社団法人 情報処理学会 研究報告
IPSJ SIG Technical Reports

2006－ＳＬＤＭ－123（２）
2006／1／17

FTTSSに基づいた音声認識LSIの開発

中野 智崇† 朴 虎岡‡ 船田 哲男‡ 北川 章夫‡

†金沢大学 〒920-1192 石川県金沢市角間町

E-mail: †tomotaka@merl.ec.t.kanazawa-u.ac.jp, ‡{piaohg@merl.ec.t, funada@t,kitagawa@is.t}.kanazawa-u.ac.jp

あらまし 近年、パーソナルコンピュータや携帯電話などのような情報機器が低価格化、高性能化により広く普及してきている。これらの情報機器に対して、誰にとっても扱いやすいユーザーインターフェイスとして音声認識ツールが注目されている。現在の音声認識ツールはソフトウェアで行うのが一般的である。携帯電話などのモバイル機器に搭載する場合にはできるだけ消費電力を抑えたい。そこで、我々はリアルタイム音声認識LSIの回路設計を行い、CMOS0.35μm プロセスを用いて性能評価を行った。本稿では、ビタビアルゴリズムを用いた音声認識処理部について述べる。回路シミュレーションにより、リアルタイム認識に十分な認識処理時間である6.1msを得た。

キーワード 音声認識, FTTSS, HMM, ビタビアルゴリズム

Voice Recognition LSI based on FTTSS

Tomotaka NAKANO† Park Hu Gang‡ Tetsuo FUNADA‡ and Akio KITAGAWA‡

† Faculty of Engineering, Kanazawa University kakumamachi, Kanazawa-shi, Ishikawa, 920-1192 Japan

E-mail: †tomotaka@merl.ec.t.kanazawa-u.ac.jp, ‡{piaohg@merl.ec.t, funada@t,kitagawa@is.t}.kanazawa-u.ac.jp

Abstract Recently, information devices such as personal computers and cellular phones is widespread by reducing the price and making to high performance. The voice recognition tool is paid to attention for these information devices as a user interface. It is popularly practiced to equipped a voice recognition tool as a software or a middleware. In the application of the mobile devices such as cellular phones, it is suitable to implement the voice recognition on the wired logic in terms of the power consumption. Then, we designed of real-time voice recognition LSI with the CMOS0.35μm technology. In this paper, we present the voice recognition process based on Viterbi algorithm and its performance. 6.1ms the processing time for the voice recognition is enough short for real-time recognition.

Keyword voice recognition, FTTSS, HMM, viterbi argorithm

1. はじめに

半導体技術の急速な発展に伴い、パーソナルコンピュータや携帯電話の普及率は爆発的に伸びた。それに伴い様々なユーザインターフェイスが存在している。その中でも音声認識は、どのようなユーザに対しても扱いやすいため期待されている。これは例えば、ブラインドタッチのような特別な訓練をする必要がないからである。また、手足や目などがふさがっている場合などでも情報を入力できるという利点を持っている。これらの特長を生かしてパーソナルコンピュータの入力デバイスや電話サービスの自動化[1]、音声認証などの用途に用いられている[2]。

しかし、モバイル機器、特に携帯電話においては、ソフトウェアで処理を行うとバッテリー容量の問題や、携帯電話に搭載されているプロセッサの演算処理不足などにより、リアルタイム処理を実現するのが困難である。そこで、これらの問題を解決するために音声認識システムをハードウェア上で実現することは大きな利点がある。音声認識システムを考えた場合、DSPでも音声認識処理を行うことができる。しかし、現在の音声認識に用いられているHMMは認識性能が高い反面、認識のための演算に多大な負荷がかかる。よって、専用ハードウェアの開発が効果的である。

また、一般に雑音のある環境下で音声認識を行う場合、その認識率は著しく低下する。そこで我々は、FTTSS(mel-Frequency Fourier Transform of Ternarized Spectral Slope)という音声特徴パラメータを用いている。FTTSSを用いることで雑音のある環境下でも認識率が高いことが証明されている[3]。

