

R32C/100 シリーズ

RJJ05B1268-0100

CPU書き換えモード(EW0モード)を使った
内蔵フラッシュメモリ書き換え

Rev.1.00

2010.05.28

1. 要約

この資料では、CPU書き換えモードのEW0モードを使った内蔵フラッシュメモリ書き換えの使用例を紹介しします。

2. はじめに

この資料で説明する応用例は、次のマイコンに適用されます。
マイコン: R32C/118グループ

R32C/118グループと同様のSFR(周辺機能制御レジスタ)を持つ他のR32C/100シリーズでも本プログラムを使用することができます。ただし、一部の機能を追加等で変更している場合がありますのでユーザーズマニュアルで確認してください。このアプリケーションノートのご使用に際しては十分な評価を行ってください。

3. 設定手順

3.1 CPU書き換えモード

CPU書き換えモードは、CPUがソフトウェアコマンドを実行することによって、フラッシュメモリを書き換えるモードです。CPU書き換えモードでは、CPUバスから直接フラッシュメモリをアクセスせず、フラッシュメモリ書き換え専用のバスを経由してフラッシュメモリにアクセスします。

図 3.1にCPU書き換えモード時のフラッシュメモリアクセスバスを示します。

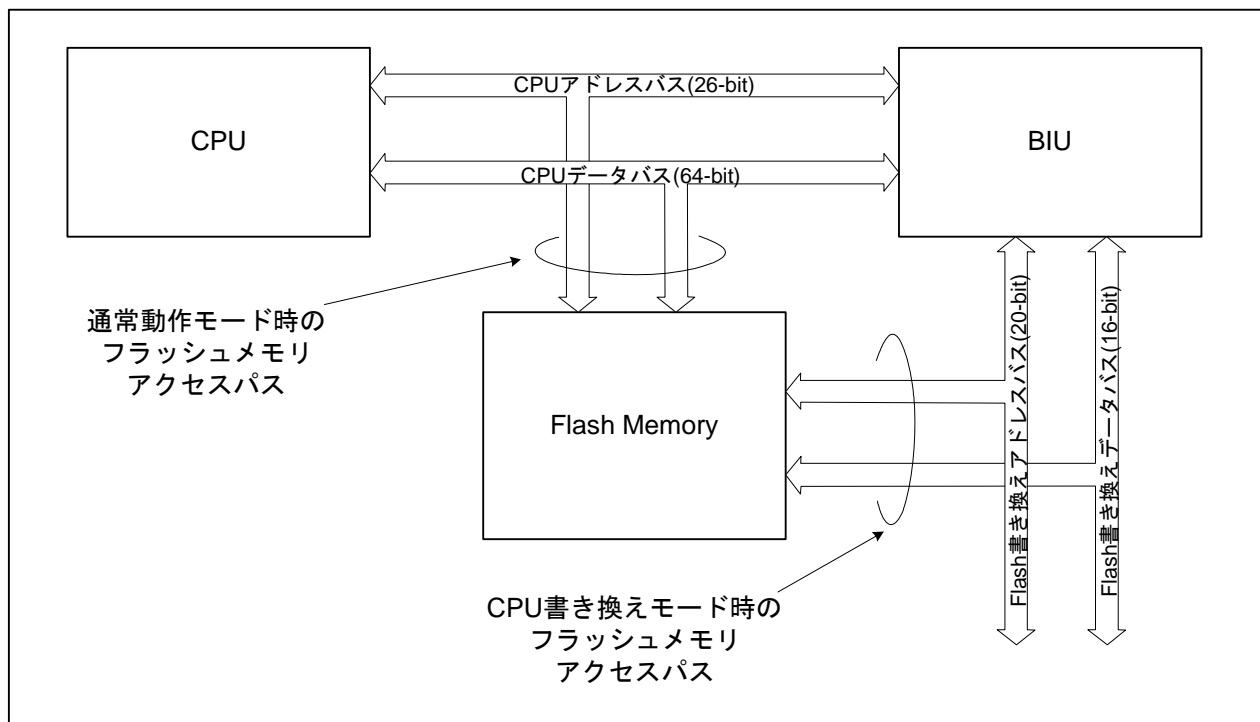


図 3.1 CPU書き換えモード時のフラッシュメモリアクセスバス

3.2 EW0モードの特徴

EW0モードでは、あらかじめRAM上に転送したCPU書き換えプログラムから、プログラムコマンド、イレーズコマンドを発行することで、ユーザROM領域とデータ領域を書き換えることができます。

EW0モードでは、プログラム中、イレーズ中でもCPUは動作しているため、周辺機能割り込みはベクタテーブルと割り込みプログラムをRAM上に配置することで、プログラム中、イレーズ中に割り込みを受け付けることができます。表3.1にEW0モードの特徴を示します。

表 3.1 EW0モードの特徴

項目	内容
CPU動作モード	シングルチップモード メモリ拡張モード
書き換えプログラムを実行できる領域	内蔵フラッシュメモリ以外の領域
プログラム、イレーズ後のモード	リードステータスレジスタモード
プログラム、イレーズ中のCPUの状態	動作
フラッシュメモリのステータス検知	・プログラムでFMSR0レジスタを読む ・リードステータスレジスタコマンドを実行し、データを読む

3.3 フラッシュメモリ書き換えバスタイミング設定

フラッシュメモリ書き換えのためのバス設定はFEBC0、FEBC3レジスタで行います。FEBC0、FEBC3レジスタは、フラッシュメモリ書き換えバスタイミング設定およびフラッシュメモリCPU書き換えモードタイミングに関する電気的特性を参照して、書き換え条件を満たすように設定してください。

EBC0、EBC3レジスタとFEBC0、FEBC3レジスタは同じレジスタを共用していますので、フラッシュメモリ書き換え後はEBC0、EBC3レジスタを再設定してください。

以下にフラッシュメモリ書き換えバスタイミングの算出例を示します。

(1) クロック条件

EW0モード遷移前は、シングルチップモード、メインクロック逡倍モードで動作しているものとします。表3.2にメインクロック逡倍モード遷移後のクロック条件、表3.3にクロック関連レジスタの設定値を示します。

表 3.2 クロック条件

クロック名	周波数
メインクロック	16MHz
PLLクロック	100MHz
ベースクロック	50MHz
CPUクロック	50MHz
周辺バスクロック	25MHz
周辺機能クロック源	25MHz

表 3.3 クロック関連レジスタの設定値

レジスタ名	設定値	備考
PLC0	04h	詳細は「ユーザズマニュアル」を参照してください
PLC1	03h	同上
PM3	40h	PM36~PM35ビット=“10b”(周辺機能クロック源 4分周)
CCR	1Fh	BCD1~BCD0ビット=“11b”(ベースクロック 2分周) CCD1~CCD0ビット=“11b”(CPUクロック 分周なし) PCD1~PCD0ビット=“01b”(周辺バスクロック 2分周) BCSビット=“0”(ベースクロック源 PLLクロック選択)
PBC	0504h	詳細は「ユーザズマニュアル」を参照してください
EBC0	0000h	シングルチップモード動作のためリセット後の値のまま

(2) フラッシュメモリ条件

表 3.4 にフラッシュメモリの規格値を示します。この規格値は、変更になる可能性があります。ユーザーマニュアルの電気的特性を参照してください。

表 3.4 フラッシュメモリCPU書き換えモードタイミング

記号	項目	規格値		単位
		最小	最大	
tcR	リードサイクル時間	200		ns
tsu(S-R)	リード前チップセレクトセットアップ時間	200		ns
th(R-S)	リード後チップセレクトホールド時間	0		ns
tsu(A-R)	リード前アドレスセットアップ時間	200		ns
th(R-A)	リード後アドレスホールド時間	0		ns
tw(R)	リードパルス幅	100		ns
tcW	ライトサイクル時間	200		ns
tsu(S-W)	ライト前チップセレクトセットアップ時間	0		ns
th(W-S)	ライト後チップセレクトホールド時間	30		ns
tsu(A-W)	ライト前アドレスセットアップ時間	0		ns
th(W-A)	ライト後アドレスホールド時間	30		ns
tw(W)	ライトパルス幅	50		ns

(3) 必要サイクル数算出

設定値検討にあたり、クロック条件およびフラッシュメモリ条件からフラッシュメモリアクセスに必要なサイクル数を算出します。

バスタイミングの基準となるクロックは、CCRレジスタのBCD1~BCD0ビットで設定されるベースクロックです。表 3.2 に示すクロック条件でベースクロックは50MHzですので、1サイクル当たり20nsになります。このサイクル時間を元に表 3.4 のフラッシュメモリ条件をサイクル数単位に換算した結果を表 3.5 に示します。

表 3.5 必要サイクル数条件

記号	規格値		単位	サイクル数		単位
	最小	最大		最小	最大	
tcR	200		ns	10		サイクル
tsu(S-R)	200		ns	10		サイクル
th(R-S)	0		ns	0		サイクル
tsu(A-R)	200		ns	10		サイクル
th(R-A)	0		ns	0		サイクル
tw(R)	100		ns	5		サイクル
tcW	200		ns	10		サイクル
tsu(S-W)	0		ns	0		サイクル
th(W-S)	30		ns	1.5		サイクル
tsu(A-W)	0		ns	0		サイクル
th(W-A)	30		ns	1.5		サイクル
tw(W)	50		ns	2.5		サイクル

(4) バスタイミング選択

表 3.3の周辺バスクロック及び表 3.5を元にバスタイミングを決定します。

バスタイミングの決定には、周辺バスクロックにより決まるMPY1~MPY0ビットの値及びフラッシュメモリのリードサイクル、ライトサイクルが必要になります。

表 3.6に周辺バスクロック2分周時におけるMPY1~MPY0ビット、FWR4~FWR0ビットとリードサイクルの関係を示します。

また、表 3.7に周辺バスクロック2分周時におけるMPY1~MPY0ビット、FSUW1~FSUW0ビット、FWW1~FWW0ビットとライトサイクルの関係を示します。ここで、網掛け部分はクロック条件、サイクル数の条件を共に満たす箇所を示します。

なお、周辺バスクロックが3分周、4分周の場合については、ユーザーズマニュアルを参照してください。

表 3.6 周辺バスクロック2分周時におけるMPY1~MPY0ビット、FWR4~FWR0ビットとリードサイクルの関係
(単位：サイクル)

FWR3~FWR0 ビットの設定値		FWR 4 ビット の設定値	MPY1~MPY0ビットの設定値							
			10b				11b			
			mpy = 3				mpy = 4			
			tsu(S-R) tsu(A-R)	tw(R)	tcR	th(R-S) th(R-A)	tsu(S-R) tsu(A-R)	tw(R)	tcR	th(R-S) th(R-A)
0000b	wr=1	0	4	3	4	0	6	5	6	0
		1	6	5	6	0	6	5	6	0
0001b	wr=2	0	8	7	8	0	10	9	10	0
		1	8	7	8	0	10	9	10	0
0101b	wr=3	0	10	9	10	0	14	13	14	0
		1	12	11	12	0	14	13	14	0
0110b	wr=4	0	14	13	14	0	18	17	18	0
		1	14	13	14	0	18	17	18	0
1010b	wr=5	0	16	15	16	0	22	21	22	0
		1	18	17	18	0	22	21	22	0
1011b	wr=6	0	20	19	20	0	26	25	26	0
		1	20	19	20	0	26	25	26	0
1111b	wr=7	0	22	21	22	0	30	29	30	0
		1	24	23	24	0	30	29	30	0

表 3.7 周辺バスクロック2分周時におけるMPY1~MPY0ビット、FSUW1~FSUW0ビット、FWW1~FWW0ビットとライトサイクルの関係

(単位：サイクル)

FSUW1~FSUW0 ビットの設定値		FWW1~FWW0 ビットの設定値		MPY1~MPY0ビットの設定値							
				10b				11b			
				mpy = 3				mpy = 4			
				tsu(S-W) tsu(A-W)	tw(W)	tcW	th(W-S) th(W-A)	tsu(S-W) tsu(A-W)	tw(W)	tcW	th(W-S) th(W-A)
00b	suw=0	00b	ww=1	1	3	6	2	1	4	6	1
		01b	ww=2	1	6	8	1	1	8	10	1
		10b	ww=3	1	9	12	2	1	12	14	1
		11b	ww=4	1	12	14	1	1	16	18	1
01b	suw=1	00b	ww=1	4	3	8	1	5	4	10	1
		01b	ww=2	4	6	12	2	5	8	14	1
		10b	ww=3	4	9	14	1	5	12	18	1
		11b	ww=4	4	12	18	2	5	16	22	1
10b	suw=2	00b	ww=1	7	3	12	2	9	4	14	1
		01b	ww=2	7	6	14	1	9	8	18	1
		10b	ww=3	7	9	18	2	9	12	22	1
		11b	ww=4	7	12	20	1	9	16	26	1
11b	suw=3	00b	ww=1	10	3	14	1	13	4	18	1
		01b	ww=2	10	6	18	2	13	8	22	1
		10b	ww=3	10	9	20	1	13	12	26	1
		11b	ww=4	10	12	24	2	13	16	30	1

