

機能分散型計算機における入出力サブシステムの試作

DISTRIBUTED I/O SUBSYSTEM

正井 一夫 田中英彦 元岡 達
 Kazuo MASAI Hidehiko TANAKA Tohru MOTO-OKA

東京大学 工学部

Tokyo University

1 はじめに

最近、新しい計算機アーキテクチャとして、機能分散型計算機がいろいろ考えられている。その一つとして図 1.1 に示すように三つのサブシステムよりなる計算機が考えられる。ここで各サブシステムは次のような機能を持つ。

- ・ファンクショナルサブシステムはもっぱら演算処理を行う部分である。
- ・メモリーサブシステムは、ファイル、仮想空間の管理など記憶に関する機能を受け持ち、ディスク、ドラム等の二次記憶はここで管理される。
- ・入出力サブシステムは、外界との入出力に関する機能を受け持ち、プリンタ、キーボード、等の周辺、端末を制御する。

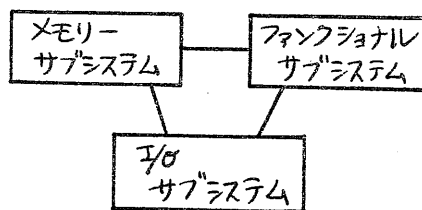


図 1.1 三つのサブシステム

又、最近ではマイクロプロセッサの発展も著しく、このような背景のもとに、従来のチャンネルの構成法を再検討し、入出力サブシステムを考えてみた。ここでは、入出力サブシステムをホスト計算機の下に実験システムとして試作することについて検討した結果を報告する。

2 入出力サブシステムの機能

入出力サブシステムは、基本的には外界との入出力の機能を受け持つが、その他に OS の一部機能を取込むことも考えられる。

2.1 データ入力と出力

入出力機器 (TTY、ディスク、通信回線等) とのデータの単なる受授を行うだけでなく、入出力機器のインテリジェント化を行う。例えば、TTY には、マルチプレクス、エディティング、エコー、書式制御などの機能をサポートする他、通信回線に対しては、フロントエンドプロセッサとして伝送制御、経路制御、パケットング、再結合などをサポートする。

2.2 OS の一部機能の代行

各入出力機器から来る多数の各種の割込は、本サブシステムで処理できるものはすべて処理し、その他の割込はその優先度の順に並べかえ一括してホストへ伝達する。これによってホストへの割込回数を減らすことができる。

又、タイマーを定期的に割込む疑似 I/O とみなし、本サブシステムに取込み、従来、CPU に一定間隔で発生していた割込をなくすることができる。更に、タスクをタイマーキューに登録したり、そのタスクを所定時刻に起動することも考えられる。

更に、本サブシステムでは、ジョブによって生成された各タスクの情報を受け取り、各々の優先度を考えて、どのタスクにどの程度のタイ

ムスライズを与えるかをすべて計算し、その結果をデータとしてホストプロセッサに渡し、タスク切替時のみ割込むことも考えられる。割込まれたときホストは、単にそのスケジューリングデータを見て、タスクの切替を行うだけでよい。このようにして、スケジューリングというOSの重要な機能を取込むこともでき、ホスト側からは、スケジューリングを行う専用ハードウェアに見え、ホストの負担をかなり軽減できる。

3. ハードウェア構成

3.1 構成とその特徴⁽¹⁾

現在のチャンネル形式の入出力は、一種の入出力サブシステムと考えることもできるが、より高度な処理を行なわせるためには、チャンネルの各部をプロセッサ化することが考えられる。すなわち、チャンネル制御装置の代りに高速の制御

