リアルタイム大語彙連続音声認識システムのハードウエア化に関する研究

報告者　大学院総合理工学研究科　エレクトロニクス系工学専攻　教授　神戸尚志
共同研究者　理工学部電気電子工学科　助手　谷本浩一

1．背景

音声認識技術は、キーボードやマウスより便利で自然な形での操作が可能な入力インターフェイスとして期待されている。現在用いられる音声認識は、主に決められた単語に対して認識可能であり、文章による入力には対応していない。文章による音声認識を大語彙連続音声認識、または連続音声認識と呼ばれ、現在、パソコン上で動作するソフトウエアとして実現されている。大語彙連続音声認識では、音響モデル(HMM)や言語モデル(N-gram)を多く用いられている。音声認識の根幹技術の一つである HMM を用いた認識は、認識性能が高い反面、計算量が多くなる。そのため、ソフトウエアでリアルタイム音声認識を行うには高い処理能力を持つ CPU が必要となっている。

2．目的

本研究では、大語彙連続音声認識エンジン「Julius」をもとに、ハードウエアとソフトウエアで構成するシステムを設計し、携帯端末へ搭載し、キーボード操作など専門技術を必要としない人間にやさしい電子機器利用環境の実現を目指している。

3．研究組織

神戸…研究開発の企画立案、学生指導、論文作成指導、成果のまとめ
谷本…FPGA ボードによるシステム試作、学生指導、論文作成
大学院生、学部生…ハードウエア・ソフトウエアの設計とその最適化、
FPGA ボードへの実装、システムテストなど

4．研究方法

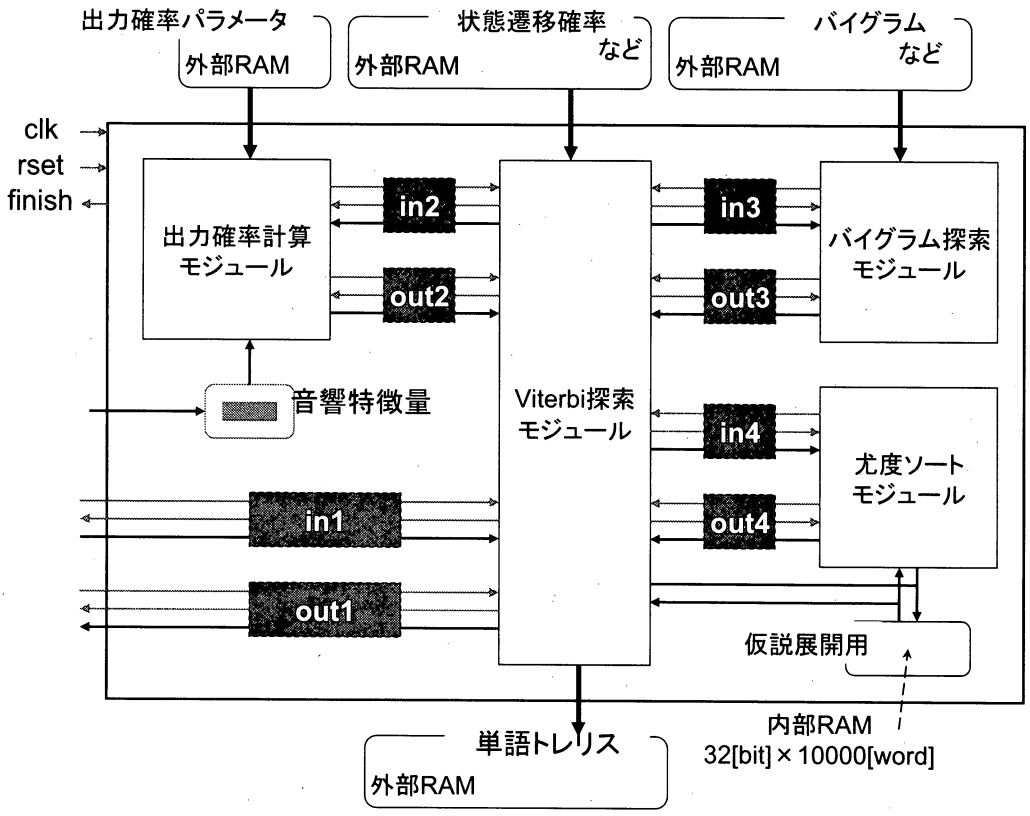
大語彙連続音声認識ソフトウエア「Julius」の第 1 パス部をハードウエア化し、その高速化・回路規模削減を行い、入出力部と第 2 パスを実行する CPU とのインターフェイスを構築し、FPGA ボードによりシステム全体の動作を検証・評価する。

