

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、誤りが無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乗算

要旨

汎用レジスタに設定された単精度浮動小数点の乗算を行い、乗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	6
7. プログラムリスト.....	16
単精度浮動小数点について <参考>	21

1. 引数

内容		格納場所	データ長 (バイト)
入力	被乗数	R0, R1	4
	乗数	R2, R3	4
出力	乗算結果	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)	348
データメモリ (バイト)	0
スタック (バイト)	16
クロックサイクル数	1078
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：被乗数の上位 2 バイト

R1：被乗数の下位 2 バイト

R2：乗数の上位 2 バイト

R3：乗数の下位 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：乗算結果の上位 2 バイト

R1：乗算結果の下位 2 バイト

2. 図 1 にソフトウェア FMUL の実行例を示します。

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

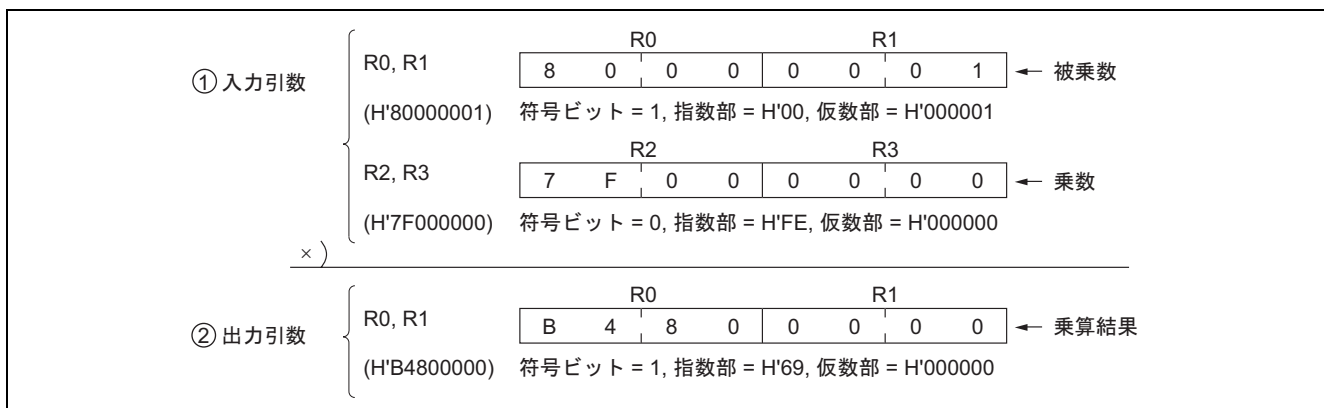


図 1 ソフトウェア FMUL の実行例

5.2 使用上の注意

1. ソフトウェア FMUL 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
	H'00000001
正の最小値	H'80000001
	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 $\infty \times 100 = \infty$, $\infty \times (-100) = -\infty$ となります (表 1 参照)。

表 1 最大値を引数とした演算例

被乗数	乗数	結果
> H'7F800000 ($+\infty$)	正の数	H'7F800000 ($+\infty$)
	負の数	H'FF800000 ($-\infty$)
< H'FF800000 ($-\infty$)	正の数	H'FF800000 ($-\infty$)
	負の数	H'7F800000 ($+\infty$)
正の数	> H'7F800000 ($+\infty$)	H'7F800000 ($+\infty$)
	< H'FF800000 ($-\infty$)	H'FF800000 ($-\infty$)
負の数	> H'7F800000 ($+\infty$)	H'FF800000 ($-\infty$)
	< H'FF800000 ($-\infty$)	H'7F800000 ($+\infty$)

4. H'80000000 は H'00000000 (ゼロ) として扱います。
5. ソフトウェア FMUL 実行後，汎用レジスタの被乗数，乗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 FMUL では，データメモリを使用していません。

5.4 使用例

被乗数および乗数を設定し，ソフトウェア FMUL をサブルーチンコールします。

WORK1	. RES. B	2	...	ユーザプログラムで 被加数 加数 加算結果 } を設定するデータメモリエリアを 確保します。
WORK2	. RES. B	2		
WORK3	. RES. B	2		
	⋮			
	MOV. W	@WORK1,R0	} ... {	ユーザプログラムで設定した被乗数を 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乗数を 入力引数に設定します。
	MOV. W	@WORK2+2,R3		
	JSR	@FMUL	...	ソフトウェア FMUL 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乗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乗算は次のように行います。

1. 被乗数, 乗数が"0"であるかを判定します。
 - a. 被乗数, 乗数のどちらかが"0"であれば, H'00000000 を出力値とします。
2. 被乗数, 乗数が無限大であるかを判定します。
無限大であれば, 表 1 に示す結果を出力値とします。
3. 被乗数を R_1 (符号ビット: S_1 , 指数部 α_1 , 仮数部 β_1), R_2 (符号ビット: S_2 , 指数部 α_2 , 仮数部 β_2) とすると,

$$R_1 = (-1)^{S_1} \times 2^{\alpha_1 - 127} \times \beta_1$$

$$R_2 = (-1)^{S_2} \times 2^{\alpha_2 - 127} \times \beta_2$$

となり, 乗算は,

$$R_1 \times R_2 = (-1)^{S_1 + S_2} \times 2^{\alpha_1 + \alpha_2 - 127 - 127} \times \beta_1 \times \beta_2$$

となります。浮動小数点フォーマットでは, 指数部の乗算結果に H'7F (D'127) を加えるため,

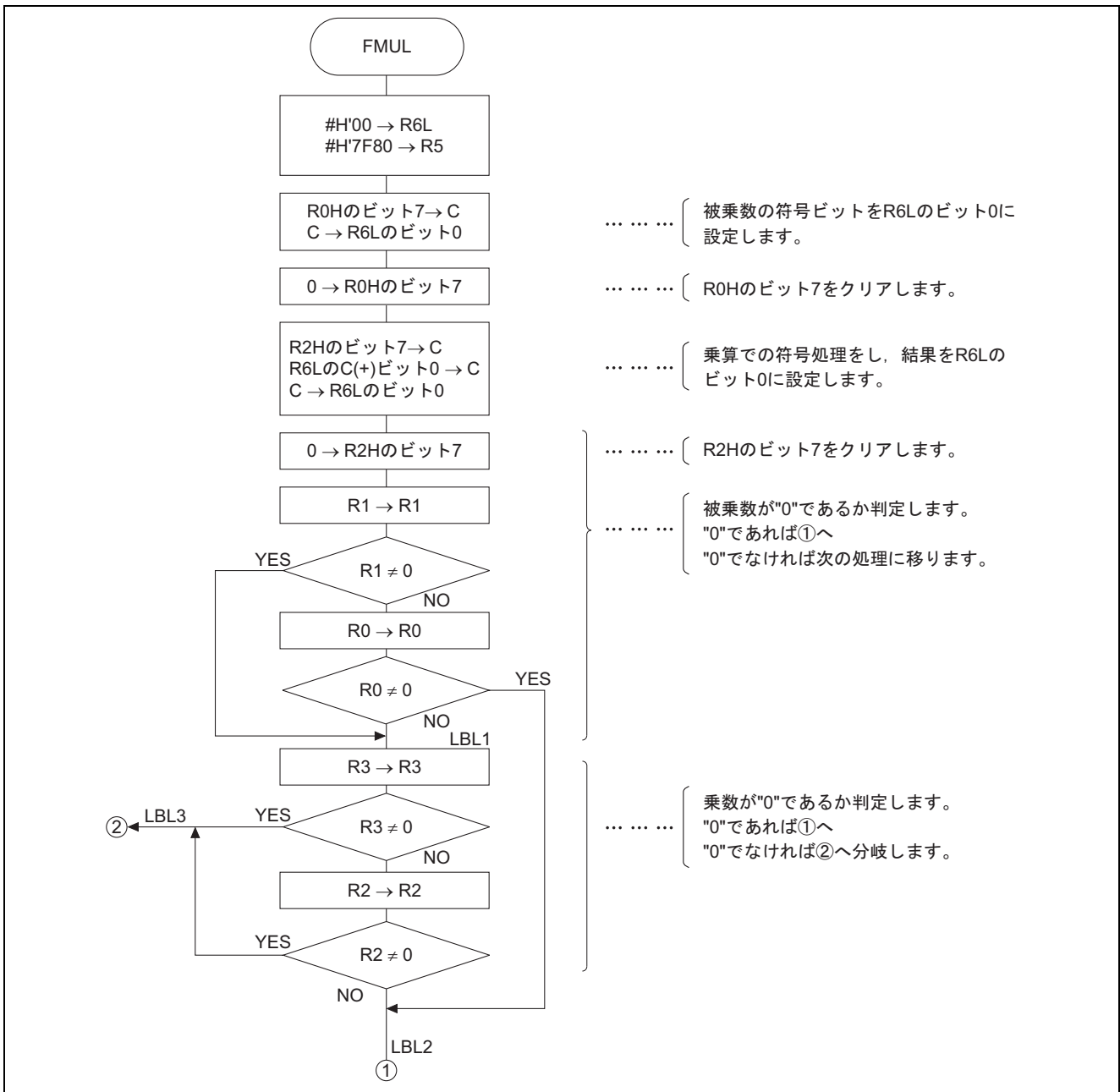
$$R_1 \times R_2 = (-1)^{S_1 + S_2} \times 2^{\alpha_1 + \alpha_2 - 127} \times \beta_1 \times \beta_2$$

