

複数の誤り訂正符号に対応する 再構成可能デコーダモデルの提案

谷口 一徹[†] 小林 礼貴^{††} 坂主 圭史^{††}
武内 良典^{††} 今井 正治^{††}

[†]立命館大学 理工学部 電子情報デザイン学科

^{††}大阪大学 大学院情報科学研究科 情報システム工学専攻

本研究では、複数の誤り訂正符号に対応する再構成可能デコーダモデルを提案する。提案する再構成可能デコーダモデルは、さまざまなビタビ復号とターボ復号に対応し、デザインパラメータを設定することでさまざまな再構成可能デコーダを実現できる。評価実験より、提案する再構成可能デコーダは、実用的な性能及び面積や消費電力で実現できることが確認できた。

Parameterized Reconfigurable Decoder Model for Multiple FEC Standards

Ittetsu Taniguchi[†] Ayataka Kobayashi^{††} Keishi Sakanushi^{††}
Yoshinori Takeuchi^{††} and Masaharu Imai^{††}

[†]College of Science and Engineering, Ritsumeikan University, Japan

^{††}Graduate School of Information Science and Technology,
Osaka University, Japan

Abstract : This paper proposes a parameterized reconfigurable decoder model for multiple FEC standards. Proposed decoder model enables to realize a dedicated reconfigurable decoder which executes pre-defined multiple FEC standards easily. Experimental results shows that proposed reconfigurable decoder model is quite reasonable as FEC decoder.

1. 研究背景

近年、携帯電話、PDA などのさまざまな小型の組み込みシステムが広く普及している。これらの組み込みシステムは、年々多機能化が進んでおり、音楽の再生、携帯ゲームなどの様々なアプリケーションが充実している。このような組み込みシステムにおいて、無線通信機能は必要不可欠になっている。現在までに、無線通信において様々な符号化方式や変調方式が提案されている。特に、演算の並列度が高く、多量の処理を必要とする処理として誤り訂正復号処理が知られている。誤り訂正復号処理は、通信路上のノイズにより誤ったデータを復号する処理である。近年の無線通信規格では、ビタビ復号やターボ復号などの誤り訂正復号処理が広く使用されている。

そこで本研究では、複数の誤り訂正符号に対応する再構成可能デコーダモデルを提案する。提案する再構成可能デコーダモデルは、ビタビ復号やターボ復号を実現できる再構成可能アーキテクチャで、デザインパラメータを適切に設定することで、決められた複数の誤り訂正符号に対応するデコーダを実現する。

以下、本稿の構成を述べる。第 2 章では、現在用いられる通信規格とそこで使用されている復号方式をまとめる。第 3 章では、本研究で対象とする復号方式であるビタビ復号とターボ復号についてまとめる。第 4 章では、提案する再構成可能デコーダモデルについて述べる。そして、第 5 章で評価実験、第 6 章でまとめと今後の課題について述べる。

2. さまざまな通信規格とその復号方式

表 1 に現在提案されている通信規格とそこで用いられている復号方式についてまとめる。さまざまな通信規格においてビタビ復号やターボ復号が用いられているが、それぞれの復号方式において、拘束長や対応する符号化率、最大ブロック長や最大スループットが異なる。本研究では、単純にビタビ復号とターボ復号といった区別だけでなく、拘束長や符号化率などをあらかじめ決め、その範囲での再構成可能なデコーダを実現する、再構成可能デコーダモデルを提案する。

3. 対象とする誤り訂正符号

3.1 ビタビ復号

ビタビ復号²⁾は、畳み込み符号化された系列から入力データを復号するアルゴリズムである。ビタビ復号は、最尤復号化の一種であり、受信した系列が出力される尤もらし

表 1. 誤り訂正復号処理のパラメタとスループット¹⁾

通信規格	復号化	拘束長	符号化率	最大ブロック長	最大スループット
GSM	Viterbi	5	1/2	876	12Kbps
GSM-EDGE	Viterbi	7	1/2	870	62Kbps
CDMA2000	Viterbi	9	1/2, ..., 1/6	744	28Kbps
UMTS	Viterbi	9	1/2, 1/3	504	32Kbps
DAB	Viterbi	7	1/4	***	1.1Mbps
DVB-T	Viterbi	7	1/2	1,624	32Mbps
DVB-H	Viterbi	7	1/2	1,624	32Mbps
IEEE802.11a/g	Viterbi	7	1/2	4,095	54Mbps
IEEE802.16	Viterbi	7	1/2	4,800	20Mbps
CDMA2000	Turbo	4	1/5	20,730	2Mbps
UMTS	Turbo	4	1/3	5,114	2Mbps
HSDPA	Turbo	4	1/3	5,114	14.4Mbps
DVB-RCT	Turbo	4	2/4	648	31Mbps
LTE	Turbo	4	1/3	5,114	100Mbps