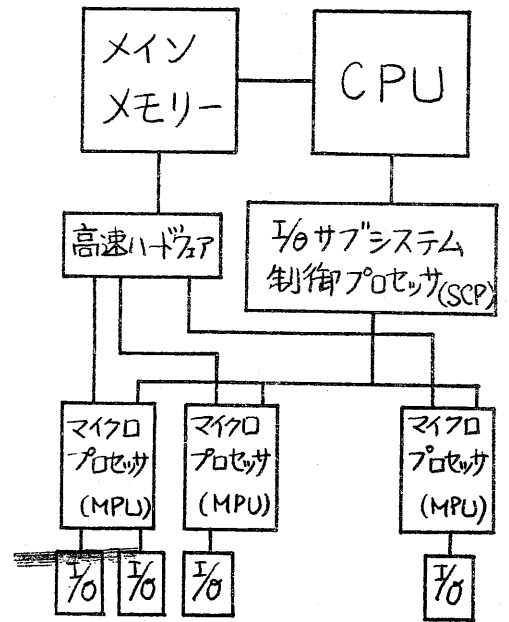


図3.1 ハードウェア構成

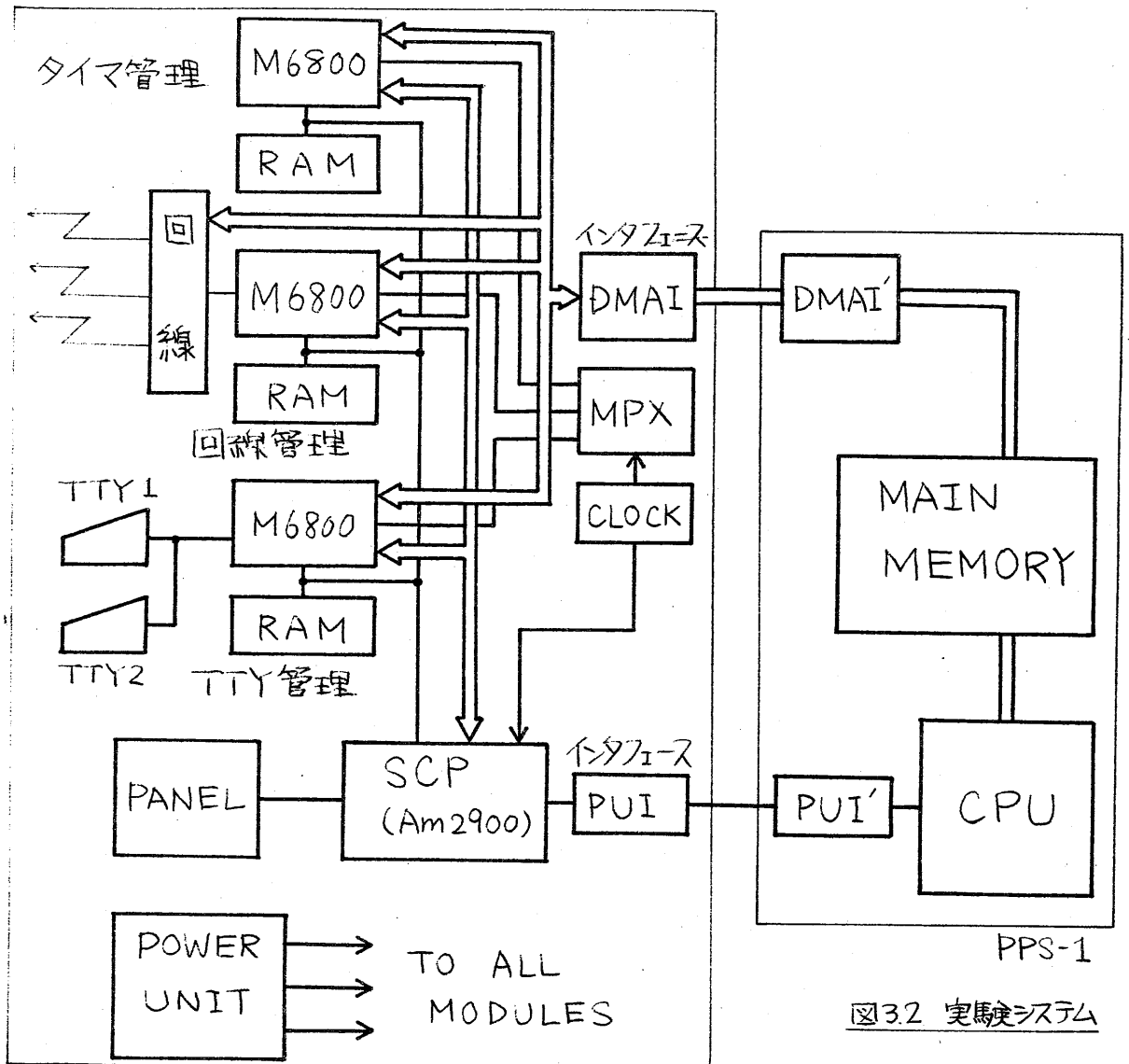


図3.2 実験システム

プロセッサを用い、各チャンネルとデバイスコントローラには、各々独立のマイクロプロセッサを用いる方法が考えられる。しかし、現在のマイクロプロセッサは処理速度の面で、ハードウェアモードロジックに1桁以上劣ってしまう。そこで、図3.1に示すように、高速性が要求されるDMA制御部とチャンネルマルチプレクサ部に対しては、従来通りのハードウェア構成とし、各チャンネル毎の制御には中～低速のマイクロプロセッサ(以後MPUと記す)を用い、各MPUを制御するために高速の制御プロセッサを(以下この部分を入出力サブシステム制御プロセッサと言い、SCPと記す)用いる形式を考えた。ただし、デバイスコントローラは、場合によってはそのまま使用してもよいと考えられる。

このようなシステム構成には次のような特徴が考えられる。

- (1) マイクロプロセッサを用いているので、エディット、書式制御などのインテリジェントな機能を容易に実現できる。
- (2) 入出力処理をマイクロプロセッサのソフトウェアで記述し、その入れ換えは容易なので、柔軟性、適応性に富んでいる。
- (3) 各I/Oクラスごとにマイクロプロセッサを用いてモジュール化しているので、拡張性に富んでいる。
- (4) 並列処理によって処理能力が向上し、比較的安価であるマイクロプロセッサを用いているので価格性能比も向上する。

このハードウェア構成を実現した実験システムは、図3.2に示すようになっている。この図において、MPXはマルチプレクサ、DMAIはDMAインタフェース(共に高速ハードウェア部)であり、PUIはプロセッサユニットインタフェースでSCPの一部である。また実験システムなので、SCPにはその内部状態がわかるようにパネルがつけられている。

以下各部の説明を行う。

3.2 高速ハードウェア部

高速ハードウェア部は、各I/Oを制御するMPUから来るDMA要求を順に受け付け、ホストのメインメモリーにDMAを行う

マルチプレクサ部は、図3.3に示すように、各MPUからのDMA要求信号をスキャンし、

要求を検出すると、そのMPUにDMA用バスの使用可信号を返し、DMA制御部へDMA開始指令を出す。するとDMA制御部は、バスに乗っているデータを見て、メモリーにDMAを行い、ステータスを返す。

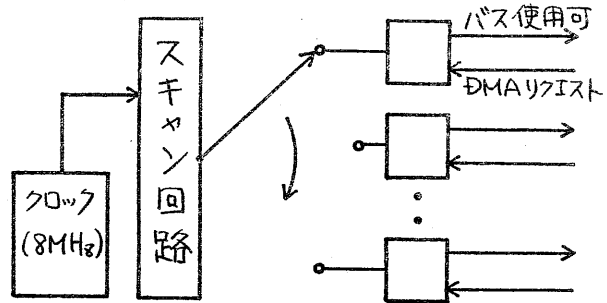


図3.3 マルチプレクサ

実際のマルチプレクサ 125 nsec に1回のスキャンを行っており、2回に1回必ずスキャンを行う高速用のポート1つと、その他の低速用のポート8つを持っている。ここで、メインメモリーのサイクルタイムは660 nsecなので、高速ポートは等価的にサイクルタイムは、約1 μsec、低速ポートは約3 μsec(共に、他のメモリー使用者がないとき)となる。高速ポートは本サブシステム以外で使用するためのものであるが、メモリーが16ビットなので、2メガバイト/秒というかなり高速な転送も可能となっている。

3.3 SCP (I/O Subsystem Control Processor)

3.3.1 SCPの構成

SCPはホストよりのコマンドを各MPUに伝達したり、各MPUより来る割込をホストへととりまとめて伝達するMPU管理用のプロセッサである。本プロセッサはマイクロプログラム方式のプロセッサであり、それ自身の持つRAMに蓄えられている機械語をマイクロ命令で実現して走っている。

SCPの構成は、図3.4に示すようになっている。プロセッシング部にはAMD社のAm2901を、シーケンサ部にはAm2909を用いている⁽²⁾。ここで、Am2901は4ビットスライスのALUと16個のレジスタを持っている。又、Am2909は、マイクロプログラムカウンタ、スタック、レジスタ、ダイレクト

