

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>

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。

H8/300L Super Low Power 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

H8/38024

目次

1. 引数	2
2. 内部レジスタ変化およびフラグ変化.....	2
3. プログラミング仕様	2
4. 注意事項	3
5. 説明	3
6. フローチャート	7
7. プログラムリスト.....	15
単精度浮動小数点について <参考>	19

1. 引数

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

2. 内部レジスタ変化およびフラグ変化

R0	R1	R2	R3	R4	R5	R6	R7
		×	×	×	×	×	—

I	U	H	U	N	Z	V	C
—	—	×	—	×	×	×	×

【記号説明】

- : 不変
- ×: 不定
- : 結果

3. 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
クロックサイクル数	268
リエントラント	可
リロケーション	可
途中割り込み	可

4. 注意事項

仕様のクロックサイクル数は、図 1 の実行例を行った時の値です。

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

5. 説明

5.1 機能詳細

1. 引数の詳細は以下のとおりです。

a. 入力引数として、単精度浮動小数点を以下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. 出力引数として単精度浮動小数点が以下のように設定されます。

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。

のように入力引数を設定すると、 のように加算結果が R0, R1 に設定されます。

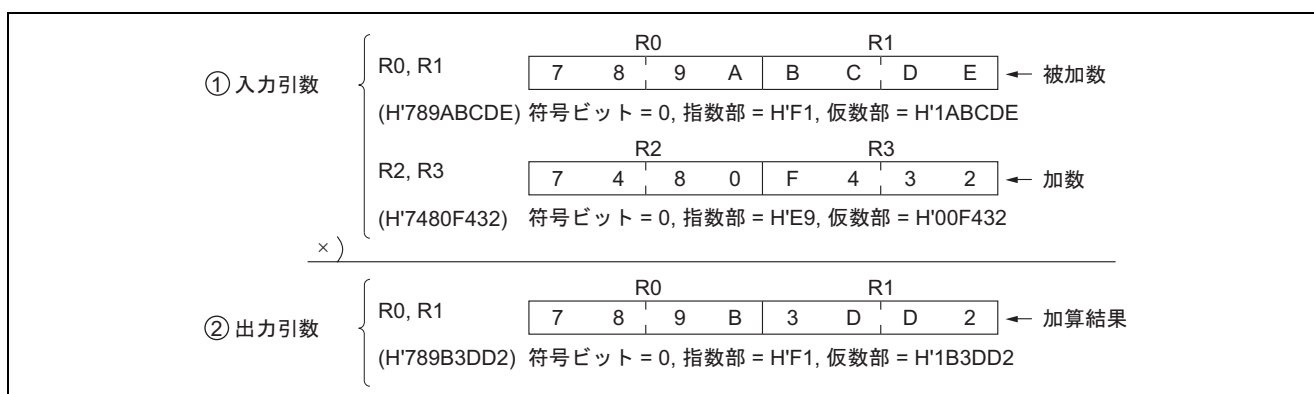


図 1 ソフトウェア FADD の実行例

5.2 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'7F800000
{ 正の最小値	H'00000001
{ 負の最大値	H'80000001
{ 負の最小値	H'FF800000

2. 単精度の正の数の H'7F800001 ~ H'7FFFFFFF はすべて最大値 (H'7F800000) として扱います。負の数の 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 ~ H'7FFFFFFF	*****	H'7F800000
H'7F800000 ~ H'FFFFFFF 以外	H'7F800000 ~ H'7FFFFFFF	H'7F800000
H'FF800000 ~ H'FFFFFFF	*****	H'FF800000
H'FF800000 ~ H'FFFFFFF 以外	H'FF800000 ~ H'FFFFFFF	H'FF800000

【注】 *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。

5.3 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。

5.4 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 { ユーザプログラムで 被加数 } 加数 } を設定するデータメモリアreaを 加算結果 } 確保します。
WORK2	. RES. W	2	
WORK3	. RES. W	2	
	⋮		
	MOV. W	@WORK1, R0	}... .. { ユーザプログラムで設定した被加数を 入力引数に設定します。
	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 入力引数に設定します。
	MOV. W	@WORK2+2, R3	
	JSR	@FADD { ソフトウェアFADDをサブルーチンコールします。
	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	
	⋮		

5.5 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 $+\infty$, $-\infty$ であるかを判定します。
 - a. 被加数の指数部が H'FF のとき

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

を出力値とします

- 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"であるかを判定します。
 - 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(特異表現の場合, 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 = $1.2088876915 \times 2^{14}$
 (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 1 1 0 1 0 1 0 1 1 1 1 0 0 0
	H'F1	H'9ABCDE

加数	1 1 1 1 0 1 0 0	1 . 0 0 1 1 0 1 1 0 0 1 1 1 1 0 1 1 0 1 0 0 1 0
	H'F4	H'9B3CC2

↓ 指数部を合わせる (被加数に3を加える) ↓ 被加数の仮数を右に3ビットシフト

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 0 0 1 1 0 1 1
-----	-----------------	---

