

動的計画法向け汎用ハードウェアアクセラレータの提案

田中 飛鳥[†] 宮崎 敏明[†]

会津大学大学院 コンピュータ理工学研究科[†]

1. はじめに

動的計画法(Dynamic Programming, DP)は、最適化問題を多段決定問題に分解し、逐次的に最適解を求める手法である。DPは、バイオインフォマティクスにおける配列アライメントや、音声認識、画像マッチングなど、数多くの実用的な応用分野が存在する。そのためDPを高速処理する専用ハードウェアが提案されてきた。しかし、それらは、ある特定のDP問題に特化しており、他のDP問題に適用することは困難であった。本稿では、複数のDP問題に対応できる汎用性の高いハードウェアアクセラレータを提案する。

2. 関連研究

文献[1]は、DPに基づいたアルゴリズムであるSmith-Watermanアルゴリズムのハードウェアによる高速化を目指したものである。また、音声認識用に特化したDP処理ハードウェアも提案されている[2]。それらハードウェアは個々に専用化されており、他のDP問題にそのまま適用できるものではない。文献[3]は、多くのDP問題を汎用的に解くハードウェアを目指したものであるが、入力データの初期設定が複雑であり、実用性に欠ける。

3. 提案回路

図1にDP問題で使用される3つの演算パス(以下、単にパス)の例を示す。図1(a)は、隣接する過去に求めた3つの部分最適解を用いて、今の部分最適解を得るもので、Smith-Watermanアルゴリズム[1]などで用いられる。図1(b)と(c)のパスは、音声認識

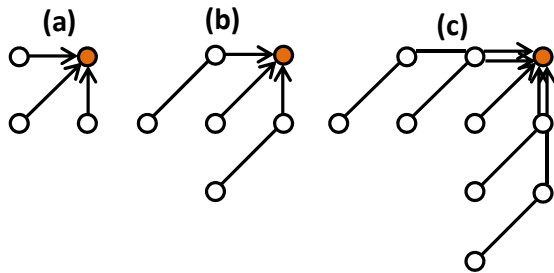


図1. 動的計画法の演算パス例

や画像処理を実現するDP問題に用いられている[4]。

図1に示した各パスは下式(1)(2)(3)の演算処理(以下、DP演算)に対応する。

$$g(i,j) = \min/\max \begin{cases} g(i-1,j) + f_1(i,j) \\ g(i-1,j-1) + f_3(i,j) \\ g(i,j-1) + f_2(i,j) \end{cases} \quad (1)$$

$$g(i,j) = \min \begin{cases} g(i-1,j-2) + 2f_3(i,j-1) + f_1(i,j) \\ g(i-1,j-1) + 2f_3(i,j) \\ g(i-2,j-1) + 2f_3(i-1,j) + f_2(i,j) \end{cases} \quad (2)$$

$$g(i,j) = \min \begin{cases} g(i-1,j-3) + 2f_1(i,j-2) + f_4(i,j-1) + f_5(i,j) \\ g(i-1,j-2) + 2f_1(i,j-1) + f_4(i,j) \\ g(i-1,j-2) + 2f_1(i,j) \\ g(i-2,j-1) + 2f_1(i-1,j) + f_2(i,j) \\ g(i-3,j-1) + 2f_1(i-2,j) + f_2(i-1,j) + f_3(i,j) \end{cases} \quad (3)$$

ここで、DP演算を行う2つの入力データ系列をそれぞれ $A=\{a_1, a_2, \dots, a_i, \dots, a_j\}$ 、 $B=\{b_1, b_2, \dots, b_j, \dots, b_i\}$ とすると、各式の左辺 $g(i,j)$ は、 $I \times J$ の2次元行列に保持できる。また、右辺の $f_k(i,j) \mid k \in \{1,2,3,4,5\}$ は、応用アプリケーションごとに定義される。例えば、式(1)の形式を持つSmith-Watermanアルゴリズムでは、 $f_1(i,j)$ 、 $f_2(i,j)$ は定数、 $f_3(i,j)$ はBLOSUMなどの置換行列(定数行列)を参照して得られる値となる。音声認識では、 $f_k(i,j)$ は局所距離となる。