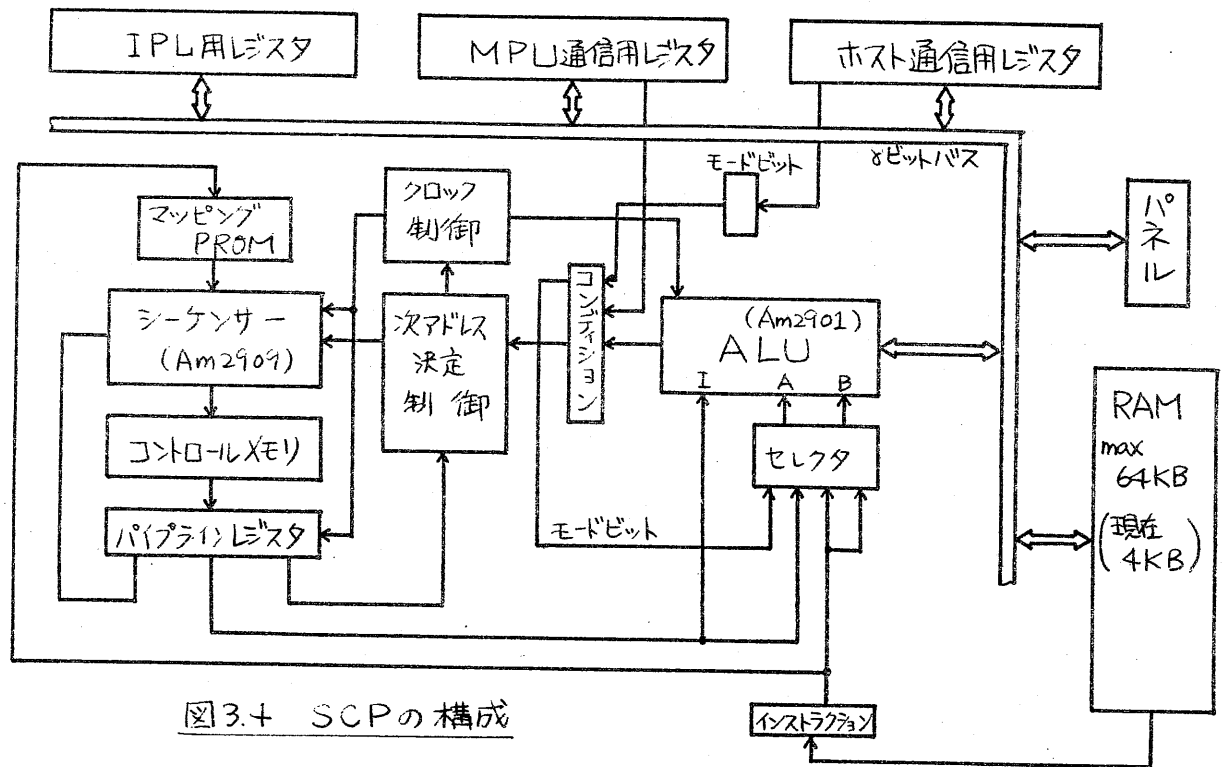


図3.4 SCPの構成

インプット端子を持ち、この4種類よりマイクロ命令のアドレスが選択できるようになっている。SCPでは、これらのLSIを用い、8ビットのALUと12ビットのシーケンサーを構成している。

シーケンシングとALU処理をパイプライン化するためにコントロールメモリ（以下CMと記す）の出力はパイプラインレジスタに蓄えられるが、SCPには他にレジスタとして、内部レジスタ（Am2901内）と外部レジスタが存在する。内部レジスタは演算用であるが、外部レジスタは各々機能を持ち、次のようなものがある。

- (1) MPUへのIPLを行うためのもの
アドレス・データレジスタ、IPLセレクトレジスタなどがある。
- (2) MPUと通信を行うためのもの
MPUを選択するためのレジスタ、データレジスタ、割込レジスタなどがある。
- (3) ホストとの通信を行うためのもの
バッファレジスタ、ステータスレジスタなどがある。
- (4) RAMを使用するためのもの
メモリアドレスレジスタとメモリアドレスレジスタがある。

(5) その他

パネルとのインタフェースをとるレジスタ、機械語のためのインストラクションレジスタなどがある。

これらのレジスタは図3.4に示すように、ほとんどSCP内の同じバスに接続されている。

3.3.2 ホスト、MPUとの通信

(1) MPUとの通信

MPUとSCPの間には図3.5に示すように、8ビットのバスと各MPUへの割込線があり、MPUがSCPへデータを転送したい場合には、MPUはデータ出力レジスタにセットすればよい。その際MSBをゼロにセットするが、そのことによってSCPのスクアンレジスタのMPU番号に対応する位置の1ビットがセットされ、SCPへの割込が発生する。SCPは、MPUセレクトレジスタを用いてバスを切換えて、要求元のデータ出力レジスタの内容を読み転送は終了する。ただし、転送は2度に分けて、2バイト行う。

SCPがMPUへデータを転送したい場合には、SCPはMPUセレクトレジスタを用いて目的の相手のレジスタへデータを2度に分けて書き込む。その後、割込レジスタの相手のMPUに対応するビットをセットすることによって、

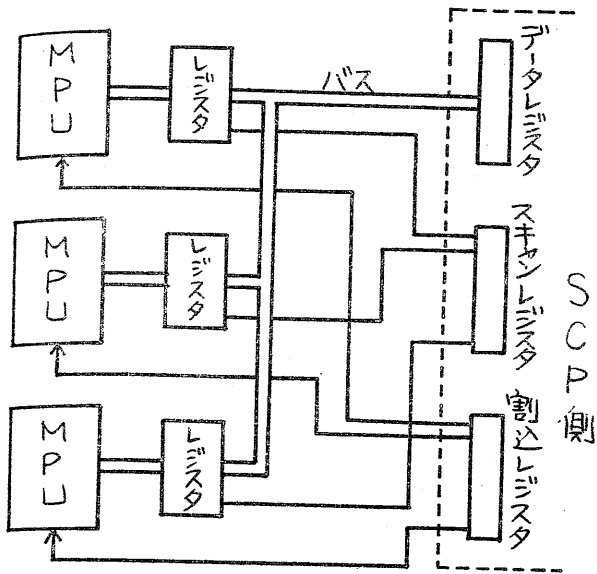


図3.5 各MPUとの通信

そのMPUに割込を発生させる。MPUは割込を受けたらそのデータを受け取り、転送は終了する。

このような通信は、バスの使用権に関する制御をすべて高速側のSCPが行い、一回の通信は2バイトで十分なので、簡単なハードウェアであるが、能率よく転送が行える。

(2) ホストとの通信

SCPとホストの間には、図3.6に示すようにバッファレジスタ(8ビット)とステータスレジスタがあり、ホストからSCPへデータを送る場合には、ホストはI/Oを起動する手順を経てバッファレジスタにデータを出力する。このことによってSCPに対する割込が発生し、SCPはバッファレジスタを読みに行く。このとき、さらに連続してデータを転送する時には、お互いにステータスレジスタをソフトウェアで確かめねばならない。すなわち、ホストがレジスタにデータを書き込むと、ステータスビットがセットされ、これを見てSCPはデータを取込む。SCPがデータを取込むと、ステータスビットがリセットされ、これを見てホストはさらに転送を継続できる。

