パルス演算に基づくLDPCエラー訂正処理とそのアーキテクチャ

LSI Architectures for LDPC Error Correction based on Stochastic Logic

宮曦媛　浅井哲也　本村真人
Xiyuan Gong　Tetsuya Asai　Masato Motomura

北海道大学大学院情報科学研究科
Graduate School of Information Science and Technology, Hokkaido University

1　まえがき

LDPC 符号（低密度パリティ検査符号）は Gallager により 1960 年代に提案された (R. Gallager, *Low Density Parity-Check Codes. MIT Press*, Cambridge, MA, 1963)。LDPC 符号は強力なエラー訂正符号であるが、訂正のアルゴリズムは複雑である。通常は乗算回数を減らして処理を簡略する手法 (BCJR アルゴリズム) が用いられるが、本稿ではあえて乗算回数を減らさず、回路構成が容易なパルス変調型乗算器を複数用いたストリーム処理型の LDPC エラー訂正回路を提案する。

2　LDPC の積和演算アルゴリズム

通信路が 2 元対称通信路であるとし、4 ビットの送受信語を仮定してアルゴリズムを説明する。送信語 $x = (0, 1, 1, 1)$、受信語 $y = (0, 0, 1, 1)$、$m \times n$ の検査行列

$$H = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \quad (1)$$

を例とする。送信語の i 番目のビットを 1 として送信した時に、そのビットが 0 または 1 として受信される確率 P_i は $P(y_i|x_i = 1)$ で表される。この確率を用いて、行列

$$P = \begin{bmatrix} P_{11} & P_{21} & * & P_{41} \\ P_{12} & * & P_{32} & P_{42} \\ * & P_{23} & P_{33} & * \end{bmatrix}, P_{ij} = P_i \ (for\ all\ j) \quad (2)$$

を求める。図 1 に 1 行分の計算例を示す。ステップ 1 ですべての行について上記の演算を行い、下記の行列

$$PP = \begin{bmatrix} PP_{11} & PP_{21} & * & PP_{41} \\ PP_{12} & * & PP_{32} & PP_{42} \\ * & PP_{23} & PP_{33} & * \end{bmatrix} \quad (3)$$

を求める。ステップ 2 では列方向に同様な演算を行い、PP から新しい P 行列を生成する ($P_{ij} = P_{ij}(0)/(P_{ij}(0) + P_{ij}(1))$)。上記の演算を繰り返し行う (ステップ 1 からステップ 2 までが 1 つのループとなる)。n 回演算を繰り返した後、n+1 回目に行演算を行い、ステップ 3 の演算を行う (ここでビット推定を行う)。

3　提案アーキテクチャ

上記のように、LDPC エラー訂正アルゴリズムの演算の中心は「乗算」である。通常は、A × B を求めるために多ビットの乗算器が必要であるが、この A, B をランダムなパルス列 (ただし、パルス密度は A, B に比例) に変換すると、乗算器は AND ゲートのみで実現できる (T. Kondo, et. al., *IEEE Trans. Appl. Superconduct.*, 2005)。図 2 に提案するアーキテクチャの概略を示す。送信語データ取り組み段 (以下 DF) でワード 1 をパイプラインレジスタ 1 に書き込む。次に、演算段 (以下 EX) でワード 1 をパルス密度に変換して積和演算を行う。同時にワード 2 を読み込む。ワード 1 がパリティ検査と出力段 (以下 PC) のパイプラインレジスタに取り込まれるのと同時に、ワード 2 が EX 段に与えられる。PC 段に取り込まれた (積和演算処理後の) ワード 1 のパルス列をカウントし、パリティ検査を行う。DF および PC にかかる時間 ($T_{DF/PC}$) は、データ長で決まる。また、EX にかかる時間 (T_{EX}) は繰り返しの処理回数ではなく、パルス演算の精度 (パルスカウンタのビット数) により決まる。$T_{DF/PC} = T_{EX}$ であれば、パイプラインを乱すことなくエラー訂正が可能である。パイプラインステージの概略を図 3 に示す。

謝辞：本研究の遂行にあたり、有益なご助言を頂いた (株) 東芝研究開発センター 丸亀孝生氏, 西義史氏, 木下敦寛氏に感謝の意を表します。

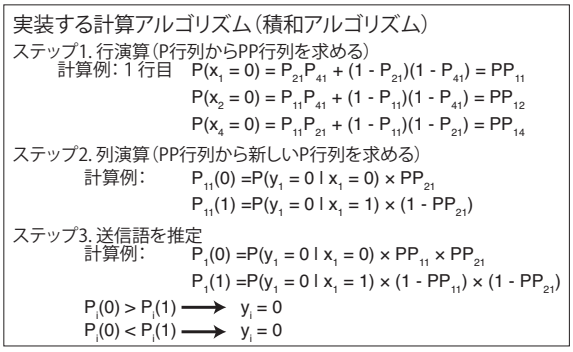

図 1　積和演算アルゴリズムの計算例

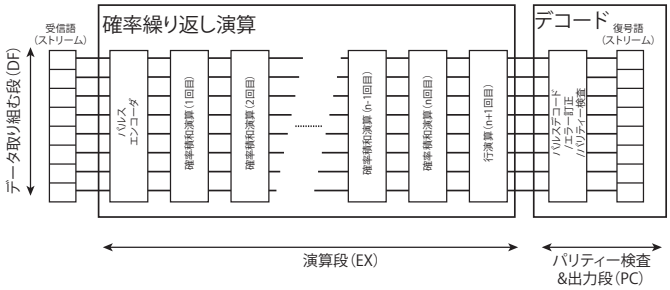

図 2　提案アーキテクチャ

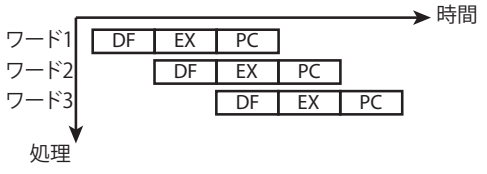

図 3　復号処理のパイプラインステージ