

ソフトウェアマニュアル
概説&マクロ仕様

コンパクトPMS V5

HIDIC
S10 シリーズ

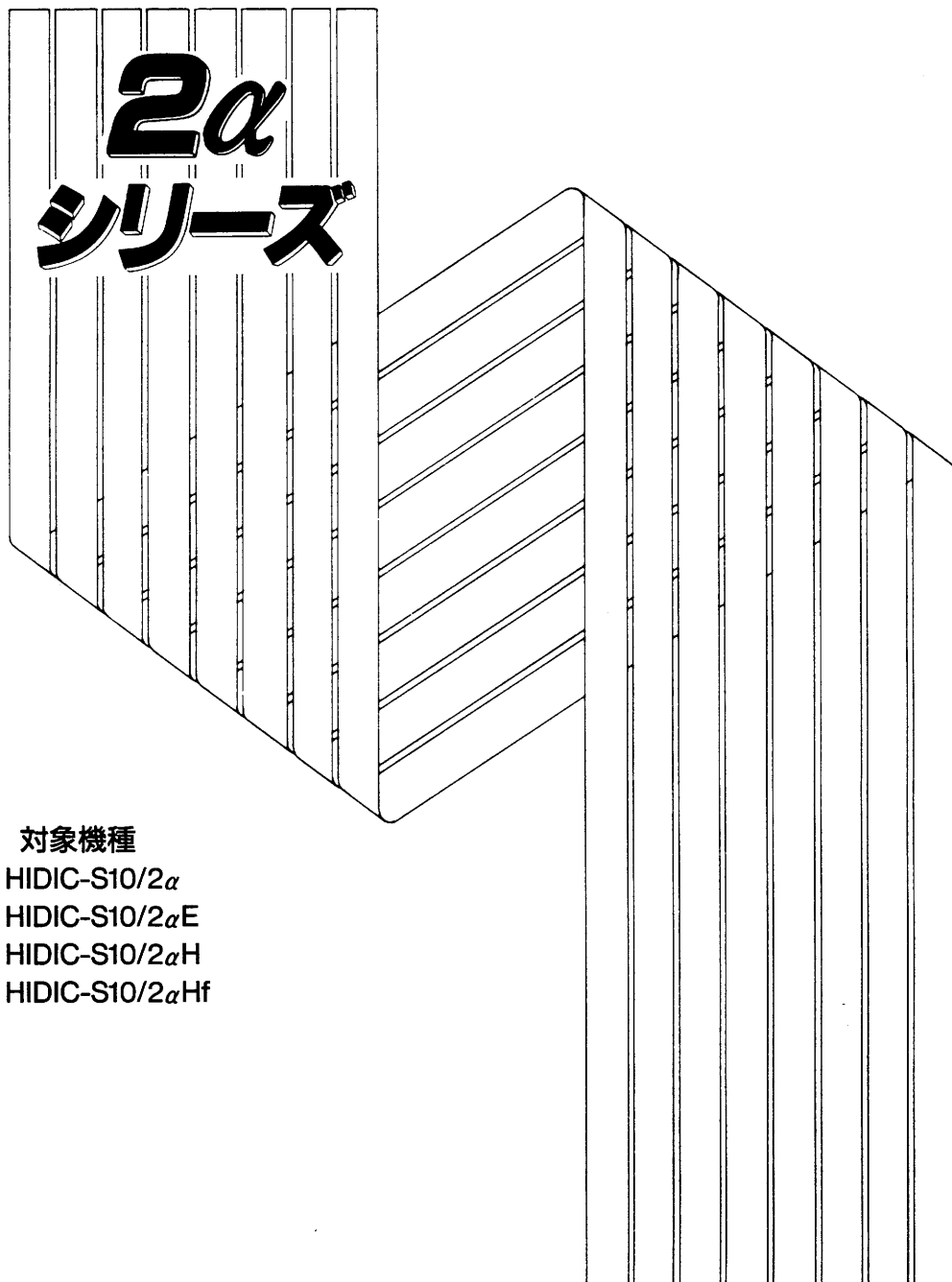
2α
シリーズ

対象機種
HIDIC-S10/2
HIDIC-S10/2 E
HIDIC-S10/2 H
HIDIC-S10/2 Hf



ソフトウェアマニュアル
概説&マクロ仕様

コンパクトPMS V5



対象機種
HIDIC-S10/2α
HIDIC-S10/2αE
HIDIC-S10/2αH
HIDIC-S10/2αHf

HITACHI

本製品を輸出される場合には、外国為替及び外国貿易法の規制並びに米国輸出管理規則など外国の輸出関連法規をご確認の上、必要な手続きをお取りください。
なお、不明な場合は、弊社担当営業にお問合わせください。

1987年 8月 (第1版) SP-3-020 (廃版)
1990年 5月 (第2版) SP-3-120 (廃版)
1990年 9月 (第3版) SP-3-220 (廃版)
1995年 5月 (第4版) SAJ-3-201 (A)

- このマニュアルの一部、または全部を無断で転写したり複写することは、固くお断りいたします。
- このマニュアルの内容を、改良のため予告なしに変更することがあります。

はじめに

このたびは、日立プログラマブルコントローラ（PCs）をお求めいただきありがとうございます。

このマニュアルは、HIDIC S10/2 α シリーズ（2 α シリーズ）のシーケンサ機能にリアルタイム性およびマルチタスキング性を拡充したOS、コンパクトPMSについて解説しています。

このコンパクトPMSを使用することにより、2 α シリーズを本格的にマイコンとして幅広く制御に活用できます。

<マニュアル構成>

1 概 説

コンパクトPMSの全体の構成について説明しています。

2 タスク管理

タスクのスケジューリング、動きなどリアルタイムシステムを構築するうえで必要な機能について説明しています。

3 システム管理

システムの立上げ処理や異常発生の処理、およびユーザ固有の処理を行うための組込みサブルーチンについて説明しています。

4 マクロ仕様

マクロ命令の仕様、リンケージ方法について説明しています。

付 録

一覧表など補足資料を記載しています。

このマニュアルは下記バージョンのシステムF/Dに対応しています。

対象ツール	システムF/D名称	バ ー ジ ョ ン
PSE α	Compact PMS SYS	Ver. 5.0 Rev. 0.0以降
PC98	CPMSロードシステム	Ver. 4.2 Rev. 0.0以降
	CPMSEロードシステム	Ver. 2.2 Rev. 0.0以降

コンパクトPMSのバージョンとサポートマクロ命令の関係を下表に示します。

マクロ命令	CPMSバージョン		
	V1	V4	V5
r l e a s	○	○	○
q u e u e	○	○	○
a b o r t	○	○	○
d e l a y	○	○	○
t i m e r	○	○	○ *1
c t i m e	○	○	○
c h a p	○	○	○
c h m o d	○	○	○
s f a c t	○	○	○
g f a c t	○	○	○
u s p c h k	○	○	○
s t i m e		○	○
g t i m e		○	○
w a k e		○	○
c w a k e		○	○
r s e r v		○	○
f r e e		○	○
m v m e m		○	○

○印：サポートマクロ命令

*1：機能追加あり

目 次

1 概 説	
1.1 コンパクトPMS	2
1.2 ハードウェア	2
1.3 ソフトウェア	2
1.3.1 プログラムの種類	2
1.3.2 プログラムのローディング	3
1.3.3 プログラムのチェック	3
1.4 2 α シリーズのOSとコンパクトPMSの関係	4
2 タスク管理	
2.1 タスクの構成	8
2.1.1 動 作	9
2.1.2 コンパクトPMSとタスクのリンケージ	10
2.1.3 タスクのレベル	11
2.1.4 サブルーチン	12
2.1.5 データテーブル	13
2.2 タスクの状態	14
2.3 タスクの動作	16
3 システム管理	
3.1 システム立上げ	20
3.2 異常処理	20
3.3 組込みサブルーチン	21
3.3.1 種 類	21
3.3.2 登 録	22
3.3.3 入力情報	22
3.4 エラーログ	23
3.5 D H P	29
3.6 浮動小数点演算機能	30
3.6.1 浮動小数点演算使用時の注意	31
3.6.2 浮動小数点用エラーログ	33
4 マクロ仕様	
4.1 マクロ命令	36
4.1.1 マクロライブラリ	36
4.1.2 一般規則	37
4.1.3 パラメータチェック	39

4. 2 種 類	40
4. 2. 1 r l e a s	41
4. 2. 2 q u e u e	42
4. 2. 3 a b o r t	43
4. 2. 4 d e l a y	44
4. 2. 5 t i m e r	45
4. 2. 6 c t i m e	46
4. 2. 7 c h a p	47
4. 2. 8 c h m o d	48
4. 2. 9 s f a c t	49
4. 2. 10 g f a c t	50
4. 2. 11 u s p c h k	51
4. 2. 12 s t i m e	52
4. 2. 13 g t i m e	55
4. 2. 14 w a k e	57
4. 2. 15 c w a k e	60
4. 2. 16 r s e r v	61
4. 2. 17 f r e e	63
4. 2. 18 m v m e m	65
4. 3 補 足	67
4. 3. 1 r s e r vとf r e eの関連	67
4. 3. 2 w a k eとc w a k eの関連	68
4. 3. 3 s t i m eとg t i m eの関連	69
4. 4 マクロ命令使用例	70

5 付 録

付録1 H-S10/2αシリーズの拡張メモリ割付	82
付録2 プロテクトキースイッチについて	83
付録3 DHP情報一覧	84
付録4 マクロ命令処理時間一覧	85

1 概 説

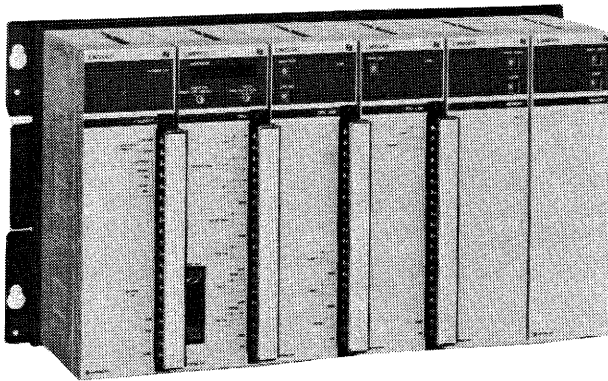
1.1 コンパクトPMS

コンパクトPMSは、時々刻々変化する現場信号をリアルタイムに同時処理する制御用システムに優れた機能を発揮するリアルタイムOSです。

1.2 ハードウェア

コンパクトPMSを使用する場合は拡張メモリが必要となります。拡張メモリにはプログラムのデバッグを行うデバッガとユーザー作成のプログラムが入ります。プログラムの量によりメモリ容量を選択してください。

尚、将来HI-FLOWシステムを使用する計画がある場合、ユーザー作成プログラムは先頭256Kバイト以降をご使用ください。



・写真はオプションモジュールのCPU間リンク、PSEリンクも実装されています。

■構成例

拡張マウントベース : HPC-1000
 CPU電源モジュール : LWV000
 CPUモジュール : LWP000(2α)
 拡張メモリ : LUM△△△

1.3 ソフトウェア

1.3.1 プログラムの種類

コンパクトPMSの下で実行できるプログラム言語は次の3種類があります。システムの処理時間、ユーザーのプログラムの作成し易さ、慣れ等を考慮して選択してください。

- 68000アセンブラ言語
- C言語
- F A - B A S I C言語

命令語の
処理スピード

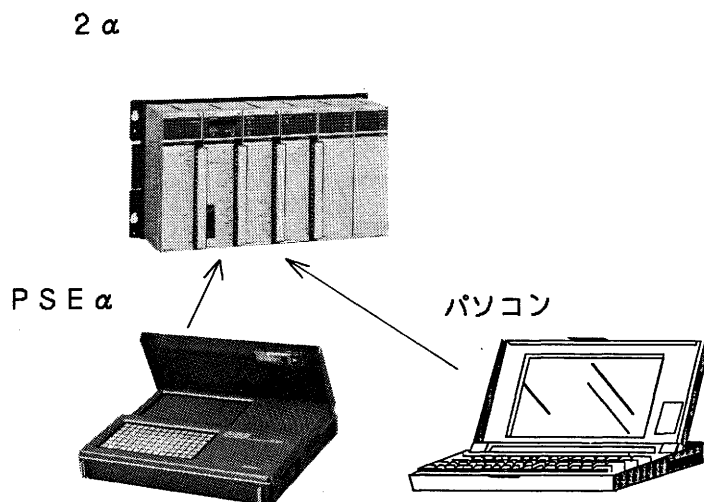
プログラムの
作成し易さ



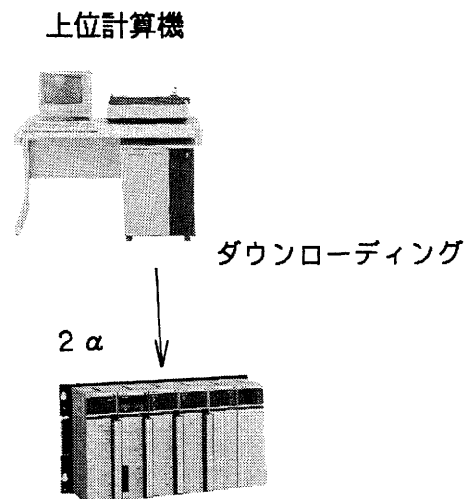
1.3.2 プログラムのローディング

プログラムはツールによりローディングする方法と、上位計算機（V90シリーズなど）よりダウンロードする方法があります。

▼ ツールより

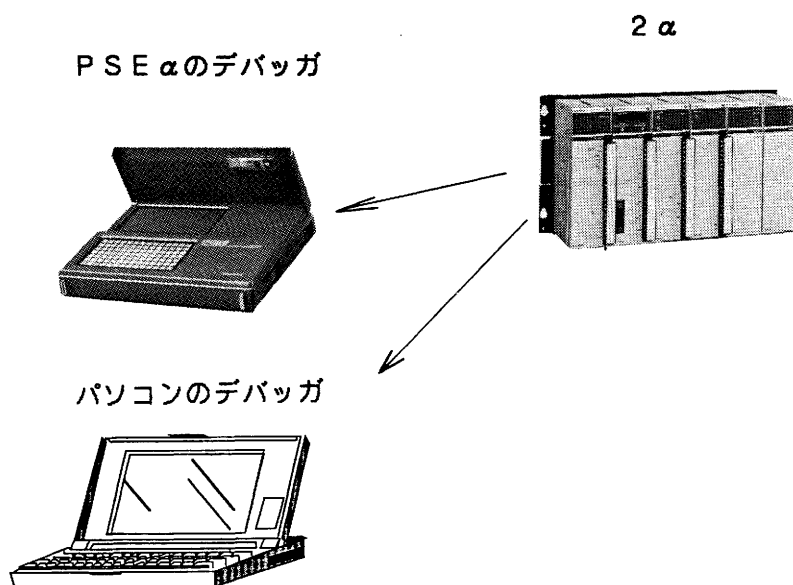


▼ 上位計算機より



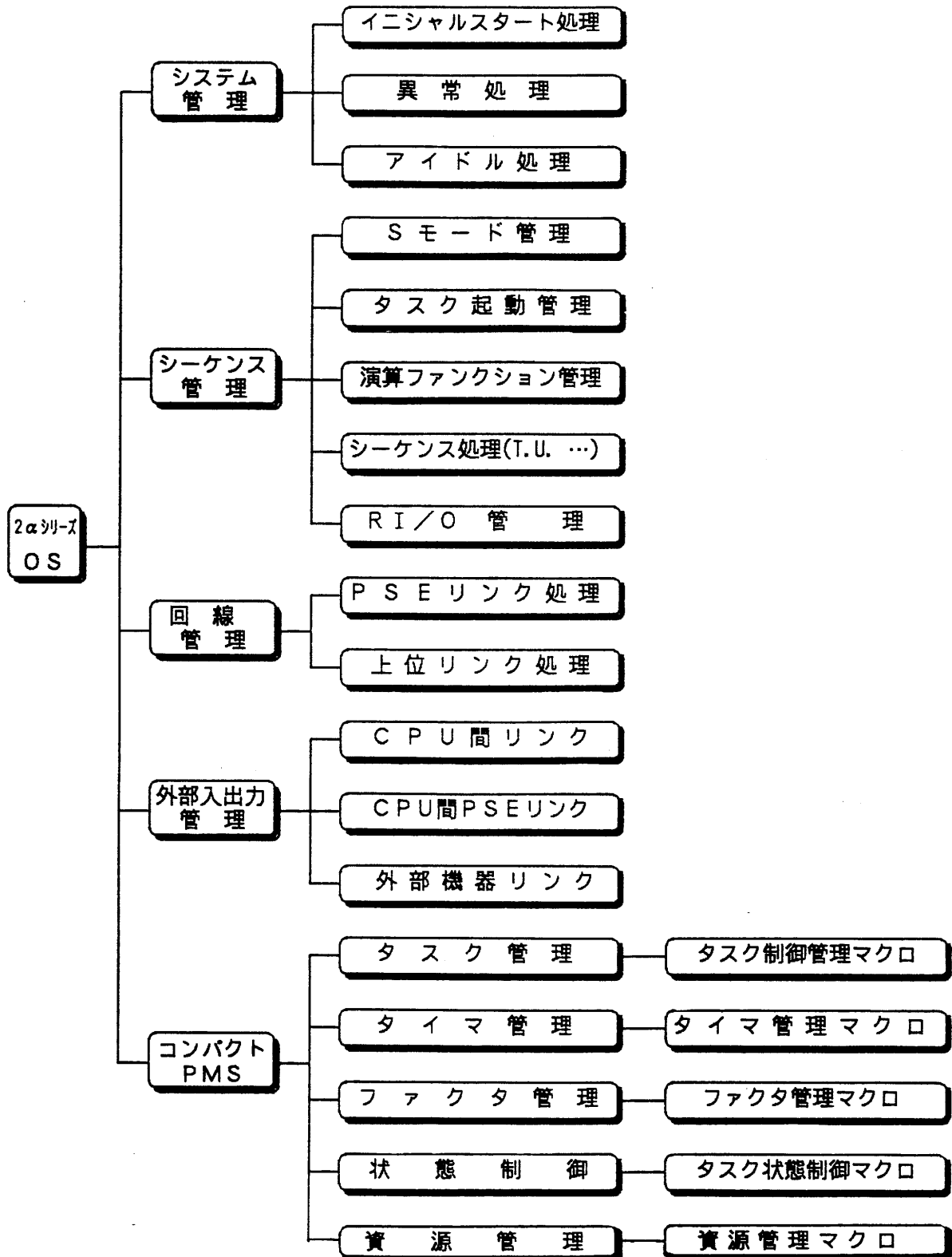
1.3.3 プログラムのチェック

ユーザーが作成したプログラムはツールのデバッガにより効率よくチェックする事ができます。



1.4 2αシリーズのOSとコンパクトPMSの関係

コンパクトPMSは2αシリーズ(2α, 2αE, 2αH, 2αHf)のOSの下で動く、リアルタイムOSです。このコンパクトPMSはユーザーが作成したタスクの動作管理を行います。



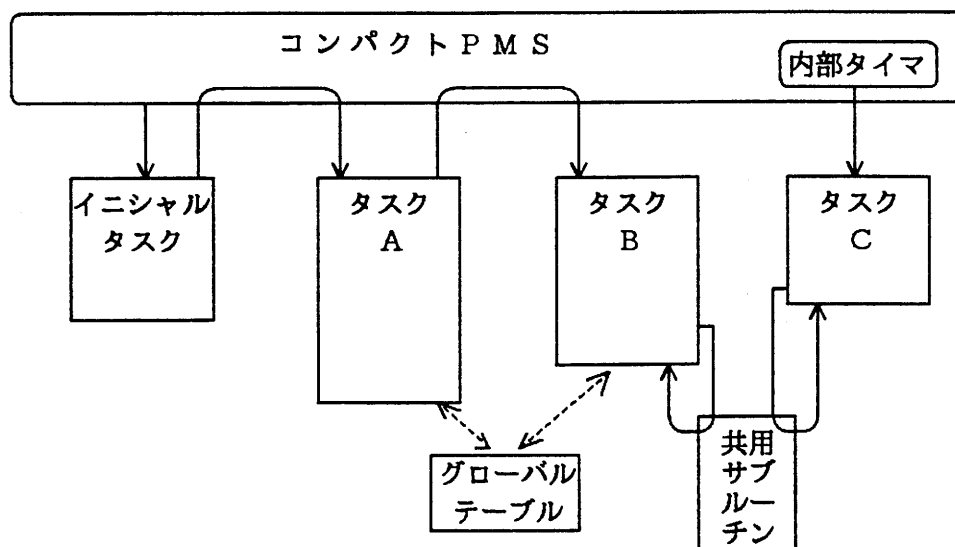
●タスク：コンパクトPMSのもとで実行される制御の流れに対応したプログラムの最小単位

- **イニシャルスタート処理**
復電時にシステムテーブル及びハードレジスタの初期化を行います。
- **異常処理**
CPU異常（エラー）が発生した時に、異常データを収集するとともに他の処理が続行可能な状態とします。
- **アイドル処理**
CPU異常及びハード異常時の情報をコード化し、CPUコンソールLEDに表示します。またアイドル時間のカウントを行います。
- **Sモード管理**
Sモードプログラムの起動、再起動管理を行います。
- **タスク起動管理**
Pコイル（Sモードプログラムより起動）により起動されるタスクのスケジューリングを行います。
- **演算ファンクション管理**
演算命令により起動される演算ファンクションを実行します。
- **シーケンス処理**
シーケンス制御に必要なタイマ（T），ワンショット（U），等の制御を行います。
- **R I/O管理**
R I/Oの転送終了を受け、R I/Oの状態データを収集するとともに起動を行います。
- **PSEリンク処理**
PSE専用回線のH-7338手順の制御を行います。
- **上位リンク処理**
上位計算機専用回線のH-7338手順の制御を行います。
- **CPU間リンク**
グローバルエリア（P I/OのGエリア）の内容を他CPU（ 2α ， $2\alpha E$ ， $2\alpha H$ ， $2\alpha Hf$ ）へ指定されたアドレスより指定された語数を転送します。
- **CPU間PSEリンク**
1台のPSEより複数のCPU（ 2α ， $2\alpha E$ ， $2\alpha H$ ， $2\alpha Hf$ ）にリンケージします。
- **外部機器リンク**
外部機器（パソコン，CRT，T/W，etc）とリンケージする為RS-422インターフェイスを持つ事が出来ます。外部機器により手順を選択する事が可能です。
- **タスク管理**
タスクの起動，再起動及びタスクのIdle，Dormant状態を管理します。タスクの起動，再起動に関しては、タスクレベルによりスケジューリングを行います。
- **タイマ管理**
周期起動の登録，周期起動のキャンセル，遅延処理を行います。
- **ファクタ管理**
タスクの起動要因（ファクタ）のセット，取り出し後のリセットを行います。
- **状態制御**
実行レベルの変更（ハードレベル）及びタスクレベルの変更を行います。
- **資源管理**
ユーザが定義したタスク間共有資源の排他的制御を行います。