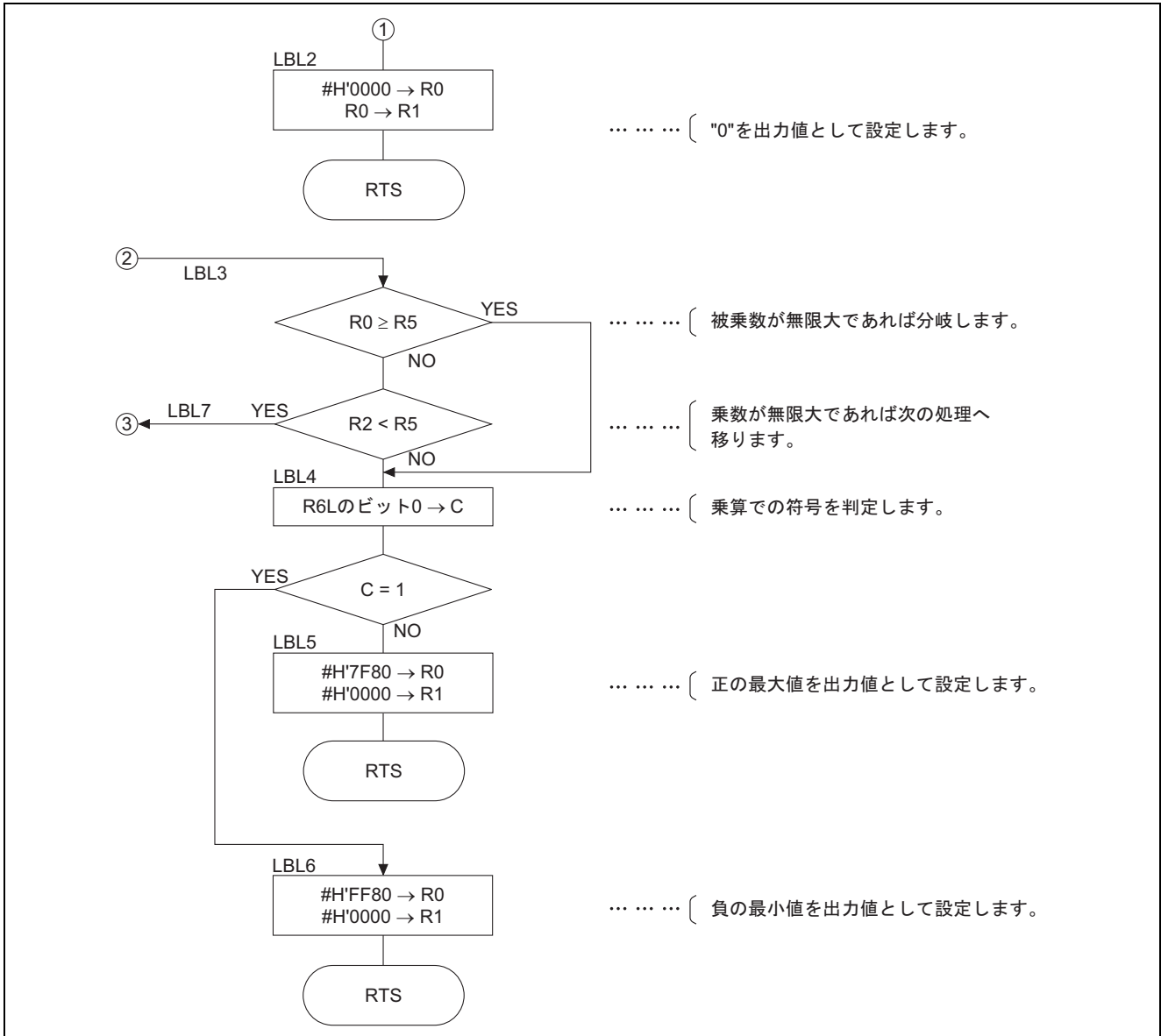
となります。よって処理は次のように行います。

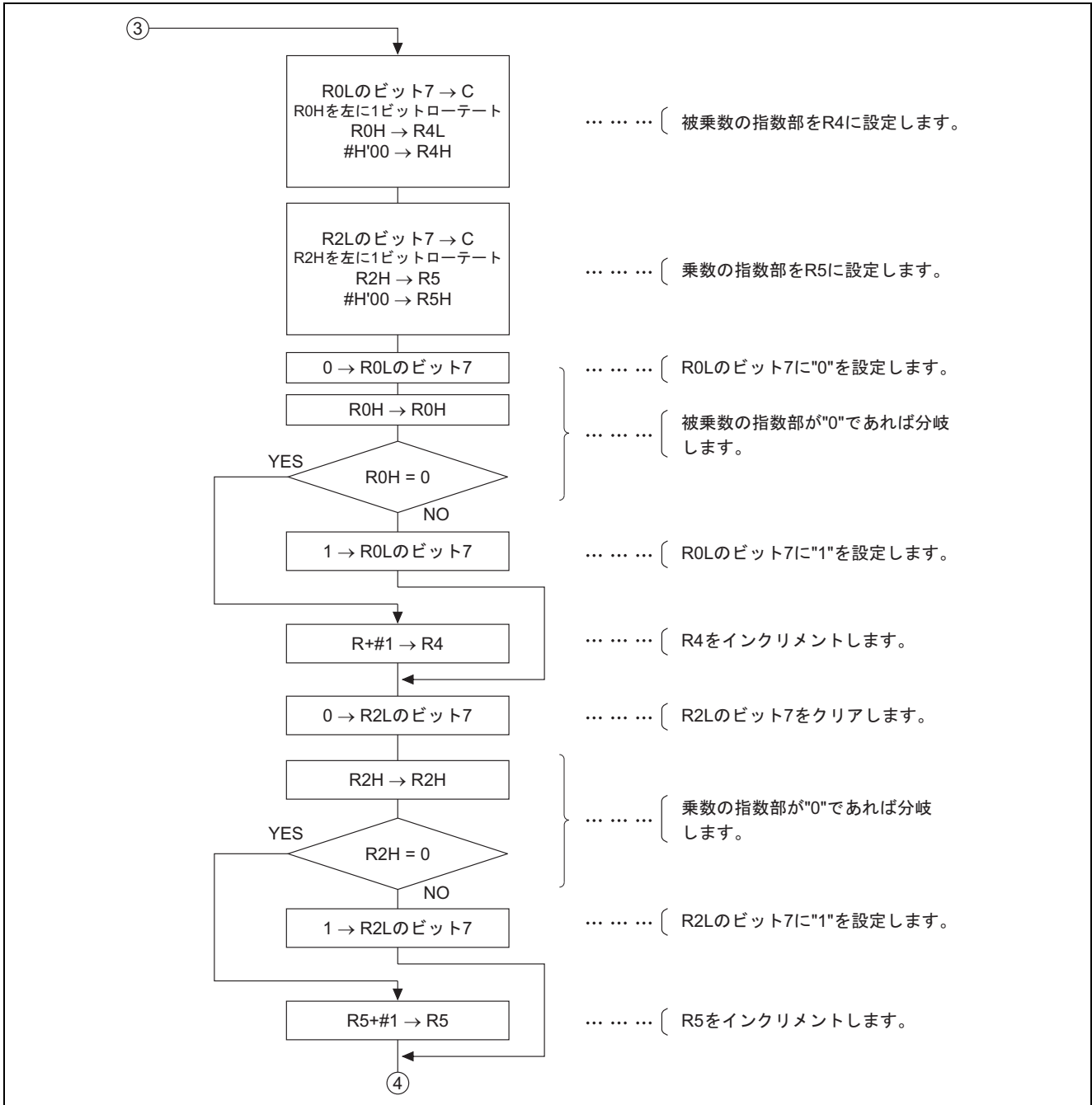
- a. $R_1 \times R_2$ の符号ビットを判定します。
- b. 指数部を加算処理します。
 α_1, α_2 はともに浮動小数点フォーマットに添って H'7F (D'127) 分の加算を行っています。乗算結果にも H'7F (D'127) を加えるため,