SCPからホストへデータを送る場合には、SCPは割込レジスタのホストに対応するビットをセットし、ホストへ割込を起す。その後、ホストから割込受付信号が返ってくるまで一般に時間があるので、他の処理を実行する。ホス

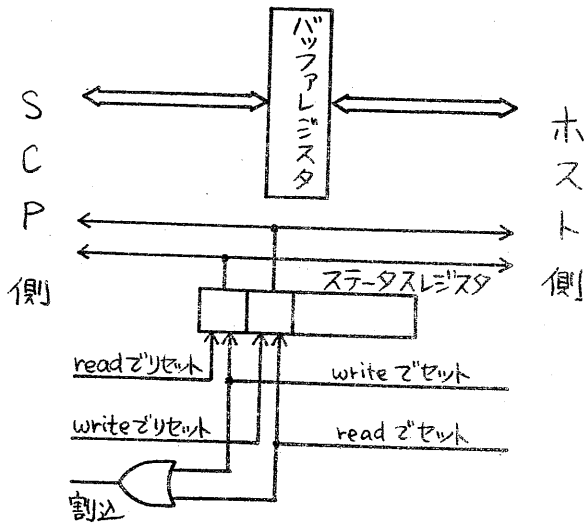


図3.6 ホストとの通信

トからの割込受付信号は、SCPに対して割込を発生し、これによって転送が始まる。連続して転送する場合、ステータスビットを見て同期をとるのは、ホストからSCPへ転送する場合と同様である。

3.3.3 マイクロ命令

SCPの1マイクロ命令は24ビットで、図3.7に示すように、ALU命令とBRANCH命令がある。

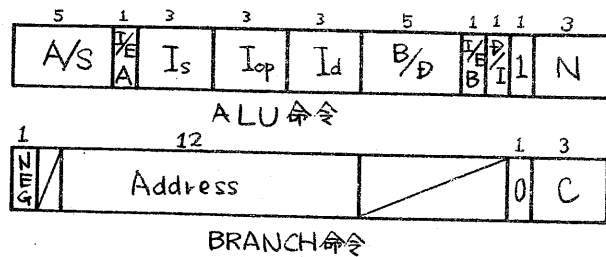


図3.7 マイクロ命令

(1) ALU命令

ALU命令は、Am2901のALUを使用する命令で、各フィールドは次のような意味を持つ。

- <A/Sフィールド、I/E Aビット>：ソースとなるレジスタの指定
- <B/Dフィールド、I/E Bビット>：デスティネーションとなるレジスタの指定
- <Is, Iop, Idフィールド>：ALUの機能を決定する⁽²⁾
- <Nフィールド>：次に実行するアドレスの決定法を指定する。指定には、プログラムカウンタ

ンタ、スタック、マッピングPROMによるものなど8種類が存在する。

〈D/Iビット〉：D/I = 1のとき内部レジスタはマイクロ命令によって指定されたものとなり、D/I = 0のときは、機械語によって指定されたものとなる。機械語によって指定した場合は、Nフィールドの値によってさらに2種類に分かれる。(3.3.4(3)参照)

(2) BRANCH 命令

この命令はALUを使わない命令で、この場合、ALUは動作しない。各フィールドは次のような意味を持つ。

〈Address フィールド〉：ブランチを行う場合のアドレスとなる。

〈C フィールド〉：ブランチの種類を指定するフィールドで、ALUのステータスやホストからの要求の有無などによるコンディショナルブランチや、サブルーチンコールなど8種類指定できる。

〈NEG ビット〉：コンディションを否定するかどうかを決める。

NEG = 1 : 条件が成立したときブランチ

NEG = 0 : 条件が成立しなかったときブランチ

3.3.4 機械語のためのハードウェア

SCPはマイクロプログラムによって機械語を設定するが、機械語を実現することを容易にするために、いくつかのハードウェア機能がある。これには次のようなものがある。

(1) 割込機能

SCPはマイクロレベルに対する割込機能を持たないが、MPUもしくはホストより要求のある場合には、各々コンディションコードの対応するビットがセットされ、割込が存在することが1マイクロ命令で検出できるようになっている。これによって機械語を実現する場合には、自動割込機能(インタラプト)を容易に実現できる。又、ここでホストよりの要求に対する処理を行うモードをホストモード、その他の場合をMPUモードと呼ぶ。したがって、ホストの要求を示すビットはモードを示すビットでもある。

(2) インストラクションレジスタとマッピングPROM
SCPには、図3.4に示したように、メモリーより直接セットされるインストラクションレジスタと言うバスには無関係なレジスタが存在し、その出力はマッピングPROMによって、CM上のアドレス(12ビット)に変換される。これによって、機械語を実現する各マイクロプログラムへのブランチが1サイクルタイムで可能となる。

マッピングは、図3.8に示すように行うが、これは機械語の上位5ビットがオペレーション部で、次の3ビットは補助部であることに対応している。

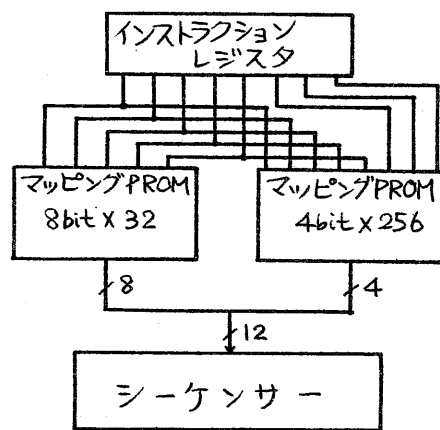


図3.8 マッピングPROM

(3) 内部レジスタ選択機構

内部レジスタは16個存在し、その選択は一般にはマイクロ命令の1フィールドを使用するが、機械語で内部レジスタを使用する場合のために、インストラクションレジスタによって内部レジスタを選択することが出来る。この選択の方法には、I₁アドレッシングとI₂アドレッシングの2つの方法があり、マイクロ命令で指定できる。

