

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が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。

H8/300H Tiny シリーズ

単精度浮動小数点の加算

要旨

汎用レジスタに設定された単精度浮動小数点の加算を行ない、加算結果を汎用レジスタに設定します。

動作確認デバイス

H8/300H Tiny シリーズ

目次

1. 機能	2
2. 引数	2
3. 内部レジスタ変化およびフラグ変化.....	2
4. プログラミング仕様.....	3
5. 注意事項	3
6. 説明	4
7. フローチャート	7
8. プログラムリスト.....	14
< 参考 > 単精度浮動小数点フォーマットについて	18

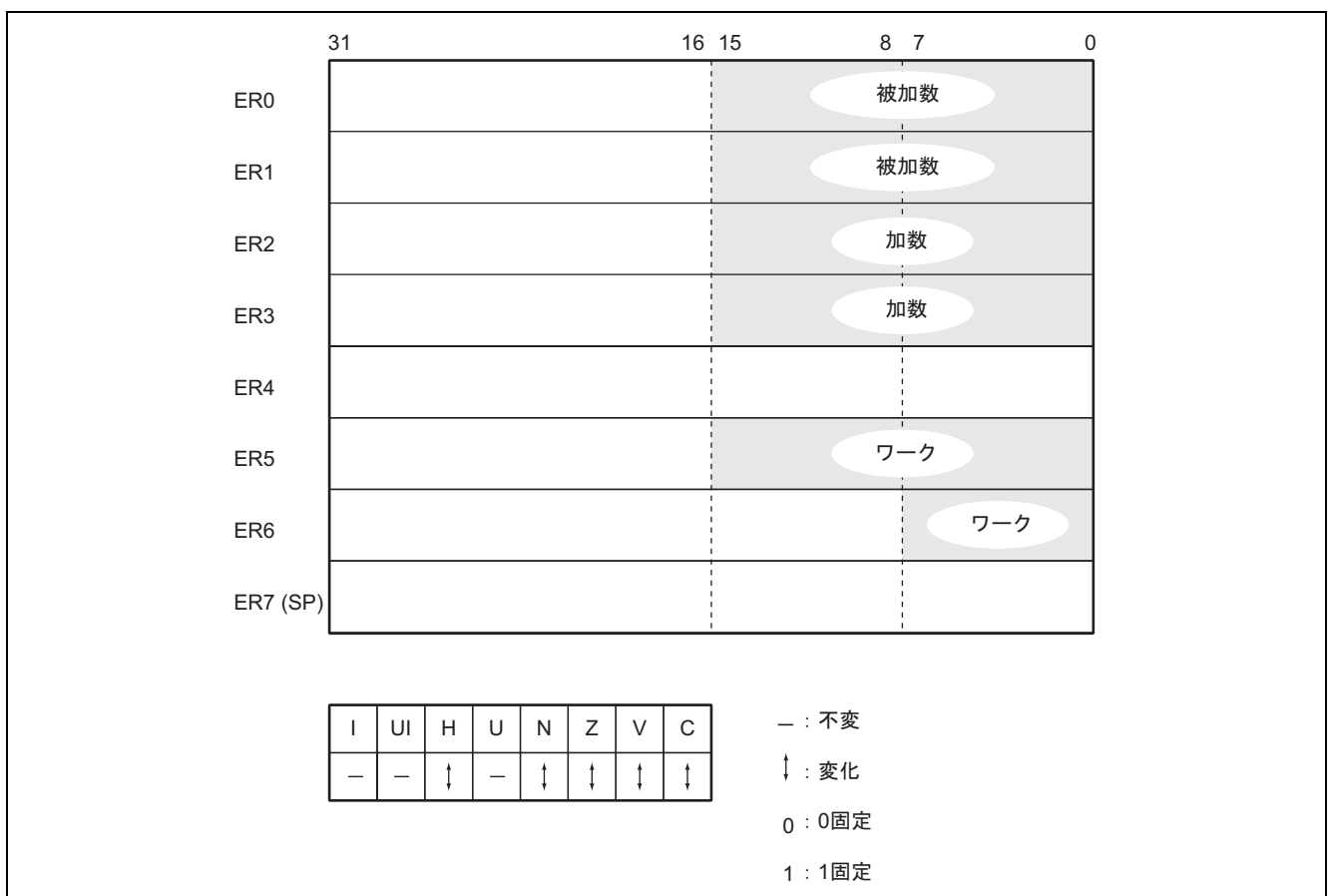
1. 機能

- (1) 汎用レジスタに設定された単精度浮動小数点の加算を行い、加算結果を汎用レジスタに設定します。
- (2) 引数は、すべて単精度浮動小数点フォーマットです。

2. 引数

	内容	格納場所	データ長
入力	被加数	R0, R1	4
	加数	R2, R3	4
出力	加算結果	R0, R1	4

3. 内部レジスタ変化およびフラグ変化



4. プログラミング仕様

プログラムメモリ (バイト)
280
データメモリ (バイト)
0
スタック (バイト)
0
ステート数
268
リエントラント
可
リロケーション
可
途中割り込み
可