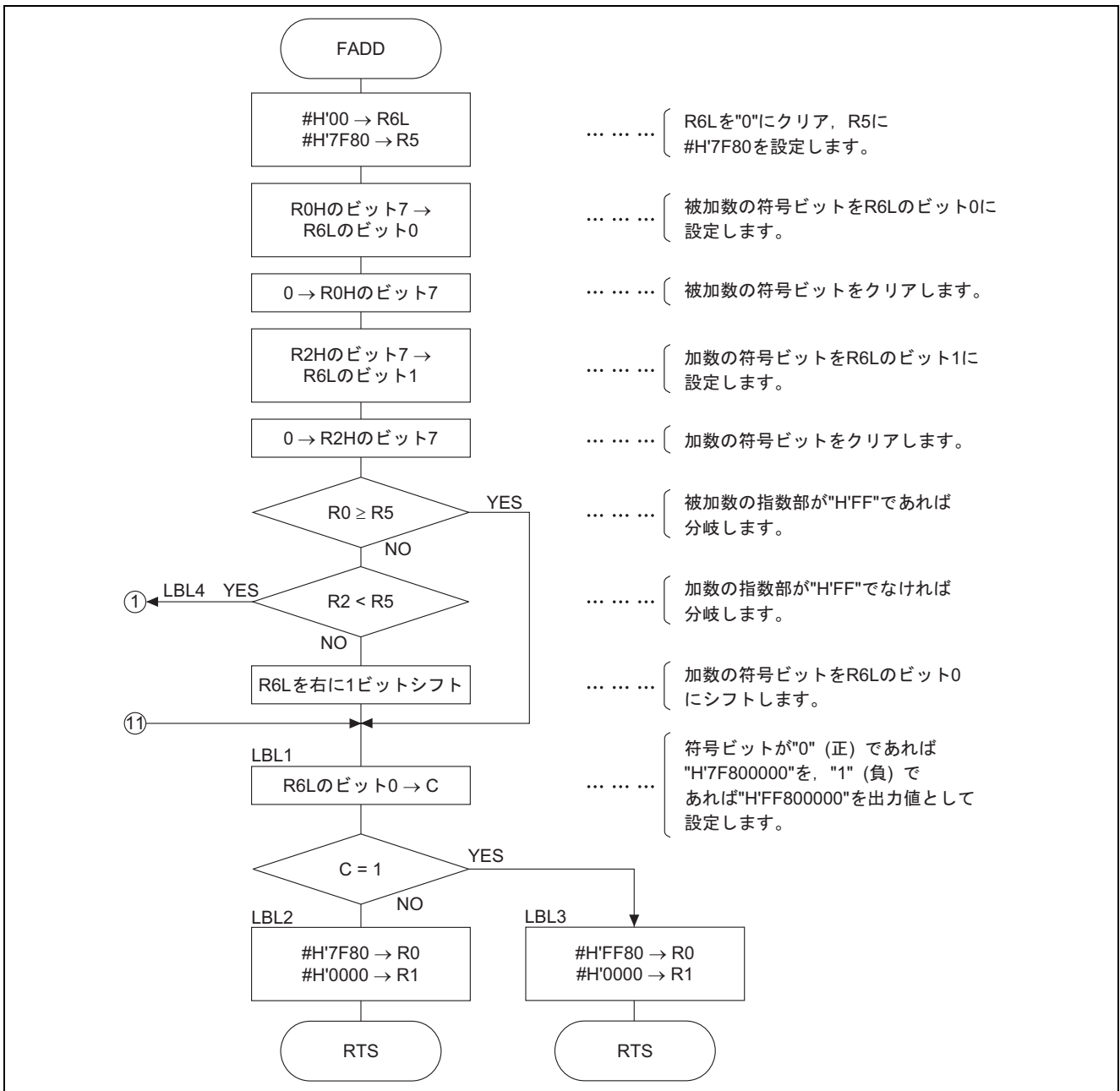
加数	1 1 1 1 0 1 0 0	1 . 0 1 1 0 0 1 1 0 0 1 1 1 1 0 1 1 1 0 1 0 0 1 0
(×)		

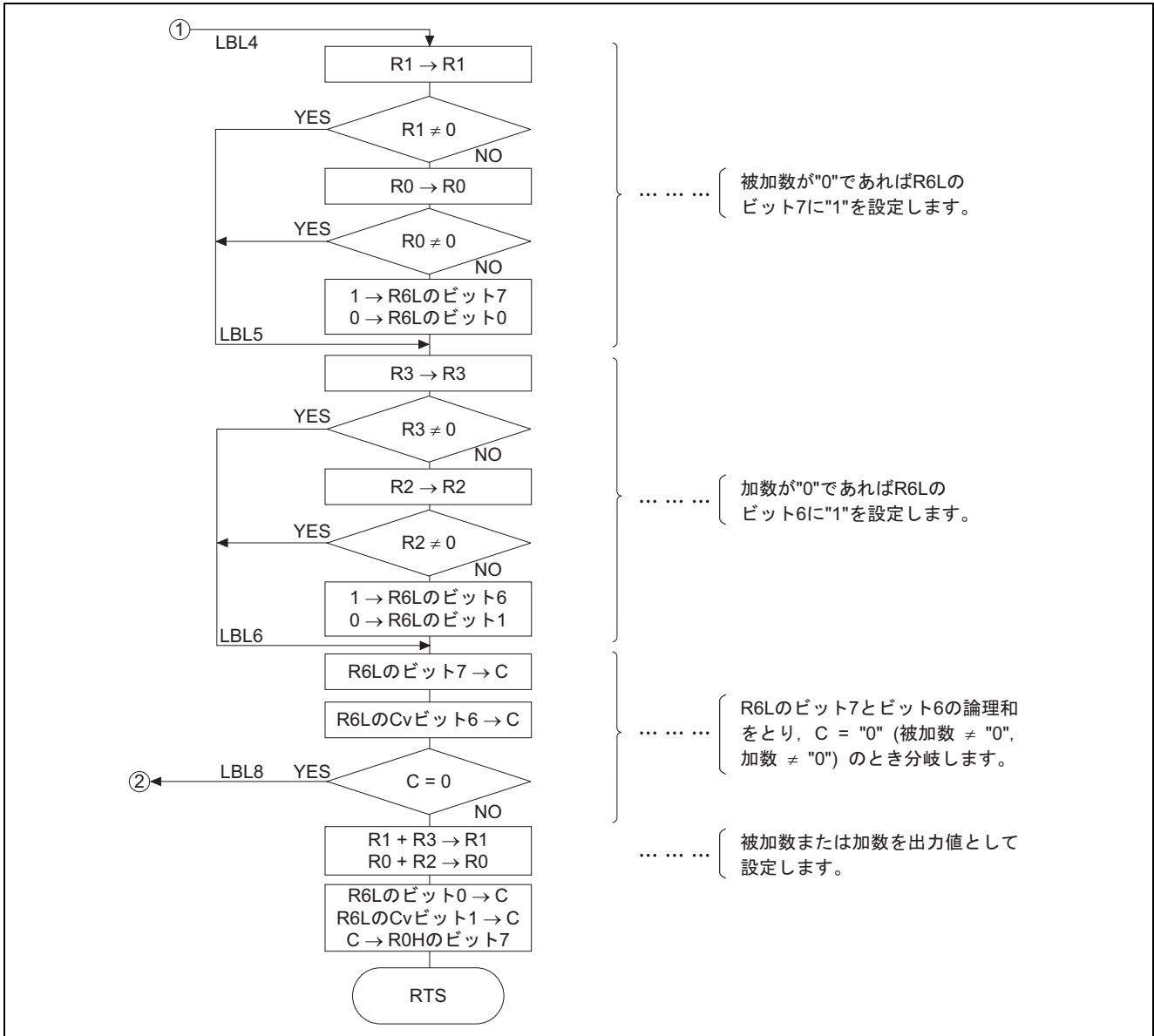
加算結果	1 1 1 1 0 1 0 0	1 . 0 0 1 1 1 1 0 1 0 0 1 0 1 0 1 0 1 1 1 0 1
------	-----------------	---

