

SIMD 型並列処理における分岐支援機構に関する提案

佐野雅彦*¹ 高橋義造*²

徳島大学総合情報処理センター*¹ 大阪工業大学情報科学部*²

SIMD 型並列計算機において SPMD 的処理を実現するための分岐支援機構を提案する。本方式は、従来の SIMD で採用されているフラグやマスクによるアクティビティ制御とは異なり、命令アドレスを命令と共に PE に放送する、命令アドレス放送方式を用いたアクティビティ制御方式により、SPMD で必要とされる複雑な条件分岐を可能とする。本稿では、提案する分岐支援機構の概要と特徴について述べる。

Branch support mechanisms for SIMD parallel computing

Masahiko Sano*¹ Yoshizo Takahashi*²

Information Processing Center, The University of Tokushima*¹

Faculty of Information Science, Osaka Institute of Technology*²

New branch support mechanisms for SPMD paradigm on SIMD machine have been proposed. Unlike to the ordinary SIMD machine with activity control mechanism by using of condition flags and execution masks policies, our proposal activity control mechanism using a instruction address broadcast method enables complex conditional jumps required for SPMD paradigm on SIMD machine. The instruction address broadcast method is that a control processor broadcasts instruction address to processor elements together with its instruction word. In this paper, we describe our proposed branch support mechanisms.

1. はじめに

並列処理ではプログラミング手法が重要である。MIMD マシンでの典型的な手法である SPMD(single program, multiple data)手法 [1]は同じプログラムのコピーを全てのプロセッサ要素(PE)で非同期に実行するため、幅広く使用されている。一方 SIMD マシンは、全ての PE が同じ命令を同期して実行する特性から、データ並列モデル[2]には適するが、SPMD 的な運用が難しく、MIMD マシンと比較すると汎用性が低く応用範囲が制限される。しかし、汎用性が向上すれば応用範囲が広がるため、価格性能比に優れる SIMD マシンの有効性は向上する。このような観点から 1つの解決策として、MIMD マシンの機能(ま

として、MIMD マシンの機能(または命令)を SIMD マシンでシミュレーションする方式が提案されている。この種の方式では、基本的に、MIMD マシンで動作するアプリケーション(アルゴリズム)は SIMD 上で処理可能であるが、処理性能はシミュレーション速度に依存しており、改善課題とされる[3]。

対照的に、本研究は前述の方法とは異なり、SIMD マシン上で直接処理する手法に主眼を置く。そして従来の SIMD マシンの問題点であるアクティビティ制御の改善を検討した結果、本稿で述べる分岐支援機構[4]による SIMD アーキテクチャを考案した。以降では、SIMD における SPMD 処理の概要と、提案する分岐支援機構について述べる。

2. SIMD における SPMD 処理

典型的な SIMD マシンの概略図を図 1 に示す。SIMD マシンは複数の PE を単一の制御プロセッサ(CP)で制御する。命令語は CP から全 PE に放送され、全 PE は同期して実行する。SIMD の特徴は、大半の制御信号が CP から命令として PE に供給されるので、汎用プロセッサと比較して PE の構造が簡単である。従って MIMD と比較してチップ上により多くの PE を実装できるので、超並列マシンを構成し易く、価格性能比に優れていることにある[4]。

MIMD では多くの場合、SPMD 手法が採られる。SPMD プログラムは、プロセッサ間通信と同期を除けば、これまでの単一プロセッサのプログラムと同様の手法である。SPMD プログラムでは無条件/条件付分岐命令が含まれているが、各 PE ごとに異なる部分を実行している場合が多いため、複数の分岐が同時に発生する場合がある。しかし各 PE が同じ命令を同期して実行する SIMD では異なる分岐を同時処理できない。このため、SIMD で SPMD 処理を行うためには、実行の可能性のある命令全てを順発行的に発行する必要がある。これに関して次のプログラム (1) を例にして説明する。

```

Program {
    a();
    if(cond1) {
        b();
        if(cond2) c();          (1)
    } else d();
    e();
    while(cond3) f();
    g();
}

```

a(),b(),... は分岐命令を含まない命令語列であり、cond1,cond2,... は分岐条件を表す。このプログラム構造は図 2 の様に表すことができ、分岐、ループおよび合流点を節、分岐命令を含まない命令語列を実線、分岐のみで命令語

