

お客様各位

カタログ等資料中の旧社名の扱いについて

2010年4月1日を以ってNECエレクトロニクス株式会社及び株式会社ルネサステクノロジが合併し、両社の全ての事業が当社に承継されております。従いまして、本資料中には旧社名での表記が残っておりますが、当社の資料として有効ですので、ご理解の程宜しくお願い申し上げます。

ルネサスエレクトロニクス ホームページ (<http://www.renesas.com>)

2010年4月1日

ルネサスエレクトロニクス株式会社

【発行】ルネサスエレクトロニクス株式会社 (<http://www.renesas.com>)

【問い合わせ先】 <http://japan.renesas.com/inquiry>

ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連し発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、当社は、一切その責任を負いません。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。
3. 当社製品を改造、改変、複製等しないでください。
4. 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お客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
5. 輸出に際しては、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。本資料に記載されている当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途の目的で使用しないでください。また、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に使用することができません。
6. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りが無いことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されていない用途に当社製品を使用したことによりお客様または第三者に生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。
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
高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）
特定水準： 航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障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。
9. 当社は、当社製品の品質および信頼性の向上に努めておりますが、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、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害など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、機器またはシステムとしての出荷保証をお願いいたします。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、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、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。お客様が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。
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固くお断りいたします。
12. 本資料に関する詳細についてのお問い合わせその他お気付きの点等がございましたら当社営業窓口までご照会ください。

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とその総株主の議決権の過半数を直接または間接に保有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。

H8S/20103, H8S/20203, H8S/20223 グループ

データフラッシュ書き換え (EW1 モード)

要旨

H8S/20103, H8S/20203, H8S/20223 グループ内蔵のデータフラッシュ領域をEW1モードで書き換え動作を実現します。

対象デバイス

H8S/20103 (R4F20103)

H8S/20203 (R4F20203)

H8S/20223 (R4F20223)

動作確認条件

システムクロック $\phi = \phi_{osc} = 20\text{MHz}$

目次

1. 仕様	2
2. 使用機能説明	4
3. 動作原理	24
4. ソフトウェア説明	25
5. フローチャート	30
6. プログラムリスト	39

1. 仕様

本アプリケーションにおける仕様を以下に説明します。図 1 にデータフラッシュ書き換えの概要を示します。本アプリケーションノートでは、データフラッシュ A 領域の先頭から 128 バイトの書き換えをします。

- (1) データバックアップのため、データフラッシュ A 領域の 4K バイトをバックアップ用 RAM 領域に転送します。
- (2) データフラッシュ A 領域をイレーズします。
- (3) データフラッシュ A 領域をブランクチェックします。
- (4) バックアップ用 RAM 領域の先頭から 128 バイトに書き換えデータを作成します。
- (5) 書き換えをしたバックアップ用 RAM 領域 4K バイトをデータフラッシュ A 領域にライトします。
- (6) (1)~(5)より、データフラッシュ A 領域の先頭から 128 バイトを書き換えします。

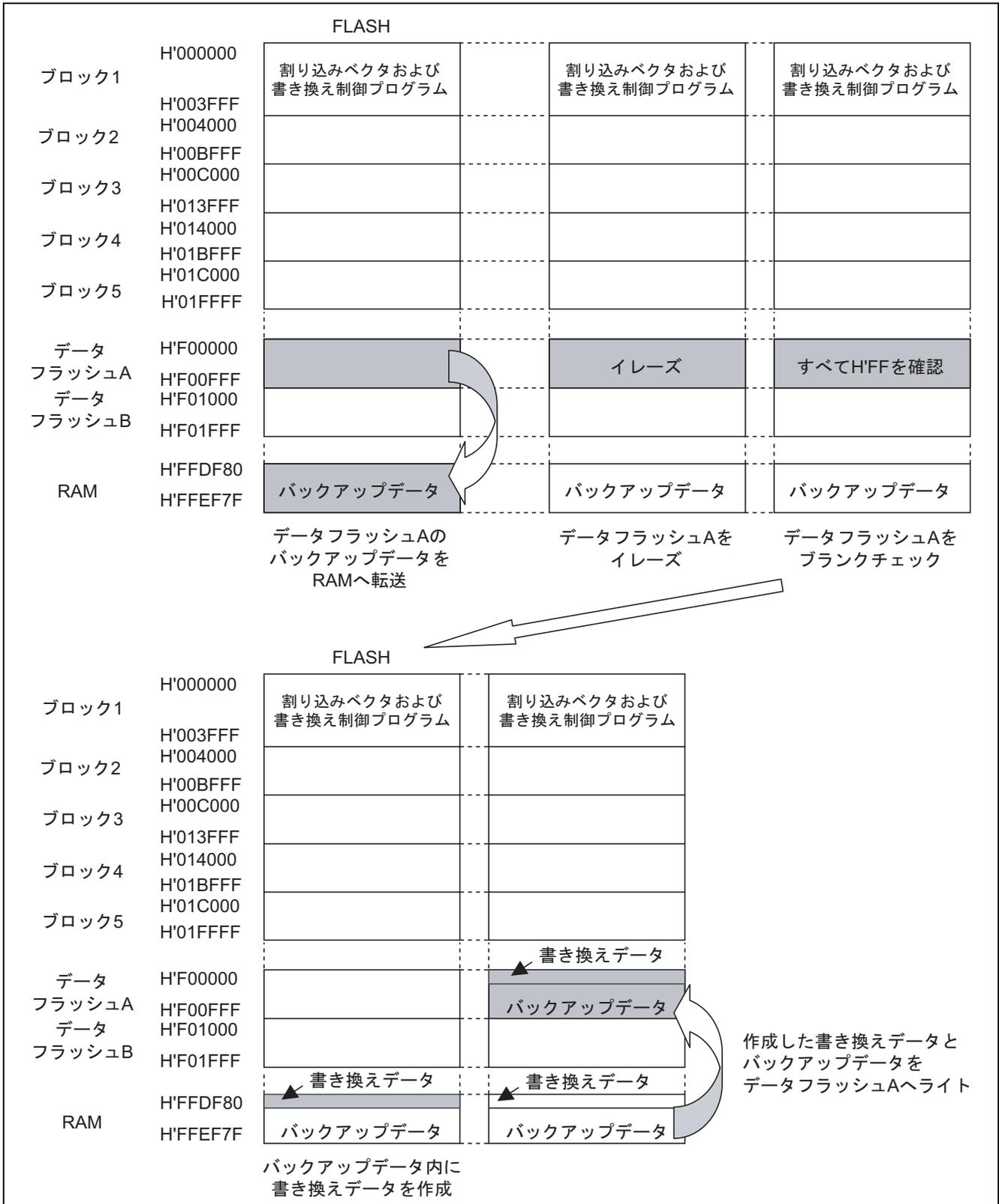


図1 データフラッシュ書き換え (EW1 モード) の概要

2. 使用機能説明

2.1 ROM

内蔵されているフラッシュメモリの特長は以下のとおりです。

- 書き込み/消去方式
書き込みは4バイトごとの同時書き込み方式です。消去はブロック単位で行います。全面消去を行う場合も1ブロックずつ消去してください。
- 書き込み/消去時間
プログラム ROM 書き込み時間：4バイト同時書き込み 150 μ s (Typ), 1バイト当たり換算 38 μ s (Typ)
データフラッシュ書き込み時間：4バイト同時書き込み 300 μ s (Typ), 1バイト当たり換算 75 μ s (Typ)
消去時間：プログラム ROM, データフラッシュ共に1ブロック当たり 200ms (Typ)
- 書き換え回数
プログラム ROM は1,000回まで, データフラッシュは10,000回まで書き換え可能です。
- 2種類のオンボードプログラミング
ブートモード：内蔵 SCI を使用して, ユーザ ROM 領域の書き込み/消去が行えます。
ブートモードでは, ホストと本 LSI 間のビットレートを自動で合わせることができます。
ユーザモード：任意のインタフェースでユーザ ROM 領域の書き込み/消去が行えます。
- ライタモード
PROM ライタを用いて書き込み/消去を行います。
- プロテクト機能
誤書き込み誤消去プロテクト
ソフトウェアによりフラッシュメモリの書き込み/消去に対するプロテクト機能のロックビットプロテクトを設定可能
- PROM ライタプロテクト/ブートモードプロテクト
ユーザ ROM 領域の指定アドレスに指定データを書くと, PROM ライタまたはブートモードで, ユーザ ROM 領域の指定アドレスにプロテクトをかけることができます。
- アクセスサイクル
プログラム ROM：1ステート
データフラッシュ：2ステート

2.1.1 ブロック構成

図2にフラッシュメモリのブロック構成を示します。ユーザROM領域には、マイコンの動作プログラムを格納する領域（プログラムROM）とは別に、データを格納する領域（データフラッシュ）があります。太線枠は消去ブロックを表します。細線枠は書き込み単位を表し、枠線内の数値はアドレスを示します。消去は図2に示す消去ブロック単位で行います。書き込みは下位アドレス4ビットがH'0, H'4, H'8, H'Cで始まる2ワードもしくは4バイト単位で行います。

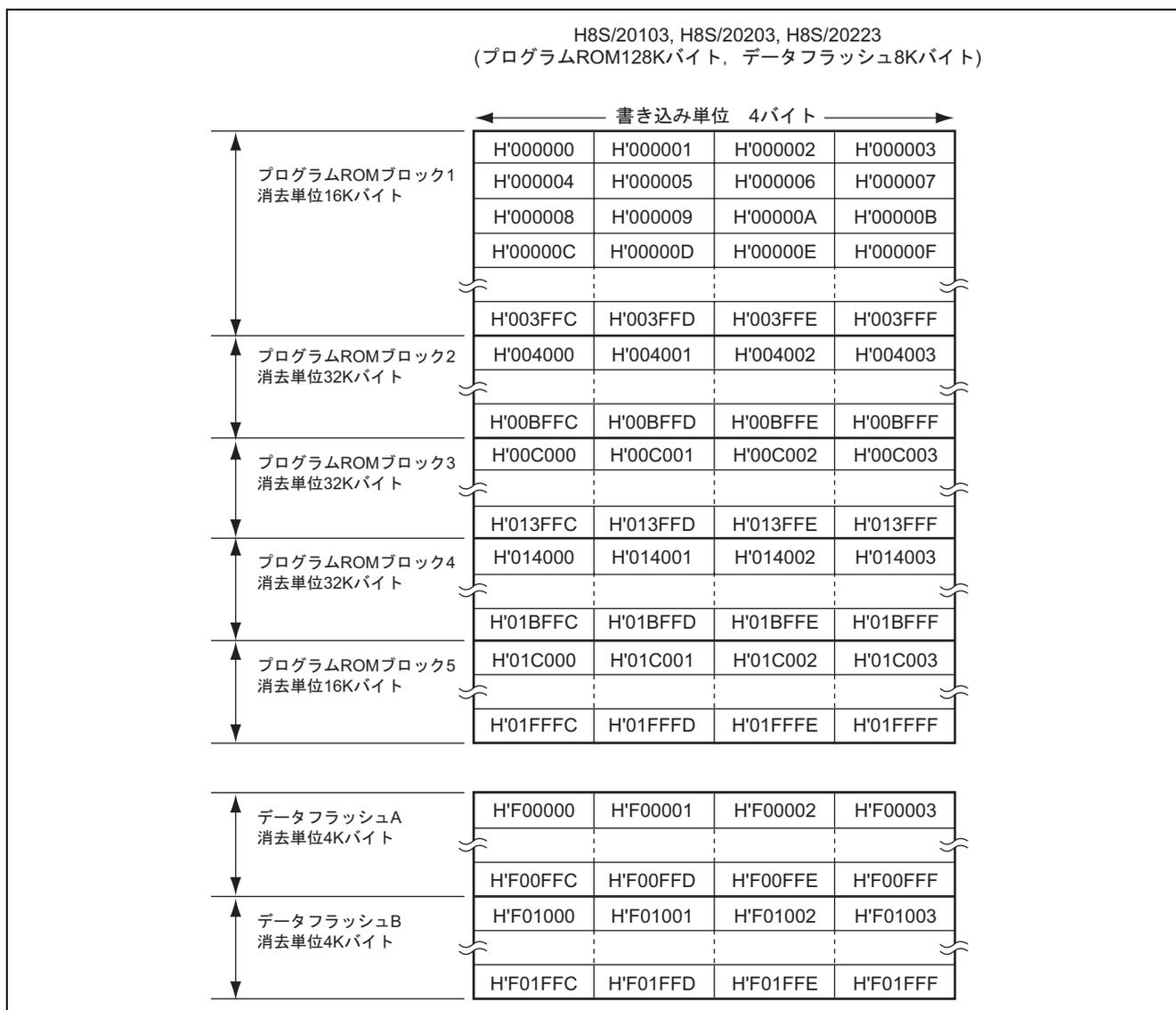


図2 フラッシュメモリのブロック構成

2.1.2 CPU 書き換えモード

CPU 書き換えモードでは、CPU がソフトウェアコマンドを実行することにより、ユーザ ROM 領域を書き換えることができます。ソフトウェアコマンドはユーザ ROM 領域の書き換えをしたい領域に対して発行してください。

CPU 書き換えモードでは、消去動作中に割り込み要求が発生した場合に、消去動作を一時中断して割り込み処理を行うイレーズサスペンド機能があります。イレーズサスペンド中はプログラムでユーザ ROM 領域を読み出すことが可能です。

CPU 書き換えモードには EW0 モードと、EW1 モードがあります。

表 1 に EW0 モードと EW1 モードの違いを示します。

表 1 EW0 モードと EW1 モードの違い

項目	EW0 モード	EW1 モード
書き換え制御プログラムを配置できる領域	ユーザ ROM 領域	
書き換え制御プログラムを実行できる領域	RAM へ転送してから実行する必要あり	ユーザ ROM 領域上で実行可能
書き換えられる領域	ユーザ ROM 領域	ユーザ ROM 領域 ただし、書き換え制御プログラムがあるブロックを除く
ソフトウェアコマンドの制限	なし	プログラム、イレーズコマンドは書き換え制御プログラムがあるブロックに対する実行禁止
ソフトウェアコマンド実行後のモード	リードアレイモード	
自動書き込み、自動消去時 CPU 状態	動作	ホールド状態 (入出力ポートはコマンド実行前の状態を保持)
フラッシュメモリのステータス検知	プログラムで FLMSTR レジスタの FMPSRF, FMERSF, FMEBSF をリード	
イレーズサスペンドへの移行条件	FLMCR2 の FMSPEN ビットが"1"かつ FMSPREQ を"1"にする もしくは、FLMCR2 の FMSPEN ビットが"1"かつ FLMCR2 の FMISPE ビットが"1"かつ割り込み要求が発生	FLMCR2 の FMSPEN ビットが"1"かつ、割り込み要求が発生
割り込み発生条件	<ul style="list-style-type: none"> • ビジー状態からレディ状態に復帰したとき*¹ • ビジー状態でユーザ ROM 領域をリードしたとき*¹ 	使用禁止
DTC の使用	使用可能* ²	使用可能* ^{2*3}

- 【注】
1. ユーザ ROM 領域へのアクセスを発生させないために VOFR を設定し、ベクタアドレスを RAM 上に配置し、かつ割り込み処理ルーチンを RAM 上に配置してください。
 2. DTC ベクタ・処理ルーチンを RAM 上に配置してください。E/W 処理中は DTC でユーザ ROM 領域をアクセスしないでください。アクセスした場合は不定値が読み出されます。
 3. RAM 上に書き換え制御プログラムを配置する場合は、DTC を使用しないでください。

2.1.3 EW0 モード

RAM へ書き換え制御プログラムを転送し、RAM 上のプログラムへ分岐したところで、FLMCR1 の FMEWMOD ビットを"0"にした後、FLMCR1 の FMCMDEN ビットを"1" (ソフトウェアコマンド許可) にすると、EW0 モードになります。

