

資 料

マイクロコンピュータ用
リアルタイム・モニタプログラム
開 発 報 告 書

昭和 57 年 3 月



財団法人 日本情報処理開発協会

この報告書は、日本自転車振興会から競輪収益の一部である機械工業振興資金の補助を受けて、昭和56年度に実施した「マイクロコンピュータの応用に関する調査研究」の一環としてとりまとめたものであります。

は　じ　め　に

当協会マイクロコンピュータ振興センター（MCC）では、マイコン産業振興の一環として昭和53年度以来マイクロコンピュータ応用システムの高度化、システム開発の効率化などにつながる基礎的、共通的、先導的技術について、システムハウスを中心に委託開発を行うことになり、我が国のマイコン産業の技術力の育成・強化につとめているが、昭和56年度においては次のテーマについて委託開発を行った。〔（ ）内は委託先〕

- ① I/O シミュレータ〔㈱ティー・エス・ディ〕
- ② マイクロコンピュータ用リアルタイム・モニタプログラム〔日本電気ソフトウェア㈱〕
- ③ インテリジェントディスクユニット〔萩原電気㈱〕
- ④ マルチプロセッサ用開発支援システム〔㈱デジタル〕
- ⑤ パケット交換網用汎用端末機〔コンピューター・ネットワーク・サービス㈱〕
- ⑥ リアルタイムFFT演算装置〔㈱エー・ディー・エス〕

本報告書は上記のテーマのうち「マイクロコンピュータ用リアルタイム・モニタプログラム」の開発に関する成果をまとめたものである。

ここに委託開発にあたりご指導・ご協力いただいた関係各位に対し厚くお礼申し上げますとともに、これらの開発システムが広くマイクロコンピュータ応用システムの開発に携わる方々に利用され、我が国のマイクロコンピュータ産業の一層の発展に寄与することができれば幸いである。

昭和57年3月

マイクロコンピュータプロジェクト委員会

(敬称略)

委員長	田村 浩一郎	電子技術総合研究所 制御部論理システム研究室室長
委員	寺田 浩詔	大阪大学工学部 電子工学科 教授
"	福村 晃夫	名古屋大学工学部 情報工学科 教授
"	山上 喜吉	新エネルギー総合開発機構 太陽技術開発室 副主任研究員
"	出口 光一郎	東京大学 工学部 計数工学科 助手
"	前田 英明	マイクロコンピュータシステムコンサルタント
オブザーバ	稻積 義登	通商産業省機械情報産業局 情報処理振興課
"	佐藤 昌彦	通商産業省機械情報産業局 電子政策課

事務局 (財)日本情報処理開発協会 マイクロコンピュータ振興センター

マイクロコンピュータプロジェクト委員会小委員会

(敬称略)

委員	○田村 浩一郎	電子技術総合研究所 制御部論理システム研究室室長
"	○山上 喜吉	新エネルギー総合開発機構 太陽技術開発室 副主任研究員
"	出口 光一郎	東京大学工学部 計数工学科 助手
"	前田 英明	マイクロコンピュータシステムコンサルタント
"	福村 晃夫	名古屋大学工学部 情報工学科 教授
"	吉田 雄二	名古屋大学工学部 電気工学第2学科 助教授
"	今井 正治	豊橋技術科学大学 第4工学系 講師
"	寺田 浩詔	大阪大学工学部 電子工学科 教授
"	島崎 眞昭	京都大学工学部 情報工学科 助教授
"	河田 亨	大阪大学工学部 電子工学科 助手

事務局 (財)日本情報処理開発協会 マイクロコンピュータ振興センター

(注) 本開発においては、上記○印の委員にご担当いただきました。

とりまとめにあたって

本書は、Z-80 マイクロプロセッサ上で動作するリアルタイム・マルチタスク・モニタプログラムの開発概要について説明するとともに、プログラムの使用方法、さらに利用者の対象とするシステムに適するモニタプログラムの作成方法すなわちシステムジェネレーションについて説明しており、取扱い説明書としての性格を合わせて持たせた。

なお、本書を読む場合および実際にシステムジェネレーションを行う場合、下記のマニュアル類を参照いただきたい。

- 各種のZ-80 マイクロプロセッサ説明書
- マイクロコンピュータユーザズマニュアル MCS-85 割り込みコントローラ 8259 など周辺デバイス関係
- MICROSOFT utility software manual
(ユーティリティ・ソフトウェア説明書) プログラム開発用モニタプログラム CP/M 関係

目 次

1. 概 要	1
1.1 目的と特徴	1
1.1.1 開発の目的	1
1.1.2 リアルタイムモニタとは	1
1.1.3 プログラムの特徴	2
1.2 機 能	2
1.3 利用者と適用分野	3
1.4 システムの構成	3
1.4.1 ハードウェア	3
1.4.2 プログラム	4
1.4.3 メモリ	5
2. モニタプログラムの機能	8
2.1 システムの初期化处理	8
2.2 割り込み分析	9
2.3 割り込み処理	11
2.3.1 外部入出力機器	11
2.3.2 インターバル・タイマ	12
2.4 タスクプログラムの管理	14
2.4.1 タスクプログラムの番号	14
2.4.2 タスクプログラムの優先順位	14
2.4.3 タスクプログラムの状態と遷移	14
2.5 マクロ処理	18
3. マクロの説明と使用方法	21
3.1 タスクプログラムの実行管理	21
3.1.1 RQST\$, \$RQST マクロ	21
3.1.2 EXIT\$ マクロ	22
3.1.3 STIM\$, \$STIM マクロ	22
3.1.4 RTIM\$, \$RTIM マクロ	23
3.2 タスクプログラムの同期制御	24
3.2.1 WAIT\$ マクロ	24
3.2.2 POST\$ マクロ	25

3.2.3	ONTM\$ マクロ	26
3.2.4	OFTM\$ マクロ	26
3.3	システム共有資源の管理	27
3.3.1	ATCH\$ マクロ	27
3.3.2	DTCH\$ マクロ	28
3.3.3	GTBF\$ マクロ	28
3.3.4	PTBF\$ マクロ	30
3.4	タスクプログラム間の相互通信	30
3.4.1	RQST\$, \$RQST マクロ	30
3.4.2	RTRV\$ マクロ	30
3.5	外部入出力機器の制御	32
3.5.1	IORQ\$ マクロ	32
4.	システム共通領域の説明	35
4.1	モニタプログラム作業領域	36
4.1.1	実行タスク情報	36
4.1.2	割り込みフラグ	36
4.2	タスクプログラム管理領域	37
4.2.1	タスク管理情報	37
4.2.2	タスク管理作業領域	38
4.3	キュー領域	41
4.3.1	キュー開始アドレス	41
4.3.2	レディ・キュー	42
4.3.3	タイマ・キュー	43
4.3.4	ウェイト・キュー	43
4.3.5	キュー・バッファ	44
4.4	スタック領域	45
4.4.1	モニタ・スタックエリア	45
4.4.2	タスク・スタックエリア	45
4.5	デバイス管理情報	46
4.5.1	割り込み処理入口アドレス	46
4.5.2	デバイス割り込みマスク	46
4.5.3	デバイス管理情報	47
4.6	バッファ管理領域	48
4.6.1	ブロック管理情報	48

4.6.2	ブロック使用情報	49
4.6.3	バッファ・エリヤ	50
5.	システム・ジェネレーションの説明	51
5.1	システム・ジェネレーションの作業と手順	52
5.2	システム共通領域の生成	53
5.3	OWNコーディング	55
5.3.1	システムの初期化处理	56
5.3.2	外部入出力機器割り込み処理	58
5.3.3	最優先割り込み処理	59
5.4	機能の選択と追加	60
5.5	タスクプログラムの作成	63
5.6	汎用モニタプログラムCP/Mの利用	63
付録Ⅰ	名称付与規則	64
付録Ⅱ	マクロ命令とパラメータ	65
付録Ⅲ	モジュールの大きさ	66
付録Ⅳ	システム・ジェネレーションの実例	67
付録Ⅴ	デモプログラムによる操作説明	78

1. 概 要

1.1 目的と特徴

1.1.1 開発の目的

本モニタプログラムは、マイクロコンピュータZ-80上で動作する汎用リアルタイム・モニタプログラムである。

本プログラムは利用者の開発するシステムの外部環境に合わせ、リアルタイム・モニタプログラムとしての機能の拡張、変更が容易にでき、マイクロコンピュータ・システム開発の生産性、信頼性の向上を図ることを目的として作成されたものである。

マイクロコンピュータの多くは各種機器あるいは、システムに組み込まれて利用されているが、これらのマイクロコンピュータ応用システムでは、特殊な外部入出力機器に接続されることが多く、外部入出力機器からの割り込み処理とそれに応じたプログラム実行管理が特に重要となる。

割り込み処理あるいはプログラム実行管理は、すべてのモニタプログラムが持つ共通的な機能である。しかしマイクロコンピュータ・システムでは、開発システム毎に新たにモニタプログラムを開発する傾向にある。これはシステムに接続される入出力機器などの外部環境が特定のものとならないため、その入出力部分の処理が一義的に決定できない理由による。

本プログラムは、リアルタイム・モニタプログラムを基本的な機能で分割したモジュールで構成され、適用するシステムに合わせて、モニタプログラムを特殊化、統合することができる。これにより、マイクロコンピュータ応用システム開発の生産性、信頼性の大幅な向上が期待できる。

1.1.2 リアルタイムモニタとは

8ビットマイクロコンピュータの初期の頃の使われ方は、いくつかの機能を前もって決められた順番に従って実行するだけのもので、例えば一定時間隔でセンサからのデータをサンプリングして、それに必要な処理を加え、結果を制御機器へ出力するだけといった形態であった。

このように、各機能に対応したプログラムを順番に実行させるだけならそれぞれのプログラムの実行を管理する（モニタする）ための特別のシステムは必要としない。

しかし、マイクロコンピュータの応用技術は急速に進んでおり、マイクロコンピュータ組込みシステムに要求される機能は、ますます複雑なものとなり、各機能が複雑に入り組み、かつそれらの機能が実行される順番が前もって決められない処理を実現するため、各機能を処理するプログラムの実行を管理するためのシステム、すなわちモニタプログラムが必要となる。特に、データの発生が変化が割り込みを使ってマイクロコンピュータに通知され、それに基づいて対応するプログラムを直ちに実行させて、結果を返す方式を即時処理（リアルタイムプロセッシング）というが、このようなシステムを制御するためのモニタプログラムをリアルタイムモニタと呼んでいる。

また、CRTディスプレイやシリアルプリンタなどの速度は、マイクロプロセッサの内部処理速度に比べて遅く、このような装置の制御の合い間に、もっと速応性の要求される処理を並行して実

行し、システムの処理効率を上げることが必要なことがある。これを多重処理（マルチプロセッシング）といい、リアルタイムモニタには、この多重処理機能、すなわちマルチタスキングが必要不可欠なものとなっている。

1.1.3 プログラムの特徴

本モニタプログラムの特徴と、利用することで期待できる効果について以下に説明する。

(1) モニタプログラムの汎用性

マイクロコンピュータ応用システムで必要とする高い応答性、多重処理あるいは並行処理機能を持ち、基本的なマクロ命令を備えた汎用的なリアルタイム・モニタプログラムである。

(2) システムへの適応性

計測、制御あるいは通信など幅広い分野を持つマイクロコンピュータ応用システムの個々のシステムに必要なモニタプログラムの最小限度の機能の選択ができ、最小の記憶容量で利用できる。さらに、モニタプログラムでシステムに依存する部分である外部入出力機器に関係する機能あるいは処理を、利用者が作成し追加することで、柔軟でシステムに最適なモニタプログラムを作成できる。

(3) システムの信頼性

タスクプログラム管理などモニタプログラムの主要な機能、基本的なマクロ命令など、モニタプログラムとして本来持つべき機能は提供されているので、利用者が作成した機能以外は、その内容の検査は必要ない。そのために、モニタプログラムの信頼性の向上がはかれる。

(4) システムの生産性

モニタプログラム作成の簡単化と効率化を行なっているために、モニタプログラム作成時の生産性を上げることができる。信頼性のあるモニタプログラムを核として組み上げられるシステム全体の生産性も向上させることができる。

本モニタプログラムで提供するマクロ命令を利用することで、モニタプログラムの種々の機能を利用できタスクプログラムの作成が容易になる。また、本モニタプログラムを使用するシステム間では、タスクプログラムの流用、再利用ができ資源の有効利用も可能となる。

1.2 機能

外部入出力機器で発生する非同期の割り込みと、その割り込みに対応し多重並行処理で動作するタスクプログラムの監視が、リアルタイム・マルチタスク・モニタプログラムの主な役割である。

以下に本モニタプログラムの機能の概要を示す。詳細については後述の2. 機能を参照のこと。

(1) タスクプログラムの管理

タスクプログラムの管理は、モニタプログラムの中心でスケジューラあるいはディスパッチャーともいわれ、外部入出力機器からの割り込みやマクロ命令による要求に応じて利用者の定義したタスクプログラムの実行管理、同期制御などをタスクプログラムの優先順位に基づいて行う。

(2) マクロサービス

タスクプログラムの起動、実行または終了などの実行管理、同期制御、タスクプログラム間のデータ交換、さらに、外部入出力機器などの共有資源の管理などモニタプログラムの機能を利用者に提供している。

(3) 割り込みの管理

システムに接続している外部入出力機器とインターバルタイマの割り込み、あるいは最優先（マスク不能）の割り込みを分析、判定し各割り込み処理へ制御をわたす。

外部入出力機器との実際のデータ入出力は、利用者の作成したタスクプログラムで行なわれる。

(4) インターバルタイマの管理

システムに内蔵されているインターバルタイマを使用して、周期あるいは遅延時間起動タスクプログラムの管理、また、外部入出力機器と入出力動作中の時間監視を行う。

(5) 外部入出力機器の制御

システムに接続する外部入出力機器の使用状態の管理、および割り込みマスクの設定、解除など機器の制御を行う。

1.3 利用者と適用分野

(1) 利用者

利用者は本書に記述された内容に従って作業を行うことで、容易に利用者のシステムに合うモニタプログラムを作成できるが、モニタプログラムの作成の作業にはZ-80アセンブラ言語、また、モニタプログラム一般の知識を必要とする。そのため本モニタプログラムの作成、また、利用者として一応システム設計従事者を対象としている。

(2) 適用分野

本モニタプログラムの適用分野として、マイクロプロセッサZ-80を使用する計測、制御、あるいは通信などの各種システムが考えられる。本モニタプログラムは、それら各種システムの利用者プログラム（タスクプログラム）を管理し制御する。

1.4 システムの構成

1.4.1 ハードウェア

モニタプログラムを動作させるための中央処理装置と、中央処理装置に接続するデータ入出力、あるいは、制御機器など、モニタプログラムを使用した開発対象システムでのハードウェア構成例を図1-1に示す。

図1-1には、開発支援システムとしてのハードウェア構成も含めてある。これはモニタプログラムの作成で、プログラム開発用汎用モニタプログラムとCP/M(Control Program for Microcomputers)を使用するとき最低限必要とするコマンド等の入出力装置である。

上記の他に、中央処理装置にはプログラマブルタイマーが、中央処理装置と外部入出力機器との間には、プログラマブル割り込みコントローラーが接続される。

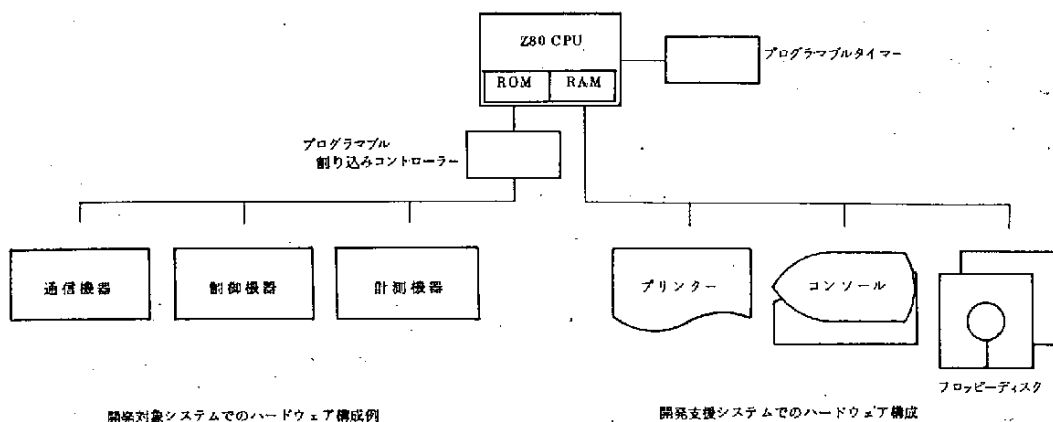


図 1-1 ハードウェアの構成例

1.4.2 プログラム

モニタプログラム全体の構成を図 1-2 のようにする。この構成はモニタプログラムをその代表的な機能に基づいて分割したものである。以後本書でのモニタプログラムの説明は図 1-2 で示した構成に従って進める。

モニタプログラムを構成する各機能については、後述の 2. 機能で詳細に説明する。ここでは、モニタプログラムを構成する各機能の概要について説明する。

(1) 初期化処理

リセット、電源投入の割り込みで本モニタプログラムが最初に行う処理である。実際の処理としてはモニタプログラムで使用する領域、外部入出力機器、割り込みコントローラ、あるいはタイマなどの初期設定を行う。

(2) 割り込み分析

システムに接続している外部入出力機器、あるいは、タイマーからの割り込みを解析し各割り込み処理へ分岐させる。

(3) 割り込み処理

外部入出力機器からの割り込み処理、実際には割り込みマスク開閉の管理、さらに、利用者の作成した外部入出力機器と実際にデータ入出力を行うタスクプログラムの起動準備を行う。

(4) タスク管理

タスクプログラム多重並行処理の管理を行う。具体的な処理としては外部入出力機器、あるいは、マクロ命令を介してのタスクプログラムからの要求に応じて、レディ状態にあるタスクプログラムから、一番実行優先度の高いタスクプログラムを探し、そのタスクプログラムに実行権を与える。

(5) マクロ処理

タスクプログラムの実行管理、同期制御、タスクプログラム間のデータ交換、さらに、システムの共有資源の管理などモニタプログラムの機能をマクロとして利用者に提供している。

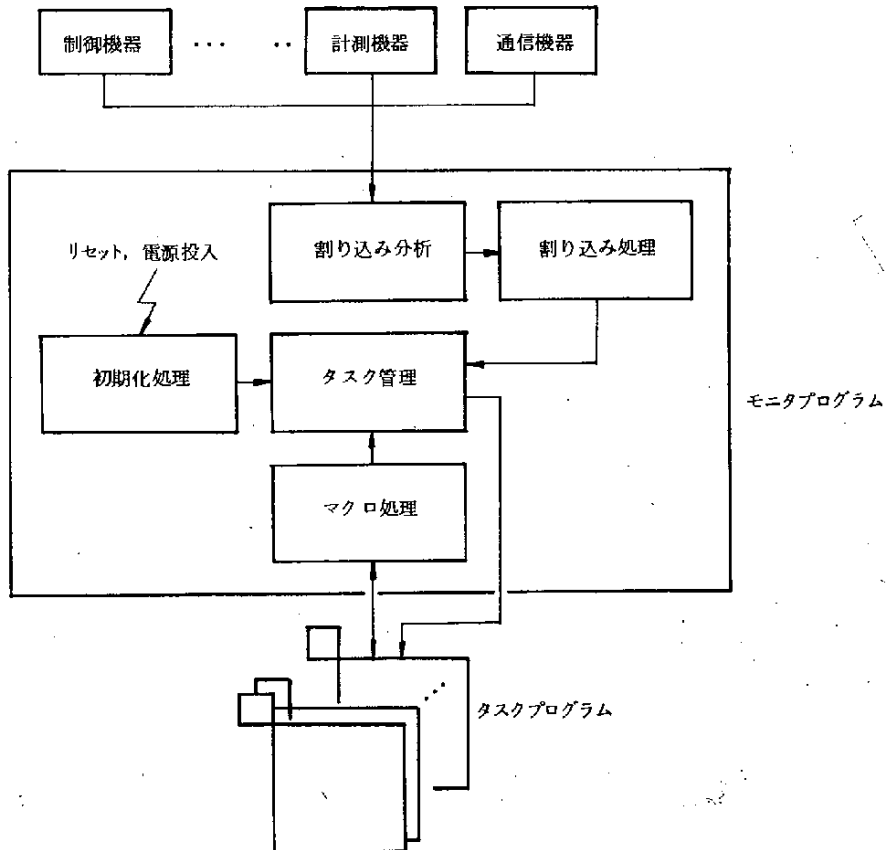


図 1-2 モニタプログラムの構成

1.4.3 メモリ

(1) 容 量

本モニタプログラムに必要な主記憶域の容量の概算値を表 1-1 に示す。表 1-1 に示した容量で処理部分はモニタプログラムのマクロなどすべての処理を含めたときの値であり、テーブルなどに使用される領域は、モニタプログラムで管理するタスクプログラムが使用する領域の概算値をも

とに計算している。

実際には、モニタプログラムに組み込むマクロなどの機能、あるいは、タスクプログラムのスタック領域など確保する領域の大きさ、さらにタスクプログラム数など、いろいろな条件で必要なメモリ容量は変化する。

表 1-1 必要メモリ容量

内 容	種別	容 量
処 理 部 分	ROM	約 2.0 K バイト
領 域 (テーブルなど)	ROM	約 $12 * N + 30$ バイト
	RAM	約 $60 * N + 120$ バイト

注意：Nはタスクプログラムの個数

(2) 配 置

モニタプログラムの主記憶域上での配置は、利用者が任意に指定可能である。ただし、以下の3点について注意が必要である。

① 割り込みベクタアドレス

外部入出力機器からの割り込みの入る位置は利用者が任意に指定できるが、その位置をモニタプログラムの所定の位置にする必要がある。

② 最優先（マスク不能）割り込みアドレス

最優先割り込みの位置は 66H と固定となっているために、その位置をモニタプログラムの所定の位置に指定するか、最優先割り込みのときモニタプログラムの最優先割り込み処理を起動できるようにする必要がある。