い入力系列を求める方式である。ビタビ復号の処理は大きく分けて以下の3つの処理から構成される。

1. Branch Metric 演算

Branch Metric 演算とは、トレリス線図における全ての遷移に対して、復号したい符号語と各内部状態において出力される符号語との間のハミング距離を算出する処理である。このハミング距離を Branch Metric と呼ぶ。拘束長を K とすると、ある時刻 t における内部状態は 2^{K-1} 存在するため、これらの処理を並列実行可能である。

2. ACS 演算

ACS (Add Compare Select) 演算とは、ある時刻 $t+1$ の任意の状態に遷移する2つのパスに対し、尤度の大きいパスを選択し、Path Metric を算出する。Path Metric とは、そのパスを構成する全ての遷移の Branch Metric の総和である。ここで、尤度の大きいパスとは、Path Metric が小さいパスであると言える。そして、ACS 演算では、Path Metric がもっとも小さい生き残りパスを出力する。

3. Trace Back 演算

Trace Back 演算とは、ACS 演算により求められた生き残りパスから、最終的な復号結果

を出力する演算である。

3.2 ターボ復号

ターボ復号³⁾とはターボ符号化された系列を復号する処理である。ターボ符号化器は、複数の畳み込み符号化器とインタリーバから構成される。一般的なターボ符号は、情報ビット1ビットに対して3bitの符号語を出力することから、符号化率は1/3となる。本研究ではターボ復号を実現するアルゴリズムとして Berrou らのアルゴリズム³⁾を採用する。Berrou らのターボ復号アルゴリズムを実現する符号器、復号器として図1に示す構成が提案されている。

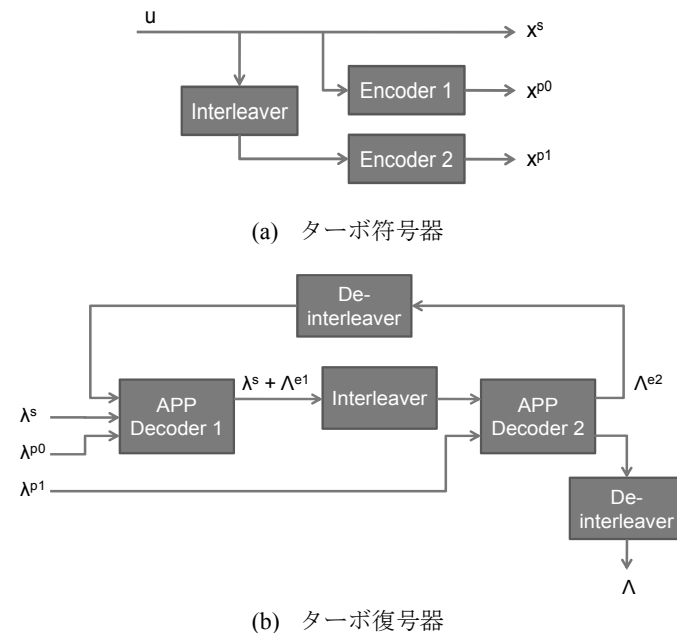


図 1. Berrou のターボ符号器とターボ復号器

図1に示すターボ符号器は2つの畳み込み符号器と1つのインタリーバから構成される。ターボ復号器は、2つのAPP (A Posteriori Probability) 復号器ならびにインタリーバとデインタリーバから構成される。APP 復号器はSISO (Soft Input Soft Output) 復号器から

構成される。SISO 復号とは、入力データ系列と事前外部値から対数尤度比と外部値を計算する復号法である。本研究では、SISO 復号のアルゴリズムとして、Max-Log-MAP アルゴリズム⁴⁾を採用する。Max-Log-MAP アルゴリズムでは、SISO 復号処理は以下に示す4つの演算から構成される。