(5) バスタイミング設定値

表 3.6、表 3.7により、クロック条件とフラッシュメモリ条件を満たし、バスタイミングが最適な設定値は、表 3.8の通りになります。

表 3.8 バスタイミング最適値

項目	設定値	備考
FWR3~FWR0	0101b	wr = 3
FWR4	0	微調整しない
FSUW1~ FSUW0	00b	suw = 0
FWW1~ FWW0	10b	ww = 3
MPY1~ MPY0	10b	mpy = 3 (周辺バスクロック2分周)

表 3.8のバスタイミングをFEBC0レジスタの各ビットに設定するとFEBC0 = 5885hとなります。

図 3.2にFEBC0レジスタの設定内容を示します。

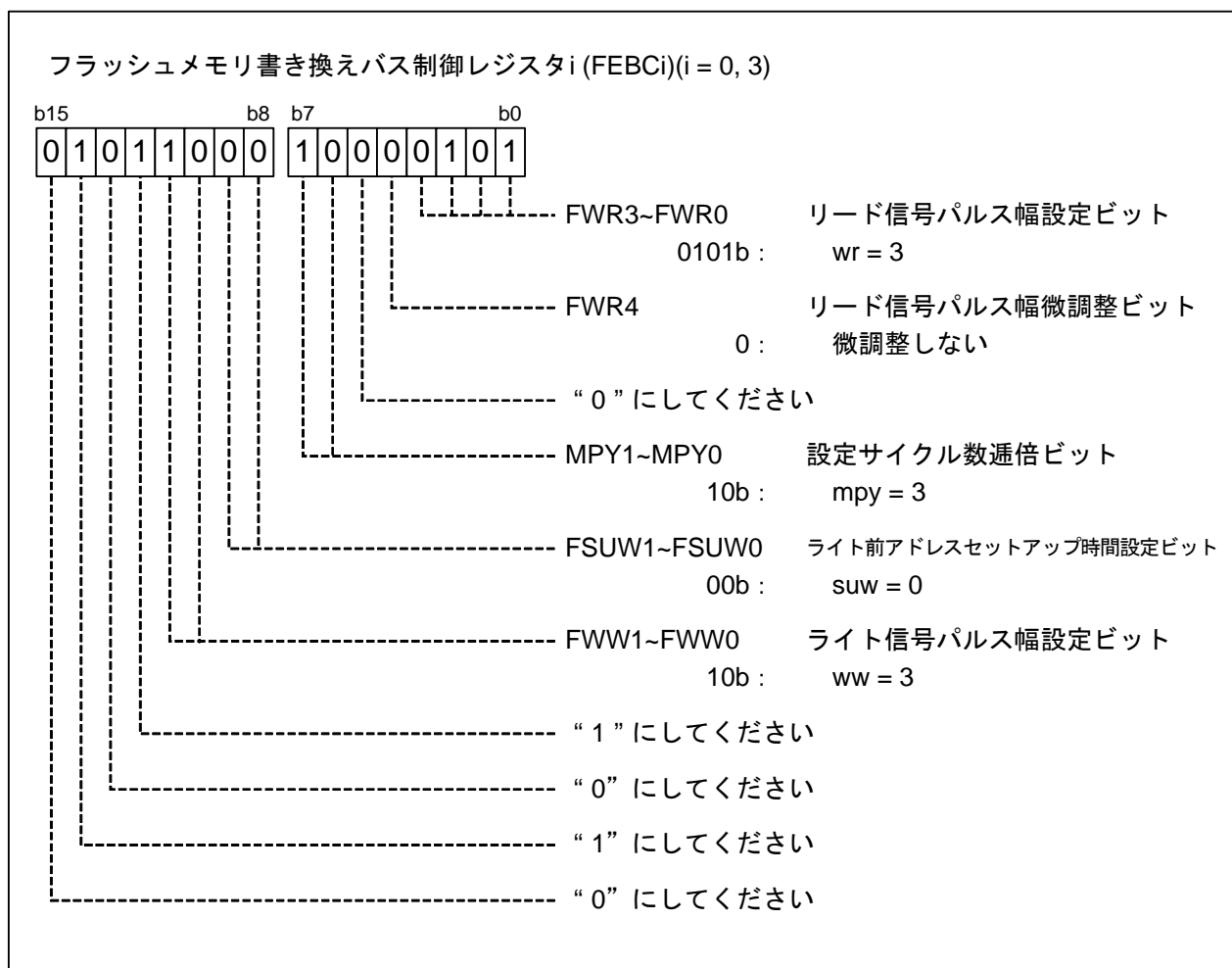


図 3.2 フラッシュメモリ書き換えバス制御レジスタ0(FEBC0)設定値

(6) CPU動作モードとフラッシュメモリ書き換え

フラッシュメモリ書き換えのバスタイミング設定に使用するレジスタはCPUの動作モードによって異なります。

メモリ拡張モードを使用せずシングルチップモードのみを使用する場合、CB01、CB12、CB23レジスタはリセット後の値(“00h”)のまま変更しないでください。このとき、プログラム領域、データ領域ともFEBC0レジスタでバス設定が行えます。

メモリ拡張モードを使用する場合、CB01、CB12、CB23レジスタを各レジスタの設定範囲に従って、値を変更してください。このとき、プログラム領域はFEBC0レジスタ、データ領域はFEBC3レジスタでバス設定が行えます。

また、メモリ拡張モードで使用するEBC0、EBC3レジスタとFEBC0、FEBC3レジスタは同じレジスタを共用していますので、フラッシュメモリ書き換えのためにFEBCiレジスタを変更するとEBCiレジスタの設定値も変化します。このため、CPU書き換えモード中には $\overline{CS0}$ 領域、 $\overline{CS3}$ 領域に配置された外部デバイスにアクセスできなくなる場合があります。

表 3.9にこれらの詳細と制限事項を記載します。

表 3.9 CPU動作モードとフラッシュメモリ書き換え

項目	CPU動作モード	
	シングルチップモード	メモリ拡張モード
CB01レジスタ	00hのまま変更しないで下さい	04h~F8hの範囲で、かつCB12レジスタの設定値より大きい値を設定してください
CB12レジスタ	00hのまま変更しないで下さい	03h~F7hの範囲で、かつCB23レジスタの設定値より大きくCB01レジスタの設定値より小さい値を設定してください
CB23レジスタ	00hのまま変更しないで下さい	02h~F6hの範囲で、かつCB12レジスタの設定値より小さい値を設定してください
プログラム領域のバス設定	FEBC0レジスタ	FEBC0レジスタ
データ領域のバス設定	FEBC0レジスタ	FEBC3レジスタ
FEBCiレジスタ設定後の $\overline{CS0}$ 領域、 $\overline{CS3}$ 領域の状態	—	<ul style="list-style-type: none"> セパレートバス 16ビット幅 RDY信号無視
制限事項	なし	<ul style="list-style-type: none"> HOLD信号は無視されます $\overline{CS0}$領域、$\overline{CS3}$領域をマルチプレクスバスで使用していた場合、CPU書き換えモード時は外部デバイスにアクセスできません $\overline{CS0}$領域、$\overline{CS3}$領域のバスタイミングが変わりますので、外部デバイスにアクセスできなくなる場合があります

