

## 並列信号処理装置における高信頼化と動的負荷分散の実現方式

高橋勝己, 水野政治, 古市昌一, 宮田裕行  
三菱電機(株) 情報技術総合研究所

画像や信号など、定期的に連続して送られてくるデータを複数のDSPを用いて処理する並列信号処理装置を設計中である。本装置では、対象データの性質を過去の処理履歴から予測し、処理内容を変化させる、いわゆるアダプティブ処理を実現する。本論文では、この動的に変化する処理を複数のDSPを用いて行なう、動的負荷分散方式と高信頼化を実現する方式について報告する。

### Fault Tolerant Control and Load Distribution Algorithm Scheme in Parallel Signal Processor System

Katsumi TAKAHASHI, Masaji MIZUNO, Masakazu FURUICHI,  
Hiroyuki MIYATA  
Infomation Technology R&D Center, Mitsubishi Electric Corporation

We are developing high performance parallel signal processor system which employs multi-DSP architecture. Primary applications are signal and image data processing which require realtime response and high performance. In our load distribution algorithm, we utilize the history of load of processed data to determine the best parameter for the incoming data. In this paper, we describe the architecture of this system, the load distribution algorithm and the fault tolerant control scheme.

#### 1. はじめに

近年、マルチメディアデータの処理要求が高まり、動画像や連続する信号などの高速処理が重要となってきている。画像や信号など定期的に連続して送られてくるデータに対し、定型的な処理を、指定時間内に遅滞なく行なう装置では、従来、専用のハードウェアが多く用いられてきた。しかし、専用のハードウェアは、信頼性の確保が難しく、データの諸条件の変化にも対応できなかった。

近年のLSI技術の進歩に伴い、多数のDSPを接続したアーキテクチャ<sup>1)</sup>の実現可能性が高まり、更に、その高速化から、リアルタイム処理などの時間的制約を満たすことが可能になってきた<sup>2)</sup>。この構成では、論理的に使用するDSP数の増減や使用プログラムの変更を動的に行なうことによって、データの諸条件の変化に対応することができ、DSPに故障が発生した場合にも、他のDSPが代替することによって、処理を維持することができる<sup>3)</sup>。

信号処理分野の中には、信号の性質を過去の処理履歴から予測し、処理内容を動的に変更できるものがある。我々は、信号の持つ性質の予測に基づいて、信号処理の内容を動的に変更する並列信号処理装置を設計してい

る。

以下、本論文では、並列信号処理装置のアーキテクチャ、信号の性質予測に基づいて行なう動的負荷分散方式、高信頼化方式、及び、シミュレーションによる負荷分散動作検証について報告する。

#### 2. 並列処理アーキテクチャ

図1は、本並列信号処理のアーキテクチャを示したものである。本アーキテクチャは、以下の信号の処理を行なうプロセッサ(DSP)と装置全体の制御を行なうプロセッサ(マイクロプロセッサ; $\mu P$ )、及び、これらを接続するバスから構成される。

- 前処理プロセッサ  
装置外部からデータを受けとり、データを複数に分割するための準備等、信号処理の前処理を行なう。
- 信号処理プロセッサ群  
各プロセッサは必要なデータをバスから取り込み、処理を行なう。信号処理の主な部分は、全てこのプロセッサ群で処理される。
- 後処理プロセッサ  
信号処理プロセッサ群の処理結果を収集し、信号処理の後処理を行なう。
- 制御プロセッサ

