

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ひ申し上げます。

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>

日立半導体技術情報

〒100-0004
 東京都千代田区大手町2丁目6番2号
 (日本ビル)
 TEL (03)5201-5022 (ダイヤルイン)
 株式会社 日立製作所 半導体グループ

製品分類	開発環境		発行番号	TN-OS*-066A	Rev.	第1版
題名	HI7750/4におけるFPU使用時の注意事項		情報分類	1. 仕様変更 2. ドキュメント訂正追加等 ③. 使用上の注意事項 4. マスク変更 5. ライン変更		
適用製品	HS0775ITI41SRE, HS0775ITI41SRE-E, HS0775ITI41SRB, HS0775ITI41SRB-E, HS0775ITI41SRS, HS0775ITI41SRS-E	対象ロット等 V1.00r1, V1.01r1, V1.0Ar1, V1.0Br1, V1.0Cr1	関連資料	HI7000/4 シリーズユーザーズマニュアル ADJ-702-298A 第2版		有効期限 永年

拝啓、貴社益々御清栄のこととお慶び申し上げます。平素は格別の御高配を賜り、感謝申し上げます。

HI7750/4でFPUを使用する場合は、下記の添付資料に記載の注意事項にご留意いただけますよう、お願いいたします。

添付資料：「HI7750/4におけるFPU使用時の注意」(HI7750/4-NOTE-FPU-021105)

以上

HI7750/4 における FPU 使用時の注意

FPU を使用する場合は、本資料に記載の注意事項に留意いただけますよう、お願い申し上げます。

1. タスク、タスク例外処理ルーチン

1.1 TA_COP1, TA_COP2 属性

SH-4 は、FPU レジスタバンク 0, 1 を持っています。一方、TA_COP1, TA_COP2 属性は、それぞれ FPU バンク 0, FPU バンク 1 を使用する、という意味です。

TA_COP1, TA_COP2 属性は、表 1 に示すように指定してください。なお、起動時はバンク 0 (FPSCR.FR=0) になっています。

表 1 TA_COP1, TA_COP2 属性の指定

ケース	属性	備考
マトリクス演算などを行う場合(両方の FPU レジスタバンクを使用する場合)	TA_COP1 TA_COP2	
浮動小数点演算を行う場合	TA_COP1	通常の浮動小数点演算は片方の FPU レジスタバンクのみを使用します。 また、TA_COP2 を指定すると、タスク、タスク例外処理ルーチンのエントリ関数の先頭で FPSCR.FR=1 に設定する必要があるため、推奨しません。
浮動小数点演算を行わない場合	指定不要	

1.2 コンパイラオプションとの関係

起動時の FPSCR の各ビットは、「4.1 タスクやハンドラなどの起動時の状態」に示す通りです。

浮動小数点演算を行う場合、コンパイラオプションによってはこの初期値のままでは正常に動作しないため、タスク、タスク例外処理ルーチンのエントリ関数の先頭で、表 2 に示すように FPSCR の各ビットを設定する必要があります。

表 2 FPSCR に設定すべき値

FPSCR のビット	コンパイラオプション		設定すべき値
精度モード(FPSCR.PR)	FPU オプション	DOUBLE	1
		上記以外	(設定不要)
非正規化モード(FPSCR.DN)	Denormalize オプション	OFF	(設定不要)
		ON	0
丸めモード(FPSCR.RM)	Round オプション	Zero	(設定不要)
		Nearest	B'00

以下に、下記の条件におけるタスクの FPSCR 設定例を以下に示します。

[コンパイラオプション]

- ・ FPU = DOUBLE
- ・ Denormalize = ON
- ・ Round = Nearest

```
#include <machine.h> /* 組み込み関数 set_fpscr() を使用するためにインクルード */
#define INI_FPSCR 0x00080000 /* PR=1, DN=0, SZ=0, RM=B'00 */
#pragma noregsave(Task)
void Task(VP_INT exinf)
{
    set_fpscr(INI_FPSCR); /* 先頭で FPSCR を設定 */
    /* タスクの処理 */
    ext_tsk();
}
```