3.4 設定手順概略

CPU書き換えモード(EW0モード)の実行フローを図3.3に示します。

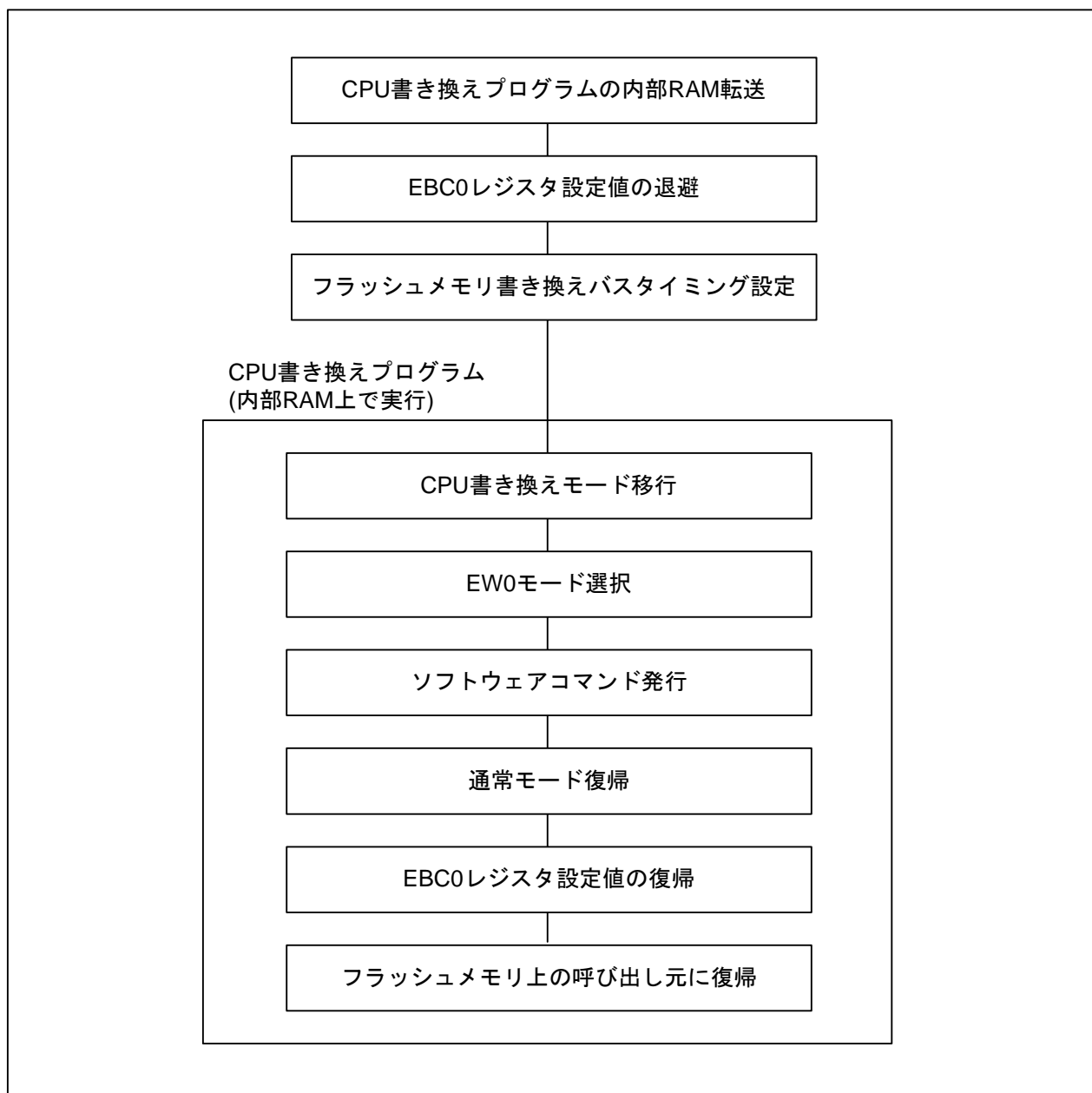


図 3.3 CPU書き換えモード(EW0モード)実行フロー

3.5 設定手順詳細

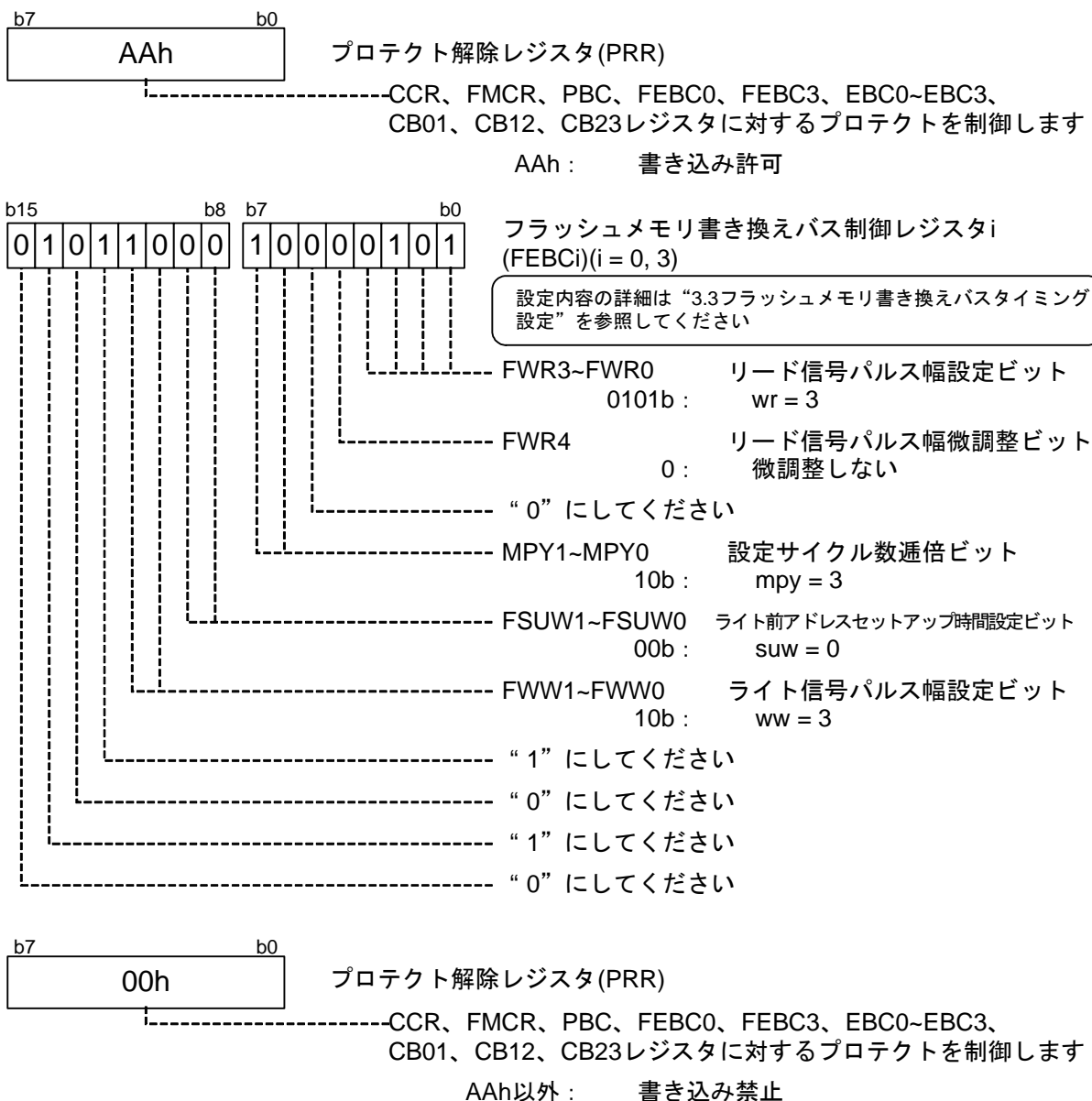
CPU書き換えプログラムのRAM転送

3.6に示す方法でフラッシュメモリ上のCPU書き換えプログラムを内部RAMに転送してください。

EBC0レジスタ設定値の退避

EBC0レジスタとFEBC0レジスタはアドレスを共有しますので、EBC0レジスタの設定値を内部RAMに退避してください。

フラッシュメモリ書き換えバスタイミング設定



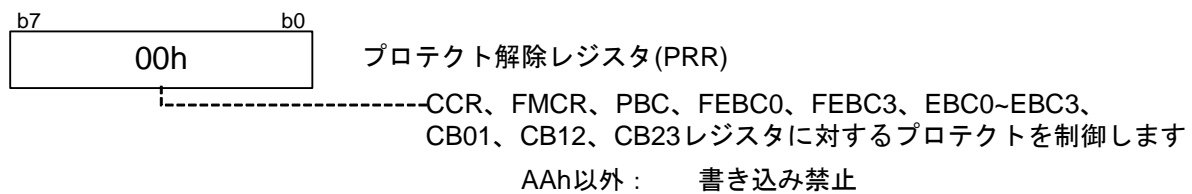
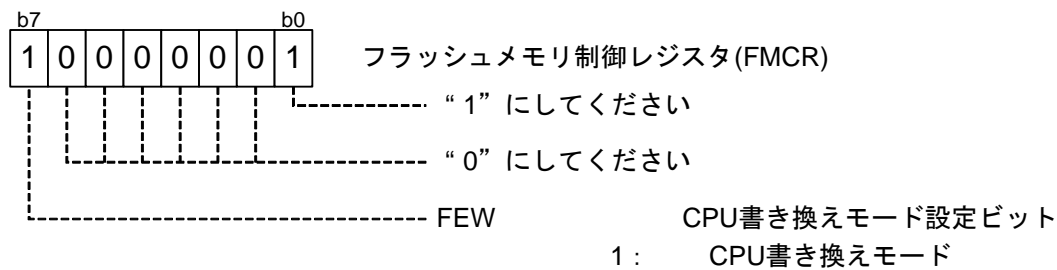
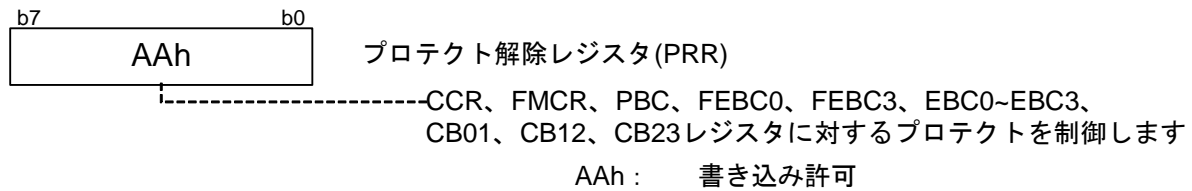
PRRレジスタを“AAh”(書き込み許可)にした後は、必ず“AAh以外”(書き込み禁止)にしてください

次ページへ続く

前ページから

CPU書き換えプログラムの呼び出し
内部RAMに転送したCPU書き換えプログラムを呼び出してください。

CPU書き換えモード移行



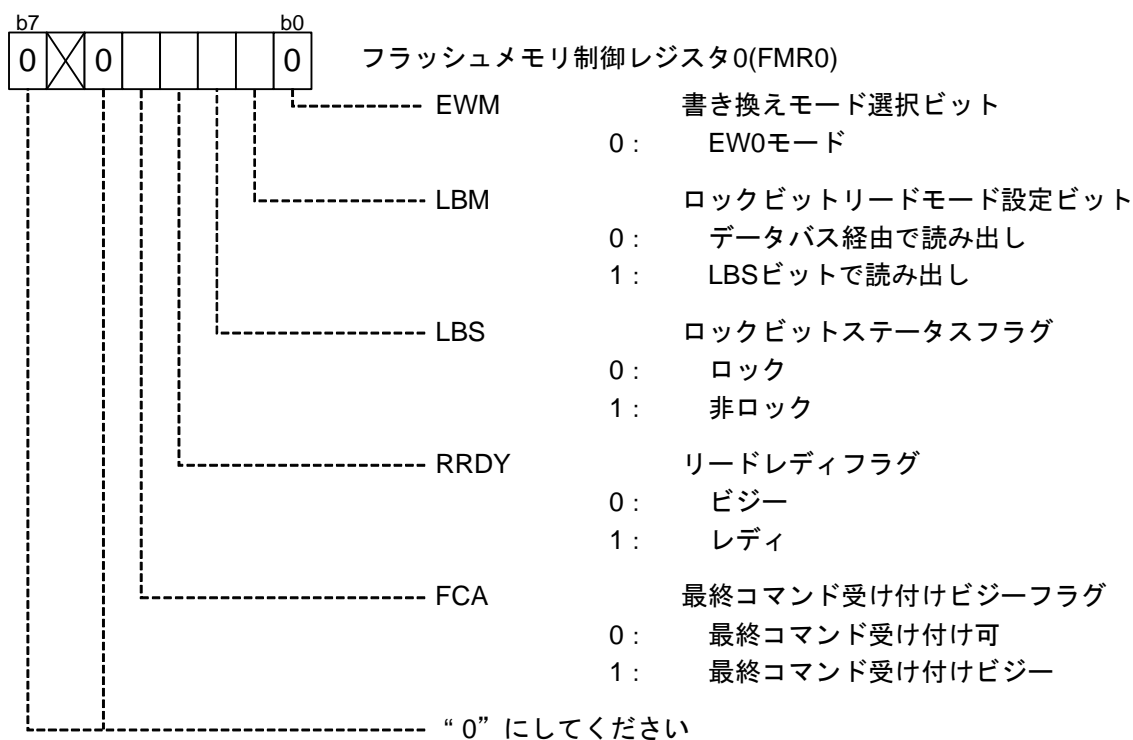
PRRレジスタを“AAh”(書き込み許可)にした後は、必ず“AAh以外”(書き込み禁止)にしてください

次ページへ続く

前ページから

ソフトウェアコマンド発行(第一コマンド)
内蔵フラッシュメモリのソフトウェアコマンド(第一コマンド)を発行してください。

ソフトウェアコマンド発行(第二コマンド以降)
複数のコマンドからなるフラッシュメモリの操作を行う場合、次の処理を行ってください。
最後のコマンドを発行する前に、フラッシュメモリ制御レジスタ0の最終コマンド受け付けビジーフラグが“0”になるまで待ってください。

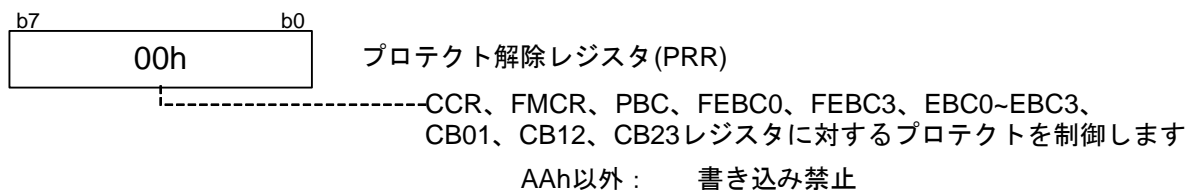
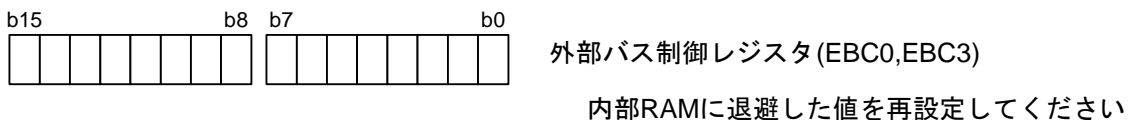
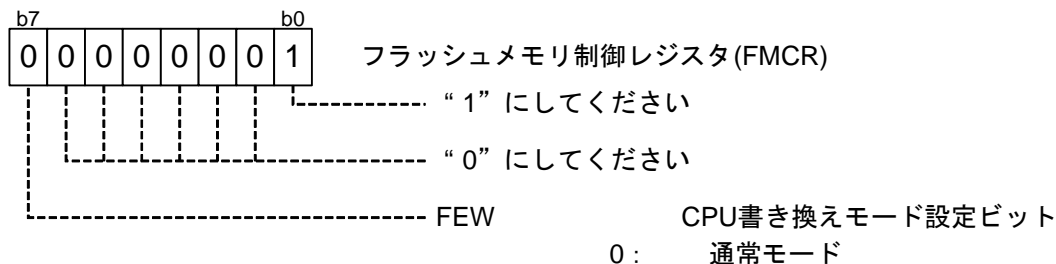
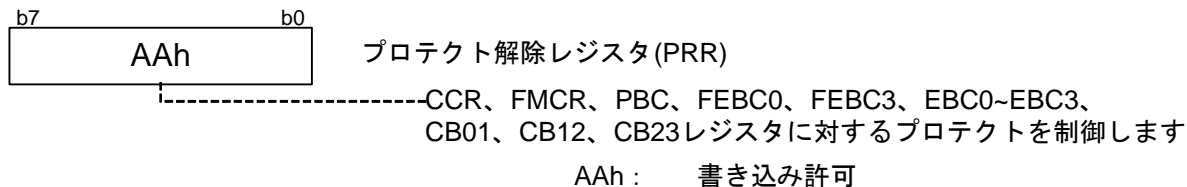


続いて、内蔵フラッシュメモリのソフトウェアコマンド(最終コマンド)を発行してください。

次ページへ続く

前ページから

通常モードに復帰、EBC0レジスタ設定値の復帰



PRRレジスタを“AAh”(書き込み許可)にした後は、必ず“AAh以外”(書き込み禁止)にしてください

フラッシュメモリ上の呼び出し元に復帰

3.6 CPU書き換え制御プログラムのRAMへの転送

CPU書き換え制御プログラムはRAM上で動作させる必要があります。ここでは、FFFD0000h番地に格納されたCPU書き換え制御プログラムをRAM上に転送する例を説明します。図3.4プログラムの転送イメージを示します。

