

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。

# 17Kシリーズ

## 4ビット・シングルチップ・マイクロコントローラ

浮動小数点演算パッケージ

概 説	1
ハードウェア構成	2
電卓仕様	3
システム制御部概要	4
RAMレイアウトおよび変数説明	5
システム制御部フロー・チャート	6
浮動小数点形式（RAM上のデータの格納形式）	7
演算パッケージ説明	8
電卓プログラム	9

本資料に掲載の応用回路および回路定数は、例示的に示したものであり、量産設計を対象とするものではありません。

本製品が外国為替および外国貿易管理法の規定による戦略物資等（または役務）に該当するか否かは、ユーザー（仕様を決定した者）が判定してください。

- 文書による当社の承諾なしに本資料の転載複製を禁じます。
  - この製品を使用したことにより、第三者の工業所有権等にかかわる問題が発生した場合、当社製品の構造製法に直接かかわるもの以外につきましては、当社はその責を負いませんのでご了承ください。
  - 当社は品質、信頼性の向上に努めていますが、半導体製品はある確率で故障が発生します。当社半導体製品の故障により結果として、人身事故、火災事故、社会的な損害等を生じさせない冗長設計、延焼対策設計、誤動作防止設計等安全設計に十分ご注意ください。
  - 当社は、当社製品の品質水準を「標準水準」、「特別水準」およびお客様に品質保証プログラムを指定して頂く「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の上ご使用願います。
    - 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
    - 特別水準：輸送機器（自動車、列車、船舶等）、交通用信号機器、防災／防犯装置、各種安全装置、生命維持を直接の目的としない医療機器
    - 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、生命維持のための装置またはシステム等
- 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。当社製品を上記の「標準水準」の用途以外でご使用をお考えのお客様は、必ず事前に当社販売窓口までご相談頂きますようお願い致します。
- この製品は耐放射線設計をしておりません。

巻末にアンケート・コーナーを設けております。このドキュメントに対するご意見をお気軽にお寄せください。

# はじめに

**ご利用対象者** このアプリケーション・ノートは、17Kシリーズの機能を理解し、それを用いたアプリケーション・プログラムを設計するエンジニアを対象とします。

**目的** このアプリケーション・ノートは、17Kシリーズの持つ機能を、応用プログラム例を用いてユーザーに理解していただくことを目的としています。

**構成** このアプリケーション・ノートは、大きく分けて次の内容で構成しています。

- 概説
- ハードウェア構成
- 電卓仕様
- システム制御部概要
- RAMレイアウトおよび変数説明
- システム制御部フロー・チャート
- 浮動小数点形式
- 演算パッケージ説明
- プログラム・リスト

**凡例** データ表記の重み：左が上位桁、右が下位桁

注：本文中につけた注の説明

注意：気をつけて読んでいただきたい内容

備考：本文の補足説明

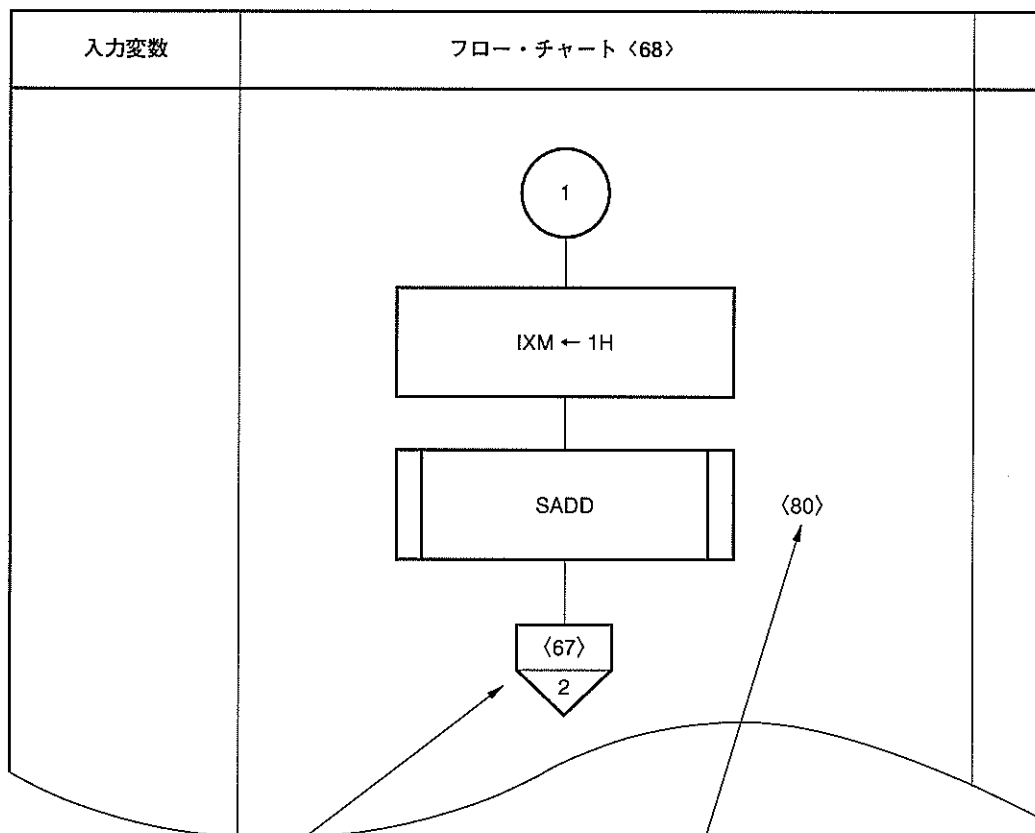
数の表記：10進数……××××  
16進数……××××H

**関連資料** 関連資料は、以下のとおりです。あわせてご利用ください。

- $\mu$ PD17201A, 17207 データ・シート (IC-8194)
- $\mu$ PD172××シリーズ ユーザーズ・マニュアル (IEU-762)

## フロー・チャートの見方

このアプリケーション・ノートで使われているフロー・チャートの見方を次の例によって示します。



フロー・チャート <67> の  
②に飛びます。

フロー・チャート <80> を  
参照してください。



# 目 次

第1章	概 説	…	1
第2章	ハードウェア構成	…	3
2.1	システム・ブロック図	…	3
2.2	回路図	…	4
2.3	端子割り付け	…	5
2.4	端子機能説明	…	8
2.5	表 示	…	9
2.6	キー入力	…	11
第3章	電卓仕様	…	13
3.1	仕 様	…	13
3.2	キー	…	13
3.3	状態遷移	…	16
3.4	表 示	…	18
3.5	オートパワーオフ	…	18
3.6	演算例	…	18
第4章	システム制御部概要	…	27
4.1	メイン処理	…	27
4.2	キー・スキャン処理	…	28
4.3	キー・ブランチ処理	…	30
4.4	演算分岐処理, エラー処理	…	31
4.5	オートパワーオフ・タイマ管理	…	35
4.6	STOPモード処理	…	36
第5章	RAMレイアウトおよび変数説明	…	37
5.1	RAMレイアウト	…	37
5.2	RAMおよびフラグ説明	…	40
5.3	処理およびRAMの名称説明	…	43
第6章	システム制御部フロー・チャート	…	45
6.1	初期設定	…	46
6.2	メイン処理	…	49
6.3	キー・スキャン処理	…	51
6.4	キー・リターン値判定処理	…	55
6.5	インプット・コード生成処理	…	57
6.6	オートパワーオフ・タイマ管理	…	58

6.7	STOPモード処理	…	59
6.8	キー・ブランチ処理	…	61
6.9	オール・クリア・キー処理	…	65
6.10	クリア・キー処理	…	66
6.11	数値キー，小数点キー処理	…	68
6.12	符号キー処理	…	75
6.13	演算子キー処理	…	77
6.14	パーセント・キー処理	…	82
6.15	イコール・キー処理	…	84
6.16	演算分岐処理，エラー処理	…	86
6.17	RAMオール・クリア処理	…	90
6.18	表示データ・エリア・イニシャライズ処理	…	92
6.19	REGYクリア処理	…	93
6.20	表示データ・エリア・ダウン・シフト処理	…	94
6.21	表示データ・エリア・アップ・シフト処理	…	95
6.22	REGYアップ・シフト処理	…	96
6.23	データ転送処理	…	97
6.24	演算結果変換処理	…	98
6.25	表示データ変換処理	…	102
6.26	表示データ出力処理	…	106
第7章	浮動小数点形式（RAM上のデータの格納形式）	…	109
7.1	数値の管理構成	…	109
7.2	浮動小数点レジスタ	…	110
7.3	浮動小数点形式の各部の説明	…	111
7.4	浮動小数点レジスタへの格納例	…	112
第8章	演算パッケージ説明	…	113
8.1	サブルーチン一覧表	…	113
8.2	四則演算	…	115
8.3	その他のサブルーチン（ユーザから呼ばれることのない内部ルーチン）	…	144
第9章	電卓プログラム	…	163
9.1	システム制御部プログラム	…	163
9.2	浮動小数点演算部プログラム	…	232

## 図 の 目 次

図番号	タイトル, ページ
2-1	システム・ブロック図 … 3
2-2	回路図 … 4
2-3	LCDパネルの結線例 … 10
2-4	キー入力回路図 … 12
3-1	状態遷移図 … 17
3-2	表 示 … 18
5-1	RAMレイアウト … 38
7-1	数値の管理構成 … 109
7-2	浮動小数点レジスタのRAM配置イメージ … 110
7-3	指数部と指数値の対応 … 111
8-1	筆算による乗算例 … 125
8-2	復帰法による除算例 … 138

## 表 の 目 次

表番号	タイトル, ページ
2-1	端子割り付け … 5
4-1	入力キーとキー・コードの対応 … 28
4-2	有効なキー入力 … 28
4-3	演算子データの変化 … 32
7-1	正 規 化 … 111
8-1	サブルーチン一覧表 … 114

# 第1章 概 説

制御機器の分野では、マイクロコンピュータにより、外部からのアナログ信号入力をデジタル信号に変換し、各種の演算を加えた値を使用するケースが多くあります。

今回、4ビット・シングルチップ・マイクロコントローラの17Kシリーズのうち、赤外線リモート・コントローラ用の $\mu$ PD17201A, 17207を使用し、浮動小数点演算パッケージ、およびその応用例として電卓を制御するプログラムを作成しましたので、紹介します。

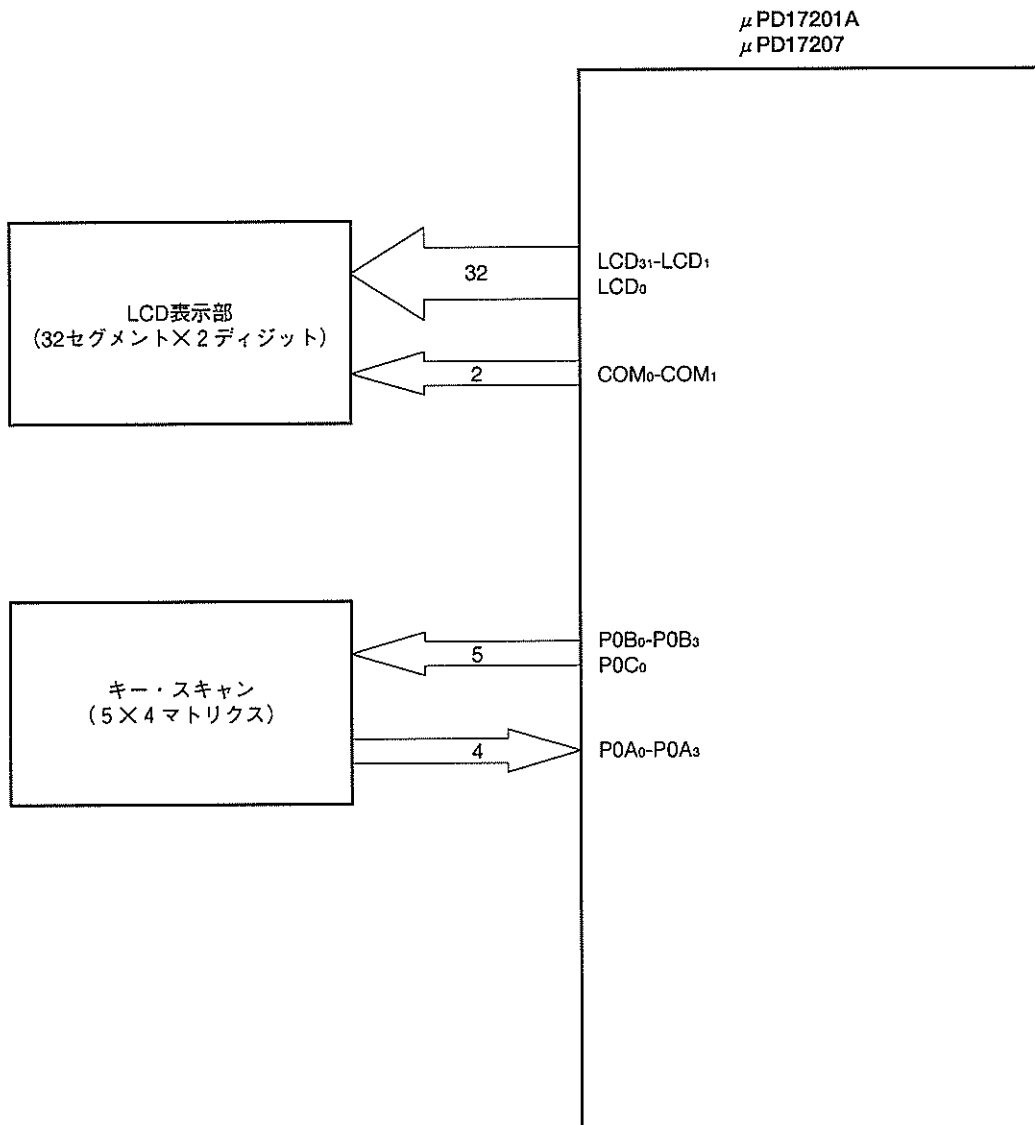


## 第2章 ハードウェア構成

### 2.1 システム・ブロック図

図2-1に、このアプリケーション・ノートで紹介する電卓のシステム・ブロック図を示します。

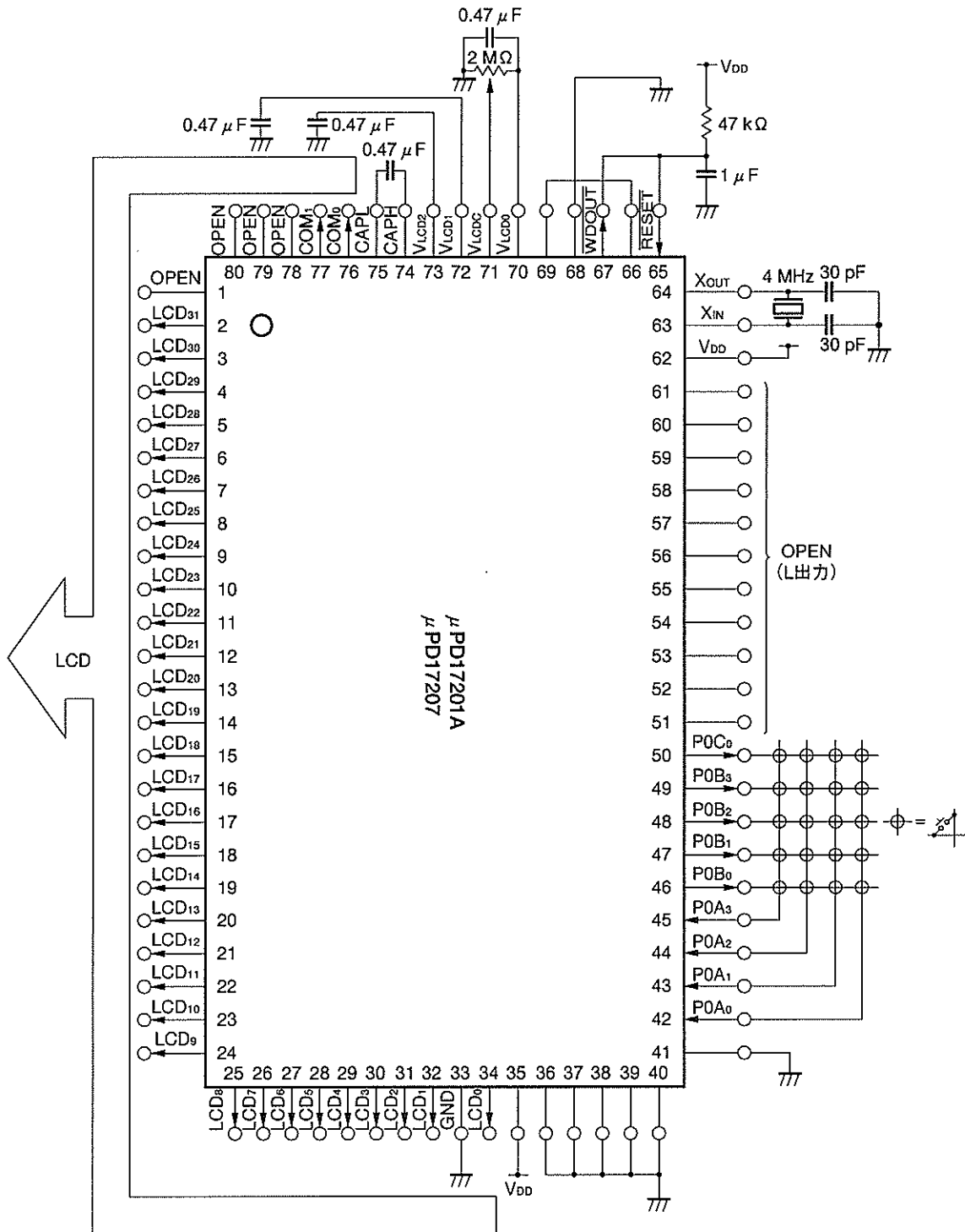
図2-1 システム・ブロック図



## 2.2 回路図

図2-2に、この電卓の回路図を示します。

図2-2 回路図



## 2.3 端子割り付け

表 2-1 端子割り付け (1/3)

端子番号	端子名称	入出力	論理	初期 設定	機 能	マスク・ オプション
1	LCD32	出力	—	—	未使用 (オープン)	
2	LCD31	出力	—	—	LCDコントローラ/ドライバのセグメント信号出力	
3	LCD30	出力	—	—	LCDコントローラ/ドライバのセグメント信号出力	
4	LCD29	出力	—	—	LCDコントローラ/ドライバのセグメント信号出力	
5	LCD28	出力	—	—	LCDコントローラ/ドライバのセグメント信号出力	
6	LCD27	出力	—	—	LCDコントローラ/ドライバのセグメント信号出力	
7	LCD26	出力	—	—	LCDコントローラ/ドライバのセグメント信号出力	
8	LCD25	出力	—	—	LCDコントローラ/ドライバのセグメント信号出力	
9	LCD24	出力	—	—	LCDコントローラ/ドライバのセグメント信号出力	
10	LCD23	出力	—	—	LCDコントローラ/ドライバのセグメント信号出力	
11	LCD22	出力	—	—	LCDコントローラ/ドライバのセグメント信号出力	
12	LCD21	出力	—	—	LCDコントローラ/ドライバのセグメント信号出力	
13	LCD20	出力	—	—	LCDコントローラ/ドライバのセグメント信号出力	
14	LCD19	出力	—	—	LCDコントローラ/ドライバのセグメント信号出力	
15	LCD18	出力	—	—	LCDコントローラ/ドライバのセグメント信号出力	
16	LCD17	出力	—	—	LCDコントローラ/ドライバのセグメント信号出力	
17	LCD16	出力	—	—	LCDコントローラ/ドライバのセグメント信号出力	
18	LCD15	出力	—	—	LCDコントローラ/ドライバのセグメント信号出力	
19	LCD14	出力	—	—	LCDコントローラ/ドライバのセグメント信号出力	
20	LCD13	出力	—	—	LCDコントローラ/ドライバのセグメント信号出力	
21	LCD12	出力	—	—	LCDコントローラ/ドライバのセグメント信号出力	
22	LCD11	出力	—	—	LCDコントローラ/ドライバのセグメント信号出力	
23	LCD10	出力	—	—	LCDコントローラ/ドライバのセグメント信号出力	
24	LCD9	出力	—	—	LCDコントローラ/ドライバのセグメント信号出力	
25	LCD8	出力	—	—	LCDコントローラ/ドライバのセグメント信号出力	
26	LCD7	出力	—	—	LCDコントローラ/ドライバのセグメント信号出力	
27	LCD6	出力	—	—	LCDコントローラ/ドライバのセグメント信号出力	
28	LCD5	出力	—	—	LCDコントローラ/ドライバのセグメント信号出力	
29	LCD4	出力	—	—	LCDコントローラ/ドライバのセグメント信号出力	
30	LCD3	出力	—	—	LCDコントローラ/ドライバのセグメント信号出力	
31	LCD2	出力	—	—	LCDコントローラ/ドライバのセグメント信号出力	
32	LCD1	出力	—	—	LCDコントローラ/ドライバのセグメント信号出力	
33	GND	—	—	—	グラウンド	



表2-1 端子割り付け (2/3)

端子番号	端子名称	入出力	論理	初期 設定	機能	マスク・ オプション
34	LCD <sub>0</sub>	出力	—	—	LCDコントローラ/ドライバのセグメント信号出力	
35	V <sub>ADC</sub>	—	—	—	未使用 (V <sub>DD</sub> に接続)	
36	ADC <sub>0</sub>	入力	—	—	未使用 (GNDに接続)	
37	ADC <sub>1</sub>	入力	—	—	未使用 (GNDに接続)	
38	ADC <sub>2</sub>	入力	—	—	未使用 (GNDに接続)	
39	ADC <sub>3</sub>	入力	—	—	未使用 (GNDに接続)	
40	GND <sub>ADC</sub>	—	—	—	未使用 (GNDに接続)	
41	INT	入力	—	—	未使用 (GNDに接続)	
42	POA <sub>0</sub>	入力	—	—	キー入力	
43	POA <sub>1</sub>	入力	—	—	キー入力	
44	POA <sub>2</sub>	入力	—	—	キー入力	
45	POA <sub>3</sub>	入力	—	—	キー入力	
46	POB <sub>0</sub>	出力	L	H	キー・スキャン出力	
47	POB <sub>1</sub>	出力	L	H	キー・スキャン出力	
48	POB <sub>2</sub>	出力	L	H	キー・スキャン出力	
49	POB <sub>3</sub>	出力	L	H	キー・スキャン出力	
50	POC <sub>0</sub>	出力	L	H	キー・スキャン出力	
51	POC <sub>1</sub>	出力	—	L	未使用 (オープン)	
52	POC <sub>2</sub>	出力	—	L	未使用 (オープン)	
53	POC <sub>3</sub>	出力	—	L	未使用 (オープン)	
54	POD <sub>0</sub> /LED	出力	—	L	未使用 (オープン)	
55	POD <sub>1</sub> /TMOUT	出力	—	L	未使用 (オープン)	
56	POD <sub>2</sub>	出力	—	L	未使用 (オープン)	
57	POD <sub>3</sub>	出力	—	L	未使用 (オープン)	
58	P1A <sub>0</sub> /SCK	出力	—	L	未使用 (オープン)	
59	P1A <sub>1</sub> /SO	出力	—	L	未使用 (オープン)	
60	P1A <sub>2</sub> /SI	出力	—	L	未使用 (オープン)	
61	REM	出力	—	L	未使用 (オープン)	
62	V <sub>DD</sub>	—	—	—	正電源 (+5 V)	
63	X <sub>IN</sub>	—	—	—	メイン・クロック接続	
64	X <sub>OUT</sub>	—	—	—	メイン・クロック接続	
65	RESET	入力	L	—	リセット入力	ブルアップ
66	V <sub>REG</sub>	—	—	—	未使用 (X <sub>TOUT</sub> に接続)	
67	WDO <sub>UT</sub>	出力	L	—	暴走検出出力	
68	X <sub>TIN</sub>	—	—	—	未使用 (GNDに接続)	

表2-1 端子割り付け (3/3)

端子番号	端子名称	入出力	論理	初期 設定	機 能	マスク・ オプション
69	XTOUT	—	—	—	未使用 (VREGに接続)	
70	V <sub>LCD0</sub>	出力	—	—	LCDドライブ用基準電圧出力	
71	V <sub>LCD0C</sub>	出力	—	—	LCDドライブ用基準電圧調整出力	
72	V <sub>LCD1</sub>	出力	—	—	LCDドライブ用ダブル出力	
73	V <sub>LCD2</sub>	出力	—	—	LCDドライブ用トリプル出力	
74	CAPH	—	—	—	LCDドライブ電圧昇圧用コンデンサ接続	
75	CAPL	—	—	—	LCDドライブ電圧昇圧用コンデンサ接続	
76	COM <sub>0</sub>	出力	—	—	LCDコントローラ/ドライバのコモン信号出力	
77	COM <sub>1</sub>	出力	—	—	LCDコントローラ/ドライバのコモン信号出力	
78	LCD <sub>35</sub> /COM <sub>2</sub>	出力	—	—	未使用 (オープン)	
79	LCD <sub>34</sub> /COM <sub>3</sub>	出力	—	—	未使用 (オープン)	
80	LCD <sub>33</sub>	出力	—	—	未使用 (オープン)	

## 2.4 端子機能説明

端子番号	端子名称	機能説明
2   32  34	LCD <sub>31</sub>   LCD <sub>1</sub>  LCD <sub>0</sub>	LCDコントローラ/ドライバのセグメント信号出力です。 COM <sub>0</sub> -COM <sub>1</sub> とともにマトリクス (32×2) を構成します。
33	GND	グラウンドです。
42   45	POA <sub>0</sub>   POA <sub>3</sub>	キー入力です。 POB <sub>0</sub> -PCB <sub>3</sub> , POC <sub>0</sub> とともにマトリクス (5×4) を構成します。
46   49  50	POB <sub>0</sub>   POB <sub>3</sub>  POC <sub>0</sub>	キー・スキャン出力です。 POA <sub>0</sub> -POA <sub>3</sub> とともにマトリクス (5×4) を構成します。
62	V <sub>DD</sub>	正電源です。
63  64	X <sub>IN</sub>  X <sub>OUT</sub>	メイン・クロック発振用の4 MHzセラミック発振子を接続します。
65	$\overline{\text{RESET}}$	システム・リセット用の入力です。 ロウ・レベル入力によってリセットがかかります。 ロウ・レベル入力中は、メイン・クロックの発振が停止します。
67	$\overline{\text{WDOUT}}$	暴走検出用の出力です。 ウォッチドッグ・タイマのオーバフローやスタックのオーバフロー/アンダフロー時にロウ・レベルを出力します。 $\overline{\text{RESET}}$ 端子と接続してください。
71	V <sub>LCD0C</sub>	LCDドライブ用基準電圧を調整するための出力です。
70 71 72	V <sub>LCD0</sub> V <sub>LCD1</sub> V <sub>LCD2</sub>	LCDドライブ用基準電圧の出力です。 ・ V <sub>LCD0</sub> : 基準電圧出力 ・ V <sub>LCD1</sub> : ダブラ出力 (2倍の電圧) ・ V <sub>LCD2</sub> : トリブラ出力 (3倍の電圧)
74 75	CAPH CAPL	LCDドライブ電圧の昇圧用コンデンサを接続します。
76  77	COM <sub>0</sub>  COM <sub>1</sub>	LCDコントローラ/ドライバのコモン信号出力です。 LCD <sub>31</sub> -LCD <sub>0</sub> とともにマトリクス (32×2) を構成します。

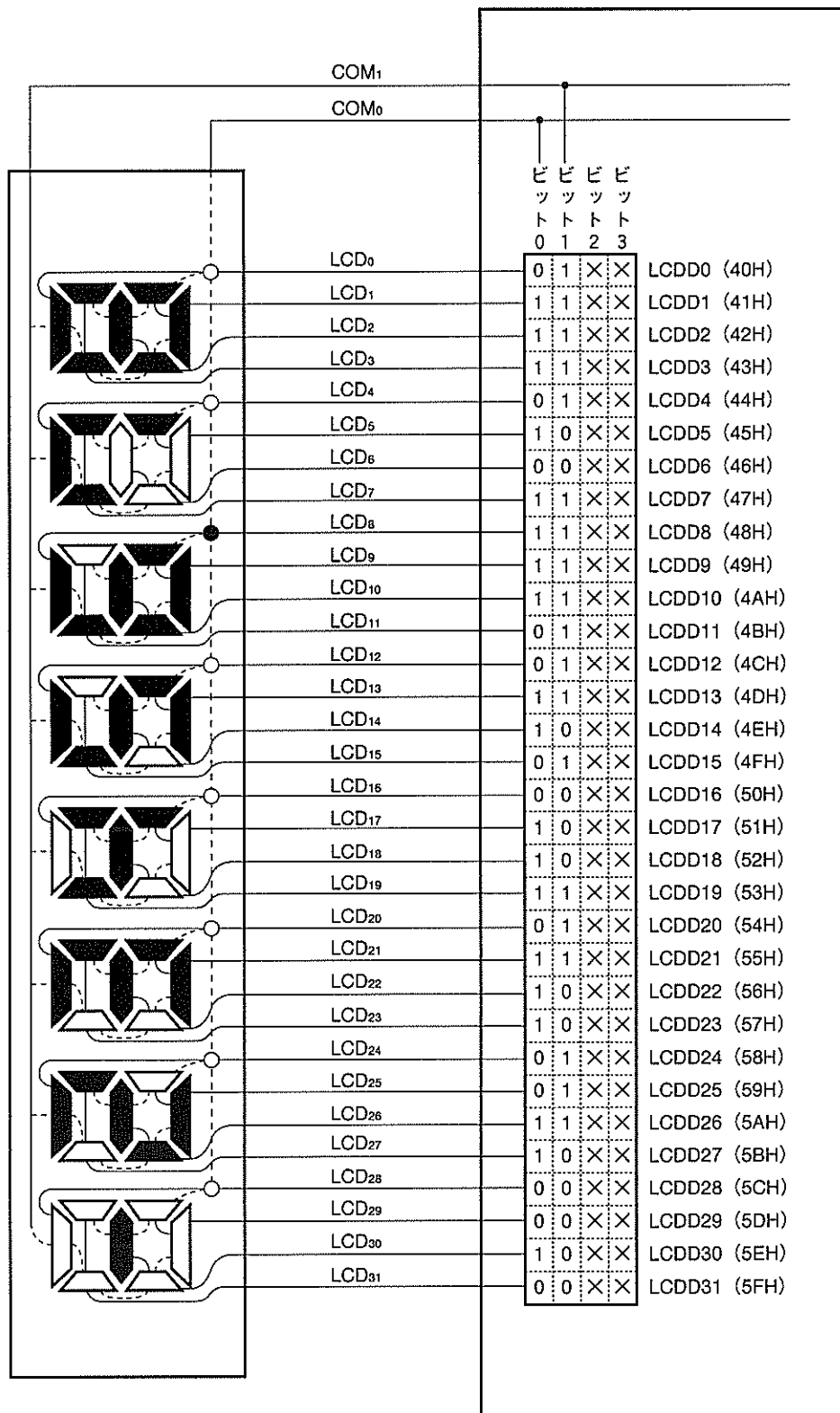
## 2.5 表 示

このアプリケーション・ノートで紹介する電卓の表示は、 $2 \times 32$ のマトリクスを用いたLCD表示を使用します。

表示の制御は、 $\mu$ PD17201A, 17207内蔵のLCDコントローラ/ドライバにより行います。LCDコントローラ/ドライバは、DMA (Direct Memory Access) 動作により、表示データ・メモリのセグメント・データを自動的に読み出してセグメント信号とコモン信号を発生します。表示データ・メモリはLCDD0-LCDD35 (BANK0の40H番地-63H番地) にマッピングされていますが、この電卓ではこのうちLCDD0-LCDD31 (BANK0の40H番地-5FH番地) を使用しています。

図2-3に、この電卓のLCDパネルの結線例を示します。

図 2-3 LCDパネルの結線例



備考 このアプリケーション・ノートで紹介する電卓仕様では2時分割表示を使用しているため、表示データ・メモリのビット2、ビット3（Xの箇所）は使用しません。

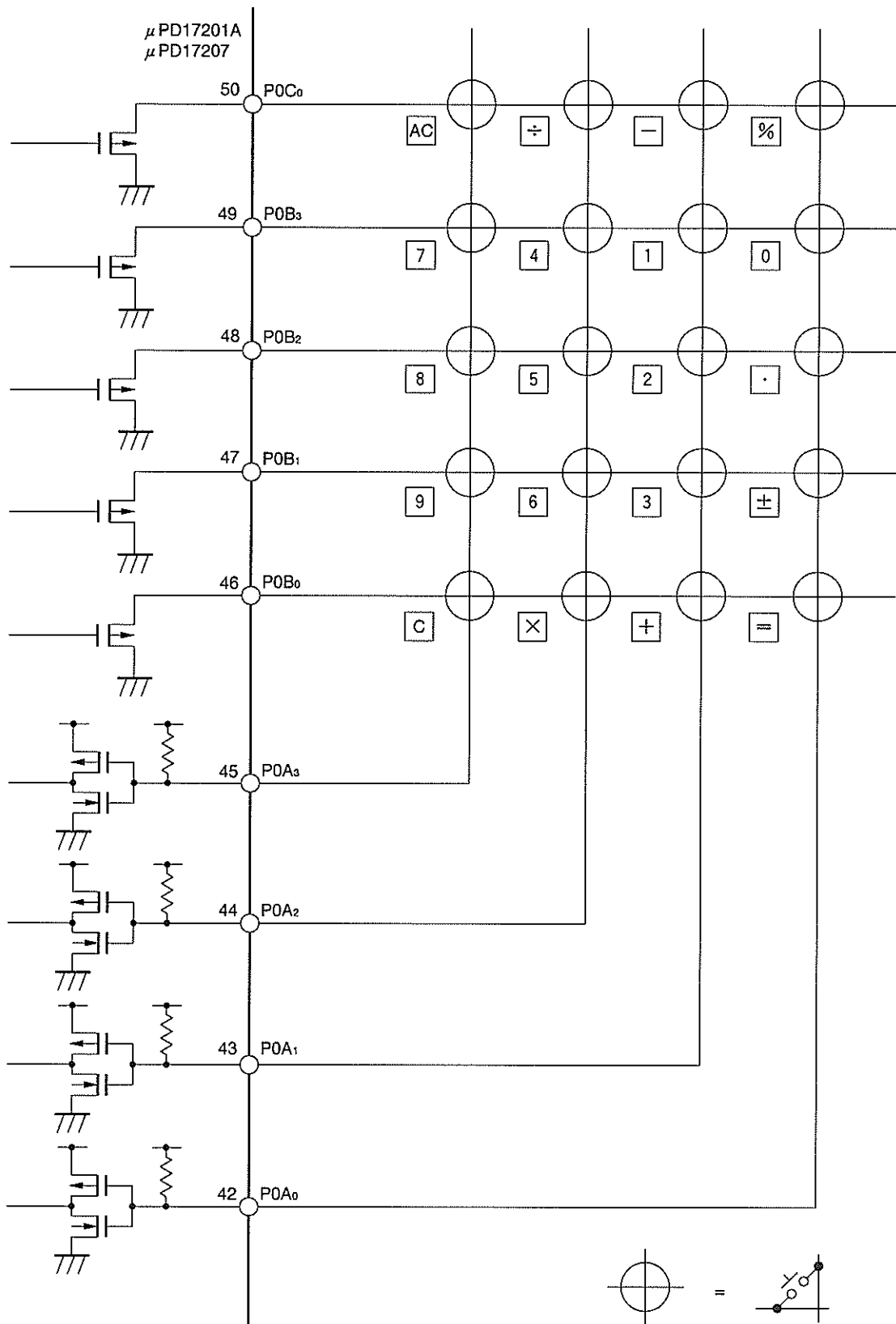
## 2.6 キー入力

このアプリケーション・ノートで紹介する電卓のキーは、 $5 \times 4$ のマトリクスで構成されています。P0B0-P0B3, P0C0から10msごとにキー・スキャン信号を出力し、P0A0-P0A3にポートの状態を入力し、押されているキーを判別します。キー・スキャンはロウ・アクティブで行います。P0B0-P0B3, P0C0のおおのこの端子を時分割でロウ・レベルにし、これと同期してP0A0-P0A3を読み出し、ロウ・レベルの検出でキー入力を判定します。

同じキー状態を3回検出した時点でキー入力を確定し、キー動作を行います。キー入力確定すると、そのキーが離されるまでは（キーが何も押されていない状態が確定するまでは）、他のキー入力を受け付けません。また、2個以上のキーが同時に押されたときは多重押しと判定し、すべてのキーが離されるまで他のキー入力を受け付けません。

図2-4に、この電卓のキー入力の回路図を示します。

図2-4 キー入力回路図



## 第3章 電卓仕様

### 3.1 仕様

表示機能：8桁LCD表示

(符号1桁+数値7桁)

入力方法：5×4マトリクス・キー入力

ロック・アウト方式<sup>※</sup>

入力/表示可能桁数：最大7桁

(±0.000001～±9999999)

特殊機能：オートパワーオフ機能(3分間)

演算種類：・四則演算

・パーセント計算、割り増し、割り引き計算

注 押されているキーが離れるまでは、他のキー入力を受け付けません。

### 3.2 キー

このアプリケーション・ノートで紹介する電卓には、次のようなキーがあります。

(1) 数値キー (           )

数値の入力に使用します。入力できる数値は、符号を除いて最大7桁までです。

(2) 小数点キー (  )

小数点の入力に使用します。

(3) 符号キー (  )

入力中の数値の符号を変えるときに使用します。



(4) 演算子キー (  $\boxed{+}$   $\boxed{-}$   $\boxed{\times}$   $\boxed{\div}$  )

演算子の入力に使用します。演算子キーを続けて押したときは、最後に押された演算子キーが有効になります。

キー入力	表 示
$\boxed{2}$ $\boxed{5}$ $\boxed{0}$	2 5 0.
$\boxed{+}$	2 5 0.
$\boxed{-}$	2 5 0.
$\boxed{\times}$	2 5 0.
$\boxed{2}$	2.
$\boxed{=}$	5 0 0.

この例の場合、 $[250 \times 2]$  が実行され、計算結果は500になります。

(5) パーセント・キー (  $\boxed{\%}$  )

パーセント計算、割り増し、割り引き計算を行うときに使用します。

(6) イコール・キー (  $\boxed{=}$  )

計算結果を求めるときに使用します。

(7) クリア・キー (  $\boxed{C}$  )

入力中の数値の訂正に使用します。以前に入力された数値や演算子は保持されます。

キー入力	表 示
$\boxed{5}$ $\boxed{0}$ $\boxed{0}$	5 0 0.
$\boxed{+}$	5 0 0.
$\boxed{4}$ $\boxed{0}$ $\boxed{0}$	4 0 0.
$\boxed{C}$	0.
$\boxed{3}$ $\boxed{0}$ $\boxed{0}$	3 0 0.
$\boxed{=}$	8 0 0.

この例の場合、 $[500 + 400]$  が  $[500 + 300]$  に訂正され、計算結果は800になります。

キー入力	表 示
7 5 0	7 5 0.
×	7 5 0.
C	0.
=	0.

この例の場合、 $[750 \times 0]$  が実行され、計算結果は0になります。

(8) オール・クリア・キー (AC)

それまでの入力をすべて無効にし、新たな計算を始めるときに使用します。

キー入力	表 示
9 0 0	9 0 0.
—	9 0 0.
1 0 0	1 0 0.
AC	0.
7 0 0	7 0 0.
×	7 0 0.
5	5.
=	3 5 0 0.

この例の場合、 $[900 - 100]$  の入力がすべて無効になり、 $[700 \times 5]$  が実行され、計算結果は3500になります。

また、電源オン・スイッチを兼ねています。

### 3.3 状態遷移

このアプリケーション・ノートで紹介する電卓には、次の6つのモードがあります。

(1) 左項モード

演算の初項を入力します。

(2) 演算子選択モード

演算子を入力します。

(3) 右項モード

演算の次項を入力します。

(4) パーセント計算モード

パーセント計算、割り増し、割り引き計算を行います。

(5) エラー・モード

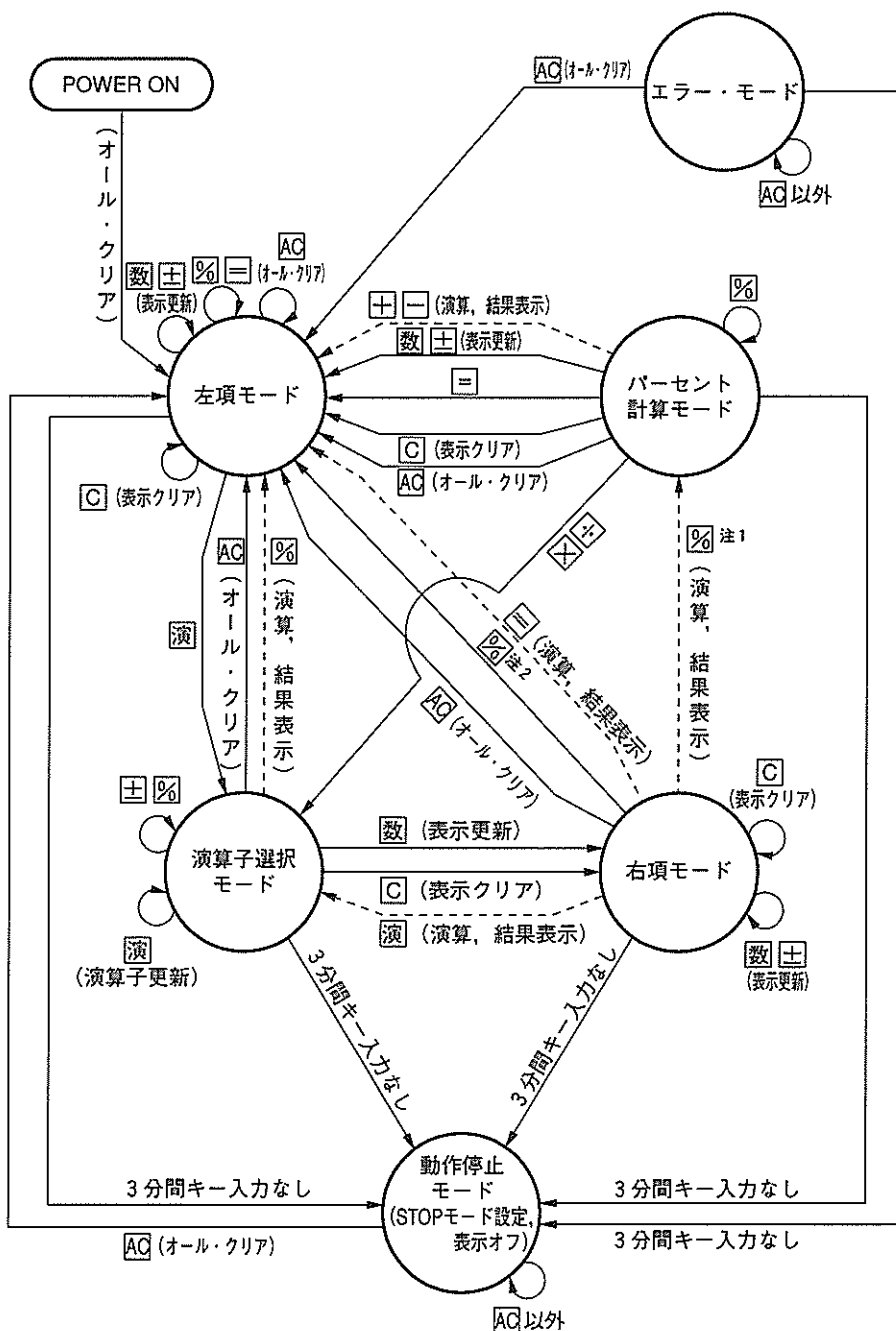
除算で除数が0の場合、および演算結果にオーバーフローが発生した場合に、エラー表示を行います。

(6) 動作停止モード

3分間キー入力がない場合、STOPモードを設定します。

図3-1に、状態遷移図を示します。

図3-1 状態遷移図



- 注1. 演算子として  $\times$  または  $\div$  が入力されている場合の処理です。  
 2. 演算子として  $+$  または  $-$  が入力されている場合の処理です。

備考1. 数: 0 - 9, .  
 演: +, -,  $\times$ ,  $\div$

2. ( ) : キー入力による処理内容を表します。

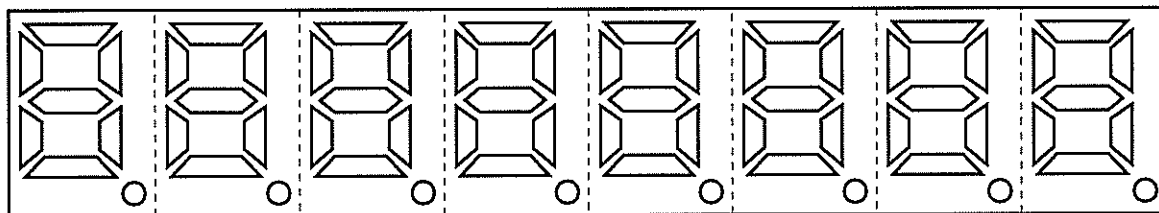
————— : 通常の状態遷移を表します。

----- : 演算結果にエラーが発生した場合は、エラー・モードに遷移します。

### 3.4 表 示

入力した数値、および演算結果の表示には、図3-2に示すように8セグメント×8桁のLCDを使用します。

図3-2 表 示



### 3.5 オートパワーオフ

この電卓は、キー入力が3分間行われなかったとき、自動的に電源オフになります。この場合は、**AC**キーを押すと、再び電源オンになります。

### 3.6 演 算 例

次に、このアプリケーション・ノートで紹介する電卓仕様による演算について、キー入力の手順および表示例を示します。

(1) 加算

$$123 + 4560.7 = 4683.7$$

キー入力	表 示
1 2 3	1 2 3.
+	1 2 3.
4 5 6 0 . 7	4 5 6 0.7
=	4 6 8 3.7

$$-852+147 = -705$$

キー入力	表示
8 5 2 ±	- 8 5 2.
+	- 8 5 2.
1 4 7	1 4 7.
=	- 7 0 5.

$$611+0 = 611$$

キー入力	表示
6 1 1	6 1 1.
+	6 1 1.
=	6 1 1.

(2) 減算

$$951-627.84 = 323.16$$

キー入力	表示
9 5 1	9 5 1.
-	9 5 1.
6 2 7 . 8 4	6 2 7.8 4
=	3 2 3.1 6

$$-753.94-186 = -939.94$$

キー入力	表示
7 5 3 . 9 4 ±	- 7 5 3.9 4
-	- 7 5 3.9 4
1 8 6	1 8 6.
=	- 9 3 9.9 4

$$-2098 - 0 = -2098$$

キー入力	表示
2 0 9 8 ±	- 2 0 9 8.
-	- 2 0 9 8.
=	- 2 0 9 8.

(3) 乗算

$$279 \times (-0.58) = -161.82$$

キー入力	表示
2 7 9	2 7 9.
×	2 7 9.
. 5 8 ±	- 0.5 8
=	- 1 6 1.8 2

$$35.704 \times 22.194 = 792.414576$$

キー入力	表示
3 5 . 7 0 4	3 5.7 0 4
×	3 5.7 0 4
2 2 . 1 9 4	2 2.1 9 4
=	7 9 2.4 1 4 5

備考 表示桁未満は切り捨てて表示されます。

$$519 \times 519 = 269361$$

キー入力	表示
5 1 9	5 1 9.
×	5 1 9.
=	2 6 9 3 6 1.

(4) 除算

$$-769.3 \div (-245) = 3.14$$

キー入力	表 示
<input type="text" value="7"/> <input type="text" value="6"/> <input type="text" value="9"/> <input type="text" value="."/> <input type="text" value="3"/> <input type="text" value="±"/>	<input type="text" value="- 7 6 9.3"/>
<input type="text" value="÷"/>	<input type="text" value="- 7 6 9.3"/>
<input type="text" value="2"/> <input type="text" value="4"/> <input type="text" value="5"/> <input type="text" value="±"/>	<input type="text" value="- 2 4 5."/>
<input type="text" value="="/>	<input type="text" value="3.1 4"/>

$$412 \div (-618) = -0.666666\dots$$

キー入力	表 示
<input type="text" value="4"/> <input type="text" value="1"/> <input type="text" value="2"/>	<input type="text" value="4 1 2."/>
<input type="text" value="÷"/>	<input type="text" value="4 1 2."/>
<input type="text" value="6"/> <input type="text" value="1"/> <input type="text" value="8"/> <input type="text" value="±"/>	<input type="text" value="- 6 1 8."/>
<input type="text" value="="/>	<input type="text" value="- 0.6 6 6 6 6 6"/>

備考 表示桁未満は切り捨てて表示されます。

$$-9.98 \div (-9.98) = 1$$

キー入力	表 示
<input type="text" value="9"/> <input type="text" value="."/> <input type="text" value="9"/> <input type="text" value="8"/> <input type="text" value="±"/>	<input type="text" value="- 9.9 8"/>
<input type="text" value="÷"/>	<input type="text" value="- 9.9 8."/>
<input type="text" value="="/>	<input type="text" value="1."/>



(5) 混合計算

$$\{(45+38) \times 0.75 - 25\} \div (-50) = -0.745$$

キー入力	表示
4 5	4 5.
+	4 5.
3 8	3 8.
×	8 3.
· 7 5	0.7 5
-	6 2.2 5
2 5	2 5.
÷	3 7.2 5
5 0 ±	- 5 0.
=	- 0.7 4 5

備考 四則の混合計算は、乗除算が優先されず、すべて入力順に計算されます。

(6) パーセント計算

$$645 \times 32.9 \% = 212.205$$

キー入力	表示
6 4 5	6 4 5.
×	6 4 5.
3 2 . 9	3 2.9
%	2 1 2.2 0 5

(7) 割り増し計算

$$400 + (400 \times 60\%) = 640$$

キー入力	表 示
4 0 0	4 0 0.
×	4 0 0.
6 0	6 0.
%	2 4 0.
+	6 4 0.

(8) 割り引き計算

$$32.5 - (32.5 \times 70\%) = 9.75$$

キー入力	表 示
3 2 . 5	3 2.5
×	3 2.5
7 0	7 0.
%	2 2.7 5
-	9.7 5

(9) エラー発生例

① 除数が0の除算を行ったとき

$$500 \div 0$$

キー入力	表 示
5 0 0	5 0 0.
÷	5 0 0.
0	0.
=	E.

② 計算結果 $\geq 1.0 \times 10^7$ , または計算結果 $\leq -1.0 \times 10^7$ のとき

$$4500 \times 4500 = 2.025 \times 10^7$$

キー入力	表 示
4 5 0 0	4 5 0 0 .
×	4 5 0 0 .
=	E .

(10) アンダフロー発生例

$-1.0 \times 10^{-6} < \text{計算結果} < 1.0 \times 10^{-6}$ のとき

$$0.0063 \div (-9000) = -7 \times 10^{-7}$$

キー入力	表 示
. 0 0 6 3	0 . 0 0 6 3
÷	0 . 0 0 6 3
9 0 0 0 ±	- 9 0 0 0 .
=	0 .