I₁アドレッシングは図3.9(a)に示すように、モードビットとインストラクションレジスタの下位3ビットを連結してアドレスとする方法で、16個の内部レジスタを8個ずつの2組に分け、モードによって使いわけることを可能としている。これによって、機械語から見ると2つのモードが存在し、モードの切換によってレジスタも自動的に切り換わると考えられる。

I₂アドレッシングは図3.9(b)に示すように、I

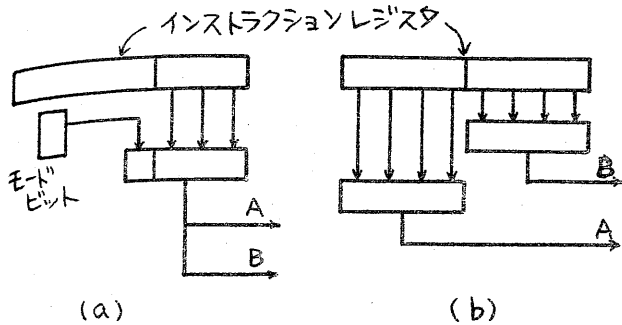


図3.9 内部レジスタ選択機構

インストラクションレジスタの内容全体で、内部レジスタのAポート（ソースとして使う）、Bポート（デスティネーションとして使う）の両方を別々にアドレスする方法で、機械語でレジスタレジスタオペレーションに近い命令を実現できるようになる。

3.3.5 機械語

SCPでは先に述べたマイクロ命令を用いて機械語を設定するが、SCPの機械語にはかなりマクロ的な命令が多く、SCPの基本機能の記述を容易に行える。しかし、このような専用の機械語の他に柔軟性を持たせるために汎用機械語も用意している。

機械語はマッピングPROMを用いてデコードするので、図3.10に示すように上位5ビットがオペレーションフィールドとなり、次の3ビットは、普通、内部レジスタの指定に用いる。2バイト目以下は、RAM上のアドレス、又はその機械語に必要な情報を指定するオペランドとして用いる。

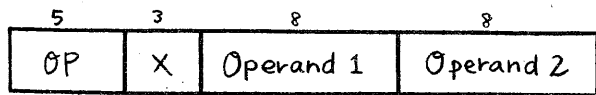


図3.10 機械語のフォーマット(例)

さて、専用の機械語としては次のような種類があり、各々数個の命令が存在する。

(1) 割込リスト処理

SCPでは各MPUよりホストへ伝えるべき割込を優先度順に並べて、一括してホストへ転送する。この時、MPUからの割込は、図3.11に示すようなリスト構造にして蓄える。そこで、リスト処理用に次のような機械語を用意する。

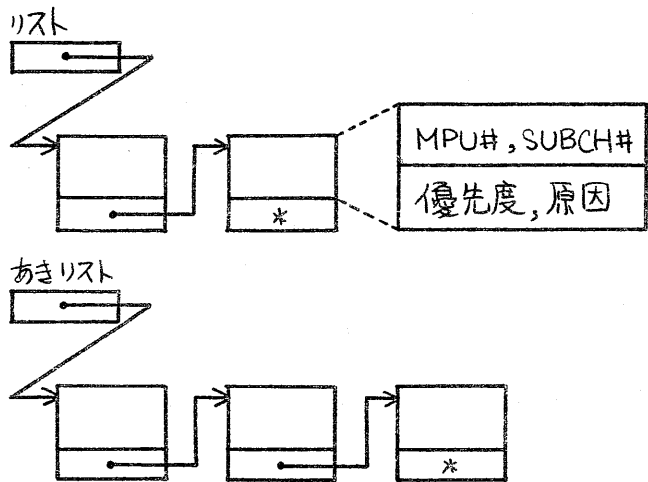


図3.11 割込リスト

Insert List : RAM上に存在する要素を、優先度順に割込リストに挿入する。

Initialize List : あきリストを、指定した数だけ作り、割込リストのイニシャライズを行う。

Delete List : 現在のリストをすべてあきリストにつなぎ換えて、リストを消去する。

一般のリスト処理を行うには、この他にも必要な処理があるが、ここでは(専用機械語なので)設けていない。

(2) 管理用テーブル処理

SCPでは各MPUのサブチャネルの状態を管理するために、図3.12に示すようなテーブルを用いるが、このテーブルに対する処理用には、次のような機械語を用意している。

MPU# SUBCH#	ステータス
⋮	⋮
-1	-1

図3.12 管理テーブル

Search and Read : レジスタに指定された、(MPU#, SUBCH#)をキーとして、管理テーブルをサーチし、それに対応するステータスをレジスタにもどす。もしキーが存在しないときは、レジスタに-1を入れる。

Write into Table : サーチの結果、読み出したステータスの所へ、レジスタに指定された内容を格納(上書き)する。

(3) データ転送

3.3.2 で述べたようにSCPはMPU、ホストと通信を行うが、その手続をすべて見えなくした次のような機械語を用意する。

Read from Host: RAMの指定されたアドレスへ、レジスタによって指定されたバイト数だけホストよりデータを受け取る。

Write to Host: RAMの指定されたアドレスより、レジスタによって指定されたバイト数だけホストへデータを送る。

Send List: 割込リストをホストへ転送する。

Read from MPU: レジスタによって指定されたMPUからデータを2バイト受け取り、RAMに指定されたアドレスへ格納する。

Write to MPU: レジスタによって指定されたMPUへ、RAMの指定されたアドレスのデータを2バイト転送する。

IPL to MPU: RAMの指定されたアドレスに存在する図3.13に示すようなIPL用データを、指定されたMPUのRAMに、IPLルートを用いて直接格納する。

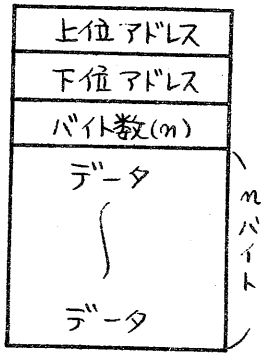


図3.13 IPLデータ

(4) その他

Interrupt to Host: ホストへ割込を起こす。ただし、レジスタの値がゼロのときには割込を起こさない。

Select one MPU: 多数のMPUからの要求から1つを選ぶ。(レジスタの1のビットを1つ選び、その1をリセットする。)