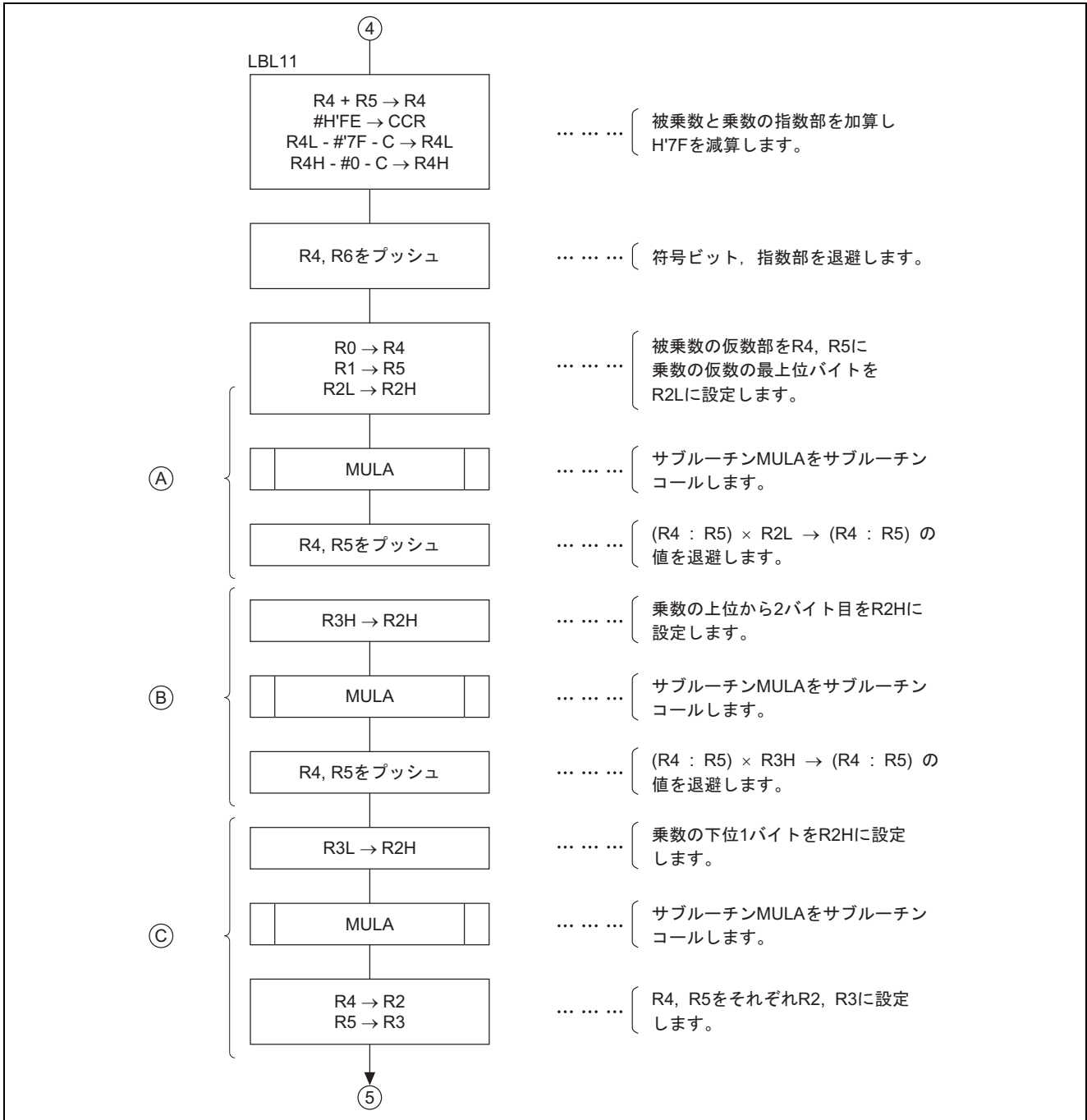
$$(\alpha_1 - H'7F) + (\alpha_2 - H'7F) + H'7F = \alpha_1 + \alpha_2 - H'7F$$
 とします。
(特異表現の場合, 指数部にあらかじめ 1 を加算して処理します)。
- c. 仮数部を乗算処理します。
インプリシット MSB も含めて処理します。
(特異表現の場合, 仮数部のインプリシット MSB は"0"として処理します)。
- d. 演算結果を浮動小数点フォーマットにします。