## (11) その他

キー入力	表 示
<input type="text" value="3"/> <input type="text" value="8"/>	<input type="text" value="3 8."/>
<input style="width: 20px; height: 20px; vertical-align: middle;" type="text" value="%"/>	<input type="text" value="3 8."/>
<input style="width: 20px; height: 20px; vertical-align: middle;" type="text" value="×"/>	<input type="text" value="3 8."/>
<input type="text" value="5"/> <input type="text" value="7"/>	<input type="text" value="5 7."/>
<input style="width: 20px; height: 20px; vertical-align: middle;" type="text" value="="/>	<input type="text" value="2 1 6 6."/>
<input type="text" value="1"/> <input type="text" value="9"/>	<input type="text" value="1 9."/>
<input style="width: 20px; height: 20px; vertical-align: middle;" type="text" value="="/>	<input type="text" value="1 9."/>
<input style="width: 20px; height: 20px; vertical-align: middle;" type="text" value="−"/>	<input type="text" value="1 9."/>
<input style="width: 20px; height: 20px; vertical-align: middle;" type="text" value="%"/>	<input type="text" value="1 9."/>
<input style="width: 20px; height: 20px; vertical-align: middle;" type="text" value="+"/>	<input type="text" value="1 9."/>
<input type="text" value="4"/> <input type="text" value="8"/>	<input type="text" value="4 8."/>
<input style="width: 20px; height: 20px; vertical-align: middle;" type="text" value="="/>	<input type="text" value="6 7."/>

上記の例の場合、はじめに [38×57] が、次に [19+48] が実行されます。

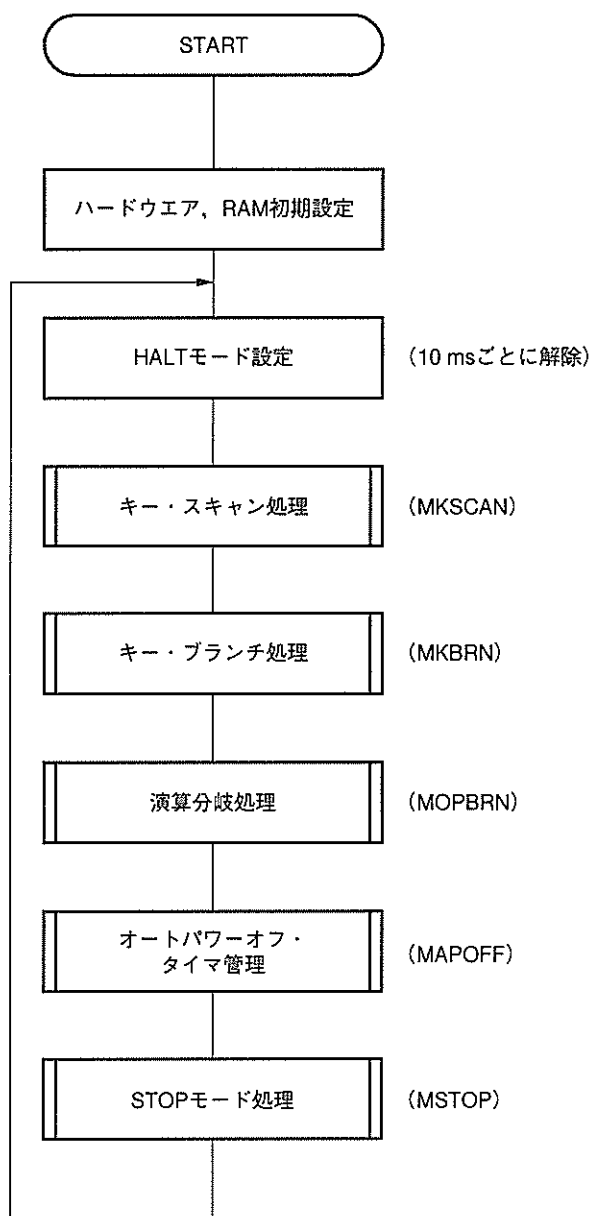


## 第4章 システム制御部概要

次に、このアプリケーション・ノートで紹介する電卓のシステム制御部の主な処理について、概要説明、および概略フロー・チャートを示します。

### 4.1 メイン処理

この電卓プログラムがスタートすると、ハードウェア、RAMの初期設定を行ったのち、メイン処理に入ります。この電卓では、消費電流の低減のため、10 msごとにメイン処理を行い、メイン処理が1回終了するごとにHALTモードを設定します。



## 4.2 キー・スキャン処理

POB<sub>0</sub>-POB<sub>3</sub>, POC<sub>0</sub>からキー・スキャン信号を出力し、POA<sub>0</sub>-POA<sub>3</sub>にキー状態を入力します。10ms×3回のチャタリング処理を行い、チャタリング終了時にキー・コードを生成します。このとき、有効なキー入力の場合はキー処理要求フラグをセットします。

表4-1に、入力キーとキー・コードの対応を、表4-2に、有効なキー入力を示します。

表4-1 入力キーとキー・コードの対応

上位桁 / 下位桁		RKCODH (0.30H) の値					
		7H	BH	DH	EH	0H	FH
RKCODL (0.31H) の値	7H	7	8	9	C	AC	—
	BH	4	5	6	×	÷	—
	DH	3	2	1	+	—	—
	EH	0	.	±	=	%	—
	FH	—	—	—	—	—	キー・オフ <sup>注</sup>
	0H	—	—	—	—	—	多重押し

注 キーが何も押されていない状態です。

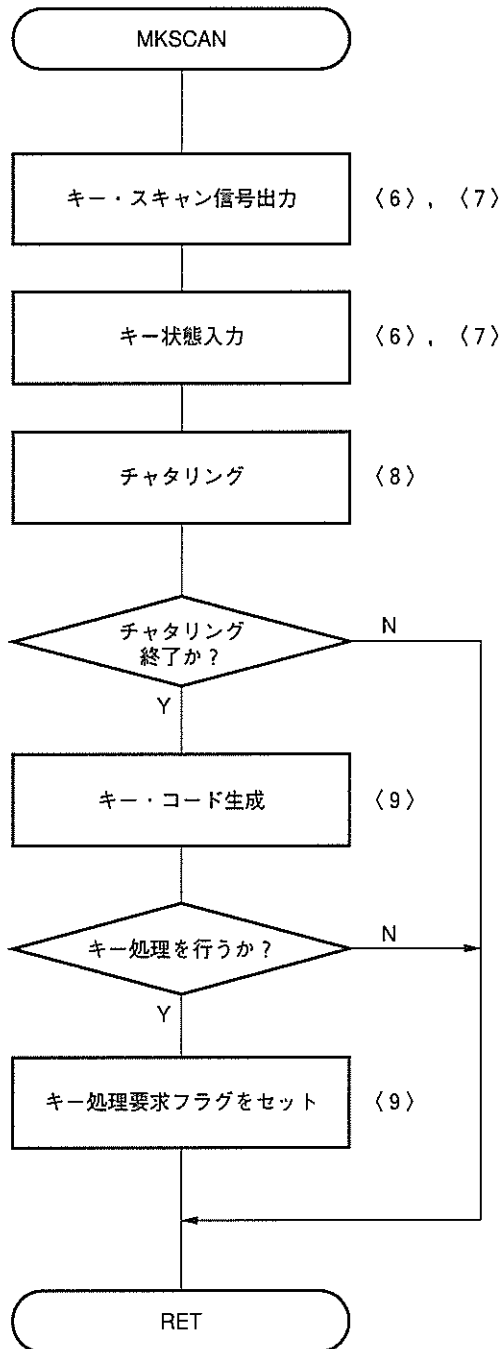
表4-2 有効なキー入力

		前回のキー・コード		
		キー・オフ <sup>注</sup>	1個押し	多重押し
今回のキー・コード	キー・オフ <sup>注</sup>	—	○	○
	1個押し	○	×	×
	多重押し	×	×	—

注 キーが何も押されていない状態です。

備考 ○の場合にキー入力が有効となります。

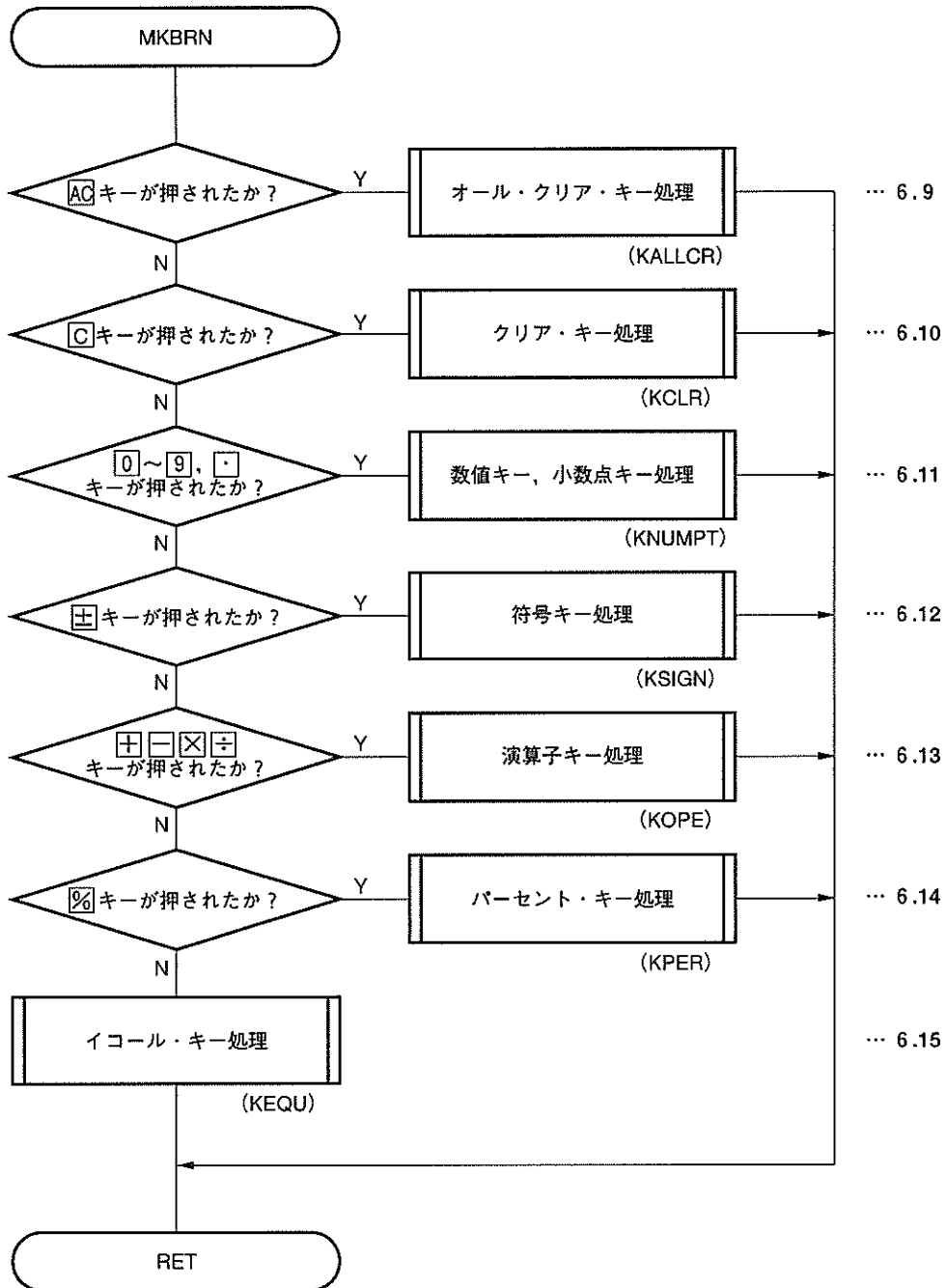
—は、この電卓では存在しないケースを示します。





### 4.3 キー・ブランチ処理

キー・コードをデコードし、各キー処理を呼び出します。



## 4.4 演算分岐処理, エラー処理

加減乗除の各演算処理を呼び出し、計算結果を表示します。除数が0の除算を行ったとき、およびオーバーフロー発生時（計算結果 $\geq 1.0 \times 10^7$ 、または計算結果 $\leq -1.0 \times 10^7$ のとき）には、エラー表示を行います。また、アンダフロー発生時（ $-1.0 \times 10^{-6} < \text{計算結果} < 1.0 \times 10^{-6}$ のとき）は、計算結果として0を表示します。

どの演算処理を呼び出すかの判定には、演算子データを使用します。演算子データは、演算子キーの入力により変化します。演算子データを格納するエリアとして、ROPE（BANK0の60H）、およびRCOM（BANK0の61H）を用意しています。

表4-3に、演算子データの変化の具体例を示します。

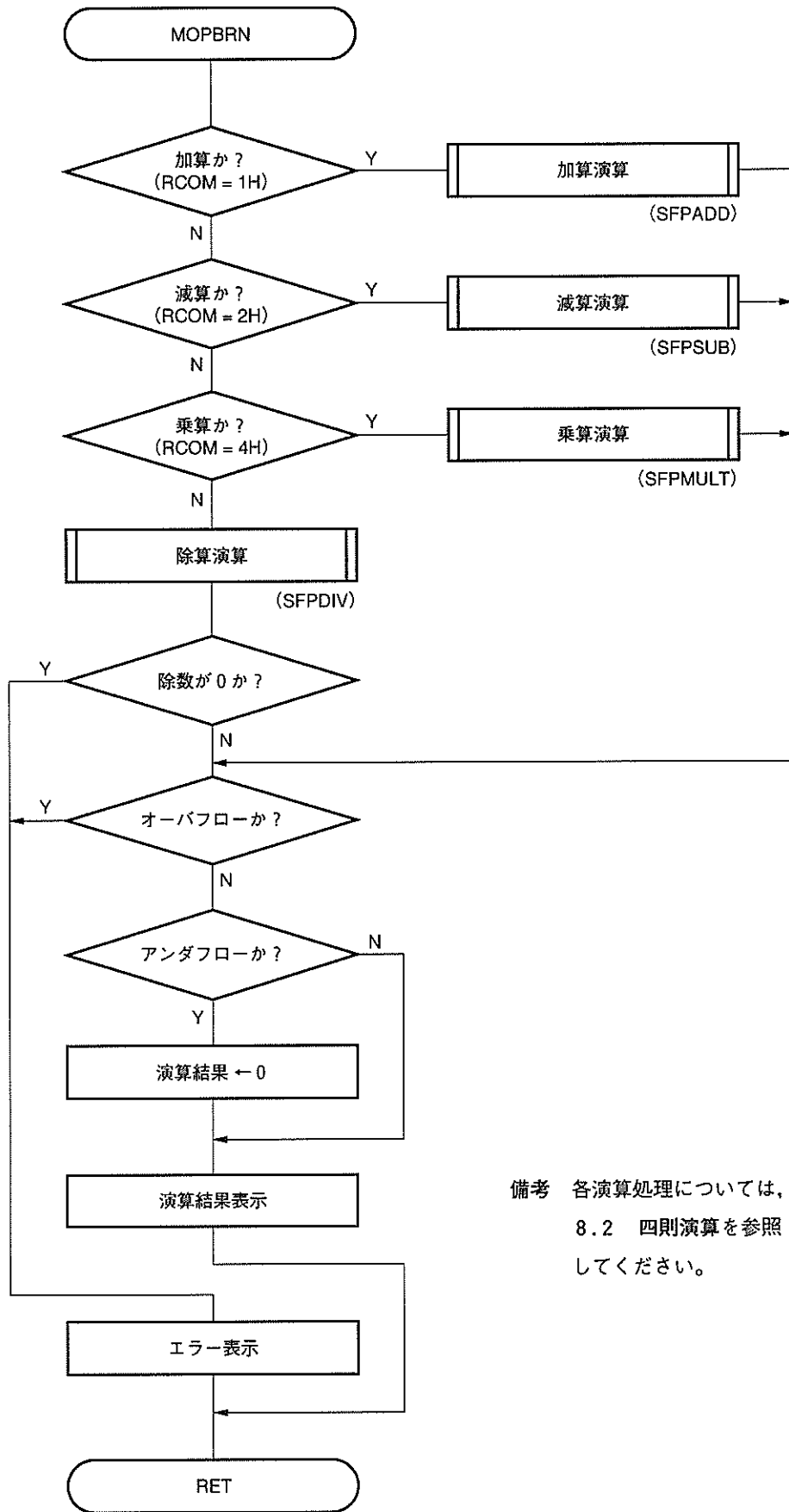
表4-3 演算子データの変化 (1/2)

入 力 キ ー	ROPE (0.60H) の値	RCOM (0.61H) の値
<div style="border: 1px solid black; padding: 2px; display: inline-block;">AC</div>	0H	0H
RAMのオール・クリア処理により、ROPE, RCOMに0Hが入ります。		
<div style="display: inline-block; margin-right: 10px;"> <div style="border: 1px solid black; padding: 2px;">3</div> <div style="border: 1px solid black; padding: 2px;">0</div> <div style="border: 1px solid black; padding: 2px;">+</div> </div> <div style="border: 1px solid black; padding: 2px; display: inline-block;">-</div>	1H  2H	0H  0H
1つ目の数値に続く演算子キーの入力により、ROPEに演算子データが格納されます。演算子キーが続けて押されたときは、ROPEの値が更新されます。RCOMの値はそのままです。 演算子キーと演算子データの対応は、次のとおりです。  <div style="display: flex; flex-direction: column; align-items: center;"> <div style="margin-bottom: 10px;"><div style="border: 1px solid black; padding: 2px;">+</div> キー：1H</div> <div style="margin-bottom: 10px;"><div style="border: 1px solid black; padding: 2px;">-</div> キー：2H</div> <div style="margin-bottom: 10px;"><div style="border: 1px solid black; padding: 2px;">×</div> キー：4H</div> <div style="margin-bottom: 10px;"><div style="border: 1px solid black; padding: 2px;">÷</div> キー：8H</div> </div>		
<div style="display: inline-block; margin-right: 10px;"> <div style="border: 1px solid black; padding: 2px;">2</div> <div style="border: 1px solid black; padding: 2px;">0</div> <div style="border: 1px solid black; padding: 2px;">=</div> </div>	2H	2H
<div style="display: inline-block; margin-right: 10px;"> <div style="border: 1px solid black; padding: 2px;">2</div> <div style="border: 1px solid black; padding: 2px;">0</div> <div style="border: 1px solid black; padding: 2px;">×</div> </div>	4H	2H
2つ目の数値に続くイコール・キーの入力により、ROPEの演算子データがRCOMに転送されます。演算分岐処理では、RCOMの値により演算子を識別し、各演算処理を呼び出し、結果を表示します。 また、2つ目の数値に続いて演算子キーが入力されたときは、ROPEに新たな演算子データが格納されます。		

表4-3 演算子データの変化 (2/2)

入 力 キ ー	ROPE (0.60H) の値	RCOM (0.61H) の値
AC	0H	0H
5 0 ×	4H	0H
7 0 % +	4H 1H	4H 1H

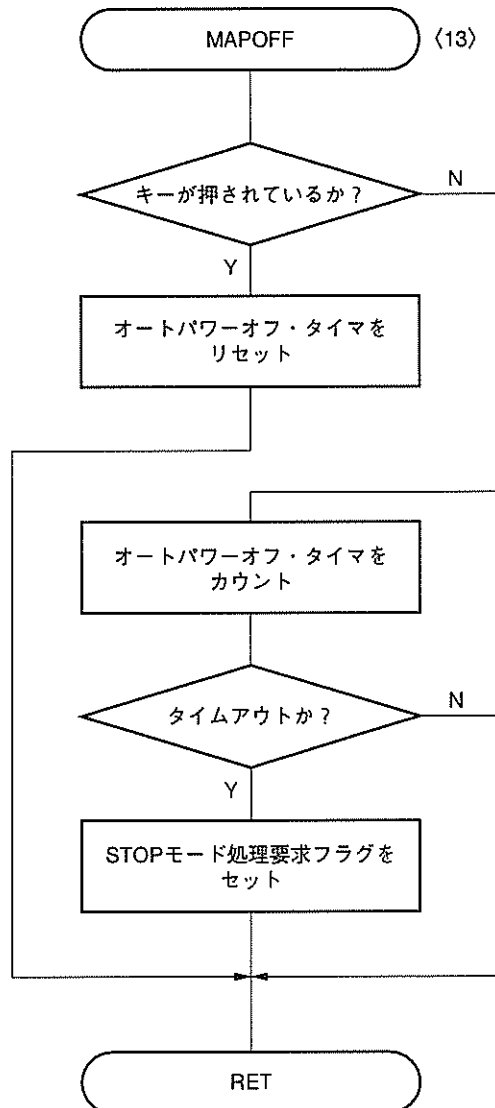
パーセント計算のときは、パーセント・キーが押された時点でROPEの値を判定し、4Hまたは8Hの場合（演算子キーとして× または ÷ が押された場合）のみ、ROPEの演算子データがRCOMに転送され、パーセント計算が実行されます。また、パーセント計算に続いて+ または - が押されたときは、演算子データがROPEとRCOMに格納され、それぞれ割り増し計算、割り引き計算が実行されます。



備考 各演算処理については、  
8.2 四則演算を参照  
してください。

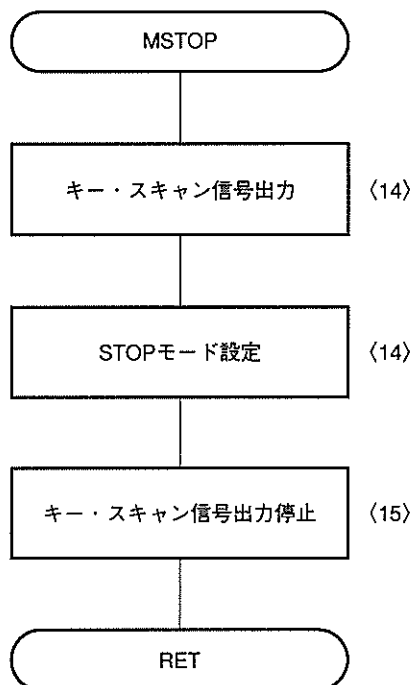
## 4.5 オートパワーオフ・タイマ管理

キーが押されているときはオートパワーオフ・タイマをリセットします。キー・オフのときはオートパワーオフ・タイマをカウントし、タイムアウトの場合はSTOPモード処理要求フラグをセットします。



## 4.6 STOPモード処理

P0B0-P0C0のキー・スキャン信号をすべて出力状態にし、STOPモードを設定します。キー入力によりSTOPモードが解除されたのち、キー・スキャン信号の出力を停止します。



## 第 5 章 RAMレイアウトおよび変数説明

### 5.1 RAMレイアウト

図 5-1 にこのアプリケーション・ノートで紹介するプログラムのRAMレイアウトを示します。



図5-1 RAMレイアウト (1/2)

[BANK0]		アドレス															F
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	3210	RREG0	RREG1	RREG2	RREG3	RREG4	RREG5	RREG6	RREG7	RREG8	RREG9	RREG10	RREG11	DBF3	DBF2	DBF1	DBF0
1	3210	RNUMC	RPTLOC	RSINLOC	RDSIGN	RDEXP				RZLSD							RDMSD
2	3210				RZLSD												RZMSD
3	3210	RKCODH	RKCODL	RCHCODH	RCHCODL	RIPCODH	RIPCODL	RCHATC		RSTOP3	RSTOP2	RSTOP1	RSTOP0				
4	3210	LCDD0	LCDD1	LCDD2	LCDD3	LCDD4	LCDD5	LCDD6	LCDD7	LCDD8	LCDD9	LCDD10	LCDD11	LCDD12	LCDD13	LCDD14	LCDD15
5	3210	LCDD16	LCDD17	LCDD18	LCDD19	LCDD20	LCDD21	LCDD22	LCDD23	LCDD24	LCDD25	LCDD26	LCDD27	LCDD28	LCDD29	LCDD30	LCDD31
6	3210	ROPE	RCOM			RMODE				RSYSFLG	REVEFLG						
7	3210	P0A	P0B	P0C	P0D	AR3	AR2	AR1	AR0	WR	BANK	IX/AMPH	IX/MPL	IXL	IRPH	RPL	PSW CMP CY Z IXE
												MPE				BCD	



## 5.2 RAMおよびフラグ説明

RAMおよび フラグ名	RAMアドレス	名 称	説 明
RREG0   RREG11	0.00H   0.0BH	レジスタ0   レジスタ11	ワーク・レジスタとして使用します。
RNUMC	0.10H	数値キー・カウンタ	数値キーの押された回数をカウントします。
RPTLOC	0.11H	小数点表示位置 エリア	小数点の表示位置を格納します。
RSINLOC	0.12H	符号表示位置エリア	符号の表示位置を格納します。
REGD		表示データ・ レジスタ	
RDSIGN	0.13H	演算結果符号エリア	演算結果の正負をチェックします。  ビット0が0のとき：正 ビット0が1のとき：負
RDEXP	0.14H	表示データ指数 エリア	表示データの正規化、および演算結果を表示データに変換する ときに使用します。
RDLSD   RDMSD	0.18H   0.1FH	表示データ・エリア	表示データを格納します。
RZLSD   RZMSD	0.23H   0.2FH	退避用レジスタ (REGZ)	データ退避エリアです。パーセント計算のときに、被演算データ を退避します。
RKCODH   RKCODL	0.30H   0.31H	キー・コード・ エリア	チャタリング処理の終了したキー・コードを格納します。
RCHCODH   RCHCODL	0.32H   0.33H	チャタリング・ コード・エリア	チャタリング処理中のキー・コードを格納します。
RIPCODH   RIPCODL	0.34H   0.35H	インプット・ コード・エリア	現在押されているキーのキー・コードを格納します。
RCHATC	0.36H	チャタリング・ カウンタ	チャタリング処理の回数をカウントします。
RSTOP3   RSTOP2   RSTOP1   RSTOP0	0.38H   0.39H   0.3AH   0.3BH	オートパワーオフ・ タイマ	オートパワーオフ用の3分間タイマです。
LCDD0   LCDD31	0.40H   0.5FH	LCDセグメント・ データ・レジスタ	LCDに表示するセグメント・データを格納します。

RAMおよび フラグ名	RAMアドレス	名 称	説 明
ROPE	0.60H	演算子エリア	最後に押された演算子キーのデータを格納します。  1H：加算（ $\boxed{+}$ キー）  2H：減算（ $\boxed{-}$ キー）  4H：乗算（ $\boxed{\times}$ キー）  8H：除算（ $\boxed{\div}$ キー）
RCOM	0.61H	実行演算子エリア	実行対象となる演算子を格納します。
RMODE	0.64H	モード・エリア	現在のモードを格納します。  1H：右項モード  2H：演算子選択モード  4H：左項モード  8H：エラー・モード
RSYSFLG	0.68H	システム・フラグ・ エリア	システム制御部で使用するフラグです。
FPER	0.68H.0	パーセント・フラグ	パーセント・キーが押されたときにセットします。
FOPEND	0.68H.1	演算終了フラグ	演算終了時にセットします。
FFALSE	0.68H.2	不正入力フラグ	パーセント・キー、またはイコール・キーが不正な位置で押されたときにセットします。
FSTOP	0.68H.3	動作停止モード・ フラグ	STOPモードの設定時にセットします。
REVEFLG	0.69H	イベント・フラグ・ エリア	各種の処理要求フラグです。
FKEYREQ	0.69H.0	キー処理要求フラグ	有効なキー入力（キー・オフを含む）の確定時にセットします。
FMULTI	0.69H.1	多重押しフラグ	キー・スキャン処理で、多重押しの検出時にセットします。
FOPREQ	0.69H.2	演算要求フラグ	演算可能なキーが押された時にセットします。
FSTPREQ	0.69H.3	STOPモード 処理要求フラグ	オートパワーオフ・タイマのタイム・アウト時にセットします。
REGX		浮動小数点 レジスタ1	演算用レジスタです。被演算データ、および演算結果を格納します。
RXSIGN	1.03H		REGXの符号部です。
RXEXP	1.04H 1.05H		REGXの指数部です。
RXLSD   RXMSD	1.06H   1.0FH		REGXの仮数部です。

RAMおよび フラグ名	RAMアドレス	名 称	説 明
REGY		浮動小数点 レジスタ 2	演算用レジスタです。演算データを格納します。
RYSIGN	1.13H		REGYの符号部です。
RYEXP	1.14H 1.15H		REGYの指数部です。
RYLSD   RYMSD	1.16H   1.1FH		REGYの仮数部です。
REGW		浮動小数点 レジスタ 3	演算用のワーク・レジスタです。
RWSIGN	1.23H		REGWの符号部です。
RWEXP	1.24H 1.25H		REGWの指数部です。
RWLSD   RWMSD	1.26H   1.2FH		REGWの仮数部です。
ROPFLG	1.30H	演算フラグ	浮動小数点演算で使用するフラグです。
FEXCHG	1.30H.0	レジスタ交換フラグ	加減算時に、浮動小数点レジスタ1と浮動小数点レジスタ2の内容を交換したときにセットします。
FZERO	1.30H.1	演算結果ゼロ・ フラグ	演算結果が0のときにセットします。
FDVERR	1.30H.2	除数ゼロ・エラー・ フラグ	除算で、除数が0のときにセットします。
FOVER	1.30H.3	オーバフロー・ フラグ	演算によりオーバフローが発生したときにセットします。

## 5.3 処理およびRAMの名称説明

このアプリケーション・ノートで紹介する電卓のプログラム記述では、作業の効率を良くするため、処理名称、レーベル名称、RAM名称を、先頭アルファベットにより識別できるように付けました。

次に、処理およびRAMの名称の意味を、先頭アルファベット別に示します。

### (1) 処理名称

M\*\*\*\*：メイン・ルーチン

S\*\*\*\*：サブルーチン

I\*\*\*\*：割り込みルーチン

### (2) レーベル名称

L\*\*\*\*：メイン・ルーチン内の分岐先レーベル

J\*\*\*\*：サブルーチン内の分岐先レーベル

H\*\*\*\*：割り込みルーチン内の分岐先レーベル

T\*\*\*\*：テーブルのレーベル

### (3) RAM名称

R\*\*\*\*：ニブル（4ビット）単位で使用しているRAMエリア名称

F\*\*\*\*：フラグ名称



## 第6章 システム制御部フロー・チャート

次に、このアプリケーション・ノートで紹介する電卓のシステム制御部のフロー・チャートを示します。



## 6.1 初期設定

入力変数	フロー・チャート〈1〉	処理, 備考	出力変数
	<pre> graph TD     IRESET([IRESET]) --&gt; DI[DI]     DI --&gt; SP[SP ← 5H]     SP --&gt; PCC[PCC ← 3H]     PCC --&gt; P1A["P1A ← 0000B P0B ← 1111B P0C ← 0001B P0D ← 0000B"]     P1A --&gt; PM1["PM1 ← 1110B PM0 ← 1111B"]     PM1 --&gt; TMCT[TMCT ← 0110B]     TMCT --&gt; TMM[TMM ← 9CH]     TMM --&gt; WTC[WTC ← 0000B]     WTC --&gt; Note{&lt;2&gt; 注 1}             </pre> <p>注 フロー・チャート〈2〉の①へ飛びます。</p>	<p>割り込み禁止</p> <p>スタック・ポインタ設定</p> <p>システム・クロック設定 (メイン・クロック)</p> <p>出力ポート初期設定</p> <p>ポート入出力設定 出力: P0B P0C P0D P1A 入力: P0A</p> <p>8ビット・タイマ動作クロック設定 (システム・クロック/256)</p> <p>8ビット・タイマ・モジュロ・レジスタ設定</p> <p>時計用タイマ・セレクト設定</p>	<p>P1A (1.70H) P0B (0.71H) P0C (0.72H) P0D (0.73H)</p>

入力変数	フロー・チャート〈2〉	処理, 備考	出力変数
	<pre> graph TD     Start((1)) --&gt; Step1[LCDM ← 0001B LCDC ← 0001B]     Step1 --&gt; Step2[ADM ← 0000B]     Step2 --&gt; Step3[PSL ← 0000B SIC ← 0000B]     Step3 --&gt; Step4[IPF ← 0000B]     Step4 --&gt; Step5[RPH ← 0000B RPL ← 0000B]     Step5 --&gt; Step6[SRAMCR (45)]     Step6 --&gt; End{ (3) 1 }                     </pre>	<p>LCDコントローラ/ドライバ設定 表示モード : 1/2 デューティフレーム周 波数 : メイン・クロック/2<sup>13</sup></p> <p>A/Dコンバータ設定 (スタンバイ 状態)</p> <p>シリアル・インタフェース設定 (未使用)</p> <p>割り込み許可フラグ設定 (すべて 禁止)</p> <p>汎用レジスタ設定 (BANK0の00H -0FH)</p> <p>RAMクリア</p>	<p>RPH RPL</p>

入力変数	フロー・チャート〈3〉	処理, 備考	出力変数
	<pre> graph TD     Start((1)) --&gt; Step1[RSTOP0 ← FH RSTOP1 ← 4H RSTOP2 ← 6H RSTOP3 ← 4H]     Step1 --&gt; Step2[RDEXP ← 8H]     Step2 --&gt; Step3[RKCODL ← FH RKCODH ← FH RCHCODL ← FH RCHCODH ← FH]     Step3 --&gt; Step4[RMODE ← 4H]     Step4 --&gt; Step5[SDPINI &lt;47&gt;]     Step5 --&gt; Step6[SDISP &lt;61&gt;]     Step6 --&gt; Step7[LCDEN ← 1]     Step7 --&gt; Step8[TMEN ← 1]     Step8 --&gt; Step9([MMAIN])                     </pre>	<p>オートパワーオフ・タイマをリセット</p> <p>表示データ指数エリアをリセット</p> <p>キー・コード、チャタリング・コードをリセット</p> <p>モードを左項モードに設定</p> <p>表示データ・エリアをイニシャライズ</p> <p>表示データをLCDセグメント・データに変換</p> <p>LCD表示オン</p> <p>8ビット・タイマのカウントを開始</p>	<p>RSTOP0 RSTOP1 RSTOP2 RSTOP3 (0.38H-0.3BH)</p> <p>RDEXP (0.14H)</p> <p>RKCODL RKCODH RCHCODL RCHCODH (0.30H-0.31H) (0.32H-0.33H)</p> <p>RMODE (0.64H)</p>

## 6.2 メイン処理

入力変数	フロー・チャート〈4〉	処理, 備考	出力変数
<p>FKEYREQ (0.69H.0)</p> <p>RKCODL (0.31H)</p> <p>RKCODH (0.30H)</p> <p>FSTOP (0.68H.3)</p>	<pre> graph TD     MMAIN([MMAIN]) --&gt; IRQTM[IRQTM ← 0]     IRQTM --&gt; HALT2H[HALT 2H]     HALT2H --&gt; IRQTM1{IRQTM = 1?}     IRQTM1 -- Y --&gt; WDTRES[WDTRES ← 1]     WDTRES --&gt; MKSCAN[MKSCAN]     MKSCAN --&gt; FKEYREQ1{FKEYREQ = 1?}     FKEYREQ1 -- Y --&gt; FKEYREQ0[FKEYREQ ← 0]     FKEYREQ1 -- N --&gt; C5_1[&lt;5&gt; 1]     FKEYREQ0 --&gt; RKCODL{RKCODL = FH?}     RKCODL -- Y --&gt; RKCODH{RKCODH = FH?}     RKCODL -- N --&gt; C5_1     RKCODH -- Y --&gt; FSTOP{FSTOP = 1?}     RKCODH -- N --&gt; C5_1     FSTOP -- Y --&gt; C5_3[&lt;5&gt; 3]     FSTOP -- N --&gt; C5_2[&lt;5&gt; 2]     </pre>	<p>8ビット・タイマの割り込み要求フラグをクリア</p> <p>HALTモード設定 (10 msごとに解除)</p> <p>ウォッチドッグ・タイマをリセット</p> <p>キー・スキャン処理</p> <p>有効なキー入力 (キー・オフを含む) が確定したか?</p> <p>キー処理要求フラグをクリア</p> <p>キー・オフか?</p> <p>動作停止モードか?</p>	

入力変数	フロー・チャート〈5〉	処理, 備考	出力変数
<p>FOPREQ (0.69H.2)</p> <p>FSTPREQ (0.69H.3)</p>	<pre> graph TD     Start((1)) --&gt; MKBRN[MKBRN&lt;br/&gt;(16)]     MKBRN --&gt; FOPREQ{FOPREQ = 1?}     FOPREQ -- Y --&gt; MOPBRN[MOPBRN&lt;br/&gt;(41)]     FOPREQ -- N --&gt; Conn2((2))     Conn2 --&gt; MOPBRN     MOPBRN --&gt; MAPOFF[MAPOFF&lt;br/&gt;(13)]     MAPOFF --&gt; FSTPREQ{FSTPREQ = 1?}     FSTPREQ -- Y --&gt; MSTOP[MSTOP&lt;br/&gt;(14)]     FSTPREQ -- N --&gt; Conn3((3))     MSTOP --&gt; MMAIN([MMAIN])     Conn3 --&gt; MSTOP     </pre>	<p>キー・ブランチ処理</p> <p>演算可能か?</p> <p>演算分岐処理</p> <p>オートパワーオフ・タイマ管理</p> <p>タイムアウトか?</p> <p>STOPモード処理</p>	

### 6.3 キー・スキャン処理

入力変数	フロー・チャート〈6〉	処理, 備考	出力変数
<p>FMULTI (0.69H.1)</p>	<pre> graph TD     Start([MKSCAN]) --&gt; R1[RIPCODL ← FH RIPCODH ← FH]     R1 --&gt; R2[RREG0 ← 7H]     R2 --&gt; R3[RREG2 ← 0H]     R3 --&gt; R4[P0B ← RREG0]     R4 --&gt; R5[NOP 3回]     R5 --&gt; R6[RREG9 ← P0A]     R6 --&gt; R7[P0B ← FH]     R7 --&gt; R8[SKRET]     R8 --&gt; D{FMULTI = 1?}     D -- Y --&gt; T1[&lt;8&gt; 1]     D -- N --&gt; T2[&lt;7&gt; 1]     </pre>	<p>インプット・コードをリセット</p> <p>キー・スキャン信号の初期値を RREG0 にセット</p> <p>キー数カウンタをリセット</p> <p>P0B のキー・スキャン信号出力</p> <p>12 <math>\mu</math>s ウェイト</p> <p>キー・リターン値読み出し</p> <p>P0B のキー・スキャン信号出力停止</p> <p>キー・リターン値判定</p> <p>多重押しか?</p>	

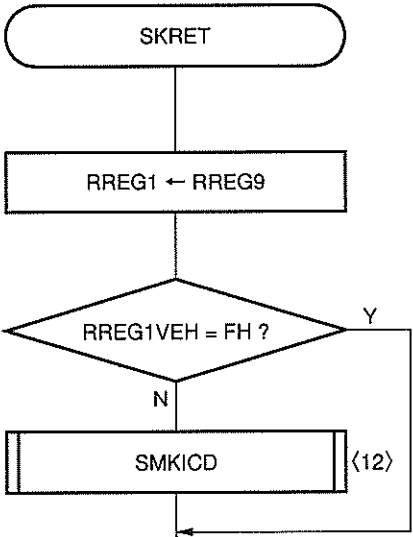
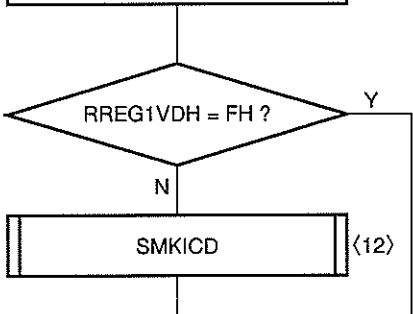
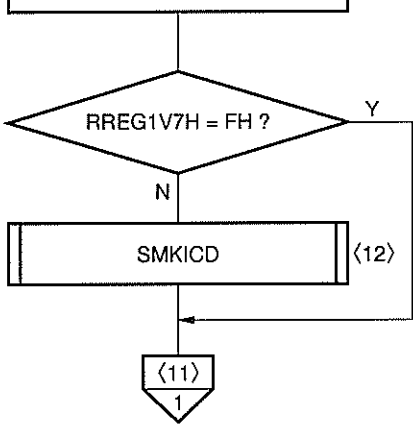
入力変数	フロー・チャート〈7〉	処理, 備考	出力変数
<p>FMULTI (0.69H.1)</p>	<pre> graph TD     Start((1)) --&gt; CY1[CY ← 1]     CY1 --&gt; Shift[RREG0とCYを 1ビット右シフト]     Shift --&gt; CY0{CY = 0?}     CY0 -- N --&gt; Conn6[&lt;6&gt; 1]     CY0 -- Y --&gt; RREG0[RREG0 ← 0H P0C ← RREG0]     RREG0 --&gt; NOP3[NOP 3回]     NOP3 --&gt; RREG9[RREG9 ← P0A]     RREG9 --&gt; P0C1[P0C ← 1H]     P0C1 --&gt; SKRET[SKRET]     SKRET --- Conn10[&lt;10&gt;]     SKRET --&gt; FMULTI{FMULTI = 1?}     FMULTI -- Y --&gt; Conn8_1[&lt;8&gt; 1]     FMULTI -- N --&gt; Conn8_2[&lt;8&gt; 2]     </pre>	<p>P0Bのキー・スキャン信号出力が終了したか?</p> <p>P0Cのキー・スキャン信号出力</p> <p>12 μsウエイト</p> <p>キー・リターン値読み出し</p> <p>P0Cのキー・スキャン信号出力停止</p> <p>キー・リターン値判定</p> <p>多重押しか?</p>	

入力変数	フロー・チャート<8>	処理, 備考	出力変数
	<pre> graph TD     Start((1)) --&gt; Init[FMULTI ← 0]     Init --&gt; Set[RIPCODL ← 0H RIPCODH ← FH]     Set --&gt; Join((2))     Join --&gt; KeyH{RCHCODH = RIPCODH?}     KeyH -- N --&gt; KeyL{RCHCODL = RIPCODL?}     KeyH -- Y --&gt; KeyL     KeyL -- Y --&gt; Update[RCHCODH ← RIPCODH RCHCODL ← RIPCODL]     KeyL -- N --&gt; Update     Update --&gt; Reset[RCHATC ← 2H]     Reset --&gt; DoneH{RCHATC = 0H?}     DoneH -- Y --&gt; Join     DoneH -- N --&gt; Dec[RCHATC ← RCHATC - 1H]     Dec --&gt; DoneL{RCHATC = 0H?}     DoneL -- Y --&gt; Exit1{&lt;9&gt; 1}     DoneL -- N --&gt; Join     </pre>	<p>多重押しフラグをクリア</p> <p>多重押しフラグをクリア</p> <p>入力・コードに多重押しのコードをセット</p> <p>キーの状態が変化しなかったか?</p> <p>チャタリング・コードを更新</p> <p>チャタリング・カウンタをリセット</p> <p>チャタリング処理は終了済みか?</p> <p>チャタリング・カウンタをカウント</p> <p>チャタリング処理が終了したか?</p>	<p>FMULTI (0.69H.1)</p>



入力変数	フロー・チャート〈9〉	処理, 備考	出力変数
	<pre> graph TD     1((1)) --&gt; D1{RCHCODL = 0H?}     D1 -- Y --&gt; D2{RCHCODH = FH?}     D1 -- N --&gt; J1(( ))     D2 -- Y --&gt; D3{RCHCODL = FH?}     D2 -- N --&gt; J1     D3 -- Y --&gt; D4{RCHCODH = FH?}     D3 -- N --&gt; J1     D4 -- Y --&gt; D5{RKCODL = FH?}     D4 -- N --&gt; J1     D5 -- Y --&gt; D6{RKCODH = FH?}     D5 -- N --&gt; J1     D6 -- Y --&gt; P1[FKEYREQ ← 1]     D6 -- N --&gt; J1     P1 --&gt; P2[RKCODH ← RCHCODH RKCODL ← RCHCODL]     P2 --&gt; D4     P2 --&gt; 2((2))     2 --&gt; P2     P2 --&gt; RET([RET])     J1 --&gt; RET     </pre>	<p>多重押しか?</p> <p>キー・オフか?</p> <p>前のキー・コードがキー・オフか?</p> <p>キー処理要求フラグをセット</p> <p>キー・コードにチャタリング・コードをセット</p>	<p>FKEYREQ (0.69H.0)</p> <p>RKCODH (0.30H) RKCODL (0.31H)</p>

### 6.4 キー・リターン値判定処理

入力変数	フロー・チャート <10>	処理, 備考	出力変数
RREG9 (0.09H)	 <pre> graph TD     SKRET([SKRET]) --&gt; RREG1[RREG1 ← RREG9]     RREG1 --&gt; RREG1VEH{RREG1VEH = FH?}     RREG1VEH -- Y --&gt; J1(( ))     RREG1VEH -- N --&gt; SMKICD1[SMKICD &lt;12&gt;]     J1 --&gt; SMKICD1     </pre>	<p>P0A<sub>0</sub>のキー・リターン信号がハイか?</p> <p>インプット・コード生成</p>	
RREG9 (0.09H)	 <pre> graph TD     RREG1[RREG1 ← RREG9] --&gt; RREG1VDH{RREG1VDH = FH?}     RREG1VDH -- Y --&gt; J2(( ))     RREG1VDH -- N --&gt; SMKICD2[SMKICD &lt;12&gt;]     J2 --&gt; SMKICD2     </pre>	<p>P0A<sub>1</sub>のキー・リターン信号がハイか?</p> <p>インプット・コード生成</p>	
RREG9 (0.09H)	 <pre> graph TD     RREG1[RREG1 ← RREG9] --&gt; RREG1V7H{RREG1V7H = FH?}     RREG1V7H -- Y --&gt; J3(( ))     RREG1V7H -- N --&gt; SMKICD3[SMKICD &lt;12&gt;]     J3 --&gt; SMKICD3     SMKICD3 --&gt; Exit[&lt;11&gt; 1]     </pre>	<p>P0A<sub>2</sub>のキー・リターン信号がハイか?</p> <p>インプット・コード生成</p>	

入力変数	フロー・チャート <11>	処理, 備考	出力変数
<p>RREG9 (0.09H)</p>	<pre> graph TD     Start((1)) --&gt; RREG1[RREG1 ← RREG9]     RREG1 --&gt; FH{RREG1V7H = FH?}     FH -- Y --&gt; SMKICD[SMKICD &lt;12&gt;]     FH -- N --&gt; RREG2{RREG2 ≥ 2H?}     RREG2 -- Y --&gt; FMULTI[FMULTI ← 1]     SMKICD --&gt; RET([RET])     FMULTI --&gt; RET     </pre>	<p>POA<sub>3</sub>のキー・リターン信号がハイか?</p> <p>インプット・コード生成</p> <p>多重押しか?</p> <p>多重押しフラグをセット</p>	<p>FMULTI (0.69H.1)</p>

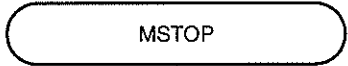
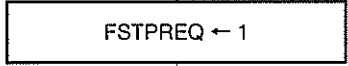

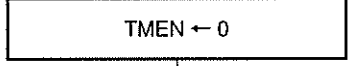
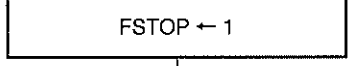
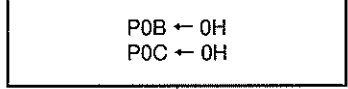
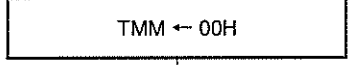
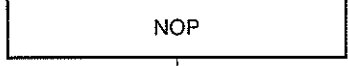
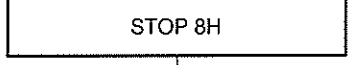

### 6.5 インプット・コード生成処理

入力変数	フロー・チャート〈12〉	処理, 備考	出力変数
RREG0 (0.00H) RREG9 (0.09H)	<pre>                     graph TD                         Start([SMKICD]) --&gt; Process1[RIPCODH ← RREG0 RIPCODL ← RREG9]                         Process1 --&gt; Process2[RREG2 ← RREG2 + 1H]                         Process2 --&gt; End([RET])                     </pre>	インプット・コード生成    キー数をカウント	RIPCODH RIPCODL (0.34H-0.35H)   RREG2 (0.02H)

### 6.6 オートパワーオフ・タイマ管理

入力変数	フロー・チャート <13>	処理、備考	出力変数
<p>RKCODL (0.31H)</p> <p>RKCODH (0.30H)</p>	<pre> graph TD     Start([MAPOFF]) --&gt; D1{RKCODL = FH?}     D1 -- Y --&gt; P1[RSTOP0 ← RSTOP0 - 1H RSTOP1 ← RSTOP1 - CY RSTOP2 ← RSTOP2 - CY RSTOP3 ← RSTOP3 - CY]     D1 -- N --&gt; D2{RKCODH = FH?}     D2 -- Y --&gt; P1     D2 -- N --&gt; P2[RSTOP0 ← FH RSTOP1 ← 4H RSTOP2 ← 6H RSTOP3 ← 4H]     P1 --&gt; D3{CY = 1?}     D3 -- Y --&gt; P3[FSTPREQ ← 1]     D3 -- N --&gt; P2     P3 --&gt; P2     P2 --&gt; End([RET])     </pre>	<p>現在キー・オフか？</p> <p>オートパワーオフ・タイマをカウント</p> <p>タイムアウトか？</p> <p>STOPモード処理要求フラグをセット</p> <p>オートパワーオフ・タイマをリセット</p>	

## 6.7 STOPモード処理

入力変数	フロー・チャート〈14〉	処理, 備考	出力変数
			
		STOPモード処理要求フラグをクリア	FSTPREQ (0.69H.3)
		LCD表示オフ	
		8ビット・タイマの動作停止	
		動作停止モード・フラグをセット	FSTOP (0.68H.3)
		キー・スキャン信号出力	
		8ビット・タイマ・モジュロ・レジスタにSTOPモード解除時のウェイト値を設定	
			
		STOPモード設定 (キー入力で解除)	
			

入力変数	フロー・チャート <15>	処理, 備考	出力変数
	<pre> graph TD     Start((1)) --&gt; Step1[TMEN ← 0]     Step1 --&gt; Step2["P0B ← FH P0C ← 1H"]     Step2 --&gt; Step3[TMM ← 9CH]     Step3 --&gt; Step4[TMEN ← 1]     Step4 --&gt; Step5([RET])                     </pre>	<p>8ビット・タイマの動作停止</p> <p>キー・スキャン信号出力停止</p> <p>8ビット・タイマ・モジュロ・レジスタを再設定</p> <p>8ビット・タイマの動作開始</p>	

### 6.8 キー・ブランチ処理

入力変数	フロー・チャート <16>	処理, 備考	出力変数
<p>RKCODH (0.30H)</p> <p>RKCODL (0.31H)</p> <p>FSTOP (0.68H.3)</p> <p>RMODE (0.64H)</p> <p>RKCODH (0.30H)</p> <p>RKCODL (0.31H)</p>	<pre> graph TD     Start([MKBRN]) --&gt; D1{RKCODH = 0H?}     D1 -- N --&gt; C18_5["&lt;18&gt; 5"]     D1 -- Y --&gt; D2{RKCODL = 7H?}     D2 -- Y --&gt; C18_1["&lt;18&gt; 1"]     D2 -- N --&gt; D3{FSTOP = 1?}     D3 -- Y --&gt; C19_3["&lt;19&gt; 3"]     D3 -- N --&gt; D4{RMODE = 8H?}     D4 -- Y --&gt; C19_3     D4 -- N --&gt; D5{RKCODH = 0H?}     D5 -- Y --&gt; C17_1["&lt;17&gt; 1"]     D5 -- N --&gt; D6{RKCODL = EH?}     D6 -- Y --&gt; C19_1["&lt;19&gt; 1"]     D6 -- N --&gt; C18_5     </pre>	<p>AC キーが押されたか?</p> <p>動作停止モードか?</p> <p>エラー・モードか?</p> <p>Ⓜ キーが押されたか?</p>	



入力変数	フロー・チャート<17>	処理, 備考	出力変数
<p>RKCODH (0.30H)</p> <p>RKCODL (0.31H)</p> <p>RKCODL (0.31H)</p> <p>RKCODH (0.30H)</p> <p>RKCODL (0.31H)</p>	<pre> graph TD     Start((1)) --&gt; D1{RKCODH = EH?}     D1 -- Y --&gt; D2{RKCODL = EH?}     D1 -- N --&gt; D3{RKCODH = DH?}     D2 -- Y --&gt; T1[&lt;19&gt; 2]     D2 -- N --&gt; D4{RKCODL = 7H?}     D4 -- Y --&gt; T2[&lt;18&gt; 2]     D4 -- N --&gt; T3[&lt;18&gt; 5]     D3 -- Y --&gt; D5{RKCODL = EH?}     D3 -- N --&gt; T4[&lt;18&gt; 3]     D5 -- Y --&gt; T5[&lt;18&gt; 4]     D5 -- N --&gt; T4     </pre>	<p>☐ キーが押されたか?</p> <p>☐ キーが押されたか?</p> <p>☐ キーが押されたか?</p>	

入力変数	フロー・チャート〈18〉	処理, 備考	出力変数
	<pre> graph TD     1((1)) --&gt; KALLCR[KALLCR &lt;20&gt;]     KALLCR --&gt; 2((2))     2 --&gt; KCLR[KCLR &lt;21&gt;]     KCLR --&gt; 3((3))     3 --&gt; KNUMPT[KNUMPT &lt;23&gt;]     KNUMPT --&gt; 4((4))     4 --&gt; KSIGN[KSIGN &lt;30&gt;]     KSIGN --&gt; 5((5))     5 --&gt; KOPE[KOPE &lt;32&gt;]     KOPE --&gt; 19{&lt;19&gt; 3}     KCLR --&gt; KALLCR     KNUMPT --&gt; KALLCR     KSIGN --&gt; KALLCR     </pre>	オール・クリア・キー処理	
		クリア・キー処理	
		数値キー, 小数点キー処理	
		符号キー処理	
		演算子キー処理	

入力変数	フロー・チャート<19>	処理, 備考	出力変数
	<pre> graph TD     1((1)) --&gt; KPER[KPER &lt;37&gt;]     KPER --&gt; 2((2))     2 --&gt; KEQU[KEQU &lt;39&gt;]     KEQU --&gt; RET([RET])     3((3)) --&gt; 2     KPER --&gt; 3             </pre>	<p>パーセント・キー処理</p> <p>イコール・キー処理</p>	

