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

2007－ＳＬＤＭ－132（31）
2007／11／22

Min-Sumアルゴリズムを用いた 高速無線LANシステム用LDPC復号器の設計

濱　　希†　島尻　寛之†　吉田たけお†

† 琉球大学 工学部 情報工学科
E-mail: †{hama,shimajiri,tyoshida}@fts.ie.u-ryukyu.ac.jp

あらまし　本稿では，Min-Sum アルゴリズムを用いた高速無線 LAN システム用 LDPC 復号器を設計する．設計する復号器は，次世代無線 LAN システム規格である IEEE 802.11n に準拠した LDPC 符号を使用する．また，これは符号長 648, 1296, 1944 ビットと符号化率 1/2, 2/3, 3/4, 5/6 に対応した復号器である．設計した復号器の総面積は 69,467,024[nm^2] となった．

キーワード　低密度パリティ検査符号, IEEE 802.11n, 復号器, Min-Sum アルゴリズム, 無線 LAN

An LDPC Decoder Based on the Min-Sum Algorithm for High Speed WLAN Systems

Nozomu HAMA†, Hiroyuki SHIMAJIRI†, and Takeo YOSHIDA†

† Department of Information Engineering, Faculty of Engineering, University of the Ryukyus
E-mail: †{hama,shimajiri,tyoshida}@fts.ie.u-ryukyu.ac.jp

Abstract　In this paper, we show an architecture of low density parity check (LDPC) decoders based on the Min-Sum algorithm for high speed WLAN systems. The decoder supports twelve combinations of code lengths 648, 1296, 1944 bits and code rates 1/2, 2/3, 3/4 , 5/6 based on IEEE 802.11n standard. The total cell area of our decoder is 69,467,024 [nm^2].

Key words　LDPC Code, IEEE 802.11n, Decoder, Min-Sum Algorithm, WLAN

1. はじめに

低密度パリティ検査 (Low Density Parity Check : LDPC) 符号は，Shannon 限界に迫る高い誤り訂正能力を持つことが確認され，近年，次世代の誤り訂正符号として注目を集めている [1]～[5]．LDPC 符号は，一般に，Sum-Product アルゴリズムなどを用いた繰り返し処理によって復号される．この繰り返し処理は，ハードウェアの並列実装に適しており，高いスループットが要求される分野への応用が期待されている [6]．しかし，Sum-Product アルゴリズムを並列実装した場合，回路規模が非常に大きくなってしまうという問題がある．

一方，復号処理における演算量を低減するために，Sum-Product アルゴリズムの近似法である Min-Sum アルゴリズムが提案されている．Min-Sum アルゴリズムは，Sum-Product アルゴリズムで必要としている乗算を比較演算に置き換えることによって，演算量を低減しており，LDPC 復号器の実装の際に採用されることが多い [7], [8]．しかし，Min-Sum アルゴリズムの復号能力は，Sum-Product アルゴリズムの復号能力に比べて若干劣ることが知られている [9]．

文献 [10] では，次世代無線 LAN システム規格である IEEE 802.11n で検討されている LDPC 符号を対象として，Min-Sum アルゴリズムをハードウェア実装する際の，パラメータ値の検討を行っている．文献 [10] の結果に基づいて，Min-Sum アルゴリズムをハードウェア実装することにより，実用上十分な復号能力を持った LDPC 復号器の実現が可能となる．本稿では，文献 [10] で示されているパラメータ値を用いて，Min-Sum アルゴリズムをハードウェア実装し，回路規模の評価を行う．

以下 **2.** で，LDPC 符号の概要について述べる．続いて **3.** では，設計した LDPC 復号器の設計要件について述べる．次に **4.** で LDPC 復号器のアーキテクチャについて述べる．そして **5.** では，設計した回路の評価を示し，最後に **6.** で，本稿のまとめと今後の課題について述べる．

2. LDPC 符号とその復号法

2.1 LDPC 符号の概要

LDPC 符号は，非常に疎な検査行列によって定義される線形ブロック符号である．LDPC 符号は，その検査行列の行重みと列重みがともに一定値である正則 (regular) LDPC 符号と，一定

値でない非正則 (irregular) LDPC 符号に分けられる．以下では，IEEE 802.11n で検討されている LDPC 符号の検査行列の表現方法を説明する．なお，IEEE 802.11n の LDPC 符号は，非正則 LDPC 符号である．