列を含まないものを点線で表す。この有向グラフから、a,b,c,d,e,f,...,g の順で命令列を発行することにより、処理可能となる。

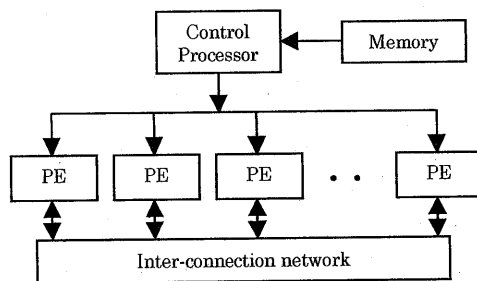
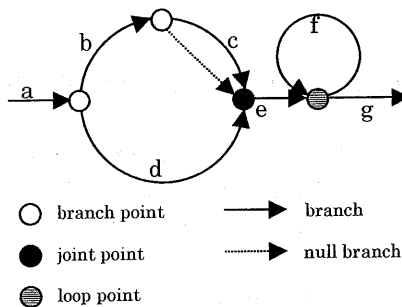


図 1 SIMD マシンの概略



○ branch point → branch
● joint point - - - - - null branch
● loop point

図 2 プログラム構造

条件分岐命令を持たない典型的な SIMD マシンではマスクやフラグによる PE のアクティビティ制御をもちいて条件分岐処理される。具体的に言えば、分岐条件が満たされた PE はフラグがセットされるので、該当 PE のみ実行できるように、CP は命令にタグを付加して発行することにより実現する。この方式は構造が簡単な上に、分岐命令を必要としない利点があるが、1分岐に対して1フラグを必要とする。例えば、プログラム (1) では、PE に c の命令列を実行させるためには、2つのフラグが必要である。代表的な SIMD マシンである CM-2[6]では4つのフラグを持つが、SPMD プログラムを処理するには多数のフラグが必要であり、不十分である。

この問題を解決するため、我々は従来方式と併用可能な、アドレス放送を用いた新しい分岐支援機構を提案した。次節ではその機構

するものと仮定する。そしてアドレス6の命令が発行されたとき、待機しているPEはTARの値と一致するので、この命令から実行を再開する。分岐条件が成立しないPEはアドレス5の命令で無条件分岐するが、アドレス6の命令を実行しないので、TARにアドレス7を設定して待機する。最終的にアドレス7の命令がCPから発行された時点で全PEは活動状態となる。この時のアクティビティの変化を表1に示す。これは2台のPEで処理する例で、アドレス3の分岐命令で、分岐するPE0と分岐しないPE1の場合を想定している。最上段はCPが発行する命令語のアドレスを、時間順に並べたものである。

```

1.   lda   x
2.   cmp   y
3.   jm    els
4.   lda   b
5.   jmp   fi
6. els:  lda   c
7. fi:   sta   a

```

(2)

・後方分岐

後方分岐はループ処理などに使用される。条件分岐の場合はORバスなどを用いて条件成立のPEを検出する手法が一般的である。存在する場合はCPが後方分岐し、分岐先アドレスから命令発行を継続するため、PEは動作継続する。条件が成立しないPEは待機する。ただし無条件後方分岐については、前方分岐と異なりCPは直ちに分岐処理を行う。プログラム(3)は、次のプログラム、

```
do { x=x-y } while(x>=y);
```

をコンパイルしたものを想定したものである。

```

1.   lda   x
2. do:  sub   y
3.   cmp   y
4.   jm    do
5.   sta   x

```

(3)

CPはアドレス1から順次命令発行を行い、アドレス4の分岐命令で、条件成立するPEが存在すると仮定して、アドレス2から命令

発行を行う。不成立の場合、AF=0および分岐命令の次命令語アドレス5をTARに設定し、待機状態となる。条件成立するPEがなくなるとCPはアドレス5の命令を発行し、待機状態にあるPEは動作を再開する。この時のアクティビティの変化も表1に示す。仮定は表1の場合と同様であるが、アドレス4の命令語で分岐する点で異なる。