本稿では、IEEE754に準拠した認識回路に必要な演算器を設計し、リアルタイム処理に必要な、つまりFTTSSの出力10ms以内の時間で認識部の処理が完了するかを、回路シミュレーションにより考察したのでここに報告する。

2. 音声認識システム

音声認識とは人間が発声した音声は波形に含まれる情報の中で周波数成分やその時間的変化など最も基本的な意味内容に関する情報を、自動的に抽出、解析し言語情報として認識することである。情報を抽出する部分が音声特徴抽出部、解析する部分が音声認識部と呼ばれている。以下で音声特徴抽出部と、音声認識部について説明する。

2.1. 音声特徴抽出

音声認識では認識させたい音声信号の時間的な変化をそのまま利用するのではなく、音声信号に含まれている周波数成分を利用するのが一般的である。理由として、もとのままの波形では情報が冗長すぎて扱いにくいということが挙げられる。また、もとの波形は外部環境の影響を受けやすいという欠点があり、周波数成分を利用することで耐性を持たすことができる。この周波数成分は特徴パラメータと呼ばれている。一般に特徴パラメータはスペクトル包絡のパラメータを指す。音声の言語音の音韻性は主に、音声のスペクトル包絡とその時間的推移によって特徴付けられるので特徴パラメータは音声を特徴付けるのに適している。

音声特徴パラメータの抽出の手法として、フィルタバンク分析と、LPC(Linear Predictive Coding)などが挙げられる。我々は耐雑音性をもっている FTTSS パラメータを用いている。FTTSS については第 3 章で説明する。

2.2. 音声認識

かつての音声認識には DTW(Dynamic Time Warping)が用いられていた。しかし、現在の音声認識では HMM(Hidden Marcov Model)[4]を用いた認識法が広く用いられている。DTW では単語や、音素に関して、標準的な時系列を標準パターンとして用いるが、HMM 法では各単語や音素を標準的な時系列マルコフモデルで表現する。HMM 法は DTW に比べてスペクトル時系列の統計的変動をモデルのパラメータに反映させることができる特徴があるが、逆にモデルパラメータを決定するための学習処理が複雑になる。実際は、対象に応じて適切に状態数やモデル構造を決定するとともに、スペクトルパターンの表現法を決定する必要がある。これらの数や複雑さ、すなわち、モデルの自由度を大きくすれば、きめ細かい変動が表現できるが、モデルの推定すべきパラメータが多くなり、精度が悪くなる。

各単語に対して HMM を構築したら各単語の観測ベクトルに対し、尤度を最大化するモデルを推定する。尤度を最大化するアルゴリズムとしてトレリスアルゴリズムとビタビアルゴリズムがあるが、計算値のダイナミックレンジが少ないなどの利点があるビタビアルゴリズム[5]を認識アルゴリズムとして用いる。

3. FTTSS

音声認識に用いられる代表的な特徴量として、メル LPC ケプストラムや FFT ケプストラムが挙げられるが、雑音のある環境下でこれらの特徴量を利用して音声認識すると、認識率が著しく低下する。そこで耐雑音性を持っている PSD フィルタを用いた特徴量 FTTSS を用いる。この方法は、音声認識のための特徴を周波数帯域の複数個の周波数点上でパワースペクトルの傾きを表現する量から抽出しようということに基づいている。図 1 に FTTSS 抽出のブロック図を示す。

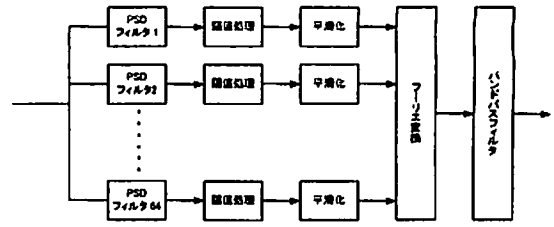

図 1　FTTSS 抽出のブロック図

抽出過程を説明する。まず入力された音声は PSD フィルタに入力される。64 チャネル PSD フィルタはそれぞれのチャネルの中心周波数が mel スケール上で等間隔となるように配置されていて、入力音声のそれぞれの中心周波数におけるスペクトルの傾斜値を求める。図 2 に各チャネルの中心周波数を示す。

