

# 1N-2 並列処理システム-晴-の待ち合わせ記憶構成

山名早人、草野義博、萩原孝、村岡洋一  
(早稲田大学 理工学部)

## 1. はじめに

我々は、主に科学技術計算を目的とした並列処理システム-晴-を提案している[1]。-晴-では、プログラムに内在する並列性を十分に引き出す為にデータフロー実行を取り入れている。データフロー実行では、ノードの発火制御を司る待ち合わせ記憶(WM:Waiting Memory)の高速化がシステム全体の高速化において重要なポイントとなる。

本稿では、-晴-の試作機で用いる待ち合わせ記憶WMの構成について述べると共に、ソフトウェアシミュレータによる簡単な評価を行う。

## 2. -晴-の要素プロセッサ構成

-晴-の要素プロセッサ(PE)の構成を図1に示す。BCU(Broadcast control Unit)はBCN(Broadcast Network)からの実行命令を受け取る。これによって各PEは実行を開始する。-晴-では、プログラムをマクロブロックという単位に分割し[2]、マクロブロック単位で実行を行っている。このため実行開始時に、実行マクロブロックに属するデータ(以後パケットと呼ぶ)をDM(Data Memory)からWM(Waiting Memory unit)へ転送する。WMは、対になったパケットをLCU(Local Control Unit)へ送る。LCUは、共有メモリアクセス関係の命令を処理し、それ以外の命令はEX(Execution unit)へそのまま送られる。EXは、演算を実行すると共に、データフローグラフ上の次のノードに対するパケットを作成しSW(Switching unit)へ送る。SWでは、パケットを他のPE宛のもの、自PE宛のものに振り分け、他PE宛のものは、ネットワークを通じて外部に送り出す。また、自PE宛のものはDMC(Data Memory Controller)へ送る。DMCでは、パケットが現在実行中のマクロブロックに属するかを調べ、実行中のマクロブロックに属するものであればWMに、他のマクロブロックに属するものであればDMに格納する。これによってWMへの負荷を軽くしている。

## 3. PE試作機におけるWM構成

### 3.1. 容量

-晴-では、MEB(Multiple Exit Block)やMEIFB(Multiple Exit + perfect IF Block)というマクロブロック[3]単位で実行を行う。現在実行中のマクロブロックに必要なパケット、すなわち他のマクロブロック宛のパケットはWMへ入力されずにDMに格納される。このためWM内に溜まるパケット数を少なく抑えることが出来る。現在、-晴-のソフトウェアによるシステムシミュレータ(HSSV2:-Harray- System Simulator Version 2)により、この容量についての検討を行っている。

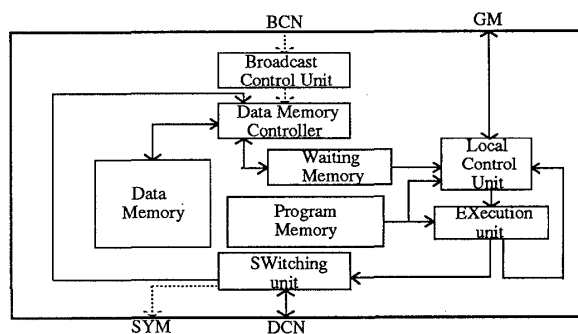


図1 PEの構成

### 3.2. 処理速度

試作機では、システムクロックを最大10(MHz)まで上げることが目標にしている。パケットは2語に分割して送るので、第2章で述べたWM→LCU→EX→SW→DMC→WMのパイプラインは、各ステージが200(ns)で処理を終了しなければならない。したがって、WMも200(ns)間隔で処理可能な構成にする。

### 3.3. ハードウェア構成

ここでは、連想メモリ(CAM)[4]を例に挙げて、CAMを用いた場合の構成方法を述べる。CAMを用いる場合の利点として、

1. ワード並列に検索が出来る為に高速な検索が可能
2. 検索によって一致した複数語の連続読み出しが可能で、スティッキーの処理が容易
3. アドレス指定による読み出し/書き込みが出来るため、データのダンプが容易であり、デバッグ時に便利
4. 一般にCAMは大容量化が困難であるが、-晴-ではマクロブロック単位での実行を行う為、現在試作されている容量(512W/CAM)で対応可能。

などが挙げられる。しかしながら、データフロー実行専用のチップではないので、待ち合わせ記憶を実現するには図2に示すような手順を踏む必要がある。この為、1パケットの処理に最低400(ns)、最大700(ns)の時間が必要となる。

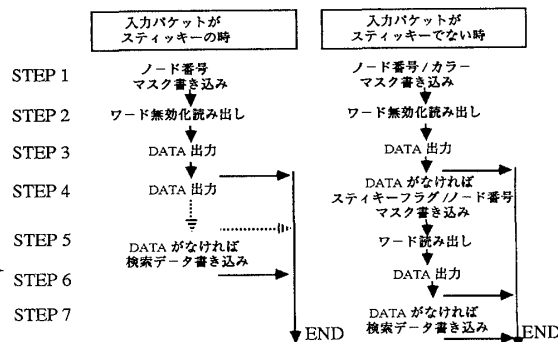


図2 CAMによる待ち合わせ記憶実現の手順

A Construction of Waiting Memory for Parallel Processing System -Harray-

Hayato YAMANA, Yoshihiro KUSANO,

Takashi HAGIWARA and Yoichi MURAOKA

School of Science and Engineering, WASEDA University

第3.2節で示した処理間隔200[ns]を満足させる為に文献[5]で提案を行ったWMのバンク化を行う。図3にバンク化を行ったWM構成を示す。検索に必要なバケットのタグ部(スティッキーフラグ/ノード番号/カー)をCAMに入れ、他のデータはSRAMに記憶させてる。WMのバンクは、0から3までの4つに別れており、バンク0には、バケットの行き先ノード番号下位2bitが"00"のものが、バンク1には"01"のものがというように、バケットのノード番号下位2bitによって各バンクにバケットを振り分ける。バケットが行き先ノード番号で見たときに平均的に到着した場合、1バンク時に比較して最大4倍の処理性能向上が見込まれる。