表1 アクティビティの変化例
プログラム(2)の場合

ADR	1	2	3	4	5	6	7
PE0	A	A	A	N	N	A	A
PE1	A	A	A	A	A	N	A

プログラム(3)の場合

ADR	1	2	3	4	2	3	4	2	3	4	5
PE0	A	A	A	A	A	A	A	A	A	A	A
PE1	A	A	A	A	N	N	N	N	N	N	A

A:Active, N:Not active をそれぞれ示す。

3.3 サブルーチンコール

動作中の全PEを対象とするコール、リターン処理は、従来のSIMDと同様CPのみの処理で、動作中のPEは実行を継続する。条件付コールの場合は後方条件分岐と同様である。また、コール命令の次命令アドレスが戻りアドレスであるから、このアドレスをTARに設定することで、条件リターンが可能である。これは前方条件分岐と同様な動作である。

ここで、再帰コールのように、サブルーチンからサブルーチンが呼び出される場合には、MIMDと同様、リターンアドレスをPEのローカルメモリに保存する必要がある。再帰コールを例に、この場合の動作状況を図3に示す。この例では2台のPEが関数の再帰コールを行った場合の実行状況である。PE0は5回、PE1は3回コールした場合を想定する。横軸は発行された命令列、縦軸は処理中のcall levelを示す。図中のc,rはそれぞれcall,return命令を意味し、その他の空欄には何らかの命令列が発行されるものとする。図中の○印で

挟まれた点線部分は、PE が待機中であることを示している。この図の PE1 のように、CP が発行するコール/リターン命令と PE のコールレベルは一致しない場合があり、正しい動作を行うには、リターンアドレスをローカルメモリなどに保存する必要がある。

4. 命令発行順序制御

この節では CP の命令発行順序制御、および命令スキップ機構について述べる。プログラム (1) の例では、a,b,c,d,e,f,f...,g の命令列発行順序により処理が可能である。ここで、命令列 a を終了し、cond1 において、命令列 b に分岐する PE が存在しない場合、命令列 b,c の処理は不要となるので、これらの命令発行省略により処理時間の短縮が可能になる。従来の SIMD では、分岐条件判定の後、PE のアクティビティを検査した上で、命令発行省略を明示的に処理するようにプログラムされる。一方、本方式ではこれらの命令発行省略を一連のハードウェアで行うことを提案する。つまり、命令列 b の最初の命令語を発行した時点でアクティブな PE が存在しない場合、自動的に命令発行を中断し、命令列 d を発行させることにより、命令発行省略を行うものである。実現にあたり、予めコンパイラが命令列発行順序を記した順序テーブルを作成しておく。内容は、命令列中の命令数、命令列の先頭アドレス、次命令列および中断時の命令列の情報に対するインデックスが含まれている。CP は順序表の先頭から解釈して処理する。表 2 はプログラム (2) および (3) の順序テーブルの例である。

条件前方分岐では、CP は分岐条件を満たさない PE が存在すると仮定して命令発行を継続、PE が存在しない場合は自動的に中断して分岐先の命令列を発行する。例えばプログラム (2) のアドレス 3 の分岐命令で、全ての PE が分岐条件を満たす場合、アドレス 4 の命令語を発行した時点で、この命令語を実行する PE が存在しないため、自動的に命

令発行が中断される。その後、順序テーブルの内容からアドレス 6 の命令から発行される。同様に条件後方分岐では、条件を満たす PE が存在すると仮定して後方分岐する。存在しない場合は中断する。これら CP と PE の動作関係を表 3 に示す。

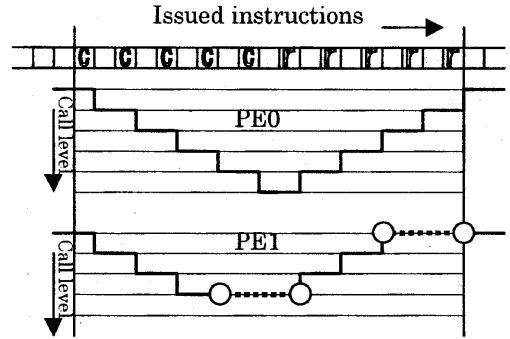


図 3 再帰コールにおける PE の動作状況

表 2 順序テーブルの例

プログラム (2) の例

命令列番号	命令数	先頭アドレス	次命令列番号	中断時の次命令列番号
1	3	1	2	3
2	2	4	3	
3	1	7		

プログラム (3) の例

命令列番号	命令数	先頭アドレス	次命令列番号	中断時の次命令列番号
1	4	1	1	2
2	1	5		

表 3 CP と PE の動作関係

	CP	PE		
	PC	AF	TAR	
前方条件付	PC+1	真	0	分岐先 ADR
		偽	不変	不変
前方条件無	PC+1		0	分岐先 ADR
後方条件付	分岐先 ADR	真	不変	不変
		偽	0	PC+1
後方条件無	分岐先 ADR		不変	不変