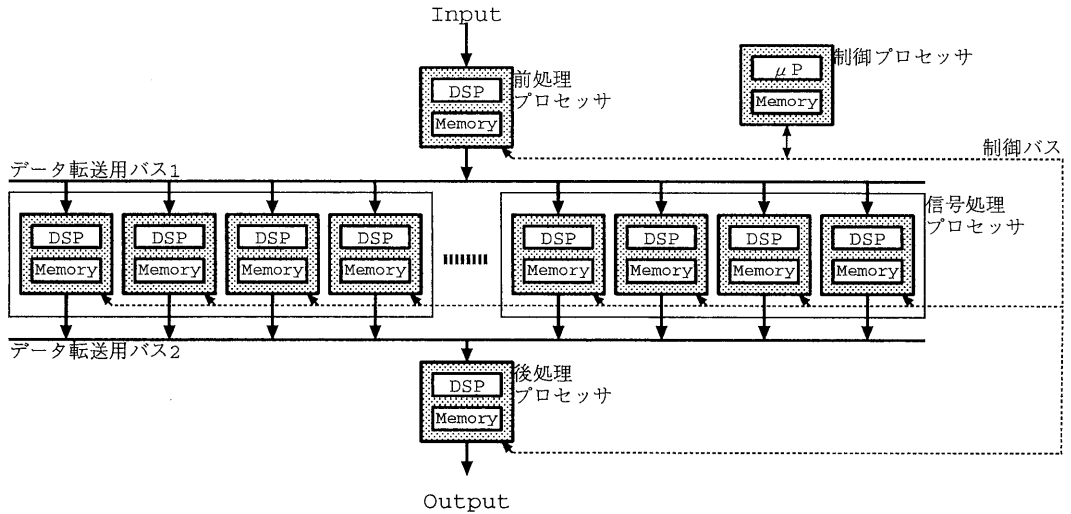


図1 並列信号処理装置の構成

個々のプロセッサを監視し、データの分割や信号処理プロセッサ群への割り当てなど、装置全体を制御する。

- データ転送用バス  
信号処理プロセッサ群へのデータ分配、及び、プロセッサ群からのデータ収集を行なうためのバス。
- 制御用バス  
制御プロセッサからの制御データを各プロセッサに伝達するためのバス。各プロセッサの状況把握にも用いられる。  
これらの各プロセッサは、ローカルメモリを持ち、バスを介してデータの送受信を行なう。

### 3. 動的負荷分散方式

#### 3.1 入力データの分割

入力データは、互いに独立なデータが定期的に送られてくる。また、個々のデータも、互いに独立な領域に分割することが可能である。

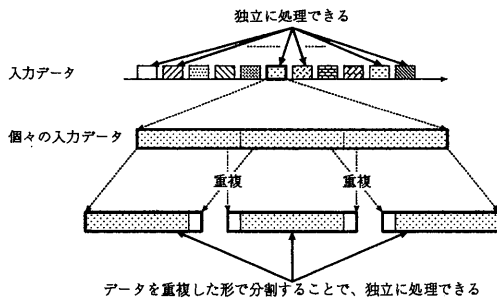


図2 入力データの性質

図2は、入力データの性質を示したものである。図のように、入力データは、一定周期毎に入力される独立なデータとして提供される。この個々の入力データは、そのまま、独立に処理が可能である。また、個々のデータは、内部に局所性を含んでいるため、更に分割を行ない、それぞれを独立に処理させることも可能である。しかし、データ内の分割では、局所的に関連しあっているため、図のようにデータを重複させた形で分割を行なう必要が生じる。

入力データが以上のような性質を持つため、これらを実行する方法としては、大きく2つあり、次のようになる。

- (1) 入力時に分離されている個々のデータを、そのまま、1つのプロセッサで処理させる方式
  - 1つ分の入力データを1つのプロセッサで扱うため必要メモリ量が多い
  - 1つ分の入力データを1つのプロセッサで処理するため、データ入力から処理結果が得られるまでのターンアラウンド時間が長い
  - 並列に実行するためのオーバーヘッドが少ない
- (2) 入力時に分離されている個々のデータを、更に分割し、複数のプロセッサで処理させる方式
  - 1つ分の入力データを更に分割したものを1つのプロセッサで扱うため必要メモリ量が小さい
  - 1つ分の入力データの一部分を1つのプロセッサで処理するため、データ入力から処理結果が得られるまでのターンアラウンド時間が短い
  - 1つ入力データを更に重複させて分割し、重複した部分をそれぞれ独立に処理する分、発生するオーバーヘッドが大きい。

