

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>

## — 日立マイクロコンピュータ技術情報 —

〒 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)-5201-5191

株式会社 日立製作所 半導体グループ

題 目	FPU倍精度演算命令使用上の注意		発行番号	TN-SH7-142A	
			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	
適 用 製 品	HD6417091BP200, HD6417091RBP200, HD6417750BP200, HD6417750F167, HD6417750VF128	対象ロット等	関 連 資 料	SH7091ハードウェアマニュアル, SH7091プログラミングマニュアル SH7750ハードウェアマニュアル, SH7750プログラミングマニュアル	有効期限
		全ロット			永年

## 1. 概要

倍精度FDIV, FADD, FSUB, FMULで非正規化数を入力とし、非正規化数を扱うモードにおいて演算結果が不正となる場合があります。

## 1.1 対象ユーザ

科学技術計算で厳密解を求めるプログラム開発者が対象です。倍精度浮動小数点命令を使用し、非正規化数を扱う場合に限られます。倍精度浮動小数点命令を使用するが、非正規化数を0として扱う場合、あるいは単精度浮動小数点命令しか使用しない場合は対象外です。

## 1.2 不良の概要

不良には2種類有り、非正規化数入力時に結果を誤るケース(a,b)、非正規化数とqNaN入力時に結果を誤るケース(c)があります。

(a)倍精度FDIVで、非正規化数を入力すると結果が誤って0または無限大となる場合があります。

(b)倍精度FMULで、非正規化数を入力すると結果が誤って無限大となる場合があります。

(c)倍精度FDIV, FADD, FSUB, FMULで、非正規化数とqNaNを入力すると誤って例外トラップが起動され、結果が不正となる場合があります。

## 1.3 不良の影響度

影響度が大きい不良として、倍精度FDIV, FMULで非正規化数入力時に誤った値をレジスタに書き込む場合(a,b)があります。特に、非正規化数/非正規化数=0, 非正規化数/0=0は数学的に不適切な値となります。

## 2. 対応策

通常は[1]、科学技術計算で非正規化数での厳密解を必要とする場合は[2]で対策して下さい。

[1]倍精度浮動小数点命令は、FPSCR.DN=1 すなわち非正規化数を0として扱うモードで使用します。  
本対策による性能低下はありません。

[2]非正規化数入力時に結果を誤るケース(a,b)は、ソフトウェアにより回避して下さい。詳細は、4.ソフトウェアの変更を参照してください。

(i)ソースかつディスティネーションとなるレジスタ(DRn)を退避。

(ii)倍精度FDIVで結果が0または無限大の場合、ユーザ定義の非正規化数処理用関数をコールする。

非正規化数とqNaN入力時に結果を誤るケース(c)は、TRAPルーチンにより回避して下さい。詳細は、5.TRAPルーチンの変更を参照してください。

(i)倍精度FDIV, FADD, FSUB, FMULで一方の入力が非正規化数で他方の入力がqNaNの場合、トラップルーチンでqNaN(64h'7fff\_ffff)がディスティネーションレジスタに書かれる。

### 3. 詳細

#### 3.1 定義

不良となるデータパターンを定義します。表中の(A)～(D)は下記データパターンに該当します。

◎倍精度非正規化数(A)

64h'00000000\_XXXXXXXX または 64h'80000000\_XXXXXXXX (X:0 or 1)

但し, 32h'XXXXXXXX!=32h'00000000

◎倍精度非正規化数(B)

64h'000YYYYY\_XXXXXXXX または 64h'800YYYYY\_XXXXXXXX (X:0 or 1)

但し, 20h'YYYYY!=20h'00000

◎倍精度qNaN(C)

64h'7ff00000\_XXXXXXXX または 64h'fff00000\_XXXXXXXX (X:0 or 1)

但し, 32h'XXXXXXXX!=32h'00000000

◎倍精度qNaN(D) \*定義通り

64h'7ffXXXXX\_XXXXXXXX または 64h'fffXXXXX\_XXXXXXXX (X:0 or 1)

