゲートされてから次のリードアクセス開始までの時間を設定します。 HYPERBUS_CSHI_1P5_CYCLE : 1.5 クロックサイクル ... HYPERBUS_CSHI_16P5_CYCLE : 16.5 クロックサイクル
	e_hyperbus_cshi_t wcshi1	HM_CS1#信号がネゲートされてから次のライトアクセス開始までの時間を設定します。 HYPERBUS_CSHI_1P5_CYCLE : 1.5 クロックサイクル ... HYPERBUS_CSHI_16P5_CYCLE : 16.5 クロックサイクル

表5.15 HyperBus コントローラ初期設定構造体 (st_hyperbus_cfg_t) (2/2)

空間	メンバ名	内容
CS1	e_hyperbus_css_t rcss1	HM_CS1#信号がアサートされてから次のリードアクセス開始までの時間を設定します。 HYPERBUS_CSS_1_CYCLE : 1 クロックサイクル ... HYPERBUS_CSS_16_CYCLE : 16 クロックサイクル
	e_hyperbus_css_t wcsc1	HM_CS1#信号がアサートされてから次のライトアクセス開始までの時間を設定します。 HYPERBUS_CSS_1_CYCLE : 1 クロックサイクル ... HYPERBUS_CSS_16_CYCLE : 16 クロックサイクル
	e_hyperbus_csh_t rcsh1	リードアクセス終了から HM_CS1#信号がネゲートされるまでの時間を設定します。 HYPERBUS_CSH_1_CYCLE : 1 クロックサイクル ... HYPERBUS_CSH_16_CYCLE : 16 クロックサイクル
	e_hyperbus_csh_t wcsh1	ライトアクセス終了から HM_CS1#信号がネゲートされるまでの時間を設定します。 HYPERBUS_CSH_1_CYCLE : 1 クロックサイクル ... HYPERBUS_CSH_16_CYCLE : 16 クロックサイクル
	e_hyperbus_ltcy_t operate_ltcy1	動作時のリードライトレイテンシを設定します。 HYPERBUS_LTCY_5_CYCLE : 5 クロックレイテンシ HYPERBUS_LTCY_6_CYCLE : 6 クロックレイテンシ

5.7 変数一覧

表5.16にローダプログラムで使用する変数一覧を示します。

表5.16 ローダプログラムで使用する変数一覧

変数名	説明	備考
st_hyperbus_cfg_t HYPERBUS_CFG_TABLE[0]	HyperBus コントローラ用の設定テーブルデータ	表6.1参照

HYPERBUS_CFG_TABLE[0]は、アプリケーションプログラムでも HyperRAM にアクセスするためにテーブルデータを使用します。

5.8 関数一覧

サンプルコードは、周辺機能を使用するためのインタフェース関数（API 関数）、ユーザシステムの用途に合わせてユーザで準備が必要なユーザ定義関数（API 関数からコールされる関数）、サンプルコードを動作させるために必要なサンプル関数から構成されています。

表5.17にサンプル関数一覧を、表5.18にAPI関を、表5.19にHyperFlashサンプル関数一覧を、表5.20にHyperRAMサンプル関数一覧を、表5.21にユーザ定義関数一覧を示します。

表5.17 サンプル関数一覧

関数名	概要
reset_handler	リセットハンドラ処理（アセンブラ関数）
INITSCT	プログラムのセクション初期化（アセンブラ関数）
R_SC_HardwareSetup	HyperBus コントローラの初期設定
r_memclk_setup	メモリクロックの設定処理

表5.18 API 関数一覧

関数名	概要
R_HYPERBUS_Setup	HyperBus コントローラと HyperBus に接続されたデバイスの初期設定
R_HYPERBUS_SelectSpace	HyperBus コントローラのアクセス領域の選択
R_CPG_InitialiseHwlf	CPG の初期設定

表5.19 HyperFlash サンプル関数一覧

関数名	概要
HyperFlash_ReadVCR	HyperFlash の Volatile Configuration Register (VCR)のリード関数
HyperFlash_WriteVCR	HyperFlash の Volatile Configuration Register (VCR)へのライト関数
HyperFlash_EraseSect	HyperFlash の指定したセクタのイレース関数
HyperFlash_WriteWord	HyperFlash の指定したアドレスへのライト関数(1 ワード(16 ビット)単位)
HyperFlash_ReadROMInfo	HyperFlash の Device ID および Common Flash Interface(CFI)情報のリード関数

表5.20 HyperRAM サンプル関数一覧

関数名	概要
HyperRAM_ReadID0	HyperRAM の Identification Register 0 (ID0)のリード関数
HyperRAM_ReadID1	HyperRAM の Identification Register 1 (ID1)のリード関数
HyperRAM_ReadCR0	HyperRAM の Configuration Register 0 (CR0)のリード関数
HyperRAM_WriteCR0	HyperRAM の Configuration Register 0 (CR0)へのライト関数
HyperRAM_ReadCR1	HyperRAM の Configuration Register 1 (CR1)のリード関数
HyperRAM_WriteCR1	HyperRAM の Configuration Register 1 (CR1)へのライト関数

表5.21 ユーザ定義関数一覧

関数名	概要
Userdef_PreHardwareSetup	HyperBus コントローラ初期化前に必要なハードウェア初期化处理
Userdef_PostHardwareSetup	HyperBus コントローラ初期化後に必要なハードウェア初期化处理

5.9 関数仕様

サンプルコードの関数仕様を示します。

reset_handler	
概 要	リセットハンドラ処理
宣 言	reset_handler
説 明	ローダプログラムおよびアプリケーションプログラムのエントリ関数です。 ローダプログラムでは HyperFlash に対して高速なアクセスができるように設定処理を行います。 アプリケーションプログラムでは HyperRAM に対して高速なアクセスができるように設定処理を行います。
引 数	なし
リターン値	なし

INITSCT	
概 要	プログラムのセクション初期化
宣 言	void INITSCT(void)
説 明	RAM 領域に転送する必要がある初期値ありデータ（RAM 領域で実行する必要があるコードおよび定数データを含む）を ROM 領域から転送し、RAM 領域の初期値なしデータの初期化を行います。
引 数	p_dtbl : 初期値ありデータのセクション情報が格納された領域へのポインタ p_btbl : 初期値なしデータのセクション情報が格納された領域へのポインタ
リターン値	なし

R_SC_HardwareSetup	
概 要	HyperBus コントローラの初期設定
宣 言	void R_SC_HardwareSetup(void)
説 明	ローダプログラムでは、使用する HyperFlash に最適な設定を行います。 — CPU および周辺クロックの設定 — HyperFlash とのアクセスを高速に行えるように HyperBus コントローラおよび HyperFlash のレジスタを設定 アプリケーションプログラムでは、使用する HyperRAM に最適な設定を行います。 — HyperRAM とのアクセスを高速に行えるように HyperBus コントローラおよび HyperRAM のレジスタを設定
引 数	なし
リターン値	なし
注意事項	本関数は、HyperFlash および HyperRAM に配置して実行することができません。 本関数は、HyperFlash および HyperRAM 以外の領域に配置する必要があります。

r_memclk_setup	
概 要	メモリクロックの設定処理
宣 言	void r_memclk_setup (void)
説 明	R_SC_HardwareSetup 関数の実行前に、メモリクロックの設定を行います。 HyperFlash ブートの場合、ここではメモリクロックの設定は行いません。
引 数	なし
リターン値	なし
注意事項	本関数は、HyperFlash に配置して実行することができません。 本関数は、HyperFlash 以外の領域に配置する必要があります。

R_HYPERBUS_Setup	
概 要	HyperBus コントローラと HyperBus に接続されたデバイスの初期設定
宣 言	void R_HYPERBUS_Setup (void)
説 明	ローダプログラムでは HyperBus コントローラおよび HyperFlash のレジスタの初期設定を行います。 アプリケーションプログラムでは HyperBus コントローラおよび HyperRAM のレジスタの初期設定を行います。
引 数	なし
リターン値	なし
注意事項	本関数は、HyperFlash および HyperRAM に配置して実行することができません。 本関数は、HyperFlash および HyperRAM 以外の領域に配置する必要があります。

R_HYPERBUS_SelectSpace	
概 要	HyperBus コントローラのアクセス領域の選択
宣 言	int_t R_HYPERBUS_SelectSpace(e_hyperbus_access_area_t area, e_hyperbus_space_select_t space)
説 明	HyperBus コントローラのアクセス領域を選択します。HyperRAM のレジスタをアクセスする場合は、引数 space に HYPERBUS_REGISTER_SPACE を指定して本関数をコールしてください。また、HyperRAM のメモリ領域をアクセスする場合は、引数 space に HYPERBUS_MEMORY_SPACE を指定してください。
引 数	e_hyperbus_access_area_t area : HyperBus のアクセスする空間の指定 HYPERBUS_CS0_AREA (使用禁止) HYPERBUS_CS1_AREA e_hyperbus_space_select_t space : HyperRAM のアクセス対象領域の指定 HYPERBUS_MEMORY_SPACE HYPERBUS_REGISTER_SPACE
リターン値	DRV_SUCCESS : 正常終了 DRV_ERROR : エラー終了
注意事項	本関数は、HyperRAM に配置して実行することができません。 本関数は、HyperRAM 以外の領域に配置する必要があります。 本関数は、HyperRAM 空間に対してのみ設定が可能です。

```
R CPG InitialiseHwlf
```