5．研究成果

下図に、今回設計した第一パス部の回路構造を示す。主な各関数を 4 つのモジュールに分

割(Vitabi 探索モジュール、出力確率計算モジュール、バイグラム探索モジュール、尤度ソートモジュール)した。本回路の動作は、以下の通りである。１フレーム分の音声特徴量をレジスタに格納し同期信号（in1）で第一パス処理を開始する。Vitabi 探索モジュールは、同期信号（in2, in3）を用い、出力確率計算モジュールによる出力確率計算、バイグラム探索モジュールによる N-gram 探索を開始させる。同期信号（out2, out3）より得られた出力確率と状態遷移確率から確率の高さを示すスコアを算出し、最終的にスコアの高いものを候補として残すため尤度ソートモジュールにより尤度ソートが行われる。

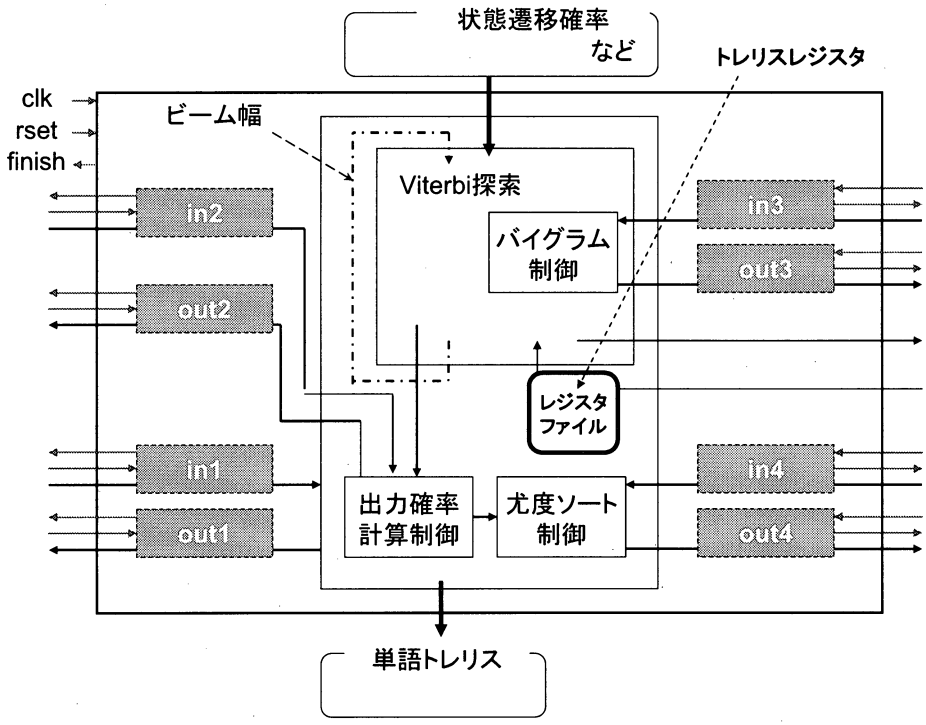
以下に、各モジュールの設計最適化について述べる。



Vitabi 探索モジュールの最適化

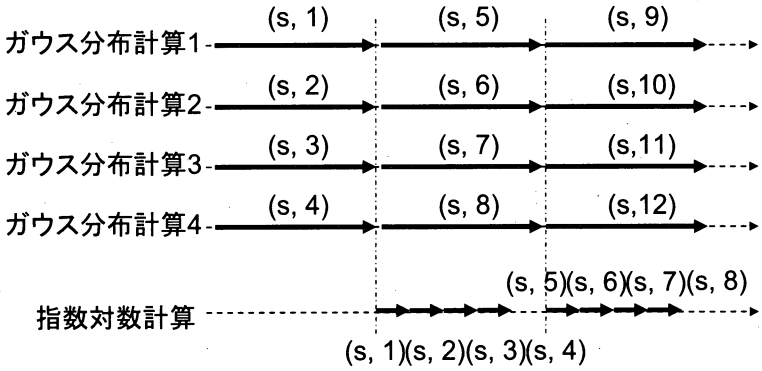
Vitabi 探索は、音素の状態遷移すべてを網羅するのではなく、同じ状態に遷移するパスのうち、スコアの低いパスは除外し、よりスコアの高い後続可能なパスを残して確率計算を行う Vitabi アルゴリズムが用いている。