提案回路は式(1)~(3)のどの形式のDP演算も解くことができるもので、Processing Element(以下、PE)を線形接続した1次元PEアレイからなる。提案回路全体の概要とPEの内部構造を図2に示す。PEは左隣接PEから受け取った値を使って計算を行い、終了後に右隣接PEへ計算結果などを転送する。隣接するPE同士の入出力線は対応する番号、例えば、in1はout1に、in2はout2に接続される。図2中、破線で囲ったFunction部は、Function base部と $Func_k$ 部からなる。ここで、 $Func_k \mid k=1,2,3$ は、式(1)~(3)中の $f_k(i,j)$ に対応する。各 $Func_k$ の出力は、

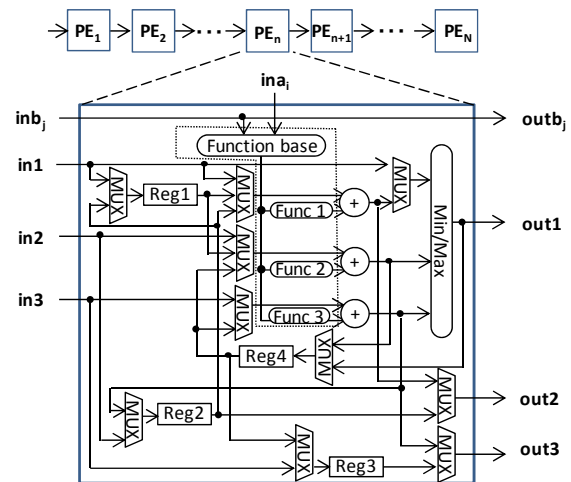


図2. 提案回路(PEアレイ)全体とPEの内部構造

A General Hardware Accelerator for Dynamic Programming

[†]Asuka Tanaka, [†]Toshiaki Miyazaki

[†]Graduate School of Information Systems, The University of Aizu

専用加算器に入力され、式(1)~(3)の右辺の各値が計算される。扱う DP 演算に合わせて本 Function 部を設定する。例えば、Smith-Waterman アルゴリズムの場合は、Function Base から置換行列の値が出力される。Func₃ は、Function Base 部から出力された値をそのまま加算器に送るが、Func₂ と Func₃ は、ユーザによって定めた定数を出力する。Min/Max 部では3つの入力値の最小値または最大値を出力する。

