Project	IEEE 802.16 Broadband Wireless Access Working Group <**http://ieee802.org/16**>
Title	LDPC coding for OFDMA PHY
Date Submitted	2004-11-04

Source(s)

Brian Classon	brian.classon@motorola.com
Yufei Blankenship	yufei.blankenship@motorola.com
Motorola	
Jerry Kim	kimjy@samsung.com
Panyuh Joo	panyuh@samsung.com
Gyubum Kyung	gyubum.kyung@samsung.com
Hongsil Jeong	hongsil.jeong@samsung.com
Sang-Hyo Kim	sanghyo7.kim@samsung.com
Hanju Kim	hanju.kim@samsung.com
DS Park	dspark@samsung.com
Jaeho Jeon	jhjeon@samsung.com
Samsung	
Eric Jacobsen	eric.a.jacobsen@intel.com
Bo Xia	bo.xia@intel.com
Intel	
Kyuhyuk Chung	kyuhyuk@lge.com
Min-seok Oh	minoh@lge.com
LG Electronics	
Brian Johnson	brjohnso@nortelnetworks.com
Alek Purkovic	apurkovi@nortelnetworks.com
Nina Burns	nburns@nortelnetworks.com
Sergey Sukobok	ssukobok@nortelnetworks.com
Nortel Networks	
Eli Sasha	shasha@runcom.co.il
Zion Hadad	zionh@runcom.co.il
Yigal Leiba	yigall@runcom.co.il
Itzik Kitroser	itzikk@runcom.co.il
Yossi Segal	yossis@runcom.co.il
Oren Elkayam	Elkayam@runcom.co.il
Runcom Technologies	
Simon Litsyn	litsyn@eng.tau.ac.il
Eran Sharon	eransh@eng.tau.ac.il
Tel-Aviv University.	

Dale Hocevar hocevar@ti.com
Anuj Batra batra@ti.com
Texas Instruments

Victor Stolpman Victor.Stolpman@nokia.com
Jianzhong (Charlie) Zhang Charlie.Zhang@nokia.com
Nico Van Waes
Nokia

Masoud Olfat masoud.olfat@nextel.com
Nextel Communications

Re:	IEEE P802.16-REVe/D5-2004, sponsor ballot
Abstract	This contribution contains additional text output from an informal LDPC group.
Purpose	Provide additional LDPC specification text.
Notice	This document has been prepared to assist IEEE 802.16. It is offered as a basis for discussion and is not binding on the contributing individual(s) or organization(s). The material in this document is subject to change in form and content after further study. The contributor(s) reserve(s) the right to add, amend or withdraw material contained herein.
Release	The contributor grants a free, irrevocable license to the IEEE to incorporate material contained in this contribution, and any modifications thereof, in the creation of an IEEE Standards publication; to copyright in the IEEE's name any IEEE Standards publication even though it may include portions of this contribution; and at the IEEE's sole discretion to permit others to reproduce in whole or in part the resulting IEEE Standards publication. The contributor also acknowledges and accepts that this contribution may be made public by IEEE 802.16.
Patent Policy and Procedures	The contributor is familiar with the IEEE 802.16 Patent Policy and Procedures <http://ieee802.org/16/ipr/patents/policy.html>, including the statement "IEEE standards may include the known use of patent(s), including patent applications, provided the IEEE receives assurance from the patent holder or applicant with respect to patents essential for compliance with both mandatory and optional portions of the standard." Early disclosure to the Working Group of patent information that might be relevant to the standard is essential to reduce the possibility for delays in the development process and increase the likelihood that the draft publication will be approved for publication. Please notify the Chair <mailto:chair@wirelessman.org> as early as possible, in written or electronic form, if patented technology (or technology under patent application) might be incorporated into a draft standard being developed within the IEEE 802.16 Working Group. The Chair will disclose this notification via the IEEE 802.16 web site <http://ieee802.org/16/ipr/patents/notices>.

Overview

An informal LDPC group has been working on the goal of achieving consensus on a proposed LDPC code design as an optional advanced code for the OFDMA PHY. Many excellent code designs have been submitted. The codes have been qualitatively and quantitatively characterized, and it is clear that a LDPC code with excellent flexibility and performance, as well as low encoding and decoding complexity, can be defined for 802.16e. This contribution provides additional LDPC specification text.

