

FPGA を用いたスポッティバイト誤り制御符号の能力検証

Verification of Spotty Byte Error Control Codes Using FPGA

笹田 武志[†]
Takeshi Sasada

金子 晴彦[†]
Haruhiko Kaneko

1. まえがき

宇宙環境下では、半導体メモリ素子に記憶されるデータが放射線の影響によりビット反転するため、データ誤りを訂正・検出する誤り制御符号 (ECC) が必須である。宇宙機のデータレコーダは質量・処理性能等の搭載制約から、多ビットエラーに対する誤り制御能力が高く、かつ符号化率の高い ECC が要求される。宇宙航空研究開発機構では、誤り制御能力、符号化率、回路規模、動作速度等の観点から搭載用として実現可能性の高い Reed-Solomon 符号及びスポッティバイト誤り制御符号を FPGA に実装した[1]。本稿では、ECC 評価回路を用いて 5 種類の誤り制御符号能力を検証した結果を示す。

2. 誤り制御符号

2.1 搭載した符号の概要

本研究で検証する ECC は表 1 に示す 5 種類である。実際には符号①-1 として、2 バイト誤り訂正 (D₁₆EC) Reed-Solomon 符号も FPGA に搭載しているが本研究では検証していない。いずれの符号もパリティ検査行列により定義される線形ブロック符号であり、符号①系は Reed-Solomon (RS) 符号、符号②系はスポッティバイト誤り (*t/b* 誤り) 制御符号である。なお *t/b* 誤りとは *b* ビットバイト中のランダム *t* ビット誤りのことである。

表 1 ECC の機能

ECC	Reed-Solomon code		Spotty byte ECC		
	①-2	①-3	②-1	②-2	②-3
Check-bit length R	48	32	31	32	13
Information-bit length K	128	128	129	128	131
N(=R+K)	176	160	160	160	144
Correctable error	Single-byte error	Single-byte error	3-bit in 2-byte error	Single 4/16-error	Single 2/16-error
Detectable error	Double-byte error		Single-byte error	Double 4/16-error	

符号①-2 : (S₁₆EC-D₁₆ED) Reed-Solomon 符号, 符号①-3 : (S₁₆EC) Reed-Solomon, 符号②-1 : ([TEC]₂-S₁₆ED) 符号, 符号②-2 : (S_{4/16}EC-D_{4/16}ED) 符号, 符号②-3 : (S_{2/16}EC) 符号

スポッティバイト誤り制御符号は、RS 符号と異なりバイト単位での誤り訂正を行わない。任意の *t/b* 誤りを訂正・検出することで符号化率を高めている。符号の機能は数学的に証明されている[2-4]。

RS 符号は、スポッティバイト誤り制御符号との回路規模、動作速度等の比較のため実装した。但し、高速な復

号のため並列復号アルゴリズム[5]を採用した点が、従来の誤り位置多項式に基づく RS 符号の復号と異なる。本アルゴリズムでは、誤りバイト位置及びバイト誤りパターンをシンドロームと部分行列の積により算出できることから、従来よりも少ない遅延で復号が可能となる。

2.2 アルゴリズム

表 1 の ECC は全て線形ブロック符号であり、*R* 行 *N* 列を有する検査行列 **H** の零空間として定義できる。符号②-1 の検査行列の例を式(1)に示す[4]。

$$\mathbf{H} = \begin{bmatrix} I & I & I & I & \dots & I \\ H'' & \gamma^1 H'' & \gamma^2 H'' & \gamma^3 H'' & \dots & \gamma^9 H'' \\ H'' & \gamma^2 H'' & \gamma^4 H'' & \gamma^6 H'' & \dots & \gamma^{18} H'' \\ H'' & \gamma^3 H'' & \gamma^6 H'' & \gamma^9 H'' & \dots & \gamma^{27} H'' \end{bmatrix} \quad (1)$$

ただし、*I* は 16×16 単位行列であり、 $\gamma \in \text{GF}(2^5)$ は GF(2) 上の 5 次の原始多項式 $p(x)=1+x^2+x^5$ の根である。