概要	CPG の初期設定
宣言	int_t R_CPG_InitialiseHwlf(void)
説明	<p>r_cpg_drv_sc_cfg.h の CPG コンフィグレーションデータを使用して、CPG レジスタ（FRQCR、CKIOSEL、SCLKSEL）の設定処理を行います。</p> <p>本サンプルコードでは、「表5.2 ブート起動用内蔵ROMプログラムおよびローダプログラムの設定内容」の「動作クロックの設定」および「HYPCLK の選択」に記載されている動作周波数となるように CPG の設定を行います。</p>
引数	なし なし
リターン値	<p>DRV_SUCCESS : 正常終了</p> <p>DRV_ERROR : r_cpg_drv_sc_cfg.h の CPG コンフィグレーションデータが不正</p>
注意事項	本関数は、大容量内蔵 RAM に配置する必要があります。

HyperFlash ReadVCR

概 要	HyperFlash の Volatile Configuration Register (VCR) のリード関数
宣 言	uint16_t HyperFlash_ReadVCR(uint32_t baddr)
説 明	HyperFlash の Volatile Configuration Register をリードし、リードした値を戻り値として返します。
引 数	uint32_t baddr HyperFlash 空間のベースアドレス
リターン値	Volatile Configuration Register をリードした値
注意事項	<p>本関数は、HyperFlash に配置して実行することができません。</p> <p>本関数は、HyperFlash 以外の領域に配置する必要があります。</p> <p>MMU が有効の状態で本関数をコールする場合は、引数 baddr にはメモリ属性としてストロングリオーダ属性またはデバイス属性を設定した HyperFlash 空間のアドレスを指定してください。</p>

HyperFlash WriteVCR

概要	HyperFlash の Volatile Configuration Register (VCR) へのライト関数
宣言	int_t HyperFlash_WriteVCR (uint32_t baddr, uint16_t wdata)
説明	引数 wdata で指定した値を HyperFlash の Volatile Configuration Register に書き込みを行います。
引数	uint32_t baddr HyperFlash 空間のベースアドレス uint16_t wdata Volatile Configuration Register にライトする値
リターン値	DRV_SUCCESS : 正常終了
注意事項	本関数は、HyperFlash に配置して実行することができません。 本関数は、HyperFlash 以外の領域に配置する必要があります。 MMU が有効の状態で本関数をコールする場合は、引数 baddr にはメモリ属性としてストロングリオーダ属性またはデバイス属性を設定した HyperFlash 空間のアドレスを指定してください。

HyperFlash_EraseSect	
概 要	HyperFlash の指定したセクタのイレース関数
宣 言	int_t HyperFlash_EraseSect (uint32_t baddr, uint32_t saddr)
説 明	引数 saddr で指定した HyperFlash のセクタを消去します。
引 数	uint32_t baddr HyperFlash 空間のベースアドレス uint32_t saddr イレースする HyperFlash のセクタアドレス
リターン値	DRV_SUCCESS : 正常終了 DRV_ERROR : エラー終了
注意事項	本関数は、HyperFlash に配置して実行することができません。 本関数は、HyperFlash 以外の領域に配置する必要があります。 MMU が有効の状態では本関数をコールする場合は、引数 baddr にはメモリ属性としてストロングリオーダ属性またはデバイス属性を設定した HyperFlash 空間のアドレスを指定してください。

HyperFlash_WriteWord	
概 要	HyperFlash の指定したアドレスへのライト関数(1 ワード(16 ビット)単位)
宣 言	uint16_t HyperFlash_WriteWord (uint32_t baddr, uint32_t waddr, uint16_t wdata)
説 明	引数 wdata で指定した値を、引数 waddr で指定した HyperFlash のアドレスへ書き込みを行います。
引 数	uint32_t baddr HyperFlash 空間のベースアドレス uint32_t waddr 書き込む HyperFlash のアドレス uint16_t wdata 書き込みデータ
リターン値	DRV_SUCCESS : 正常終了
注意事項	本関数は、HyperFlash に配置して実行することができません。 本関数は、HyperFlash 以外の領域に配置する必要があります。 MMU が有効の状態では本関数をコールする場合は、引数 baddr にはメモリ属性としてストロングリオーダ属性またはデバイス属性を設定した HyperFlash 空間のアドレスを指定してください。

HyperFlash_ReadROMInfo	
概 要	HyperFlash の Device ID および Common Flash Interface(CFI)情報のリード関数
宣 言	void HyperFlash_ReadROMInfo (uint32_t baddr, uint16_t *idbuf)
説 明	HyperFlash の Device ID および CFI 情報を読み出します。
引 数	uint32_t baddr HyperFlash 空間のベースアドレス Uint16_t *idbuf Device ID および CFI 情報を格納するバッファ(16 ビット×256 ワード)の先頭アドレス
リターン値	なし
注意事項	本関数は、HyperFlash に配置して実行することができません。 本関数は、HyperFlash 以外の領域に配置する必要があります。 MMU が有効の状態では本関数をコールする場合は、引数 baddr にはメモリ属性としてストロングリオーダ属性またはデバイス属性を設定した HyperFlash 空間のアドレスを指定してください。 Device ID および CFI 情報の詳細は、使用する HyperFlash のデータシートを参照してください。

HyperRAM_ReadID0	
概 要	HyperRAM の Identification Register 0 (ID0)のリード関数
宣 言	uint16_t HyperRAM_ReadID0(uint32_t baddr)
説 明	HyperRAM の Identification Register 0 をリードし、リードした値を戻り値として返します。
引 数	uint32_t baddr HyperRAM 空間のベースアドレス
リターン値	Identification Register 0 をリードした値
注意事項	本関数は、HyperRAM に配置して実行することができません。 本関数は、HyperRAM 以外の領域に配置する必要があります。 MMU が有効の状態では本関数をコールする場合は、引数 baddr にはメモリ属性としてストロングリオーダ属性またはデバイス属性を設定した HyperRAM 空間のアドレスを指定してください。

HyperRAM_ReadID1	
概 要	HyperRAM の Identification Register 1 (ID1)のリード関数
宣 言	uint16_t HyperRAM_ReadID1(uint32_t baddr)
説 明	HyperRAM の Identification Register 1 をリードし、リードした値を戻り値として返します。
引 数	uint32_t baddr HyperRAM 空間のベースアドレス
リターン値	Identification Register 1 をリードした値
注意事項	本関数は、HyperRAM に配置して実行することができません。 本関数は、HyperRAM 以外の領域に配置する必要があります。 MMU が有効の状態では本関数をコールする場合は、引数 baddr にはメモリ属性としてストロングリオーダ属性またはデバイス属性を設定した HyperRAM 空間のアドレスを指定してください。

HyperRAM_ReadCR0	
概 要	HyperRAM の Configuration Register 0 (CR0)のリード関数
宣 言	uint16_t HyperRAM_ReadCR0(uint32_t baddr)
説 明	HyperRAM の Configuration Register 0 をリードし、リードした値を戻り値として返します。
引 数	uint32_t baddr HyperRAM 空間のベースアドレス
リターン値	Configuration Register 0 をリードした値
注意事項	本関数は、HyperRAM に配置して実行することができません。 本関数は、HyperRAM 以外の領域に配置する必要があります。 MMU が有効の状態では本関数をコールする場合は、引数 baddr にはメモリ属性としてストロングリオーダ属性またはデバイス属性を設定した HyperRAM 空間のアドレスを指定してください。

HyperRAM_WriteCR0	
概 要	HyperRAM の Configuration Register 0 (CR0)へのライト関数
宣 言	int_t HyperRAM_WriteCR0 (uint32_t baddr, uint16_t wdata)
説 明	引数 wdata で指定したデータを HyperRAM の Configuration Register 0 へライトします。
引 数	uint32_t baddr HyperRAM 空間のベースアドレス uint16_t wdata Configuration Register 0 への書き込みデータ
リターン値	DRV_SUCCESS : 正常終了
注意事項	本関数は、HyperRAM に配置して実行することができません。 本関数は、HyperRAM 以外の領域に配置する必要があります。 MMU が有効の状態では本関数をコールする場合は、引数 baddr にはメモリ属性としてストロングリオーダ属性またはデバイス属性を設定した HyperRAM 空間のアドレスを指定してください。
HyperRAM_ReadCR1	
概 要	HyperRAM の Configuration Register 1 (CR1)のリード関数
宣 言	uint16_t HyperRAM_ReadCR1(uint32_t baddr)
説 明	HyperRAM の Configuration Register 1 をリードし、リードした値を戻り値として返します。
引 数	uint32_t baddr HyperRAM 空間のベースアドレス
リターン値	Configuration Register 1 をリードした値
注意事項	本関数は、HyperRAM に配置して実行することができません。 本関数は、HyperRAM 以外の領域に配置する必要があります。 MMU が有効の状態では本関数をコールする場合は、引数 baddr にはメモリ属性としてストロングリオーダ属性またはデバイス属性を設定した HyperRAM 空間のアドレスを指定してください。
HyperRAM_WriteCR1	
概 要	HyperRAM の Configuration Register 1 (CR1)へのライト関数
宣 言	int_t HyperRAM_WriteCR1 (uint32_t baddr, uint16_t wdata)
説 明	引数 wdata で指定したデータを HyperRAM の Configuration Register 1 へライトします。
引 数	uint32_t baddr HyperRAM 空間のベースアドレス uint16_t wdata Configuration Register 1 への書き込みデータ
リターン値	DRV_SUCCESS : 正常終了
注意事項	本関数は、HyperRAM に配置して実行することができません。 本関数は、HyperRAM 以外の領域に配置する必要があります。 MMU が有効の状態では本関数をコールする場合は、引数 baddr にはメモリ属性としてストロングリオーダ属性またはデバイス属性を設定した HyperRAM 空間のアドレスを指定してください。

Userdef_PreHardwareSetup	
概 要	HyperBus コントローラ初期化前に必要なハードウェア初期化処理
宣 言	void Userdef_PreHardwareSetup (void)
説 明	HyperBus コントローラの初期化前に実行する必要がある、ハードウェア初期化処理を記述するためのユーザ定義関数です。R_SC_HardwareSetup 関数の先頭でコールされます。 ローダプログラムでは、CPG の初期設定関数をコールします。 アプリケーションプログラムでは、IOKEEP ビットのクリアを行い、ディープスタンバイから復帰した際に保持されている端子の状態を解除します。
引 数	なし
リターン値	なし
注意事項	本関数に HyperFlash および HyperRAM で実行することができない処理を記述する場合、本関数は HyperFlash および HyperRAM 以外の領域に配置する必要があります。