まず，$z \times z$ の単位行列の各行を，s だけ右に巡回シフトした正方行列を $I(s)$ と表すことにする．また，便宜上，$z \times z$ の零行列を $I(-1)$ と表すことにする．IEEE 802.11n の LDPC 符号の検査行列は，このような $z \times z$ の正方行列 $I(s), -1 \leqq s \leqq z-1$ を部分行列として持つように構成されている．このとき，符号長 $n = z \times N$，情報記号数 k，検査記号数 $m = n - k = z \times M$ の非正則 LDPC 符号の検査行列 H は，

$$H = \begin{pmatrix} I(s_{1,1}) & I(s_{1,2}) & \cdots & I(s_{1,N}) \\ I(s_{2,1}) & I(s_{2,2}) & \cdots & I(s_{2,N}) \\ \vdots & \vdots & \ddots & \vdots \\ I(s_{M,1}) & I(s_{M,2}) & \cdots & I(s_{M,N}) \end{pmatrix} \quad (1)$$

と表される．なお実際には，$M \times N$ の行列 H' を用いて，

$$H' = \begin{pmatrix} s_{1,1} & s_{1,2} & \cdots & s_{1,N} \\ s_{2,1} & s_{2,2} & \cdots & s_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ s_{M,1} & s_{M,2} & \cdots & s_{M,N} \end{pmatrix} \quad (2)$$

のように，部分行列の情報だけを表示することによって，検査行列 H を表す．

IEEE 802.11n の LDPC 符号の検査行列は，符号長 $n = 648, 1296, 1944$ の 3 種類が定められている．このとき，正方行列 $I(s)$ のサイズは，それぞれ $z = 27, 54, 81$ となっている．また，それぞれの符号長に対して，符号化率 $R = 1/2, 2/3, 3/4, 5/6$ の 4 種類が定められている．すなわち，IEEE 802.11n の LDPC 符号の検査行列は，符号長と符号化率の組み合わせにより，合計 12 種類ある．本稿では，これら 12 種類全ての検査行列に対応した LDPC 復号器を設計する．

2.2 Min-Sum アルゴリズム

本稿では，LDPC 符号の復号法として，Min-Sum アルゴリズムを採用する．ここでは，Min-Sum アルゴリズムについて説明する．なお以下では，検査行列 H の (i,j) 成分を H_{ij} と表す．ただし，$1 \leqq i \leqq m, 1 \leqq j \leqq n$ である．また，集合 A から，要素 a を取り除いて得られる集合を $A \backslash a$ と表す．さらに，

$$A(i) = \{j | H_{ij} = 1\} \quad (3)$$

$$B(j) = \{i | H_{ij} = 1\} \quad (4)$$

とする．

以上の準備のもとで，Min-Sum アルゴリズムを以下に示す．

（1） 初期化

$j = 1, 2, \cdots, n$ に対して，受信記号 y_j から対数尤度比 λ_j を算出する．ここで，二値入力 AWGN 通信路における対数尤度比 λ_j は以下のようになる．

$$\lambda_j = \ln \frac{P(y_j | x_j = 0)}{P(y_j | x_j = 1)} = \frac{2y_j}{\sigma^2}$$

なお，上式において，x_j は送信記号を，σ^2 は分散を表している．また，$H_{ij} = 1$ を満たす (i,j) に対して，対数事前値比 β_{ij} を $\beta_{ij} = 0$ と初期化する．さらに，反復回数をカウントする変数 l に $l = 1$ を，最大反復回数を数える変数 $l_{\max}$ に適当な値を，それぞれ設定する．

（2） 行処理

$i = 1, 2, \cdots, m$ の順に，$H_{ij} = 1$ を満たす (i,j) に対して，以下の式を用いて，対数外部値比 α_{ij} を更新する．

$$\alpha_{ij} = \left(\prod_{j' \in A(i) \backslash j} \mathrm{sign}(\lambda_{j'} + \beta_{ij'}) \right) \min_{j' \in A(i) \backslash j} |\lambda_{j'} + \beta_{ij'}|$$

ただし，

$$\mathrm{sign}(x) = \begin{cases} 1 & x \geqq 0 \text{ のとき} \\ -1 & x < 0 \text{ のとき} \end{cases}$$

