Vol.2011-SLDM-149 No.38
Vol.2011-EMB-20 No.38
2011/3/19

多重高速保存型一括並列処理による省メモリな音声認識用 HMM 計算回路

島崎 亮[†1] 中村一博[†1]
高木一義[†1] 高木直史[†2]

本稿では、多重高速保存型一括並列処理による省メモリな音声認識用 HMM 計算回路を提案する。入力音声内の単語を認識する単語音声認識処理では、出力確率計算と最尤推定処理の二つの処理からが行われる。この処理は HMM 出力確率計算が主であり、これが膨大な計算時間やメモリ量を必要とする。提案する VLSI アーキテクチャでは出力確率計算と最尤推定処理を行う回路の多重化を行う事で、複数の HMM についての計算を同時に行う。入力データの効率的な共有により、処理に必要なメモリ量の削減を図る。

Memory Efficient VLSI Architecture of Output Probability and Likelihood Score Computations for HMM-based Word Recognition Using Multiple Fast Store-based Block Parallel Processing

Ryo Shimazaki, Kazuhiro Nakamura,
Kazuyoshi Takagi and Naofumi Takagi

We present a VLSI architecture for output probability computations (OPCs) of continuous Hidden Markov Models (HMMs) and likelihood scorer computations (LSCs) which supports *multiple fast store-based block parallel processing* (*MultipleFastStoreBPP*). We demonstrate the MultipleFastStoreBPP for HMM-based word recognition, which exploits full performance of the FastStoreBPP and present a high-speed VLSI architecture that supports it. A comparison demonstrates the efficiency of the architecture.

1. はじめに

将来のユビキタスコンピューティングの実現に向けて、音声認識やジェスチャー認識などの自然なヒューマンインタフェースを備えた携帯機器が求められている。音声認識はその１つとして注目されており[1)]、カーナビゲーションシステムや携帯電話、PDA などにも用いられている。しかし音声認識処理の計算量は大きく、携帯機器では精度や語彙数が制限されてしまう為、実時間低消費電力で高精度な認識を音声認識処理に特化した専用ハードウェアで行う研究が進められている[2)–5)]。この場合、認識処理に必要なデータを回路内メモリに保持し、複数の PE(Processing Element) による並列計算を行う事で高速化を実現する[2),6)]。

本稿では、一括並列処理の１つである多重高速保存型一括並列処理による省メモリな音声認識用 HMM 計算回路を提案する。入力音声内の単語を認識する音声認識処理は、出力確率計算と最尤推定処理の二つの処理から成る。この処理は HMM 出力確率計算が主であり、これが膨大な計算時間やメモリ量を必要とする。提案する VLSI アーキテクチャでは出力確率計算と最尤推定処理を行う回路の多重化を行う事で、複数の HMM についての計算を同時に行う。入力データの効率的な共有により、処理に必要なメモリ量の削減を図る。

2. 音声認識

音声認識は人が発した音声を入力とし、予め登録された音声モデルから入力信号に最も近い音声モデルを出力する処理である。音声モデルは、単語や音素、音韻等の認識対象をモデル化したものであり、音声認識などには時間的な変動に頑健な隠れマルコフモデル（Hidden Markov Model; HMM）が使用される[2)]。孤立単語音声認識では、あらかじめ単語毎に対応するモデルを用意しておき、入力音声とこれを比較して最も近いモデルを出力する。

2.1 HMM ベースの音声認識処理

HMM を利用した音声認識の処理は大きく 3 つにわかれる。特徴抽出、出力確率計算、Viterbi アルゴリズム等による尤度計算である。特徴抽出処理は認識処理の前処理として、音声信号から認識処理に必要な情報を特徴ベクトルとして取り出す。これは通常 DSP などによりリアルタイムに処理される[3)]。次に特徴ベクトルを入力として、出力確率計算を行う。

†1 名古屋大学
Nagoya University
†2 京都大学
Kyoto University

Vol.2011-SLDM-149 No.38
Vol.2011-EMB-20 No.38
2011/3/19

最後に Viterbi アルゴリズムによる尤度計算を行う。

以下に出力確率の計算式を記す。出力確率計算は加算、減算、乗算、累算により、式 (1) の $\log b_j(\mathbf{o}_t)$ を求める計算である。

$$\log b_j(\mathbf{o}_t) = \omega_j + \sum_{s=1}^{P} \sigma_{js}(o_{ts} - \mu_{js}) \quad (i = 0, 1, 2, \ldots, N) \tag{1}$$