書き込み、消去動作の制御はソフトウェアコマンドで行います。ソフトウェアコマンドの終了状態などは FLMSTR で確認できます。

消去中にイレーズサスペンドに移行する場合は、FLMCR2 の FMSPEN ビットを"1" (サスペンド許可)、FMSPREQ ビットを"1" (イレーズサスペンドリクエスト) にしてください。そして、イレーズサスペンドへの遷移時間 (約 50 μ s) 待ち、FLMSTR の FMRDY ビットが"1" (レディ) になったことを確認後、ユーザ ROM 領域にアクセスしてください。FMSPREQ ビットを"0" (イレーズレジューム) にすると、消去を再開します。

割り込みを使用する場合、ユーザ ROM 領域へのアクセスを発生させないため、割り込みベクタオフセットレジスタを設定し、ベクタアドレスを RAM 上に配置し、割り込み処理ルーチンも RAM 上に配置してください。

2.1.4 EW1 モード

FLMCR1 の FMEWMOD ビットを"1"にした後、FLMCR1 の FMCMDEN ビットを"1" (ソフトウェアコマンド許可) にすると、EW1 モードになります。

書き込み、消去動作の制御はソフトウェアコマンドで行います。ソフトウェアコマンドの終了状態などは FLMSTR で確認できます。

消去中にイレーズサスペンドに移行する場合は、FLMCR2 の FMSPEN ビットを"1" (サスペンド許可) にしてからイレーズコマンドを実行してください。またイレーズサスペンドに移行するための割り込みは、あらかじめ割り込み許可状態にしてください。イレーズコマンド実行からイレーズサスペンドへの遷移時間後、割り込み要求を受け付けられます。

割り込み要求が発生すると、FMSPREQ ビットが自動的に"1" (イレーズサスペンドリクエスト) になり、消去が中断されます。割り込み処理終了後、消去が完了していないとき (FLMSTR の FMERCF ビットが"1") は、FMSPREQ ビットを"0" (イレーズレジューム) にして消去を再開させてください。

2.1.5 書き込み/消去処理

オンボードでのフラッシュメモリの書き込み/消去は CPU がソフトウェアコマンドを実行する方式 (CPU 書き換え) を採用しています。

2.1.6 ソフトウェアコマンド

表2にワード命令ソフトウェアコマンド一覧を、表3にバイト命令ソフトウェアコマンド一覧を示します。ワード命令を使用するか、バイト命令を使用するかは、FLMCR1のFMWUSビットで指定します。

表2 ソフトウェアコマンド一覧 (ワード命令 FMWUS = 1)

ソフトウェア コマンド	第1コマンドサイクル			第2コマンドサイクル			第3コマンドサイクル			コマンド使用	
	モード	アドレス	データ	モード	アドレス	データ	モード	アドレス	データ	EW0	EW1
イレーズ	ライト	x	H'2020	ライト	BA	H'D0D0					
プログラム	ライト	WA	H'4141	ライト	WA	WD1	ライト	WA	WD2		
ブランク チェック	ライト	x	H'2525	ライト	BA	H'D0D0					
ロックビット プログラム	ライト	x	H'7777	ライト	BA	H'D0D0					
リードアレイ	ライト	x	H'FFFF								—
クリア ステータス	ライト	x	H'5050								
ロックビット リード	ライト	x	H'7171	リード	BA	H'xxxx					x

- 【記号説明】
- x: ユーザ ROM 領域の任意の番地
 - xx: 8ビットの任意のデータ
 - BA: ブロックの任意の番地
 - WA: 書き込み番地 (アドレス下位2ビットは無視されます。各コマンドサイクルの書き込み番地は同一番地を設定してください。)
 - WDn: 書き込みデータ (16ビット)

表 3 ソフトウェアコマンド一覧 (バイト命令 FMWUS = 0)

ソフトウェア コマンド	第 1 コマンドサイクル			第 2 コマンドサイクル			第 3 コマンド~第 5 コマンド サイクル			コマンド使用	
	モード	アドレス	データ	モード	アドレス	データ	モード	アドレス	データ	EW0	EW1
イレース プログラム	ライト	x	H'20	ライト	BA	H'D0					
プログラム	ライト	WA	H'41	ライト	WA	WD1	ライト	WA	WD2~ WD4		
ブランク チェック	ライト	x	H'25	ライト	BA	H'D0					
ロックビット プログラム	ライト	x	H'77	ライト	BA	H'D0					
リードアレイ	ライト	x	H'FF								—
クリア ステータス	ライト	x	H'50								
ロックビット リード	ライト	x	H'71	リード	BA	H'xx					x

- 【記号説明】
- x: ユーザ ROM 領域の任意の番地
 - xx: 8 ビットの任意のデータ
 - BA: ブロックの任意の番地
 - WA: 書き込み番地 (アドレス下位 2 ビットは無視されます。各コマンドサイクルの書き込み番地は同一番地を設定してください。)
 - WDn: 書き込みデータ (8 ビット)

(1) CPU 書き換えモードの初期化

ソフトウェアコマンド発行を行う前に CPU 書き換えモードの設定, およびソフトウェアコマンド発行を許可する必要があります。

図 3 に EW0 モード, 図 4 に EW1 モードの CPU 書き換えモード初期化を示します。

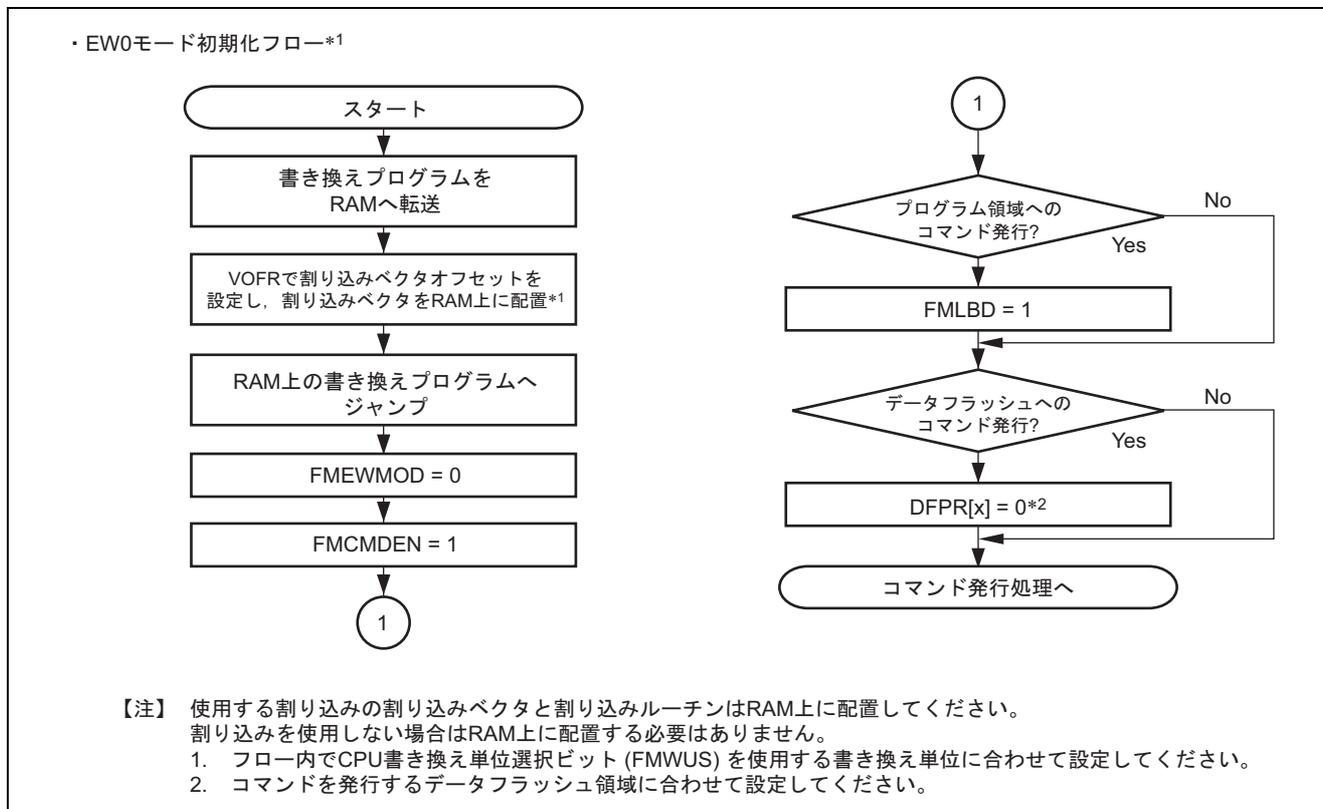
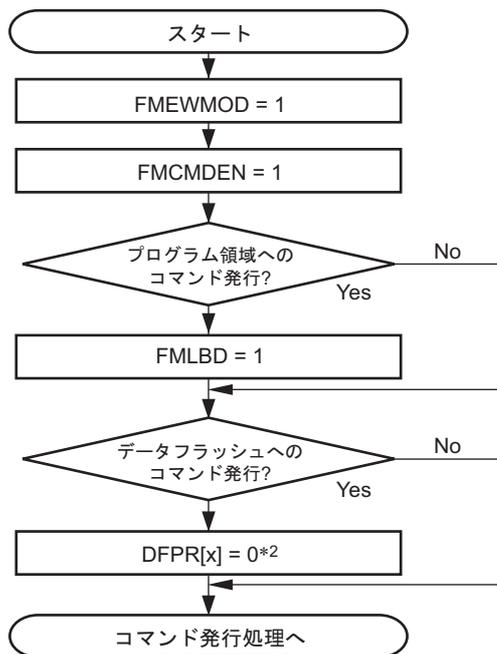


図 3 EW0 モードの初期化

・ EW1モード初期化フロー*1



- 【注】 使用する割り込みの割り込みベクタと割り込みルーチンはRAM上に配置してください。
割り込みを使用しない場合はRAM上に配置する必要はありません。
1. フロー内でCPU書き換え単位選択ビット (FMWUS) を使用する書き換え単位に合わせて設定してください。
 2. コマンドを発行するデータフラッシュ領域に合わせて設定してください。

図 4 EW1 モードの初期化

(2) イレーズコマンド

第1コマンドサイクルで H'20, 第2コマンドサイクルで H'D0 をブロックの任意の番地に書くと指定されたブロックに対し自動的に消去/消去ベリファイを開始します。

消去終了は FLMSTR の FMRDY ビットで確認できます。FMRDY ビットは消去期間中は"0", 終了後は"1"になります。

消去終了後 FLMSTR の FMEBSF ビットで消去の結果を知ることができます。(9) フルステータスチェック参照)。

また, 指定されたブロックのロックビットが"0" (ロック状態) で FMLBE ビットが"0" (ロックビット有効) の場合, 指定ブロックに対するイレーズコマンドは受け付けられません。図5にイレーズサスペンド機能未使用時のフローを, 図6に EW0 モード, 図7に EW1 モード, イレーズサスペンド機能使用時のフローを示します。

イレーズサスペンド機能使用時においてイレーズサスペンド後, イレーズレジュームにより消去を再開する場合, PC がインクリメントし正常に命令フェッチができないことがあります。これを回避するために, FMSPREQ = 0 ライト直後に NOP 命令を2個追加してください。また, RAM 上に書き換え制御プログラムを配置し, EW1 モードでイレーズサスペンドを行う場合は, DTC を使用しないでください。