とする．

（3） 列処理

$j = 1, 2, \cdots, n$ の順に，$H_{ij} = 1$ を満たす (i,j) に対して，以下の式を用いて，対数事前値比 β_{ij} を更新する．

$$\beta_{ij} = \sum_{i' \in B(j) \backslash i} \alpha_{i'j}$$

（4） 一時推定語の計算

$j = 1, 2, \cdots, n$ に対して，以下の式を用いて，一時推定語 $\hat{c}_j$ を計算する．

$$\hat{c}_j = \begin{cases} 0 & \mathrm{sign}(\lambda_j + \sum_{i' \in B(j)} \alpha_{i'j}) = 1 \text{ のとき} \\ 1 & \mathrm{sign}(\lambda_j + \sum_{i' \in B(j)} \alpha_{i'j}) = -1 \text{ のとき} \end{cases}$$

（5） パリティ検査

一時推定語 $(\hat{c}_1, \hat{c}_2, \cdots, \hat{c}_n)$ が，以下の式を満たすかどうかを検査する．満たした場合は，アルゴリズムを終了する．

$$H \cdot (\hat{c}_1, \hat{c}_2, \cdots, \hat{c}_n)^T = \mathbf{0}$$

（6） 反復回数のカウント

$l < l_{\max}$ ならば，$l = l + 1$ として (2) へ戻る．$l = l_{\max}$ ならば，$(\hat{c}_1, \hat{c}_2, \cdots, \hat{c}_n)$ を推定語として出力し，アルゴリズムを終了する．

3. LDPC 復号器の設計要件

ここでは，提案する LDPC 復号器の設計要件について述べる．

1. で述べたように，Min-Sum アルゴリズムにおける固定小数点の精度やループ回数等の設計パラメータに関して，文献 [10] に誤り訂正性能の調査結果が報告されている．

文献 [10] によれば，Min-Sum アルゴリズムにおいては，16 ビットの固定小数点 (符号ビット，整数部 11 ビット，少数部 4 ビット) でループ回数を 30 回以上とした場合に，十分な誤り訂正率を達成できることが示されている．本稿では文献 [10] の調査結果に基づき，演算精度を 16 ビット固定小数点，ループ回数を 30 回とした Min-Sum アルゴリズムによる LDPC 復号器の構成について検討する．

本稿で提案する LDPC 復号器では，IEEE 802.11n で採用が検討されている LDPC 符号の 12 種類の検査行列全てに対応する必要がある．文献 [10] によれば，Min-Sum アルゴリズムのループ処理を 30 回実行しなければならないため，LDPC 復号器が絶え間なく受信語を復号するには，1 回のループ処理を 21(= $\lfloor 648/30 \rfloor$) サイクル以内に収める必要がある．また，LDPC 符号の最大の検査行列のサイズが 1944 × 972 ビットであるため，演算量，回路規模ともに大きくなってしまう問題がある．以上のことから，本稿で提案する LDPC 復号器は，以下の要件を満たすように設計を行った．

- IEEE 802.11n の 12 種類の検査行列に対応
- 1 回のループ処理を 21 サイクル以内で実行 (符号長 648 ビットの場合)
- 回路面積の削減

4. LDPC 復号器のアーキテクチャ

ここでは，我々が設計した LDPC 復号器のアーキテクチャについて述べる．図 1 に，提案する LDPC 復号器のブロック図を示す．

4.1 MUL, IBUF, LAMD

MUL は 16 ビットの乗算器で構成され，受信語 y_j と $2/\sigma^2$ の積 λ_j を求める．16 ビット乗算を 4 サイクルで実行し，演算結果を後段の IBUF に出力する．

IBUF は 1944 × 16 ビットのシフトレジスタで構成され，16 ビット毎にシフトしながら，MUL から送られてくる λ_j を格納する．

LAMD は 1944 × 16 ビットのレジスタで構成され，復号処理中の λ_j を格納する．復号処理が完了した時点で，IBUF 内に復号可能な λ_j が格納されている場合に，LAMD は IBUF から必要なデータを取り込む．

受信データの取り込みを IBUF で行い，復号中のデータの保持を LAMD で行うことにより，絶え間なく受信語の入力と復号処理を行うことができる．なお，IBUF と LAMD は LDPC 符号の全ての検査行列に対応するため，最大の符号長 1944 ビットに合わせた容量としている．また，検査行列毎に LAMD を参照するアドレスが異なるため，LAMD の後段にマルチプレクサ (MUX) を配置している．図 1 では MUX の表示は省略している．