## 6.9 オール・クリア・キー処理

入力変数	フロー・チャート<20>	処理, 備考	出力変数
	<pre> graph TD     KALLCR([KALLCR]) --&gt; LCDEN0[LCDEN ← 0]     LCDEN0 --&gt; SRAMCR[SRAMCR &lt;45&gt;]     SRAMCR --&gt; RDEXP[RDEXP ← 8H]     RDEXP --&gt; RKCODL[RKCODL ← 7H RKCODH ← 0H RCHCODL ← 7H RKCODH ← 0H]     RKCODL --&gt; RMODE[RMODE ← 4H]     RMODE --&gt; SDPINI[SDPINI &lt;47&gt;]     SDPINI --&gt; SDISP[SDISP &lt;61&gt;]     SDISP --&gt; LCDEN1[LCDEN ← 1]     LCDEN1 --&gt; RET([RET])                     </pre>	<p>LCD表示オフ</p> <p>RAMクリア</p> <p>表示データ指数エリアをリセット</p> <p>キー・コード, チャタリング・コードに <b>AC</b> キーのコードを復帰</p> <p>モードを左項モードにセット</p> <p>表示データ・エリアをイニシャライズ</p> <p>表示データをLCDセグメント・データに変換</p> <p>LCD表示オン</p>	<p>RDEXP (0.14H)</p> <p>RMODE (0.64H)</p>

### 6.10 クリア・キー処理

入力変数	フロー・チャート〈21〉	処理, 備考	出力変数
<p>FPER (0.68H.0)</p>	<pre> graph TD     KCLR([KCLR]) --&gt; FPER{FPER = 1?}     FPER -- Y --&gt; FPER0[FPER ← 0]     FPER0 --&gt; RMODE4[RMODE ← 4H]     RMODE4 --&gt; FOPEND[FOPEND ← 0 FFALSE ← 0]     FOPEND --&gt; RNUMC[RNUMC ← 0H]     RNUMC --&gt; RDEXP[RDEXP ← 8H]     RDEXP --&gt; RMODE2{RMODE = 2H?}     RMODE2 -- Y --&gt; RMODE1[RMODE ← 1H]     RMODE2 -- N --&gt; RMODE1     RMODE1 --&gt; Exit1[1]     FPER -- N --&gt; RMODE4     </pre>	<p>☒キーが押されたか？</p> <p>パーセント・フラグをクリア</p> <p>モードを左項モードに設定</p> <p>演算終了フラグ, 不正入力フラグをクリア</p> <p>数値キー・カウンタをリセット</p> <p>表示データ指数エリアをリセット</p> <p>演算子選択モードか？</p> <p>モードを右項モードにセット</p>	<p>FPER (0.68H.0)</p> <p>RMODE (0.64H)</p> <p>FOPEND (0.68H.1) FFALSE (0.68H.2)</p> <p>RNUMC (0.10H)</p> <p>RDEXP (0.14H)</p>
<p>RMODE (0.64H)</p>	<p>FPER (0.68H.0)</p> <p>RMODE (0.64H)</p> <p>FOPEND (0.68H.1) FFALSE (0.68H.2)</p> <p>RNUMC (0.10H)</p> <p>RDEXP (0.14H)</p>	<p>モードを右項モードにセット</p>	<p>RMODE (0.64H)</p>

入力変数	フロー・チャート〈22〉	処理, 備考	出力変数
	<pre> graph TD     Start((1)) --&gt; LCDEN0[ LCDEN ← 0 ]     LCDEN0 --&gt; SDPINI[ SDPINI (47) ]     SDPINI --&gt; SDISP[ SDISP (61) ]     SDISP --&gt; LCDEN1[ LCDEN ← 1 ]     LCDEN1 --&gt; RET([ RET ])                     </pre>	<p>LCD表示オフ</p> <p>表示データ・エリアをイニシャライズ</p> <p>表示データをLCDセグメント・データに変換</p> <p>LCD表示オン</p>	

### 6.11 数値キー，小数点キー処理

入力変数	フロー・チャート <23>	処理，備考	出力変数
FFALSE (0.68H.2)		☒ キーが ☐ キーが不正に押されたか？  不正入力フラグをクリア  数値キー・カウンタをリセット  表示データ指数エリアをリセット	FFALSE (0.68H.2)  RNUMC (0.10H)  RDEXP (0.14H)
FPER (0.68H.0)		☒ キーが押されているか？  モードを左項モードに設定	RMODE (0.64H)
FPER (0.68H.0) FOPEND (0.68H.1)		パーセント・フラグ，演算終了フラグをクリア	FPER (0.68H.0) FOPEND (0.68H.1)
RKCODH (0.30H)		} ☐ キーが押されたか？	
RKCODL (0.31H)			

入力変数	フロー・チャート〈24〉	処理, 備考	出力変数
<p>RDEXP (0.14H)</p> <p>RNUMC (0.10H)</p> <p>RNUMC (0.10H)</p> <p>RMODE (0.64H)</p>	<pre> graph TD     Start((1)) --&gt; D1{RDEXP = 8H?}     D1 -- Y --&gt; D2{RNUMC = 0H?}     D1 -- N --&gt; J1(( ))     D2 -- Y --&gt; P1[RNUMC ← RNUMC + 1H]     P1 --&gt; P2[LCDEN ← 0]     P2 --&gt; P3[SDPINI &lt;47&gt;]     P3 --&gt; P4[SDISP &lt;61&gt;]     P4 --&gt; P5[LCDEN ← 1]     D2 -- N --&gt; J1     J1 --&gt; P6[RDEXP ← RNUMC]     P6 --&gt; D3{RMODE = 2H?}     D3 -- Y --&gt; P7[RMODE ← 1H]     D3 -- N --&gt; J2(( ))     P7 --&gt; J2     J1 --&gt; J2     J2 --&gt; End{&lt;29&gt; 3}     </pre>	<p>小数点キーがはじめて押されたか？</p> <p>数値キーが押されないうちに小数点キーが押されたか？</p> <p>数値キー・カウンタをカウント</p> <p>LCD表示オフ</p> <p>表示データ・エリアをイニシャライズ</p> <p>表示データをLCDセグメント・データに変換</p> <p>LCD表示オン</p> <p>表示データ指数エリアに数値キー・カウンタの値をセット</p> <p>演算子選択モードか？</p> <p>モードを右項モードに設定</p>	<p>RNUMC (0.10H)</p> <p>RDEXP (0.14H)</p> <p>RMODE (0.64H)</p>



入力変数	フロー・チャート<25>	処理, 備考	出力変数
RNUMC (0.10H)	<pre> graph TD     1((1)) --&gt; D1{RNUMC = 7H?}     D1 --&gt; C29[&lt;29&gt; 3]     D1 --&gt; D2{RKCODL = EH?}     D2 -- N --&gt; D3{RKCODL = DH?}     D2 -- Y --&gt; P1[RREG9 ← 0H]     P1 --&gt; C28[&lt;28&gt; 1]     C28 --&gt; D3     D3 -- N --&gt; C26_2[&lt;26&gt; 2]     D3 -- Y --&gt; D4{RKCODH = 7H?}     D4 -- N --&gt; D5{RKCODH = BH?}     D4 -- Y --&gt; P2[RREG9 ← 1H]     P2 --&gt; D5     D5 -- N --&gt; C26_1[&lt;26&gt; 1]     D5 -- Y --&gt; P3[RREG9 ← 2H]     P3 --&gt; C26_1             </pre>	<p>数値キーがすでに7回押されたか？</p>	
RKCODL (0.31H)		<p>[0] キーが押されたか？</p> <p>“0”の表示データをRREG9にセット</p>	
RKCODL (0.31H)		<p>[1] キーが押されたか？</p>	
RKCODH (0.30H)		<p>“1”の表示データをRREG9にセット</p>	
RKCODH (0.30H)		<p>[2] キーが押されたか？</p> <p>“2”の表示データをRREG9にセット</p>	

入力変数	フロー・チャート <26>	処理, 備考	出力変数
RKCODH (0.30H)		[3] キーが押されたか?  “3”の表示データをRREG9に セット	
RKCODL (0.31H)		[4] キーが押されたか?  (27)	
RKCODH (0.30H)		“4”の表示データをRREG9に セット	
RKCODH (0.30H)		[5] キーが押されたか?  “5”の表示データをRREG9に セット	
RKCODH (0.30H)		[6] キーが押されたか?  “6”の表示データをRREG9に セット	

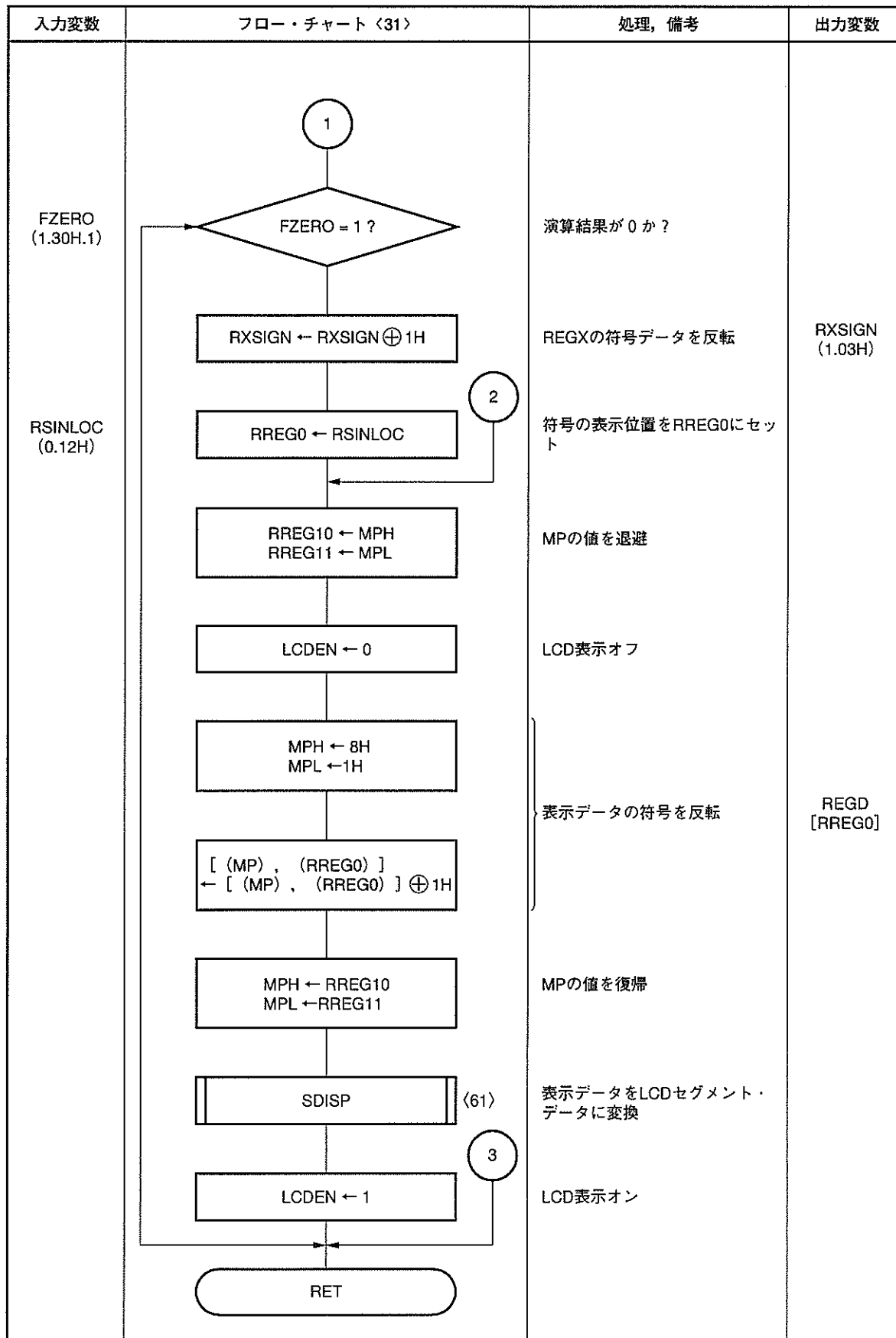
入力変数	フロー・チャート <27>	処理, 備考	出力変数
RKCODH (0.30H)	<pre>                     graph TD                         1((1)) --&gt; D1{RKCODH = 7H?}                         D1 -- Y --&gt; P1[RREG9 ← 7H]                         D1 -- N --&gt; D2{RKCODH = 8H?}                         P1 --&gt; D2                         D2 -- Y --&gt; P2[RREG9 ← 8H]                         D2 -- N --&gt; D3{RKCODH = 9H?}                         P2 --&gt; D3                         P3[RREG9 ← 9H] --&gt; E1[&lt;28&gt;]                         E1 --&gt; 1                     </pre>	[7] キーが押されたか？  “7” の表示データをRREG9に セット	
RKCODH (0.30H)		[8] キーが押されたか？  “8” の表示データをRREG9に セット	
RKCODH (0.30H)		[9] キーが押されたか？  “9” の表示データをRREG9に セット	

入力変数	フロー・チャート <28>	処理, 備考	出力変数
<p>RMODE (0.64H)</p> <p>RNUMC (0.10H)</p>	<pre> graph TD     Start((1)) --&gt; LCDEN[LCDEN ← 0]     LCDEN --&gt; RMODE4H{RMODE = 4H?}     RMODE4H -- Y --&gt; RMODE4H     RMODE4H -- N --&gt; RMODE1H[RMODE ← 1H]     RMODE1H --&gt; RMODE4H     RMODE4H --&gt; RNUMC0H{RNUMC = 0H?}     RNUMC0H -- Y --&gt; SDPINI[SDPINI &lt;47&gt;]     SDPINI --&gt; RREG90H{RREG9 = 0H?}     RREG90H -- Y --&gt; C2{&lt;29&gt; 2}     RREG90H -- N --&gt; RDLSD[AH ← RDLSD]     RDLSD --&gt; C1{&lt;29&gt; 1}     </pre>	<p>LCD表示オフ</p> <p>左項モードか？</p> <p>モードを右項モードに設定</p> <p>1回目の数値キー入力か？</p> <p>表示データ・エリアをイニシャライズ</p> <p>0キーが押されたか？</p> <p>表示データ・エリアの最下位桁にスペースのデータをセット</p>	<p>RMODE (0.64H)</p> <p>RDLSD (0.18H)</p>

入力変数	フロー・チャート <29>	処理, 備考	出力変数
<p>RDEXP (0.14H)</p>	<pre> graph TD     Start((1)) --&gt; Decision{RDEXP = 8H?}     Decision -- Y --&gt; DecStep[RPTLOC ← RPTLOC - 1H]     Decision -- N --&gt; NumStep[RNUMC ← RNUMC + 1H]     DecStep --&gt; NumStep     NumStep --&gt; Shift[SUSHFD &lt;50&gt;]     Shift --&gt; SetLSD[RDLS ← RREG9]     SetLSD --&gt; Disp[SDISP &lt;61&gt;]     Disp --&gt; LCD[LCDEN ← 1]     LCD --&gt; End([RET])     </pre>	<p>小数点キーが入力されているか?</p> <p>小数点表示位置を-1</p> <p>数値キー・カウンタをカウント</p> <p>表示データを1バイト左シフト</p> <p>新たな表示データを表示データ・エリアの最下位桁にセット</p> <p>表示データをLCDセグメント・データに変換</p> <p>LCD表示オン</p>	<p>RPTLOC (0.11H)</p> <p>RNUMC (0.10H)</p> <p>RDLS (0.18H)</p>

### 6.12 符号キー処理

入力変数	フロー・チャート <30>	処理, 備考	出力変数
<p>RMODE (0.64H)</p> <p>FPER (0.68H.0)</p> <p>FOPEND (0.68H.1)</p> <p>RNUMC (0.10H)</p>	<pre> graph TD     KSIGN([KSIGN]) --&gt; D1{RMODE = 2H?}     D1 -- Y --&gt; D2{RNUMC = 0H?}     D1 -- N --&gt; D3{FPER = 1?}     D3 -- Y --&gt; P1[FPER ← 0]     P1 --&gt; P2[FOPEND ← 1]     P2 --&gt; P3[RMODE ← 4H]     P3 --&gt; D1     D3 -- N --&gt; D2     D2 -- Y --&gt; C1{&lt;31&gt; 3}     D2 -- N --&gt; P4[RREG0 ← RNUMC + 8H RSINLOC ← RREG0]     P4 --&gt; C2{&lt;31&gt; 2}     </pre>	<p>演算子選択モードか？</p> <p><input checked="" type="checkbox"/> キーが押されているか？</p> <p>パーセント・フラグをクリア</p> <p>演算終了フラグをセット</p> <p>モードを左項モードにセット</p> <p>演算が終了済みか？</p> <p>数値キーが何も押されていないか？</p> <p>符号の表示位置を求め、RREG0とRSINLOCにセット</p>	<p>FPER (0.68H.0)</p> <p>FOPEND (0.68H.1)</p> <p>RMODE (0.64H)</p> <p>RSINLOC (0.12H)</p>



### 6.13 演算子キー処理

入力変数	フロー・チャート <32>	処理, 備考	出力変数
RKCODL (0.31H)	<pre>                     graph TD                         KOPE([KOPE]) --&gt; D1{RKCODL = DH?}                         D1 -- Y --&gt; D2{RKCODH = EH?}                         D1 -- N --&gt; D3{RKCODH = 0H?}                         D2 -- Y --&gt; P1[RREG9 ← 1H]                         D2 -- N --&gt; D3                         P1 --&gt; D3                         D3 -- Y --&gt; P2[RREG9 ← 2H]                         D3 -- N --&gt; D4{RKCODH = EH?}                     </pre>	<input type="checkbox"/> キーが押されたか？	
RKCODH (0.30H)	<pre>                     graph TD                         D2{RKCODH = EH?}                         D2 -- Y --&gt; P1[RREG9 ← 1H]                         D2 -- N --&gt; D3{RKCODH = 0H?}                     </pre>	加算の演算子データをRREG9に セット	
RKCODH (0.30H)	<pre>                     graph TD                         D3{RKCODH = 0H?}                         D3 -- Y --&gt; P2[RREG9 ← 2H]                         D3 -- N --&gt; D4{RKCODH = EH?}                     </pre>	<input type="checkbox"/> キーが押されたか？  減算の演算子データをRREG9に セット	
RKCODH (0.30H)	<pre>                     graph TD                         D4{RKCODH = EH?}                         D4 -- Y --&gt; P3[RREG9 ← 4H]                         D4 -- N --&gt; D5{RKCODH = 0H?}                     </pre>	<input checked="" type="checkbox"/> キーが押されたか？  乗算の演算子データをRREG9に セット	
RKCODH (0.30H)	<pre>                     graph TD                         D5{RKCODH = 0H?}                         D5 -- Y --&gt; P4[RREG9 ← 8H]                         D5 -- N --&gt; C1{&lt;33&gt;}                         C1 --&gt; P4                     </pre>	<input type="checkbox"/> キーが押されたか？  除算の演算子データをRREG9に セット	



入力変数	フロー・チャート <33>	処理, 備考	出力変数
RMODE (0.64H)			
FFALSE (0.68H.2)		不正入力フラグをクリア	FFALSE (0.68H.2)
RMODE (0.64H)		演算子選択モードか?	
FOPEND (0.68H.1)		演算が終了済みか?	
FOPEND (0.68H.1)		演算終了フラグをクリア	FOPEND (0.68H.1)
FPER (0.68H.0)		Ⓜキーが押されているか?	
FPER (0.68H.0)		パーセント・フラグをクリア	FPER (0.68H.0)
		ⓇかⓈキーが押されたか?	

入力変数	フロー・チャート <34>	処理, 備考	出力変数
<p>RMODE (0.64H)</p>	<pre> graph TD     Start((1)) --&gt; STRNDY[STRNDY &lt;57&gt;]     STRNDY --&gt; RMODE4H{RMODE = 4H?}     RMODE4H -- N --&gt; Exit1{&lt;35&gt; 1}     RMODE4H -- Y --&gt; RREG0[RREG0 ← 9H&lt;br&gt;RREG1 ← 8H]     RREG0 --&gt; STRAN[STRAN &lt;52&gt;]     STRAN --&gt; SRYCLR[SRYCLR &lt;48&gt;]     SRYCLR --&gt; ROPE[ROPE ← RREG9]     ROPE --&gt; RMODE2H[RMODE ← 2H]     RMODE2H --&gt; Exit2{&lt;36&gt; 3}     </pre>	<p>表示データを正規化してREGY へ転送</p> <p>左項モードか?</p> <p>REGYのデータをREGXへ転送</p> <p>REGYをクリア</p> <p>演算子エリアにRREG9の演算子 データをセット</p> <p>モードを演算子選択モードに設 定</p>	<p>ROPE (0.60H)</p> <p>RMODE (0.64H)</p>

入力変数	フロー・チャート〈35〉	処理、備考	出力変数
	<pre> graph TD     1((1)) --&gt; P1[RCOM ← ROPE ROPE ← RREG9]     P1 --&gt; P2[RMODE ← 2H]     P2 --&gt; D1{36 2}                     </pre>	<p>実行演算子エリアに演算子エリアのデータをセットし、演算子エリアにRREG9の演算子データをセット</p> <p>モードを演算子選択モードに設定</p>	<p>RCOM (0.61H) ROPE (0.60H)</p> <p>RMODE (0.64H)</p>
	<pre> graph TD     2((2)) --&gt; P3[ROPE ← RREG9 RCOM ← RREG9]     P3 --&gt; P4[RREG0 ← 8H RREG1 ← 9H]     P4 --&gt; S1[STRAN]     S1 --- L1[52]     L1 --- P5[RREG0 ← 2H RREG1 ← 8H]     P5 --&gt; S2[STRAN]     S2 --- L2[52]     L2 --&gt; D2{36 1}                     </pre>	<p>演算子エリアと実行演算子エリアにRREG9の演算子データをセット</p> <p>REGXのデータをREGYへ転送</p> <p>REGZのデータをREGXへ復帰</p>	<p>ROPE (0.60H) RCOM (0.61H)</p>

入力変数	フロー・チャート <36>	処理, 備考	出力変数
	<pre> graph TD     1((1)) --&gt; RMODE[RMODE ← 4H]     RMODE --&gt; FOPEND[FOPEND ← 1]     FOPEND --&gt; FOPREQ[FOPREQ ← 1]     FOPREQ --&gt; RET([RET])     2((2)) --&gt; FOPEND     3((3)) --&gt; FOPREQ                     </pre>	<p>モードを左項モードに設定</p> <p>演算終了フラグをセット</p> <p>演算要求フラグをセット</p>	<p>RMODE (0.64H)</p> <p>FOPEND (0.68H.1)</p> <p>FOPREQ (0.69H.2)</p>

### 6.14 パーセント・キー処理

入力変数	フロー・チャート <37>	処理, 備考	出力変数
FPER (0.68H.0)	<pre>                     graph TD                         KPER([KPER]) --&gt; D1{FPER = 1?}                         D1 -- Y --&gt; D2{RMODE = 2H?}                         D1 -- N --&gt; D2                         D2 -- Y --&gt; D3{RMODE = 1H?}                         D2 -- N --&gt; D3                         D3 -- Y --&gt; P1[FOPEND ← 0]                         D3 -- N --&gt; D4{ROPE ≥ 4H?}                         P1 --&gt; D4                         D4 -- Y --&gt; P2[RMODE ← 4H]                         D4 -- N --&gt; P3[FFALSE ← 1]                         P2 --&gt; D4                         P3 --&gt; D4                         D4 --&gt; Exit1{&lt;38&gt; 1}                     </pre>	☒ キーがすでに押されているか？	
RMODE (0.64H)		演算子選択モードか？	
RMODE (0.64H)		右項モードか？	
		演算終了フラグをクリア	FOPEND (0.68H.1)
ROPE (0.60H)		☒ か ☒ キーが押されているか？	
		モードを左項モードにセット	RMODE (0.64H)
		不正入力フラグをセット	FFALSE (0.68H.2)

入力変数	フロー・チャート〈38〉	処理, 備考	出力変数
<p>ROPE (0.60H)</p>	<pre> graph TD     Start((1)) --&gt; RREG[RREG0 ← 8H RREG1 ← 2H]     RREG --&gt; STRAN[STRAN (52)]     STRAN --&gt; STRNDY[STRNDY (57)]     STRNDY --&gt; RYEXP["RYEXP ← RYEXP - 2H (RYEXP + 1H) ← (RYEXP + 1H) - CY"]     RYEXP --&gt; FPER[FPER ← 1]     FPER --&gt; RCOM[RCOM ← ROPE]     RCOM --&gt; FOPREQ[FOPREQ ← 1]     FOPREQ --&gt; End((2))     End --&gt; RET[RET]             </pre>	<p>REGXのデータをREGZに退避</p> <p>表示データを正規化してREGYへ転送</p> <p>REGYの指数データを-2</p> <p>パーセント・フラグをセット</p> <p>実行演算子エリアに演算子エリアのデータをセット</p> <p>演算要求フラグをセット</p>	<p>RYEXP RYEXP + 1H (1.14H-1.15H)</p> <p>FPER (0.68H.0)</p> <p>RCOM (0.61H)</p> <p>FOPREQ (0.69H.2)</p>

### 6.15 イコール・キー処理

入力変数	フロー・チャート <39>	処理, 備考	出力変数
FPER (0.68H.0)	<pre>                     graph TD                         KEQU([KEQU]) --&gt; D1{FPER = 1?}                         D1 -- Y --&gt; P1[FPER ← 0]                         P1 --&gt; C1["(40) 4"]                         D1 -- N --&gt; D2{RMODE = 4H?}                     </pre>	☒ キーが押されているか?  パーセント・フラグをクリア	FPER (0.68H.0)
RMODE (0.64H)	<pre>                     graph TD                         D2{RMODE = 4H?} -- Y --&gt; P2[FFALSE ← 1]                         P2 --&gt; C2["(40) 5"]                         D2 -- N --&gt; P3[RCOM ← ROPE]                     </pre>	左項モードか?  不正入力フラグをセット	FFALSE (0.68H.2)
ROPE (0.60H)	<pre>                     graph TD                         P3[RCOM ← ROPE] --&gt; D3{RMODE = 2H?}                     </pre>	実行演算子エリアに演算子エリア のデータをセット	RCOM (0.61H)
RMODE (0.64H)	<pre>                     graph TD                         D3{RMODE = 2H?} -- Y --&gt; D4{RCOM &lt; 4H?}                         D4 -- Y --&gt; C3["(40) 1"]                         D4 -- N --&gt; C4["(40) 2"]                         D3 -- N --&gt; C5["(40) 3"]                     </pre>	演算子入力モードか?  ☕ か ☒ キーが押されているか?	

入力変数	フロー・チャート <40>	処理, 備考	出力変数
	<pre> graph TD     1((1)) --&gt; SRYCLR[SRYCLR (48)]     SRYCLR --&gt; 2((2))     2 --&gt; RREG[RREG0 ← 8H RREG1 ← 9H]     RREG --&gt; STRAN[STRAN (52)]     STRAN --&gt; 1     STRAN --&gt; 3((3))     3 --&gt; STRNDY[STRNDY (57)]     STRNDY --&gt; 4((4))     4 --&gt; FOPREQ[FOPREQ ← 1]     FOPREQ --&gt; FOPEND[FOPEND ← 1]     FOPEND --&gt; 5((5))     5 --&gt; RMODE[RMODE ← 4H]     RMODE --&gt; RET([RET])     </pre>	<p>REGYクリア</p> <p>REGXのデータをREGYへ転送</p> <p>表示データを正規化してREGYへ転送</p> <p>演算要求フラグをセット</p> <p>演算終了フラグをセット</p> <p>モードを左項モードに設定</p>	<p>FOPREQ (0.69H.2)</p> <p>FOPEND (0.68H.1)</p> <p>RMODE (0.64H)</p>



### 6.16 演算分岐処理, エラー処理

入力変数	フロー・チャート <41>	処理, 備考	出力変数
<p>RCOM (0.61H)</p> <p>RCOM (0.61H)</p> <p>RCOM (0.61H)</p>	<pre> graph TD     Start([MOPBRN]) --&gt; FOPREQ[FOPREQ ← 0]     FOPREQ --&gt; LCDEN[LCDEN ← 0]     LCDEN --&gt; ROPFLG[ROPFLG ← 0H]     ROPFLG --&gt; RCOM8H{RCOM = 8H?}     RCOM8H -- Y --&gt; C42_2{&lt;42&gt; 2}     RCOM8H -- N --&gt; RCOM4H{RCOM = 4H?}     RCOM4H -- Y --&gt; C42_1{&lt;42&gt; 1}     RCOM4H -- N --&gt; RCOM2H{RCOM = 2H?}     RCOM2H -- Y --&gt; C42_3{&lt;42&gt; 3}     RCOM2H -- N --&gt; SFPADD[SFPADD]     SFPADD -- 注 --&gt; SFPSUB[SFPSUB]     SFPSUB -- 注 --&gt; SFPSUB     SFPSUB --&gt; C42_3     </pre>	<p>演算要求フラグをクリア</p> <p>LCD表示オフ</p> <p>演算フラグをすべてクリア</p> <p>除算命令か?</p> <p>乗算命令か?</p> <p>減算命令か?</p> <p>加算演算</p> <p>減算演算</p>	<p>FOPREQ (0.69H.2)</p> <p>ROPFLG (1.30H)</p>
	<p>注 SFPADD, SFPSUBについては、第8章演算パッケージ説明を参照してください。</p>		

入力変数	フロー・チャート <42>	処理, 備考	出力変数
<p>FDVERR (1.30H.2)</p> <p>RXEXP RXEXP + 1H (1.04H-1.05H)</p>	<pre> graph TD     Start((1)) --&gt; SFPMULT[SFPMULT]     SFPMULT --&gt; SFPDIV[SFPDIV]     SFPDIV --&gt; FDVERR{FDVERR = 1?}     FDVERR -- Y --&gt; Exit1{&lt;44&gt; 1}     FDVERR -- N --&gt; RREG0[RREG0 ← RXEXP RREG1 ← (RXEXP + 1H)]     RREG0 --&gt; RREG1_8H{RREG1 &lt; 8H?}     RREG1_8H -- N --&gt; Exit2{&lt;43&gt; 1}     RREG1_8H -- Y --&gt; Adjust[RREG0 ← RREG0 - 8H RREG1 ← RREG1 - CY]     Adjust --&gt; CY{CY = 1?}     CY -- Y --&gt; Exit3{&lt;43&gt; 2}     CY -- N --&gt; Exit4{&lt;44&gt; 1}     </pre> <p>注 SFPMULT, SFPDIVについては、第8章演算パッケージ説明を参照してください。</p>	<p>乗算命令</p> <p>除算命令</p> <p>除数が0だったか?</p> <p>演算結果がオーバーフローでないか? (7桁以内か)</p>	

入力変数	フロー・チャート <43>	処理, 備考	出力変数
	<pre> graph TD     Start((1)) --&gt; Process1[RREG0 ← RREG0 - BH RREG1 ← RREG1 - FH - CY]     Process1 --&gt; Decision{CY = 0?}     Decision -- Y --&gt; Process2[SFIX &lt;53&gt;]     Decision -- N --&gt; Process3[SRXCLR 注]     Process3 --&gt; Decision     Process2 --&gt; Process4[RDEXP ← 8H (RDEXP+1H) ← 0H]     Process4 --&gt; Process5[RNUMC ← 0H]     Process5 --&gt; End{&lt;44&gt; 2}     </pre> <p>注 SRXCLRについては、第8章 演算パッケージ説明を参照してください。</p>	<p>演算結果がアンダフローでないか？ (小数点以下6桁以内か)</p> <p>REGXクリア</p> <p>演算結果を表示データに変換</p> <p>表示データ指数エリアをリセット</p> <p>数値キー・カウンタをリセット</p>	<p>RDEXP RDEXP + 1H (0.14H-0.15H)</p> <p>RNUMC (0.10H)</p>

入力変数	フロー・チャート〈44〉	処理, 備考	出力変数
	<pre> graph TD     Start((1)) --&gt; RMODE[RMODE ← 8H]     RMODE --&gt; SDPINI[SDPINI &lt;47&gt;]     SDPINI --&gt; RDLSD[RDLSD ← CH]     RDLSD --&gt; SDISP[SDISP &lt;61&gt;]     SDISP --&gt; LCDEN[LCDEN ← 1]     LCDEN --&gt; RET([RET])             </pre>	<p>モードをエラー・モードに設定</p> <p>表示データ・エリアをイニシャライズ</p> <p>表示データ・エリアの最下位桁にエラー表示のデータをセット</p> <p>表示データをLCDセグメント・データに変換</p> <p>LCD表示オン</p>	<p>RMODE (0.64H)</p> <p>RDLSD (0.18H)</p>

### 6.17 RAMオール・クリア処理

入力変数	フロー・チャート<45>	処理, 備考	出力変数
	<pre> graph TD     Start([SRAMCR]) --&gt; Init[MPH ← 8H MPL ← 0H]     Init --&gt; LoopStart((1))     LoopStart --&gt; RREG0[RREG0 ← 2H RREG1 ← 0H]     RREG0 --&gt; LoopBody["[ (MP), (RREG0) ] ← RREG1"]     LoopBody --&gt; RREG0Inc[RREG0 ← RREG0 + 1H]     RREG0Inc --&gt; RREG0Eq0{RREG0 = 0H?}     RREG0Eq0 -- N --&gt; LoopStart     RREG0Eq0 -- Y --&gt; MPLInc[MPL ← MPL + 1H]     MPLInc --&gt; MPLEq7{MPL = 7H?}     MPLEq7 -- Y --&gt; Exit1{&lt;46&gt; 1}     MPLEq7 -- N --&gt; MPLEqF{MPL = FH?}     MPLEqF -- Y --&gt; MPHSet[MPH ← 9H]     MPHSet --&gt; Exit2{&lt;46&gt; 2}     MPLEqF -- N --&gt; LoopStart     </pre>	<p>RAMの全エリア (0.00H, 0.01H を除く) をクリア (0)</p>	<p>MPL (0.7BH)</p> <p>MPH (0.7AH)</p>

入力変数	フロー・チャート <46>	処理, 備考	出力変数
	<pre> graph TD     1((1)) --&gt; D{MPH = 9H?}     D -- Y --&gt; 2((2))     D -- N --&gt; P[MPL ← MPL + 1H]     P --&gt; C46[&lt;46&gt; 1]     2 --&gt; C46     C46 --&gt; RET([RET])             </pre>	<p>RAMの全エリア (0.00H, 0.01H を除く) をクリア (0)</p>	

### 6.18 表示データ・エリア・イニシャライズ処理

入力変数	フロー・チャート <47>	処理, 備考	出力変数
	<pre> graph TD     SDPINI([SDPINI]) --&gt; RREG10[RREG10 ← MPH RREG11 ← MPL]     RREG10 --&gt; RDLSD[RDLSD ← 0H]     RDLSD --&gt; MPH[MPH ← 8H MPL ← 1H]     MPH --&gt; RREG0[RREG0 ← 9H RREG1 ← AH]     RREG0 --&gt; LoopStart[ ]     LoopStart --&gt; RREG1_Set["[ (MP) , (RREG0) ] ←RREG 1"]     RREG1_Set --&gt; RREG0_Inc[RREG0 ← RREG0 + 1H]     RREG0_Inc --&gt; RREG0_0H{RREG0 = 0H ?}     RREG0_0H -- N --&gt; LoopStart     RREG0_0H -- Y --&gt; RPTLOC[RPTLOC ← 7H]     RPTLOC --&gt; MPH_RREG10[MPH ← RREG10 MPL ← RREG11]     MPH_RREG10 --&gt; RET([RET])     </pre>	<p>MPの値を退避</p> <p>表示データ・エリアの最下位桁に0をセット</p> <p>表示データ・エリアの最下位桁以外にスペースのデータをセット</p> <p>小数点表示位置をリセット</p> <p>MPの値を復帰</p>	<p>RDLSD (0.18H)</p> <p>RDLSD + 1H -RDMSD (0.19H-0.1FH)</p> <p>RPTLOC (0.11H)</p>

### 6.19 REGYクリア処理

入力変数	フロー・チャート <48>	処理, 備考	出力変数
	<pre> graph TD     Start([SRYCLR]) --&gt; Save[RREG10 ← MPH RREG11 ← MPL]     Save --&gt; Init[MPH ← 8H MPL ← 9H]     Init --&gt; Set[RREG0 ← 3H RREG1 ← 0H]     Set --&gt; LoopStart[ ]     LoopStart --&gt; LoopBody["[ (MP), (RREG0) ] ← RREG1"]     LoopBody --&gt; LoopInc[RREG0 ← RREG0 + 1H]     LoopInc --&gt; LoopCheck{RREG0 = 0H?}     LoopCheck -- N --&gt; LoopStart     LoopCheck -- Y --&gt; Restore[MPH ← RREG10 MPL ← RREG11]     Restore --&gt; End([RET])     </pre>	<p>MPの値を退避</p> <p>REGYをクリア (0)</p> <p>MPの値を復帰</p>	<p>REGY (1.13H-1.1FH)</p>



### 6.20 表示データ・エリア・ダウン・シフト処理

入力変数	フロー・チャート〈49〉	処理, 備考	出力変数
	<pre> graph TD     Start([SDSHFD]) --&gt; Init[RREG10 ← MPH RREG11 ← MPL]     Init --&gt; SetMP[MPH ← 8H MPL ← 1H]     SetMP --&gt; SetRREG0[RREG0 ← 7H]     SetRREG0 --&gt; LoopStart(( ))     LoopStart --&gt; CalcRREG1["RREG1 ← [ (MP) , (RREG0) ]"]     CalcRREG1 --&gt; DecRREG0["RREG0 ← RREG0 - 1H"]     DecRREG0 --&gt; CalcMP["[ (MP) , (RREG0) ] ← RREG1"]     CalcMP --&gt; IncRREG0["RREG0 ← RREG0 + 2H"]     IncRREG0 --&gt; DecR["RREG0 = 0H ?"]     DecR -- N --&gt; LoopStart     DecR -- Y --&gt; SetRDMSD["RDMSD ← 0H"]     SetRDMSD --&gt; RestoreMP["MPH ← RREG10 MPL ← RREG11"]     RestoreMP --&gt; End([RET])                     </pre>	MPの値を退避	(RDLS D - 2H) - (RDMS D - 1H) (0.16H-0.1EH)
		表示データ・エリアのデータを 1バイト左シフト	
		表示データ・エリアの最上位桁 をクリア (0)	RDMSD (0.1FH)
		MPの値を復帰	

### 6.21 表示データ・エリア・アップ・シフト処理

入力変数	フロー・チャート <50>	処理, 備考	出力変数
	<pre> graph TD     Start([SUSHFD]) --&gt; RREG10[RREG10 ← MPH RREG11 ← MPL]     RREG10 --&gt; MPH[MPH ← 8H MPL ← 1H]     MPH --&gt; RREG0[RREG0 ← EH]     RREG0 --&gt; RREG1[RREG1 ← [(MP), (RREG0)]]     RREG1 --&gt; RREG0_1[RREG0 ← RREG0 + 1H]     RREG0_1 --&gt; RREG1_1["[(MP), (RREG0)] ← RREG1"]     RREG1_1 --&gt; RREG0_2[RREG0 ← RREG0 - 2H]     RREG0_2 --&gt; Decision{RREG0 &lt; 8H?}     Decision -- N --&gt; RDLSD[RDLSD ← 0H]     Decision -- Y --&gt; RDLSD     RDLSD --&gt; MPH_2[MPH ← RREG10 MPL ← RREG11]     MPH_2 --&gt; End([RET])     </pre>	<p>MPの値を退避</p> <p>表示データ・エリアのデータを 1バイト右シフト</p> <p>表示データ・エリアの最下位桁 をクリア (0)</p> <p>MPの値を復帰</p>	<p>(RDLSD + 1H) -RDMSD (0.19H-0.1FH)</p> <p>RDLSD (0.18H)</p>

### 6.22 REGYアップ・シフト処理

入力変数	フロー・チャート〈51〉	処理, 備考	出力変数
	<pre> graph TD     Start([SUSHFY]) --&gt; Assign1[RREG10 ← MPH RREG11 ← MPL]     Assign1 --&gt; Assign2[MPH ← 8H MPL ← 9H]     Assign2 --&gt; Assign3[RREG0 ← EH]     Assign3 --&gt; LoopStart(( ))     LoopStart --&gt; Calc1[RREG1 ← [ (MP), (RREG0) ] ]     Calc1 --&gt; Calc2[RREG0 ← RREG0 + 1H]     Calc2 --&gt; Calc3["[ (MP), (RREG0) ] ← RREG1"]     Calc3 --&gt; Calc4[RREG0 ← RREG0 - 2H]     Calc4 --&gt; Dec1{RREG0 &lt; 6H?}     Dec1 -- N --&gt; Assign4[RYLSD ← 0H]     Dec1 -- Y --&gt; Assign4     Assign4 --&gt; Assign5[MPH ← RREG10 MPL ← RREG11]     Assign5 --&gt; End([RET])     </pre>	<p>MPの値を退避</p> <p>REGYの仮数部のデータを1バイト右シフト</p> <p>REGYの最下位桁をクリア (0)</p> <p>MPの値を復帰</p>	<p>(RYLSD + 1H) -RYMSD (1.17H-1.1FH)</p> <p>RYLSD (1.16H)</p>

### 6.23 データ転送処理

入力変数	フロー・チャート <52>	処理, 備考	出力変数
<p>RREG0 (0.00H)</p> <p>RREG1 (0.01H)</p> <p>REGZ (0.23H-0.2FH)</p> <p>REGX (1.03H-1.0FH)</p> <p>REGY (1.13H-1.1FH)</p>	<pre> graph TD     STRAN([STRAN]) --&gt; RREG10[RREG10 ← MPH RREG11 ← MPL]     RREG10 --&gt; MPH[MPH ← 8H]     MPH --&gt; RREG2[RREG2 ← 3H]     RREG2 --&gt; MPL[MPL ← RREG0]     MPL --&gt; RREG3[RREG3 ← [(MP), (RREG2)]]     RREG3 --&gt; MPL2[MPL ← RREG1]     MPL2 --&gt; RREG3_2["[ (MP), (RREG2) ] ← RREG3"]     RREG3_2 --&gt; RREG2_2[RREG2 ← RREG2 + 1H]     RREG2_2 --&gt; DEC{RREG2 = 0H?}     DEC -- N --&gt; MPL     DEC -- Y --&gt; RREG10_2[RREG10 ← MPH RREG11 ← MPL]     RREG10_2 --&gt; RET([RET])     </pre>	<p>MPの値を退避</p> <p>表示データ・レジスタ, 退避用レジスタ, および浮動小数点レジスタ相互間のデータ転送</p> <p>RREG0 …転送元のロウ・アドレス</p> <p>RREGi …転送先のロウ・アドレス</p> <p>MPのデータを復帰</p>	<p>REGD (0.13H-0.1FH)</p> <p>REGZ (0.23H-0.2FH)</p> <p>REGX (1.03H-1.0FH)</p> <p>REGY (1.13H-1.1FH)</p>

### 6.24 演算結果変換処理

入力変数	フロー・チャート〈53〉	処理, 備考	出力変数
<p>RDEXP (0.14H)</p> <p>RDEXP + 1H (0.15H)</p> <p>RDEXP + 1H (0.15H)</p>	<pre> graph TD     Start([SFIX]) --&gt; RREG[RREG0 ← 8H RREG1 ← 1H]     RREG --&gt; STRAN[STRAN (52)]     STRAN --&gt; D1{RDEXP = 0H?}     D1 -- Y --&gt; D2{(RDEXP + 1H) = 0H?}     D1 -- N --&gt; D3{(RDEXP + 1H) = FH?}     D2 -- Y --&gt; RDEXP[RDEXP ← 1H]     D2 -- N --&gt; D3     RDEXP --&gt; D3     D3 -- Y --&gt; SDSHFD[SDSHFD (49)]     D3 -- N --&gt; D3     SDSHFD --&gt; End[1]     </pre>	<p>REGXのデータをREGDへ転送</p> <p>演算結果の指数データが00Hか?</p> <p>指数データに1をセット</p> <p>指数データが0未満か?</p> <p>表示データ・エリアのデータを1バイト左シフト</p>	<p>RDEXP (0.14H)</p>

入力変数	フロー・チャート <54>	処理, 備考	出力変数
<p>RDSIGN (0.13H)</p>	<pre> graph TD     1((1)) --&gt; D1{RDSIGN = 1H?}     D1 -- Y --&gt; P1[RDMSD ← BH]     D1 -- N --&gt; P2[RDMSD ← AH]     P1 --&gt; P2         </pre>	<p>演算結果が0未満か？</p> <p>表示データ・エリアの最上位桁に “-”の表示データをセット</p> <p>表示データ・エリアの最上位桁に スペースの表示データをセット</p>	<p>RDMSG (0.1FH)</p> <p>RDMSG (0.1FH)</p>
<p>RDEXP+1H (0.15H)</p>	<pre> graph TD     D2{(RDEXP + 1H) = FH?}     P3[RPTLOC ← RDEXP]     C2{&lt;56&gt; 2}     P4[RREG10 ← MPH RREG11 ← MPL]     P5[MPH ← 8H MPL ← 1H]     C1{&lt;55&gt; 1}      D2 -- Y --&gt; C2     D2 -- N --&gt; P3     P3 --&gt; C2     C2 --&gt; P4     P4 --&gt; P5     P5 --&gt; C1         </pre>	<p>指数データが0未満か？</p> <p>小数点表示位置エリアに指数デー タの下位桁をセット</p> <p>MPの値を退避</p> <p>MP設定</p>	<p>RPTLOC (0.11H)</p> <p>RPTLOC (0.11H)</p>



入力変数	フロー・チャート <56>	処理, 備考	出力変数
	<pre> graph TD     Start((1)) --&gt; P1[MPH ← RREG10 MPL ← RREG11]     P1 --&gt; P2[RDEXP ← 1H]     P2 --&gt; P3[RPTLOC ← 1H]     P3 --&gt; P4[RSINLOC ← FH]     P4 --&gt; D1{RDLSD = 0H?}     D1 -- Y --&gt; D2{RPTLOC = 7H?}     D1 -- N --&gt; P5[RPTLOC ← RPTLOC + 1H]     D2 -- Y --&gt; End([RET])     D2 -- N --&gt; P5     P5 --&gt; P6[SDSHFD &lt;49&gt;]     P6 --&gt; P7[RDMSD ← AH]     P7 --&gt; P8[RSINLOC ← RSINLOC - 1H]     P8 --&gt; D1     </pre>	<p>MPの値を復帰</p> <p>表示データ指数エリアに1Hをセット</p> <p>小数点表示位置エリアに1Hをセット</p> <p>符号表示位置エリアにFHをセット</p> <p>表示データのゼロ・サプレス</p>	<p>RDEXP (0.14H)</p> <p>RPTLOC (0.11H)</p> <p>RSINLOC (0.12H)</p> <p>RPTLOC (0.11H)</p> <p>RDMSD (0.1FH)</p> <p>RSINLOC (0.12H)</p>



### 6.25 表示データ変換処理

入力変数	フロー・チャート <57>	処理, 備考	出力変数
<p>RDEXP (0.14H)</p> <p>RNUMC (0.10H)</p> <p>RDLSO -RDMSO (0.18H-0.1FH)</p>	<pre> graph TD     Start([STRNDY]) --&gt; SRYCLR[SRYCLR &lt;48&gt;]     SRYCLR --&gt; RDEXP_8H{RDEXP = 8H?}     RDEXP_8H -- Y --&gt; RDEXP_ASSIGN[RDEXP ← RNUMC]     RDEXP_8H -- N --&gt; RREG10_ASSIGN[RREG10 ← MPH RREG11 ← MPL]     RDEXP_ASSIGN --&gt; RREG10_ASSIGN     RREG10_ASSIGN --&gt; MPH_ASSIGN[MPH ← 8H MPL ← 1H]     MPH_ASSIGN --&gt; RREG0_ASSIGN[RREG0 ← FH]     RREG0_ASSIGN --&gt; RREG1_ASSIGN[RREG1 ← [ (MP), (RREG0) ] ]     RREG1_ASSIGN --&gt; RREG1_AH{RREG1 = AH?}     RREG1_AH -- Y --&gt; RREG0_DEC[RREG0 ← RREG0 - 1H]     RREG0_DEC --&gt; RREG1_ASSIGN     RREG1_AH -- N --&gt; Exit((58) 1)     </pre>	<p>REGYをクリア</p> <p>小数点キーが押されなかったか?</p> <p>表示データ指数エリアに数値キー・カウンタの値をセット</p> <p>MPの値を退避</p> <p>表示データを最上位桁からチェックし、スペース以外のデータが出るまで繰り返す</p>	<p>RDEXP (0.14H)</p>

入力変数	フロー・チャート <58>	処理, 備考	出力変数
	<pre> graph TD     Start((1)) --&gt; Decision{RREG1 = BH?}     Decision -- Y --&gt; Process1[RREG0 ← RREG0 - 1H]     Process1 --&gt; Process2[RYSIGN ← 1H]     Decision -- N --&gt; Process3[RYSIGN ← 0H]     Process2 --&gt; Connector1[ ]     Process3 --&gt; Connector1     Connector1 --&gt; End{&lt;59&gt; 1}     </pre>	<p>表示データの値が0未満か?</p> <p>表示データ・エリアのアドレス・ポインタを-1</p> <p>REGYの符号エリアにマイナスのデータをセット</p> <p>REGYの符号エリアにプラスのデータをセット</p>	<p>RYSIGN (1.13H)</p> <p>RYSIGN (1.13H)</p>

入力変数	フロー・チャート <59>	処理, 備考	出力変数
<p>RDLS -RDMS (0.18H-0.1FH)</p> <p>RDEXP RDEXP + 1H (0.14H-0.15H)</p>	<pre> graph TD     Start((1)) --&gt; RREG1[RREG1 ← EH]     RREG1 --&gt; MPL1[MPL ← 1H]     MPL1 --&gt; RREG2[RREG2 ← [ (MP), (RREG0) ] ]     RREG2 --&gt; MPL2[MPL ← 9H]     MPL2 --&gt; RREG1_2["[ (MP), (RREG1) ] ← RREG2"]     RREG1_2 --&gt; RREG1_3["RREG1 ← RREG1 - 1H RREG0 ← RREG0 - 1H"]     RREG1_3 --&gt; Decide{RREG0 &lt; 8H ?}     Decide -- N --&gt; MPL1     Decide -- Y --&gt; MPH["MPH ← RREG10 MPL ← RREG11"]     MPH --&gt; RYEXP["RYEXP ← RDEXP (RYEXP + 1H) ← (RDEXP + 1H)"]     RYEXP --&gt; End{&lt;60&gt; 1}                     </pre>	<p>表示データ・エリアの数値データをREGYの仮数部へ転送</p> <p>MPのデータを復帰</p> <p>表示データ指数エリアのデータをREGYの指数部へ転送</p>	<p>RYLSD -RYMSD (1.16H-1.1FH)</p> <p>RYEXP RYEXP + 1H (1.14H-1.15H)</p>

入力変数	フロー・チャート <60>	処理, 備考	出力変数
	<pre> graph TD     Start((1)) --&gt; RREG3_8H[RREG3 ← 8H]     RREG3_8H --&gt; RREG3_minus_1H[RREG3 ← RREG3 - 1H]     RREG3_minus_1H --&gt; RREG3_0H{RREG3 = 0H?}     RREG3_0H -- Y --&gt; RYMSD_0H{(RYMSD - 1H) = 0H?}     RREG3_0H -- N --&gt; RREG3_minus_1H     RYMSD_0H -- Y --&gt; SUSHFY[SUSHFY &lt;51&gt;]     RYMSD_0H -- N --&gt; RREG3_minus_1H     SUSHFY --&gt; RYEXP_update[RYEXP ← RYEXP - 1H (RYEXP + 1H) ← (RYEXP + 1H) - CY]     RYEXP_update --&gt; RDEXP_reset[RDEXP ← 8H RDEXP + 1H ← 0H]     RDEXP_reset --&gt; RNUMC_reset[RNUMC ← 0H]     RNUMC_reset --&gt; RET([RET])     </pre>	<p>REGYのゼロ・サブレス</p> <p>表示データ指数エリアをリセット</p> <p>数値キー・カウンタをリセット</p>	<p>RYEXP RYEXP + 1H (1.14H-1.15H)</p> <p>RDEXP RDEXP + 1H (0.14H-0.15H)</p>