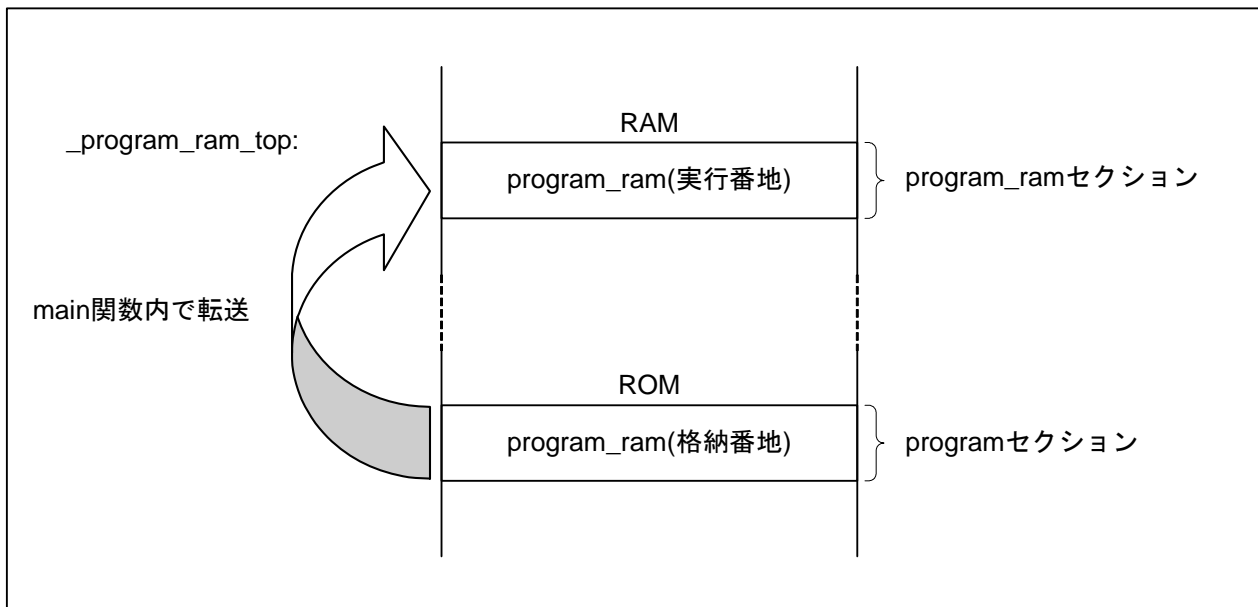


図 3.4 プログラムの転送イメージ

(1) セクション名の変更

セクション名として「program_ram」を追加し、このセクションにRAM上で動作するCPU書き換え制御プログラムを配置します。CPU書き換え制御プログラムを、programセクションからprogram_ramセクションに配置しなおすには、C言語上で下記の通り記述します。

```
void main(void)
{
/* このプログラムはprogramセクション上に配置されます */
}
/* #pragma SECTION宣言以降のプログラムは全てprogram_ramセクション上に配置されます */
#pragma SECTION program program_ram
void ew0_mode_program(void)
{
/* このプログラムはprogram_ramセクション上に配置されます */
}
```

#pragma SECTIONはコンパイラが生成するセクション名を変更する#pragma拡張機能です。

この宣言をprogramセクションに対して行った場合は、その#pragma宣言以降に記述された関数のセクション名を変更します。

(2) CPU書き換え制御プログラムの転送

CPU書き換え制御プログラムをRAMに転送する処理(マクロ定義)を追加します。

```

/*****
 * #pragma declaration *
 *****/
#define pcopy(X,Y) _asm(" .initsct "X",code,align¥n"¥
                        " .initsct "X"_INIT,rom"Y"¥n"¥
                        " mov.l #(topof "X"_INIT),A0¥n"¥
                        " mov.l #(topof "X"),A1¥n"¥
                        " mov.l #sizeof "X",R7R5¥n"¥
                        " smovf.b")

```

追加マクロ定義

さらに、プログラムの先頭でRAMに転送する処理を以下のように記述します。

```

/****FUNC COMMENT****
 * Outline      : Flash Memory Version CPU Rewrite Mode
 *              : (EW0 Mode) Sample
 * Declaration   : void main(void)
 * Description   : This program is the execution sample in CPU rewriting mode.
 * Argument      : none
 * Return Value  : none
 * Calling Functions : SetPLLClock() : Configuring PLL Mode
 *                  : ew0_mode_control() : CPU rewriting program execution
 ****FUNC COMMENT END****/
void main(void)
{
    unsigned char tmp;
    pcopy("program_ram","data,align"); // (1) Transfer rewrite program to RAM
    asm("FCLR I"); // (2) Interruption inhibiton
}

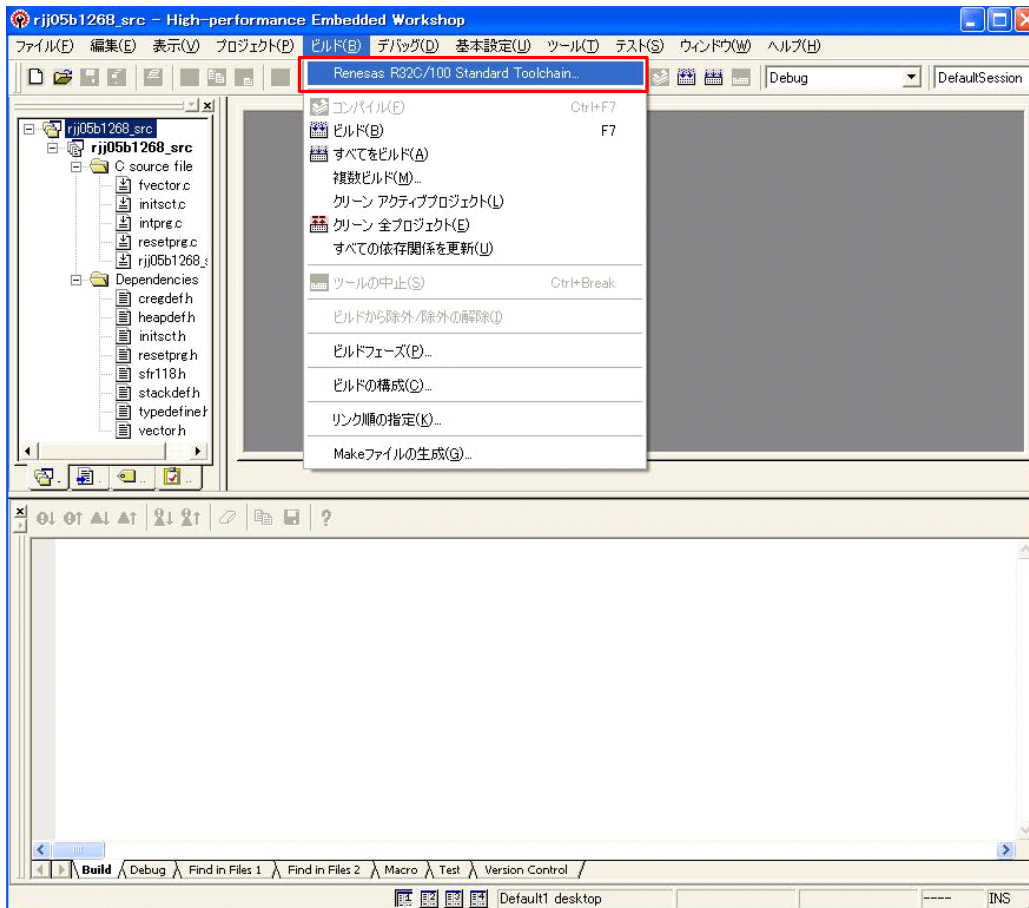
```

転送処理

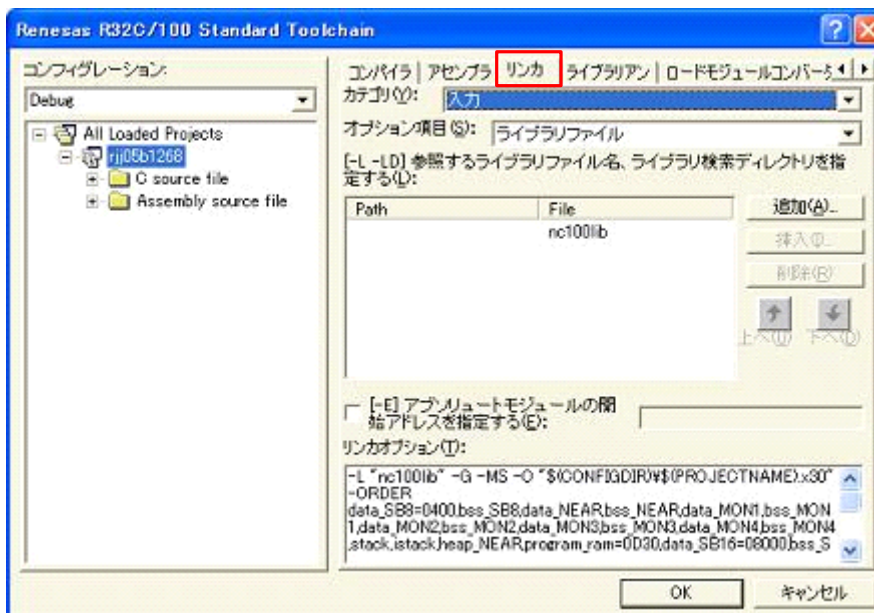
(3) プログラム格納位置の指定

RAM上に転送したプログラムを実行するには、プログラムの格納番地(ROM上)と実行番地(RAM上)を別々に配置するようにリンカ(Linker)で指定する必要があります。以下に設定方法を説明します。

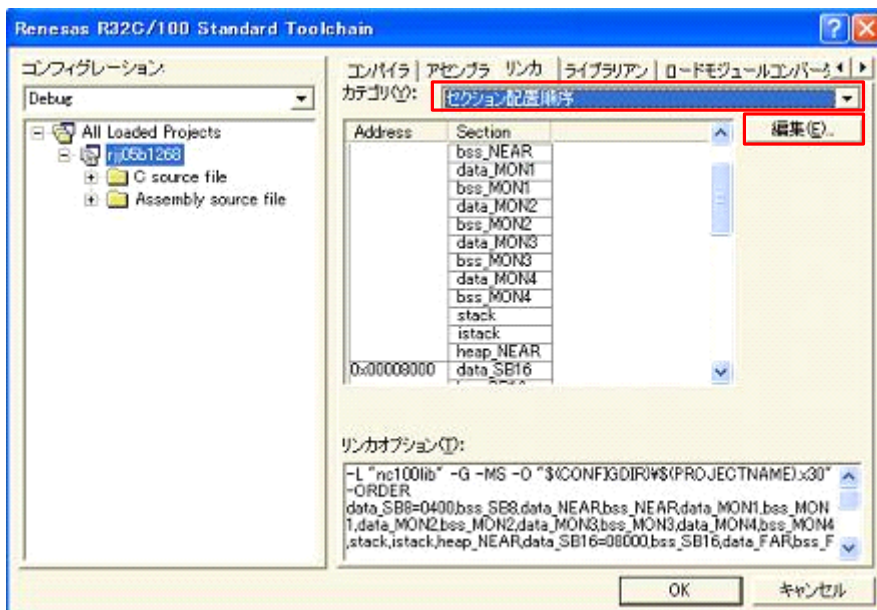
“ビルド(B)メニュー”から“Renesas R32C/100 Standard Toolchain…”を選択します。



“Renesas R32C/100 Standard Toolchain” ウィンドウにて、“リンカ”を選択します。

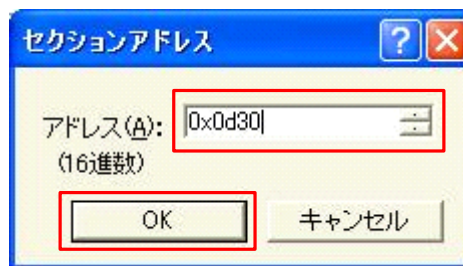
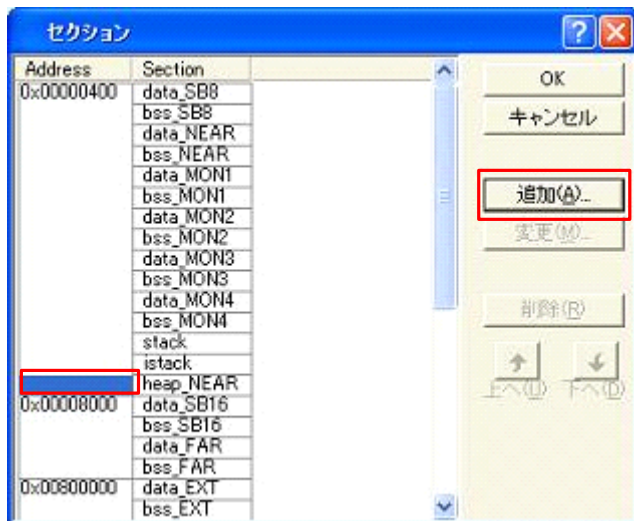


カテゴリ (Y) に “セクションの配置順序” を選択すると、セクションのメモリマップが表示されます。
“編集(E)” ボタンをクリックし、セクションウインドを表示します。