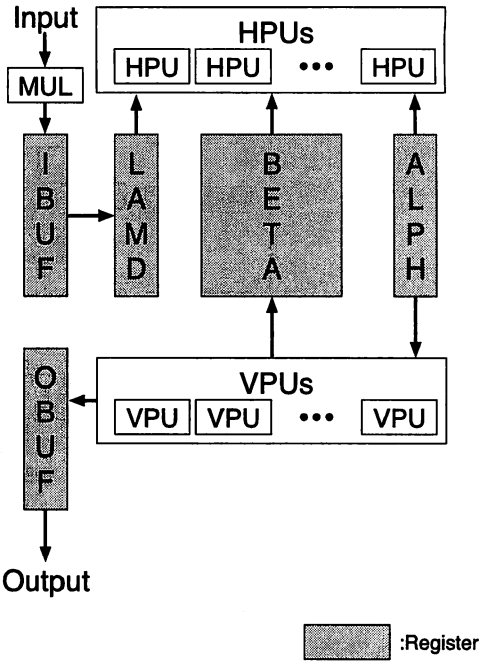

図 1 LDPC 復号器の構成

4.2 行処理回路 (HPU)

HPU は行処理を行うユニットである．**2.** で示した対数外部値比 α_{ij} を求める式より，行処理では $\lambda_i + \beta_{iq}$ の最小値を求めていることがわかる．ここで q は，検査行列中の i 行目の 1 が立っている列を表し，$q \neq j$ とする．これを i 行目の全ての α_{ij} について考えると，$\lambda_i + \beta_{ij}$ の結果が最小値となる列以外では，α_{ij} は全てこの最小値となる．一方，$\lambda_i + \beta_{ij}$ の結果が最小値となる列では，α_{ij} は，$\lambda_i + \beta_{ij}$ の結果が 2 番目に小さい値をとることになる．したがって 1 つの行に関して，$\lambda_i + \beta_{ij}$ の結果のうち，最小値と 2 番目に小さい値，最小値となった列のアドレスを求めるだけでよいことがわかる．

図 2 に HPU のブロック図を示す．行処理を行う HPU では上で述べたように，各行を処理するために，行毎に $\lambda_i + \beta_{ij}$ の最小値と 2 番目に小さい値を求めなければならない．各検査行列の各行中の 1 が立っている列の数は，高々24 列しか存在しないため，HPU は 24 個の 16 ビットデータを比較するだけで良い．なお比較器ツリーの各段の間には，HPU の遅延時間を削減するために，パイプラインレジスタを挿入している．λ_j と β_{ij} の加算を含め，24 個の 16 ビットデータの中から最小の 2 つのデータを求めるのに，7 サイクル要する．また，求めた最小値の符号ビットを求めるための XOR ツリーを HPU 内に実装している．

HPU から出力される結果は，最小の 2 つのデータと最小値の列のアドレスである．列のアドレスは，検査行列内の 24 個の単位行列に含まれるかが分かればよいため，5 ビットで表現することができる．したがって，HPU の出力のビット幅は 37