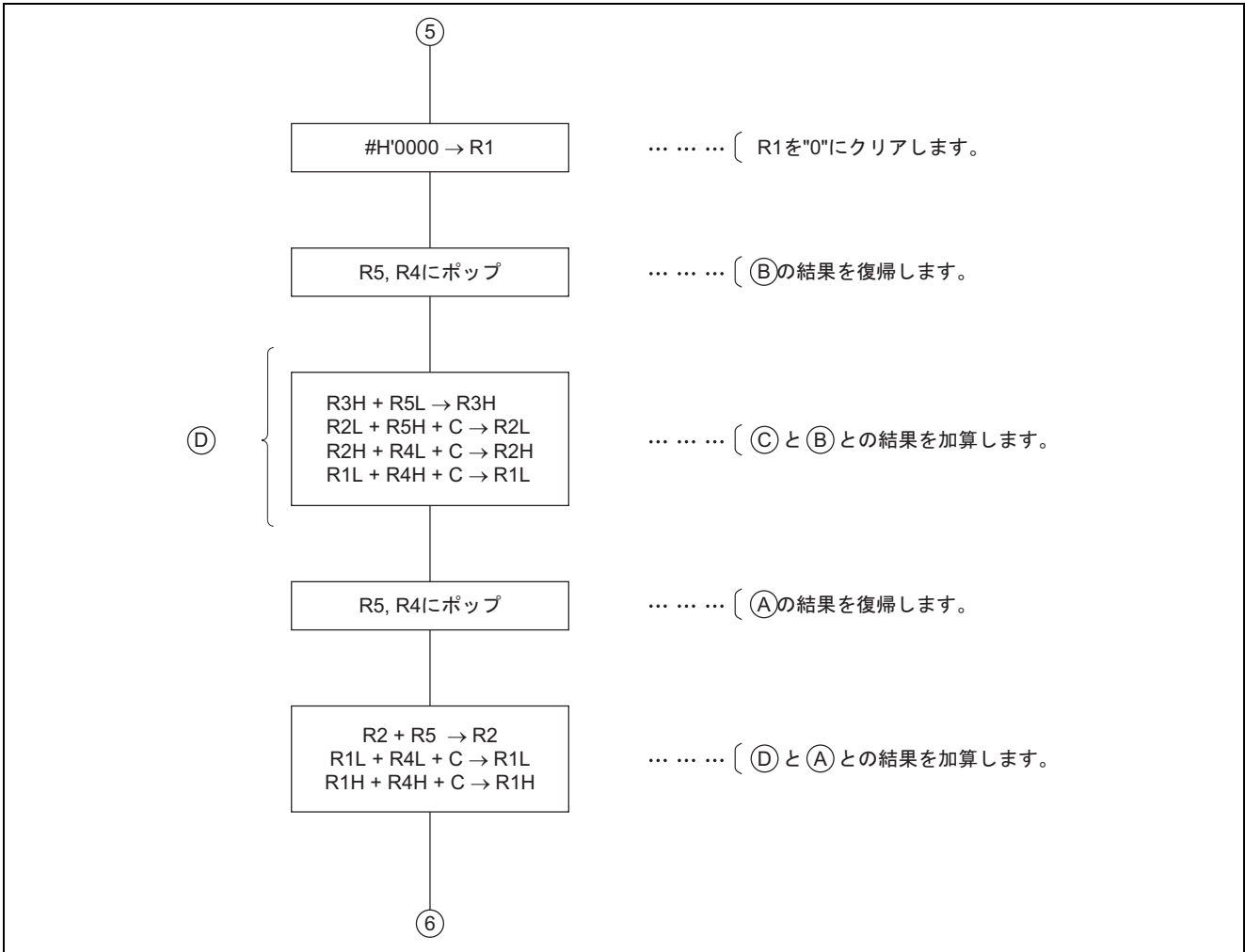
6. フローチャート

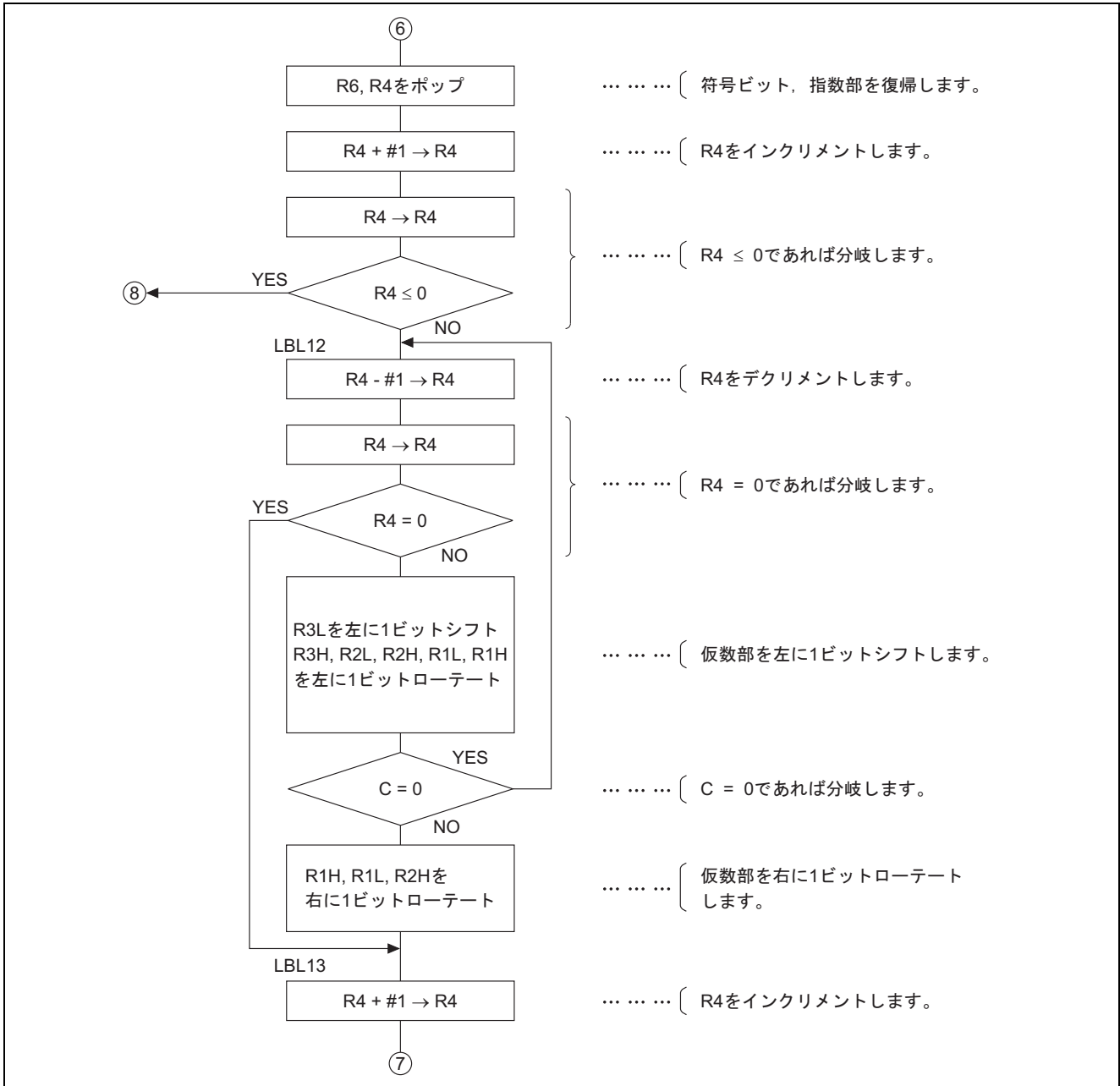


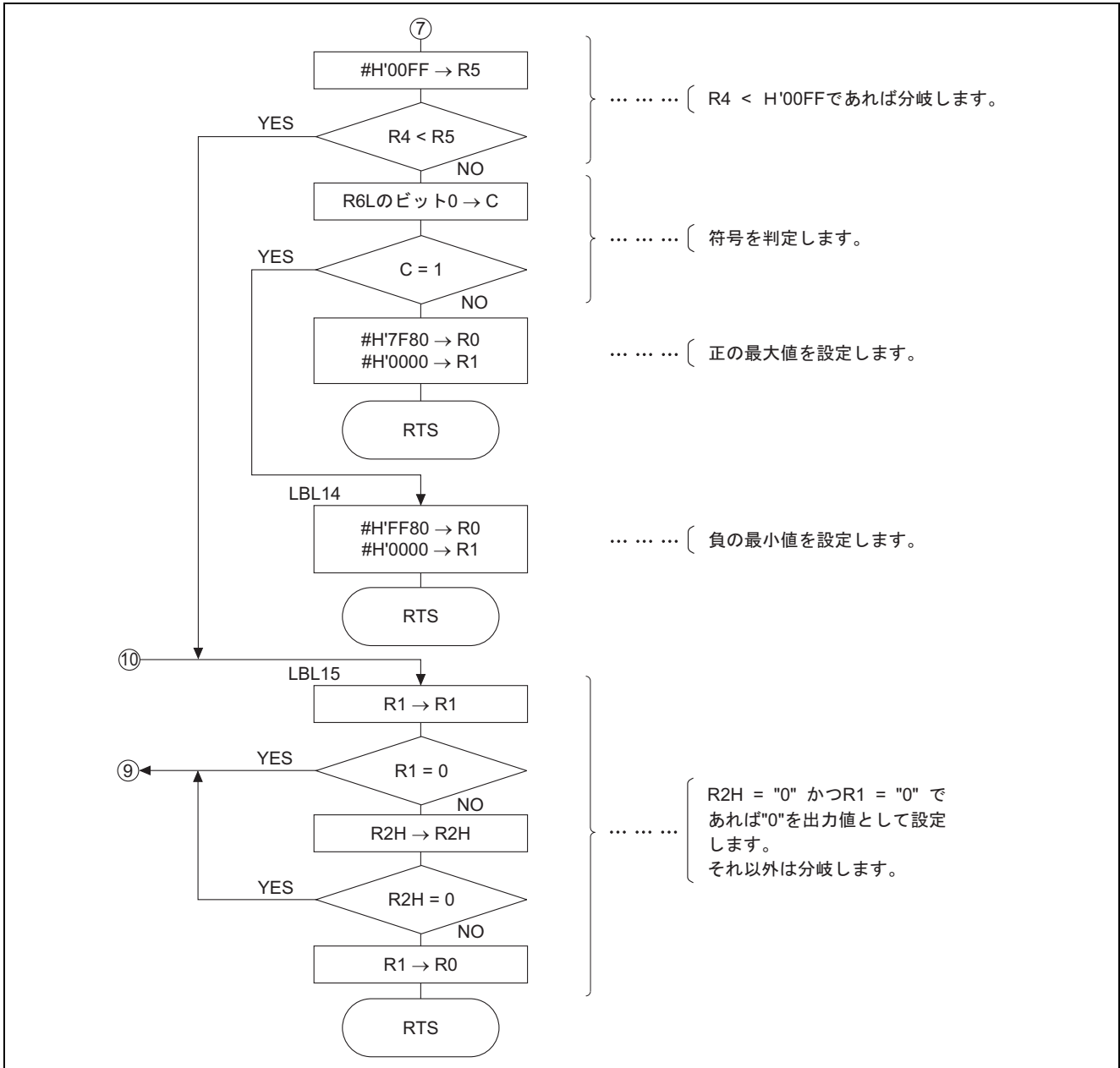


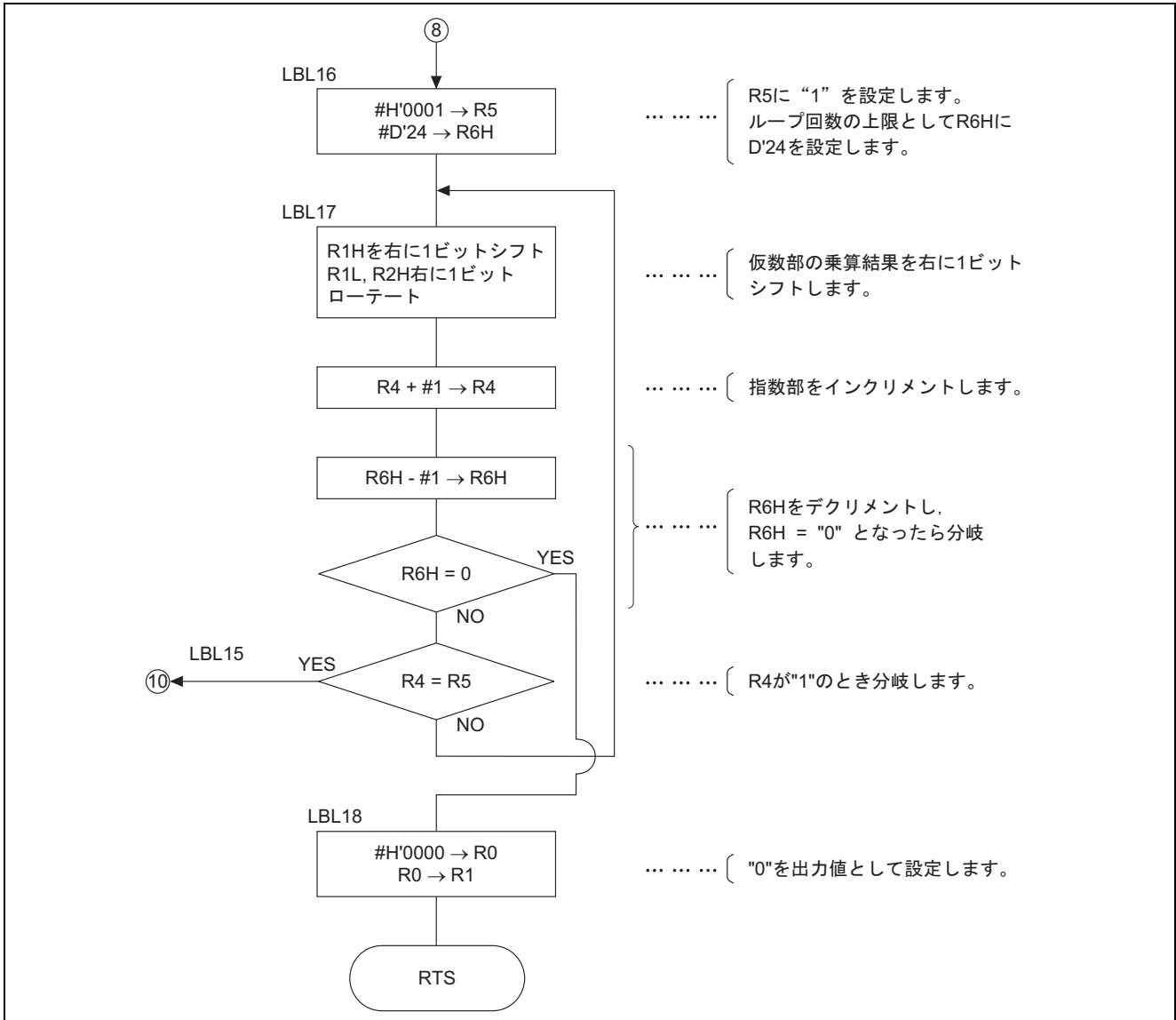


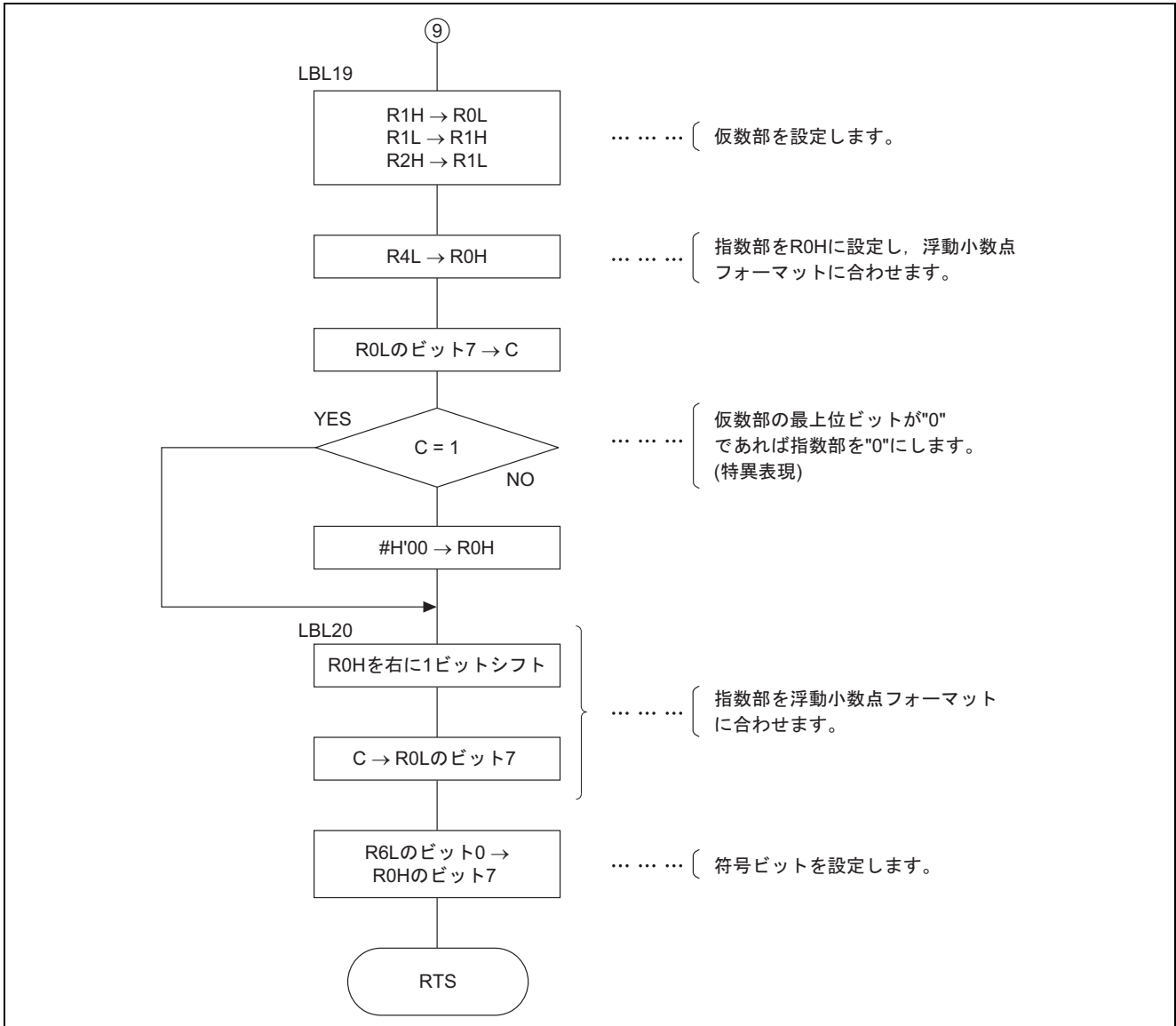


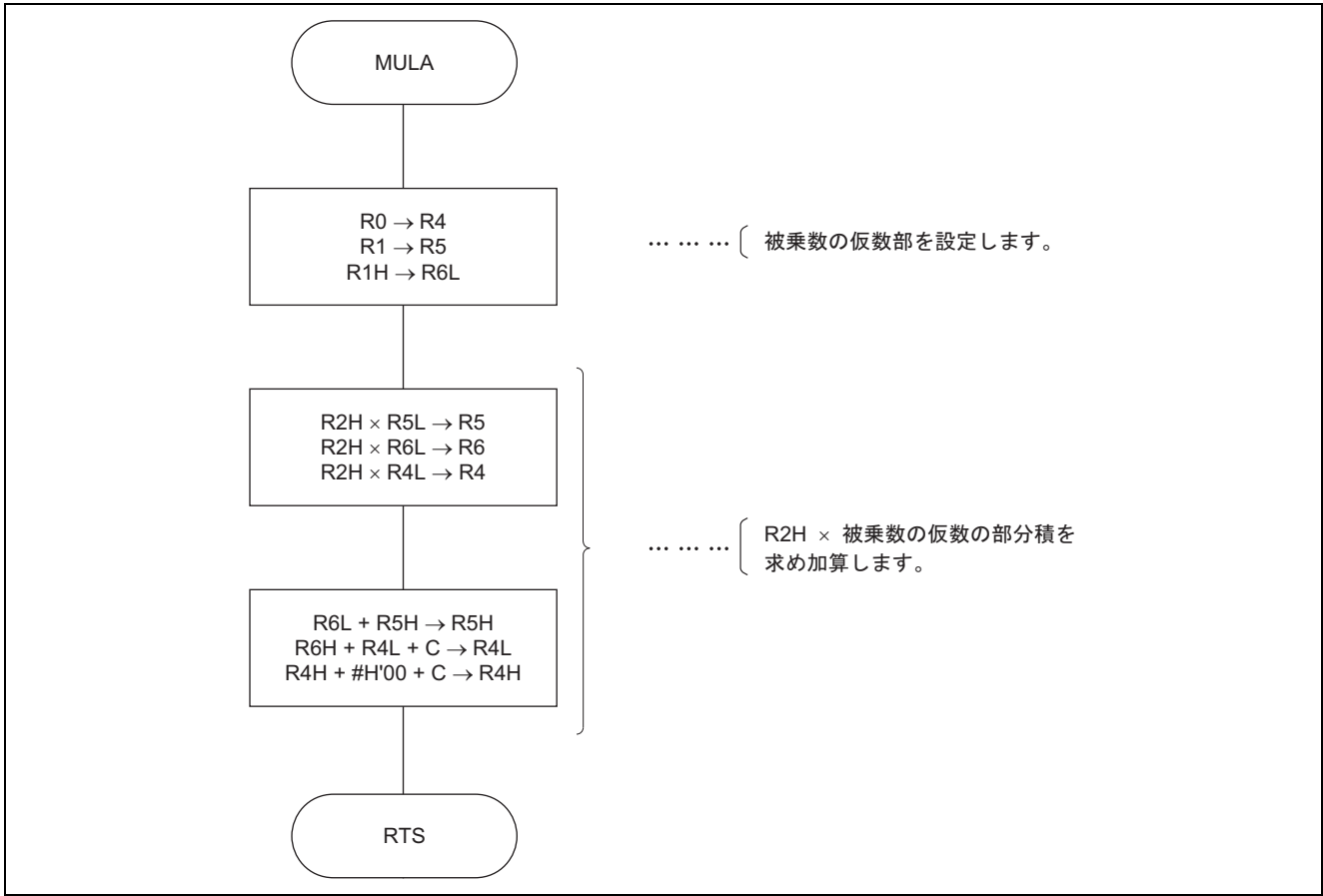












7. プログラムリスト

```

*** H8/300 ASSEMBLER  VER 1.0B **           08/18/92 10:22:23
PROGRAM NAME =
1          ;*****
2          ;*
3          ;*      00 - NAME           :FLOATING POINT MULTIPLICATION (FMUL)
4          ;*
5          ;*****
6          ;*
7          ;*      ENTRY   :R0           (UPPER WORD OF MULTI PLICAND)
8          ;*      R1      (LOWER WORD OF MULTI PLICAND)
9          ;*      R2      (UPPER WORD OF MULTIPLIER)
10         ;*      R3      (LOWER WORD OF MULTIPLIER)
11         ;*
12         ;*      RETURNS :R0           (UPPER WORD OF RESULT)
13         ;*      R1      (LOWER WORD OF RESULT)