EW1 モードでは, 書き換え制御プログラムが配置されているブロックに対して, このコマンドを実行しないでください。

また, FLMSTR の FMRDY ビットは消去開始と共に"0"となり, 終了と共に"1"に戻ります。

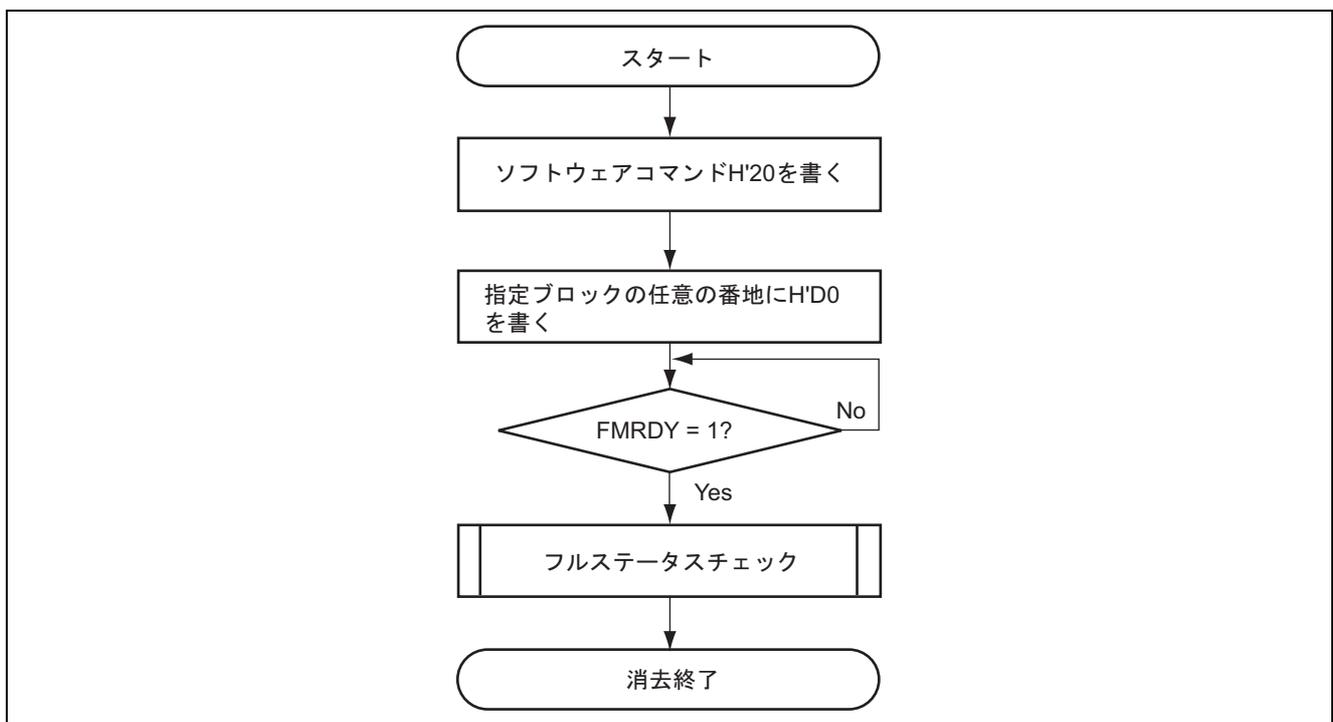


図5 イレーズサスペンド機能未使用時のフロー

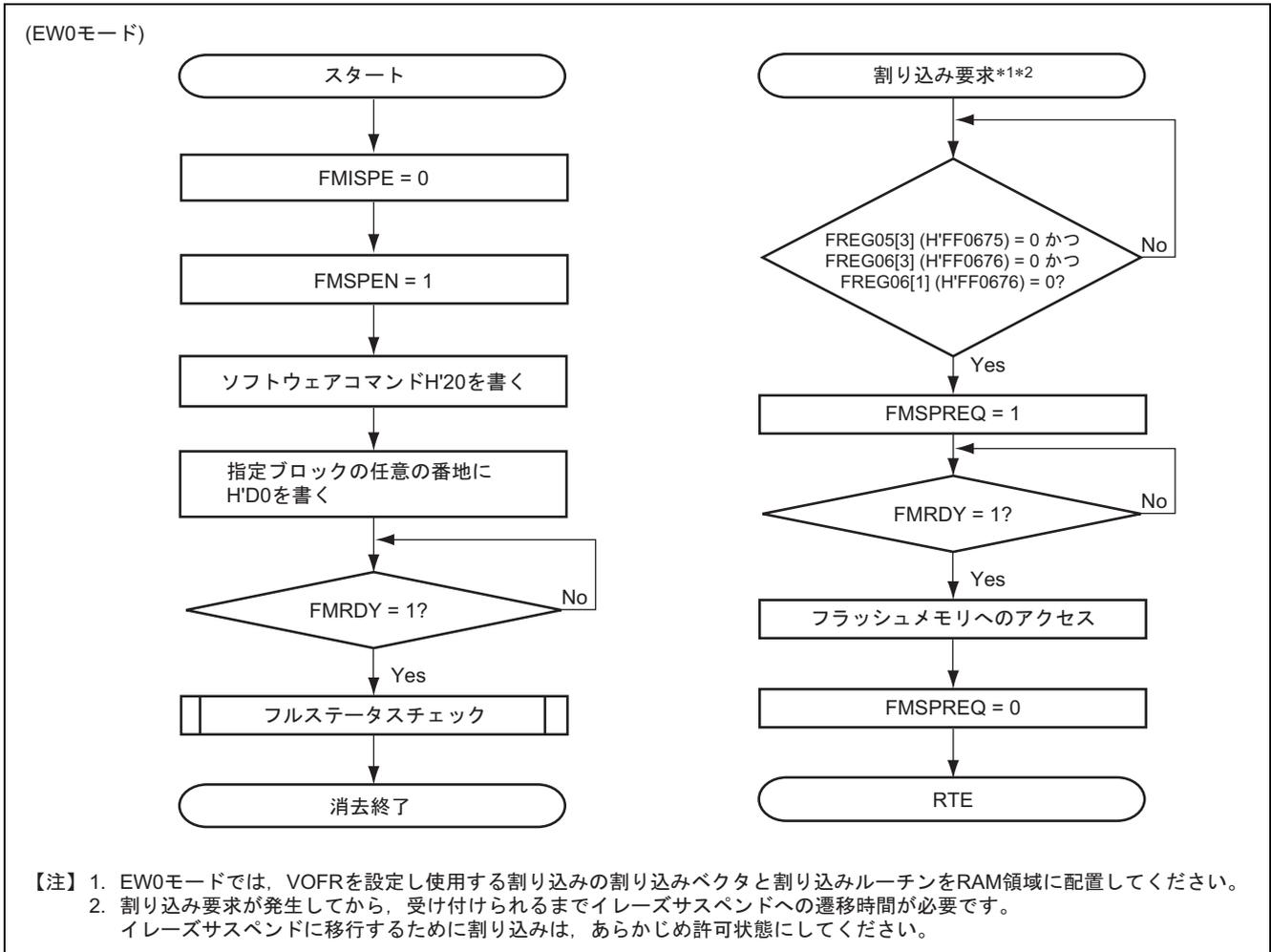


図 6 EW0 モード，イレーズサスペンド機能使用時のフロー

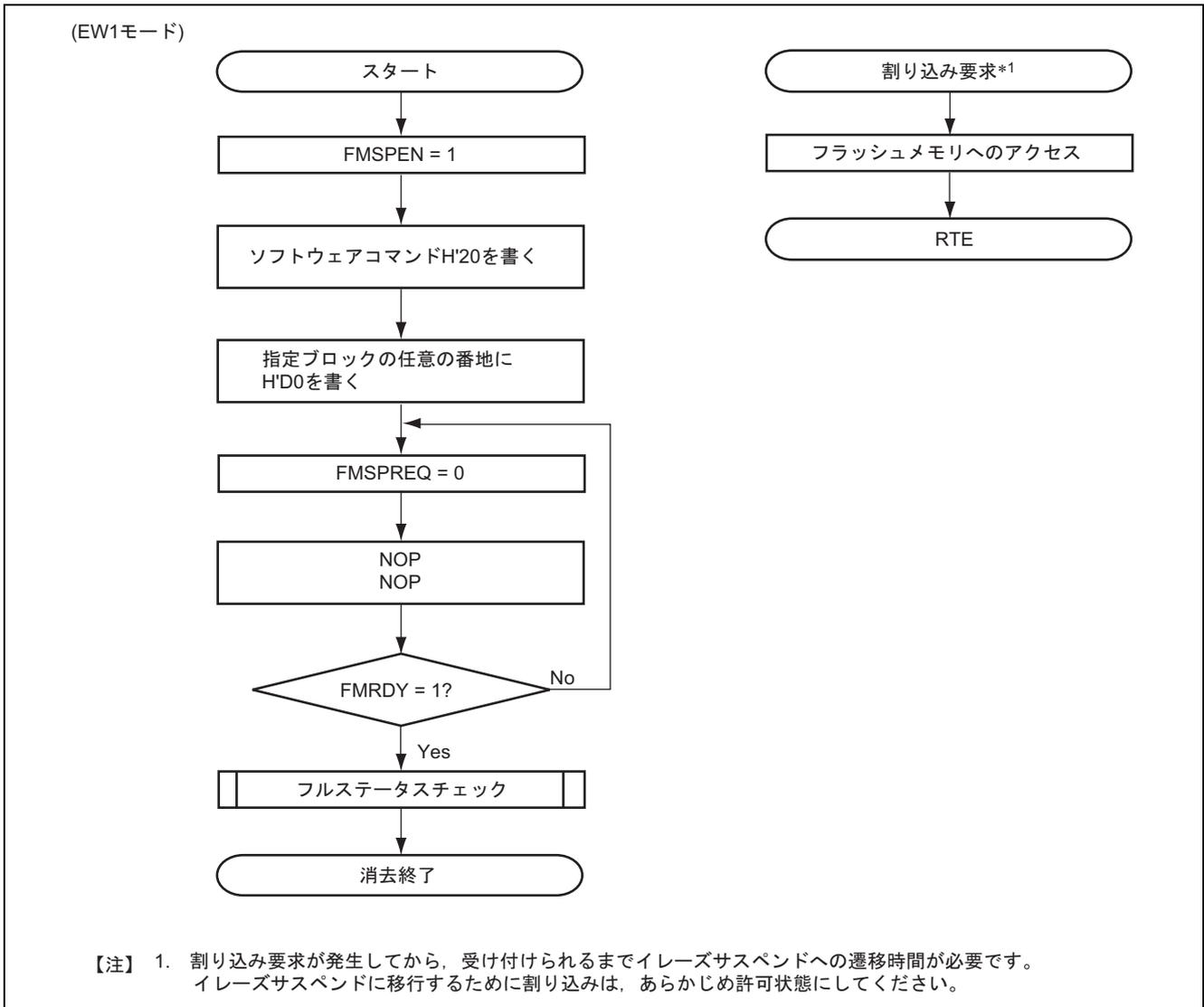


図7 EW1 モード、イレーズサスペンド機能使用時のフロー

(3) プログラムコマンド

4 バイト単位でフラッシュメモリにデータを書き込むコマンドです。

コマンドおよびデータのサイズは、FLMCR1 の FMWUS ビットで設定できます。FMWUS ビットが"0"の場合、バイト命令での書き込みになります。第 1 コマンドサイクルで H'41 を書き、第 2 コマンドサイクル～第 5 コマンドサイクルで書き込み番地にデータを書くと自動的に書き込みとベリファイを開始*します。

FMWUS ビットが"1"の場合、ワード命令での書き込みになります。第 1 コマンドサイクルで H'4141 を書き、第 2 コマンドサイクル・第 3 コマンドサイクルで書き込み番地にデータを書くと、書き込みとベリファイを開始*します。

プログラムの終了は FLMSTR の FMRDY ビットで確認できます。FMRDY ビットは書き込み期間中は"0"、終了後は"1"になります。

書き込み終了後、FLMSTR の FMPSF ビットで書き込みの結果を知ることができます。(9) フルステータスチェック参照)。図 8 に書き込みフローを示します。

既にプログラムされた番地に対する追加書き込みはしないでください。

また、指定されたブロックのロックビットが"0" (ロック状態) で FMLBD ビットが"0" (ロックビット有効) の場合、指定ブロックに対するプログラムコマンドは受け付けられません。

EW1 モードでは、書き換え制御プログラムが配置されているブロックに対して、このコマンドを実行しないでください。

また、FLMSTR の FMRDY ビットはプログラム開始と共に"0"となり、終了と共に"1"に戻ります。

【注】 * 書き込み番地の下位 2 ビットは無視されます。

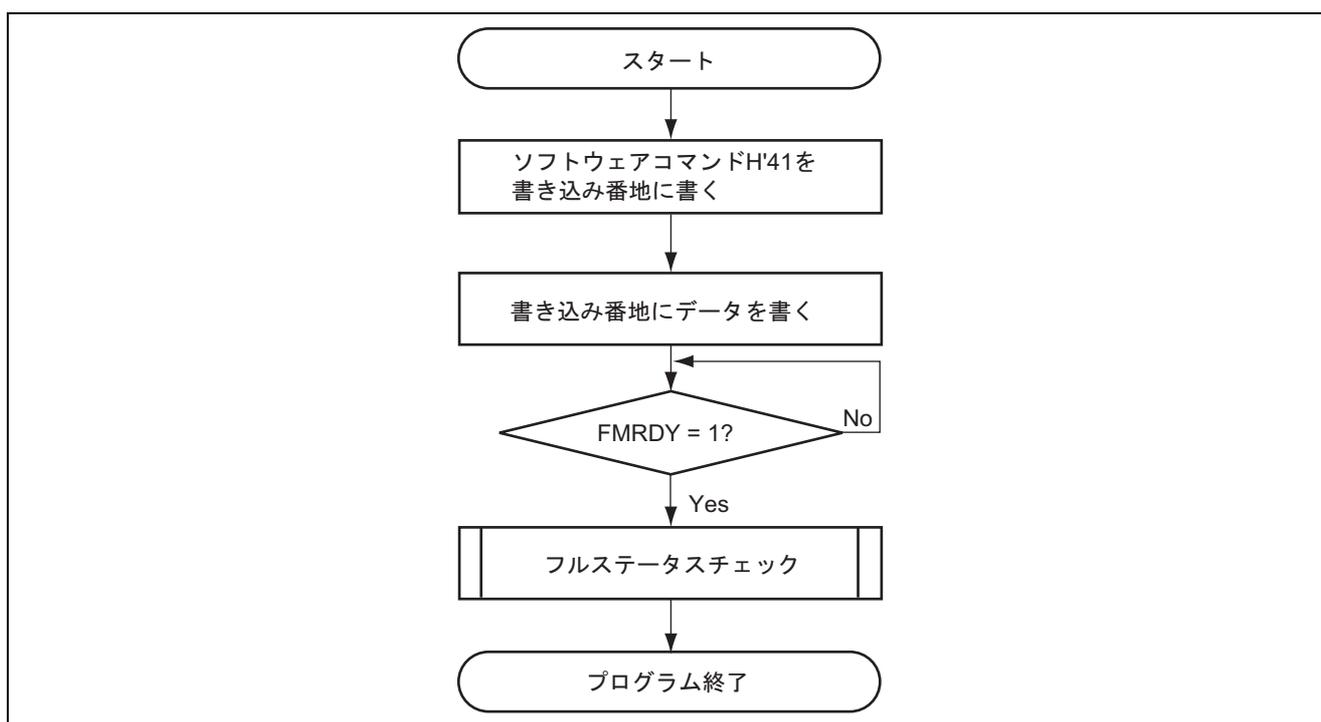


図 8 書き込みフロー

(4) ブランクチェックコマンド

第1コマンドサイクルで H'25, 第2コマンドサイクルで H'D0 をブロックの任意の番地を書く指定されたブロックに対しブランクチェックを開始します。

ブランクチェックの終了は FLMSTR の FMRDY ビットで確認できます。FMRDY ビットは, ブランクチェック期間中は"0", 終了後は"1"になります。

ブランクチェック終了後, FLMSTR の FMEBSF ビットでブランクチェックの結果を知ることができます ((9) フルステータスチェック参照)。図9にブランクチェックフローを示します。

また, FLMSTR の FMRDY ビットはブランクチェック開始と共に"0"となり, 終了と共に"1"に戻ります。

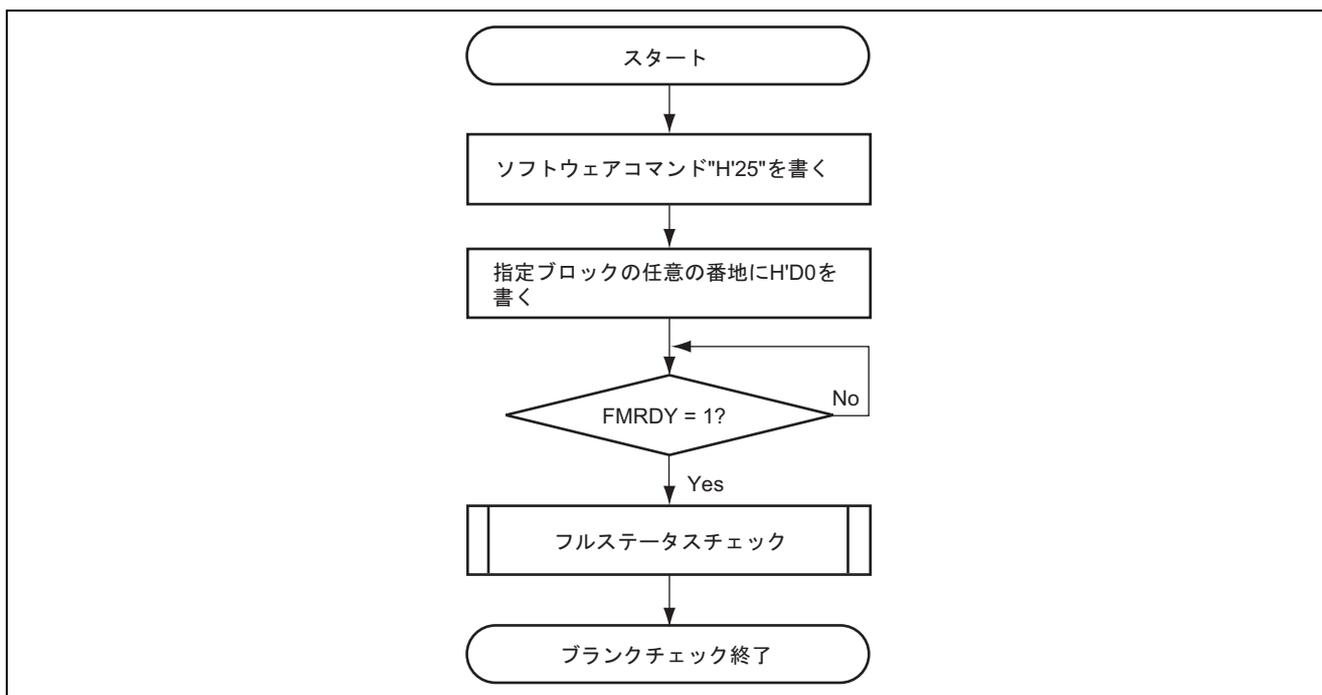


図9 ブランクチェックフロー

(5) ロックビットプログラムコマンド

第1コマンドサイクルで H'77, 第2コマンドサイクルで H'D0 をブロックの任意の番地を書く指定されたブロックに対しロックビット書き込みを開始します。

ロックビット書き込みの終了は FLMSTR の FMRDY ビットで確認できます。FMRDY ビットは、ロックビット書き込み期間中は "0", 終了後は "1" になります。

