



本ドキュメントはCypress (サイプレス) 製品に関する情報が記載されております。本ドキュメントには、仕様の開発元企業として「スパンション」, 「Spansion」, 「富士通」または「Fujitsu」の名が記載されておりますが、これらの製品は Cypress が新規および既存のお客様に引き続き提供してまいります。

商品仕様の継続性について

Cypress 製品として提供することに伴う商品仕様としての変更はなく、ドキュメントとしての変更もありません。また本ページのお知らせは、変更情報として追記いたしません。本ドキュメントに変更情報が記載されている場合、それは本お知らせを除いた前版からの変更点です。なお、今後改訂は必要に応じて行われますが、その際の変更内容は改訂後のドキュメントに記載いたします。

オーダ型格および品名について

Cypress は既存のオーダ型格および品名を引き続きサポートいたします。これらの製品をご注文の際は、このドキュメントに記載されているオーダ型格および品名をご使用ください。

詳しいお問い合わせ先

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

サイプレスについて

サイプレス (銘柄コード: CY) は、車載や産業機器、ネットワーキング プラットフォームから高機能民生機器およびモバイル機器まで、今日の最先端組み込みシステム向けに高性能で高品質のソリューションを提供します。NOR フラッシュ メモリや F-RAMTM、SRAM、TraveoTM マイクロコントローラー、業界唯一の PSoC[®] プログラマブル システムオンチップ ソリューション、アナログおよび PMIC Power Management IC、CapSense[®] 静電容量タッチセンシング コントローラー、Wireless BLE Bluetooth[®] Low-Energy、USB コネクティビティ ソリューションなど、幅広い差別化製品ポートフォリオを、一貫した革新性と業界最高クラスの技術サポート、比類のないシステム バリューとともにグローバルに提供します。

FR Family
32-BIT MICROCONTROLLER
MB91F313*

パラレルプログラマ用 FLASH メモリ書き込み手順

注意事項

- 本資料の記載内容は、予告なしに変更することがありますので、ご用命の際は営業部門にご確認ください。
- 本資料に記載された動作概要や応用回路例は、半導体デバイスの標準的な動作や使い方を示したもので、実際に使用する機器での動作を保証するものではありません。したがって、これらを使用するにあたってはお客様の責任において機器の設計を行ってください。これらの使用に起因する損害などについては、当社はその責任を負いません。
- 本資料に記載された動作概要・回路図を含む技術情報は、当社もしくは第三者の特許権、著作権等の知的財産権やその他の権利の使用権または実施権の許諾を意味するものではありません。また、これらの使用について、第三者の知的財産権やその他の権利の実施ができることの保証を行うものではありません。したがって、これらの使用に起因する第三者の知的財産権やその他の権利の侵害について、当社はその責任を負いません。
- 本資料に記載された製品は、通常の産業用、一般事務用、パーソナル用、家庭用などの一般的用途に使用されることを意図して設計・製造されています。極めて高度な安全性が要求され、仮に当該安全性が確保されない場合、社会的に重大な影響を与えかつ直接生命・身体に対する重大な危険性を伴う用途（原子力施設における核反応制御、航空機自動飛行制御、航空交通管制、大量輸送システムにおける運行制御、生命維持のための医療機器、兵器システムにおけるミサイル発射制御をいう）、ならびに極めて高い信頼性が要求される用途（海底中継器、宇宙衛星をいう）に使用されるよう設計・製造されたものではありません。したがって、これらの用途にご使用をお考えのお客様は、必ず事前に営業部門までご相談ください。ご相談なく使用されたことにより発生した損害などについては、責任を負いかねますのでご了承ください。
- 半導体デバイスはある確率で故障が発生します。当社半導体デバイスが故障しても、結果的に人身事故、火災事故、社会的な損害を生じさせないよう、お客様は、装置の冗長設計、延焼対策設計、過電流防止対策設計、誤動作防止設計などの安全設計をお願いします。
- 本資料に記載された製品を輸出または提供する場合は、外国為替及び外国貿易法および米国輸出管理関連法規等の規制をご確認の上、必要な手続きをおとりください。
- 本書に記載されている社名および製品名などの固有名詞は、各社の商標または登録商標です。