2 タスク管理

2.1 タスクの構成

ユーザーが作成したプログラムは次のような構成になります。



■ タスクナンバー

タスクには、タスクナンバー (TN) と呼ばれる番号が与えられ、このナンバーにより識別されます。

Pコイルとタスクの分類

Pコイル番号	タスクナンバー	分類	起動方法	説明
P001	1	イニシャルタスク	CPUの復電またはリセット	CPUの復電時に起動され、システムのイニシャル等を行うタスクを割付けます。
P002 ~P07F	2~127	ユーザタスク	Pコイルの励磁またはマクロ命令 (Queue) の発行	ユーザにて作成した制御タスクを割付けます。
P080	128	システムタスク	システム的にOSにより起動されます。	システムタスク (デバッグタスク) を割付けます。ユーザの使用は避けて下さい。

- 2α , $2\alpha E$ ($2\alpha H$, $2\alpha Hf$) ではシーケンスプログラムより“Pコイル”を励磁する事によりタスクを起動 (Queue) する事ができます。このPコイル番号はタスクナンバーと一致します。

■ グローバルテーブル

タスク間でデータのやり取りを行うテーブル。

■ 共用サブルーチン

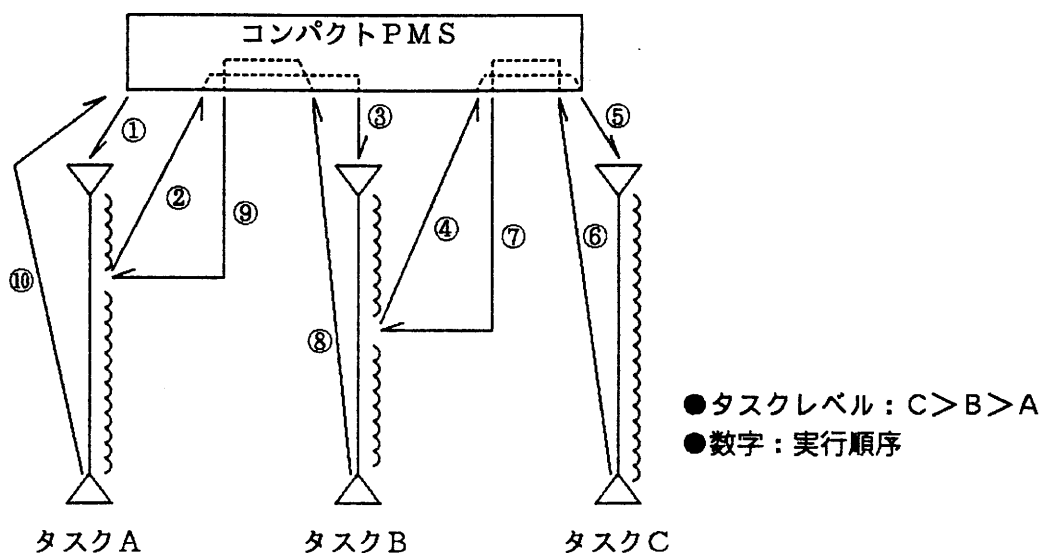
各タスクでリエントラントに使用できるサブルーチン。

2.1.1 動作

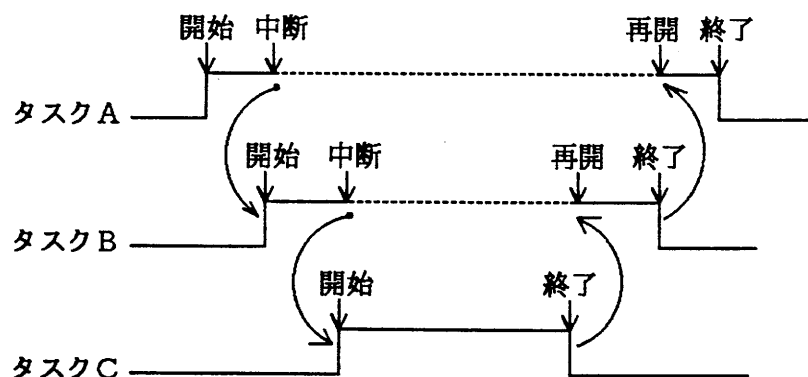
ユーザーが作成したタスクは、コンパクトPMSの下で起動、中断、終了等の管理を受けます。

また、各タスクには優先レベルがつけられ、起動がかけられればレベルの高いタスクより順次処理する事ができます。同じレベルのタスクは先に起動要求のあったタスクより処理を開始します。例えば現在実行中のタスクよりレベルの高いタスクに起動要求があった場合は、実行を一時中断し、高いレベルのタスクに実行は移ります。中断したタスクは高いレベルのプログラムが終了するまで待たされます。

■ 動作例



■ タイムチャート



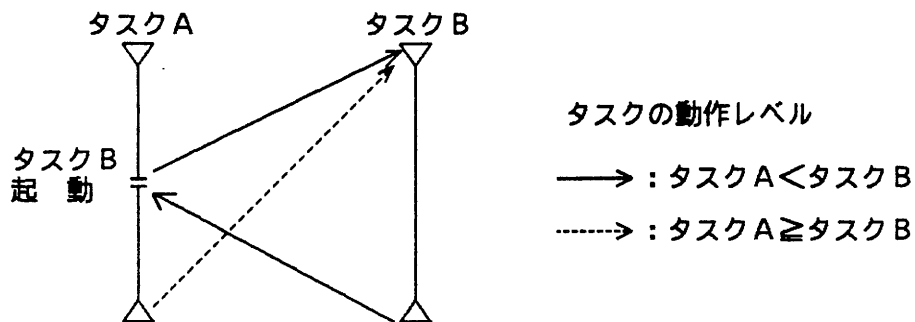
2.1.2 コンパクトPMSとタスクのリンケージ

コンパクトPMSにはタスクの起動や停止, 実行権の放棄等に関する各種の命令、つまりユーザータスクからコンパクトPMSに処理を依頼するための命令があります。これをマクロ命令といいます。

ユーザーはこのマクロ命令を使用してマルチタスクを実現します。

◆◆ マクロ命令によるタスク起動の動作例 ◆◆

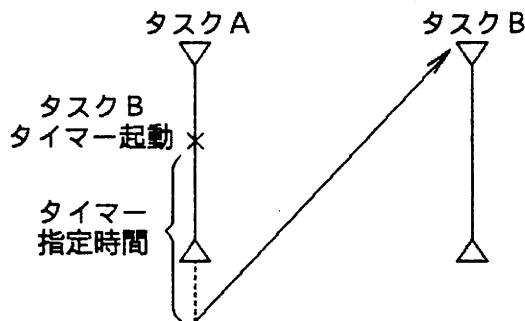
● QUEUEマクロ命令



起動要求を受けたタスクBのレベルが高い場合は起動要求時制御がタスクBへ移りますが、タスクBのレベルがタスクAと同等あるいは低い場合には、タスクAの動作終了後、タスクBに制御が移ります。

又、タスクBに制御が移る際、タスクBよりもレベルが同等以上のタスクが起動要求を受けている場合は、そのタスクの動作終了後にタスクBに制御が移ります。

● TIMERマクロ命令



タイマー指定時間後タスクBに起動要求がかかりますが、タスクBよりもレベルが同等以上のタスクが起動要求を受けている場合は、そのタスクの動作終了後にタスクBに制御が移ります。

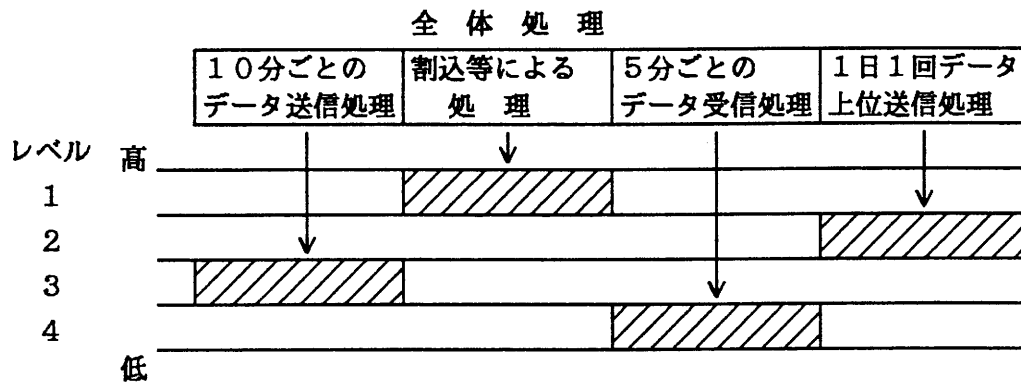
2.1.3 タスクのレベル

各タスクには優先レベルがあり、同時に処理を実行しようとした場合は、優先レベルの高いタスクより実行されます。同レベルの場合は、先に起動要求のあったタスクより実行されます。

レベルは0～4の5レベルあり、小さい方が優先順位が高くなっています。このレベルはデバッグによりタスクを登録する時に、割りつけます。

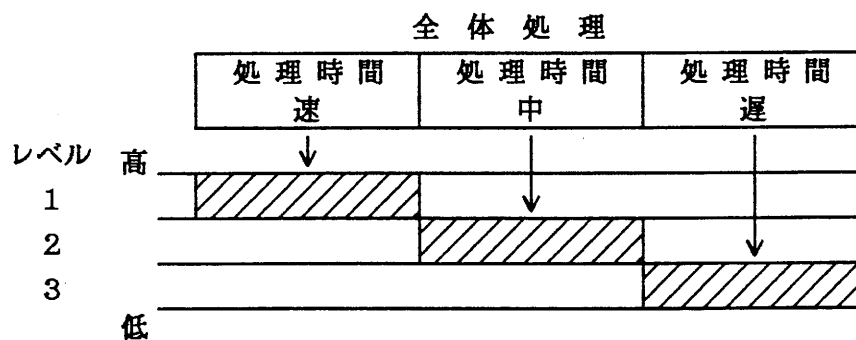
タスクの動作レベルを決定するには、そのシステム構成により変化しますが、次にその主な決め方を示します。

■ 処理内容によるレベル決定



割込を第1の優先と考え割込による処理のレベルを最高とする。
以下動作回数の少ない処理からレベルを決定する。

■ 処理時間によるレベル決定



処理時間の遅い処理のレベルを高くすると速いものの動作が遅くなると考え、処理時間の速いものからレベルを高く決定する。

2.1.4 サブルーチン

サブルーチンには、あるタスクのみが使用する I S U B と、複数のタスク間で共通に使用する R S U B とがあります。

I S U B は、それを C a l l するメインタスク（又はサブルーチン）に直接接続されます。

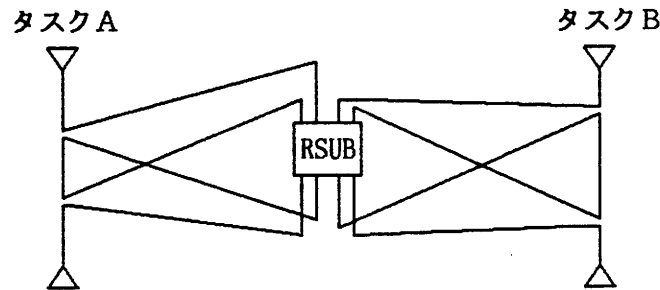
R S U B は、ある固定アドレスに格納しておき、各タスクからは、そのアドレスに J u m p する事になります。

ライブラリー等をリンクする場合は、I S U B となります。

I S U B : Internal Subroutine (内部従属サブルーチン)

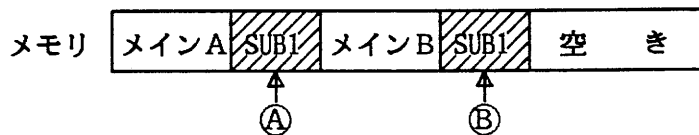


R S U B : Resident Subroutine (常駐サブルーチン)



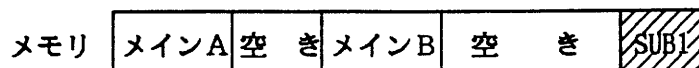
■ メモリ上の違い

◆◆ SUB 1 を I S U B とした場合 ◆◆



同一のサブルーチン (SUB 1) が、(A)(B) の 2ヶ所に格納される事になりメモリーのむだになります。

◆◆ SUB 1 を R S U B とした場合 ◆◆



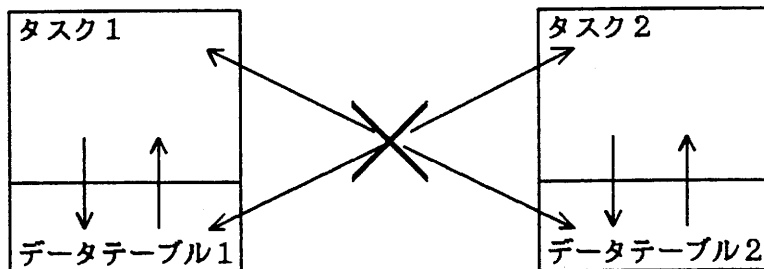
上記の様にメモリーの空きエリアに余裕が出来ます。

2.1.5 データテーブル

データテーブルにはタスク内データテーブルと、タスク間共通のグローバルテーブルの2種があります。

◆◆ タスク内データテーブル ◆◆

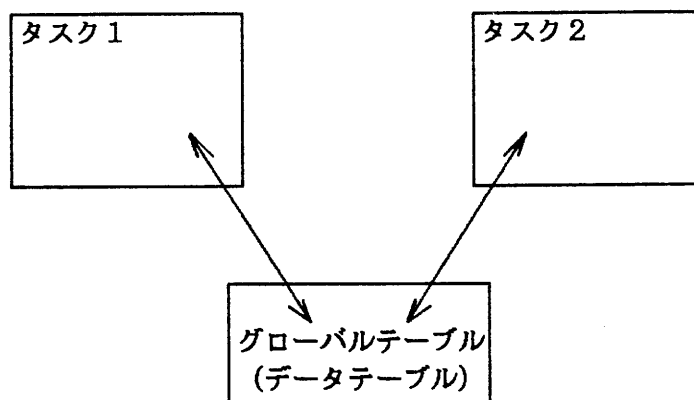
ある1つのタスク内でのみ使用されるデータテーブルで、他のタスクが使用することのないものです。



タスク1で使用できるのはデータテーブル1のみでデータテーブル2に対する読み込み、書き込みなどいっさいできません。タスク2に関して同様。

◆◆ グローバルテーブル ◆◆

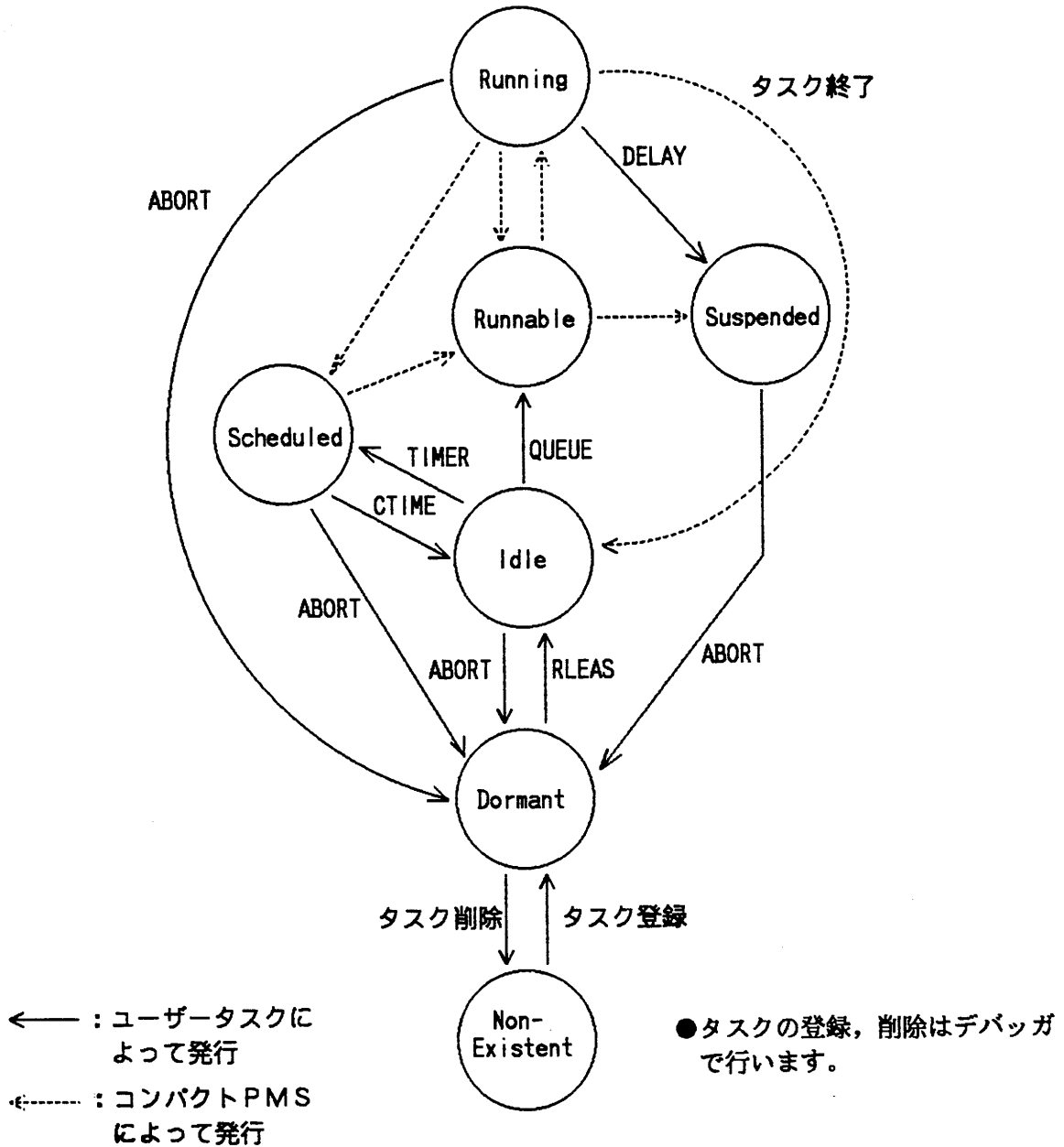
複数のタスク間で共通に使用されるデータテーブルで、複数のタスクによりデータの読み込み、書き込みが行われ、タスク間のデータの受渡しを可能とするものです。



タスク1, タスク2ともに、グローバルテーブル (データテーブル) に対する読み込み, 書き込み可能。グローバルテーブルを使ってタスク1, タスク2のデータの受渡しができます。

2.2 タスクの状態

タスクには実行中であったり中断していたりする状態があります。



- Non-Existent (ノンイグジステント)
タスクがコンパクトPMSに登録されていない状態。
- Dormant (ドーマント)
起動が禁止されている状態。
- Idle (アイドル)
起動要求を受けつけられる状態。
- Runnable (ランナブル)
起動要求は受けつけられ、実行順番待ちの状態。
- Running (ランニング)
実行中の状態。
- Scheduled (スケジュールド)
起動要求を受けつけられる状態で、かつ、一定時間後に自動的にRunnableとなる事が予定されている状態。
- Suspended (サスペンディッド)
事象 (イベント) 発生待ちで実行を中断している状態。

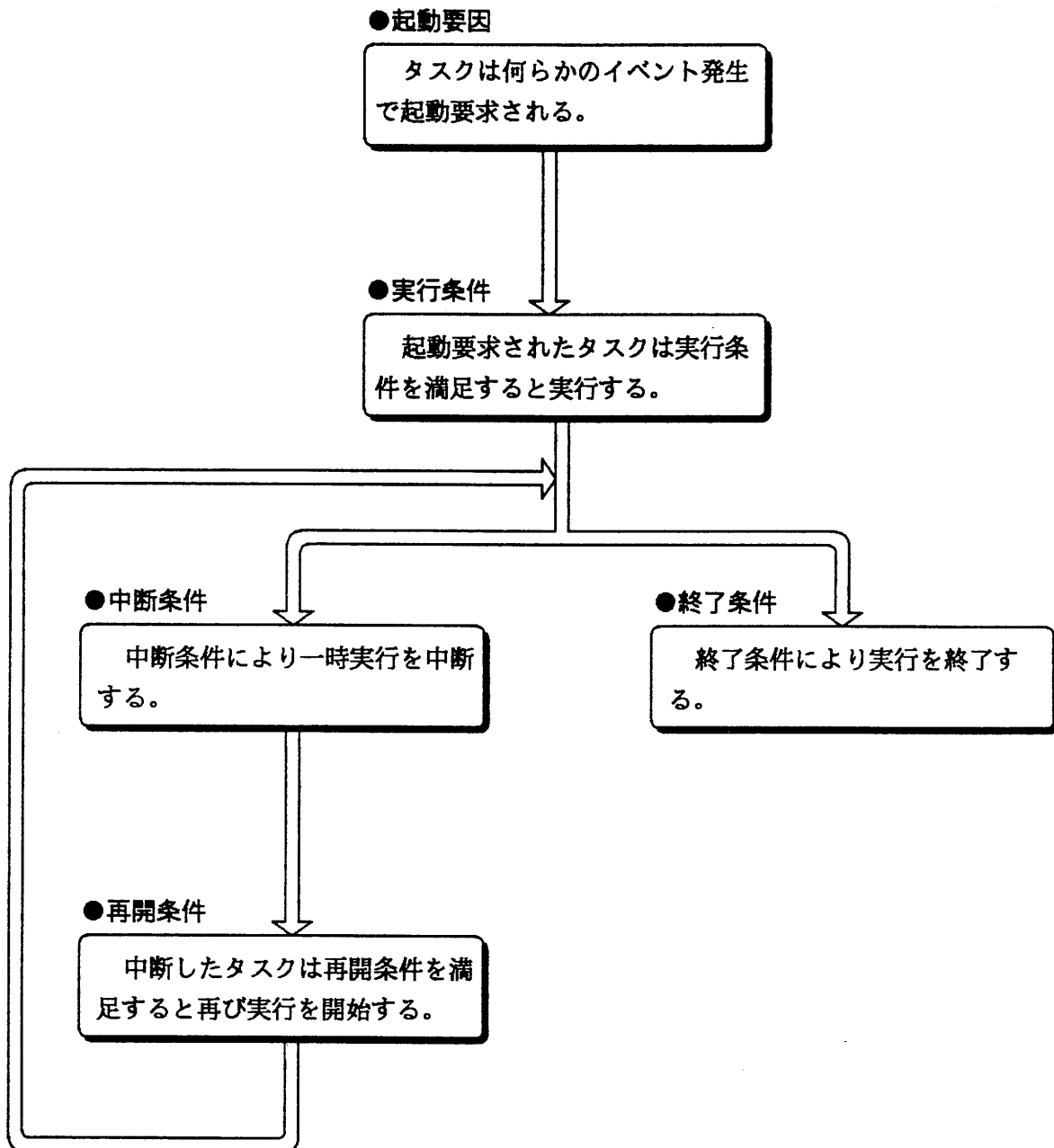
◆◆ タスクの状態とマクロ命令 ◆◆

タスクの状態を変えるためにマクロ命令が用いられます。そのマクロ命令の発行前と発行後の関係は次のようになります。

マクロ名	マクロ命令発行タスク		マクロ命令対象タスク	
	前	後	前	後
RLEAS	Running	Running	Dormant	Idle
QUEUE	Running	Running	Idle	Runnable
ABORT	Running	Running	任意	Dormant
TIMER	Running	Running	Idle	Scheduled
CTIME	Running	Running	Scheduled	Idle
DELEY	Running	Suspended	——	——
CHAP	Running	Running	Running	Runnable