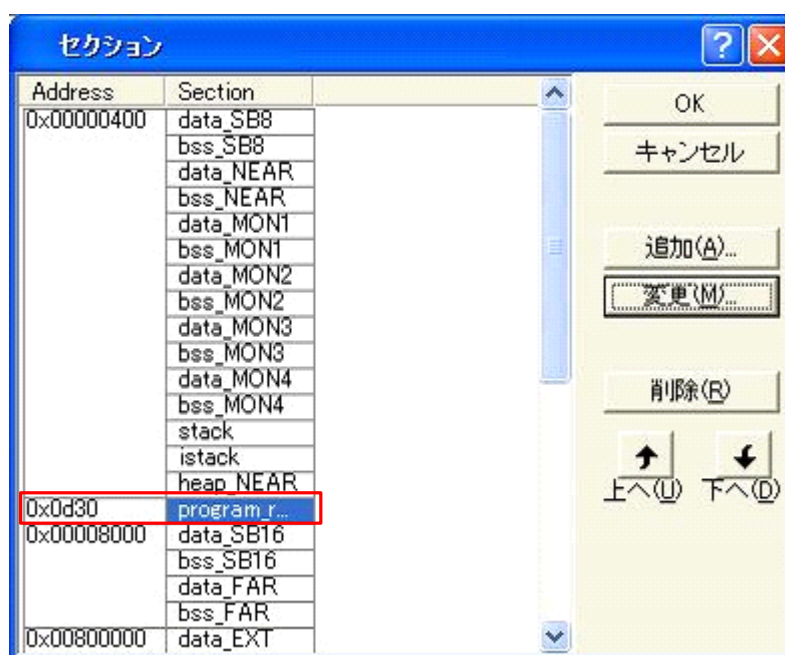
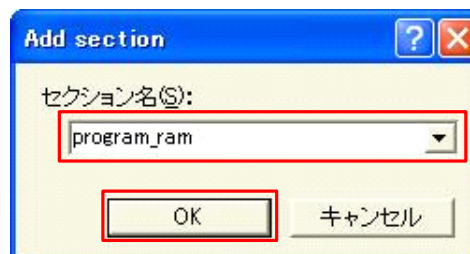
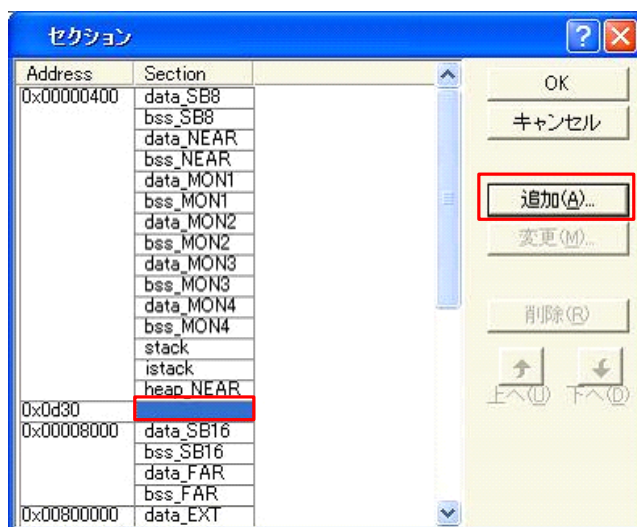


アドレスの欄の “0x00008000” の上側を選択し “追加(A)” ボタンをクリックします。セクションアドレスウインドが表示されたら “0x0d30” を設定し “OK” ボタンをクリックします。

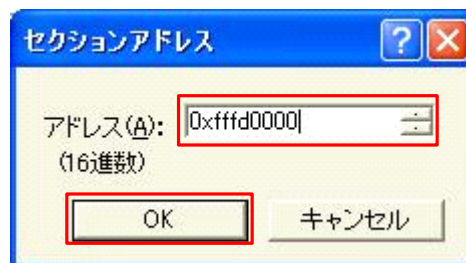
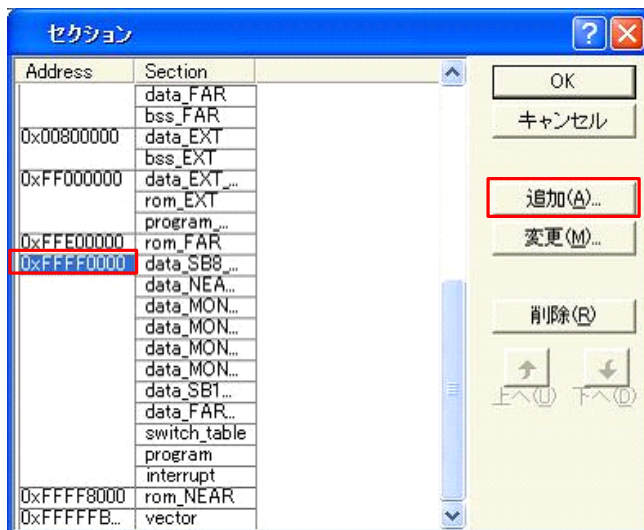
追加するセクションアドレスは、ユーザーのシステムに依存します。使用していないRAMの空領域のアドレスを設定してください。



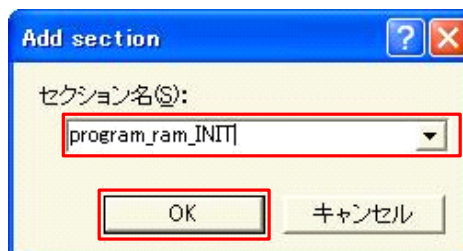
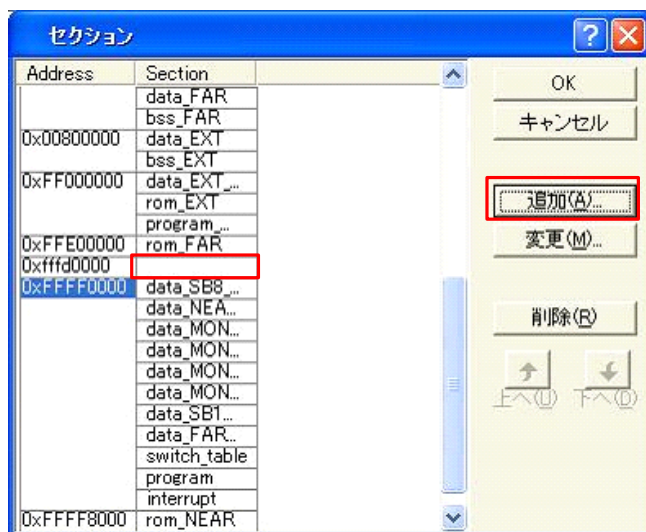
追加した”0x0d30”のアドレスのセクションの欄を選択し“追加(A)”ボタンをクリックします。“Add section”ウインドウが表示されたら、セクション名の欄に“program_ram”を入力し“OK”ボタンをクリックします。

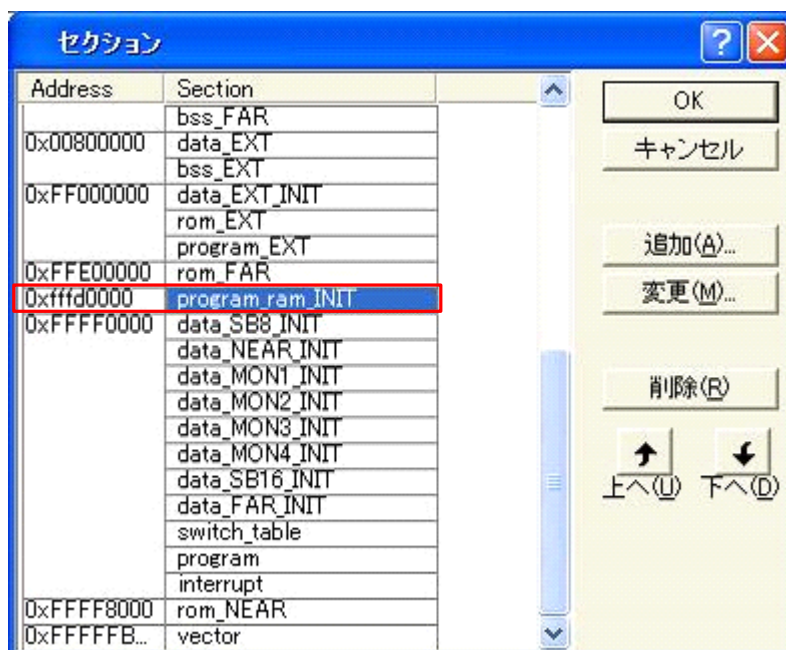


アドレスの欄の“0xFFFF0000”を選択し“追加(A)”ボタンをクリックします。セクションアドレスウインドウが表示されたら“0xfffd0000”を設定し“OK”ボタンをクリックします。



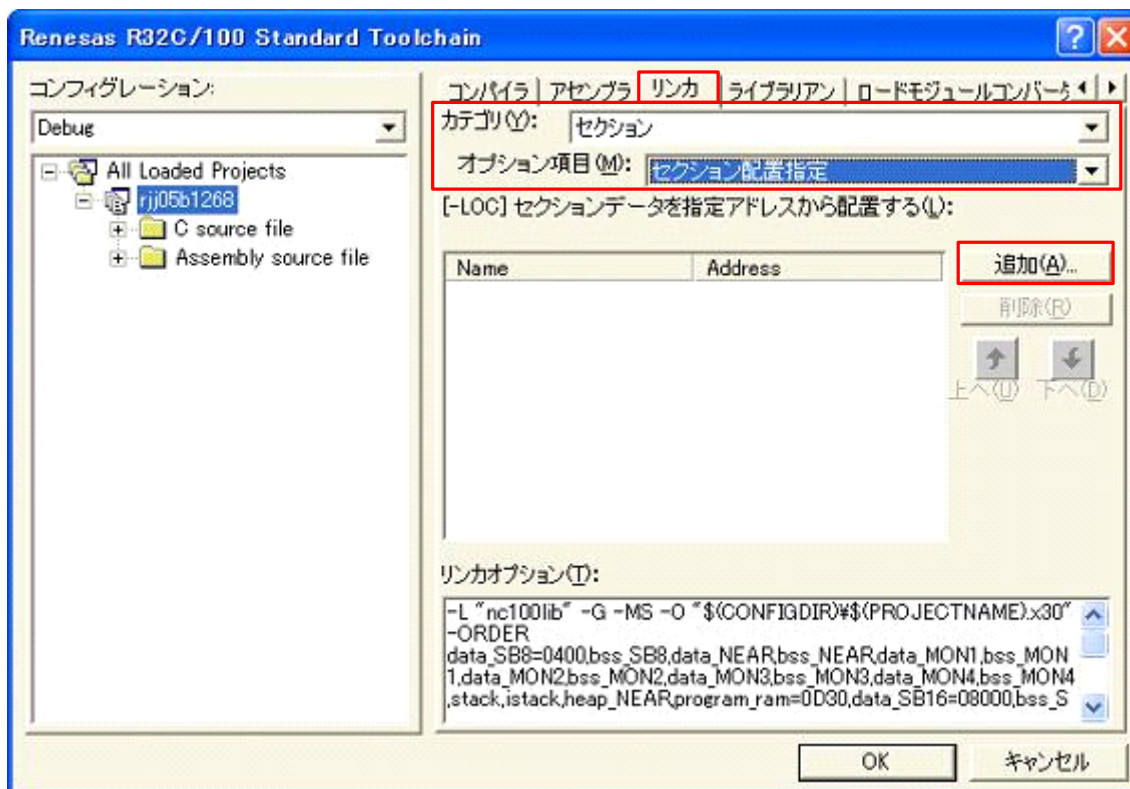
追加した“0xfffd0000”のアドレスのセクションの欄を選択し“追加(A)”ボタンをクリックします。“Add section”ウインドウが表示されたら、セクション名の欄に“program_ram_INIT”を入力し“OK”ボタンをクリックします。