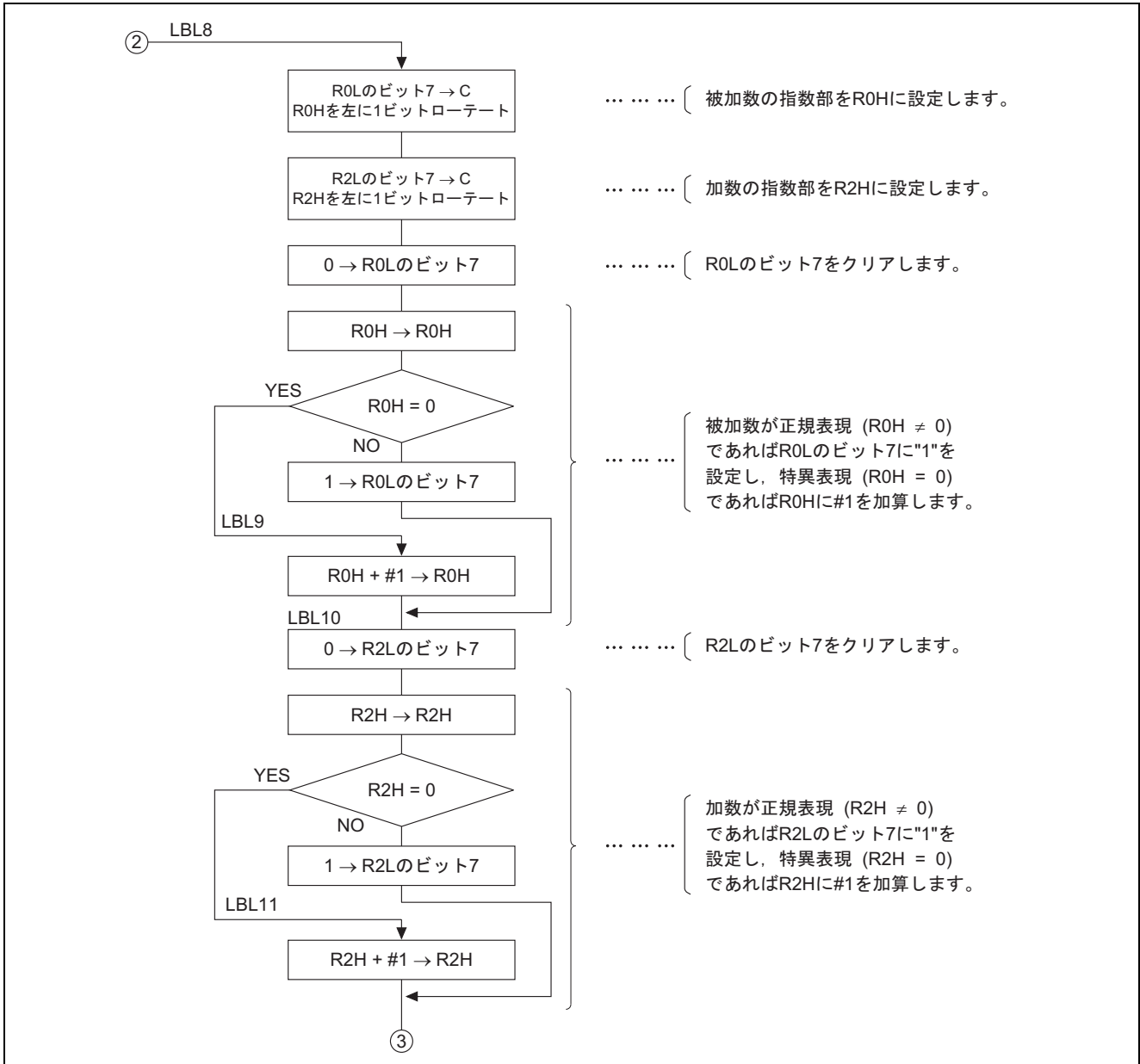
↓ 指数部はそのまま ↓ 仮数部のみ加算処理を行う

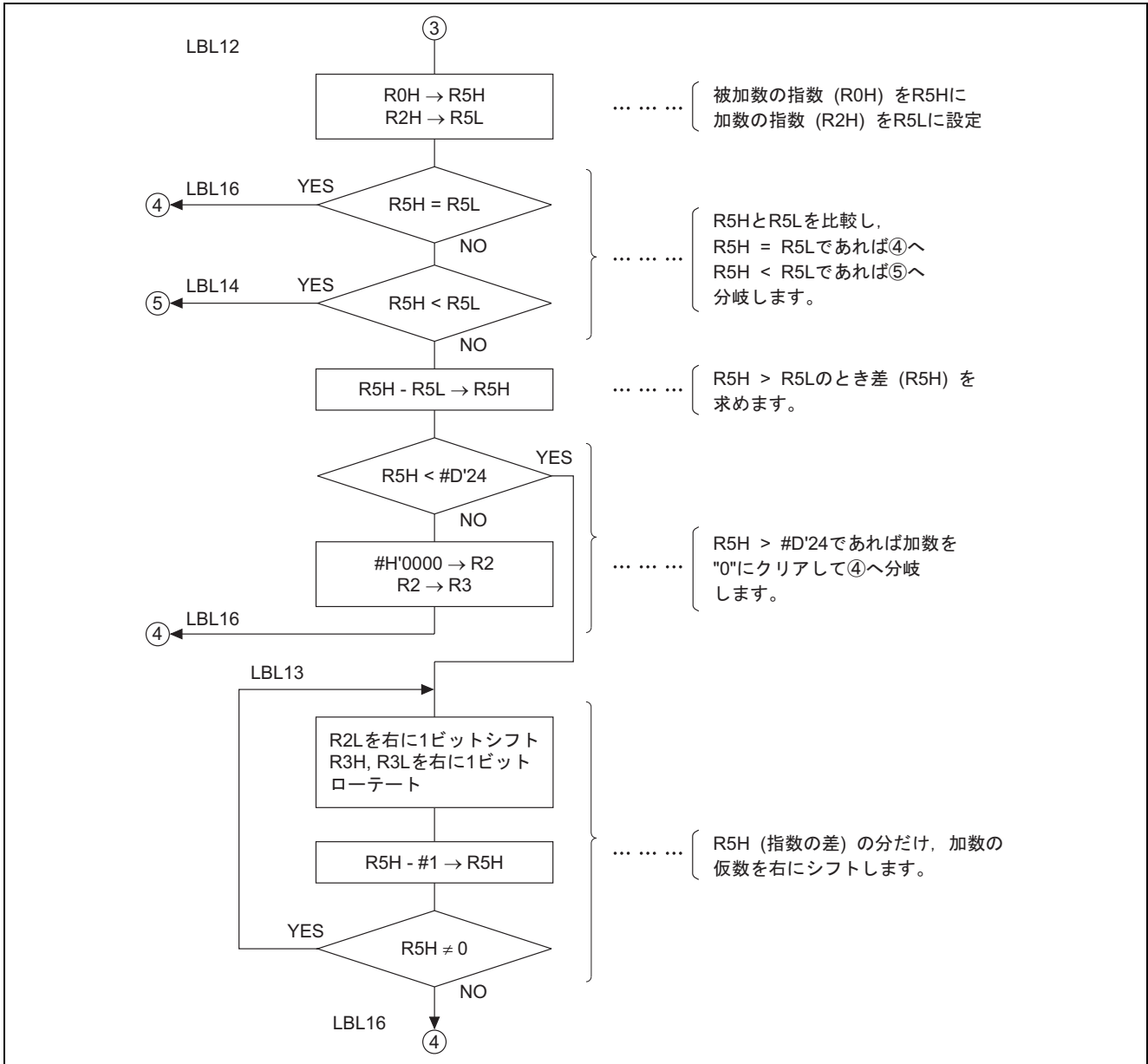
加算結果 = $1.36393511295 \times 2^{-17}$
 (H'7A2E956D)
 符号ビット = 0, 指数部 = H'F4, 仮数部 = H'2E956D
 (インプリシットMSBは含まない)