1. γ 演算

γ 演算とは、トレリス線図における全ての遷移に対し、 γ 値を計算する演算である。 γ 値とは、状態が遷移する際の尤度を表す。拘束長をKとすると、ある時刻tにおける内部状態は 2^K-1 存在するため、これらの処理を並列実行可能である。

2. β 演算

β 演算とは、ある状態から遷移する2つのパスに対して、 γ 値と遷移先の後方遷移尤度からある時刻における後方遷移尤度を求める演算である。1つの状態に遷移する2つのパスのうち、尤度の大きいパスの尤度をその状態の後方遷移尤度とする。これらの処理は、各状態に対して計算されるので、並列化が可能である。

3. α 演算

α 演算とは、ある状態に遷移する2つのパスに対し、 γ 値と前方遷移尤度からある時刻におけるその状態の前方遷移尤度を求める演算である。1つの状態に遷移する2つのパスのうち、より尤度の大きいパスの尤度をその状態の前方遷移尤度とする。これらの処理は、各状態に対して計算されるので、並列化が可能である。

4. LLR 演算

LLR 演算は、前方遷移尤度、後方遷移尤度、 γ 値、入力データ、事前外部値から、対数尤度比 (LLR) と外部値を求める演算である。

4. 再構成可能デコーダモデル

4.1 全体構成

提案する再構成可能デコーダモデルを図2に示す。提案する再構成可能デコーダモデルは、7種類のメモリやバッファ、レジスタファイル、そして、BM_GユニットやPEアレ

イ、ならびにLLR_MinユニットやTrace Backユニットから構成される。Trace Backユニットとメモリ以外のユニットは、ビタビ復号時とターボ復号時で異なる処理を行う。

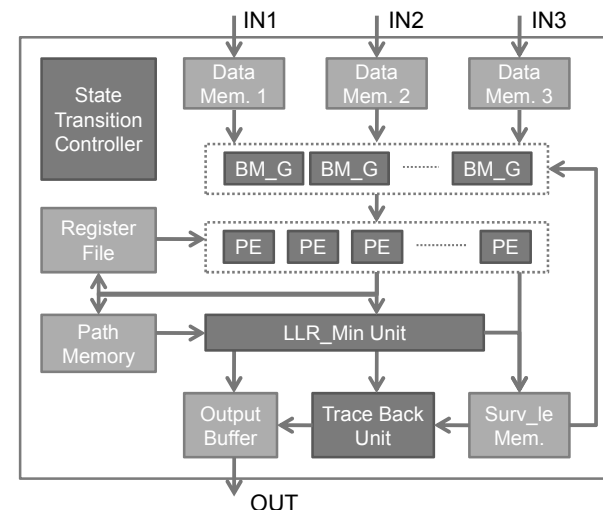


図2. 再構成可能デコーダモデル

図3にビタビ復号処理時のデータの流れを示す。入力されたデータはデータメモリで一旦保存され、BM_Gユニットに転送される。BM_GユニットはBranch Metric 演算を並列で行い、PEアレイはACS演算を並列で行う。LLR_MinユニットではTrace Back 演算に必要な最小のPath Metricを持つ状態を求める演算を行う。

図4にターボ復号処理時のデータの流れを示す。入力されたデータはデータメモリで一旦保存され、BM_Gユニットに転送される。BM_Gユニットでは γ 演算を行い、各遷移の γ 値を求める。そして、PEアレイで α 演算と β 演算を並列に行い、求められた前方遷移尤度ならびに後方遷移尤度はレジスタファイルならびにPathメモリに格納される。求められた前方遷移尤度ならびに後方遷移尤度を用い、LLRならびに外部値を出力する。