Recommended Text Changes:

Add the following text to 802.16e_D5, adjusting the numbering as required:

<Add a new paragraph to section 8.4.9.2.5.1 Code Description, the new paragraph to precede the last paragraph of the section "$\mathbf{H}_b$ is partitioned into two sections…">

The permutations used are circular right shifts, and the set of permutation matrices contains the $z \times z$ identity matrix and circular right shifted versions of the identity matrix. Because each permutation matrix is specified by a single circular right shift, the binary base matrix information and permutation replacement information can be combined into a single compact model matrix $\mathbf{H}_{bm}$. The model matrix $\mathbf{H}_{bm}$ is the same size as the binary base matrix $\mathbf{H}_b$, with each binary entry (i,j) of the base matrix $\mathbf{H}_b$ replaced to create the model matrix $\mathbf{H}_{bm}$. Each 0 in $\mathbf{H}_b$ is replaced by a blank or negative value (e.g., by –1) to denote a $z \times z$ all-zero matrix, and each 1 in $\mathbf{H}_b$ is replaced by a circular shift size $p(i,j) \geq 0$. The model matrix $\mathbf{H}_{bm}$ can then be directly expanded to $\mathbf{H}$.

<Add the material below to the end of section 8.4.9.2.5.2 LDPC encoding, directly after the sentence "The LDPC codes are defined such that very low complexity encoding directly from **H** is possible." >

The following informative subsection shows two such methods.

Direct Encoding (Informative)

For the two methods, described below, section $\mathbf{H}_{b2}$ is further partitioned into two sections, where vector $\mathbf{h}_b$ has odd weight, and $\mathbf{H}'_{b2}$ has a dual-diagonal structure with matrix elements at row i, column j equal to 1 for i=j, 1 for i=j+1, and 0 elsewhere:

$$\mathbf{H}_{b2} = \left[\mathbf{h}_b \mid \mathbf{H}'_{b2}\right] = \left[\begin{array}{c|ccccc} h_b(0) & 1 & & & & \\ h_b(1) & 1 & 1 & & \mathbf{0} & \\ . & & 1 & \ddots & & \\ . & & & \ddots & 1 & \\ . & & \mathbf{0} & & 1 & 1 \\ h_b(m_b-1) & & & & & 1 \end{array}\right].$$

The base matrix has $h_b(0)$=1, $h_b(m$-1)=1, and a third value $h_b(j)$, 0<j<(m_b-1) equal to 1. The base matrix structure avoids having multiple weight-1 columns in the expanded matrix.

In particular, the non-zero submatrices are circularly right shifted by a particular circular shift value. Each 1 in $\mathbf{H}'_{b2}$ is assigned a shift size of 0, and is replaced by a $z \times z$ identity matrix when expanding to $\mathbf{H}$. The two 1s located at the top and the bottom of $\mathbf{h}_b$ are assigned equal shift sizes, and the third 1 in the middle of $\mathbf{h}_b$ is given an unpaired shift size. The unpaired shift size is 0.

Method 1

Encoding is the process of determining the parity sequence **p** given an information sequence **s**. To encode, the information block **s** is divided into $k_b = n_b - m_b$ groups of z bits. Let this grouped **s** be denoted **u**,

$$\mathbf{u} = [\mathbf{u}(0) \quad \mathbf{u}(1) \quad \cdots \quad \mathbf{u}(k_b - 1)],$$

where each element of **u** is a column vector as follows

$$\mathbf{u}(i) = [s_{iz} \quad s_{iz+1} \quad \cdots \quad s_{(i+1)z-1}]^T$$

Using the model matrix $\mathbf{H}_{bm}$, the parity sequence **p** is determined in groups of z. Let the grouped parity sequence **p** by denoted **v**,

$$\mathbf{v} = [\mathbf{v}(0) \quad \mathbf{v}(1) \quad \cdots \quad \mathbf{v}(m_b - 1)],$$

where each element of **v** is a column vector as follows

$$\mathbf{v}(i) = [p_{iz} \quad p_{iz+1} \quad \cdots \quad p_{(i+1)z-1}]^T$$

Encoding proceeds in two steps, (a) initialization, which determines $\mathbf{v}(0)$, and (b) recursion, which determines $\mathbf{v}(i+1)$ from $\mathbf{v}(i)$, $0 \le i \le m_b - 2$.
An expression for $\mathbf{v}(0)$ can be derived by summing over the rows of $\mathbf{H}_{bm}$ to obtain

$$\mathbf{P}_{p(x,k_b)}\mathbf{v}(0) = \sum_{j=0}^{k_b-1}\sum_{i=0}^{m_b-1}\mathbf{P}_{p(i,j)}\mathbf{u}(j) \qquad (1)$$