5. 注意事項

プログラミング仕様のステート数は、図 1 の例を実行した場合の値です。

浮動小数点フォーマットに関しては「＜参考＞ 単精度浮動小数点フォーマットについて」を参照してください。

6. 説明

6.1 機能説明

(1) 引数の詳細は以下のとおりです。

(a) 入力引数は、次のように設定します。

R0：被加数の上位 2 バイト

R1：被加数の下位 2 バイト

R2：加数の上位 2 バイト

R3：加数の下位 2 バイト

(b) 出力引数は、次のように設定します。

R0：加算結果の上位 2 バイト

R1：加算結果の下位 2 バイト

(2) 図 1 にソフトウェア FADD の実行例を示します。

入力引数を図 1 のように設定すると、図 1 に示すように、加算結果が R0, R1 に設定されます。

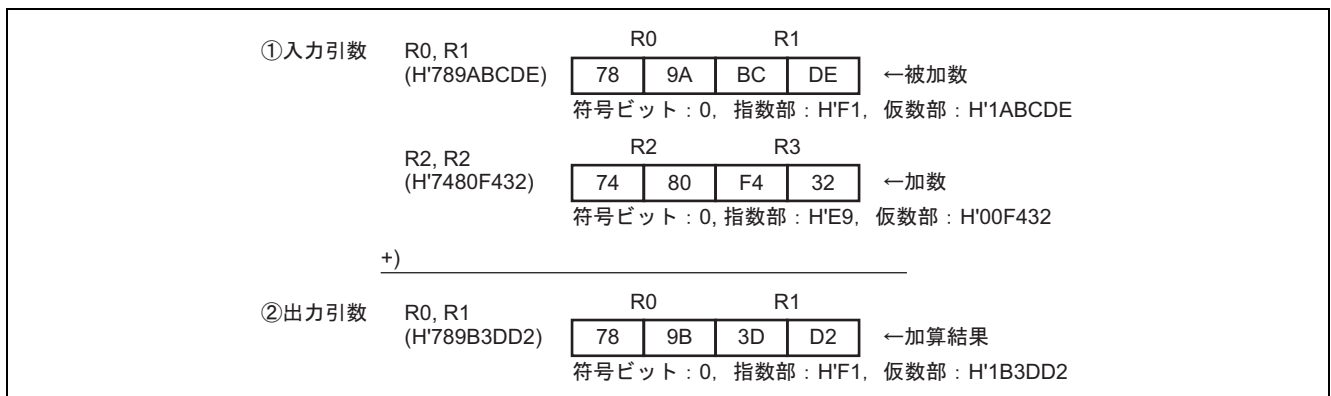


図 1 ソフトウェア FADD の実行例

6.2 使用上の注意

- (1) ソフトウェア FADD で扱える最大値，最小値は，以下となります。
 正の最大値：H'7F80000
 正の最小値：H'00000001
 負の最大値：H'80000001
 負の最小値：H'FF800000
- (2) 単精度の正の数 H'7F800001 ~ H'7FFFFFFF は，すべて最大値 (H'7F800000) として扱います。負の数 H'FF800000 ~ H'FFFFFFF は，すべて最小値 (H'FF800000) として扱います。
- (3) 最大値は無限大 (∞) として取り扱いますので， $\infty + 100$ ， $\infty - 100$ の結果も ∞ となります。
 (表 1 参照)

表 1 最大値を引数とした演算例

被加数	加数	結果
H'7F800000 ~ 7FFFFFFF	H'*****	H'7F800000
H'7F800000 ~ FFFFFFFF 以外	H'7F800000 ~ 7FFFFFFF	H'7F800000
H'FF800000 ~ FFFFFFFF	H'*****	H'FF800000
H'7F800000 ~ 7FFFFFFF 以外	H'7F800000 ~ 7FFFFFFF	H'FF800000

【注】 H'***** は，16 進数

- (4) H'80000000 は，H'00000000 (ゼロ) として扱います。
- (5) ソフトウェア FADD 実行後，汎用レジスタの被加数，加数データは破壊されます。
 実行後も入力引数を必要とする場合は，あらかじめメモリ上に退避してください。

6.3 データメモリの説明

ソフトウェア FADD では，データメモリを使用していません。

6.4 使用例

被加数および加数を設定し，ソフトウェア FADD をサブルーチンコールします。

```

WORK1  .RES.W 2      ..... 被加数を設定するデータメモリエリア
WORK2  .RES.W 2      ..... 加数を設定するデータメモリエリア
WORK3  .RES.W 2      ..... 加算結果を設定するデータメモリエリア
      .
      .
      .
MOV.W  @WORK1,R0     ..... ユーザプログラムで設定した被加数を入力引数に設定
MOV.W  @WORK1+2,R1
MOV.W  @WORK2,R2     ..... ユーザプログラムで設定した加数を入力引数に設定
MOV.W  @WORK2+2,R3
JSR    @FADD          ..... ソフトウェアFADDをサブルーチンコール
MOV.W  R0,@WORK3     ..... 出力引数に設定された加算結果を格納します。
MOV.W  R1,@WORK3+2
  
```

6.5 動作原理

単精度浮動小数点の加算は、次のように行います。

(1) 被加数, 加数が $+\infty$, $-\infty$ であるか判定します。

(a) 被加数の指数部が H'FF のとき以下を出力します。

符号ビット	出力値
0 (正)	H'7F800000 ($+\infty$)
1 (負)	H'FF800000 ($-\infty$)

(b) 被加数が $+\infty$, $-\infty$ でなく加数の指数部が H'FF のとき, (1)-(a)と同様にします。

(2) 被加数, 加数が “0” であるか判定します。

(a) 被加数, 加数のどちらかが “0” であれば “0” でない方の値を出力値とします。

(両方とも “0” であれば, “H'00000000” を出力値とします。)

(3) 被加数, 加数の指数部の値を合わせます。

(a) 被加数, 加数の指数部が一致するまで, 指数部の小さい方の数をインクリメントし, それと同時に仮数部 (インプリシット MSB も含めて) を 1 桁ずつ右にシフトします。

(非正規化表現の場合, 指数部にあらかじめ 1 を加算し, 仮数部のインプリシット MSB は, 0 として処理します。)

(4) 仮数部の加算を行います。

(5) 演算結果を浮動小数点フォーマットにします。

(例) 被加数 = $1.20888876915 \times 2^{14}$
(H'789ABCDE)
符号ビット = 0, 指数部 = H'F1, 仮数部 = H'1ABCDE
(インプリシット MSB は含まない)

加数 = $1.21282410622 \times 2^{-17}$
(H'7A1B3DD2)
符号ビット = 0, 指数部 = H'F4, 仮数部 = H'1B3DD2
(インプリシット MSB は含まない)

インプリシット MSB

被加数	1 1 1 1 0 0 0 1	1. 0 0 0 1 1 0 1 0 1 0 1 1 1 0 0 1 1 0 1 1 1 0
	H'F1	H'9ABCDE

加数	1 1 1 1 0 1 0 0	1. 0 0 0 1 1 0 1 1 0 0 1 1 1 1 0 1 1 1 0 1 0 0 1 0
	H'F4	H'9B3DD2

	↓	指数部を合わせる (被加数の指数部に3を加える)	↓	被加数の仮数を3ビット右シフト
--	---	-----------------------------	---	-----------------