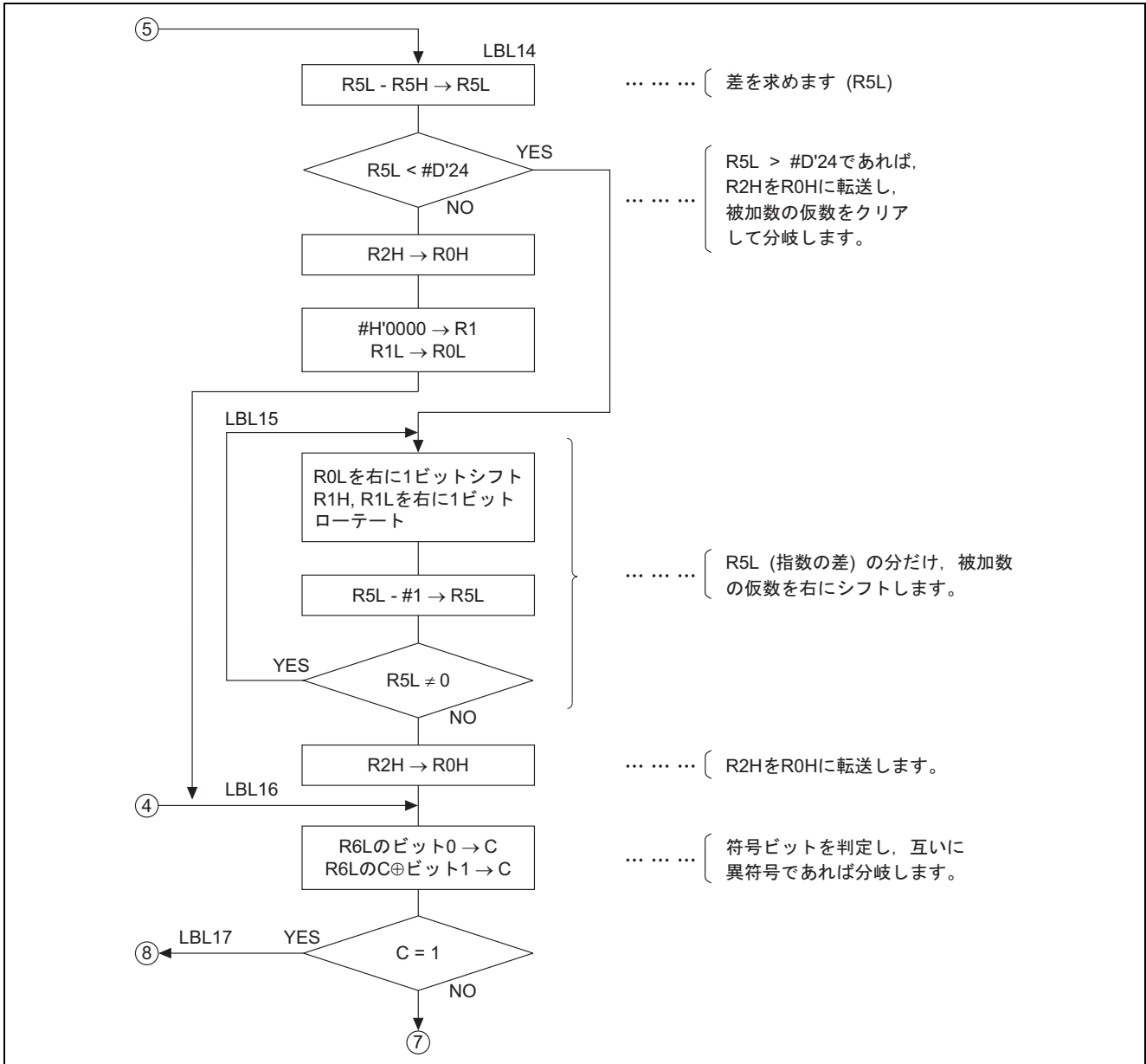
6. フローチャート

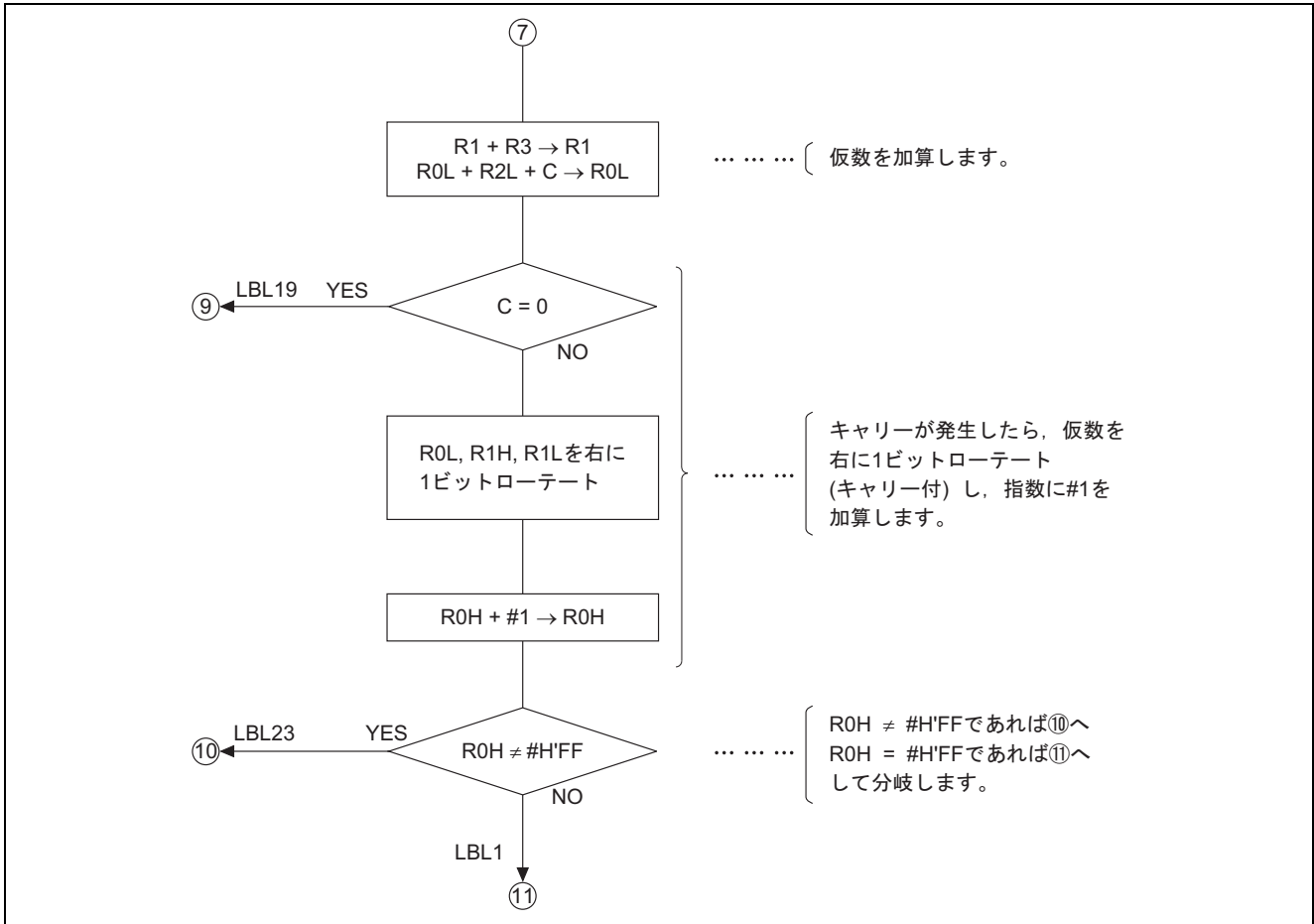


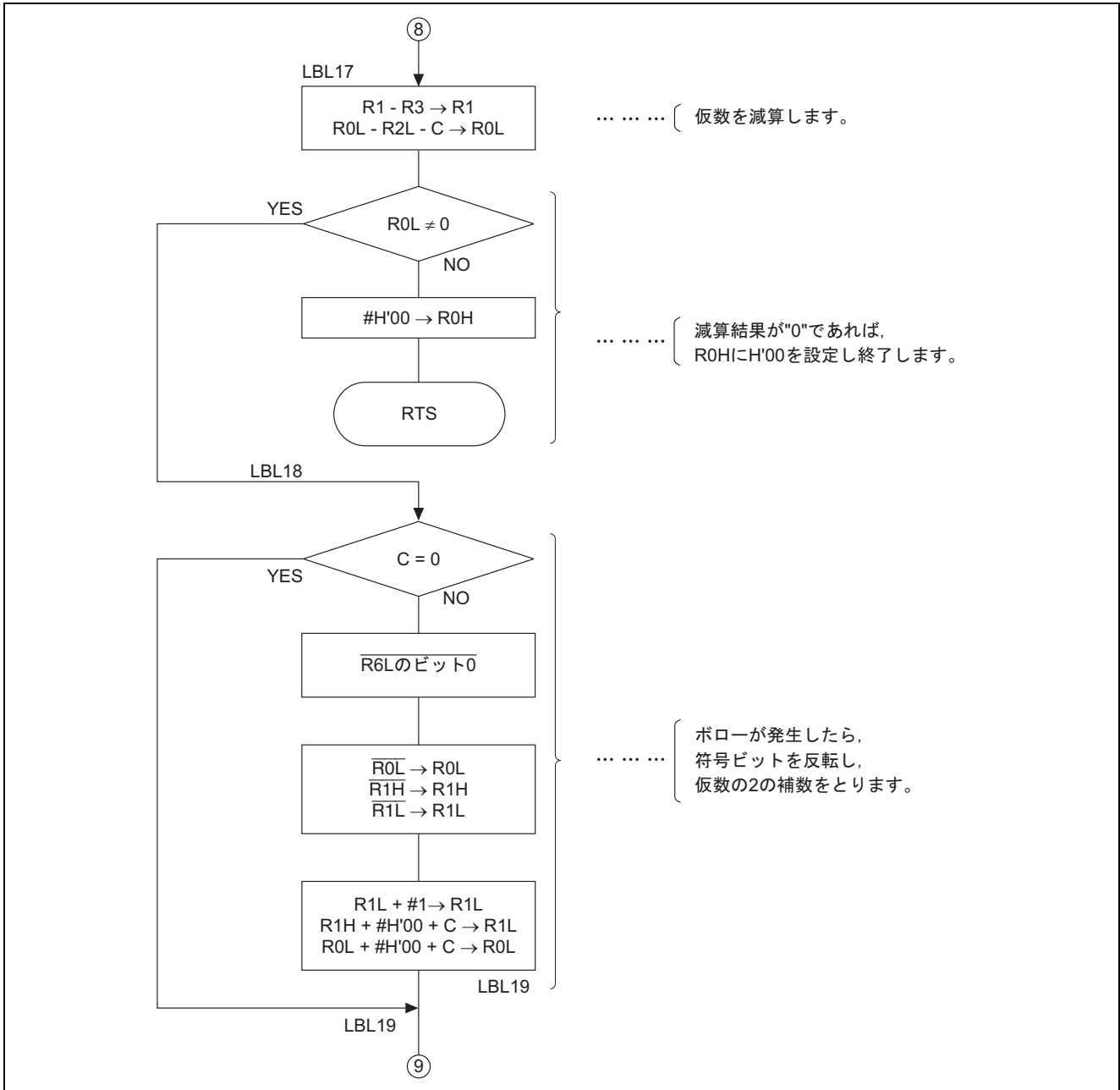


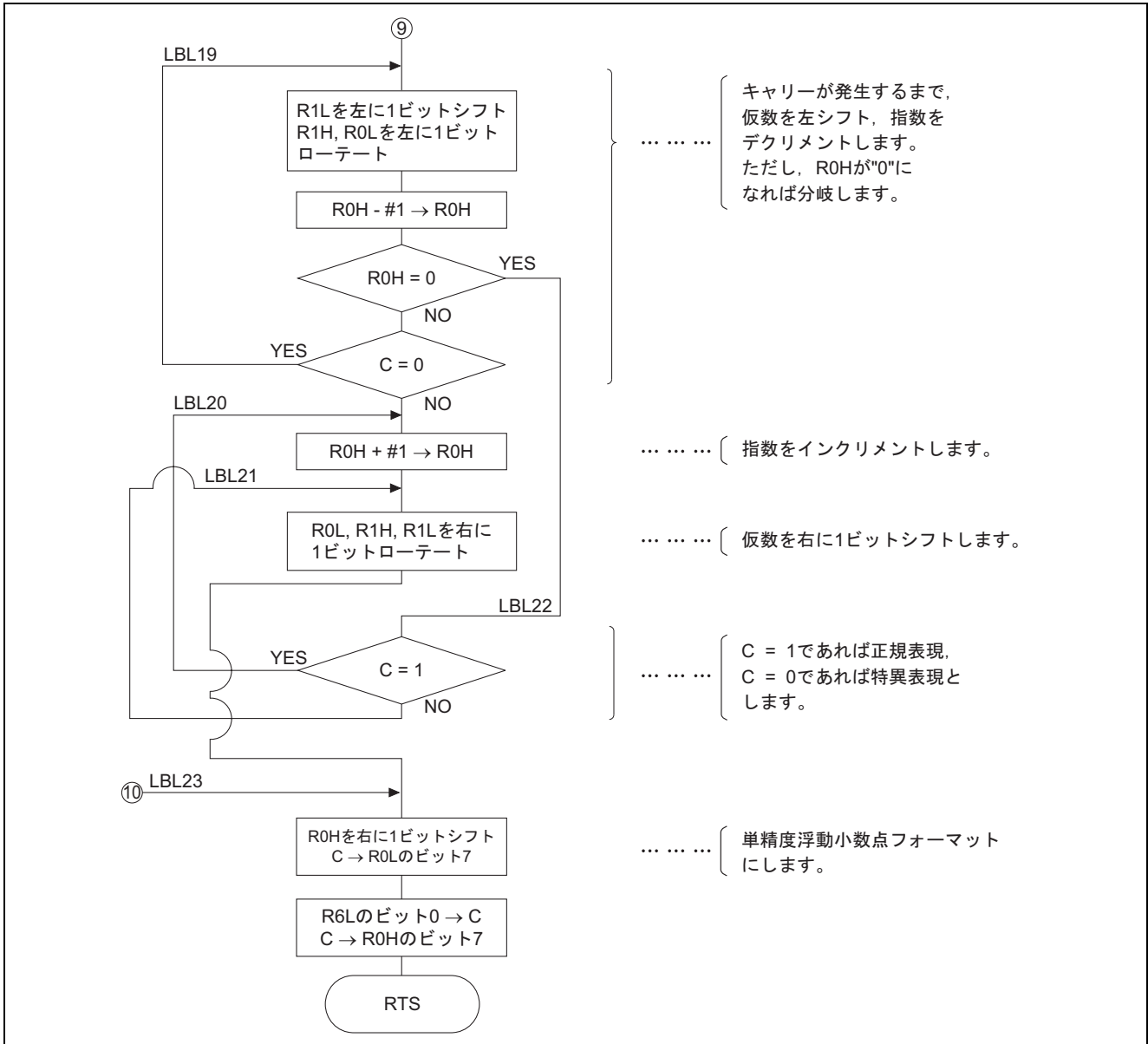












7. プログラムリスト

```

*** H8/300 ASSEMBLER  VER 1.0B **           08/18/92 10:20:43
PROGRAM NAME =
1          ;*****
2          ;*
3          ;*      00 - NAME           :FLOATING POINT ADDITION (FADD)
4          ;*
5          ;*****
6          ;*
7          ;*      ENTRY:   R0           (UPPER WORD OF SUMMAND)
8          ;*                R1           (LOWER WORD OF SUMMAND)
9          ;*                R2           (UPPER WORD OF ADDEND)
10         ;*                R3           (LOWER WORD OF ADDEND)
11         ;*
12         ;*      RETURNS: R0           (UPPER WORD OF RESULT)
13         ;*                R1           (LOWER WORD OF RESULT)
14         ;*
15         ;*****
16         ;
