

HMMの対数型アルゴリズムのFPGA上へのマッピング

2F-2

中谷 正吾、山内 宗、梶原 信樹

RWC超並列NEC研究室

1. はじめに

与えられた問題に応じて最適なハードウェアアーキテクチャに再構成可能なフレキシブルなコンピュータは、従来のノイマン型コンピュータの持つ汎用性と専用ハードウェアの持つ高速性を併せ持つものとして期待される[1]。

本研究は、このような適応型再構成可能コンピュータの有効性を具体的に示すのが目的である。そのため、現在入手可能な再構成可能デバイスであるFPGA (Field Programmable Gate Array)上にある程度複雑なアプリケーションをマッピングし、その速度を評価した。

ここで取り上げたアプリケーションは蛋白質モチーフ抽出に用いるHMM (Hidden Markov Model)の識別アルゴリズムである。従来HMMのFPGA上へのマッピングでは、簡略化されたアルゴリズムであるViterbiアルゴリズムが取り上げられてきた[2][3]。これは音声認識ではよく用いられているものであるが、蛋白質モチーフ抽出の場合これは使えず、完全なフォワードアルゴリズムを計算しなければならない。フォワードアルゴリズムをマッピングしたのは本研究が最初である。

2. 対数型フォワードアルゴリズム

HMMにおける識別は、入力シンボル列 $x_0 x_1 x_2 \dots x_{T-1}$ をそのHMM θ が出力する確率

$$P(\mathbf{X}|\theta) = \pi \mathbf{B}(x_0) \mathbf{A}\mathbf{B}(x_1) \mathbf{A}\mathbf{B}(x_2) \dots \mathbf{A}\mathbf{B}(x_{T-1}) \mathbf{F} \quad (1)$$

を計算することでなされる。ここで、 x_i は各シン

ボルを表し、 π は初期状態行ベクトル、 \mathbf{A} は状態遷移確率行列、 $\mathbf{B}(x_i)$ はシンボル出力確率行列、 \mathbf{F} は終状態列ベクトルである。式(1)の右辺を、左から右へ順に計算を進めて行くのがフォワードアルゴリズムである。しかし、この計算をそのまま行くと大きなダイナミックレンジの浮動小数点の積和演算を必要とし、現在のFPGAでは回路規模が大きくなり過ぎてマッピングは困難になる。そこで我々は対数上でこれらの演算を行う方法を用いた。

積和計算 $Z = \sum_{i=1}^N X_i Y_i$ は、つぎの漸化式で順次計算できる。

$$\begin{cases} Z_1 = X_1 Y_1 \\ Z_i = Z_{i-1} + X_i Y_i \quad \text{for } i = 2 \sim N \end{cases} \quad (2)$$

これは対数表現

$$x_i = -\log_a X_i, \quad y_i = -\log_a Y_i, \quad z_i = -\log_a Z_i \quad (3)$$

を用いると、次のように書き換えることができる。

$$\begin{cases} z_1 = x_1 + y_1 \\ z_i = z_{i-1} - F_a(x_i + y_i - z_{i-1}) \quad \text{for } x_i + y_i \geq z_{i-1} \\ \quad = x_i + y_i - F_a(z_{i-1} - (x_i + y_i)) \quad \text{for } x_i + y_i < z_{i-1} \end{cases} \quad \text{for } i = 2 \sim N \quad (4)$$

ここで a をパラメータとする関数 $F_a(x) = \log_a(1 + a^{-x})$ を用いた。関数 $F_a(x)$ はメモリーを用いてLook-Up Table方式で値を求めるようにすれば、式(4)は加減算のみで計算できるようになる。また、対数の底 a を1に十分近い値にとると、数値を整数に量子化しても十分な精度を得ることができる。さらにこの方式は比較的容易にダイナミックレンジも確保できるというメリットもある。

3. マッピング

マッピングにはザイリンクスのXC4010(1万ゲ

Implementation of Hidden Markov Model on FPGAs

Shogo Nakaya, Tsukasa Yamauchi, Nobuki Kajihara
Massively Parallel Systems NEC Laboratory, RWCP1-1, Miyazaki 4-Chome, Miyamae-ku, Kawasaki, Kanagawa
216, Japan

ト相当)を用いた。図1にその回路図の概要を示す。前述のように計算は全て整数で行われ、Look-Up Tableは外部RAMを用いる。回路規模を押さえるため、繰り返し計算は展開せずループのままにしてある。しかし、式(4)のような特殊な演算をそのままマッピングしてるため、効率的なパイプライン処理が行なわれ高速な計算が行なえるようになっている。対数上で計算すると和計算が複雑になるが、積和演算の初項計算のときはこの和計算回路をバイパスするようにし、高速化を図った。また、特殊なアンダーフロー処理やオーバーフロー処理は汎用CPUで行なうと何ステップもかかり冗長になるが、本研究のようなハードウェアマッピングでは簡単な一段の論理回路で済み、これも高速化の要因になっている。