メモリへのアクセス回数を削減するために、一時的に遷移に必要な情報を記憶するトレリスレジスタを実装した。トレリスレジスタの実装により、アクセス時間を 1/2 に削減した。トレリスレジスタを実装した Vitabi 探索モジュールを下図に示す。



出力確率計算モジュールの最適化

出力確率計算モジュールは音声特徴量を入力として、出力確率計算結果を出力するモジュールである。出力確率計算では、ガウス分布計算を計算した後、その結果を受け指数対数計算が実行される。ガウス分布計算は各混合で依存関係がないために、並列化が可能である。ガウス分布計算と指数対数計算の処理時間は約 4:1 である。下図で示す様に、ガウス分布計算を混合数に対して 4 並列で計算を行い、指数対数計算とパイプライン化した。これにより 4 倍以上の高速化となった。

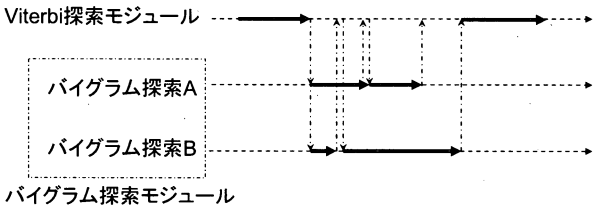


音声認識では、音素を HMM でモデル化している。本研究では状態数（1987 状態）の HMM を使用している。また、音素の種類として母音のように頻繁に出現するものや、子音のようにあまり出現しないものに分類される。そのため、出力確率の計算回数に違いが生じる。一度算出された確率に対して内部 RAM に格納し、再度計算されることを防いだ。出

力確率計算結果 32[bit]、HMM 状態数 1987[状態]の記憶が必要なため、内部 RAM の容量は約 16[KB]となった。

バイグラム探索モジュールの最適化

バイグラム探索に必要な言語確率データは外部 RAM に格納されている。マルチポートメモリを適用し、並列にアクセスする。各探索が終了すると、すぐに次の探索要求が送られるよう、探索との通信を非同期通信とし、探索要求信号の送信タイミングを管理する監視用フラグを搭載した。



尤度ソートモジュールの最適化

ソフトウエアの尤度ソートは、ヒープソートが適用されていた。しかし、有力候補を抽出さえすればよく、すべてを昇順に並べる必要はない。候補の中から尤度のビーム幅分の候補を探索しながら、有力候補と判断できた候補を、探索範囲から省きながら候補分を抽出できるハーフカッティングを考案適用した。スコアの最大値と最小値から算出される中央値に基づいて、ビーム幅分の候補を求めるアルゴリズムである。