ロックビット書き込み終了後, FLMSTR の FMPSF ビットで書き込みの結果を知ることができます (9) フルステータスチェック参照)。

図 10 にロックビット書き込みフローを示します。

また, FLMSTR の FMRDY ビットはロックビット書き込み開始と共に "0" となり, 終了と共に "1" に戻ります。

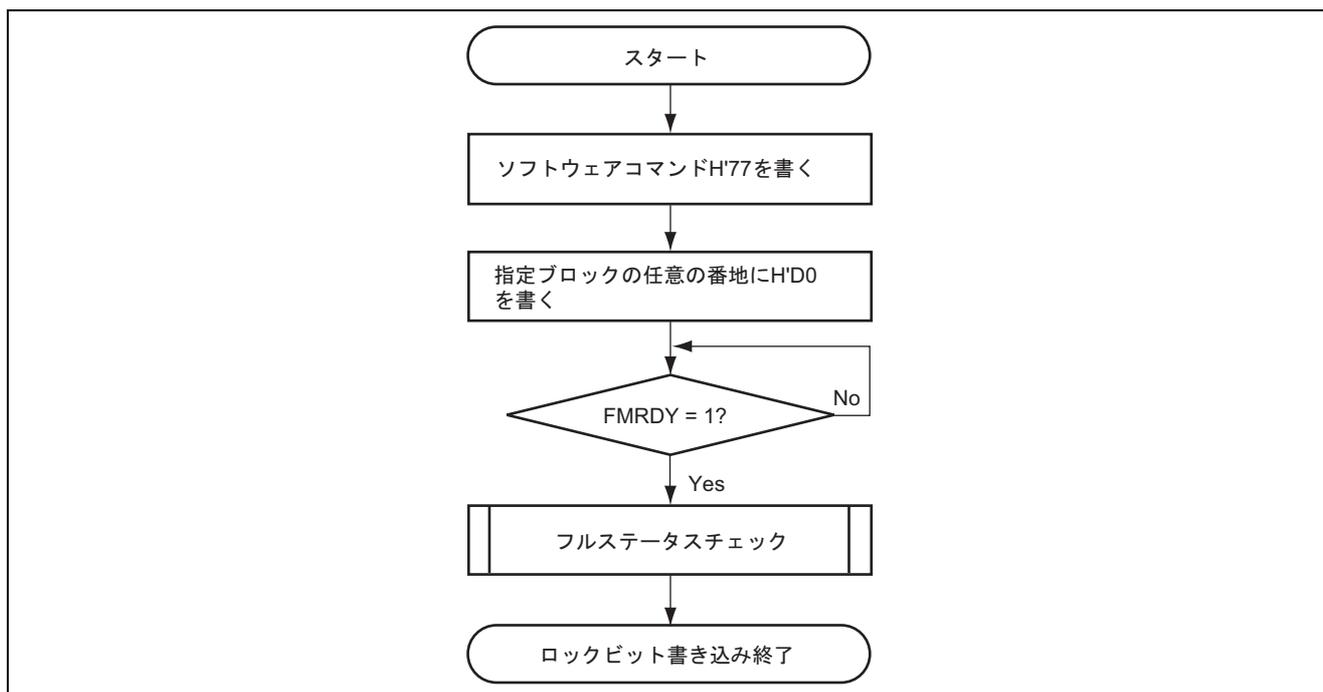


図 10 ロックビット書き込みフロー

(6) リードアレイコマンド

フラッシュメモリのデータが読めるモードへ遷移するコマンドです。

第1コマンドサイクルで H'FF を書くと、リードアレイモードになります。次のコマンドサイクル以降で指定番地を読み出すと、指定した番地のデータが読み出せます。

リードアレイモードは他のコマンドが書かれるまでは保持されるので、複数の番地を続けて読み出せます。

(7) ロックビットリードコマンド

フラッシュメモリのロックビットが読めるモードへ遷移するコマンドです。

第1コマンドサイクルで H'71、第2コマンドサイクルで H'D0 をブロックの任意の番地に書くと、指定されたブロックに対しロックビットリードを開始します。

ロックビットリードモードに遷移した後、指定ブロックアドレス BA をリードすると、リードしたデータのビット 14 の値でロックビットの値を知ることができます。ROM 上ではこのコマンドを実行しないでください。

(8) クリアステータスコマンド

ステータスフラグを"0"にするコマンドです。

第1コマンドサイクルで H'50 を書くと、FLMSTR の FMPRSF ビットと FMEBSF ビットが"0"になります。

(9) フルステータスチェック

各コマンド（リードアレイ、クリアステータスを除く）を発行した場合、フルステータスチェックを行い、コマンド実行によるエラーの有無を確認します。

エラーが発生すると、FLMSTR の FMEBSF ビットと FMPRSF ビットが"1"になり、各エラーの発生を示します。

表 4 にエラーと FLMSTR の状態を、図 11 にフルステータスチェックのフロー、図 12 に各エラー発生時の対処方法を示します。

表 4 エラーと FLMSTR の状態

FLMSTR の状態		エラー	エラー発生条件
FMEBSF	FMPRSF		
0	0	正常終了	
0	1	プログラムエラー	プログラムコマンドを実行し、正しく書き込みされなかったとき
		ロックビットプログラムエラー	ロックビットプログラムコマンドを実行し、正しく書き込みされなかったとき
1	0	イレーズエラー	イレーズコマンドを実行し、正しく消去されなかったとき
		ブランクチェックエラー	ブランクチェックコマンドを実行し、指定ブロックがブランクではなかったとき
1	1	コマンドシーケンスエラー	<ul style="list-style-type: none"> • コマンドが正しく書き込まれなかったとき • 2 サイクルのコマンドの最終サイクルでデータに書いてもよい値 (H'D0 または H'FF) 以外のデータを書いたとき • イレーズサスペンド中にイレーズコマンドを入力したとき • イレーズサスペンド中にサスペンドブロックへのプログラムコマンドを入力したとき

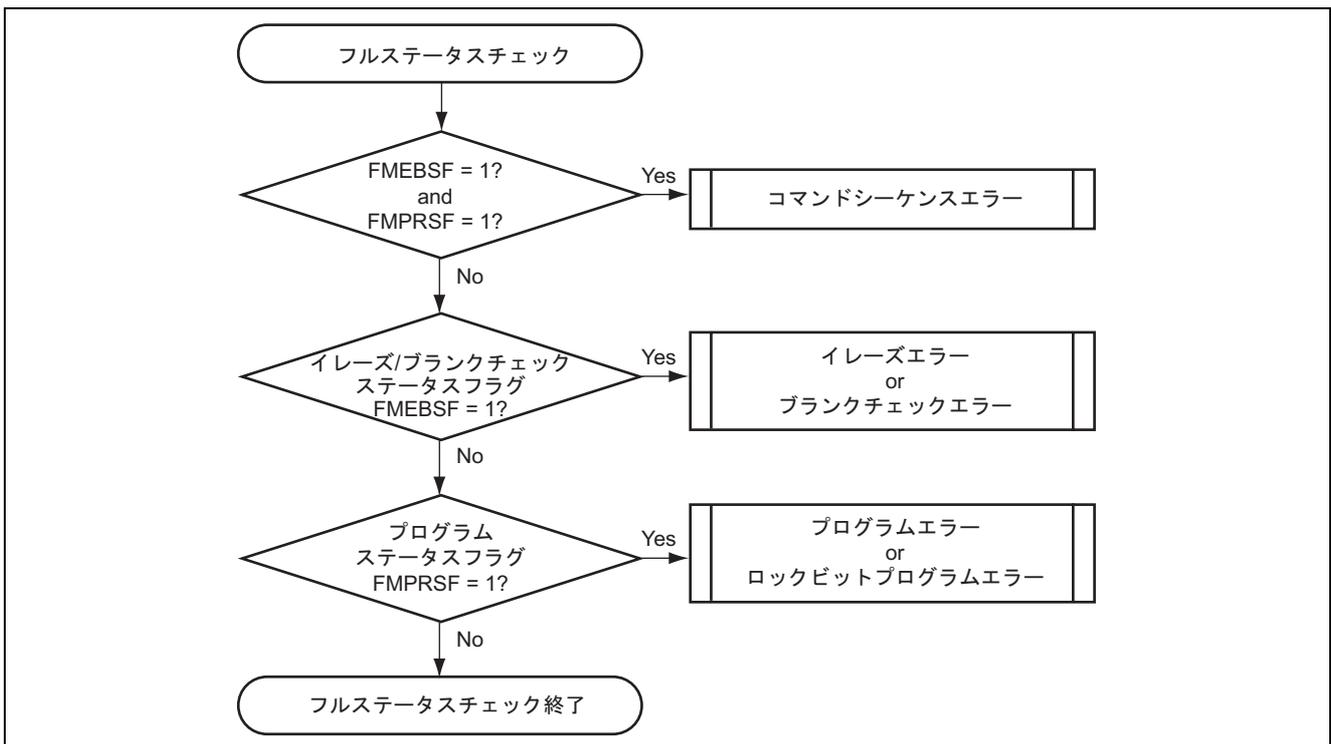


図 11 フルステータスチェックのフロー

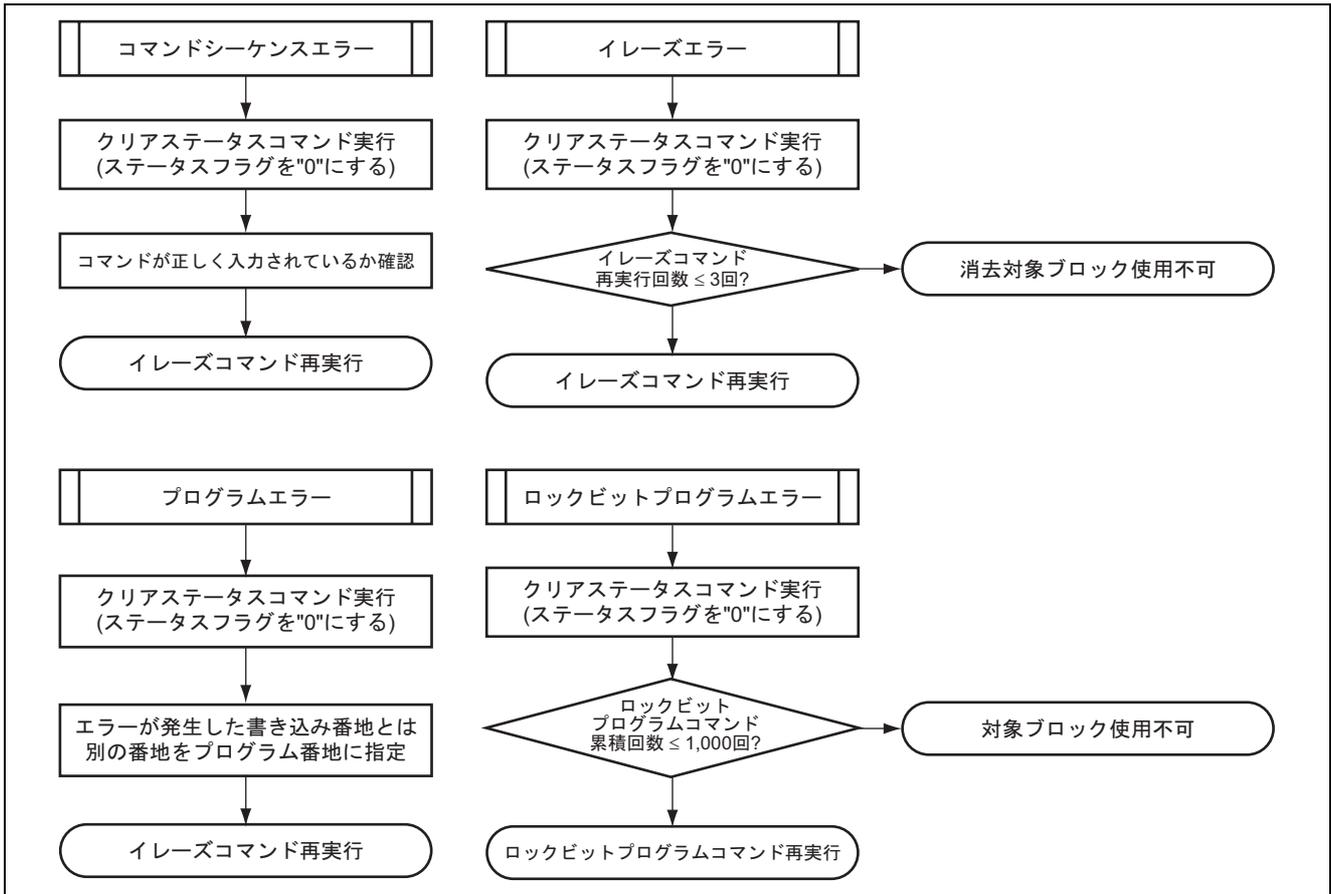


図 12 各エラー発生時の対処方法

2.1.7 プロテクト

フラッシュメモリに対する読み込み/書き込み/消去プロテクトの状態にはソフトウェアプロテクトによるもの、ロックビットプロテクトによるものおよびライターモードやブートモードでのアクセスを制限するプロテクトがあります。

2.1.8 ソフトウェアプロテクト

ソフトウェアでフラッシュメモリコントロールレジスタ (FLMCR1) の FMCMDEN をクリアすることでソフトウェアコマンド無効状態になります。この状態でソフトウェアコマンドを入力してもコマンドは実行されません。

また、フラッシュメモリデータフラッシュプロテクトレジスタ (DFPR) の設定により、データフラッシュはブロックごとにプロテクトが可能です。DFPR の DFPR1 ビットおよび DFPR0 ビットを "1" に設定すると全データフラッシュがプロテクト状態になります。

2.1.9 ロックビットプロテクト

ロックビットプロテクトはロックビットプログラムコマンドでロックビットを書き込むことで、書き込み/消去無効状態になります。この状態でイレーズ/プログラムコマンドを入力してもコマンドは実行されません。これにより、CPU の暴走などによる誤消去・誤書き込みを防止します。

プロテクトを解除する場合は、FLMCR1 の FMLBD ビットをセットすることで一時的に無効にすることができます。ロックビットをクリアする場合は、指定ブロックを消去してください。また、ロックビットはデータフラッシュには使用できません。

2.1.10 使用上の注意事項

(1) 使用禁止命令

EW0 モードでは、次の命令はフラッシュメモリ内部のデータを参照するため、使用できません。

- TRAP 命令

(2) 割り込み

表 5 に CPU 書き換えモード時の割り込みを示します。

表 5 CPU 書き換えモード時の割り込み

モード	状態	割り込み要求受け付け時	ウォッチドッグタイマリセット, LVD リセット, ソフトウェアリセット, 端子リセット発生時
EW0	イレーズコマンド 実行中	割り込みベクタを RAM に配置することで、使用することができます。(「H8S/20103, H8S/20203, H8S/20223 グループ ハードウェアマニュアル 割り込みベクタオフセットレジスタ」を参照してください。)	リセットが発生するとすぐに、ソフトウェアコマンドを強制停止し、フラッシュメモリ、LSI をリセットします。ソフトウェアコマンド実行中のブロックまたはアドレスは強制停止されるため、正常値が読み出せなくなる場合がありますので、再起動した後、再度消去を実行し、正常終了することを確認してください。 ウォッチドッグタイマは、コマンド動作中も停止しないため、定期的にウォッチドッグタイマを初期化してください。
	プログラムコマンド 実行中		
	ロックビットプログラムコマンド 実行中		
	ブランクチェック コマンド実行中		
EW1	イレーズコマンド 実行中 (イレーズサスペンド無効)	イレーズ実行が優先され、割り込み要求が待たされます。イレーズが終了した後、割り込み処理を実行します。	リセットが発生するとすぐに、ソフトウェアコマンドを強制停止し、フラッシュメモリ、LSI をリセットします。ソフトウェアコマンド実行中のブロックまたはアドレスは強制停止されるため、正常値が読み出せなくなる場合がありますので、再起動した後、再度消去を実行し、正常終了することを確認してください。 ウォッチドッグタイマは、コマンド動作中も停止しないため、ウォッチドッグタイマのオーバフロー時間を消去/書き込み実行時間以上になるように設定してください。
	イレーズコマンド 実行中 (イレーズサスペンド有効)	イレーズサスペンドへの遷移時間後にイレーズを中断し、割り込み処理を実行します。 割り込み処理が終了後、FLMCR2 の FMSPREQ ビットを "0" (イレーズリスタート) にすることにより、イレーズを再開することができます。	
	プログラムコマンド 実行中	ソフトウェアコマンド実行が優先され、割り込み要求が待たされません。ソフトウェアコマンドが終了した後、割り込み処理を実行しません。	
	ロックビットプログラム コマンド実行中		
	ブランクチェック コマンド実行中		