17 FADD_cod C  0000          .SECTION FADD_code, CODE, ALIGN=2
18         .EXPORT  FADD
19         ;
20 FADD_cod C  00000000 FADD .EQU  $           ;Entry point
21 FADD_cod C  0000 FE00     MOV.B   #H'00,R6L  ;Clear R6L
22 FADD_cod C  0002 79057F80 MOV.W   #H'7F80,R5 ;Set "H'7F80"
23         ;
24 FADD_cod C  0006 7770     BLD    #7,R0H
25 FADD_cod C  0008 670E     BST    #0,R6L  ;Set sign bit to bit 0 of R6L
26 FADD_cod C  000A 7270     BCLR  #7,R0H  ;Bit clear bit 7 of R0H
27         ;
28 FADD_cod C  000C 7772     BLD    #7,R2H
29 FADD_cod C  000E 671E     BST    #1,R6L  ;Set sign bit to bit 1 of R6L
30 FADD_cod C  0010 7272     BCLR  #7,R2H  ;Bit clear bit 7 of R2H
31         ;
32 FADD_cod C  0012 1D05     CMP.W  R0,R5
33 FADD_cod C  0014 4306     BLS    LBL1    ;Branch if "exponent of summand"="H'FF"
34 FADD_cod C  0016 1D25     CMP.W  R2,R5
35 FADD_cod C  0018 421A     BHI    LBL4    ;Branch if not "exponent of summand"="H'FF"