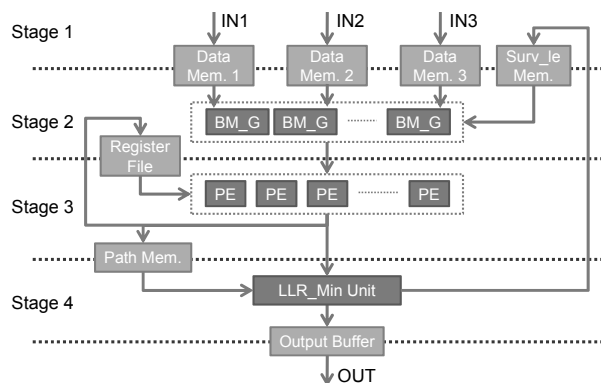


図 6. ターボ復号処理時のパイプライン構成

4.3 再構成可能デコーダモデルのデザインパラメータとコンフィギュレーション情報

再構成可能デコーダモデルはさまざまなデザインパラメータを持ち、対応したい復号方式に合った再構成可能デコーダを実現できる。一方、特定の復号化方式に特化した再構成可能デコーダにおいて、コンフィギュレーション情報を変更することで、あらかじめ決められた復号化方式に対応できる。これは再構成可能デコーダの大きな特徴であり、本節では、再構成可能デコーダモデルのデザインパラメータならびにコンフィギュレーション情報について述べる。

提案する再構成可能デコーダモデルのデザインパラメータならびに再構成可能デコーダのコンフィギュレーション情報を表 2 に示す。再構成可能デコーダモデルのデザインパラメータは、デコーダの仕様で、対応可能な復号化の情報を与える。ビタビ復号の拘束長や符号化率などは 1 つの数値を与えるのではなく、対応可能な拘束長の集合を与える。それに従い、指定された拘束長の集合全てに対応できるようなインスタンスが生成される。一方、PE 数もデザインパラメータとして与える。PE 数を変更することで内部での並列度に影響を及ぼす一方、面積や消費電力にも影響を与える。PE 数は要求されるスループットに応じてあらかじめ最適化することを想定する。

再構成可能デコーダを実際に使用する際は、コンフィギュレーション情報を与え、その動作を指定する。具体的には、符号化時の符号生成式や復号方式（ビタビまたはターボ）、ならびに拘束長やブロック長など、再構成可能デコーダが対応可能な範囲の情報を

指定する。ここで、ターボ復号時のループ回数も指定する。

表 2. 再構成可能デコーダモデルのデザインパラメータとコンフィギュレーション情報

デザインパラメータ (デコーダ仕様)	コンフィギュレーション情報
<ul style="list-style-type: none"> PE 数 入力データのビット幅 Sliding Window アルゴリズムの Window サイズ 対象復号方式 ビタビ復号の拘束長 ビタビ復号の符号化率 ビタビ復号の判定方式 ターボ復号の拘束長 最大ブロック長 	<ul style="list-style-type: none"> 符号生成式 ターボ復号時のループ回数 実行復号方式 (ビタビ/ターボ) 拘束長 (ビタビ復号時) 符号化率 (ビタビ復号時) 判定方法 (ビタビ復号時) 拘束長 (ターボ復号時) ブロック長

5. 評価実験

提案する再構成可能デコーダモデルの有効性を示すために、本章ではスループット、面積、消費電力の 3 点からさまざまな仕様の再構成可能デコーダインスタンスを評価する。生成された再構成可能デコーダは TSMC 0.18um で合成し、100MHz での動作を想定しスループットを算出した。また、ロジック部の消費電力はゲートレベルシミュレーションで、メモリ部の消費電力は TSMC 0.18um のデータシート値より算出した。

5.1 再構成可能デコーダインスタンスの評価

提案する再構成可能デコーダの有効性を示すために、以下の 3 つの仕様の再構成可能デコーダを実現し評価した。評価結果を表 3 に示す。

- Type A (ビタビ復号のみ)
 - ビット幅: 8bit, PE 数: 8, Window サイズ: 128, 最大ブロックサイズ: 5114
 - 拘束長: 9, 符号化率: 1/2, 1/3, 判定方式: 軟判定
- Type B (ターボ復号のみ)
 - ビット幅: 8bit, PE 数: 8, Window サイズ: 128, 最大ブロックサイズ: 5114
 - 拘束長: 4, 並列度: 8
- Type C (ビタビ復号&ターボ復号)
 - ビタビ復号: Type A と同じ仕様
 - ターボ復号: Type B と同じ仕様

表 3. 再構成可能デコーダの評価結果

Type	スループット [Mbps]	面積		消費電力 [mW]
		ロジック部 [Kgate]	メモリ部 [Kbit]	
A	3.0	57	73	92
B	4.1	13	242	241
C (ビタビ)	3.0	64	267	100
C (ターボ)	4.1			285