次に、“リンカ”を選択した状態で“カテゴリ(Y)”から“セクション”を選択します。“オプション項目(M)”から“セクション配置指定”を選択します。

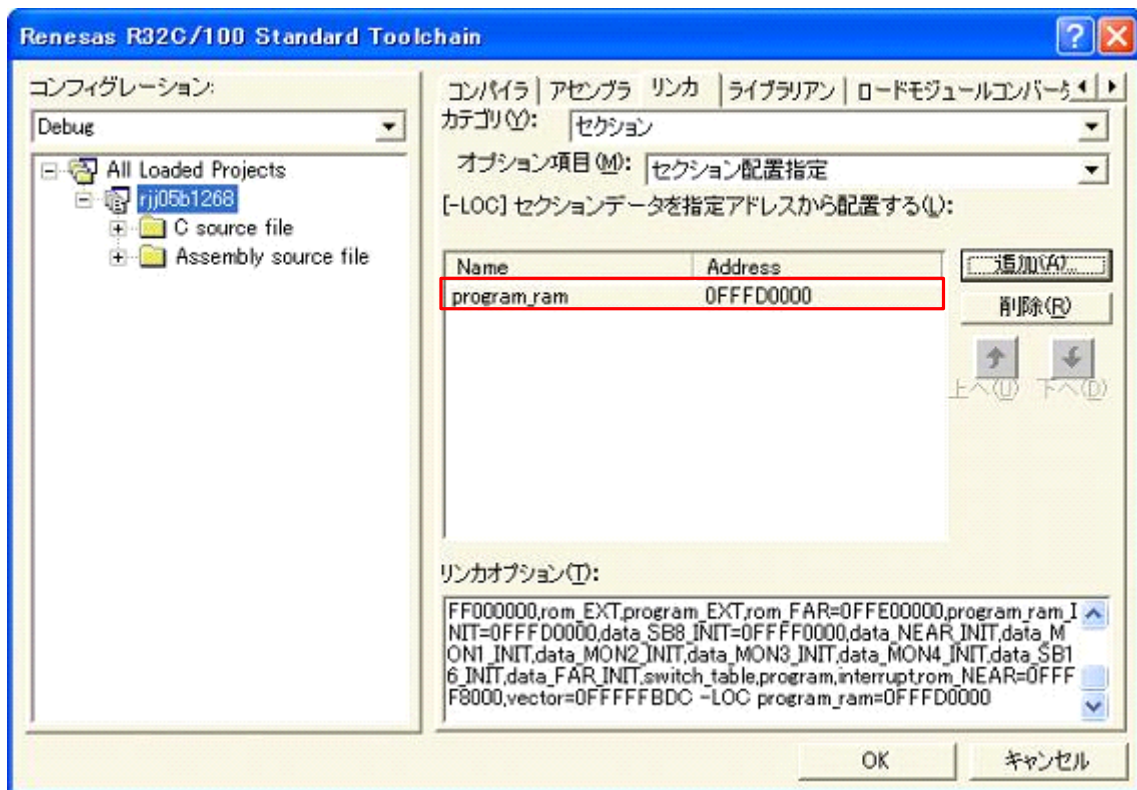
“追加”ボタンをクリックすると“Add Location”ウインドウが表示されます。



ここで、セクション名(S)に“program_ram”を、アドレス(A)に“0xFFFD0000”をそれぞれ入力し“OK”ボタンをクリックします。



上記オプション設定で、program_ramセクションを0xFFFD0000h番地(ROM上)から格納します。



3.7 フラッシュメモリ書き換えに関する注意

(1) 電源電圧に関する注意事項

- フラッシュメモリ書き換え中の電源電圧は、電気的特性に定める電圧の範囲で一定の電圧を供給してください。書き換え中に保証値を超える電圧変動があった場合、フラッシュメモリの保証はできません。

(2) ハードウェアリセットに関する注意事項

- フラッシュメモリ書き換え中は、ハードウェアリセットを行わないでください。

(3) フラッシュメモリプロテクトに関する注意点

- IDコードの格納番地に誤ったデータを書くと、標準シリアル入出力モードによるフラッシュメモリの読み書きができなくなります。

(4) プログラム作成上の注意点

- 低速モード、低消費電力モードでは、FMCRレジスタのFEWビットを“1”(CPU書き換えモード)にしないでください。
- プログラム、ブロックイレーズ、ロックビットプログラム、プロテクトビットプログラムは、NMI、ウォッチドッグタイマ割り込み、発振停止検出割り込み、電圧低下検出割り込みで中断されます。これらのソフトウェアコマンドが中断された場合、当該ブロックをイレーズした後に再度同じコマンドを実行してください。特にブロックイレーズが中断された場合、ロックビットとプロテクトビットの値は不定になりますので、ロック解除後再度ブロックイレーズを実施してください。

(5) 割り込み使用上の注意点

- 可変ベクタテーブルにベクタを持つ割り込みは、ベクタをRAM領域に移すことで使用できます。
- NMI、ウォッチドッグタイマ割り込み、発振停止検出割り込み、電圧低下検出割り込みは、割り込みが発生すると自動的にリードアレイモードになりますので、フラッシュメモリの書き換え中でも使用できます。割り込み発生時はフラッシュメモリの書き換えが中断され、FMR0、FMSR0レジスタがリセットされます。割り込み処理終了後にFMR1レジスタのLBDビットを“1”(ロックビットプロテクト無効)にしてから再度書き換えプログラムを実行してください。
- BRK命令、INT0命令、UND命令は、フラッシュメモリ上のデータを参照するため使用できません。

(6) 書き換え制御プログラムの書き換えに関する注意点

- 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電圧が低下すると、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、その後のフラッシュメモリ書き換えができなくなることがあります。書き換えできなくなった場合は、シリアルライター、パラレルライターを使用して書き換えてください。

(7) プログラム、イレーズ回数とソフトウェアコマンド実行時間

- ソフトウェアコマンド(プログラム、ブロックイレーズ、ロックビットプログラム、プロテクトビットプログラム)の実行時間は、プログラム、イレーズ回数の増加とともに長くなります。特に、プログラム、イレーズ回数が電気的特性に定めるプログラム、イレーズ回数を超えると、ソフトウェアコマンドの実行時間は著しく長くなるため、ソフトウェアコマンドの待ち時間の設定は電気的特性に定める最大時間以上に設定してください。

(8) その他の注意事項

- 電気的特性に定めるプログラム、イレーズ回数は当初の性能を保証できるプログラム、イレーズ回数の最大値です。この回数を超えると直ちにプログラム、イレーズができなくなるわけではありません。
- デバッグでプログラム、イレーズを繰り返したチップは、量産時には使用しないでください。

4. 参考プログラム

参考プログラムは、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。

4.1 参考プログラムの説明

参考プログラムでは、メインクロック逡倍モードで起動し、INT0割り込み要求の発生をトリガにCPU書き換えモード(EW0モード)に遷移して内部RAMに格納された16バイトデータをブロック7(FFFA0000h番地~FFFAFFFFh番地)へバックアップするプログラム例を示します。本プログラム例ではブロック7に対して、ブロックイレース、プログラム(RAM領域のデータ保存)を順に行います。図4.1に、参考プログラムのメモリマップを示します。

参考プログラムの実行状態は、ポートP4の状態により確認できます。表4.1にポートP4の状態を示します。

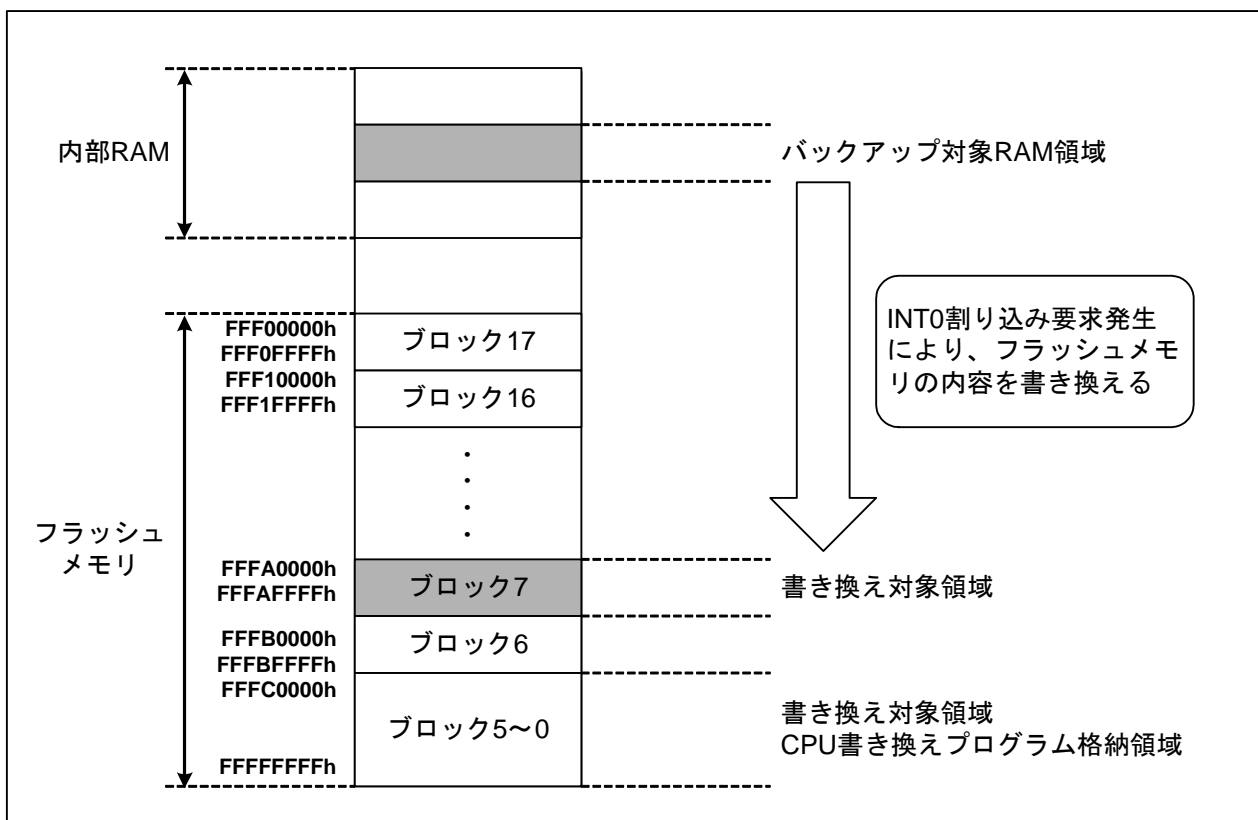


図 4.1 参考プログラムメモリマップ

表 4.1 ポートP4の状態

P4	出力状態	
	"1"	"0"
P4_0	停止	CPU書き換え実行中
P4_1	ブロックイレース正常終了	消去エラー発生
P4_2	プログラム正常終了	プログラムエラー発生