ここで $\mathbf{o}_t$ は時刻 t における入力音声の特徴ベクトル $\mathbf{o}_t = (o_{t0}, o_{t1}, \ldots, o_{tP})$、$P$ は次元数である。一つの HMM モデルは N 個の状態を持っており。HMM 毎にある状態 i からある状態 j へと移動する確率 a_{ij} が決まっている。HMM の出力確率 $b_j(\mathbf{o}_t)$ は状態 j において、シンボル $\mathbf{o}_t$ が出力される確率を表している。μ, ω, σ の各ベクトルはモデルによって値が異なり、出力確率計算を行う前にあらかじめ求めておくことが可能な各HMM の学習パラメータである[2)3)]。

次に、このようにして求まった出力確率から Viterbi アルゴリズムを用いて、モデルにおける特徴ベクトル系列の対数尤度を求める。ここで対数尤度とはそのモデルにおいて特徴ベクトル系列が出力される尤もらしさである。尤度は次の式 (2),(3),(4) によって計算される[2)3)]。

$$\delta_0(j) = \log \pi \qquad (1 \le j \le N) \tag{2}$$

$$\delta_t(j) = \min_{i=j-1,j} [\delta_{t-1}(i) + \log a_{ij}] + |\log b_j(\mathbf{o}_t)| \qquad (1 \le t \le T,\ 1 \le j \le N) \tag{3}$$

$$P^* = \min_{1 \le j \le N} [\delta_T(j)] \qquad (t = T) \tag{4}$$

T は特徴ベクトル系列の総数を表す。$\log \pi$ は HMM 毎に決められた学習パラメータの一つである。この対数尤度 P^* を求める計算を最尤推定という。あらかじめ学習されている単語 HMM の数だけこの出力確率計算と最尤推定を繰り返し、一番尤もらしいモデルが認識結果として出力される。図 1 に V 個の HMM に対する出力確率計算と最尤推定処理の流れを示す。 図 1 における"*Partial computation of* $\log b_j(\mathbf{o}_t)$"は二つの加算と二つの乗算を含む式 (1) の演算を一回行うことを意味している。この演算を Loop A 内で P 回繰り返す事で $\log b_j(\mathbf{o}_t)$ の値が得られる。図 1 における"*Partial computation of* $\log \delta_t(j)$"は三つの加算と二つの選択処理、そして一つの比較処理を含む式 (2) と (3) の演算を一回行うことを意味している。この演算を Loop B 内で N 回繰り返す事で $\log \delta_t(j)$ の値が得られる。

2.2 高速保存型一括並列処理

HMM に基づく音声認識回路において、高速かつ小型 · 省メモリな回路の実現の為に、

```
for (v = 1; v ≤ V; v = v + 1 ) {     // Loop D
 for (t = 1; t ≤ T;  t = t + 1 ) {    // Loop C
  for (j = 1; j ≤ N; j = j + 1 ) {    // Loop B
   for (p = 1; p ≤ P; p = p + 1 ) {    // Loop A
    Partial computation of log b_j(O_t);
   }  // end of Loop A
   Partial computation of log δ_t(j);
 } } }  // end of Loops B, C and D
```

図 1 HMM を利用した音声認識処理
Fig. 1 Flow of voice recognition with HMM.

StoreBPP[6] の発展的改良による出力確率計算と最尤推定処理の高速化手法が提案されている[7]。この手法は高速保存型一括並列処理 (*fast store-based block parallelprocessing, FastStoreBPP*) と呼ばれている。従来の StoreBPP では、$M/2$ 個の PE をサイズ $P \times M$ のレジスタアレイ Reg$\mathbf{O}$ に接続し、$M/2$ 個のデータを読出すと同時に各 PE に送る事で、$M/2$ 並列で式 (1) の計算を行っていた。FastStoreBPP では、M 個の PE をサイズ $P \times M$ のレジスタアレイ Reg$\mathbf{O}$ に接続し、M 個のデータを読出すと同時に各 PE に送る事で、M 並列で式 (1) の計算を行う。これは出力確率計算部の入力バスのバス幅を 2 倍に拡張することにより、StoreBPP の性能を更に引き出す改良手法であった。これにより、メモリ量を増やす事無く処理時間を短縮したり、処理時間を増大させる事なく回路面積やメモリ量を削減する事が出来た。

FastStoreBPP に基づき出力確率計算と最尤推定処理を実行するフローチャートを図 2 に示す。 P' は M 個の出力確率計算の結果が得られる周期である。$\lceil M/P' \rceil$ 個の $PE2$ をパイプライン式に動作させることにより、Loop A と Loop E' 内の処理は同時に実行できる。これにより各 Loop E' は Loop A の次の計算が始まる前に終了する。