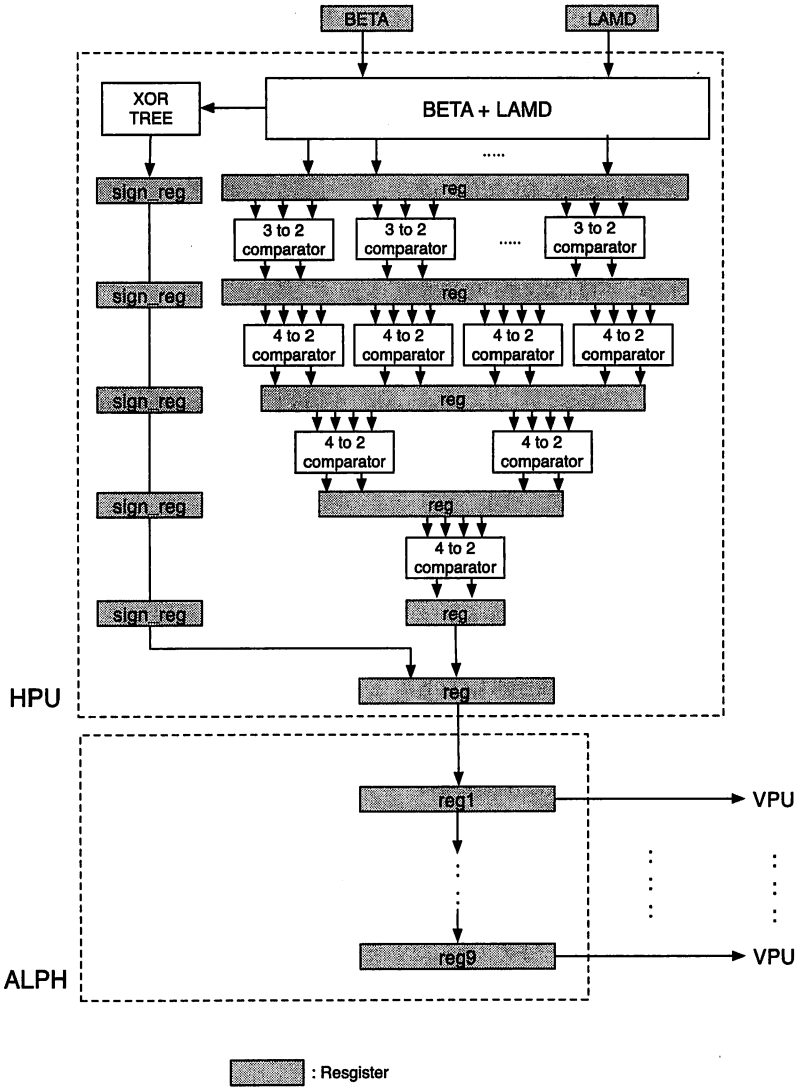

図2 HPU の構成

ビットとなる．この結果，行処理の結果 α_{ij} を格納するレジスタ ALPH のサイズは，972 × 37 ビットとなる．ここで 972 とは，LDPC 符号の最大サイズの検査行列の行数である．

各行に対する行処理は，全ての行について並列に実行することができる．しかし，全ての行を並列に処理するためには，972 個の HPU を実装しなければならない．そこで提案する LDPC 復号器では，図 2 に示すように，HPU の後段にパイプラインレジスタを追加し，1 つの HPU で複数の行をオーバーラップして処理することにする．オーバーラップして処理される行の数は，復号対象の検査行列の符号長によって変わり，符号長が 648 ビットの場合は 3 行，1296 ビットの場合は 6 行，1944 ビットの場合は 9 行処理する．これにより，必要な HPU の数を 108 個に削減することができる．また，行処理に要する実行サイクル数は，符号長が 648 ビットの場合は 10 サイクル，1296 ビットの場合は 13 サイクル，1944 ビットの場合は 16 サイクルとなる．

4.3 列処理回路 (VPU)

VPU は列処理を行うユニットである．**2.** で示した対数事前値比 β_{ij} を求める式より，列処理では α_{pj} の総和を求めている．ここで p は，検査行列中の j 列目の 1 が立っている行を表し，$p \neq i$ とする．このことから，j 列目の全ての β_{ij} に着目すると，検査行列の 1 が立っている行の α_{ij} を全て加算し，その総和から各行の α_{ij} を引くことによって，j 列目の全ての β_{ij} を求めることができる．また **2.** で示した一時推定語 $\hat{c}$ を求める式より，α_{ij} の総和から対応する列の λ_j を加算した結果から $\hat{c}$ を求めることができる．