4. 結果と議論

実際にLZモチーフ抽出に使われている状態数12のHMMを用い、FPGA(XC4010)と汎用プロセッサ(R4400・150MHz)の処理速度を比較した。その結果処理速度はXC4010の方が4倍速いことがわかった。また整数演算では現在最高速クラスのPentium Pro(200MHz)と比べても2倍近い性能をもつことがわかった。

XC4010はすでに速度の面でも集積度の面でも

旧世代のFPGAである。最新のXC4025Eのような高速高集積FPGAを用いフォワードアルゴリズムとバックワードアルゴリズムを同時に実行することにより、さらにPentium Proの7倍の処理速度を実現できると見積られる。

Pentium Pro(200MHz)は0.35ミクロンテクノロジーであるのに対し現在最新のFPGAでも0.5ミクロンテクノロジーで作られている。同じテクノロジーで比較し、かつコンピュータに適した専用の再構成可能デバイスを作成すれば、汎用プロセッサの数十倍の性能が得られるものと期待される。

参考文献

- [1] T. Yamauchi, et al., 'SOP: Adaptive Massively Parallel System', '95 RWC Joint Symp. TR-95001, pp.117-118, (1995)
- [2] J. M. Carrera, et al., 'Architecture of a FPGA-based Coprocessor:the PAR-1', IEEE Symposium on FPGAs for Custom Computing Machines, pp.20-29, (1995)
- [3] H. Schmit and D. Thomas, 'Hidden Markov Modeling and Fuzzy Controllers in FPGAs', IEEE Symposium on FPGAs for Custom Computing Machines, pp.214-221, (1995)

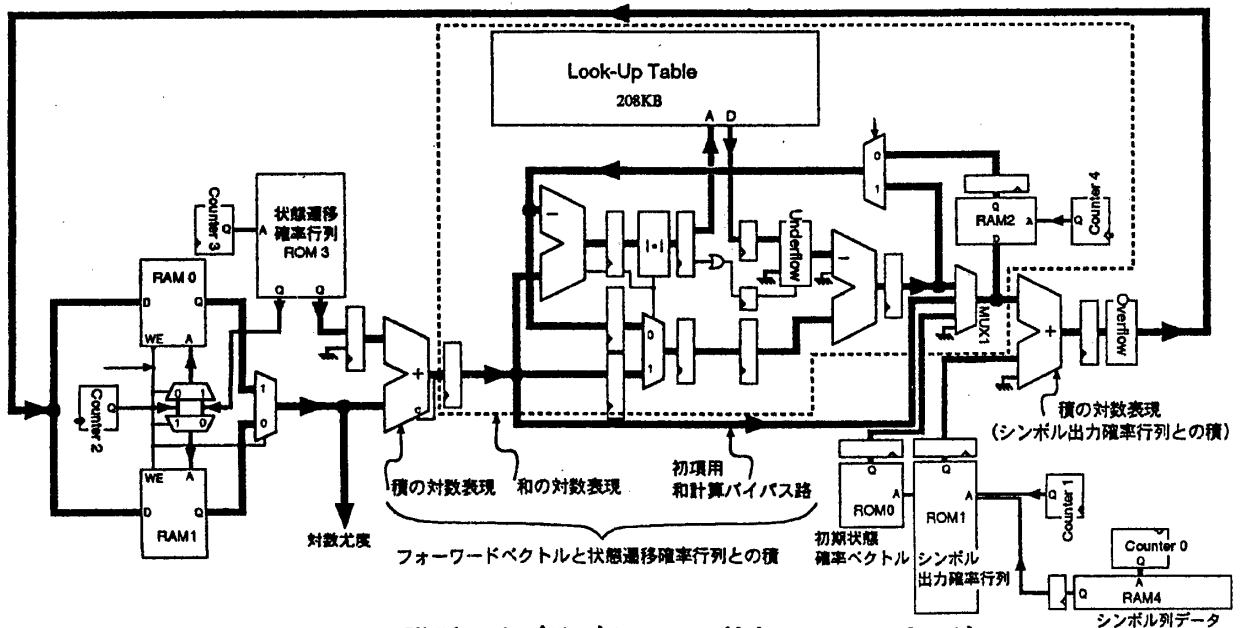


図1 HMM識別アルゴリズムのハードウェアマッピング