(3) アクセス方法

下記に示すプロテクトのあるビットに値を書く場合は、当該ビットに"0"を書いた後、続けて"1"を書くかまたは、"1"を書いた後、続けて"0"を書く必要があります。なお、"0"を書いた後、"1"を書くか、"1"を書いた後、"0"を書くまでに他の内部 I/O レジスタへのアクセスや割り込みが入らないようにしてください。また、必ず MOV 命令で書いてください。

- a. "1"にする場合、"0"を書いた後、続けて"1"を書くビット
FLMCR1 : FMLBD ビットおよび FMCMDEN ビット
FLMCR2 : FMISPE ビットおよび FMSPEN ビット
- b. "0"にする場合、"1"を書いた後、続けて"0"を書くビット
DFPR : DFPR1 ビット, DFPR0 ビット

FLMCR1 の FMCMDEN ビット"0"→"1", FMLBD ビットを"0"→"1"にする場合のプログラム例を示します。

```
MOV.B    @FLMCR1, R0L      :FLMCR1=H'04      R0L=H'04      ROH=H'xx
MOV.B    @FLMCR1, R0H      :FLMCR1=H'04      R0L=H'04      ROH=H'04
BSET     #0, R0H           :FLMCR1=H'04      R0L=H'04      ROH=H'05
BSET     #3, R0H           :FLMCR1=H'04      R0L=H'04      ROH=H'0D
MOV.B    R0L, @FLMCR1     :FLMCR1=H'04      R0L=H'04      ROH=H'0D
MOV.B    R0H, @FLMCR1     :FLMCR1=H'0D      R0L=H'04      ROH=H'0D
```

(4) ユーザ ROM 領域の書き換え

EW0 モードを使用し、書き換え制御プログラムが格納されているブロックを書き換えている最中に電源が低下すると、書き換え制御プログラムが正常に書き換えられないため、その後フラッシュメモリの書き換えができなくなる可能性があります。書き換え制御プログラムの書き換えは、電源の安定性を十分に確保した状態で行ってください。

(5) プログラム

既に書き込みされた番地に対する追加書き込みはしないでください。

(6) LSI モード遷移

ソフトウェアコマンド実行中に、スタンバイモード、スリープモードに移行しないでください。

(7) フラッシュメモリのソフトウェアコマンド実行中のリセット

プログラムコマンド, ロックビットプログラムコマンド, ブランクチェックコマンド, イレーズコマンドを実行中に端子リセット, LVD リセット, ウォッチドッグリセット, ソフトウェアリセットを入れしないでください。リセットを入れた場合, 実行中のコマンドが強制停止されます。強制停止した場合は指定されたブロックに再度イレーズコマンドを実行し, 正常終了することを確認してください。

3. 動作原理

本タスク例の動作原理を図 13 に示します。図 13 に示すようなハードウェア処理，およびソフトウェア処理によって，データフラッシュの書き換え動作をします。

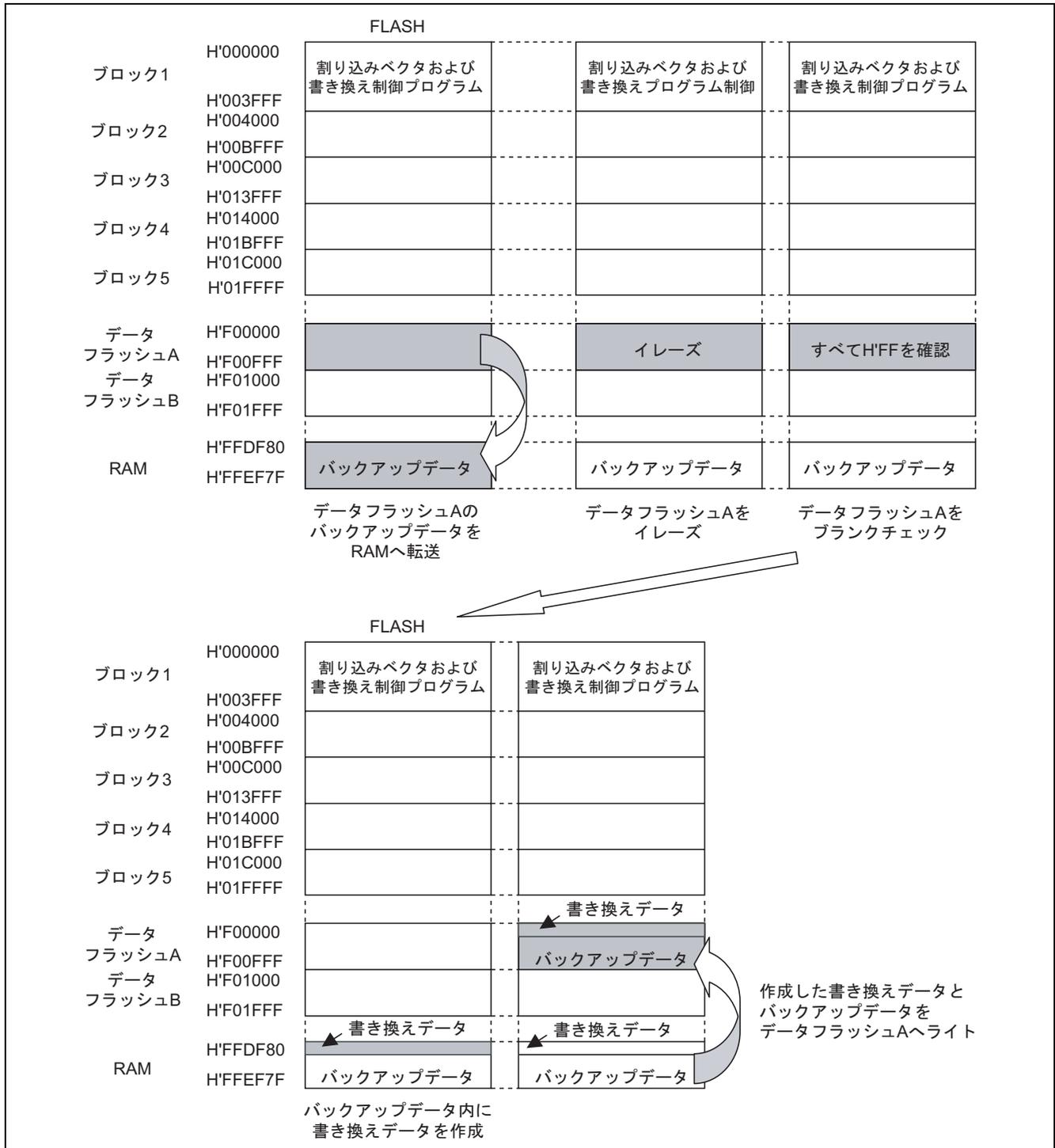


図 13 本タスク例の動作原理

4. ソフトウェア説明

4.1 モジュール説明

表 6 に本タスク例におけるモジュール説明を示します。

表 6 モジュール説明

モジュール名	メインルーチン		
宣言	void main(void)		
引数	引数名	型	意味
	なし	なし	なし
戻り値	型	値	意味
	なし	なし	なし
機能説明	各関数の呼び出し		

モジュール名	システム初期化ルーチン		
宣言	void h8s_sysinit(void)		
引数	引数名	型	意味
	なし	なし	なし
戻り値	型	値	意味
	なし	なし	なし
機能説明	モジュールスタンバイの設定, WDT の停止, システムクロックの設定, パスマスタ動作クロックの設定		

モジュール名	イレースルーチン		
宣言	unsigned char ew1_erase (unsigned char *er_blk)		
引数	引数名	型	意味
	er_blk	unsigned char *	イレース対象ブロックのアドレス設定
戻り値	型	値	意味
	unsigned char	H'00: 正常終了 H'10: イレースエラー H'18: コマンドシーケンスエラー	イレースコマンド実行結果
機能説明	EW1 モードでデータフラッシュ A 領域をイレース		

モジュール名	ブランクチェックルーチン		
宣言	unsigned char ew1_blank_check (unsigned char *blank_blk)		
引数	引数名	型	意味
	blank_blk	unsigned char *	ブランクチェック対象ブロックのアドレス設定
戻り値	型	値	意味
	unsigned char	H'00: 正常終了 H'10: ブランクチェックエラー H'18: コマンドシーケンスエラー	ブランクチェックコマンド実行結果
機能説明	EW1 モードでデータフラッシュ A 領域をブランクチェック		

モジュール名	プログラムルーチン		
宣言	unsigned char ew1_write(volatile unsigned char *wr_top, volatile unsigned char *wr_end, volatile unsigned char *wr_data)		
引数	引数名	型	意味
	wr_top	volatile unsigned char *	書き込み対象先頭アドレス
	wr_end	volatile unsigned char *	書き込み対象終了アドレス + 1
	wr_data	volatile unsigned char *	書き込みデータ格納アドレス
戻り値	型	値	意味
	unsigned char	H'00: 正常終了 H'08: プログラムエラー H'18: コマンドシーケンスエラー	プログラムコマンド実行結果
機能説明	EW1 モードでデータフラッシュ A 領域をライト		

モジュール名	フルステータスチェックルーチン		
宣言	unsigned char full_status_check(unsigned char *addr)		
引数	引数名	型	意味
	addr	unsigned char *	フルステータスチェック対象アドレス設定
戻り値	型	値	意味
	unsigned char	H'00: 正常終了 H'08: プログラムエラーまたはロックビットプログラムエラー H'10: イレーズエラーまたはブランクチェックエラー H'18: コマンドシーケンスエラー	フルステータスチェック実行結果
機能説明	フルステータスチェック		

4.2 使用内部レジスタ説明

表 7 に本タスク例における使用内部レジスタを説明します。

表 7 使用内部レジスタ説明

レジスタ名	シンボル	機能	アドレス	設定値
PMRJ	PMRJ[1:0]	PJ0, PJ1 端子を OSC1, OSC2 端子に設定	H'FF000C	B'11
FLMCR1	FMWUS	CPU をバイト命令書き換えに設定	H'FF0660	0
	FMEWMOD	EW1 モードに設定		1
	FMCM DEN* ¹ * ² * ³ * ⁴	フラッシュメモリソフトウェアコマンド許可		1
DFPR	DFPR1* ⁵ * ⁶	データフラッシュ B の E/W を禁止	H'FF0662	1
	DFPR0* ⁵ * ⁶	データフラッシュ A の E/W を許可		0
FLMSTR	FMEBSF* ⁷ * ⁸	イレーズ/ブランクチェックステータスフラグ 0: 正常終了 1: エラー終了 [1 になる条件] • イレーズコマンドを実行し, 正しく消去されなかったとき • ブランクチェックコマンドを実行し, 指定ブロックがブランクではなかったとき [0 になる条件] • クリアステータスコマンドを発行したとき	H'FF0663	—
	FMPRSF* ⁷ * ⁸	プログラムステータスフラグ 0: 正常終了 1: エラー終了 [1 になる条件] • プログラムコマンドを実行し, 正しく書き込みされなかったとき • ロックビットプログラムコマンドを実行し, 正しく書き込みされなかったとき [0 になる条件] • クリアステータスコマンドを発行したとき		—
	FMRDY	フラッシュメモリレディ/ビジーステータスフラグ 0: ビジー(書き込み, 消去実行中) 1: レディ [1 になる条件] • 書き込み/消去動作中以外するとき [0 になる条件] • 書き込み/消去動作中のとき		—
SYSCCR	PHIHSEL	ϕ high クロックソースを ϕ osc に設定	H'FF06D0	1
LPCR1	PSCSTP	PSC 分周回路動作	H'FF06D1	0
	PHIBSEL	ϕ base クロックソースを ϕ high に設定		1
LPCR2	PHI[2:0]	システムクロック ϕ を ϕ base に設定	H'FF06D2	B'000
LPCR3	PHIS[2:0]	パスマスタ動作クロック ϕ s を ϕ に設定	H'FF06D3	B'000
OSCCSR		ϕ osc 発振安定時間設定	H'FF06D5	H'0E

レジスタ名	シンボル	機能	アドレス	設定値
TMWD		WDT にクロック入力禁止	H'FFFF99	H'F7
TCSRWD		TMWD 書き込み制御	H'FFFF9A	H'A3
MSTCR1	MSTWDT	WDT モジュールスタンバイ解除	H'FFFFDC	0

- 【注】
- "1"にするときは一度"0"をライトした後、続けて"1"をライトしてください。このとき、割り込みが入らないようにしてください。
 - FMRDY ビットが"0"から"1"になるときに"0"になります。
 - FMEWMOD ビットを設定した後に、FMCMDEN ビットを"1"にしてください。
 - FMEWMOD ビットが"0"のときに FMCMDEN ビットを"1"にする場合は、必ずプログラムを RAM 上で実行してください。
 - DFPR ビットを"0"にするときは、一度"1"をライトした後、続けて"0"をライトしてください。このとき、割り込みが入らないようにしてください。
 - DFPR ビットは、FMCMDEN ビットが"0"→"1"になるときに、"1"になります。
 - ソフトウェアで"1"にすることはできません。
 - クリアステータスコマンドを実行すると"0"になります。