表 3 より、再構成可能デコーダでビタビ復号とターボ復号を実現する Type C の場合、ビタビ復号専用の再構成可能デコーダで実行時 (Type A) ならびにターボ復号専用の再構成可能デコーダで実行時 (Type B) それぞれにおいて、個別に実現する場合と同様の性能が得られることが分かる。また、4.1 節で述べたように、ビタビ復号時とターボ復号時で大半の演算ユニットやメモリを再利用しているため、単独に 2 つのデコーダを搭載する場合に比べ、ロジック部、メモリ部共に面積を節約できることが分かる。しかし、消費電力に関しては、ビタビ復号実行時、ターボ復号実行時それぞれにおいて、専用の再構成可能デコーダ (Type A, B) で実行する際よりも多くかかることが分かる。例えば、ビタビ復号実行時は、Path メモリは使用しないため、そのための消費電力が必要となる。また、ターボ復号実行時は、Trace Back ユニットが不要である。

5.2 再構成可能デコーダモデルのスケラビリティの評価

再構成可能デコーダモデルはさまざまなデザインパラメータを持ち、必要な仕様のデコーダを生成できる。本節では、性能に直接的に影響を及ぼす PE 数を変化させた際のスループット、面積、消費電力の変化を評価する。本節では、以下の仕様のビタビ専用の再構成可能デコーダを実現することを想定し、PE 数を 8, 16, 32, 64 と変化させてスループットや面積、消費電力を評価した。

- 共通仕様
 - ビット幅: 8bit, Window サイズ: 64, 最大ブロックサイズ: 4800
 - 拘束長: 5, 6, 7, 符号化率: 1/2, 1/3, 判定方式: 軟判定, 硬判定

表 4 に拘束長 7, 符号化率 1/2 で実行した際の評価結果を示す。表 4 より、再構成可能デコーダの PE 数を倍にするにつれ、スループットもほぼ倍になっていることが分かる。ビタビ復号を実現する場合、Branch Metric 演算と ACS 演算が並列可能で演算量が多い処

理である。Branch Metric 演算を行う BM_G ユニットは、対応したい拘束長により一意に決定されるため、PE 数の変化により Branch Metric 演算は影響を受けない。ACS 演算は、PE アレイで実現されるため、PE 数によって ACS 演算がボトルネックとなりスループットに差が出てくることが考えられる。これより、要求されるスループットに応じて PE 数を設定することで、最適な再構成可能デコーダが得られることが分かる。

表 4. PE 数の変化によるスループット、面積、消費電力の変化

PE 数	スループット [Mbps]	面積		消費電力 [mW]
		ロジック部 [Kgate]	メモリ部 [Kbit]	
8	12	21	13	96
16	24	26	13	112
32	49	38	13	148
64	98	59	13	193

6. まとめと今後の課題

本研究では、複数の誤り訂正符号に対応する再構成可能デコーダモデルを提案した。提案する再構成可能デコーダは、あらかじめ決められた複数の復号処理を、コンフィギュレーションを変更することにより容易に実現できる。また、再構成可能デコーダモデルによりそのような再構成可能デコーダを容易に実現できる。評価実験より、提案する再構成可能デコーダは、実用的な性能及び面積や消費電力で実現できることが確認できた。

今後の課題は、ビタビ復号やターボ復号以外の復号アルゴリズムへの対応や、与えられた仕様を実現する、最適なデザインパラメータの探索等が挙げられる。

参考文献

- 1) T. Vogt, and N. When, "A Reconfigurable Application Specific Instruction Set Processor for Convolutional and Turbo Decoding in a SDR Environment," Proc. of DATE, pp.38—43, Mar. 2008.
- 2) A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," IEEE Trans. Inform. Theory, vol. IT-13, pp.260—269, Apr. 1967.
- 3) C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon Limit Error-Correcting Coding and Decoding: Turbo-Codes," Proc. of ICC, pp. 1064—1070, vol. 2, May 1993.
- 4) P. Robertson and P. Hoeher, "Optimal and Sub-Optimal Maximum a Posteriori Algorithms Suitable for Turbo Decoding," European Trans. Telecommunication, no. 8, pp. 119—125, Mar./Apr. 1997.