2.3 タスクの動作

タスクは起動要求されてから終了するまで次のような流れになっています。



◆◆ 起動要因 ◆◆

イ ベ ント	説 明
QUEUEマクロ発行	QUEUEマクロのパラメータのタスクが起動要求されます。

◆◆ 実行条件 ◆◆

条 件	説 明
自タスクより高いレベルのタスク、かつ、同レベルで先に起動されているタスクが存在しない。 かつ、自タスクがメモリに存在してコンパクトPMSに登録。	待行列ブロックで自タスクが最優先になって、かつ、自タスクが生成されている場合を意味します。

◆◆ 中断条件 ◆◆

条 件	説 明
タスクの実行に必要な資源が使用できない	タスクの実行に必要な資源（CPU等）が他タスクにより使用されている時は、実行は中断され待ち状態となります。
優先レベルの高いタスクが動作可能の時	割込みにより、より優先レベルの高いタスクが起動されて、そのタスクの動作可能なときはそのタスクに制御が移ります。
自ら実行を中断したとき	同期をとるため等の目的で自ら実行を中断すると他タスクに制御が移ります。

◆◆ 終了条件 ◆◆

条 件	説 明
タスクが実行終了したとき	実行待行列からはずされます。
ABORTマクロ命令の対象となったとき	ABORTマクロ命令を発行すると、処理が打ち切られます。
処理継続不可能時	コンパクトPMSが自動的に異常を生じたタスクのABORT処理を行います。

◆◆ 再開条件 ◆◆

条 件	説 明
タスクの実行に必要な資源が使用可能になった	資源（CPU等）が空きとなり、使用可能となった。
発生待の事象が発生した	自らの中断（DELAY）要因が除かれる事象が発生した。
優先レベルの高いタスク総てが中断、あるいは終了した時	自分より高いレベル、あるいは同レベルで先に起動がかかっていたタスクが動作可能でいるかぎりサービスは行なわれません。

3 システム管理

3.1 システム立上げ

コンパクトPMSは、電源を投入することによりイニシャルタスクを起動します。

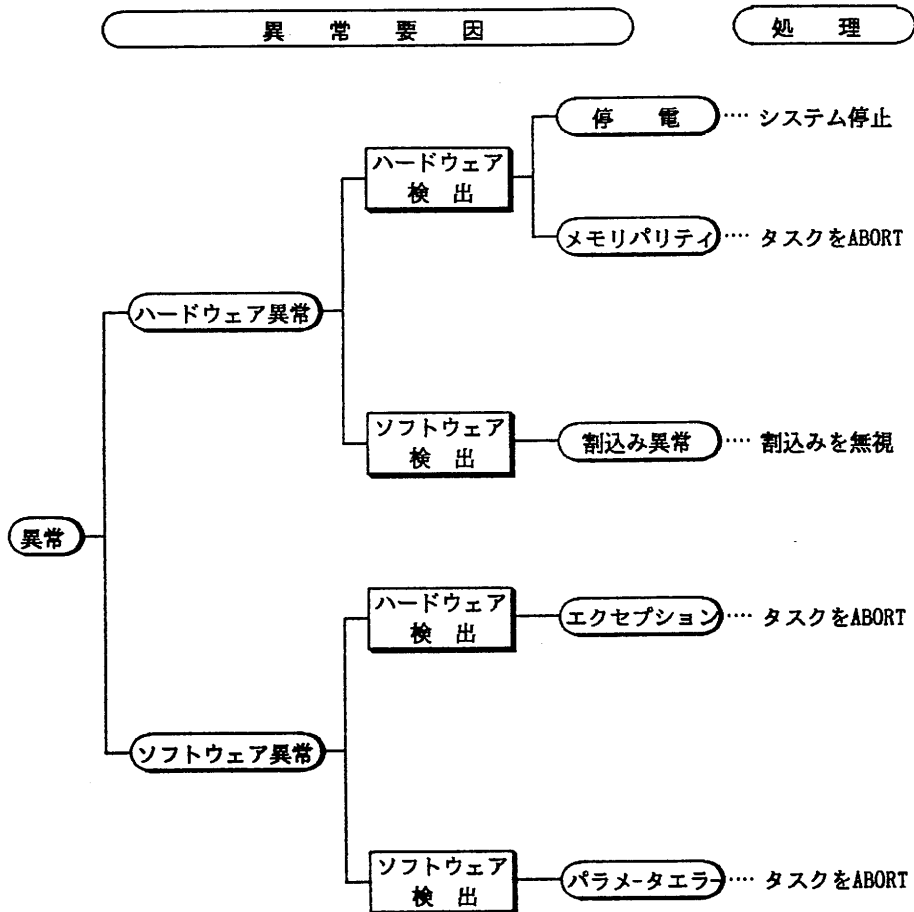
イニシャルタスクは、最初に起動されるユーザタスクですが、ユーザタスクは立上げ時には、このイニシャルタスク以外は全てDormant状態となっています。従って、イニシャルタスクでは、これから行う業務に必要なユーザタスクに対してRLEASマクロ命令を発行して、タスクが起動要求受け付け可能な状態とします。イニシャルタスクのタスク番号は“1”です。

3.2 異常処理

2αシリーズのOSは、システムの信頼性を高めるためにRAS機能を強化し、ハードウェア、ソフトウェアの各レベルでの異常チェックを行い、更にソフトウェアの一部はリトライを行っています。この概略を以下に示します。

ハードウェアレベルで、異常が発生した場合、OSレベルでエラー回復可能か否かチェックされます。OSレベルでリトライ可能と判断されるとOSは、リトライ処理を行います。OSレベルでリトライ不可な異常であると判断されると異常データの収集、ユーザの組み込みサブルーチンへのリンク、処理依頼元タスクに対する処理（タスクの打切り）などを行います。

◆◆ 異常要因と処理の概略図 ◆◆



3.3 組み込みサブルーチン

システム内部でイベント（主に異常）が発生したときOSはエラー処理を行います。これにユーザ処理を組み込むことができます。これは、組み込みサブルーチンを登録することによってできます。

組み込むサブルーチンは、1つのイベントに1つ組み込むことができます。

3.3.1 種類 表

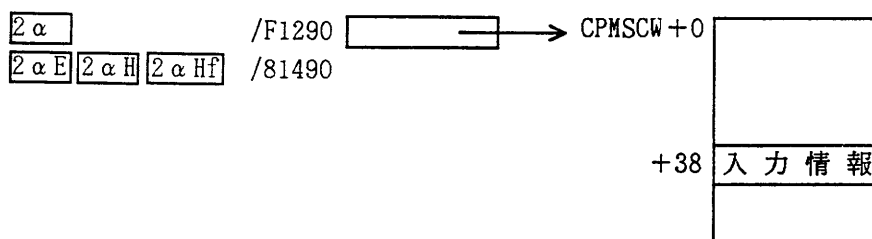
ユーザが組みめるプログラムのサブルーチンの種類を次に示します。

組み込みサブルーチン内でのレジスタ退避回復は、プログラム完成後不要となるリンケージですので、C言語で作成する事が可能です。

サブルーチン総称		動作 レベル	対応イベント	入力情報	発行可能な マクロ命令	備 考
名 称	略 称					
System Down Subroutine	SDS	6	OS実行中にハードエラー又はプログラムエラー発生	エラーフリーズエリア(ERSTK)アドレス	なし	
CPU Error Subroutine	CPES	6	タスク実行中にハードエラー又はプログラムエラー発生	同 上	queue rleas	
Exit Subroutine	EXS	5	タスク終了	終了したタスクのTN	queue rleas	資源開放後
Abort Subroutine	ABS	5	ABORTマクロ命令発行	ABORTされるタスクのTN	同 上	同 上
Parameter Check Subroutine	PCKS	5	マクロパラメータエラー発生	エラーフリーズエリア(SVCEB)アドレス	同 上	

- CPMSCWは、2αの場合は/F1290番地、2αE、2αH、2αHfの場合は/81490番地にロングワードで格納されているアドレスです。

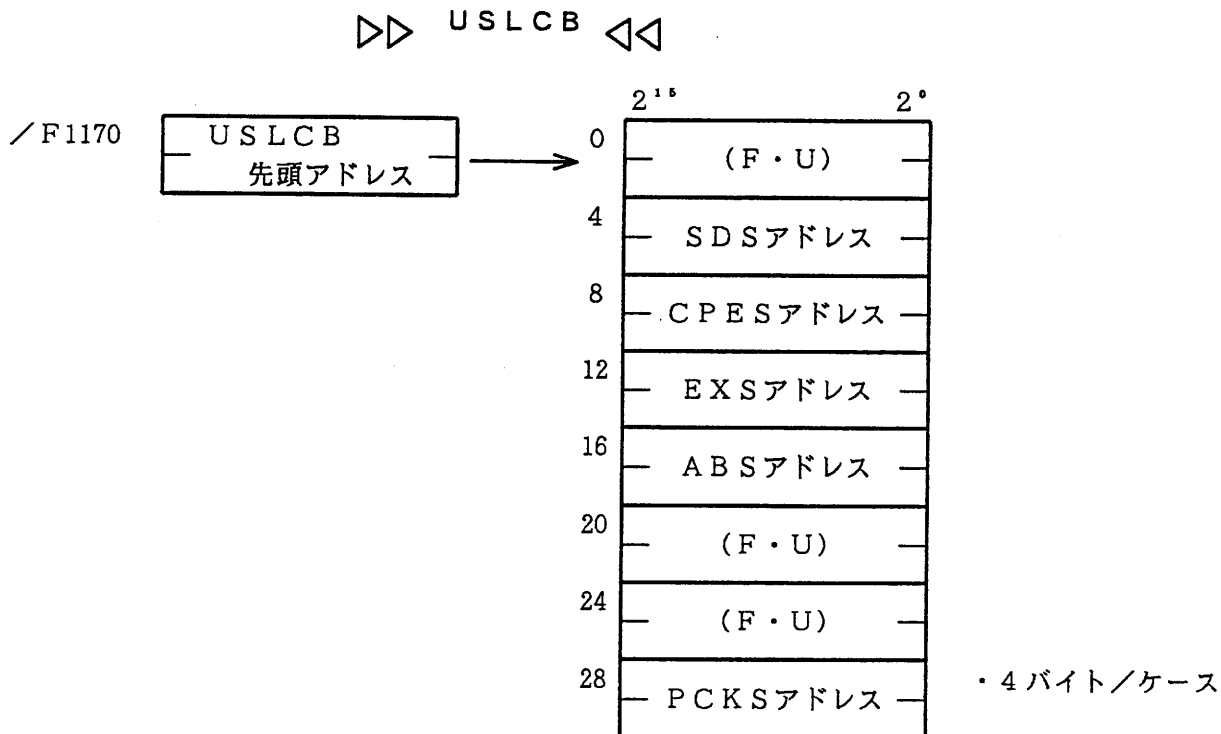
入力情報は、CPMSCWの38番目にロングワードで格納されています。



3.3.2 登録

登録は次のUSLCB : User Subroutine Link Control Block の該当するケースにアドレスを書込む事により行います。

USLCBの先頭アドレスは、/F1170番地 (SEQCB : Sequence Control Block) にLong Wordにて格納されています。内容を調べて誤りのないように登録してください。



3.3.3 入力情報

組込みサブルーチンで入力できるエラー情報には、次の2つのテーブルがあります。

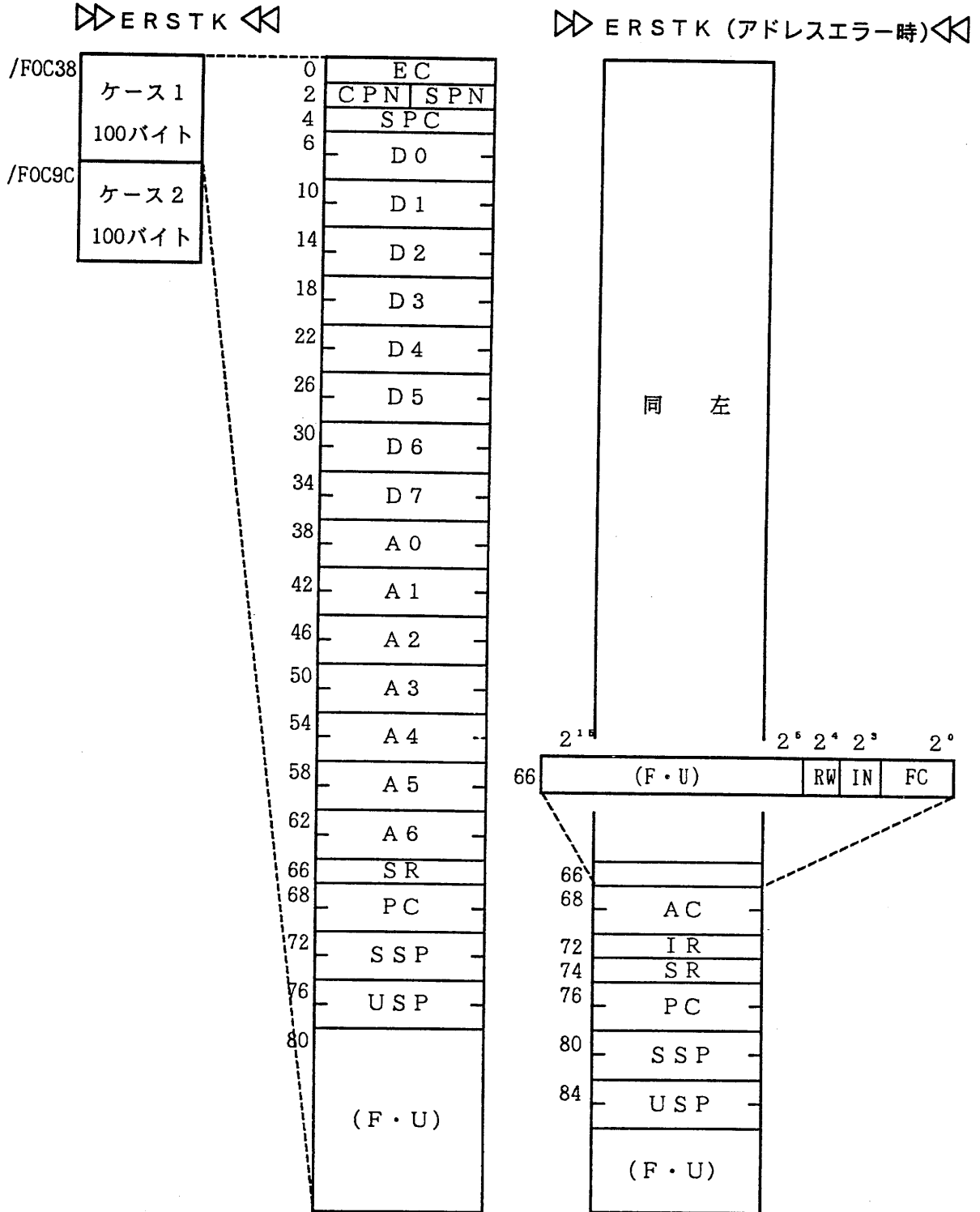
- ERSTK : Error Stack
システムダウンおよび、CPUエラーが発生した時編集されます。
- SVCEB : Supervisory Call Error Block
マクロパラメータエラー発生時に編集されます。

2 α

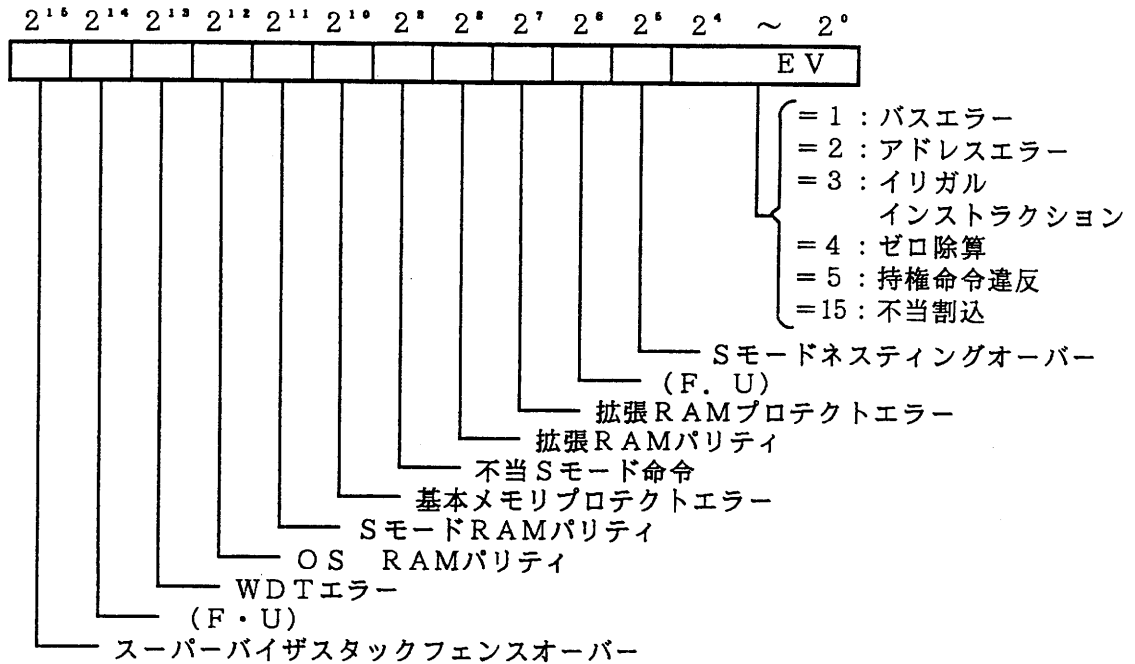
3.4 エラーログ

●● ERSTKテーブル構成 ●●

このテーブルはエラー発生が連続した場合は2ケース分格納されます。
 また、テーブル中ECの内容がアドレスエラーの場合格納される内容が変わります。



EC : エラーコード



- CPN : タスク (Pコイル) 番号
- SPN : SモードプログラムNo.
- SPC : Sモードプログラムカウンター (不当Sモード命令またはSモードRAMパリティ時のみ有効)
- DO~7 : エラー発生時のデータレジスタ内容
- AO~6 : エラー発生時のアドレスレジスタ内容
- SR : エラー発生時のステータスレジスタ内容
- PC : プログラムカウンター (MPU)
- SSP : システムスタックポインタ
- USP : ユーザスタックポインタ
- RW : Read系 (= 1), Write系 (= 0)
- IN : インストラクション (= 0), 以外 (= 1)
- FC : ファンクションコード
 - = 1 ... ユーザデータ
 - = 2 ... ユーザプログラム
 - = 5 ... スーパーバイザデータ
 - = 6 ... スーパーバイザプログラム
- AC : アクセスアドレス
- IR : インストラクションレジスタ (エラー時の命令)
- F・U : 将来拡張用

2 α E
 2 α H
 2 α H f

●● ERSTKテーブル構成 ●●

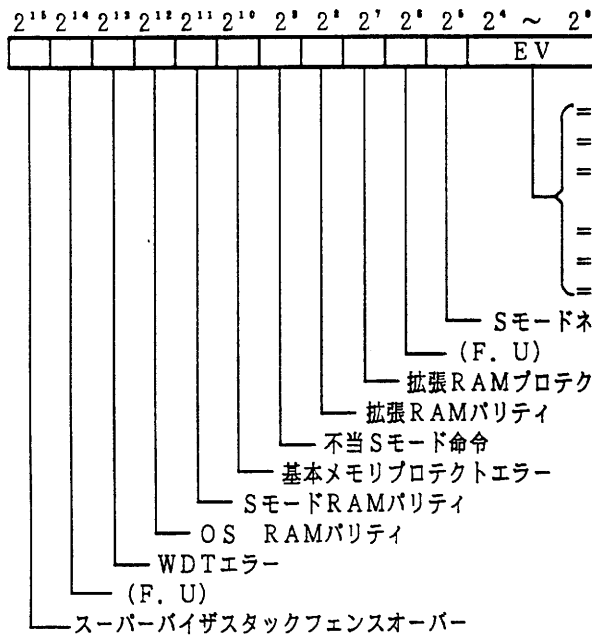
このテーブルはエラー発生が連続した場合は2ケース格納されます。

また、テーブル中「拡張情報」は“スタックフレームフォーマット”の内容により格納される情報が変わります。

/80000	ケース 1	0	CASEP	64	DO	128	拡張 情 報
	256バイト	2	TYPE	68	D 1		
/80100		ケース 2	4	F.U	72	D 2	
	6		EC	76	D 3		
8	CPN	80	D 4				
10	SPN	84	D 5				
12	SPC	88	D 6				
14	MS	92	D 7				
16	SEC	96	A 0				
20	YEAR	100	A 1				
22	MONTH DAY	104	A 2				
24	SECCNT	108	A 3				
28	SYSCNT	112	A 4				
32	SVO	116	A 5				
34	SR	120	A 6				
36	PC	124	U S P				
40	MSP						
44	ISP						
48	SFC						
50	DFC						
52	VBR						
56	CACR						
60	CAAD						

2 α E
2 α H
2 α H f

EC : エラーコード



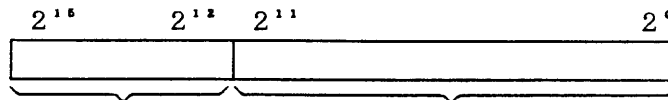
- 22=比較不能状態での分岐/セット (FPU)
- 23=不正確な結果 (FPU)
- 24=ゼロ除算 (FPU)
- 25=アンダーフロー (FPU)
- 26=オペランドエラー (FPU)
- 27=オーバーフロー (FPU)
- 28=シグナリング非数 (FPU)