③ 汎用モニタプログラム CP/M との重複

プログラム開発用モニタプログラム CP/M を使用して、モニタプログラムの作成あるいは検査を行うとき、CP/M と CP/M が使用する領域とモニタプログラムが重複しないようにする必要がある。

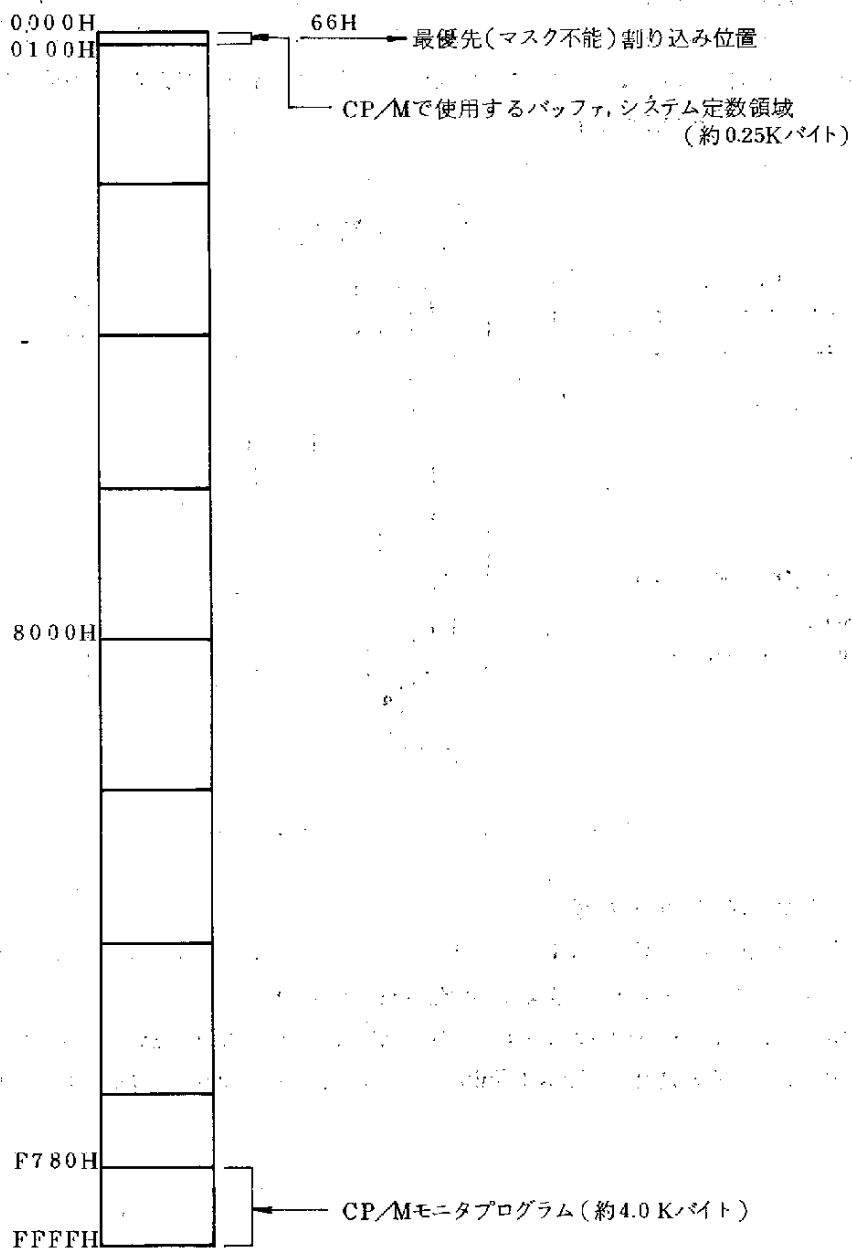


図 1-3 モニタプログラムの配置

2. モニタプログラムの機能

ここでは、図1-2で示したモニタプログラムを構成する各機能について詳細に説明する。モニタプログラムとしての一般的な機能を説明する形で、本モニタプログラムの持つ機能について説明する。

個々の機能を説明する前に、モニタプログラム全体と各機能間の流れを明瞭にする意味で図2-1にモニタプログラムの制御の流れを示す。

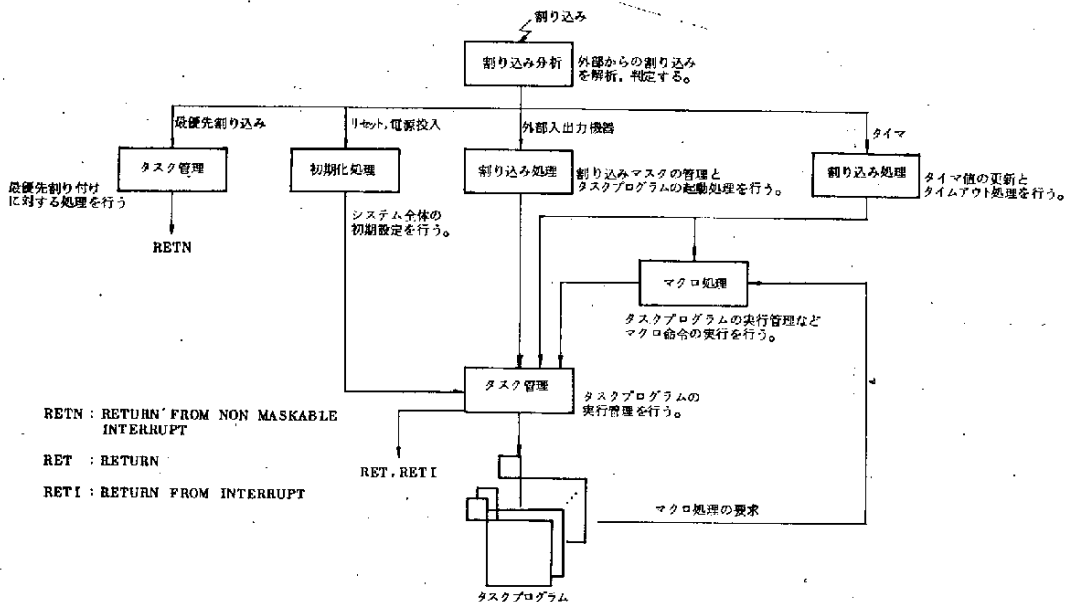


図2-1 モニタプログラムの制御の流れ

2.1 システムの初期化処理

中央処理装置に対する電源投入あるいはリセットにより、本モニタプログラムに起動がかかり、スタート・アップすなわちシステム全体の初期化処理が実行される。

初期化処理は、モニタプログラムに固有な処理と利用者システムに固有な処理に分けられる。モニタプログラムに固有な処理は、利用者に提供されるモニタプログラムの中に既に組み込まれているが、

利用者のシステムに固有な処理は、モニタプログラム作成時に利用者が追加する必要がある。

以下に初期化処理の内容を示す。

(1) モニタプログラム固有の処理

① テーブルなどの初期設定

- ・ モニタプログラムのスタックポインタを初期化する。
- ・ 実行中のタスクプログラム無しの状態（アイドル）とする。
- ・ レディ、タイマ、ウェイト各キュー領域を初期化する。

(2) 利用者システム固有の処理

① Z-80 中央処理装置（CPU）の初期設定

CPUの割り込みモードを設定する。

② デバイスの初期設定

- ・ プログラマブル割り込みコントローラを初期化する。
- ・ プログラマブル・インターバル・タイマを初期化する。

③ 外部入出力機器の初期設定

プログラマブル・コミュニケーション・インタフェース、すなわち，USART（Universe Synchronous Asynchronous Receiver Transmitter）の初期化を行う。

④ その他

システム共通領域とRAM領域の初期化，さらに，周期時間起動タスクプログラムの起動準備などを行う。

上記システム初期設定終了後，処理はタスクプログラム管理へ移る。

2.2 割り込み分析

マイクロプロセッサと外部入出力機器との関係には，割り込み機能が重要な働きをする。入出力データの転送，計時機能，さらに，マイクロプロセッサ外部の異常事態発生に対しても割り込みは有効である。割り込みは，割り込み要因に応じた処理へ制御をわたす機能である。

Z-80 中央処理装置では，割り込みモードの指定で最大128レベルの独立した番地指定割り込みを処理できる。

本モニタプログラムは，プログラマブル割り込みコントローラ8259 使用前提に設計，利用者に提供している。8259は優先順位の設定，割り込みマスクあるいはベクタアドレス，すべてを利用者が指定でき最高64レベルまでの割り込みを制御できる。

以下に割り込み分析処理の内容について説明する。

(1) タスクプログラムの情報退避

- ① 実行途中のタスクプログラムで使用中のレジスタの内容，スタックポインタを退避する。
- ② モニタプログラムのスタックポインタを初期化する。

(2) 割り込みの分析

① 割り込み要因を分析，判定し各割り込みに対応する処理へ制御をわたす。

表 2 - 1 割り込みの種類

種 類	内 容	処 理
リセット割り込み	リセットあるいは電源投入による割り込み。	モニタプログラムの起動，システム全体の初期化処理実行に入る。
外部入出力機器割り込み	システムに接続されている入出力機器からの制御あるいはデータ転送を伴う入出力割り込み。	利用者の作成した各入出力機器割り込み処理に制御がわたる。
タイマ割り込み	プログラマブル・タイマからの周期的な割り込み	モニタプログラムのタイマ割り込み処理へ制御がわたる。
最優先割り込み (マスク不能割り込み)	電源切断などシステム異常状態発生時の割り込みで一番優先度が高い。	0066H番地のマスク不能割り込み処理へ制御がわたる。

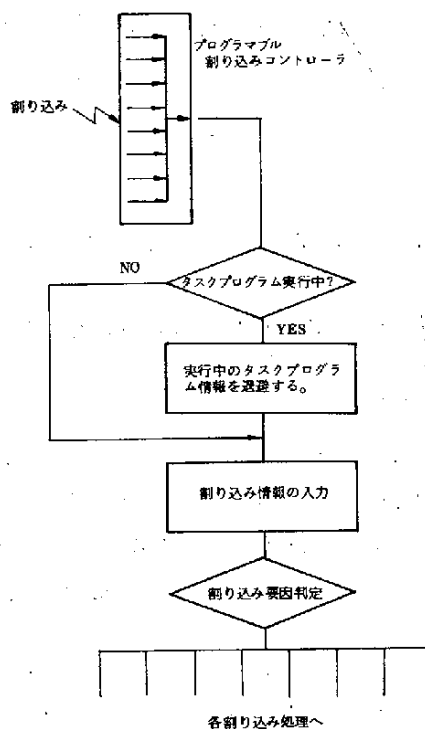


図 2 - 2 割り込み分析

2.3 割り込み処理

リセット割り込み、あるいは最優先割り込みに対しては、各々モニタプログラムの機能の一部で利用者定義可能である初期化处理、あるいは、最優先割り込み処理を直接実行する。

ここでは、利用者がシステムに接続する外部入出力機器とインターバル・タイマの割り込み処理について説明する。

外部入出力機器の割り込みは、同時に入出力データの転送を伴うことが多い。外部入出力機器の割り込み処理は、先に説明した割り込み分析処理と実際にデータ入出力を行う利用者作成のタスクプログラムとの間の連絡の中継ぎを行うだけである。

インターバル・タイマの割り込み処理は、本モニタプログラムに機能の一部として含まれ、その機能は利用者に提供されている。

2.3.1 外部入出力機器

システムに接続している外部入出力機器からの割り込みに対する処理を下に示す。

(1) 外部入出力機器の制御

① 外部入出力機器の割り込みマスクを閉じる。

(2) 外部機器入出力処理タスクプログラムの起動処理

① 外部入出力機器と入出力処理タスクプログラムの入出力待ち情報を解除する。

② デバイス・タイマ・カウンタを0とする。

③ 外部入出力機器を専有しているタスクプログラム、すなわち、入出力処理タスクプログラムをタイマキューから外しレディキューへ登録する。

④ タスク管理へ制御をわたす

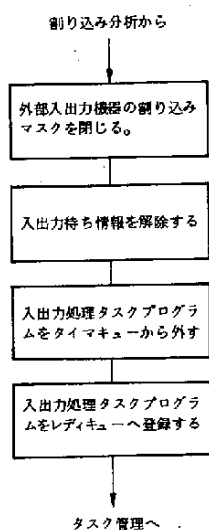


図 2-3 外部入出力機器からの割り込み処理

2.3.2 インターバル・タイマ

インターバル・タイマの割り込みに対する処理を下に示す。

(1) インターバル・タイマの制御

① タイマの割り込みマスクを閉じる。

(2) タイマキューに登録されているタスクプログラムのタイマ値更新

周期タイマ、ウェイトタイマ、デバイスタイマ各カウントエリアの値が0でないときに、以下の処理を行う。タイムアウトとなったときは該当タスクプログラムをレディキューに登録する。

① 周期タイマ値の更新

② ウェイトタイマ値の更新

タイムアウトになったときで、周期時間起動タスクプログラムとなっていないときはタイマキューから外し、タイマウェイトを解除する。

③ デバイスタイマ値の更新

タイムアウトになったときで、周期時間起動タスクプログラムとなっていないときはタイマキューから外し、入出力完了待ちを解除する。

また外部入出力機器の割り込みマスクを閉じる。

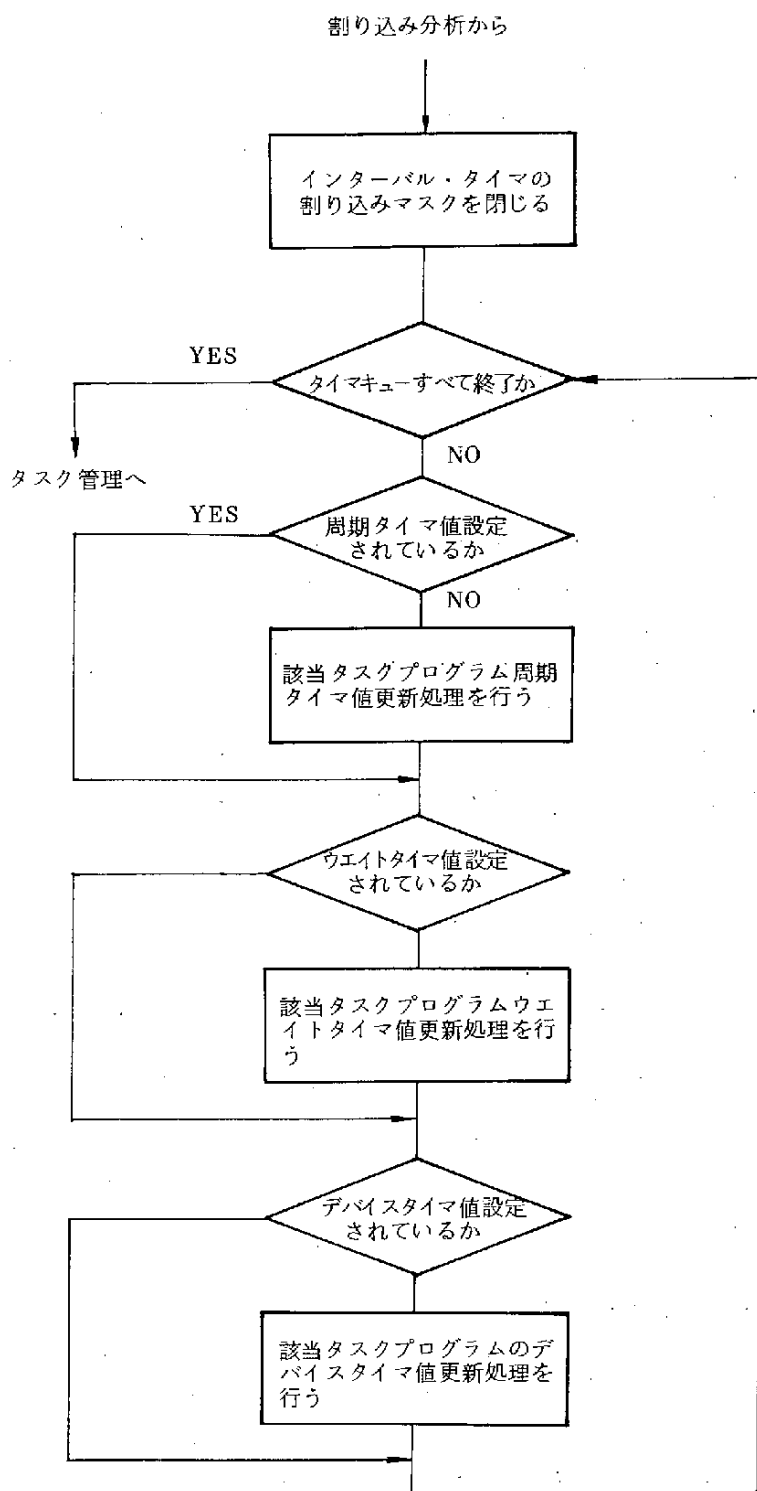


図 2-4 インターバルタイマからの割り込み処理

2.4 タスクプログラムの管理

タスクプログラムとは、利用者が作成するプログラムで、モニタプログラムが起動および実行する制御上の単位である。

タスクプログラム管理（スケジューラあるいはディスパッチャーとも言われる）は、外部入出力機器あるいはインターバル・タイマから発生する非同期の割り込みを対処する。多重並行処理、また、マクロ命令によるタスクプログラムの同期処理など、タスクプログラム起動および実行の制御（スケジューリング）を行う。

タスクプログラムの具体的な処理としては、READY（実行可能）状態にあるタスクプログラムのなかで、もっとも実行優先度の高いタスクプログラムに実行権を与える。

2.4.1 タスクプログラムの番号

本モニタプログラムでは、利用者の作成したタスクプログラムの識別、および実行の優先度の順位づけのためにタスクプログラムに番号（タスク番号）を付与する。モニタプログラムでは、タスクプログラムの管理あるいは制御を、すべてこのタスク番号をもとに行う。

タスク番号としては、タスク管理情報@TCBC内にタスクプログラムの固定情報を定義するために、確保した領域の順番が割り付けられる。

最小タスク番号は0かつ最大タスク番号は127で、最大128個までのタスクプログラムを利用者は定義できる。また、タスク番号は0で始まる若番順の値となる。

2.4.2 タスクプログラムの優先順位

本モニタプログラムでは、タスク番号で実行の優先順位づけを行うことができ、若いタスク番号を持つタスクプログラムの方が優先順位が高くなっている。そのため利用者は緊急度あるいは重要度の高いタスクプログラムには若いタスク番号を割り付けることで、実行の優先度を高くすることができる。

いま、あるタスクプログラムの実行中に、それよりも優先順位が高いタスクプログラム実行要求があった場合、モニタプログラムは実行中のタスクプログラムを一時中断し、優先順位の高いタスクプログラムを実行させる。その処理が終了すると、中断されていたタスクプログラムを中断したところから再開する。

2.4.3 タスクプログラムの状態と遷移

タスクプログラムには、図2-5に示す4種類の状態がある。各状態は外部入出力機器からの割り込み、あるいは、マクロ命令実行による事象の発生で他の状態に移る。モニタプログラムに管理されているタスクプログラムは必ずこの中のいずれかの状態に存在する。

モニタプログラムは、実行可能状態（READY）に存在するタスクプログラムの中で、最も優先順位の高いタスクプログラムに実行権をわたす。

(1) 静止状態 (DEAD)

タスクプログラムは、以下の要因で起動される。

① 外部入出力機器からの割り込み

② モニタプログラム、あるいはタスクプログラムからのマクロ命令の実行

RQST\$, \$RQST, STIM\$, \$STIM, DOST\$, ONTM\$ 各マクロ

静止状態とは、これらの要因の発生がまったくない状態である。また、実行状態のタスクプログラムが処理を終了しモニタプログラムに、その旨の通知 (EXIT\$ マクロ) を行くと、そのタスクプログラムは静止状態となる。

(2) 実行可能状態 (READY)

実行可能状態に存在するタスクプログラムとは、実行準備は整っているが、そのタスクプログラムより優先順位の高いタスクプログラムが実行中であるため、中央処理装置 (CPU) を使用できない状態にあるタスクプログラムである。

実行可能状態に存在するタスクプログラムは、レディキュー @ RQUE に優先順位の高い順に登録されている。

(3) 実行状態 (RUN)

