

# ハードウェアモニタを使用したデュアルシステムの試験

阿江 勉, 安川克彦, 各藤 薫 (日本電信電話公社 データ通信本部)  
三輪康祐, 夜川峰晴 (日本電気)

## 1. はじめに

コンピュータシステムの性能評価を行うためのデータ収集手段としては、一般に数学的手法、シミュレーション手法、マニュアルによる計測、測定用ツールをソフトウェアに直接組み込む方法、及びハードウェアモニタを使用する方法等がある。上述の色々な方法により、データを収集し、評価を行ってシステムの改善に役立てているが、このうちハードウェアモニタを使用する方法は、次に示す点で他の方法より優れている。

- ① 被測定システムに外乱を与えない。
- ② 被測定システムのプログラムを改変する必要がないため、ソフトウェアのオーバーヘッドに影響を与えない。
- ③ 微妙なタイミングの測定が可能で、しかも、被測定システムの動作状態 (Run - Wait - Stop) とは非同期に測定できる。

本稿では、データ収集手段としてハードウェアモニタを用い、システムの改善及び処理能力の評価に効果を上げたケースとして、次の2項目を紹介する。

- ① システムの状態遷移処理の評価。
- ② タスクプライオリティの評価。

## 2. 評価対象システム

評価の対象とするシステムは、クロック同期等のハードウェア的同期方法のこれない汎用計算機を用いて、ソフトウェア的に並列運転を可能にする「主従型同期方式」 (Master-slave dual system) をとっている。この方式は、図 2.1 で示すように両 CPU で同一処理を並列に行っているが、ターミナルへの入出力処理、及び同期処理の制御は 1 台の CPU のみで行い、処理能力の向上を計っているものである。

ここでは、対象システムにおける 2 台の CPU による運転状態の遷移、及び業務処理を行うためのタスク構成について以下に述べる。

### 2.1 CPU の状態遷移

本システムは、2 台の CPU による並列運転を行っており、その運転形態をモードとステータスを区別している。運転モードとは、システムが何台の CPU でオンライン運転が行われているかを定義するもので、これには、デュアル、シングル、及びダウンの 3 種類がある。また、運転ステータスとは、CPU の処理内容によるソフトウェア上の区別で、システムの処理に対する主導権を持っている CPU をマスタ、マスタ CPU の制御下でデュアル運転に必要な各種処理をする CPU をスレーブと呼ぶ。

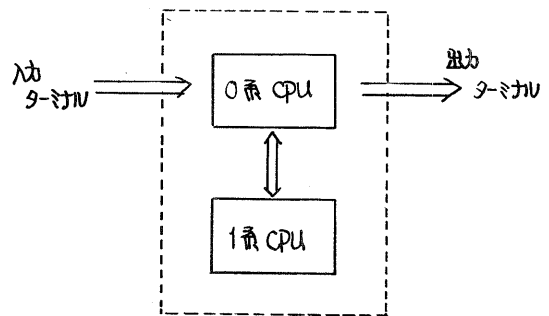


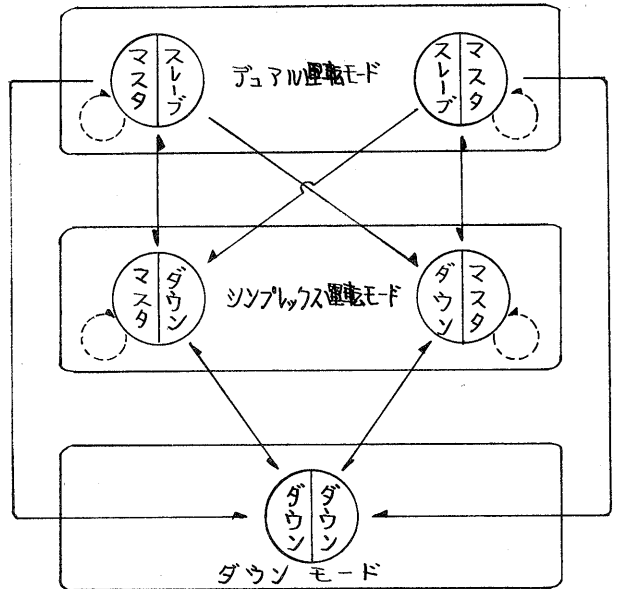
図 2.1. 対象システム概念図

システムが処理を行う上でこの運転モードの遷移図を図 2.2 に示す。本システムでは、1つの安定状態にある運転モードから、他の運転モードへの移行が不可欠な障害に起因する場合、ハードウェアの自動監視制御部(ASC)によりダウンした CPU に自動的に IPL (Initial Program Load) をかけ、元の運転モードに復帰させている。この運転モードの移行時には、一時的に不安定な状態が生じるが、運用への影響を少なくするため短時間で移行されるように設計されている。