図 3 に FastStoreBPP に基づいた出力確率計算部と最尤推定処理部の VLSI アーキテクチャを示す。このアーキテクチャは出力確率計算の為に 2 つのレジスタ群と 2 つのレジスタ、$PE1$ を M 固有している。また最尤推定処理の為に 4 つのレジスタ群、2 つのレジスタ、単一の $PE2$ を保有している。Reg$\mathbf{O}$ には M の入力特徴ベクトル $\mathbf{O}_{t'+1}$, ..., $\mathbf{O}_{t'+M}$ を格納する。Regμ と Regσ にはそれぞれ HMM パラメータ μ_{jp} と σ_{jp} を格納する。Regω には HMM パラメータ ω_j と中間結果を格納する。Regδ は算出された出力確率値を保存する。Regδ_l は中間結果として、$(v'+1)$ 番目から $(v'+L)$ 番目の L 個の HMM に対する $\log \delta_{t'+M}(j)$ $(1 \le j \le N)$ をそれぞれ N 個格納する。Regδ_j と Regδ_{j-1} はそれぞれ、$\log \delta_t(j)$ と $\log \delta_t(j-1)$ $(t'+1 \le t \le t'+M)$ の中間結果を M 個格納する。Reg$\mathbf{a}_{j,j}$ と Reg$\mathbf{a}_{j-1,j}$ はそれぞれ、v 番目の HMM における HMM パラメータ $\log a_{j,j}$ と $\log a_{j-1,j}$ を格納する。

Vol.2011-SLDM-149 No.38
Vol.2011-EMB-20 No.38
2011/3/19

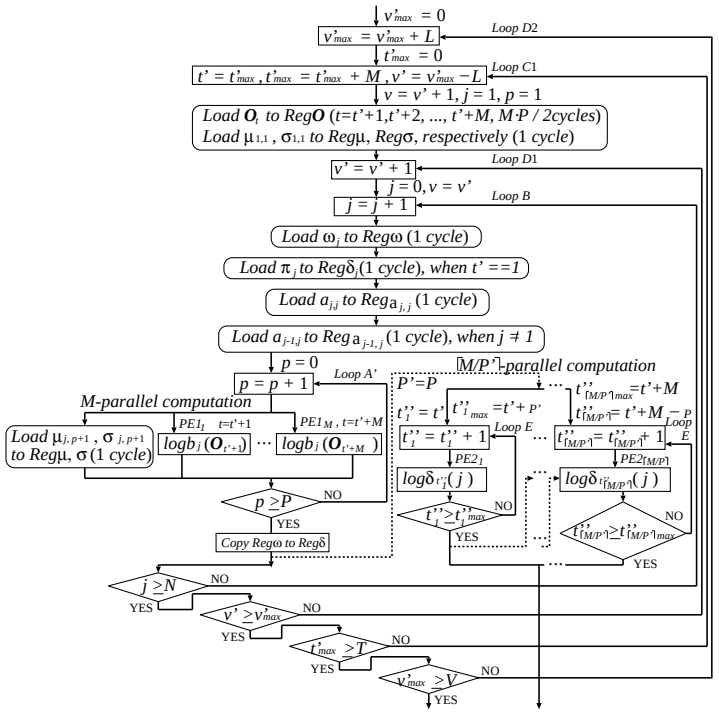

図 2 FastStoreBPP に基づく出力確率計算と最尤推定処理のフローチャート
Fig. 2 Flowchart of OPC and LSC based on FastStoreBPP.

出力確率計算部は図 2 に示す Loop C1 内で RAM から M 個の入力特徴ベクトルを読み込みそれらを Reg$\mathbf{O}$ に格納する処理から動作を開始する。v 番目の HMM のパラメータは ROM から読み出され Regμ,Regσ,Regω,Regδ_j そして Reg$\mathbf{a}_{j,j}$ に格納される。格納された M 個の入力特徴ベクトル $\mathbf{O}_{t'+1}$, ...,$\mathbf{O}_{t'+M}$ に対し、M 個の $PE1$ が格納された μ_{ij} と σ_{ij} の値を用いて、M 個の中間結果を同時に算出する。全ての $PE1$ は HMM パラメータの値を共有している。これと同時に v 番目の HMM における HMM パラメータ μ_{jp+1} と σ_{jp+1} が ROM から読み出され、それぞれ Regμ と Regσ に格納される。次の M 並列計算においては、その格納された二つのパラメータが利用される。v 番目の HMM に対する M 個の出力確率 $\log b_j(\mathbf{O}_{t'+1})$,$\cdots$, $\log b_j(\mathbf{O}_{t'+M})$ は図 2 に示される