36 FADD_cod C  001A 110E     SHLR  R6L     ;Shift R6L 1 bit right
37 FADD_cod C  001C          LBL1
38 FADD_cod C  001C 770E     BLD    #0,R6L  ;Bit load sign bit
39 FADD_cod C  001E 450A     BCS    LBL3    ;Branch if sign bit=1
40 FADD_cod C  0020          LBL2
41 FADD_cod C  0020 79007F80 MOV.W   #H'7F80,R0 ;Set plus maximum number
42 FADD_cod C  0024 79010000 MOV.W   #H'0000,R1
43 FADD_cod C  0028 5470     RTS
44 FADD_cod C  002A          LBL3
45 FADD_cod C  002A 7900FF80 MOV.W   #H'FF80,R0 ;Set minus minimum number
46 FADD_cod C  002E 79010000 MOV.W   #H'0000,R1
47 FADD_cod C  0032 5470     RTS
48         ;
49 FADD_cod C  0034          LBL4
50 FADD_cod C  0034 0D11     MOV.W  R1,R1   ;
51 FADD_cod C  0036 4608     BNE    LBL5    ;Branch if Z=0
52 FADD_cod C  0038 0D00     MOV.W  R0,R0
53 FADD_cod C  003A 4604     BNE    LBL5    ;Branch if Z=0