Copyright© 2008 FUJITSU MICROELECTRONICS LIMITED all rights reserved

改版履歴

版数	日付	内容
1.0 版	2008/10/14	新規作成

目次

注意事項	1
改版履歴	2
目次	3
1 はじめに	4
2 パラレルプログラマを使用する場合の書き込みフロー	5
3 メモリマップ	6
3.1 CPUモードのメモリマップ	6
3.2 FLASHメモリモードのメモリマップ	7
4 CPUモード用データをFLASHメモリ用データへ変換する方法	8
4.1 用意するもの	8
4.2 書き込み用バイナリファイルへの変換方法	8
5 パラレルプログラマでの書き込み	9
5.1 エンディアンの変換が必要な場合	9
6 付録	10
6.1 バッチファイル(convert_313.bat)	10
6.2 FSG製パラレルプログラマ(1個書きタイプ)の使用方法	12
6.2.1 用意するもの	12
6.2.2 プログラマの設定手順	13
6.2.3 プログラマへのデータ転送手順	14
6.2.4 書き込み手順	15

1 はじめに

本資料では MB91F313* の FLASH 書き込み時にパラレルプログラマを用いる場合の手順について説明します。プログラマの機能や使用方法につきましては、ご使用のパラレルプログラマの製造元メーカーへお問い合わせください。

2 パラレルプログラマを使用する場合の書き込みフロー

FLASH 書き込み時にパラレルプログラマを使用する場合のフローを図 2-1 に示します。

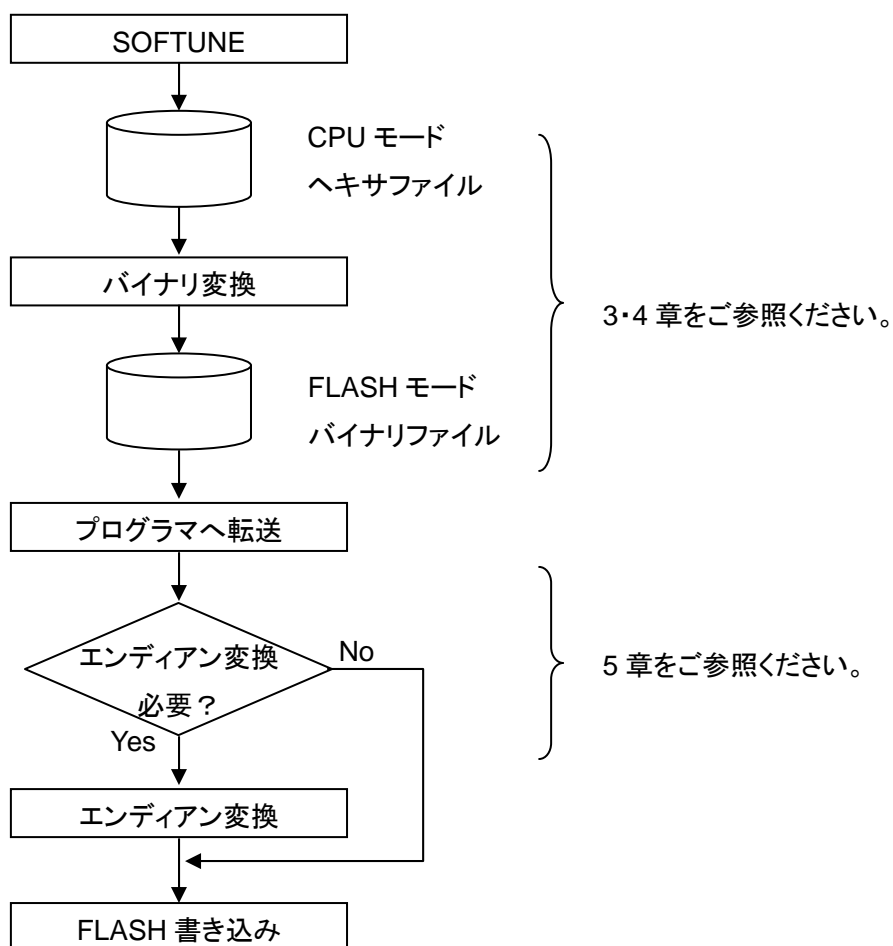


図 2-1 パラレルプログラマを使用する場合の書き込みフロー

3 メモリマップ

FLASH メモリのアドレスマッピングは FLASH メモリモードと CPU モードで異なります。SOFTUNE で生成したヘキサファイル(***.mhx)やシリアルプログラマを用いて抽出したデータは、CPU モード用のデータになっています。そのため、パラレルプログラマを用いて書き込む場合は、CPU モード用のデータを FLASH メモリモード用のデータに変換する必要があります。