14         ;*
15         ;*****
16         ;
17 FMUL_cod C  0000          .SECTION FMUL_code, CODE, ALIGN=2
18                   .EXPORT  FMUL
19         ;
20 FMUL_cod C           00000000 FMUL  .EQU    $           ;Entry point
21 FMUL_cod C  0000 FE00      MOV.B   #H'00,R6L   ;Clear R6L
22 FMUL_cod C  0002 79057F80  MOV.W   #H'7F80,R5 ;Set "H'7F80"
23         ;
24 FMUL_cod C  0006 7770      BLD     #7,R0H    ;Set sign bit of multiplicand
25 FMUL_cod C  0008 670E      BST     #0,R6L    ; to bit 0 of R6L
26 FMUL_cod C  000A 7270      BCLR   #7,R0H    ;Bit clear bit 7 of R0H
27         ;
28 FMUL_cod C  000C 7772      BLD     #7,R2H    ;
29 FMUL_cod C  000E 750E      BXOR   #0,R6L    ;Set sign bit of result
30 FMUL_cod C  0010 670E      BST     #0,R6L    ; to bit 0 of R6L
31 FMUL_cod C  0012 7272      BCLR   #7,R2H    ;Bit clear bit 7 of R2H
32         ;
33 FMUL_cod C  0014 0D11      MOV.W   R1,R1