実行状態に存在するタスクプログラムとは、モニタプログラムから実行権を与えられ、中央処理装置 (CPU) を専有しているタスクプログラムである。

実行状態に存在するタスクプログラムは 1 個しかない。

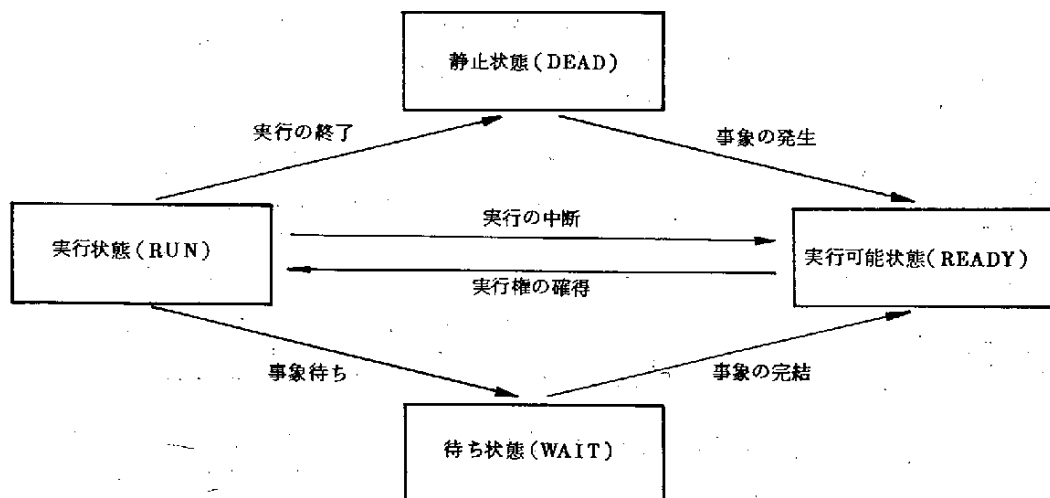


図 2-5 タスクプログラムの状態と遷移

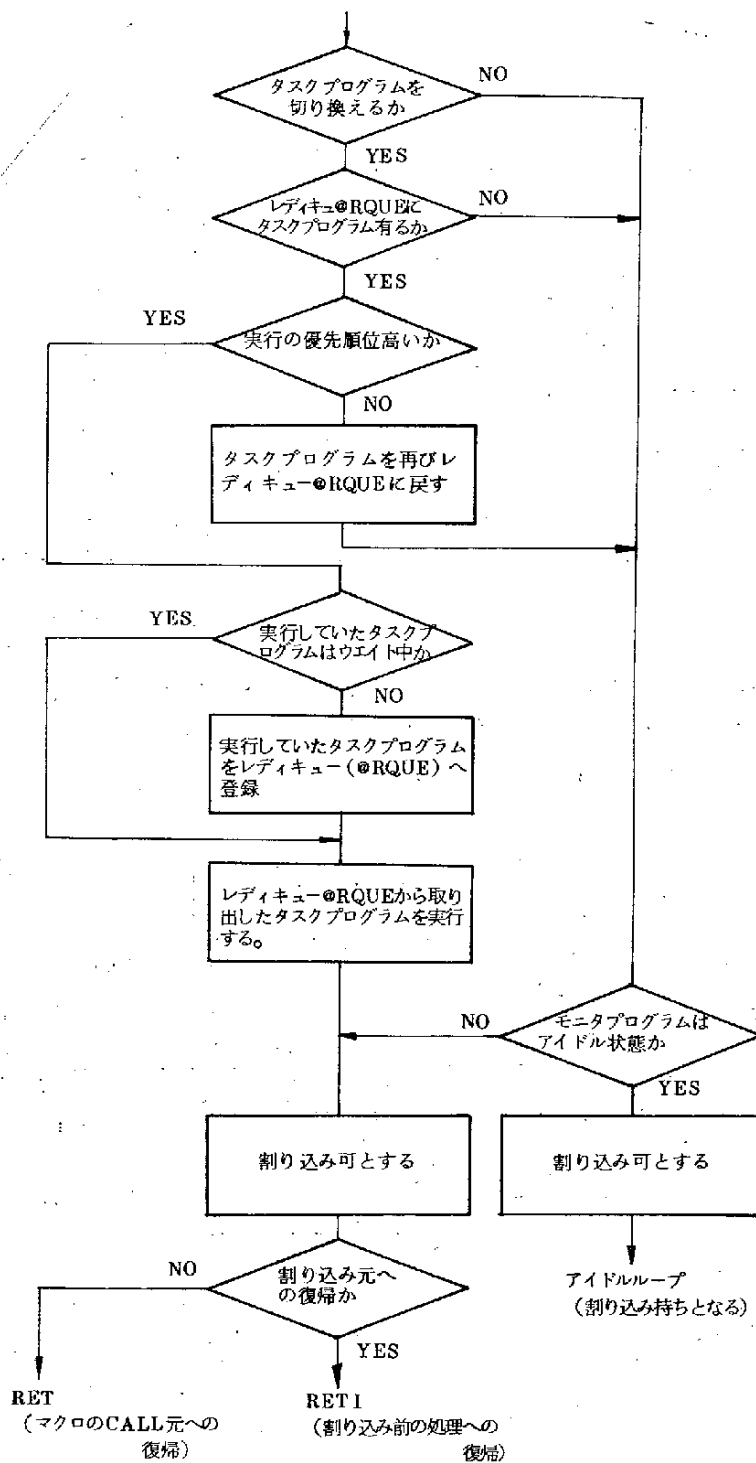


図 2 - 6 タスクプログラムの管理

(4) 待ち状態 (WAIT)

待ち状態に存在するタスクプログラムとは、実行可能状態のタスクプログラムで外部入出力機器、あるいは、マクロ命令の実行など何らかの事象の発生、あるいは、完了を待って、一時実行権を放棄しているタスクプログラムをいう。

待ち状態に存在するタスクプログラムは、タイマキュー@TQUE、あるいは、ウェイトキュー@WQUEに一時登録され、事象が発生あるいは完了することでキューから外されレディ・キュー@RQUEへ登録される。

① 外部入出力機器の入出力動作完了待ち

IORQ\$ マクロ

② 周期あるいは遅延時間の経過待ち

STIM\$, ONTM\$ マクロ

③ タスクプログラム間の同期制御

WAIT\$ マクロ

④ システム共有資源 (リソース) の管理

ATCH\$ マクロ

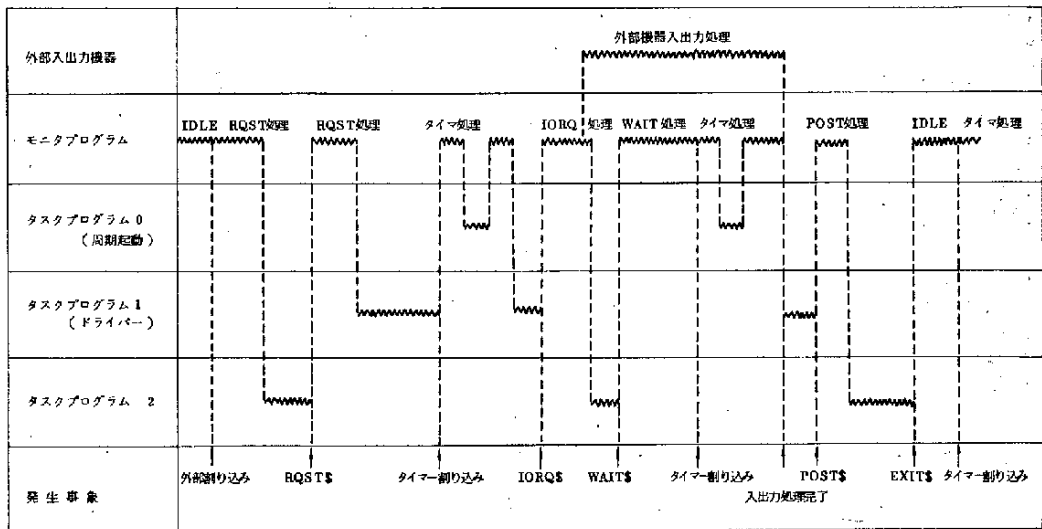


図 2-7 タスクプログラムの多重並行処理の例

2.5 マクロ処理

モニタプログラムの管理下で走行する利用者作成のタスクプログラムは、モニタプログラムで提供している機能を利用することで目的としている機能を実現できる。モニタプログラムで備えている機能はモニタプログラム、また、タスクプログラムからのマクロ命令の発行で利用できる。

タスクプログラムからモニタプログラムへの機能の使用要求は、すべてマクロ命令を通して行う。マクロはモニタマクロとタスクマクロの2種類に分けられる。モニタマクロはモニタプログラム内で、タスクマクロはタスクプログラム内で使用する。モニタマクロ名は先頭1文字が、タスクマクロ名は最後の1文字が記号“\$”となっている。

以降、機能別に分類したマクロを表2-2～表2-6に示す。

(1) タスクプログラムの実行管理

モニタプログラム、あるいは、タスクプログラムから他のタスクプログラムの起動と実行、周期時間間隔でのタスクプログラムの起動と実行を行う。また、タスクプログラムの実行権の放棄およびタスクプログラムの周期時間起動を解除する。

表2-2 タスクプログラム実行管理マクロ

マクロ名	機 能 概 要
RQST\$, \$RQST	タスクプログラムを実行可能状態 (READY) にする。 優先順位が高ければ即座に実行に入る。 優先順はタスクプログラム番号の若番順である。
EXIT\$	タスクプログラムを静止状態 (DEAD) にする。
STIM\$, \$STIM	インターバルタイマを利用し指定した時間間隔でタスクプログラムを周期起動する。
RTIM\$, \$RTIM	STIM\$, \$STIMマクロの処理を解除する。

(2) タスクプログラムの同期制御

他のタスクプログラムが処理終了するまで処理を開始できないなど、2つのタスクプログラム間で処理の都合上、待ち合わせが必要なとき、また、指定時間経過待ちなど事象待ちとなるとき、自らの実行権を放棄して他のタスクプログラムに実行権を譲る。処理が実行可能となったとき、あるいは、待ち事象が完了したときに再び実行権放棄中のタスクプログラムに制御を返す。

表 2-3 タスクプログラムの同期制御マクロ

マクロ名	機 能 概 要
WAIT\$	タスクプログラムの実行権を放棄して待ち状態とする。
POST\$	WAIT\$ マクロの処理を解除する。
ONTM\$	タスクプログラムを指定した時間だけ実行権を放棄し待ち状態とする。
OFTM\$	ONTM\$ マクロの処理を解除する。

(3) システム共有資源の管理

システムが保有している資源（たとえば外部入出力機器、バッファなど）を、複数のタスクプログラムで同時に使用できないように資源を管理する。ある資源が他のタスクプログラムに占有され使用中のとき、その資源の占有要求を出したタスクプログラムを占有解除まで待ち状態にするか、使用不可能ということで起動元のタスクプログラムへ通知する。

表 2-4 システム共有資源管理マクロ

マクロ名	機 能 概 要
ATCH\$	システム共有資源の占有を要求する。
DTCH\$	ATCH\$ マクロで占有したシステム共有資源を解放する。
GTBF\$	動的バッファ領域を確保する。
PTBF\$	GTBF\$ マクロで確保してある動的バッファ領域を解放する。

(4) タスクプログラム間の相互通信

相互に連絡しシステムを構成しているタスクプログラム間では、互いにデータを交換する必要がある。本モニタプログラムでは、タスクプログラム間で簡単にデータ交換が行なえる機構を備えている。データの交換はキューバッファ@QBUFを介して行なわれ、送信ではキューバッファにデータが格納され受信ではキューバッファからデータが取り出される。

表 2-5 タスクプログラム間相互通信マクロ

マクロ名	機能概要
RQST\$, \$RQST	タスクプログラムにデータ(パラメータ)をわたす。
RTRV\$	RQST\$, \$RQST マクロでわたされたデータ(パラメータ)を受けとる。

(5) 外部入出力機器の制御

外部入出力機器の動作は、中央処理装置(CPU)の処理時間に較べて非常に遅い。この入出処理時間に中央処理装置を有効に活用するために、入出力処理要求を出したタスクプログラムの実行権を放棄し他のタスクプログラムに実行権を譲る。入出力処理が完了した時点、再び実行権を放棄し待ち状態となっているタスクプログラムに制御を返す。

表 2-6 外部入出力機器制御マクロ

マクロ名	機能概要
IORQ\$	タスクプログラムが命令した外部入出力機器との入出力処理が完了するまで実行権を放棄して待ち状態とする。

3. マクロの説明と使用方法

ここでは本モニタプログラムで提供するマクロの機能の説明と使用方法について説明する。

マクロと受けわたしを行うパラメータは、すべてレジスタ経由で行う。また、マクロをタスクプログラム内で使用するときマクロを起動する前にDI命令で割り込み不可とする。マクロの実行が終了し、起動元へ戻ってきたときは、すでにEI命令で割り込み可の状態になっている。

3.1 タスクプログラムの実行管理

3.1.1 RQST\$, \$RQST マクロ

(1) 機能

タスクプログラムを起動し、要求先タスクプログラム実行に必要なパラメータをわたすことができる。わたされたパラメータは、要求先タスクプログラムのキューバッファ@QUEBに一時格納され、RTRV\$マクロで取り出される。

(2) 起動方法

① 形式

```
DI  
CALL RQST$
```

タスクマクロ

あるいは

```
CALL $RQST
```

モニタマクロ

② パラメータ

C Reg : 起動するタスクプログラムの番号を設定する。

CE Reg : 要求先タスクプログラムへわたすパラメータが格納されている領域の先頭アドレスを設定する。

(3) 注意事項

① わたすパラメータの大きさ(バイト数)は要求先タスクプログラムで定められ固定である。パラメータの大きさはモニタプログラム作成時にタスク管理情報@TCBCに設定する。

② 本マクロ実行時に要求元タスクプログラムの領域から要求先タスクプログラムのキューバッファ@QUEB内にパラメータの転送が行なわれる。

パラメータの受けわたしについてはRTRV\$マクロの注意事項も参照のこと。

③ タスクプログラムの番号を T_1 と T_2 とし、タスクプログラム T_1 からタスクプログラム T_2 を本マクロで起動要求したとき、実行タスクプログラムは2つのタスク番号の違いにより次のよう

になる。

- $T_1 < T_2$ (優先順位は T_1 の方が高い) のとき
タスクプログラム T_1 がそのまま実行を継続し、タスクプログラム T_2 は、レディキュー@
RQUE に登録され実行待ち状態となる。
- $T_1 > T_2$ (優先順位は T_2 の方が高い) のとき
タスクプログラム T_2 が先に実行され、タスクプログラム T_1 は実行中断となりレディキュー
@RQUE に登録される。

3.1.2 EXIT\$マクロ

(1) 機能

タスクプログラムの実行を終了させる。

(2) 起動方法

① 形式

```
DI  
CALL EXIT$
```

② パラメータ

なし

(3) 注意事項

なし

3.1.3 STIM\$, \$STIMマクロ

(1) 機能

タスクプログラムを指定した時間間隔で周期時間起動する。

(2) 起動方法

① 形式

```
DI  
CALL STIM$
```

タスクマクロ

あるいは

```
CALL $STIM
```

モニタマクロ

② パラメータ

C Reg : 周期時間起動するタスクプログラムの番号を設定する。

DE Reg : 周期時間間隔 ($\times 100\text{msec}$) を設定する。

(3) 注意事項

- ① すでに周期時間起動されているタスクプログラムに対して、再度周期時間起動の指定を本マクロで行うと、再指定された周期時間間隔で起動される。
- ② 周期時間を $I\text{msec}$ ($DE \times 100\text{msec}$), タスクプログラムの実行時間を $T\text{msec}$ として、タスクプログラムの周期時間起動を図示すると次のようになる。

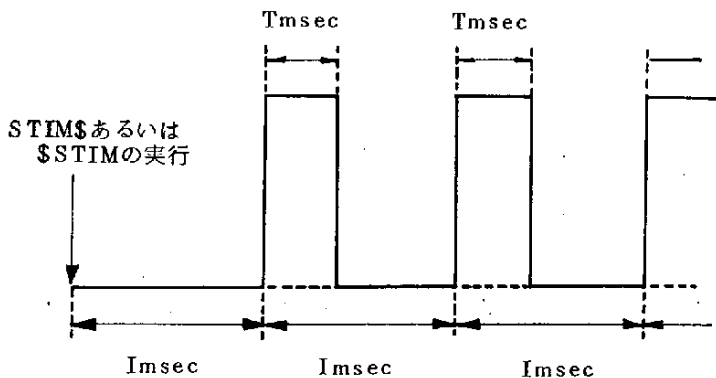


図 3-1 STIM\$と\$STIMマクロの実行

3.1.4 RTIM\$, \$RTIMマクロ

(1) 機能

STIM\$あるいは\$STIMマクロで指定したタスクプログラムの周期時間起動を解除する。

(2) 起動方法

① 形式

```
DI
CALL RTIM$
```

タスクマクロ

あるいは

```
CALL $RTIM
```

モニタマクロ

② パラメータ

C Reg : 周期時間起動を解除するタスクプログラムの番号を設定する。

(3) 注意事項

- ① 周期時間起動していないタスクプログラムに対して、本マクロで周期時間起動の解除を行なったときは無視される。

3.2. タスクプログラムの同期制御

3.2.1 WAIT\$マクロ

(1) 機能

タスクプログラムの待ち合わせを行う。

(2) 起動方法

① 形式

DI CALL WAIT\$

② パラメータ

C Reg : 待ち合わせる (WAITさせる) タスクプログラムの番号を設定する。

(3) 注意事項

- ① WAITするタスクプログラムが自分自身のときは、他のタスクプログラムでPOST\$マクロを実行しWAITを解除するまでWAITする。
- ② WAIT\$とPOST\$マクロを使用したタスクプログラムの同期制御を例を用いて説明する。
ここではタスク番号T₁のアプリケーション・タスクプログラムとタスク番号T₂の入出力処理(ドライバ)タスクプログラムを考える。タスクプログラムT₁、タスクプログラムT₂を起動する。タスクプログラムT₂は入出力機器からデータを入力し、バッファエリアに設定する。タスクプログラムT₁は、タスクプログラムT₂の入力したバッファエリアにあるデータをもとに処理を継続する。この一連の処理を正しく実行するには、WAIT\$とPOST\$マクロを使用して次に示す図のようにタスクプログラムT₁とT₂の同期をとる必要がある。

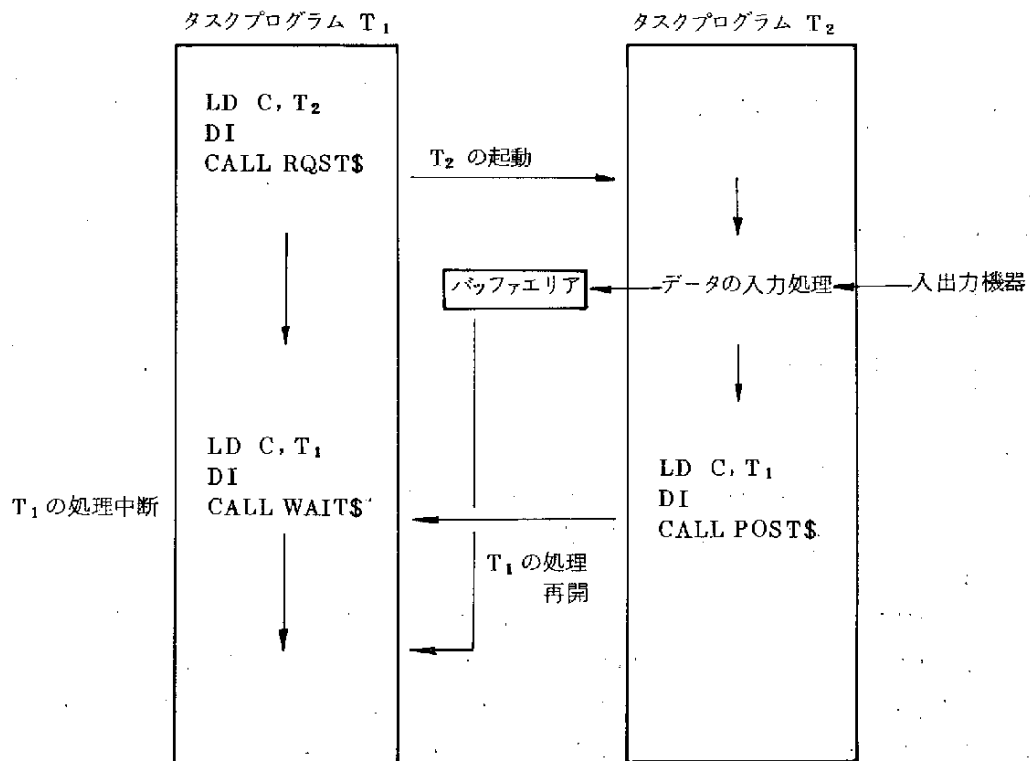


図 3-2 タスクプログラムの同期制御

3.2.2 POST\$ マクロ

(1) 機能

WAIT\$ マクロで指定したタスクプログラムの待ち合わせを解除する。

(2) 起動方法

① 形式

```

DI
CALL POST$

```

② パラメータ

C Reg : 待ち合わせを解除するタスクプログラムの番号を設定する。

(3) 注意事項

- ① WAIT 状態となっていないタスクプログラムに対して本マクロを実行したときは無視される。
- ② POST\$ マクロを使用したタスクプログラムの同期制御の例については、WAIT\$ マクロの注意事項を参照のこと。

3.2.3 ONTMSマクロ

(1) 機能

タスクプログラムを指定時間(遅延時間)だけ待ち状態にして、遅延時間経過後タスクプログラムを起動する。

(2) 起動方法

① 形式

```
DI  
CALL ONTMS
```

② パラメータ

C Reg : 遅延時間起動するタスクプログラムの番号を設定する。

DE Reg : 遅延時間($\times 100\text{ms}$)を設定

(3) 注意事項

- ① 遅延時間起動するタスクプログラムが自分自身のタスクプログラムのとき、遅延時間が経過するか他のタスクプログラムでOFTMSマクロを実行するまで待ち状態となる。
- ② 遅延時間起動するタスクプログラムは、本マクロ実行時に実行状態(RUNすなわち自分自身のタスクプログラム)か実行可能状態(READY)にななければならない。
- ③ すでに遅延時間起動指定されているタスクプログラムに対して本マクロを実行すると、その時点から再指定した遅延時間後起動される。

3.2.4 OFTMSマクロ

(1) 機能

タスクプログラムの遅延時間起動を解除する。

(2) 起動方法

① 形式

```
DI  
CALL OFTMS
```

② パラメータ

C Reg : 遅延時間起動を解除するタスクプログラム番号を設定する。

(3) 注意事項

- ① 本マクロで指定したタスクプログラムが遅延時間起動待ち状態となっていないときは無視される。

3.3 システム共有資源の管理

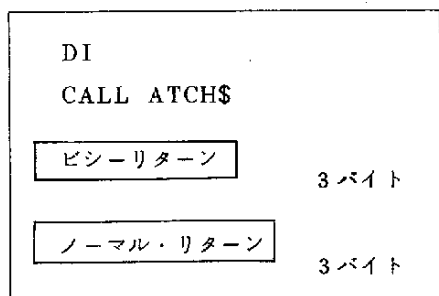
3.3.1 ATCH\$マクロ

(1) 機能

デバイス（外部入出力機器、バッファ）などのシステム共有資源（リソース）の専有を宣言する。

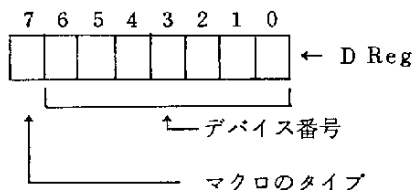
(2) 起動方法

① 形式



② パラメータ

D Reg : 専有するデバイス番号を設定する



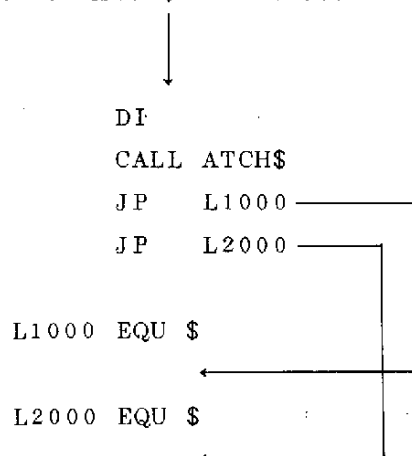
0 : WAITタイプ

1 : DEFERタイプ

(3) 注意事項

- ① 本マクロ実行時にデバイスが専有されているとき、WAITタイプであるとデバイスの専有が解除されるまで待ち状態となり、専有が解除されたときにノーマル・リターンする。DEFERタイプであるとデバイス専有中のときはビジー・リターンし、専有されていないときはノーマル・リターンする。
- ② エンプティあるいはノーマルリターン先には3バイトの領域があり、利用者はJP命令を使用して、各々のリターン別に処理の振り分けができる。