表中の (真、偽) は PE の分岐条件を示す。

現在の問題点は、順序テーブルをコンパイラ

ル時に静的に作成する方針のため、分岐先が動的に変化するプログラム構造に対して適用が難しいことから、分岐先は既知アドレスに限定されている。対策として、動的順序決定を行う方法が考えられるが、現在の所、検討課題である。

5. 考察

本方式を他の方式と比較すると、近いものと思われる方式に SM-1[7]等で採用されている命令によるアクティビティ制御方式がある。この方法は、本方式と同様に PE 側でも分岐命令を実行するものであるが、分岐アドレスの代わりにコンパイル時に決定される値を使用する点、そしてこの値を用いた PE 活性化命令によるアクティビティを制御する点で異なる。ハードウェア量の側面からみると、TAR、アドレス比較回路およびアドレス線が各 PE に必要であるが、アクティビティ制御のための専用命令が不要なため、PE の制御ロジックが複雑化しない利点がある。また、同一チップへの集積化を想定すれば、IO ピン数の制約が無いので、新たにアドレス線が追加されても問題はない。

提案した CP の命令スキップ機構は、中断時に順序テーブル情報から別の命令列を発行する仕組みをもつが、これは一般的なプロセッサにおける分岐予測や投機実行に関連し、これらに関する一般論が本方式にも適用できると思われる。ただし、CP と PE がオンチップ化される場合、PE の内部状態を CP から観測可能なことから、何らかの分岐予測が可能であると思われるが、それ以外の場合では、PE 側の情報が CP には未知なことから、データ依存により PE の分岐方向が揃う確率が低いことから、効果は小さいと推測される。よって、本方式で用いる一方向に仮定する分岐予測方式で十分であると思われる。

本方式に限らず、SPMD を SIMD 上で処理数場合、本質的に PE の有効利用率が低下することは不可避である。理由の一つに、CP

からの命令発行数が 1 であることが挙げられる。一つの改善手段として、命令発行数を増やす方法を検討中である。この方法は、プログラム (2) の様な例に効果を発揮する。つまり、アドレス 4 とアドレス 6 の命令は同時発行可能なため、アドレス 2 の比較命令の結果を用いて、命令語を選択して実行することにより、両方の条件の場合を同時実行するものである。この場合、分岐命令が省略できるので、少なくとも 4 命令語分の実行時間が短縮される。

6. おわりに

本稿では、従来の SIMD マシンではにおけるアクティビティ制御の改善し、MIMD マシンにおける SPMD 処理を実現するためのアドレス放送方式を用いた分岐支援機構を提案した。本方式は、簡単なハードウェアで実現され、SIMD アーキテクチャ上で SPMD に準ずるプログラムの実行を可能する特徴をもつが、今後の課題として、その処理性能を詳しく検証する必要がある。

参考文献

- [1] Patterson, D.A., Hennessy, J.L.: "Computer Organization & Design: Hardware/Software Interface", pp. 606-607, Morgan Kaufmann, 1994.
- [2] Hwang, K.: "Advanced Computer Architecture: Parallelism, Scalability, Programmability", pp. 554-556, McGraw-Hill, 1993.
- [3] Marayanan, P.J.: "Processor Autonomy on SIMD Architectures", International Conf. on SUPER-COMPUTING, pp. 127-136, 1993.
- [4] Takahashi, Y., Sano, M., Inoe, T.: "A SIMD Machine for Massively Parallel Genetic Algorithms", PDPTA '97, pp. 915-923, 1997.
- [5] Allen, J.D., Schimmel, D.E.: "Issues in the Design of High Performance SIMD Architectures", IEEE Trans. Parallel and Distributed Systems, Vol. 7, No. 8, pp. 818-829, 1996.
- [6] Tanenbaum, A.S.: "Structured Computer Organization" 3rd Ed., pp. 512-520, Prentice-Hall, 1990.
- [7] 松田、湯浅: SIMD 型超並列マシン SM-1 の概要、情処研報、92-ARC-95, pp. 127-134, 1992.