### 6.26 表示データ出力処理

入力変数	フロー・チャート <61>	処理, 備考	出力変数
<p>RPTLOC (0.11H)</p>	<pre> graph TD     SDISP([SDISP]) --&gt; RREG10[RREG10 ← MPH RREG11 ← MPL]     RREG10 --&gt; MPH[MPH ← 8H]     MPH --&gt; RREG0[RREG0 ← 8H RREG1 ← 4H RREG2 ← 0H]     RREG0 --&gt; RREG3[RREG3 ← FH - RPTLOC]     RREG3 --&gt; AR[AR ← TSEG DAT]     AR --&gt; MPL[MPL ← 1H]     MPL --&gt; RREG4[RREG4 ← [ (MP), (RREG0) ]]     RREG4 --&gt; RREG4EQ{RREG4 = 0H?}     RREG4EQ -- Y --&gt; Exit1[&lt;62&gt; 1]     RREG4EQ -- N --&gt; ARplus[AR ← AR + 1H]     ARplus --&gt; RREG4minus[RREG4 ← RREG4 - 1H]     RREG4minus --&gt; RREG4EQ                     </pre>	<p>MPの値を退避</p> <p>MPEセット</p> <p>表示データ・エリアの先頭の カラム・アドレスをRREG0に、セ グメント・データ・エリアの先 頭アドレスをRREG1-RREG2に セット</p> <p>小数点の表示位置をRREG3にセ ット</p> <p>セグメント・データ・テーブル の先頭アドレスをARにセット</p>	
<p>RDLS -RDMSD (0.18H-0.1FH)</p>	<p>表示データ読み出し</p>	<p>読み出した表示データから、参 照するセグメント・データのアド レスを指定</p>	

入力変数	フロー・チャート <62>	処理, 備考	出力変数
	<pre> graph TD     Start((1)) --&gt; DBF[DBF ← (AR)]     DBF --&gt; Decision{RREG0 = RREG3?}     Decision -- Y --&gt; DBF0[DBF0 ← DBF0V1H]     Decision -- N --&gt; DBF0     DBF0 --&gt; MPL[MPL ← RREG1]     MPL --&gt; DBF0_ptr["[ (MP), (RREG2) ] ← DBF0"]     DBF0_ptr --&gt; RREG2_1[RREG2 ← RREG2 + 1H]     RREG2_1 --&gt; DBF1_ptr["[ (MP), (RREG2) ] ← DBF1"]     DBF1_ptr --&gt; RREG2_2[RREG2 ← RREG2 + 1H]     RREG2_2 --&gt; DBF2_ptr["[ (MP), (RREG2) ] ← DBF2"]     DBF2_ptr --&gt; End{&lt;63&gt;}     </pre>	<p>セグメント・データ読み出し</p> <p>小数点の表示位置か?</p> <p>セグメント・データを小数点付きに変換</p> <p>セグメント・データをセグメント・データ・エリアに出力</p>	<p>LCDD0 -LCDD31 (0.40H-0.5FH)</p>

入力変数	フロー・チャート <63>	処理, 備考	出力変数
	<pre> graph TD     Start((1)) --&gt; P1[RREG2 ← RREG2 + 1H]     P1 --&gt; P2["[ (MP), (RREG2) ] ← DBF3"]     P2 --&gt; P3["RREG2 ← RREG2 + 1H RREG1 ← RREG1 + CY"]     P3 --&gt; P4[RREG0 ← RREG0 + 1H]     P4 --&gt; D1{RREG0 = 0H?}     D1 -- N --&gt; C1["(61) 1"]     D1 -- Y --&gt; P5["MPH ← RREG10 MPL ← RREG11"]     P5 --&gt; End([RET])     </pre>	<p>セグメント・データをセグメント・データ・エリアに出力</p> <p>セグメント・データ出力が終了したか?</p> <p>MPの値を復帰</p>	

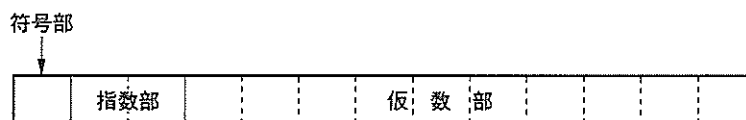
## 第7章 浮動小数点形式 (RAM上のデータの格納形式)

### 7.1 数値の管理構成

この演算パッケージでは、被演算データ、演算データ、および演算結果を、図7-1に示すような浮動小数点付き10進数で扱います。

各演算データは、正負の符号を表す符号部 (1ビット)、小数点位置 ( $10^n$ ) を格納する指数部 (8ビット)、および、符号も小数点も付いていない10進数を扱う仮数部 (40ビット) に分けて管理します。

図7-1 数値の管理構成



- 符号部 : 1ビット (RAM上では1ニブル割り付け)
- 指数部 (16進 2桁) :  $2 \times 4$ ビット
- 仮数部 (10進10桁) :  $10 \times 4$ ビット 整数部 1桁  
小数部 9桁

この数値の管理構成を式で表すと、次のようになります。

$$(-1)^{\text{(符号部値)}} \times \text{(仮数部値)} \times 10^{\text{(指数部値)}}$$

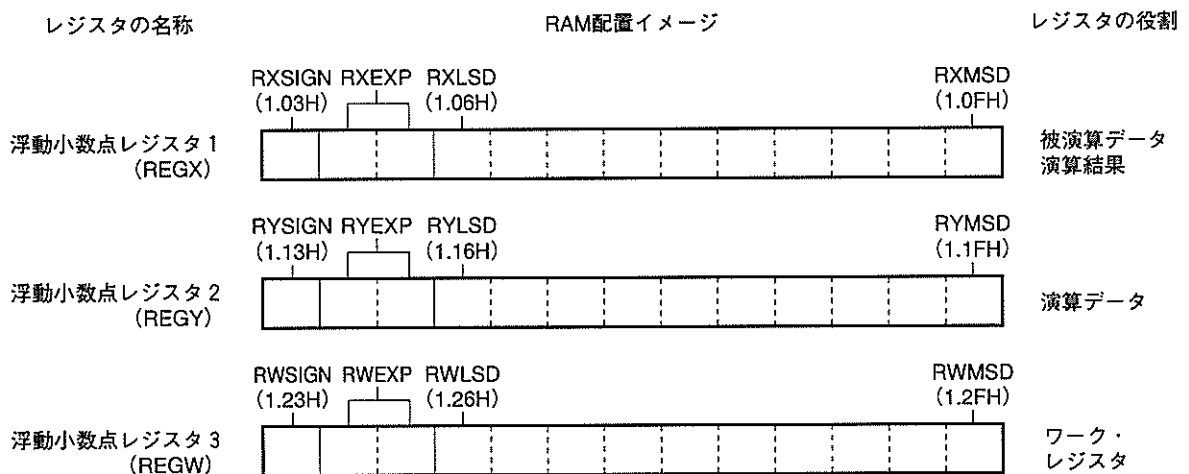


## 7.2 浮動小数点レジスタ

この演算パッケージでは、浮動小数点付き10進数を扱うレジスタとして、3つの浮動小数点レジスタを用意しています。

図7-2に、浮動小数点レジスタのRAM配置イメージを示します。

図7-2 浮動小数点レジスタのRAM配置イメージ



この演算パッケージを使用するときは、REGXに被演算数を、REGYに演算数をセットしたあとに、各演算処理を呼び出します。各演算処理は、REGXに演算結果を格納して終了します。

各レジスタ間のデータの流れの詳細については、8.2 四則演算を参照してください。

## 7.3 浮動小数点形式の各部の説明

この節では、浮動小数点付き10進数の各部について説明します。

### (1) 仮数部

仮数部は有効数字の部分を、整数部1桁と小数部9桁の合計10桁で表現します。仮数部には、正規化された10進数の絶対値が格納されます。

正規化とは、仮数部をあらかじめ定められた範囲内に収めるために、指数部と仮数部を調整することです。この演算パッケージでは、数値が0の場合を除いて、仮数部の範囲を $0.1 \leq \text{仮数部} < 1$ としていますので、正規化を行うと整数部に“0”、かつ小数部最上位桁に“0”以外の数が格納されるように指数部が調整されます。

表7-1に、正規化の例を示します。

表7-1 正規化

正規化される前の値	正規化された値
3729.45	$0.372945 \times 10^4$
0.8765	$0.8765 \times 10^0$
0.00054	$0.54 \times 10^{-3}$

### (2) 指数部

指数部は、表現する数値の底10の指数を、2桁の16進数で構成しています。

負の場合は2つの補数表現になります。

指数部と指数値の対応は、図7-3の通りです。

図7-3 指数部と指数値の対応

指数値 …	$10^7$	$10^6$	…	$10^2$	$10^1$	$10^0$	$10^{-1}$	$10^{-2}$	…	$10^{-5}$	$10^{-6}$ …
指数部 …	07	06	…	02	01	00	FF	FE	…	FB	FA …

備考 2の補数とは、n桁の2進数に対し、 $2^n$ からその数を引いた数をいいます。

### (3) 符号部

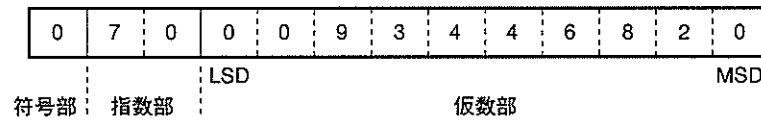
数値の符号は1ビットで表現します。

正数の場合は、符号に対応するビットをリセット(0)、負数の場合はセット(1)します。

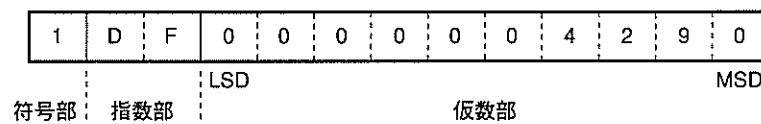
## 7.4 浮動小数点レジスタへの格納例

この節では、浮動小数点付き10進数の浮動小数点レジスタへの格納例を示します。

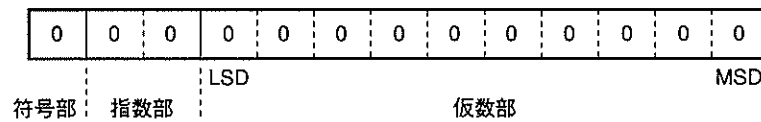
(1)  $2864439 \rightarrow 0.286443900 \times 10^7$



(2)  $-0.000924 \rightarrow -0.924000000 \times 10^{-3}$



(3)  $0 \rightarrow 0.000000000 \times 10^0$



## 第 8 章 演算パッケージ説明

### 8.1 サブルーチン一覧表

表 8-1 にサブルーチン一覧表を示します。

表 8-1 サブルーチン一覧表

サブルーチン名	処理説明	参照箇所
SFPADD	REGXとREGYの浮動小数点数を10進加算し、結果を正規化してREGXに格納します。	8.2.1
SFPSUB	REGXとREGYの浮動小数点数を10進減算し、結果を正規化してREGXに格納します。	8.2.2
SFPMULT	REGXとREGYの浮動小数点数を10進乗算し、結果を正規化してREGXに格納します。	8.2.3
SFPDIV	REGXとREGYの浮動小数点数を10進除算し、結果を正規化してREGXに格納します。	8.2.4
SNML	REGXに格納された演算結果を正規化します。	8.3.1
SADD	REGXとREGYの仮数部を10進加算し、演算結果をREGXに格納します。	8.3.2
SSUB	REGXとREGY、またはREGWの仮数部を10進減算し、演算結果をREGXに格納します。	8.3.3
SADDEX	REGXとREGYの指数部を16進加算し、演算結果をREGXの指数部に格納します。	8.3.4
SSUBEX	REGXとREGYの指数部を16進減算し、演算結果をREGXの指数部に格納します。	8.3.5
SCHGXYEX	REGXとREGYの指数部、仮数部を交換します。	8.3.6 (1)
SCHGXY	REGXとREGYの仮数部を交換します。	8.3.6 (2)
SCHGXW	REGXとREGWの仮数部を交換します。	8.3.6 (3)
SUSHFX	REGXの仮数部を1桁アップ・シフトします。	8.3.7 (1)
SUSHFW	REGWの仮数部を1桁アップ・シフトします。	8.3.7 (2)
SDSHFX	REGXの仮数部を1桁ダウン・シフトします。	8.3.8 (1)
SDSHFY	REGYの仮数部を1桁ダウン・シフトします。	8.3.8 (2)
SDSHFW	REGWの仮数部を1桁ダウン・シフトします。	8.3.8 (3)
SRXCLR	REGXの符号部、指数部、仮数部をクリア (0) します。	8.3.9 (1)
SRWCLR	REGWの仮数部をクリア (0) します。	8.3.9 (2)

## 8.2 四則演算

この演算パッケージでは、四則演算関数として、次の4つを用意しました。

(1) 浮動小数点加算 (SFPADD)

REGXの値を被加数、REGYの値を加数として加算を行います。

(2) 浮動小数点減算 (SFPSUB)

REGXの値を被減数、REGYの値を減数として減算を行います。

(3) 浮動小数点乗算 (SFPMULT)

REGXの値を乗数、REGYの値を被乗数として乗算を行います。

(4) 浮動小数点除算 (SFPDIV)

REGXの値を被除数、REGYの値を除数として除算を行います。

備考 演算結果は、各演算処理の終了時にREGXに格納されます。

### 8.2.1 浮動小数点加算 (SFPADD)

(1) 処理内容

REGXとREGYの浮動小数点数を10進加算し、結果を正規化してREGXに格納します。

(2) 入力条件

REGX、REGYにそれぞれ正規化された被加数、加数を格納します。

(3) 出力結果

- ・REGXには正規化された演算結果を格納しています。
- ・REGYには加数を保持しています。ただし正規化されてはいません。

(4) 使用ワーク・エリア

REGX

REGY

REGW (符号の退避、指数差、演算結果がマイナスになったときの結果補正)

RREG0, RREG1 (REGXとREGYのデータ交換)

## (5) ネスティング・レベル

2レベル

## (6) 処理手順

加算の処理は、(a) → (b) → (c) → (d) → (e) の順で行います。

## (a) 被仮数と仮数の桁合わせ

REGXとREGYの桁を合わせます。固定小数点形式では桁をそろえて演算を行いますが、この演算パッケージで使用している浮動小数点形式では、指数を調整することで、仮数部が固定小数点の加算を行えるようにします。

(i) REGXとREGYの指数値の差を求めます。

(ii) REGXとREGYの指数差を求めた結果、大きい指数値を持っているレジスタの内容がREGXとなるように、必要に応じてREGXとREGYの内容を交換します。

これは、のちに加算の処理を行う際、REGYにある仮数部の内容を、REGXの指数部に合わせてダウン・シフトさせるため、REGYにはREGXより絶対値の小さい値が入っている必要があるためです。また、レジスタの交換には、ワーク・レジスタRREG0、RREG1を使用します。

なお、指数差がない場合、およびREGXの指数部の方がREGYの指数部より大きい場合は、REGXとREGYの内容交換は行いません。

(iii) 指数差が10未満のときは演算処理を継続します。

指数差が10以上のときは、REGXを演算結果として演算を終了します。これは、この演算パッケージで用意した仮数部が10桁なので、小さい数値が格納できないためです。

(iv) 加算の演算処理は、REGXの仮数部にREGYの仮数部を足し込めるようにREGXの指数部の値にそろえるため、REGYの仮数部のダウン・シフト（桁合わせ）を行います。

桁合わせの例を次に示します。

$$68.372 \text{ (REGX)} + 245.91 \text{ (REGY)}$$

$$= (0.68372 \times 10^2) + (0.24591 \times 10^3)$$

；REGYの指数値がREGXの指数値より大きいため、REGXとREGYを交換します。

$$= (0.24591 \times 10^3) \text{ (REGX)}$$

$$+ (0.68372 \times 10^2) \text{ (REGY)}$$

$$= (0.24591 \times 10^3)$$

$$+ (0.068372 \times 10^3)$$

；指数値を $10^3$ にそろえるため、REGYの仮数部をダウン・シフトします。

(b) 被演算データと演算データが同符号の場合の加算処理

REGXとREGYが同符号の場合は、符号を変えずに仮数部の絶対値の加算で演算結果が求められます。そこで、仮数部を10進加算するルーチン（SADD）を呼び出します。

(c) 被演算データと演算データが異符号の場合の加算処理

REGXとREGYが異符号の場合は、加数、被加数をそれぞれ減数、被減数と考えることにより、減算の形で演算結果が求められます。そこで、仮数部を10進減算するルーチン（SSUB）を呼び出します。演算結果はREGXの仮数部に格納されます。

(d) 演算結果の補正


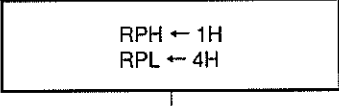
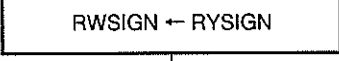
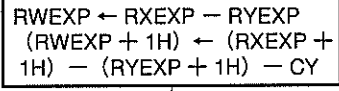
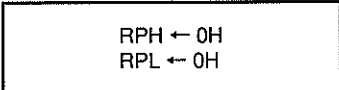
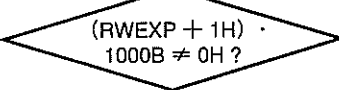
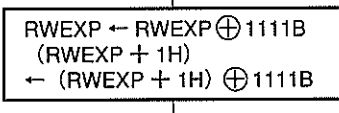
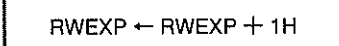
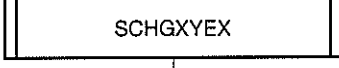
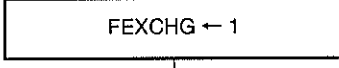
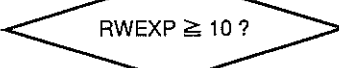
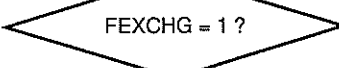

(c)の演算結果がマイナスになった（演算前のREGXの仮数部よりREGYの仮数部が大きい）場合は、仮数部の補数を取り、符号を反転し、演算結果とします。

(e) 正規化

演算結果が格納されたREGXを正規化するルーチン（SNML）を呼び出し、処理を終了します。



(7) 浮動小数点加算処理フロー・チャート

入力変数	フロー・チャート〈64〉	処理, 備考	出力変数
REGY (1.13H-1.1FH)			
REGX (1.03H-1.0FH)		ジェネラル・レジスタをBANK1 のロウ・アドレス2Hに設定	RPH RPL
		加数の符号保持	
		指数差を求める (桁合わせを行い, 加減算を可 能にするため)	
		ジェネラル・レジスタをBANK0 のロウ・アドレス0Hに設定	RPH RPL
		REGY (演算数) の指数値の方 が大きいか?	
		指数差の絶対値をとる (桁合わせのため)	
			
		REGX (被演算数) ↔ REGY (演算数)	
		レジスタ交換フラグをセット	
		指数差が10以上か?	
		レジスタ交換フラグがオンか?	
			
	注 フロー・チャート〈66〉の①へ飛びます。		

入力変数	フロー・チャート〈65〉	処理, 備考	出力変数
	<pre> graph TD     Start((1)) --&gt; P1[RPH ← 1H RPL ← 0H]     P1 --&gt; P2[RYEXP ← RXEXP (RYEXP + 1H) ← (RXEXP + 1H)]     P2 --&gt; P3[RYLSL ← RXLSL (RYLSL + 1H) ← (RXLSL + 1H) (RYLSL + 2H) ← (RXLSL + 2H) (RYLSL + 3H) ← (RXLSL + 3H) (RYLSL + 4H) ← (RXLSL + 4H) (RYLSL + 5H) ← (RXLSL + 5H) (RYLSL + 6H) ← (RXLSL + 6H) (RYLSL + 7H) ← (RXLSL + 7H) (RYLSL + 8H) ← (RXLSL + 8H) RYMSD ← RXMSD]     P3 --&gt; P4[RPH ← 0H]     P4 --&gt; P5[RXSIGN ← RWSIGN]     P5 --&gt; P6[FEXCHG ← 0]     P6 --&gt; End([RET])     </pre>	<p>ジェネラル・レジスタをBANK1の ロウ・アドレス0Hに設定</p> <p>REGYの指数部を復帰</p> <p>REGYの仮数部を復帰</p> <p>ジェネラル・レジスタをBANK0の ロウ・アドレス0Hに設定</p> <p>演算結果符号を格納</p> <p>レジスタ交換フラグをクリア</p>	<p>RPH RPL</p> <p>RPH</p> <p>REGX (1.03H-1.0FH)</p>

入力変数	フロー・チャート <66>	処理, 備考	出力変数
	<pre> graph TD     Start((1)) --&gt; P1[RWEXP ← RWEXP - 1H]     P1 --&gt; D1{CY = 1?}     D1 -- Y --&gt; ExitR[ ]     D1 -- N --&gt; P2[SDSHFY &lt;90&gt;]     P2 --&gt; P1     P1 --&gt; P3[RYEXP ← RXEXP]     P3 --&gt; D2{FEXCHG = 1?}     D2 -- Y --&gt; P4[SCHGXY &lt;85&gt;]     P4 --&gt; P5[FEXCHG ← 0]     P5 --&gt; D3{RXSIGN = RYSIGN?}     D2 -- N --&gt; ExitR     D3 -- Y --&gt; T1[&lt;68&gt; 1]     D3 -- N --&gt; T2[&lt;67&gt; 1]     </pre>	<p>指数差デクリメント</p> <p>指数差がなくなったか?</p> <p>桁合わせ (指数値の大きい方に小さい方を合わせる)</p> <p>演算値ダウン・シフト</p> <p>指数部値をそろえる</p> <p>REGX (被演算数) と REGY (演算数) を交換したか?</p> <p>REGX ↔ REGY (元に戻す)</p> <p>レジスタ交換フラグをクリア</p> <p>REGX と REGY は同符号か?</p>	

入力変数	フロー・チャート <67>	処理, 備考	出力変数
	<pre> graph TD     Start((1)) --&gt; IXM1[IXM ← 1H]     IXM1 --&gt; SSUB1[SSUB &lt;81&gt;]     SSUB1 --&gt; CY{CY = 1?}     CY -- Y --&gt; RXSIGN[RXSIGN ← RXSIGN ⊕ 0001B]     RXSIGN --&gt; SRWCLR[SRWCLR &lt;93&gt;]     SRWCLR --&gt; SCHGXW[SCHGXW &lt;86&gt;]     SCHGXW --&gt; IXM2[IXM ← 2H]     IXM2 --&gt; SSUB2[SSUB &lt;81&gt;]     SSUB2 --&gt; SNML[SNML &lt;76&gt;]     SNML --&gt; RET([RET])     CY -- N --&gt; SSUB2     </pre>	<p>インデクス・レジスタにREGYのロウ・アドレスをセット</p> <p>REGX(被演算数) - REGY (演算数)</p> <p>減算でポローがでたか?</p> <p>演算結果符号反転</p> <p>REGW ← ALL 0</p> <p>REGX(演算結果) ↔ REGW</p> <p>インデクス・レジスタ ← REGWのロウ・アドレス</p> <p>REGX(0) - REGW(演算結果)</p> <p>演算結果の正規化</p> <p>補数表現を補数なし表現に改める</p>	<p>IXM</p> <p>REGX (1.03H-1.0FH)</p> <p>IXM</p>

入力変数	フロー・チャート <68>	処理, 備考	出力変数
	<pre> graph TD     Start((1)) --&gt; SetIXM[IXM ← 1H]     SetIXM --&gt; SADD[SADD]     SADD --- 80["&lt;80&gt;"]     SADD --&gt; End{&lt;67&gt; 2}                     </pre>	<p>インデクス・レジスタにREGYの ロウ・アドレスをセット</p> <p>REGX(被演算数) + REGY (演算数)</p>	<p>IXM</p>

## 8.2.2 浮動小数点減算 (SFPSUB)

### (1) 処理内容

REGXとREGYの浮動小数点数を10進減算し、結果を正規化してREGXに格納します。

### (2) 入力条件

REGX, REGYにそれぞれ正規化された被減数, 減数を格納します。

### (3) 出力結果

- ・REGXには正規化された演算結果を格納しています。
- ・REGYには減数を保持しています。ただし正規化されてはいません。

### (4) 使用ワーク・エリア

REGX

REGY

REGW (符号の退避, 指数差, 演算結果がマイナスになったときの結果補正)

RREG0, RREG1 (REGXとREGYのデータ交換)

### (5) ネスティング・レベル

2レベル

### (6) 処理手順

減算は、減数の符号を反転することにより、加算として考えることができます。

このため、(a) → (b) の順で処理を行います。

#### (a) 減数の符号反転

REGYの符号部を反転します。

#### (b) 浮動小数点加算処理

この後の処理は加算と同様なので、浮動小数点加算 (SFPADD) に処理を移行します。

詳細は、浮動小数点加算 (SFPADD) の処理手順を参照してください。

(7) 浮動小数点減算処理フロー・チャート

入力変数	フロー・チャート〈69〉	処理、備考	出力変数
<p>REGY (1.13H-1.1FH)</p>	<pre> graph TD     Start([SFPSUB]) --&gt; Process1[RPH ← 1H&lt;br/&gt;RPL ← 4H]     Process1 --&gt; Process2[RWSIGN ← RYSIGN ⊕ 0001B]     Process2 --&gt; End[&lt;64&gt; 1]             </pre>	<p>ジェネラル・レジスタをBANK1の ロウ・アドレス2Hに設定</p> <p>減数の符号反転したものを退避</p>	<p>RPH RPL</p>

### 8.2.3 浮動小数点乗算 (SFPMULT)

#### (1) 処理内容

REGXとREGYの浮動小数点数を10進乗算し、結果を正規化してREGXに格納します。

#### (2) 入力条件

REGX, REGYにそれぞれ正規化された乗数, 被乗数を格納します。

備考 乗算処理に限り、REGXを演算データ、REGYを被演算データとして扱います。

#### (3) 出力結果

- ・REGXには正規化された演算結果を格納しています。
- ・REGYには乗数を保持しています。

#### (4) 使用ワーク・エリア

REGX

REGY

REGW (演算結果の符号, 指数値の退避, 乗数の格納, カウンタ)

RREG0 (退避した符号の復帰)

#### (5) ネスティング・レベル

2レベル

#### (6) 処理手順

この演算パッケージの乗算は、筆算の要領で行っています。

具体例として、 $25.7 \times (-0.32)$  を挙げて、筆算による乗算の手順を考えます。

図8-1 筆算による乗算例

$$\begin{array}{r}
 + 25.7 \\
 \times (-0.32) \\
 \hline
 514 \quad \dots \textcircled{2} (1) \\
 771 \quad \dots \textcircled{2} (2) \\
 \hline
 - 8.224 \\
 \begin{array}{ccc}
 | & | & | \\
 \textcircled{1} & \textcircled{4} & \textcircled{3}
 \end{array}
 \end{array}$$

- ① 演算結果の符号を求めます。被乗数と乗数が同符号なら結果の符号は正で、異符号なら負となります。この例は正数と負数の乗算なので、結果は負となります。



② 被乗数と、乗数のそれぞれの桁の乗算を、乗数の下位桁から行います。このとき、小数点は無視して演算を行います。

(1)  $257 \times 2$  を行います。

(2)  $257 \times 3$  を行います。乗数の 3 は 10 の位にあたるので、結果は 1 桁左にずらして書きます。

③ 桁ごとの乗算の結果を加算します。

④ 演算結果の小数点の位置を求めます。この例の場合、被乗数が  $257 \times 10^{-1}$ 、乗数が  $32 \times 10^{-2}$  なので、③の加算結果を  $10^{-3}$  倍します。つまり、下から 3 桁目と 4 桁目の間に小数点を置きます。

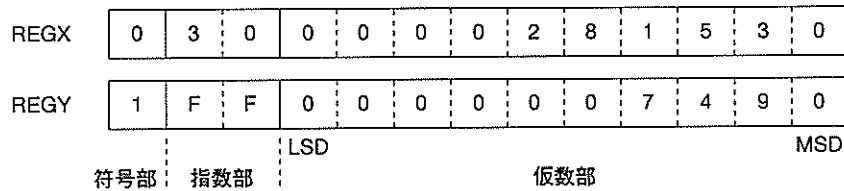
以上より、 $25.7 \times (-0.32)$  の結果は  $-8.224$  となります。

この演算パッケージでは、筆算による乗算を、次のように実現しています。

ここでは具体例として、 $351.82 \text{ (REGX)} \times (-0.0947) \text{ (REGY)}$  を挙げて説明します。

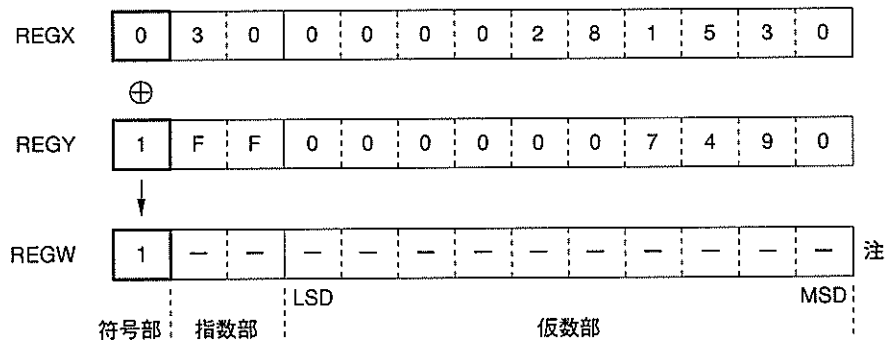
$$351.82 = 0.35182 \times 10^3$$

$$-0.0947 = -0.947 \times 10^{-1}$$



(a) 符号部の乗算

演算結果の符号を求めます (REGXの符号部 ⊕ REGYの符号部)。乗数 (REGX) と被乗数 (REGY) が同符号なら結果の符号は正 (0) で、異符号なら負 (1) となります。結果は REGW の符号部に退避します。

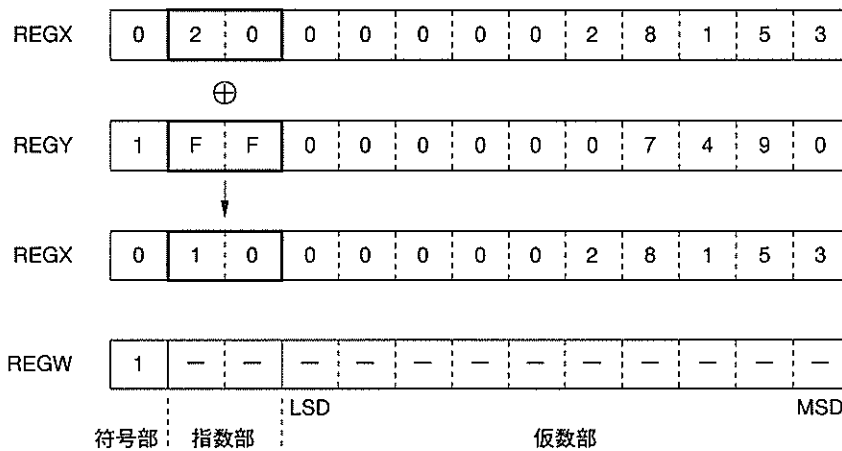


注 “-” の部分には不定の値が入っています。



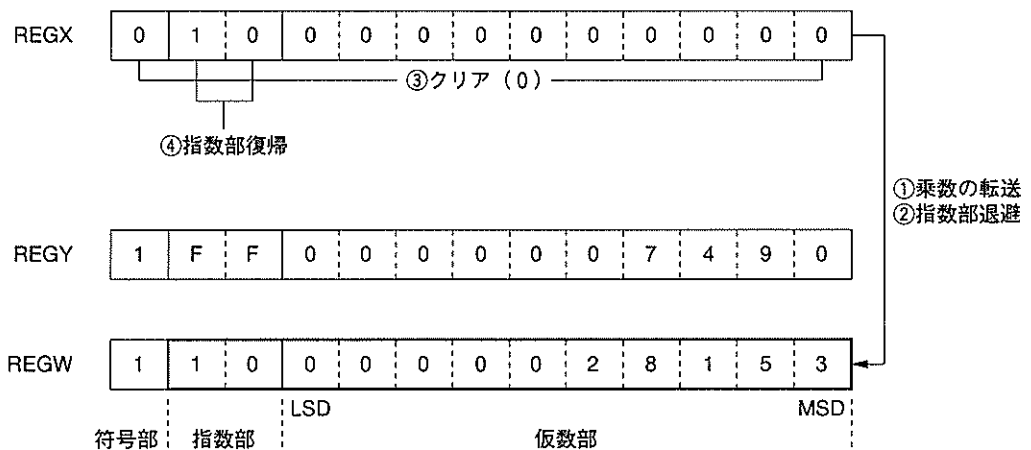
(c) 指数部の加算

演算結果の指数を求めます。乗数 (REGX) と被乗数 (REGY) の指数部を加算 (SADDEX) し、結果をREGXの指数部に格納します。



(d) 乗数の転送と演算結果エリアのクリア

REGXとREGWの仮数部を交換 (SCHGXW) し、REGXを0にクリア (SRXCLR) します。これは、REGXに演算結果を格納するためです。また、クリア (0) ののち、REGXの指数部には、演算結果の指数値を復帰します。



(e) 仮数部10桁ダウン・シフト・カウンタの設定

REGWの指数部の下位桁を仮数部10桁ダウン・シフト・カウンタとします。



仮数部10桁ダウン・シフト・カウンタは、仮数部の桁数を初期値として設定します。これは、乗数の下位桁から順に乗算（桁乗数<sup>注</sup>×REGY）を行うため、仮数部のダウン・シフトの回数を記憶しておく必要があるためです。

注 (f) の処理で、ダウン・シフトによる最下位桁オーバーフロー格納エリアとして使用します。

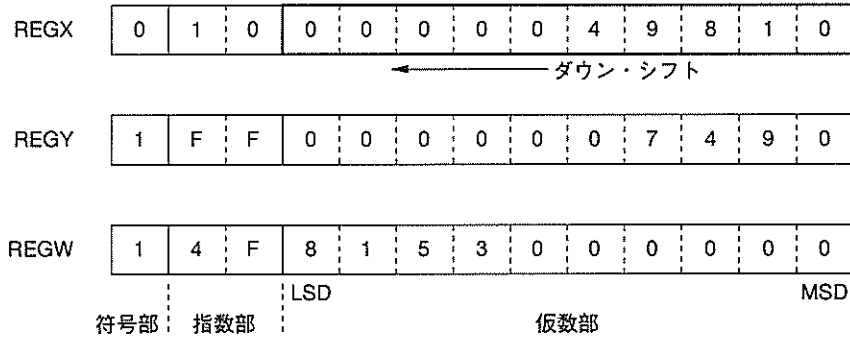
(f) 乗数の1桁ごとの乗算

(i) 仮数部10桁ダウン・シフト・カウンタをダウン・カウントします。カウンタの値がFHになったところで、(f) の処理を終了します。





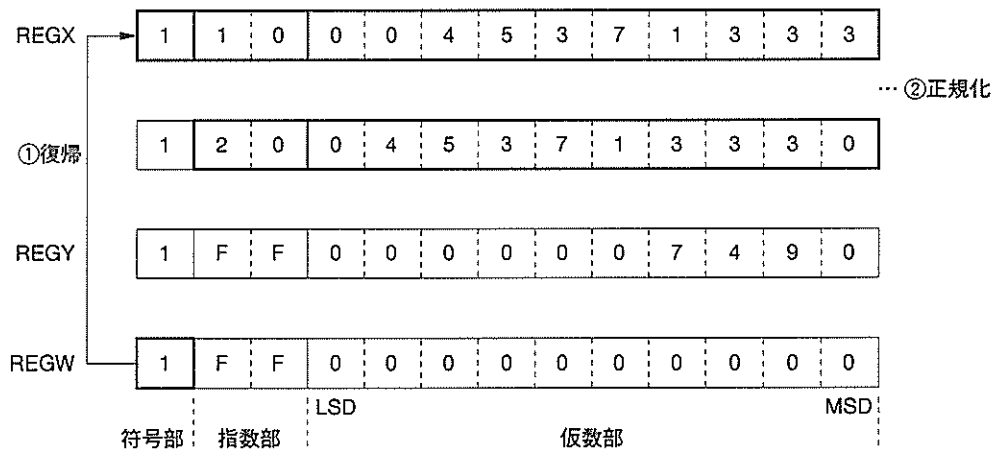
(iv) REGXの仮数部を1桁ダウン・シフト (SDSHFX) します。筆算による乗算では、乗数の桁ごとに乗算結果を上位桁にずらして書きますが、この演算パッケージでは、逆に、それまでの桁の乗算結果の合計を下位桁にずらすことにより、同じ処理を実現しています。




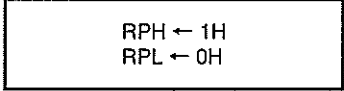

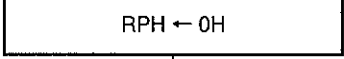
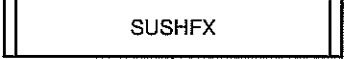
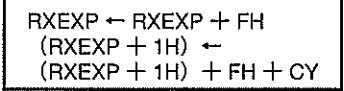

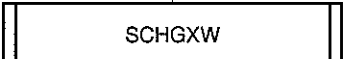
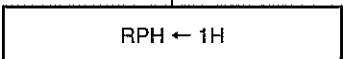
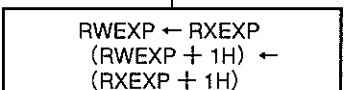
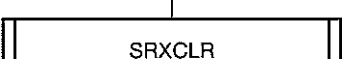
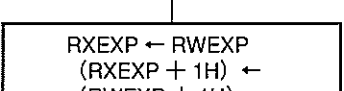

(v) (i) - (iv) の動作を仮数部10桁分 (仮数部10桁ダウン・シフト・カウンタがFHになるまで) 行います。

(g) 正規化

REGWに退避しておいた結果符号をREGXに復帰し、正規化 (SNML) を行い、処理を終了します。



(7) 浮動小数点乗算処理フロー・チャート

入力変数	フロー・チャート<70>	処理, 備考	出力変数															
REGX (1.03H-1.0FH) REGY (1.13H-1.1FH)																		
		ジェネラル・レジスタをBANK1の ロウ・アドレス0Hに設定	RPH RPL															
		演算結果符号を求め, 退避しておく  <table border="1" data-bbox="890 703 1204 837"> <tr> <td>RXSIGN</td> <td>RYSIGN</td> <td>→RWSIGN</td> </tr> <tr> <td>+</td> <td>+</td> <td>+</td> </tr> <tr> <td>+</td> <td>-</td> <td>-</td> </tr> <tr> <td>-</td> <td>+</td> <td>-</td> </tr> <tr> <td>-</td> <td>-</td> <td>+</td> </tr> </table>	RXSIGN	RYSIGN	→RWSIGN	+	+	+	+	-	-	-	+	-	-	-	+	
	RXSIGN	RYSIGN	→RWSIGN															
	+	+	+															
	+	-	-															
	-	+	-															
	-	-	+															
		ジェネラル・レジスタをBANK0の ロウ・アドレス0Hに設定	RPH															
		乗数のアップ・シフト																
	指数部値-1																	
	指数部加算 (REGX - REGY) (積の指数値を求める)																	
	REGX ↔ REGW																	
	ジェネラル・レジスタをBANK1の ロウ・アドレス0Hに設定	RPH																
	指数部退避 (REGXクリア(0)のため)																	
	REGX (積) ← ALL 0																	
	指数部復帰																	
																		

入力変数	フロー・チャート <71>	処理, 備考	出力変数
	<pre> graph TD     1((1)) --&gt; RPH[RPH ← 0H]     RPH --&gt; RWEXP[RWEXP ← AH]     RWEXP --&gt; RWEXP2[RWEXP ← RWEXP - 1H]     RWEXP2 --&gt; CY{CY = 1?}     CY -- Y --&gt; RXSIGN[RXSIGN ← RWSIGN]     CY -- N --&gt; 72{&lt;72&gt; 1}     72 --&gt; RXSIGN     RXSIGN --&gt; SNML[SNML]     SNML --&gt; RET([RET])     </pre>	<p>ジェネラル・レジスタをBANK0のロウ・アドレス0Hに設定</p> <p>仮数部10桁アップ・シフト・カウンタ初期設定</p> <p>仮数部10桁アップ・シフト・カウンタ・デクリメント</p> <p>仮数部10桁分の加算が終了したか?</p> <p>演算結果符号を格納</p> <p>積の正規化</p>	<p>RPH</p> <p>REGX (1.03H-1.0FH)</p>



入力変数	フロー・チャート <72>	処理, 備考	出力変数
<p>REGW (1.23H-1.2FH)</p>	<pre> graph TD     Start((1)) --&gt; SDSHFX[SDSHFX &lt;89&gt;]     SDSHFX --&gt; SDSHFW[SDSHFW &lt;91&gt;]     SDSHFW --&gt; Decrement["(RWEXP + 1H) ← (RWEXP + 1H) - 1H"]     Decrement --&gt; CY{CY = 1?}     CY -- Y --&gt; SDSHFW     CY -- N --&gt; IXM["IXM ← 1H"]     IXM --&gt; SADD[SADD &lt;80&gt;]     SADD --&gt; Decrement     </pre>	<p>積のエリアのダウン・シフト</p> <p>乗数のエリアのダウン・シフト</p> <p>桁乗数デクリメント</p> <p>加算回数終了したか?</p> <p>インデクス・レジスタにREGYの ロウ・アドレスをセット</p> <p>REGX (積) + REGY (被乗数)</p>	<p>IXM</p>

### 8.2.4 浮動小数点除算 (SFPDIV)

(1) 処理内容

REGXとREGYの浮動小数点を10進除算し、結果を正規化してREGXに格納します。

(2) 入力条件

REGX, REGYにそれぞれ正規化された被除数, 除数を格納します。

(3) 出力結果

- ・REGXには正規化された演算結果を格納しています。  
除数が0の場合, エラー (FDVERRフラグをセット) となり, REGXに格納されている被除数を保持したまま処理を終了します。
- ・REGYには除数を保持しています。

(4) 使用ワーク・エリア

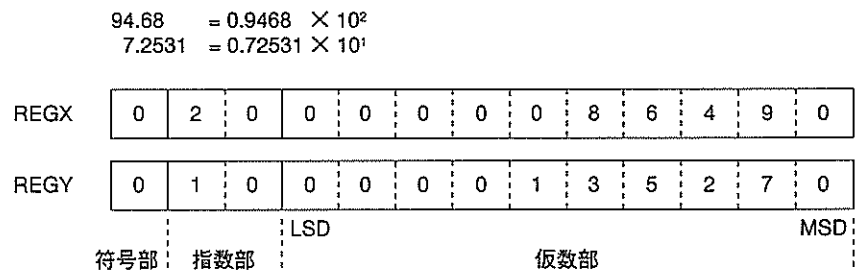
- REGX
- REGY
- REGW (演算結果, カウンタ)
- RREG0 (除数のゼロ・チェック)

(5) ネスティング・レベル

2レベル

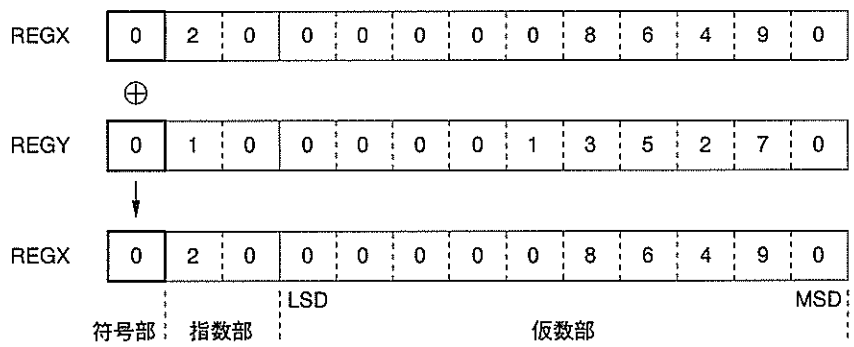
(6) 処理手順

具体例として, 94.68 (REGX) ÷ 7.2531 (REGY) を挙げて説明します。



(a) 符号部の除算

演算結果の符号を求め (REGXの符号部 ⊕ REGYの符号部) , REGXの符号部に格納します。被除数 (REGX) と除数 (REGY) が同符号なら結果の符号は正 (0) で、異符号なら負 (1) となります。



(b) 除数のゼロ・チェック

REGYの仮数部が0ならエラー (FDVERRフラグをセット) となり、処理を終了します。

(c) 指数部の減算

演算結果の指数を求めます。被除数 (REGX) と除数 (REGY) の指数部を減算 (SSUBEX) し、結果をREGXの指数部に格納します。

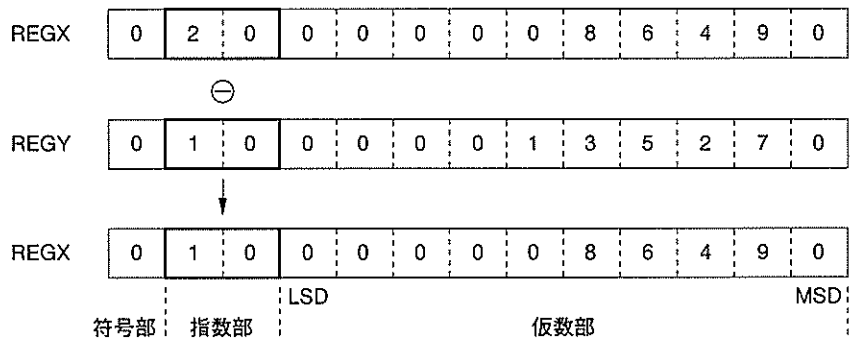




図8-2 復帰法による除算例

除算例	処理内容	商
$\begin{array}{r} 1.2 \\ 5 \overline{) 6} \rightarrow \\ \underline{5} \quad \dots \textcircled{1} \\ 10 \quad \dots \textcircled{2} \\ \underline{10} \quad \dots \textcircled{3} \\ 0 \end{array}$	$\begin{array}{r} \textcircled{1} \quad 6 \\ \underline{-5} \\ 1 \end{array}$ 被除数から除数が減算できたので、商として1をたてます。	1
$\begin{array}{r} 1 \\ \underline{-5} \\ -4 \end{array}$	減算結果が負になったので、商は数えません。	1
$\begin{array}{r} \textcircled{2} \quad -4 \\ +5 \\ \hline 1 \\ \downarrow \\ 10 \end{array}$	減算結果を正数に戻すため、除数を加算します。さらに正数になった減算結果をアップ・シフトします。それにより商の桁が1つ下がります。	1
$\begin{array}{r} \textcircled{3} \quad 10 \\ \underline{-5} \\ 5 \end{array}$	減算できたので、商として0.1をたて、現在の商の1と加算します。	1.1
$\begin{array}{r} \textcircled{3} \quad 5 \\ \underline{-5} \\ 0 \end{array}$	減算できたので、商に0.1を加算します。減算結果が0になったので終了します。	1.2

(i) 仮数部10桁アップ・シフト・カウンタをダウン・カウントします。

REGX	0	1	0	0	0	0	0	0	0	0	0	0
------	---	---	---	---	---	---	---	---	---	---	---	---

REGY	0	1	0	0	0	0	0	1	3	5	2	7	0
------	---	---	---	---	---	---	---	---	---	---	---	---	---

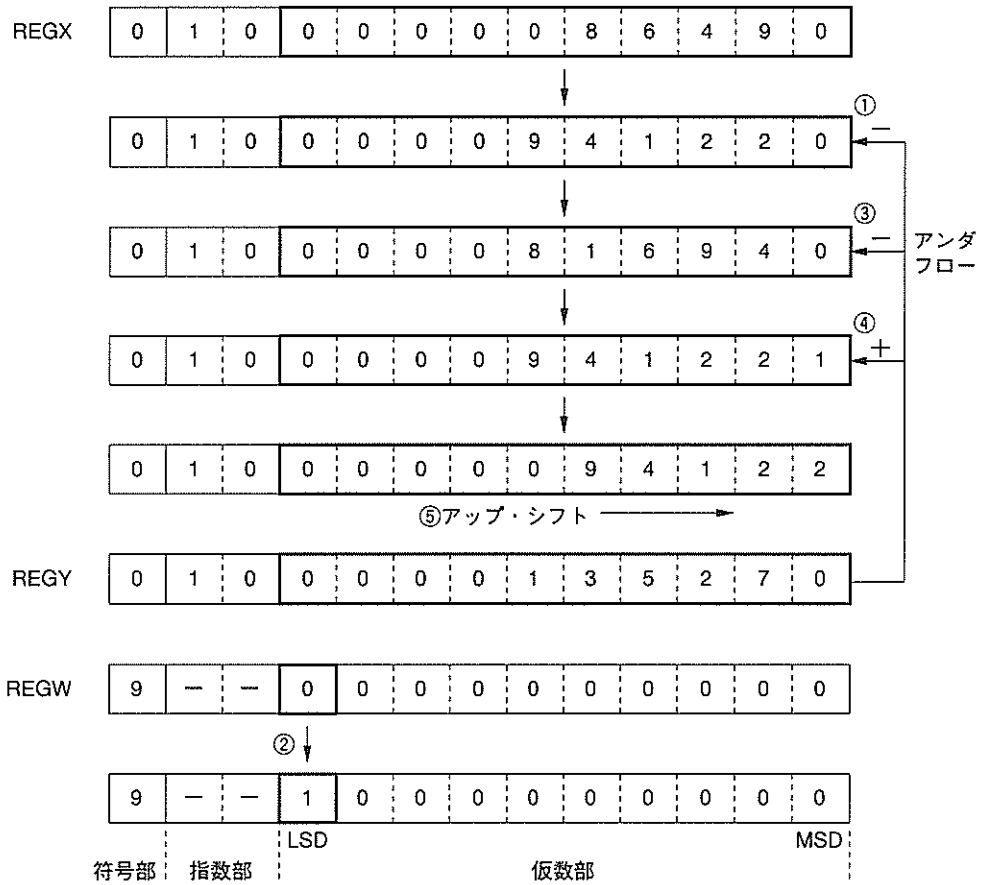
REGW	9	-	-	0	0	0	0	0	2	8	1	5	3
------	---	---	---	---	---	---	---	---	---	---	---	---	---

仮数部10桁アップ・シフト・カウンタ-1

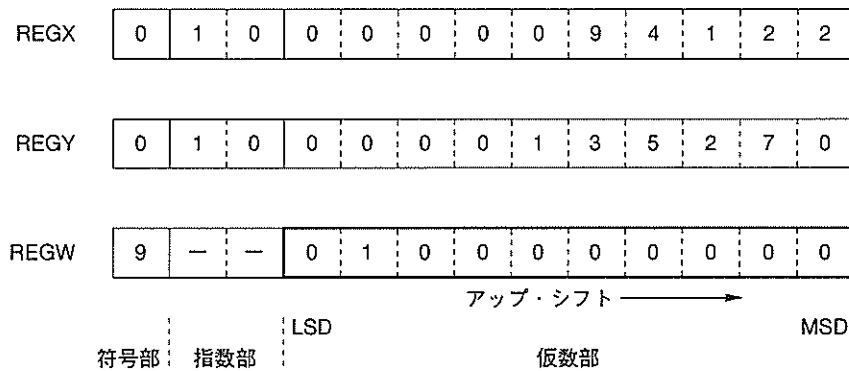
符号部	指数部	LSD	仮数部	MSD
-----	-----	-----	-----	-----

(ii) REGXからREGYを減算します (SSUB)。アンダフローしなければ、REGWの最下桁に商として1を加えます。

アンダフローした場合は、REGXとREGYを加算 (SADD) し、REGXを1桁アップ・シフトします (SUSHFX)。これは、減算結果をアップ・シフトすることにより次の減算に備えるためです。



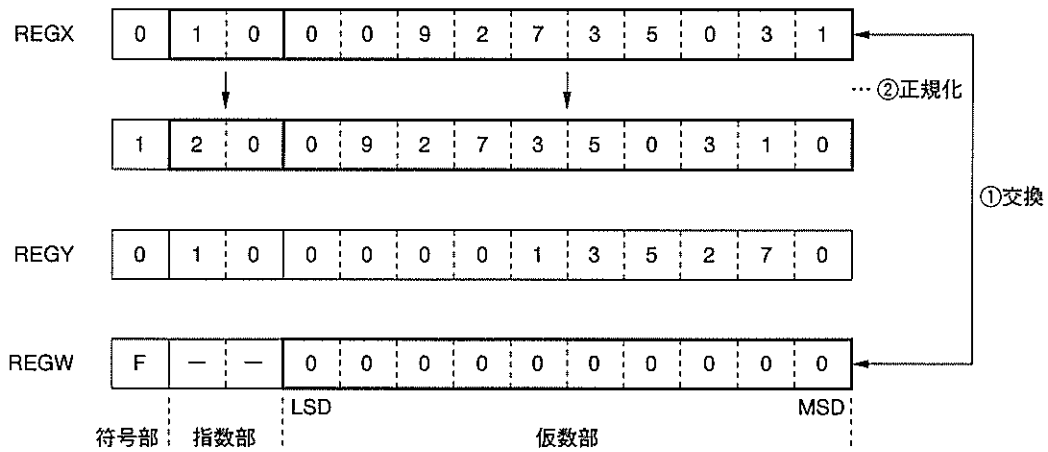
(iii) REGWの仮数部を1桁アップ・シフト (SUSHFW) します。これは、演算結果をアップ・シフトすることにより、商の桁を更新するためです。