```

```

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

```

110 FADD_cod C 0096          LBL13
111 FADD_cod C 0096 110A          SHLR    R2L      ;Shift mantissa of addend 1 bit left
112 FADD_cod C 0098 1303          ROTXR   R3H
113 FADD_cod C 009A 130B          ROTXR   R3L
114 FADD_cod C 009C 1A05          DEC.B   R5H      ;Decrement bit counter
115 FADD_cod C 009E 46F6          BNE     LBL13    ;Branch Z=0
116 FADD_cod C 00A0 401C          BRA     LBL16    ;Branch always
117                               ;
118 FADD_cod C 00A2          LBL14
119 FADD_cod C 00A2 185D          SUB.B   R5H,R5L
120 FADD_cod C 00A4 AD18          CMP.B   #D'24,R5L
121 FADD_cod C 00A6 450A          BCS     LBL15    ;Branch if R5L<D'24
122 FADD_cod C 00A8 0C20          MOV.B   R2H,R0H
123 FADD_cod C 00AA 79010000      MOV.W   #H'0000,R1 ;Clear summand
124 FADD_cod C 00AE 0C98          MOV.B   R1L,R0L
125 FADD_cod C 00B0 400C          BRA     LBL16    ;Branch always
126 FADD_cod C 00B2          LBL15
127 FADD_cod C 00B2 1108          SHLR    R0L      ;Shift mantissa of summand 1 bit right
128 FADD_cod C 00B4 1301          ROTXR   R1H
129 FADD_cod C 00B6 1309          ROTXR   R1L
130 FADD_cod C 00B8 1A0D          DEC.B   R5L      ;Decrement bit counter
131 FADD_cod C 00BA 46F6          BNE     LBL15    ;Branch if Z=0
132 FADD_cod C 00BC 0C20          MOV.B   R2H,R0H
133                               ;
134 FADD_cod C 00BE          LBL16
135 FADD_cod C 00BE 770E          BLD     #0,R6L
136 FADD_cod C 00C0 751E          BXOR    #1,R6L
137 FADD_cod C 00C2 4516          BCS     LBL17    ;Branch if different sign bit
138                               ;
139 FADD_cod C 00C4 0931          ADD.W   R3,R1    ;Addition mantissa
140 FADD_cod C 00C6 0EA8          ADDX.B  R2L,R0L
141 FADD_cod C 00C8 442A          BCC     LBL19    ;Branch if C=0
142 FADD_cod C 00CA 1308          ROTXR   R0L      ;Rotate mantissa 1 bit right
143 FADD_cod C 00CC 1301          ROTXR   R1H
144 FADD_cod C 00CE 1309          ROTXR   R1L
145 FADD_cod C 00D0 8001          ADD.B   #H'01,R0H ;Increment exponent
146 FADD_cod C 00D2 A0FF          CMP.B   #H'FF,R0H
147 FADD_cod C 00D4 4638          BNE     LBL23    ;Branch if not exponent=H'FF
148 FADD_cod C 00D6 5A000000      JMP     @LBL1    ;Jump
149                               ;
150 FADD_cod C 00DA          LBL17
151 FADD_cod C 00DA 1931          SUB.W   R3,R1    ;Substruct mantissa
152 FADD_cod C 00DC 1EA8          SUBX.B  R2L,R0L
153 FADD_cod C 00DE 4604          BNE     LBL18    ;Branch if Z=0
154 FADD_cod C 00E0 F000          MOV.B   #H'00,R0H ;Clear R0H
155 FADD_cod C 00E2 5470          RTS
156 FADD_cod C 00E4          LBL18
157 FADD_cod C 00E4 440E          BCC     LBL19    ;Branch if C=0
158 FADD_cod C 00E6 710E          BNOT    #0,R6L   ;Bit not sign bit
159 FADD_cod C 00E8 1708          NOT     R0L      ;2's complement mantissa
160 FADD_cod C 00EA 1701          NOT     R1H
161 FADD_cod C 00EC 1709          NOT     R1L
162 FADD_cod C 00EE 8901          ADD.B   #H'01,R1L
163 FADD_cod C 00F0 9100          ADDX.B  #H'00,R1H
164 FADD_cod C 00F2 9800          ADDX.B  #H'00,R0L
165                               ;
    