4.2 処理フロー

図 4.2に参考プログラム全体の処理フローを示します。

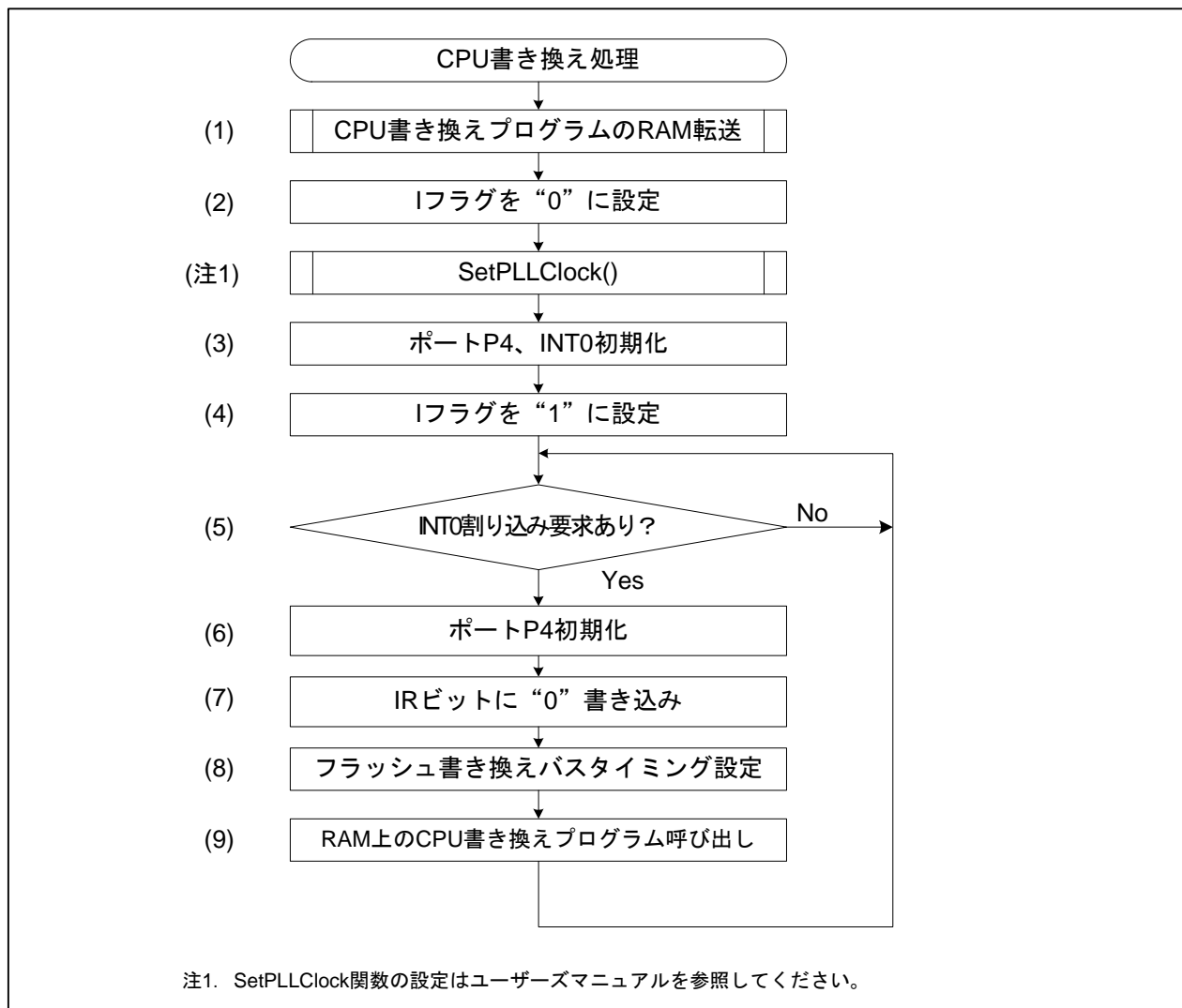


図 4.2 CPU書き換え処理

図 4.3にRAM上のCPU書き換えプログラムの処理フローを示します。

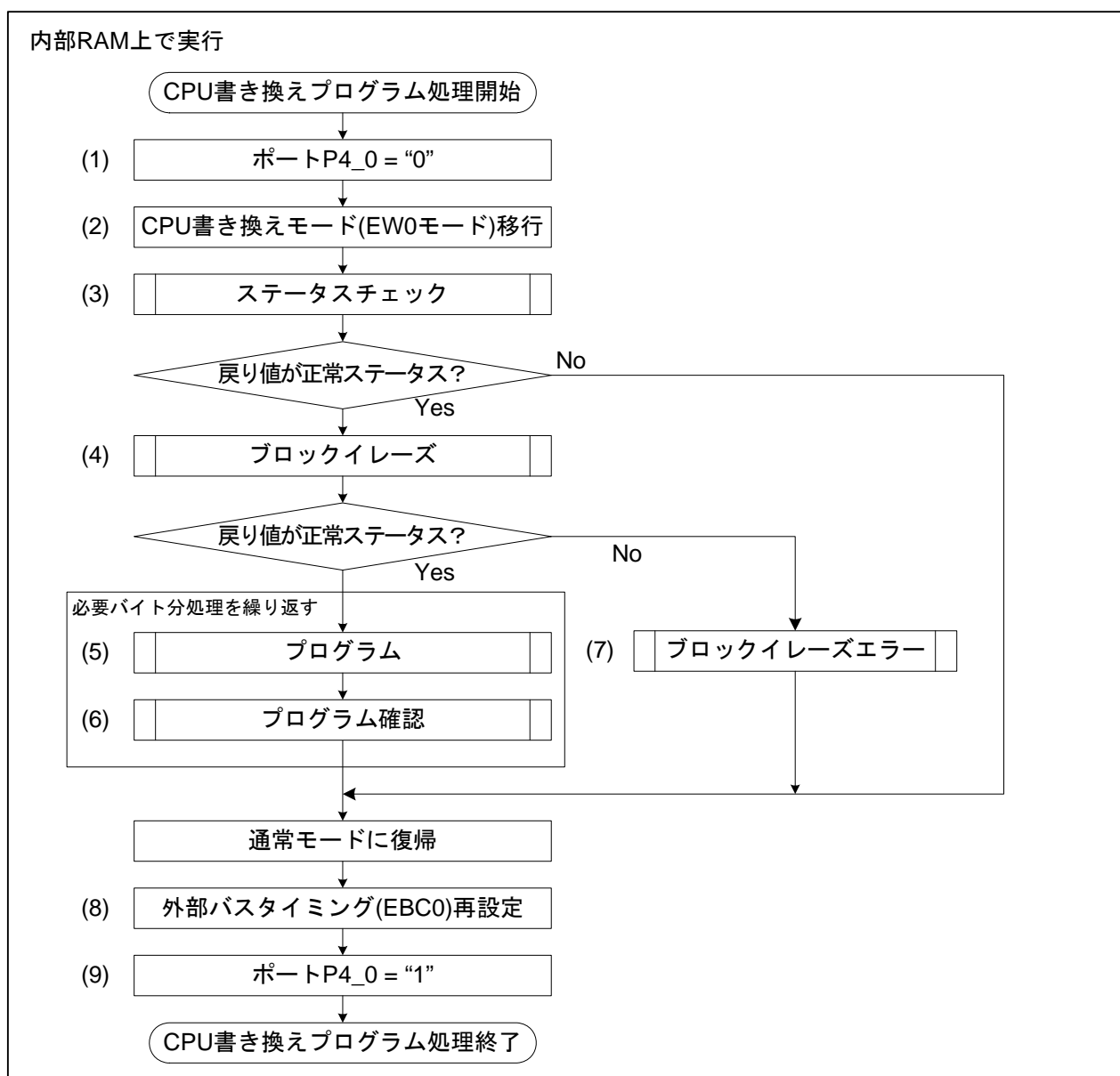


図 4.3 RAM上のCPU書き換えプログラム処理

(1) ブロックイレーズ

ブロックイレーズ関数のフローを図 4.4 に示します。

フラッシュメモリへのコマンド書き込みは16ビット単位で行ってください。

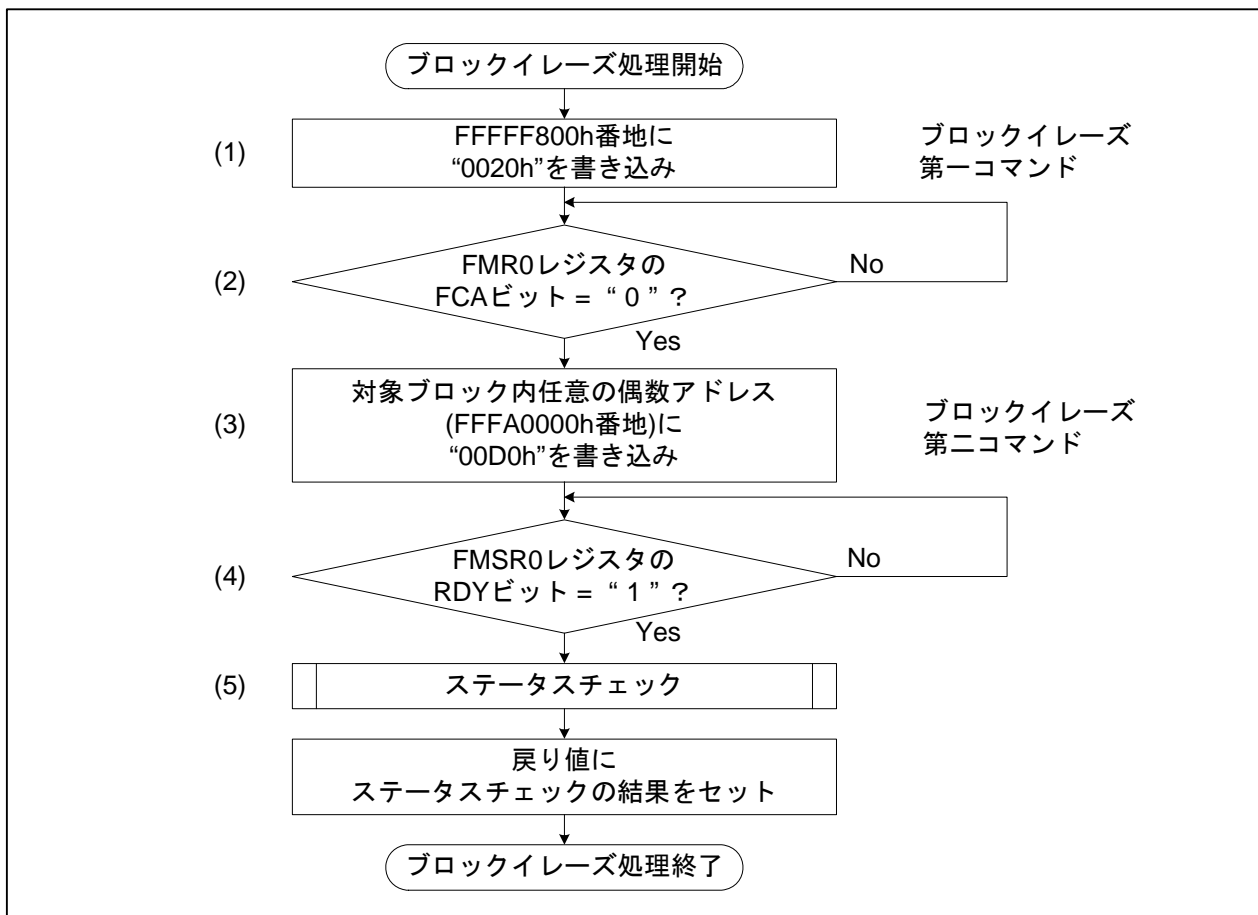


図 4.4 ブロックイレーズ関数

(2) ブロックイレーズエラー

ブロックイレーズエラー関数のフローを図 4.5 に示します。

フラッシュメモリへのコマンド書き込みは16ビット単位で行ってください。ブロックイレーズコマンドで消去エラーが発生した場合、クリアステータスレジスタコマンド発行後再度ブロックイレーズコマンドを発行し、消去エラーが発生しなくなってから少なくとも3回ブロックイレーズを実施してください。

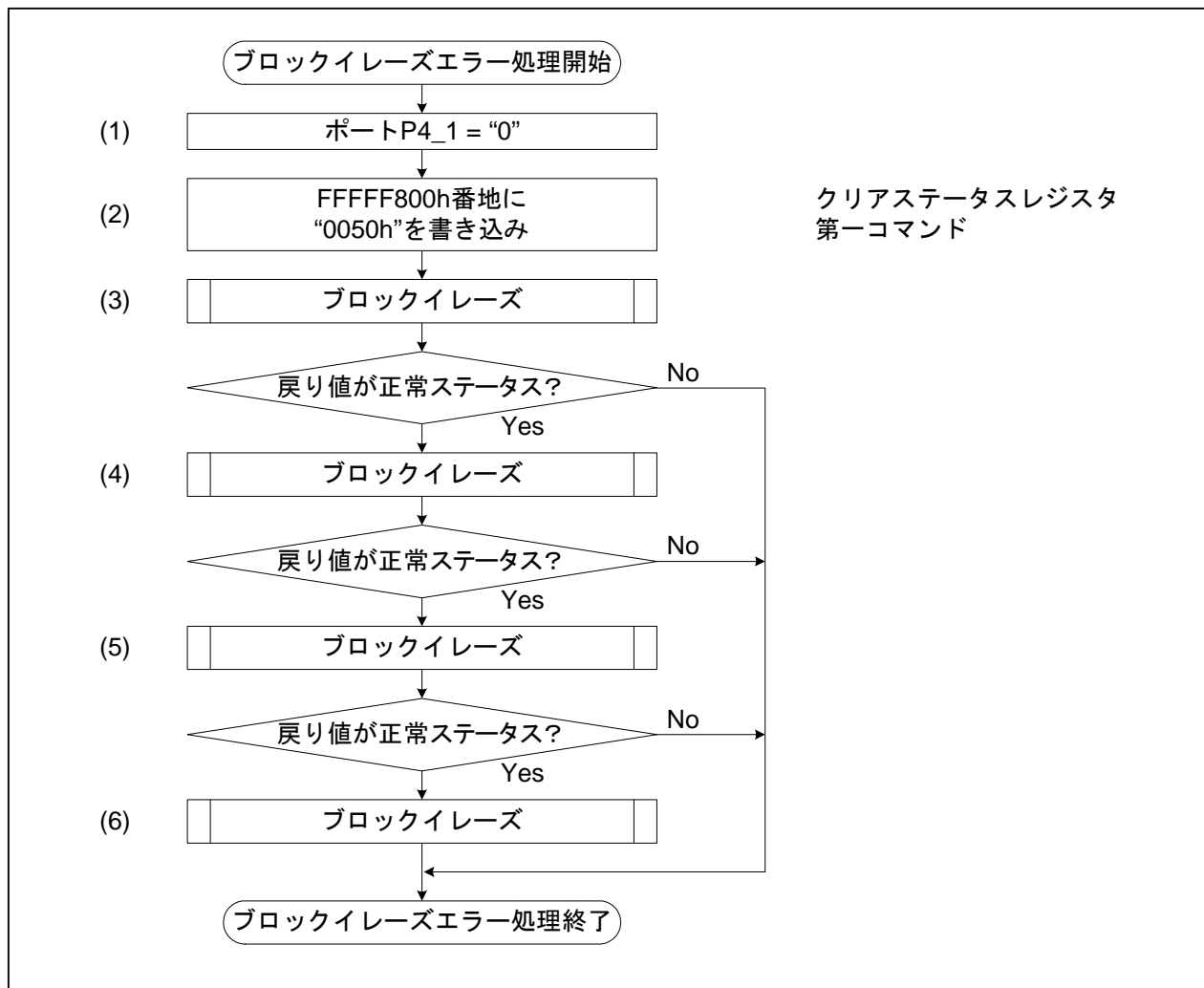


図 4.5 ブロックイレーズエラー関数

(3) プログラム

プログラム関数のフローを図 4.6 に示します。

フラッシュメモリへのコマンド書き込みは16ビット単位で行ってください。

プログラムは64ビット(4ワード)単位で行います。第二コマンド以降第五コマンドまでが一連のコマンドです。

書き込みアドレスの上位29ビットは固定、下位3ビットは、第二コマンドから順に、“000b”-“010b”-“100b”-“110b”(0h-2h-4h-6hまたは8h-Ah-Ch-Eh)となるように指定してください。