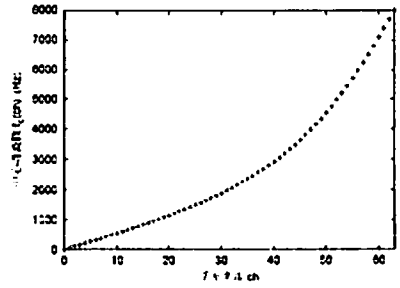

図 2　各チャネルの中心周波数

次に、求めた傾斜値にある閾値を設け、その絶対値が閾値よりも小さいなら 0、閾値よりも大きいもので符号が正なら＋1、負なら－1 とする。これにより、その帯域のパワースペクトルが雑音によって多少の影響を受けても、入力音声ホルマントに由来する正負の明確な傾斜を求めることにより、音声固有のスペクトルに基づく特徴は雑音の影響を受けにくい。閾値処理された後、64 チャネルの値を 10ms 毎に取り出し、フーリエ変換を行う。得られた特徴値は、ケプストラム計算に類似した処理で求められるが、傾斜値を求めてい

ること、閾値による二値化処理を行っていること、対数をとっていないことから異なる。そこでこの特徴値を FTTSS と呼んでいる。我々はこの FTTSS を用いて我々の設定した雑音下で、特徴量として一般的に用いられている MFCC を用いた時と比べて 8%の認識率向上を実験の結果により確認している。

4. 用意したモデル

HMM には離散分布型 HMM と連続分布型 HMM がある。離散分布型 HMM は、任意の分布形状を表現できる利点があるが、量子化に伴う歪が生じる難点がある。このため連続分布型 HMM を用いることにする。連続型 HMM は多次元正規分布を出力確率の密度関数としている。このときの出力確率は(1)式で表される。

$$f_x(x)=\frac{1}{(2\pi)^{P/2}|\Sigma|^{1/2}}\exp\left[-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right] \quad (1)$$

ここで、μは平均値ベクトル、Σは共分散行列である。今 P=30 次元としている。よってμも 30 次元、Σは 30×30 次元である。また、共分散行列は無相関正規分布を用いている。

モデルは left-to-right モデルを用い、10 状態 8 出力分布 1 混合の HMM である。遷移は自己遷移と次状態への遷移のみを許す。初期状態からの遷移と最終状態への遷移は 1 つに限定している。また初期状態と最終状態は分布を持たない。このようなモデルを 50 個音声認識に用いる。

5. 認識アルゴリズム

認識アルゴリズムとしてビタビアルゴリズムを用いた。ビタビアルゴリズムとは与えられた信号系列を最も高い確率で生成する状態遷移系列を求めるアルゴリズムである。時刻 t 状態 j において生成確率を最大にする経路を$\phi_t(j)$、最適状態の確率をδ_tとする。また、最適経路の生成確率を P、最適経路上の最終状態s_{t_r}とすると、最適経路およびその生成確率は以下の手順で求めることができる。ϕ

$$\begin{aligned}\delta_0(i) &= \pi_i \\ \phi_0(i) &= 0 \qquad (1\le i\le N)\end{aligned} \quad (2)$$

$$\begin{aligned}\delta_t(j) &= \max_{1\le i\le N}[\,\delta_{t-1}(i)a_{ij}\,b_j(o_t)] \\ \phi_t(j) &= \operatorname{argmax}_{1\le i\le N}[\,\delta_{t-1}(i)a_{ij}\,b_j(o_t)]\end{aligned} \qquad (1\le t\le T,\ 1\le j\le N) \quad (3)$$

$$P=\max_{1\le i\le N}[\,\delta_T(i)] \quad (4)$$

ここで、a_{ij} は状態 i から状態 j への遷移確率、$b_j(o_t)$は状態 j で観測系列 o_t を出力する確率、π_i は初期状態確率である。(3)式の反復をすることで尤度の最も高いモデルを選ぶことができる。そして、そのモデルが認識結果となる。また音声認識の概念図を図 3 に示す。