図 3-3 ATCH\$ マクロの戻り先



3.3.2 DTCH\$ マクロ

(1) 機能

デバイスの専有を解除する。

(2) 起動方法

① 形式

```

DI
CALL DTCH$
    
```

② パラメータ

D Reg : 専有を解除するデバイス番号を設定する。

(3) 注意事項

なし

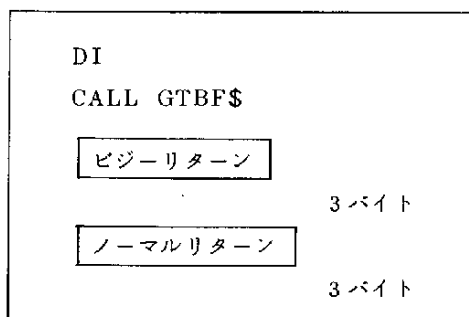
3.3.3 GTBF\$ マクロ

(1) 機能

バッファエリア@BUFFから指定された連続するブロックの領域を確保し、確保した領域の先頭アドレスを返す。

(2) 起動方法

① 形式



② パラメータ

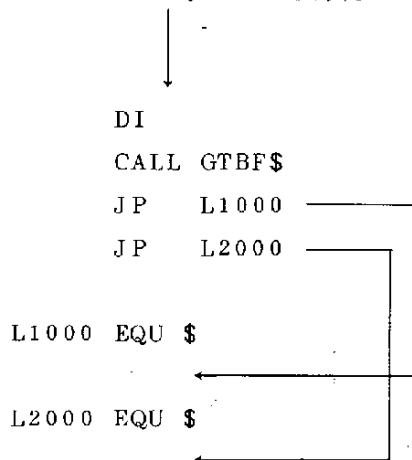
C Reg : 確保するブロック数を設定する。

DE Reg : 確保したバッファエリアの領域の先頭アドレスが設定される。

(3) 注意事項

- ① 1 ブロックの大きさ (バイト) とバッファエリア@BUFF内の全ブロック数は利用者が定義しブロック管理情報@BLOKに設定する。
- ② 本マクロを使用して複数のブロックを同時に確保することができたとき、それらのブロックはアドレス上連続している。
- ③ 指定されたブロック数の領域を確保できたときはノーマルリターンし、ブロック数が十分であっても連続したブロックを確保できなかったとき、あるいはブロック数が不足のときはビジーリターンする。
- ④ ビジーあるいはノーマルリターン先には3バイトの領域があり、利用者はJP 命令を使用して、各々のリターン別に処理の振り分けができる。

図 3-4 GTBF\$ マクロの戻り先



3.3.4 PTBF\$マクロ

(1) 機能

確保してあるバッファエリアの領域を解放する。

(2) 起動方法

① 形式

```
DI  
CALL PTBF$
```

② パラメータ

C Reg : 解放する領域のブロック数を設定する。

DE Reg : 解放するバッファエリア内の領域の先頭アドレスを設定する。

(3) 注意事項

なし

3.4 タスクプログラム間の相互通信

3.4.1 RQST\$, \$RQSTマクロ

3.4.1.1 RQST\$, \$RQSTマクロの説明を参照のこと

3.4.2 RTRV\$マクロ

(1) 機能

本マクロ実行タスクプログラムのキューバッファ@QUEB内に待ち行列を作っているパラメータを取り出す。

(2) 起動方法

① 形式

```
DI  
CALL RTRV$  
  
エンブティリターン 3バイト  
  
ノーマルリターン 3バイト
```

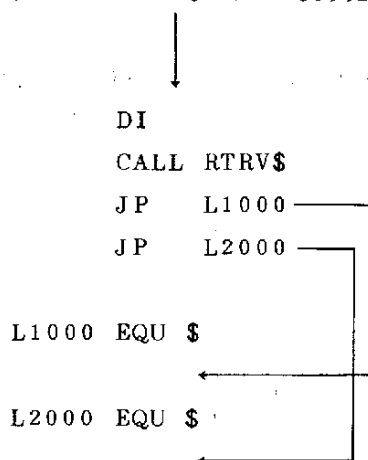
② パラメータ

DE Reg : 受けとるパラメータを格納する領域の先頭アドレスを設定する。

(3) 注意事項

- ① 本マクロ実行後でもキューバッファ@QUEB内にパラメータが残っていればEXIT\$マクロ実行後タスクプログラムは再起動される。
- ② 本マクロ実行時にタスクプログラムのキューバッファ@QUEBからタスクプログラムの領域へパラメータの転送が行なわれる。
- ③ パラメータを持たないタスクプログラムでもその最初の処理で本マクロを1度起動すること。タスク管理作業領域@TCBWの中のキューリザーブ数はRQST\$(\$RQST)マクロで+1されRTRV\$マクロで-1される。
- ④ キューバッファ@QUEBが空のときはエンプティリターンし、パラメータをタスクプログラムの領域に転送終了時はノーマルリターンする。
エンプティあるいはノーマルリターン先には3バイトの領域があり、利用者はJP.命令を使用して、各々のリターン別に処理の振り分けをできる。

図 3-5 RTRV\$ マクロの戻り先



- ⑤ RQST\$あるいは\$RQSTとRTRV\$マクロでの実際のパラメータ受けわたしの状態を次の図に示す。

パラメータを送る側のタスクプログラムの番号を T_1 , 受けとる側のタスクプログラムの番号を T_2 とする。

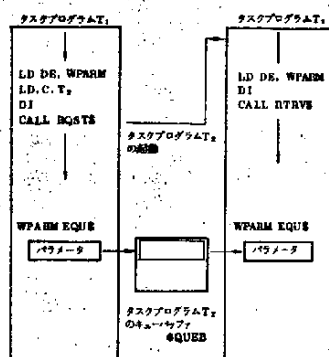


図 3-6 RQST\$(\$RQST)とRTRV\$マクロのパラメータ受け渡し

3.5 外部入出力機器の制御

3.5.1 IORQ\$マクロ

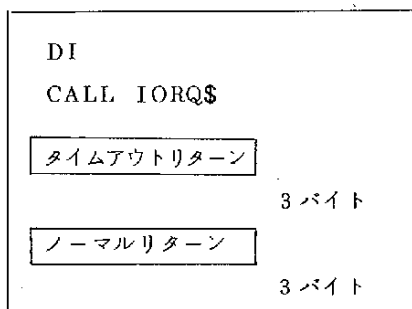
(1) 機能

本マクロを起動したタスクプログラムをデバイス（外部入出力機器）の入出力完了待ち状態とする。デバイスからの割り込みがあるか、指定入出力時間の超過（タイムアウト）で、タスクプログラムの入出力完了待ちを解除する。

本マクロの実行前にATCH\$マクロで、デバイスの専有宣言をしておく必要がある。

(2) 起動方法

① 形式



② パラメータ

D Reg : デバイス（外部入出力機器）番号を設定する。

HL Reg : タイムアウト時間（×100msec）を設定する。

(3) 注意事項

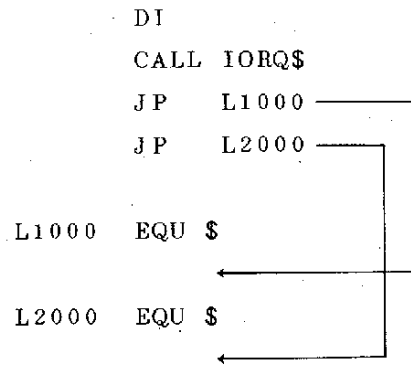
① デバイス番号は利用者が外部入出力機器に割り当てた番号である。

デバイス割り込みマスク@DMSK上の該当する外部入出力機器の順序と対応をとる必要がある。

② 指定した入出力時間内にデバイス（外部入出力機器）からの応答があったときはノーマルリターンし、入出力時間が経過しても応答がないときはタイムアウトリターンする。

ノーマルリターンあるいはタイムアウトリターン先には3バイトの領域があり、利用者はJJP命令を使用して、各々のリターン別に処理の振り分けができる。

図3-7 IORQ\$ マクロの戻り先



⑧ IORQ\$ マクロの処理概要を次に示す。

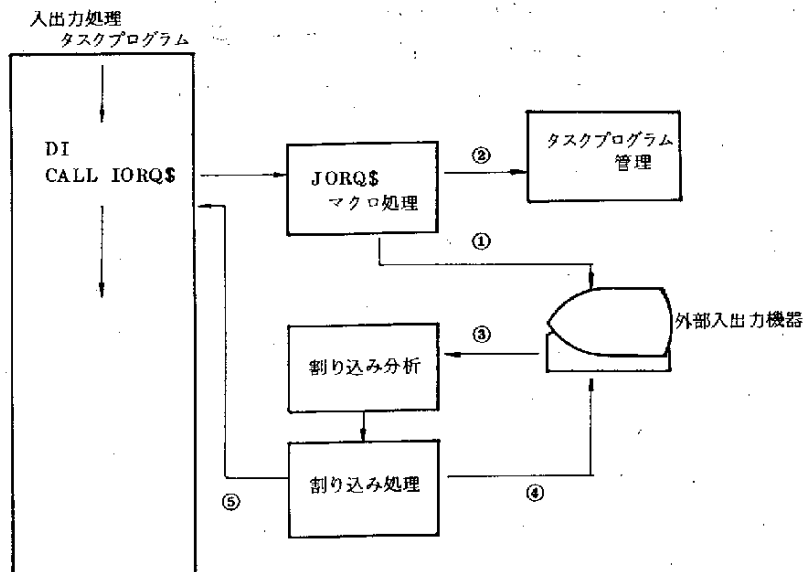


図3-8 IORQ\$ マクロの処理

- ① 外部入出力機器の割り込みマスクを開ける。
- ② 外部入出力機器との入出力処理実行中、実行可能状態 (READY) の他のタスクプログラムを実行させるためにタスクプログラム管理へ制御がわたる。
- ③ 外部入出力機器からの割り込みが入る。
- ④ 外部入出力機器の割り込みマスクを閉じる。
- ⑤ 入出力処理タスクプログラムへ制御が戻り処理が続行される。

4. システム共通領域の説明

システム共通領域とは、モニタプログラムがタスクプログラムの実行管理を行うとき、タスクプログラム自身が走行するとき、さらにモニタプログラムが提供しているマクロの実行処理などに必要とする領域のことである。

システム共通領域のほとんどは、利用者が作成するシステムごとに異なる。

システム共通領域でモニタプログラム、あるいは、利用者が作成するタスクプログラムで作業領域として使用する領域以外は、モニタプログラム作成時に個々の領域の形式に従って、利用者のシステム用に内容を定義しなければならない。また作業領域の中には、その大きさを利用者のシステムに合わせた大きさとしなければならないものもある。

本モニタプログラムでは、利用者が容易に内容定義の必要な領域の展開ができるようなMACRO-80アセンブラ言語のマクロ定義命令を使用したマクロを用意してある。利用者は、可変値をパラメータで与えマクロを呼び出すことで容易に要求する内容を持つ領域を展開できる。

システム共通領域には、利用者が、その領域を主記憶域上に確保しておくだけで良いものもある。しかし、利用者がモニタプログラムの処理内容を把握しようとするとき理解の助けになるように、ここでは、一応モニタプログラムで使用する、すべての共通領域について説明してある。

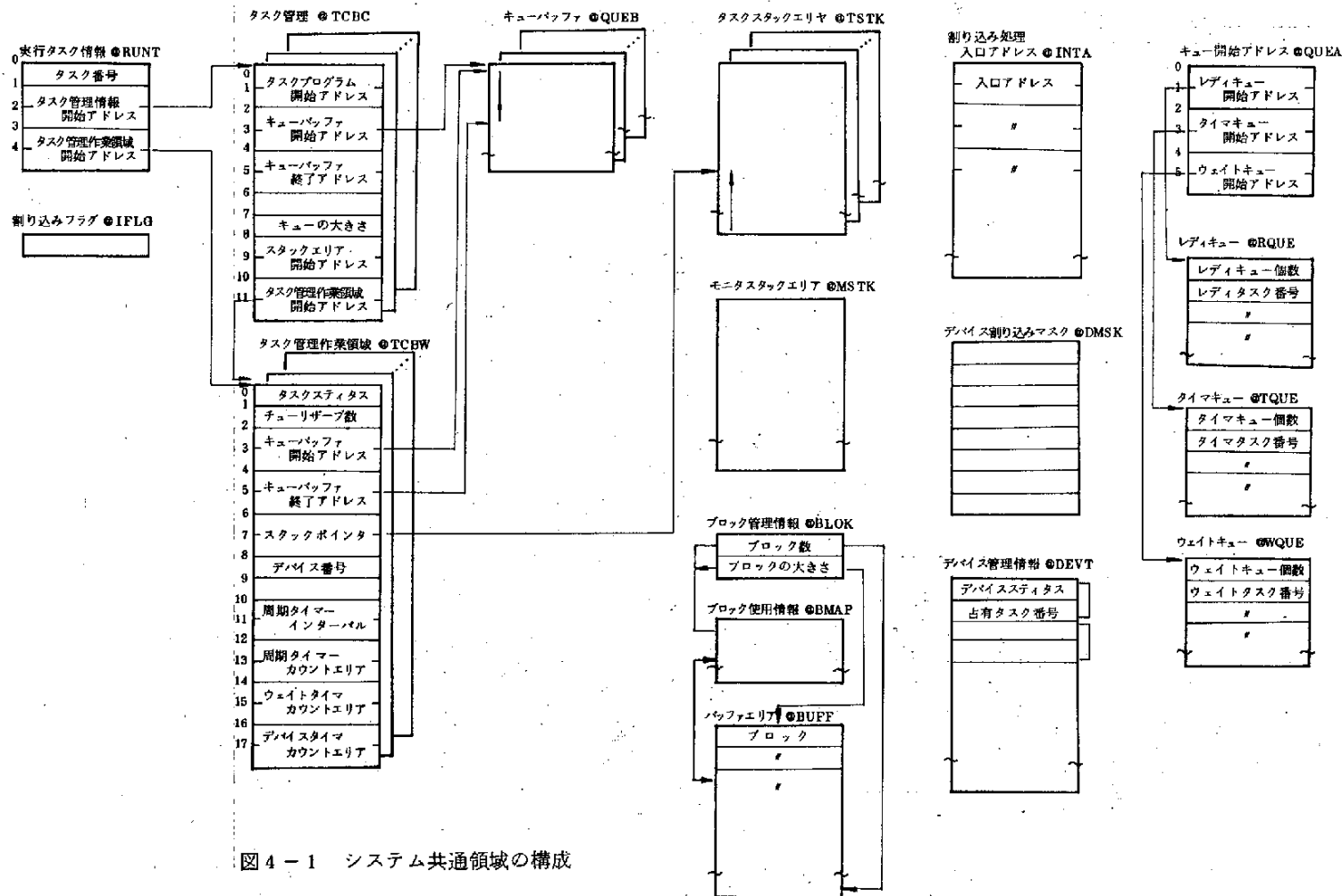


図4-1 システム共通領域の構成

4.1 モニタプログラム作業領域

4.1.1 実行タスク情報 (@RUNT)

(1) 機能

現在，中央処理装置 (CPU) を専有して走行しているタスクプログラムの情報を抑えている。

(2) 形式

@RUNT

0	タスク番号
1	
2	タスク管理情報 開始アドレス
3	
4	タスク管理作業領域 開始アドレス

タスク番号：現在モニタプログラム配下で走行しているタスクプログラムの番号が設定される。

アイドル状態のときは FFH が設定される。

タスク管理情報開始アドレス

：タスク番号に該当するタスク管理情報 @TCBC の開始位置が設定される。

タスク管理作業領域開始アドレス

：タスク番号に該当するタスク管理作業領域 @TCBW の開始位置が設定される。

(3) 注意事項

- ① モニタプログラムが作業領域的に使用する領域であり，モニタプログラム作成時に 5 バイトの領域を確保するだけで良い。
- ② モニタプログラムのスタートアップ時に，初期化処理でタスク番号に FFH すなわちアイドル状態が設定される。

4.1.2 割り込みフラグ (@IFLG)

(1) 機能

外部割り込みかプログラム割り込みかの識別に使用する。

(2) 形式

@IFLG

割り込みフラグ

割り込みフラグ：割り込みの識別に使用する。

0：プログラム割り込みである。

1：外部割り込みである。

(3) 注意事項

- ① 外部割り込みは外部入出力機器からの割り込みで、プログラム割り込みはタスクプログラムからモニタプログラムのマクロ実行要求である。
- ② タスクプログラム管理の最後で割り込み元への復帰、すなわち RET か RETI いずれの命令の実行かは、このフラグを判定して行う。
- ③ モニタプログラム作成時に 1 バイトの領域を確保するだけで良い。
- ④ モニタプログラムのスタートアップ時に、初期化処理で 0 が設定される。

4.2 タスクプログラム管理領域

4.2.1 タスク管理情報(@TCBC)

(1) 機能

タスクプログラムの位置、あるいは、実行時に使用する領域などタスクプログラムの固定情報を格納する。

モニタプログラム作成時に、タスク管理情報でモニタプログラム配下で走行するタスクプログラムを登録し、その情報を定義する。

(2) 形式

@TCBC

0	
1	タスクプログラム 開始アドレス
2	
3	キューバッファ 開始アドレス
4	
5	キューバッファ 終了アドレス
6	
7	
8	キューの大きさ
9	スタックエリア 開始アドレス
10	
11	タスク管理作業領域 開始アドレス

タスクプログラム開始アドレス

：タスクプログラムの最初の実行命令の位置を設定する。

タスクプログラム起動時に、この開始位置に制御がわたる。

キューバッファ開始アドレス

：タスクプログラム起動時にわたされるパラメータ格納用領域である。キュー

バッファ@QUEBの開始位置を設定する。

キューバッファ終了アドレス

：キューバッファ@QUEBの終了位置を設定する。

キューの大きさ

：タスクプログラム起動時にわたされるパラメータの大きさ（バイト）を設定する。

スタックエリア開始アドレス

：タスクプログラムが使用するスタックエリア@TSTKの開始位置を設定する。

タスク管理作業領域開始アドレス

：タスクプログラム管理作業領域@TCBWの開始位置を設定する。

(3) 注意事項

- ① モニタプログラム作成時に、タスクプログラムごとにタスク管理情報の領域を確保し、各タスクプログラムの情報を設定する。
- ② タスク番号はタスク管理情報領域内での各タスクプログラムの定義されている順番である。
すなわち、最初に定義されている管理情報を持つタスクプログラムの番号が0で以降1つずつ昇順となっている。
- ③ タスク管理情報にはタスクプログラムの固定情報が設定され、システムの立ち上げから終了まで一定であり更新されない。タスクプログラムの可変情報はタスク管理作業領域@TCBWに設定される。

4.2.2 タスク管理作業領域（@TCBW）

(1) 機能

タスクプログラム実行時の情報、また、実行中のタスクプログラムの処理が中断するとき実行再開に必要な情報を格納する。

(2) 形式

@TCBW

0	タスクステータス
1	キューリザーブ数
2	キューバッファ
3	開始アドレス
4	キューバッファ
5	終了アドレス
6	
7	スタックポインタ
8	デバイス番号
9	
10	
11	周期タイム インターバル
12	
13	周期タイム カウントエリア
14	
15	ウェイトタイム カウントエリア
16	
17	デバイスタイム カウントエリア

タスクステータス

：タスクプログラムの現在の状態を示す。

注意事項を参照のこと。

キューリザーブ数

：キューバッファ@QUEB上にある現在のキュー（パラメータ）の個数が設定される。

キューバッファ開始アドレス

：次のRTRV\$マクロで取り出されるキューの位置が設定される。

キューバッファ終了アドレス

：次のPQST\$(\$RQST)マクロで登録されるキューの位置が設定される。

スタックポインタ

：タスクスタックエリア@TSTK内の位置を指す。

デバイス番号

：デバイス専有解除待ち、あるいは入出力完了待ちとなっているとき、その対象デバイスの番号が設定される。

周期タイマインターバル

：周期時間起動タスクプログラムとなったときに、その周期時間間隔が設定される。

周期タイマカウントエリア

：周期時間起動タスクプログラムのとき、その周期時間経過を記録する。

ウェイトタイマカウントエリア

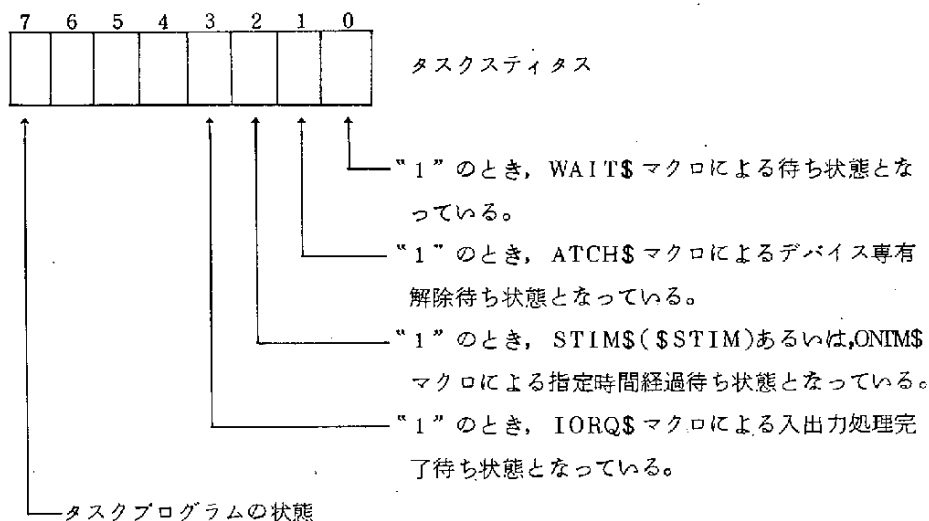
：遅延時間起動タスクプログラムのとき、その遅延時間経過を記録する。

デバイスタイマカウントエリア

：デバイスの入出力処理時間の経過を記録する。

(3) 注意事項

- ① タスク管理作業領域は、モニタプログラム作成時に各々のタスクプログラムに対して、18バイトの大きさの領域を確保するだけで良い。
- ② タスクステータスの詳細を次に示す。