Userdef_PostHardwareSetup	
概 要	HyperBus コントローラ初期化後に必要なハードウェア初期化処理
宣 言	void Userdef_PostHardwareSetup (void)
説 明	HyperBus コントローラの初期化後に実行する必要がある、ハードウェア初期化処理を記述するためのユーザ定義関数です。R_SC_HardwareSetup 関数の終端でコールされます。 ローダプログラムでは、何も行いません。 アプリケーションプログラムでは、保持用内蔵 RAM 各ページのライトイネーブルをライト有効に設定します。
引 数	なし
リターン値	なし
注意事項	本関数に HyperFlash および HyperRAM で実行することができない処理を記述する場合、本関数は HyperFlash および HyperRAM 以外の領域に配置する必要があります。

5.10 ロードプログラムのフローチャート

5.10.1 ロードプログラム全体

図5.6にロードプログラム（全体）のフローチャートを示します。

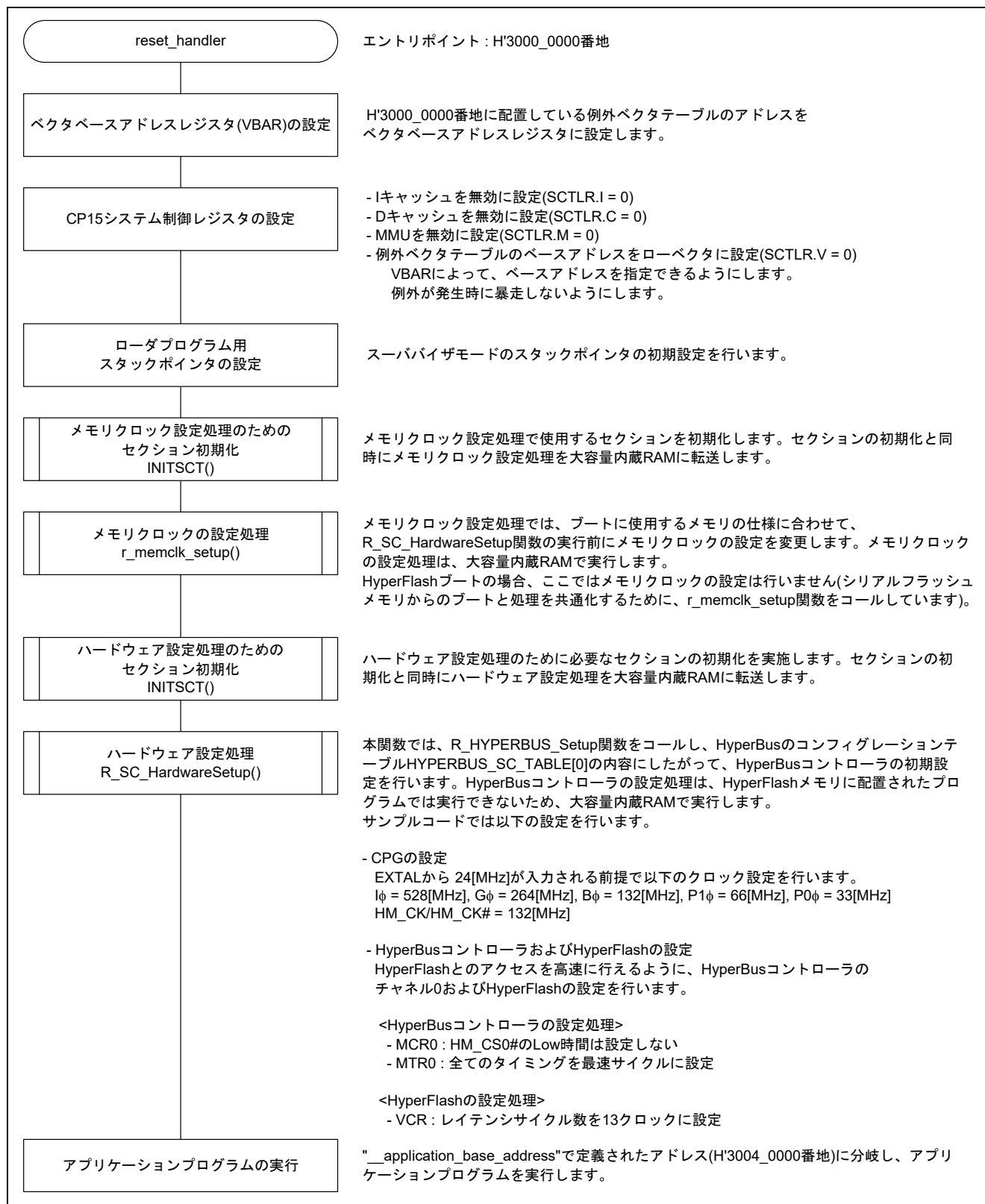


図5.6 ロードプログラム（全体）のフローチャート

5.10.2 R_SC_HardwareSetup 関数

ハードウェアの初期設定を行うための関数です。ローダプログラムでは、高速に HyperFlash にアクセスができるように、HyperBus コントローラおよび HyperFlash の設定を行います（HyperFlash に供給するクロック（HM_CK/HM_CK#）の設定も行います）。

HyperBus コントローラのレジスタおよび HyperFlash のレジスタを設定する処理は、HyperFlash 空間に配置されたプログラムでは実行できないため、大容量内蔵 RAM に展開し、大容量内蔵 RAM で実行します。

