

## On Chip SIMD プロセッサ用のC言語拡張sDPCEの提案

4 E - 8

坪田浩乃 Linda Lu 田村俊之 田中健一 久間和生

三菱電機 (株) 先端技術総合研究所

E-mail tsubota@qua.crl.melco.co.jp

## 1. はじめに

LSIチップ上にSIMD(Single Instruction stream Multiple Data stream)型の並列処理機構を搭載したプロセッサ用のC言語拡張sDPCE(small Data Parallel C Extension)を提案する。

インテル社のMMXテクノロジーをはじめとして、データ並列処理機構を搭載するプロセッサが注目されている<sup>1)2)</sup>。これらのon chip SIMD機構は実現が比較的容易で、マルチメディア処理だけでなく産業分野で高速化要求の強い画像認識や音声認識処理などへの適用も有望である。標準的なS/W開発環境が整えば、今後、より多くのCPUやマイクロコントローラに採用されると予想される。しかし、現状のS/W開発環境は、標準的なライブラリ関数はあるもののアセンブリ言語によるものが主流である。本稿では、ANSIテクニカルレポートのDPCE(Data Parallel C Extension)<sup>3)</sup>を基に、on chip上の小規模並列処理に特化した言語仕様sDPCEを提案する。また、移植性および実現の容易さを考慮してC言語のプリプロセッサとする実現手法についても述べる。

## 2. sDPCE言語仕様

DPCEでは、多次元の並列型が定義できるが、sDPCEでは、認識処理でよく用いられる行列演算をそのまま記述できるようにするために、並列型を1次元のベクトル型と2次元の行列型に限定し、並列型変数の演算で、要素ごとの演算とともに、ベクトル・行列の演算も表現可能とする。また、DPCEで想定されているさまざまな大規模並列システム用のデータ配置指定やノードごとの動作指定といったシンタックスは言語仕様から削除した。

以下にDPCEとの違いを明確にしながらsDPCE言語仕様の概要をまとめる。

## 2. 1 並列型変数

(a)は、並列型変数の記述例である。キーワード

A proposal of sDPCE language for on chip SIMD processors,  
Hirono TSUBOTA, Linda Lu, Toshiyuki Tamura,  
Ken-ichi TANAKA, Kazuo KYUMA,  
Mitsubishi Electric Corporation Advanced Technology R&D  
Center

shapeを用いてベクトル型あるいは行列型の並列型指定子を宣言する。並列型変数の宣言では、データ型と並列型を併記する。並列型変数は、配列宣言も可能であり、3次元以上のデータも扱うことができる。行列型の各行は、ベクトル型にキャストすることができる。

## (a)並列型変数記述例

```
shape [SIZE]VectorS;
shape [SIZE][SIZE]MatrixS;
short:VectorS v1, v2, v3[4];
short:MatrixS m;

v3[2] = v1 + v2;
r = v1 ** v2; /* ベクトル内積 */
v2 = m ** v1; /* 行列・ベクトル積 */
[3][*]m = v1;
v2 = [4][*]m;
```

## 2. 2 並列型変数のインデックス

DPCEでは、並列型変数の各要素は、[2][3]mのように変数名の左側にインデックスを記述して表す。また、流体シミュレーションや画像フィルタリングなどでよく用いられる近傍データ間での演算を行うための特殊な表現が許されている。これらのシンタックスは、sDPCEでも記述できる。

本来演算すべき要素より一つ先の要素を演算させたい場合には、[.+1]と表現できる。近傍4点の平均を計算したい場合の記述を示す。

## (b)特殊なインデックス表現

```
shape [128][128]S;
float:S a, b;

a = ([.-1][.]b + [.+1][.]b +
[.][.-1]b + [.] [+1]b) / 4.0;
```

## 2. 3 演算子

sDPCEでは、ベクトルの内積、行列積および行列・ベクトル積用に新たに演算子\*\*を用意した。これ以外の演算子は、DPCEと同じである。以下に、DPCEで記述性を向上させている2種類の演算子を紹介する。

(1) <?, >?