```

```

166 FADD_cod C 00F4          LBL19
167 FADD_cod C 00F4 1009      SHLL    R1L      ;Shift mantissa 1 bit left
168 FADD_cod C 00F6 1201      ROTXL   R1H
169 FADD_cod C 00F8 1208      ROTXL   R0L
170 FADD_cod C 00FA 1A00      DEC.B   R0H      ;Decrement exponent
171 FADD_cod C 00FC 470C      BEQ     LBL22     ;Branch if exponent=0
172 FADD_cod C 00FE 44F4      BCC     LBL19     ;Branch if exponent>0
173 FADD_cod C 0100          LBL20
174 FADD_cod C 0100 0A00      INC.B   R0H      ;Increment exponent
175 FADD_cod C 0102          LBL21
176 FADD_cod C 0102 1308      ROTXR   R0L      ;Rotate mantissa 1 bit right
177 FADD_cod C 0104 1301      ROTXR   R1H
178 FADD_cod C 0106 1309      ROTXR   R1L
179 FADD_cod C 0108 4004      BRA     LBL23     ;Branch always
180 FADD_cod C 010A          LBL22
181 FADD_cod C 010A 45F4      BCS     LBL20     ;Branch if C=1
182 FADD_cod C 010C 40F4      BRA     LBL21     ;Branch always
183                               ;
184 FADD_cod C 010E LBL23      ;Chage floating point format
185 FADD_cod C 010E 1100      SHLR    R0H
186 FADD_cod C 0110 6778      BST     #7,R0L
187 FADD_cod C 0112 770E      BLD     #0,R6L
188 FADD_cod C 0114 6770      BST     #7,R0H
189 FADD_cod C 0116 5470      RTS
190                               ;
191                               .END
*****TOTAL ERRORS      0
*****TOTAL WARNINGS    0

```

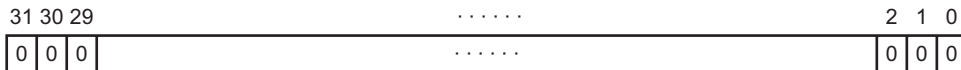
単精度浮動小数点について < 参考 >

単精度浮動小数点のフォーマットについて

1. 単精度浮動小数点数の内部表現について

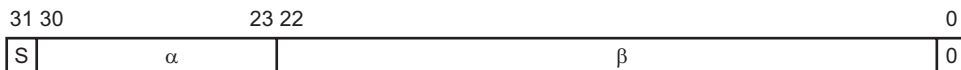
本アプリケーションノートでは、単精度浮動小数点の値 ($R = \text{実数}$) により、以下の表現方法を用います。

A. $R = 0$ のときの内部表現



32 ビットのすべてのビットが 0 です。

B. 正規表現 (Normalized Format)



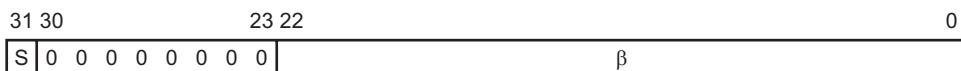
ここで α はフィールド長が 8 ビットの指数です。 β はフィールド長が 23 ビットの仮数です。

このときの R の値は、次の式で表現できます ($1 \leq \alpha \leq 254$ の場合)。

$$R = 2^S \times 2^{\alpha-127} \times (1 + 2^{-1} \times \beta_{22} + 2^{-2} \times \beta_{21} + \dots + 2^{-23} \times \beta_0)$$

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。

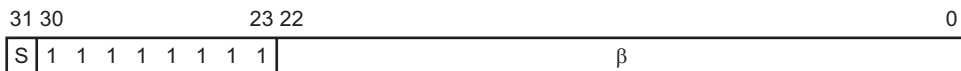
C. 特異表現 (Denormalized Format)



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。このときの 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

$$\begin{aligned} \text{If } S &= B'0 \text{ (2進)} \\ \alpha &= B'10000011 \text{ (2進)} \\ \beta &= B'1011100\dots\dots 0 \text{ (2進)} \end{aligned}$$

としたとき、これに対応した実数は次のようになります。

$$\begin{aligned} R &= 2^0 \times 2^{131-127} \times (1 + 2^{-1} + 2^{-3} + 2^{-4} + 2^{-5}) \\ &= 16 + 8 + 2 + 1 + 0.5 = 27.5 \end{aligned}$$

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。

以下の数値まで表すことができます。

$$\begin{aligned} R_{MAX} &= 2^{254-127} \times (1 + 2^{-1} + 2^{-2} + 2^{-3} \dots\dots + 2^{-23}) \\ &= 3.37 \times 10^{38} \\ R_{MIN} &= 2^{-126} \times 2^{-23} = 2^{-140} = 1.40 \times 10^{-45} \end{aligned}$$

ホームページとサポート窓口

ルネサステクノロジホームページ

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

お問合せ先

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

csc@renesas.com

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2003.09.19	—	初版発行
2.00	2006.10.13	19, 20	< 参考 > を追加

本資料ご利用に際しての留意事項

1. 本資料は、お客様に用途に応じた適切な弊社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弊社または第三者の知的財産権その他の権利の実施、使用を許諾または保証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など全ての情報の使用に起因する損害、第三者の知的財産権その他の権利に対する侵害に関し、弊社は責任を負いません。
3. 本資料に記載の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、あるいはその他軍事事業の目的で使用しないでください。また、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、それらの定めるところにより必要な手続を行ってください。
4. 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などの全ての情報は本資料発行時点のものであり、弊社は本資料に記載した製品または仕様等を予告なしに変更することがあります。弊社の半導体製品のご購入およびご使用に当たりますは、事前に弊社営業窓口で最新の情報をご確認頂きますとともに、弊社ホームページ (<http://www.renesas.com>) などを通じて公開される情報に常にご注意下さい。
5. 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のですが、万一本資料の記述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、弊社はその責任を負いません。
6. 本資料に記載の製品データ、図、表などに示す技術的な内容、プログラム、アルゴリズムその他応用回路例などの情報を流用する場合は、流用する情報を単独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下さい。弊社は、適用可否に対する責任を負いません。
7. 本資料に記載された製品は、各種安全装置や運輸・交通用、医療用、燃焼制御用、航空宇宙用、原子力、海底中継用の機器・システムなど、その故障や誤動作が直接人命を脅かしあるいは人体に危害を及ぼすおそれのあるような機器・システムや特に高度な品質・信頼性が要求される機器・システムでの使用を意図して設計、製造されたものではありません（弊社が自動車用と指定する製品を自動車に使用する場合を除きます）。これらの用途に利用されることをご検討の際には、必ず事前に弊社営業窓口へご照会下さい。なお、上記用途に使用されたことにより発生した損害等について弊社はその責任を負いかねますのでご了承願います。
8. 第7項にかかわらず、本資料に記載された製品は、下記の用途には使用しないで下さい。これらの用途に使用されたことにより発生した損害等につきましては、弊社は一切の責任を負いません。
 - 1) 生命維持装置。
 - 2) 人体に埋め込み使用するもの。
 - 3) 治療行為（患部切り出し、薬剤投与等）を行なうもの。
 - 4) その他、直接人命に影響を与えるもの。
9. 本資料に記載された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、実装条件およびその他諸条件につきましては、弊社保証範囲内でご使用ください。弊社保証値を越えて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、弊社はその責任を負いません。
10. 弊社は製品の品質及および信頼性の向上に努めておりますが、特に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。弊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、誤動作防止設計などの安全設計（含むハードウェアおよびソフトウェア）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致します。
11. 本資料に記載の製品は、これを搭載した製品から剥がれた場合、幼児が口に入れて誤飲する等の事故の危険性があります。お客様の製品への実装後に容易に本製品が剥がれることがなきよう、お客様の責任において十分な安全設計をお願いします。お客様の製品から剥がれた場合の事故につきましては、弊社はその責任を負いません。
12. 本資料の全部または一部を弊社の文書による事前の承諾なしに転載または複製することを固くお断り致します。
13. 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弊社営業窓口までご照会下さい。