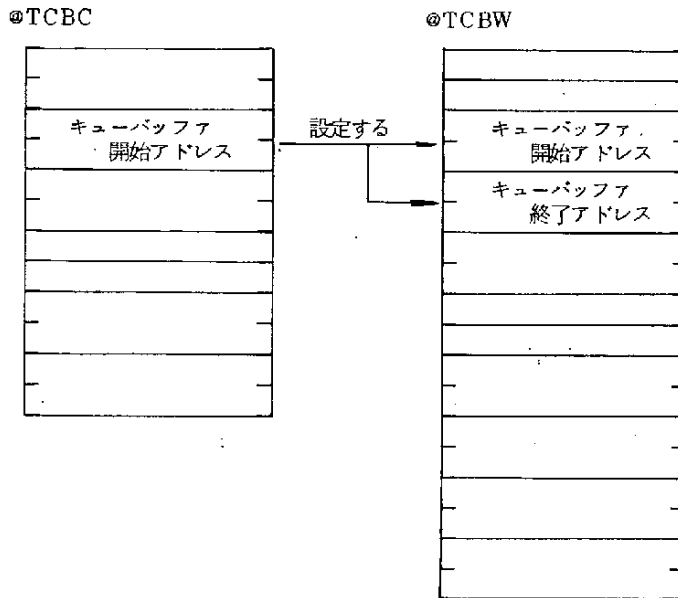


“0”：DEAD状態にある

“1”：RUN, READYあるいはWAIT状態にある。

- ③ タスク管理作業領域は、モニタプログラムのタスクプログラム実行管理で作業領域的に使用される。
- ④ モニタプログラムのスタートアップ時に、初期化処理でタスク管理作業領域のキューバッファ開始、終了アドレスにタスク管理情報@TCBCに設定されているキューバッファ開始アドレスを設定する。タスク管理作業領域の他の項目はすべてゼロクリアする。

図 4-2 タスク管理作業領域 @TCBW の初期設定



@TCBW のキューバッファ開始、終了アドレスに @TCBC に設定されているキューバッファ開始アドレスを設定する。

他の項はすべてゼロクリヤする。

この初期設定をすべてのタスクプログラムの @TCBW に対して行う。

4.3 キュー領域

4.3.1 キュー開始アドレス (@QUEA)

(1) 機能

レディキュー (@RQUE)、タイマキュー (@TQUE) さらにウェイトキュー (@WQUE) の開始アドレスを管理する。

(2) 形式

@QUEA

0	
1	レディキュー開始アドレス
2	
3	タイマキュー開始アドレス
4	
5	ウェイトキュー開始アドレス

レディキュー開始アドレス

: レディキュー@RQUEの開始位置が設定される。

タイマキュー開始アドレス

: タイマキュー@TQUEの開始位置が設定される。

ウェイトキュー開始アドレス

: ウェイトキュー@WQUEの開始位置が設定される。

(3) 注意事項

- ① 利用者はキュー開始アドレスの領域に手を加える必要はない。上記の形式で定義されたものをソース・プログラムで提供している。

4.3.2 レディキュー (@RQUE)

(1) 機能

実行可能状態 (READY) のタスクプログラムを登録しておく領域である。

タスクプログラムは実行優先順位の高い順にタスク番号で登録される。

(2) 形式

@RQUE	
0	レディキューの個数
1	レディタスク番号
2	レディタスク番号
3	レディタスク番号
	⋮

レディキューの個数

: レディキューに登録されているタスクプログラムの個数が設定される。

レディタスク番号

: 実行可能状態 (READY) となっているタスクプログラムの番号が設定される。

(3) 注意事項

- ① 利用者のシステムで生成すると考えられる実行可能状態のタスクプログラムを見積もり、レディキューの領域の大きさを決める必要がある。
- ② モニタプログラムのスタートアップ時に初期化処理でレディキューの個数に 0 を設定している。

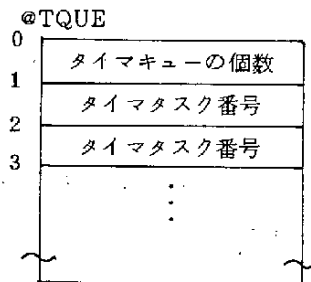
4.3.3 タイマキュー (@TQUE)

(1) 機能

IORQ\$ マクロによる入出力処理完了待ちや、STIM\$(\$STIM)マクロによる周期時間、あるいは、ONTM\$ マクロによる遅延時間経過待ちのタスクプログラムを登録しておく領域である。

タスクプログラムは実行優先順位の高い順にタスク番号で登録される。

(2) 形式



タイマキューの個数

：タイマキューに登録されているタイマタスクプログラムの個数が設定される。

タイマタスク番号

：入出力処理完了待ちや周期時間、あるいは遅延時間経過待ちとなっているタイマタスクプログラムの番号が設定される。

(3) 注意事項

- ① 利用者のシステムで生成すると考えられるタイマタスクプログラムを見積もり、タイマキューの領域の大きさを決める必要がある。
- ② モニタプログラムのスタートアップ時に、初期化処理でタイマキューの個数に0を設定している。

4.3.4 ウェイトキュー (@WQUE)

(1) 機能

WAIT\$ マクロにより実行権を一時放棄しているタスクプログラム、あるいはATCH\$ マクロを実行したが、デバイスが専有されていて解除されるまで実行に入ることができず待ち状態となっているタスクプログラムを登録しておく領域である。

タスクプログラムは実行優先順位の高い順にタスク番号で登録される。

(2) 形式

@WQUE

0	ウェイトキューの個数
1	ウェイトタスク番号
2	ウェイトタスク番号
3	ウェイトタスク番号
	⋮

ウェイトキューの個数

：ウェイトキューに登録されているウェイトタスクプログラムの個数が設定される。

ウェイトタスク番号

：POST\$ マクロ実行，あるいはデバイス専有解除待ちとなっているウェイトタスクプログラムの番号が設定される。

(3) 注意事項

- ① 利用者のシステムで生成すると考えられるウェイトタスクプログラムを見積もり，ウェイトキューの領域の大きさを決める必要がある。
- ② モニタプログラムのスタートアップ時に初期化処理でウェイトキューの個数に 0 を設定している。

4.3.5 キューバッファ (@QUEB)

(1) 機能

タスクプログラム間で受けわたすキュー（パラメータ）を一時的に格納する領域である。

(2) 形式

@QUEB

0	キュー
1	キュー
2	キュー
3	キュー
	⋮

キュー ：タスクプログラムのパラメータが設定される。

キューの大きさは，タスク管理情報 @TCBC に定義されている。

(3) 注意事項

- ① キュー（パラメータ）を持つタスクプログラム（タスク管理情報 @TCBC のキューの大きさ

が1以上)は、必ずキューバッファの領域を確保する必要がある。

② キューバッファ領域の大きさは、タスク管理情報@TCBCのキューの大きさの整数倍(1, 2, 3, ……)であり、タスクプログラムが受けとりの処理を開始するまでにわたされるキュー(パラメータ)の個数を考慮する必要がある。

③ キューバッファのキュー(パラメータ)の受けわたしはRQST\$(\$RQST)で行ない、受けとりはRTRV\$マクロで行う。

4.4 スタック領域

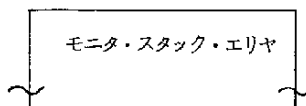
4.4.1 モニタ・スタック・エリア(@MSTK)

(1) 機能

モニタプログラムが使用する作業領域である。

(2) 形式

@MSTK



(3) 注意事項

① モニタスタックエリアはモニタプログラムでアセンブラ命令, CALL, PUSH, あるいは, POPなどを用いたときは使用される。

② モニタプログラム作成時に、利用者のシステムで必要な大きさの領域を確保するだけで良い。

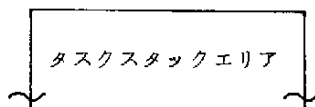
4.4.2 タスクスタックエリア(@TSTK)

(1) 機能

タスクプログラムが使用する作業領域である

(2) 形式

@STTK



(3) 注意事項

① タスクスタックエリアは、タスクプログラムでアセンブラ命令, CALL, PUSH, あるいは POPなどを用いたときに使用される。

- ② モニタプログラム作成時に、タスクプログラムで必要な大きさの領域を確保するだけで良い。
- ③ タスクスタックエリアは、利用者が作るタスクプログラム各々に必要である。

4.5 デバイス管理領域

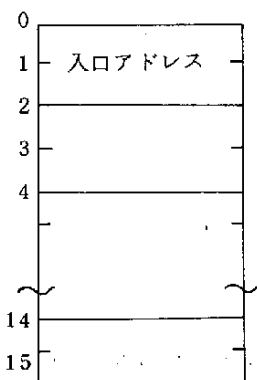
4.5.1 割り込み処理入口アドレス (@INTA)

(1) 機能

システムに接続している外部入出力機器に対する利用者作成の割り込み処理モジュールの入口を定義する領域である。

(2) 形式

@INTA



入口アドレス

：外部入出力機器の割り込み処理モジュールの最初の実行命令の位置を設定する。

(3) 注意事項

- ① 割り込み処理入口アドレスの最後 (@INTA+14) には、タイマ割り込み処理モジュール (ITMR\$) の入口アドレスが設定されている。

4.5.2 デバイス割り込みマスク (@DMSK)

(1) 機能

外部入出力機器から入る割り込みのマスクパターンを定義する。

(2) 形式

◎DMSK

0	マスクパターン
1	マスクパターン
2	#
3	#
4	#
5	#
6	#
7	#

マスクパターン

：外部入出力機器から割り込みが入ったときのマスクのビット位置を定義する。

(3) 注意事項

- ① デバイス（外部入出力機器）の割り込みマスクオープンは、マスクパターンで定義されているマスクを使用する。
- ② デバイス（外部入出力機器）とマスクパターンは、0 から 7 までの若番順のデバイス番号とデバイス割り込みマスク領域のアドレスの低い側からのマスクボタンで対応させる。

4.5.3 デバイス管理情報（@DEVT）

(1) 機能

外部入出力機器、バッファなどのシステム共有資源であるデバイス（リソース）の使用状態を管理する領域である。

(2) 形式

◎DEVT

0	デバイスステータス
1	占有タスク番号
2	デバイスステータス
3	占有タスク番号
4	⋮
	⋮

デバイスステータス

：デバイスの使用状態を示す。

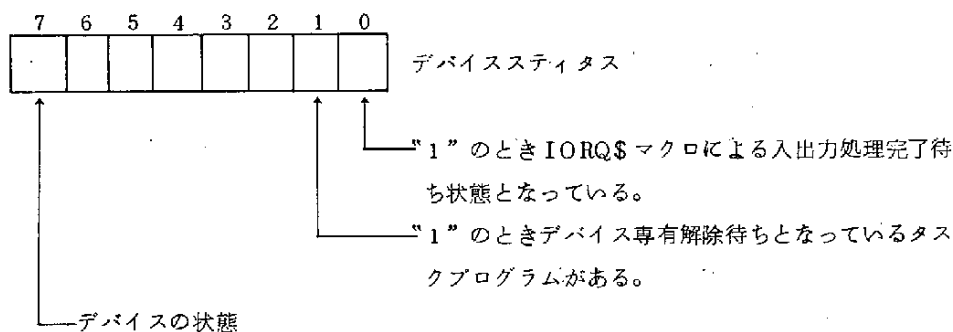
注意事項を参照のこと。

占有タスク番号

:ATCH\$ マクロでデバイス専有宣言し、現在使用中のタスクプログラムの番号が設定されている。

(3) 注意事項

- ① デバイス番号は、デバイス管理情報領域のデバイスにアドレスの低い側から順に0から始まる昇順で付与した番号である。
- ② デバイス管理情報の領域は、システムで保有するデバイス（リソース）の個数分確保すれば良い。領域の大きさは、デバイスの個数×2（バイト）である。割り込み関係の処理で使用する外部入出力機器用に最低デバイス8個分の領域は確保しておくこと。
- ③ デバイスステータスの詳細を次に示す。



"0" : 未使用である。

"1" : 使用（専有）中である。

- ④ 利用者はモニタプログラムのスタートアップ時に初期化処理でデバイス管理情報の領域を、すべてゼロクリアしなければならない。

4.6 バッファ管理領域

4.6.1 ブロック管理情報 (@BLOK)

(1) 機 器

システムで確保、保管するバッファ・エリアの1ブロックの大きさとブロック数を定義する。

(2) 形 式

@BLOK

0	ブロックの数
1	ブロックの大きさ

ブロックの数

：確保するバッファエリアを構成する
ブロックの全個数を設定する。

ブロックの大きさ

：1ブロックの大きさ（バイト）を設定する。

(3) 注意事項

- ① システムのバッファエリアを使用しないとき、すなわち、GTBF\$,あるいはPTBF\$マクロを使用しないときは、ブロック管理情報は必要ない。
- ② モニタプログラム作成時に、ブロック管理情報の領域を確保し、ブロック情報を定義すること。

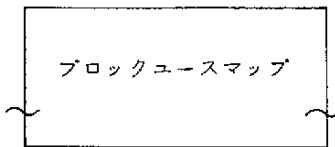
4.6.2 ブロック使用情報 (@BMAP)

(1) 機能

システムバッファエリア内のブロック使用状態を記録する領域である。

(2) 形式

@BMAP



ブロックユースマップ

：1ブロックを領域内の1ビットに対応させブロック単位に、その使用、未使用を記録する。

“0”：未使用である。

“1”：使用中である。

(3) 注意事項

- ① モニタプログラム作成時に、ブロック使用情報として必要な領域を確保するだけで良い。
領域の大きさは

$$\frac{\text{ブロック数}(\text{@BLOK})}{8} \text{ バイト (切り上げ)}$$

である。

- ② システムのバッファエリアを使用しないとき、すなわち、GTBF\$,あるいは、PTBF\$マクロを使用しないときは、ブロック使用情報は必要ない。
- ③ 利用者はモニタプログラムのスタートアップ時に初期化処理でブロック使用情報の領域を、

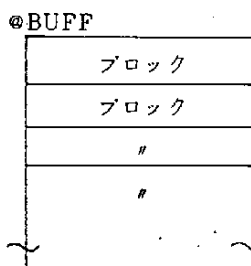
すべてゼロクリアしなければならない。

4.6.3 バッファエリア (@BUFF)

(1) 機能

GTBF\$ と PTBF\$ マクロを使用してブロック単位に動的に確保・解放が可能なシステムバッファエリアである。

(2) 形式



ブロック : ブロック管理情報で定義したバッファエリアの管理 (確保, 解放) 上の単位である。

(3) 注意事項

- ① モニタプログラム作成時に, バッファエリアとして必要な領域を確保するだけで良い。
領域の大きさは, ブロック管理情報 @BLOK で定義されている。ブロックの大きさとブロック数の積だけ確保すること。
- ② システムのバッファエリアを使用しないとき, すなわち GTBF\$ あるいは PTBF\$ マクロを使用しないときは, バッファエリアは確保する必要はない。

5. システム・ジェネレーションの説明

汎用的な機能を持ったモニタプログラムを利用者の開発システムに合わせ作成することを、システム・ジェネレーションという。実際には、2章で説明したモニタプログラムの5つの基本的な機能（利用者にソースプログラムの形式で提供）に対して、以下にあげる一連の操作を行うことで利用者の開発システム用に特殊化することである。

(1) システム共通領域の生成

モニタプログラムおよびタスクプログラムで使用するシステム共通領域の確保と定義を行う。

(2) オウンコーディング

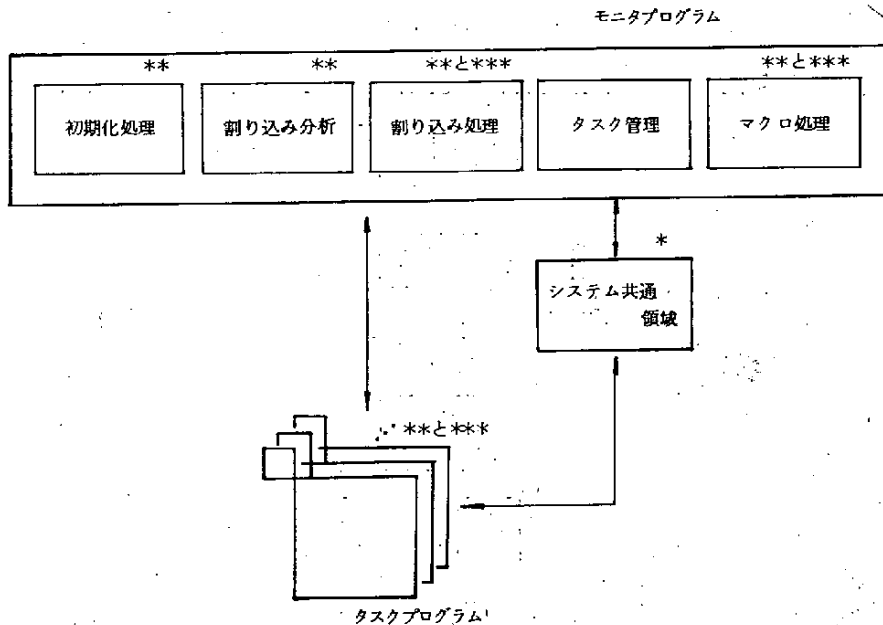
利用者のシステムに固有な処理、たとえば、初期化処理や外部入出力機器割り込み処理の部分を作成する。

(3) 機能の選択と追加

利用者のシステムに必要なマクロ処理などの機能の選択と利用者のシステムに固有な処理を追加する。

システム・ジェネレーションの操作には各種のユーティリティ・プログラムが必要である。

本モニタプログラムのシステム・ジェネレーションでは、プログラム開発用汎用モニタプログラムCP/Mと、その下で動くユーティリティ・プログラムの利用を考えている。



- * : システム共通領域の生成
- ** : オウンコーディング
- *** : 機能の選択と追加

図5-1 システム・ジェネレーションの範囲

5.1 システム・ジェネレーションの作業と手順

システム・ジェネレーションの作業は、利用者に提供されるモニタプログラムのソースプログラムに対して開発システム用の特殊化の操作を施し、最終的に実行形式のプログラムを作ることである。

システム・ジェネレーションの作業では、汎用モニタプログラムCP/Mを使用できる環境とCP/Mの下で動くテキストエディタ、アセンブラ、また、リンカーの各プログラムを必要とする。

表 5 - 1 システム・ジェネレーションの作業

作 業	作 業 内 容	使用プログラム
システム共通領域の生成	MACRO-80の擬似命令あるいはマクロ機能を使用して、ソースプログラムの中にシステム共通領域の確保と定義を行う。	MACRO-80 EDIT-80
OWNコーディング	MACRO-80の命令を使用して、初期化あるいは割り込みなどシステムに固有な処理のソースプログラムへの追加あるいは新規作成を行う。	MACRO-80 EDIT-80
機能の選択と追加	システムで使用するマクロ処理の選択あるいはシステムに固有な処理のモニタプログラムへの組み込みを行う。	LINK-80

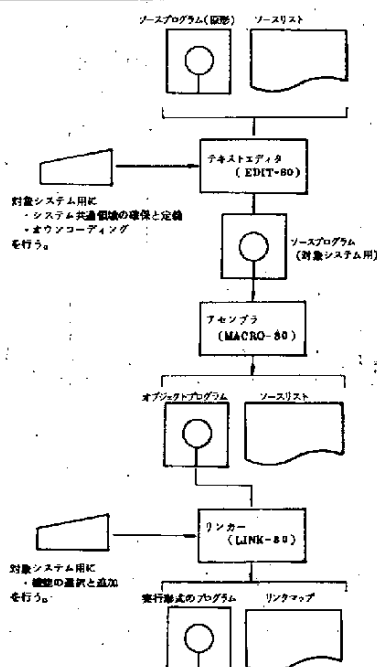


図 5 - 2 システム・ジェネレーションの手順

5.2 システム共通領域の生成

モニタプログラムとタスクプログラムで使用するシステムの共通領域は、システムジェネレーション時に対象とするシステムに合わせて、利用者が作成しなければならない。ここでは、このシステム共通領域の生成方法について説明する。

システム共通領域には大きく分けて、ただその領域を確保しておけばよいものと領域の確保と同時にその内容をあらかじめ定義しておくものがある。前者は、モニタプログラム、あるいはタスクプログラムで作業領域的に使用される領域でRAM(Random Access Memory)を使用する。それに対して後者はシステムの固定情報を定義しておく領域でモニタプログラムの処理部分と同じように一般にROM(Read Only Memory)に書き込まれて使用される。

システム共通領域の確保、および、内容の定義には汎用モニタプログラムCP/Mの配下で利用できるMACRO-80のマクロ機能あるいは擬似命令を使用する。

利用ごとのシステムで異なる領域の大きさ、あるいは、内容を持つと考えられる共通領域に対しては、その可変値をパラメータで与えることにより共通領域のイメージを展開することができるマクロをモニタプログラムで提供している。