34 FMUL_cod C  0016 4604      BNE     LBL1
35 FMUL_cod C  0018 0D00      MOV.W   R0,R0
36 FMUL_cod C  001A 4708      BEQ     LBL2      ;Branch if R1=R0=0
37 FMUL_cod C  001C          LBL1
38 FMUL_cod C  001C 0D33      MOV.W   R3,R3
39 FMUL_cod C  001E 460C      BNE     LBL3      ;Branch if not R3=0
40 FMUL_cod C  0020 0D22      MOV.W   R2,R2
41 FMUL_cod C  0022 4608      BNE     LBL3      ;Branch if not R2=0
42         ;
43 FMUL_cod C  0024          LBL2
44 FMUL_cod C  0024 79000000  MOV.W   #H'0000,R0 ;Set 0 to result
45 FMUL_cod C  0028 0D01      MOV.W   R0,R1
46 FMUL_cod C  002A 5470      RTS
47         ;
48 FMUL_cod C  002C          LBL3
49 FMUL_cod C  002C 1D05      CMP.W   R0,R5
50 FMUL_cod C  002E 4304      BLS     LBL4      ;Branch if R0>=R5
51 FMUL_cod C  0030 1D25      CMP.W   R2,R5
52 FMUL_cod C  0032 4218      BHI     LBL7      ;Branch if R2>=R5