また、*H''* 及び $\gamma^i H''$ は任意の 3 列ベクトルが線形独立である 5×16 部分行列であり、以下の式(2)により定義する。

$$\begin{aligned} H'' &= [\gamma^0 \gamma^1 \gamma^2 \gamma^3 \gamma^4 \gamma^{11} \gamma^{27} \gamma^{17} \gamma^8 \gamma^{22} \gamma^{25} \gamma^{12} \gamma^{28} \gamma^9 \gamma^{13} \gamma^{15}] \\ \gamma^i H'' &= [\gamma^{0+i} \gamma^{1+i} \gamma^{2+i} \gamma^{3+i} \gamma^{4+i} \gamma^{11+i} \gamma^{27+i} \gamma^{17+i} \\ &\quad \gamma^{8+i} \gamma^{22+i} \gamma^{25+i} \gamma^{12+i} \gamma^{28+i} \gamma^{9+i} \gamma^{13+i} \gamma^{15+i}] \end{aligned} \quad (2)$$

ここで、 γ^i は長さ 5 を有する列ベクトルである。

3. 訂正能力の検証

3.1 誤り制御符号評価回路

評価回路は、USB 制御ブロック、USB ホストからアクセス可能な制御レジスタ、情報語・符号語・受信語・復号語のためのバッファメモリ、6 種類のアルゴリズムより生成した符号化回路、符号化制御回路、符号化回路に対応する 6 種類の並列復号回路、復号制御回路の 7 種類の機能ブロックから成る。

配線リソース制約から 6 種の ECC は 1 個の FPGA に格納できなかった。回路規模の大きい符号①-1 とそれ以外の 5 種の符号は、FPGA にロードし使用の都度入れ替える。

3.2 検証方法

実験機器は、前述の評価回路 (市販の FPGA 及びボード) とそれを制御するホスト PC (PC/AT 互換機) から構成され、機器間は USB ケーブルで接続される。ホスト PC は源泉データである情報語を評価回路に入力する。評価回路は選択された符号化アルゴリズムにて情報語を符号化し、その符号語をホスト PC に返す。ホスト PC は符号語に対して任意の誤りモデルに従ったビット/バイト誤

[†] 宇宙航空研究開発機構, JAXA

りを付加し、受信語として評価回路に出力する。評価回路は誤りを含む受信語に対し誤り制御（復号処理）後、復号語をホスト PC に返す。ホスト PC にて情報語と復号語を比較して誤り制御能力を検証する。

3.3 誤りパターン

評価回路に入力する誤りは、軌道上の観測結果から t/b 誤りとした。すなわち 1 ビット誤りである Single-Event Upset (SEU) が支配的であるが、確率的に t/b 誤りが発生する。本研究ではランダム t ビット誤り率を指数関数の逆数で定義した。また Single-Event Latchup (SEL) 等による回路故障を考慮しバイト誤りをパラメータに加えた。誤りパターンはホスト PC のプログラムで生成し、乱数の巡回を防ぐため十分な精度（配列）を確保した。無線通信路で発生するバースト誤りはここでは考慮していない。

メモリ素子が故障しないケース 1（バイト誤り率 0）と、8 個ある情報部メモリ素子の内 1 個が常に故障状態にあるケース 2（バイト誤り率は検査部も含めて平均的に 0.1 とした）で分け、 t/b 誤り率を変化 ($3 \times 10^{-6} \sim 10^{-2}$) させ誤り制御後の BER を評価した。図 1 に前提としたメモリ素子の構成を示す。

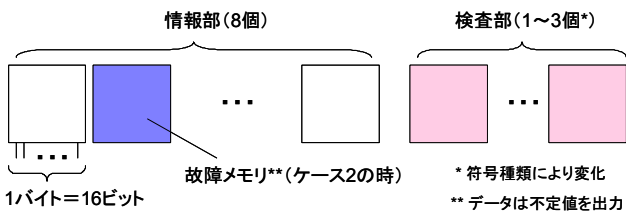


図 1 メモリ素子構成（バイト長=16ビット）

4. 評価結果

図 2 にケース 1 の時間経過に伴う BER 推移（符号①-3 及び②-3）を示す。時間経過とともに BER が収束するが、母数（シミュレーション時間）が十分に多くないため、強い訂正能力の符号ほど散発的な誤りの影響による BER 変動が残存する。