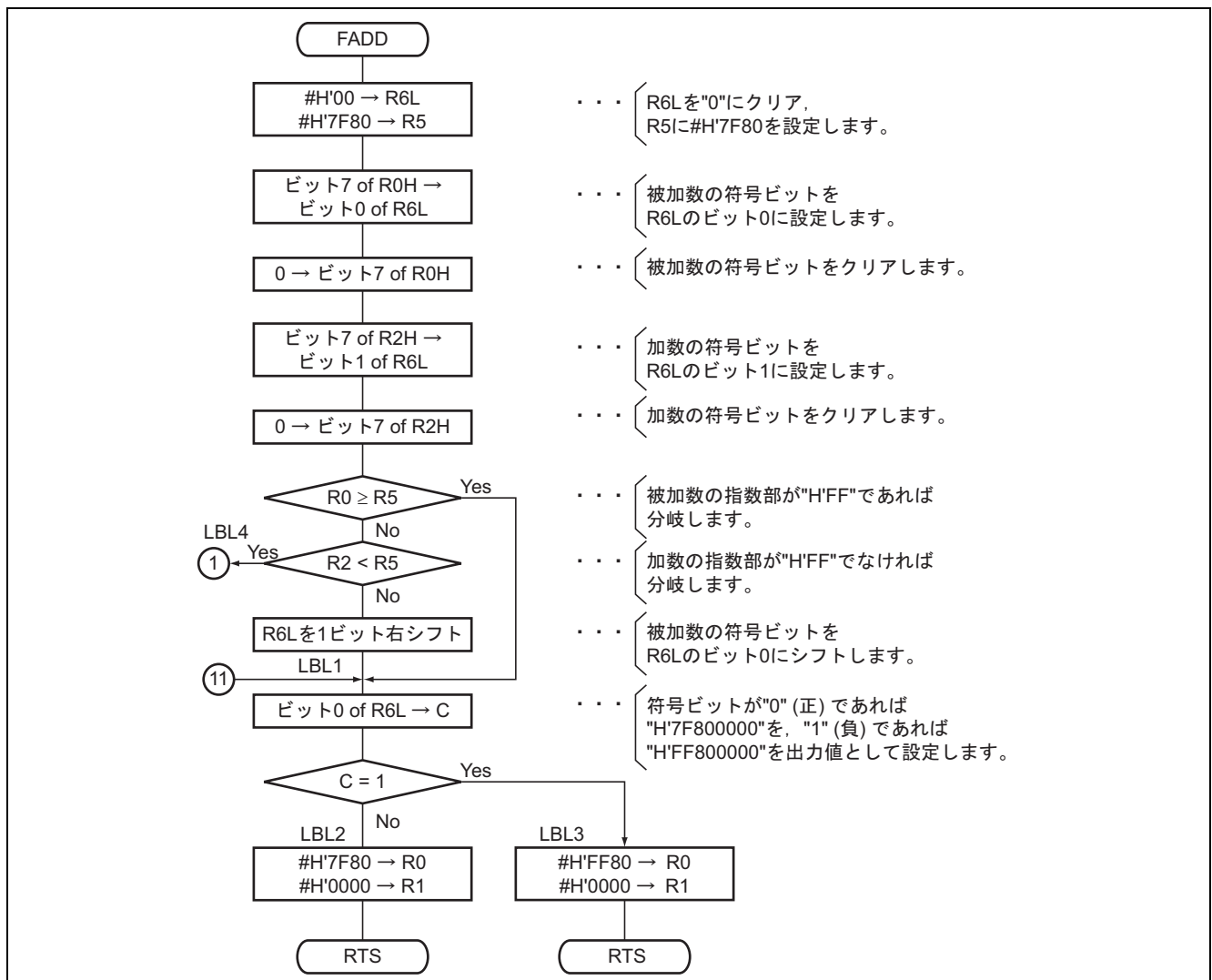
被加数	1 1 1 1 0 1 0 0	0. 0 0 1 0 0 1 1 0 1 0 1 0 1 1 1 1 0 0 1 1 0 1 1
加数	1 1 1 1 0 1 0 0	1. 0 0 0 1 1 0 1 1 0 0 1 1 1 1 0 1 1 1 0 1 0 0 1 0

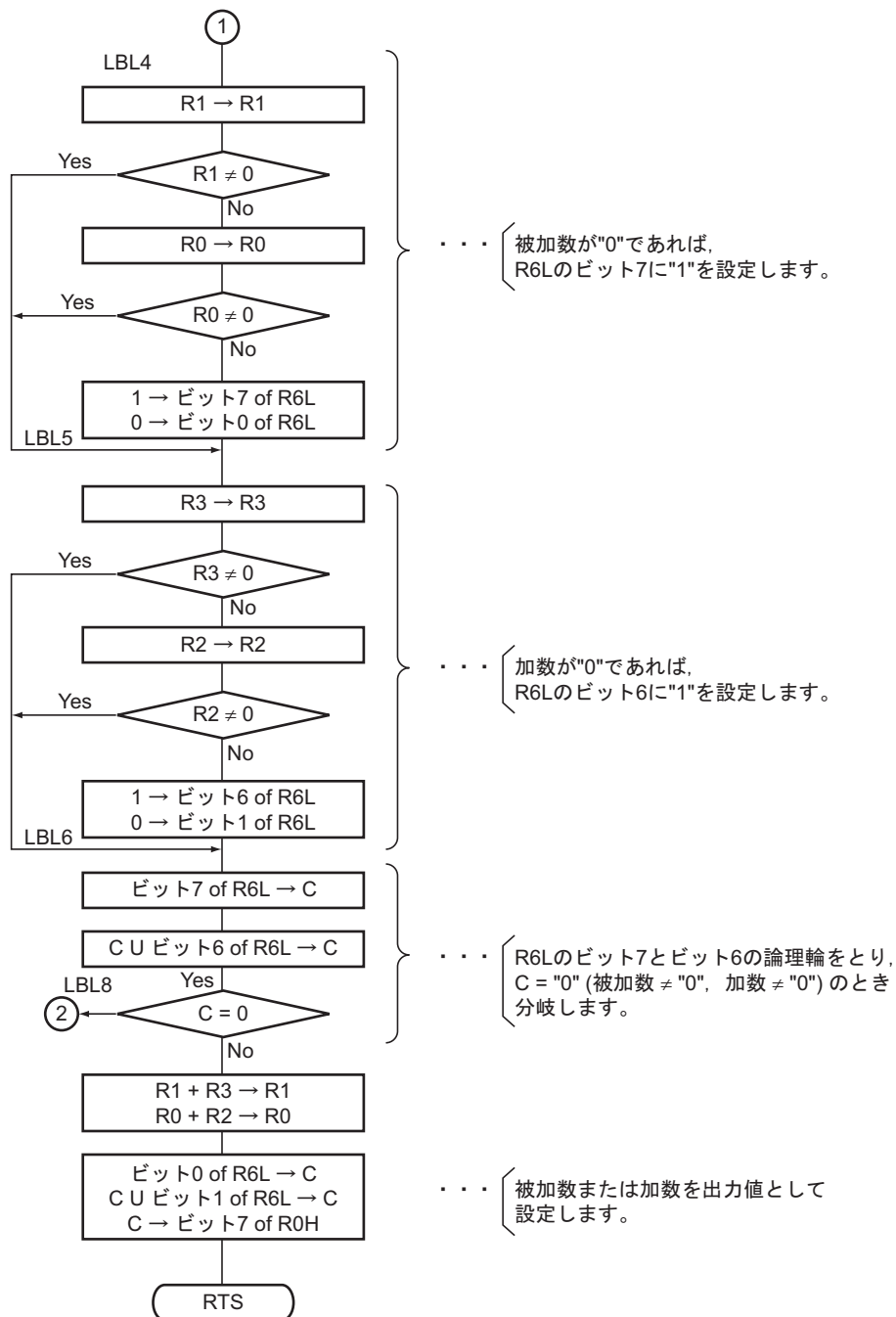
+)