図5.7にR_SC_HardwareSetup（ローダプログラム）のフローチャートを示します。

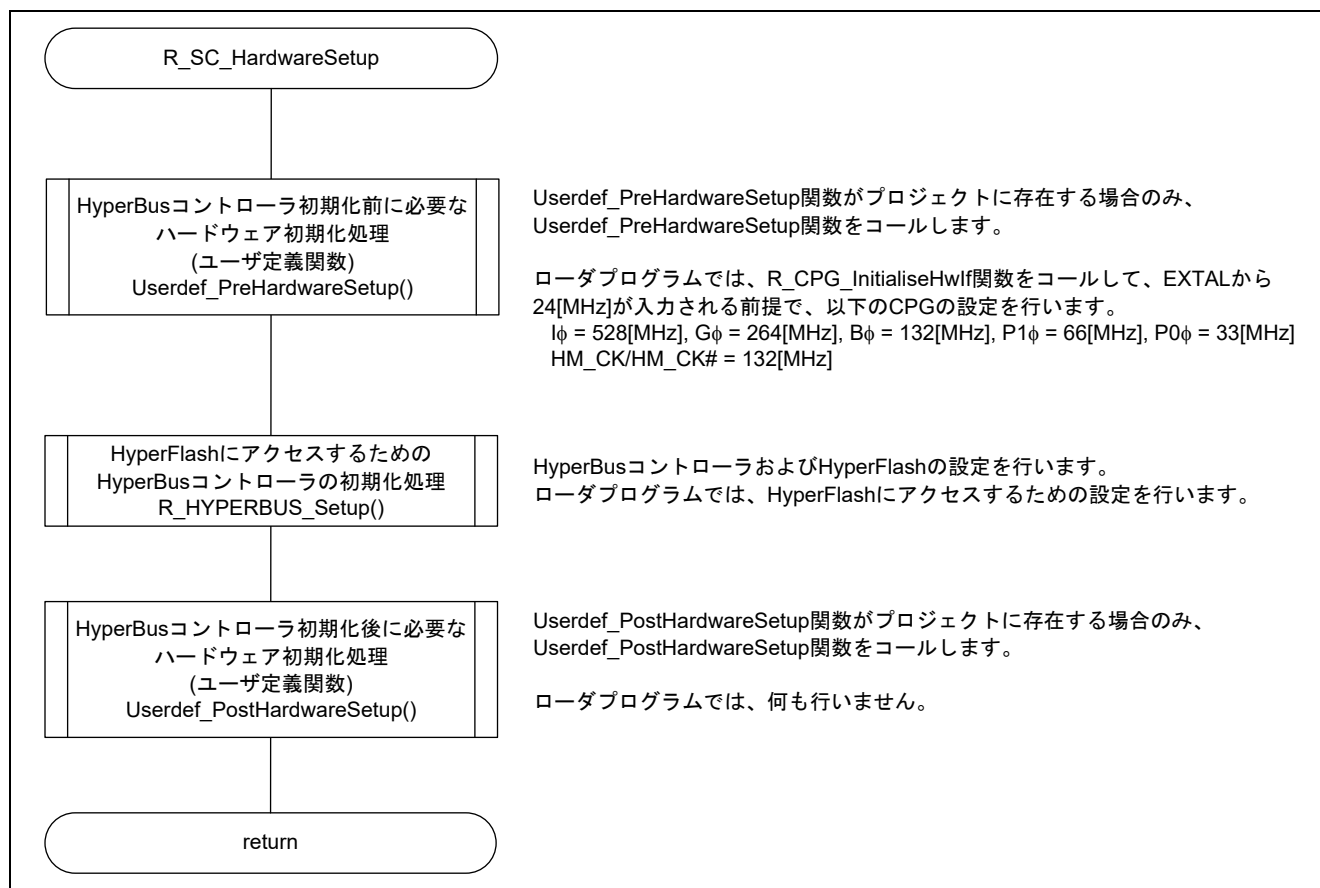


図5.7 R_SC_HardwareSetup（ローダプログラム）のフローチャート

5.10.3 HyperBus コントローラの初期設定

図5.8にR_HYPERBUS_Setup（ローダプログラム）のフローチャートを示します。

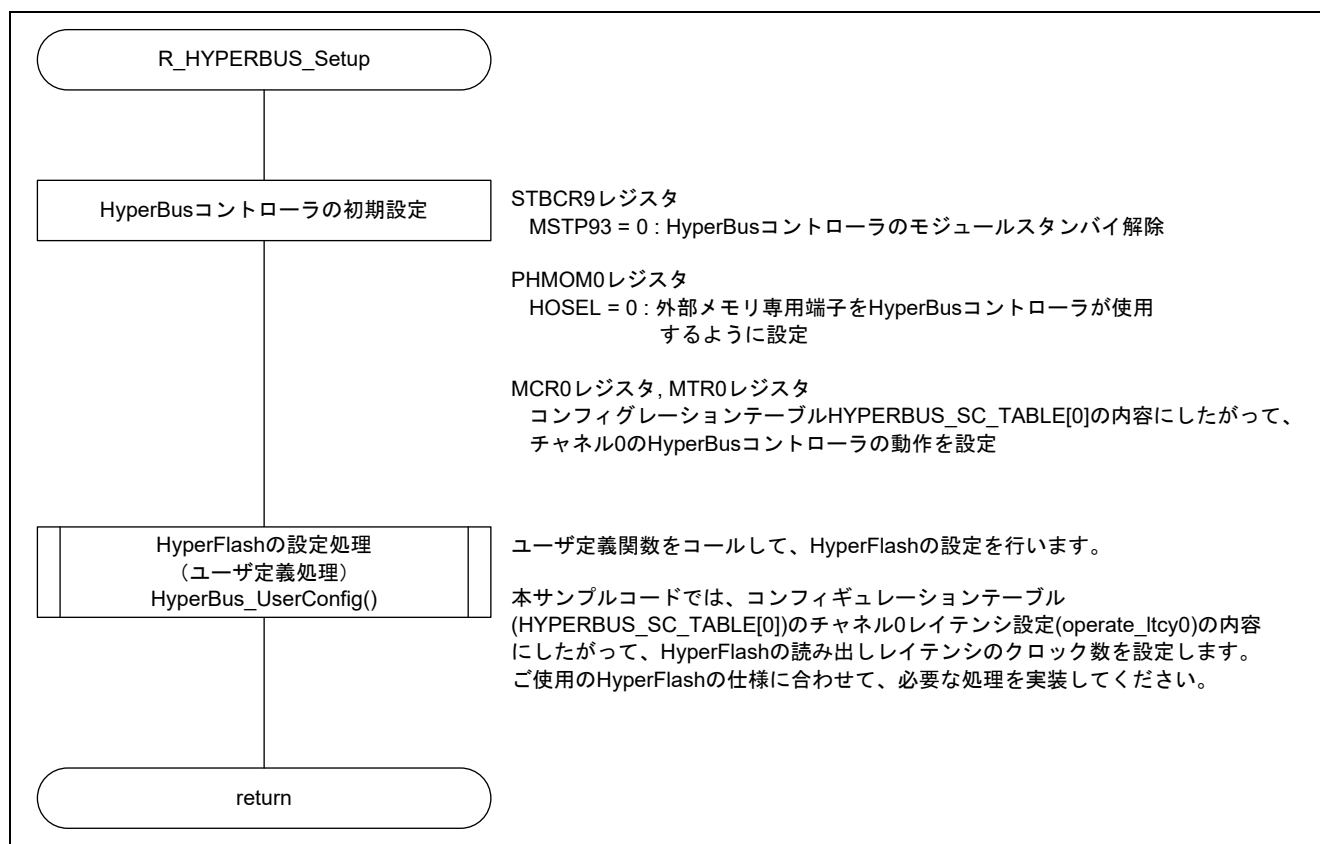


図5.8 R_HYPERBUS_Setup（ローダプログラム）のフローチャート

5.10.4 HyperFlash の設定

図5.9にHyperBus_UserConfig関数（ローダプログラム）のフローチャートを示します。本関数はユーザ定義関数です。ご使用の HyperFlash の仕様に合わせて必要な処理を実装してください。

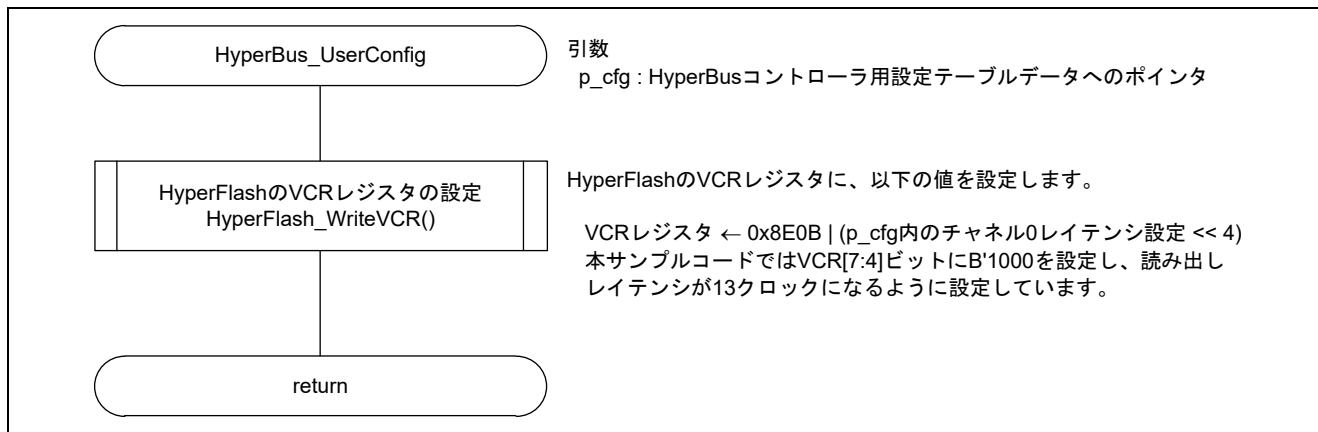


図5.9 HyperBus_UserConfig 関数（ローダプログラム）のフローチャート

5.11 アプリケーションプログラムのフローチャート

5.11.1 アプリケーションプログラム全体

アプリケーションプログラムの詳細については、「RZ/A2M グループ 初期設定例」のアプリケーションノートをご参照ください。

本アプリケーションプログラムでは、上記に加え、HyperRAM の初期設定および HyperRAM のキャッシュ無効領域へのライトを行います。

5.11.2 R_SC_HardwareSetup 関数

ハードウェアの初期設定を行うための関数です。アプリケーションプログラムでは HyperBus コントローラに接続された HyperRAM にアクセスできるように、HyperBus コントローラおよび HyperRAM の設定を行います。