信号処理装置では、処理結果の利用用途によって許容

処理遅延時間が規定される。本装置では、この許容処理遅延時間に合わせて、2つの方式を統合した分割方式をとる。

### 3.2 負荷分散

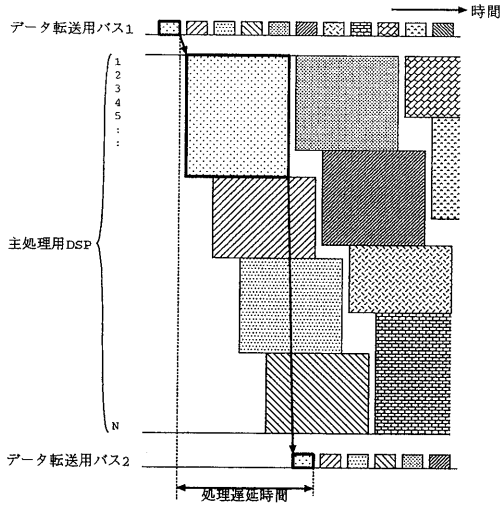


図3 信号処理プロセッサと入力データ

図3は、許容処理遅延時間がデータ入力間隔の4周期分であるとして、信号処理プロセッサと入力データとの関係を示したタイムチャートである。図では、入力データは、信号処理プロセッサが4周期で処理できる分量毎に分割され、個々のプロセッサに分配/処理される。個々の入力データに施す処理は、過去の処理結果に基づいて変化するため、負荷量も変化する。この負荷量の変化は、データの分割数、処理を割り当てるプロセッサ数として現れることになる。

このように、本分割方式では、入力データ1つに対し割り当てるプロセッサ数は、処理内容に応じて変化する。このため、 $L$ 周期で信号処理を行ない、最大負荷の時のデータ分割数  $M$  の時、本信号処理装置は、必ずしも、 $LM$  個のプロセッサを必要としない。 $L$ 周期で信号処理をする場合には、 $L$  個の入力データが同時に処理されることになるが、データの全ての組合せにおいて、分割数が  $M$  個未満の入力データが、常に1つ以上あれば、 $LM$  個未満のプロセッサで信号の処理を行なうことができる。

### 3.3 負荷の予測と制御

本並列信号処理装置では、入力データの性質を、データの周期性と過去の処理結果から予測し、処理内容を決する。本装置では、データの性質を予測するための管理テーブルを用意し、そのテーブルの要素毎に、

- サンプルしたデータを補間によって増やし、処理する。
- サンプルしたデータをマージによって減らし、

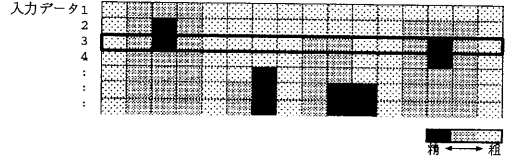


図4 個々のデータと信号処理の精粗

処理する。

といった、信号処理の精粗を決定する。この精粗は、そのままデータを処理する際の負荷の大小に対応し、信号処理結果によって常に更新される。図4は、この精粗を表のマス濃淡で表したものである。図中の太枠で囲まれている部分が、1つの入力データである。入力データは、幾つかの領域からなり、それぞれ実施する信号処理の精粗が指定されている。

制御プロセッサは、個々の領域の精粗に基づいて処理を決定し、領域毎に処理内容からプロセッサでの実行時間を予測、分割の際に発生するオーバーヘッドを換算しながら、データを分割してゆく。分割後の各データ群と処理するプロセッサとの関連付けは、制御プロセッサが、信号処理プロセッサのスケジュール管理に基づいて行なう。