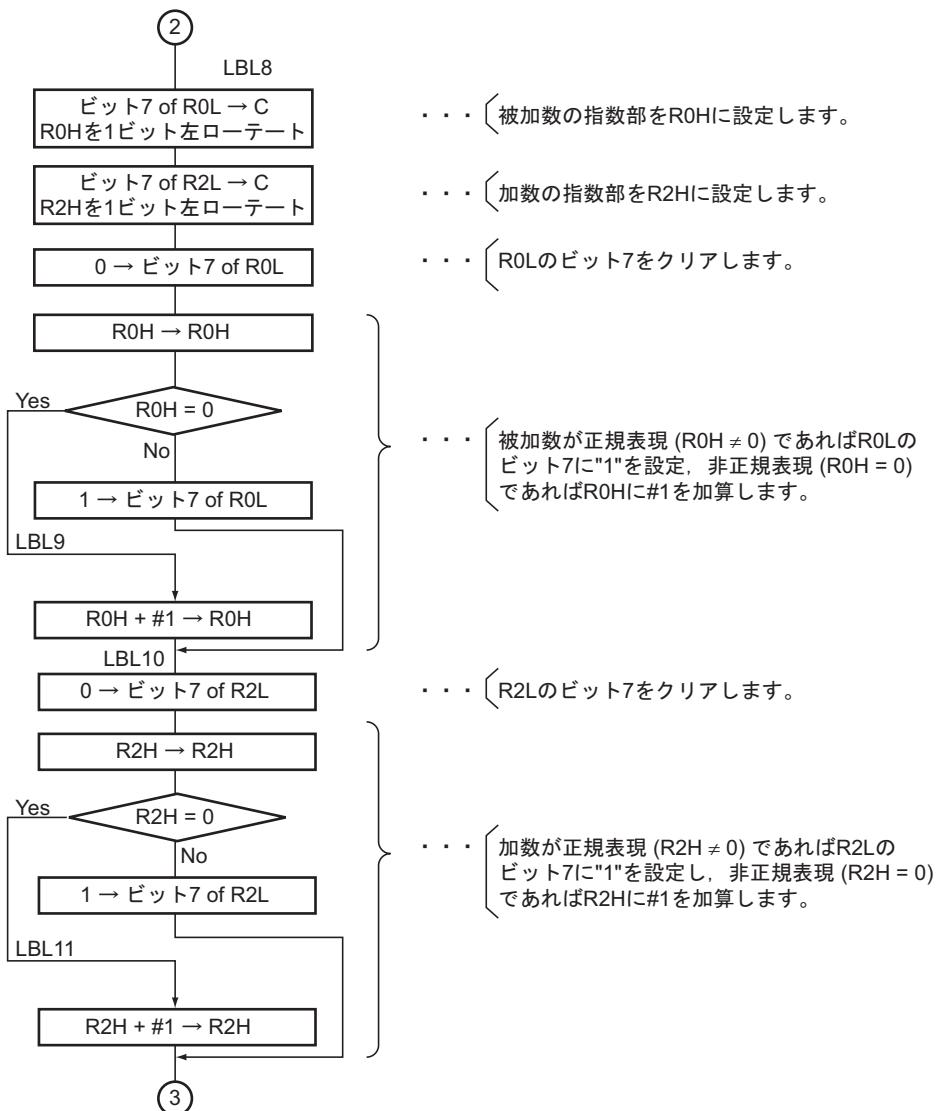
加算結果	1 1 1 1 0 1 0 0	1. 0 1 0 1 1 0 1 0 0 1 0 1 0 1 0 1 0 1 1 0 1 1 0 1		
	↓	指数部はそのまま	↓	仮数部のみ加算処理を行う

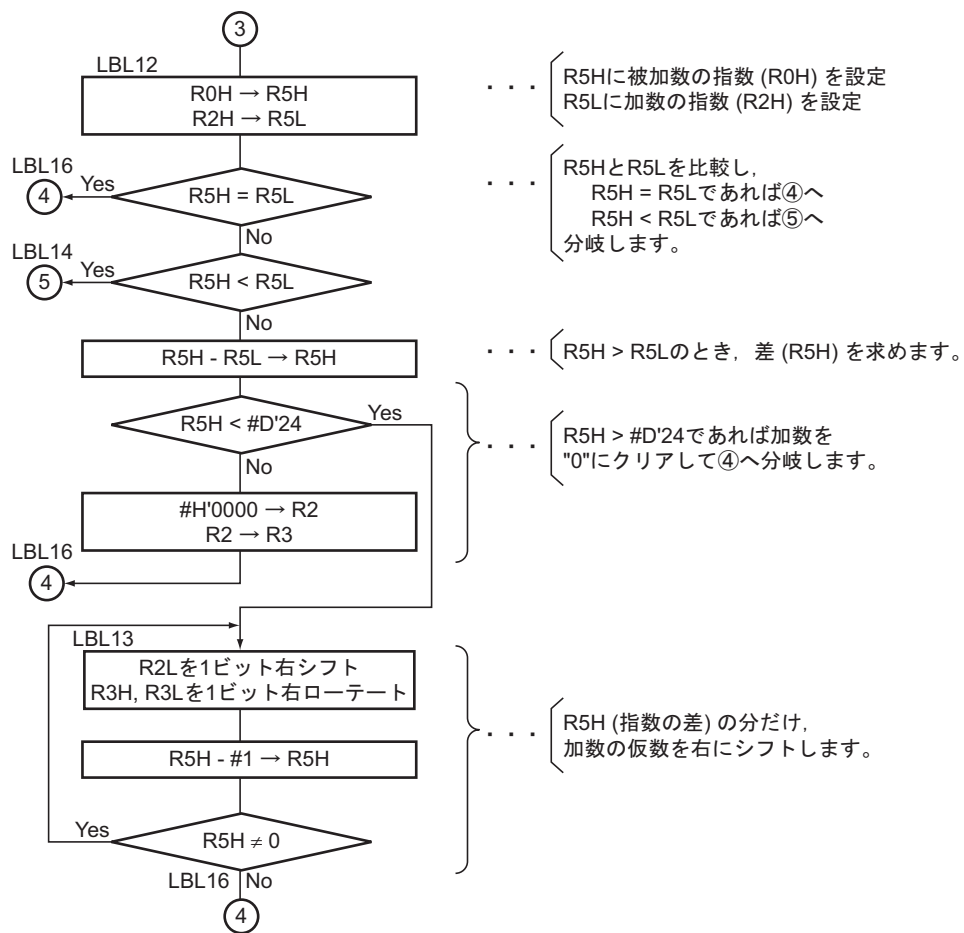
加算結果 = $1.36393511295 \times 2^{-17}$
(H'7A2E956D)
符号ビット = 0, 指数部 = H'F4, 仮数部 = H'2E956D
(インプリシット MSB は含まない)