※ 22~28は、2 α H fのみ

- CASEP : ケースポイント
- TYPE : RAM-OS時= /6820, ROM-OS時= /6821
- EC : エラーコード
- CPN : タスク (Pコイル)No.
- SPN : SモードプログラムNo.
- SPC : Sモードプログラムカウンター
(不当Sモード命令または、SモードRAMパリティ時のみ有効)
- MS : ミリ秒
- SEC : 秒
- YEAR : 年
- MONTH : 月
- DAY : 日
- SECCNT : 秒カウンター
- SYSCNT : システムカウンター
- SVO : スタックフレームフォーマット+ベクタオフセット
- SR : ステータスレジスタ内容
- PC : プログラムカウンター (MPU)
- MSP : マスタスタックポインタ
- ISP : 割込みスタックポインタ
- SFC : ソースファンクションコード
- DFC : ディスティネーションファンクションコード
- VBR : ベクタベースレジスタ
- CACR : キャッシュコントロールレジスタ
- CAAR : キャッシュアドレスレジスタ
- D0~D7 : データレジスタ内容
- A0~A6 : アドレスレジスタ内容
- USP : ユーザスタックポインタ

2 α E
2 α H
2 α Hf

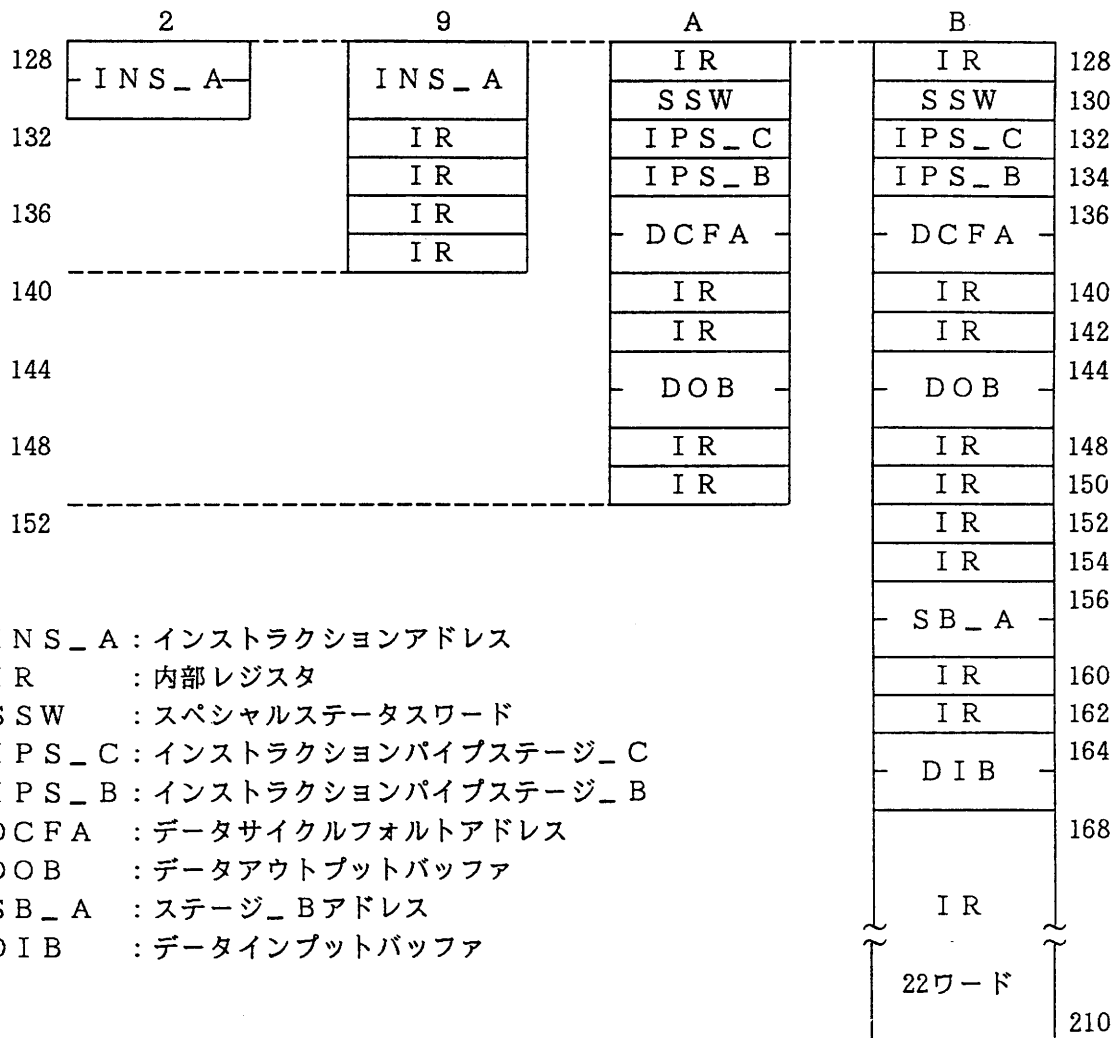
SVO : スタックフレームフォーマット+ベクタオフセット



スタックフレームフォーマット ベクタオフセット

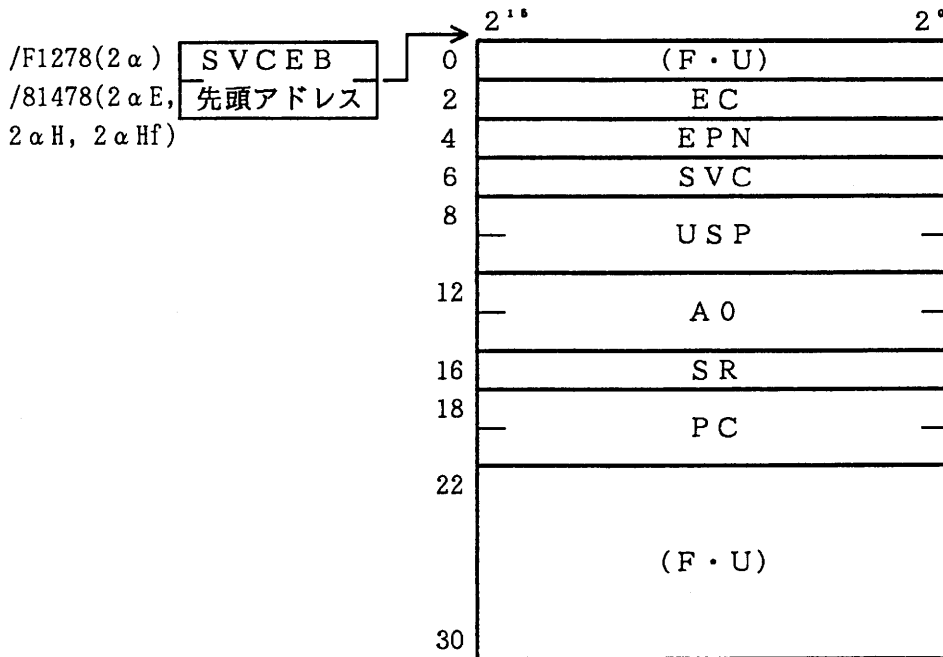
スタックフレームフォーマットと拡張情報

スタックフレームフォーマットが、2, 9, A, B以外の場合は拡張情報は、ありません。

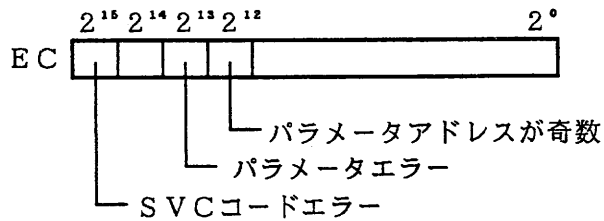


- INS_A : インストラクションアドレス
- IR : 内部レジスタ
- SSW : スペシャルステータスワード
- IPS_C : インストラクションパイプステージ_C
- IPS_B : インストラクションパイプステージ_B
- DCFA : データサイクルフォルトアドレス
- DOB : データアウトプットバッファ
- SB_A : ステージ_Bアドレス
- DIB : データインプットバッファ

●● SVCEBテーブル構成 ●●



EC : エラーコード

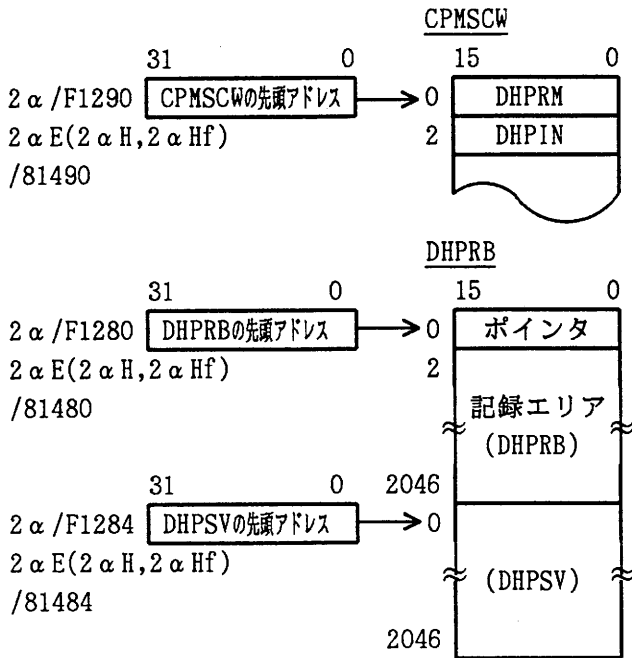


- EPN : エラープログラムNo.
- SVC : SVCマクロNo.
- USP : エラー発生時のUSP
- A0 : エラー発生時のA0レジスタ
- SR : マクロ発行時のSR
- PC : マクロ命令の戻りアドレス
- F·U : 将来拡張用

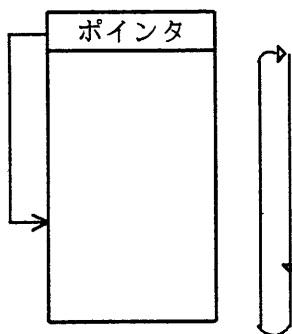
3.5 DHP

DHP (Debugging HelPer) は、OSの遷移状態をトレースする機能です。この機能により不具合発生前のOSの状態等を調べることができます。

DHP関連テーブルを以下に示します。



- DHPRM (DHP記録モード)
 - 2 : 全記録モード
 - 1 : CPMSマクロ記録モード
 - 他 : DHPは記録されません
- DHPIN (DHPイニシャルモード)
 - 1 : リセット (GR時) にDHPRB及びDHPSVをゼロクリアします。
 - 他 : ゼロクリアしません。
- DHPRB (DHP記録エリア)
 - 2048バイト容量のDHPのトレースバッファです。先頭1ワードは次記録ポインタです。
- DHPSV (DHP退避エリア)
 - DHPRBの内容を退避するエリアです。この退避処理は、
 - 1 : CPUダウン時
 - 2 : CPUエラー発生時に退避されます。



DHPは順次記録されます。

3.6 浮動小数点演算機能

浮動小数点演算機能は、2αシリーズのH-S10/2αHfで使用できます。

浮動小数点は、IEEE (The Institute of Electrical and Electronics Engineers) 型式です。

浮動小数点レジスタ、浮動小数点の関数などの詳細は、CP/M-68K V1.2オペレーティングシステムガイド、コンパイラのマニュアル等を参照してください。

ここでは、簡単に浮動小数点レジスタ、エラー発生時の浮動小数点演算用エラーログについて紹介します。

<浮動小数点レジスタ>

79	63	0		
				FP0
				FP1
				FP2
				FP3
				FP4
				FP5
				FP6
				FP7

Floating Point
Data Registers

31	23	15	7	0		
0		Exception Enable	Mode Control		FP	Control Register
Condition Code	Quotient	Exception Status	Accrued Exception		FP	Status Register
					FP	Instruction Address Register

3.6.1 浮動小数点演算使用時の注意

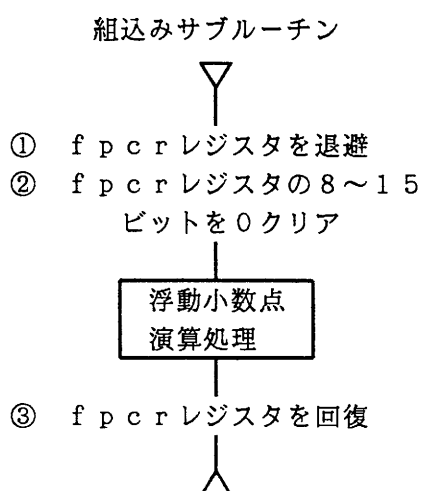
組込みサブルーチン内で浮動小数点演算処理を行う場合は、システムエラーを防止するために例外トラップの発生を防止します。組込みサブルーチンの先頭でfpcp（浮動小数点演算コプロセッサ）のc r（コントロールレジスタ）の8～15ビットを0に書換えることで、例外トラップの発生を防止できます。そのため、組込みサブルーチン内の浮動小数点演算処理の前後には、以下に示すようにfpcpのc r（fpcrと称する）の退避・書換え・回復処理が必要です。

また、アセンブラでのfpcpのc rの退避・書換え・回復処理例を次ページに示します。

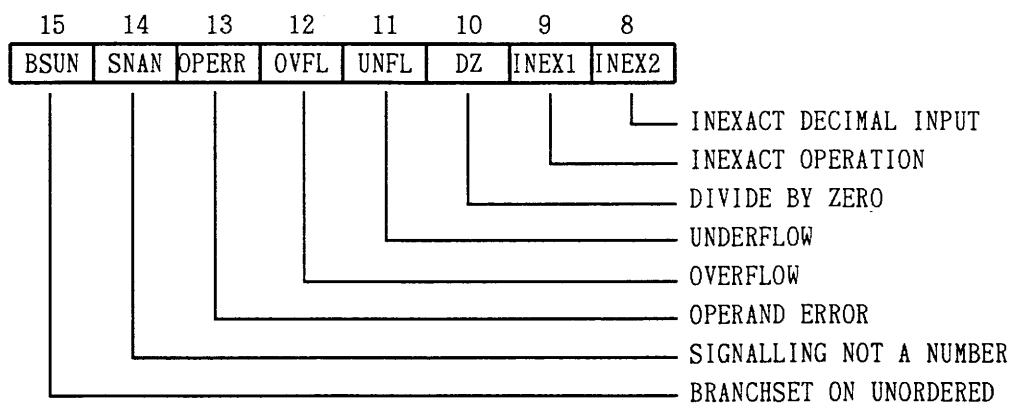
C言語でのfpcpのc rを書換えるライブラリ関数は用意されていません。

ライブラリ関数の詳細については、「CRM-68K V1.2 オペレーティングシステムガイド」を参照してください。

<組込みサブルーチン内におけるfpcpのc rに関する処理とc rの構成>



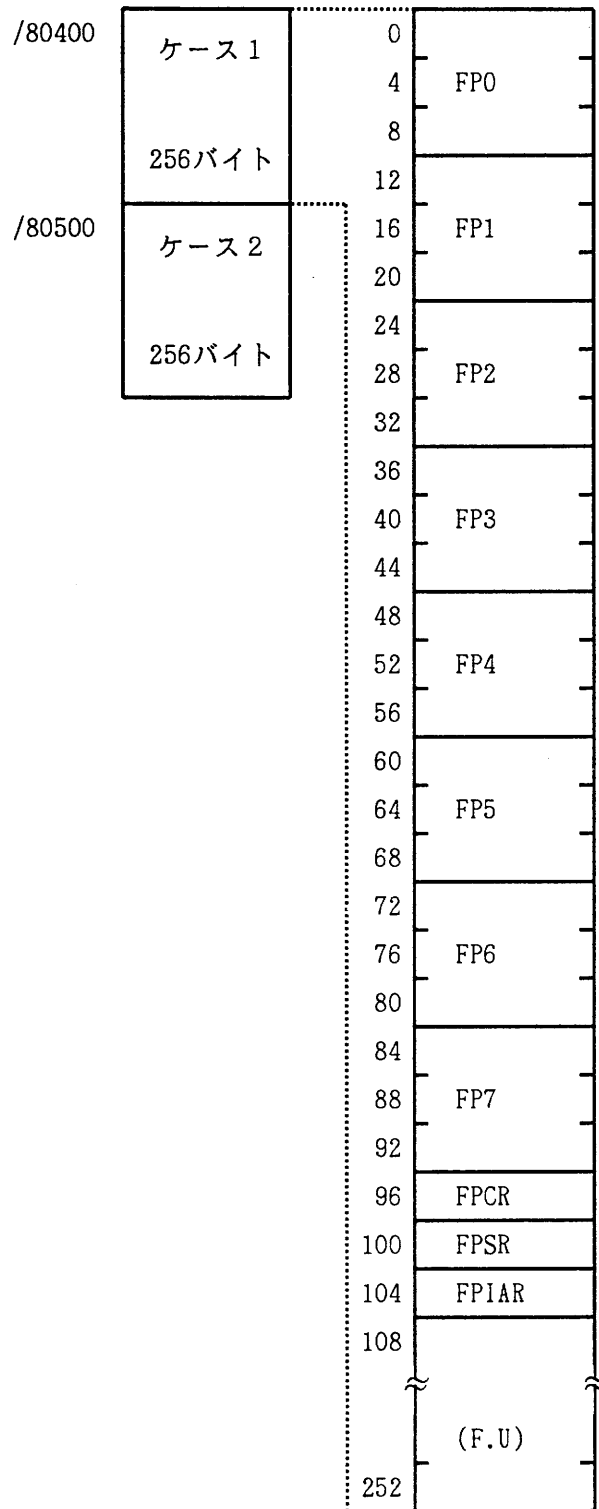
< f p c p の c r の構成 >



<fpcrのc rの退避・書換え・回復処理>

	コーディング
	⋮
	fmove.l fpcr,(退避エリア) … ①
ア	fand.l 0xff,fpcr …………… ②
セ	⋮
ン	(浮動小数点演算処理)
ブ	⋮
ラ	fmove.l (退避エリア),fpcr …③
	⋮

3.6.2 浮動小数点用エラーログ



4 マクロ仕様

4.1 マクロ命令

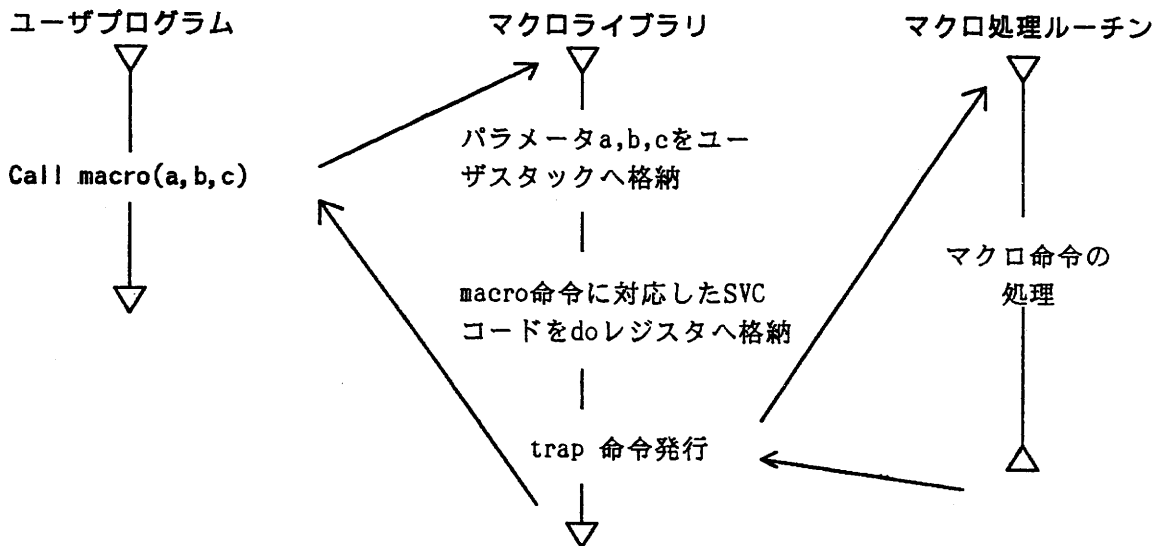
マクロ命令は、ユーザプログラム（タスク）からコンパクトPMSに処理を依頼するための命令であり、ユーザプログラムでは `call` 文等で記述しておけば、マクロライブラリによりOS呼出し命令である `trap` 命令に展開されます。

プログラムが実行されたとき、この `trap` 命令によりOSへリンクし、処理が行われます。

`trap` 命令とは、ソフト的に割込みを発生させ、ユーザモードからスーパーバイザモードに変更します。ユーザモードとは、ユーザプログラムが実行するモードであり、スーパーバイザモードとは、OS処理等のシステムタスク処理モードです。

4.1.1 マクロライブラリ

マクロライブラリは、高級言語で記述されたマクロ命令を `trap` 命令に展開するためのサブルーチン群です。マクロライブラリは呼出されると、パラメータ（引数）を各マクロ命令ごとに定められた順にユーザスタックエリアに格納し、`trap` 命令を発行します。C言語でユーザプログラムを記述する場合は、マクロ命令レベルの記述で済み `trap` 命令を意識して記述する必要はありません。アセンブラ言語でユーザプログラムを記述する場合は、`trap` 命令を意識して記述してください。



マクロライブラリは、開発環境により提供方法が異なりますが、同じ機能です。

- PSEαで開発：コンパクトPMSのシステムF/DでCPMSMLIBファイルとして提供します。
- パソコンで開発：C言語プログラム開発環境システム（RPDP/S10）のシステムF/Dで、CPMS.LIBファイルとして提供します。

4.1.2 一般夫見貝リ

◆◆パラメータの渡し方◆◆

マクロライブラリを使用する場合、パラメータはすべてアドレス渡しとなります。

■ C言語で作成する場合

```
long tn ;
  ⋮
tn=100 ;
  ⋮
macro(&tn) ;
  ⋮
```

左のようにtn(=100)が格納されたアドレスを引数に記述します。
& tn はtnへのポインタ, tn が格納されているアドレスを示します。
これをmacro(tn)と書いてはなりません。

C言語の場合は、この他にもいろいろな記述方法があります。要は、パラメータのアドレスがマクロライブラリに渡るように記述すればよく、以下の記述は同等ですのでコーディングしやすい方法でプログラムしてください。第2章のマクロ命令の記述方法は、一例を述べたものであり、同等のことが行えれば別の記述方法を用いてもかまいません。

● パラメータが配列全体であるとき

```
  ⋮
long X[n] ;
  ⋮
macro(X) ;
  ⋮
```

● パラメータが配列の要素の1つであるとき(下の3つは同等です)