また、図3における裏バンク用CAMは、次に実行されるマクロブロックに属するバケットを前もってロードしておくためのものである。これによって、マクロブロック実行開始時のDM→WM間のバケット転送オーバーヘッドを隠すことが可能となる。

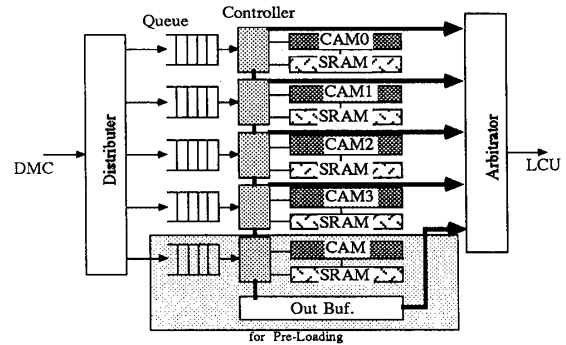


図3 WM構成

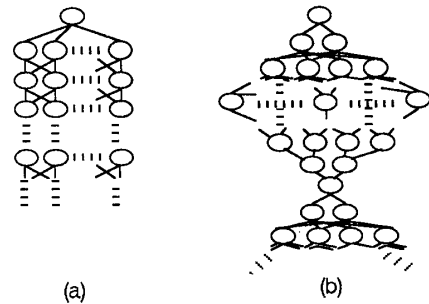


図4 使用プログラムのデータフローグラフ

3.4. バンク化の評価

バンク化によって処理の向上が図れることを確認するために、一晴-のソフトウェアシミュレータHSSV2を用いてシミュレーションを行った。用いた2種類のプログラムを図4に示す。図4の(a)は、常に一定の並列度のあるプログラムであり、(b)は並列度の変化するプログラムである。またノード番号は、図4のグラフの上から下へ、同一段では左から右へ0から順に付けてある。シミュレーションではそれぞれ1024個のノードを実行した時の実行時間を求め、バンク化しない時を1として処理速度の向上比を求めた。これを図5に示す。

図5(a)(b)共に4バンクと8バンク化時の結果が等しくなっている。これは、WMへのバケット入力間隔が最高200[ns]であり、CAMでの処理時間が400/700[ns]、CAMの処理時間が入力間隔の2/3.5倍である為、4バンクよりも多いCAMを設けても無意味であることを示している。

図5(a)では、並列度が約8のところまで2/4バンク時共に速度向上比が飽和している。これに対して(b)では、並列度が約16のところまで飽和しており、飽和並列度が大きくなっている。これは、(a)が常に並列度が一定であるプログラムに対し、(b)は並列度が変化するプログラムである為にWM→LCU→EX→SW→DMC→WMのパイプラインがみだれることに起因していると考えられる。

データフロー実行を行うことによって、プログラムの並列性が十分に引き出せると考えると、この結果より、WMは2バンクよりも4バンク化するのが適当であることがわかる。また4バンク化した時、並列度が平均で8~16以上であれば、バンク化しない時に比べて約3倍の処理速度向上が期待出来ることがわかる。

今回使用した図4のプログラムでは、ノード番号が順番に付けられていた為、連続して同一バンクにバケットが入力されるという状況がほとんど発生しなかった。しかし、ノード番号の付け方によって、処理時間に影響が出て来ると考えられるので、今後シミュレータによりさらに検討を行っていく。

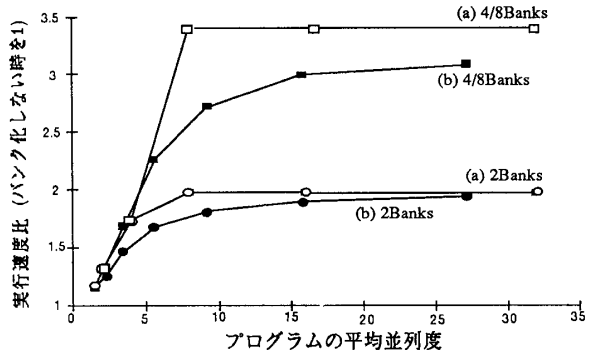


図5 WMバンク化時の処理速度向上

4. おわりに

並列処理システム一晴-における待ち合わせ記憶の構成について述べ、バンク化により高速化が可能であることを示した。現在、要素プロセッサ(PE)の回路設計を行っており、今年度中にPE試作機1台を動作させる予定である。

参考文献

- (1)丸島他：並列処理システム一晴-の実行方式，情処研報，88-CA-69, pp.9-16, 1988
- (2)萩原他：科学技術計算用並列処理システム一晴-のマクロブロック化手法の検討，情処34回全大 1Q-9, 1987
- (3)萩原他：並列処理システムにおけるFortranプログラムのマクロブロック化の評価，情処35回全大 1C-4, 1987
- (4)小倉他：20Kb CAM(Content Addressable Memory LSI)，信学技報 CPSY87-33, pp.31-37, 1988
- (5)山名他：並列処理システム一晴-の要素プロセッサ構成，情処研報，88-CA-70-4, 1988