(1) 共通領域の確保

モニタプログラム、あるいは、タスクプログラム実行時に1時的なデータ格納用に使用する領域で、MACRO-80の擬似命令DEFSで必要な大きさの領域を確保するだけである。利用者は提供されているMACRO-80のマクロ機能にパラメータを与えて、必要な領域を確保する。

大きさが固定となっている場合、モニタプログラムでその領域は確保してあるので利用者は、改めて領域を確保する必要はない。

領域確保だけの共通領域は、すべてモジュール@RAMT内で定義される。

(2) 共通領域の内容定義

システムジェネレーション時に固定情報として利用者システム用に内容定義が必要なものはすべて提供してある。MACRO-80のマクロ機能にパラメータを与えて領域確保と内容定義を同時に行う。

内容が利用者のシステムに依存しないものは、すでに、その領域確保と内容定義は行なわれている。利用者は内容定義など行う必要はない。

内容定義が必要な領域は、すべて、モジュール@ROMT内で定義される。

表 5 - 2 システム共通領域の生成

種 別	名 称	領域確保	内容定義	
モニタプログラム作業領域	ⓈRUNT	○		領域確保済
	ⓈIFLG	○		領域確保済
タスクプログラム管理領域	ⓈTCBC		○	
	ⓈTCBW	○		
キュー領域	ⓈQUEA		○	内容定義済
	ⓈRQUE	○		
	ⓈTQUE	○		
	ⓈWQUE	○		
	ⓈQUEB	○		
スタック領域	ⓈMSTK	○		領域確保済
	ⓈTSTK	○		
デバイス管理領域	ⓈINTA		○	
	ⓈDMSK		○	内容定義済
	ⓈDEVT	○		
バッファ管理領域	ⓈBLOK		○	システムで必要なときに領域確保、内容定義する。
	ⓈBMAP	○		
	ⓈBUFF	○		

注意 : ○が領域の確保あるいは内容の定義を必要とすることを意味する。

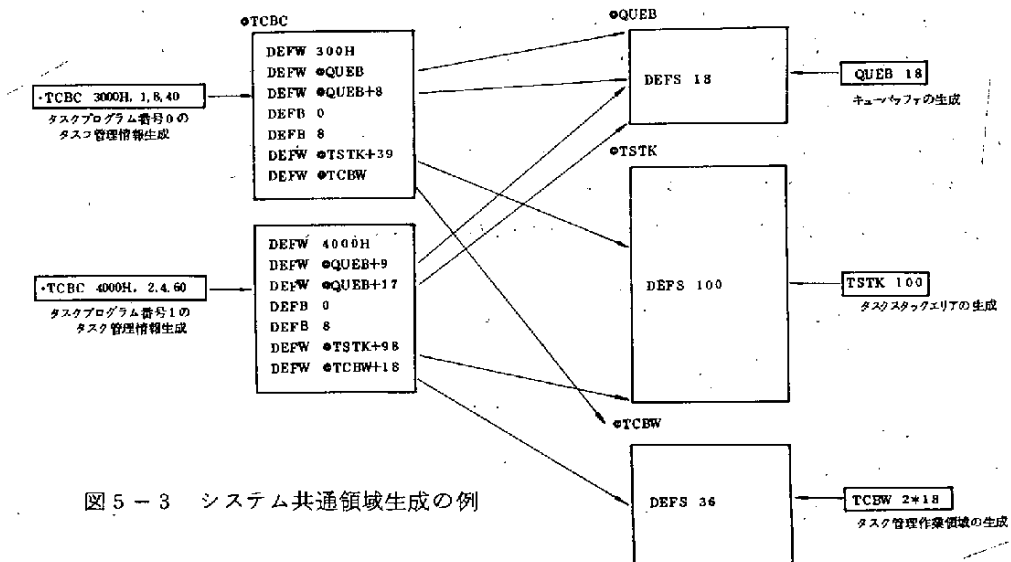


図 5 - 3 システム共通領域生成の例

表 5-3 システム共通領域を生成するマクロとパラメータ

種 別	マクロ 名 称	展 開 領域名	マクロ呼び 出し形式	パ ラ メ ー タ	備 考
タスクプログラム 管理領域	.TCBC	@TCBC	.TCBC P ₁ , P ₂ , P ₃ , P ₄	P ₁ :タスクプログラムの開始アドレス P ₂ :キューの最大個数 P ₃ :キューの大きさ P ₄ :使用するスタックエリアの大きさ	パラメータは 個々のタスク プログラムの 情報である。
	.TCBW	@TCBW	.TCBW P ₁	P ₁ :タスクプログラムの全個数	
キュー領域	.RQUE	@RQUE	.RQUE P ₁	P ₁ :レディキュー(タスクプログラム) の最大個数	
	.TQUE	@TQUE	.TQUE P ₁	P ₁ :タイマキュー(タスクプログラム) の最大個数	
	.WQUE	@WQUE	.WQUE P ₁	P ₁ :ウェイトキュー(タスクプログラ ム)の最大個数	
	.QUEB	@QUEB	.QUEB P ₁ , P ₂	P ₁ :すべてのキューバッファの大きさ P ₂ :タスクプログラムの全個数	
スタック領域	.TSTK	@TSTK	.TSTK P ₁ , P ₂	P ₁ :すべてのタスクスタックエリアの 大きさ P ₂ :タスクプログラムの全個数	
デバイス 管理領域	.INTA	@INTA	.INTA P ₁ , P ₂ ,...,P ₈	P ₁ :外部入出力機能の割り込み処理入 口アドレス(1≤n≤8)	P ₁ ≥ 8であ る。
	.DEVT	@DEVT	.DEVT P ₁	P ₁ :外部入出力機器などデバイス(リ ソース)の個数	
バッファ 管理領域	.BLOK	@BLOK	.BLOK P ₁ , P ₂	P ₁ :ブロックの個数 P ₂ :1ブロックの大きさ	
	.BMAP	@BMAP	.BMAP P ₁	P ₁ :ブロックユースマップ領域の大き さ	P ₁ :ブロック の個数/ 8の端数 を切り上 げた値
	.BUFF	@BUFF	.BUFF P ₁	P ₁ :バッファエリアの大きさ	P ₁ :ブロック の個数× 1ブロッ クの大き さ

注意:パラメータで大きさの単位はバイトである。

5.3 オウンコーディング

システム・ジェネレーションでは、モニタプログラムの初期化処理と外部入出力機器などの割り込み処理、あるいは、入出力処理に関係する部分については、利用者のシステムの状態に合わせてオウンコーディングする必要がある。

また、システム共通領域の生成も MACRO-80 のマクロ展開を使用せずに直接オウンコーディング、すなわち、アセンブラ命令を使用して領域の確保と内容の定義ができる。

ここで、オウンコーディングとは、利用者が自ら M-80 用アセンブラ言語を使用して、モニタプログラムの中に処理の追加、削除などの変更、また、新たな処理の作成を行うことを意味する。

本モニタプログラムは、機能で分割したモジュール単位で利用者に提供、公開している。また、モジュールには豊富な注釈が挿入されているのでオウンコーディングのときなどにモジュールの内容の理解、把握がしやすいようになっている。

5.3.1 システムの初期化処理

利用者が対象とするシステムで、スタートアップ時に行うシステム固有の初期化処理は、利用者が直接オウンコーディングでモジュール IDEV\$ に組み込まなければならない。

利用者のシステム固有の初期化処理として以下に示すものがある。

(1) 中央処理装置 (CPU) の初期設定

中央処理装置の割り込みモードをモード 0 とする。

(2) デバイスの初期設定

① インターバルタイマ

インターバルタイマに 100msec と周期の割り込みを発生させる。

② 外部入出力機器

中央処理装置のコミュニケーション・インタフェース、すなわち、USART の初期化を行う。

③ 割り込みコントローラ

割り込みベクタアドレスの設定などプログラマブル割り込みコントローラの初期化を行う。

(3) システム共通領域の初期設定

① デバイス管理情報 @DEVT

デバイス管理情報の領域の初期化 (ゼロクリア) する。

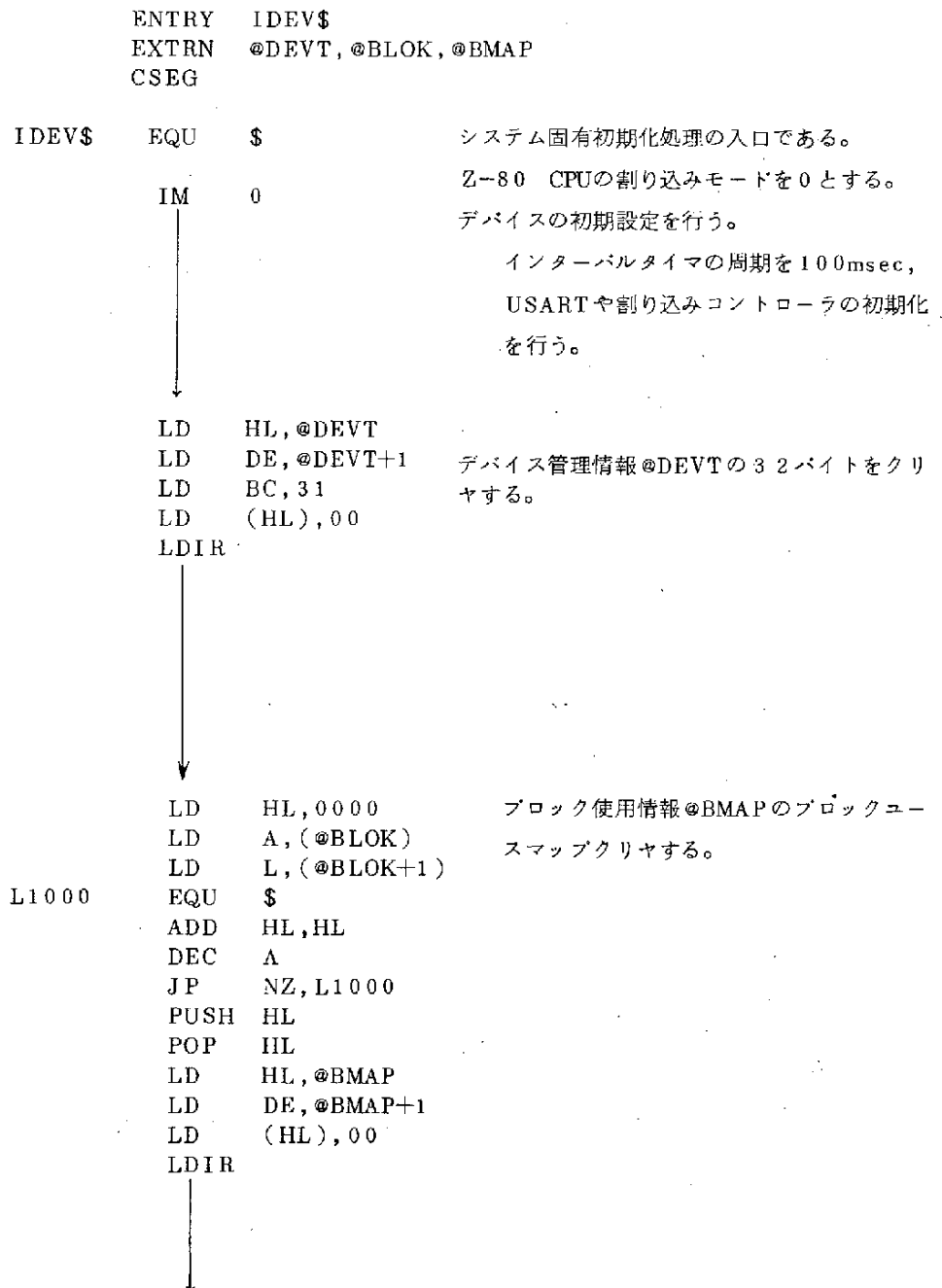
② ブロック使用情報 @BMAP

マクロ GTBF\$ あるいは PTBF\$ でシステムバッファエリアを使用するときブロック使用情報の領域を初期化 (ゼロクリア) する。

(4) RAM 領域の初期設定

必要によりシステムで使用する RAM 領域の初期化を行う。

図 5-4 システムに固有な初期化処理の例



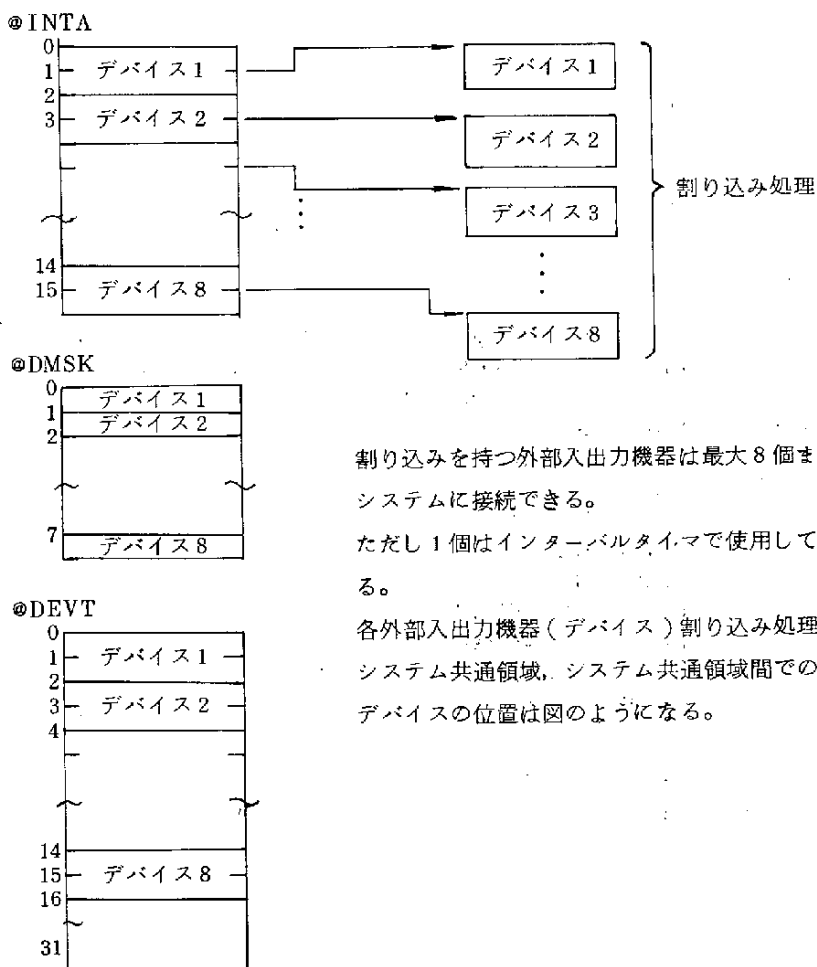
5.3.2 外部入出力機器割り込み処理

割り込み処理は外部入出力機器ごとに異なるために、本モニタプログラムでは利用者に提供していない。割り込み処理は、利用者のシステムに接続する外部入出力機器ごとに利用者がOWNコーディングで作成しなければならない。

割り込み処理は、モニタプログラムの割り込み分析機能と入出力処理（ドライバ）タスクプログラム起動までの中継ぎをする部分である。割り込み処理の共通部分はモニタプログラムの機能の一部として提供（RQUE\$）されているので、利用者は内容的には外部入出力機器の割り込みマスクのクローズとデバイス管理情報@DEVT上の位置を連絡するだけで良い。

上記の外部入出力機器割り込み処理を1つのモジュールで作成し、その入口アドレスを割り込み処理入口アドレス@INTAに設定する。このとき、割り込み処理入口アドレス@INTA上での位置は、デバイス割り込みマスク@DMSK上の位置と対応させること。

図5-5 外部入出力機器割り込み処理とシステム共通領域



割り込みを持つ外部入出力機器は最大8個までシステムに接続できる。

ただし1個はインターバルタイマで使用している。

各外部入出力機器（デバイス）割り込み処理とシステム共通領域、システム共通領域間での各デバイスの位置は図のようになる。

ENTRY	XXIO\$	
EXTRN	@DEVT, RQUE\$	
CSEG		
XXIO\$	EQU	\$ 外部入出力機器割り込み処理の入口である。
	IN	A, 80H 割り込みマスクの状態を入力する。
	RES	2, A 割り込みマスクの中の該当する外部入出力
	OUT	40H, A 機器のビットを"0"とする。
	LD	HL, @DEVT+4 割り込みマスク閉じる。
	JP	RQUE\$ 割り込み共通処理へ制御をわたす。
END		

図 5-6 外部入出力機器割り込み処理の例

5.3.3 最優先割り込み処理

外部の異常事態発生による最優先（マスク不能）割り込みが起きたときは 0066H 番地に制御がわたる。このとき本モニタプログラムの NMIP\$ モジュールが起動する。

利用者は最優先割り込みに対する処理を考えると、この NMIP\$ モジュールをOWNコーディングで作成しなければならない。割り込み処理終了後、再び割り込み前のもとの処理へ復帰するときは、レジスタの内容退避などの現状保存の対策をしなければならない。

ENTRY	NMIP\$	
CSEG		
NMIP\$	EQU	\$ 最優先割り込み処理の入口である。
	PUSH	AF 割り込み処理終了後、割り込み前の処理へ復帰する
	PUSH	BC ときは、使用していたレジスタの内容を退避する。
	PUSH	DE
	PUSH	HL
	↓	最優先割り込みに対する処理
	POP	HL 退避してあるレジスタの内容を回復する。
	POP	DE
	POP	BC
	POP	AF
	RETN	割り込み前の処理へ復帰する。
END		

図 5-7 最優先割り込み処理の例

5.4 機能の選択と追加

本モニタプログラムは2章で説明したモニタプログラムを構成する5つの基本的な機能を、さらに細分化した機能を持つリロケートブルなモジュールとして分割され、ソース・プログラム・ファイルに登録されている。

利用者のシステムに依存する機能、たとえば、外部入出力機器割り込み処理などは、利用者が作成したときに、ソース・プログラム・ファイルへ登録しておく。

システムジェネレーションのときに、リンカーLINK-80を使用して、ソース・プログラム・ファイルから必要な機能を選択、結合しモニタプログラムを作成する。

(1) 機能の選択

利用者はモニタプログラムのソース・プログラム・ファイルの中から利用者のシステムに必要なモジュール、すなわち、機能のみを選択できる。利用者は選択できる機能は原則としてマクロ処理の機能に限る。マクロ処理は各マクロごとに機能分割、すなわち、モジュール化してあるので、システムで必要なマクロ（ただし、\$RQSTは必須）のみを選択できる。

また、本モニタプログラムを利用している他のシステムの機能、たとえば、外部入出力機器の割り込み処理などは、システムジェネレーションの機能の選択で再利用することが可能となる。

機能の選択を行うことで、機能的に過不足のない記憶容量的に無駄のないモニタプログラムを作成できる。

(2) 機能の追加

利用者がOWNコーディングしたモジュールは機能の追加ということで、システムジェネレーション時にモニタプログラムの中に組み込む。OWNコーディングとしては、初期化処理（IDEV\$）、最優先割り込み処理（NMIP\$）、外部入出力機器割り込み処理、さらに、入出力処理が対象となる。

また、利用者が独自に作成したマクロ処理なども機能の追加ということでモニタプログラムの中に組み込むことができる。

表 5-4 モニタプログラムを構成する機能一覧

分 類	モジュール	機 能 概 要	選択可能	OWNコーディング	
				新規作成	機能追加
タスクプログラムの管理	EXEC	モニタプログラムの中心となりタスクプログラムの実行管理を行う			
システムの初期処理	INIT	モニタプログラムの恒常的な初期設定を行う			
	IDEV	利用者のシステムに固有な初期設定を行う			○
割り込み分析	INTR	外部入出力機器からの割り込みの分析する			○
割り込み処理	NMIP	最優先(マスク不能)割り込み時の処理を行う			
	ITMR	インターバルタイマからの割り込みに対する処理を行う			○
		外部入出力機器からの割り込み処理を行う		○	
	RQUE	割り込みに応じた入出力処理タスクプログラムの起動を行う			
マクロ処理	RQST	RQST\$, (\$RQST) マクロの処理を行う			
	EXIT	EXIT\$ マクロの処理を行う	○		
	STIM	STIM\$(\$STIM) マクロの処理を行う	○		
	RTIM	RTIM\$(\$RTIM) マクロの処理を行う	○		
	WAIT	WAIT\$ マクロの処理を行う	○		
	POST	POST\$ マクロの処理を行う	○		
	ONTM	ONTM\$ マクロの処理を行う	○		
	OFTM	OFTM\$ マクロの処理を行う	○		
	ATCH	ATCH\$ マクロの処理を行う	○		
	DTCH	DTCH\$ マクロの処理を行う	○		
	GTBF	GTBF\$ マクロの処理を行う	○		
	PTBF	PTBF\$ マクロの処理を行う	○		
	RTRV	RTRV\$ マクロの処理を行う	○		
	IORQ	IORQ\$ マクロの処理を行う	○		
モニタプログラム共通処理	EQUE	レディ、タイマあるいはウェイトキューにタスク番号を登録する			
	DQUE	レディ、タイマあるいはウェイトキューから一番優先順位の高いタスク番号を外す			
	SQUE	レディ、タイマあるいはウェイトキューから指定されたタスク番号を外す			
	GQUE	レディ、タイマあるいはウェイトキューの先頭アドレスを返す			
	TQUE	タイマキューから一番優先順位の高いタスク番号を外す			
	GTCB	タスク管理情報とタスク管理作業領域の先頭アドレスを返す			
	EXXR	レジスタの切換えを行う			
システム共通領域	ROMT	システムで共通的に使用される固定情報領域			○
	RAMT	システムで共通的に使用される可変情報領域			○

注意：○印が各項目の対象となるファイルである。

A>L80
Link-80 343 14-Apr-81...