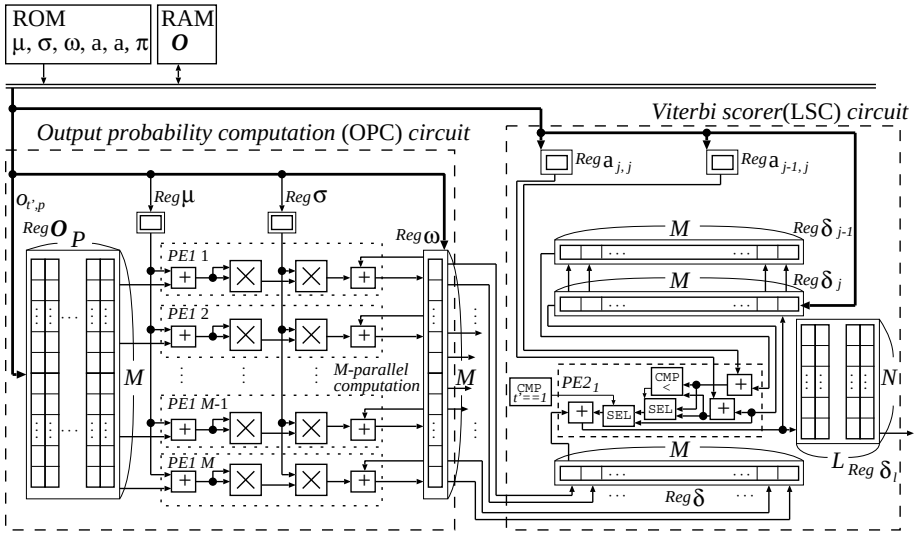

図 3 FastStoreBPP に基づく VLSI アーキテクチャ ($\lceil M/P' \rceil = 1$)
Fig. 3 VLSI architecture of OPC and LSC based on FastStoreBPP ($\lceil M/P' \rceil = 1$)

Loop A' の中で同時に計算される。算出された結果が Regω から Regδ へとコピーされてから、次の $\log b_{j+1}(\mathbf{O}_{t'+1})$,$\cdots$, $\log b_{j+1}(\mathbf{O}_{t'+M})$ について演算が始まる。$\log \delta_t(j)$ $(t''+1 \le t \le t''+M)$ は式 (2.11) に基づき、単体の $PE2$ がコピーされた出力確率を用い、複数サイクルかけて算出される。v 番目の HMM についての $M \cdot N$ 個の出力確率値は図 2 における Loop B 内で算出される。$v'+1$ 番目から $v'+L$ 番目までの L 個の HMM についての $M \cdot N \cdot L$ 個の出力確率値は、同一の入力特徴ベクトル $\mathbf{O}_{t'+1}$, ..., $\mathbf{O}_{t'+M}$ を用いて、図 2 における Loop D1 内で算出される。L 個の HMM についての N 個の中間結果 $\log \delta_{t'+M}(j)$ $(1 \le j \le N)$ は Regδ_l の中に格納される。Loop C1 において $v'+1$ から $v'+L$ までの HMM に対する $M \cdot N \cdot L \cdot \lceil T/M \rceil$ の出力確率値が算出され、最終的に Loop D2 において全ての HMM に対する出力確率と $\log \delta_t(j)$ $(1 \le j \le N, 1 \le t \le T)$ が求められる。図 3 に示される FastStoreBPP の為の最尤推定のアーキテクチャは $\lceil M/P' \rceil \ge 2$ となる場合にも適用可能である。この最尤推定計算部はパイプライン用のレジスタと $\lceil M/P' \rceil$ 個の $PE2$ から構成される。

Vol.2011-SLDM-149 No.38
Vol.2011-EMB-20 No.38
2011/3/19

3. 提案する音声認識用 HMM 計算回路

FastStoreBPP において、一つの単語 HMM に対する演算は図 2 における Loop D1 内で実行されている。この一つの HMM に対する計算を多重化することで、複数の単語に対する HMM 計算を同時に実行する。