(iv) (i) - (iii) の動作を仮数部10桁分（仮数部10桁アップ・シフト・カウンタがFHになるまで）行います。

(f) 正規化

REGXとREGWの仮数部を交換（SCHGXW）し、演算結果をREGXに格納し、正規化（SNML）を行い、処理を終了します。



(7) 浮動小数点除算処理フロー・チャート

入力変数	フロー・チャート <73>	処理, 備考	出力変数
<p>REGX (1.03H-1.0FH) REGY (1.13H-1.1FH)</p>	<pre> graph TD     SFPDIV([SFPDIV]) --&gt; FVERR0[FVERR ← 0]     FVERR0 --&gt; RPH1[RPH ← 1H RPL ← 0H]     RPH1 --&gt; RXSIGN[RXSIGN ← RXSIGN ⊕ RYSIGN]     RXSIGN --&gt; RPH0[RPH ← 0H]     RPH0 --&gt; IXH0[IXH ← 0H IXM ← 8H IXL ← 6H]     IXH0 --&gt; IXE1[IXE ← 1]     IXE1 --&gt; RREG0[RREG0 ← REGY: 除数 [IX]]     RREG0 --&gt; IXE0[IXE ← 0]     IXE0 --&gt; RREG0_0H{RREG0 = 0H?}     RREG0_0H -- N --&gt; 74((74))     RREG0_0H -- Y --&gt; IX_1H[IX ← IX + 1H]     IX_1H --&gt; IXL_0H{IXL = 0H?}     IXL_0H -- N --&gt; IXE1     IXL_0H -- Y --&gt; FVERR1[FVERR ← 1]     FVERR1 --&gt; RET([RET])     </pre>	<p>除数ゼロ・エラー・フラグをクリア</p> <p>ジェネラル・レジスタをBANK1のロウ・アドレス0Hに設定</p> <p>演算結果の符号を求める</p> <p>ジェネラル・レジスタをBANK0のロウ・アドレス0Hに設定</p> <p>インデクス・レジスタをBANK1のカラム・アドレス6Hに設定</p> <p>除数 (REGY仮数部) のゼロ・チェック</p> <p>除数ゼロ・エラー・フラグをセット</p>	<p>FVERR (1.30H.2)</p> <p>RPH RPL</p> <p>REGX (1.03H-1.0FH)</p> <p>RPH</p> <p>FVERR (1.30H.2)</p>



入力変数	フロー・チャート <74>	処理, 備考	出力変数
	<pre> graph TD     1((1)) --&gt; SSUBEX[SSUBEX &lt;83&gt;]     SSUBEX --&gt; SRWCLR[SRWCLR &lt;93&gt;]     SRWCLR --&gt; RWSIGN_AH[RWSIGN ← AH &lt;2&gt;]     RWSIGN_AH --&gt; RWSIGN_1H[RWSIGN ← RWSIGN - 1H]     RWSIGN_1H --&gt; CY{CY = 1?}     CY -- Y --&gt; SCHGXW[SCHGXW &lt;86&gt;]     CY -- N --&gt; 75{&lt;75&gt; 1}     75 --&gt; SCHGXW     SCHGXW --&gt; SNML[SNML &lt;76&gt;]     SNML --&gt; RET([RET])         </pre>	<p>指数部減算 (REGX - REGY) (商の指数値を求める)</p> <p>REGW (商) ← ALL 0</p> <p>仮数部10桁アップ・シフト・カウンタ初期設定</p> <p>仮数部10桁アップ・シフト・カウンタ・デクリメント</p> <p>仮数部10桁分の減算が終了したか?</p> <p>REGX ↔ REGW (REGWの商をREGXに転送)</p> <p>商の正規化</p>	

入力変数	フロー・チャート <75>	処理, 備考	出力変数
	<pre> graph TD     Start((1)) --&gt; SUSHFW[SUSHFW (88)]     SUSHFW --&gt; IXM[IXM ← 1H]     IXM --&gt; SSUB[SSUB (81)]     SSUB --&gt; CY{CY = 1?}     CY -- Y --&gt; SADD[SADD (80)]     SADD --&gt; SUSHFX[SUSHFX (87)]     SUSHFX --&gt; LoopStart{&lt;74&gt; 2}     LoopStart --&gt; RWLSD[RWLSD ← RWLSD + 1H]     RWLSD --&gt; SSUB     CY -- N --&gt; RWLSD     </pre>	<p>商のエリアのアップ・シフト</p> <p>インデクス・レジスタにREGYの ロウ・アドレスをセット</p> <p>REGX (被除数) - REGY (除数)</p> <p>減算でボローがでたか?</p> <p>REGX (被除数) + REGY (除数)</p> <p>被除数のエリアのアップ・シフト</p> <p>商の最下位桁に1を加算</p>	<p>IXM</p>

## 8.3 その他のサブルーチン (ユーザから呼ばれることのない内部ルーチン)

### 8.3.1 正規化 (SNML)

#### ●処理内容

REGXに格納された演算結果を正規化します。

演算結果が0の場合、演算結果ゼロ・フラグ (FZERO) がセット (1) されます。

演算結果がオーバーフロー、またはアンダフローした場合、オーバーフロー・フラグ (FOVER) がセット (1) されます。

オーバーフロー、アンダフローの検出は、REGXの指数値により行います。指数値は、16進数2桁で、2の補数表現を用いて表していますので、表せる指数値の範囲は $-128 \leq \text{指数値} \leq 127$ となりますが、この演算パッケージでは、指数値が $-64 \leq \text{指数値} \leq 63$ の範囲を越えた場合をオーバーフロー、アンダフローとし、オーバーフロー・フラグをセットしています。オーバーフロー・フラグをシステム制御部でチェックすることにより、オーバーフロー、アンダフローの発生による演算結果の破壊を防ぐことができます。

また、演算結果が桁上がりしたときは、仮数部最下位桁を切り捨てます。

$$\begin{aligned}
 &0.77777 \times 0.88888 \\
 &= 0.77777 \times (8.8888 \times 10^{-1}) \\
 &= (0.77777 \times 8.8888) \times 10^{(0+(-1))} \\
 &= 6.913441976 \times 10^{-1} \quad ; \text{最下位桁を切り捨てます} \\
 &= 0.691344197 \times 10^0
 \end{aligned}$$

**備考** このアプリケーション・ノート (この浮動小数点演算パッケージを応用した電卓のアプリケーション) では、オーバーフロー・フラグを無視しています。

●正規化処理フロー・チャート

入力変数	フロー・チャート <76>	処理, 備考	出力変数
REGX (1.03H-1.0FH)			
		演算結果ゼロ・フラグをクリア	FZERO (1.30H.1)
		オーバフロー・フラグをクリア	FOVER (1.30H.3)
		ジェネラル・レジスタをBANK0 のロウ・アドレス0Hに設定	RPH RPL
		インデクス・レジスタを仮数部 の最下位桁に設定	
		演算結果 (REGX仮数部) のゼロ・チェック	

入力変数	フロー・チャート <77>	処理, 備考	出力変数
	<pre> graph TD     Start((1)) --&gt; D1{RXMSD = 0H?}     D1 -- Y --&gt; D2{(RXMSD - 1H) = 0H?}     D1 -- N --&gt; D3{(RXEXP + 1H) · 1000B = 0H?}     D2 -- Y --&gt; P1[SUSHFX &lt;87&gt;]     P1 --&gt; U1["RXEXP ← RXEXP + FH (RXEXP + 1H) ← (RXEXP + 1H) + FH + CY"]     U1 --&gt; D3     D2 -- N --&gt; P2[SDSHFX &lt;89&gt;]     P2 --&gt; U2["RXEXP ← RXEXP + 1H (RXEXP + 1H) ← (RXEXP + 1H) + CY"]     U2 --&gt; D3     D3 -- Y --&gt; T1[&lt;78&gt;]     T1 --&gt; D4{(RXEXP + 1H) &lt; 4H?}     D3 -- N --&gt; D4     D4 -- Y --&gt; P3[FOVER ← 1]     D4 -- N --&gt; P3     P3 --&gt; End([RET])     </pre>	<p>演算結果の整数部が0か?</p> <p>演算結果の小数部最上位桁が0か?</p> <p>演算結果のアップ・シフト</p> <p>指数部値-1</p> <p>演算結果のダウン・シフト</p> <p>指数部値+1</p> <p>指数部値は正か?</p> <p>指数部値は64未満か? } オーバフロー判定</p> <p>オーバーフロー・フラグをセット</p>	<p>REGX (1.03H-1.0FH)</p> <p>REGX (1.03H-1.0FH)</p> <p>FOVER (1.30H.3)</p>

入力変数	フロー・チャート〈78〉	処理, 備考	出力変数
	<pre> graph TD     Start((1)) --&gt; Decision{ (RXEXP + 1H) ≥ CH? }     Decision -- Y --&gt; Process[FOVER ← 1]     Decision -- N --&gt; End([RET])     End --&gt; Process     </pre>	<p>指数部値が -64以上か?</p> <p>アンダフロー 判定</p> <p>オーバフロー・ フラグをセット</p>	<p>FOVER (1.30H.3)</p>

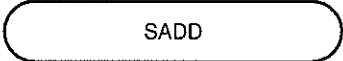
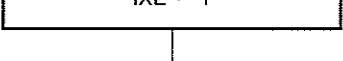
入力変数	フロー・チャート <79>	処理, 備考	出力変数
	<pre> graph TD     Start((1)) --&gt; SRXCLR[SRXCLR &lt;92&gt;]     SRXCLR --&gt; FZERO[FZERO ← 1]     FZERO --&gt; RET([RET])             </pre>	<p>演算結果 ← 0</p> <p>演算結果ゼロ・フラグをセット</p>	<p>FZERO (1.30H.1)</p>

### 8.3.2 仮数部加算 (SADD)

●処理内容

REGXとREGYの仮数部を10進加算し、演算結果をREGXに格納します。10進加算は、プログラム・ステータス・ワードのBCDフラグをセットすることにより行っています。

●仮数部加算処理フロー・チャート

入力変数	フロー・チャート〈80〉	処理, 備考	出力変数
IXM  REGX (1.03H-1.0FH)		インデクス・レジスタをクリア  ジェネラル・レジスタをBANK1 のロウ・アドレス0Hに設定 BCDフラグをセット  REGXの仮数部とインデクス修 飾されたレジスタの仮数部を10 進加算  ジェネラル・レジスタをBANK0 のロウ・アドレス0Hに設定 BCDフラグをクリア	RPH RPL      REGX (1.03H-1.0FH)      RPH RPL
	IXH ← 0H IXL ← 0H		
	RPH ← 1H RPL ← 1H		
	IXE ← 1		
	REGX仮数部 ← REGX仮数部 + REGX仮数部 [IX]		
	IXE ← 0		
	RPH ← 0H RPL ← 0H		
			

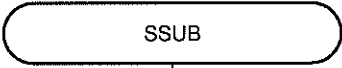
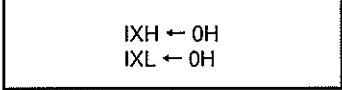
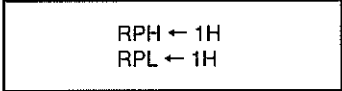
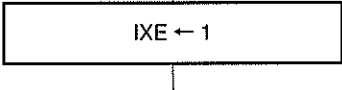
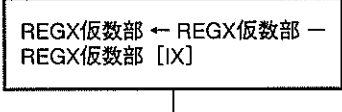
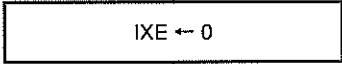
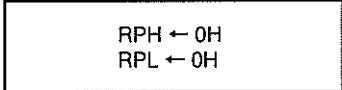
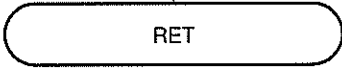


### 8.3.3 仮数部減算 (SSUB)

●処理内容

REGXとREGY, またはREGWの仮数部を10進減算し, 演算結果をREGXに格納します。10進減算は, プログラム・ステータス・ワードのBCDフラグをセットすることにより行っています。

●仮数部減算処理フロー・チャート

入力変数	フロー・チャート <81>	処理, 備考	出力変数
IXM  REGX (1.03H-1.0FH)			
		インデクス・レジスタをクリア	
		ジェネラル・レジスタをBANK1のロウ・アドレス0Hに設定 BCDフラグをセット	RPH RPL
			
		REGXの仮数部とインデクス修飾されたレジスタの仮数部を10進減算	REGX (1.03H-1.0FH)
			
		ジェネラル・レジスタをBANK0のロウ・アドレス0Hに設定 BCDフラグをクリア	RPH RPL
			

### 8.3.4 指数部加算 (SADDEX)

●処理内容

REGXとREGYの指数部を16進加算し、演算結果をREGXの指数部に格納します。

●指数部加算処理フロー・チャート

入力変数	フロー・チャート <82>	処理, 備考	出力変数
REGX (1.03H-1.0FH) REGY (1.13H-1.1FH)	<pre>                     graph TD                         Start([SADDEX]) --&gt; SetRPH[RPH ← 1H RPL ← 0H]                         SetRPH --&gt; Calc[RXEXP ← RXEXP + RYEXP (RXEXP + 1H) ← (RXEXP + 1H) + (RYEXP + 1H) + CY]                         Calc --&gt; SetRPH0[RPH ← 0H]                         SetRPH0 --&gt; End([RET])                     </pre>	ジェネラル・レジスタをBANK1の ロウ・アドレス0Hに設定  REGXとREGYの指数部値加算  ジェネラル・レジスタをBANK0の ロウ・アドレス0Hに設定	RPH RPL  REGX (1.03H-1.0FH)  RPH

### 8.3.5 指数部減算 (SSUBEX)

●処理内容

REGXとREGYの指数部を16進減算し、演算結果をREGXの指数部に格納します。

●指数部減算処理フロー・チャート

入力変数	フロー・チャート <83>	処理, 備考	出力変数
<p>REGX (1.03H-1.0FH) REGY (1.13H-1.1FH)</p>	<pre> graph TD     Start([SSUBEX]) --&gt; SetRPH[RPH ← 1H RPL ← 0H]     SetRPH --&gt; Calc[RXEXP ← RXEXP - RYEXP (RXEXP + 1H) ← (RXEXP + 1H) - (RYEXP + 1H) - CY]     Calc --&gt; SetRPH0[RPH ← 0H]     SetRPH0 --&gt; End([RET])             </pre>	<p>ジェネラル・レジスタをBANK1の ロウ・アドレス0Hに設定</p> <p>REGXとREGYの指数部値減算</p> <p>ジェネラル・レジスタをBANK0の ロウ・アドレス0Hに設定</p>	<p>RPH RPL</p> <p>REGX (1.03H-1.0FH)</p> <p>RPH</p>

### 8.3.6 レジスタ交換

#### (1) レジスタ交換1 (SCHGX YEX)

●処理内容

REGXとREGYの指数部、仮数部を交換します。

●レジスタ交換処理1フロー・チャート

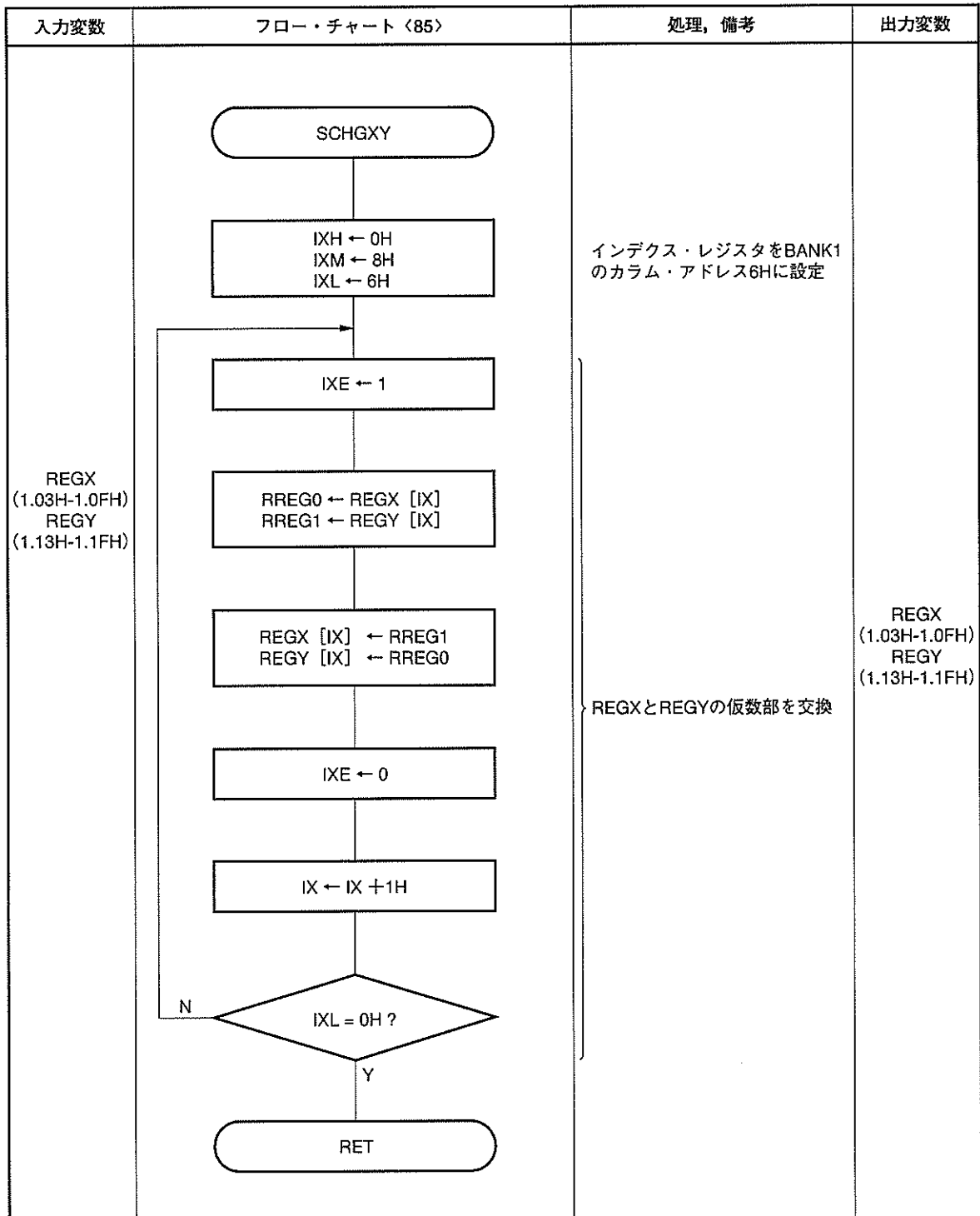
入力変数	フロー・チャート <84>	処理, 備考	出力変数
<p>REGX (1.03H-1.0FH) REGY (1.13H-1.1FH)</p>	<pre> graph TD     Start([SCHGX YEX]) --&gt; Init[IXH ← 0H IXM ← 8H IXL ← 4H]     Init --&gt; LoopStart[IXE ← 1]     LoopStart --&gt; LoopBody[RREG0 ← REGX [IX] RREG1 ← REGY [IX]]     LoopBody --&gt; Exchange[REGX [IX] ← RREG1 REGY [IX] ← RREG0]     Exchange --&gt; LoopEnd[IXE ← 0]     LoopEnd --&gt; IncIX[IX ← IX + 1H]     IncIX --&gt; Decision{IXL = 0H?}     Decision -- N --&gt; LoopStart     Decision -- Y --&gt; End([RET])     </pre>	<p>インデクス・レジスタをBANK1 のカラム・アドレス4Hに設定</p> <p>REGXとREGYの指数部、仮数部 を交換</p>	<p>REGX (1.03H-1.0FH) REGY (1.13H-1.1FH)</p>

(2) レジスタ交換2 (SCHGXY)

●処理内容

REGXとREGYの仮数部を交換します。

●レジスタ交換処理2フロー・チャート







(2) レジスタ・アップ・シフト 2 (SUSHFW)

●処理内容

REGWの仮数部を1桁アップ・シフトします。

●レジスタ・アップ・シフト処理2フロー・チャート

入力変数	フロー・チャート <88>	処理, 備考	出力変数
<p>REGW (1.23H-1.2FH)</p>		<p>ジェネラル・レジスタをBANK1の ロウ・アドレス2Hに設定</p> <p>REGWの仮数部を1桁アップ・シ フト</p> <p>ジェネラル・レジスタをBANK0の ロウ・アドレス0Hに設定</p>	<p>RPH RPL</p> <p>REGW (1.23H-1.2FH)</p> <p>RPH RPL</p>



### 8.3.8 レジスタ・ダウン・シフト

#### (1) レジスタ・ダウン・シフト1 (SDSHFX)

●処理内容

REGXの仮数部を1桁ダウン・シフトします。

●レジスタ・ダウン・シフト処理1フロー・チャート

入力変数	フロー・チャート <89>	処理, 備考	出力変数
REGX (1.03H-1.0FH)	<pre>                     graph TD                         Start([SDSHFX]) --&gt; RPH1H[RPH ← 1H RPL ← 0H]                         RPH1H --&gt; Loop                         subgraph Loop [ ]                             direction TB                             L1[RXLSD ← (RXLSD + 1H)]                             L2["(RXLSD + 1H) ← (RXLSD + 2H)"]                             L3["(RXLSD + 2H) ← (RXLSD + 3H)"]                             L4["(RXLSD + 3H) ← (RXLSD + 4H)"]                             L5["(RXLSD + 4H) ← (RXLSD + 5H)"]                             L6["(RXLSD + 5H) ← (RXLSD + 6H)"]                             L7["(RXLSD + 6H) ← (RXLSD + 7H)"]                             L8["(RXLSD + 7H) ← (RXLSD + 8H)"]                             L9["(RXLSD + 8H) ← (RXLSD + 9H)"]                             L10[RXMSD ← 0H]                         end                         Loop --&gt; RPH0H[RPH ← 0H RPL ← 0H]                         RPH0H --&gt; End([RET])                     </pre>	ジェネラル・レジスタをBANK1の ロウ・アドレス0Hに設定	RPH RPL
	REGXの仮数部を1桁ダウン・シフト	ジェネラル・レジスタをBANK0の ロウ・アドレス0Hに設定	REGX (1.03H-1.0FH)  RPH RPL

(2) レジスタ・ダウン・シフト2 (SDSHFY)

●処理内容

REGYの仮数部を1桁ダウン・シフトします。

●レジスタ・ダウン・シフト処理2フロー・チャート

入力変数	フロー・チャート <90>	処理, 備考	出力変数
<p>REGY (1.13H-1.1FH)</p>		<p>ジェネラル・レジスタをBANK1の ロウ・アドレス1Hに設定</p> <p>REGYの仮数部を1桁ダウン・シ フト</p> <p>ジェネラル・レジスタをBANK0の ロウ・アドレス0Hに設定</p>	<p>RPH RPL</p> <p>REGY (1.13H-1.1FH)</p> <p>RPH RPL</p>

(3) レジスタ・ダウン・シフト3 (SDSHFW)

●処理内容

REGWの仮数部を1桁ダウン・シフトします。

●レジスタ・ダウン・シフト処理3フロー・チャート

入力変数	フロー・チャート〈91〉	処理, 備考	出力変数
<p>REGW (1.23H-1.2FH)</p>	<pre> graph TD     Start([SDSHFW]) --&gt; Step1["(RWEXP + 1H) ← RWLSD"]     Step1 --&gt; Step2["RPH ← 1H RPL ← 4H"]     Step2 --&gt; Step3["RWLSD ← (RWLSD + 1H) (RWLSD + 1H) ← (RWLSD + 2H) (RWLSD + 2H) ← (RWLSD + 3H) (RWLSD + 3H) ← (RWLSD + 4H) (RWLSD + 4H) ← (RWLSD + 5H) (RWLSD + 5H) ← (RWLSD + 6H) (RWLSD + 6H) ← (RWLSD + 7H) (RWLSD + 7H) ← (RWLSD + 8H) (RWLSD + 8H) ← (RWLSD + 9H) RWMSD ← 0H"]     Step3 --&gt; Step4["RPH ← 0H RPL ← 0H"]     Step4 --&gt; End([RET])                     </pre>	<p>仮数部最下位桁 (加算カウンタ) を退避</p> <p>ジェネラル・レジスタをBANK1のロウ・アドレス2Hに設定</p> <p>REGWの仮数部を1桁ダウン・シフト</p> <p>ジェネラル・レジスタをBANK0のロウ・アドレス0Hに設定</p>	<p>RPH RPL</p> <p>REGW (1.23H-1.2FH)</p> <p>RPH RPL</p>

### 8.3.9 レジスタ・クリア (0)

#### (1) レジスタ・クリア (0) 1 (SRXCLR)

●処理内容

REGXの符号部、指数部、仮数部をクリア (0) します。

●レジスタ・クリア (0) 処理1フロー・チャート

入力変数	フロー・チャート <92>	処理, 備考	出力変数
REGX (1.03H-1.0FH)	<pre> graph TD     Start([SRXCLR]) --&gt; Init[IXH ← 0H IXM ← 8H IXL ← 3H]     Init --&gt; LoopStart[IXE ← 1]     LoopStart --&gt; Clear[REGX [IX] ← 0H]     Clear --&gt; DecIXE[IXE ← 0]     DecIXE --&gt; IncIX[IX ← IX + 1H]     IncIX --&gt; Decision{IXL = 0H?}     Decision -- N --&gt; LoopStart     Decision -- Y --&gt; End([RET])                     </pre>	<p>インデクス・レジスタをBANK1 のカラム・アドレス3Hに設定</p> <p>REGXの符号部、指数部、仮数 部をクリア (0)</p>	REGX (1.03H-1.0FH)

(2) レジスタ・クリア (0) 2 (SRWCLR)

●処理内容

REGWの仮数部をクリア (0) します。

●レジスタ・クリア (0) 処理2フロー・チャート

入力変数	フロー・チャート<93>	処理, 備考	出力変数
<p>REGW (1.23H-1.2FH)</p>	<pre> graph TD     Start([SRWCLR]) --&gt; Init[IXH ← 0H IXM ← 8H IXL ← 6H]     Init --&gt; LoopStart(( ))     LoopStart --&gt; IXE1[IXE ← 1]     IXE1 --&gt; Clear[REGW [IX] ← 0H]     Clear --&gt; IXE0[IXE ← 0]     IXE0 --&gt; IncIX[IX ← IX + 1H]     IncIX --&gt; Decision{IXL = 0H?}     Decision -- N --&gt; LoopStart     Decision -- Y --&gt; End([RET])             </pre>	<p>インデクス・レジスタをBANK1 のカラム・アドレス6Hに設定</p> <p>REGWの仮数部をクリア (0)</p>	<p>REGW (1.23H-1.2FH)</p>

## 第 9 章 電卓プログラム

### 9.1 システム制御部プログラム

次に、このアプリケーション・ノートで紹介する電卓のシステム制御部のプログラム・リストを示します。

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:22:42 12/20/93 PAGE 01-001

PROG =

SOURCE = CALCDEF.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
1
2 ;*****
3 ;* USER NAME : NEC CORPORATION *
4 ;* *
5 ;* SYSTEM NAME : 17K APPLICATION (CALCULATOR) *
6 ;* *
7 ;* CPU : uPD17201A *
8 ;* *
9 ;* LAST UPDATE : '93/12/20 11:00 *
10 ;* *
11 ;* FILE NAME : CALCDEF.ASM *
12 ;* *
13 ;*****
14 ;*****
15 ;* *
16 ;* コントロール・レジスタ定義 *
17 ;* *
18 ;*****
19 PUBLIC BELOW
20 0082 PCC MEM 0.82H ;システム・クロック制御レジスタ
21 0083 WTC MEM 0.83H ;時計用タイマ/ウォッチドッグ・
22 ; タイマ制御レジスタ
23 00A1 ADM MEM 0.0A1H ;A/Dコンバータ動作モード/端子選択
24 00A2 SIC MEM 0.0A2H ;シリアル・インタフェース制御レジスタ
25 00A3 PSL MEM 0.0A3H ;ポート/タイマ出力, LED出力, シリアル
26 ;・インタフェース入力/出力制御レジスタ
27 00A7 PM0 MEM 0.0A7H ;ビットI/O入力/出力制御レジスタ
28 00AF IPF MEM 0.0AFH ;割り込み許可フラグ
29 00B1 LCDC MEM 0.0B1H ;LCDコントローラ/ドライバ制御レジスタ
30 00B2 LCDM MEM 0.0B2H ;表示モード・レジスタ
31 00B3 TMCT MEM 0.0B3H ;8ビット・タイマ/カウンタ制御レジスタ
32 00B7 PM1 MEM 0.0B7H ;グループI/O入力/出力制御レジスタ
33
34 EJECT
    
```

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:22:42 12/20/93 PAGE 01-002

PROG =

SOURCE = CALCDEF.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
35 ;*****
36 ;*
37 ;*          R A M定義          *
38 ;*
39 ;*****
40 ;*****
41 ;*          B A N K O          *
42 ;*****
43 ;+++      汎用レジスタ      +++
44 0000 RREG0 MEM 0.00H
45 0001 RREG1 MEM 0.01H
46 0002 RREG2 MEM 0.02H
47 0003 RREG3 MEM 0.03H
48 0004 RREG4 MEM 0.04H
49 ;RREG5 MEM 0.05H
50 ;RREG6 MEM 0.06H
51 ;RREG7 MEM 0.07H
52 ;RREG8 MEM 0.08H
53 0009 RREG9 MEM 0.09H
54 000A RREG10 MEM 0.0AH
55 000B RREG11 MEM 0.0BH
56
57 ;+++      表示データ・レジスタ (R E G D) +++
58 0010 RNUMC MEM 0.10H ;数値キー・カウンタ
59 0011 RPTLOC MEM 0.11H ;小数点表示位置エリア
60 0012 RSINLOC MEM 0.12H ;符号表示位置エリア
61 0013 RDSIGN MEM 0.13H ;演算結果符号エリア
62 0014 RDEXP MEM 0.14H ;表示データ指数エリア
63 0018 RDLSD MEM 0.18H ;表示データ・エリア (最下位桁)
64 001F RDMSD MEM 0.1FH ; " (最上位桁)
65
66 ;+++      退避用レジスタ (R E G Z)      +++
67 0023 RZLSD MEM 0.23H
68 ;RZMSD MEM 0.2FH
69
70 EJECT
    
```



AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:22:42 12/20/93 PAGE 01-003

PROG =

SOURCE = CALCDEF.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
71 ;+++ キー・スキャン +++
72 0030 RRCODH MEM 0.30H ;キー・コード・エリア (上位4ビット)
73 0031 RRCODL MEM 0.31H ; " (下位4ビット)
74 0032 RCHCODH MEM 0.32H ;チャタリング・コード・エリア
75 ; (上位4ビット)
76 0033 RCHCODL MEM 0.33H ; " (下位4ビット)
77 0034 RIPCODH MEM 0.34H ;インプット・コード・エリア (上位4ビット)
78 0035 RIPCODL MEM 0.35H ; " (下位4ビット)
79 0036 RCHATC MEM 0.36H ;チャタリング・カウンタ
80
81 ;+++ オートパワーオフ・タイマ +++
82 0038 RSTOP3 MEM 0.38H ;3分間カウンタ (10ms x 18,000)
83 0039 RSTOP2 MEM 0.39H
84 003A RSTOP1 MEM 0.3AH
85 003B RSTOP0 MEM 0.3BH
86
87 ;+++ 演算子 +++
88 0060 ROPE MEM 0.60H ;演算子エリア
89 0061 RCOM MEM 0.61H ;実行演算子エリア
90
91 ;+++ モード +++
92 0064 RMODE MEM 0.64H
93
94 ;+++ システム・フラグ・エリア +++
95 0068 RSYSFLG MEM 0.68H
96 0681 FPER FLG RSYSFLG.0 ;パーセント・フラグ
97 0682 FOPEND FLG RSYSFLG.1 ;演算終了フラグ
98 0684 FFALSE FLG RSYSFLG.2 ;不正入力フラグ
99 0688 FSTOP FLG RSYSFLG.3 ;動作停止モード・フラグ
100
101 ;+++ イベント・フラグ・エリア +++
102 0069 REVEFLG MEM 0.69H
103 0691 FKEYREQ FLG REVEFLG.0 ;キー処理要求フラグ
104 0692 FMULTI FLG REVEFLG.1 ;多重押しフラグ
105 0694 FOPREQ FLG REVEFLG.2 ;演算要求フラグ
106 0698 FSTPREQ FLG REVEFLG.3 ;STOPモード処理要求フラグ
107
108 ;+++ ポート・レジスタ +++
109 0070 POA MEM 0.70H
110 0071 POB MEM 0.71H
111 0072 POC MEM 0.72H
112 0073 POD MEM 0.73H
113 0170 PIA MEM 1.70H
114
115 EJECT
    
```

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:22:42 12/20/93 PAGE 01-004

PROG =

SOURCE = CALCDEF.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
116 ;*****
117 ;* B A N K 1 *
118 ;*****
119 ;+++ 浮動小数点レジスタ 1 +++
120 0103 RXSIGN MEM 1.03H ;符号部
121 0104 RXEXP MEM 1.04H ;指数部
122 0106 RXLSD MEM 1.06H ;仮数部 (最下位桁)
123 010F RXMSD MEM 1.0FH ; " (最上位桁)
124
125 ;+++ 浮動小数点レジスタ 2 +++
126 0113 RYSIGN MEM 1.13H ;符号部
127 0114 RYEXP MEM 1.14H ;指数部
128 0116 RYLSD MEM 1.16H ;仮数部 (最下位桁)
129 011F RYMSD MEM 1.1FH ; " (最上位桁)
130
131 ;+++ 浮動小数点レジスタ 3 +++
132 0123 RWSIGN MEM 1.23H ;符号部
133 0124 RWEXP MEM 1.24H ;指数部
134 0126 RWLSD MEM 1.26H ;仮数部 (最下位桁)
135 012F RWMSD MEM 1.2FH ; " (最上位桁)
136
137 ;+++ 演算フラグ +++
138 0130 ROPFLG MEM 1.30H
139 1301 FEXCHG FLG ROPFLG.0 ;レジスタ交換フラグ
140 1302 FZERO FLG ROPFLG.1 ;演算結果ゼロ・フラグ
141 1304 FDEVERR FLG ROPFLG.2 ;除算エラー・フラグ
142 1308 FOVER FLG ROPFLG.3 ;オーバフロー・フラグ
143
144 EJECT
    
```

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:22:42 12/20/93 PAGE 01-005

PROG =

SOURCE = CALCDEF.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
145 ;*****
146 :*
147 :*          コンスタント値定義          *
148 :*
149 ;*****
150 ;*****
151 :*          ハードウェア          *
152 ;*****
153 ;+++      8ビット・モジュロ・レジスタ      +++
154 009C CRLHLT DAT 9CH          ;HALTモード解除 (10ms)
155 0000 CRLSTP DAT 0H          ;STOPモード解除
156
157 ;*****
158 :*          システム・レジスタ          *
159 ;*****
160 ;+++      データ・メモリ・ロウ・アドレス      +++
161          ・ポインタ (MP)          +++
162 0008 CSTMPE DAT 1000B          ;MPH: 間接アドレス指定 (BANK 0)
163 0009 CSTBK2 DAT 1001B          ; " : (BANK 2)
164 0008 CSTBK1 DAT 1000B          ;MPL: " (BANK 1)
165
166 ;*****
167 :*          ポート入出力データ/キー・コード          *
168 ;*****
169 ;+++      キー・スキャン信号          +++
170          → キー・コード (上位4ビット)          +++
171 000E CKSRB0 DAT 1110B          ;P0B0出力
172 000D CKSRB1 DAT 1101B          ;P0B1出力
173 000B CKSRB2 DAT 1011B          ;P0B2出力
174 0007 CKSRB3 DAT 0111B          ;P0B3出力
175 0000 CKSRC0 DAT 0000B          ;P0C0出力
176 0000 CALLLOW DAT 0000B
177 000F CALLHI DAT 1111B
178 0001 COCOHI DAT 0001B
179
180 ;+++      キー・リターン値          +++
181          → キー・コード (下位4ビット)          +++
182 000E CKRET0 DAT 1110B          ;P0A0入力
183 000D CKRET1 DAT 1101B          ;P0A1入力
184 000B CKRET2 DAT 1011B          ;P0A2入力
185 0007 CKRET3 DAT 0111B          ;P0A3入力
186
187 ;+++      キー・コード (その他)          +++
188 00F0 CKMLT DAT 0F0H          ;多重押し
189 00FF CKOFF DAT 0FFH          ;キー・オフ
190
191          EJECT
    
```

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:22:42 12/20/93 PAGE 01-006

PROG =

SOURCE = CALCDEF.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
192 ;*****
193 ;* R A Mデータ *
194 ;*****
195 ;+++ 数値キー・カウンタ +++
196 0000 CNUM DAT 0H ;数値キー入力数: 0
197 0007 CFLNUM DAT 7H ; " : 7
198
199 ;+++ 小数点表示位置エリア +++
200 0007 CPLINI DAT 7H ;小数点表示位置エリア初期値
201
202 ;+++ 表示データ指数エリア +++
203 0008 CEXPINI DAT 08H ;表示データ指数エリア初期値
204
205 ;+++ オーバフロー判定 +++
206 00FB CEXPMIN DAT 0FBH ;指数の範囲: -5~7
207 0007 CEXPMAX DAT 07H
208
209 ;+++ 符号データ +++
210 0000 CSINOFF DAT 0000B ;符号表示なし
211 0001 CSINMN DAT 0001B ;マイナス符号表示
212 0001 CCHSIN DAT 0001B ;符号反転
213
214 EJECT
    
```

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:22:42 12/20/93 PAGE 01-007

PROG =

SOURCE = CALCDEF.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
215 ;+++ 表示データ +++
216 0000 CZERO DAT 0H ; [ 0 ]
217 0001 CONE DAT 1H ; [ 1 ]
218 0002 CTWO DAT 2H ; [ 2 ]
219 0003 CTHREE DAT 3H ; [ 3 ]
220 0004 CFOUR DAT 4H ; [ 4 ]
221 0005 CFIVE DAT 5H ; [ 5 ]
222 0006 CSIX DAT 6H ; [ 6 ]
223 0007 CSEVEN DAT 7H ; [ 7 ]
224 0008 CEIGHT DAT 8H ; [ 8 ]
225 0009 CNINE DAT 9H ; [ 9 ]
226 000A CSPACE DAT 0AH ;スペース
227 000B CMINUS DAT 0BH ;[-]
228 000C CERRDP DAT 0CH ;[E]
229
230 ;+++ チャタリング +++
231 0002 CCHATS DAT 2H ;チャタリング開始
232 0000 CCHATE DAT 0H ;チャタリング終了
233
234 ;+++ オートパワーオフ・タイマ +++
235 464F CSTSTP DAT 464FH ;3分間タイマ(10ms×18,000)
236 ; →STOPモード設定
237
238 ;+++ LCDセグメント・データ +++
239 0001 CDPON DAT 0001B ;小数点表示
240
241 ;+++ 演算子 +++
242 0001 CADD DAT 0001B ; [+ ]
243 0002 CSUB DAT 0010B ; [- ]
244 0004 CMUL DAT 0100B ; [ × ]
245 0008 CDIV DAT 1000B ; [ ÷ ]
246
247 ;+++ モード +++
248 0001 CRIGHT DAT 0001B ;右項モード
249 0002 COPSEL DAT 0010B ;演算子選択モード
250 0004 CLEFT DAT 0100B ;左項モード
251 0008 CERROR DAT 1000B ;エラー・モード
252
253 EJECT
    
```

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:22:42 12/20/93 PAGE 01-008

PROG =

SOURCE = CALCDEF.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
254 ;*****
255 ;* 浮動小数点演算部 *
256 ;*****
257 CGBO DAT 0H ;汎用レジスタ BANK 0
258 CGB1 DAT 1H ;汎用レジスタ BANK 1
259 CROWG DAT 0H ;汎用レジスタ ロウ・アドレス 0H
260 CROWX DAT 0H ;汎用レジスタ ロウ・アドレス 0H
261 CROWY DAT 2H ;汎用レジスタ ロウ・アドレス 10H
262 CROWW DAT 4H ;汎用レジスタ ロウ・アドレス 20H
263 CROWXBCD DAT 1H ;汎用レジスタ ロウ・アドレス 0H
264 ; BCDフラグ セット
265 CIXMRY DAT 1H ;インデクス修飾 (REGY)
266 CIXMRW DAT 2H ;インデクス修飾 (REGW)
267 CIXMB1 DAT 8H ;インデクス修飾 (BANK1)
268 CIXLMANT DAT 6H ;インデクス修飾 (仮数部 LSD)
269 CIXLEXP DAT 4H ;インデクス修飾 (指数部)
270 CIXLSIGN DAT 3H ;インデクス修飾 (符号部)
271 CMDRX DAT 0H ;インデクス修飾されるカラム・アドレス 0H
272 CMDRY DAT 10H ;インデクス修飾されるカラム・アドレス 10H
273 CMDRW DAT 20H ;インデクス修飾されるカラム・アドレス 20H
274 CJUDGE DAT 1000B ;指数部値の正負判定
275 CCPTURN DAT 1111B ;ビット反転マスク
276 CSUBTURN DAT 0001B ;減数符号反転
277 CRKTURN DAT 0001B ;REGX 符号反転
278 CEXDIF DAT 0AH ;指数差判定
279 CSIGNCK DAT 0001B ;符号部チェック
280 CMANTCNT DAT 0AH ;仮数部10桁カウンタ
281 CEXOVER DAT 0100B ;指数部値 オーバフロー
282 CEXUNDER DAT 1100B ;指数部値 アンダフロー
283
284 ENDP
285
286 EJECT
    
```

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:22:42 12/20/93 PAGE 01-009

PROG =

SOURCE = CALCDEF.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
  287 ;*****
  288 ;*
  289 ;*          マスク・オプション定義          ;*
  290 ;*
  291 ;*****
  292 OPTION
  293 OPTRES RESPLUP ;リセット端子: プルアップ抵抗内蔵
+ 12 0001 1
  294 OPTCK USEX,NOXT ;システム・クロック: メイン・クロック
+ 24 0006 1
  295 ENDOP
  296
  297 END
    
```

TOTAL ERRORS = 0

TOTAL WARNINGS = 0

END OF LIST

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:23:25 12/20/93 PAGE 02-001

PROG =

SOURCE = CALC1.ASM

```
E STNO LOC. OBJ. M I SOURCE STATEMENT
  1      :*****
  2      :*
  3      :* USER NAME : NEC CORPORATION *
  4      :*
  5      :* SYSTEM NAME : 17K APPLICATION (CALCULATOR) *
  6      :*
  7      :* CPU : uPD17201A *
  8      :*
  9      :* LAST UPDATE : '93/12/20 11:00 *
 10     :*
 11     :*****
 12     :*****
 13     :*
 14     :* FILE NAME : CALC1.ASM *
 15     :*
 16     :* INCLUDE 7 ROUTINES: *
 17     :* IRESET : RESET ROUTINE *
 18     :* MMAIN : MAIN ROUTINE *
 19     :* MKSCAN : KEY SCAN *
 20     :* SKRET : KEY RETURN *
 21     :* SMKIPC : MAKE INPUT CODE *
 22     :* MAPOFF : AUTO POWER OFF *
 23     :* COUNTER CONTROL *
 24     :* MSTOP : STOP MODE PROCESS *
 25     :*
 26     :*****
 27
 28     EJECT
```



AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:23:25 12/20/93 PAGE 02-002

PROG =

SOURCE = CALC1.ASM

E STNO LOC. OBJ. M I SOURCE STATEMENT

```

29 ;*****
30 ;*          外部参照          *
31 ;*****
32 EXTRN LAB: MKBRN          ;キー・ブランチ処理
33 EXTRN LAB: MOPBRN        ;演算分岐処理/エラー処理
34 EXTRN LAB: SRAMCR        ;RAMオール・クリア処理
35 EXTRN LAB: SDPINI        ;表示データ・エリア・イニシャライズ処理
36 EXTRN LAB: SDISP         ;表示データ出力処理
37
38 EXTRN MEM: PCC           ;システム・クロック制御レジスタ
39 EXTRN MEM: WTC           ;時計用タイマ/ウォッチドッグ・タイマ制御レジスタ
40 EXTRN MEM: ADM           ;A/Dコンバータ動作モード/端子選択
41 EXTRN MEM: SIC           ;シリアル・インタフェース制御レジスタ
42 EXTRN MEM: PSL           ;ポート/タイマ出力, LED出力, シリアル・
43 ;                       ;インタフェース入力/出力制御レジスタ
44 EXTRN MEM: PMO           ;ビットI/O入力/出力制御レジスタ
45 EXTRN MEM: IPF           ;割り込み許可フラグ
46 EXTRN MEM: LCDC          ;LCDコントローラ/ドライバ制御レジスタ
47 EXTRN MEM: LCDM          ;表示モード・レジスタ
48 EXTRN MEM: TMCT          ;8ビット・タイマ/カウンタ制御レジスタ
49 EXTRN MEM: PM1           ;グループI/O入力/出力制御レジスタ
50
51 EXTRN MEM: RREG0, RREG1, RREG2, RREG9 ;汎用レジスタ
52 EXTRN MEM: RDEXP         ;表示データ指数エリア
53 EXTRN MEM: RKCODH, RKCODL ;キー・コード・エリア
54 EXTRN MEM: RCHCODH, RCHCODL ;チャタリング・コード・エリア
55 EXTRN MEM: RIPCODH, RIPCODL ;インプット・コード・エリア
56 EXTRN MEM: RCHATC        ;チャタリング・カウンタ
57 EXTRN MEM: RSTOP3, RSTOP2 ;オートパワーオフ・タイマ
58 EXTRN MEM: RSTOP1, RSTOP0
59 EXTRN MEM: RMODE         ;モード・エリア
60 EXTRN MEM: POA, POB, POC, POD, P1A ;ポート・レジスタ
61
62 EXTRN FLG: FSTOP         ;ストップ・モード・フラグ
63 EXTRN FLG: FKEYREQ       ;キー処理要求フラグ
64 EXTRN FLG: FMULTI        ;多重押しフラグ
65 EXTRN FLG: FOPREQ        ;演算要求フラグ
66 EXTRN FLG: FSTPREQ       ;STOPモード処理要求フラグ
67
68 EXTRN DAT: CKSRB3, CKSRC0 ;ポート出力データ
69 EXTRN DAT: CALLOW, CALLHI, COCOHI
70 EXTRN DAT: CKRET0, CKRET1, CKRET2, CKRET3 ;ポート入力データ
71 EXTRN DAT: CKMLT, CKOFF ;キー・コード
72 EXTRN DAT: CRLHLT        ;8ビット・モジュロ・
73 EXTRN DAT: CRLSTP        ;レジスタ・データ
74 EXTRN DAT: CEXPINI       ;表示データ指数エリア初期値
75 EXTRN DAT: CCHATC, CCHATE ;チャタリング開始/終了カウント値
76 EXTRN DAT: CSTSTP        ;オートパワーオフ・カウント値
77 EXTRN DAT: CLEFT         ;左項モード
78
79 EJECT
    
```

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:23:25 12/20/93 PAGE 02-003

PROG =

SOURCE = CALC1.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
 80
 81
 82
 83
 84
 85 0000          ORG      000H
 86 0000 0C005   BR       IRESET
 87
 88              ;+++   割り込みベクタ・アドレス      +++
 89 0001 0C005   BR       IRESET
 90 0002 0C005   BR       IRESET
 91 0003 0C005   BR       IRESET
 92 0004 0C005   BR       IRESET
 93
 94              IRESET:
 95 0005 071F0   DI
 96 0006 1D785   MOV      WR, #5H
 97 0007 07021   POKE     SP, WR
 98 0008 1D783   MOV      WR, #3H          ;システム・クロック： メイン・クロック
 99 0009 07022   POKE     PCC, WR
100
101
102
103
104              ;*****
105              ;*      ポート初期設定      *
106              ;*****
107              BANK1
+ 1 000A 1D791   1      MOV      BANK, #01H
108 000B 1D700   1      MOV      P1A, #CALLLOW AND OFH      ;P1A: 未使用
109
110              BANK0
+ 1 000C 1D790   1      MOV      BANK, #00H
111 000D 1D71F   MOV      POB, #CALLHI AND OFH      ;POB, POC: キー・ソース
112 000E 1D721   MOV      POC, #COCOHI AND OFH      ;      (ロウ・アクティブ)
113 000F 1D730   MOV      POD, #CALLLOW AND OFH      ;POD: 未使用
114 0010 1D78E   MOV      WR, #1110B      ;POB, POC, P1A: 出力
115 0011 07327   POKE     PM1, WR          ;POA: 入力(キー・リターン)
116 0012 1D78F   MOV      WR, #1111B      ;POD.-POD: 出力
117 0013 07227   POKE     PM0, WR
118
119
120              EJECT

```

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:23:25 12/20/93 PAGE 02-004

PROG =

SOURCE = CALC1.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
116 ;*****
117 ;* タイマ初期設定 *
118 ;*****
119 ;+++ 8ビット・タイマ +++
120 0014 1D786 MOV WR,#0110B ;動作クロックの周期: f SYS/256
121 0015 07323 POKE TMCT,WR
122 0016 1D0FC MOV DBFO,#CRLHLT AND OFH ;モジュロ・レジスタ・
123 0017 1D0E9 MOV DBF1,#CRLHLT SHR 4 AND OFH ; リセット(10ms)
124 0018 070A2 PUT TMM,DBF
125
126 ;+++ 時計用タイマ / ウォッチドッグ・タイマ +++
127 0019 1D780 MOV WR,#0000B ;時計用タイマ: 未使用
128 001A 07023 POKE WTC,WR
129
130 ;*****
131 ;* LCDコントローラ/ドライバ初期設定 *
132 ;*****
133 001B 1D781 MOV WR,#0001B ;表示モード: 1/2デューティ
134 001C 07322 POKE LCDM,WR
135 ; MOV WR,#0001B ;フレーム周波数: 512Hz
136 001D 07321 POKE LCDC,WR
137
138 ;*****
139 ;* その他のハードウェアの初期設定 *
140 ;*****
141 ;+++ A/Dコンバータ +++
142 001E 1D780 MOV WR,#0000B ;動作モード:
143 001F 07221 POKE ADM,WR ; スタンバイ状態
144
145 ;+++ シリアル・インタフェース +++
146 ; MOV WR,#0000B ;未使用
147 0020 07223 POKE PSL,WR
148 ; MOV WR,#0000B
149 0021 07222 POKE SIC,WR
150
151 ;*****
152 ;* 割り込み初期設定 *
153 ;*****
154 ; MOV WR,#0000B ;すべての割り込みを禁止
155 0022 0722F POKE IPF,WR
156
157 EJECT
    
```

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:23:25 12/20/93 PAGE 02-005

PROG =

SOURCE = CALC1.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
158 ;*****
159 ;* R A M初期設定 *
160 ;*****
161 0023 ID7D0 MOV RPH, #0000B ;汎用レジスタ:
162 0024 ID7E0 MOV RPL, #0000B ; 0. 0 0 H-0. 0 F H
163 0025 EXTRN CALL SRAMCR ;R A Mオール・クリア
164
165 ;+++ オートパワーオフ・タイマ・リセット +++
166 0026 ID3BF MOV RSTOP0, #CSTSTP AND 0FH ;10ms*18000 カウント
167 0027 ID3A4 MOV RSTOP1, #CSTSTP SHR 4 AND 0FH
168 0028 ID396 MOV RSTOP2, #CSTSTP SHR 8 AND 0FH
169 0029 ID384 MOV RSTOP3, #CSTSTP SHR 12 AND 0FH
170
171 002A ID148 MOV RDEXP, #CEXPINI AND 0FH
172 002B ID31F MOV RKCODL, #CKOFF AND 0FH ;キー・コード,
173 002C ID30F MOV RKCODH, #CKOFF SHR 4 AND 0FH ; チャタリング・コード
174 002D ID33F MOV RCHCODL, #CKOFF AND 0FH ; ←キー・オフのコード
175 002E ID32F MOV RCHCODH, #CKOFF SHR 4 AND 0FH
176 002F ID644 MOV RMODE, #CLEFT AND 0FH
177
178 0030 EXTRN CALL SDPINI ;表示データ・エリア・イニシャライズ
179 0031 EXTRN CALL SDISP ;表示データ出力
180 SETI LCDEN ;L C D表示オン
+ 1 0032 07331 1 PEEK WR, . MF. LCDEN SHR 4
+ 2 0033 16788 1 OR WR, #. DF. LCDEN AND 0FH
+ 3 0034 07321 1 POKE . MF. LCDEN SHR 4, WR
181 SETI TMEN ; 8ビット・タイマ・カウント開始
+ 1 0035 07333 1 PEEK WR, . MF. TMEN SHR 4
+ 2 0036 16788 1 OR WR, #. DF. TMEN AND 0FH
+ 3 0037 07323 1 POKE . MF. TMEN SHR 4, WR
182
183 EJECT
    
```