3.1 CPUモードのメモリマップ

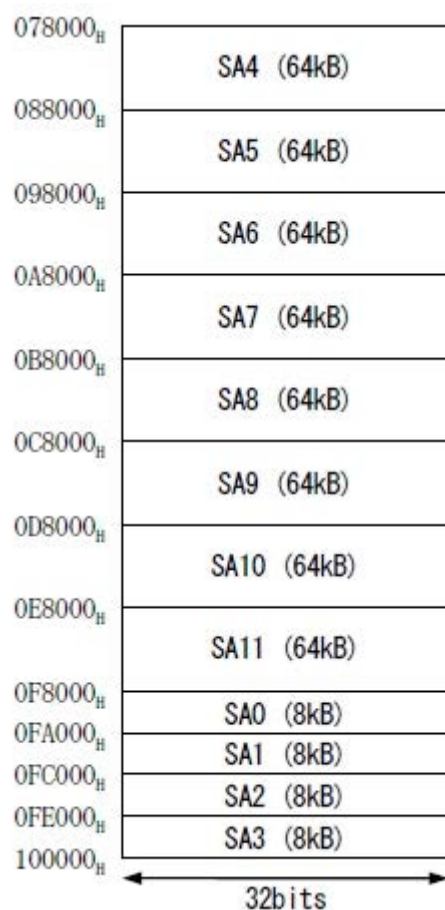


図 3.1-1 MB91F313*の CPU モードのメモリマップ

3.2 FLASHメモリモードのメモリマップ

078000 _H	SA0 (8kB)
07A000 _H	SA1 (8kB)
07C000 _H	SA2 (8kB)
07E000 _H	SA3 (8kB)
080000 _H	SA4 (64kB)
090000 _H	SA5 (64kB)
0A0000 _H	SA6 (64kB)
0B0000 _H	SA7 (64kB)
0C0000 _H	SA8 (64kB)
0D0000 _H	SA9 (64kB)
0E0000 _H	SA10 (64kB)
0F0000 _H	SA11 (64kB)
100000 _H	

16bits

図 3.2-1 MB91F313*の FLASH メモリモードのメモリマップ

4 CPUモード用データをFLASHメモリ用データへ変換する方法

4.1 用意するもの

- SOFTUNE 添付の m2bs(モトローラヘキサファイルをバイナリデータへ変換するツール)
- CPU モードのヘキサファイル(SOFTUNE で生成した”***.mhx”ファイルやシリアルプログラマで抽出したデータ)
- バッチファイル(convert_313.bat (本資料の付録に添付しています。))

4.2 書き込み用バイナリファイルへの変換方法

1. コマンドプロンプト上で CPU モードのヘキサファイル(***.mhx)があるディレクトリに移動します。
2. “m2bs”コマンドにパスを通します。
(以下は一例です。作業ディレクトリ、SOFTUNE がインストールされているディレクトリについてはお客様の環境をご確認ください。)

```
C:¥work¥abs>path ¥Softune6¥bin
```

3. バッチファイル(convert_313.bat)を実行します。

```
C:¥work¥abs>convert_313.bat
```

4. 変換するヘキサファイル(***.mhx)のファイル名を入力してください。

```
C:¥work¥abs>convert_313.bat
C:¥work¥abs>input hex file name? -> _____.mhx
```

5. 作成するバイナリファイルの名前を指定してください。

```
C:¥work¥abs>convert_313.bat
C:¥work¥abs>input hex file name? -> _____.mhx
C:¥work¥abs>output bin file name? -> _____.bin
```