本関数は、大容量内蔵 RAM に展開し、大容量内蔵 RAM で実行します。

図5.10にR_SC_HardwareSetup（アプリケーションプログラム）のフローチャートを示します。

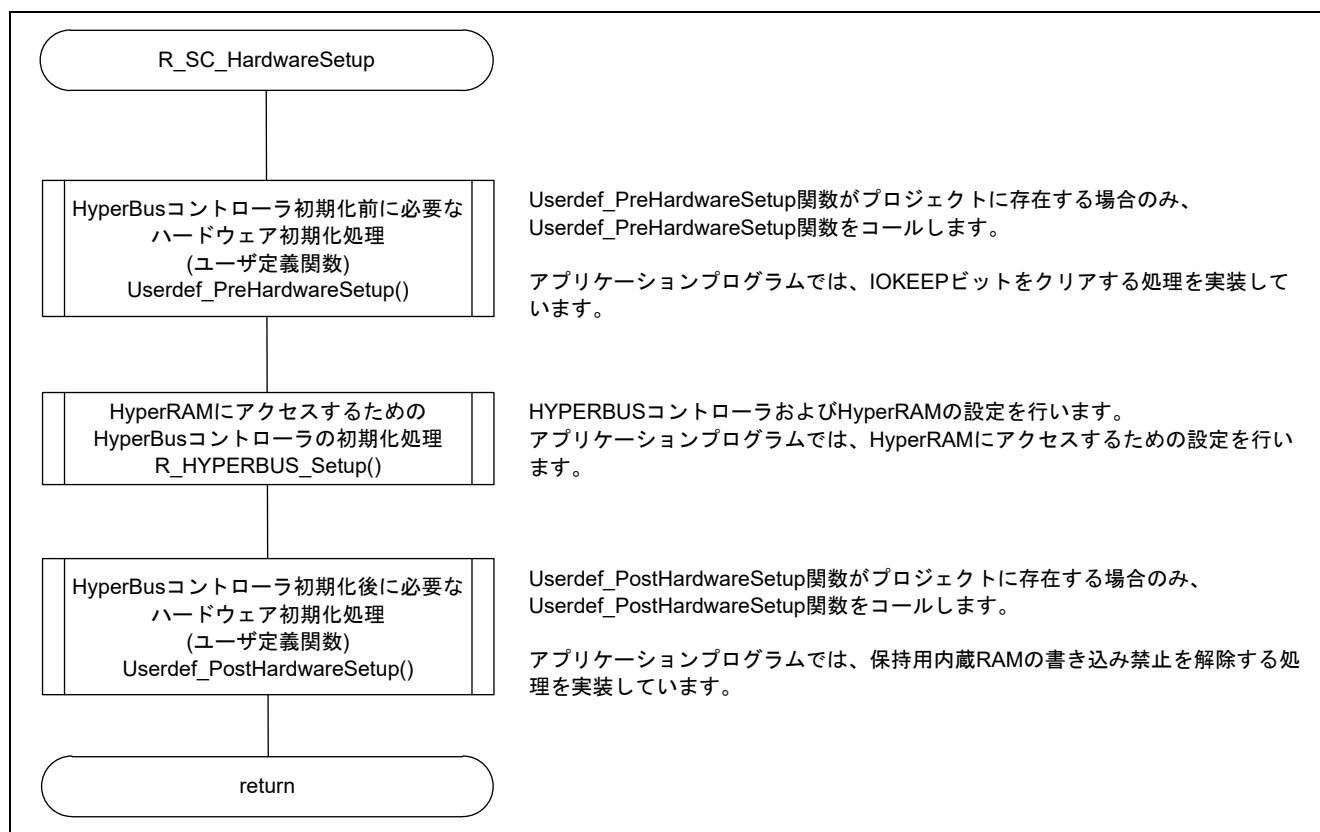


図5.10 R_SC_HardwareSetup（アプリケーションプログラム）のフローチャート

5.11.3 HyperBus コントローラの初期設定

図5.11にR_HYPERBUS_Setup（アプリケーションプログラム）のフローチャートを示します。

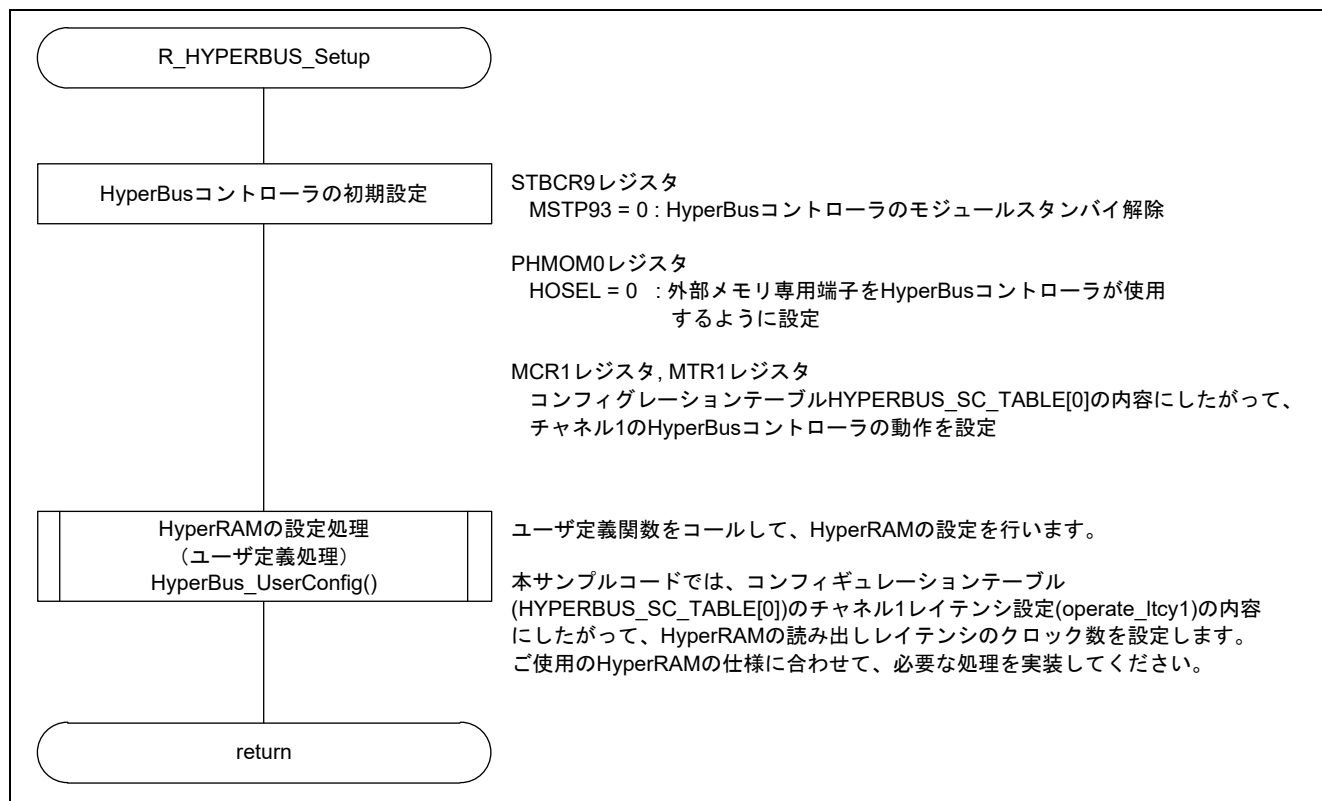


図5.11 R_HYPERBUS_Setup（アプリケーションプログラム）のフローチャート

5.11.4 HyperRAM の初期設定

図5.12にHyperBus_UserConfig関数（アプリケーションプログラム）のフローチャートを示します。本関数はユーザ定義関数です。ご使用の HyperRAM の仕様にあわせて必要な処理を実装してください。