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:23:25 12/20/93 PAGE 02-006

PROG =

SOURCE = CALC1.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
184 ;*****
185 ;*
186 ;*          メイン処理
187 ;*
188 ;*****
189 MMAIN:
190          CLR1  IRQTM
+ 1 0038 1D780 1      MOV    WR,#0000B
+ 2 0039 0732E 1      POKE   .MF.IRQTM SHR 4,WR
191          LMN200:
192          HALT  2H          ;10msごとに解除
193          SKT1  IRQTM
+ 1 003B 0733E 1      PEEK   WR,.MF.IRQTM SHR 4
+ 2 003C 1E781 1      SKT    WR,#.DF.IRQTM AND OFH
194          BR    LMN200
195          SET1  WDTRES      ;ウォッチドッグ・タイマをリセット
+ 1 003E 07033 1      PEEK   WR,.MF.WDTRES SHR 4
+ 2 003F 16788 1      OR     WR,#.DF.WDTRES AND OFH
+ 3 0040 07023 1      POKE   .MF.WDTRES SHR 4,WR
196          CALL MKSCAN      ;キー・スキャン
197          SKT1  FKEYREQ     ;チャタリング終了か?
+ 1 0042 1E691 1      SKT    .MF.FKEYREQ SHR 4,#.DF.FKEYREQ AND OFH
198          BR    LMN600      ;No →オートパワーオフ・
                               ;      タイマ管理
199
200
201 ;*****
202 ;      動作停止モード・チェック
203 ;*****
204          CLR1  FKEYREQ
+ 1 0044 1469E 1      AND    .MF.FKEYREQ SHR 4,#.DF.(NOT FKEYREQ AND OFH)
205          SKNE  RKCODL,#CKOFF AND OFH      ;オフ・チャタリング
206          SKE   RKCODH,#CKOFF SHR 4 AND OFH ;      終了か?
207          BR    LMN400      ;No: 1個押し
208          SKF1  FSTOP      ;動作停止モードか?
+ 1 0048 1F688 1      SKF    .MF.FSTOP SHR 4,#.DF.FSTOP AND OFH
209          BR    LMN800      ;Yes →STOPモード再設定
210          BR    LMN600      ;No →オートパワーオフ・
                               ;      タイマ管理
211
212
213          LMN400:
214          CALL  MKBRN      ;キー・ブランチ
215          SKF1  FOPREQ     ;演算可能か?
+ 1 004C 1F694 1      SKF    .MF.FOPREQ SHR 4,#.DF.FOPREQ AND OFH
216          CALL  MOPBRN      ;Yes: 演算分岐
217          LMN600:
218          CALL  MAPOFF     ;オートパワーオフ・
                               ;      タイマ管理
219          SKF1  FSTPREQ    ;タイム・アウトか?
+ 1 004F 1F698 1      SKF    .MF.FSTPREQ SHR 4,#.DF.FSTPREQ AND OFH
220          LMN800:
221          CALL  MSTOP      ;Yes →STOPモード設定
222          BR    MMAIN
223          CALL  MMAIN
224
225          EJECT
    
```

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:23:25 12/20/93 PAGE 02-007

PROG =

SOURCE = CALC1.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
226 ;*****
227 ;*
228 ;* キー・スキャン処理 *
229 ;*
230 ;* INPUT : Nothing *
231 ;* OUTPUT : RKCODH - RKCODL (キー・コード)*
232 ;* : FKEYREQ (キー処理要求フラグ) *
233 ;*
234 ;*****
235 MKSCAN:
236 0052 1D35F MOV RIPCODL, #CKOFF AND 0FH ;インプット・コードを
237 0053 1D34F MOV RIPCODH, #CKOFF SHR 4 AND 0FH ; リセット
238 0054 1D007 MOV RREG0, #CKSRB3 AND 0FH
239 0055 1D020 MOV RREG2, #0H ;キー数カウンタをリセット
240
241 ;*****
242 ;* P O Bs - P O Bs 出力 *
243 ;*****
244 LKS100:
245 0056 18710 ST POB, RREG0 ;P O B ←キー・スキャン信号
246 0057 074F0 NOP ;12us ウェイト
247 0058 074F0 NOP
248 0059 074F0 NOP
249 005A 08709 LD RREG9, POA ;キー・リターン値を退避
250 005B 1D71F MOV POB, #CALLHI AND 0FH
251 005C 1C094 CALL SKRET ;キー・リターン判定
252 1 SKF1 FMULTI ;多重押しか?
+ 1 005D 1F692 SKF .MF, FMULTI SHR 4, #.DF, FMULTI AND 0FH
253 005E 0C06D BR LKS200 ;Y e s
254 1 SET1 CY
+ 1 005F 167F4 OR .MF, CY SHR 4, #.DF, CY AND 0FH
255 0060 07070 RORC RREG0
256 1 SKF1 CY ;P O Bs - P O Bsが終了したか?
+ 1 0061 1F7F4 SKF .MF, CY SHR 4, #.DF, CY AND 0FH
257 0062 0C056 BR LKS100 ;N o
258
259 EJECT
    
```

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:23:25 12/20/93 PAGE 02-008

PROG =

SOURCE = CALC1.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
 260 ;*****
 261 ;* P O C。出力 *
 262 ;*****
 263 0063 1D000 MOV RREG0,#CKSRC0 AND OFH
 264 0064 18720 ST POC,RREG0 ;P O C ←キー・スキャン信号
 265 0065 074F0 NOP ;12us ウェイト
 266 0066 074F0 NOP
 267 0067 074F0 NOP
 268 0068 08709 LD RREG9,POA ;キー・リターン値を退避
 269 0069 1D721 MOV POC,#COCOHI AND OFH
 270 006A 1C094 CALL SKRET ;キー・リターン値判定
 271 SKTI FMULTI ;多重押しか?
+ 1 006B 1E692 SKT .MF.FMULTI SHR 4,#.DF.FMULTI AND OFH
 272 006C 0C070 BR LKS300 ;N o
 273
 274 ;*****
 275 ;* 多重押し *
 276 ;*****
 277 LKS200:
+ 1 006D 1469D CLR1 FMULTI
 279 006E 1D350 AND .MF.FMULTI SHR 4,#.DF.(NOT FMULTI AND OFH)
 280 006F 1D34F MOV RIPCODL,#CKMLT AND OFH ;インプット・コード
 281 MOV RIPCODH,#CKMLT SHR 4 AND OFH ; ←多重押しのコード
 282 ;*****
 283 ;* キー状態判定 *
 284 ;*****
 285 LKS300:
 286 0070 08320 LD RREG0,RCHCODH ;インプット・コードが変化したか?
 287 0071 05340 XOR RREG0,RIPCODH
 288 0072 09000 SKE RREG0,#0H
 289 0073 0C078 BR LKS400
 290 0074 08330 LD RREG0,RCHCODL
 291 0075 05350 XOR RREG0,RIPCODL
 292 0076 0B000 SKNE RREG0,#0H
 293 0077 0C07E BR LKS500 ;N o →チャタリング
 294 LKS400:
 295 0078 08340 LD RREG0,RIPCODH ;チャタリング・コード
 296 0079 18320 ST RCHCODH,RREG0 ; ←インプット・コード
 297 007A 08350 LD RREG0,RIPCODL
 298 007B 18330 ST RCHCODL,RREG0
 299 007C 1D362 MOV RCHATC,#CCHATS AND OFH ;チャタリング・カウンタ・リセット
 300 007D 0C093 BR LKS999 ;R E T
 301
 302 EJECT

```

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:23:25 12/20/93 PAGE 02-009

PROG =

SOURCE = CALC1.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
303 ;*****
304 ;* チャタリング *
305 ;*****
306 LKS500:
307 007E 0B360 SKNE RCHATC, #CCHATE AND OFH ;チャタリングが終了済みか?
308 007F 0C093 BR LKS999 ;Y e s →R E T
309 0080 11361 SUB RCHATC, #1H
310 0081 09360 SKE RCHATC, #CCHATE AND OFH ;チャタリングが終了したか?
311 0082 0C093 BR LKS999 ;N o →R E T
312
313 ;*****
314 ;* チャタリング終了 *
315 ;*****
316 0083 0B330 SKNE RCHCODL, #CKMLT AND OFH ;多重押しか?
317 0084 0932F SKE RCHCODH, #CKMLT SHR 4 AND OFH
318 0085 0C087 BR LKS600 ;N o
319 0086 0C08F BR LKS900 ;Y e s
320 LKS600:
321 0087 0B33F SKNE RCHCODL, #CKOFF AND OFH ;オフ・チャタリング
322 0088 0932F SKE RCHCODH, #CKOFF SHR 4 AND OFH ; 終了か?
323 0089 0C08B BR LKS700 ;N o : 1個押し
324 008A 0C08E BR LKS800 ;Y e s
325
326 LKS700:
327 008B 0B31F SKNE RKCODL, #CKOFF AND OFH ;←1個押し
328 008C 0930F SKE RKCODH, #CKOFF SHR 4 AND OFH ;キー・オフ→1個押しか?
329 008D 0C093 BR LKS999 ;N o →R E T
330
331 LKS800:
332 SET1 FKEYREQ ;←キー・オフ
+ 1 008E 16691 1 OR .MF. FKEYREQ SHR 4, #. DF. FKEYREQ AND OFH ;キー処理要求セット
333
334 LKS900:
335 008F 08320 LD RREGO, RCHCODH ;←多重押し
336 0090 18300 ST RKCODH, RREGO ;キー・コード
337 0091 08330 LD RREGO, RCHCODL ;
338 0092 18310 ST RKCODL, RREGO ; ←チャタリング・コード
339
340 LKS999:
341 0093 070E0 RET
342
343 EJECT
    
```



AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:23:25 12/20/93 PAGE 02-010

PROG =

SOURCE = CALC1.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
344 ;*****
345 ;*
346 ;* キー・リターン値判定処理 *
347 ;* *
348 ;* INPUT : RREG9(キー・リターン値) *
349 ;* OUTPUT : FMULTI (多重押しフラグ) *
350 ;* *
351 ;*****
352 SKRET:
353 ;+++
354 0094 08091 LD RREG1,RREG9 +++
355 0095 1601E OR RREG1,#CKRETO AND OFH
356 0096 0B01F SKNE RREG1,#CALLHI AND OFH ;P0A0がロウか?
357 0097 0C099 BR JKR100 ;No
358 0098 1C0AB CALL SMKICD ;Yes: インプット・コード生成
359
360 ;+++
361 JKR100: P0A1チェック +++
362 0099 08091 LD RREG1,RREG9
363 009A 1601D OR RREG1,#CKRET1 AND OFH
364 009B 0B01F SKNE RREG1,#CALLHI AND OFH ;P0A1がロウか?
365 009C 0C09E BR JKR200 ;No
366 009D 1C0AB CALL SMKICD ;Yes: インプット・コード生成
367
368 ;+++
369 JKR200: P0A2チェック +++
370 009E 08091 LD RREG1,RREG9
371 009F 1601B OR RREG1,#CKRET2 AND OFH
372 00A0 0B01F SKNE RREG1,#CALLHI AND OFH ;P0A2がロウか?
373 00A1 0C0A3 BR JKR300 ;No
374 00A2 1C0AB CALL SMKICD ;Yes: インプット・コード生成
375
376 ;+++
377 JKR300: P0A3チェック +++
378 00A3 08091 LD RREG1,RREG9
379 00A4 16017 OR RREG1,#CKRET3 AND OFH
380 00A5 0B01F SKNE RREG1,#CALLHI AND OFH ;P0A3がロウか?
381 00A6 0C0A8 BR JKR400 ;No
382 00A7 1C0AB CALL SMKICD ;Yes: インプット・コード生成
383
384 ;+++
385 JKR400: 多重押しチェック +++
386 00A8 1B022 SKLT RREG2,#2H ;多重押しか?
387 00A9 16692 SETI FMULTI ;Yes
+ 1 00A9 16692 OR .MF.FMULTI SHR 4,#.DF.FMULTI AND OFH
388
389 00AA 070E0 RET
390
391 EJECT
    
```

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:23:25 12/20/93 PAGE 02-011

PROG =

SOURCE = CALC1.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
392 ;*****
393 ;*
394 ;*      インプット・コード生成処理      *
395 ;*
396 ;*      INPUT   : RREG0 (キー・スキャン信号) *
397 ;*                : RREG9 (キー・リターン値) *
398 ;*      OUTPUT  : RREG2 (キー数カウンタ) *
399 ;*                : RIPC0DH - RIPC0DL *
400 ;*                  (インプット・コード) *
401 ;*
402 ;*****
403 SMK1CD:
404 00AB 18340      ST      RIPC0DH, RREG0      ;インプット・コード生成
405 00AC 18359      ST      RIPC0DL, RREG9
406 00AD 10021      ADD     RREG2, #1H      ;キー数カウント
407
408 00AE 070E0      RET
409
410                EJECT
    
```

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:23:25 12/20/93 PAGE 02-012

PROG =

SOURCE = CALC1.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
411 ;*****
412 ;*
413 ;* オートパワーオフ・タイマ管理 *
414 ;*
415 ;* INPUT : RKCODH - RKCODL (キー・コード)*
416 ;* OUTPUT : RSTOP3 - RSTOP0 *
417 ;* (オートパワーオフ・タイマ) *
418 ;* : FSTPREQ (STOPモード処理) *
419 ;* 要求フラグ) *
420 ;*
421 ;*****
422 MAPOFF:
423 BANK0
+ 1 00AF 1D790 1 MOV BANK.#00H
424 00B0 0B31F 1 SKNE RKCODL,#CKOFF AND OFH ;キー・オフか?
425 00B1 0930F SKE RKCODH,#CKOFF SHR 4 AND OFH
426 00B2 0C0BA BR LAP200 ;No
427
428 ;*****
429 ;* オートパワーオフ・タイマ・カウント *
430 ;*****
431 00B3 113B1 SUB RSTOP0,#1H
432 00B4 133A0 SUBC RSTOP1,#0H
433 00B5 13390 SUBC RSTOP2,#0H
434 00B6 13380 SUBC RSTOP3,#0H
435 SKF1 CY ;3分間タイム・アウトか?
+ 1 00B7 1F7F4 1 SKF .MF.CY SHR 4,#.DF.CY AND OFH
436 1 SET1 FSTPREQ ;Yes
+ 1 00B8 16698 1 OR .MF.FSTPREQ SHR 4,#.DF.FSTPREQ AND OFH
437 00B9 0C0BE 1 BR LAP400 ;RET
438
439 ;*****
440 ;* オートパワーオフ・タイマ・リセット *
441 ;*****
442 LAP200:
443 00BA 1D3BF MOV RSTOP0,#CSTSTP AND OFH
444 00BB 1D3A4 MOV RSTOP1,#CSTSTP SHR 4 AND OFH
445 00BC 1D396 MOV RSTOP2,#CSTSTP SHR 8 AND OFH
446 00BD 1D384 MOV RSTOP3,#CSTSTP SHR 12 AND OFH
447
448 LAP400:
449 00BE 070E0 RET
450
451 EJECT
    
```

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:23:25 12/20/93 PAGE 02-013

PROG =

SOURCE = CALC1.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
452 ;*****
453 ;*
454 ;* S T O Pモード処理 *
455 ;* *
456 ;* INPUT : Nothing *
457 ;* OUTPUT : FSTOP (動作停止モード・フラグ)*
458 ;* : FSTPREQ (S T O Pモード処理 *
459 ;* : 要求フラグ) *
460 ;* *
461 ;*****
462 ;*****
463 ;* S T O Pモード設定 *
464 ;*****
465 MSTOP:
466 CLR1 FSTPREQ
+ 1 00BF 14697 1 AND .MF.FSTPREQ SHR 4, #.DF. (NOT FSTPREQ AND OFH)
467 CLR1 LCDEN ; L C D表示オフ
+ 1 00C0 07331 1 PEEK WR, .MF.LCDEN SHR 4
+ 2 00C1 14787 1 AND WR, #.DF. (NOT LCDEN) AND OFH
+ 3 00C2 07321 1 POKE .MF.LCDEN SHR 4, WR
468 CLR1 TMEN ; 8ビット・タイマ・カウント停止
+ 1 00C3 07333 1 PEEK WR, .MF.TMEN SHR 4
+ 2 00C4 14787 1 AND WR, #.DF. (NOT TMEN) AND OFH
+ 3 00C5 07323 1 POKE .MF.TMEN SHR 4, WR
469 SETI FSTOP
+ 1 00C6 16688 1 OR .MF.FSTOP SHR 4, #.DF.FSTOP AND OFH
470 MOV POB, #CALLLOW AND OFH
471 MOV POC, #CALLLOW AND OFH
472 MOV DBF0, #CRLSTP AND OFH
473 MOV DBF1, #CRLSTP SHR 4 AND OFH
474 PUT TMM, DBF
475 NOP
476 STOP 8H ; S T O Pモード設定
477
478 EJECT
    
```

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:23:25 12/20/93 PAGE 02-014

PROG =

SOURCE = CALC1.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
479 ;*****
480 ;*          S T O Pモード解除          *
481 ;*****
482          CLR1      TMEN                ; 8ビット・タイマ・カウント停止
+ 1 00CE 07333 1      PEEK      WR, .MF. TMEN SHR 4
+ 2 00CF 14787 1      AND       WR, #. DF. (NOT TMEN) AND 0FH
+ 3 00D0 07323 1      POKE      .MF. TMEN SHR 4, WR
483 00D1 1D71F        MOV      POB, #CALLHI AND 0FH                ; ポート・リセット
484 00D2 1D721        MOV      POC, #COCOHI AND 0FH
485 00D3 1D0FC        MOV      DBF0, #CRLHLT AND 0FH                ; 8ビット・モジュロ・
486 00D4 1D0E9        MOV      DBF1, #CRLHLT SHR 4 AND 0FH        ; レジスタ・リセット
487 00D5 070A2        PUT      TMM, DBF
488          SET1      TMEN                ; 8ビット・タイマ・カウント開始
+ 1 00D6 07333 1      PEEK      WR, .MF. TMEN SHR 4
+ 2 00D7 16788 1      OR       WR, #. DF. TMEN AND 0FH
+ 3 00D8 07323 1      POKE      .MF. TMEN SHR 4, WR
489 00D9 070E0        RET
490
491          END
    
```

TOTAL ERRORS = 0

TOTAL WARNINGS = 0

END OF LIST

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:24:27 12/20/93 PAGE 03-001

PROG =

SOURCE = CALC2.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
  1      ;*****
  2      ;*
  3      ;* USER NAME : NEC CORPORATION *
  4      ;*
  5      ;* SYSTEM NAME : 17K APPLICATION (CALCULATOR) *
  6      ;*
  7      ;* CPU : uPD17201A *
  8      ;*
  9      ;* LAST UPDATE : '93/12/20 11:00 *
 10     ;*
 11     ;*****
 12     ;*****
 13     ;*
 14     ;* FILE NAME : CALC2.ASM *
 15     ;*
 16     ;* INCLUDE 9 ROUTINES: *
 17     ;* MKBRN : KEY BRANCH *
 18     ;* KALLCR : ALL CLEAR KEY PROCESS *
 19     ;* KCLR : CLEAR KEY PROCESS *
 20     ;* KNUMPT : NUMBER AND POINT KEY *
 21     ;* PROCESS *
 22     ;* KSIGN : SIGN KEY PROCESS *
 23     ;* KOPE : OPERATOR KEY PROCESS *
 24     ;* KPER : PERCENT KEY PROCESS *
 25     ;* KEQU : EQUAL KEY PROCESS *
 26     ;* MOPBRN : OPERATOR BRANCH *
 27     ;* AND ERROR PROCESS *
 28     ;*
 29     ;*****
 30     EJECT
 31

```

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:24:27 12/20/93 PAGE 03-002

PROG =

SOURCE = CALC2.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
32 ;*****
33 ;* 外部参照 *
34 ;*****
35 EXTRN LAB: SRAMCR ;RAMオール・クリア処理
36 EXTRN LAB: SDPINI ;表示データ・エリア・イニシャライズ処理
37 EXTRN LAB: SUSHFD ;表示データ・エリア・アップ・シフト処理
38 EXTRN LAB: STRAN ;データ転送処理
39 EXTRN LAB: SFIX ;演算結果変換処理
40 EXTRN LAB: STRNDY ;表示データ変換処理
41 EXTRN LAB: SDISP ;表示データ出力処理
42 EXTRN LAB: SRXCLR
43 EXTRN LAB: SRYCLR
44 EXTRN LAB: SFPADD
45 EXTRN LAB: SFPSUB
46 EXTRN LAB: SFPMULT
47 EXTRN LAB: SFPDIV
48
49 EXTRN MEM: RREG0, RREG1, RREG9 ;汎用レジスタ
50 EXTRN MEM: RNUMC ;数値キー・カウンタ
51 EXTRN MEM: RPTLOC ;小数点表示位置エリア
52 EXTRN MEM: RSINLOC ;符号表示位置エリア
53 EXTRN MEM: RDEXP ;表示データ指数エリア
54 EXTRN MEM: RDLSD ;表示データ・エリア
55 EXTRN MEM: RZLSD ;退避用レジスタ
56 EXTRN MEM: RKCODH, RKCODL ;キー・コード・エリア
57 EXTRN MEM: RCHCODH, RCHCODL ;チャタリング・コード・エリア
58 EXTRN MEM: ROPE ;演算子エリア
59 EXTRN MEM: RCOM ;実行演算子エリア
60 EXTRN MEM: RMODE ;モード・エリア
61 EXTRN MEM: RXSIGN ;浮動小数点レジスタ1 符号部
62 EXTRN MEM: RXEXP ;浮動小数点レジスタ1 指数部
63 EXTRN MEM: RYSIGN ;浮動小数点レジスタ2 符号部
64 EXTRN MEM: RYEXP ;浮動小数点レジスタ2 指数部
65 EXTRN MEM: ROPFLG ;演算フラグ
66
67 EXTRN FLG: FPER ;パーセント・フラグ
68 EXTRN FLG: FOPEND ;演算終了フラグ
69 EXTRN FLG: FFALSE ;不正入力フラグ
70 EXTRN FLG: FSTOP ;ストップ・モード・フラグ
71 EXTRN FLG: FOPREQ ;演算要求フラグ
72 EXTRN FLG: FZERO ;演算結果ゼロ・フラグ
73 EXTRN FLG: FDVERR ;除算エラー・フラグ
74
75 EJECT
    
```

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:24:27 12/20/93 PAGE 03-003

PROG =

SOURCE = CALC2.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
76 EXTRN DAT: CKSRB0, CKSRB1, CKSRB2 ;キー・コード (上位4ビット)
77 EXTRN DAT: CKSRB3, CKSRC0
78 EXTRN DAT: CKRET0, CKRET1, CKRET2, CKRET3 ; " (下位4ビット)
79 EXTRN DAT: CSTMPE ;MPEセット
80 EXTRN DAT: CSTBK1 ;MP ←BANK1
81 EXTRN DAT: CNONUM, CFLNUM ;数値キー入力数: 0/7
82 EXTRN DAT: CEXPINI ;表示データ指数エリア初期値
83 EXTRN DAT: CEXPMIN, CEXPMAX ;オーバーフロー判定
84 EXTRN DAT: CCHSIN ;符号データ反転
85 EXTRN DAT: CZERO ; [0] 表示データ
86 EXTRN DAT: CONE ; [1] "
87 EXTRN DAT: CTWO ; [2] "
88 EXTRN DAT: CTHREE ; [3] "
89 EXTRN DAT: CFOUR ; [4] "
90 EXTRN DAT: CFIVE ; [5] "
91 EXTRN DAT: CSIX ; [6] "
92 EXTRN DAT: CSEVEN ; [7] "
93 EXTRN DAT: CEIGHT ; [8] "
94 EXTRN DAT: CNINE ; [9] "
95 EXTRN DAT: CSPACE ;スペース "
96 EXTRN DAT: CERRDP ; [E] "
97 EXTRN DAT: CADD ; [+] 演算子データ
98 EXTRN DAT: CSUB ; [-] "
99 EXTRN DAT: CMUL ; [×] "
100 EXTRN DAT: CDIV ; [÷] "
101 EXTRN DAT: CRIGHT ;右項モード
102 EXTRN DAT: COPSEL ;演算子選択モード
103 EXTRN DAT: CLEFT ;左項モード
104 EXTRN DAT: CERROR ;エラー・モード
105
106 ;*****
107 ;* 外部定義 *
108 ;*****
109 PUBLIC MKBRN, MOPBRN
110
111 EJECT
    
```



AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:24:27 12/20/93 PAGE 03-004

PROG =

SOURCE = CALC2.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
112 ;*****
113 ;*
114 ;* マクロ定義ファイル読み込み ;*
115 ;* ;*
116 ;*****
117 1 INCLUDE 'PUSHMP.ASM'
+ 1 1
+ 2 1 EXTRN MEM: RREG10,RREG11
+ 3 1
+ 4 1 ;*****
+ 5 1 ;*
+ 6 1 ;* データ・メモリ・ロウ・アドレス・ ;*
+ 7 1 ;* ポインタ退避マクロ定義 ;*
+ 8 1 ;*
+ 9 1 ;*****
+ 10 1 APUSHMP MACRO
+ 11 1 LD RREG10, MPH
+ 12 1 LD RREG11, MPL
+ 13 1 ENDM
+ 14 1
+ 15 1 ;*****
+ 16 1 ;*
+ 17 1 ;* データ・メモリ・ロウ・アドレス・ ;*
+ 18 1 ;* ポインタ復帰マクロ定義 ;*
+ 19 1 ;*
+ 20 1 ;*****
+ 21 1 APOPMP MACRO
+ 22 1 ST MPH, RREG10
+ 23 1 ST MPL, RREG11
+ 24 1 ENDM
+ 25 1
+ 26 1 EOF
118 EJECT
119

```

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:24:27 12/20/93 PAGE 03-005

PROG =

SOURCE = CALC2.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
120 ;*****
121 ;*
122 ;* キー・ブランチ処理 ;*
123 ;* ;*
124 ;* INPUT : RKCODH - RKCODL (キー・コード)*
125 ;* : RMODE (モード・エリア)*
126 ;* : FSTOP (動作停止モード・フラグ)*
127 ;* OUTPUT : Nothing ;*
128 ;* ;*
129 ;*****
130 MKBRN:
131 OODA OB300 SKNE RKCODH, #CKSRC0 AND OFH ; [A C] キーが押されたか?
132 OODB OB317 SKE RKCODL, #CKRET3 AND OFH
133 OODC OCODE BR LKD050 ;N o
134 OODD OCOF1 BR LKD250 ;Y e s
135
136 LKD050:
137 SKT1 FSTOP ;動作停止モードか?
+ 1 00DE 1E688 1 .MF. FSTOP SHR 4, #. DF. FSTOP AND OFH
138 OODF OB648 SKNE RMODE, #CERROR AND OFH ;エラー・モードか?
139 OOE0 OCOFE BR LKD900 ;Y e s →R E T
140
141 OOE1 09300 SKE RKCODH, #CKSRC0 AND OFH
142 OOE2 OCOE6 BR LKD100
143 OOE3 OB31E SKNE RKCODL, #CKRETO AND OFH ; [%] キーが押されたか?
144 OOE4 OCOFB BR LKD700 ;Y e s
145 OOE5 OCOF9 BR LKD600 ;N o →演算子キー処理
146 LKD100:
147 OOE6 0930E SKE RKCODH, #CKSRB0 AND OFH
148 OOE7 OCOED BR LKD200
149 OOE8 OB31E SKNE RKCODL, #CKRETO AND OFH ; [=] キーが押されたか?
150 OOE9 OCOFD BR LKD800 ;Y e s
151 OOE A OB317 SKNE RKCODL, #CKRET3 AND OFH ; [C] キーが押されたか?
152 OOE B OCOF3 BR LKD300 ;Y e s
153 OOE C OCOF9 BR LKD600 ;N o →演算子キー処理
154 LKD200:
155 OOE D OB30D SKNE RKCODH, #CKSRB1 AND OFH ; [±] キーが押されたか?
156 OOE E OB31E SKE RKCODL, #CKRETO AND OFH
157 OOE F OCOF5 BR LKD400 ;N o →数値キー・小数点キー処理
158 OOE F OCOF7 BR LKD500 ;Y e s
159
160 EJECT
    
```

AS17K V1.10 V4 &lt;&lt; D17201A ASSEMBLE LIST &gt;&gt; 11:24:27 12/20/93 PAGE 03-006

PROG =

SOURCE = CALC2.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
161 ;+++ オール・クリア・キー +++
162 LKD250:
163 00F1 1COFF CALL KALLCR
164 00F2 0COFE BR LKD900 ;R E T
165
166 ;+++ クリア・キー +++
167 LKD300:
168 00F3 1C10F CALL KCLR
169 00F4 0COFE BR LKD900 ;R E T
170
171 ;+++ 数値キー・小数点キー +++
172 LKD400:
173 00F5 1C121 CALL KNUMPT
174 00F6 0COFE BR LKD900 ;R E T
175
176 ;+++ 符号キー +++
177 LKD500:
178 00F7 1C171 CALL KSIGN
179 00F8 0COFE BR LKD900 ;R E T
180
181 ;+++ 演算子キー +++
182 LKD600:
183 00F9 1C198 CALL KOPE
184 00FA 0COFE BR LKD900 ;R E T
185
186 ;+++ パーセント・キー +++
187 LKD700:
188 00FB 1C1D0 CALL KPER
189 00FC 0COFE BR LKD900 ;R E T
190
191 ;+++ イコール・キー +++
192 LKD800:
193 00FD 1C1EA CALL KEQU
194
195 LKD900:
196 00FE 070E0 RET
197
198 EJECT

```

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:24:27 12/20/93 PAGE 03-007

PROG =

SOURCE = CALC2.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
199 ;*****
200 ;*
201 ;* オール・クリア・キー処理 *
202 ;* *
203 ;* INPUT : Nothing *
204 ;* OUTPUT : RDEXP (表示データ指数エリア) *
205 ;* : RMODE (モード・エリア) *
206 ;* *
207 ;*****
208 KALLCR:
209 CLR1 LCDEN ;LCD表示オフ
+ 1 00FF 07331 1 PEEK WR,.MF.LCDEN SHR 4
+ 2 0100 14787 1 AND WR,#.DF.(NOT LCDEN) AND OFH
+ 3 0101 07321 1 POKE .MF.LCDEN SHR 4,WR
210 0102 EXTRN CALL SRAMCR ;RAMオール・クリア
211
212 0103 1D148 MOV RDEXP,#CEXPINI AND OFH
213 0104 1D317 MOV RKCODL,#CKRET3 AND OFH ;キー・コード,
214 0105 1D300 MOV RCHODH,#CKSRC0 SHR 4 AND OFH ;チャタリング・コード
215 0106 1D337 MOV RCHODL,#CKRET3 AND OFH ; ← [AC] キーの
216 0107 1D320 MOV RCHODH,#CKSRC0 SHR 4 AND OFH ; コードを復帰
217 0108 1D644 MOV RMODE,#CLEFT AND OFH
218
219 0109 EXTRN CALL SDPINI ;表示データ・エリア・イニシャライズ
220 010A EXTRN CALL SDISP ;表示データ出力
221 SET1 LCDEN ;LCD表示オン
+ 1 010B 07331 1 PEEK WR,.MF.LCDEN SHR 4
+ 2 010C 16788 1 OR WR,#.DF.LCDEN AND OFH
+ 3 010D 07321 1 POKE .MF.LCDEN SHR 4,WR
222
223 010E 070E0 RET
224
225 EJECT
    
```

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:24:27 12/20/93 PAGE 03-008

PROG =

SOURCE = CALC2.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
226 ;*****
227 ;*
228 ;* クリア・キー処理 *
229 ;*
230 ;* INPUT : RMODE (モード・エリア) *
231 ;* : FPER (パーセント・フラグ) *
232 ;* OUTPUT : RNUMC (数値キー・カウンタ) *
233 ;* : RDEXP (表示データ指数エリア) *
234 ;* : RMODE (モード・エリア) *
235 ;* : FPER (パーセント・フラグ) *
236 ;* : FOPEND (演算終了フラグ) *
237 ;* : FFALSE (不正入力フラグ) *
238 ;*
239 ;*****
240 KCLR:
241 SKT1 FPER ; [%] キーが押されたか?
+ 1 010F 1E681 1 SKT .MF.FPER SHR 4, #.DF.FPER AND OFH
242 0110 0C113 BR NCR200 ; N o
243 CLR1 FPER ; Y e s
+ 1 0111 1468E 1 AND .MF.FPER SHR 4, #.DF.(NOT FPER AND OFH)
244 0112 1D644 MOV RMODE, #CLEFT AND OFH ; 左項モード設定
245 NCR200:
246 CLR2 FOPEND, FFALSE
+ 1 0113 14689 1 AND .MF.FOPEND SHR 4, #.DF.(NOT (FOPEND OR FFALSE) AND OFH)
247 0114 1D100 MOV RNUMC, #CNUM AND OFH
248 0115 1D148 MOV RDEXP, #CEXPINI AND OFH
249 0116 0B642 SKNE RMODE, #COPSEL AND OFH ; 演算子選択モードか?
250 0117 1D641 MOV RMODE, #CRIGHT AND OFH ; Y e s : 右項モード設定
251 CLR1 LCDEN
+ 1 0118 07331 1 PEEK WR, .MF.LCDEN SHR 4
+ 2 0119 14787 1 AND WR, #.DF.(NOT LCDEN) AND OFH
+ 3 011A 07321 1 POKE .MF.LCDEN SHR 4, WR
252 011B EXTRN CALL SDPINI
253 011C EXTRN CALL SDISP
254 SET1 LCDEN
+ 1 011D 07331 1 PEEK WR, .MF.LCDEN SHR 4
+ 2 011E 16788 1 OR WR, #.DF.LCDEN AND OFH
+ 3 011F 07321 1 POKE .MF.LCDEN SHR 4, WR
255
256 0120 070E0 RET
257
258 EJECT
    
```

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:24:27 12/20/93 PAGE 03-009

PROG =

SOURCE = CALC2.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
259 ;*****
260 ;*
261 ;* 数值キー・小数点キー処理 *
262 ;*
263 ;* INPUT : RNUMC (数值キー・カウンタ) *
264 ;* : RKCODH - RKCODL (キー・コード)*
265 ;* : RMODE (モード・エリア) *
266 ;* : FPER (パーセント・フラグ) *
267 ;* : FFALSE (不正入力フラグ) *
268 ;* OUTPUT : RNUMC (数值キー・カウンタ) *
269 ;* : RPTLOC (小数点表示位置エリア) *
270 ;* : RDEXP - RDMSD (表示データ) *
271 ;* : RMODE (モード・エリア) *
272 ;* : FPER (パーセント・フラグ) *
273 ;* : FOPEND (演算終了フラグ) *
274 ;* : FFALSE (不正入力フラグ) *
275 ;*
276 ;*****
277 KNUMPT:
278 SKT1 FFALSE
+ 1 0121 1E684 1 SKT .MF.FFALSE SHR 4, #.DF.FFALSE AND OFH
279 0122 0C126 1 BR NNP100
280 CLR1 FFALSE
+ 1 0123 1468B 1 AND .MF.FFALSE SHR 4, #.DF. (NOT FFALSE AND OFH)
281 0124 1D100 MOV RNUMC, #CNONUM AND OFH
282 0125 1D148 MOV RDEXP, #CEXPINI AND OFH
283
NNP100:
284 SKF1 FPER
+ 1 0126 1F681 1 SKF .MF.FPER SHR 4, #.DF.FPER AND OFH
285 0127 1D644 MOV RMODE, #CLEFT AND OFH
286 CLR2 FPER, FOPEND
+ 1 0128 1468C 1 AND .MF.FPER SHR 4, #.DF. (NOT (FPER OR FOPEND) AND OFH)
287
288 0129 0B30B SKNE RKCODH, #CKSRB2 AND OFH ;小数点キーが押されたか?
289 012A 0931E SKE RKCODL, #CKRETO AND OFH
290 012B 0C13E BR NNP300 ;No →数值キー処理
291
292 EJECT
    
```

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:24:27 12/20/93 PAGE 03-010

PROG =

SOURCE = CALC2.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
 293 ;*****
 294 ;*          小数点キ-処理          *
 295 ;*****
 296 012C 09148          SKE  RDEXP,#CEXPINI AND OFH ;小数点キ-がすでに押されたか?
 297 012D 0C170          BR   NNP999          ;Y e s  →R E T
 298 012E 09100          SKE  RNUMC,#CNUM          ;数值キ-が押される前に小数点キ-
 299 012F 0C139          BR   NNP200          ;          が押されたか?
 300 0130 10101          ADD  RNUMC,#1H          ;Y e s
 301                    CLR1  LCDEN
+ 1 0131 07331 1        PEEK WR,.MF.LCDEN SHR 4
+ 2 0132 14787 1        AND  WR,#.DF.(NOT LCDEN) AND OFH
+ 3 0133 07321 1        POKE .MF.LCDEN SHR 4,WR
 302 0134 EXTRN          CALL SDPINI
 303 0135 EXTRN          CALL SDISP
 304                    SET1  LCDEN
+ 1 0136 07331 1        PEEK WR,.MF.LCDEN SHR 4
+ 2 0137 16788 1        OR   WR,#.DF.LCDEN AND OFH
+ 3 0138 07321 1        POKE .MF.LCDEN SHR 4,WR
 305                    NNP200:
 306 0139 08100          LD   RREGO,RNUMC
 307 013A 18140          ST   RDEXP,RREGO
 308 013B 0B642          SKNE RMODE,#COPSEL AND OFH ;演算子選択モードか?
 309 013C 1D641          MOV  RMODE,#CRIGHT AND OFH ;Y e s : 右項モード設定
 310 013D 0C170          BR   NNP999          ;R E T
 311
 312                    EJECT
    
```

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:24:27 12/20/93 PAGE 03-011

PROG =

SOURCE = CALC2.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
313 ;*****
314 ;* 数値キー処理 *
315 ;*****
316 NNP300:
317 013E OB107 SKNE RNUMC,#CFLNUM AND OFH ;数値キーがすでに7回押されたか?
318 013F OC170 BR NNP999 ;Yes →RET
319
320 ;+++ キー・デコード +++
321 0140 0931E SKE RKCODL,#CKRETO AND OFH ;[0]が押されたか?
322 0141 OC144 BR NNP400
323 0142 1D090 MOV RREG9,#CZERO AND OFH ;Yes
324 0143 OC15C BR NNP700
325
326 NNP400:
327 0144 0931D SKE RKCODL,#CKRET1 AND OFH
328 0145 OC14D BR NNP500
329 0146 OB307 SKNE RKCODH,#CKSRB3 AND OFH ;[1]が押されたか?
330 0147 1D091 MOV RREG9,#CONE AND OFH ;Yes
331 0148 OB30B SKNE RKCODH,#CKSRB2 AND OFH ;[2]が押されたか?
332 0149 1D092 MOV RREG9,#CTWO AND OFH ;Yes
333 014A OB30D SKNE RKCODH,#CKSRB1 AND OFH ;[3]が押されたか?
334 014B 1D093 MOV RREG9,#CTHREE AND OFH ;Yes
335 014C OC15C BR NNP700
336
337 NNP500:
338 014D 0931B SKE RKCODL,#CKRET2 AND OFH
339 014E OC156 BR NNP600
340 014F OB307 SKNE RKCODH,#CKSRB3 AND OFH ;[4]が押されたか?
341 0150 1D094 MOV RREG9,#CFOUR AND OFH ;Yes
342 0151 OB30B SKNE RKCODH,#CKSRB2 AND OFH ;[5]が押されたか?
343 0152 1D095 MOV RREG9,#CFIVE AND OFH ;Yes
344 0153 OB30D SKNE RKCODH,#CKSRB1 AND OFH ;[6]が押されたか?
345 0154 1D096 MOV RREG9,#CSIX AND OFH ;Yes
346 0155 OC15C BR NNP700
347
348 NNP600:
349 0156 OB307 SKNE RKCODH,#CKSRB3 AND OFH ;[7]が押されたか?
350 0157 1D097 MOV RREG9,#CSEVEN AND OFH ;Yes
351 0158 OB30B SKNE RKCODH,#CKSRB2 AND OFH ;[8]が押されたか?
352 0159 1D098 MOV RREG9,#CEIGHT AND OFH ;Yes
353 015A OB30D SKNE RKCODH,#CKSRB1 AND OFH ;[9]が押されたか?
354 015B 1D099 MOV RREG9,#CNINE AND OFH ;Yes
355
356 EJECT
    
```



AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:24:27 12/20/93 PAGE 03-012

PROG =

SOURCE = CALC2.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
  357
  358          1          CLR1   LCDEN           ;LCD表示オフ
+   1 015C 07331 1          PEEK   WR,.MF.LCDEN SHR 4
+   2 015D 14787 1          AND    WR,#.DF.(NOT LCDEN) AND OFH
+   3 015E 07321 1          POKE   .MF.LCDEN SHR 4,WR
  359 015F 09644          SKE    RMODE,#CLEFT AND OFH ;左項モードか?
  360 0160 1D641          MOV    RMODE,#CRIGHT AND OFH ;No: 右項モード設定
  361 0161 09100          SKE    RNUMC,#CNONUM AND OFH ;1回目の数値キー入力か?
  362 0162 0C167          BR     NNP800 ;No
  363 0163 EXTRN          CALL   SDPINI ;Yes: 表示データ・エリア
  364                                     ; .イニシャライズ
  365 0164 0B090          SKNE   RREG9,#CZERO AND OFH ;[0]が押されたか?
  366 0165 0C16C          BR     NNP900 ;Yes
  367 0166 1D18A          MOV    RDLS,#CSPACE AND OFH ;No: 表示データの最下位桁
  368                                     ; ←スペース表示データ
  369          NNP800:
  370 0167 09148          SKE    RDEXP,#CEXPINI AND OFH ;小数点キーがすでに押されたか?
  371 0168 11111          SUB    RPTLOC,#1H ;Yes
  372 0169 10101          ADD    RNUMC,#1H
  373 016A EXTRN          CALL   SUSHFD ;表示データ・エリア右シフト
  374 016B 18189          ST     RDLS,RREG9 ;表示データの最下位桁
  375                                     ; ←押された数値の表示データ
  376          NNP900:
  377 016C EXTRN          CALL   SDISP ;表示データ出力
  378          SET1   LCDEN ;LCD表示オン
+   1 016D 07331 1          PEEK   WR,.MF.LCDEN SHR 4
+   2 016E 16788 1          OR     WR,#.DF.LCDEN AND OFH
+   3 016F 07321 1          POKE   .MF.LCDEN SHR 4,WR
  379
  380          NNP999:
  381 0170 070E0          RET
  382
  383          EJECT
    
```

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:24:27 12/20/93 PAGE 03-013

PROG =

SOURCE = CALC2.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
384 ;*****
385 ;*
386 ;* 符号キー処理 *
387 ;*
388 ;* INPUT : RNUMC (数値キー・カウンタ) *
389 ;* : RSINLOC (符号表示位置エリア) *
390 ;* : RMODE (モード・エリア) *
391 ;* : FPER (パーセント・フラグ) *
392 ;* : FOPEND (演算終了フラグ) *
393 ;* : FZERO (演算結果ゼロ・フラグ) *
394 ;* OUTPUT : RSINLOC (符号表示位置エリア) *
395 ;* : RDLS - RDMSD (表示データ) *
396 ;* : RMODE (モード・エリア) *
397 ;* : RASIGN (浮動小数点レジスタ1 *
398 ;* ;符号部) *
399 ;* : FOPEND (演算終了フラグ) *
400 ;*
401 ;*****
402 KSIGN:
403 0171 0B642 SKNE RMODE, #COPSEL AND OFH ;演算子選択モードか?
404 0172 0C197 BR MSG500 ;Yes →RET
405 1 SKT1 FPER ; [%] キーが押されたか?
+ 1 0173 1E681 1 SKT .MF.FPER SHR 4, #.DF.FPER AND OFH
406 0174 0C178 BR NSG100 ;No
407 1 CLR1 FPER ;Yes
+ 1 0175 1468E 1 AND .MF.FPER SHR 4, #.DF.(NOT FPER AND OFH)
408 1 SET1 FOPEND
+ 1 0176 16682 1 OR .MF.FOPEND SHR 4, #.DF.FOPEND AND OFH
409 0177 1D644 MOV RMODE, #CLEFT AND OFH ;左項モード設定
410 NSG100:
411 1 SKF1 FOPEND
+ 1 0178 1F682 1 SKF .MF.FOPEND SHR 4, #.DF.FOPEND AND OFH
412 0179 0C180 BR NSG200
413
414 017A 0B100 SKNE RNUMC, #CNONUM ;表示が [ _____ 0. ] か?
415 017B 0C197 BR MSG500 ;Yes →RET
416 017C 08100 LD RREGO, RNUMC ;符号表示位置計算
417 017D 10008 ADD RREGO, #.DM.RDLS AND OFH
418 017E 18120 ST RSINLOC, RREGO
419 017F 0C186 BR NSG300
420
421 EJECT
    
```

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:24:27 12/20/93 PAGE 03-014

PROG =

SOURCE = CALC2.ASM

E	STNO	LOC.	OBJ.	M	I	SOURCE STATEMENT	
	422					NSG200:	
	423			1		BANK1	
+	1	0180	1D791	1		MOV BANK, #01H	
	424			1		SKF1 FZERO	
+	1	0181	1F302	1		SKF .MF.FZERO SHR 4, #.DF.FZERO AND OFH	
	425	0182	0C196			BR NSG400	
	426	0183	15031			XOR RXSIGN, #CCHSIN AND OFH	
	427			1		BANK0	
+	1	0184	1D790	1		MOV BANK, #00H	
	428	0185	08120			LD RREG0, RSINLOC	
	429						
	430					NSG300:	
	431			1		APUSHMP	;メモリ・ポインタ退避
+	1	0186	087AA	1		LD RREG10, MPH	
+	2	0187	087BB	1		LD RREG11, MPL	
	432						
	433			1		CLR1 LCDEN	;LCD表示オフ
+	1	0188	07331	1		PEEK WR, .MF.LCDEN SHR 4	
+	2	0189	14787	1		AND WR, #.DF.(NOT LCDEN) AND OFH	
+	3	018A	07321	1		POKE .MF.LCDEN SHR 4, WR	
	434	018B	1D7A8			MOV MPH, #CSTMPE AND OFH	
	435	018C	1D7B1			MOV MPL, #.DM.RDLSH SHR 4 AND OFH	
	436	018D	1A010			MOV RREG1, @RREG0	
	437	018E	15011			XOR RREG1, #CCHSIN AND OFH	;表示データの符号を反転
	438	018F	0A010			MOV @RREG0, RREG1	
	439			1		APOPMP	;メモリ・ポインタ復帰
+	1	0190	187AA	1		ST MPH, RREG10	
+	2	0191	187BB	1		ST MPL, RREG11	
	440						
	441	0192	EXTRN			CALL SDISP	;表示データ出力
	442			1		SET1 LCDEN	;LCD表示オン
+	1	0193	07331	1		PEEK WR, .MF.LCDEN SHR 4	
+	2	0194	16788	1		OR WR, #.DF.LCDEN AND OFH	
+	3	0195	07321	1		POKE .MF.LCDEN SHR 4, WR	
	443						
	444					NSG400:	
	445			1		BANK0	
+	1	0196	1D790	1		MOV BANK, #00H	
	446						
	447					NSG500:	
	448	0197	070E0			RET	
	449						
	450					EJECT	

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:24:27 12/20/93 PAGE 03-015

PROG =

SOURCE = CALC2.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
451 ;*****
452 ;*
453 ;* 演算子キー処理 *
454 ;* *
455 ;* INPUT : RKCODH - RKCODL (キー・コード)*
456 ;* : RZLSD - RZMSD (退避用レジスタ)*
457 ;* : ROPE (演算子エリア) *
458 ;* : RCOM (実行演算子エリア) *
459 ;* : RMODE (モード・エリア) *
460 ;* : RXSIGN - RXMSD *
461 ;* (浮動小数点レジスタ1) *
462 ;* : RYSIGN - RYMSD *
463 ;* (浮動小数点レジスタ2) *
464 ;* : FPER (パーセント・フラグ) *
465 ;* : FOPEND (演算終了フラグ) *
466 ;* : FFALSE (不正入力フラグ) *
467 ;* OUTPUT : ROPE (演算子エリア) *
468 ;* : RCOM (実行演算子エリア) *
469 ;* : RMODE (モード・エリア) *
470 ;* : FOPEND (演算終了フラグ) *
471 ;* : FFALSE (不正入力フラグ) *
472 ;* : FOPREQ (演算要求フラグ) *
473 ;* : RXSIGN - RXMSD *
474 ;* (浮動小数点レジスタ1) *
475 ;* : RYSIGN - RYMSD *
476 ;* (浮動小数点レジスタ2) *
477 ;*
478 ;*****
479
480 EJECT
    
```

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:24:27 12/20/93 PAGE 03-016

PROG =

SOURCE = CALC2.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
481 ;+++ キー・デコード +++
482 KOPE:
483 0198 0931D SKE RKCODL, #CKRET1 AND OFH
484 0199 0C19F BR NOP100
485 019A 0B30E SKNE RKCODH, #CKSRBO AND OFH ; [+] キーが押されたか?
486 019B 1D091 MOV RREG9, #CADD AND OFH ; Yes
487 019C 0B300 SKNE RKCODH, #CKSRCO AND OFH ; [-] キーが押されたか?
488 019D 1D092 MOV RREG9, #CSUB AND OFH ; Yes
489 019E 0C1A3 BR NOP200
490 NOP100:
491 019F 0B30E SKNE RKCODH, #CKSRBO AND OFH ; [x] キーが押されたか?
492 01A0 1D094 MOV RREG9, #CMUL AND OFH ; Yes
493 01A1 0B300 SKNE RKCODH, #CKSRCO AND OFH ; [+/-] キーが押されたか?
494 01A2 1D098 MOV RREG9, #CDIV AND OFH ; Yes
495
496 NOP200:
497 CLR1 FFALSE
+ 1 01A3 1468B 1 AND .MF. FFALSE SHR 4, #. DF. (NOT FFALSE AND OFH)
498 01A4 0B642 1 SKNE RMODE, #COPSEL AND OFH ; 演算子選択モードか?
499 01A5 0C1B9 BR NOP500 ; Yes
500 SKT1 FOPEND
+ 1 01A6 1E682 1 SKT .MF. FOPEND SHR 4, #. DF. FOPEND AND OFH
501 01A7 0C1AA BR NOP300
502 CLR1 FOPEND
+ 1 01A8 1468D 1 AND .MF. FOPEND SHR 4, #. DF. (NOT FOPEND AND OFH)
503 01A9 0C1B9 BR NOP500
504 NOP300:
505 SKT1 FPER ; パーセント演算中か?
+ 1 01AA 1E681 1 SKT .MF. FPER SHR 4, #. DF. FPER AND OFH
506 01AB 0C1B0 BR NOP400 ; No
507 CLR1 FPER ; Yes
+ 1 01AC 1468E 1 AND .MF. FPER SHR 4, #. DF. (NOT FPER AND OFH)
508 01AD 1B094 SKLT RREG9, #CMUL AND OFH ; [+] か [-] キーが押されたか?
509 01AE 0C1B9 BR NOP500 ; No
510 01AF 0C1C1 BR NOP700 ; Yes →パーセント演算
511
512 EJECT
    
```

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:24:27 12/20/93 PAGE 03-017

PROG =

SOURCE = CALC2.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
513 ;+++ データ転送 (表示データ・エリア
514 ;                                     →浮動小数点レジスタ2) +++
515 ;
NOP400: CALL STRNDY
516 01B0 EXTRN
517
518 01B1 09644 SKE RMODE, #CLEFT AND OFH ;左項モードか?
519 01B2 0C1BC BR NOP600 ;N o : 右項モード
520
521 ;+++ データ転送 (浮動小数点レジスタ2
522 ;                                     →浮動小数点レジスタ1) +++
523 01B3 1D001 MOV RREG0, #. DM. RYSIGN SHR 4 AND OFH
524 01B4 16008 OR RREG0, #CSTBK1 AND OFH
525 01B5 1D010 MOV RREG1, #. DM. RXSIGN SHR 4 AND OFH
526 01B6 16018 OR RREG1, #CSTBK1 AND OFH
527 01B7 EXTRN CALL STRAN
528 01B8 EXTRN CALL SRYCLR ;浮動小数点レジスタ2クリア
529
NOP500: ST ROPE, RREG9 ;退避させた演算子データを復帰
530 01B9 18609 MOV RMODE, #COPSEL AND OFH ;演算子選択モード設定
531 01BA 1D642 BR NOP900 ;R E T
532 01BB 0C1CF
533
NOP600: LD RREG0, ROPE
534
535 01BC 08600 ST RCOM, RREG0
536 01BD 18610 ST ROPE, RREG9
537 01BE 18609 MOV RMODE, #COPSEL AND OFH ;演算子選択モード設定
538 01BF 1D642 BR NOP800
539 01C0 0C1CE
540
541 EJECT
    