バッチファイルが実行され FLASH メモリモードのバイナリファイル(***.bin)が作成されます。これをパラレルプログラマの書き込みデータとしてご使用ください。

5 パラレルプログラマでの書き込み

FLASHメモリモードのバイナリファイルをパラレルプログラマへ転送し、マイコンのFLASHメモリへの書き込みを行ってください。この際、ご使用のパラレルプログラマによってはエンディアンの変換が必要となります。

5.1 エンディアンの変換が必要な場合

ご使用のパラレルプログラマとマイコンの FLASH メモリモードのエンディアンが一致していない場合はバイナリファイルをプログラマに転送した後、エンディアンの変換が必要になります。

エンディアンの変換が必要な場合は、プログラマの SWAP 機能を使用してバイト単位でバイナリデータの入れ替えを行ってください。

表 5.1-1 エンディアンの変換が必要なパラレルプログラマ

プログラマ型格	仕様	メーカー
AF9708	1 個書きプログラマ	フラッシュサポートグループ(株)
AF9709/09B/09C	1 個書きプログラマ	
AF9710	1 個書きプログラマ	
AF9834	ギャングユニット(※)	
AF9845/45B/45C	ギャングユニット(※)	

(※) ギャングユニットは AF9723/23B 本体と組み合わせてご使用ください。

上記表中に記載のないプログラマをご使用のお客様でエンディアンの変換が必要かどうか不明な場合は、弊社またはご使用のパラレルライターメーカーまでお問い合わせください。

6 付録

6.1 バッチファイル(convert_313.bat)

```

echo off

set /P hex_file="input hex file name? -> "

if exist %hex_file% ( goto FOUND ) else goto NOTFOUND

:FOUND

set /P bin_file="output bin file name? -> "

echo %hex_file% is found. Binary file %bin_file% is made...


m2bs %hex_file% -p 0xff -ran 0x78000,0x87fff -o tmpbin4
m2bs %hex_file% -p 0xff -ran 0x88000,0x97fff -o tmpbin5
m2bs %hex_file% -p 0xff -ran 0x98000,0xa7fff -o tmpbin6
m2bs %hex_file% -p 0xff -ran 0xa8000,0xb7fff -o tmpbin7
m2bs %hex_file% -p 0xff -ran 0xb8000,0xc7fff -o tmpbin8
m2bs %hex_file% -p 0xff -ran 0xc8000,0xd7fff -o tmpbin9
m2bs %hex_file% -p 0xff -ran 0xd8000,0xe7fff -o tmpbin10
m2bs %hex_file% -p 0xff -ran 0xe8000,0xf7fff -o tmpbin11
m2bs %hex_file% -p 0xff -ran 0xf8000,0xf9fff -o tmpbin0
m2bs %hex_file% -p 0xff -ran 0xfa000,0xfbfff -o tmpbin1
m2bs %hex_file% -p 0xff -ran 0xfc000,0xfdfff -o tmpbin2
m2bs %hex_file% -p 0xff -ran 0xfe000,0xfffff -o tmpbin3
copy /b tmpbin0.bin+tmpbin1.bin+tmpbin2.bin+tmpbin3.bin+tmpbin4.bin+tmpbin5.bin+tmpbin6.bin+tmpbin7.bin+tmpbin8.bin+tmpbin9.bin+tmpbin10.bin+tmpbin11.bin %bin_file%

echo Binary file %bin_file% was made in this folder.

del tmpbin*.b*

```

```
:NOTFOUND
```

```
    echo %hex_file% is not found.
```

```
    goto END
```

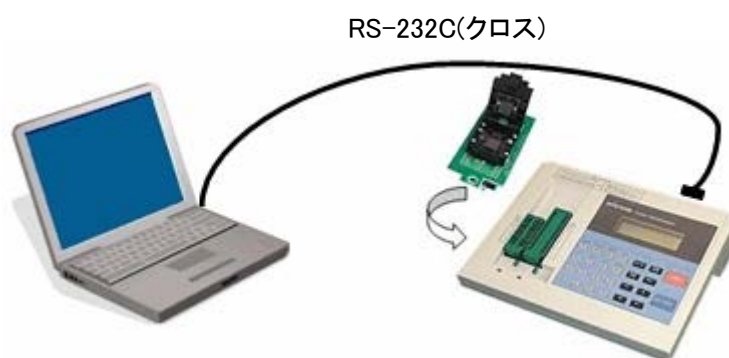
```
:END
```

6.2 FSG製パラレルプログラマ(1個書きタイプ)の使用方法

例として、フラッシュサポートグループ(FSG)製パラレルプログラマ(シリアル I/F:RS-232C クロス、1 個書きタイプ)を使用し MB91F313 の書き込みを行う際の手順を以下に示します。