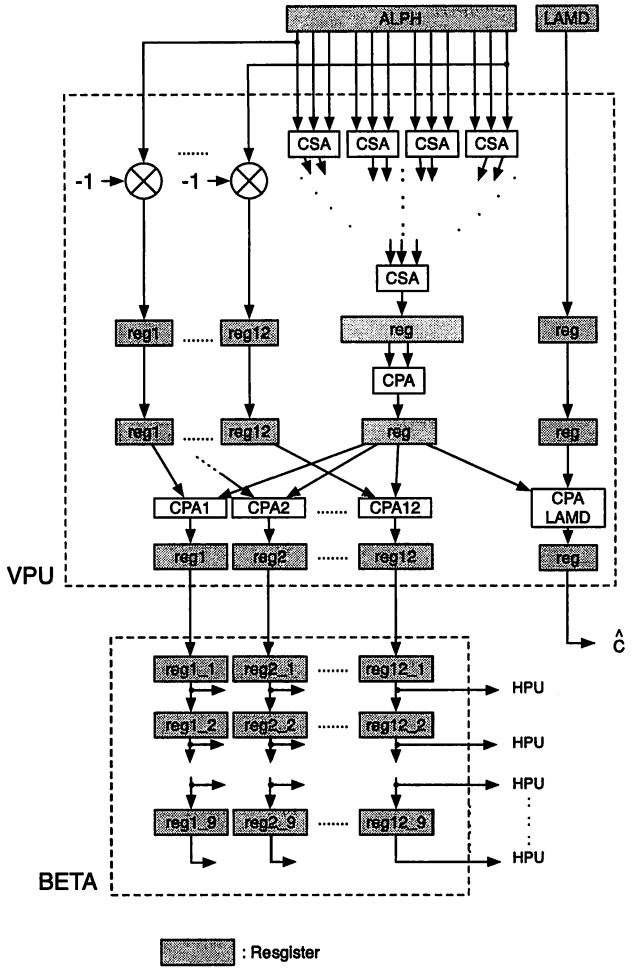

図3 VPU の構成

図 3 に VPU のブロック図を示す．列処理を行う VPU では上で述べたように，各列を処理するために，検査行列の 1 が立っている行の α_{ij} を全て加算しなければならない．各検査行列の各列中の 1 が立っている行の数は，高々12 行しか存在しないため，VPU は 12 個の 16 ビットデータを加算するだけで良い．VPU の内部には，12 個の 16 ビットデータを加算する CSA(Carry Save Adder) ツリーと CPA(Carry Propagation Adder)，1 列中の各 β_{ij} を求めるための 12 個の CPA と符号反転回路を実装する．この他，一時推定語 $\hat{c}$ を求めるための λ_j との加算を行う CPA も実装する．なお HPU と同様に，CSA ツリーと CPA，CPA と CPA の間には，遅延時間を削減するためにパイプラインレジスタを挿入している．VPU では 1 列分の β_{ij} を求めるのに，4 サイクル要する．

VPU から出力される結果は，12 個の 16 ビットの β_{ij} と 1 ビットの一時推定語 $\hat{c}$ である．列処理の結果 β_{ij} を格納するレジスタ BETA のサイズは，$1944 \times 12 \times 16$ ビットとなる．また一時推定語 $\hat{c}$ は，ループ処理が 30 回繰り返された後，OBUF に格納される．

各列に対する列処理は，全ての列について並列に実行することができる．しかし，全ての列を並列に処理するためには，1944 個の VPU を実装しなければならない．ここで 1944 とは，LDPC 符号の最大サイズの検査行列の列数である．提案する LDPC 復号器では HPU と同様に図 3 に示すように，VPU の後段にパイプラインレジスタを追加し，1 つの VPU で複数の列をオーバーラップして処理することにする．オーバーラップして処理される列の数は HPU と同じである．これにより，必要な VPU の数を 216 個に削減することができる．また，列処理に要する実行サイクル数は，符号長が 648 ビットの場合は 7 サイクル，1296 ビットの場合は 10 サイクル，1944 ビットの場合は 13 サイクルとなる．この結果，行処理と列処理に要するサイクル数，すなわち 1 回のループ処理に要するサイクル数は，符号長が 648 ビットの場合は 17 サイクル，1296 ビットの場合は 23 サイクル，1944 ビットの場合は 29 サイクルとなる．この 1 回のループ処理に要するサイクル数は，それぞれの符号長

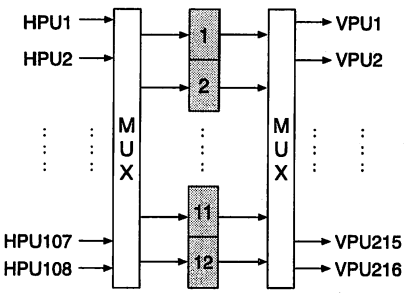

図4 ALPH の構成

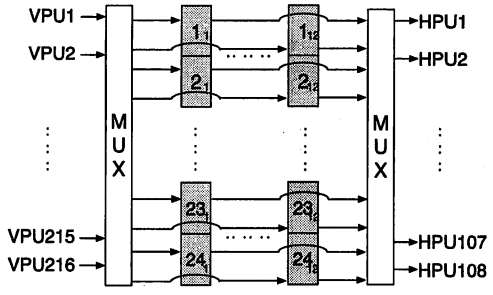

図5 BETA の構成

の符号語が絶え間なく LDPC 復号器に入力された場合でも，30 回のループ処理を実行するのに十分間に合うサイクル数である．

4.4 ALPH と BETA

行処理と列処理の結果を格納する ALPH と BETA のブロック図を，図 4 と図 5 に示す．

この 2 つのレジスタは，IEEE 802.11n の LDPC 符号の全ての検査行列に対応するため，最大サイズの検査行列に合わせた容量を実装している．また，復号中の検査行列に応じたデータを取得，格納するために，これらのレジスタの前後にはマルチプレクサを配置している．