設計結果

第 1 パスの各モジュールに最適化を行った結果について下表に示す。高精度版（ビーム幅、1500 個）とし動作周波数を 100MHz とした。

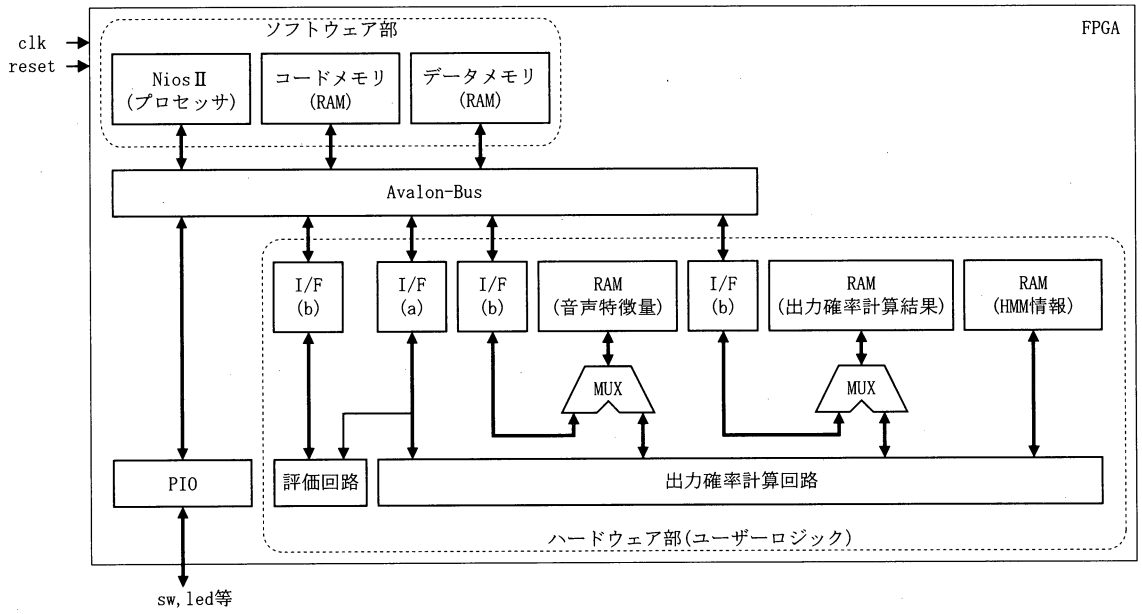
	回路構成	回路規模[ゲート数]
①	シーケンシャル	195, 512
②	① + 関数単位のパイプライン +式変形+ループ展開	463, 420
③	② + 出力確率計算キャッシュ実装	477, 603
④	③ + バイグラム並列探索	496, 817
⑤	④ + トレリスレジスタ	500, 903
⑥	⑤ + ハーフカッティング	499, 210

回路規模は、②から⑥の最適化を行うことで、①に比べ、約 7%増加したが、処理時間は、約 776.7 倍の高速化ができた。ガウス分布/指数対数計算の最適化により回路規模が 267,908[ゲート]増加している。これは第 1 パスの処理でボトルネックとなっているガウス分布計算を高速化するため、ガウス分布計算を 4 並列する関数単位のパイプラインおよび並列化による規模の増大が原因である。それ以降の最適化回路については、ほとんど回路規模の増加はなく最適化を実現した。

最適化⑥の回路においては、測定を行った 20 フレームすべてで音声認識のリアルタイムな動作時間である 10[ms]以下の処理時間を達成できた。

出力確率計算部の FPGA ボードへの実装

FPGA 実装の第 1 段階として、ソフトウエアにより出力確率計算回路の制御を行う試作システムを設計し、購入した FPGA ボードを用いてハードウエア／ソフトウエア協調検証を行った。試作システムは、HMM の状態数 8 とし、HMM 情報をオンチップメモリに実装し、各種高速化手法(並列化、RAM 分割、パイプライン処理)を適用した出力確率計算回路の性能を評価した。試作システムのアーキテクチャを下図に示す。



出力確率計算について Nios II プロセッサによるソフトウエア処理と、出力確率計算回路によるハードウエア処理のサイクル数を比較測定した。4 混合 8 状態の HMM 情報で 1 フレーム分の出力確率計算を行った結果を下表に示す。

ハードウエア処理では、回路規模はソフトウエア処理と比べて約 20,000[LE]の増加となるが、計算を 2,890 倍高速化できた。また、出力確率計算回路の FPGA による検証は、ソフトウエアシミュレーションよりも約 3,000～15,000 高速であり、検証期間の短縮に有効であることを確認した。

出力確率計算	回路規模 [LE]	処理サイクル数	高速化比
ソフトウェア処理	3,184	20,096,721	1.00
ソフトウェア処理（整数演算化）	3,042	1,831,250	10.97
シーケンシャル（ハードウェア化）	10,060	27,176	369.75
2 並列	15,695	14,938	672.67
2 並列＋RAM 分割	19,127	14,466	694.62
4 並列	27,415	7,585	1,324.77
4 並列＋RAM 分割	27,988	7,274	1,381.41
パイプライン処理	23,862	7,397	2,716.87
パイプライン処理＋ RAM 分割	23,336	6,948	2,892.45

6．今後の展開

本研究では、音声認識システムの計算時間を多く要している部分（第一パス処理）をハードウエア化することにより、システム全体の効率化を提案した。第一パス処理の中では Vitabi 探索モジュール・出力確率計算モジュール・バイグラム探索モジュール・尤度ソートモジュールの最適化をすることで第一パス処理の高速化を行い、リアルタイムな処理を実現した。また、FPGA ボードを用いて出力確率計算部の動作検証評価を行った。

今後は、第二パスを含めたソフト・ハードシステムの開発と FPGA ボードによる協調検証を進める。