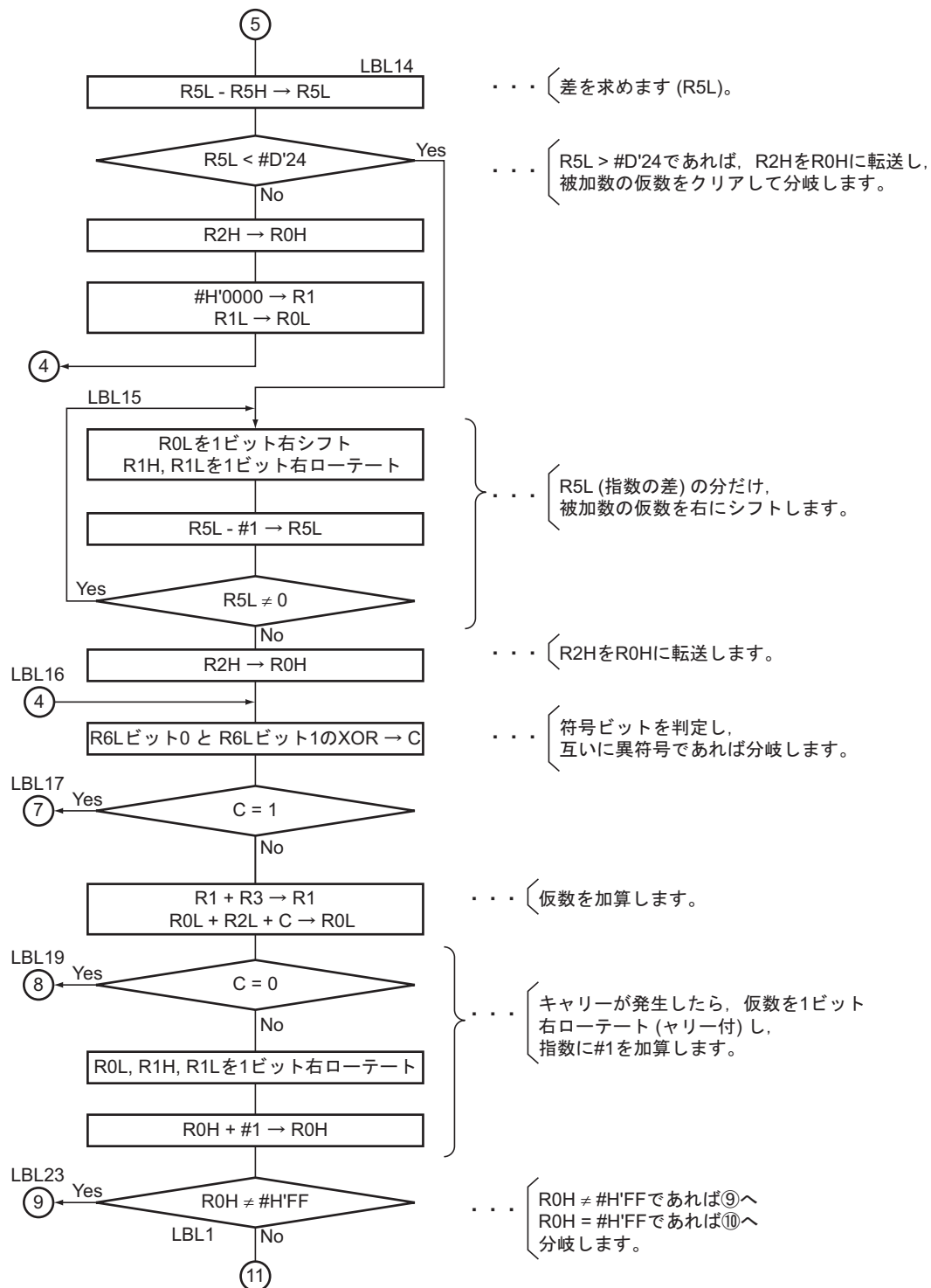
7. フローチャート

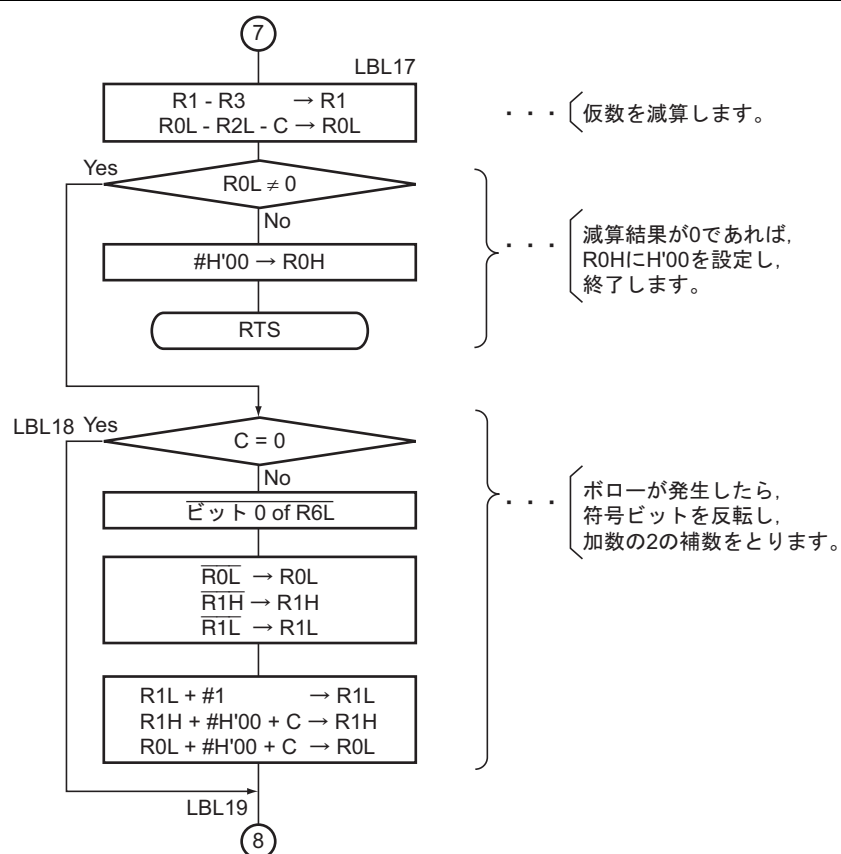


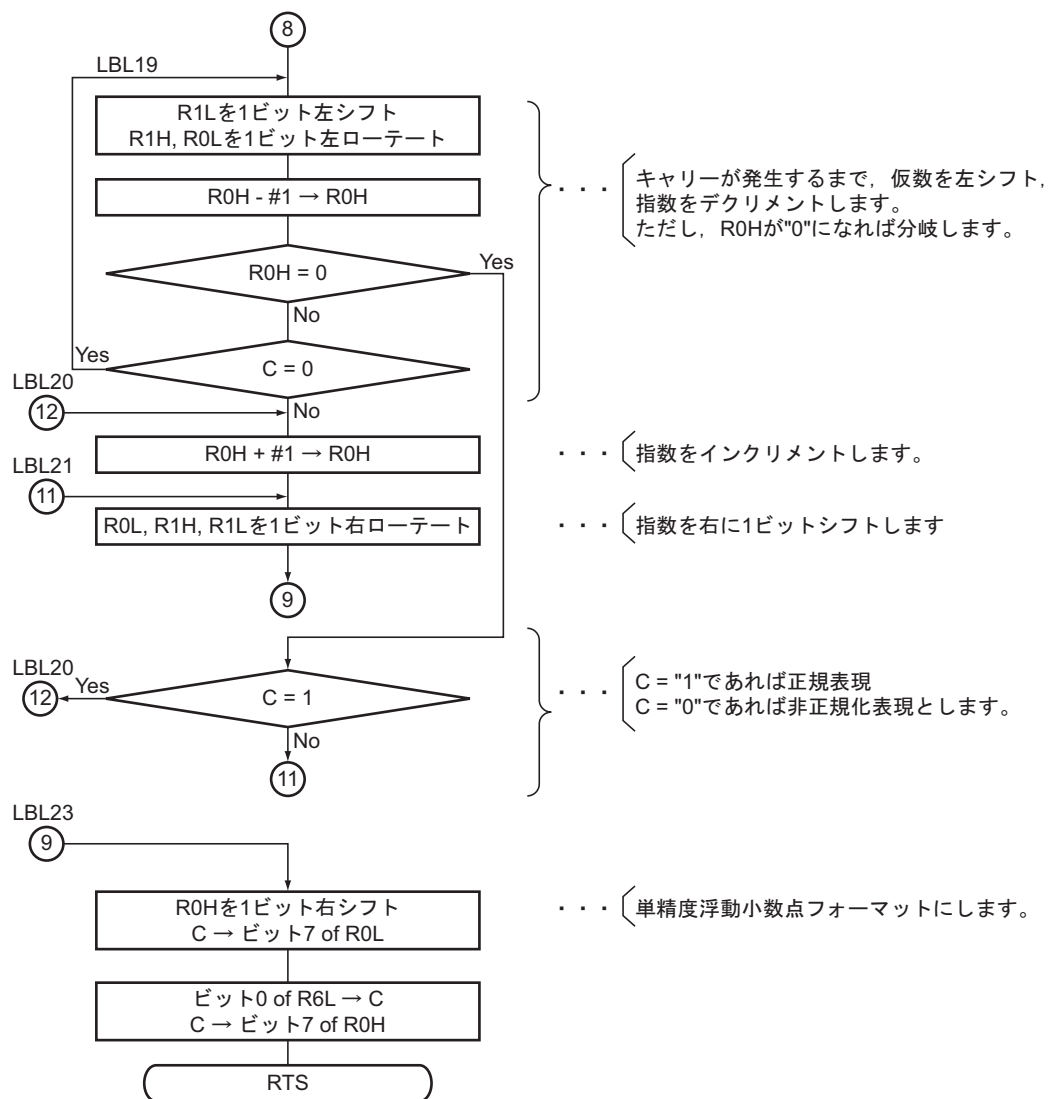












8. プログラムリスト

```

1          1  ;*****
2          2  ;*
3          3  ;*      NAME:    FLOATING POINT ADDITION
4          4  ;*                      (FADD)
5          5  ;*
6          6  ;*****
7          7  ;*
8          8  ;*      ENTRY:    R0      (HIGHER WORD OF AUGEND)
9          9  ;*                      R1      (LOWER WORD OF AUGEND)
10         10 ;*                      R2      (HIGHER WORD OF ADDEND)
11         11 ;*                      R3      (LOWER WORD OF ADDEND)
12         12 ;*
13         13 ;*      RETURNS:  R0      (HIGHER WORD OF RESULT)
14         14 ;*                      R1      (LOWER WORD OF RESULT)
15         15 ;*
16         16 ;*****
17         17 ;
18         18          .CPU      300HN
19 0000      19          .SECTION FADD_code, CODE, ALIGN=2
20         20          .EXPORT  FADD
21         21          ;
22          22 FADD      .EQU    $          ;Entry point
23 0000 FE00      23      MOV.B   #H'00, R6L      ;Clear R0L
24 0002 7905F80   24      MOV.W   #H'7F80, R5      ;Set H'7F80
25         25      ;
26 0006 7770      26          BLD    #7, R0H
27 0008 670E      27          BST    #0, R6L      ;Set sign bit to bit 0 of R6L
28 000A 7270      28          BCLR   #7, R0H      ;Bit clear bit 7 of R0H