2. 割込みハンドラ、CPU 例外ハンドラ、タイムイベントハンドラ、初期化ルーチン

これらのハンドラで浮動小数点演算を行う場合は、「HI7000/4 シリーズユーザズマニュアル(第2版, ADJ-702-298A)」の「4.14 FPUを使用する場合(HI7750/4)」を参考に、ハンドラの手前で FPSCR を初期設定してください。なお、ハンドラ終了時に FPSCR を元に戻す必要はありません。

表 3に、FPSCR の各ビットに設定すべき値を示します。

なお、浮動小数点演算を行わない場合は、FPSCR の初期設定は不要です。

表 3 FPSCR に設定すべき値

FPSCR のビット	コンパイラオプション		設定すべき値
精度モード(FPSCR.PR)	FPU オプション	DOUBLE	1
		上記以外	0
非正規化モード(FPSCR.DN)	Denormalize オプション	OFF	1
		ON	0
丸めモード(FPSCR.RM)	Round オプション	Zero	B'01
		Nearest	B'00
転送サイズモード(FPSCR.SZ)			0
FPU レジスタバンク(FPSCR.FR)			0
FPSCR のその他のビット			0

3. 拡張サービスコールルーチン

3.1 浮動小数点演算を行う場合

拡張サービスコールの発行は、コンパイラは関数呼び出しと扱います。しかし、呼び出し側と拡張サービスコールルーチンは別のリンク単位である場合には、呼び出し側と拡張サービスコールルーチンとでコンパイラオプションが異なるケースがあります。この場合は、拡張サービスコールルーチンのエントリ関数の先頭で表 3と同様に FPSCR を初期設定し、拡張サービスコールルーチン終了時に FPSCR を元の値に戻してください。

3.2 浮動小数点演算を行わない場合

基本的には、拡張サービスコールとその呼び出し側は、以下のコンパイラオプションを同じにしてください。

- FPU オプション
- FPSCR オプション
- Denormalize オプション
- Round オプション

特に、拡張サービスコールルーチンのコンパイラオプションが

- FPU オプションの指定無し(MIX)
- FPSCR オプション = Aggressive

で、かつ拡張サービスコールの呼び出し側のコンパイラオプションが上記とは異なる場合は、拡張サービスコールルーチンの先頭で FPSCR レジスタを退避し、終了時に復帰する必要があります。

以下に、その例を示します。

```
#include <machine.h> /* 組み込み関数 set_fpscr() を使用するためにインクルード */
ER ExtendedSVCRoutine(VP_INT par1)
{
    UW old_fpscr;
    old_fpscr = get_fpscr(); /* FPSCR を退避 */
    /* 拡張サービスコールルーチンの処理 */
    set_fpscr(old_fpscr); /* FPSCR の復帰 */
    return E_OK;
}
```

4. 参考情報

4.1 タスクやハンドラなどの起動時の状態

表 4に、タスクやハンドラなどの起動時の状態を示します。

表 4 タスクやハンドラなどの起動時の状態

起動時の状態	タスク, タスク例外処理ルーチン	拡張サービスコールルーチン	割込みハンドラ, タイムイベントハンドラ, 初期化ルーチン	CPU 例外ハンドラ (TRAPA 含む)
FPSCR の値	H'00040001	拡張サービスコール発行前と同じ	不定	例外発生前と同じ
精度モード (FPSCR.PR)	単精度(0)	拡張サービスコール発行前と同じ	不定	例外発生前と同じ
非正規化モード (FPSCR.DN)	0 と扱う(1)	拡張サービスコール発行前と同じ	不定	例外発生前と同じ
丸めモード (FPSCR.RM)	0 に丸める (B'01)	拡張サービスコール発行前と同じ	不定	例外発生前と同じ
転送サイズモード (FPSCR.SZ)	32 ビット(0)	拡張サービスコール発行前と同じ	不定	例外発生前と同じ
FPU レジスタバンク (FPSCR.FR)	バンク 0(0)	拡張サービスコール発行前と同じ	不定	例外発生前と同じ
FPSCR のその他のビット	0	拡張サービスコール発行前と同じ	不定	例外発生前と同じ