### 2.2 タスク構成

本システムでは、両 CPU の処理を一致させるために業務処理を行うタスクを3種類設定し、システムスループットの向上を計っている。それぞれのタスクを、JOB タスク、I/O タスク、及び MAP タスクと名付け、各タスクは、表 2.1 に示す特性を持つようにしている。

また、デュアル運転時におけるシステムの動作状態を図 2.3 に示す。



(注) ① 印の左側は0番CPU, 右側は1番CPUの運転ステータスを示す。

--- は、ASCの自動IPLによる運転モードの復帰状態を示す。

図 2.2 システムの状態遷移図

表 2.1 タスクの特性

項番	項目	JOB タスク	I/O タスク	MAP タスク
1	処理内容	デュアル運転時の同期対象となる入出力メッセージの処理 <ul style="list-style-type: none"> <li>コア上のテーブルの更新</li> <li>処理結果をターミナル又は他タスクへ出力</li> </ul>	両表示装置に対する入出力処理 <ul style="list-style-type: none"> <li>リカバリデータのセーブ</li> <li>ターミナル・メッセージの取得</li> </ul>	ターミナルからの MAP 転送要求の処理 <ul style="list-style-type: none"> <li>MAP データの読み出し及び編集</li> </ul>
2	処理要求、及び初期データの発生頻度	最低 500ms に1回は発生	メッセージ発生時で、最頻時は 2~3秒に1回発生	ターミナルからの処理要求時で他のメッセージに比べるとほとんどない。
3	結果の出力	ターミナル 及び 他タスク	磁気テープ 又は 磁気ドラム	ターミナル
4	処理メッセージの性質	マスタ CPU 及び スレーブ CPU で 同期が必要	マスタ CPU 及び スレーブ CPU で 非同期	マスタ CPU 及び スレーブ CPU で 非同期
5	タスクの動作状態	タスク サスペンド 無し	タスク サスペンド あり	タスク サスペンド あり