4.3 使用 RAM 説明

表 8 に本タスク例における使用 RAM 説明を示します。

表 8 使用 RAM 説明

ラベル名	機能	メモリ使用量	使用モジュール名
—	データフラッシュ A バックアップ領域 (H'FFDF80 ~ H'FFEF7F)	4KB	—

4.4 使用定義説明

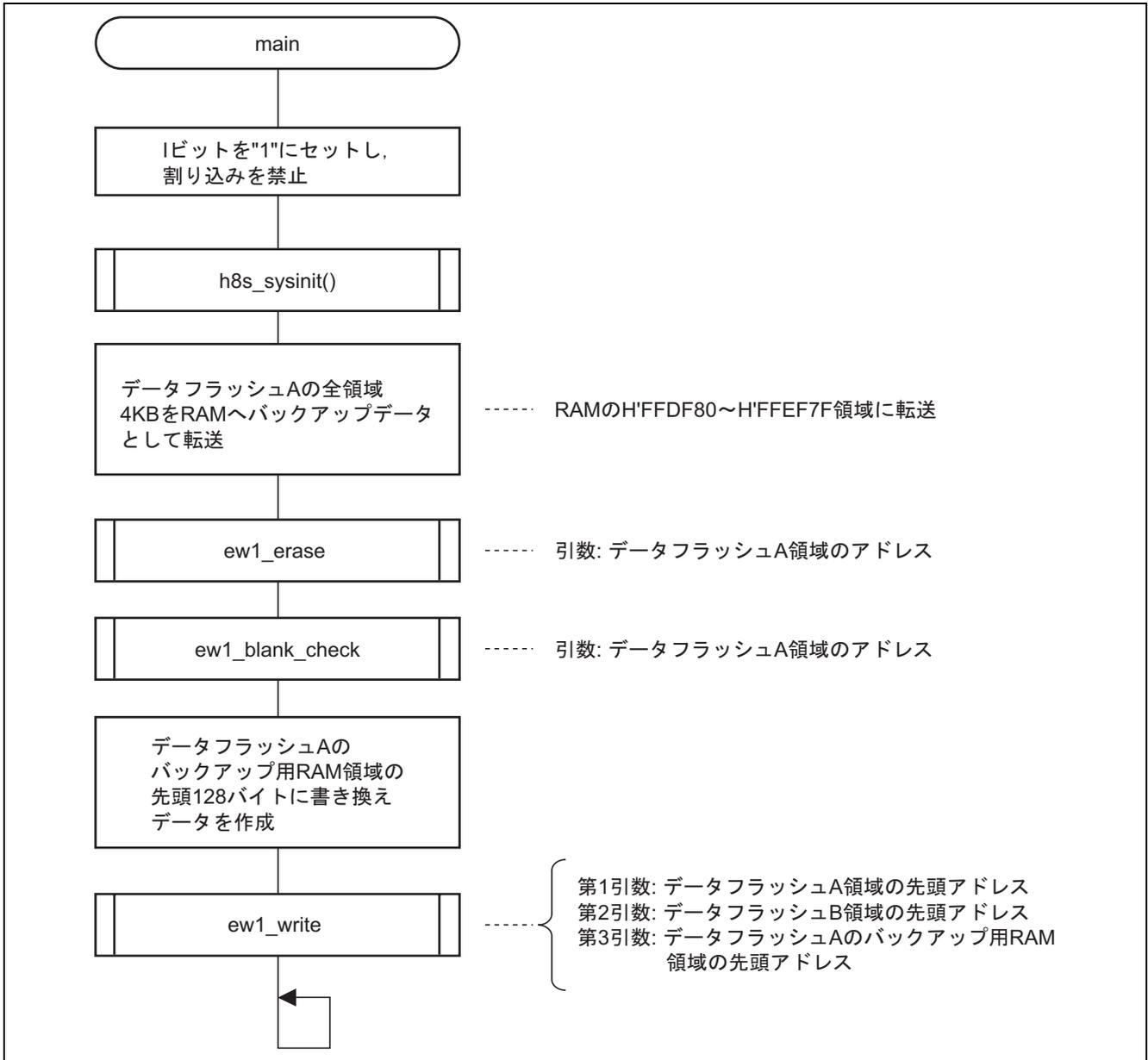
表 9 に本タスク例における使用定義説明を示します。

表 9 使用定義説明

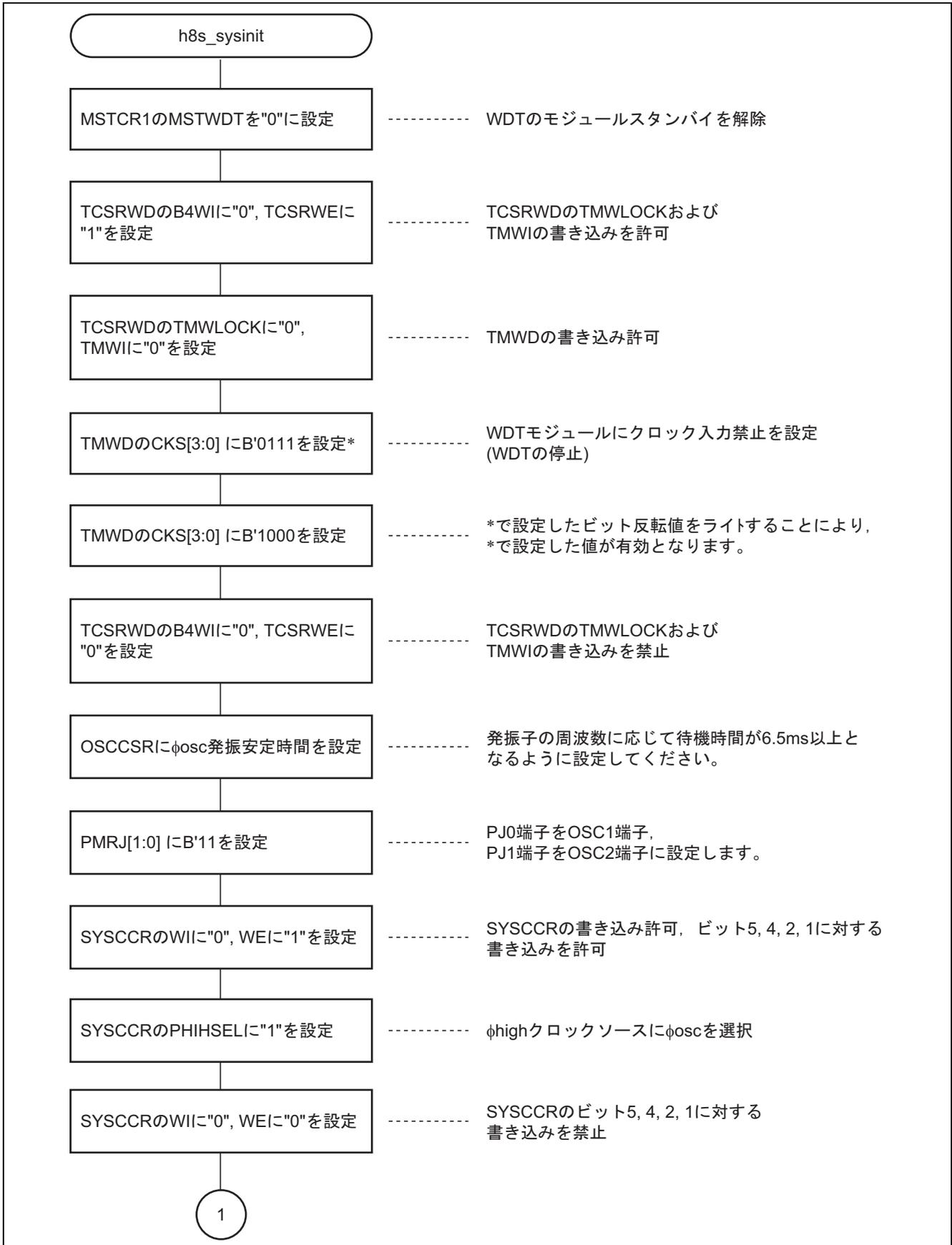
ラベル名	説明	定数値
FULL_STATUS	FLMSTR のフルステータスチェックのマスクデータ	H'28
COMMAND_SEQUENCE_ERR	FLMSTR のフルステータスチェックのコマンドシーケンスエラーデータ	H'28
ERASE_BLANK_ERR	FLMSTR のフルステータスチェックのイレーズエラー , ブランクチェックエラーデータ	H'20
PRG_LOCKBIT_ERR	FLMSTR のフルステータスチェックのプログラムエラー , ロックビットプログラムエラーデータ	H'08
NO_ERR	FLMSTR のフルステータスチェックの正常終了データ	H'00
WRITE_SIZE	データフラッシュ A 書き換えデータサイズ	H'80
BACK_UP_AREA	データフラッシュ A のバックアップ用 RAM 領域先頭 アドレス	H'FFDF80
FLASH_BLK_A	データフラッシュ A のブロック先頭アドレス	H'F00000
FLASH_BLK_B	データフラッシュ B のブロック先頭アドレス	H'F01000
FLASH_BLK_B_END	データフラッシュ B のブロック末尾アドレス	H'F01FFF