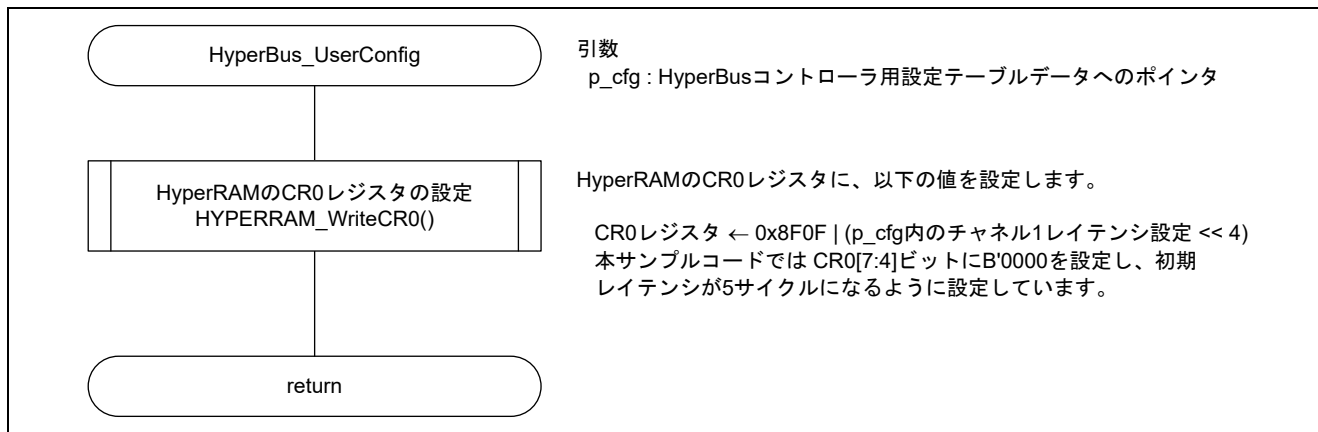


図5.12 HyperBus_UserConfig 関数（アプリケーションプログラム）のフローチャート

5.12 HyperFlash および HyperRAM をアクセスするための関数のフローチャート

5.12.1 HyperFlash の Volatile Configuration Register のリード関数

図5.13にHyperFlashのVolatile Configuration Register(VCR)へのリード関数のフローチャートを示します。

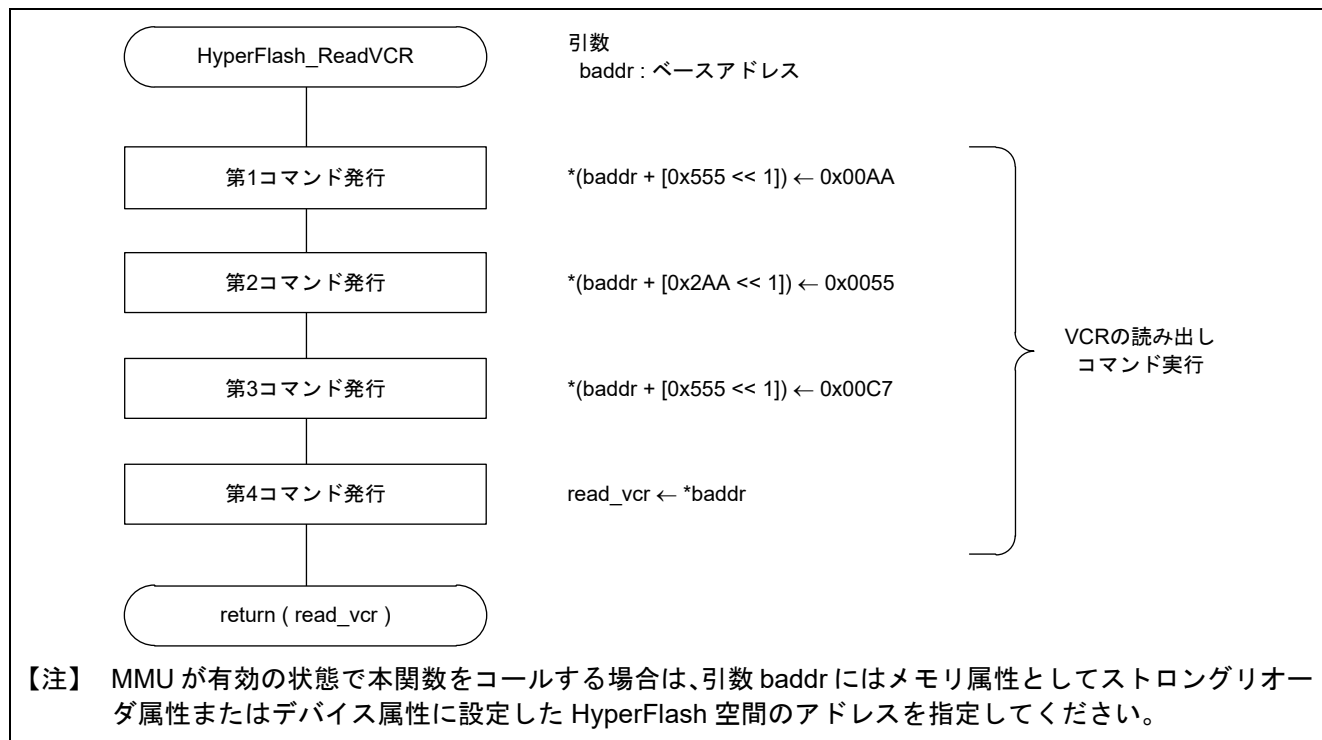


図5.13 HyperFlash の Volatile Configuration Register(VCR)へのリード関数

5.12.2 HyperFlash の Volatile Configuration Register (VCR)へのライト関数

図5.14にHyperFlashのVolatile Configuration Register(VCR)へのライト関数のフローチャートを示します。

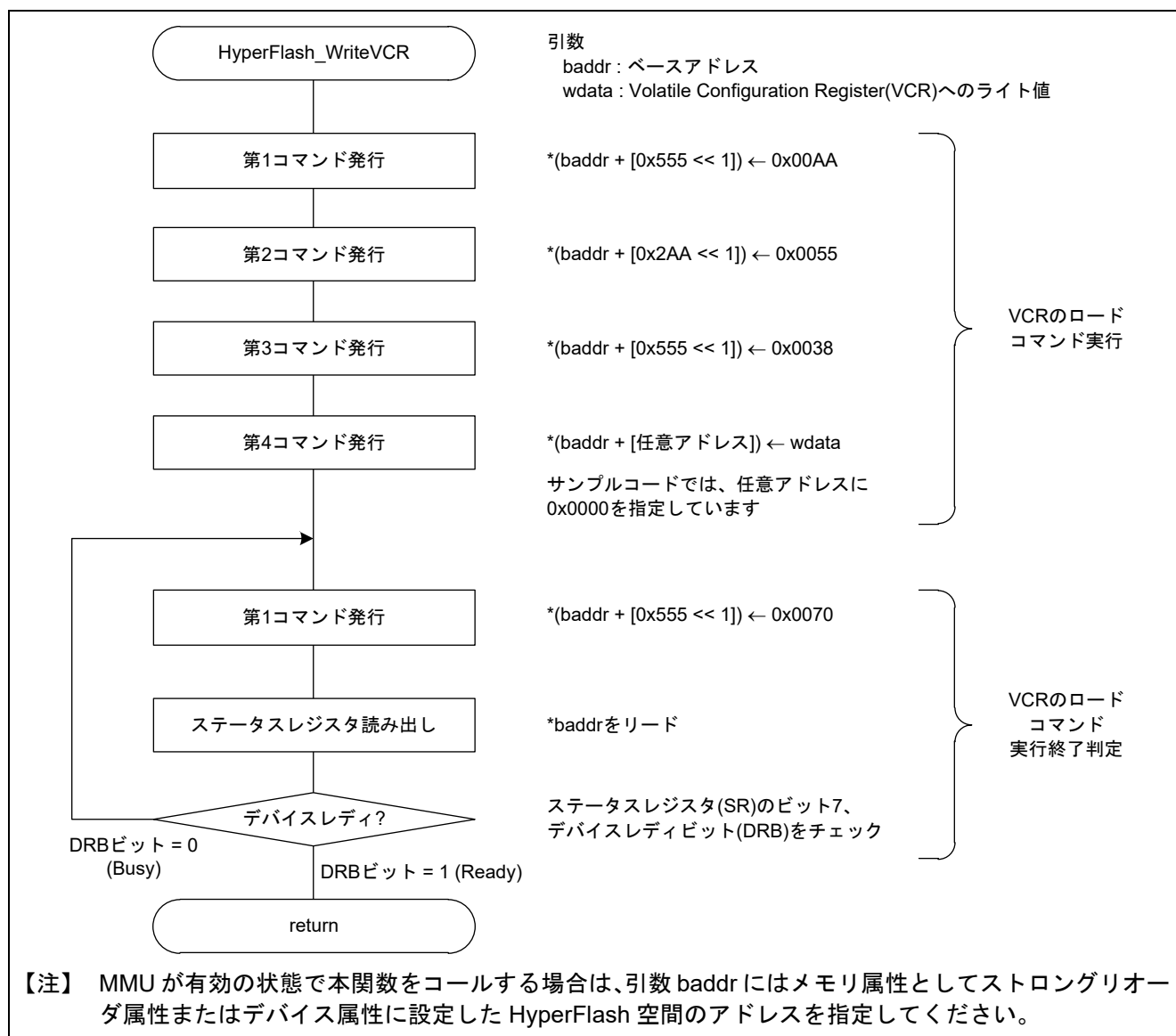


図5.14 HyperFlash の Volatile Configuration Register(VCR)へのライト関数

5.12.3 HyperRAM の Configuration Register 0 (CR0)のリード関数

図5.15にHyperRAMのConfiguration Register 0 (CR0)のリード関数のフローチャートを示します。

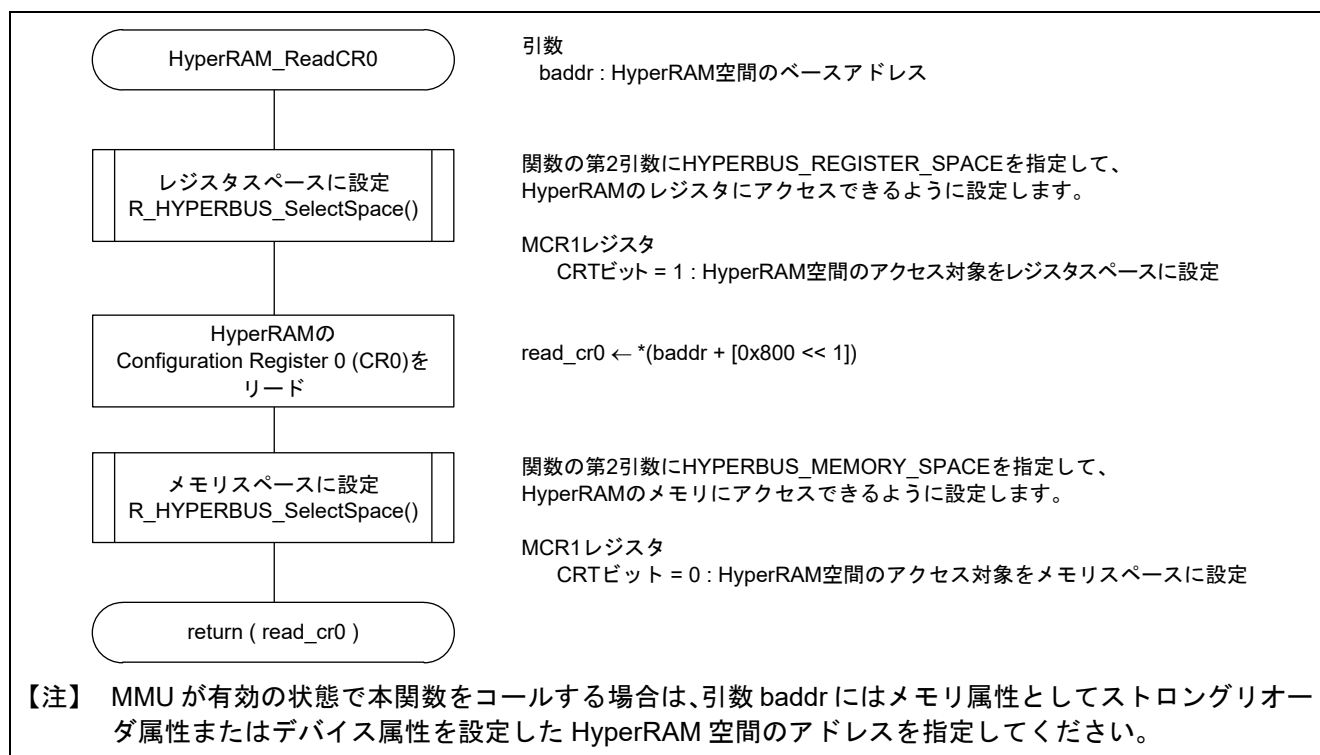


図5.15 HyperRAM の Configuration Register 0 (CR0)のリード関数

5.12.4 HyperRAM の Configuration Register 0 (CR0)へのライト関数

図5.16にHyperRAMのConfiguration Register 0 (CR0)へのライト関数のフローチャートを示します。

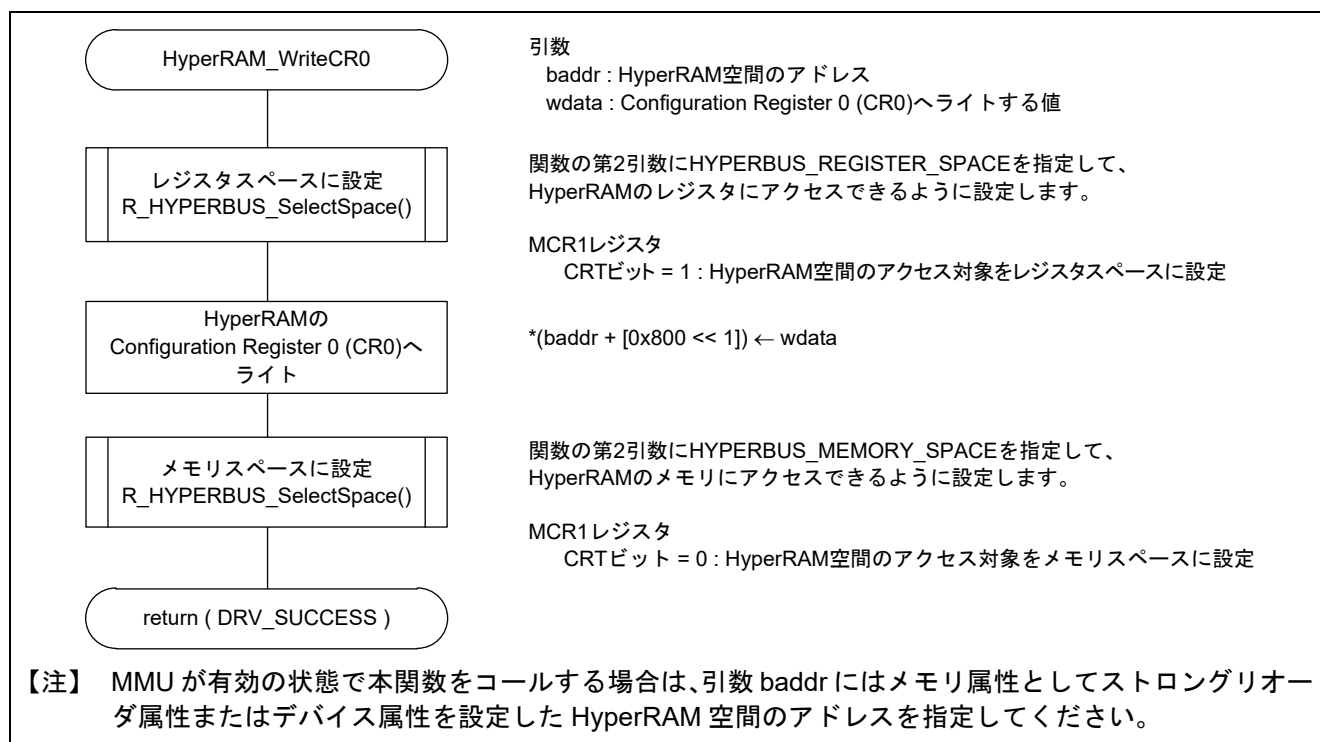


図5.16 HyperRAM の Configuration Register 0 (CR0)へのライト関数

5.12.5 HyperRAM をアクセスする際の領域（メモリまたはレジスタ）の指定関数

図5.17にHyperRAMをアクセスする際の領域（メモリまたはレジスタ）の指定関数のフローチャートを示します。

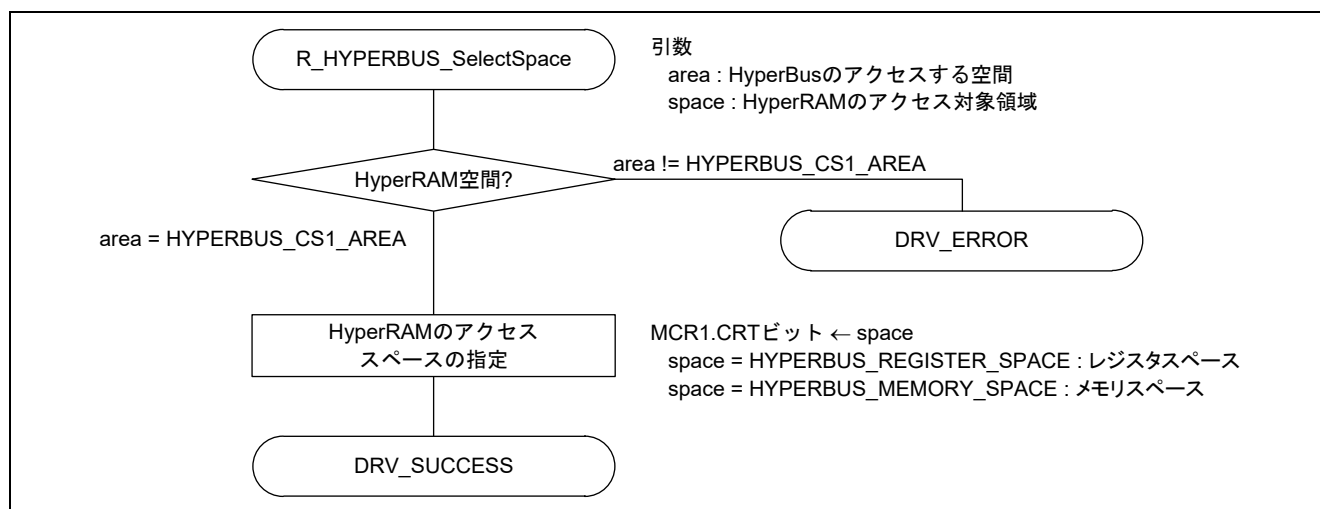


図5.17 HyperRAM をアクセスする際の領域（メモリまたはレジスタ）の指定関数

6. 応用例

6.1 HyperBus コントローラの設定

HyperBus コントローラ用の設定テーブルデータの内容を参照して、HyperBus コントローラに設定します。表6.1に、サンプルコードのHyperBusコントローラ用の設定テーブルデータ（HYPERBUS_SC_TABLE[0]）の設定値を示します。Hyperbus コントローラの初期設定関数（R_HYPERBUS_Setup）がこのテーブルデータを用いて、Hyperbus コントローラの動作タイミングに関する CS0 メモリ設定レジスタ(MCR0)、CS1 メモリ設定レジスタ 1(MCR1)、CS0 メモリタイミングレジスタ(MTR0)、および CS1 メモリタイミングレジスタ (MTR1)の設定を行います。

ご使用の HyperFlash または HyperRAM の仕様に合わせて設定してください。

表6.1 HyperBus コントローラ用の設定テーブルデータ（HYPERBUS_SC_TABLE[0]）の設定値

空間	メンバ	設定値	対応レジスタ
HyperFlash	e_hyperbus_init_control init_flag0	HYPERBUS_INIT_AT_LOADER : ローダプログラムで初期化	—
	e_hyperbus_maxen_t maxen0	HYPERBUS_MAXEN_OFF : 設定しない	MCR0.MAXEN
	uint16_t maxlen0	0 : 1 クロックサイクル	MCR0.MAXLEN[8:0]