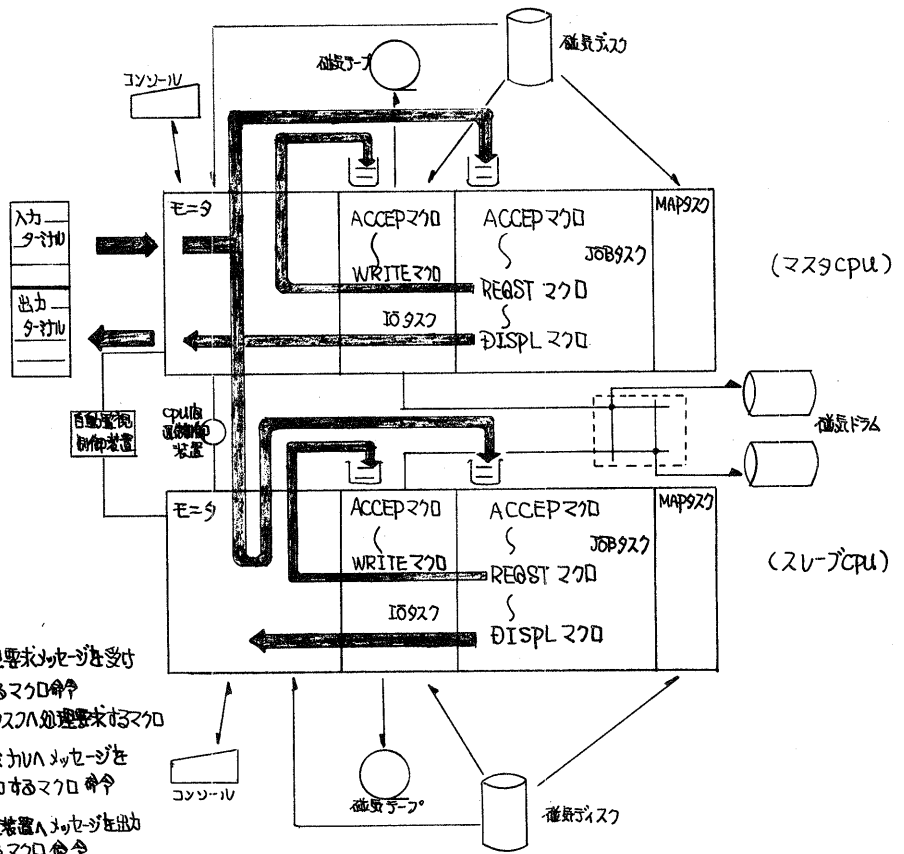


図 2.3 デュアル回転時のシステム動作状況

### 3. ハードウェアモニタ

使用したハードウェアモニタは、被測定システムに取付けたプローブにより、システムの動作に応じたレベル信号をサンプリングしてミニコンピュータに送りそこで処理したデータを磁気テープに出力すると同時に、CRT上に表示するコンピュータシステムである。

ここでは、ハードウェアモニタの持つ測定機能を以下に示す。

#### (1) カウンタ機能

カウンタ機能は、システムの状態を表わす信号のうち、性能評価に関する信号を取り出し、得られた信号をカウンタにより時間、又は発生回数として測定するもので、4種類の測定モードがある。

(i) PT (Processing Time) モード：本モードは、処理時間等の測定に利用し、主としてCPUの特性に関連した測定が行われる。

(ii) CH (Channel) モード：本モードは、入出力チャンネルの使用状況に関連した測定が行われる。

(iii) TR (Trunk) モード：本モードは、入出力バンクの使用状況に関連した測定が行われる。

(iv) OP (Operation code) モード：本モードは、命令の使用状況に関連した測定が

行われる。

(2) トレース機能

トレース機能は、被測定システムの注目するプログラム、又はアドレスの時間的・空間的な動きを測定するものであり、3種類ある。

- (i) データトレース機能：本機能は、指定されたアドレス（最大16個所）の内容を追跡し、そのアドレスの内容が変更された時、又は変更された可能性があるとき、新しい内容とその時の時刻を測定する。
- (ii) アドレストレース機能：本機能は、指定されたアドレスの時間的な動きを追跡するものであり、プログラムカウンタの値と着目アドレスとが一致した時、一致のパターンとその時刻を測定する。
- (iii) プログラムデバッグ機能：本機能は、指定されたアドレスへ書き込まれた値が、予想されていたものと等しいか、又は等しくない時、その時点のプログラムカウンタの内容を測定する。

(3) マップ機能

マップ機能は、測定したいメモリ範囲に対するCPUのアクセス頻度を測定する機能である。

4. 評価例

ここでは、3章で示した機能を持つハードウェアモニタを用い、システムのダウン状態からの立上げ、及びシンプレックス運転モードからデュアル運転モードへの遷移状態における評価、並びにターンアラウンドタイムに影響を与えるタスクのプライオリティに関する評価事例を以下に述べる。

4.1 システムの状態遷移

(1) 自動IPLによるシステムの再立上げ処理

システムがシンプレックス又はデュアル運転モード動作中、向欠障害等の発生によりダウンした場合、その状態をハード的に検出しダウン直前の運転ステータスがマスタであったCPUにIPL (ASC-IPL) をかけて、システムをシンプレックスモードで再スタートさせている。これは、本システムの業務上の性格