3.1 多重高速保存型一括並列処理

FastStoreBPP は StoreBPP に対して入力バス幅の拡張を行なう事で、PE やレジスタの追加を伴わない改良で高速化を実現していた。この時のバス幅拡張は従来手法の 2 倍に留めるのが最も効率的で、これ以上に拡張しても実行速度を向上させる事は処理上不可能だった。今回アーキテクチャにおけるバス幅拡張のスケーラビリティをより高める為に、FastStoreBPP を発展させた手法を提案する。FastStoreBPP では、M 個の PE をサイズ $P \times M$ のレジスタアレイ Reg**O** に接続し、M 個のデータを読出すと同時に各 PE に送る事で、M 並列で式 (1) の計算を行っていた。提案手法では、$L' \times M$ 個の PE をサイズ $P \times M$ のレジスタアレイ Reg**O** に接続し、M 個のデータを読出すと同時に各 PE に送る事で、$L' \times M$ 並列で式 (1) の計算を行う。ここで L' は多重化する単語 HMM の数を指す。これは出力確率計算部の入力バス幅の拡張と、出力確率計算・最尤推定処理部を多重化することにより、FastStoreBPP の性能を更に引き出す手法である。これにより更なる回路面積やメモリ量の削減が可能になる。以降本研究ではこの新たな手法を MultipleFastStoreBPP と呼ぶ。MultipleFastStoreBPP に基づき出力確率計算と最尤推定処理を実行するフローチャートを図 4 に示す。 このフローチャートは OPC(出力確率計算) と LSC(最尤推定部分) の 2 つの処理のセット L' 個から構成される。2 つの処理である OPC(出力確率計算) と LSC(最尤推定部分) の構造は図 2 に示した FastStoreBPP と同じである。

図 5 に MultipleFastStoreBPP に基づいた出力確率計算部と最尤推定処理部の VLSI アーキテクチャを示す。 図 3 に示した FastStoreBPP の回路アーキテクチャと比較して、出力確率計算部と尤度推定処理部の組が L' 個存在する特徴を持つ。FastStoreBPP の回路アーキテクチャにおいては、特徴ベクトル系列を格納する為の入力バッファのサイズが $P \times M$ である時、同時に出力確率計算を行える $PE1$ の数は M であり、その全てがある一つの単語 HMM のパラメータを入力にして並列に計算を行っていた。MultipleFastStoreBPP に基づくアーキテクチャでは入力バッファのサイズ $P \times M'$ について、ある一つの単語 HMM に対して出力確率計算を行える $PE1$ の数は M' であり、同時に計算を行える単語 HMM の数は L' である。出力確率計算の際に単語 HMM 毎に異なる値である μ, σ を格納する為