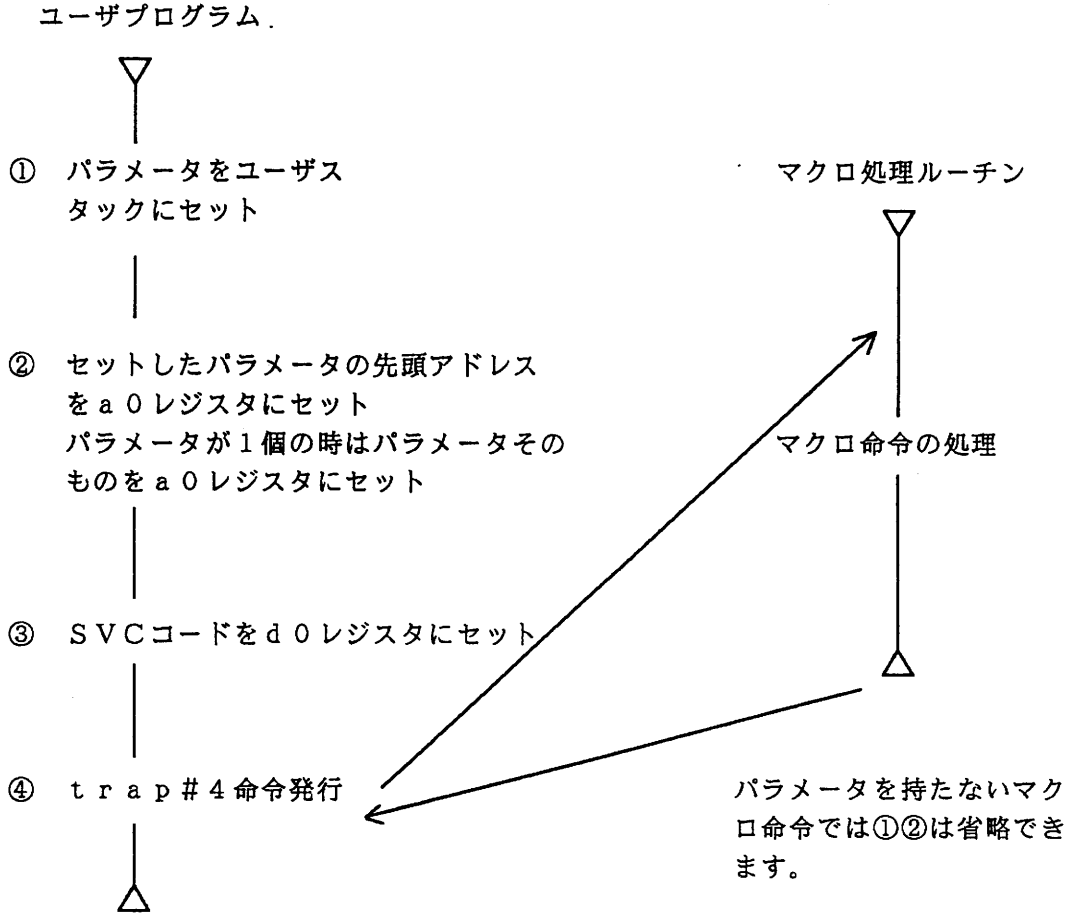
<pre> ⋮ long X[n] ; ⋮ X[i] =100 ; ⋮ macro(&X[i]); ⋮</pre>	<pre> ⋮ long X[n] ; ⋮ X[i] =100 ; ⋮ macro(X+i) ; ⋮</pre>	<pre> ⋮ long *X[n] ; ⋮ * X[i] =100 ; ⋮ macro(X [i]) ; ⋮</pre>
--	---	---

● パラメータが単純変数のとき(下の2つは同等です)

<pre> ⋮ long X ; ⋮ X=100 ; ⋮ macro(&X) ; ⋮</pre>	<pre> ⋮ long *X ; ⋮ *X=100 ; ⋮ macro(X) ; ⋮</pre>
---	--

■ アセンブラで作成する場合

ユーザスタックにパラメータを格納したあと trap # 4 命令を発行します。



◆◆リターンコード◆◆

マクロ命令の実行結果は d 0 レジスタに格納され、マクロ処理ルーチンから戻されます。マクロライブラリを使用している場合は、マクロ命令を関数として使用した場合、その値がリターンコードとなります。

```

long macro ;
long *X ;
...
if (macro(X)) {.....}
...
  
```

マクロライブラリの値はマクロ命令処理結果のリターンコードが返されますので左のようにリターンコードを判定できます。

マクロ命令の処理が正常に行われるとリターンコードは 0 が返されますが、マクロ命令によってはリターンコードが 0 以外であっても正常に処理が行われているものもあります。

◆◆マクロ命令のパラメータチェック◆◆

マクロ命令はユーザプログラムとコンパクトPMSの直接のデータのやりとりですから、パラメータを誤るとシステムの誤動作、ダウンとなる場合があります。コンパクトPMSでは特定のマクロ命令について、パラメータのチェックを行っています。もしパラメータの異常を検出したときは、そのマクロ命令を発行したタスクの処理を打ち切ります。

4.1.3 パラメータチェック

コンパクトPMSのマクロ命令は、そのパラメータの主要なものは、ソフトウェア的にそのパラメータの合理的をチェックします。チェックにより不合理とみなされると、マクロパラメータエラーとして報告され、マクロ命令発行タスクはABORTされます。

4.2 種類

種類	名称	内容
タスク制御 管理マクロ	RLEAS	パラメータ指定のタスクがDormantならばIdleにします。 Dormant以外は何もしません。
	QUEUE	パラメータ指定のタスクがDormantでも未登録でもないとき、 実行待ち行列につながります。
	ABORT	パラメータ指定のタスクを強制終了させDormantにします。
タイマ 管理マクロ	DELAY	本命令発行タスク（自タスク）をパラメータ指定時間だけ中断しま す。
	TIMER	パラメータ指定のタスクを指定時間経過後、起動要求し、その後、 指定周期ごとに起動要求します。
	CTIME	TIMERマクロ命令を取り消します。
	STIME	絶対時刻の設定又は更新を行います。 * 1
	GTIME	現在の時刻を取出します。 * 1
	WAKE	指定タスクを時刻管理テーブルに登録しscheduled状態と します。 * 1
	CWAKE	WAKEマクロ命令による登録を取消します。 * 1
タスク状態 管理マクロ	CHAP	パラメータ指定のタスクの優先レベルを変更します。
	CHMOD	ステータスレジスタのレベルを変更し、割込禁止状態とします。
ファクタ 管理マクロ	SFACT	パラメータ指定のタスクに起動要因（0～16）をセットします。
	GFACT	本マクロ命令発行タスク（自タスク）に対してセットされている起 動要因を取り出します。
	USPCHK	本マクロ命令発行タスク（自タスク）がパラメータで指定されたバ イト数を越えて、スタックを使用しているかをチェックします。
資源管理 マクロ	RSERV	タスク間の共有エリアを占有します。
	FREE	RSERVマクロ命令により占有したタスク間 共有エリアを占有解除します。
情報関連 マクロ	MVMEM	プロテクトエリアヘデータを転送します。

* 1 : 2 α , 2 α Eの場合、オプションの時計付メモリが必要です。

4.2.1 r l e a s (r e l e a s e)

機能	パラメータ tn で指定される被要求タスクが Dormant 状態 (休止中) か否かチェックします。もし Dormant 状態ならば Idle 状態 (実行可能) とします。Dormant 状態でないときは何もしません。													
発行手順	アセンブラ	C 言語												
	<pre> ⋮ MOVE.L #2, D0 MOVE.L TN, A0 TRAP #4 ⋮ </pre>	<pre> ⋮ long tn ; ⋮ rleas (&tn) ; ⋮ </pre>												
パラメータ	tn : 倍精度整数型変数又は定数 Idle 状態にしたい被要求タスクのタスク番号であり、次の形にしてください。													
	<div style="text-align: center;"> <table style="border: 1px solid black; margin: auto;"> <tr> <td style="text-align: center;">31</td> <td style="text-align: center;">8</td> <td style="text-align: center;">7</td> <td style="text-align: center;">0</td> </tr> <tr> <td style="text-align: center;">0</td> <td colspan="2" style="text-align: center;">←————→</td> <td style="text-align: center;">0</td> </tr> <tr> <td colspan="4" style="text-align: center;">タスク番号</td> </tr> </table> </div>		31	8	7	0	0	←————→		0	タスク番号			
31	8	7	0											
0	←————→		0											
タスク番号														
リターンコード	0 : 正常終了 常に正常としてリターンコードを返します。													
パラメータチェック	0 < tn ≤ 128 でないときはパラメータエラーとして、(OSは) 要求タスクの処理を打ち切り、Dormant 状態にします。													

- ※ 本マクロ命令仕様の説明には、二種類のタスクがあります。
要求タスク (calling task) と被要求タスク (called task) です。
要求タスクはマクロ命令を使って、OS(compact PMS) に Tn で指名したタスク (被要求タスク) の状態変更処理を要求します。マクロ命令パラメータで Tn を指定しない場合は、要求タスク自身 (要求タスク = 被要求タスク) についての処理を行ないません。

4.2.2 queue (queue)

機能	<p>(OSは) Idle 状態にある被要求タスクを実行待ち状態とします。即ち、被要求タスクが Dormant 状態でも、未登録でもないとき、OSはそのタスクを実行待ち行列につながります。</p> <p>実行待ち行列につながれたタスクは、そのレベル待ち行列の中では、先着順 (First-In, First-out) で実行されます。待ち行列は実行レベルごとであり、各レベルにはそれぞれ優先順位 (プライオリティ) がついています。</p> <p>したがって、被要求タスクのレベルが要求タスクのレベルよりも高い場合、プログラム制御はOSから被要求タスクに移ります。被要求タスクのレベルが要求タスクと同じか、低いと、OSはqueueマクロの処理後、制御を要求タスクに戻します。</p> <p>OSが被要求タスクを実行待ち行列につながるとき、パラメータで指定したfactも起動要因テーブルに格納します。OSにより起動されたタスクはGFACTマクロ命令により起動要因を読み出すことができます。</p>	
発行手順	<p style="text-align: center;">ア セ ン ブ ラ</p> <pre> ⋮ MOVE.L #3, D0 LEA PARA, A0 TRAP #4 ⋮ PARA : DC.L TN DC.L FACT </pre>	<p style="text-align: center;">C 言 語</p> <pre> ⋮ long tn, fact ; ⋮ queue(&tn, &fact) ; </pre>
パラメータ	<p>t n : 倍精度整数型変数又は定数 起動したいタスクのタスク番号</p> <p>f a c t : 倍精度整数型変数又は定数 起動するタスクに渡す起動要因</p>	
リターンコード	<p>0 : 正常終了 (被要求タスクは実行待ちにつながれた)</p> <p>1 : 指定された被要求タスクは Dormant 状態であった。</p>	
パラメータチェック	<p>(OSは) $0 < t n \leq 128$ でないとき、又は $0 \leq f a c t \leq 16$ でないとき、その要求をパラメータエラーとし、要求タスクの処理を打ち切り、そのタスクを Dormant 状態にします。</p>	
備 考	<ul style="list-style-type: none"> ・要求タスクが本マクロを発行した時に被要求タスクが実行待ち行列につながっている場合、そのタスクは2回起動されることとなります。同一タスクの待ち行列登録は2度までです。 ・被要求タスクが fact 読み出しをする前に同じタスクが同じ起動要因 (fact) で行列に登録された時、そのタスクの起動要因は0が設定されます。同じタスクについて、同じ fact は一回しか登録されません。 	

4.2.3 abort (abort)

機能	パラメータで示された被要求タスク番号 (tn) を持つ被要求タスクを強制終了させ、Dormant 状態にします。該当タスクに対してすでに設定されていた起動要因があれば取消します。									
発行手順	ア セ ン ブ ラ	C 言 語								
	<pre> : MOVE.L #1, D0 MOVE.L TN, A0 TRAP #4 : </pre>	<pre> : long tn ; : abort (&tn) ; : </pre>								
パラメータ	tn : 倍精度整数型変数又は定数 Dormant 状態としたい被要求タスクのタスク番号であり次の形にしてください。									
	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="text-align: center;">31</td> <td style="text-align: center;">8</td> <td style="text-align: center;">7</td> <td style="text-align: center;">0</td> </tr> <tr> <td style="text-align: center;">0 ←</td> <td colspan="2" style="text-align: center;">→ 0</td> <td style="text-align: center;">タスク番号</td> </tr> </table>		31	8	7	0	0 ←	→ 0		タスク番号
31	8	7	0							
0 ←	→ 0		タスク番号							
リターンコード	0 : 正常終了 常に正常としてリターンコードを返します。									
パラメータチェック	0 < tn ≤ 128 でないときパラメータエラーとして要求タスクの処理を打ち切り Dormant 状態とします。									
備考	・対象となった被要求タスクは、一時的に変更されていた状態が初期設定されていた状態に戻されます。									

4.2.4 delay (delay)

機能	このマクロ命令発行要求タスクをパラメータで指定された時間だけ中断します。中断中は他タスクに制御が移ります。指定された時間中断後、他に動作可能なタスク（このマクロ命令発行タスクのレベルより高レベルないし、同レベルで先に起動されているもの）がなければこのマクロ命令発行タスクに制御が戻ります。	
発行手順	<p style="text-align: center;">ア セ ン ブ ラ</p> <pre> ⋮ MOVE.L #6, D0 MOVE.L T, A0 TRAP #4 ⋮ </pre>	<p style="text-align: center;">C 言 語</p> <pre> ⋮ long t ; ⋮ delay (&t) ; ⋮ </pre>
パラメータ	t : 倍精度整数型変数又は定数 中断したい時間を msec 単位で指定します。	
リターンコード	<p>0 : 正常終了</p> <p>1 : タイマーテーブル (TRB) エリア満杯により中断できなかった。</p> <p>2 : スタック退避エリア (ARSB) 満杯により中断できなかった。</p>	
パラメータチェック	0 < t ≤ 86400000 でないときパラメータエラーとし、要求タスクの処理を打ち切り、Dormant 状態とします。	
備 考	<ul style="list-style-type: none"> ・パラメータは 0 < t ≤ 86400000 (msec) でなければなりません。 ・リターンコードが 0 でないとき中断はされていません。 	

4.2.5 timer (timer)

機能	<p>指定した tn をもつ被要求タスクをタイマーテーブル (TRB) に登録し、指定時間経過後に起動するとともに、その後周期時間経過するごとに起動します。ただし、周期時間 (cyt) = 0 のときは、指定した tn を指定時間経過後に一度起動します。起動するときにパラメータで指定された起動要因も渡します。</p>	
発行手順	<p style="text-align: center;">ア セ ン ブ ラ</p> <pre> ⋮ MOVE.L #7, D0 LEA PARA, A0 TRAP #4 ⋮ PARA : DC.L TN DC.L T DC.L CYT DC.L FACT </pre>	<p style="text-align: center;">C 言 語</p> <pre> ⋮ long tn, t, cyt, fact ; ⋮ timer (&tn, &t, &cyt, &fact) ; ⋮ </pre>
パラメータ	<p>tn : 倍精度整数型変数又は定数 起動したい被要求タスクのタスク番号を指定します。</p> <p>t : 倍精度整数型変数又は定数 最初に起動をかけるまでの時間 (msec)</p> <p>cyt : 倍精度整数型変数又は定数 周期時間 (msec) を指定します。</p> <p>$fact$: 倍精度整数型変数又は定数 起動する被要求タスクに渡す起動要因を指定します。</p>	
リターンコード	<p>0 : 正常終了</p> <p>1 : タイマーテーブル (TRB) 満杯で処理不成功</p>	
パラメータチェック	<p>$0 < tn \leq 128$ でないとき又は、$0 < t \leq 84600000$ でないとき又は $0 \leq cyt \leq 84600000$ でないとき又は $0 \leq fact \leq 16$ でないとき又は $para$ が WORD 境界にならないときは、パラメータエラーとし、要求タスクの処理を打ち切り、Dormant 状態とします。</p>	
備考	<ul style="list-style-type: none"> ・パラメータ t は $0 < t \leq 86400000$ ms でなければなりません。 ・パラメータ cyt は $0 \leq cyt \leq 86400000$ ms でなければなりません。 ・パラメータ tn で指定した被要求タスクが Dormant 状態であると起動はされません。 ・このマクロ命令を取消するには $ctime$ マクロ命令を使用しなければなりません。 	

4.2.6 c t i m e (cancel time)

機能	<p>TIMER マクロ命令で登録されたタスクの起動要求を取消します。即ち、OSはパラメータで指定された t n, fact がタイマーテーブル (TRB) に登録されているか否か調べ、登録されていればこれを取消します。取消されるものは t n, fact が一致しているものすべてです。</p>	
発行手順	<p style="text-align: center;">ア セ ン ブ ラ</p> <pre> : MOVE.L #8, D0 LEA PARA TRAP #4 : PARA : DC.L TN DC.L FACT : </pre>	<p style="text-align: center;">C 言 語</p> <pre> : long tn, fact ; : ctime (&tn, &fact) ; : </pre>
パラメータ	<p>t n : 倍精度整数型変数又は定数 取消したい TIMER マクロ命令で指定した被要求タスク番号</p> <p>f a c t : 倍精度整数型変数又は定数 取消したい TIMER マクロ命令で指定した起動要因</p>	
リターンコード	<p>0 : 正常終了</p> <p>1 : パラメータで指定された t n 及び fact はタイマーテーブル (TRB) に登録されていなかった。</p>	
パラメータチェック	<p>$0 < t n \leq 128$ でないとき又は $0 \leq f a c t \leq 16$ でないとき又は para が Word境界にないときパラメータエラーとし、要求タスク処理を打切り Dormant 状態とします。</p>	
備 考	<ul style="list-style-type: none"> ・すでに起動されて動作中の被要求タスクをこのマクロ命令により打切ることはできません。すでに起動されて動作中のときは以降の起動が行われなくなります。 ・複数の timer マクロ命令も t n, fact がこのマクロ命令のパラメータと一致している限り1つのこのマクロ命令ですべて取消されます。 	

4.2.7 chap (change priority level)

機 能	<p>パラメータで示された被要求タスクの優先レベルを変更します。即ち、パラメータで指定した被要求タスクの実行レベルを変更します。このマクロ命令発行要求タスクよりも高いレベルに変更すると、対象被要求タスクが動作可能であれば対象被要求タスクに制御が移ります。</p>	
発行手順	<p style="text-align: center;">ア セ ン ブ ラ</p> <pre> ⋮ MOVE.L #10, D0 LEA PARA, A0 TRAP #4 ⋮ PARA : DC.L TN DC.L LEVEL ⋮ </pre>	<p style="text-align: center;">C 言 語</p> <pre> ⋮ long tn, level ; ⋮ chap (&tn, &level) ; ⋮ </pre>
パラメータ	<p><code>tn</code> : 倍精度整数型変数又は定数 レベルを変更したい被要求タスクのタスク番号</p> <p><code>level</code> : 倍精度整数型変数又は定数 タスクに与える優先レベル</p>	
リターンコード	<p>0 : 正常終了</p> <p>1 : スタック退避エリア (ARSB) が満杯で処理不成功</p>	
パラメータチェック	<p>$0 < tn \leq 128$でないとき又は、$0 \leq level \leq 4$でないとき又は <code>para</code> が word 境界にないときはパラメータエラーとし、被要求タスクの処理を打ち切り、Dormant 状態とします。</p>	
備 考	<ul style="list-style-type: none"> ・ 自タスクに対してこのマクロ命令を発行し、レベルを下げると他一般タスクに制御が移ることがあります。 ・ 対象となった被要求タスクは、それがすでに起動されている時、パラメータで指定されたレベルのタスクの中では、最もあとから起動されたものとして扱われます。 ・ このマクロ命令の効果は、対象被要求タスクが終了したときに失われます。次に起動されたときは元のレベルで動作します。 	

4.2.8 chmod (change mode)

機能	自タスクのステータスレジスタの内容を変更します。即ち、ステータスレジスタの割込禁止レベルを変更し、割込み禁止状態を作り出します。ただし、割込み禁止状態とするのはレベル3までとします。	
発行手順	アセンブラ ⋮ MOVE.L #9, D0 MOVE.W WSR, A0 TRAP #4 ⋮	C 言語 ⋮ short wsr ; ⋮ chmod (&wsr) ; ⋮
パラメータ	wsr : 単精度整数型変数又は定数 変更するステータスレジスタの内容	
リターンコード	変更する前のステータスレジスタの内容	
パラメータチェック	このマクロ命令はパラメータチェックを行いません。	
備考	<ul style="list-style-type: none"> ステータスレジスタ (SR) の割込みマスクレベル及びコンディションコードを変更します。 <div style="text-align: center;"> </div> <ul style="list-style-type: none"> 割込禁止レベルは3までとし、それ以上を指定した場合は3とします。 このマクロ命令でタスクの割込禁止レベルのタスクを変更できますが、割込み禁止時間は必要最小限に止めなければなりません。(2 msecを超えて割込み禁止にしてはなりません) このマクロ命令で変更されたSRは、このマクロ命令発行タスク内でのみ有効です。他のタスクにまで効力は及びません。 このマクロ命令発行タスクが終了又は ABORT したとき、このマクロ命令の効果は失われます。 リターンコードの下 word にこのマクロ命令発行時のSRが格納されます。上 word は“0”となります。 	

4.2.9 s f a c t (set factor)

機 能	パラメータで指定した被要求タスクに起動要因をセットします。即ち、パラメータで指定された被要求タスクの起動要因テーブルに起動要因をセットします。セットされた起動要因は GFACT マクロ命令により取込むことができます。	
発行手順	<pre> ア セ ン ブ ラ ... MOVE.L #4, D0 LEA PARA, A0 TRAP #4 ... PARA : DC.L TN DC.L FACT ... </pre>	<pre> C 言 語 ... long tn, fact ; ... sfact (&tn, &fact) ; ... </pre>
パラメータ	<p>t n : 倍精度整数型変数又は定数 起動要因を設定したい被要求タスク番号を与えます。</p> <p>f a c t : 倍精度整数型変数又は定数 起動要因を与えます。</p>	
リターンコード	<p>0 : 正常終了</p> <p>1 : 指定起動要因は登録済み</p> <p>2 : 指定被要求タスクは Dormant 状態</p>	
パラメータチェック	<p>$0 < t n \leq 128$ でないとき又は $0 \leq f a c t \leq 16$ でないときパラメータエラーとし要求タスクの処理を打切り Dormant 状態とします。</p>	
備 考	<ul style="list-style-type: none"> ・ 起動要因は 0 ~ 16 の範囲で指定可能です。 ・ 対象被要求タスクが Dormant 状態であるとき起動要因はセットされません。 ・ 同一起動要因を多重にセットすることはできません。多重にセットしようとした場合、リターンコード=1 がセットされます。 ・ セットされた起動要因は、gfact マクロ命令で取込まれると起動要因テーブルから削除されます。また、abort マクロ命令を発行されたときは、全要因がクリアされます。 	

4.2.10 g f a c t (get factor)