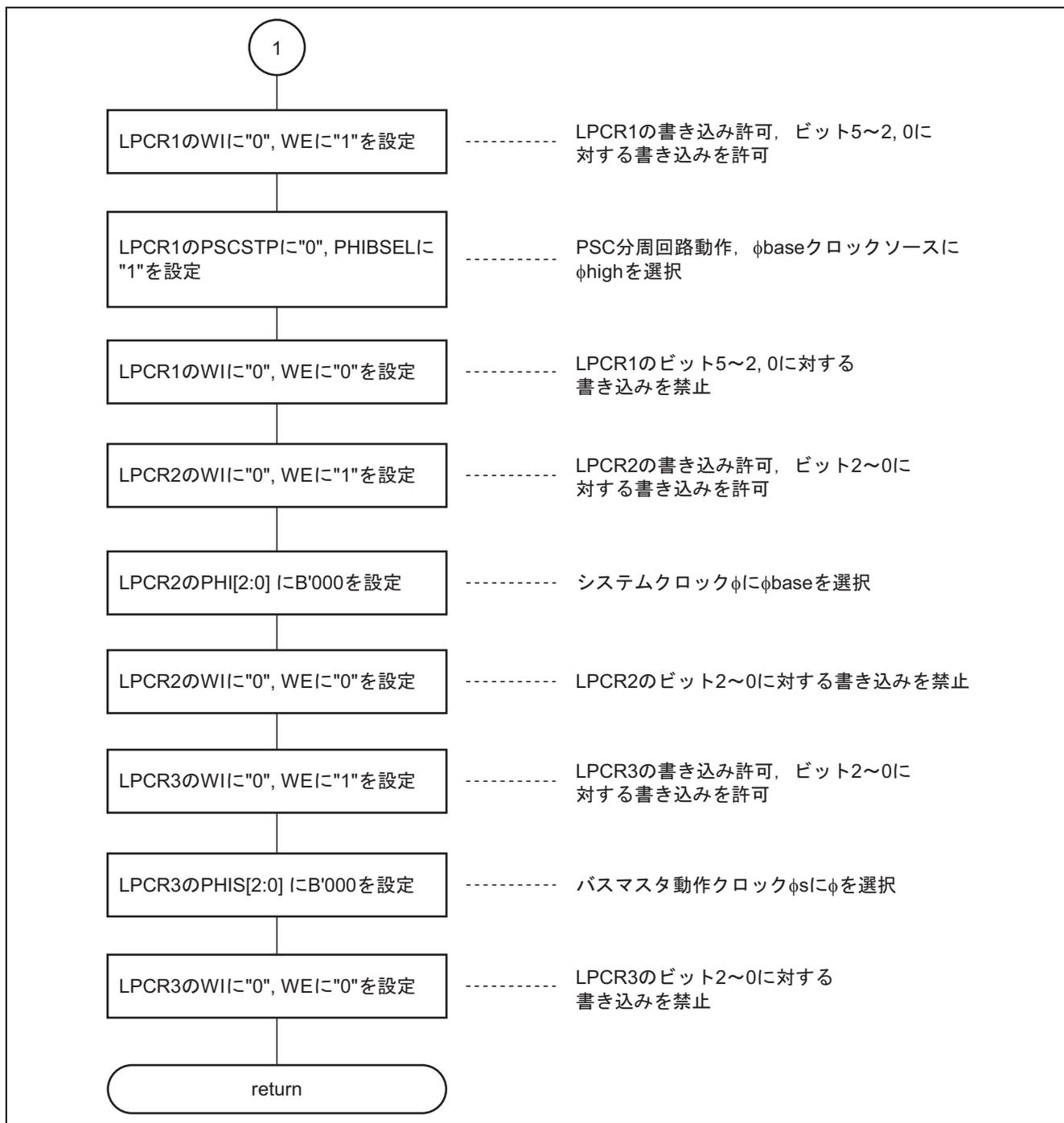
5. フローチャート

5.1 メインルーチン

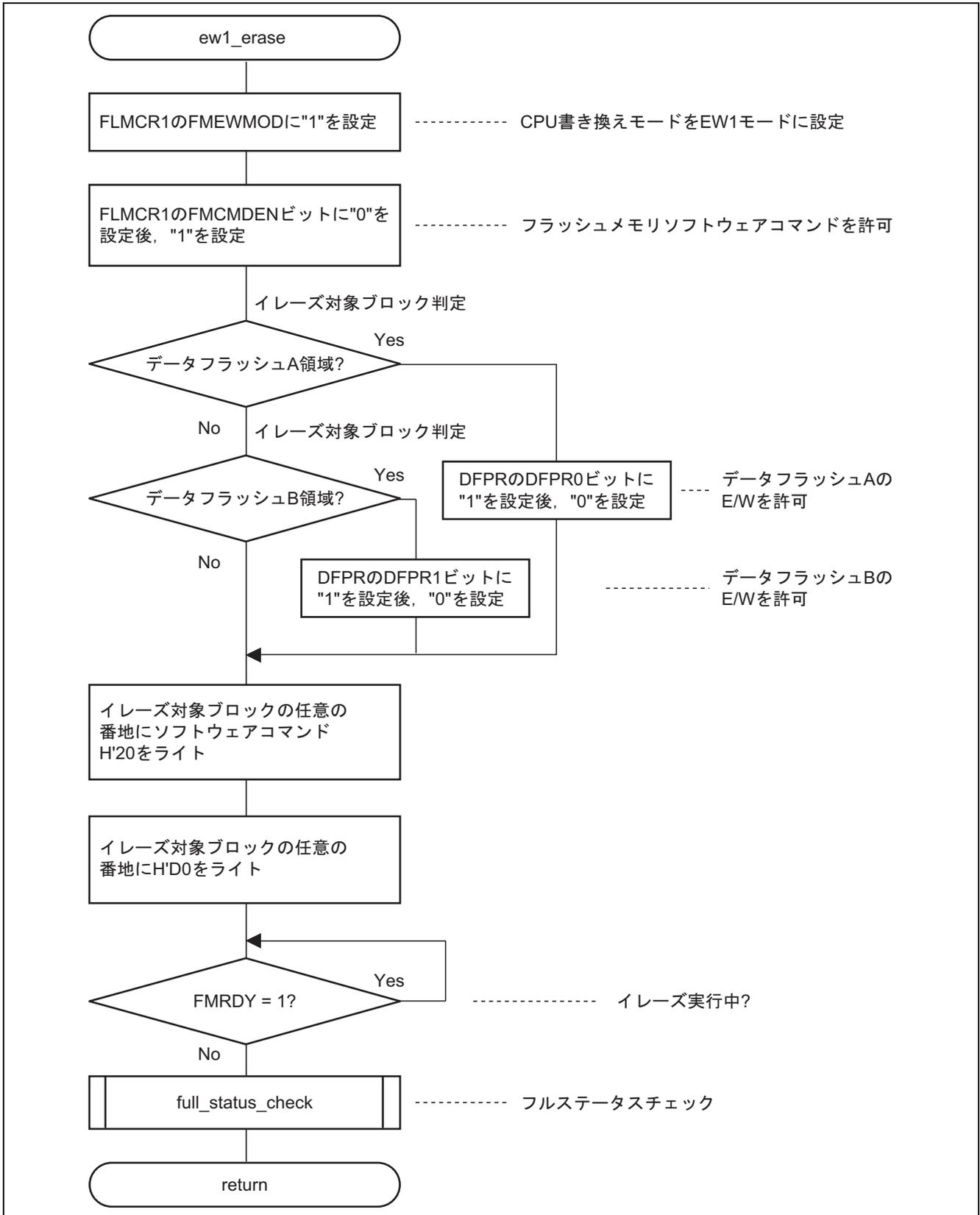


5.2 システムの初期化ルーチン

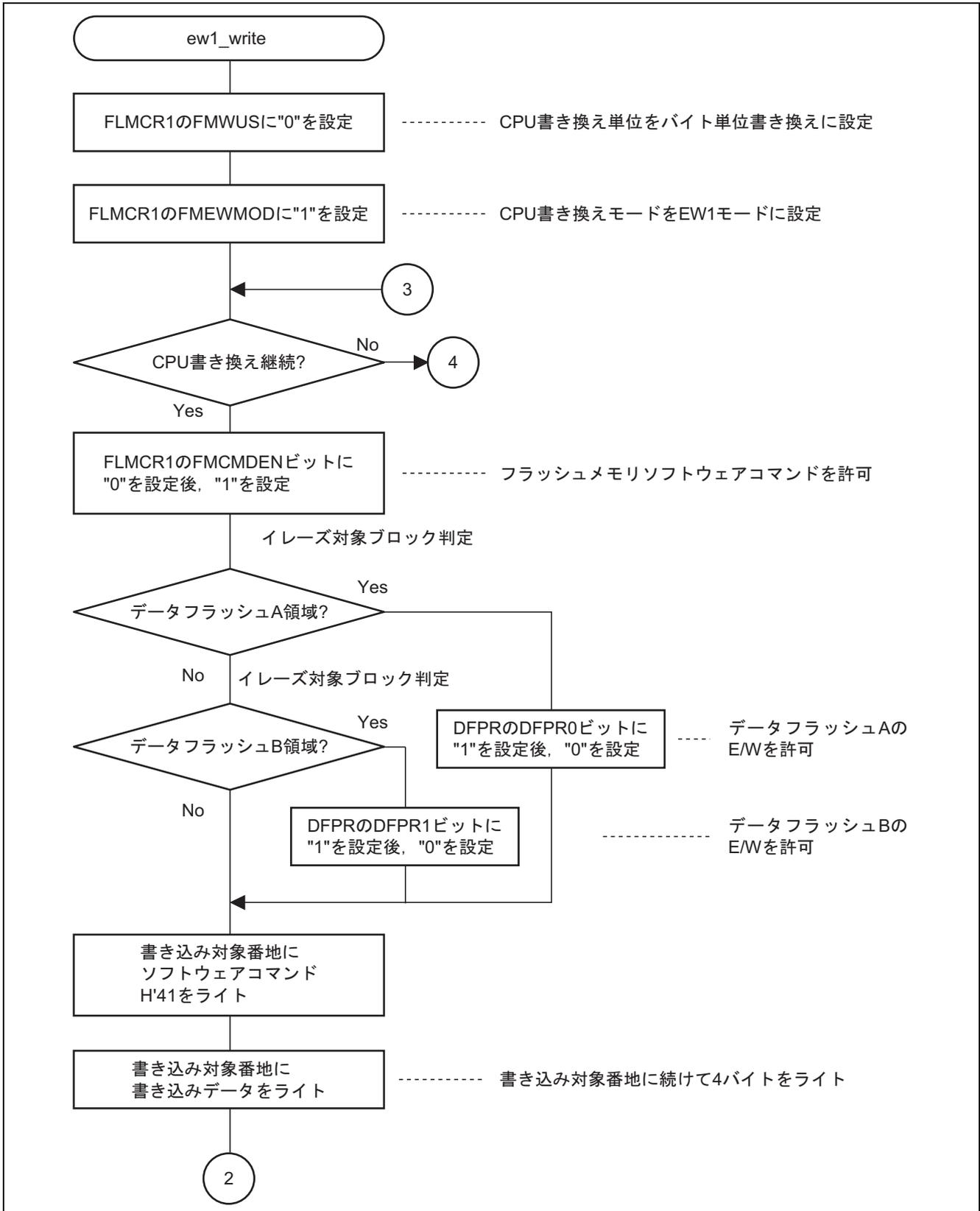


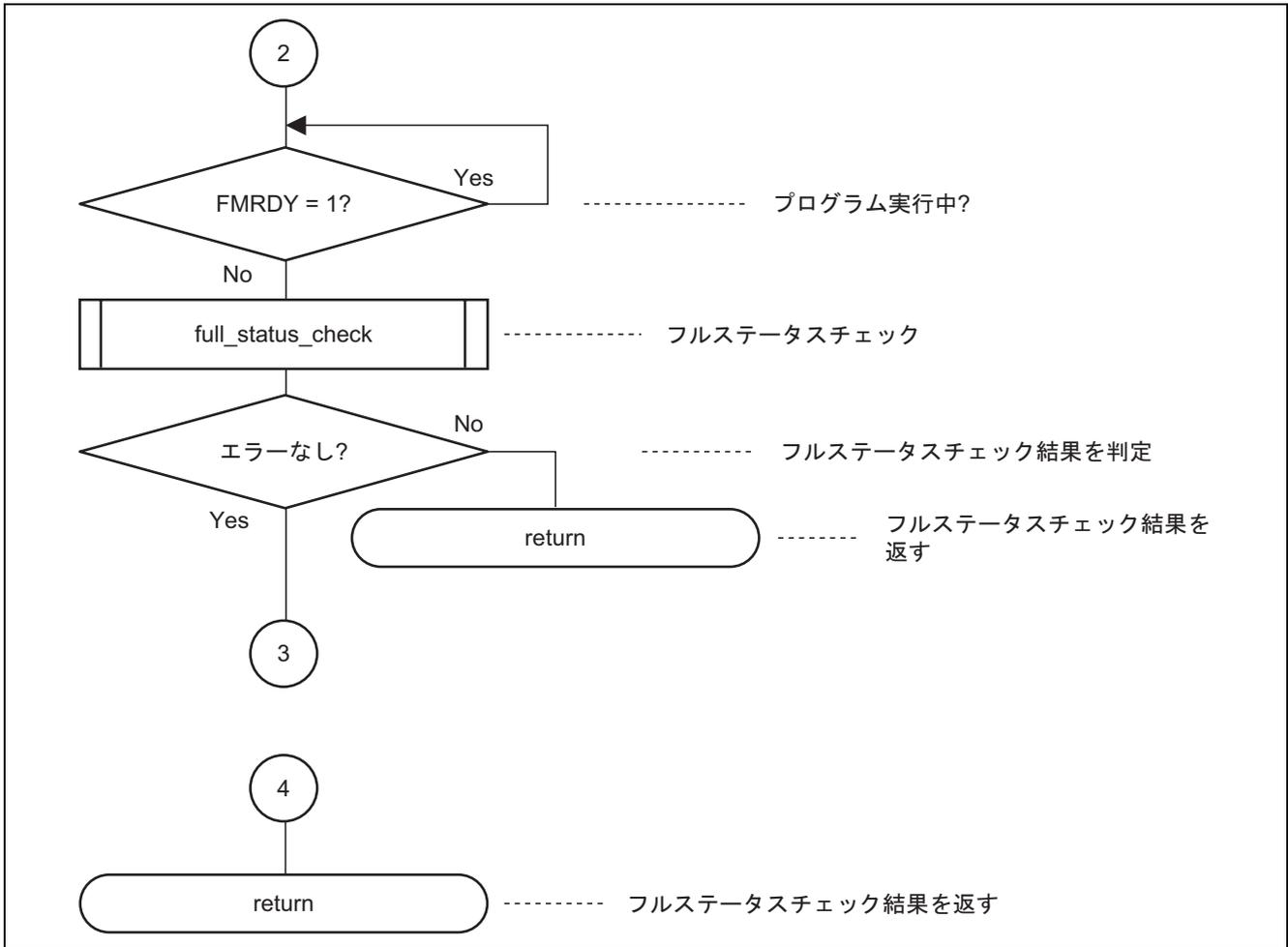


5.3 イレーズルーチン

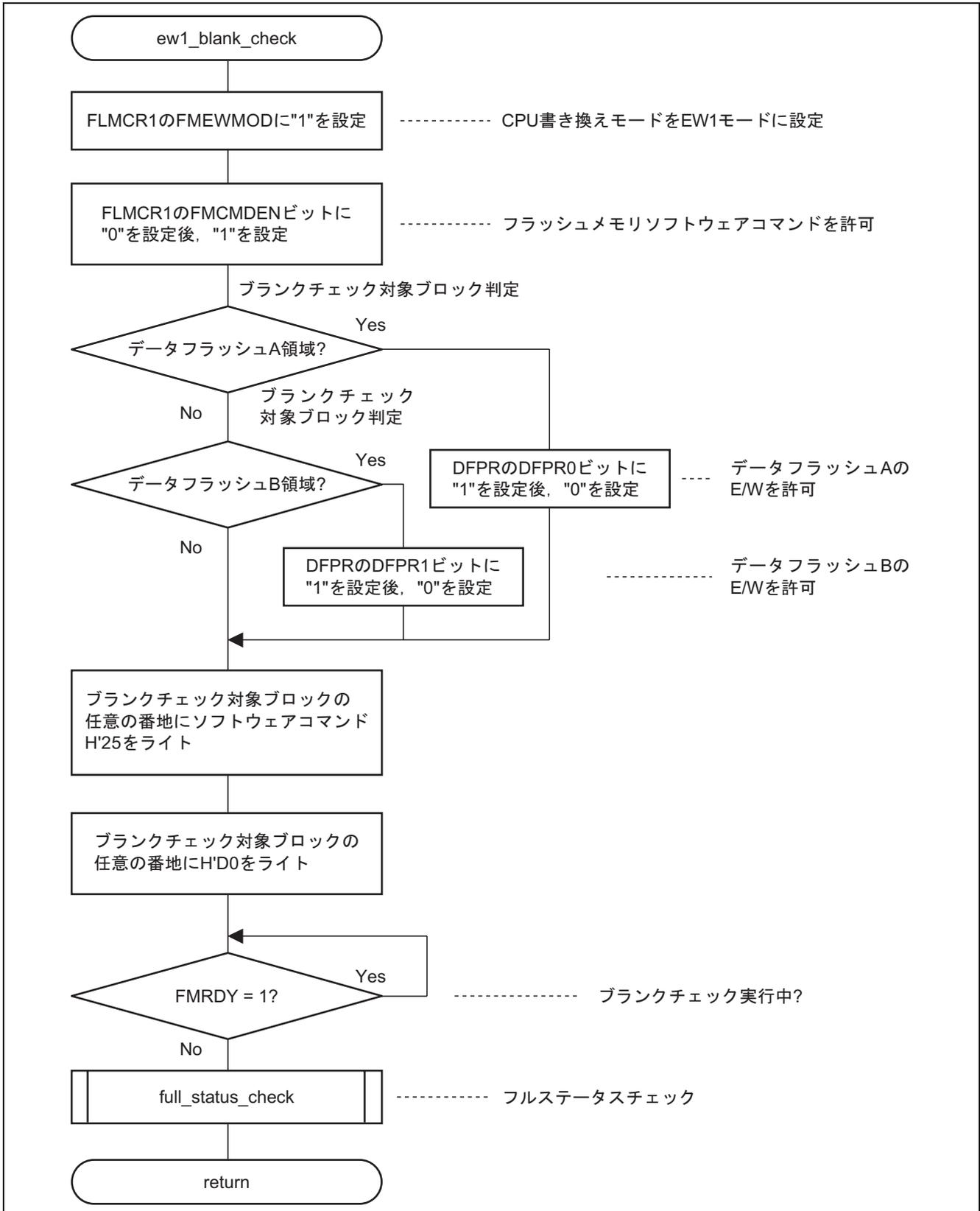


5.4 プログラムルーチン

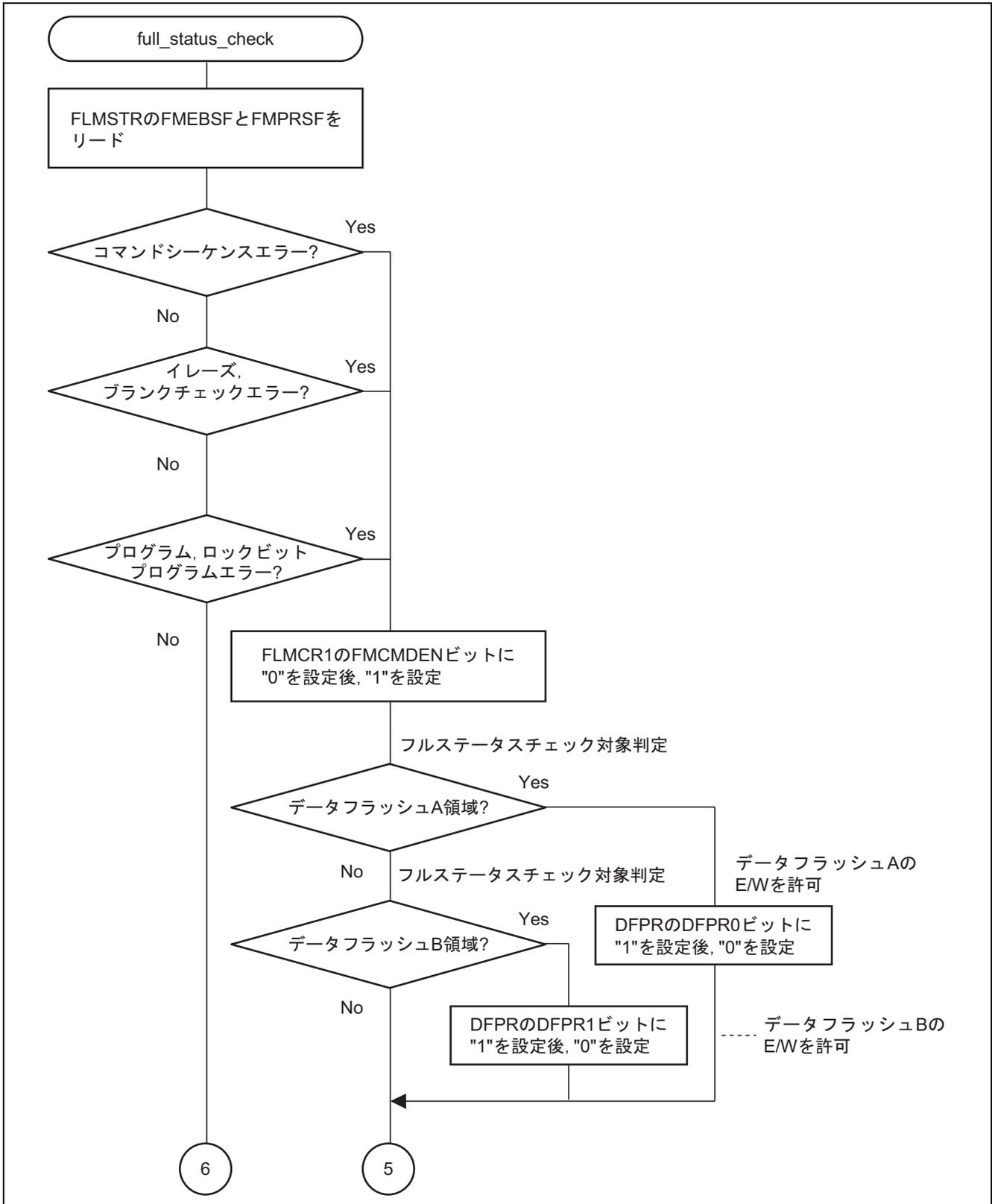


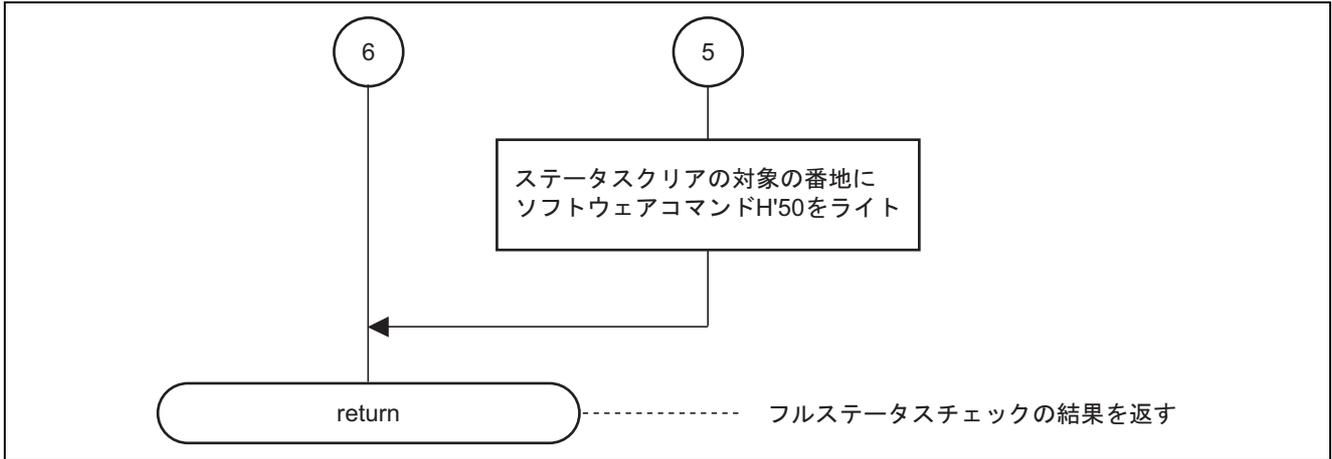


5.5 ブランクチェックルーチン



5.6 フルステータスチェックルーチン





6. プログラムリスト

```

/*****/
/* H8S/2000 Tiny Series -H8S/20203- */
/* Application Note */
/* */
/* data flash read and write */
/* */
/* Function : data flash read and write ( EW1 mode ) */
/* */
/* External Clock : 20MHz */
/* Internal Clock : 20MHz */
/*****/
#include <machine.h>
#include "iodefine.h"

#define FULL_STATUS          0x28      /* mask FLMSTR of FMEBSF,FMPSRF */
#define COMMAND_SEQUENCE_ERR 0x28      /* FMEBSF=1, FMPSRF=1 */
#define ERASE_BLANK_ERR      0x20      /* FMEBSF=1, FMPSRF=0 */
#define PRG_LOCKBIT_ERR      0x08      /* FMEBSF=0, FMPSRF=1 */
#define NO_ERR                0x00      /* FMEBSF=0, FMPSRF=0 */

#define WRITE_SIZE           0x80      /* data size written in data flash A */
#define BACK_UP_AREA        (volatile unsigned char *)0xFFDF80

/* Data Flash block area */
#define FLASH_BLK_A          (volatile unsigned char *)0xF00000
#define FLASH_BLK_B          (volatile unsigned char *)0xF01000
#define FLASH_BLK_B_END     (volatile unsigned char *)0xF01FFF

/*****/
/*Declaration of function prototype */
/*****/
void main(void);
unsigned char ewl_write( volatile unsigned char *wr_top,
                        volatile unsigned char *wr_end, volatile unsigned char *wr_data );
unsigned char ewl_erase( unsigned char *er_blk );
unsigned char ewl_blank_check( unsigned char *blank_blk );
unsigned char full_status_check( unsigned char *adrs );
void h8s_sysinit(void);