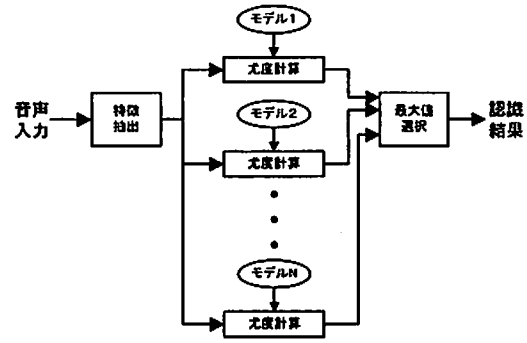

図 3　音声認識の概念図

6. 回路設計

音声認識処理に必要な平方根演算回路、対数演算回路、指数演算回路の設計を Verilog-HDL を用いて行った。また、すべての演算回路は IEEE754 のフォーマットに準拠させている。これはハードウェアに組み込むことを考えると便利だからである。

6.1. 平方根演算回路

6.1.1. ニュートン法

平方根の演算をするために、ニュートン法を用いた平方根演算回路を設計した。ニュートン法とは方程式を数値計算によって解くためのアルゴリズムである。平方根を求めたい値を a とし、$f(x)$を(5)のようにとり、$f(x)$の 1 次微分した関数 $g(x)$は(6)式となる。

$$f(x) = x^2 - a \quad (5)$$

$$g(x) = 2x \quad (6)$$

点 $x=x_0$における $f(x)$の接線と x 軸との交点を x_1 とすると

$$x_1 = x_0 - f(x)/f'(x_0) \quad (6)$$

と表すことができ、この式は漸化式で表すことができることが知られており、次のような形で表すことができる。

$$x_n = x_{n-1} - f(x_{n-1})/g(x_{n-1}) \quad (7)$$

この x_nを漸化的に求めていけば平方根の値を求めることができる。これをフローチャートにすると図 4

になる。aが求めたい値である。

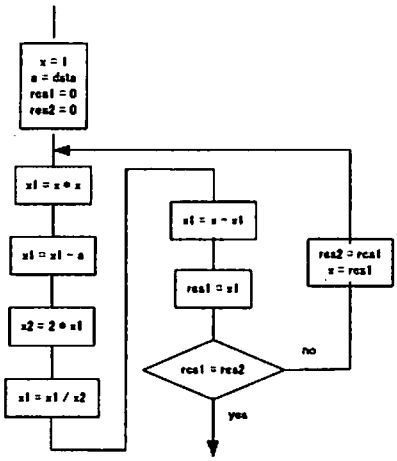

図4　ニュートン法を用いた平方根演算の演算フロー

6.1.2. 初期値の選定

また、初期値の取り方について考察する。初期値はどのようにとっても収束することが証明されている。しかし、選んだ初期値によっては値が収束するまでに時間がかかってしまう場合がある。平方根回路に入力されるモデルパラメータの取りうる値の範囲は最大値は 3.096892×10^27 である。また、最小値は 2.313191×10^16 である。このときの値が収束するまでのニュートン法で用いる初期値と収束反復回数の関係を図 5 に示す。

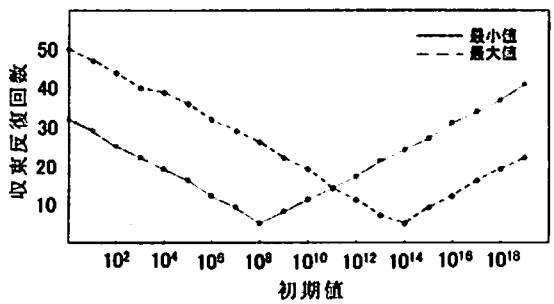

図5　入力値の初期値と収束反復回数の関係

図 5 より、最大値の時は 10^14 の時に、また最小値の時は 10^8 のときに最も反復回数が少なくなる。よって、初期値は中間の 10^11 とした。

6.2. 対数演算回路

対数演算を行う回路のフローチャートを図 6 に示す。

底は 2 で考える。まず、最初に正規化を行う。ここで言う正規化とは浮動小数点の指数部を 8'b01111111 とすることである。つまり、求めたい数を a とすると