Table Branch: テーブルを参照してブランチを行う。これは、ホストより来るコマンドのデコードに用いる。

汎用命令は、普通のマイクロプロセッサほど多くはないが、基本的なものはあり、次のようなものがある。

- (1) Load , (2) Store , (3) Shift
- (4) Jump , (5) Skip on Condition
- (6) Reset Mode and Return

(7) Interrupt enable , (8) Interrupt disable
(9) Wait , (10) Halt , (11) No Operation
又、マイクロ命令に似た機能を持つ機械語として次のようなものもある。

(12) レジスタ・レジスタオペレーション: 2つのレジスタAとBをALUで演算して、またレジスタBにしまう

(13) 外部レジスタLoad, Store: 外部レジスタへデータをStoreしたり、外部レジスタよりLoadする。

これらの命令によって、現在機械語として実装されていない機能で実現できないものは、ほとんどないと思われる。

3.4 MPU

MPUは先にも述べたように入出力を直接制御する部分で、入出力のインテリジェント化を行う部分でもある。

3.4.1 MPUの構成

MPUは、Motorola社のM6800⁽³⁾を中心として構成され、図3.14に示すように、ハードウェアは、IPL制御部分、ホストへのDMA制御部分、SCPとの通信制御部分と入出力機器制御部分に分けられ、すべてM6800のバスに結合されている。

IPL制御部はM6800を停止させ、そのバスをすべてIPLに開放させる機構で、DMA制御部分、SCPとの通信制御部分には、転送するデータを蓄えるレジスタと若干の制御用ハ

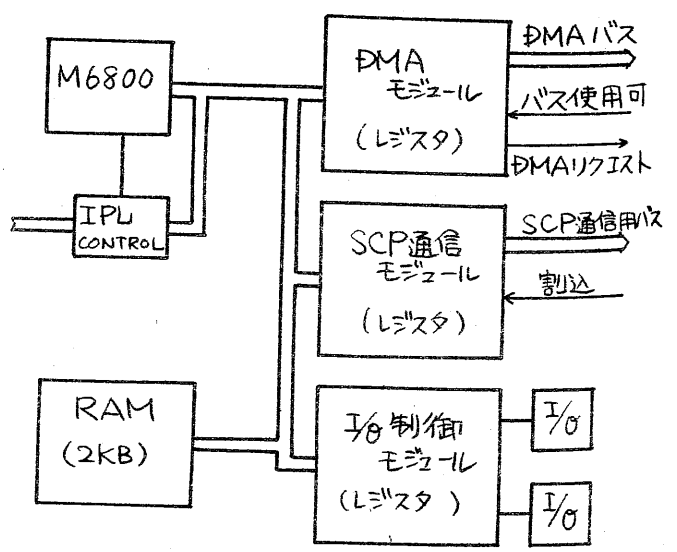


図3.14 MPUの構成

ードウェアとがある。

入出力制御部分は、制御すべき入出力機器によって異なり、ここではMPU内のソフトウェアではサポートしきれない部分をハードウェアでサポートする形になっている。

3.4.2 ホストへのDMA

各MPUは、高速ハードウェア部の助けを借りて、ホストのメインメモリを自由に参照、書き込みができるようになっている。

DMAを行う場合MPUは、まずアドレスレジスタにメインメモリのアドレスを、コントロールレジスタにメインメモリに対するコマンド(Read、Write、Test & Set等)を、又、書き込み時にはデータレジスタにデータをそれぞれセットし、次に、タグレジスタを用いてDMAリクエスト信号をオンにする。この信号は、マルチプレクサ(高速ハードウェア部)へ伝達され、マルチプレクサでは先に述べたように、そのリクエスト信号を順に受け、MPUに対して、DMA用バス使用可信号を返すとともにDMA制御部(高速ハードウェア部)へDMA開始指令を出す。DMAが終了し、終了信号が返って来たら、それをMPUはセンスによって知り、リクエスト信号をオフにし、メモリに対するDMAサイクルは終了する。

このようにして、どのMPUもメインメモリを使用することができ、もともとMPUは処理速度が速くないので、DMAは十分高速と考えられる。

3.5 ホスト計算機 PPS-1⁽⁴⁾

本入出力サブシステムは、ホストの下に結合されるが、そのホストとしては、我々の研究室にあるPPS-1というポリプロセッサシステムを用いる。

PPS-1の構成は図3.15に示すようになっており、3台のプロセッサが主記憶を共有している。又、各プロセッサはI/Oスイッチを介して入出力と結合されている。そこで、その部分を本サブシステムにつなぎ換え、メインメモリにつながっているBFC(Basic File Channel)の下にあるDMA用ポートを本サブシステムのDMAバスに結合して、本サブシステムをPPS-1の下に結合する。

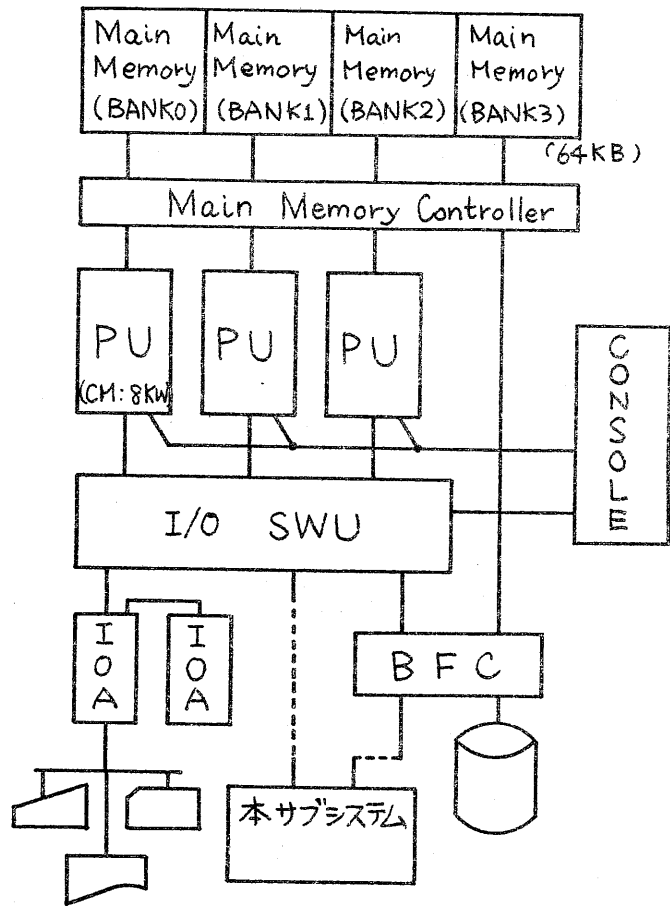


図3.15 PPS-1の構成

3.6 IPL機構

本入出力サブシステムのプログラムは、すべてホスト側から送られ、ロードされる。本システムのIPLは、まず最初にSCPへIPLを行い、次にSCP経由で、各MPUへIPLを行う。