## 4. 高信頼化方式

本章では、並列信号処理装置において行なった高信頼化方式について述べる。

### 4.1 故障検出

信号処理プロセッサの故障は、自己診断プログラムやパリティなどによって検出する。信号処理プロセッサは、1つのデータに対する信号処理を終える毎に自己診断を実施し、自己の動作を確認する。制御用のプロセッサは、この診断結果と、ハードウェアが提供する故障情報を収集し、プロセッサの故障状態を把握する。

### 4.2 故障回避

本信号処理装置では、故障発生時の処理の補償は行なわず、故障が検出されたプロセッサへの負荷配分の停止を行なう。

図5は、個々のデータの負荷を一定として、予備のプロセッサが用意されている時の故障回避の流れを示したものである。動作は次の順序で行なわれる。

- (1) データ1の処理が、各プロセッサ(1-7)に割り当てられる。
- (2) データ1の信号処理中にプロセッサ4番に故障が発生する。
- (3) データ2の処理が、各プロセッサ(1-7)に割り当てられる。
- (4) データ1信号処理を終えた後、自己診断によって故障が検出される。
- (5) 制御プロセッサが自己診断結果を収集し、プロセッサ4番の故障を認識する。

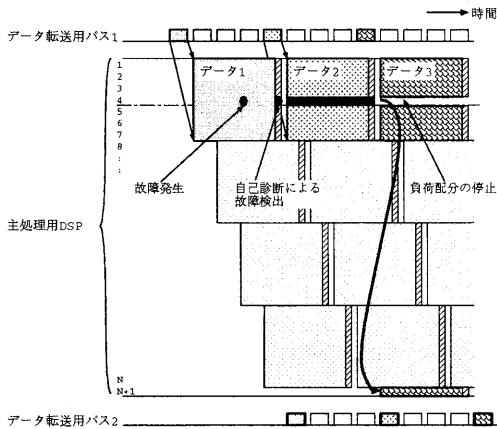


図5 故障回避

- (6) データ3の処理が、各プロセッサ(1-3, 5-7, N+1)に割り当てられる。この時、プロセッサ4番への処理は、故障のため、新たにプロセッサN+1番に割り当てられる。

データ2のプロセッサの割り当ては、故障検出の前であるため、この場合、データ1とデータ2において、プロセッサ4番に割り当てられたデータの処理結果は、得られない。データ3以降は、故障を認識した上で割り当てを行なうため、損失なく処理結果を得ることができるようになる。

本構成では、信号処理プロセッサでの故障検出後、制御プロセッサが故障を認識し、故障を回避した負荷配分を行なえるようになるまでには、2周期分の遅れが生じる。したがって、自己診断と次の処理割り当てとの間には、2周期以上あくことが望ましい。このため、本装置では、プロセッサのスケジューリングとして、割り当てられた処理の終了後、新たな処理が割り当てられるまでの時間が長い、キュー方式を採用している。

#### 4.3 縮退処理

代替できる予備プロセッサがない状況で、処理を継続させるためには、

- データの一部を処理しない。
- 処理内容を負荷の少ないものに切替える。

といった対処が必要になる。後者は、縮退と呼ばれる処理であり、本装置では、この方式を選択している。

##### 4.3.1 データ単体での縮退

データの処理に割り当て可能なプロセッサ数が必要数を下回る場合、本信号処理装置では、処理の精粗を粗側にシフトすることによって、処理に必要なプロセッサ数を削減する。この粗側に処理をシフトさせる領域は、予め設定しておく縮退優先度に基づいて選択する。

図6は、図4と同じ表記を用い、縮退優先度を

- (1) 負荷の低い領域(粗の領域)ほど縮退優先度が高い
- (2) 負荷が同じ領域は、右側のものほど縮退優先度が高い

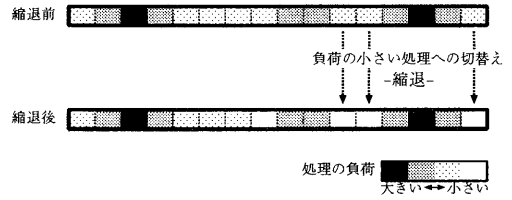


図6 データと負荷の小さい処理への切替え

い

として、図4の太枠で囲んだ入力データ3について、3つの領域の縮退を行なった時の例を示したものである。図6のデータの処理量は、3つの領域についてより負荷の小さい処理への切替えを行なっているため、減少する。