機 能	このマクロ命令発行要求タスクに対してセットされている起動要因を取出します。即ち、自タスクの起動要因テーブルからセットされている起動要因を数値の小さい順に1つ取出します。取出された要因は、起動要因テーブルから削除されます。取出されずに残っている起動要因は、再度このマクロ命令を発行することにより取込まれます。	
発行手順	ア セ ン ブ ラ ⋮ MOVE.L #5, D0 LEA FACT, A0 TRAP #4 ⋮	C 言 語 ⋮ long fact ; ⋮ gfact (&fact) ; ⋮
パラメータ	f a c t : 倍精度型変数 このマクロ命令発行タスクに対してセットされている起動要因が、このマクロ命令の実行結果として格納されます。	
リターンコード	0 : 正常終了 常に正常としてリターンコードを返します。	
パラメータチェック	fact が word 境界にないときパラメータエラーとしタスクの処理を打切り Dormant 状態とします。	
備 考	<ul style="list-style-type: none"> ・タスクは起動されたら、このマクロ命令を発行して起動要因を取込み、起動要因がすべて取込まれてから終了してください。 ・起動要因がセットされていない状態でこのマクロ命令を発行すると、fact には、0 が渡されます。上記文で起動要因がすべて取込まれた状態とは、このマクロ命令発行結果 fact に0が渡されることにより知ることができます。 	

4.2.11 `uspchk` (user stack pointer check)

機能	このマクロ命令発行タスクが、パラメータで指定されたバイト数を超えてスタックを使用しているか否かを Check します。このマクロ命令をプログラムのどの位置で Call するかは、ユーザで検討してください。	
発行手順	ア セ ン ブ ラ	C 言 語
	<pre> ⋮ MOVE.L #11, D0 LEA PARA, A0 TRAP #4 ⋮ PARA : DC.L USEBYT DC.L ADDR ⋮ </pre>	<pre> ⋮ long usebyt, addr ; ⋮ uspchk (&usebyt, &addr); ⋮ </pre>
パラメータ	<p><code>usebyt</code> : 倍精度整数型変数又は定数 該当要求タスクで確保しているスタックエリアのバイト長を与えます。</p> <p><code>addr</code> : 倍精度型変数 正常終了時、未使用となっているバイト数が、異常終了時オーバーしているバイト数がこのマクロの実行結果として格納されます。</p>	
リターンコード	<p>0 : 正常終了 (スタックエリアに空あり) <code>addr</code> に空きバイト数を格納します。</p> <p>1 : スタックエリアに空なし <code>addr</code> にオーバーしたバイト数を格納します。</p>	
備 考	<ul style="list-style-type: none"> ・プログラムのネスティングの最も深い所でこのマクロ命令を実行する事が最も効果的です。 ・デバック終了後は、このマクロ命令をプログラムより削除する事をお勧めします。 ・ユーザは、リターンコード判定により、異常処理を記述する必要があります。 	

4.2.12 `stime` (`set time` (1))

機能	このマクロ命令は実時刻を設定します。即ち、時刻を管理しているオプションの時計付拡張メモリにパラメータで指定された時刻を設定します。	
発行手順	<p style="text-align: center;">アセンブラ</p> <pre> ⋮ MOVE.L #13,D0 LEA PARA,A0 TRAP #4 TST.L D0 ⋮ PARA: DC.L SEC DC.W DAY DC.W MONTH DC.W YEAR DC.W WEEK ⋮ </pre>	<p style="text-align: center;">C 言語</p> <pre> typedef struct { long sec ; short day ; short month ; short year ; short week ; } TIME; main() { long rtn; static TIME time { SEC, DAY, MONTH, YEAR, WEEK, }; ⋮ rtn = stime(&time); ⋮ } </pre>
パラメータ	<p><code>time</code> …… 設定する時刻が格納されているエリアです。</p> <p><code>sec, day, month, year, 及び week</code>には下記を与えます。</p> <p><code>sec</code> : 午前0時(深夜)を0として秒単位で時刻を与えます。 <code>day</code> : 日を与えます。 <code>month</code>: 月を与えます。 <code>year</code> : 西暦年を与えます。 <code>week</code> : 曜日を与えます。</p>	

(set time (2))

リターン コード	<p>0 : 正常終了 1 : RTCハードウェア未実装 リターンコード=0 以外の時 このマクロ命令は無効です。</p>																										
パラメータ チェック	<p>パラメータが以下の場合、パラメータエラーとなり、タスクの実行を打ち切り、タスクをDormant状態とします。</p> <p style="text-align: center;"> $0 \leq \text{sec} < 86400$ でない時 $1 \leq \text{day} \leq 31$ でない時 $1 \leq \text{month} \leq 12$ でない時 $1900 \leq \text{year} \leq 2199$ でない時 $1 \leq \text{week} \leq 7$ でない時 </p>																										
備 考	<p>(a) このマクロ命令を使用するにはオプションの時計付拡張メモリが必要です。</p> <p>(b) 時刻の内 時, 分, 秒はSecのエリアに次のようにセットします。 (例) A時 B分 C秒をセットする場合 $\text{SEC} = A \times 3600 + B \times 60 + C$</p> <p>(c) 曜日 (week) の数字は次のように対応します。</p> <table border="1" data-bbox="478 1265 1181 1355" style="margin-left: auto; margin-right: auto;"> <tr> <td>week</td> <td>1</td> <td>2</td> <td>3</td> <td>4</td> <td>5</td> <td>6</td> <td>7</td> </tr> <tr> <td>曜日</td> <td>日</td> <td>月</td> <td>火</td> <td>水</td> <td>木</td> <td>金</td> <td>土</td> </tr> </table> <p>(d) 時計付拡張メモリの制約上、下記の時刻を設定すると日付の更新を誤って行う事があります。したがって、下記の時刻をさけて設定してください。</p> <table border="1" data-bbox="438 1500 1300 1892" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>設定日付、時刻および日付更新後の状態</th> <th>具体例</th> </tr> </thead> <tbody> <tr> <td>すべての月の29日23時59分59秒を設定すると次の月の1日に更新される。 (1~12月ただし閏年の2月29日は除く)</td> <td>3月29日→ 4月1日</td> </tr> <tr> <td>4月, 6月, 9月, 11月の30日23時59分59秒を設定するとその月の31日に更新される。</td> <td>4月30日→ 4月31日</td> </tr> <tr> <td>閏年でない年の2月28日23時59分59秒を設定すると2月29日に更新される。</td> <td>83年2月28日→ 83年2月29日</td> </tr> <tr> <td>閏年の2月28日23時59分59秒を設定すると3月1日に更新される。</td> <td>84年2月28日→ 84年3月1日</td> </tr> </tbody> </table>	week	1	2	3	4	5	6	7	曜日	日	月	火	水	木	金	土	設定日付、時刻および日付更新後の状態	具体例	すべての月の29日23時59分59秒を設定すると次の月の1日に更新される。 (1~12月ただし閏年の2月29日は除く)	3月29日→ 4月1日	4月, 6月, 9月, 11月の30日23時59分59秒を設定するとその月の31日に更新される。	4月30日→ 4月31日	閏年でない年の2月28日23時59分59秒を設定すると2月29日に更新される。	83年2月28日→ 83年2月29日	閏年の2月28日23時59分59秒を設定すると3月1日に更新される。	84年2月28日→ 84年3月1日
week	1	2	3	4	5	6	7																				
曜日	日	月	火	水	木	金	土																				
設定日付、時刻および日付更新後の状態	具体例																										
すべての月の29日23時59分59秒を設定すると次の月の1日に更新される。 (1~12月ただし閏年の2月29日は除く)	3月29日→ 4月1日																										
4月, 6月, 9月, 11月の30日23時59分59秒を設定するとその月の31日に更新される。	4月30日→ 4月31日																										
閏年でない年の2月28日23時59分59秒を設定すると2月29日に更新される。	83年2月28日→ 83年2月29日																										
閏年の2月28日23時59分59秒を設定すると3月1日に更新される。	84年2月28日→ 84年3月1日																										

(set time (3))

備 考	(e) このマクロ命令発行時にwakeマクロ命令によりscheduled状態となっているタスクがある場合、このマクロ命令発行前の時刻とこのマクロ命令により新たに設定された時刻との関係により、起動時刻は次のようになります。	
	起動予定種別	時刻を遅らせる場合
	時刻指定による起動	<p>(Don't care指定ありの場合) 時刻を遅らせることにより起動時刻が24時間以上遠のいたものは修正後の日付の同一時刻に起動予定が移されます。</p> <p>(絶対時刻指定の場合) 時刻の変更に影響されません。</p>
時刻指定による周期起動	同 上	<p>時刻を進めることにより起動時刻が過ぎてしまったものは、最初の起動予定時刻に周期時間を加えていったものが新しく設定した時刻以降となる時刻に起動予定が移されます。</p>
(f) 時刻の変更は現在時刻から±24時間未満でなければなりません。これ以外を指定したときは、日付の遅進には無関係に、現在時刻とこのマクロで指定した新時刻(sec)を比較し、24時間以内の変更と同一の再スケジューリングを行います。		

4.2.13 g t i m e (g e t t i m e (1))

機 能	このマクロ命令は実時刻を取込みます。即ち時計付拡張メモリで管理している時刻をパラメータで指定したエリアへ格納します。	
発行手順	ア セ ン ブ ラ	C 言 語
	<pre> ⋮ MOVE.L #14,D0 LEA PARA,A0 TRAP #4 TST.L D0 ⋮ PARA: DS.L 1 (sec) DS.W 1 (day) DS.W 1 (month) DS.W 1 (year) DS.W 1 (week) ⋮ </pre>	<pre> typedef struct (long sec ; short day ; short month ; short year ; short week ;) TIME ; main() { long rtn; static TIME time ; ⋮ rtn = gtime(&time); ⋮ } </pre>
パラメータ	<p>time 時刻を取込むエリアです。12バイトの大きさが必要です。</p> <p>sec,day,month,year及びweekにはこのマクロ命令の実行結果として下記が格納されます。</p> <p>sec : 時刻が午前0時(深夜)を0として秒単位で格納されます。 day : 日が格納されます。 month: 月が格納されます。 year : 西暦年が格納されます。 week : 曜日が格納されます。</p>	

(get time (2))

リターン コード	0 : 正常終了 1 : 時計付拡張メモリ未実装																
備 考	<p>(a) このマクロ命令を使用するにはオプションの時計付拡張メモリが必要です。</p> <p>(b) 時刻は sec のエリアに次の形で格納されます。 A時 B分 C秒の場合 $sec = A \times 3600 + B \times 60 + C$</p> <p>(c) 曜日は次のように数字で対応します。</p> <table border="1" data-bbox="488 848 1190 927"><tr><td>week</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td></tr><tr><td>曜日</td><td>日</td><td>月</td><td>火</td><td>水</td><td>木</td><td>金</td><td>土</td></tr></table>	week	1	2	3	4	5	6	7	曜日	日	月	火	水	木	金	土
week	1	2	3	4	5	6	7										
曜日	日	月	火	水	木	金	土										

4.2.14 wake (wakeup task (1))

機能	<p>このマクロ命令はパラメータで指定された番号の被要求タスクをシステムテーブル (ARB) に登録し、Scheduled状態にします。Scheduled状態となった被要求タスクはパラメータで指定された時刻に起動されます。周期起動のときはさらにその後周期時間経過するごとに起動されます。起動するときにはパラメータで指定された fact を起動要因としてタスクに渡します。</p>	
発行手順	<pre> ⋮ MOVE.L #15, D0 LEA PARA, A0 TRAP #4 TST.L D0 ⋮ PARA: DC.L ID DC.L TN DC.L FACT DC.L SEC DC.W DAY DC.W MONTH DC.W YEAR DC.W DUMMY DC.L CYCLE ⋮ </pre>	<pre> typedef struct { long sec ; short day ; short month ; short year ; short week ; } TIME ; main() { long rtn; static long id, tn, fact, cycle; static TIME time { SEC, DAY, MONTH, YEAR, 0 }; ⋮ rtn = wake (&id, &tn, &fact, &time, &cycle); ⋮ } </pre>
パラメータ	<pre> id : 倍精度整数型変数又は定数 起動モード (0 : 時刻起動, 1 : 周期起動) です。 tn : 倍精度整数型変数又は定数 起動したいタスクの番号です。 fact : 倍精度整数型変数又は定数 起動したいタスクに渡す起動要因です。 time : 12バイトの起動する時刻を設定したテーブルです。 sec : 午前0時 (深夜) を0として秒単位で与えます。 day : 日を与えます。 month: 月を与えます。 year : 西暦年を与えます。 week : 使用しませんので0をセットしてください。 cycle : 倍精度整数型変数又は定数 周期時間です。 </pre>	

(wakeup task (2))

<p>パラメータ</p>	<p>(1) idとtime及びcycleの関係は次のようになります。</p> <table border="1" data-bbox="359 459 1428 772"> <thead> <tr> <th>id</th> <th>time</th> <th>cycle</th> <th>内 容</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>起動時刻</td> <td>0</td> <td>[時刻起床] パラメータtimeで指定した時刻に1回だけ指定された被要求タスクを起動します。</td> </tr> <tr> <td>1</td> <td>最初の起動時刻</td> <td>最初の起動時刻以降、周期的に起動する時間</td> <td>[時刻周期起床] パラメータtimeで指定した時刻に指定されたタスクを起動します。その後、cycleで指定された周期ごとに被要求タスクを起動します。</td> </tr> </tbody> </table> <p>(2) 起動時刻はDon't careコード(=-1)を使用する事により次のような設定が可能です。</p> <table border="1" data-bbox="359 918 1428 1265"> <thead> <tr> <th>No.</th> <th>year</th> <th>time</th> <th>day</th> <th>sec</th> <th>起 動 時 間</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>1990</td> <td>1</td> <td>10</td> <td>36610</td> <td>1990年 1月 10日 10時 10分 10秒に起動されます。(絶対時刻指定)</td> </tr> <tr> <td>2</td> <td>-1 Don't care</td> <td>1</td> <td>10</td> <td>36610</td> <td>今年又は翌年の1月 10日 10時 10分 10秒に起動されます。(**)</td> </tr> <tr> <td>3</td> <td>(*)</td> <td>-1 Don't care</td> <td>10</td> <td>36610</td> <td>今月又は翌月の10日 10時 10分 10秒に起動されます。</td> </tr> <tr> <td>4</td> <td>(*)</td> <td>(*)</td> <td>-1 Don't care</td> <td>36610</td> <td>今日又は翌日の10時 10分 10秒に起動されます。(**)</td> </tr> </tbody> </table> <p>(*) Don't careコードより位の大きいデータは無視されます。 (**) 現在時刻より前の場合翌年, 翌月又は翌日に起動されます。また後の場合は今年, 今月又は今日に起動されます。</p>	id	time	cycle	内 容	0	起動時刻	0	[時刻起床] パラメータtimeで指定した時刻に1回だけ指定された被要求タスクを起動します。	1	最初の起動時刻	最初の起動時刻以降、周期的に起動する時間	[時刻周期起床] パラメータtimeで指定した時刻に指定されたタスクを起動します。その後、cycleで指定された周期ごとに被要求タスクを起動します。	No.	year	time	day	sec	起 動 時 間	1	1990	1	10	36610	1990年 1月 10日 10時 10分 10秒に起動されます。(絶対時刻指定)	2	-1 Don't care	1	10	36610	今年又は翌年の1月 10日 10時 10分 10秒に起動されます。(**)	3	(*)	-1 Don't care	10	36610	今月又は翌月の10日 10時 10分 10秒に起動されます。	4	(*)	(*)	-1 Don't care	36610	今日又は翌日の10時 10分 10秒に起動されます。(**)
id	time	cycle	内 容																																								
0	起動時刻	0	[時刻起床] パラメータtimeで指定した時刻に1回だけ指定された被要求タスクを起動します。																																								
1	最初の起動時刻	最初の起動時刻以降、周期的に起動する時間	[時刻周期起床] パラメータtimeで指定した時刻に指定されたタスクを起動します。その後、cycleで指定された周期ごとに被要求タスクを起動します。																																								
No.	year	time	day	sec	起 動 時 間																																						
1	1990	1	10	36610	1990年 1月 10日 10時 10分 10秒に起動されます。(絶対時刻指定)																																						
2	-1 Don't care	1	10	36610	今年又は翌年の1月 10日 10時 10分 10秒に起動されます。(**)																																						
3	(*)	-1 Don't care	10	36610	今月又は翌月の10日 10時 10分 10秒に起動されます。																																						
4	(*)	(*)	-1 Don't care	36610	今日又は翌日の10時 10分 10秒に起動されます。(**)																																						
<p>リターンコード</p>	<p>0 : 正常終了 1 : システムテーブル (ARB) 満杯 2 : 時計付拡張メモリ未実装 リターンコード=0以外の場合このマクロ命令は無効です。</p>																																										
<p>パラメータチェック</p>	<p>パラメータが次の場合、タスクの実行を打切りDormant状態とします。</p> <p>0 ≤ id ≤ 1 以外 1 ≤ tn ≤ 128 以外 0 ≤ fact ≤ 16 以外 0 ≤ sec < 86400 以外 1 ≤ cycle ≤ 86400 以外 1 ≤ day ≤ 31 以外 1 ≤ month ≤ 12 以外 1900 ≤ year ≤ 2199 以外</p>																																										

(wakeup task (3))

備 考	<ul style="list-style-type: none">(a) パラメータfactは $0 \leq \text{fact} \leq 16$ でなければなりません。(b) パラメータtime中のsecは $0 \leq \text{sec} < 86400$ でなければなりません。この時刻は午前0時を起点に秒単位で時刻を設定してください。(c) 起動予定時刻にパラメータtnで指定された被要求タスクがDormant状態であると起動されません。(d) このマクロを取消するにはcwakeマクロ命令を使用しなければなりません。(e) 周期起動のときcycleは $0 < \text{cycle} \leq 86400$ でなければなりません。(f) このマクロ命令によってシステムテーブル(ARB)に登録されている被要求タスクをScheduled状態にしても、cwakeマクロ命令により登録を取消しないと、システムテーブル(ARB)は解放されません。(g) 現在時刻より前の時刻をパラメータで指定した場合は翌日の同一時刻に起動されます。
-----	--

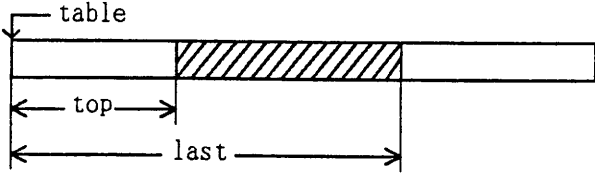
4.2.15 `cwake` (cancel wakeup task)

機能	このマクロ命令は <code>wake</code> マクロ命令で登録された被要求タスクの起動要求を取消します。即ち、パラメータで指定された番号の被要求タスクがシステムテーブル (ARB) に登録されているか否か調べ、登録されていればパラメータで指定された起動要因と一致するかを調べます。その後起動要因が一致していれば、システムテーブルから削除し、登録を取消します。パラメータで指定されたタスク番号及び起動要因が一致する被要求タスクすべてが取消の対象となります。	
発行手順	<pre> アセンブラ ... MOVE.L #16, D0 LEA PARA, A0 TRAP #4 TST.L D0 ... PARA: DC.L TN DC.L FACT ... </pre>	<pre> C言語 main() { long tn, fact, rtn; ... rtn = cwake (&tn, &fact); ... } </pre>
パラメータ	<p><code>tn</code> : 倍精度整数型変数又は定数 タスク番号を指定します。</p> <p><code>fact</code> : 倍精度整数型変数又は定数 起動要因を指定します。</p>	
リターンコード	<p>0 : 正常終了</p> <p>1 : パラメータで指定された<code>tn</code>, 及び<code>fact</code>と一致するものはシステムテーブル (ARB) に登録されていなかったため処理不成功。</p> <p>2 : 時計付拡張メモリ未実装</p>	
パラメータチェック	<p>パラメータが次の場合被要求タスクの実行は打切られDormant状態となります。</p> <p>1 : $1 \leq tn \leq 128$ 以外</p> <p>2 : $0 \leq fact \leq 16$ 以外</p>	
備考	<p>(a) すでに起動されて動作中の被要求タスクをこのマクロ命令により打切ることはできません。 すでに起動されて動作中のときは、以降の起動予定が取消されます。</p> <p>(b) 複数個の<code>wake</code>マクロ命令も、<code>tn</code>, <code>fact</code>がこのマクロ命令のパラメータと一致している限り、1つの<code>cwake</code>マクロ命令ですべて取消されます。</p>	

4.2.16 r s e r v (r e s e r v (1))

機能	<p>このマクロ命令発行要求タスクがパラメータで指定されたタスク間共通メモリを占有します。このマクロ命令発行タスクがすでに r s e r v マクロ命令により資源占有の場合今回のマクロ発行は無効となります。</p> <p>占有中でない場合、今回占有しようとするエリアが他の一般タスクにより占有されていないかをチェックし、占有しようとするエリアすべてが他のどの一般タスクからも占有されていない時に限りパラメータで指定されたエリアを占有します。そうでない時は該当エリアが f r e e マクロ命令により占有解除になるまでこのマクロ命令発行要求タスクを待ち状態とします。</p> <p>待ち状態となった要求タスクは該当エリアが解放され、かつパラメータで指定したすべてのエリアが占有可能となった時、エリアを占有し制御を受取ります。</p> <p>このマクロ命令によるエリアの占有はハードウェア機構によるインターロックではありませんので、このマクロ命令によりエリアを占有しても、他のタスクからの該当エリアに対するアクセスを防止することはできません。これを防止するには該当エリアをアクセスする前に必ず該当エリアをこのマクロ命令により占有してからアクセスするという約束を該当するエリアをアクセスするすべてのタスクが守らなければなりません。</p>	
発行手順	<pre> アセンブラ : MOVE.L #17, D0 LEA PARA, A0 TRAP #4 TST.L D0 : PARA: DC.L CASE DC.L TYPE1 DC.L TABLE1 DC.L TOP1 DC.L LAST1 DC.L TYPE2 DC.L TABLE2 DC.L TOP2 DC.L LAST2 : : : </pre> <p style="text-align: right;">} ケース #1</p> <p style="text-align: right;">} ケース #2</p>	<pre> C 言 語 : typedef struct { long type ; long *table ; long top ; long last ; } para ; main() { long case, rtn; static para para1 { 0, &table1(0), top1, last1}; static para para2 { 0, &table2(0), top2, last2}; : rtn = rserv (&case, &para1, &para2, ...); : } </pre>

(r e s e r v (2))