```

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:24:27 12/20/93 PAGE 03-018

PROG =

SOURCE = CALC2.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
542 ;+++ データ復帰 (退避用レジスタ→浮動小数点
543 ; レジスタ1) - パーセント演算 +++
544 NOP700:
545 01C1 18609 ST ROPE,RREG9
546 01C2 18619 ST RCOM,RREG9
547 01C3 1D000 MOV RREG0,#.DM.RXSIGN SHR 4 AND OFH
548 01C4 16008 OR RREG0,#CSTBK1 AND OFH
549 01C5 1D011 MOV RREG1,#.DM.RYSIGN SHR 4 AND OFH
550 01C6 16018 OR RREG1,#CSTBK1 AND OFH
551 01C7 EXTRN CALL STRAN
552 01C8 1D002 MOV RREG0,#.DM.RZLSD SHR 4 AND OFH
553 01C9 1D010 MOV RREG1,#.DM.RXSIGN SHR 4 AND OFH
554 01CA 16018 OR RREG1,#CSTBK1 AND OFH
555 01CB EXTRN CALL STRAN
556 01CC 1D644 MOV RMODE,#CLEFT AND OFH ;左項モード設定
557 SET1 FOPEN ;左項モード設定
+ 1 01CD 16682 1 OR .MF.FOPEN SHR 4,#.DF.FOPEN AND OFH
558
559 NOP800:
560 SET1 FOPREQ ;演算要求セット
+ 1 01CE 16694 1 OR .MF.FOPREQ SHR 4,#.DF.FOPREQ AND OFH
561
562 NOP900:
563 01CF 070E0 RET
564
565 EJECT
    
```

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:24:27 12/20/93 PAGE 03-019

PROG =

SOURCE = CALC2.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
566 ;*****
567 ;*
568 ;* パーセント・キー処理 *
569 ;* *
570 ;* INPUT : ROPE (演算子エリア) *
571 ;* : RMODE (モード・エリア) *
572 ;* : FPER (パーセント・フラグ) *
573 ;* : RXLSD - RXMSD (浮動小数点 *
574 ;* レジスタ1 仮数部) *
575 ;* OUTPUT : RZLSD - RZMSD (退避用レジスタ)*
576 ;* : RCOM (実行演算子エリア) *
577 ;* : RYEXP (浮動小数点レジスタ2 *
578 ;* 指数部) *
579 ;* : FOPEND (演算終了フラグ) *
580 ;* : FOPREQ (演算要求フラグ) *
581 ;* : FPER (パーセント・フラグ) *
582 ;* : FFALSE (不正入力フラグ) *
583 ;*
584 ;*****
585 KPER:
586 SKT1 FPER ; [%] キーがすでに押されたか?
+ 1 01D0 1E681 1 SKT .MF.FPER SHR 4, #.DF.FPER AND OFH
587 01D1 0B642 1 SKNE RMODE, #COPSEL AND OFH ;演算子選択モードか?
588 01D2 0C1E9 BR NPR800 ;Yes →RET
589 01D3 0B641 SKNE RMODE, #CRIGHT AND OFH ;右項モードか?
590 01D4 0C1D7 BR NPR200 ;Yes
591 CLR1 FOPEND ;No
+ 1 01D5 1468D 1 AND .MF.FOPEND SHR 4, #.DF.(NOT FOPEND AND OFH)
592 01D6 0C1DA BR NPR400
593
594 NPR200:
595 01D7 1B604 SKLT ROPE, #CMUL AND OFH ; [×] か [÷] キーが押されたか?
596 01D8 0C1DC BR NPR600 ;Yes →パーセント演算
597 01D9 1D644 MOV RMODE, #CLEFT AND OFH ;No: 左項モード設定
598
599 NPR400:
+ 1 01DA 16684 1 SET1 FFALSE
600 01DB 0C1E9 1 OR .MF.FFALSE SHR 4, #.DF.FFALSE AND OFH
601 BR NPR800 ;RET
602 EJECT
    
```



AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:24:27 12/20/93 PAGE 03-020

PROG =

SOURCE = CALC2.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
603 ;+++ データ退避 (浮動小数点レジスタ1 → 退避
604 ; 用レジスタ) - パーセント演算 +++
605 NPR600:
606 01DC 1D000 MOV RREG0,#.DM.RXSIGN SHR 4 AND 0FH
607 01DD 16008 OR RREG0,#CSTBK1 AND 0FH
608 01DE 1D012 MOV RREG1,#.DM.RZLSD SHR 4 AND 0FH
609 01DF EXTRN CALL STRAN
610 01E0 EXTRN CALL STRNDY
611 1 BANK1
+ 1 01E1 1D791 1 MOV BANK,#01H
612 01E2 11142 SUB RYEXP,#2H
613 01E3 13150 SUBC RYEXP+1,#0H
614 1 BANK0
+ 1 01E4 1D790 1 MOV BANK,#00H
615 1 SET1 FPER
+ 1 01E5 16681 1 OR .MF.FPER SHR 4,#.DF.FPER AND 0FH
616 01E6 08600 LD RREG0,ROPE
617 01E7 18610 ST RCOM,RREG0
618 1 SET1 FOPREQ ;演算要求セット
+ 1 01E8 16694 1 OR .MF.FOPREQ SHR 4,#.DF.FOPREQ AND 0FH
619
620 NPR800:
621 01E9 070E0 RET
622
623 EJECT
    
```

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:24:27 12/20/93 PAGE 03-021

PROG =

SOURCE = CALC2.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
624 ;*****
625 ;*
626 ;*      イコール・キー処理
627 ;*
628 ;*      INPUT  : ROPE (演算子エリア)
629 ;*              : RMODE (モード・エリア)
630 ;*              : FPER (パーセント・フラグ)
631 ;*              : RXSIGN - RXMSD
632 ;*              (浮動小数点レジスタ1)
633 ;*      OUTPUT : RCOM (実行演算子エリア)
634 ;*              : RMODE (モード・エリア)
635 ;*              : FOPEND (演算終了フラグ)
636 ;*              : FFALSE (不正入力フラグ)
637 ;*              : FOPREQ (演算要求フラグ)
638 ;*              : RYSIGN - RYMSD
639 ;*              (浮動小数点レジスタ2)
640 ;*
641 ;*****
642 KEQU:
643          SKT1  FPER          ; [%] キーが押されたか?
+ 1 01EA 1E681 1          SKT  .MF.FPER SHR 4, #.DF.FPER AND 0FH
644 01EB 0C1EE          BR    NEQ100          ; N o
645          CLR1  FPER          ; Y e s
+ 1 01EC 1468E 1          AND  .MF.FPER SHR 4, #.DF.(NOT FPER AND 0FH)
646 01ED 0C203          BR    NEQ600
647
648          NEQ100:
649 01EE 09644          SKE  RMODE, #CLEFT AND 0FH      ; 左項モードか?
650 01EF 0C1F2          BR    NEQ200          ; N o
651          SET1  FFALSE        ; Y e s
+ 1 01FO 16684 1          OR   .MF.FFALSE SHR 4, #.DF.FFALSE AND 0FH
652 01F1 0C205          BR    NEQ700          ; R E T
653
654          EJECT
    
```

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:24:27 12/20/93 PAGE 03-022

PROG =

SOURCE = CALC2.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
655 NEQ200:
656 01F2 08600 LD RREG0, ROPE
657 01F3 18610 ST RCOM, RREG0
658 01F4 09642 SKE RMODE, #COPSEL AND 0FH ;演算子選択モードか?
659 01F5 0C201 BR NEQ400 ;No
660 01F6 1B614 SKLT RCOM, #CMUL ;[+] か [-] キーが押されたか?
661 01F7 0C1FB BR NEQ300 ;No
662 01F8 EXTRN CALL SRYCLR ;Yes: 浮動小数点レジスタ2クリア
663 1 BANK0
+ 1 01F9 1D790 1 MOV BANK, #00H
664 01FA 0C202 BR NEQ500
665
666 ;+++ データ転送(浮動小数点レジスタ1
667 ; →浮動小数点レジスタ2) +++
668 NEQ300:
669 01FB 1D000 MOV RREG0, #.DM.RXSIGN SHR 4 AND 0FH
670 01FC 16008 OR RREG0, #CSTBK1 AND 0FH
671 01FD 1D011 MOV RREG1, #.DM.RYSIGN SHR 4 AND 0FH
672 01FE 16018 OR RREG1, #CSTBK1 AND 0FH
673 01FF EXTRN CALL STRAN
674 0200 0C202 BR NEQ500
675
676 NEQ400:
677 0201 EXTRN CALL STRNDY
678
679 NEQ500:
680 1 SET1 FOPREQ ;演算要求セット
+ 1 0202 16694 1 OR .MF.FOPREQ SHR 4, #.DF.FOPREQ AND 0FH
681
682 NEQ600:
683 1 SET1 FOPEND
+ 1 0203 16682 1 OR .MF.FOPEND SHR 4, #.DF.FOPEND AND 0FH
684 0204 1D644 MOV RMODE, #CLEFT AND 0FH ;左項モード設定
685
686 NEQ700:
687 0205 070E0 RET
688
689 EJECT
    
```

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:24:27 12/20/93 PAGE 03-023

PROG =

SOURCE = CALC2.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
690 ;*****
691 ;*
692 ;* 演算分岐処理 , エラー処理 *
693 ;*
694 ;* INPUT : RCOM (実行演算子エリア) *
695 ;* OUTPUT : RNUMC (数値キー・カウンタ) *
696 ;* : RDEXP - RDMSD (表示データ) *
697 ;* : RMODE (モード・エリア) *
698 ;* : FOPREQ (演算要求フラグ) *
699 ;*
700 ;*****
701 MOPBRN:
702 CLR1 FOPREQ
+ 1 0206 1469B 1 AND .MF.FOPREQ SHR 4, #.DF. (NOT FOPREQ AND OFH)
703 CLR1 LCDEN ;LCD表示オフ
+ 1 0207 07331 1 PEEK WR, .MF.LCDEN SHR 4
+ 2 0208 14787 1 AND WR, #.DF. (NOT LCDEN) AND OFH
+ 3 0209 07321 1 POKE .MF.LCDEN SHR 4, WR
704 BANK1
+ 1 020A 1D791 1 MOV BANK, #01H
705 020B 1D300 1 MOV ROPFLG, #0000B ;演算フラグ・クリア
706 BANK0
+ 1 020C 1D790 1 MOV BANK, #00H
707 020D 0B618 1 SKNE RCOM, #CDIV AND OFH ;除算命令か?
708 020E 0C219 1 BR LOB300 ;Yes
709 020F 0B614 1 SKNE RCOM, #CMUL AND OFH ;乗算命令か?
710 0210 0C217 1 BR LOB200 ;Yes
711 0211 0B612 1 SKNE RCOM, #CSUB AND OFH ;減算命令か?
712 0212 0C215 1 BR LOB100 ;Yes
713
714 ;+++ 加算演算 +++
715 0213 EXTRN CALL SFPADD
716 0214 0C21D BR LOB400
717
718 ;+++ 減算演算 +++
719 LOB100: CALL SFPSUB
720 0215 EXTRN BR LOB400
721 0216 0C21D
722
723 ;+++ 乗算演算 +++
724 LOB200: CALL SFPMULT
725 0217 EXTRN BR LOB400
726 0218 0C21D
727
728 ;+++ 除算演算 +++
729 LOB300: CALL SFPDIV
730 0219 EXTRN CALL SFPDIV
731 BANK1
+ 1 021A 1D791 1 MOV BANK, #01H
732 SKF1 FDVERR ;除数が0か?
+ 1 021B 1F304 1 SKF .MF.FDVERR SHR 4, #.DF.FDVERR AND OFH
733 021C 0C231 BR LOB700 ;Yes →エラー処理
734
735 EJECT
    
```

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:24:27 12/20/93 PAGE 03-024

PROG =

SOURCE = CALC2.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
736 ;+++ オーバフロー +++
737 LOB400:
738 BANK1
+ 1 021D 1D791 1 MOV BANK,#01H
739 021E 08040 LD RREG0,RXEXP
740 021F 08051 LD RREG1,RXEXP+1H
741 BANK0
+ 1 0220 1D790 1 MOV BANK,#00H
742 0221 1B018 SKLT RREG1,#8H ;指数が0未満か?
743 0222 0C228 BR LOB500 ;Yes
744 0223 11008 SUB RREG0,#(CEXPMAX+1H) AND OFH ;指数が0~7か?
745 0224 13010 SUBC RREG1,#(CEXPMAX+1H) SHR 4 AND OFH
746 SKF1 CY
+ 1 0225 1F7F4 1 SKF .MF.CY SHR 4,#.DF.CY AND OFH
747 0226 0C22C BR LOB600 ;Yes →演算結果表示
748 0227 0C231 BR LOB700 ;No →エラー処理
749
750 LOB500:
751 0228 1100B SUB RREG0,#CEXP MIN AND OFH ;指数が-5~-1か?
752 0229 1301F SUBC RREG1,#CEXP MIN SHR 4 AND OFH
753 SKF1 CY
+ 1 022A 1F7F4 1 SKF .MF.CY SHR 4,#.DF.CY AND OFH
754 022B EXTRN CALL SRXCLR ;No: 演算結果←0
755
756 ;+++ 演算結果変換 +++
757 LOB600:
758 022C EXTRN CALL SFIX ;演算結果→表示データ
759 022D 1D148 MOV RDEXP,#CEXP INI AND OFH
760 022E 1D150 MOV RDEXP+1H,#CEXP INI SHR 4 AND OFH
761 022F 1D100 MOV RNUMC,#CNONUM AND OFH
762 0230 0C235 BR LOB800 ;→演算結果表示
763
764 ;+++ エラー処理 +++
765 LOB700:
766 BANK0
+ 1 0231 1D790 1 MOV BANK,#00H
767 0232 1D648 MOV RMODE,#CERROR AND OFH ;エラー・モード設定
768 0233 EXTRN CALL SDPINI
769 0234 1D18C MOV RDLS,#CERRDP AND OFH ;表示データの最下位桁
770 ; ← [E] の表示データ
771
772 LOB800:
773 0235 EXTRN CALL SDISP ;表示データ出力
774 SET1 LCDEN ;LCD表示オン
+ 1 0236 07331 1 PEEK WR,.MF.LCDEN SHR 4
+ 2 0237 16788 1 OR WR,#.DF.LCDEN AND OFH
+ 3 0238 07321 1 POKE .MF.LCDEN SHR 4,WR
775
776 0239 070E0 RET
777
778 END

```

TOTAL ERRORS = 0  
TOTAL WARNINGS = 0

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:24:27 12/20/93 PAGE 03-025

PROG =

SOURCE = CALC2.ASM

E STNO LOC. OBJ. M I SOURCE STATEMENT

END OF LIST

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:25:27 12/20/93 PAGE 04-001

PROG =

SOURCE = CALC3.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
1          ;*****
2          ;*
3          ;* USER NAME   : NEC CORPORATION      *
4          ;*
5          ;* SYSTEM NAME  : 17K APPLICATION (CALCULATOR) *
6          ;*
7          ;* CPU         : uPD17201A           *
8          ;*
9          ;* LAST UPDATE : '93/12/20  11:00    *
10         ;*
11         ;*****
12         ;*****
13         ;*
14         ;* FILE NAME    : CALC3.ASM           *
15         ;*
16         ;* INCLUDE 10 ROUTINES:              *
17         ;*   SRAMCR   : RAM ALL CLEAR       *
18         ;*   SDPINI   : DISPLAY DATA AREA  *
19         ;*             INITIALIZE          *
20         ;*   SRYCLR   : SECOND OPERAND DATA *
21         ;*             AREA CLEAR         *
22         ;*   SDSHFD   : DISPLAY DATA AREA  *
23         ;*             DOWN SHIFT        *
24         ;*   SUSHFD   : DISPLAY DATA AREA  *
25         ;*             UP SHIFT          *
26         ;*   SUSHFY   : SECOND OPERAND DATA *
27         ;*             AREA DOWN SHIFT   *
28         ;*   STRAN    : TRANSFER DATA      *
29         ;*   SFIX     : TRANSFER FLOATING  *
30         ;*             POINT NUMBER TO   *
31         ;*             FIXED POINT NUMBER *
32         ;*   STRNDY   : TRANSFER DISPLAY DATA *
33         ;*             TO SECOND OPERAND  *
34         ;*   SDISP    : DISPLAY DATA OUTPUT *
35         ;*
36         ;*****
37
38         EJECT
    
```

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:25:27 12/20/93 PAGE 04-002

PROG =

SOURCE = CALC3.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
39 ;*****
40 ;*          外部参照          *
41 ;*****
42 EXTRN MEM: RREG0, RREG1, RREG2 ;汎用レジスタ
43 EXTRN MEM: RREG3, RREG4
44 EXTRN MEM: RNUMC ;数値キー・カウンタ
45 EXTRN MEM: RPTLOC ;小数点表示位置エリア
46 EXTRN MEM: RSINLOC ;符号表示位置エリア
47 EXTRN MEM: RDSIGN ;演算結果符号エリア
48 EXTRN MEM: RDEXP ;表示データ指数エリア
49 EXTRN MEM: RDLSD, RDMSD ;表示データ・エリア
50 EXTRN MEM: RXSIGN ;浮動小数点レジスタ 1 符号部
51 EXTRN MEM: RYSIGN ;浮動小数点レジスタ 2 符号部
52 EXTRN MEM: RYEXP ;浮動小数点レジスタ 2 指数部
53 EXTRN MEM: RYMSD, RYLSD ;浮動小数点レジスタ 2 仮数部
54
55 EXTRN DAT: CSTMPE ;MPH: 間接アドレス指定 (BANK 0)
56 EXTRN DAT: CSTBK2 ; ; (BANK 2)
57 EXTRN DAT: CSTBK1 ;MPL: 間接アドレス指定 (BANK 1)
58 EXTRN DAT: CPLINI ;小数点表示位置エリア初期値
59 EXTRN DAT: CEXPINI ;表示データ指数エリア初期値
60 EXTRN DAT: CZERO ;[0]表示データ
61 EXTRN DAT: CSPACE ;スペース表示データ
62 EXTRN DAT: CMLNUS ;[-]表示データ
63 EXTRN DAT: CDPON ;小数点表示セグメント・データ
64 EXTRN DAT: CSINOFF, CSINMN ;符号表示なし/マイナス
65
66 ;*****
67 ;*          外部定義          *
68 ;*****
69 PUBLIC SRAMCR, SDPINI, SDSHFD, SUSHFD, STRAN, SFIX, STRNDY, SRYCLR, SDISP
70
71 EJECT
    
```



AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:25:27 12/20/93 PAGE 04-003

PROG =

SOURCE = CALC3.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
  72      ;*****
  73      ;*
  74      ;*      マクロ定義ファイル読み込み      *
  75      ;*
  76      ;*****
  77      1      INCLUDE 'PUSHMP.ASM'
+ 1      1
+ 2      1      EXTRN  MEM: RREG10,RREG11
+ 3      1
+ 4      1 ;*****
+ 5      1 ;*
+ 6      1 ;*      データ・メモリ・ロウ・アドレス・      *
+ 7      1 ;*      ポインタ退避マクロ定義      *
+ 8      1 ;*
+ 9      1 ;*****
+ 10     1      APUSHMP MACRO
+ 11     1      LD      RREG10,MPH
+ 12     1      LD      RREG11,MPL
+ 13     1      ENDM
+ 14     1
+ 15     1 ;*****
+ 16     1 ;*
+ 17     1 ;*      データ・メモリ・ロウ・アドレス・      *
+ 18     1 ;*      ポインタ復帰マクロ定義      *
+ 19     1 ;*
+ 20     1 ;*****
+ 21     1      APOPMP  MACRO
+ 22     1      ST      MPH,RREG10
+ 23     1      ST      MPL,RREG11
+ 24     1      ENDM
+ 25     1
+ 26     1      EOF
  78
  79      EJECT
    
```

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:25:27 12/20/93 PAGE 04-004

PROG =

SOURCE = CALC3.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
80 ;*****
81 ;*
82 ;* R A Mオール・クリア処理 *
83 ;* *
84 ;* INPUT : Nothing *
85 ;* OUTPUT : ALL RAM AREA *
86 ;* *
87 ;*****
88 SRAMCR:
89 1 BANK0
+ 1 023A 1D790 1 MOV BANK, #00H
90 023B 1D7A8 MOV MPH, #CSTMPE AND 0FH ;MPEセット
91 023C 1D7B0 MOV MPL, #. DM. RREG2 SHR 4 AND 0FH
92 023D 1D002 MOV RREG0, #. DM. RREG2 AND 0FH
93 023E 1D010 MOV RREG1, #CZERO AND 0FH
94 JRC200:
95 023F 0A010 MOV @RREG0, RREG1
96 0240 10001 ADD RREG0, #1H
97 0241 09000 SKE RREG0, #0H
98 0242 0C23F BR JRC200
99
100 0243 107B1 ADD MPL, #1H
101 0244 0B7B7 SKNE MPL, #7H ;BANK0かBANK2が終了したか?
102 0245 0C24A BR JRC400 ;Yes
103 0246 097BF SKE MPL, #0FH ;BANK1が終了したか?
104 0247 0C23F BR JRC200 ;No →次の列
105 0248 1D7A9 MOV MPH, #CSTBK2 AND 0FH ;Yes →BANK2
106 0249 0C24C BR JRC600
107
108 JRC400:
109 024A 0B7A9 SKNE MPH, #CSTBK2 AND 0FH ;BANK2が終了したか?
110 024B 0C24E BR JRC800 ;Yes →RET
111 JRC600:
112 024C 107B1 ADD MPL, #1H
113 024D 0C23F BR JRC200
114
115 JRC800:
116 024E 070E0 RET
117
118 EJECT
    
```

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:25:27 12/20/93 PAGE 04-005

PROG =

SOURCE = CALC3.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
119 ;*****
120 ;*
121 ;* 表示データ・エリア・イニシャライズ処理 *
122 ;*
123 ;* INPUT : Nothing *
124 ;* OUTPUT : RPTLOC (小数点表示位置エリア) *
125 ;* : RDLS D - RDMS D (表示データ) *
126 ;*
127 ;*****
128 SDPINI:
129 APUSHMP ;メモリ・ポインタ退避
+ 1 024F 087AA 1 LD RREG10, MPH
+ 2 0250 087BB 1 LD RREG11, MPL
130
131 BANK0
+ 1 0251 1D790 1 MOV BANK, #00H
132 0252 1D180 MOV RDLS D, #CZERO AND 0FH ;最下桁 ← [0] 表示データ
133
134 0253 1D7A8 MOV MPH, #CSTMPE AND 0FH ;MPEセット
135 0254 1D7B1 MOV MPL, #.DM. RDLS D SHR 4 AND 0FH ;表示データ・エリアの
;ロウ・アドレスをセット
136
137 0255 1D009 MOV RREG0, #.DM. (RDLS D+1H) AND 0FH ;カラム・アドレスをセット
138 0256 1D01A MOV RREG1, #CSPACE AND 0FH ;スペース表示データ
139 JDI200:
140 0257 0A010 MOV @RREG0, RREG1
141 0258 10001 ADD RREG0, #1H
142 0259 09000 SKE RREG0, #0H ;イニシャライズ終了?
143 025A 0C257 BR JDI200 ;No → 次の桁
144 025B 1D117 MOV RPTLOC, #CPLINI AND 0FH ;小数点表示位置をリセット
145
146 APOPMP ;メモリ・ポインタ復帰
+ 1 025C 187AA 1 ST MPH, RREG10
+ 2 025D 187BB 1 ST MPL, RREG11
147
148 025E 070E0 RET
149
150 EJECT
    
```

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:25:27 12/20/93 PAGE 04-006

PROG =

SOURCE = CALC3.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
151 ;*****
152 ;*
153 ;* 浮動小数点レジスタ 2 クリア処理 *
154 ;* *
155 ;* INPUT : Nothing *
156 ;* OUTPUT : RYSIGN - RDMSD *
157 ;* (浮動小数点レジスタ 2) *
158 ;*
159 ;*****
160 SRYCLR:
161 1 APUSHMP ;メモリ・ポインタ退避
+ 1 025F 087AA 1 LD RREG10, MPH
+ 2 0260 087BB 1 LD RREG11, MPL
162
163 1 BANK0
+ 1 0261 1D790 1 MOV BANK, #00H
164 0262 1D7A8 MOV MPH, #CSTMPE AND OFH ;MPEセット
165 0263 1D7B1 MOV MPL, #.DM.RYSIGN SHR 4 AND OFH ;浮動小数点レジスタ 2 の
166 0264 167B8 OR MPL, #CSTBK1 AND OFH ;ロウ・アドレスをセット
167 0265 1D003 MOV RREG0, #.DM.RYSIGN AND OFH ;カラム・アドレスをセット
168 0266 1D010 MOV RREG1, #CZERO AND OFH ;スペース表示データ
169 JYC200:
170 0267 0A010 MOV @RREG0, RREG1
171 0268 10001 ADD RREG0, #1H
172 0269 09000 SKE RREG0, #0H ;クリア終了?
173 026A 0C267 BR JYC200 ;No →次の桁
174
175 1 APOPMP ;メモリ・ポインタ復帰
+ 1 026B 187AA 1 ST MPH, RREG10
+ 2 026C 187BB 1 ST MPL, RREG11
176
177 026D 070E0 RET
178
179 EJECT
    
```

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:25:27 12/20/93 PAGE 04-007

PROG =

SOURCE = CALC3.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
180 ;*****
181 ;*
182 ;* 表示データ・エリア・ダウン・シフト処理 *
183 ;*
184 ;* INPUT : RDLS - RDMSD (表示データ) *
185 ;* OUTPUT : RDLS - RDMSD (表示データ) *
186 ;*
187 ;*****
188 SDSHFD:
189 1 APUSHMP ;メモリ・ポインタ退避
+ 1 026E 087AA 1 LD RREG10, MPH
+ 2 026F 087BB 1 LD RREG11, MPL
190
191 1 BANK0
+ 1 0270 1D790 1 MOV BANK, #00H
192 0271 1D7A8 MOV MPH, #CSTMPE AND 0FH ;MPEセット
193 0272 1D7B1 MOV MPL, #.DM. RDLS - SHR 4 AND 0FH ;表示データ・エリアの
194 ;로우・アドレスをセット
195 0273 1D007 MOV RREG0, #.DM. (RDLS-1H) AND 0FH ;カラム・アドレスをセット
196 JDD200:
197 0274 1A010 MOV RREG1, @RREG0 ;1バイト・ダウン・シフト
198 0275 11001 SUB RREG0, #1H
199 0276 0A010 MOV @RREG0, RREG1
200 0277 10002 ADD RREG0, #2H
201 0278 09000 SKE RREG0, #0H ;ダウン・シフト終了?
202 0279 0C274 BR JDD200 ;No → 次の桁
203 027A 1D1F0 MOV RDMSD, #CZERO AND 0FH ;最上位桁 ← [0] 表示データ
204
205 1 APOPMP ;メモリ・ポインタ復帰
+ 1 027B 187AA 1 ST MPH, RREG10
+ 2 027C 187BB 1 ST MPL, RREG11
206
207 027D 070E0 RET
208
209 EJECT
    
```

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:25:27 12/20/93 PAGE 04-008

PROG =

SOURCE = CALC3.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
210 ;*****
211 ;*
212 ;* 表示データ・エリア・アップ・シフト処理 *
213 ;*
214 ;* INPUT : RDLS - RDMSD (表示データ) *
215 ;* OUTPUT : RDLS - RDMSD (表示データ) *
216 ;*
217 ;*****
218 SUSHPD:
219 1 APUSHMP ;メモリ・ポインタ退避
+ 1 027E 087AA 1 LD RREG10, MPH
+ 2 027F 087BB 1 LD RREG11, MPL
220
221 1 BANK0
+ 1 0280 1D790 1 MOV BANK, #00H
222 0281 1D7A8 MOV MPH, #CSTMPE AND 0FH ;MPEセット
223 0282 1D7B1 MOV MPL, #.DM.RDLS SHR 4 AND 0FH ;表示データ・エリアの
224 ;ロウ・アドレスをセット
225 0283 1D00E MOV RREG0, #.DM.(RDMSD-1H) AND 0FH ;カラム・アドレスをセット
226 JUD200:
227 0284 1A010 MOV RREG1, @RREG0 ;1バイト・アップ・シフト
228 0285 10001 ADD RREG0, #1H
229 0286 0A010 MOV @RREG0, RREG1
230 0287 11002 SUB RREG0, #2H
231 0288 1B008 SKLT RREG0, #.DM.RDLS AND 0FH ;アップ・シフト終了?
232 0289 0C284 BR JUD200 ;N o → 次の桁
233 028A 1D180 MOV RDLS, #CZERO AND 0FH ;最下位桁 ← [0] 表示データ
234
235 1 APOPMP ;メモリ・ポインタ復帰
+ 1 028B 187AA 1 ST MPH, RREG10
+ 2 028C 187BB 1 ST MPL, RREG11
236
237 028D 070E0 RET
238
239 EJECT
    
```

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:25:27 12/20/93 PAGE 04-009

PROG =

SOURCE = CALC3.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
240 ;*****
241 ;*
242 ;* 浮動小数点レジスタ2 アップ・シフト処理 *
243 ;*
244 ;* INPUT : RYLS - RYMSD *
245 ;* (浮動小数点レジスタ2 仮数部) *
246 ;* OUTPUT : RYLS - RYMSD *
247 ;* (浮動小数点レジスタ2 仮数部) *
248 ;*
249 ;*****
250 SUSHFY:
251 1 APUSHMP ;メモリ・ポインタ退避
+ 1 028E 087AA 1 LD RREG10, MPH
+ 2 028F 087BB 1 LD RREG11, MPL
252
253 1 BANK0
+ 1 0290 1D790 1 MOV BANK, #00H
254 0291 1D7A8 MOV MPH, #CSTMPE AND OFH ;MPEセット
255 0292 1D7B1 MOV MPL, #.DM. RYLS SHR 4 AND OFH ;浮動小数点レジスタ2の
256 0293 167B8 OR MPL, #CSTBK1 AND OFH ;ロウ・アドレスをセット
257 0294 1D00E MOV RREG0, #.DM. (RYMSD-1H) AND OFH ;コラム・アドレスをセット
258
259 0295 1A010 JULY200: MOV RREG1, @RREG0 ;1バイト・アップ・シフト
260 0296 10001 ADD RREG0, #1H
261 0297 0A010 MOV @RREG0, RREG1
262 0298 11002 SUB RREG0, #2H
263 0299 1B006 SKLT RREG0, #.DM. RYLS AND OFH ;アップ・シフト終了?
264 029A 0C295 BR JULY200 ;No →次の桁
265 1 BANK1
+ 1 029B 1D791 1 MOV BANK, #01H
266 029C 1D160 MOV RYLS, #CZERO AND OFH ;最下位桁 ← [0] 表示データ
267 1 BANK0
+ 1 029D 1D790 1 MOV BANK, #00H
268
269 1 APOPMP ;メモリ・ポインタ復帰
+ 1 029E 187AA 1 ST MPH, RREG10
+ 2 029F 187BB 1 ST MPL, RREG11
270
271 02A0 070E0 RET
272
273 EJECT
    
```

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:25:27 12/20/93 PAGE 04-010

PROG =

SOURCE = CALC3.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
274 :*****
275 :*
276 :*          データ転送処理
277 :*
278 :*          INPUT  : RREG0 (転送元データ・エリアの
279 :*                    ロウ・アドレス)
280 :*                    : RREG1 (転送先データ・エリアの
281 :*                    ロウ・アドレス)
282 :*                    : RZLSD - RZMSD (退避用レジスタ)
283 :*                    : RXSIGN - RXMSD
284 :*                    (浮動小数点レジスタ1)
285 :*                    : RYSIGN - RYMSD
286 :*                    (浮動小数点レジスタ2)
287 :*          OUTPUT : RDSIGN - RDMSD
288 :*                    (表示データ・レジスタ)
289 :*                    : RZLSD - RZMSD (退避用レジスタ)
290 :*                    : RXSIGN - RXMSD
291 :*                    (浮動小数点レジスタ1)
292 :*                    : RYSIGN - RYMSD
293 :*                    (浮動小数点レジスタ2)
294 :*
295 :*****
296
297          EJECT
    
```



AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:25:27 12/20/93 PAGE 04-011

PROG =

SOURCE = CALC3.ASM

E	STNO	LOC.	OBJ.	M	I	SOURCE STATEMENT	
	298					STRAN:	
	299			1		APUSHMP	;メモリ・ポインタ退避
+	1	02A1	087AA	1		LD RREG10, MPH	
+	2	02A2	087BB	1		LD RREG11, MPL	
	300						
	301			1		BANK0	
+	1	02A3	1D790	1		MOV BANK, #00H	
	302	02A4	1D7A8			MOV MPH, #CSTMPE AND OFH	;MPEセット
	303	02A5	1D023			MOV RREG2, #. DM. RDSIGN AND OFH	;カラム・アドレスをセット
	304				JTR200:		
	305	02A6	187B0			ST MPL, RREG0	;転送元のロウ・アドレスをセット
	306	02A7	1A032			MOV RREG3, @RREG2	
	307	02A8	187B1			ST MPL, RREG1	;転送先のロウ・アドレスをセット
	308	02A9	0A032			MOV @RREG2, RREG3	
	309	02AA	10021			ADD RREG2, #1H	
	310	02AB	09020			SKE RREG2, #0H	;転送終了?
	311	02AC	0C2A6			BR JTR200	;No →次の桁
	312						
	313			1		APOPMP	;メモリ・ポインタ復帰
+	1	02AD	187AA	1		ST MPH, RREG10	
+	2	02AE	187BB	1		ST MPL, RREG11	
	314						
	315	02AF	070E0			RET	
	316						
	317					EJECT	

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:25:27 12/20/93 PAGE 04-012

PROG =

SOURCE = CALC3.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
318 ;*****
319 ;*
320 ;* 演算結果変換処理 *
321 ;*
322 ;* INPUT : RXSIGN - RXMSD (演算結果) *
323 ;* OUTPUT : RPTLOC (小数点表示位置エリア) *
324 ;* : RSINLOC (符号表示位置エリア) *
325 ;* : RDSIGN - RDMSD *
326 ;* (表示データ・レジスタ) *
327 ;*
328 ;*****
329 ;+++ 演算結果を表示データ・エリアへ転送 +++
330 SFIX:
331 1 BANKO
+ 1 02B0 1D790 1 MOV BANK, #00H
332 02B1 1D000 MOV RREGO, #. DM. RXSIGN SHR 4 AND OFH
333 02B2 16008 OR RREGO, #CSTBK1
334 02B3 1D011 MOV RREG1, #. DM. RDSIGN SHR 4 AND OFH
335 02B4 1C2A1 CALL STRAN
336
337 ;+++ 指数データ判定 +++
338 02B5 0B140 SKNE RDEXP, #0H ;指数データが0か?
339 02B6 09150 SKE RDEXP+1H, #0H
340 02B7 0C2BA BR JFX100
341 02B8 1D141 MOV RDEXP, #1H ;Yes: 指数データ←1
342 02B9 0C2BC BR JFX200
343 JFX100:
344 02BA 0915F SKE RDEXP+1H, #0FH ;指数データが0未満か?
345 02BB 0C2BD BR JFX300 ;No
346 JFX200:
347 02BC 1C26E CALL SDSHFD
348
349 ;+++ 演算結果の正負判定 +++
350 JFX300:
351 02BD 09131 SKE RDSIGN, #CSINMN AND OFH ;演算結果が0未満か?
352 02BE 0C2C1 BR JFX400 ;No
353 02BF 1D1FB MOV RDMSD, #CMINUS AND OFH ;Yes: 表示データの最上位桁に
354 02C0 0C2C2 BR JFX500 ; [-] の表示データをセット
355 JFX400:
356 02C1 1D1FA MOV RDMSD, #CSPACE AND OFH ;表示データの最上位桁にスペースの
357 ; 表示データをセット
358 EJECT
    
```

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:25:27 12/20/93 PAGE 04-013

PROG =

SOURCE = CALC3.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
359 ;+++ 表示データ・ダウン・シフト
360 ; (最上位桁を除く) +++
361 JFX500:
362 02C2 0B15F SKNE RDEXP+1H, #0FH ;指数データが0未満か?
363 02C3 0C2C7 BR JFX600 ;Yes →ダウン・シフト
364 02C4 08140 LD RREG0, RDEXP ;No
365 02C5 18110 ST RPTLOC, RREG0
366 02C6 0C2DC BR JFX800
367
368 JFX600:
369 APUSHMP ;メモリ・ポインタ退避
+ 1 02C7 087AA 1 LD RREG10, MPH
+ 2 02C8 087BB 1 LD RREG11, MPL
370
371 02C9 1D7A8 MOV MPH, #CSTMPE AND 0FH ;MPEセット
372 02CA 1D7B1 MOV MPL, #.DM.RDLSH SHR 4 AND 0FH ;表示データ・エリアの
;ロウ・アドレスをセット
373
374 JFX650:
375 02CB 1D007 MOV RREG0, #.DM.(RDLSH-1H) AND 0FH ;カラム・アドレスをセット
376 JFX700:
377 02CC 1A010 MOV RREG1, @RREG0 ;1バイト・ダウン・シフト
378 02CD 11001 SUB RREG0, #1H
379 02CE 0A010 MOV @RREG0, RREG1
380 02CF 10002 ADD RREG0, #2H
381 02D0 0900F SKE RREG0, #.DM.RDMSD AND 0FH ;ダウン・シフト終了?
382 02D1 0C2CC BR JFX700 ;No →次の桁
383 02D2 1D1E0 MOV RDMSD-1H, #CZERO AND 0FH
384
385 1 SET1 Z
+ 1 02D3 167F2 1 OR .MF.Z SHR 4, #.DF.Z AND 0FH
386 02D4 10141 ADD RDEXP, #1H ;指数データをインクリメント
387 02D5 12150 ADDC RDEXP+1H, #0H
388 1 SKT1 Z ;指数データが0になったか?
+ 1 02D6 1E7F2 1 SKT .MF.Z SHR 4, #.DF.Z AND 0FH
389 02D7 0C2CB 1 BR JFX650 ;No →ダウン・シフトを繰り返す
390
391 1 APOPMP ;Yes: メモリ・ポインタ復帰
+ 1 02D8 187AA 1 ST MPH, RREG10
+ 2 02D9 187BB 1 ST MPL, RREG11
392
393 02DA 1D141 MOV RDEXP, #1H
394 02DB 1D111 MOV RPTLOC, #1H
395
396 EJECT
    
```



AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:25:27 12/20/93 PAGE 04-015

PROG =

SOURCE = CALC3.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
416 ;*****
417 ;*
418 ;* 表示データ変換処理 *
419 ;* *
420 ;* INPUT : RNUMC (数値キー・カウンタ) *
421 ;* : RDEXP - RDMSD (表示データ) *
422 ;* OUTPUT : RYSIGN - RYMSD *
423 ;* (浮動小数点レジスタ2) *
424 ;* *
425 ;*****
426 STRNDY:
427 02E6 1C25F CALL SRYCLR ;浮動小数点レジスタ2 クリア
428 02E7 09148 SKE RDEXP, #CXPINI AND OFH ;小数点キーが押されたか?
429 02E8 0C2EB BR JTD100 ;Y e s
430 02E9 08100 LD RREG0, RNUMC ;N o : 表示データ指数エリア
431 02EA 18140 ST RDEXP, RREG0 ; ←数値キー・カウンタ
432
433 ;+++ 表示データからスペースを削除 +++
434 JTD100:
435 1 APUSHMP ;メモリ・ポインタ退避
+ 1 02EB 087AA 1 LD RREG10, MPH
+ 2 02EC 087BB 1 LD RREG11, MPL
436
437 02ED 1D7A8 MOV MPH, #CSTMPE AND OFH ;M P E セット
438 02EE 1D7B1 MOV MPL, #. DM. RDMSD SHR 4 AND OFH ;表示データ・エリアの
439 ;ロウ・アドレスをセット
440 02EF 1D00F MOV RREG0, #. DM. RDMSD AND OFH ;カラム・アドレスをセット
441 JTD200:
442 02F0 1A010 MOV RREG1, @RREG0 ;スペースの表示データを
443 02F1 0901A SKE RREG1, #CSPACE AND OFH ; チェック
444 02F2 0C2F5 BR JTD300
445 02F3 11001 SUB RREG0, #1H
446 02F4 0C2F0 BR JTD200
447
448 EJECT
    
```

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:25:27 12/20/93 PAGE 04-016

PROG =

SOURCE = CALC3.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
449 ;+++ 表示データの符号判定 +++
450 JTD300:
451 02F5 0901B SKE RREG1, #CMINUS AND OFH ; [-] が表示されて
452 02F6 0C2FB BR JTD400 ; いるか?
453 02F7 11001 SUB RREG0, #1H
454 BANK1
+ 1 02F8 1D791 1 MOV BANK, #01H
455 02F9 1D131 MOV RYSIGN, #CSINMN AND OFH ; Y e s
456 02FA 0C2FD BR JTD500
457 JTD400:
458 BANK1
+ 1 02FB 1D791 1 MOV BANK, #01H
459 02FC 1D130 MOV RYSIGN, #CSINOFF AND OFH ; N o
460
461 ;+++ データ転送 +++
462 JTD500:
463 BANK0
+ 1 02FD 1D790 1 MOV BANK, #00H
464 02FE 1D01E MOV RREG1, #.DM. (RYMSD-1H) AND OFH
465 JTD600:
466 02FF 1D7B1 MOV MPL, #.DM. RDMSD SHR 4 AND OFH ;表示データを浮動小数点
467 0300 1A020 MOV RREG2, @RREG0 ; レジスタ2へ転送
468 0301 1D7B1 MOV MPL, #.DM. RYMSD SHR 4 AND OFH
469 0302 167B8 OR MPL, #CSTBK1
470 0303 0A021 MOV @RREG1, RREG2
471 0304 11011 SUB RREG1, #1H
472 0305 11001 SUB RREG0, #1H
473 0306 1B008 SKLT RREG0, #.DM. RDLS D AND OFH ;転送終了か?
474 0307 0C2FF BR JTD600 ;N o →次の桁
475
476 APOPMP ;メモリ・ポインタ復帰
+ 1 0308 187AA 1 ST MPH, RREG10
+ 2 0309 187BB 1 ST MPL, RREG11
477
478 EJECT
    
```

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:25:27 12/20/93 PAGE 04-017

PROG =

SOURCE = CALC3.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
479 ;+++ 指数データ転送 +++
480 030A 08140 LD RREG0,RDEXP
481 030B 08151 LD RREG1,RDEXP+1H
482 1 BANK1
+ 1 030C 1D791 1 MOV BANK,#01H
483 030D 18140 ST RYEXP,RREG0
484 030E 18151 ST RYEXP+1H,RREG1
485
486 ;+++ ゼロ・サブレス +++
487 1 BANK0
+ 1 030F 1D790 1 MOV BANK,#00H
488 0310 1D038 MOV RREG3,#8H
489 JTD700:
490 0311 11031 SUB RREG3,#1H
491 0312 0B030 SKNE RREG3,#0H ;ゼロ・サブレス終了か?
492 0313 0C31F BR JTD800 ;Y e s
493 1 BANK1
+ 1 0314 1D791 1 MOV BANK,#01H
494 0315 081E4 LD RREG4,RYMSD-1H
495 1 BANK0
+ 1 0316 1D790 1 MOV BANK,#00H
496 0317 09040 SKE RREG4,#0H
497 0318 0C31F BR JTD800 ;Y e s
498 0319 1C28E CALL SUSHFY ;N o : 浮動小数点レジスタ2
499 ; アップ・シフト
500 1 BANK1
+ 1 031A 1D791 1 MOV BANK,#01H
501 031B 11141 SUB RYEXP,#1H
502 031C 13150 SUBC RYEXP+1H,#0H
503 1 BANK0
+ 1 031D 1D790 1 MOV BANK,#00H
504 031E 0C311 BR JTD700
505
506 ;+++ 表示データ指数エリア リセット +++
507 JTD800:
508 031F 1D148 MOV RDEXP,#CEXPINI AND OFH
509 0320 1D150 MOV RDEXP+1H,#CEXPINI SHR 4 AND OFH
510 0321 1D100 MOV RNUMC,#0H
511
512 0322 070E0 RET
513
514 EJECT
    
```

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:25:27 12/20/93 PAGE 04-018

PROG =

SOURCE = CALC3.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
515 :*****
516 :*
517 :* 表示データ出力処理 *
518 :*
519 :* INPUT : RPTLOC (小数点表示位置エリア) *
520 :* : RDLSD - RDMSD (表示データ) *
521 :* OUTPUT : LCDD0 - LCDD31 (LCD *
522 :* セグメント・データ) *
523 :*
524 :*****
525 :*****
526 :* セグメント・データ・テーブル *
527 :*****
528 0323 3232 TSEGDAT:DW 3232H ; [0]
529 0324 1010 DW 1010H ; [1]
530 0325 1322 DW 1322H ; [2]
531 0326 1132 DW 1132H ; [3]
532 0327 3110 DW 3110H ; [4]
533 0328 2132 DW 2132H ; [5]
534 0329 2332 DW 2332H ; [6]
535 032A 3012 DW 3012H ; [7]
536 032B 3332 DW 3332H ; [8]
537 032C 3132 DW 3132H ; [9]
538 032D 0000 DW 0000H ;スペース
539 032E 0100 DW 0100H ; [-]
540 032F 2322 DW 2322H ; [E]
541
542 EJECT
    
```



AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:25:27 12/20/93 PAGE 04-019

PROG =

SOURCE = CALC3.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
543 ;*****
544 ;*          初期値設定          *
545 ;*****
546 SDISP:
547          1          APUSHMP          ;メモリ・ポインタ退避
+ 1 0330 087AA 1          LD          RREG10, MPH
+ 2 0331 087BB 1          LD          RREG11, MPL
548
549 0332 1D7A8          MOV          MPH, #CSTMPE AND 0FH          ;MPEセット
550 0333 1D008          MOV          RREG0, #.DM. RDLS D AND 0FH          ;表示データ・エリアの
551 ;*****          ;カラム・アドレスをセット
552 0334 1D014          MOV          RREG1, #.DM. LCDDO SHR 4 AND 0FH          ;LCDセグメント・データ
553 0335 1D020          MOV          RREG2, #.DM. LCDDO AND 0FH          ;エリアのアドレスをセット
554 0336 1D03F          MOV          RREG3, #0FH
555 0337 01113          SUB          RREG3, RPTLOC
556
557 ;*****
558 ;*          セグメント・データ読み出し          *
559 ;*****
560 JDP200:
561 0338 1D773          MOV          ARO, #.DL. TSEGDAT AND 0FH          ;セグメント・データ・
562 0339 1D762          MOV          AR1, #.DL. TSEGDAT SHR 4 AND 0FH          ;テーブルの先頭
563 033A 1D753          MOV          AR2, #.DL. TSEGDAT SHR 8 AND 0FH          ;
564 033B 1D740          MOV          AR3, #.DL. TSEGDAT SHR 12 AND 0FH          ;アドレスをセット
565
566 033C 1D7B1          MOV          MPL, #.DM. RDLS D SHR 4 AND 0FH          ;表示データ・エリアの
567 ;*****          ;ロウ・アドレスをセット
568 033D 1A040          MOV          RREG4, @RREG0          ;表示データ読み出し
569 JDP400:
570 033E 0B040          SKNE         RREG4, #0H
571 033F 0C343          BR          JDP600
572 0340 07090          INC          AR          ;参照するセグメント・データの
573 0341 11041          SUB          RREG4, #1H          ;アドレスを指定
574 0342 0C33E          BR          JDP400
575
576 JDP600:
577 0343 07010          MOVT         DBF, @AR          ;セグメント・データ読み出し
578
579          EJECT
    
```

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:25:27 12/20/93 PAGE 04-020

PROG =

SOURCE = CALC3.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
580 ;*****
581 ;* 小数点表示位置チェック *
582 ;*****
583 0344 08004 LD RREG4,RREG0
584 0345 05034 XOR RREG4,RREG3 ;小数点表示位置か?
585 0346 0B040 SKNE RREG4,#0H
586 0347 160F1 OR DBF0,#CDPON AND 0FH ;Y e s
587
588 ;*****
589 ;* セグメント・データ出力 *
590 ;*****
591 0348 187B1 ST MPL,RREG1
592 0349 0A0F2 MOV @RREG2,DBF0
593 034A 10021 ADD RREG2,#1H
594 034B 0A0E2 MOV @RREG2,DBF1
595 034C 10021 ADD RREG2,#1H
596 034D 0A0D2 MOV @RREG2,DBF2
597 034E 10021 ADD RREG2,#1H
598 034F 0A0C2 MOV @RREG2,DBF3
599 0350 10021 ADD RREG2,#1H
600 0351 12010 ADDC RREG1,#0H
601 ;表示データ・エリアのカラム・
602 0352 10001 ADD RREG0,#1H ; アドレスをインクリメント
603 0353 09000 SKE RREG0,#0H ; データ出力終了?
604 0354 0C338 BR JDP200 ; N o → 次の桁
605
606 1 APOPMP ;メモリ・ポインタ復帰
+ 1 0355 187AA 1 ST MPH,RREG10
+ 2 0356 187BB 1 ST MPL,RREG11
607
608 0357 070E0 RET
609
610 END

```

TOTAL ERRORS = 0  
TOTAL WARNINGS = 0

END OF LIST

## 9.2 浮動小数点演算部プログラム

次に、このアプリケーション・ノートで紹介する電卓の浮動小数点演算部のプログラム・リストを示します。

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:26:25 12/20/93 PAGE 05-001

PROG =

SOURCE = ARITH.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
1          ;*****
2          ;*
3          ;*
4          ;*
5          ;* SYSTEM NAME :FLOATING POINT ARITHMETIC PACKAGE
6          ;*
7          ;*
8          ;* CPU           : μ PD17201A
9          ;*
10         ;*
11         ;* FILE NAME    : ARITH.ASM
12         ;*
13         ;*
14         ;*****
15
16         ;*****
17         ;* PUBLIC
18         ;*****
19
20         PUBLIC SFPADD,SFPDIB,SFPMULT,SFPDIV
21         PUBLIC SRXCLR
22
23         ;*****
24         ;* EXTRN
25         ;*****
26         ;***** RAMデータ *****
27
28         EXTRN MEM: RREG0,RREG1           ;汎用レジスタ
29
30
31         EXTRN MEM: RXSIGN                 ;浮動小数点レジスタ1
32         EXTRN MEM: RXEXP                 ;符号部
33         EXTRN MEM: RXLSD                 ;指数部
34         EXTRN MEM: RXMSD                 ;仮数部 最下位桁
35         EXTRN MEM: RYSIGN,RYEXP,RYLSD,RYMSD ;仮数部 最上位桁
36         EXTRN MEM: RWSIGN,RWEXP,RWLSD,RWMSD ;浮動小数点レジスタ2
37
38         ;***** フラグ *****
39
40         EXTRN FLG: FEXCHG                 ;レジスタ交換フラグ
41         EXTRN FLG: FZERO                 ;演算結果ゼロ・フラグ
42         EXTRN FLG: FOVER                 ;オーバーフロー・フラグ
43         EXTRN FLG: FDVERR                 ;除数ゼロ・フラグ
44         EJECT
    