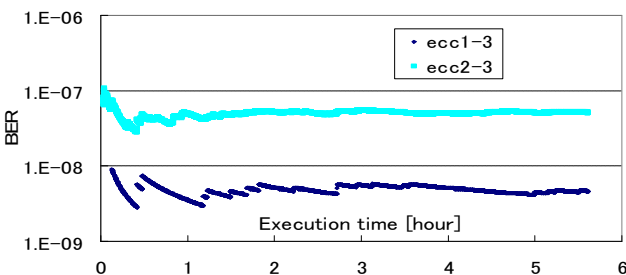


図 2 BER の推移（ケース 1, t/b 誤り率 10^{-4} ）

図 3 にケース 1 のシミュレーション結果を示す。横軸が符号語に印加した t/b 誤り率、縦軸が復号後の BER である。 t/b 誤り率が増加するほど ECC 無しの場合との BER の差が減少する。 t/b 誤り率が 3×10^{-6} 以下では、確率的に誤りが発生しない符号がある（評価データ： 3×10^{11} bit）。ケース 2 では、符号能力を超える高い確率でバイト誤りが発生するため、いずれの符号でも BER 改善は起こらず ECC 無しの場合とあまり変わらない。

図 4 はケース 2 の誤訂正率である。ここで誤訂正率は、誤りを含むワードを全ワードで除したものである。バイ

ト誤り率が高いため、各符号ほぼ一定の誤訂正率を示した。機能的に 1 バイト誤り検出能力を保証しない符号②-2 が低い誤訂正率であった。これは 1 バイト誤りが不定値のため $1/2$ の確率で 8 ビット以下の誤りになること、及びシンδροーム空間における誤り訂正のためのシンδροームの割合が小さいためと考えられる。最も検出能力が高い符号①-2 は誤訂正が発生していない。

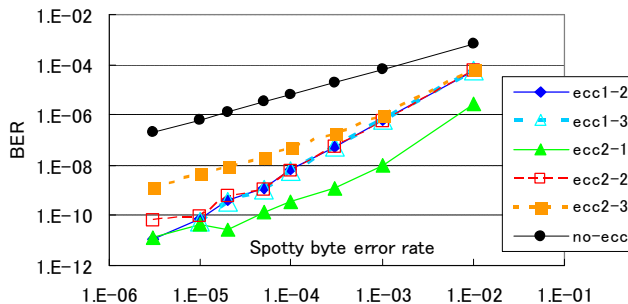


図 3 BER 測定結果（ケース 1）

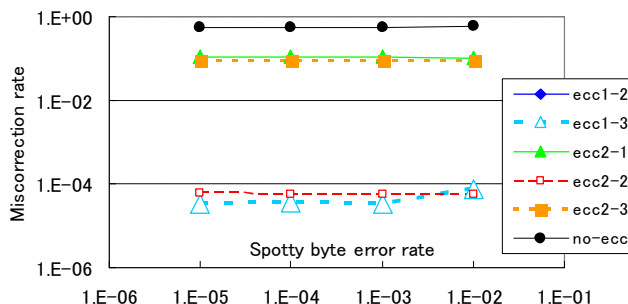


図 4 誤訂正率測定結果（ケース 2）

5. まとめ

並列復号法による RS 符号及びスポッティバイト誤り制御符号のアルゴリズムを、回路 (FPGA) に実装し符号能力を検証した。理論的に扱うことが困難な誤り制御能力外の誤りについて、ハードウェアの高速性を活かし多くの誤りパターンで BER 及び誤検出率を求めることができた。今後は残る符号①-1 の検証を行う予定である。

(株) 日立インフォメーションテクノロジー 今城英樹氏は評価回路製作で、宇宙技術開発 (株) 金井俊樹氏はシミュレーションで協力頂きました。ここに感謝致します。

参考文献

- [1] 笹田 武志, 金子 晴彦, 今城 英樹: スポッティバイト誤り制御符号評価回路の開発, 信学技報 Vol.105, No.102 SAT2006-2, pp.7-12, Feb. 2006.
- [2] 榎山 俊彦, 鈴木 一克, 藤原 英二: スポッティバイト誤り制御符号; 2004 年情報科学技術フォーラム講演論文集, LC-006, Sep. 2004.
- [3] Kazuyoshi Suzuki, Toshihiko Kashiyama, and Eiji Fujiwara, "A General Class of M-Spotty Byte Error Control Codes," Proc. Asia-Europe Workshop on Information Theory, pp.24-26, October 2004.
- [4] 清水 護, 鈴木 一克, 榎山 俊彦, 藤原 英二: 誤りバイト数を制限した m-スポッティバイト誤り訂正符号; 2005 年電子情報通信学会総合大会講演論文集, D-10-9, March 2005.
- [5] 北神 正人, 小林 鉄太郎, 藤原 英二, "バイト誤り訂正符号における並列復号法," 1995 年電子情報通信学会ソサイエティ大会論文集, A-114, Sep. 1995.