パラメータ	<p>case : 倍精度整数型変数又は定数 占有したい資源のケース数です。</p> <p>type : 倍精度整数型変数又は定数 占有したい資源の種別です。CPMSでは必ず'0'を指定してください。</p> <p>table : 倍精度整数型変数又は定数 占有したいエリアのテーブル先頭アドレスです。</p> <p>top : 倍精度整数型変数又は定数 占有したいエリア先頭からの相対バイトアドレスです。</p> <p>last : 倍精度整数型変数又は定数 占有したいエリア先頭からの相対バイトアドレスです。 占有されるエリアは下記の斜線部です。</p> 
リターンコード	<p>0 : 占有成功</p> <p>1 : すでに r e s e r v マクロ命令発行済</p> <p>2 : パラメータに不合理な指定があった。</p> <p>3 : A R S B 満杯</p> <p>リターンコードが0以外の時このマクロ命令は無効です。</p>
パラメータチェック	<p>パラメータが次の場合要求タスクの実行は打切られDormant状態とします。</p> <p>$1 \leq \text{case} \leq 32$ 以外</p>
備考	<p>(a) このマクロ命令の効力はタスクが終了 (E X I T 又は A B O R T) したときには失われます。</p> <p>(b) 占有しようとしたエリアが他タスクにより占有されているエリアと重複しているときは占有できません。この時このマクロ命令を発行したタスクは待ち (実行抑止) 状態となります。</p> <p>(c) このマクロ命令は同一タスクから2重発行できません。占有したい資源は1回にすべて占有しなければなりません。ただ f r e e マクロ命令ですべての資源を解放した後は再びこのマクロ命令を発行できます。</p> <p>(d) このマクロ命令により資源を占有する期間はできるだけ短くなるようにしてください。</p> <p>(e) このマクロ命令はタスク間の排他制御を行いますが、OS及びハードウェアの制御は行いません。</p> <p>(f) 他タスクの実行を抑止するマクロ命令を発行後、このマクロ命令を発行することは避けてください。該当タスクが資源をすでに占有中であるとき、デッドロックとなります。</p> <p>(g) 占有した資源を f r e e マクロ命令ですべて解放しないまま E X I T した場合、CPUのコンソールLEDに“ R S V E R R ”と表示されます。</p>

4.2.17 free (free (1))

機能	<p>このマクロ命令は r s e r v マクロ命令により占有している資源の占有を解除し、他タスクへ解放します。</p> <p>即ち、このマクロ命令発行要求タスクが、r s e r v マクロ命令により占有中の資源と、このマクロ命令のパラメータで指定された資源が一致していれば、その資源の占有を解除します。</p>	
発行手順	<p style="text-align: center;">ア セ ン ブ ラ</p> <pre> ⋮ MOVE.L #18, D0 LEA PARA, A0 TRAP #4 TST.L D0 ⋮ PARA: DC.L CASE DC.L TYPE1 DC.L TABLE1 DC.L TOP1 DC.L LAST1 DC.L TYPE2 DC.L TABLE2 DC.L TOP2 DC.L LAST2 ⋮ </pre> <p style="margin-left: 150px;">} ケース #1</p> <p style="margin-left: 150px;">} ケース #2</p>	<p style="text-align: center;">C 言 語</p> <pre> ⋮ typedef struct { long type ; long *table ; long top ; long last ; } para ; ⋮ main() { long case, rtn; static para para1 { 0, &table1(0), top1, last1}; static para para2 { 0, &table2(0), top2, last2}; ⋮ rtn = free (&case, &para1, &para2, ...); ⋮ </pre>
パラメータ	このマクロ命令のパラメータは r s e r v マクロ命令のパラメータと同一です。	
リターンコード	<p>0 : 指定された全ケースについて占有解除が成功した。</p> <p>1 : 指定されたパラメータのうち少なくとも1つは、自分が占有している資源ではなかった。</p> <p>リターンコード=1であっても、指定したパラメータが占有中の資源と一致したものについては該当資源の占有は解除されます。</p>	

(free(2))

<p>パラメータ チェック</p>	<p>$1 \leq \text{case} \leq 32$ 以外の時要求タスクの実行は打切られDormant状態となります。</p>
<p>備考</p>	<p>(a) このマクロ命令で占有中の資源を解放したとき、該当資源を占有しようとして待ちとなっていたタスクの待ち状態は解除されます。</p> <p>(b) freeマクロ命令のパラメータはrserveマクロ命令で指定したパラメータと一致していなければなりません。不一致の場合占有中の資源を解放する事はできません。</p> <div style="text-align: center;"> <p style="text-align: right;">} RSERVマクロ命令の パラメータ1ケースで 指定した資源</p> <p style="text-align: right;">} 解放 されない</p> <p style="text-align: right;">} 解放される</p> <p style="text-align: right;">} FREEマクロ命令の パラメータ1ケースで指 定した資源</p> </div> <p>資源解放の条件</p> <p>table(R) = table(F)</p> <p>top(R) = top(F)</p> <p>last(R) = last(F)</p>

4.2.18 mvmem (move memory (1))

機能	このマクロ命令はパラメータで指定されたデータを指定されたエリアへ転写します。このマクロ命令を使用するとプロテクトメモリ（OSプログラムエリアは除く）へのデータの書き込みが可能となります。	
発行手順	<pre> アセンブラ ⋮ MOVE.L #19, D0 LEA PARA, A0 TRAP # 4 TST.L D0 ⋮ PARA: DC.L WNO DC.L DADDR DC.L SADDR ⋮ </pre>	<pre> C言語 ⋮ main() { long rtn, wno, saddr, daddr; ⋮ rtn = mvmem (&wno, &daddr, &saddr); ⋮ } </pre>
パラメータ	<p>wno : 倍精度整数型変数又は定数 word(16ビット)単位の転送語数を与えます。</p> <p>daddr : 倍精度整数型変数又は定数 データの移動先先頭アドレスを与えます。</p> <p>saddr : 倍精度整数型変数又は定数 データの移動元先頭アドレスを与えます。</p>	
リターンコード	0 : 常に正常	

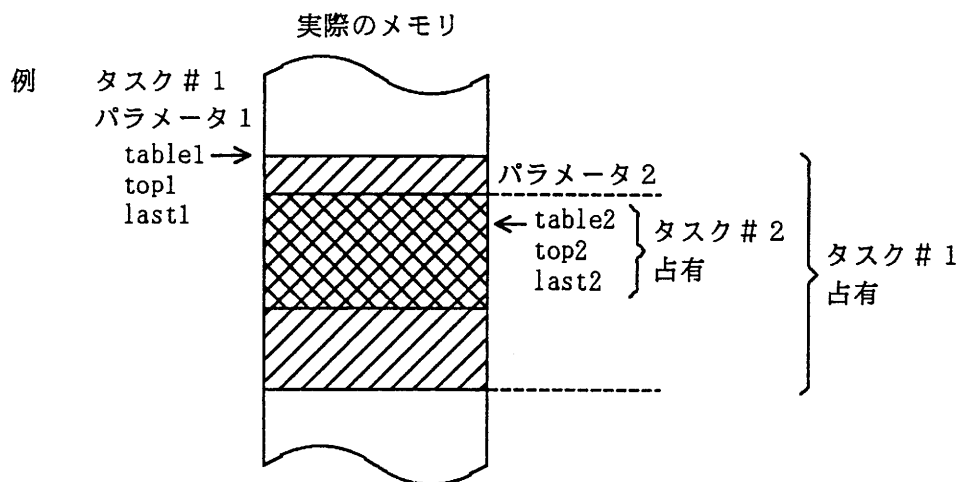
(move memory (2))

<p>パラメータ チェック</p>	<p>パラメータが次の場合要求タスクの実行は打切られ、Dormant状態とします。</p> <p>$0 < wno \leq 256$ 以外 daddrで指定されたエリアがOSにより書き込み禁止されているエリアの場合。</p>
<p>備 考</p>	<p>(a) 転送語数は $0 < wno \leq 256$ でなければなりません。</p> <p>(b) 転送元のエリアと転送先のエリアに重複があってはなりません。データは必ずアドレスの低い方から転送されるため、重複があった場合正常に転写されなくなる場合があります。</p> <div data-bbox="427 763 1412 1093" style="text-align: center;"> <p>転送元 転送先</p> <p>saddr daddr</p> <p>wno</p> <p>重複のため正常に転送されない。</p> </div> <p>(c) このマクロ命令を使用するとプロテクトスイッチのON/OFFに関係なくプロテクトエリアヘータを書込む事ができます。ただし、OSプログラムエリア及びシステムエリアへの書き込みは禁止されています。</p> <div data-bbox="389 1249 1380 1870" style="text-align: center;"> <p>000000 (メモリプロテクトエリア) FFFFF</p> <p>シーケンスプログラムエリア OSプログラムエリア</p> <p>60000 60100 60BFO 61000 63000 63800 7FFFF</p> <p>S E T Dレジスタ T.U.C 設定値 シーケンスプログラム</p> <p>F0000 F0800 (*) FC000 FFFFF</p> <p>T.U.C 計数值 OSプログラム+ワーク TCB他システムテーブル 拡張OSエリア</p> </div> <p>▨ のエリアは書き込み禁止エリアです。</p> <p>(*) OSのバージョンにより異なる。</p>

4.3 補 足

4.3.1 r s e r v と f r e e の 関 連

- rservマクロ命令発行時のパラメータを記憶するためのRSVB (Reserve Block) は32ケースまで使用可能です。したがって、freeマクロ命令を発行しないまま、rservマクロ命令により32個を超えるエリアを占有しようとした場合はエラーとなります。
- rserv及びfreeマクロ命令のパラメータ中、tableが異なるものは別のエリアとして取扱われます。したがって、tableのデータを誤って設定した場合、正常に処理されない場合がありますのでご注意ください。



上記の場合パラメータ 1 と 2 で指定されたエリアは実際には重複しているにもかかわらず、tableの値が異なるため、OSでは別エリアとして扱われ、タスク # 1、タスク # 2とも、エリアをReservしてしまい、排他的制御が正常に行われません。

- タスクがrserv命令で資源を占有したまま、freeマクロ命令でその占有を解除しないまま終了(EXIT)した場合、CPUのコンソールLEDに“RSV ERR”と表示します。

4.3.2 wakeとcwakeの関連

- 時計により時刻起動待ちとなったタスク (Scheduled状態) は最大 8 個までしか記憶されません。8 個以上のタスクを使用しようとした場合、システムテーブル (ARB: Alarm Recording Block) 満杯エラーとなりますのでご注意ください。
- WAKEマクロ命令によりスケジューリングされたものは、cwakeマクロ命令の発行又はCPUのリセット (停復電) 時のみスケジューリングが取消されます。したがって、abort命令によりタスクを実行禁止としても、スケジュールされた時刻にはOSによりQUEUEされます (実際にはIDLE状態ではないため起動されません)。
- wakeマクロ命令によるタスクのスケジューリングはstimeマクロ命令により影響を受けますのでご注意ください。できればstimeマクロ命令はイニシャルタスクで発行する等、タスクにwakeマクロ命令を発行する前に発行するようにしてください。
- 日付のチェックは、1~31の範囲で行っているため、実際に存在しない日付をセットした場合、翌日に起動されます。
(例) 1988年4月31日10時と指定した場合、1988年5月1日10時に起動されます。
なお、Don't Careコードを使用し、上記のように4月31日と指定した場合は、起動されなくなりますのでご注意ください。

4.3.3 `stime`と`gtime`の関連

- 曜日データと日付の関係はOS内部ではチェックしません。したがって、ユーザが`stime`マクロ命令で設定する時に誤りがないよう設定してください。
- システム納入時には1900年1月1日0時0分0秒がセットされますので、`stime`マクロ命令で正しく設定し直してください。
- H-S10/2 α ・2 α Eは内部にカレンダー機能を持っていません。このため、閏年は年が4で割り切れる時にのみ考慮されます。

4.4 マクロ命令使用例

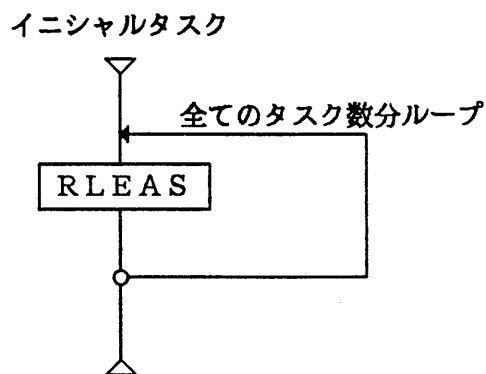
■ RLEAS (release)

RLEASマクロは、Dormant状態のタスクをIdle状態にする為のマクロ命令です。

タスクは、CPU復電時、イニシャルタスク（タスク番号1）を除き、全てDormant（タスクの実行禁止）状態になっていますので、CPUは何も処理を行なう事は出来ません。

通常、システムを作成する場合、イニシャルタスク（CPU復電時にコンパクトPMSから必ず起動されるタスクです）にて、システムを構成している全てのタスクをIdle（タスクの実行可能）状態にします。

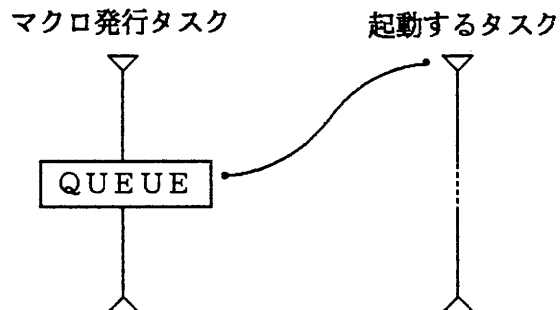
使用例



■ QUEUE (queue)

QUEUEマクロはIdle（タスクの実行可能）状態のタスクを起動し処理を行なわせます。

使用例



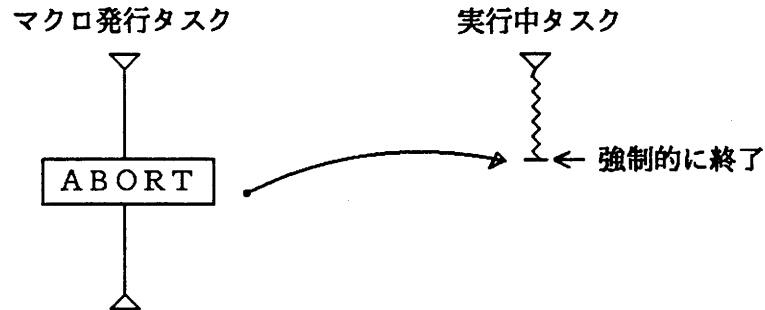
- 起動するタスクがDormant（タスクの実行禁止）状態の場合、起動しません。

■ ABORT (abort)

ABORTマクロは、タスクをDormant (タスクの実行禁止) 状態にします。

ABORTマクロによりDormant状態にされるタスクが現在実行中である場合でも強制的に終了させ、Dormant状態にします。

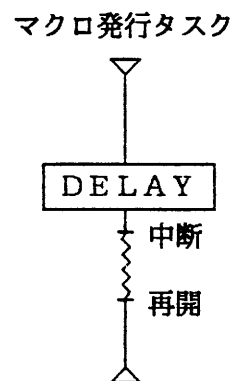
使用例



■ DELAY (delay)

DELAYマクロは、マクロ発行タスクの実行を一時中断し、指定時間経過後、再開させます。

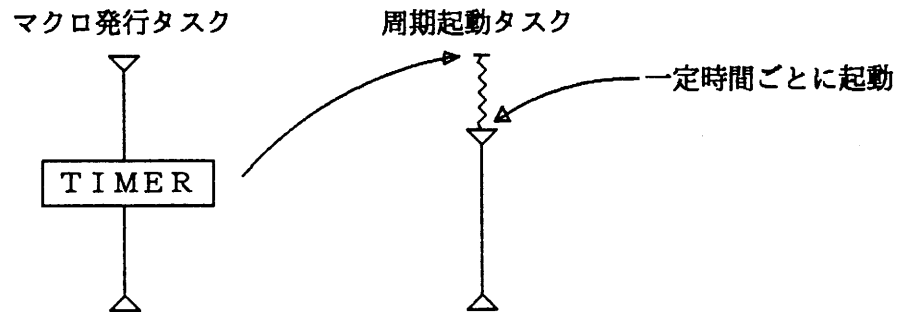
使用例



■ TIMER (timer)

TIMERマクロは、指定したタスクを指定時間後起動し、その後指定した周期時間が経過するごとに指定したタスクを起動します。つまり、あるタスクを周期起動させる為に用いるマクロ命令です。

使用例

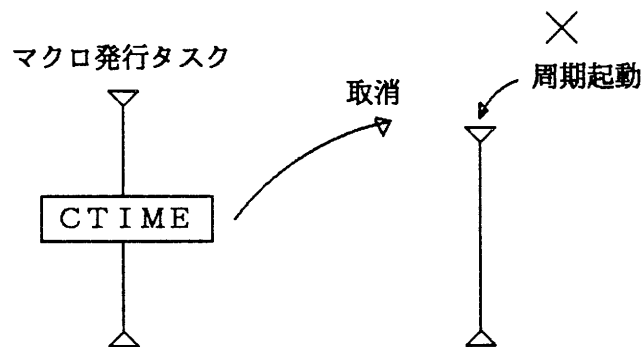


■ CTIME (cancel time)

CTIMEマクロは、TIMERマクロにて、周期起動となったタスクの周期起動要求を取消します。

このマクロを発行後は、周期起動中であったタスクは、起動されなくなります。

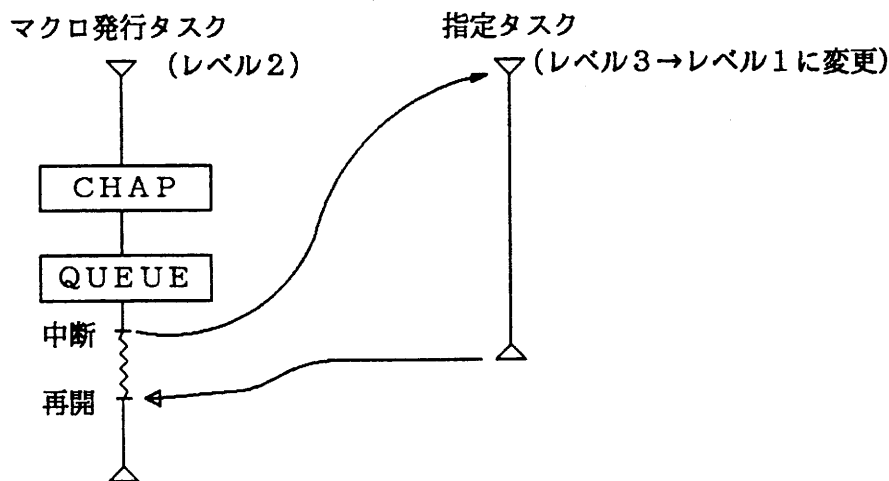
使用例



■ CHAP (change priority level)

CHAPマクロは、指定したタスクの優先レベルを一時的に変更したい場合に使用します。
タスクの優先レベルとは、タスクの実行順序を決定するものであり、2つのタスクに同時に起動要求が発生した場合、レベルの高いタスクを優先して実行します。

使用例



上記の例は、指定タスクを自タスクよりも、先に実行させる為、指定タスクのレベルを自タスクより高くし、QUEUEマクロにより起動したものであり、自タスクは、一時実行を中断し、指定したタスクが先に実行されます。

■ CHMOD (change mode)

CHMODマクロは、ステータスレジスタの内容を変更します。

ステータスレジスタには、コンディションコード（オーバーフロー，ゼロ，ネガティブ等を記憶）、割込みマスク（8レベル）等がありますが、割込みタスクを書換える事により、指定されたマスクレベル以下の割込を禁止させる事が出来ますので、ある処理をハードウェアよりも優先して処理させたい場合に有効です。

ハードウェアの割込には、次のものがあります。

- レベル1：Pコイル
- " ：Sモード終了
- レベル2：タイマー
- レベル3：リモートI/O

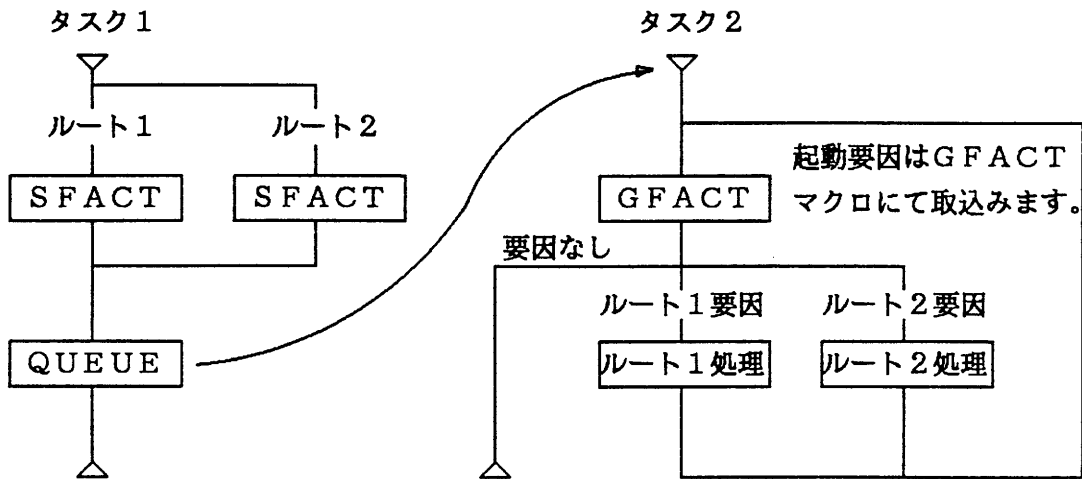
■ SFACT (set factor)

■ GFACT (get factor)

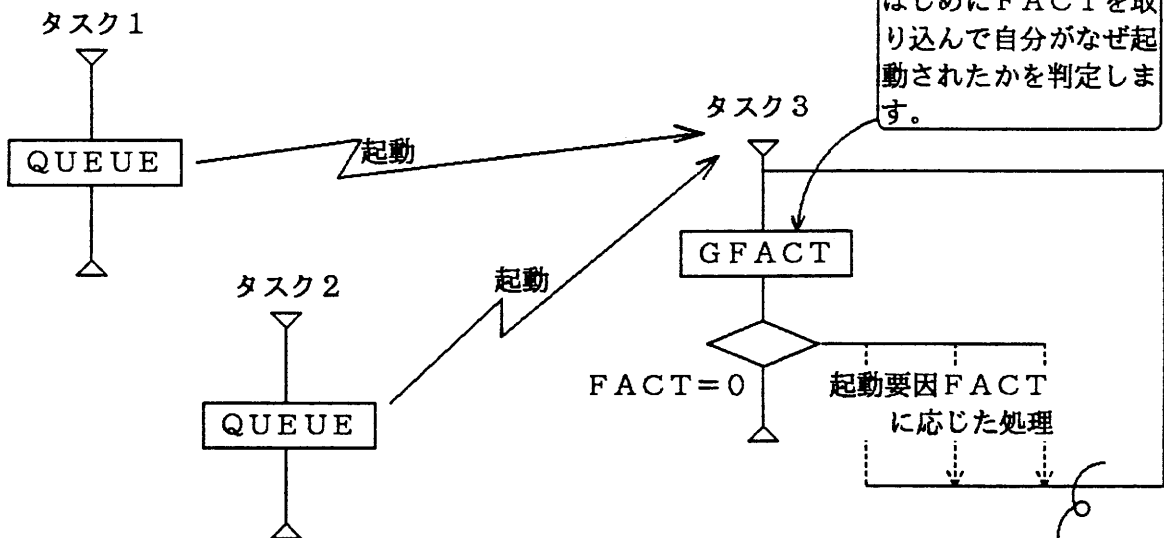
SFACTマクロは、指定したタスクの起動要因をセットし、GFACTマクロは、セットされている起動要因を取込みます。