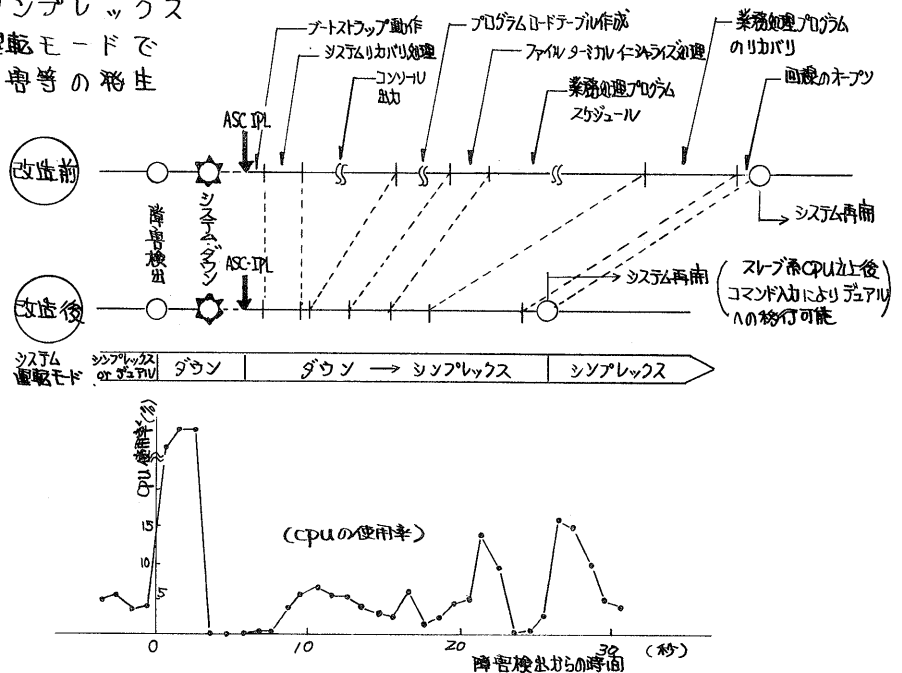


図4.1.1 システム再立上げ処理

と全国的なコンピュータネットワークを形成する関係からこのような方式を採用したのであるが、システムの立上げまでの時間を30秒以内にしなければならない厳しい条件が要求されている。システム設計、及びプログラム設計時において上述の条件を満たすようにシステムを作成したが、実際に出来上がった結果を測定したところ、30秒をはるかに越えて約60秒程度かゝることが判明した。このダウン状態を含んだ運転モードの遷移における処理時間短縮を計るため、ハードウェアモニタを使用し、再スタートのためのインシャリゼーション処理モジュール群のうちで、ボトルネックとなるモジュールを検出し、処理方式の変更を行った。

測定方法は、ハードウェアモニタのアドレストレース機能を用い、CPUがダウンする状態から再立上げが完了し、通常処理に移行するまでに処理されるモジュール内のアドレスを予め何個所か設定しておき、そのアドレスを通過する時刻を追跡した。

図4.1.1にシステム再立上げ処理の概略を、又表4.1.1にハードウェアモニタで測定した再立上げ処理の所要時間を示す。

向標点として、イニシャルメッセージのコンソール出力方法、プログラムロードテーブルの作成方法、及び業務処理プログラムのスケジュール方法等が検出された。このため、上述の処理を変更することにより、再立上げ時間が30秒以内に済む見通しとなり、システムの当初の要求を満たすことが可能になった。

(2) 自動IPLによるデュアル運転モードへの復帰処理

システムがデュアル運転を動作中、どちらかのCPUに何か障害が発生しダウンした場合、両CPU間では定期的にハルシエックを行っているため瞬間的に相手側CPUのダウン状態を検出できる。ダウンを検出すると、速やかにデュアル運転モードに復帰するため残ったCPUがマスタCPUとなり、ダウンしている相手側CPUへIPL(ソフトIPLを言う)をかけるスレーブCPUとして立上げた後デュアル組込み動作までを自動的にやっている。