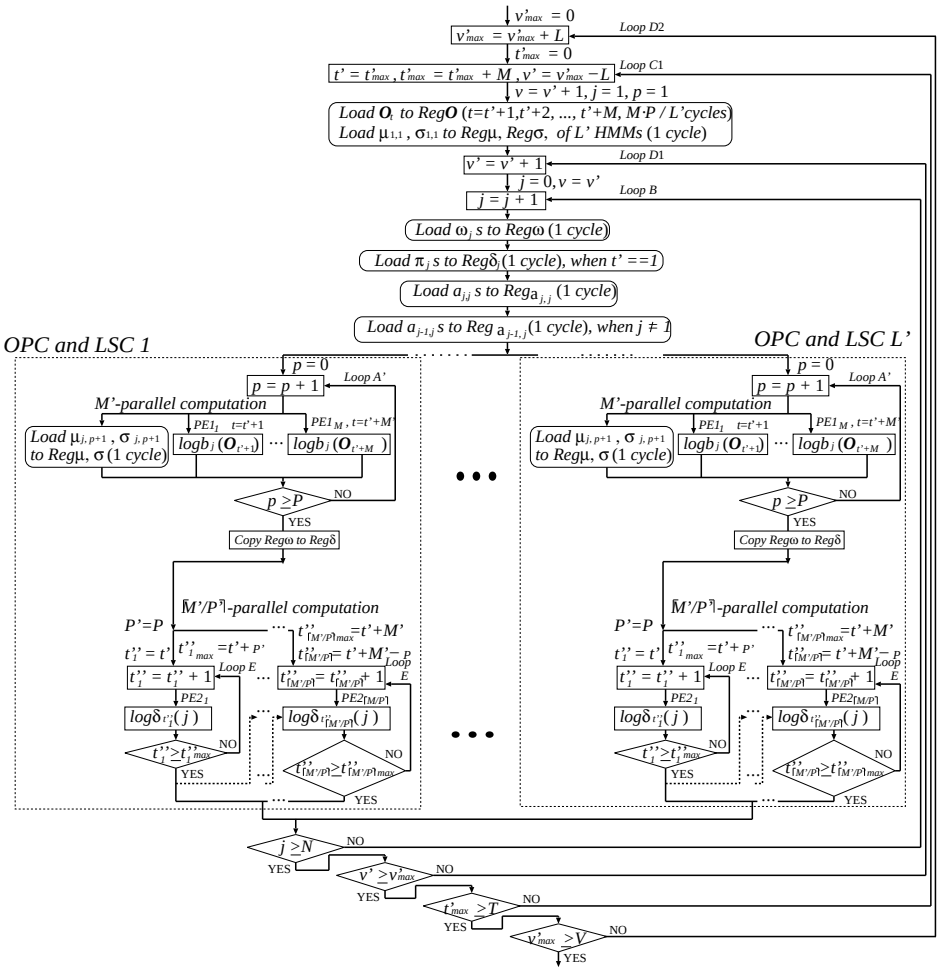

図 4 MultipleFastStoreBPP に基づく出力確率計算と最尤推定処理のフローチャート
Fig. 4 Flowchart of OPC and LSC based on MultipleFastStoreBPP.

の Regμ, Regσ が L' 組存在する。

FastStoreBPP による出力確率計算では、1 サイクル毎に Regμ, Regσ に格納される値を更新しつつ計算を行っていた為、1 サイクルで μ, σ の 2 データを送る為の入力バス幅が必要だったが、これ以上増やしても性能向上にはつながらなかった。本手法である Multi-

Vol.2011-SLDM-149 No.38
Vol.2011-EMB-20 No.38
2011/3/19

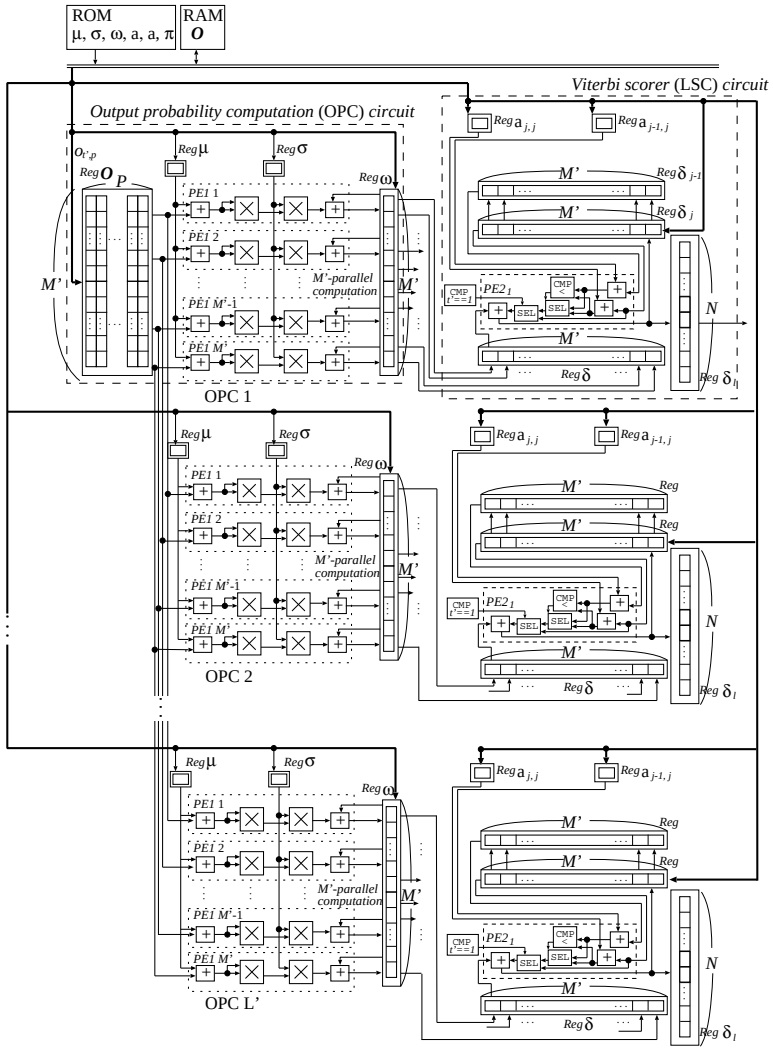

図 5 MultipleFastStoreBPP に基づく VLSI アーキテクチャ ($\lceil M/P' \rceil = 1$)
Fig. 5 VLSI architecture of OPC and LSC based on MultipleFastStoreBPP.