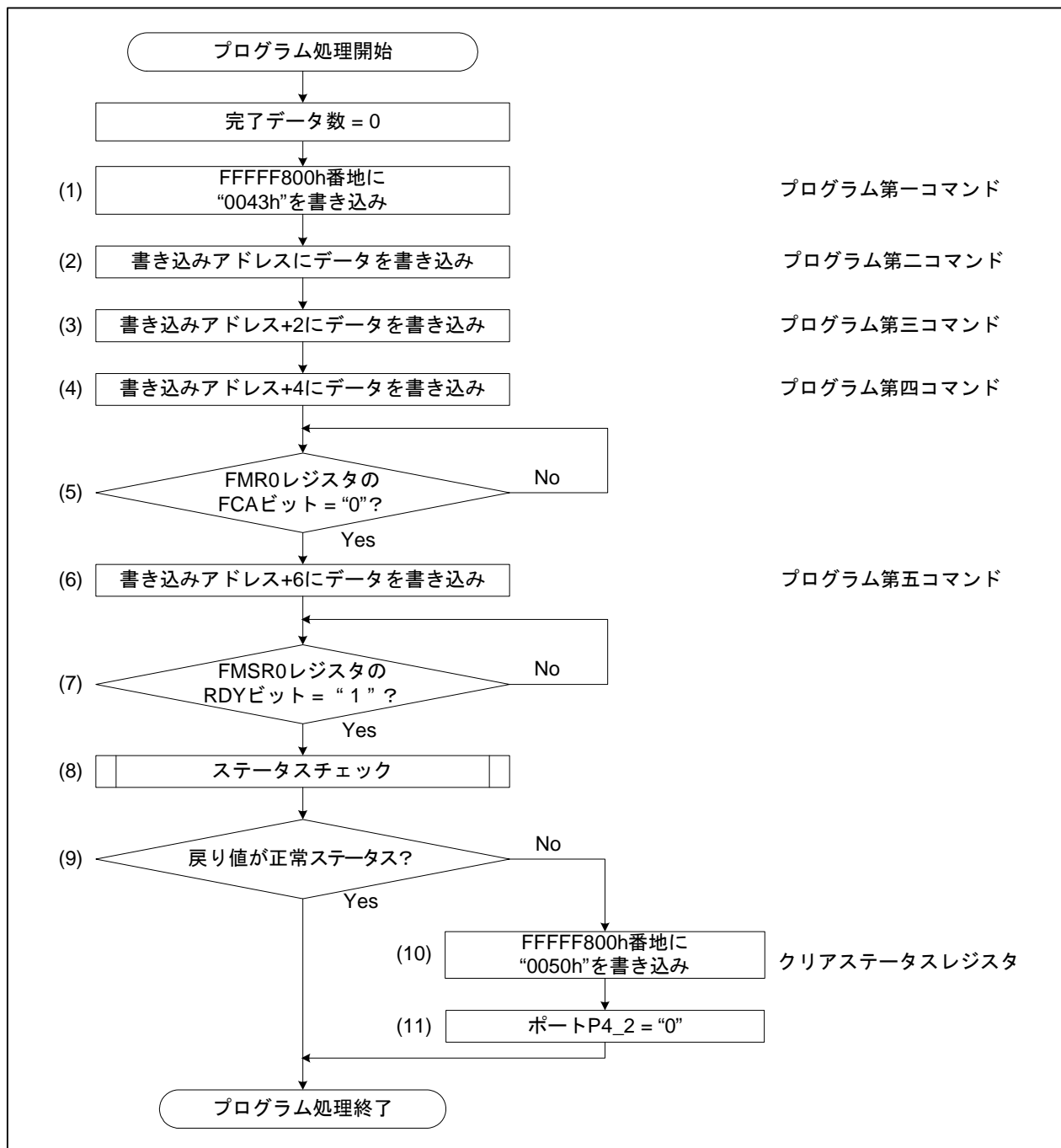
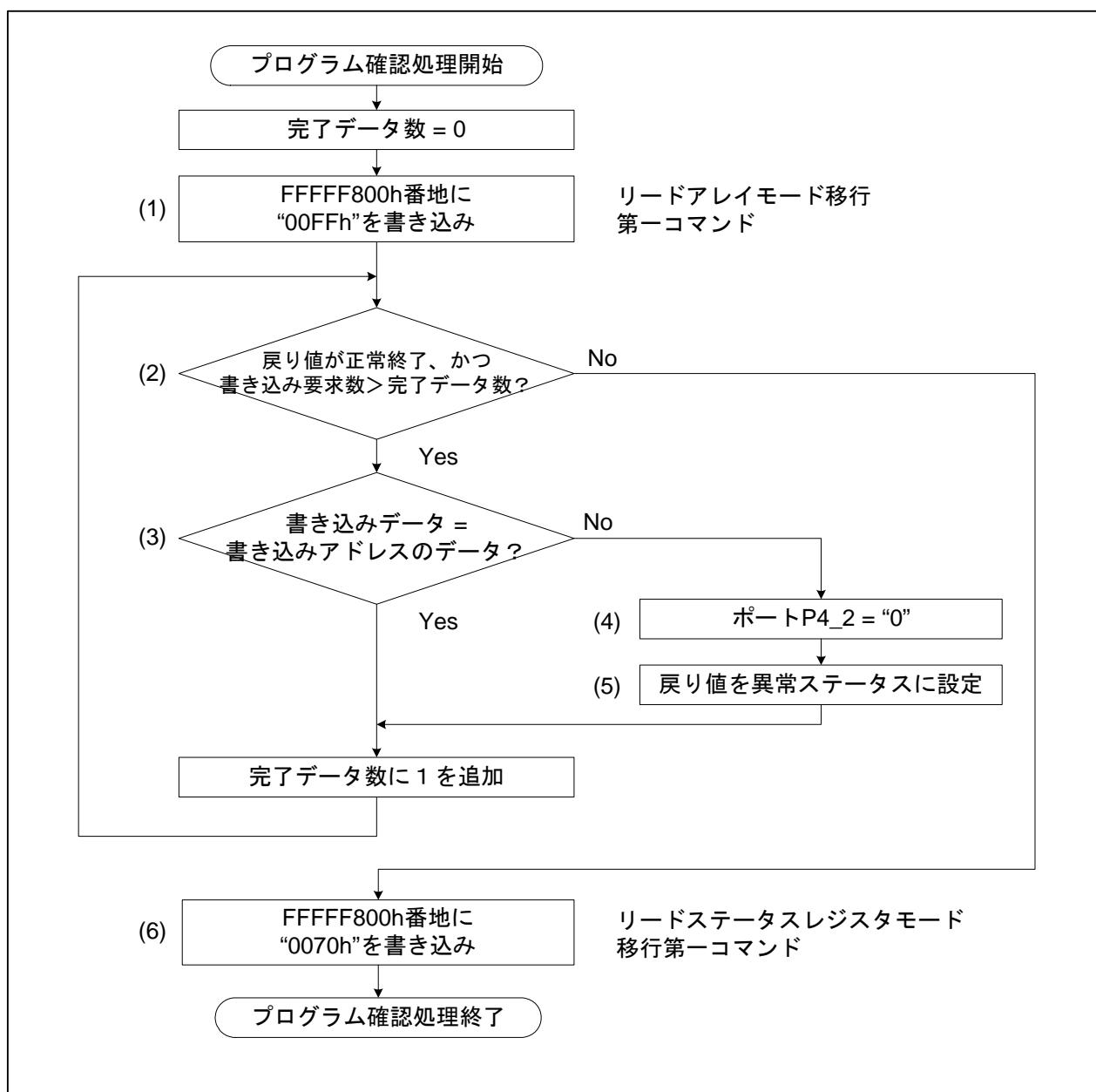


図 4.6 プログラム関数

(4) プログラム確認

プログラム確認関数のフローを図 4.7に示します。

プログラム関数で書き込まれた値が期待通りか確認します。



(5) ステータスチェック

ステータスチェック関数のフローを図 4.8 に示します。

エラー発生時はクリアステータスレジスタコマンドを発行してからエラー処理を行ってください。

プログラムが正しいにもかかわらずイレーズエラーやプログラムエラーが頻発する場合、当該ブロックは使用できなくなった可能性があります。

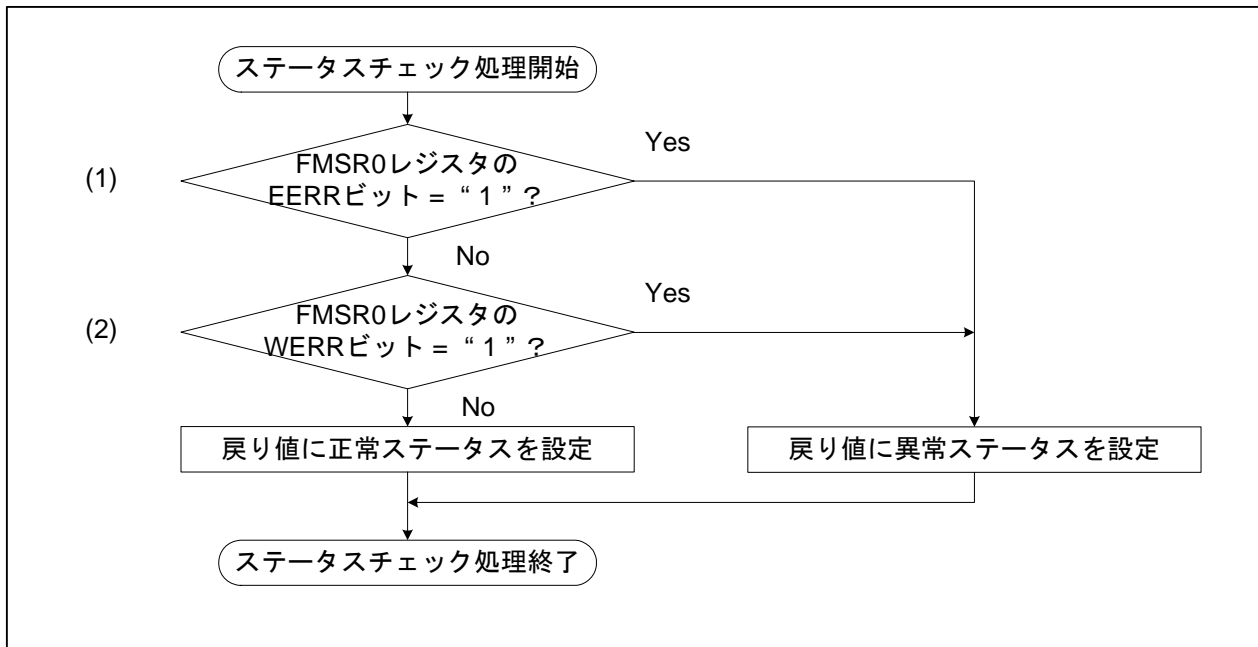


図 4.8 ステータスチェック関数

5. 参考ドキュメント

R32C/118 グループ ユーザーズマニュアル Rev.1.00

(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。)

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。)

Cコンパイラマニュアル

R32C/100シリーズ用CコンパイラパッケージV.1.02 Cコンパイラユーザーズマニュアル Rev.2.00

(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。)

ホームページとサポート窓口

ルネサス エレクトロニクスホームページ

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

お問合せ先

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

改訂記録	R32C/100 シリーズ CPU 書き換えモード(EW0モード)を使った内蔵フラッシュメモリ書き換え
------	--

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2010.05.28	-	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。

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、本文を参照してください。なお、本マニュアルの本文と異なる記載がある場合は、本文の記載が優先するものとします。

1. 未使用端子の処理

【注意】未使用端子は、本文の「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。未使用端子は、本文「未使用端子の処理」で説明する指示に従い処理してください。

2. 電源投入時の処置

【注意】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. リザーブアドレス（予約領域）のアクセス禁止

【注意】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、将来の機能拡張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。

4. クロックについて

【注意】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。

5. 製品間の相違について

【注意】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っていると、内部 ROM、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、ノイズ輻射量などが異なる場合があります。型名が異なる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、誤りがな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。



ルネサス エレクトロニクス株式会社

■営業お問合せ窓口

<http://www.renesas.com>

※営業お問合せ窓口の住所・電話番号は変更になることがあります。最新情報につきましては、弊社ホームページをご覧ください。

ルネサス エレクトロニクス販売株式会社 〒100-0004 千代田区大手町2-6-2（日本ビル）

(03)5201-5307

■技術的なお問合せおよび資料のご請求は下記へどうぞ。

総合お問合せ窓口：<http://japan.renesas.com/inquiry>