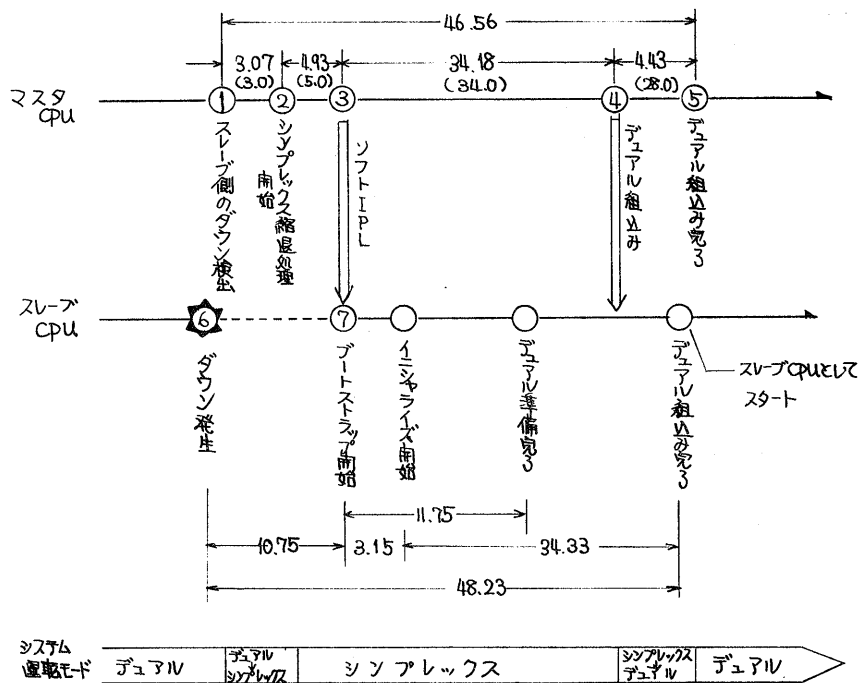
表4.1.1 自動IPLによる処理時間の比較

項番	処理内容	立上げ所要時間(秒)		改善率(%)	改善方法
		改造前	改造後		
1	ブートストラップ動作 (障害検出までの時間)	6.97	7.42		
2	システムリカバリ 処理	2.93	2.73		
3	コンソール出力	13.48	0.07	99.5	タイアアウト法の変更
4	プログラムのロード テーブル作成	12.20	2.22	81.8	テーブル作成処理の変更
5	ファイル及びタリ インシャリゼーション 処理	2.62	2.87		
6	業務処理プログラ ムのスケジュール	23.35	2.58	89.0	プログラムスケジュール 法の変更
7	業務処理プログラ ムのリカバリ	6.67	6.67		
8	回線のオープン	0.53	0.56		
合計時間		68.75	25.12	63.5	

表4.1.2 システム切替用IPLタイム

項番	タイムの種類	図4.1.2の 図番
1	システム異常検出タイム	⑥
2	異常原因のコンソール出力タイム	⑦
3	他CPU転送メッセージのバジ処理タイム	①-②
4	ソフトIPL起動タイム	②-③
5	デュアル組込み起動タイム	③-④
6	デュアル組込み取消しタイム	④-⑤

このデュアル運転モードへの復帰処理は、表4.1.2に示す各種のインターバルタイムの起動により動作するので、このインターバルタイムの値が適切であるか否向類となる。特に、信頼度を上げるためにデュアル運転を必要とする本システムにおいて、シンプレックス運転モードからデュアル運転モードへの復帰のタイミングは、マスタCPUからのデュアル粗込み開始の起動までに、ダウン側のCPUがスレーブとして立上げ処理が終るしている必要があるの



(注) ① 図中の数字は測定された処理時間である。単位：秒  
 ② ( ) 内の値は、設定したインターバルタイムの値である。

図 4.1.2 スレーブCPUダウン時のソフトIPL処理

で重要である。これらのインターバルタイム値をパラメータとして変更し、最適なタイム値。及び各処理上の動作状況を把握するために、ハードウェアモニタのアドレステレース機能を用いて測定した。

図4.1.2に、スレーブCPUのダウンから、それをマスタCPUが検出しデュアル運転モードに自動復帰するまでの処理における測定結果を示す。

測定結果から、スレーブCPUのブートストラップ開始からデュアル準備完了までの時間は平均して11秒以上かかり、自動IPL機能に許されている2回のIPLリトライ時間を考慮すると、デュアル粗込み起動インターバルタイム値(相手側CPUへ第1回目のソフトIPLをかけてからデュアル粗込み処理開始までの値)は33秒以上に設定する必要があることが分かった。また、その他のインターバルタイム値は、図4.1.2の中で示した値がほぼ妥当なものであることも分かった。

#### 4.2 タスクプライオリティ

オンラインリアルタイムにおけるタスクのプライオリティを設計する場合、システムのスループットとターンアラウンドとの関連を解明する必要がある。本システムの設計では、プライオリティレベルがI/Oタスク→JOBタスク→MAPタスクとした場合のパターン1と、JOBタスク→I/Oタスク→MAPタスクとした場合のパターン2の2種類のタスクプライオリティを検討したが、実際にどちらのパターンが良いかをハードウェアモニタで測定し、実験的評価を行った。