29         29      ;
30 000C 7772      30          BLD    #7, R2H      ;Set sign bit to bit 1 of R6L
31 000E 671E      31          BST    #1, R6L
32 0010 7272      32          BCLR   #7, R2H      ;Bit Clear bit 7 of R2H
33         33      ;
34 0012 1D05      34          CMP.W   R0, R5
35 0014 4306      35          BLS     LBL1      ;Branch if "exponent of augend"= "H'FF"
36 0016 1D25      36          CMP.W   R2, R5
37 0018 421A      37          BHI     LBL4      ;Branch if not "exponent of augend"= "H'FF"
38 001A 110E      38          SHLR    R6L      ;Shift R6L 1 bit right
39 001C           39 LBL1
40 001C 770E      40          BLD    #0, R6L      ;Bit load sign bit
41 001E 450A      41          BCS     LBL3      ;Branch if sign bit=1
42 0020           42 LBL2
43 0020 7900F80   43          MOV.W   #H'7F80, R0      ;Set maximum pos. number
44 0024 79010000   44          MOV.W   #H'0000, R1
45 0028 5470      45          RTS
46 002A           46 LBL3
47 002A 7900FF80   47          MOV.W   #H'FF80, R0      ;Set maximum neg. number
48 002E 79010000   48          MOV.W   #H'0000, R1
49 0032 5470      49          RTS
50         50      ;
51 0034           51 LBL4
52 0034 0D11      52          MOV.W   R1, R1
53 0036 4608      53          BNE     LBL5      ;Branch if Z=0
54 0038 0D00      54          MOV.W   R0, R0
55 003A 4604      55          BNE     LBL5      ;Branch if Z=0