	e_hyperbus_cshi_t rcshi0	HYPERBUS_CSHI_1_5 : 1.5 クロックサイクル	MTR0.RCSHI[3:0]
	e_hyperbus_cshi_t wcshi0	HYPERBUS_CSHI_1_5 : 1.5 クロックサイクル	MTR0.WCSHI[3:0]
	e_hyperbus_css_t rcss0	HYPERBUS_CSS_1 : 1 クロックサイクル	MTR0.RCSS[3:0]
	e_hyperbus_css_t wcss0	HYPERBUS_CSS_1 : 1 クロックサイクル	MTR0.WCSS[3:0]
	e_hyperbus_csh_t rcsh0	HYPERBUS_CSH_1 : 1 クロックサイクル	MTR0.RCSH[3:0]
	e_hyperbus_csh_t wcsh0	HYPERBUS_CSH_1 : 1 クロックサイクル	MTR0.WCSH[3:0]
	e_hyperbus_ltcy_t operate_ltcy0	HYPERBUS_LTCY_13 : 13 クロックレイテンシ	—
HyperRAM	e_hyperbus_init_control init_flag1	HYPERBUS_INIT_AT_APP : アプリケーションプログラムで初期化	—
	e_hyperbus_maxen_t maxen1	HYPERBUS_MAXEN_OFF : 設定しない	MCR1.MAXEN
	uint16_t maxlen1	0 : 1 クロックサイクル	MCR1.MAXLEN[8:0]
	e_hyperbus_cshi_t rcshi1	HYPERBUS_CSHI_1_5 : 1.5 クロックサイクル	MTR1.RCSHI[3:0]
	e_hyperbus_cshi_t wcshi1	HYPERBUS_CSHI_1_5 : 1.5 クロックサイクル	MTR1.WCSHI[3:0]
	e_hyperbus_css_t rcss1	HYPERBUS_CSS_1 : 1 クロックサイクル	MTR1.RCSS[3:0]
	e_hyperbus_css_t wcss1	HYPERBUS_CSS_1 : 1 クロックサイクル	MTR1.WCSS[3:0]
	e_hyperbus_csh_t rcsh1	HYPERBUS_CSH_1 : 1 クロックサイクル	MTR1.RCSH[3:0]
	e_hyperbus_csh_t wcsh1	HYPERBUS_CSH_1 : 1 クロックサイクル	MTR1.WCSH[3:0]
	e_hyperbus_ltcy_t operate_ltcy1	HYPERBUS_LTCY_5 : 5 クロックレイテンシ	MTR1.LTCY[3:0]

6.2 HyperFlash のレイテンシロックの設定

表6.2にRZ/A2M CPUボードに搭載しているHyperFlashのレイテンシロックと最大動作周波数の関係を示します。サンプルコードでは、132MHz 時に最小のレイテンシとなるように、ローダプログラムの処理で Volatile Configuration Register (VCR) の xVCR[7:4]ビットの設定値を変更して HyperFlash のレイテンシを 13 サイクルに設定しています。

ご使用の HyperFlash の仕様に合わせて、HyperBus コントローラおよび HyperFlash にレイテンシロックを設定してください。

表6.2 HyperFlash のレイテンシロックと最大動作周波数

レイテンシコード xVCR[7:4]	レイテンシロック	最大動作周波数(MHz)
0000	5	52
0001	6	62
0010	7	72
0011	8	83
0100	9	93
0101	10	104
0110	11	114
0111	12	125
1000	13	135
1001	14	145
1010	15	156
1011	16	166(工場出荷時)
1110~1111	予約済み	該当なし

【注】 表中のレイテンシコードは、HyperFlash の VCR/NVCR のビット 7~4 への設定値です。

6.3 HyperRAM のレイテンシロックの設定

表6.3にRZ/A2M CPUボードに搭載しているHyperRAMのレイテンシロックと最大動作周波数の関係を示します。サンプルコードでは、132MHz 時に最小のレイテンシとなるように、アプリケーションプログラムの処理で CR0 の[7:4]ビットの設定値を変更して HyperRAM のレイテンシを 5 サイクルに設定しています。