(1) SCPへのIPL

ホストからマスターリセット信号がSCPに送られて来ると、SCPの各部のステータスはクリアされ、SCPはその後CMの0番地より走り出す。ここには、SCP自身のRAMにIPLを行うマイクロプログラムがROM内に置かれており、ホストより送られて来るプログラムをRAMに格納する。(CMには、機械語を実現するためのマイクロプログラムと、SCPへのIPLを実現するためのマイクロプログラム、及びデバッグ用のマイクロプログラムが収められている。)

(2) MPUへのIPL

SCPへのIPLが完了すると、SCPの制御はいまロードされたプログラムの方へ移され、ホストからのIPL指令によって、MPUへのIPLが開始される。SCPからMPUのRAMにプログラムを格納するには、そのために設けたIPLルートを用いる。これは図3.16に示すように、IPLセレクタレジスタによって選択した相手のMPUを停止させ、そのバスの開放させ、その後IPLアドレス・データレジスタに値をセットすることによって、相手のRAMに直接格納するルートがある。プログラムの格納が終了したら、そのMPUにリセット信号を渡し、停止を解除し、MPUへのIPLは終了する。

このように、本サブシステムでは、ホストよりまずこのプログラムをロードするのど、プログラムを変化させることは容易であり、又、各MPUにROMをおく必要はない。

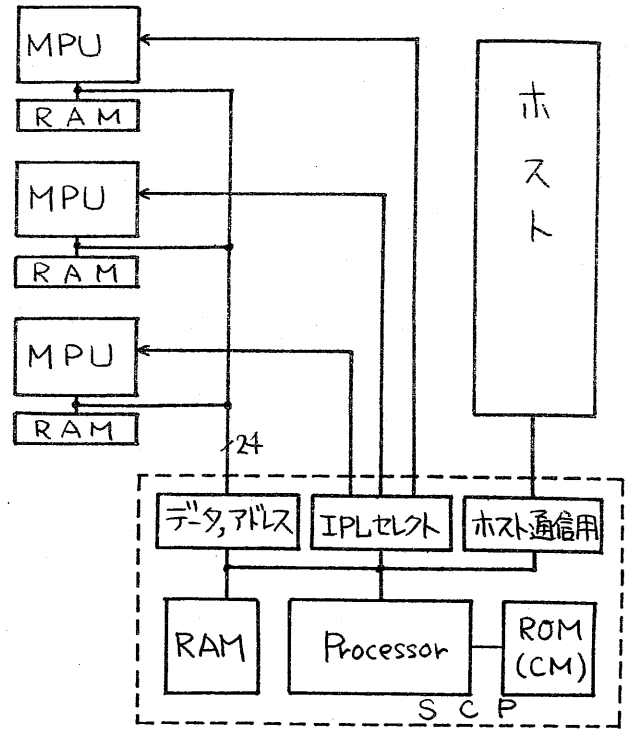


図3.16 各MPUへのIPL

3.7 各部の実装

本サブシステムは、エプレン社の ebr-400/UC56¹¹ というラッピングボード（以下単にボードと記す）を用いて構成されている。以下、各部のボード構成について述べる。

高速ハードウェア部は3ボードよりなり、各ボードのIC数を表1に示す。この部分は、高速転が要求されるため、ほとんどショットキー

TTLによって構成されている。

SCPは、AMD社の Am2900ファミリーを用いていて、ホスト計算機 PPS-1 とのプログラムバスインタフェースを含め、8ボードよりなる。各ボードのIC数は、表2に示すようになり、約190個のIC（うちLSIは37個）から構成され、クロックとしては8MHz（サイクルタイム4250nsec）を用いるため、

ボード名	機能	IC数
MPX	マルチプレクサ	10
DMA-1	DMAバスのドライバー/レシーバー	12
DMA-2	DMA制御回路	31

表1. 高速ハードウェア部ボード構成

ボード名	機能	IC数(うちLSI数)
AM-1	ALU (Am2901)、シーケンサー (Am2909)	28 (5)
AM-2	CM (24ビットX1Kワード, アクセス: 70nsec)	40 (PROM: 24)
AM-3	RAM (4Kバイト, アクセス: 250nsec)	21 (RAM: 8)
AM-4	PPS-1とのインタフェースレジスタ	25
AM-5	MPUに対するIPLレジスタ	21
AM-6	MPUとの通信用レジスタ	16
AM-7	クロック制御回路 (ホールド・ストップコントロール)	16
PUI	PPS-1とのインタフェース制御部	22

表2. SCPボード構成

ボード名	機能	IC数(うちLSI数)
MPU-1	M6800とIPLを行うための制御回路	20 (1)
MPU-2	PPS-1へのDMA用レジスタ	28 (4)
MPU-3	SCPとの通信用レジスタ	22 (3)
MPU-4	入出力機器とのインタフェース	30 (6)
RAMボード	メモリー(Motorola社製) 2KB	27 (RAM:16)

表3. MPUボード構成

バス名	使用目的	信号線数(うちタグ線)
DMA用バス	MPUがPPS-1とDMAを行う	48 (2)
MPU通信用バス	MPUとSCPとが通信を行う	20 (4)
IPL用バス	SCPが各MPUへIPLを行う	27 (3)
プログラムバス	SCPとPPS-1のPUとが通信を行う	19 (6)

表4. バス構成

やはりかなりの部分はショットキーTTLである。

最後に、MPUはMotorola社のM6800を中心にこのファミリーのICを用いて、4ボードとRAMボード(メモリーボード)で1組となり、各ボードのIC数を表3に示す。このMPUはTTYを3台制御するMPUの例で、ボードMPU-4にはデバッグ用ROM&RAMもあり複雑となっているが、約130個(うちLSIは30個)で構成されていて、ここではMOS-ICと普通のTTLを用いている。