4.5 OBUF

OBUF は 30 回のループ処理後に VPU から出力される一時推定語 $\hat{c}$ を格納し，外部に出力する．OBUF は 1640 ビットのシフトレジスタで構成され，格納された一時推定語 $\hat{c}$ を 1 ビットずつシフトしながら出力する．ここで 1640 ビットとは，符号長 1944 ビット，符号化率が 5/6 の場合の符号語の情報部のビット数である．OBUF により，復号処理と復号語の出力処理を並列に行うことができる．

なお，今回提案した LDPC 復号器では，ループ回数を常に 30 回としているため，パリティ検査のための回路を省略している．

5. 評　価

本稿では，次世代無線 LAN システム規格である IEEE 802.11n に準拠した LDPC 復号器を設計した．今回設計した LDPC 復号器の諸元を，表 1 に示す．なお，論理合成には，Synopsys 社の Design Compiler Ver.2006.06, TSMC 社の 0.13μm Standard Cell Library を使用した．また，設計した復号器の各回路の合成結果を，表 2 に示す．

表1 LDPC 復号器の諸元

動作周波数	500MHz
符号長	648 , 1296 , 1944
符号化率	$\frac{1}{2}, \frac{2}{3}, \frac{3}{4}, \frac{5}{6}$
総面積 (nm^2)	69,467,024
スループット	250Mbps

表2 各回路の合成結果

	回路面積 (nm^2)	遅延時間 (ns)
MUL	17,486	1.89
IBUF	1,637,287	1.78
LAMD	1,637,287	1.78
HPU	7,560,067	1.93
ALPH	6,150,692	1.80
VPU	16,087,680	1.71
BETA	36,261,840	1.93
OBUF	109,691	1.60

6. おわりに

本稿では，IEEE 802.11n に準拠し，Min-Sum アルゴリズムを用いた LDPC 復号器の設計を行った．またこれは，12 種類の検査行列に対応できる復号器である．さらに，1 つの HPU, VPU において複数の行，列をオーバーラップして処理を行い，回路面積を削減した．列処理を行う VPU と列処理の結果を保持する BETA の回路面積が大半を占めているので，今後の課題としては，VPU と BETA の最適化，さらなるオーバーラップによる回路面積の削減等が挙げられる．

謝辞　本研究 (の一部) は，文部科学省知的クラスター創成事業 (第 II 期) の支援による．

文　献

[1] R.G. Gallager, "Low-density parity-check codes," Cambridge, MA : MIT press 1963.
[2] R.G. Gallager, "Low-density parity-check codes," IRE Trans. Inf. Theory, vol. 8, pp21-28, Jan.1962.
[3] D.J.C. Mackay, "Good error-correcting codes based on very sparse matricces,"IEEE Tans.Inf.Theory, vol.45, pp399-432, Mar. 1999.
[4] D.J.C. Mackay, "Information Theory, Inference, and Learning Algorithms," Cambridg Univ. Press, Cambridge, 2003.
[5] Yang Sun , Marhan Karkooti, R. Cavallaro, "High Throughput, Parallel, Scallable LDPC Encoder/Decoder Architecture for OFDM Systems." IEEE CAS Workshop, pp39-42, Oct 2006.
[6] S.A. Mujtaba, "TGn sync proposal technical specfication," IEEE 802.11-04/0889r6, May 2005.
[7] M. Fossorier, M. Mihaljevic and H. Imai, "Reduced complexity iterative decoding of low-density parity check codes based on belief propagation," IEEE Trans.Commun., vol.47, pp673-680, May 1999.
[8] J. Chen and M. Fossorier, "Density evolution of two improved BP-based algorithms for LDPC decoding,"IEEE Trans. Commun.Lett, vol.2, pp208-210, March, 2002.
[9] 和田山 正, "低密度パリティ検査符号とその復号法," トリケップス, 2002.
[10] 嘉村 隆, 黒崎正行, 尾知 博, "次世代無線 LAN システム用 LDPC 符号の固定小数点化に関する一検討," CAS2005-133, Vol.105, No.634, pp.91-96, 2005.