測定方法は、I/OタスクとJOBタスクのプライオリティレベルを変更し、それ

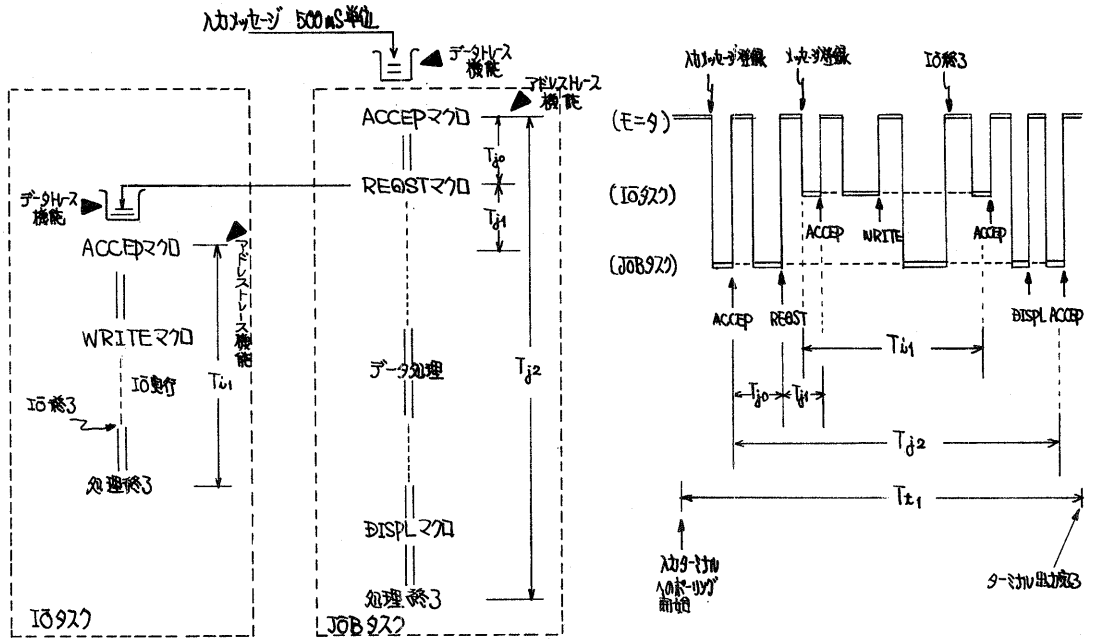


図 4.2.1 タスク制御概念図 (19-1)

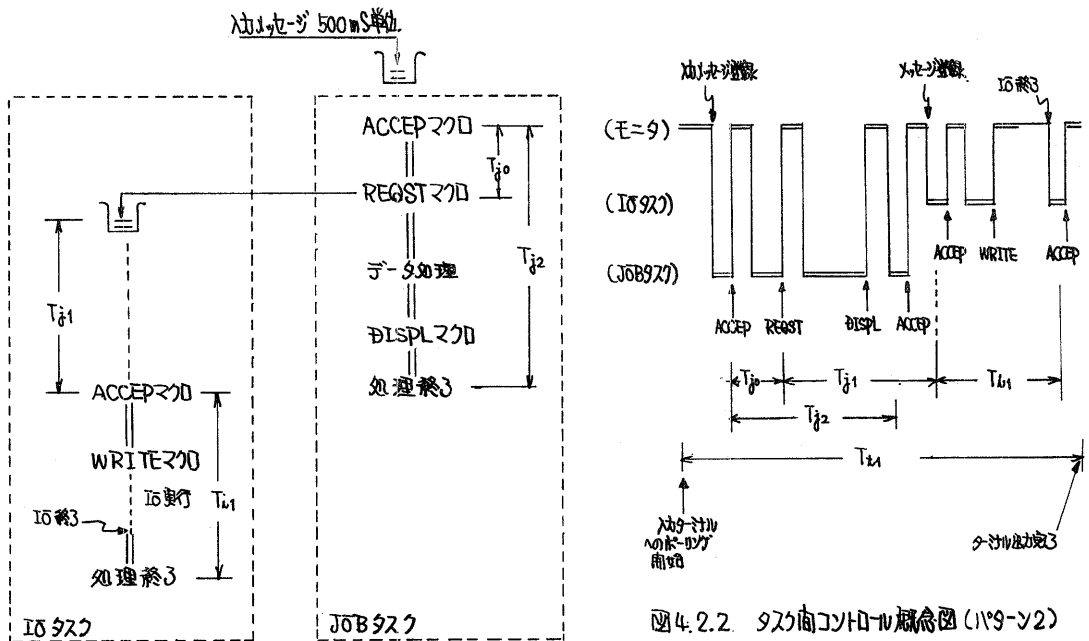


図 4.2.2 タスク制御概念図 (19-2)

がシステムに与える影響を測定するため、データ要求機能、アドレストレース機能、及びカウンタ機能のPTモードを用いた。図4.2.1にプライオリティレベルがパターン1の場合、図4.2.2にプライオリティレベルがパターン2の場合のタ