縮退処理では、処理に必要なプロセッサ数が割り当て可能な数になるまで、この処理の切替を繰り返す。このため、その時割り当て可能なプロセッサ数やデータの領域毎の精粗状況によってその縮退の範囲は大きく異なる。

##### 4.3.2 複数データ間での縮退調整

本信号処理装置では、図3, 図5で示したように、複数の入力データを同時に処理している。このため、複数のデータ間での縮退を調整することができる。複数データ間で縮退を調整する場合には、縮退優先度の低い領域しかないデータでの縮退を避け、その前後のより縮退優先度の高いデータで縮退を行なうことができる。同様に、縮退優先度が低いだけでなく、縮退を禁止する領域や入力データを設定することも可能になる。

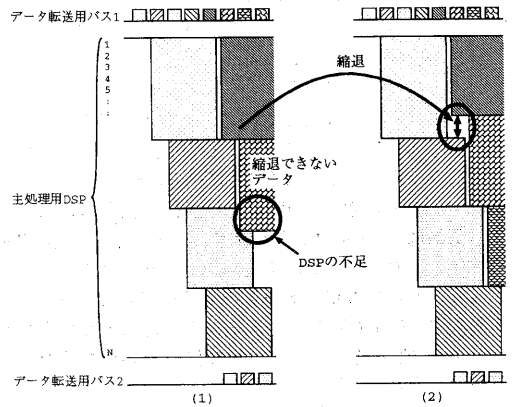


図7 縮退処理

図7は、複数データ間での縮退調整を行なう例を示したものである。図7(1)のタイムチャートは、縮退を禁止したデータの割当を行なう時点で、プロセッサ不足が生じ、処理不可能になった状態を示している。

これを、予め数周期分のデータ処理負荷を予測し、縮

退による調整を行なうことで対処可能にしたのが、図7(2)のタイムチャートである。この例では、縮退できないデータの前に処理するデータを割り当てる時点で、後続のデータで必要となるプロセッサ数と割り当て可能なプロセッサ数を予測し、その時点で必要なプロセッサ数を確保するために、縮退を行なっている。このような対処を行なうためには、同時に処理するデータの個数分先までこの予測を行なう必要がある。

また、複数のデータ間で縮退を行なう場合には、データ間の縮退優先度も設定する必要がある。本装置では、同等の縮退優先度が同等であれば、入力時期の遅いデータの方が縮退優先度が高いとして処理する。

#### 4.4 故障したプロセッサの対処

プロセッサの故障は、リセットによって回復するものと、回復しないものの2つに分けることができる。前者は、制御プロセッサが行なう、プロセッサのリセット後、診断プログラムの実行による正常動作確認がなされれば、配分の停止を解除し、処理の割り当てを再開する。一方、後者は、プロセッサの交換によって対処する。本装置では、このプロセッサを交換を、装置を稼働させたまま行なう。

### 5. シミュレーションによる負荷分散検証

負荷分散方式については、様々な条件下でその動作を見るためにシミュレータを作成し、検証を行なった。

#### 5.1 シミュレーションモデル

本シミュレータは、実際のデータの処理は行なわず、処理した際の時間の算出を基準にプロセッサをモデル化した。プロセッサは、信号処理中、待機中、故障中という3つの内部状態と、処理を終了した入力データのIDを持ち、負荷分散をシミュレーションする。