ご使用の HyperRAM の仕様に合わせて、HyperBus コントローラおよび HyperRAM にレイテンシサイクルを設定してください。

表6.3 HyperRAM のレイテンシロックと最大動作周波数

レイテンシコード CR0 の[7:4]	レイテンシロック	最大動作周波数(MHz)
0000	5	133
0001	6	166(工場出荷時)
0010~1101	予約済み	該当なし
1110	3	83
1111	4	100

【注】 表中のレイテンシコードは、HyperRAM のレジスタベースにあるコンフィギュレーションレジスタ 0(CR0)のビット 7~4 への設定値です。

7. サンプルコードの注意事項

7.1 HyperFlash および HyperRAM のレジスタアクセス

MMU が有効の状態、HyperBus コントローラに接続されたデバイス（HyperFlash および HyperRAM）のレジスタにアクセスする場合は、メモリ属性としてストロングリオーダ属性またはデバイス属性に設定した各デバイスの空間にアクセスする必要があります。

サンプルコードでは、HyperFlash 空間および HyperRAM 空間に、キャッシュ有効領域とキャッシュ無効領域を準備しています。

HyperRAM 空間には、キャッシュ有効領域およびキャッシュ無効領域にノーマルメモリ属性の領域を準備しています。どちらの領域にアクセスした場合でも、HyperRAM のレジスタには正しいアクセスができません。HyperRAM のレジスタにアクセスする場合には、HyperRAM 空間にストロングリオーダ属性またはデバイス属性に設定した領域を準備してアクセスしてください。

HyperFlash 空間には、キャッシュ有効領域にノーマルメモリ属性の領域を、キャッシュ無効領域にストロングリオーダ属性の領域を準備しています。HyperFlash のレジスタにアクセスする場合には、HyperFlash 空間のキャッシュ無効領域（H'A000 0000 番地～H'AFF FFFF 番地）をアクセスしてください。

また、HyperFlash のメモリ領域に書き込む場合にも、レジスタのアクセスと同様に、HyperFlash 空間のストロングリオーダ属性またはデバイス属性に設定した領域にアクセスする必要があります。

8. サンプルコード

サンプルコードは、ルネサス エレクトロニクスホームページから入手してください。

9. 参考ドキュメント

ユーザーズマニュアル：ハードウェア

RZ/A2Mグループ ユーザーズマニュアル ハードウェア編

（最新版をルネサス エレクトロニクスホームページから入手してください。）

RTK7921053C00000BE（RZ/A2M CPUボード）ユーザーズマニュアル

（最新版をルネサス エレクトロニクスホームページから入手してください。）

RTK79210XXB00000BE（RZ/A2M SUBボード）ユーザーズマニュアル

（最新版をルネサス エレクトロニクスホームページから入手してください。）

Arm Architecture Reference Manual ARMv7-A and ARMv7-R edition Issue C

（最新版を Arm ホームページから入手してください。）

Arm Cortex™-A9 Technical Reference Manual Revision: r4p1

（最新版を Arm ホームページから入手してください。）

Arm Generic Interrupt Controller Architecture Specification - Architecture version2.0

（最新版を Arm ホームページから入手してください。）

Arm CoreLink™ Level 2 Cache Controller L2C-310 Technical Reference Manual Revision: r3p3

（最新版を Arm ホームページから入手してください。）

テクニカルアップデート／テクニカルニュース

（最新の情報をルネサス エレクトロニクスホームページから入手してください。）

ユーザーズマニュアル：統合開発

統合開発環境 e² studio のユーザーズマニュアルは、ルネサス エレクトロニクスホームページから入手してください。

（最新版をルネサス エレクトロニクスホームページから入手してください。）

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
Rev.1.00	Dec.20.19	－	初版発行

製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本ドキュメントおよびテクニカルアップデートを参照してください。

1. 静電気対策

CMOS 製品の取り扱いの際は静電気防止を心がけてください。CMOS 製品は強い静電気によってゲート絶縁破壊を生じることがあります。運搬や保存の際には、当社が出荷梱包に使用している導電性のトレイやマガジンケース、導電性の緩衝材、金属ケースなどを利用し、組み立て工程にはアースを施してください。プラスチック板上に放置したり、端子を触ったりしないでください。また、CMOS 製品を実装したボードについても同様の扱いをしてください。

2. 電源投入時の処置

電源投入時は、製品の状態は不定です。電源投入時には、LSI の内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。外部リセット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。同様に、内蔵パワーオンリセット機能を使用してリセットする製品の場合、電源投入からリセットのかかる一定電圧に達するまでの期間、端子の状態は保証できません。

3. 電源オフ時における入力信号

当該製品の電源がオフ状態のときに、入力信号や入出力ブルアップ電源を入れないでください。入力信号や入出力ブルアップ電源からの電流注入により、誤動作を引き起こしたり、異常電流が流れ内部素子を劣化させたりする場合があります。資料中に「電源オフ時における入力信号」についての記載のある製品は、その内容を守ってください。

4. 未使用端子の処理

未使用端子は、「未使用端子の処理」に従って処理してください。CMOS 製品の入力端子のインピーダンスは、一般に、ハイインピーダンスとなっています。未使用端子を開放状態で動作させると、誘導現象により、LSI 周辺のノイズが印加され、LSI 内部で貫通電流が流れたり、入力信号と認識されて誤動作を起こす恐れがあります。

5. クロックについて

リセット時は、クロックが安定した後、リセットを解除してください。プログラム実行中のクロック切り替え時は、切り替え先クロックが安定した後に切り替えてください。リセット時、外部発振子（または外部発振回路）を用いたクロックで動作を開始するシステムでは、クロックが十分安定した後、リセットを解除してください。また、プログラムの途中で外部発振子（または外部発振回路）を用いたクロックに切り替える場合は、切り替え先のクロックが十分安定してから切り替えてください。

6. 入力端子の印加波形

入力ノイズや反射波による波形歪みは誤動作の原因になりますので注意してください。CMOS 製品の入力がノイズなどに起因して、 V_{IL} (Max.) から V_{IH} (Min.) までの領域にとどまるような場合は、誤動作を引き起こす恐れがあります。入力レベルが固定の場合はもちろん、 V_{IL} (Max.) から V_{IH} (Min.) までの領域を通過する遷移期間中にチャタリングノイズなどが入らないように使用してください。

7. リザーブアドレス（予約領域）のアクセス禁止

リザーブアドレス（予約領域）のアクセスを禁止します。アドレス領域には、将来の拡張機能用に割り付けられている リザーブアドレス（予約領域）があります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。

8. 製品間の相違について

型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。同じグループのマイコンでも型名が違うと、フラッシュメモリ、レイアウトパターンの相違などにより、電気的特性の範囲で、特性値、動作マージン、ノイズ耐量、ノイズ幅射量などが異なる場合があります。型名が違う製品に変更する場合は、個々の製品ごとにシステム評価試験を実施してください。

ご注意書き

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器・システムの設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因して生じた損害（お客様または第三者いずれに生じた損害も含まれます。以下同じです。）に関し、当社は、一切その責任を負いません。
 2. 当社製品、本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害またはこれらに関する紛争について、当社は、何らの保証を行うものではなく、また責任を負うものではありません。
 3. 当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
 4. 当社製品を、全部または一部を問わず、改造、改変、複製、リバースエンジニアリング、その他、不適切に使用しないでください。かかる改造、改変、複製、リバースエンジニアリング等により生じた損害に関し、当社は、一切その責任を負いません。
 5. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。
標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット等
高品質水準： 輸送機器（自動車、電車、船舶等）、交通制御（信号）、大規模通信機器、金融端末基幹システム、各種安全制御装置等
当社製品は、データシート等により高信頼性、Harsh environment 向け製品と定義しているものを除き、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（宇宙機器と、海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等）に使用されることを意図しておらず、これらの用途に使用することは想定していません。たとえ、当社が想定していない用途に当社製品を使用したことにより損害が生じても、当社は一切その責任を負いません。
 6. 当社製品をご使用の際は、最新の製品情報（データシート、ユーザーズマニュアル、アプリケーションノート、信頼性ハンドブックに記載の「半導体デバイスの使用上の一般的な注意事項」等）をご確認の上、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他指定条件の範囲内でご使用ください。指定条件の範囲を超えて当社製品をご使用された場合の故障、誤動作の不具合および事故につきましては、当社は、一切その責任を負いません。
 7. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は、データシート等において高信頼性、Harsh environment 向け製品と定義しているものを除き、耐放射線設計を行っておりません。仮に当社製品の故障または誤動作が生じた場合であっても、人身事故、火災事故その他社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
 8. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。かかる法令を遵守しないことにより生じた損害に関し、当社は、一切その責任を負いません。
 9. 当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。当社製品および技術を輸出、販売または移転等する場合は、「外国為替及び外国貿易法」その他日本国および適用される外国の輸出管理関連法規を遵守し、それらの定めるところに従い必要な手続きを行ってください。
 10. お客様が当社製品を第三者に転売等される場合には、事前に当該第三者に対して、本ご注意書き記載の諸条件を通知する責任を負うものといたします。
 11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。
 12. 本資料に記載されている内容または当社製品についてご不明な点がございましたら、当社の営業担当者までお問合せください。
- 注 1. 本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社が直接的、間接的に支配する会社をいいます。
- 注 2. 本資料において使用されている「当社製品」とは、注 1 において定義された当社の開発、製造製品をいいます。

(Rev.4.0-1 2017.11)

本社所在地

〒135-0061 東京都江東区豊洲 3-2-24（豊洲フォレシア）

www.renesas.com

お問合せ窓口

弊社の製品や技術、ドキュメントの最新情報、最寄の営業お問合せ窓口に関する情報などは、弊社ウェブサイトをご覧ください。

www.renesas.com/contact/

商標について

ルネサスおよびルネサスロゴはルネサス エレクトロニクス株式会社の商標です。すべての商標および登録商標は、それぞれの所有者に帰属します。