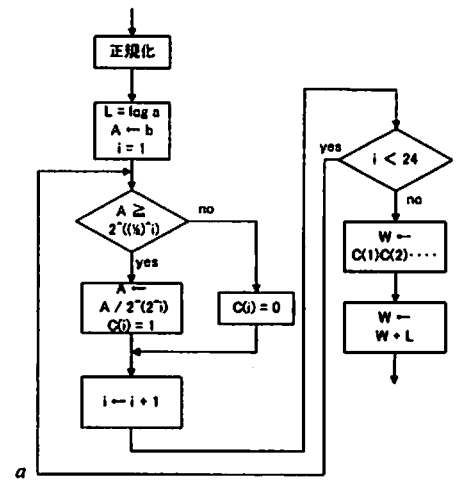

図6　対数演算の演算フロー

を $1.0 \leqq A < 10$(二進数表記)と 2 のべき乗の形に分解する。a の対数をとったものは対数の加算の性質より、$1.0 \leqq A < 10$ の対数と、2 のべき乗の対数との加算をすればよいということになる。底を 2 でとった理由は、2 のべき乗の対数は底が 2 の場合、整数になるのでこのような正規化を行うことにより回路の複雑性を減らした。またこれは、a が 2 のべき乗だった場合に演算量が小さくなる。A の対数の計算は 2^((1/2)^i)との大小比較によって行う。まずは i=1 のとき、つまり 2^(1/2)との比較になる。もし $A \geqq$ 2^(1/2)が成立するならば、$C(1)$=1 とする。そして、A/2^(1/2)を計算して A を更新する。対数の減算の性質に基づき、除算 A/2^(1/2)を利用している。一方、$A \leqq$ 2^(1/2)ならば $C(1)$=0 として A の値は何もせずにそのままにしておく。この操作を i が仮数部のビット数である 23 になるまで続ける。できあがった $C(i)$を最上位ビットから $C(1)C(2)\cdots C(23)$ とすることで A の対数を得ることができる。最後に出来上がった A の対数と 2 のべき乗の対数を加算することで a の対数が求められる。このようにして、簡単なアルゴリズムにより対数演算を行うことができる。これにより回路設計の複雑さを減らすことができる。

6.3. 指数演算回路

6.3.1. 指数演算回路の動作説明

浮動小数点は 2 のべき乗の各桁の加算で表されている。また、指数の計算において指数部の加算は指数同士の乗算で表すことができる。つまり、浮動小数点の指数は 2 のべき乗の指数の乗算で表すことができるということがわかる。具体的な処理としては、あらかじめ 2 のべき乗の指数の値を計算しレジスタに格納しておく。求めたい数の指数部により、その数の最大の桁を調べる。この最大の桁と仮数部を用いて指数の計

算を行う。仮数部が1の桁はそのべき乗の桁に対応した指数の値をレジスタから読み込む。仮数部が0の場合はその指数は1となる。最後にレジスタより読み出したすべての桁の指数の乗算をすることで全体の指数の値を求めることができる。

6.3.2. 浮動小数点の表現可能範囲

浮動小数点で表せる数には限りがある。つまり、IEEE754 の浮動小数点では-3.40282×10^38～3.40282×10^38 の間の数しか表すことができない。これを2のべき乗の指数に当てはめると、あらかじめ計算してレジスタに格納しておかなければならないのは、exp(2^-24)～exp(2^6)と exp(-2^-25)～exp(-2^6)ということになる。例えば、exp(2^6)=6.23514×10^27 となり、これが浮動小数点で表せる最大の数となる。この数を超えた桁を持つ浮動小数点の指数の値は無限とした。また exp(2^-24)を計算すると 1.00000006 となり、べき乗の最大の桁数が1のときの浮動小数点が表すことができる最小の数である。