データの処理に必要な時間は、プロセッサで行なう処理毎に、データの量を引数とする関数を用意し、この値を元に算出している。この関数は、処理内容を数式表現した時の、演算子の出現数に基づいて設定している。このように、実行時間の算出は、実際に使用するプログラムを実行し、測定したものではない。このため、今後、測定した結果を用いて、この関数を修正し、シミュレーションの精度を上げる予定である。

#### 5.2 入力データの種類

本シミュレータでは、想定される入力データや、精粗に基づき選択される処理の組合せとして20～30のパターンを用意し、シミュレーションで負荷分散動作を検証した。

図8は、プロセッサに割り当てる入力データの分割を示したものである。データの分割は、図4、図6で示したような信号処理の精粗を管理する領域毎に行なわれる。このため、データ分割は連続する領域の任意の点による分割ではなく、領域の切れ目による分割という制約を受ける。このため、分割後のデータを割り当てられた

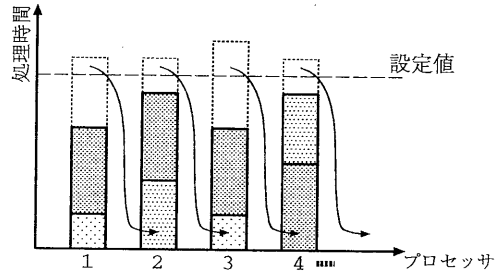


図8 プロセッサに割り当てるデータの分割

プロセッサの稼働率は、必ずしも100%になるわけではない。このように、領域毎の負荷の状況や設定したプロセッサの処理時間によって、分割後の個々のデータ群の負荷量は、大きく変動する。図3、図5、図7では、データの入力間隔の4周期分をプロセッサの処理時間として、信号処理を行なっているが、処理時間を5周期とした場合、1つのプロセッサが処理できるデータの量が増加する。そのため、1つの入力データを処理するために必要となるプロセッサの数は減少するが、5つの入力データが同時に処理されることになる。従って、入力データの処理に割り当てる時間は、信号処理に許されている遅延時間よりも短い時間のほうが、装置として処理に必要なプロセッサ数が少なくなる可能性がある。

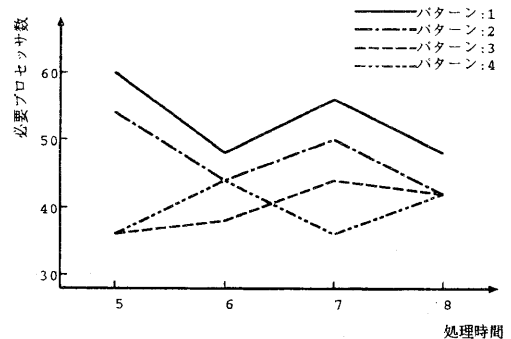


図9 処理時間と必要プロセッサ数

図9は、本信号処理装置が対象とする問題の中で、代表的な4つのパターンについて、データの入力間隔を単位として、処理時間(5～8)と必要プロセッサ数の関係を示したものである。本装置は、これら4つの全てを処理対象とするため、この結果から、装置に許容される処理遅延時間が、7周期分であっても6周期分を処理時間として設定した方が、より少ないプロセッサで信号処理を実現できることが分かる。