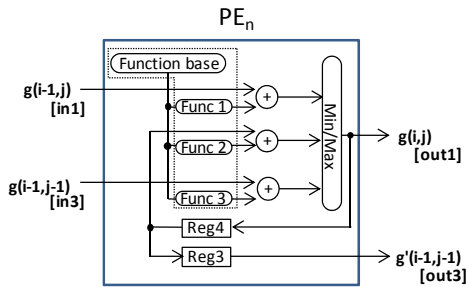


図3. 図1の演算パス (a) を実行する際に使用する PE 内のハードウェア資源

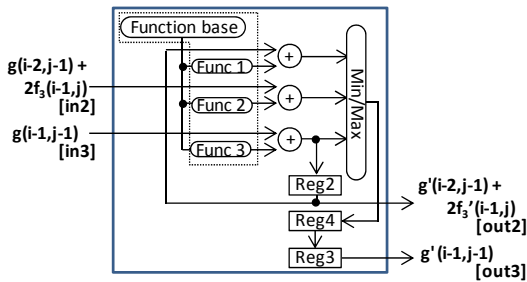


図4. 図1の演算パス (b) を実行する際に使用する PE 内のハードウェア資源

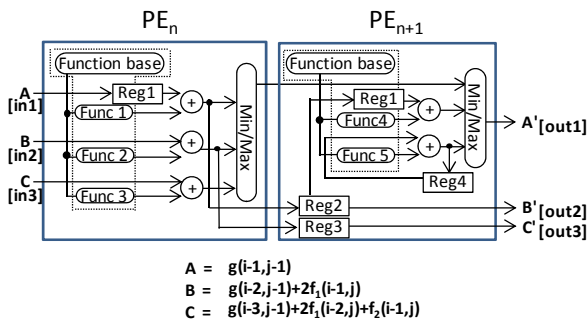


図5. 図1の演算パス (c) を実行する際に使用する PE 内のハードウェア資源

PE 内部に設けたマルチプレクサを制御することにより、図2のPEアレイを用いて図1に示した3つのパスの計算を行うことができる。図3-5に、各パスに対応したDP演算を実行する際に使用するPE内のハードウェア資源を示す。図中において、各入出力信号名に対応して[]付きで記述した信号名は、図2の入出力信号名に対応する。なお、マルチプレクサ部の表記は省略した。式(1)(2)では右辺の大小比

較の対象となる式が3つなのに対して、式(3)のそれは5つと多い。対応する演算機能を全て1つのPE内に実現すると回路規模が増大するだけでなく、式(1)(2)で表現される他のDP演算を実行する際、動作しない回路部分が多く、ハードウェア資源が無駄となる。それを解決するために、1つPEが扱うDP演算の右辺の式は最大3つとし、それ以上必要な場合は隣接するPEを組合せて、論理的に新たなPEとし、処理を実行できる機構とした(図5参照)。

4. 評価

提案回路の機能を、C言語を用いてモデル化し、動作確認を行った。結果を表1に示す。使用したPEは100個である。表中A、Bとは2つの入力データ系列A、Bの長さを表す。また、ステップ数とは、対応する演算パスを実行するのに消費するシステムクロック数のことである。本回路をシステムクロック周波数100MHzで動作させる場合、どの演算パスも約3usで処理を終了する。文献[1]は、Smith-Watermanアルゴリズムを実行する専用回路を実装し、本評価と同様のデータ系列を入力した場合、処理時間が6.87usであり、それはソフトウェアに比べ約62倍であったと報告している。我々の提案回路は、それよりも高速な動作が期待でき、しかも異なるDP演算にも対応できる汎用性がある。

表1. 図1の各演算パスの計算に必要なステップ数

演算パス	A	B	ステップ数
(a)	100	200	299
(b)	100	200	299
(c)	50	200	298

5. まとめ

本稿では、DPに基づいた複数のアルゴリズムに適用可能なハードウェアアクセラレータを提案した。提案アクセラレータは1次元のアレイプロセッサ構造であり、PEの内部構成を工夫することにより、複数の異なるDP演算を実行できる。今後は、FPGAを用いた提案回路の実機テストを実施する予定である。

参考文献

- [1] K. Benkrid, Y. Liu, and A. Benkrid, "A Highly Parameterized and Efficient FPGA-Based Skeleton for Pairwise Biological Sequence Alignment," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 17, No. 4, pp. 561-570, 2009.
- [2] J. Takahashi, S. Hamaguchi, K. Tansho, "A Modularized Processor LSI with a Highly Parallel Structure for Continuous Speech Recognition," IEEE Journal of Solid-State Circuits, Vol. 26, No. 6, pp. 833-843, June 1991.
- [3] K. I. Diamantaras, W. H. Chou, and S. Y. Kung, "Dynamic Programming Implementation on Array Processor Architectures," Journal of VLSI Signal Processing Systems, Vol. 13, No. 1, pp. 27-35, 1996.
- [4] H. Sakoe, and S. Chiba, "Dynamic Programming Algorithm Optimization for Spoken Word Recognition", IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. 26, No. 4, pp. 43-49, 1978.