但し, 52h'XXXXX\_XXXXXXXX!=52h'00000\_00000000

#### 3.2 不良の詳細

FPSCR.DN=1'b0(非正規化数をそのまま扱うモード)において、不正な演算結果となる命令及びデータの組合わせを表1に示します。

入力(A)～(C)は3.1で定義されたデータパターン、NG typeの(1)～(7)は表2～4の不正な演算結果をタイプ別に分類してあります。

(1),(2),(3),(7)は不正な演算結果で0または無限大となります。

(4),(5),(6)はFPU Errorの例外トラップが発生し、qNaNを出力しません。

1.2不良の概要の項の(a)は表1の(1),(2),(3)に、(b)は(7)に、(c)は(4),(5),(6)に相当します。

表1 不正な演算結果

NG type	命令	入力		SH-4	期待値
		DRm	DRn		
(1)	FDIV	+0/-0	(A)DENORM	+0/-0	DZ
(2)	FDIV	(A)DENORM	+0/-0	+0/-0	FPU Error
		(A)DENORM	(A)DENORM		
(3)	FDIV	(A)DENORM	+INF/-INF	+INF/-INF	FPU Error
(4)	FDIV	(C)qNaN	(A)DENORM	FPU Error	qNaN*
		(C)qNaN	(B)DENORM		
		(B)DENORM	(C)qNaN		
(5)	FADD/FSUB	(C)qNaN	DENORM	FPU Error	qNaN*
		DENORM	(C)qNaN		
(6)	FMUL	(C)qNaN	(B)DENORM	FPU Error	qNaN*
		(B)DENORM	(C)qNaN		
(7)	FMUL	(A)DENORM	+INF/-INF	+INF/-INF	FPU Error
		+INF/-INF	(A)DENORM		

\*qNaN:64h'7ff7ffff\_ffffff

FPSCR.DN=1'b1(非正規化数を0として扱うモード)の場合は全て正常に動作します。

倍精度FDIV, FADD, FSUB, FMUL命令での特殊ケースをまとめます。

- 点箱部は正常に動作します。
- 白箱部はSH-4が出力する値であり、不正な演算結果となります。

表2 FDIV DRm, DRn (DRn / DRm → DRn)

DRm \ DRn	NORM	+0	-0	+INF	-INF	(A) positive DENORM	(A) negative DENORM	(B) DENORM	(C) qNaN	(D) qNaN	sNaN	
NORM	DIV	0		INF		Error		■	■	■	■	
+0	DZ	Invalid		+INF	-INF	+0 (1)	-0 (1)					DZ
-0				-INF	+INF	-0 (1)	+0 (1)					
+INF	0	+0	-0	Invalid		Error						qNaN
-INF		-0	+0									
(A) positive DENORM		+0(2)	-0(2)	(3) +INF	(3) -INF	+0 (2)	-0 (2)					
(A) negative DENORM		-0(2)	+0(2)	(3) -INF	(3) +INF	-0 (2)	+0 (2)					
(B) DENORM	■								Error (4)			
(C) qNaN	■								Error (4)			
(D) qNaN	■											
sNaN	■											

表3 FADD DRm, DRn (DRn + DRm → DRn) FSUB DRm, DRn (DRn - DRm → DRn)

DRm \ DRn	NORM	+0	-0	+INF	-INF	(A) positive DENORM	(A) negative DENORM	(B) DENORM	(C) qNaN	(D) qNaN	sNaN		
NORM	ADD	0		+INF	-INF	Error		■	■	■	■		
+0		+0										qNaN	Invalid
-0			-0										
+INF				Invalid									
-INF	INF			Invalid	-INF								
(A) positive DENORM	■								Error (5)				
(A) negative DENORM	■												
(B) DENORM	■												
(C) qNaN	■								Error (5)				
(D) qNaN	■												
sNaN	■												