スケジューリング概念図を示す。また、測定の結果を表4.2.1に示す。

パターン1、及びパターン2におけるコントロールの遷移状態が解明された。パターン1のJOBタスクの処理は、IOタスクにおける周辺装置に対するIO実行中に終了することが分かった。IO実行中の空時間を有効に使っているパターン1の方が、システムスループットの向上が計られている。

ターンアラウンドタイムは表4.2.1に示すとおりである。JOBタスクの経過時間は、パターン2の方がパターン1より短縮されているが、入力メッセージに関するターンアラウンドタイム( $T_{t1}$ )は、パターン2を採用するほどの効果は見られない。図4.2.3にIOタスクとJOBタスクのターンアラウンドを示す。

パターン2の処理方式を採用した場合、IOタスク、JOBタスクを総括したターンアラウンドタイム  $T_T$  は、 $T_{j2}$  と  $T_{i1}$  とが直列的な処理形態をとった値でありパターン1の  $T_x$  より長くなり、トラフィックの過負荷状態においてはIOタスクの処理待ちキューに処理要求がたまりシステムのボトルネックとなる可能性が

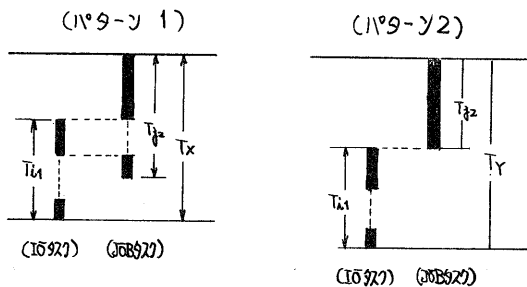


図4.2.3. ターンアラウンド概念図

考えられる。従って、オンライントリアルタイムシステムにおいては、あるタスクで処理されるメッセージのターンアラウンドタイムの短縮化のみを考えず、システムスループットをも考慮し、システム全般のメッセージに関するターンアラウンドタイムを短縮するような方向を設計しなければならないことが分かる。

以上より、パターン1のスケジューリングの方が、システムスループット、及びターンアラウンドタイムの見地から見て妥当な優先度構成であるので、パターン1を採用することにした。

表4.2.1 測定結果

項目	測定項目	パターン1	パターン2
1	ターンアラウンドタイム		
	$T_{i1}$ (IOタスク経過時間)	51 mS	51 mS
	$T_{j2}$ (JOBタスク経過時間)	45 mS	30 mS
	$T_{j0}$	21 mS	21 mS
	$T_{j1} + T_{i1}$	52 mS	60 mS
	$T_{t1}$	341 mS	333 mS
2	IOタスクの処理待ち時間 $T_{d1}$	640 $\mu$ S	8,660 $\mu$ S
3	ディスパッチング回数 $N_D$	31 回	34 回
4	メモリアバインド時間 $T_0$	42 mS	45 mS
5	CPU 使用率 $\rho$	11%	12%

(注) 上表の値は入力メッセージ到着間隔である500mSを基準にしたものである。



## 5. おわりに

本稿では、デュアルシステムにおいてタイミング的向類を含む運転モードの性能遷移、及びタスクプライオリティの評価を例にとり、ハードウェアモニタを用いたことによりシステムのボトルネックの検出、又は設計条件等の裏付けとなるデータの採取に有効であったことを述べた。このように、ハードウェアモニタはコンピュータシステムの性能評価におなり有用な道具であると言える。しかし、広く利用されるようになるためには、今後、装置自体の小型・軽量化を計るとともに、機能・性能の拡充、及び得られた評価データを処理するソフトウェアの充實が望まれる。

最後に、ハードウェアモニタによる測定のあたり、御協力をいただいた関係各位に感謝の意を表します。

## 参考文献

- 1) 若谷 阿江 ほか：「民間航空路レダ情報システム-ソフトウェア」施設 vol.27 no.3
- 2) 島谷 春彦：「ハードウェアモニタによるコンピュータシステムの評価方法」施設 vol.28, NO.5
- 3) 箱崎 ほか：「性能評価システム; SYDAS-II - 総括・ハードウェア編」NEC資料 LR-3115
- 4) 箱崎 ほか：「性能評価システム; SYDAS-II - 制御プログラム編」NEC資料 LR-3142