起動要因とは、あるタスクがなぜ起動されたかを知り、起動要因ごとに違った処理を行なわせるのに有効なものです。

使用例



使用例



タスク3はマクロ命令GFACTにより、自分が誰から起動されたのか(タスク1あるいはタスク2)を知ることができます。即ち、タスク1がタスク3を起動するときの(起動要因)FACTと、タスク2がタスク3を起動するときのFACTを変えておけば、タスク3がFACTを判定して、タスク1, 2いずれから起動されたかを知ることができます。

タスクは終了前にFACTを再チェックしなければなりません。

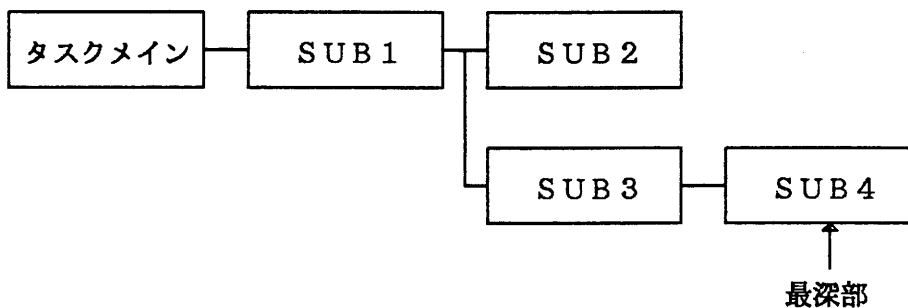
■ USPCHK (user stack pointer check)

USPCHKマクロは、自タスクが指定したスタック容量を超えて、スタックエリアを使用しているか否かをチェックします。

USPCHKマクロを、タスク内の最深部にて、発行する事でスタックエリアを確保する上での目安を得ることが出来ます。

使用例

次の様な、プログラム構成を持つタスクを例に記します。



このタスクは、SUB 4 が最深部となっていますので、SUB 4 にてUSPCHKマクロを発行する事で、スタックの使用状況がわかります。

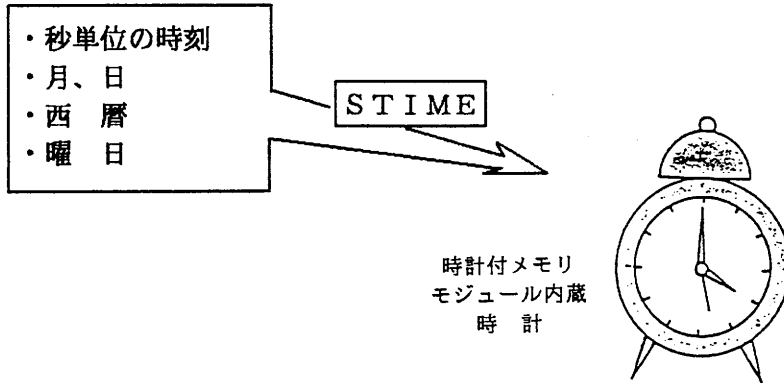
但し、他のルートにて、ローカルエリアを多大に使用している場合はそちらのルートにおける最深部にセットしてください。

- ローカルエリア：スタックエリアに確保される、プログラムのワークエリアです。

■ STIME (set time (時計付メモリが必要です))

STIMEマクロは、時刻を管理している時計付メモリへパラメータで指定された実時刻を設定します。つまり、絶対時刻の設定または更新を行うマクロ命令です。

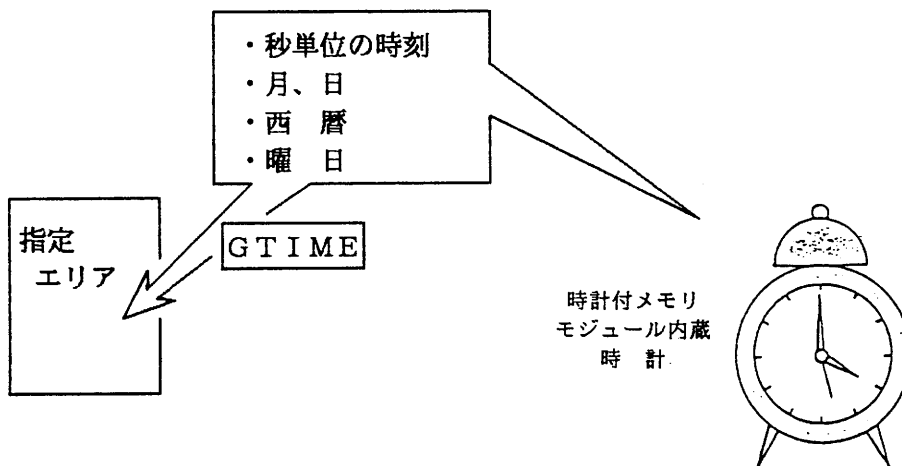
設定できるパラメータには、次のものがあります。



■ GTIME (get time (時計付メモリが必要です))

GTIMEマクロは、時計付メモリで管理している時刻をパラメータで指定したエリアへ格納します。つまり、実時刻を取込むマクロ命令です。

実行結果として下記が格納されます。

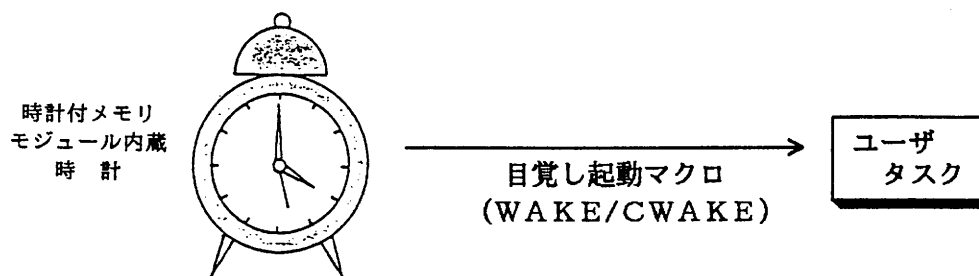


■ WAKE (wakeup task (時計付メモリが必要です))

WAKEマクロは、パラメータで指定された番号のタスクをシステムテーブルへ登録し、scheduled 状態にします。

scheduled 状態となったタスクは、パラメータで指定された時刻に起動されます。

周期起動を指定すると、さらにその後周期時間経過するごとに起動されます。

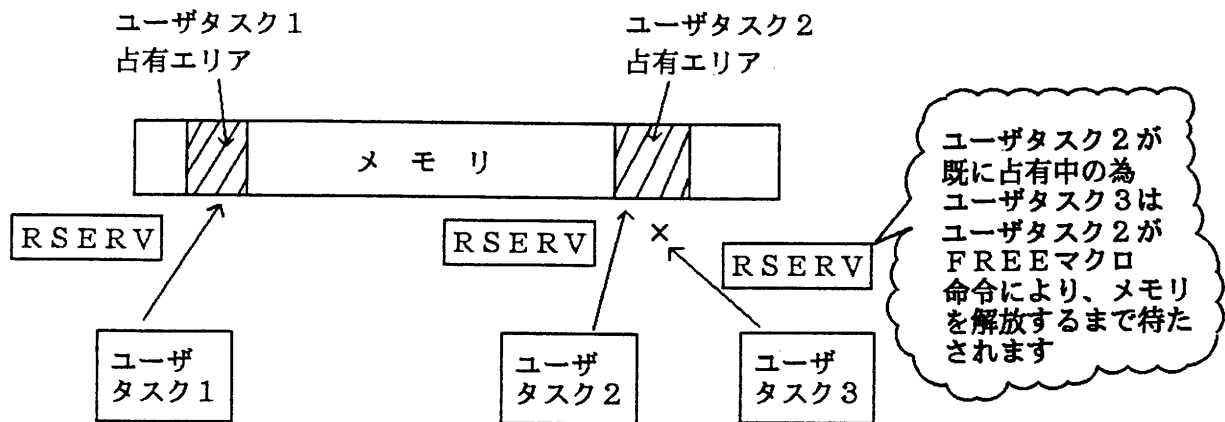


■ CWAKE (cancel wakeup task (時計付メモリが必要です))

CWAKEマクロは、WAKEマクロ命令で登録するシステムテーブルから、パラメータで指定されたタスク番号及び起動要因が一致するタスクをすべて取消します。つまり、WAKEマクロ命令を取消すマクロ命令です。

■ R S E R V (r e s e r v e)

R S E R Vマクロ命令は、パラメータで指定された領域分タスク間共通メモリを占有します。

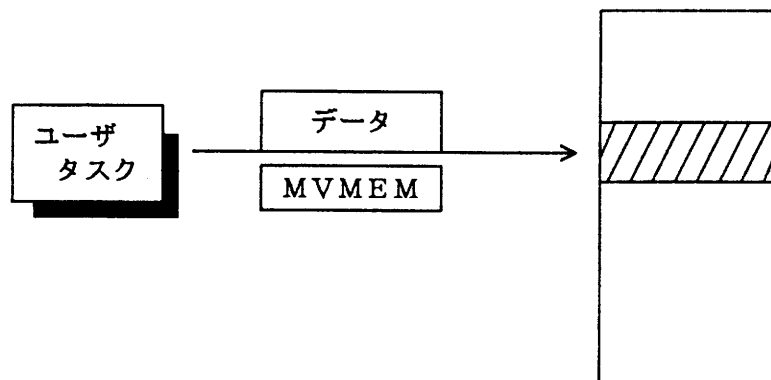


■ F R E E (f r e e)

F R E Eマクロ命令は、R S E R Vマクロ命令により占有しているメモリを解放します。つまり、R S E R V命令を解消すマクロ命令です。

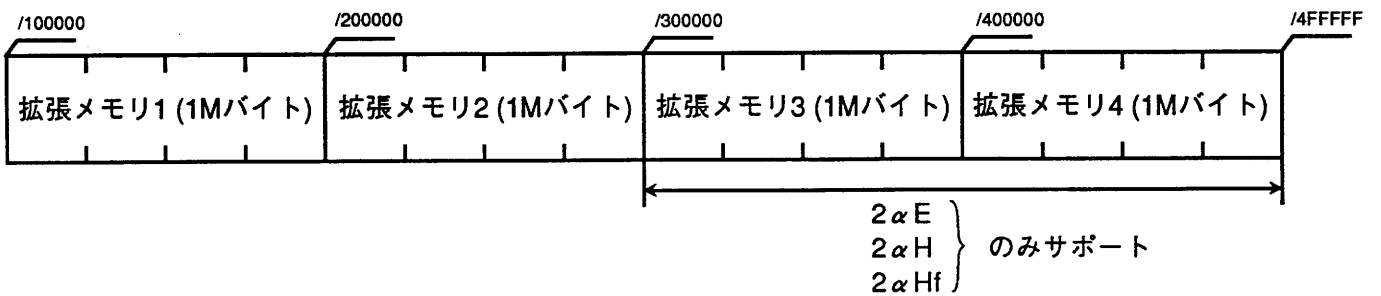
■ MVMEN (move memory)

MVMEMマクロ命令は、パラメータで指定されたデータを指定されたエリアへ転写します。本マクロ命令は、プロテクトメモリ（OSプログラムエリアを除く）へのデータの書込が可能となります。

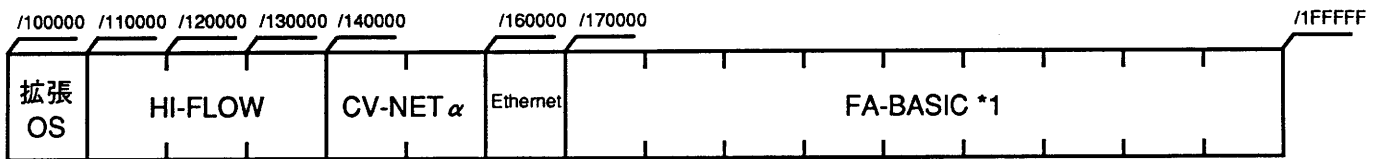


5 付 録

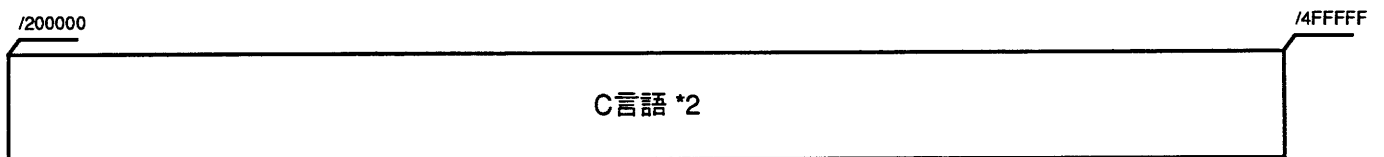
付録1 H-S10/2 α シリーズの拡張メモリ割付



拡張メモリ1



拡張メモリ2~4



*1 FA-BASICは/110000~/1FFFFの範囲内でユーザが自由に割付けできます。

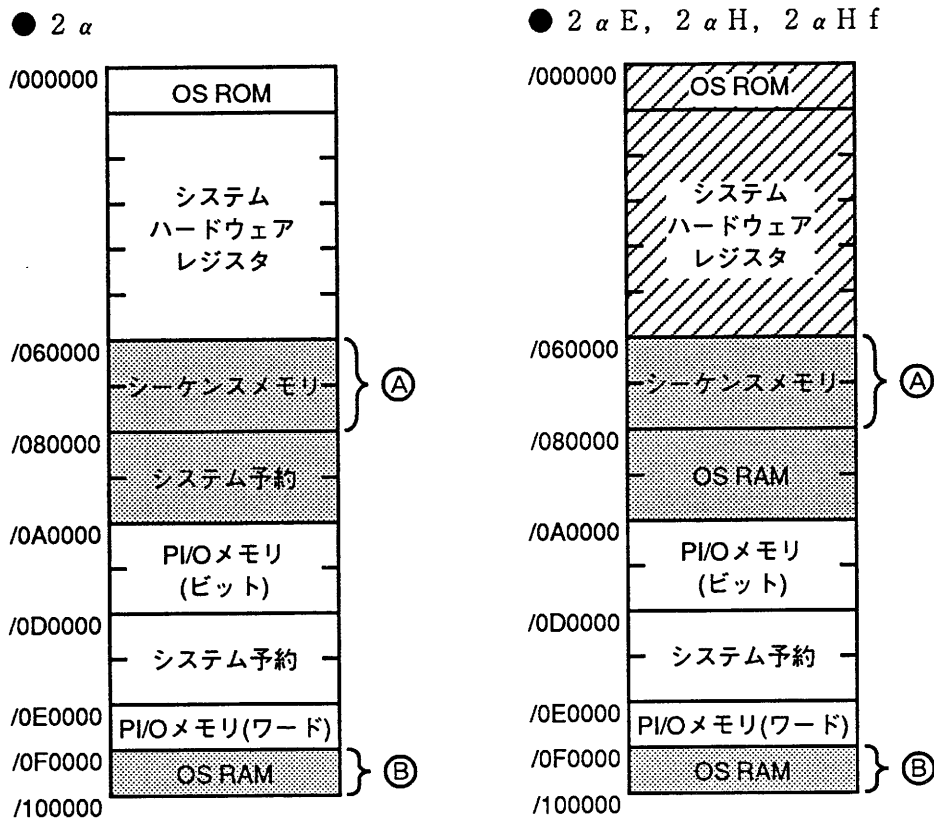
*2 C言語は/110000~/4FFFFの範囲内でユーザが自由に割付けできます。

付録2 プロテクトキースイッチについて

(1) 目 的

ユーザタスク (C言語、F A-BASIC) が誤ってシステムエリアを破壊することを防ぐためのメモリプロテクト機能です。この機能は演算ファンクション (ユーザ演算ファンクション含む) に対しては無効です。

(2) メモリプロテクトの有効範囲



▨ : 読出し/書込み時プロテクト有効

▨ : 書込み時のみプロテクト有効、読出し時はプロテクト無効

(A) : LPET, DWレジスタ, TUC設定値含む

(B) : UFET, PRET, TUC計数值含む

(3) プロテクトエラー (インジケータ表示 "PROT ERR")

プロテクトキースイッチがONのときに、ユーザタスクがプロテクトエリアをアクセスするとプロテクトエラーとなり、そのユーザタスクのみアポートされます。

ユーザタスクがプログラムエディションテーブル (PRET) ではなく、ユーザ演算ファンクション登録テーブル (UFET) に登録されているときは、プロテクトは無効となります。

付録3 DHP情報一覧

No.	機 能	DHPコード (D7.L)			備 考	
		No.	パ'ラメ-タ1	パ'ラメ-タ2		パ'ラメ-タ3
1	QUEUE	01	TN	-	-	TN: タスク番号
2	タスクEXIT	02	ETN	-	-	ETN: 終了タスク番号
3	ABORT	03	TN	-	-	
4	RLEAS	04	TN	-	-	
5	SFACT	05	TN	-	-	
6	GFACT	06	CTNO	-	-	CTNO: マクロ発行タスク番号
7	DELAY	07	CTNO	-	-	
8	TIMER	08	TN	-	-	
9	CTIME	09	TN	-	-	
10	CHMOD	0A	CTNO	-	-	
11	CHAP	0B	LEVEL	-	-	LEVEL: 変更レベル
12	タスク新起動	0C	TN	-	-	
13	タスク再起動	0D	TN	-	-	
14	CPMSイニシャル	0E	00	-	-	
15	アイドル	0F	00	-	-	
16	BCAN	10	TN	-	-	
17	USPCHK	11	CTNO	-	-	
18	STIME	12	CTNO	-	-	
19	GTIME	13	CTNO	-	-	
20	WAKE	14	TN	-	-	
21	CWAKE	15	TN	-	-	
22	RSERV	16	CTNO	-	-	
23	FREE	17	CTNO	-	-	
24	WATE	18	CTNO	-	-	
25	POST	19	CTNO	-	-	
26	DEFCD	1A	CTNO	-	-	
27	ENQ	1B	CTNO	-	-	
28	DEQ	1C	CTNO	-	-	
29	SUSP	1D	TN	-	-	
30	RSUM	1E	TN	-	-	
31	パラメータエラー	1F	CTNO	EC		EC: マクロパラメータエラーコード
32	エラー割込	20	TN	ISW		IN=0はシステムダウン ISW=割込要因
33	拡張ボード割込	21				拡張ボードサポートOSに設定
34	MVMEM	00	CTNO	-	-	
35	コイル割込	22	COIL	CNO		COIL=0: Pコイル割込 1: Nコイル割込 2: カウンタ割込 3: 演算命令割込 CNO=コイル番号
36	タイマ割込	23	K	-	-	K=1: SEQタイマ 2: T.Uタイマ 3: タスクタイマ割込
37	RI/O終了	24	00	-	-	
38	SEND割込	25	CSPN	-	-	CSPN: 終了したSモードプログラム番号

DHPMD=1の場合

DHPMD=2の場合

将来用
予約

付録4 マクロ命令処理時間一覧

この処理時間は、それぞれの条件下の処理（ステップ数）から計算した値です。

No.	項 目	$2\alpha H$ $2\alpha Hf$ (μs)	$2\alpha E$ (μs)	2α (μs)	備 考
1	マクロQUEUEからタスク起動まで	115.6	154.9	442.9	対象タスクはRLEAS状態でマクロ発行タスクよりレベルは高い。
2	マクロ終了からIDLEまで	32.5	43.5	124.4	正常終了かつ2重起動なし
3	マクロABORTから発行タスクに戻るまで	68.6	91.9	262.8	対象タスクはRLEAS状態
4	マクロSFACTから発行タスクに戻るまで	64.8	86.8	248.2	タスクはマクロ発行タスクのみ
5	マクロGFACTから発行タスクに戻るまで	61.4	82.3	235.3	fact=8を取出した場合
6	マクロDELAYからIDLEまで	94.1	126.1	360.5	タスクはマクロ発行タスクのみ
7	タイマ割込からタスク再起動まで	124.2	166.4	475.8	遅延中タスクの再開 割込時はIDLE
8	マクロTIMERから発行タスクに戻るまで	107.8	144.4	412.9	タスクはマクロ発行タスクのみ
9	タイマ割込からタスク再起動まで	119.3	159.8	456.9	割込時はIDLE、スケジューリング中タスクは1、対象タスクはRLEAS、
10	マクロCTIMEから発行タスクへ戻るまで	64.6	86.5	247.3	スケジューリング中タスクは1個
11	マクロCHMODから発行タスクへ戻るまで	52.4	70.2	200.7	タスクはマクロ発行タスクのみ
12	マクロCHAPから発行タスクへ戻るまで	110.2	147.6	422.0	他にタスクなし 対象タスクはRLEAS
13	マクロRLEASから発行タスクへ戻るまで	64.3	86.1	246.2	対象タスクはDormant
14	マクロUSPCHKから発行タスクへ戻るまで	62.3	83.5	238.7	タスクはマクロ発行タスクのみ
15	マクロSTIMEから発行タスクへ戻るまで	137.7	184.5	527.5	タスク再スケジューリングなし
16	マクロGTIMEから発行タスクへ戻るまで	54.8	73.4	209.9	マクロ発行タスクのみ
17	マクロWAKEから発行タスクへ戻るまで	101.3	135.7	388.0	マクロ発行タスクのみ ARBは空 (待タスクなし)
18	マクロCWAKEから発行タスクへ戻るまで	67.6	90.5	258.8	スケジューリング中タスクは1個
19	マクロRSERVから発行タスクへ戻るまで	77.2	103.4	295.6	パラメータは1ケース、RSERV可 他タスクでリソース待タスクなし
20	マクロFREEから発行タスクへ戻るまで	96.7	129.6	370.5	占有ケースは1個、リソース待中タスクなし
21	マクロMVMBENから発行タスクへ戻るまで	68.6	91.9	262.8	WNO=1
22	時刻割込からタスク再起動まで	137.8	184.6	527.8	タスクは時刻起動 (周期ではない。) 他の起動タスクはなし
23	GRからイニシャルタスク起動まで	226636.4	303607.2	868060.0	RTC(IM)あり。パリティエラーなし。ラダー回路なし