#### 5.3 プロセッサの状態表示

本並列信号処理では、システムの運用中に故障したプロセッサの交換などを実施するため、プロセッサの稼働

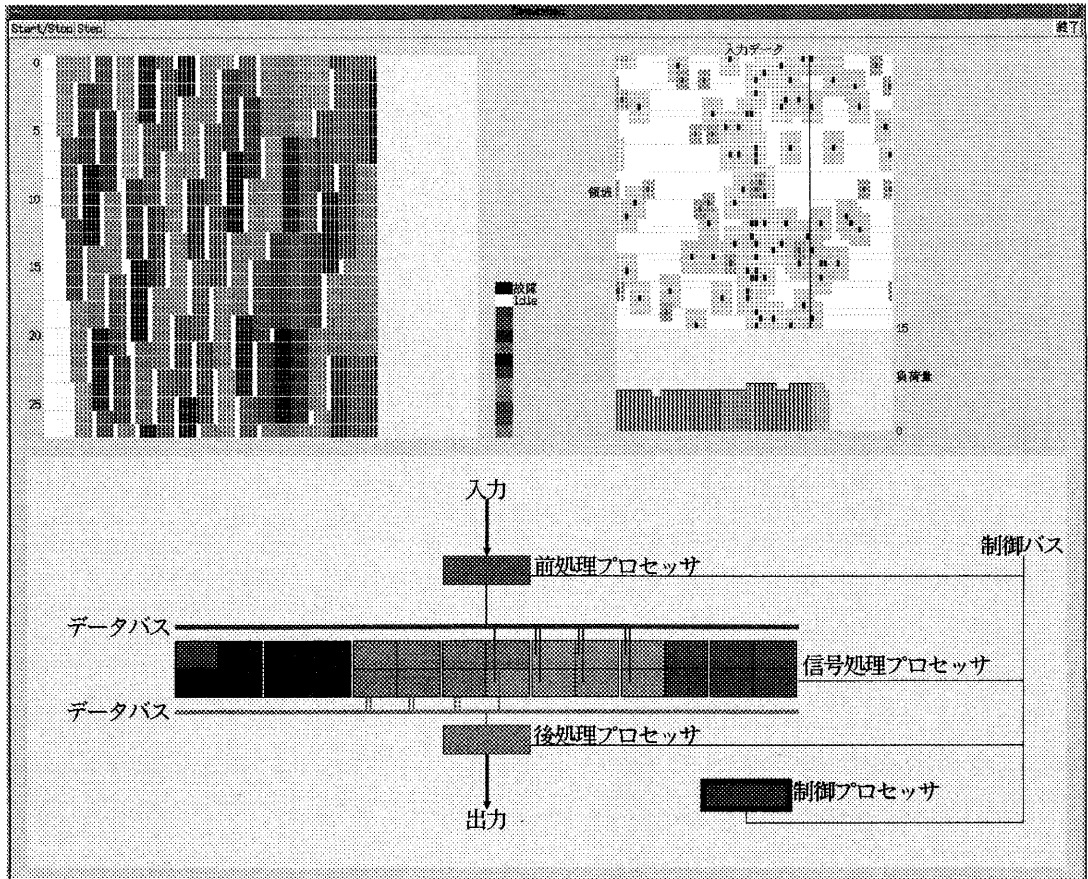


図 10 シミュレーションの画面表示例

状況を含む情報のモニタが必要になる。これらの情報を、シミュレータから収集し、表示したものが、図10の画面である。本画面では、プロセッサの稼働状況以外にも、各プロセッサが現在処理中の入力データのIDや、図3のようなタイムチャート、入力データの処理内容を決定するための精粗情報などが表示されている。本表示は、実機にも、同様に使用する予定である。

## 6. おわりに

本論文では、現在作成中の並列信号処理装置の構成や、その中で実現している負荷分散方式、及び、高信頼化方式などについて述べた。これらの動作については、シミュレータを作成し、負荷分散方式を検証と、システム動作の可視化を行ってきた。本並列信号処理装置のアーキテクチャやアルゴリズムは、現在試作中の信号処理装置に適用しており、今後、シミュレーションでの検証内容との比較を含めた評価を行っていく予定である。

る。

謝辞 本研究において、様々な御尽力と御助言を頂いた三菱電機(株)通信機製作所 芝崎直人氏と同情報総合技術研究所 真野清司氏に感謝します。

## 参考文献

- 1) 奥川峻史著:『並列計算機アーキテクチャ』, コロナ社
- 2) 水野, 宮田, 管:『高速信号処理向き並列アーキテクチャの提案』, 情報処理学会 第50回全国大会
- 3) 向殿政男編:『フォールト・トレラント・コンピュータリング』, 丸善