本サブシステムでは、多数のバスを用いているが、これらのバスについて信号線数をまとめたのが表4であり、これからも分かるように、信号線数はかなり多い。これは、SCPとMPUの速度差(10:1)が与える能率の悪化をできるだけ少なくする為にMPUの信号はできるだけ並列化したことによる。

4 入出力サブシステムの動作

4.1 ホストより見た本サブシステム

ホストより本サブシステムを見ると、チャンネルと同様に見える。したがって、主記憶に図4.1に示すように、MPUの数だけのCAWベクターを用意する。各CAWベクターの要素は、対応するMPUのCCB(Channel Control Block)へのポイントとなっていて、CCBは、図4.2に示すように各サブチャンネルに対応するCCWよりなる。CCWはまた、チェイニング

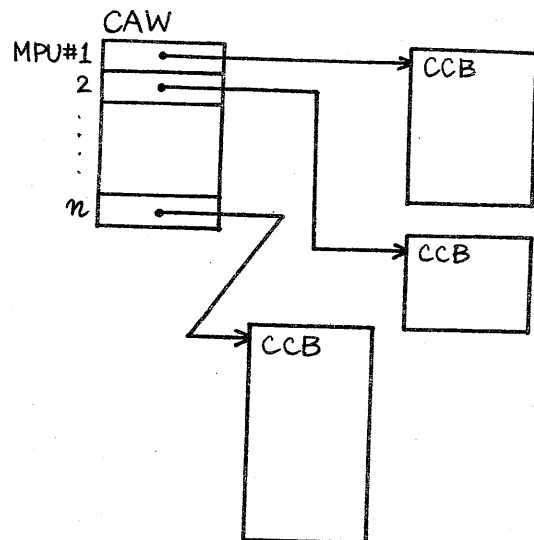


図4.1 チャンネルアドレスワード

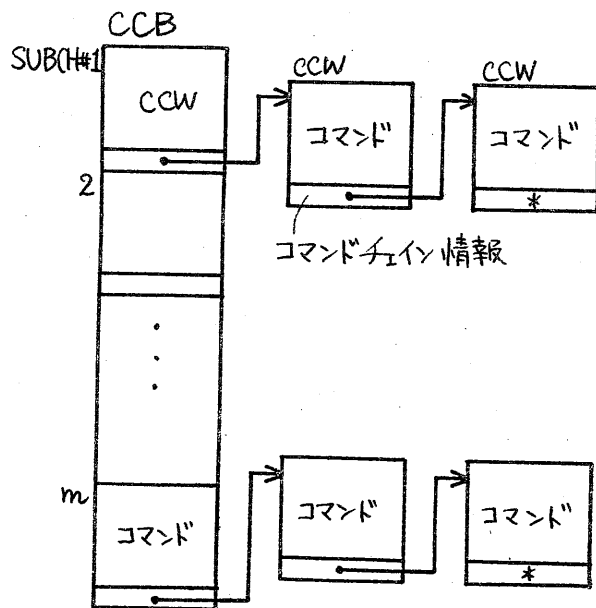


図4.2 チャンネルコントロールブロック

されてリスト構造をしていてもよく、各CCWにはコマンドが格納される。ただし、そのサイズはMPUごとに決まっていれば一定ではなく、CCBの大きさも一定ではない。

ホストからSCPを経由して各MPUに伝わるコマンドには次の五種類がある。

- (1) START I/O (MPU#, SUBCH#)
指定したMPU中のサブチャンネルに対して起動をかける。
- (2) HALT I/O (MPU#, SUBCH#)
指定したMPU中のサブチャンネルの入出力を停止させる。
- (3) SENSE I/O (MPU#, SUBCH#)
指定したMPU中のサブチャンネルのステータスを読む。
- (4) TEST CH (MPU#, SUBCH#)
指定したMPU中のサブチャンネルに、その詳しい状態を主記憶へ報告させる。
- (5) IPL (MPU#)
指定したMPUへIPLを行う。

これらのコマンドを出す時、その後SCPから割込が発生するが、同時に多数の割込要求が転送されて来るので、それらをすべて解釈して処理する必要があるのである。

4.2 動作例

本サブシステムがホストに依頼された仕事をする様子を、TTYへのデータ出力の例と、タスクのタイマMPUによる切替の例によって示す。

(1) ホストからTTYへのデータ出力

ホストはまずTTY用のCCWを設け、その中に出力データへのポインタと、出力であることを記述し、SCPにスタートI/Oを伝えるように依頼する。SCPは、指定されたMPUのステータスをチェックした後、MPUへスタートI/Oを伝達する。

MPUは起動がかかると、ホストのメインメモリーのCCWからCCBをたどり、TTYへのデータ出力を開始する。出力が終了したら、メモリーにステータス(正常終了、エラー等)を書き込み、SCPへ終了を知らせる。

SCPは、この終了事象をホストへの割込リストに挿入し、ホストへ適当な時期に割込む。

ホストに割込が受け付けられるまでに、さらに他のMPUよりの終了事象が発生した時には、その事象も割込リストに挿入し、割込が受け付けられたら優先度順に知らせる。ホストはこれによって、TTYへの出力が終了したことを知る。

(2) タイマMPUによるタスクの切替

タイマ用のMPUは、最初にホストによってSCP経由で起動されると、タスクに関する情報を受け取り、以後それを管理する。すなわち、現在ホストの上で実行されているタスクのタイムスライスが満了し、タスクの切替が必要な時には、次に走るタスクとそのタイムスライスを決定した上、ホストへSCP経由で割込む。ホストはその割込によって、スケジュールされたタスクへの切替を行う。

5 おわりに

入出力サブシステムを用いる機能分散システムは、入出力をインテリジェント化する他、割込を簡単にすることなどで、かなり他のサブシステムに対する負荷を軽減できると期待できる。今後は、実際の評価を行うとともに、他のサブシステムとの関係、全体のシステム構成、OSの分割問題等を更に検討する必要があると思われる。

現在、ハードウェアは7~8割完成し、各部のテストも順調に進んでいて、今年中には動作する予定である。

[参考文献]

- (1) 正井一夫、田中英彦、元岡達、「機能分散型I/Oサブシステムの一構成法について」、昭和52年度電子通信学会情報・半導体部門全国大会 No. 304
- (2) "The Am2900 Family Data Book" Advanced Micro Devices, Inc.
- (3) "M6800 Microprocessor Applications Manual" Motorola Inc.
- (4) 元岡達、山室良夫、「ポリプロセッサ・システムPPS-1」情報処理, Vol 15, No. 7 1974. 7月