```

```

/*****/
/*Name:      main          */
/*Parameters: None        */
/*Returns:   None         */
/*Description: User main  */
/*****/
void main(void)
{
    unsigned char ii, chk;
    volatile unsigned char *df_p, *ram_p;

    set_ccr(0x80);                /* set CCR-Ibit */

    h8s_sysinit();                /* initialize system */

    /* back up data of data flash A(4KB) */
    for ( df_p=FLASH_BLK_A, ram_p=BACK_UP_AREA; df_p<FLASH_BLK_B; df_p++, ram_p++ ){
        (*ram_p) = (*df_p);
    }

    /* EW1 erase of data flash A */
    chk = ewl_erase( FLASH_BLK_A );

    /* blank check data flash A */
    chk = ewl_blank_check( FLASH_BLK_A );

    /* create write data of data flash */
    for ( ii=0, ram_p=BACK_UP_AREA; ram_p<(BACK_UP_AREA+WRITE_SIZE); ii++, ram_p++ ){
        (*ram_p) = ii;
    }

    /* EW1 write of data flash */
    chk = ewl_write( FLASH_BLK_A, FLASH_BLK_B, BACK_UP_AREA );

    while(1);
}

```

```

/*****/
/*Name:          ewl_write          */
/*Parameters:    (wr_top)address of write top      */
/*              (wr_end)address of write end      */
/*Returns:       write result          */
/*Description:   data flash write program        */
/*              of EW1 mode            */
/*****/
unsigned char ewl_write( volatile unsigned char *wr_top,
                        volatile unsigned char *wr_end, volatile unsigned char *wr_data )
{
    volatile unsigned char *ptr;
    unsigned char result;
    unsigned char ii;

    FLASH.FLMCR1.BIT.FMWUS = 0;                /* byte write */

    FLASH.FLMCR1.BIT.FMEWMOD = 1;              /* select EW1 mode */

    for( ptr=wr_top; ptr<wr_end; ptr+=4 ){
        FLASH.FLMCR1.BIT.FMCM DEN = 0;
        FLASH.FLMCR1.BIT.FMCM DEN = 1;          /* flash memory software command enable */

        /* Data Flash A ? */
        if ( (FLASH_BLK_A <= ptr) && (ptr < FLASH_BLK_B) ){
            FLASH.DFPR.BIT.DFPR0 = 1;          /* E/W enable of Data Flash A */
            FLASH.DFPR.BIT.DFPR0 = 0;          /* E/W enable of Data Flash A */
        }
        /* Data Flash B ? */
        else if ( (FLASH_BLK_B <= ptr) && (ptr <= FLASH_BLK_B_END) ){
            FLASH.DFPR.BIT.DFPR1 = 1;          /* E/W enable of Data Flash B */
            FLASH.DFPR.BIT.DFPR1 = 0;          /* E/W enable of Data Flash B */
        }

        (*ptr) = 0x41;                          /* software command 0x41 */

        /* 4byte write */
        for( ii=0; ii<4; ii++, wr_data++){
            (*ptr) = (*wr_data);
        }

        while( FLASH.FLMSTR.BIT.FMRDY != 1 );    /* write complete ? */

        result = full_status_check(ptr);         /* full status check */

        if ( result != NO_ERR ){
            return;
        }
    }

    return (result);
}

```

```

/*****/
/*Name:          ewl_erase          */
/*Parameters:    (er_blk)address of erase BLOCK  */
/*Returns:       erase result        */
/*Description:   data flash erase program        */
/*              of EW1 mode          */
/*****/
unsigned char ewl_erase( unsigned char *er_blk )
{
    unsigned char result;

    FLASH.FLMCR1.BIT.FMEWMOD = 1;          /* select EW1 mode */

    FLASH.FLMCR1.BIT.FMCM DEN = 0;
    FLASH.FLMCR1.BIT.FMCM DEN = 1;          /* flash memory software command enable */

    /* Data Flash A ? */
    if ( (FLASH_BLK_A <= er_blk) && (er_blk < FLASH_BLK_B) ){
        FLASH.DFPR.BIT.DFPR0 = 1;          /* E/W enable of Data Flash A */
        FLASH.DFPR.BIT.DFPR0 = 0;          /* E/W enable of Data Flash A */
    }
    /* Data Flash B ? */
    else if ( (FLASH_BLK_B <= er_blk) && (er_blk <= FLASH_BLK_B_END) ){
        FLASH.DFPR.BIT.DFPR1 = 1;          /* E/W enable of Data Flash B */
        FLASH.DFPR.BIT.DFPR1 = 0;          /* E/W enable of Data Flash B */
    }

    (*er_blk) = 0x20;                       /* write software command H'20 */
    (*er_blk) = 0xD0;                       /* write software command H'D0 */

    while( FLASH.FLMSTR.BIT.FMRDY != 1 );    /* erase complete ? */

    result = full_status_check(er_blk);      /* full status check */

    return (result);
}

```

```

/*****/
/*Name:          ewl_blank_check          */
/*Parameters:    (blk_top)address of blank check    */
/*Returns:       blank check result          */
/*Description:   blank check program          */
/*****/
unsigned char ewl_blank_check( unsigned char *blank_blk )
{
    unsigned char result;

    FLASH.FLMCR1.BIT.FMEWMOD = 1;          /* select EW1 mode */

    FLASH.FLMCR1.BIT.FMCM DEN = 0;
    FLASH.FLMCR1.BIT.FMCM DEN = 1;          /* flash memory software command enable */

    /* Data Flash A ? */
    if ( (FLASH_BLK_A <= blank_blk) && (blank_blk < FLASH_BLK_B) ){
        FLASH.DFPR.BIT.DFPR0 = 1;          /* E/W enable of Data Flash A */
        FLASH.DFPR.BIT.DFPR0 = 0;          /* E/W enable of Data Flash A */
    }
    /* Data Flash B ? */
    else if ( (FLASH_BLK_B <= blank_blk) && (blank_blk <= FLASH_BLK_B_END) ){
        FLASH.DFPR.BIT.DFPR1 = 1;          /* E/W enable of Data Flash B */
        FLASH.DFPR.BIT.DFPR1 = 0;          /* E/W enable of Data Flash B */
    }

    (*blank_blk) = 0x25;                    /* blank check software command H'25 */
    (*blank_blk) = 0xD0;                    /* blank check software command H'D0 */

    while( FLASH.FLMSTR.BIT.FMRDY != 1 );  /* blank check complete ? */

    result = full_status_check(blank_blk);  /* full status check */

    return (result);
}

```

```

/*****/
/*Name:          full_status_check          */
/*Parameters:    (addr)E/W block address    */
/*Returns:       full status check result   */
/*Description:   full status check         */
/*****/
unsigned char full_status_check( unsigned char *addr )
{
    unsigned char tmp_flmstr;

    /* Full status check */
    tmp_flmstr = FLASH.FLMSTR.BYTE & FULL_STATUS; /* read FLMSTR */

    switch ( tmp_flmstr ){
        case COMMAND_SEQUENCE_ERR: /* command sequence error */
        case ERASE_BLANK_ERR: /* erase or blank check error */
        case PRG_LOCKBIT_ERR: /* program or lock bit program error */

            /* error processing */
            FLASH.FLMCR1.BIT.FMCM DEN = 0;
            FLASH.FLMCR1.BIT.FMCM DEN = 1; /* flash memory software command enable */

            /* Data Flash A ? */
            if ( (FLASH_BLK_A <= addr) && (addr < FLASH_BLK_B) ){
                FLASH.DFPR.BIT.DFPR0 = 1; /* E/W enable of Data Flash A */
                FLASH.DFPR.BIT.DFPR0 = 0; /* E/W enable of Data Flash A */
            }
            /* Data Flash B ? */
            else if ( (FLASH_BLK_B <= addr) && (addr <= FLASH_BLK_B_END) ){
                FLASH.DFPR.BIT.DFPR1 = 1; /* E/W enable of Data Flash B */
                FLASH.DFPR.BIT.DFPR1 = 0; /* E/W enable of Data Flash B */
            }

            (*addr) = 0x50; /* Clear status command */

            break;

        default : /* No error */

            break;
    }

    return (tmp_flmstr);
}

```

```

/*****/
/*Name:      h8s_sysinit      */
/*Parameters: None          */
/*Returns:   None           */
/*Description: initialize H8S/20203 */
/*****/
void h8s_sysinit(void)
{
    MSTCR1.BIT.MSTWDT = 0;          /* WDT module standby off */

    /* stop WDT */
    WDT.TCSRWD.BYTE = 0x97;        /* write enable TMWLOCK, TMWI */
    WDT.TCSRWD.BYTE = 0xA3;        /* write enable TMWD */
    WDT.TMWD.BYTE = 0xF7;         /* Not select clock source */
    WDT.TMWD.BYTE = 0xF8;         /* write bit inversion */
    WDT.TCSRWD.BYTE = 0x87;        /* write disable TMWLOCK, TMWI */

    CPG.OSCCSR.BYTE = 0x0E;        /* wait over 6.5ms, Phi_osc=20MHz */
    PMRJ.BYTE = 0x03;             /* select OSC1,OSC2 */

    CPG.SYSCCR.BYTE = (CPG.SYSCCR.BYTE & 0x7F) | 0x40; /* WI=0, WE=1 */
    CPG.SYSCCR.BYTE = 0x60;        /* high=Phi_osc, Phi_low=Phi_loco */
    CPG.SYSCCR.BYTE = CPG.SYSCCR.BYTE & 0x3F; /* WI=0, WE=0 */

    CPG.LPCR1.BYTE = (CPG.LPCR1.BYTE & 0x7F) | 0x40; /* WI=0, WE=1 */
    CPG.LPCR1.BYTE = 0x41;        /* PSC on, Phi_base=Phi_high */
    CPG.LPCR1.BYTE = CPG.LPCR1.BYTE & 0x3F; /* WI=0, WE=0 */

    CPG.LPCR2.BYTE = (CPG.LPCR2.BYTE & 0x7F) | 0x40; /* WI=0, WE=1 */
    CPG.LPCR2.BYTE = 0x40;        /* select system clock */
    CPG.LPCR2.BYTE = CPG.LPCR2.BYTE & 0x3F; /* WI=0, WE=0 */

    CPG.LPCR3.BYTE = (CPG.LPCR3.BYTE & 0x7F) | 0x40; /* WI=0, WE=1 */
    CPG.LPCR3.BYTE = 0x40;        /* select clock of bus master */
    CPG.LPCR3.BYTE = CPG.LPCR3.BYTE & 0x3F; /* WI=0, WE=0 */
}

```

6.1 リンクアドレス指定

セクション名	アドレス
PRResetPRG, PIntPRG	H'000400
P, C\$DSEC, C\$BSEC, D	H'000800
B, R	H'FFEF80
S	H'FFFD80

ホームページとサポート窓口

ルネサス テクノロジホームページ

<http://japan.renesas.com/>

お問合せ先

<http://japan.renesas.com/inquiry>

csc@renesas.com

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2008.10.31	—	初版発行

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。

本資料ご利用に際しての留意事項

1. 本資料は、お客様に用途に応じた適切な弊社製品をご購入いただくための参考資料であり、本資料中に記載の技術情報について弊社または第三者の知的財産権その他の権利の実施、使用を許諾または保証するものではありません。
2. 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他応用回路例など全ての情報の使用に起因する損害、第三者の知的財産権その他の権利に対する侵害に関し、弊社は責任を負いません。
3. 本資料に記載の製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的、あるいはその他軍事情報の目的で使用しないでください。また、輸出に際しては、「外国為替および外国貿易法」その他輸出関連法令を遵守し、それらの定めるところにより必要な手続を行ってください。
4. 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他応用回路例などの全ての情報は本資料発行時点のものであり、弊社は本資料に記載した製品または仕様等を予告なしに変更することがあります。弊社の半導体製品のご購入およびご使用に当たりますと、事前に弊社営業窓口で最新の情報をご確認いただきますとともに、弊社ホームページ (<http://www.renesas.com>) などを通じて公開される情報に常にご注意ください。
5. 本資料に記載した情報は、正確を期すため慎重に制作したものです。万一本資料の記述の誤りに起因する損害がお客様に生じた場合においても、弊社はその責任を負いません。
6. 本資料に記載の製品データ、図、表などに示す技術的な内容、プログラム、アルゴリズムその他応用回路例などの情報を流用する場合は、流用する情報を単独で評価するだけでなく、システム全体で十分に評価し、お客様の責任において適用可否を判断してください。弊社は、適用可否に対する責任を負いません。
7. 本資料に記載された製品は、各種安全装置や運輸・交通用、医療用、燃焼制御用、航空宇宙用、原子力、海底中継用の機器・システムなど、その故障や誤動作が直接人命を脅かしあるいは人体に危害を及ぼすおそれのあるような機器・システムや特に高度な品質・信頼性が要求される機器・システムでの使用を意図して設計、製造されたものではありません（弊社が自動車用と指定する製品を自動車に使用する場合を除きます）。これらの用途に利用されることをご検討の際には、必ず事前に弊社営業窓口へご照会ください。なお、上記用途に使用されたことにより発生した損害等については弊社はその責任を負いかねますのでご了承願います。
8. 第7項にかかわらず、本資料に記載された製品は、下記の用途には使用しないでください。これらの用途に使用されたことにより発生した損害等につきましては、弊社は一切の責任を負いません。
 - 1) 生命維持装置。
 - 2) 人体に埋め込み使用するもの。
 - 3) 治療行為（患部切り出し、薬剤投与等）を行うもの。
 - 4) その他、直接人命に影響を与えるもの。
9. 本資料に記載された製品のご使用につき、特に最大定格、動作電源電圧範囲、放熱特性、実装条件およびその他諸条件につきましては、弊社保証範囲内でご使用ください。弊社保証値を越えて製品をご使用された場合の故障および事故につきましては、弊社はその責任を負いません。
10. 弊社は製品の品質および信頼性の向上に努めておりますが、特に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。弊社製品の故障または誤動作が生じた場合も人身事故、火災事故、社会的損害など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、誤動作防止設計などの安全設計（含むハードウェアおよびソフトウェア）およびエージング処理等、機器またはシステムとしての出荷保証をお願いいたします。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、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
11. 本資料に記載の製品は、これを搭載した製品から剥がれた場合、幼児が口に入れて誤飲する等の事故の危険性があります。お客様の製品への実装後に容易に本製品が剥がれることがなきよう、お客様の責任において十分な安全設計をお願いします。お客様の製品から剥がれた場合の事故につきましては、弊社はその責任を負いません。
12. 本資料の全部または一部を弊社の文書による事前の承諾なしに転載または複製することを固くお断りいたします。
13. 本資料に関する詳細についてのお問い合わせ、その他お気付きの点等がございましたら弊社営業窓口までご照会ください。

D039444