リンカープログラム LINK-80 を起動する。

* /P:2000
*ROMT

固定情報を持つシステム共通領域の開始位置を指定する。

リンク情報

リンク情報としてプログラムおよびデータ領域の最初と最後の開始位置と未定義の汎用参照記号が出力される。

* /D:2800
*RAMT

可変なシステム共通領域の開始位置を指定する。

リンク情報

* /P:1000
*EXEC

次に指定するモジュールの位置すなわちモニタプログラムの開始位置を指定する。

リンク情報

*INIT

リンク情報

*IDEV

利用者のシステム用にOWNコーディングを行なった機能を追加する。

リンク情報

*RQST

リンク情報

*EXIT

利用者のシステムで使用するマクロ機能を選択する。

リンク情報

*WAIT

* /M

ここでは、先に説明したリンク情報の他に定義済汎用参照記号と、その他が出力される。

リンク情報

*RTM/N

モニタプログラムを格納するファイル名を指定する。

* /E

リンカープログラム LINK-80 の処理を終了し CP/M へ制御を返す。

リンク情報

図 5-8 LINK-80 を使用した機能の選択と追加の例

5.5 タスクプログラムの作成

タスクプログラムとは、プログラムの起動と実行においてモニタプログラムが管理かつ制御する1つの単位である。タスクプログラムはモニタプログラムとは独立に利用者が作成する。

以下にタスクプログラム作成の注意について説明する。

(1) タスクプログラムの分割

タスクプログラムの分割は、多重処理での走行を考慮して行うべきであり、意味のない多くのタスクプログラム分割は、システム共通領域の増加をまねき好ましくない。本モニタプログラムでは0から127までのタスク番号を使用でき、最大128個までのタスクプログラムを定義できる。

タスクプログラムは、タスク番号でその実行優先順位が決まる。そのため非同期に発生する割り込みに対する処理の実行順序を入出力処理（ドライバー）タスクプログラムのタスク番号で設定できる。

入出力処理タスクプログラムは、外部入出力機器と実際にデータなどの転送を行うタスクプログラムでシステムに接続する外部入出力機器ごとに必要である。

(2) タスクプログラムの製造

タスクプログラムの製造には、Z-80用アセンブラ言語の使用を前提とする。高級言語も利用できるが、モニタプログラムとの連絡部分であるマクロの起動処理は、アセンブラ命令で記述しなければならない。

タスクプログラムは、すべて、RTRV\$マクロで始まりEXIT\$マクロで終了する。

タスクプログラムで使用されているマクロ参照記号のアドレス解決は、モニタプログラムのリンクマップに出てくるマクロのアドレスを使用するか、あるいは、モニタプログラムとタスクプログラムの実行形式のプログラム作成を同時に行うことで可能である。

Z-80中央処理装置（CPU）は、構成の同じレジスタ（AF, BC, DE, HL）を2組持っている。これらのレジスタは、レジスタ切換え命令（EXあるいはEXX）でどちらか1組を選択できる。しかし、1組のレジスタは、モニタプログラム専用で使用しているためにタスクプログラムでのレジスタ切換えは禁止する。

5.6 汎用モニタプログラムCP/Mの利用

汎用モニタプログラムCP/Mとその下で動く各種ユーティリティ・プログラムは、マイクロコンピュータシステム開発支援用に非常に強力かつ有効なシステムである。

また、CP/Mはハードウェアに依存する割り込みを使用していないので、Z-80中央処理装置（CPU）を持つシステムへ簡単に移植できる利点を持っている。

モニタプログラムのシステムジェネレーションだけでなく、タスクプログラムの作成にもCP/Mとそのユーティリティプログラムを活用できる。

システムジェネレーションで作成するモニタプログラムと汎用モニタプログラムCP/Mは、ともに中央処理装置（CPU）としてZ-80を対象としている。このため、システムジェネレーション

したモニタプログラムさらにタスクプログラムもCP/Mを利用した実行かつデバッグが可能である。

表5-5に、CP/Mの下で利用できるユーティリティプログラムからシステムジェネレーションに関係する代表的なものを示す。

表5-5 CP/Mの下で利用できるユーティリティ・プログラムの例

プログラム	内 容	利 用 目 的
MACRO-80	インテル標準のマクロ機能を含み、高速かつ強力なZ-80マクロアセンブラである。	モニタプログラムまたタスクプログラム作成用言語である。マクロはシステム共通領域の生成に使用する。
EDIF-80	行単位にランダムアクセスできるテキスト・エディタである。ほとんどの編集機能を備えている。	システム共通領域の生成あるいはOWNコーディングで使用する。
LINK-80	リンキング・ローダ。 未定義の汎用参照記号のアドレスを解決する。	実行形式のモニタプログラムあるいはタスクプログラムを作成する。モニタプログラムの機能の選択と追加を行う。
ZSID	Z-80用シンボリックデバッガ。	メモリ内容の更新、参照、また、ブレークポイントの設定などモニタプログラム、タスクプログラムのデバッグに使用する。

付録 I 名称付与規則

利用者に公開し提供しているソース・リストに現われるモジュール名などの名称の命名規則について説明する。

MACRO-80 の汎用参照記号に該当するもの、すなわち、モジュール（入口）名などは他の記号と区別する意味で、名称の先頭あるいは最後の1文字に記号“\$”あるいは“@”を付与してある。

表1-1 名称付与規則

名称規則	意 味
C ₁ C ₂ C ₃ C ₄ \$ あるいは \$C ₁ C ₂ C ₃ C ₄	1つのアセンブル単位である（外部）モジュールの入口名を規定する。 あるいは、汎用参照記号として使用する（二次）入口名を規定する。
C ₁ C ₂ C ₃ C ₄ \$\$	1つの（外部）モジュール内で定義かつ使用される（内部）モジュール名を規定する。
@C ₁ C ₂ C ₃ C ₄	システム共通領域名を規定する。

付録Ⅱ マクロ命令とパラメータ

モニタプログラムで提供しているマクロ命令と、そのパラメータおよびマクロ処理終了時の情報を表Ⅱ-1に示す。パラメータの受けわたしは、すべて、タスクプログラムで使用するレジスタ經由で行う。

マクロの起動方法は次のとおりである。

○ モニタマクロ命令

CALL \$C₁C₂C₃C₄

○ タスクマクロ命令

DI

CALL C₁C₂C₃C₄\$

アブノーマルリターン

3 バイト

ノーマル・リターン

表Ⅱ-1 マクロ命令とパラメータ

情報 マクロ命令	パラメータ (レジスタ)						リターン	
	B	C	D	E	H	L	アブノーマルリターン	ノーマルリターン
RQST\$, \$RQST	—	タスク番号	パラメータ領域先頭アドレス		—	—	—	
EXIT\$	—	—	—	—	—	—	—	—
STIM\$, \$STIM	—	タスク番号	周期時間間隔		—	—	—	
RTIM\$, \$RTIM	—	タスク番号	—	—	—	—	—	
WAIT\$	—	タスク番号	—	—	—	—	—	
POST\$	—	タスク番号	—	—	—	—	—	
ONTM\$	—	タスク番号	遅延時間		—	—	—	
OFTM\$	—	タスク番号	—	—	—	—	—	
ATCH\$	—	—	デバイス番号	—	—	—	デバイス専有中	
DTCH\$	—	—	デバイス番号	—	—	—	—	
GTBF\$	—	ブロック数	バッファエリアの先頭アドレス		—	—	ブロック数不足	
PTBF\$	—	ブロック数	バッファエリアの先頭アドレス		—	—	—	
RTRV\$	—	—	パラメータ領域先頭アドレス		—	—	パラメータなし	
IORQ\$	—	—	デバイス番号	—	タイムアウト時間		タイムアウト発生	

付録Ⅲ モジュールの大きさ

モニタプログラムを構成しているモジュールの大きさ(バイト)を以下に示す。

表Ⅲ-1 モジュールの大きさ

モジュール	大きさ
EXEC	249
INIT	80
IDEV	1+X
INTR	71
NMIP	2+X
ITMR	188
RQUE	86
RQST	144
EXIT	62
STIM	48
RTIM	70
WAIT	40
POST	53
ONTM	49
OFTM	60
ATCH	83

モジュール	大きさ
DTCH	106
GTBF	143
PTBF	95
RTRV	105
IORQ	57
EQUE	40
DQUE	39
SQUE	32
GQUE	17
TQUE	23
GTCB	32
EXXR	29
ROMT	16+X
RAMT	36+X

注意：Xは利用者のシステムで可変な値

付録Ⅳ システムジェネレーションの実例

システムジェネレーションを実際のシステムでの例を使用して説明する。

先にシステムの外部環境などの設計条件について説明し、その後システム共通領域の生成などシステムジェネレーションの実例を説明している。タスクプログラムの作成例としてシリアルプリンタ出力処理タスクプログラムの中身を公開してある。

1. システムの概要

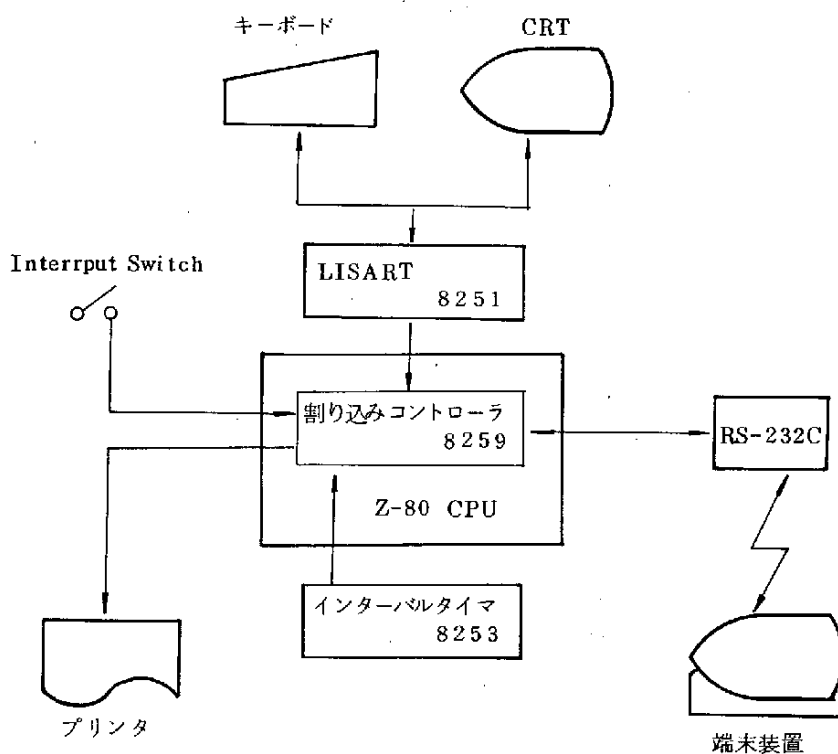
1.1 概 要

ここでは、Z-80中央処理装置(CPU)と端末装置を通信回線で結びコンソールのキーから入力する電文の送信した電文をプリンタへ出力する簡単なシステムを考える。システムジェネレーションするのは、上記中央処理装置(ホスト)上で走行するモニタプログラムである。

1.2 システムの構成

1.2.1 ハードウェア

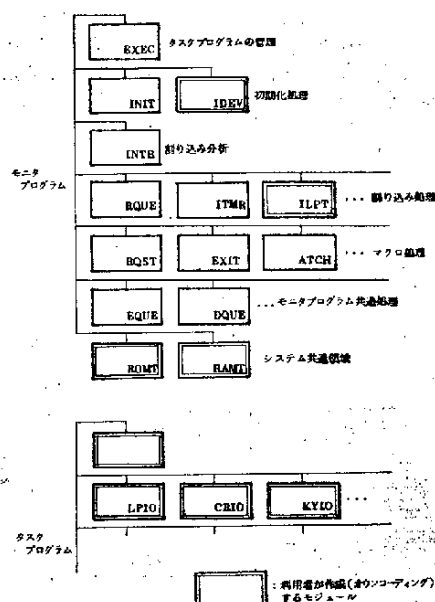
システムのハードウェア構成を図Ⅳ-1に示す。



図Ⅳ-1 ハードウェア構成

1.2.2 ソフトウェア

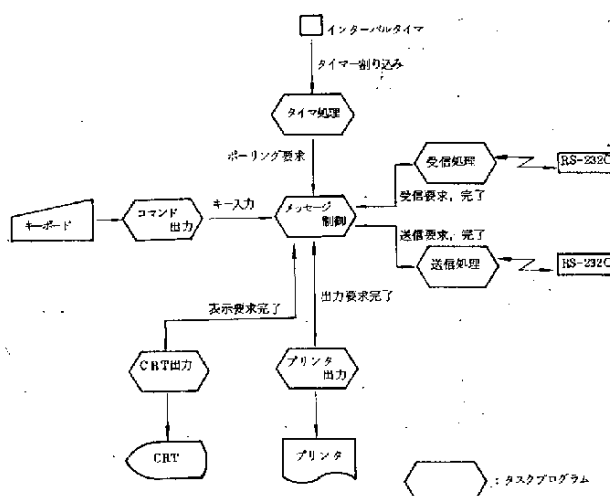
システムのソフトウェア構成を図Ⅳ-2に示す。



図Ⅳ-2 ソフトウェア構成

1.3 处理概要

システムの処理概要を図Ⅳ-3タスクプログラムの連絡の例で示す。



図Ⅳ-3 タスクプログラム間の連絡

1.4 ハードウェアインタフェース

1.4.1 インタフェース仕様

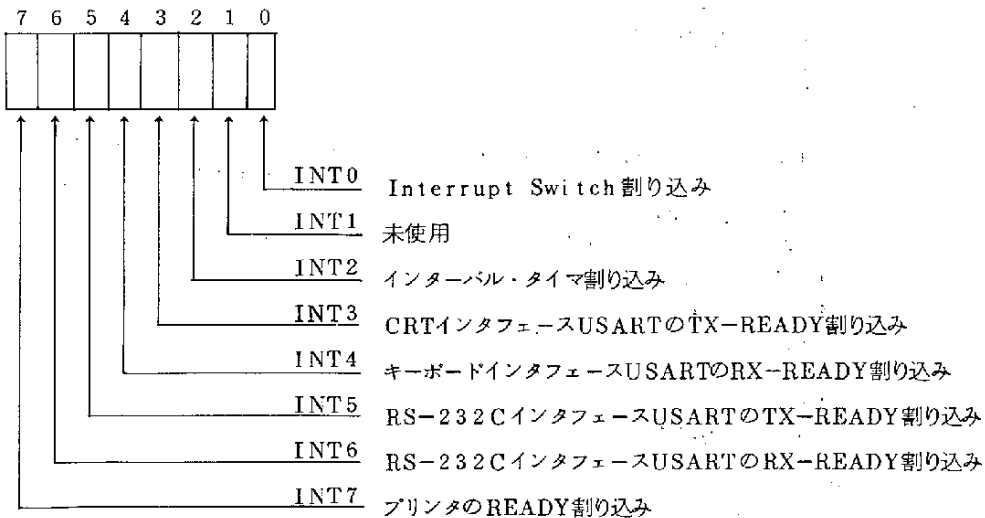
システムに接続している外部装置とのインタフェース仕様を表Ⅳ-1にまとめる。

表Ⅳ-1 インタフェース仕様

外部装置	インタフェース用LSI	インタフェース仕様	備 考
CRT	USART (Intel 8251 相当品)	調歩同期方式 2400 bit/sec	CRTとキーボードは同一のUSART を使用
キーボード	USART (Intel 8251 相当品)	調歩同期方式 2400 bit/sec	
プリンタ	パラレル I/O (Intel 8255 相当品)	セントロニクス・ コンパチブル 8bitパラレル インタフェース	
RS-232C	USART (Intel 8251 相当品)	調歩同期方式, 半二重 1200 bit/sec	

1.4.2 割り込みコントロール

割り込みコントローラとして専用LSI (Intel 8259 相当品) を用いている。8259は8レベルの割り込みを受け付ける。各割り込みの受け付け端子の位置 (優先順位) を図Ⅳ-4 にまとめる。



図Ⅳ-4 割り込み位置

2. システムジェネレーション

2.1 システム共通領域の生成

システムで使用する共通領域の確保およびその内容定義を提供されている MACRO-80 のマクロ機能にパラメータを与えることで行う。

```

EXTRY  @TCBC,...
EXTRN  @RQUE,...
EXTRN  @QUEB,...
EXTRN  CRIO$, LPIO$,...
EXTRN  ISWT$, ITMR$, ICRT$, ..... ILPT$  入出力処理タスクプログラ
                                           ムの入口(モジュール)名
                                           割り込み処理の入口(モジ
                                           ュール)名

CSEG

@TCBC  EQU    $

      • TCBC  CRIO$, 1,3,32  CRT出力タスクプログラムのタスク管理情報
      ↓                                     @TCBC展開指定
      • TCBC  LPIO$, 1,3,64  プリント出力タスクプログラムのタスク管理情報
                                     @TCBC展開指定

      DEFW  LPIO$
      DEFW  @QUEB+9
      DEFW  @QUEB+9+3*1
      DEFB  0
      DEFB  3
      DEFW  @TSTK+32+64-1
      DEFW  @TCBW+18
      ↓
      DEFW  OFFFHH

@INTA  EQU    $

      • INTA  ISWT$, 0, ITMR$, ICRT$, ..... , ILPT$  割り込み処理入口アドレス
                                                         の展開指定

      DEFW  ISWT$
      DEFW  0
      DEFW  ITMR$
      DEFW  ICRT$
      DEFW
      DEFW  :
      DEFW
      DEFW  ILPT$
      ↓
      DEFW

      END
  
```

図 IV-5 システム共通領域 (@ROMT) の作成

	ENTRY	@RUNT,...	
	ENTRY	@RQUE,...	
	ENTRY	@MSTK,...	
DSEG			
@RUNT	EQU	\$	
	DEFS	5	
@IFLG	EQU	\$	
	DEFS	1	
@DEVT	EQU	\$	
	.DEVT	8	デバイス管理情報の展開指定
	DEFS	8*2	
@TCBW	EQU	\$	
	.TCBW	7	タスク管理作業領域@TCBWの展開指定
	DEFS	7*18	
@QUEB	EQU	\$	
	.QUEB	256,7	キューバッファ@QUEBの展開指定
	DEFS	256+7	
@RQUE	EQU	\$	
	.RQUE	7	レディキュー@RQUEの展開指定
	DEFS	7+2	
@TQUE	EQU	\$	
	.TQUE	7	タイマキュー@TQUEの展開指定
	DEFS	7+2	
@WQUE	EQU	\$	
	.WQUE	7	ウェイトキュー@WQUEの展開指定
	DEFS	7+2	
@MSTK	EQU	\$	
	DEFS	50	
@TSTK	EQU	\$	
	.TSTK	320,7	タスクスタックエリア@TSTKの展開指定
	DEFS	320+7*14	
END			

図Ⅳ－６ システム共通領域（@RAMT）の作成

2.2 オウンコーディング

2.2.1 初期化処理

利用者のシステムに合わせた初期化処理は、モジュール IDEV\$ に利用者がオウンコーディングで組み込む。

ENTRY	IDEV\$	
EXTRN	\$STIM	
EXTRN	@DEVT	
IDEV\$	EQU	\$
IM	0	CPUの割り込みモードを0に設定
LD	A, 37H	] USART(8251)の初期化。コンソールのエラーリセット
OUT	0EDH, A	
LD	A, 37H	] USART(8251)の初期化。RS-232Cのエラーリセット
OUT	01-DH, A	
LD	C, 0DFH	インターバル・タイマ(8253)の初期設定
LD	A, 34H	] カウンタ0のモード設定
OUT	(C), A	
LD	A, 74H	] カウンタ1のモード設定
OUT	(C), A	
LD	A, 0B6H	] カウンタ2のモード設定
OUT	(C), A	
DEC	C	カウンタ2のI/Oアドレス設定
LD	A, 0AH	カウンタ2に値設定
OUT	(C), A	
LD	A, 00H	$1536000\text{Hz} \times 1 / 10$
OUT	(C), A	
DEC	C	カウンタ1のI/Oアドレス設定
LD	A, 00H	カウンタ1に値設定
OUT	(C), A	
LD	A, 06H	$15360\text{Hz} \times 1 / 1536$
OUT	(C), A	
DEC	C	カウンタ0のI/Oアドレス設定
LD	A, 64H	カウンタ0に値設定
OUT	(C), A	
LD	A, 00H	$1536000\text{Hz} \times 1 / 100$
OUT	(C), A	

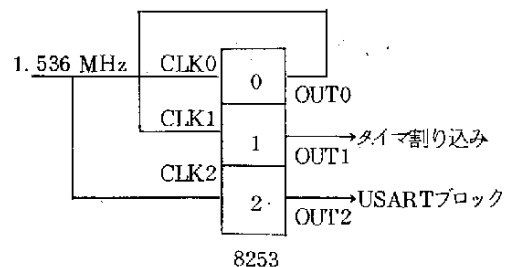


図 IV - 7 初期化処理 (IDEV\$) (1/2)

LD	C, 0D8H	] 割り込みコントローラ(8259)の初期設定
LD	HL, 0200H	
LD	A, L	
AND	0E0H	
ADD	A, 12H	] コマンドの内容・設定(ICW ₀)
OUT	(C), A	
INC	C	
OUT	(C), H	
LD	A, 02H	] コマンドの設定(ICW ₁ , ICW ₂ , OCW ₁)
OUT	(C), A	
LD	C, 04	] タスク番号4のタスクプログラムを100msec 間隔の周期起動させる。
LD	HL, 01	
CALL	\$STIM	
LD	HL, @DEVT	] デバイス管理情報の初期設定
LD	DE, @DEVT+1	
LD	BC, 16	
LD	(HL), 00	
LDIR	:	
RET		
END		

図Ⅳ-7 初期化処理(IDEV\$)(2/2)

2.2.2 割り込み処理

本システムでは、割り込みは全部で7レベルあり、各割り込みに対する処理を作成しなければならない。そのうちインターパルタイム割り込み処理は、モニタプログラムで提供しているのでそれを利用する。

実際例として、プリンタの割り込み処理をあげる。

ENTRY	ILPT\$	
EXTRN	RQUE\$	
EXTRN	@DEVT	
CSEG		
ILPT\$	EQU	\$
LD	HL, @DEVT+OEH	割り込みマスクは常に開けた状態にしている ので閉じる処理はない。 デバイス管理情報 @DEVT 内のプリンタ の位置アドレス設定。
JP	RQUE\$	
END		