pleFastStoreBPP による出力確率計算では、1 サイクル毎に L' 個の単語 HMM それぞれの Regμ, Regσ に格納される値を更新しつつ計算を行っていた為、1 サイクルで μ, σ の $2L'$ データを送る為の入力バス幅が必要になる。これは回路のバス幅を 2 倍以上に拡張する事で、データ転送を一度で行える。結果入力バッファのサイズ $P \times M'$ に対し同時に計算を行える出力確率計算の PE 数は $L' \times M'$ に増え、FastStoreBPP よりも少ない入力バッファのサイズで同時に多くの計算ができる。これにより処理にかかる実時間を抑えたまま、入力バッファのメモリ量を削減できる。最尤推定処理部分も同時に L' 個の単語の尤度が計算できるように L' 組用意されている。図 5 における各 Viterbi Scorer 内の Regδ_l のサイズは FastStoreBPP では $N \times L$ だったのに対し、N に減っている。

4. 詳細設計と評価

本章では、提案する MultipleFastStoreBPP アーキテクチャと従来の FastStoreBPP アーキテクチャ[7] との比較を行う。表 1 に従来の FastStoreBPP に基づく回路と、提案する MultipleFastStoreBPP に基づく回路の各アーキテクチャにおいて必要なレジスタサイズを示す。 ここで x_μ,x_σ,x_o ,x_a はそれぞれデータ μ_{jp},σ_{jp},o_{tp},a_{jj} のビット幅を表している。

表 1 各回路に必要なレジスタサイズの比較
Table 1 Comparison with register size of each circuits

FastStoreBPP[7] [bit]	
OPC	$P \cdot M \cdot x_o + x_\mu + x_\sigma + M \cdot x_f$
Viterbi scorer	$(N \cdot L + 2 \cdot M + (M \cdot (\lceil M/P \rceil + 1))/2) \cdot x_f + 2 \cdot x_a$
total	$(P \cdot M \cdot x_o + x_\mu + x_\sigma + M \cdot x_f) + (N \cdot L + 2 \cdot M + (M \cdot (\lceil M/P \rceil + 1))/2) \cdot x_f + 2 \cdot x_a$
MultipleFastStoreBPP [bit]	
OPC	$P \cdot M' \cdot x_o + (x_\mu + x_\sigma + M' \cdot x_f) \cdot L'$
Viterbi scorer	$\{(N + 2 \cdot M' + (M' \cdot (\lceil M'/P \rceil + 1))/2) \cdot x_f + 2 \cdot x_a\} \cdot L'$
total	$P \cdot M' \cdot x_o + \{x_f \cdot (N + 3M' + M' \cdot (\lceil M'/P \rceil + 1)/2) + 2 \cdot x_a + x_\mu + x_\sigma\} \cdot L'$

N,P,M はそれぞれ HMM の状態数、入力の特徴ベクトルの次元数、そして一度に処理する入力特徴ベクトルの数を指す。L' は多重化により一度に計算する単語 HMM の数で、M' は多重化をした際の一度に処理する入力特徴ベクトルの数を指す。OPC と Viterbi scorer,total はそれぞれ出力確率計算部と最尤推定部のレジスタサイズとその合計を指し示している。

表 2 に従来の FastStoreBPP の手法に基づく回路と、提案する MultipleFastStoreBPP

Vol.2011-SLDM-149 No.38
Vol.2011-EMB-20 No.38
2011/3/19

に基づく回路の各アーキテクチャにおいて出力確率計算と最尤推定処理の計算に必要な処理時間をサイクル単位で示す。 ここで V,T はそれぞれ HMM のモデル数、総特徴ベクトル

表 2 各回路に必要な処理時間の比較
Table 2 Comparison with processing time of each circuits

FastStoreBPP[7] [cycle]	
OPC	$\lceil V/L \rceil \{P \cdot \lceil M/2 \rceil + (1+P) \cdot L \cdot N\} \lceil T/M \rceil$
Viterbi scorer	$\lceil V/L \rceil \{L \cdot N\} \lceil T/M \rceil$
total	$\lceil V/L \rceil \{\lceil P \cdot M/2 \rceil + (2+P) \cdot L \cdot N\} \lceil T/M \rceil$
MultipleFastStoreBPP [cycle]	
OPC	$\lceil V/L' \rceil \{\lceil P \cdot M'/L' \rceil + (1+P) \cdot N\} \lceil T/M' \rceil$
Viterbi scorer	$\lceil V/L' \rceil \{3 \cdot N\} \lceil T/M' \rceil$
total	$\lceil V/L' \rceil \{\lceil P \cdot M'/L' \rceil + (4+P) \cdot N\} \lceil T/M' \rceil$