7. 演算回路の論理合成結果

6章で述べた各演算回路について Synopsys 社の Design Compiler を用いて論理合成を行った。表1に最大動作周波数、演算に必要なクロック数、ゲート数の結果を示す。テクノロジは CMOS0.35μm、ゲート数は 2NAND 換算で求めた。

表1　各演算回路の論理合成結果

	最大動作周波数 (MHz)	演算クロック数 (クロック)	ゲート数 (ゲート)
平方根演算回路	40	69	23608
対数演算回路	40	101	20989
指数演算回路	40	12	50056

8. 認識部回路の回路シミュレーション

8.1. 音声認識回路

ビタビアルゴリズムを用いた音声認識回路の設計を行った。認識候補のモデル数は 50 とした。以下の段落でその回路と結果について述べる。

8.2. 音声認識回路

図7に認識部回路のブロック図を示す。まず、入出力関係から説明する。fxin は特徴抽出部からの 30 次元の出力である。fin は特徴抽出部の終了信号である。また、start は認識の開始信号である。特徴抽出部の出力のフレーム間隔は 10ms 毎となっている。このため、10ms 以内に認識処理を完了する必要がある。音声認識

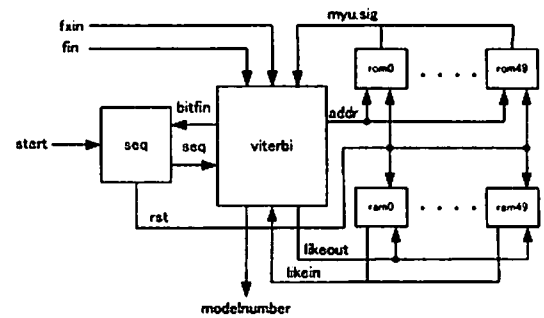

図7　認識部回路のブロック図

のシミュレーションのためにレジスタ rom0～rom49 には 50 モデル分のモデルパラメータが格納されている。rom0 にはモデル番号0のパラメータが格納されており、rom1 にはモデル番号1、rom2 にはモデル番号2、……のように割り当てあられている。モデルパラメータとは平均値ベクトル、共分散ベクトル、遷移確率である。また、計算した尤度の値はモデルごとに保存しておく必要がある。そこでレジスタ ram0～ram49 に途中結果を保存しておく。romN、ramN(N=0～49)は共通のリセット信号入力を持つ。viterbi はビタビアルゴリズムを用いた尤度計算を 50 モデル分行う回路である。またモデル1つ分の尤度計算が終わるたびに尤度計算終了信号 bitfin 信号を制御シーケンサ seq に送る。seq ではその bitfin 信号をカウントして rom、ram 部に rst 信号を送っている。seq で bitfin 信号をカウントし rom、ram を順番にセット状態にする。この操作により候補モデルを順に選んでいく。セット状態になった rom から viterbi へ格納されている各パラメータが送られる。このとき、どのモデルパラメータのどの状態からどの状態への遷移のパラメータを読むかを決めているのが addr 信号である。計算された途中の尤度は ramN に書き込まれる。50 モデル数分計算が終わって、seq が start 信号を検出し、seq 信号を1にするまで viterbi は次のフレームの特徴ベクトルが入力されるまで待つことになる。seq が start 信号を検出して、そのたびに seq 信号を 10ms 毎に viterbi ブロックに送る。これにより、次のフレームの処理へと移行する。この seq 信号により、viterbi ブロックは次のモデルの尤度計算を行う。これを全モデル分、全フレーム分計算する。そして fin 信号が1になったら ram0～ram49 に格納されている最終尤度をそれぞれのモデル分比較を行う。この中で最も大きいもののモデル番号出力とする。

認識部のフローチャートを図 8 に示す。まず、bitfin=1 としてモデル 1 から尤度を計算していく。addr=0 は 0→0 の遷移、addr=1 は 0→1 の遷移のパラメータを rom から読み込ませる信号である。これを、