```

```

53 FMUL_cod C 0034          LBL4
54 FMUL_cod C 0034 770E      BLD    #0,R6L    ;Load sign bit
55 FMUL_cod C 0036 450A      BCS    LBL6      ;Branch if C=1
56 FMUL_cod C 0038          LBL5
57 FMUL_cod C 0038 79007F80  MOV.W  #H'7F80,R0 ;Set #H'7F800000 to result
58 FMUL_cod C 003C 79010000  MOV.W  #H'0000,R1
59 FMUL_cod C 0040 5470      RTS
60 FMUL_cod C 0042          LBL6
61 FMUL_cod C 0042 7900FF80  MOV.W  #H'FF80,R0 ;Set #H'FF800000 to result
62 FMUL_cod C 0046 79010000  MOV.W  #H'0000,R1
63 FMUL_cod C 004A 5470      RTS
64                          ;
65 FMUL_cod C 004C          LBL7
66 FMUL_cod C 004C 7778      BLD    #7,R0L    ;
67 FMUL_cod C 004E 1200      ROTXL  R0H        ;
68 FMUL_cod C 0050 0C0C      MOV.B  R0H,R4L   ;Set exponent of multiplicand to R4
69 FMUL_cod C 0052 F400      MOV.B  #H'00,R4H
70                          ;
71 FMUL_cod C 0054 777A      BLD    #7,R2L
72 FMUL_cod C 0056 1202      ROTXL  R2H
73 FMUL_cod C 0058 0C2D      MOV.B  R2H,R5L   ;Set exponent of multiplier to R5
74 FMUL_cod C 005A F500      MOV.B  #H'00,R5H
75                          ;
76 FMUL_cod C 005C 7278      BCLR   #7,R0L    ;Clear bit 7 of R0L
77 FMUL_cod C 005E 0C00      MOV.B  R0H,R0H
78 FMUL_cod C 0060 4704      BEQ    LBL8      ;Branch if multiplicand is denormalized
79 FMUL_cod C 0062 7078      BSET   #7,R0L    ;Set implicit MSB
80 FMUL_cod C 0064 4002      BRA    LBL9      ;Branch always
81 FMUL_cod C 0066          LBL8
82 FMUL_cod C 0066 0B04      ADDS.W #1,R4
83                          ;
84 FMUL_cod C 0068          LBL9
85 FMUL_cod C 0068 727A      BCLR   #7,R2L    ;Clear bit 7 of R2L
86 FMUL_cod C 006A 0C22      MOV.B  R2H,R2H
87 FMUL_cod C 006C 4704      BEQ    LBL10     ;Branch if multiplier is denormalized
88 FMUL_cod C 006E 707A      BSET   #7,R2L    ;Set implicit MSB
89 FMUL_cod C 0070 4002      BRA    LBL11     ;Branch always
90 FMUL_cod C 0072          LBL10
91 FMUL_cod C 0072 0B05      ADDS.W #1,R5
92                          ;
93 FMUL_cod C 0074          LBL11
94 FMUL_cod C 0074 0954      ADD.W  R5,R4     ;addition exponents
95 FMUL_cod C 0076 06FE      ANDC   #H'FE,CCR ;Clear C flag of CCR
96 FMUL_cod C 0078 BC7F      SUBX.B #H'7F,R4L ;R4L - #H'7F - C -> R4L
97 FMUL_cod C 007A B400      SUBX.B #H'00,R4H
98                          ;
99 FMUL_cod C 007C 6DF4      PUSH   R4        ;Push R4
100 FMUL_cod C 007E 6DF6     PUSH   R6        ;Push R6
101                          ;
102 FMUL_cod C 0080 0D04     MOV.W  R0,R4     ;
103 FMUL_cod C 0082 0D15     MOV.W  R1,R5
104                          ;
105 FMUL_cod C 0084 0CA2     MOV.B  R2L,R2H
106 FMUL_cod C 0086 5E000000 J      SR    @MULA     ;R2L * (R0L:R1) -> (R4:R5)
107 FMUL_cod C 008A 6DF4     PUSH   R4        ;Push R4
108 FMUL_cod C 008C 6DF5     PUSH   R5        ;Push R5
109                          ;

```

```

110 FMUL_cod C 008E 0C32 MOV.B R3H,R2H
111 FMUL_cod C 0090 5E000000 JSR @MULA ;R3L * (R0L:R1) -> (R4:R5)
112 FMUL_cod C 0094 6DF4 PUSH R4 ;Push R4
113 FMUL_cod C 0096 6DF5 PUSH R5 ;Push R5
114 ;
115 FMUL_cod C 0098 0CB2 MOV.B R3L,R2H ;
116 FMUL_cod C 009A 5E000000 JSR @MULA ;R3L * (R0L:R1) -> (R4:R5)
117 FMUL_cod C 009E 0D42 MOV.W R4,R2 ;Push R4
118 FMUL_cod C 00A0 0D53 MOV.W R5,R3 ;Push R5
119 ;
120 FMUL_cod C 00A2 79010000 MOV.W #H'0000,R1 ;Clear R1
121 FMUL_cod C 00A6 6D75 POP R5 ;Pop R5
122 FMUL_cod C 00A8 6D74 POP R4 ;Pop R4
123 ;
124 FMUL_cod C 00AA 08D3 ADD.B R5L,R3H ;R3H + R5L -> R3H
125 FMUL_cod C 00AC 0E5A ADDX.B R5H,R2L ;R2L + R5H + C -> R2L
126 FMUL_cod C 00AE 0EC2 ADDX.B R4L,R2H ;R2H + R4L + C -> R2H
127 FMUL_cod C 00B0 0E49 ADDX.B R4H,R1L ;R1L + R4H + C -> R1L
128 ;
129 FMUL_cod C 00B2 6D75 POP R5 ;Pop R5
130 FMUL_cod C 00B4 6D74 POP R4 ;Pop R4
131 FMUL_cod C 00B6 0952 ADD.W R5,R2 ;R2 + R5 -> R2
132 FMUL_cod C 00B8 0EC9 ADDX.B R4L,R1L ;R1L + R4L + C -> R1L
133 FMUL_cod C 00BA 0E41 ADDX.B R4H,R1H ;R1H + R4H + C -> R1H
134 ;
135 FMUL_cod C 00BC 6D76 POP R6 ;Pop R6
136 FMUL_cod C 00BE 6D74 POP R4 ;Pop R4
137 FMUL_cod C 00C0 0B04 ADDS.W #1,R4
138 FMUL_cod C 00C2 0D44 MOV.W R4,R4
139 ;
140 FMUL_cod C 00C4 474A BEQ LBL16 ;Branch if R4=0
141 FMUL_cod C 00C6 4B48 BMI LBL16 ;Branch if R4<0
142 FMUL_cod C 00C8 LBL12
143 FMUL_cod C 00C8 1B04 SUBS.W #1,R4
144 FMUL_cod C 00CA 0D44 MOV.W R4,R4
145 FMUL_cod C 00CC 4714 BEQ LBL13 ;Branch if R4=0
146 FMUL_cod C 00CE 100B SHLL R3L ;Shift mantissa 1 bit left
147 FMUL_cod C 00D0 1203 ROTXL R3H
148 FMUL_cod C 00D2 120A ROTXL R2L
149 FMUL_cod C 00D4 1202 ROTXL R2H
150 FMUL_cod C 00D6 1209 ROTXL R1L
151 FMUL_cod C 00D8 1201 ROTXL R1H
152 FMUL_cod C 00DA 44EC BCC LBL12 ;Branch if C=0
153 FMUL_cod C 00DC 1301 ROTXR R1H ;Rotate mantissa 1 bit right
154 FMUL_cod C 00DE 1309 ROTXR R1L
155 FMUL_cod C 00E0 1302 ROTXR R2H
156 FMUL_cod C 00E2 LBL13
157 FMUL_cod C 00E2 0B04 ADDS.W #1,R4
158 ;
159 FMUL_cod C 00E4 790500FF MOV.W #H'00FF,R5 ;
160 FMUL_cod C 00E8 1D45 CMP.W R4,R5
161 FMUL_cod C 00EA 4418 BCC LBL15 ;Branch if R5>R4
162 FMUL_cod C 00EC 770E BLD #0,R6L ;Load sign bit
163 FMUL_cod C 00EE 450A BCS LBL14 ;Branch if C=1
164 FMUL_cod C 00F0 79007F80 MOV.W #H'7F80,R0 ;Set H'7F800000 to result
165 FMUL_cod C 00F4 79010000 MOV.W #H'0000,R1
166 FMUL_cod C 00F8 5470 RTS