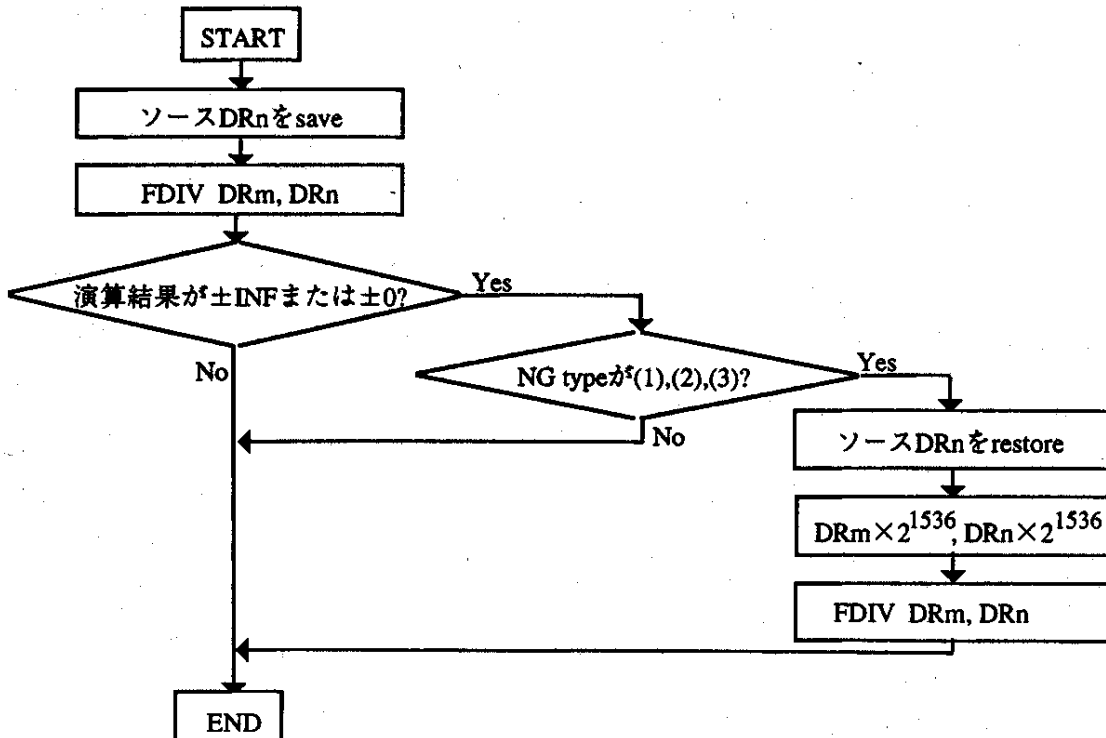
表4 FMUL DRm, DRn (DRn\*DRm→DRn)

DRm \ DRn	NORM	+0	-0	+INF	-INF	(A)positive DENORM	(A)negative DENORM	(B) DENORM	(C)qNaN	(D)qNaN	sNaN	
NORM	MUL	0	0	INF	INF	Error						
+0	0	+0	-0	Invalid	Invalid							
-0	0	-0	+0	Invalid	Invalid							
+INF	INF	Invalid	Invalid	+INF	-INF	+INF(7)	-INF(7)	qNaN				Invalid
-INF	INF	Invalid	Invalid	-INF	+INF	-INF(7)	+INF(7)	qNaN				Invalid
(A) positive DENORM				+INF(7)	-INF(7)							
(A) negative DENORM				-INF(7)	+INF(7)							
(B) DENORM									Error(6)			
(C) qNaN									Error(6)			
(D) qNaN									Error(6)			
sNaN												

#### 4. ソフトウェアの変更

##### 4.1 NG typeが(1),(2),(3)の場合

表1のNG type(1),(2),(3)は以下のフローに従ってソフトウェアにて対策してください。  
 ソースオペランドに定数 $2^{1536}$ を乗じて補正し、正規化数として演算してください。  
 NG typeが(1)で、ゼロ除算例外イネーブルの場合、ゼロ除算による例外トラップが発生し、ディステーションレジスタは変更しません。ゼロ除算例外ディスエーブルの場合、ディステーションレジスタは入力の符号に基づいた無限大となります。



#### 4.2 NG typeが(7)の場合

表1のNG type(7)はFPU Errorは発生しませんが、演算結果は正しく、ソフトウェアによる対策は必要有りません。

#### 5. TRAPルーチンの変更

表1のNG type(4),(5),(6)の対策として、表5に示すようにTRAPルーチンで命令と入力データのチェックを行ない、qNaNをディスティネーションレジスタに書き込むように処理を追加してください。

ここでqNaNは常に64h'7ff7ffff\_ffffffffにして下さい。

表5 TRAPルーチンでの処理

NG type	命令チェック	入力チェック		演算結果
		DRm	DRn	
(4)	FDIV	qNaN	DENORM	qNaN
	FDIV	qNaN	DENORM	qNaN
	FDIV	DENORM	qNaN	qNaN
(5)	FADD/FSUB	qNaN	DENORM	qNaN
	FADD/FSUB	DENORM	qNaN	qNaN
(6)	FMUL	qNaN	DENORM	qNaN
	FMUL	DENORM	qNaN	qNaN