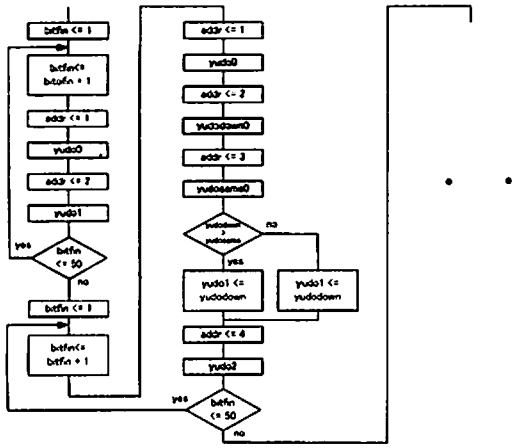

図 8　音声認識部のブロック図

bitfin=50 つまり、50 モデル分繰り返す。それが終わると次のフレームの処理へと移る。ここで同じ状態へ、やってくる遷移が 2 通りあるので、尤度の最大値選択という処理を行わなければならない。ここで yudodown は自分の状態よりひとつ前の状態から来る遷移を表しており、yudosame は自分と同じ状態から来る遷移である。これらのうちの大きい方をその状態の尤度とする。これらの処理をフレーム毎に計算する。遷移パスは一番多い時で 14 パスあり、このときの尤度計算が最も処理時間が長い場合となる。

8.3. シミュレーション結果

認識部回路の演算時間を見積もるために、cadence 社の NC-Verilog を用いて図 7 の回路のシミュレーションを行った。一番多いときの遷移の場合について、演算時間の見積もりを行ったところ 6.1ｍｓを見積もった。FTTSS 特徴抽出部からの出力フレーム間隔である 10ms よりも短い時間で認識処理が完了することが確認された。

9. まとめ

ビタビアルゴリズムを用いた音声認識回路のための平方根、対数、指数演算回路の設計を行った。平方根演算回路ではニュートン法を用いて減算器、乗算器、除算器のみで実現した。対数、指数演算回路は浮動小数点の性質を生かし回路設計の複雑さを減らした。また、それらの回路を用いて音声認識部の設計を行った。ここではその制御回路と、モデルパラメータを格納しておく rom、さらに各モデルの尤度結果を一時的に格納しておく ram の設計を行った。音声認識回路シミュレーションにおいて、FTTSS の出力 10ms 毎に対して、リアルタイム処理が十分可能である処理時間 6.1ms を見積もることができた。

10. 謝辞

本研究は東京大学大規模集積システム設計教育研究センター(VDEC)を通し、ケイデンス、シノプシス株式会社の協力で行われたものである。

文　献

[1] M. Fujioka, S. Yamamoto, M. Nkamura, T. Mukasa, and N. Inoue, "EXPERIENCE AND EVOLUTION OF VOICE RECOGNITION APPLICATIONS FOR TELECOMMUNICATIONS SERVICES" Global Telecommunications Conference, 1998. GLOBECOM98. The Bridge to Global Integration. IEEE, pp.1338-1343, Sydney, Australia, Nov.1998.

[2] J.-I.Dugelay, et al., "RECENT ADVANCES IN BIOMETRIC PERSON AUTHENTICATION", Acoustics, Speech, and Signal Processing, 2002. Proceedings. (ICASSP '02). IEEE International Conference on, pp.4060-4063, 2002.

[3] 船田哲男, 統木貴史, "スペクトル傾斜に着目した音声認識のための特徴抽出, " 電子情報通信学会論文誌, Vol. J82-d-Ⅱ, No.11, pp.2184-2187, Nov, 1999.

[4] Lawrence. R. Rabiner, A Tutorial on Hidden Marcov Models and Selected Applications in Speech Recognition, Proc. IEEE, vol.77, no.2, pp.257-286, February 1989.

[5] G. David. Forney, JR, The Viterbi Algorithm, Proc. IEEE, vol.61,, pp.268-278, March 1973.