```

56	003C 707E	56	BSET	#7,R6L	;Bit set bit 7 of R6L
57	003E 720E	57	BCLR	#0,R6L	;Bit clear bit 0 of R6L
58	0040	58	LBL5		
59	0040 0D33	59	MOV.W	R3,R3	
60	0042 4608	60	BNE	LBL6	;Branch if Z=0
61	0044 0D22	61	MOV.W	R2,R2	
62	0046 4604	62	BNE	LBL6	;Branch if Z=0
63	0048 706E	63	BSET	#6,R6L	;Bit set bit 6 of R6L
64	004A 721E	64	BCLR	#1,R6L	;Bit clear bit 1 of R6L
65	004C	65	LBL6		
66	004C 777E	66	BLD	#7,R6L	
67	004E 746E	67	BOR	#6,R6L	
68	0050 440C	68	BCC	LBL8	;Branch if not augend=addend=0
69	0052 0931	69	ADD.W	R3,R1	;Set augend and addend to result
70	0054 0920	70	ADD.W	R2,R0	
71	0056 770E	71	BLD	#0,R6L	
72	0058 741E	72	BOR	#1,R6L	
73	005A 6770	73	BST	#7,R0H	;Set sign bit
74	005C 5470	74	RTS		
75		75	;		
76	005E	76	LBL8		
77	005E 7778	77	BLD	#7,R0L	
78	0060 1200	78	ROTXL	R0H	;Set exponent of augend to R0H
79		79	;		
80	0062 777A	80	BLD	#7,R2L	
81	0064 1202	81	ROTXL	R2H	;Set exponent of addend to R0L
82		82	;		
83	0066 7278	83	BCLR	#7,R0L	
84	0068 0C00	84	MOV.B	R0H,R0H	
85	006A 4704	85	BEQ	LBL9	;Branch if augend is normalized
86	006C 7078	86	BSET	#7,R0L	;Set implicit MSB to augend
87	006E 4002	87	BRA	LBL10	;Branch always
88	0070	88	LBL9		
89	0070 8001	89	ADD.B	#H'01,R0H	
90	0072	90	LBL10		
91	0072 727A	91	BCLR	#7,R2L	
92	0074 0C22	92	MOV.B	R2H,R2H	
93	0076 4704	93	BEQ	LBL11	;Branch if addend is normalized
94	0078 707A	94	BSE	#7,R2L	;Set implicit MSB to augend
95	007A 4002	95	BRA	LBL12	;Branch always
96	007C	96	LBL11		
97	007C 8201	97	ADD.B	#H'01,R2H	
98		98	;		
99	007E	99	LBL12		
100	007E 0C05	100	MOV.B	R0H,R5H	
101	0080 0C2D	101	MOV.B	R2H,R5L	
102	0082 1CD5	102	CMP.B	R5L,R5H	
103	0084 4738	103	BEQ	LBL16	;Branch if R5H = R5L
104	0086 451A	104	BCS	LBL14	;Branch if R5H < R5L
105		105	;		
106	0088 18D5	106	SUB.B	R5L,R5H	
107	008A A518	107	CMP.B	#D'24,R5H	;Set bit counter
108	008C 4508	108	BCS	LBL13	;Branch if R5H < D'24
109	008E 79020000	109	MOV.W	#H'0000,R2	;Clear addend
110	0092 0D23	110	MOV.W	R2,R3	
111	0094 4028	111	BRA	LBL16	;Branch always
112	0096	112	LBL13		
113	0096 110A	113	SHLR	R2L	;Shift mantissa of addend 1 bit left

114	0098 1303	114	ROTXR	R3H	
115	009A 130B	115	ROTXR	R3L	
116	009C 1A05	116	DEC.B	R5H	;Decrement bit counter
117	009E 46F6	117	BNE	LBL13	;Branch Z=0
118	00A0 401C	118	BRA	LBL16	;Branch always
119	00A2	119	LBL14		
120	00A2 185D	120	SUB.B	R5H,R5L	
121	00A4 AD18	121	CMP.B	#D'24,R5L	
122	00A6 450A	122	BCS	LBL15	;Branch if R5L<D'24
123	00A8 0C20	123	MOV.B	R2H,R0H	
124	00AA 79010000	124	MOV.W	#H'0000,R1	;Clear augend
125	00AE 0C98	125	MOV.B	R1L,R0L	
126	00B0 400C	126	BRA	LBL16	;Branch always
127		127	;		
128	00B2	128	LBL15		
129	00B2 1108	129	SHLR	R0L	;Shift mantissa of augend 1 bit right
130	00B4 1301	130	ROTXR	R1H	
131	00B6 1309	131	ROTXR	R1L	
132	00B8 1A0D	132	DEC.B	R5L	;Decrement bit counter
133	00BA 46F6	133	BNE	LBL15	;Branch if Z=0
134	00BC 0C20	134	MOV.B	R2H,R0H	
135		135			
136		136	;		
137	00BE	137	LBL16		
138	00BE 770E	138	BLD	#0,R6L	
139	00C0 751E	139	BXOR	#1,R6L	
140	00C2 4516	140	BCS	LBL17	;Branch if different sign bit
141		141	;		
142	00C4 0931	142	ADD.W	R3,R1	;Addition mantissa
143	00C6 0EA8	143	ADDX.B	R2L,R0L	
144	00C8 442A	144	BCC	LBL19	;Branch if C=0
145	00CA 1308	145	ROTXR	R0L	;Rotate mantissa 1 bit right
146	00CC 1301	146	ROTXR	R1H	
147	00CE 1309	147	ROTXR	R1L	
148	00D0 8001	148	ADD.B	#H'01,R0H	;Increment exponent
149	00D2 A0FF	149	CMP.B	#H'FF,R0H	
150	00D4 4638	150	BNE	LBL23	;Branch if not exponent=H'FF
151	00D6 5A000000	151	JMP	@LBL1	;Jump
152	00DA	152	LBL17		
153	00DA 1931	153	SUB.W	R3,R1	;Subtract mantissa
154	00DC 1EA8	154	SUBX.B	R2L,R0L	
155	00DE 4604	155	BNE	LBL18	;Branch if Z=0
156	00E0 F000	156	MOV.B	#H'00,R0H	;Clear R0H
157	00E2 5470	157	RTS		
158	00E4	158	LBL18		
159	00E4 440E	159	BCC	LBL19	;Branch if C=0
160	00E6 710E	160	BNOT	#0,R6L	;Bit not sign bit
161	00E8 1708	161	NOT	R0L	;2's complement mantissa
162	00EA 1701	162	NOT	R1H	
163	00EC 1709	163	NOT	R1L	
164	00EE 8901	164	ADD.B	#H'01,R1L	
165	00F0 9100	165	ADDX.B	#H'00,R1H	
166	00F2 9800	166	ADDX.B	#H'00,R0L	
167		167	;		
168	00F4	168	LBL19		
169	00F4 1009	169	SHLL	R1L	;Shift mantissa 1 bit left
170	00F6 1201	170	ROTXL	R1H	
171	00F8 1208	171	ROTXL	R0L	

```

172 00FA 1A00      172      DEC.B   R0H      ;Decrement exponent
173 00FC 470C      173      BEQ     LBL22    ;Branch if exponent = 0
174 00FE 44F4      174      BCC     LBL19    ;Branch if exponent > 0
175 0100           175 LBL20
176 0100 0A00      176      INC.B   R0H      ;Increment exponent
177 0102           177 LBL21
178 0102 1308      178      ROTXR   R0L      ;Rotate mantissa 1 bit right
179 0104 1301      179      ROTXR   R1H
180 0106 1309      180      ROTXR   R1L
181 0108 4004      181      BRA     LBL23    ;Branch always
182 010A           182 LBL22
183 010A 45F4      183      BCS     LBL20    ;Branch if C=1
184 010C 40F4      184      BRA     LBL21    ;Branch always
185              185 ;
186 010E           186 LBL23    ;Correct into floating-point format
187 010E 1100      187      SHLR    R0H
188 0110 6778      188      BST     #7,R0L
189 0112 770E      189      BLD     #0,R6L
190 0114 6770      190      BST     #7,R0H
191 0116 5470      191      RTS
192              192 ;
193              193      .END
*****TOTAL ERRORS      0
*****TOTAL WARNINGS    0

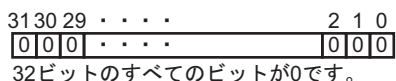
```