```

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:26:25 12/20/93 PAGE 05-002

PROG =

SOURCE = ARITH.ASM

E	STNO	LOC.	OBJ.	M	I	SOURCE STATEMENT	
	45					;***** コンスタント	*****
	46						
	47				EXTRN	DAT: CGB0	;汎用レジスタ BANK 0
	48				EXTRN	DAT: CGB1	;汎用レジスタ BANK 1
	49				EXTRN	DAT: CROWG	;汎用レジスタ ロウ・アドレス 0 H
	50				EXTRN	DAT: CROWX	;汎用レジスタ ロウ・アドレス 0 H
	51				EXTRN	DAT: CROWY	;汎用レジスタ ロウ・アドレス 1 0
	52				EXTRN	DAT: CROWW	;汎用レジスタ ロウ・アドレス 2 0
	53				EXTRN	DAT: CROWXBCD	;汎用レジスタ ロウ・アドレス 0 H
	54						; BCDフラグ セット
	55				EXTRN	DAT: CIXMRY	;インデクス修飾(REGY)
	56				EXTRN	DAT: CIXMRW	;インデクス修飾(REGW)
	57				EXTRN	DAT: CIXMB1	;インデクス修飾(BANK 1)
	58				EXTRN	DAT: CIXLMANT	;インデクス修飾(仮数部 LSD)
	59				EXTRN	DAT: CIXLEXP	;インデクス修飾(指数部)
	60				EXTRN	DAT: CIXLSIGN	;インデクス修飾(符号部)
	61				EXTRN	DAT: CMDRX	;インデクス修飾される
	62						; カラム・アドレス 0 H
	63				EXTRN	DAT: CMDRY	;インデクス修飾される
	64						; カラム・アドレス 1 0 H
	65				EXTRN	DAT: CMDRW	;インデクス修飾される
	66						; カラム・アドレス 2 0 H
	67				EXTRN	DAT: CJUDGE	;指数部値の正負判定
	68				EXTRN	DAT: CCPTURN	;ビット反転マスク
	69				EXTRN	DAT: CSUBTURN	;減数符号反転
	70				EXTRN	DAT: CRXTURN	;R E G X符号反転
	71				EXTRN	DAT: CEXDIF	;指数差判定
	72				EXTRN	DAT: CSIGNCK	;符号部チェック
	73				EXTRN	DAT: CMANTCNT	;仮数部10桁カウンタ
	74				EXTRN	DAT: CEXOVER	;指数部値 オーバフロー
	75				EXTRN	DAT: CEXUNDER	;指数部値 アンダフロー
	76						
	77				EJECT		

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:26:25 12/20/93 PAGE 05-003

PROG =

SOURCE = ARITH. ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
78 ;*****
79 ;*
80 ;* 浮動小数点演算 *
81 ;* *
82 ;* (加・減算) *
83 ;* *
84 ;* REGX [I] :被加・減数 (正規化済み) *
85 ;* REGY [I] :加・減数 { " } *
86 ;* REGX [O] :演算結果 { " } *
87 ;* IXM [O] :インデクス・レジスタ *
88 ;* FEXCHG [I/O]:レジスタ交換フラグ *
89 ;* *
90 ;* 使用汎用レジスタ : RREG0,RREG1 *
91 ;* *
92 ;*****
93
94 SFPSUB:
95 ;*****
96 ;* 減算 *
97 ;*****
98 BANK1
+ 1 0358 1D791 1 1
99 0359 1D7D1 MOV BANK,#01H ;汎用レジスタ BANK 1
100 035A 1D7E4 MOV RPH,#CGB1 AND OFH ;ロウ・アドレス 20H
101 035B 08133 LD RWSIGN,RYSIGN ;減数の符号反転したものを
102 035C 15231 XOR RWSIGN,#SUBTURN AND OFH ; RWSIGNに格納
103 ; RYSIGNは保持
104 035D 0C362 BR JAD005
105
106 SFPADD:
107 ;*****
108 ;* 加算 *
109 ;*****
+ 1 035E 1D791 1 1
110 035F 1D7D1 MOV BANK,#01H ;汎用レジスタ BANK 1
111 0360 1D7E4 MOV RPL,#CROWW AND OFH ;ロウ・アドレス 20H
112 0361 08133 LD RWSIGN,RYSIGN ;RYSIGN保持
113
EJECT
    
```

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:26:25 12/20/93 PAGE 05-004

PROG =

SOURCE = ARITH. ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
114 ;***** 指数部をそろえる *****
115 JAD005:
116 0362 08044 LD RWEXP, RYEXP
117 0363 08055 LD RWEXP+1H, RXEXP+1H
118 0364 01144 SUB RWEXP, RYEXP ;
119 0365 03155 SUBC RWEXP+1H, RYEXP+1H ;RXEXP - RYEXP
120 0366 1D7D0 MOV RPH, #CGBO AND OFH ;汎用レジスタ BANK 0
121 0367 1D7E0 MOV RPL, #CROWG AND OFH ;ロウ・アドレス 0H
122 0368 1E258 SKT RWEXP+1H, #CJUDGE AND OFH ;RYEXP > RXEXP ?
123 0369 0C370 BR JAD010
124 ;RYEXPの方が大きい場合
125 036A 1524F XOR RWEXP, #CCPTURN AND OFH ;
126 036B 1525F XOR RWEXP+1H, #CCPTURN AND OFH ;
127 036C 10241 ADD RWEXP, #1H ;
128 036D 12250 ADDC RWEXP+1H, #0H ;指数差の補数をとる
129 036E 1C467 CALL SCHGXYEX ;REGX, REGYを交換
130 ; (指数部, 仮数部)
131 SETI FEXCHG ;レジスタ交換フラグ セット
+ 1 036F 16301 I OR .MF. FEXCHG SHR 4, #.DF. FEXCHG AND OFH
132 JAD010:
133 SETI CMP ;CMPフラグ セット
+ 1 0370 167F8 I OR .MF. CMP SHR 4, #.DF. CMP AND OFH
134 0371 1124A SUB RWEXP, #CEXDIF AND OFH ;
135 0372 13250 SUBC RWEXP+1H, #0H ;指数差 ≥ 10 ?
136 CLR1 CMP
+ 1 0373 147F7 I AND .MF. CMP SHR 4, #.DF. (NOT CMP AND OFH)
137
138 ;***** 大きい方の数値を答えとして終了 *****
139
140 SKFI CY
+ 1 0374 1F7F4 I SKF .MF. CY SHR 4, #.DF. CY AND OFH
141 0375 0C38B BR JAD020 ;指数差 ≤ 9
142 SKT1 FEXCHG ;レジスタ交換フラグ = 0 ?
+ 1 0376 1E301 I SKT .MF. FEXCHG SHR 4, #.DF. FEXCHG AND OFH
143 0377 070E0 RET ;REGXを答えとして終了
144 EJECT
    
```

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:26:25 12/20/93 PAGE 05-005

PROG =

SOURCE = ARITH.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
145 ;***** 転送 ( REGX → REGY ) *****
146
147 0378 1D7D1 MOV RPH,#CGB1 AND OFH ;汎用レジスタ BANK1
148 0379 1D7E0 MOV RPL,#CROWX AND OFH ;ロウ・アドレス 0H
149 037A 18144 ST RYEXP,RXEXP
150 037B 18155 ST RYEXP+1H,RXEXP+1H
151 037C 18166 ST RYLSL,RXLSL
152 037D 18177 ST RYLSL+1H,RXLSL+1H
153 037E 18188 ST RYLSL+2H,RXLSL+2H
154 037F 18199 ST RYLSL+3H,RXLSL+3H
155 0380 181AA ST RYLSL+4H,RXLSL+4H
156 0381 181BB ST RYLSL+5H,RXLSL+5H
157 0382 181CC ST RYLSL+6H,RXLSL+6H
158 0383 181DD ST RYLSL+7H,RXLSL+7H
159 0384 181EE ST RYLSL+8H,RXLSL+8H
160 0385 181FF ST RYMSD,RXMSD ;REGYに元の値を復帰
161 0386 1D7D0 MOV RPH,#CGB0 AND OFH ;汎用レジスタ BANK0
162 0387 08230 LD RREGO,RWSIGN ;
163 0388 18030 ST RXSIGN,RREGO ;退避していた結果符号を格納
164 CLR1 FEXCHG
+ 1 0389 1430E 1 AND .MF.FEXCHG SHR 4,#.DF.(NOT FEXCHG AND OFH)
165 038A 070E0 1 RET
166 JAD020:
167 ;***** 仮数部調整 *****
168
169 BANK1
+ 1 038B 1D791 1 MOV BANK,#01H
170 038C 11241 1 SUB RWEXP,#1H ;指数差デクリメント
171 SKT1 CY
+ 1 038D 1E7F4 1 SKT .MF.CY SHR 4,#.DF.CY AND OFH
172 038E 0C398 BR JAD030
173 038F 08040 LD RREGO,RXEXP ;
174 0390 08051 LD RREG1,RXEXP+1H ;
175 0391 18140 ST RYEXP,RREGO ;
176 0392 18151 ST RYEXP+1H,RREG1 ;RYEXP ← RXEXP
177 SKT1 FEXCHG ;レジスタ交換フラグ=1?
+ 1 0393 1E301 1 SKT .MF.FEXCHG SHR 4,#.DF.FEXCHG AND OFH
178 0394 0C39A BR JAD040
179 0395 1C475 CALL SCHGX ;REGX,REGYを交換(仮数部)
180 CLR1 FEXCHG
+ 1 0396 1430E 1 AND .MF.FEXCHG SHR 4,#.DF.(NOT FEXCHG AND OFH)
181 0397 0C39A BR JAD040
182 JAD030:
183 0398 1C4C1 CALL SDSHFY ;REGYを1桁ダウン・シフト
184 0399 0C38B BR JAD020
185 EJECT
    
```



AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:26:25 12/20/93 PAGE 05-006

PROG =

SOURCE = ARITH. ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
186 JAD040:
187 ;***** 符号チェック *****
188
189 BANK1
+ 1 039A 1D791 1 MOV BANK, #01H
190 039B 08230 LD RREG0, RWSIGN
191 039C 05030 XOR RREG0, RXSIGN
192 BANK0
+ 1 039D 1D790 1 MOV BANK, #00H
193 039E 1E001 SKT RREG0, #CSIGNCK AND OFH ;異符号ならスキップ
194 039F 0C3AB BR JAD999
195
196 ;***** 仮数部 計算 (異符号時) *****
197
198 03A0 1D7B1 MOV IXM, #CIXMRY AND OFH ;インデクス修飾 (REGY)
199 03A1 1C444 CALL SSUB ;仮数部減算 (REGX - REGY)
200 SKT1 CY ;減算でポローがでたか?
+ 1 03A2 1E7F4 1 SKT .MF.CY SHR 4, #.DF.CY AND OFH
201 03A3 0C3A9 BR JAD050
202 03A4 15031 XOR RXSIGN, #CRXTURN AND OFH ;符号反転
203 03A5 1C4EE CALL SRWCLR ;REGWをクリア (0)
204 03A6 1C483 CALL SCHGXW ;REGX, REGWを交換 (仮数部)
205 03A7 1D7B2 MOV IXM, #CIXMRW AND OFH ;インデクス修飾 (REGW)
206 03A8 1C444 CALL SSUB ;仮数部減算 (REGX - REGW)
207 ;仮数部の補数をとる
208 JAD050:
209 ;***** 正規化 *****
210
211 03A9 1C3FC CALL SNML ;正規化
212 03AA 070E0 RET
213 JAD999:
214 ;***** 仮数部 計算 (同符号時) *****
215
216 03AB 1D7B1 MOV IXM, #CIXMRY AND OFH ;インデクス修飾 (REGY)
217 03AC 1C42F CALL SADD ;仮数部加算 (REGX + REGY)
218 03AD 0C3A9 BR JAD050
219 EJECT
    
```

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:26:25 12/20/93 PAGE 05-007

PROG =

SOURCE = ARITH.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
220 ;*****
221 ;*
222 ;* 浮動小数点演算 *
223 ;* *
224 ;* ( 乗算 ) *
225 ;* *
226 ;* RegX [I]:被乗数 (正規化済み) *
227 ;* RegY [I]:乗数 ( " ) *
228 ;* RegX [O]:演算結果 ( " ) *
229 ;* IXM [O]:インデクス・レジスタ *
230 ;* *
231 ;* 使用汎用レジスタ : RREGO *
232 ;* *
233 ;*****
234 SFPMULT:
235 ;***** 結果符号を求める *****
236
237 ; BANK1
+ 1 03AE 1D791 1 MOV BANK,#01H
238 03AF 1D7D1 1 MOV RPH,#CGB1 AND 0FH ;汎用レジスタ BANK 1
239 03B0 1D7E0 MOV RPL,#CROWX AND 0FH ;ロウ・アドレス 0H
240 03B1 05133 XOR RXSIGN, RYSIGN ;結果符号をを求める
241 03B2 18233 ST RWSIGN, RXSIGN
242
243 ;***** 指数計算 *****
244
245 03B3 1D7D0 MOV RPH,#CGB0 AND 0FH ;汎用レジスタ BANK 0
246 03B4 1C491 CALL SUSHFX ;REGXを1桁アップ・シフト
247 ; (誤差対策)
248 03B5 1004F ADD RXEXP,#0FH ;
249 03B6 1205F ADDC RXEXP+1H,#0FH ;指数-1
250 03B7 1C459 CALL SADDEX ;指数加算
251
252 ;***** 繰り返し加算カウンタ作成 *****
253
254 03B8 1C483 CALL SCHGXW ;REGX, REGWを交換 (仮数部)
255 03B9 1D7D1 MOV RPH,#CGB1 AND 0FH ;汎用レジスタ BANK 1
256 03BA 18244 ST RWEXP, RXEXP ;
257 03BB 18255 ST RWEXP+1H, RXEXP+1H ;指数退避
258 03BC 1C4E3 CALL SRXCLR ;REGXをクリア (0)
259 03BD 08244 LD RXEXP, RWEXP ;
260 03BE 08255 LD RXEXP+1H, RWEXP+1H ;指数復帰
261 03BF 1D7D0 MOV RPH,#CGB0 AND 0FH ;汎用レジスタ BANK 0
262 03C0 1D24A MOV RWEXP,#CMANTCNT AND 0FH ;仮数部10桁カウンタ
263
EJECT
    
```

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:26:25 12/20/93 PAGE 05-008

PROG =

SOURCE = ARITH.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
264 JMT010:
265 ;***** 乗算 *****
266
267 BANK1
+ 1 03C1 1D791 1 MOV BANK, #01H
268 03C2 11241 1 SUB RWEXP, #1H
269 SKT1 CY
+ 1 03C3 1E7F4 1 SKT .MF, CY SHR 4, #. DF. CY AND OFH
270 03C4 0C3C9 BR JMT020
271 03C5 08230 LD RREG0, RWSIGN ;
272 03C6 18030 ST RXSIGN, RREG0 ;結果符号復帰
273 03C7 1C3FC CALL SNML ;正規化
274 03C8 070E0 RET
275 JMT020:
276 03C9 1C4B1 CALL SDSHFX ;REGXを1桁ダウン・シフト
277 03CA 1C4D1 CALL SDSHFW ;REGWを1桁ダウン・シフト
278 JMT030:
279 03CB 11251 SUB RWEXP+1H, #1H ;RWEXP+1 = REGW最下位桁
280 ; <加算カウンタ>
281 SKT1 CY
+ 1 03CC 1E7F4 1 SKT .MF, CY SHR 4, #. DF. CY AND OFH
282 03CD 0C3CF BR JMT999
283 03CE 0C3C1 BR JMT010
284 JMT999:
285 03CF 1D7B1 MOV IXM, #CIXMRY AND OFH ;インデクス修飾 (REGY)
286 03D0 1C42F CALL SADD ;仮数部加算 (REGX+REGY)
287 03D1 0C3CB BR JMT030
288
289 EJECT
    
```

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:26:25 12/20/93 PAGE 05-009

PROG =

SOURCE = ARITH.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
290 ;*****
291 ;*
292 ;* 浮動小数点演算 *
293 ;* *
294 ;* ( 除算 ) *
295 ;* *
296 ;* RegX [I]:被除数 (正規化済み) *
297 ;* RegY [I]:除数 { " } *
298 ;* RegX [O]:演算結果 { " } *
299 ;* FDVERR [O]:除数ゼロ・フラグ *
300 ;* IXM [O]:インデクス・レジスタ *
301 ;* *
302 ;* 使用汎用レジスタ : RREGO *
303 ;* *
304 ;*****
305 SFPDIV:
306 BANK1
+ 1 03D2 1D791 1 MOV BANK,#01H
307 CLR1 FDVERR ;除数ゼロ・フラグをクリア
+ 1 03D3 1430B 1 AND .MF.FDVERR SHR 4,#.DF.(NOT FDVERR AND OFH)
308 03D4 1D7D1 MOV RPH,#CGB1 AND OFH ;汎用レジスタ BANK 1
309 03D5 1D7E0 MOV RPL,#CROWX AND OFH ;ロウ・アドレス 0H
310 03D6 05133 XOR RXSIGN,RYSIGN ;結果符号を求める
311 03D7 1D7D0 MOV RPH,#CGB0 AND OFH ;汎用レジスタ BANK 0
312
313 ;***** 除数ゼロチェック*****
314
315 03D8 1D7A0 MOV IXH,#0H ;
316 03D9 1D7B8 MOV IXM,#CIXMB1 AND OFH ;BANK 1
317 03DA 1D7C6 MOV IXL,#CIXLMANT AND OFH ;カラム・アドレス(仮数部)
318 ; IX ← 00010000110B
319
320 JDV010:
+ 1 03DB 1D791 1 BANK1
321 MOV BANK,#01H
322 SETI IXE
+ 1 03DC 167F1 1 OR .MF.IXE SHR 4,#.DF.IXE AND OFH
322 03DD 08100 LD RREGO,.MD.CMDRY ;RREGO ← REGY
323 CLR1 IXE
+ 1 03DE 147FE 1 AND .MF.IXE SHR 4,#.DF.(NOT IXE AND OFH)
324 BANK0
+ 1 03DF 1D790 1 MOV BANK,#00H
325 03E0 09000 SKE RREGO,#0H ;仮数部 0 か?
326 03E1 0C3E8 BR JDV020 ;0ではない
327 03E2 07080 INC IX
328 03E3 097C0 SKE IXL,#0H ;1FHまで転送終了したか?
329 03E4 0C3DB BR JDV010
330 BANK1
+ 1 03E5 1D791 1 MOV BANK,#01H
331 SETI FDVERR ;除数ゼロ・フラグ
+ 1 03E6 16304 1 OR .MF.FDVERR SHR 4,#.DF.FDVERR AND OFH
332 03E7 070E0 RET
333 EJECT
    
```

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:26:25 12/20/93 PAGE 05-010

PROG =

SOURCE = ARITH. ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
334          JDV020:
335          ;***** 指数計算 *****
336
337 03E8 1C460          CALL    SSUBEX          ;指数減算
338
339          ;***** 除算 *****
340
341 03E9 1C4EE          CALL    SRWCLR          ;REGWをクリア(0)
342 03EA 1D23A          MOV     RWSIGN,#CMANTCNT AND OFH ;仮数部10桁カウンタ
343
344          JDV030:
+ 1 03EB 1D791          MOV     BANK1
345 03EC 11231          SUB     BANK,#01H
346          SKT1          SUB     RWSIGN,#1H
+ 1 03ED 1E7F4          SKT1          SKT1          CY
347 03EE 0C3F2          BR     .MF.CY SHR 4,#.DF.CY AND OFH
348 03EF 1C483          CALL   SCHGXW          ;REGX,REGWを交換(仮数部)
349 03F0 1C3FC          CALL   SNML            ;正規化
350 03F1 070E0          RET
351          JDV040:
352 03F2 1C4A1          CALL   SUSHFW          ;REGWを1桁アップ・シフト
353          JDV050:
354 03F3 1D7B1          MOV     IXM,#CIXMRY AND OFH ;インデクス修飾(REGY)
355 03F4 1C444          CALL   SSUB            ;仮数部減算(REGX-REGY)
356          SKT1          SKT1          CY
+ 1 03F5 1E7F4          SKT1          .MF.CY SHR 4,#.DF.CY AND OFH
357 03F6 0C3FA          BR     JDV999
358 03F7 1C42F          CALL   SADD            ;仮数部加算(REGX+REGY)
359 03F8 1C491          CALL   SUSHFX          ;REGXを1桁アップ・シフト
360 03F9 0C3EB          BR     JDV030
361          JDV999:
362 03FA 10261          ADD    RWLSD,#1H       ;商+1
363 03FB 0C3F3          BR     JDV050
364          EJECT
    
```

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:26:25 12/20/93 PAGE 05-011

PROG =

SOURCE = ARITH.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
365 ;*****
366 ;*
367 ;*      正規化
368 ;*
369 ;*
370 ;*      RegX      [I]:正規化前  演算データ (結果)
371 ;*      RegX      [0]:正規化済み 演算データ (結果)
372 ;*      FZERO     [0]:演算結果ゼロ・フラグ
373 ;*      FOVER     [0]:オーバーフロー・フラグ
374 ;*                   オーバフロー : 63 < 指数値
375 ;*                   アンダフロー : -64 > 指数値
376 ;*
377 ;*      使用汎用レジスタ : RREG0
378 ;*
379 ;*****
380
381 SNML:
382
+ 1 03FC 1D791 1 BANK1
383 MOV BANK,#01H
384 CLR1 FZERO ;演算結果ゼロ・フラグをクリア
+ 1 03FD 1430D 1 AND .MF.FZERO SHR 4,#.DF.(NOT FZERO AND OFH)
384 CLR1 FOVER ;オーバーフロー・フラグをクリア
+ 1 03FE 14307 1 AND .MF.FOVER SHR 4,#.DF.(NOT FOVER AND OFH)
385 BANK0
+ 1 03FF 1D790 1 MOV BANK,#00H
386 0400 1D7D0 MOV RPH,#CGB0 AND OFH ;汎用レジスタ BANK 0
387 0401 1D7E0 MOV RPL,#CROWG AND OFH ;ロウ・アドレス 0H
388
389 ;***** 仮数部 0 検査 *****
390
391 0402 1D7A0 MOV IXH,#0H ;
392 0403 1D7B8 MOV IXM,#CIXMB1 AND OFH ;BANK 1
393 0404 1D7C6 MOV IXL,#CIXLMANT AND OFH ;カラム・アドレス (仮数部)
394 ; IX ← 00010000110B
395
JNM010:
396 BANK1
+ 1 0405 1D791 1 MOV BANK,#01H
397 SET1 IXE
+ 1 0406 167F1 1 OR .MF.IXE SHR 4,#.DF.IXE AND OFH
398 0407 08000 LD RREG0,.MD.CMDRX ;RREG0 ← REGX
399 CLR1 IXE
+ 1 0408 147FE 1 AND .MF.IXE SHR 4,#.DF.(NOT IXE AND OFH)
400 BANK0
+ 1 0409 1D790 1 MOV BANK,#00H
401 040A 09000 SKE RREG0,#0H ;仮数部 0 か?
402 040B 0C410 BR JNM020 ;0 ではない
403 040C 07080 INC IX
404 040D 097C0 SKE IXL,#0H ;0FHまで転送終了したか?
405 040E 0C405 BR JNM010
406 040F 0C42C BR JNM999 ;仮数部すべて0
407 EJECT
    
```

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:26:25 12/20/93 PAGE 05-012

PROG =

SOURCE = ARITH. ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
408 JNM020:
409 ;***** 正規化 *****
410
411 BANK1
+ 1 0410 1D791 1 MOV BANK, #01H
412 0411 090F0 1 SKE RXMSD, #0H ;整数部が 0 か?
413 0412 0C419 BR JNM040 ;0 ではない
414 JNM030:
415 0413 090E0 SKE RXMSD-1H, #0H ;小数部最上位桁が 0 か?
416 0414 0C41C BR JNM050 ;0 ではない
417 0415 1C491 CALL SUSHFH ;REGXを1桁アップ・シフト
418 0416 1004F ADD REXP, #0FH ;
419 0417 1205F ADDC REXP+1H, #0FH ;指数-1
420 0418 0C413 BR JNM030
421 JNM040:
422 0419 1C4B1 CALL SDSHFH ;REGXを1桁ダウン・シフト
423 041A 10041 ADD REXP, #1H ;
424 041B 12050 ADDC REXP+1H, #0H ;指数+1
425 JNM050:
426 041C 1F058 SKF REXP+1H, #CJUDGE AND OFH ;指数は正か?
427 041D 0C425 BR JNM060
428
429 ;***** オーバフロー判定 *****
430 SET1 CMP ;CMPフラグをセット
+ 1 041E 167F8 1 OR .MF. CMP SHR 4, #. DF. CMP AND OFH
431 041F 11054 SUB REXP+1H, #CEXOVER AND OFH ;指数値 < 64 ?
432 CLR1 CMP
+ 1 0420 147F7 1 AND .MF. CMP SHR 4, #. DF. (NOT CMP AND OFH)
433 SKF1 CY
+ 1 0421 1F7F4 1 SKF .MF. CY SHR 4, #. DF. CY AND OFH
434 0422 070E0 RET
435 SET1 FOVER ;オーバフロー・フラグをセット
+ 1 0423 16308 1 OR .MF. FOVER SHR 4, #. DF. FOVER AND OFH
436 0424 070E0 RET
437 JNM060:
438 ;***** アンダフロー判定 *****
439 SET1 CMP
+ 1 0425 167F8 1 OR .MF. CMP SHR 4, #. DF. CMP AND OFH
440 0426 1105C SUB REXP+1H, #CEXUNDER AND OFH ;指数値 > -65 ?
441 CLR1 CMP
+ 1 0427 147F7 1 AND .MF. CMP SHR 4, #. DF. (NOT CMP AND OFH)
442 SKT1 CY
+ 1 0428 1E7F4 1 SKT .MF. CY SHR 4, #. DF. CY AND OFH
443 0429 070E0 RET
444 SET1 FOVER ;オーバフロー・フラグをセット
+ 1 042A 16308 1 OR .MF. FOVER SHR 4, #. DF. FOVER AND OFH
445 042B 070E0 RET
446 JNM999:
447 ;***** クリア (0) *****
448 042C 1C4E3 CALL SRXCLR ;REGXをクリア (0)
449 SET1 FZERO
+ 1 042D 16302 1 OR .MF. FZERO SHR 4, #. DF. FZERO AND OFH
450 042E 070E0 RET
451 EJECT
    
```

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:26:25 12/20/93 PAGE 05-013

PROG =

SOURCE = ARITH. ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
452 ;*****
453 ;*
454 ;* 仮数部加算
455 ;*
456 ;*
457 ;* RegX [I]:演算データ
458 ;* IXM [I]:インデクス・レジスタ
459 ;* RegX [O]:REGXとインデクス修飾されるレジスタとの
460 ;* 加算結果*
461 ;*****
462
463 SADD:
464 BANK1
+ 1 042F 1D791 1 MOV BANK,#01H
465 0430 1D7A0 MOV IXH,#0H
466 0431 1D7C0 MOV IXL,#0H
467 0432 1D7D1 MOV RPH,#CGB1 AND 0FH ;汎用レジスタ BANK 1
468 0433 1D7E1 MOV RPL,#CROWXBCD AND 0FH ;ロウ・アドレス 0H
469 ;BCDフラグをセット
470 SETI IXE
+ 1 0434 167F1 1 OR .MF.IXE SHR 4,#.DF.IXE AND 0FH
471 CLR1 CY
+ 1 0435 147FB 1 AND .MF.CY SHR 4,#.DF.(NOT CY AND 0FH)
472 0436 00066 ADD RXLSD,RXLSD
473 0437 02077 ADDC RXLSD+1H,RXLSD+1H
474 0438 02088 ADDC RXLSD+2H,RXLSD+2H
475 0439 02099 ADDC RXLSD+3H,RXLSD+3H
476 043A 020AA ADDC RXLSD+4H,RXLSD+4H
477 043B 020BB ADDC RXLSD+5H,RXLSD+5H
478 043C 020CC ADDC RXLSD+6H,RXLSD+6H
479 043D 020DD ADDC RXLSD+7H,RXLSD+7H
480 043E 020EE ADDC RXLSD+8H,RXLSD+8H
481 043F 020FF ADDC RXLSD+9H,RXLSD+9H
482 CLR1 IXE
+ 1 0440 147FE 1 AND .MF.IXE SHR 4,#.DF.(NOT IXE AND 0FH)
483 0441 1D7D0 MOV RPH,#CGB0 AND 0FH ;汎用レジスタ BANK 0
484 0442 1D7E0 MOV RPL,#CROWG AND 0FH ;ロウ・アドレス 0H
485 ;BCDフラグをクリア
486 0443 070E0 EJECT RET
487

```



AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:26:25 12/20/93 PAGE 05-014

PROG =

SOURCE = ARITH. ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
488 ;*****
489 ;*
490 ;* 仮数部減算 ;*
491 ;* ;*
492 ;* ;*
493 ;* RegX [I]:演算データ ;*
494 ;* IXM [I]:インデクス・レジスタ ;*
495 ;* RegX [O]:REGXとインデクス修飾されるレジスタとの ;*
496 ;* ;* 減算結果*
497 ;*****
498
499
500 SSUB:
+ 1 0444 1D791 1 BANK1
MOV BANK, #01H
501 0445 1D7A0 1 MOV IXH, #0H
502 0446 1D7C0 MOV IXL, #0H
503 0447 1D7D1 MOV RPH, #CGB1 AND 0FH ;汎用レジスタ BANK 1
504 0448 1D7E1 MOV RPL, #CROWXBCD AND 0FH ;ロウ・アドレス 0H
505 ; BCDフラグをセット
506 SETI IXE
+ 1 0449 167F1 1 OR .MF. IXE SHR 4, #.DF. IXE AND 0FH
507 CLR1 CY
+ 1 044A 147FB 1 AND .MF. CY SHR 4, #.DF. (NOT CY AND 0FH)
508 044B 01066 SUB RXLSD, RXLSD
509 044C 03077 SUBC RXLSD+1H, RXLSD+1H
510 044D 03088 SUBC RXLSD+2H, RXLSD+2H
511 044E 03099 SUBC RXLSD+3H, RXLSD+3H
512 044F 030AA SUBC RXLSD+4H, RXLSD+4H
513 0450 030BB SUBC RXLSD+5H, RXLSD+5H
514 0451 030CC SUBC RXLSD+6H, RXLSD+6H
515 0452 030DD SUBC RXLSD+7H, RXLSD+7H
516 0453 030EE SUBC RXLSD+8H, RXLSD+8H
517 0454 030FF SUBC RXLSD+9H, RXLSD+9H
518 CLR1 IXE
+ 1 0455 147FE 1 AND .MF. IXE SHR 4, #.DF. (NOT IXE AND 0FH)
519 0456 1D7D0 MOV RPH, #CGB0 AND 0FH ;汎用レジスタ BANK 0
520 0457 1D7E0 MOV RPL, #CROWG AND 0FH ;ロウ・アドレス 0H
521 ; BCDフラグをクリア
522 0458 070E0 RET
523 EJECT
    
```

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:26:25 12/20/93 PAGE 05-015

PROG =

SOURCE = ARITH.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
524 ;*****
525 ;*
526 ;*      指数部演算 (加算)
527 ;*
528 ;*      RegX   [I]:被加数
529 ;*      RegY   [I]:加数
530 ;*      RegX   [O]:指数部演算結果
531 ;*
532 ;*****
533 SADDEX:
534      BANK1
+ 1 0459 1D791 1      MOV      BANK,#01H
535 045A 1D7D1 1      MOV      RPH,#CGB1 AND OFH      ;汎用レジスタ BANK 1
536 045B 1D7E0      MOV      RPL,#CROWX AND OFH     ;ロウ・アドレス 0H
537 045C 00144      ADD      RXEXP,RYEXP
538 045D 02155      ADDC     RXEXP+1H,RYEXP+1H
539 045E 1D7D0      MOV      RPH,#CGB0 AND OFH     ;汎用レジスタ BANK 0
540 045F 070E0      RET
541 ;*****
542 ;*
543 ;*      指数部演算 (減算)
544 ;*
545 ;*      RegX   [I]:被減数
546 ;*      RegY   [I]:減数
547 ;*      RegX   [O]:指数部演算結果
548 ;*
549 ;*****
550 SSUBEX:
551      BANK1
+ 1 0460 1D791 1      MOV      BANK,#01H
552 0461 1D7D1 1      MOV      RPH,#CGB1 AND OFH     ;汎用レジスタ BANK 1
553 0462 1D7E0      MOV      RPL,#CROWX AND OFH     ;ロウ・アドレス 0H
554 0463 01144      SUB      RXEXP,RYEXP
555 0464 03155      SUBC     RXEXP+1H,RYEXP+1H     ;RXEXP - RYEXP
556 0465 1D7D0      MOV      RPH,#CGB0 AND OFH     ;汎用レジスタ BANK 0
557 0466 070E0      RET
558 EJECT
    
```

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:26:25 12/20/93 PAGE 05-016

PROG =

SOURCE = ARITH.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
559 *****
560 :*
561 :* レジスタ交換1 (REGX ←→ REGY)
562 :* (指数部, 仮数部)
563 :*
564 :* RegX [I]:演算データ
565 :* RegY [I]:演算データ
566 :* RegX [0]:REGY
567 :* RegY [0]:REGX
568 :*
569 :* 使用汎用レジスタ : RREG0, RREG1
570 :*
571 :*
572 *****
573
574 SCHGXYEX:
+ 1 0467 1D791 1 BANK1
575 0468 1D7A0 1 MOV BANK, #01H
576 0469 1D7B8 MOV IXH, #0H ;
577 046A 1D7C4 MOV IXM, #CIXMB1 AND OFH ;BANK 1
;カラム・アドレス (指数部)
;IX ← 00010000100B
578
579 JEXY999:
580 SET1 IXE
+ 1 046B 167F1 1 OR .MF. IXE SHR 4, #.DF. IXE AND OFH
581 046C 08000 LD RREG0, .MD. CMDRX ;RREG0 ← REGX
582 046D 08101 LD RREG1, .MD. CMDRY ;RREG1 ← REGY
583 046E 18001 ST .MD. CMDRX, RREG1 ;REGX ← REGY
584 046F 18100 ST .MD. CMDRY, RREG0 ;REGY ← REGX
585 CLR1 IXE
+ 1 0470 147FE 1 AND .MF. IXE SHR 4, #.DF. (NOT IXE AND OFH)
586 0471 07080 INC IX
587 0472 097C0 SKE IXL, #0H ;*FHまで転送終了したか?
588 0473 0C46B BR JEXY999
589 0474 070E0 RET
590
591 EJECT
    
```

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:26:25 12/20/93 PAGE 05-017

PROG =

SOURCE = ARITH.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
592 ;*****
593 ;*
594 ;* レジスタ交換 2 (REG X ←→ REG Y) ;*
595 ;* (仮数部) ;*
596 ;* ;*
597 ;* RegX [I]:演算データ ;*
598 ;* RegY [I]:演算データ ;*
599 ;* RegX [0]:REG Y ;*
600 ;* RegY [0]:REG X ;*
601 ;* ;*
602 ;* 使用汎用レジスタ : RREG0, RREG1 ;*
603 ;* ;*
604 ;*****
605
606
607 SCHGX:
+ 1 0475 1D791 1 BANK1
MOV BANK, #01H
608 0476 1D7A0 1 MOV IXH, #0H ;
609 0477 1D7B8 1 MOV IXM, #CIXMB1 AND 0FH ;BANK 1
610 0478 1D7C6 1 MOV IXL, #CIXLMANT AND 0FH ;カラム・アドレス (仮数部)
611 ; IX←00010000110B
612
613 JXY999:
+ 1 0479 167F1 1 SETI IXE
OR .MF. IXE SHR 4, #.DF. IXE AND 0FH
614 047A 08000 1 LD RREG0, .MD. CMDRX ;RREG0 ← REGX
615 047B 08101 1 LD RREG1, .MD. CMDRY ;RREG1 ← REGY
616 047C 18001 1 ST .MD. CMDRX, RREG1 ;REGX ← REGY
617 047D 18100 1 ST .MD. CMDRY, RREG0 ;REGY ← REGX
618 1 CLR1 IXE
+ 1 047E 147FE 1 AND .MF. IXE SHR 4, #.DF. (NOT IXE AND 0FH)
619 047F 07080 1 INC IX
620 0480 097C0 1 SKE IXL, #0H ;*FHまで転送終了したか?
621 0481 0C479 1 BR JXY999
622 0482 070E0 1 RET
623 EJECT
    
```

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:26:25 12/20/93 PAGE 05-018

PROG =

SOURCE = ARITH.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
624
625 ;*****
626 ;*
627 ;* レジスタ交換3 (REGX ↔ REGW) *
628 ;* (仮数部) *
629 ;*
630 ;* RegX [I]:演算データ *
631 ;* RegW [I]:演算用ワーク・エリア *
632 ;* RegX [O]:REGW *
633 ;* RegW [O]:REGX *
634 ;*
635 ;* 使用汎用レジスタ : RREG0,RREG1 *
636 ;*
637 ;*****
638
639 SCHGXW:
640 BANK1
+ 1 0483 1D791 1 MOV BANK,#01H
641 0484 1D7A0 1 MOV IXH,#0H
642 0485 1D7B8 MOV IXH,#CIXMB1 AND OFH ;BANK1
643 0486 1D7C6 MOV IXL,#CIXLMANT AND OFH ;カラム・アドレス(仮数部)
644 ; IX←00010000110B
645
646 JXW010:
+ 1 0487 167F1 1 SETI IXE
647 0488 08000 OR .MF.IXE SHR 4,#.DF.IXE AND OFH
648 0489 08201 LD RREG0,.MD.CMDRX ;RREG0 ← REGX
649 048A 18001 LD RREG1,.MD.CMDRW ;RREG1 ← REGW
650 048B 18200 ST .MD.CMDRX,RREG1 ;REGX ← REGW
651 CLR1 IXE ;MD.CMDRW,RREG0 ;REGW ← REGX
+ 1 048C 147FE 1 AND .MF.IXE SHR 4,#.DF.(NOT IXE AND OFH)
652 048D 07080 INC IX
653 048E 097C0 SKE IXL,#0H ;*FHまで転送終了したか?
654 048F 0C487 BR JXW010
655 0490 070E0 RET
656 EJECT
    
```

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:26:25 12/20/93 PAGE 05-019

PROG =

SOURCE = ARITH.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
657 *****
658 ;*
659 ;* アップ・シフト・レジスタ 1 (REGX) ;*
660 ;* ;*
661 ;* RegX [I]:演算データ(結果) ;*
662 ;* RegX [O]:アップ・シフト済み ;*
663 ;* ;*
664 *****
665
666 SUSHFX:
667 BANK1
+ 1 0491 1D791 1 MOV BANK,#01H
668 0492 1D7D1 1 MOV RPH,#CGB1 AND 0FH ;汎用レジスタ BANK1
669 0493 1D7E0 MOV RPL,#CROWX AND 0FH ;ロウ・アドレス 0H
670 0494 080EF LD RXMSD,RXMSD-1H ;
671 0495 080DE LD RXMSD-1H,RXMSD-2H ;
672 0496 080CD LD RXMSD-2H,RXMSD-3H ;
673 0497 080BC LD RXMSD-3H,RXMSD-4H ;
674 0498 080AB LD RXMSD-4H,RXMSD-5H ;
675 0499 0809A LD RXMSD-5H,RXMSD-6H ;
676 049A 08089 LD RXMSD-6H,RXMSD-7H ;
677 049B 08078 LD RXMSD-7H,RXMSD-8H ;
678 049C 08067 LD RXMSD-8H,RXMSD-9H ;アップ・シフト
679 049D 1D060 MOV RXLSD,#0H ;LSD ← 0
680 049E 1D7D0 MOV RPH,#CGB0 AND 0FH ;汎用レジスタ BANK0
681 049F 1D7E0 MOV RPL,#CROWG AND 0FH ;ロウ・アドレス 0H
682 04A0 070E0 RET
683 EJECT
    
```

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:26:25 12/20/93 PAGE 05-020

PROG =

SOURCE = ARITH. ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
684 *****
685 ;*
686 ;* アップ・シフト・レジスタ2 (REGW) ;*
687 ;* ;*
688 ;* RegW [I]:演算データ(結果) ;*
689 ;* RegW [O]:アップ・シフト済み ;*
690 ;* ;*
691 *****
692
693 SUSHFW:
694 BANK1
+ 1 04A1 1D791 1 MOV BANK,#01H
695 04A2 1D7D1 1 MOV RPH,#CGB1 AND 0FH ;汎用レジスタ BANK1
696 04A3 1D7E4 MOV RPL,#CROWW AND 0FH ;ロウ・アドレス 2H
697 04A4 082EF LD RWMSD,RWMSD-1H ;
698 04A5 082DE LD RWMSD-1H,RWMSD-2H ;
699 04A6 082CD LD RWMSD-2H,RWMSD-3H ;
700 04A7 082BC LD RWMSD-3H,RWMSD-4H ;
701 04A8 082AB LD RWMSD-4H,RWMSD-5H ;
702 04A9 0829A LD RWMSD-5H,RWMSD-6H ;
703 04AA 08289 LD RWMSD-6H,RWMSD-7H ;
704 04AB 08278 LD RWMSD-7H,RWMSD-8H ;
705 04AC 08267 LD RWMSD-8H,RWMSD-9H ;アップ・シフト
706 04AD 1D260 MOV RWLSD,#0H ;LSD ← 0
707 04AE 1D7D0 MOV RPH,#CGB0 AND 0FH ;汎用レジスタ BANK0
708 04AF 1D7E0 MOV RPL,#CROWG AND 0FH ;ロウ・アドレス 0H
709 04B0 070E0 RET
710 EJECT
    
```

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:26:25 12/20/93 PAGE 05-021

PROG =

SOURCE = ARITH.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
711 ;*****
712 ;*
713 ;*      ダウン・シフト・レジスタ 1 (REGX)
714 ;*
715 ;*      RegX   [I]:演算データ
716 ;*      RegX   [0]:ダウン・シフト済み
717 ;*
718 ;*****
719
720 SDSHFX:
721          BANK1
+ 1 04B1 1D791 1 1  MOV      BANK, #01H
722 04B2 1D7D1      MOV      RPH, #CGB1 AND 0FH      ;汎用レジスタ BANK 1
723 04B3 1D7E0      MOV      RPL, #CROWX AND 0FH    ;ロウ・アドレス 0H
724 04B4 08076      LD        RXLSD, RXLSD+1H
725 04B5 08087      LD        RXLSD+1H, RXLSD+2H
726 04B6 08098      LD        RXLSD+2H, RXLSD+3H
727 04B7 080A9      LD        RXLSD+3H, RXLSD+4H
728 04B8 080BA      LD        RXLSD+4H, RXLSD+5H
729 04B9 080CB      LD        RXLSD+5H, RXLSD+6H
730 04BA 080DC      LD        RXLSD+6H, RXLSD+7H
731 04BB 080ED      LD        RXLSD+7H, RXLSD+8H
732 04BC 080FE      LD        RXLSD+8H, RXLSD+9H
733 04BD 1D0F0      MOV      RXMSD, #0H      ;ダウン・シフト
734 04BE 1D7D0      MOV      RPH, #CGBO AND 0FH  ;MSD ← 0
735 04BF 1D7E0      MOV      RPL, #CROWG AND 0FH ;汎用レジスタ BANK 0
736 04C0 070E0      RET
737          EJECT

```



AS17K VL.10 V4 << D17201A ASSEMBLE LIST >> 11:26:25 12/20/93 PAGE 05-022

PROG =

SOURCE = ARITH. ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
738 ;*****
739 ;*
740 ;*      ダウン・シフト・レジスタ 2 (REGY)
741 ;*
742 ;*      RegY  [I]:演算データ
743 ;*      RegY  [O]:ダウン・シフト済み
744 ;*
745 ;*****
746
747 SDSHFY:
748
+ 1 04C1 1D791 1 BANK1
749 04C2 1D7D1 1 MOV BANK, #01H
750 04C3 1D7E2 MOV RPL, #CROWY AND OFH ;汎用レジスタ BANK 1
751 04C4 08176 LD RYLS, RYLS+1H ;ロウ・アドレス 1 H
752 04C5 08187 LD RYLS+1H, RYLS+2H
753 04C6 08198 LD RYLS+2H, RYLS+3H
754 04C7 081A9 LD RYLS+3H, RYLS+4H
755 04C8 081BA LD RYLS+4H, RYLS+5H
756 04C9 081CB LD RYLS+5H, RYLS+6H
757 04CA 081DC LD RYLS+6H, RYLS+7H
758 04CB 081ED LD RYLS+7H, RYLS+8H
759 04CC 081FE LD RYLS+8H, RYLS+9H ;ダウン・シフト
760 04CD 1D1F0 MOV RYMS, #0H ;MSD ← 0
761 04CE 1D7D0 MOV RPH, #CGB0 AND OFH ;汎用レジスタ BANK 0
762 04CF 1D7E0 MOV RPL, #CROWG AND OFH ;ロウ・アドレス 0 H
763 04D0 070E0 RET
764 EJECT
    
```

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:26:25 12/20/93 PAGE 05-023

PROG =

SOURCE = ARITH. ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
765 ;*****
766 ;*
767 ;*      ダウン・シフト・レジスタ 3 (REGW)
768 ;*
769 ;*      RegW  [I]:演算データ・ワーク・エリア
770 ;*      RegW  [O]:ダウン・シフト済み
771 ;*      RWEXP+1H[O]:ダウン・シフトによってはじきだされた最下位桁*
772 ;*
773 ;*      使用汎用レジスタ : RREGO
774 ;*
775 ;*****
776
777
778          SDSHFV:
+ 1 04D1 1D791 1 BANK1
779 04D2 08260 1 MOV      BANK, #01H
780 04D3 18250 LD      RREGO, RWLSD
781 04D4 1D7D1 ST      RWEXP+1H, RREGO ;最下位桁退避
782 04D5 1D7E4 MOV     RPH, #CGB1 AND 0FH ;汎用レジスタ BANK 1
783 04D6 08276 MOV     RPL, #CROW AND 0FH ;ロウ・アドレス 2H
784 04D7 08287 LD      RWLSD, RWLSD+1H
785 04D8 08298 LD      RWLSD+1H, RWLSD+2H
786 04D9 082A9 LD      RWLSD+2H, RWLSD+3H
787 04DA 082BA LD      RWLSD+3H, RWLSD+4H
788 04DB 082CB LD      RWLSD+4H, RWLSD+5H
789 04DC 082DC LD      RWLSD+5H, RWLSD+6H
790 04DD 082ED LD      RWLSD+6H, RWLSD+7H
791 04DE 082FE LD      RWLSD+7H, RWLSD+8H
792 04DF 1D2F0 LD      RWLSD+8H, RWLSD+9H ;ダウン・シフト
793 04E0 1D7D0 MOV     RWMSD, #0H ;MSD ← 0
794 04E1 1D7E0 MOV     RPH, #CGB0 AND 0FH ;汎用レジスタ BANK 0
795 04E2 070E0 MOV     RPL, #CROWG AND 0FH ;ロウ・アドレス 0H
796          RET
EJECT
    
```

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:26:25 12/20/93 PAGE 05-024

PROG =

SOURCE = ARITH.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
797 ;*****
798 ;*
799 ;* レジスタ クリア (0) (REGX) ;*
800 ;* (符号部, 指数部, 仮数部) ;*
801 ;*
802 ;* RegX [I]:演算データ(結果) ;*
803 ;* RegX [0]:クリア(0)済み ;*
804 ;*
805 ;*****
806
807 SRXCLR:
808 1 BANK1
+ 1 04E3 1D791 1 MOV BANK,#01H
809 04E4 1D7A0 MOV IXH,#0H
810 04E5 1D7B8 MOV IXM,#CIXMB1 AND OFH ;BANK1
811 04E6 1D7C3 MOV IXL,#CIXLSIGN AND OFH ;カラム・アドレス(符号部)
812 ;IX←00010000011B
813
814 JXC999:
+ 1 04E7 167F1 1 SETI IXE
815 04E8 1D000 1 OR .MF.IXE SHR 4,#.DF.IXE AND OFH
816 CLR1 IXE ;クリア(0)
+ 1 04E9 147FE 1 AND .MF.IXE SHR 4,#.DF.(NOT IXE AND OFH)
817 04EA 07080 INC IX
818 04EB 097C0 SKE IXL,#0H ;0FHまで転送終了したか?
819 04EC 0C4E7 BR JXC999
820 04ED 070E0 RET
821 EJECT
    
```

AS17K V1.10 V4 << D17201A ASSEMBLE LIST >> 11:26:25 12/20/93 PAGE 05-025

PROG =

SOURCE = ARITH.ASM

```

E STNO LOC. OBJ. M I SOURCE STATEMENT
822 ;*****
823 ;*
824 ;* レジスタ クリア (0) (REGW) *
825 ;* (仮数部) *
826 ;*
827 ;* RegW [I]:演算用ワークエリア *
828 ;* RegW [0]:クリア (0) 済み *
829 ;*
830 ;*****
831
832 SRWCLR:
833 BANK1
+ 1 04EE 1D791 1 MOV BANK, #01H
834 04EF 1D7A0 1 MOV IXH, #0H
835 04F0 1D7B8 MOV IXM, #CIXMB1 AND OFH ;BANK 1
836 04F1 1D7C6 MOV IXL, #CIXLMANT AND OFH ;カラム・アドレス (仮数部)
837 ;IX←00010000110B
838
839 JWC999:
+ 1 04F2 167F1 1 SET1 IXE
840 04F3 1D200 1 OR .MF. IXE SHR 4, #.DF. IXE AND OFH
841 CLR1 IXE ;クリア (0)
+ 1 04F4 147FE 1 AND .MF. IXE SHR 4, #.DF. (NOT IXE AND OFH)
842 04F5 07080 INC IX
843 04F6 097C0 SKE IXL, #0H ;2FHまで転送終了したか?
844 04F7 0C4F2 BR JWC999
845 04F8 070E0 RET
846
847 END
    
```

TOTAL ERRORS = 0  
TOTAL WARNINGS = 0

END OF LIST

## アンケート記入のお願い

お手数ですが、このドキュメントに対するご意見をお寄せください。今後のドキュメント作成の参考にさせていただきます。

[ドキュメント名] 17Kシリーズ アプリケーション・ノート 浮動小数点演算パッケージ  
(IEA-757 (第1版))

[お名前など] (さしつかえのない範囲で)

御社名(学校名, その他) ( )  
 ご住所 ( )  
 お電話番号 ( )  
 お仕事の内容 ( )  
 お名前 ( )

1. ご評価 (各欄に○をご記入ください)

項 目	大変良い	良 い	普 通	悪 い	大変悪い
全体の構成					
説明内容					
用語解説					
調べやすさ					
デザイン, 字の大きさなど					
その他 ( )					
( )					

2. わかりやすい所 (第 章, 第 章, 第 章, 第 章, その他 )  
理由 [ ]

3. わかりにくい所 (第 章, 第 章, 第 章, 第 章, その他 )  
理由 [ ]

4. ご意見, ご要望

5. このドキュメントをお届けしたのは  
NEC販売員, 特約店販売員, NEC半応技本部長, その他 ( )

ご協力ありがとうございました。

下記あてにFAXで送信いただくか, 最寄りの販売員にコピーをお渡してください。

NEC半導体インフォメーションセンター  
FAX: (044) 548-7900

— お問い合わせは、最寄りの NEC へ —

【営業関係お問い合わせ先】

コンシューマ半導体販売事業部 OA半導体販売事業部 インダストリー半導体販売事業部	〒108-01 東京都港区芝五丁目7番1号 (NEC本社ビル)	東京 (03)3454-1111 (大代表)
中部支社 半導体販売部	〒460 名古屋市中区栄四丁目14番5号 (松下中日ビル)	名古屋 (052)242-2755
関西支社 半導体第一販売部 半導体第二販売部 半導体第三販売部	〒540 大阪市中央区城見一丁目4番24号 (NEC関西ビル)	大阪 (06) 945-3178 大阪 (06) 945-3200 大阪 (06) 945-3208
北海道支社 東北支社 岩手支店 山形支店 郡山支店 いわき支店 長岡支店 水戸支店 神奈川支社 群馬支店 太田支店 宇都宮支店	札幌 (011)231-0161 仙台 (022)261-5511 盛岡 (0196)51-4344 山形 (0236)23-5511 郡山 (0249)23-5511 いわき (0246)21-5511 長岡 (0258)36-2155 水戸 (0292)26-1717 横浜 (045)324-5511 高崎 (0273)26-1255 太田 (0276)46-4011 宇都宮 (0286)21-2281	小山支店 (0285)24-5011 長野支社 (0262)35-1444 本松支店 (0263)35-1666 諏訪支店 (0266)53-5350 上甲支店 (0552)24-4141 埼玉支社 (048)641-1411 立川支社 (0425)26-5981 千葉支社 (043)238-8116 静岡支社 (054)255-2211 沼津支店 (0559)63-4455 浜松支店 (053)452-2711 北陸支社 (0762)23-1621
福井支店 富山支店 京都支社 神戸支社 中国支社 鳥取支店 岡山支店 四国支社 新居浜支店 松山支店 九州支社 北九州支店	福井 (0776)22-1866 富山 (0764)31-8461 京都 (075)344-7824 神戸 (078)332-3311 中国 (082)242-5504 鳥取 (0857)27-5311 岡山 (086)225-4455 高松 (0878)36-1200 新居浜 (0897)32-5001 松山 (0899)45-4111 福岡 (092)271-7700 北九州 (093)541-2887	

【本資料に関する技術お問い合わせ先】

半導体応用技術本部 マイクロコンピュータ技術部	〒210 川崎市幸区塚越三丁目484番地	川崎 (044)548-7923	半導体 インフォメーションセンター FAX(044)548-7900 (FAXにてお願い致します)
半導体応用技術本部 中部応用システム技術部	〒460 名古屋市中区栄四丁目14番5号 (松下中日ビル)	名古屋 (052)242-2762	
半導体応用技術本部 西日本応用システム技術部	〒540 大阪市中央区城見一丁目4番24号 (NEC関西ビル)	大阪 (06) 945-3383	