6.2.1 用意するもの

- マイコン: MB91F313 (FPT-120P-M21)
- プログラマ本体: AF9708/09/09B (いずれか)
- アダプタ: TRF110-91F313
- パソコン
- データ転送用ソフト(任意)
- RS-232C ケーブル(クロス)



6.2.2 プログラムの設定手順

プログラムの設定手順を以下に示します。

- ① プログラムの電源を入れます。
- ② デバイスコード「17257」を指定します。

ROM TYPE ▲ ([CODE]を選択)
「17257」を入力 SET/START

- ③ デバイス書き込み範囲の指定を行う際には、プログラムのバッファ RAM の 0～87FFF 番地を START/STOP ADDRESS に設定します。

FUNCTION A ([CONTROL]を選択)
▲ ([ADDRESS % DISPLAY]を選択) ▼ ▲ ([OFF]に設定) SET/START
FUNCTION 1 (START/STOP ADDRESS を選択)
RAMSTA=「0」SPA=「87FFF」になっていることを確認 SET/START

- ④ データフォーマットを「DG-BINARY」に設定し、SBF を「FF」に設定します。

FUNCTION 5 ▲ (FMT.NAME[DG-BINARY]を選択)
▼ SUB-FMT「SBF=FF」を入力 SET/START

6.2.3 プログラマへのデータ転送手順

プログラマへのデータ転送手順を以下に示します。

- ① RS-232Cインタフェースの設定を行います。(以下は設定例です。設定はPCと合わせてください。)

FUNCTION	4	
<input type="checkbox"/> BAUD RATE:	[38400]	を選択 <input type="checkbox"/>
<input type="checkbox"/> BIT LENGTH:	[8BIT]	を選択 <input type="checkbox"/>
<input type="checkbox"/> PARITY BIT:	[NON]	を選択 <input type="checkbox"/>
<input type="checkbox"/> STOP BITS:	[2BIT]	を選択 <input type="checkbox"/> SET/START

- ② 転送時 Xon-off 制御はイネーブルに設定します。

FUNCTION	A	<input type="checkbox"/> CONTROL[SERIAL XON/XOFF]	を選択 <input type="checkbox"/>
<input type="checkbox"/> [ON]	を選択	SET/START	

- ③ プログラマをデータの入力待ち状態に設定します。

FUNCTION	6	SERIAL I/F [INPUT]	を選択	SET/START
----------	---	--------------------	-----	-----------

- ④ データ転送用のソフト(任意)を使用し、PC からプログラマへ書き込みデータを転送します。
PC から送信したデータをプログラマが正常に受け付けると、プログラマの表示が READY*から RECEIVING*に変わります。

【注意】 PC からデータを転送する際は、必ずバイナリモードでデータを転送してください。

- ⑤ データ転送後、16bit 毎にデータの SWAP を行います。

EDIT	A	(SWAP を選択)	<input type="checkbox"/> 16BIT に設定	SET/START
------	---	------------	------------------------------------	-----------

6.2.4 書き込み手順

Flash 書き込みの手順を以下に示します。

- ① アダプタにマイコンを載せて、アダプタをプログラマにセットします。
- ② Flash の消去を行います。

DEVICE	ERASE	SET/START
--------	-------	-----------

- ③ Flash のブランクチェックを行います。

DEVICE	BLANK	SET/START
--------	-------	-----------

- ④ Flash の書き込みを行います。

DEVICE	PROGRAM	SET/START
--------	---------	-----------

- ⑤ Flash に書込んだデータの Verify チェックを行います。

DEVICE	VERIFY	SET/START
--------	--------	-----------