4.2 SH-4 の FPSCR 構造

31 22 21 20 19 18 17 12 11 7 6 2 1 0

予約	FR	SZ	PR	DN	Cause	Enable	Flag	RM
----	----	----	----	----	-------	--------	------	----

bit		意味		
21	FR	FPU レジスタバンク	0	バンク 0
			1	バンク 1
20	SZ	転送サイズモード	0	FMOV のデータサイズは 32 ビット
			1	FMOV のデータサイズは 32 ビットペア(64 ビット)
19	PR	精度モード	0	単精度
			1	倍精度
18	DN	非正規化モード	0	非正規化数を 0 と扱う
			1	非正規化数を非正規化数と扱う
17-12	Cause	FPU 例外要因		
11-7	Enable	FPU 例外イネーブル		
6-2	Flag	FPU 例外フラグ		
1,0	RM	丸めモード	B'00	近傍への丸め
			B'01	0 方向への丸め

4.3 コンパイラによる扱い

本節では、コンパイラ V5.1, V6, V7.1 による FPU の扱いについて解説します。なお、FPU オプションに Single または Double が指定された場合は、コンパイラが FPSCR を変更するオブジェクトコードを生成することはありません。

(1) FPSCR.PR(精度モード)

表 5 コンパイラによる FPSCR.PR ビットの扱い

コンパイラオプション		関数開始時点でコンパイラが想定している精度モード(FPSCR.PR ビット) *1	関数終了時の精度モード *2	備考
FPU オプション	FPSCR オプション *3			
Single	(指定不可)	単精度(0)	単精度(0)	コンパイラは PR ビットを変更するオブジェクトコードは生成しません。
Double	(指定不可)	倍精度(1)	倍精度(1)	
指定無し (Mix)	Safe	単精度(0)	単精度(0)	
	Aggressive	単精度(0)	不定	

【注】 *1 コンパイラは、関数先頭でこの精度モードになっている前提でコード生成を行います。

*2 コンパイラは、関数終了時にこの精度モードとなるようにコード生成を行います。

*3 コンパイラ V5.1 では、FPSCR オプションはサポートしておらず、“Aggressive”と同じ意味になります。

(2) FPSCR.DN(非正規化モード)

表 6 コンパイラによる FPSCR.DN ビットの扱い

コンパイラオプション	コンパイラが想定している非正規化モード(FPSCR.DN ビット) *1	備考
Denormalize オプション		
OFF	非正規化数を 0 と扱う(1)	コンパイラは DN ビットを変更するオブジェクトコードは生成しません。
ON	非正規化数を非正規化数と扱う(0)	

【注】 *1 コンパイラは、関数先頭でこの非正規化モードになっている前提でコード生成を行います。

(3) FPSCR.RM(丸めモード)

表 7 コンパイラによる FPSCR.RM ビットの扱い

コンパイラオプション	コンパイラが想定している丸めモード(FPSCR.RM ビット) *1	備考
Round オプション		
Zero	0 で丸める(B'01)	コンパイラは RM ビットを変更するオブジェクトコードは生成しません。
Nearest	近傍に丸める(B'00)	

【注】 *1 コンパイラは、関数先頭でこの丸めモードになっている前提でコード生成を行います。

(4) FPSCR.SZ(転送サイズモード)

コンパイラは、常に SZ=0(FMOV 命令の転送サイズは 32 ビット)を仮定しています。また、SZ ビットを変更するオブジェクトコードは生成しません。

(5) FPSCR.FR(FPU バンクレジスタ)

コンパイラは、FR ビットを変更するオブジェクトコードは生成しません。

ただし、組み込み関数 `st_ext()`、`ld_ext()` では、関数内部で一時的に FR ビットを変更します。リターン時の FR ビットは、呼び出し時と同じです。

以上