数を指す。また L は出力確率計算において図 2 に示すループ D1 内において、同一の入力特徴ベクトルで一度に計算する HMM の数を表す。OPC と Viterbi scorer,total の欄はそれぞれ出力確率計算部と最尤推定部において計算に必要なサイクル数とその合計を表している。回路内で同時に動作する $PE1$ の総数を全て同じ M 個に統一した場合 ($M' = M/L'$)、表 1,2 より、従来の FastStoreBPP のアーキテクチャと比較して、提案手法である MultipleFastStoreBPP のアーキテクチャは少ないレジスタで動作し、必要な処理サイクル数も少ない事が分かった。レジスタサイズは入力データである特徴ベクトルの効率的な共有により削減される。また必要な処理サイクル数はバス幅の拡張により、特徴ベクトルの読み込みに必要なサイクル数が削減される。

表 3 に提案する MultipleFastStoreBPP に基づく各回路に必要なレジスタサイズ、処理時間、各 PE 数を従来手法の FastStoreBPP に基づく回路と比較した結果を示す。ここでは $N = 32$, $P = 38$, $T = 86$, $x_\mu = 8$, $x_\sigma = 8$, $x_f = 24$, $x_o = 8$, $x_a = 8$ そして $V = 800$ と仮定した。これは近年の孤立単語音声認識回路[2),5)] において使用されていた値と同じである。FastStoreBPP においては、$M = 44$, $M = 29$ の場合を仮定する。MultipleFastStoreBPP においては、$M' = 11$, $L' = 4$ の場合を仮定する。

5. ま と め

本研究では多重高速保存型一括並列処理による省メモリな音声認識用 HMM 計算回路を提案した。処理部の多重化を行った MultipleFastStoreBPP は、従来の FastStoreBPP に対し高速に動作し回路に必要なレジスタサイズも削減する事を示した。MultipleFastStoreBPP

表 3 FastStoreBPP と MultipleFastStoreBPP の性能比較
Table 3 Comparison with each performances

	Register size [bit]	Processing time [cycle]	$\#PE1s$	$\#PE2s$
FastStoreBPP($M = 29$)[7]	15,472	3,345,600	29	1
FastStoreBPP($M = 44$)[7]	22,000	2,315,520	44	2
MultipleFastStoreBPP ($M = 44, M' = 11, L' = 4$)	10,768	1,107,600	44	4

アーキテクチャによって、FastStoreBPP の最大限の性能を引き出せる事が確認された。結果提案した HMM 計算回路においては大幅なメモリ量の削減や高速化を実現できた。

参 考 文 献

1) A. Lee, T. Kawahara, and K. Shikano: Julius . an opensource real-time large vocabulary recognition engine, Proc. European Conference on Speech Communication and Technology, pp. 1691-1694 (2001).
2) S. Yoshizawa, N. Wada, and N. Hayanaga: Scalable Architecture for Word HMM-based Speech Recognition and VLSI Implementation in CompleteSystem, IEEE TRANS. ON CIRCUITS AND SYSTEMS, Vol. 53, No.1, pp. 70-77 (2006).
3) K. Miura, H. Noguchi, H. Kawaguchi, and M. Yoshimoto: A Low Memry Bandwidth Gaussian Mixture Model(GMM) Processor for 20,000-word Real-Time Speech Recognition FPGA System, Proc. of the International Conference on Field-Programmable Technology, pp. 341-344 (2008).
4) J. Hashimoto, A. Eguchi, M. Saituji, A. Yamada, and T. Kanbe: An Output Probability Computation Circuit Design for Real Time Speech Recognition, SASIMI(2007)
5) 高橋 克哉, 吉澤 真吾, 早坂 昇, 宮永 喜一: FIFO を用いた HMM 音声認識回路の小型・省電力化について, 電子情報通信学会技術研究報告. SIS, スマートインフォメディアシステム 110(74), pp. 37-42 (2010).
6) K. Nakamura, M. Yamamoto, K. Takagi, and N. Takagi: *A VLSI Architecture for Output Probability Computations of HMM-Based Recognition Systems with Store-Based Block Parallel Processing*, IEICE TRANS.INF & SYST., Vol.E93-D, No.2, pp.300–305, 2010.
7) R. Shimazaki, K. Nakamura, M. Yamamoto, K. Takagi, and N. Takagi: A High-speed VLSI Architecture of Output probability and Likelihood Scorer Computation for HMM-Recognition Systems, SASIMI 2010, Oct. 2010.