2項演算子で、各々小さい方の値、大きい方の値が演算結果となる。

(2) 並列型を右辺とする代入演算子

次の9つの演算子では、右辺の全要素を演算した値を左辺に代入する。例えば、+=では、全要素の総和が<?=は、一番小さい値が左辺に代入される。

\*= /= += -= &= ^= |= <?= >?=

2.4 制御構文

where文は、演算実行をマスクし、ベクトルの中でマスクされていないデータだけを演算させるための構文である。DPCEでは、where文以外に、並列型ごとに状態という概念があり、選択的な演算実行が可能であった。しかし、小規模な並列処理では、状態を記憶するメモリをもたせたり、状態を判断させるために演算を実行するのは、大きなオーバーヘッドとなる。そこで、sDPCEでは、状態という概念は、導入せず要素ごとに実行するかどうかを決定するwhereおよびwhere-else構文のみを仕様を含める。

(c) where文

```
shape [128]S;
int:S mask, a, b;

where(mask==0)
    b = a; /* maskが0のところだけに代入*/
```

3. sDPCE言語処理系の実現

C++がサポートされていないマイクロコントローラも含め、多くのSIMD演算機構を有するプロセッサに適用するために、C言語のプリプロセッサとして実現する(図1)。このプリプロセッサにより、データ並列処理部分はライブラリに置き換えられる。生成されたCプログラムは、各々のプロセッサのCコンパイラによって、最適化された並列演算ライブラリ関数がリンクされ並列処理実行モジュールを生成することができる。

即ち、各々のプロセッサでは、最適化されたライブラリ関数を用意するだけで、汎用的な高級言語環境が実現できる。

4. sDPCEプリプロセッサの試作

PentiumII(PentiumPro+MMX)とSIMD型のNeural Network Chip(Neuro4)4)をターゲットとして、sDPCEプリプロセッサを試作した。関数コールのオーバーヘッドを削除するために、単純な関数はマクロ定義で実現

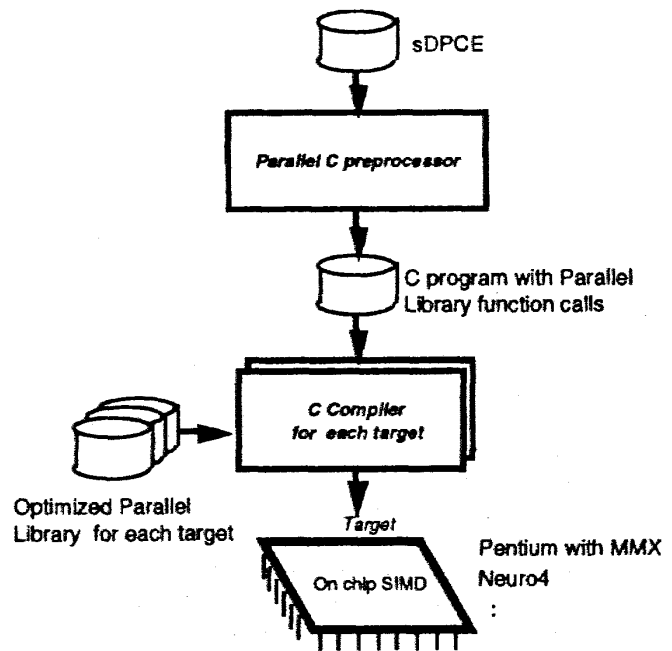


図1 sDPCEの言語処理系

するなどの最適化を行い、4並列命令を使う場合で、並列演算を行わない場合に比べ簡単な処理で2.5倍程度の性能を確認している5)。

5. おわりに

On chip SIMD機能搭載のプロセッサをターゲットとする高級言語開発環境として、sDPCE言語仕様を提案した。プロセッサエレメントごとの動作指定をwhere文に限定し、行列演算記述を可能にすることで、画像認識アルゴリズムなどをそのまま記述できるようにした。処理系は、移植性および実現の容易さを考慮してC言語のプリプロセッサとして実現する。これにより、各プロセッサは、限定された最適化ライブラリ関数を用意するだけで、汎用的な高級言語環境が実現できる。

参考文献

- 1) "The VIS Instruction Set", <http://www.sun.com/sparc/vis/index.html>
- 2) "Intel Architecture MMX Technology programmer's Reference Manual", March 1996
- 3) "Data Parallel C Extensions", Numerical C Extensions Group of X3J11 DPCE Subcommittee Technical Report, Version 1.6 X3J11/94-080 WG14/N395 December 31, 1994
- 4) Y.Kondo, "A 1.2GFLOPS Neural Network Chip Exhibiting Fast Convergence", ISSCC Digest of Technical Papers, Vol.37, pp.218-219, 1994.
- 5) Linda Lu, "An Evaluation of sDPCE language for on chip SIMDprocessors", 情報処理学会第56回全国大会論文集, 4E-09, 1998