< 参考 > 単精度浮動小数点フォーマットについて

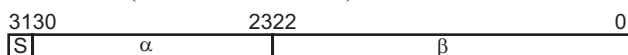
(1) 単精度浮動小数点の内部表現について

本アプリケーションノートでは、単精度浮動小数点の値により、以下の表現方法を用います。
(実数を R とする。)

(a) R = 0 のときの内部表現



(b) 正規表現 (Normalized Format)



ここでαはフィールド長が8ビットの指数です。βはフィールド長が23ビットの仮数です。
この時の R の値は、次の式で表現できます。 ($1 \leq \alpha \leq 254$ の場合)

$$\downarrow \text{インプリシット MSB}$$

$$R = 2^S \times 2^{\alpha-126} \times (1 + 2^{-1} \times \beta_{22} + 2^{-2} \times \beta_{21} + \dots + 2^{-23} \times \beta_0)$$

ここで、 β_i とはβの i ($0 \leq i \leq 22$) 番目の1ビットの値です。S は符号ビットです。

(c) 特異表現 (Denormalized Format)

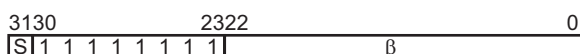


ここでβはフィールド長が23ビットの仮数です。この表現は正規表現では表現できない小さい実数を表現するときに用います。

この時の R の値は、次の式で表現できます。

$$R = 2^S \times 2^{-126} \times (2^{-1} \times \beta_{22} + 2^{-2} \times \beta_{21} + \dots + 2^{-23} \times \beta_0)$$

(d) 無限大



ここでβはフィールド長が23ビットの仮数です。ただし本アプリケーションノートでは、指数部がすべて1の場合、次のように扱います。

S = 0 の場合：正の無限大

$$R = +\infty$$

S = 1 の場合：負の無限大

$$R = -\infty$$

(2) 内部表現の例

$$S = B'0 \quad (2 \text{ 進})$$

$$\alpha = B'10000011 \quad (2 \text{ 進})$$

$$\beta = B'1011100 \cdots \cdots 0 \quad (2 \text{ 進})$$

としたとき，これに対応した実数は次のようになります。

$$\begin{aligned} R &= 20 \times 2^{13} - 126 \times (1 + 2^{-1} + 2^{-2} + 2^{-3} + 2^{-4} + 2^{-5}) \\ &= 16 + 8 + 2 + 1 + 0.5 = 27.5 \end{aligned}$$

(a) 最大値と最小値

ここでは絶対値としての最大，最小値について述べます。最大値を R_{MAX} ，最小値を R_{MIN} とします。以下の数値まで表わすことができます。

$$R_{\text{MAX}} = 2^{254-127} \times (1 + 2^{-1} + 2^{-2} + 2^{-3} + \cdots + 2^{-23})$$

$$3.27 \times 10^{38}$$

$$R_{\text{MIN}} = 2^{-126} \times 2^{-23} = 2^{-140} \quad 1.40 \times 10^{-45}$$

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
2.00	2006.02.28	-	日立版からルネサス版へフォーマット変更
3.00	2006.06.12	13	誤記修正

安全設計に関するお願い

1. 弊社は品質、信頼性の向上に努めておりますが、半導体製品は故障が発生したり、誤動作する場合があります。弊社の半導体製品の故障又は誤動作によって結果として、人身事故、火災事故、社会的損害などを生じさせないような安全性を考慮した冗長設計、延焼対策設計、誤動作防止設計などの安全設計に十分ご注意ください。

本資料ご利用に際しての留意事項

1. 本資料は、お客様が用途に応じた適切なルネサス テクノロジ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ルネサス テクノロジが所有する知的財産権その他の権利の実施、使用を許諾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の使用に起因する損害、第三者所有の権利に対する侵害に関し、ルネサス テクノロジは責任を負いません。
3. 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全ての情報は本資料発行時点のものであり、ルネサス テクノロジは、予告なしに、本資料に記載した製品または仕様を変更することがあります。ルネサス テクノロジ半導体製品のご購入に当たりましては、事前にルネサス テクノロジ、ルネサス販売または特約店へ最新の情報をご確認頂きますとともに、ルネサス テクノロジホームページ(<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。
4. 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。万一本資料の記述誤りに起因する損害がお客様に生じた場合には、ルネサス テクノロジはその責任を負いません。
5. 本資料に記載の製品データ、図、表に示す技術的な内容、プログラム及びアルゴリズムを流用する場合は、技術内容、プログラム、アルゴリズム単位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。ルネサス テクノロジは、適用可否に対する責任を負いません。
6. 本資料に記載された製品は、人命にかかわるような状況の下で使用される機器あるいはシステムに用いられることを目的として設計、製造されたものではありません。本資料に記載の製品を運輸、移動体用、医療用、航空宇宙用、原子力制御用、海底中継用機器あるいはシステムなど、特殊用途へのご利用をご検討の際には、ルネサス テクノロジ、ルネサス販売または特約店へご照会ください。
7. 本資料の転載、複製については、文書によるルネサス テクノロジの事前の承諾が必要です。
8. 本資料に関し詳細についてのお問い合わせ、その他お気づきの点がございましたらルネサス テクノロジ、ルネサス販売または特約店までご照会ください。