```

```

167                                     ;
168 FMUL_cod C 00FA LBL14
169 FMUL_cod C 00FA 7900FF80 MOV.W #H'FF80,R0 ;Set H'FF800000 to product
170 FMUL_cod C 00FE 79010000 MOV.W #H'0000,R1
171 FMUL_cod C 0102 5470 RTS
172                                     ;
173 FMUL_cod C 0104 LBL15
174 FMUL_cod C 0104 0D11 MOV.W R1,R1
175 FMUL_cod C 0106 4628 BNE LBL19 ;Branch if not R1=0
176 FMUL_cod C 0108 0C22 MOV.B R2H,R2H
177 FMUL_cod C 010A 4624 BNE LBL19 ;Branch if not R2H=0
178 FMUL_cod C 010C 0D10 MOV.W R1,R0
179 FMUL_cod C 010E 5470 RTS
180                                     ;
181 FMUL_cod C 0110 LBL16
182 FMUL_cod C 0110 79050001 MOV.W #H'0001,R5 ;Set #H'0001 to R5
183 FMUL_cod C 0114 F618 MOV.B #D'24,R6H ;Se bit counter
184 FMUL_cod C 0116 LBL17
185 FMUL_cod C 0116 1101 SHLR R1H ;Shift mantissa 1 bit right
186 FMUL_cod C 0118 1309 ROTXR R1L
187 FMUL_cod C 011A 1302 ROTXR R2H
188 FMUL_cod C 011C 0B04 ADDS.W #1,R4 ;Increment exponent
189 FMUL_cod C 011E 1A06 DEC.B R6H ;Decrement bit counter
190 FMUL_cod C 0120 4706 BEQ LBL18 ;Branch if Z=1
191 FMUL_cod C 0122 1D54 CMP.W R5,R4
192 FMUL_cod C 0124 47DE BEQ LBL15 ;Branch if R5=R4
193 FMUL_cod C 0126 40EE BRA LBL17 ;Branch always
194 FMUL_cod C 0128 LBL18
195 FMUL_cod C 0128 79000000 MOV.W #H'0000,R0 ;Clear result
196 FMUL_cod C 012C 0D01 MOV.W R0,R1
197 FMUL_cod C 012E 5470 RTS
198                                     ;
199 FMUL_cod C 0130 LBL19
200 FMUL_cod C 0130 0C18 MOV.B R1H,R0L
201 FMUL_cod C 0132 0C91 MOV.B R1L,R1H
202 FMUL_cod C 0134 0C29 MOV.B R2H,R1L
203                                     ;
204 FMUL_cod C 0136 0CC0 MOV.B R4L,R0H
205 FMUL_cod C 0138 7778 BLD #7,R0L
206 FMUL_cod C 013A 4502 BCS LBL20 ;Branch if C=1
207 FMUL_cod C 013C F000 MOV.B #H'00,R0H
208 FMUL_cod C 013E LBL20 ;Change floating point format
209 FMUL_cod C 013E 1100 SHLR R0H
210 FMUL_cod C 0140 6778 BST #7,R0L
211 FMUL_cod C 0142 770E BLD #0,R6L
212 FMUL_cod C 0144 6770 BST #7,R0H
213 FMUL_cod C 0146 5470 RTS
214                                     ;
215                                     ;-----
216                                     ;
217 MUL_cod C 0148 MULA ;R2H * (R0L:R1) -> (R4:R5)
218 MUL_cod C 0148 0D04 MOV.W R0,R4 ;R0 -> R4
219 MUL_cod C 014A 0D15 MOV.W R1,R5 ;R1 -> R5
220 MUL_cod C 014C 0C1E MOV.B R1H,R6L ;R1H -> R6L
221                                     ;

```

```

222 MUL_cod C 014E 5025 MULXU R2H,R5 ;R2H * R5L -> R5
223 MUL_cod C 0150 5026 MULXU R2H,R6 ;R2H * R6L -> R6
224 FMUL_cod C 0152 5024 MULXU R2H,R4 ;R2H * R4L -> R4
225 ;
226 FMUL_cod C 0154 08E5 ADD.B R6L,R5H ;R5H + R6L -> R5H
227 FMUL_cod C 0156 0E6C ADDX.B R6H,R4L ;R4L + R6H + C -> R4L
228 FMUL_cod C 0158 9400 ADDX.B #H'00,R4H ;R4H + #H'00 + C -> R4H
229 FMUL_cod C 015A 5470 RTS
230 ;
231 .END
*****TOTAL ERRORS 0
*****TOTAL WARNINGS 0

```

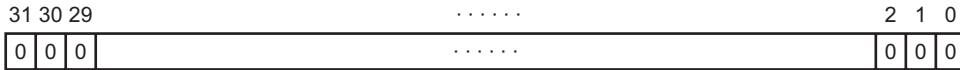
単精度浮動小数点について < 参考 >

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

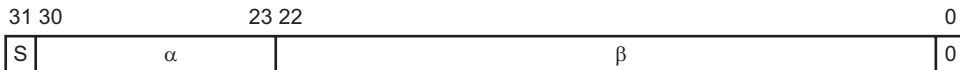
本アプリケーションノートでは、単精度浮動小数点の値 (R = 実数) により、以下の表現方法を用います。

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 ≤ α ≤ 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 ≤ i ≤ 22) 番目の 1 ビットの値です。S は符号ビットです。

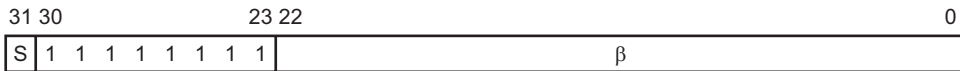
C. 特異表現 (Denormalized Format)



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。

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

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

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	21, 22	< 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。