図 IV - 8 プリンタの割り込み処理

2.3 機能の選択と追加

システムジェネレーションでの機能の選択あるいは追加は、リンカー LINK-80 を使用して行う。実際例として、LINK-80 に指定するコマンドを示す。なお各コマンド指定時に出力されるリンク情報は省略する。

A>L80	
* /P:1000	プログラム領域の番地指定
* /D:3000	データ領域の番地指定
*EXEC	タスクプログラム管理
*INIT	初期化处理
*IDEV	
*INTR	割り込み分析
*NMIP	
*ILPT	
*ICRT	
* :	割り込み処理
*	(利用者作成のものも含む)
*ITMR	
*RQUE	
*RQST	
*EXIT	
*STIM	マクロ処理
*ATCH	
*DTCH	
*RTRV	
*IORQ	
*EQUE	
*DQUE	
*SQUE	
*GQUE	タスクプログラム共通処理
*TQUE	
*GTCB	
*EXXR	
*ROMT	
*RAMT	システム共通領域
*LP10	
*CR10	タスクプログラム
* :	(利用者作成)
*	
* /M	
*RTM /N	
* /E	

注：リンク情報は省略してある。

図Ⅳ－9 MACRO-80を使用した機態の選択と追加

2.4 タスクプログラムの作成

タスクプログラムの作成例として、本システムを構成するタスクプログラムであるプリンタ出力タスクプログラムを示す。プリンタ出力タスクプログラムは、入出力処理（ドライバー）タスクプログラムの1つである。

プリンタ出力タスクプログラムは、1モジュールで構成されモジュールの入口（LPIO\$）がタスクプログラムの入口となっている。

```

ENTRY    LPIO$
EXTRN    RQST$,RTRV$,ATCH$,DTCH$,IORQ$,EXIT$

CSEG

LPIO$    EQU    $

        LD      DE,WIPAR      受けとるパラメータ格納領域の先頭アドレス
        DI
        CALL    RTRV$         パラメータの受けとり
        JP      L9000
        LD      HL,(WIPAR+1)   プリンタ出力情報先頭アドレス
        LD      D,CLPDV
        DI
        CALL    ATCH$         プリンタの専有宣言
        NOP
        NOP
        NOP

L1000    EQU    $

        DI
        LD      A,(HL)         プリンタ出力情報設定
        CPL
        OUT     (CLPDA),A      情報出力
        LD      A,80H
        OUT     (CLPST),A      STROBE OUTPUT

        LD      A,00
        OUT     (CLPST),A
        PUSH    HL
        LD      D,CLPDV        プリンタのデバイス番号設定
        LD      HL,CTOUT       出力処理タイムアウト時間の設定
        CALL    IORQ$          出力処理の制御
        JP      L2000          タイムアウト発生
        POP     HL
        INC     HL              次のプリンタ出力情報のアドレス
        LD      A,(HL)
        CP      0FFH           出力情報は終了か。
        JP      NZ,L1000

```

図Ⅳ-10 タスクプログラム(LPIO\$(1/2))

LD	D, CLPDV	プリンタのデバイス番号設定
DI		
CALL	DTCH\$	プリンタの専有解除
LD	A, CLPTN	プリンタ出力処理タスクプログラム番号の設定
LD	(WOPAR), A	
LD	A, 00	復帰情報(正常終了)の設定
LD	(WOPAR+1), A	
LD	A, (WIPAR)	プリンタ出力処理タスクプログラムを起動したタスクプログラムの番号設定
LD	C, A	
LD	DE, WOPAR	
DI		
CALL	RQST\$	
JP	L9000	
L2000	EQU \$	
	POP HL	
LD	D, CLPDV	プリンタのデバイス番号設定
DI		
CALL	DTCH\$	プリンタの専有解除
LD	A, CLPTN	プリンタ出力処理タスクプログラム番号の設定
LD	(WOPAR), A	
LD	A, 0FFH	復帰情報(異常終了)の設定
LD	(WOPAR+1), A	
LD	A, (WIPAR)	プリンタ出力処理タスクプログラムを起動したタスクプログラムの番号設定
LD	C, A	
LD	DE, WOPAR	
DI		
CALL	RQST\$	
L9000	EQU \$	
	DI	
CALL	EXIT\$	プリンタ出力処理タスクプログラムの終了処理
CLPDV	EQU 2	プリンタのデバイス番号
CLPDA	EQU 0E4H	
CLPST	EQU 0E6H	
CTOUT	EQU 40	タイムアウト時間(X100msec)
	DSEG	
WIPAR :	DEFS 3	受けとるパラメータの格納領域
WOPAR :	DEFS 8	送るパラメータの格納領域
	END	

図IV-10 タスクプログラム(LPIO\$(2/2))

付録V デモプログラムによる操作説明

1. デモの概要

デモ・プログラム実行時のシステム構成を、図V-1に示す。モニタ・プログラムおよびデモ・プログラムは、ホスト側CPU ユニット（DSC-80ZA）で走行する。また、端末側はパーソナル・コンピュータPC-8001で走行するプログラムによって制御されている。

デモ・プログラムには、プリンタ、キーボード、CRTディスプレイおよび端末側システム（端末側システムとはRS-232C インタフェースを介して接続される）が周辺装置として接続されている。デモ・プログラムは、これらの周辺装置を使用して、以下に示す処理が実行できる。

1) キー入力→CRT表示

ホスト側のキーボードより入力した文字を、ホスト側のCRT画面上に表示する。

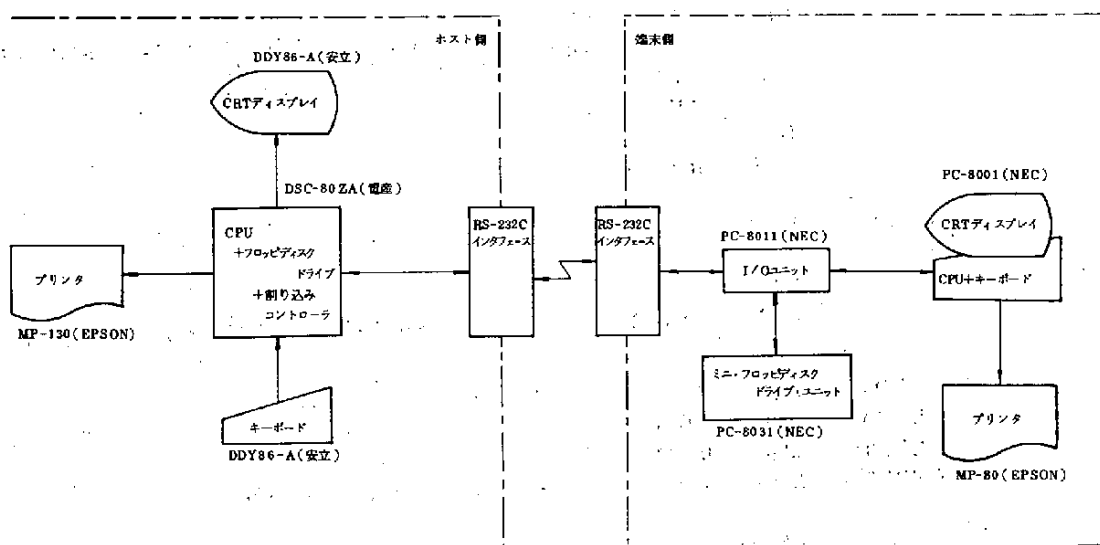
2) キー入力→端末プリンタ

ホスト側のキーボードより入力した文字を、端末側のプリンタに印字する。

3) 端末キー入力→プリンタ

端末側のキーボードより入力した文字を、ホスト側のプリンタに印字する。

デモ・プログラムでは、以上3つの処理を並列に実行する。



図V-1 デモ・システム構成

2. 操作説明

デモ・プログラムは、端末側の装置およびプログラムと共に実行される。以下に、ホスト側および端末側の操作方法を示す。

2.1 電源投入

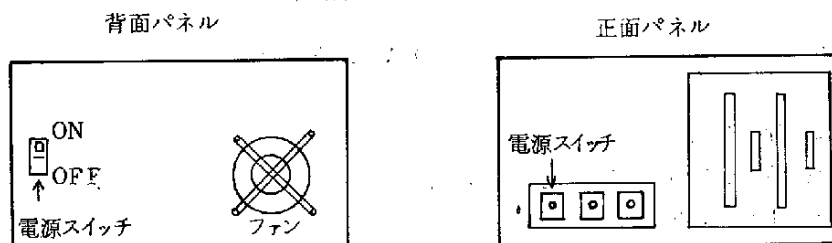
2.1.1 ホスト側

ホスト側全装置の電源投入方法を以下に示す。

1) CPUユニット(DSC-80ZA)

DSC-80ZAには、装置の正面および背面パネルに電源スイッチを有している。スイッチの投入は、まず背面パネル、続いて正面パネルの順に行なう(図V-2参照)。

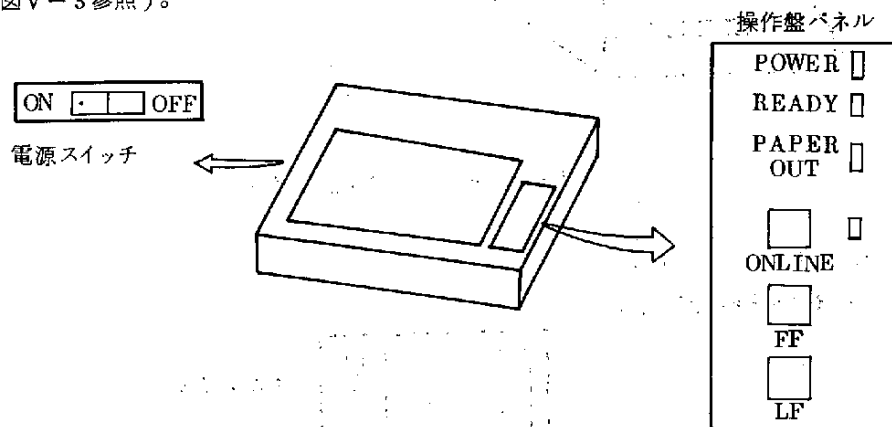
両スイッチはLED表示と一体になっており、電源投入はLEDの点灯によって確認する。



図V-2 DSC-80ZA電源スイッチ

2) プリンタ(MP-130)

プリンタ側面の電源スイッチを投入する。電源投入は、操作盤パネル上のLED表示で確認する(図V-3参照)。



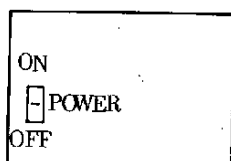
図V-3 MP-130スイッチおよび操作盤パネル

電源投入確認後、装置をオンライン状態にする。オンライン／オフラインの切換えは、操作盤上のONLINEスイッチで行なう。LEDの点灯で、オンライン状態の確認を行なう。

3) キャラクタ・ディスプレイ・ターミナル (DDY86-A)

DDY86-A 背面の電源スイッチを投入する。スイッチ投入後数秒で画面上にカーソルが表示されることで電源投入を確認する (図V-4 参照)。

背面図



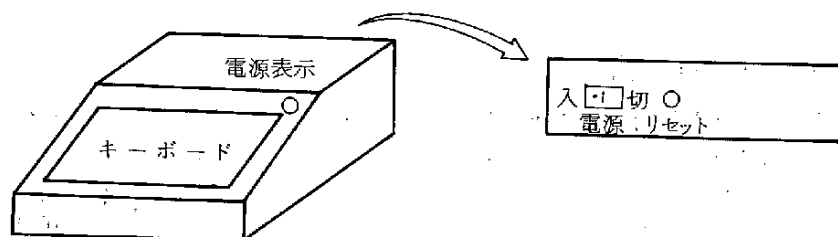
図V-4 DDY86-A 電源スイッチ

2.1.2 端末側

端末側全装置の電源投入方法を以下に示す。

1) PC-8001

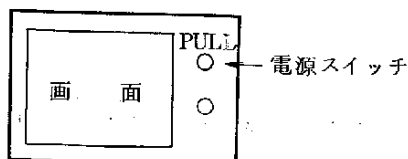
PC-8001 側面の電源スイッチを投入する。電源表示の点灯で電源投入の確認を行なう (図V-5 参照)。



図V-5 PC-8001 電源スイッチ

2) CRTディスプレイ

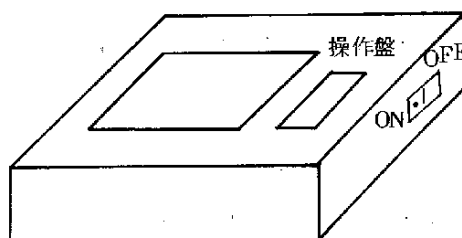
CRT前面の電源を投入する。



図V-6 CRTディスプレイ電源スイッチ

3) プリンタ (MP-80)

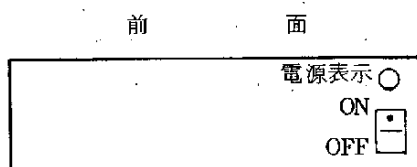
プリンタ側面の電源スイッチで電源投入を行なう。電源投入の確認およびオンライン指定は2.1.1(2)を参照。



図V-7 プリンタ電源スイッチ

4) I/Oユニット (PC-8011)

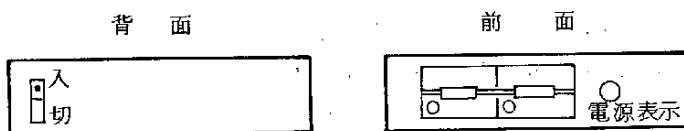
PC-8011前面の電源スイッチを投入する。電源投入は、電源表示の点灯で確認する (図V-8 参照)。



図V-8 PC-8011 電源スイッチ

5) ミニ・フロッピーディスク・ユニット (PC-8031)

PC-8031 背面の電源スイッチを投入する。前面の電源表示の点灯によって電源投入を確認する。(図V-9 参照)。



図V-9 PC-8031 電源スイッチ

2.2 プログラムの起動

ホスト側および端末側でのプログラムの起動方法を以下に示す。プログラムの起動は、まず端末側、続いてホスト側の順に実行する。

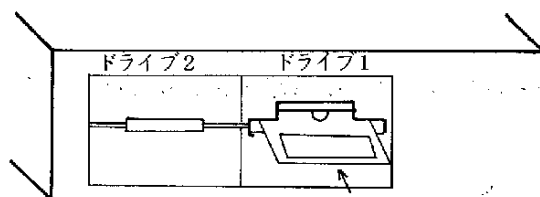
2.2.1 端末側プログラム

端末側全装置の電源が投入されていることを確認した後で、以下の手順に従ってプログラムの起動を行なう。

1) Disk Basicの起動

PC-8001 Disk Basic の起動を以下の手順で行なう。

- a) Disk Basic の格納されているミニ・フロッピーディスクをPC-8031 のドライブ1に挿入して、PC-8001 背面のリセット・スイッチを押下する(図V-5, V-10参照)。



図V-10 ミニ・フロッピーディスクの挿入

- b) Disk Basic が起動されると画面上に

DISK VERSION

How many files (0-15)?

と表示され、使用ファイル数を1と指示する(図V-11参照)。使用ファイル数入力以後、PC-8001 はコマンド入力待ち状態となる。

DISK VERSION

How many files(0-15)? 1CR^注

NEC PC-8001 BASIC Ver 1.1

Copyright 1979 (C) by Microsoft

OK

注：キーボードからの入力

図V-11 Disk Basic 起動時の画面表示

2) プログラムの起動

プログラムの格納されているミニ・フロッピーディスクをPC-8031のドライブ1に挿入し、図V-12の手順でプログラムを起動する。

```
OK
 $\frac{\text{moun} + 1 \text{ CR}}{\text{run "DEMO" CR}}$ 
(画面クリア)
*** Machine Language Loading ***
Load Complete
(画面クリア)
```

注：キーボードより入力する

図V-12 プログラム起動時の画面表示

起動後プログラムは、キー入力待ち状態になる。

2.2.2 ホスト側プログラム

ホスト側全装置の電源が投入されていることを確認した後で、以下の手順に従ってプログラムの起動を行なう。

1) CP/Mの起動

デモ・プログラムはCP/Mファイルとして登録されているため、デモ・プログラムの起動に先立って、CP/Mの起動を行なう。DSC-80ZAにおけるCP/Mの起動は、以下の手順に従う。

- 電源投入を確認する。
- 2～3秒後に“スペース”キーを入力する。キー入力を受付けられると、画面上に
DSCC SREIES II, V1.0
と表示される。
- CP/Mの格納されたフロッピーディスクをドライブ0へ挿入する(図V-13参照)。

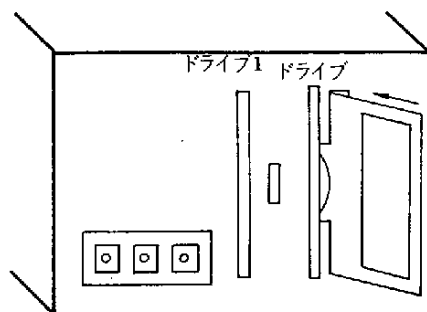


図 V-13 ディスクの挿入

- d) “S” キーを入力して、CP/Mをメモリ内にロードされる。

CP/Mは起動後画面上に

59K CP/M VERS 2.2

>

と表示する。

2) デモ・プログラムの起動

CP/M起動後、CP/Mコマンドを用いてデモ・プログラムを起動する。

以下の手順でデモ・プログラムを起動する。

- デモ・プログラムの登録されているフロッピーディスクをドライブ0に挿入する(図2-12参照)。
- CP/Mコマンドによりデモ・プログラムを起動する。

> DEMO CR

注

注：下線の文字をキー入力する。

コマンド入力後、デモ・プログラムがメモリ内にロードされ、実行が移される。

2.3 デモ・プログラム操作

ホスト側および端末側のプログラムの起動が完了した時点から、2章で示したデモ・プログラムの処理が実行可能となる。

各々の処理に対する操作方法を示す。

1) キー入力→CRT表示

ホスト側におけるキー入力は、DSC-80ZA前面パネル上のINTERRUPTスイッチを押下することで受付可能となる(図V-14参照)。スイッチ押下後、第1文字目のキー入力が“*”以外の場合(2.3.2参照)“Return”キー押下までの全文字をCRT画面上に表示する。

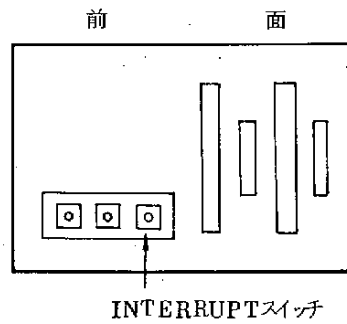


図 V-14 DSC-80ZA INTERRUPT スイッチ

2) キー入力→端末プリンタ

1)で示したように、キー入力はINTERRUPTスイッチ押下後に受け付けられる。第1文字の入力が“*”の場合、第2文字目以降“Return”キー入力までの全入力文字を端末側のプリンタに印字させる。

図 V-15.に、キー入力文字の転送の様子を示す。

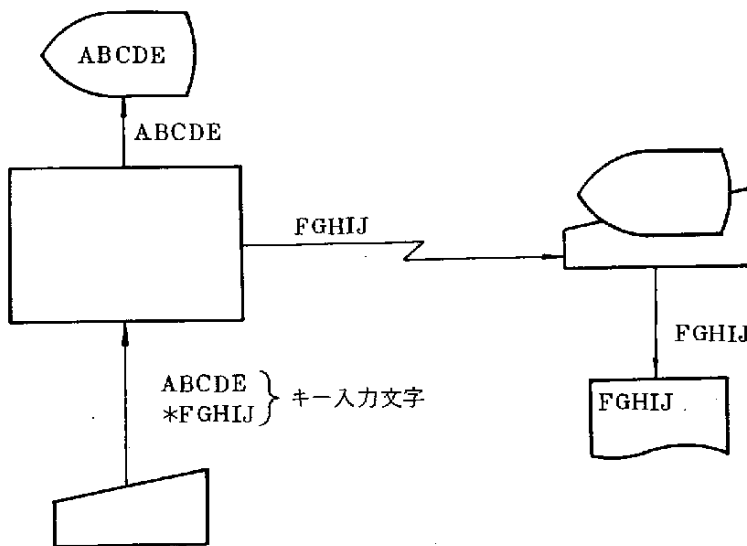


図 V-15 キー入力文字の流れ

3) 端末キー入力→プリンタ

端末側では、常にキー入力待ちの状態にある。そして、一行の先頭から終了 (Return キー入力まで一行となる) までの全文字をホスト側のプリンタに印字させる。図 V-16 にキー入力文字の流れを示す。

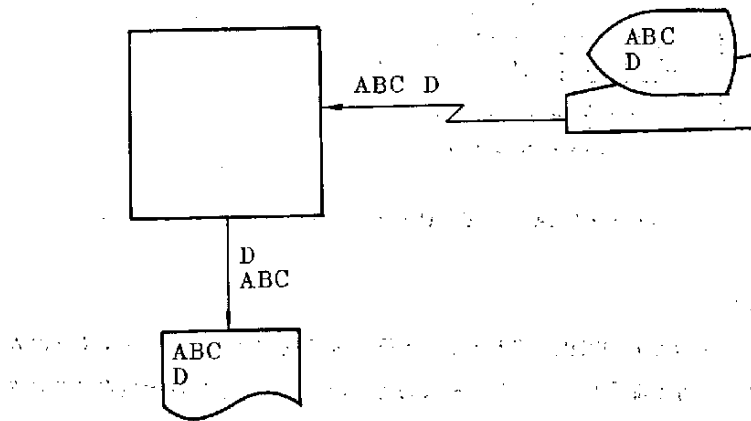


図 V-16 キー入力文字の流れ

—— 禁 無 断 転 載 ——

昭和 57 年 3 月 発行

発行所 財団法人 日本情報処理開発協会

東京都港区芝公園 3-5-8

機 械 振 興 会 館 内

TEL(434)8211(代表)

印刷所 株式会社 タケミ印刷

東京都千代田区神田司町 2-16

TEL(254)5840

原本 (持出厳禁)

受 付 No.	211-40
受付年月日	
作 成 部	