where x , $1 \le x \le m_b - 2$, is the row index of $\mathbf{h}_{bm}$ where the entry is nonnegative and unpaired, and $\mathbf{P}_i$ represents the $z \times z$ identity matrix circularly right shifted by size i. Equation (1) is solved for $\mathbf{v}(0)$ by multiplying by $\mathbf{P}^{-1}_{p(x,k_b)}$, and $\mathbf{P}^{-1}_{p(x,k_b)} = \mathbf{P}_{z-p(x,k_b)}$ since $p(x,k_b)$ represents a circular shift.
Considering the structure of $\mathbf{H}'_{b2}$, the recursion can be derived as follows,

$$\mathbf{v}(1) = \sum_{j=0}^{k_b-1}\mathbf{P}_{p(i,j)}\mathbf{u}(j) + \mathbf{P}_{p(i,k_b)}\mathbf{v}(0), \quad i = 0, \qquad (2)$$

$$\mathbf{v}(i+1) = \mathbf{v}(i) + \sum_{j=0}^{k_b-1}\mathbf{P}_{p(i,j)}\mathbf{u}(j) + \mathbf{P}_{p(i,k_b)}\mathbf{v}(0), \quad i = 1,\ldots,m_b - 2 \qquad (3)$$

where

$$\mathbf{P}_{-1} \equiv \mathbf{0}_{z\times z}.$$

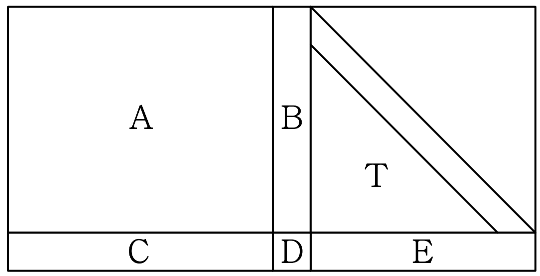
Thus all parity bits not in $\mathbf{v}(0)$ are determined by evaluating Equation (2) for $0 \le i \le m_b - 2$.
Equations (1) and (2) completely describe the encoding algorithm. These equations also have a straightforward interpretation in terms of standard digital logic architectures. Since the non-zero elements $p(i,j)$ of $\mathbf{H}_{bm}$ represent circular shift sizes of a vector, all products of the form $\mathbf{P}_{p(i,j)}\mathbf{u}(j)$ can be implemented by a size-z barrel shifter.

Method 2

For efficient encoding of LDPC, **H** are divided into the form

$$\boldsymbol{H} = \begin{pmatrix} \boldsymbol{A} & \boldsymbol{B} & \boldsymbol{T} \\ \boldsymbol{C} & \boldsymbol{D} & \boldsymbol{E} \end{pmatrix} \qquad (1)$$

where $\boldsymbol{A}$ is $\left(N_p - g\right) \times N_k$, $\boldsymbol{B}$ is $\left(N_p - g\right) \times g$, $\boldsymbol{T}$ is $\left(N_p - g\right) \times \left(N_p - g\right)$, $\boldsymbol{C}$ is $g \times N_k$, $\boldsymbol{D}$ is $g \times g$, and finally, $\boldsymbol{E}$ is $g \times \left(N_p - g\right)$. The basic structure of the **H** matrix is



Further, all these matrices are sparse and $\boldsymbol{T}$ is lower triangular with ones along the diagonal. ***B*** and ***D*** part have the column degree 3 and ***D*** has shift value of a (a is an integer, 0<=a<=z-1). B is with the shift value a of the first entry and shift value 0 in the middle of the column. This other entry is non-zero.

Let $\boldsymbol{v}$=($\boldsymbol{u}$, $\boldsymbol{p}_1$, $\boldsymbol{p}_2$) that $\boldsymbol{u}$ denotes the systematic part, $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ combined denote the parity part, $\boldsymbol{p}_1$ has length g, and $\boldsymbol{p}_2$ has length (N_p-g). The definition equation $\boldsymbol{H} \cdot \boldsymbol{v}^t = 0$ splits into two equations, as in equation 3 and 4 namely

$$\boldsymbol{A}\boldsymbol{u}^T + \boldsymbol{B}\boldsymbol{p}_1^T + \boldsymbol{T}\boldsymbol{p}_2^T = \boldsymbol{0} \qquad (2)$$

and

$$\left(\boldsymbol{-ET}^{-1}\boldsymbol{A} + \boldsymbol{C}\right)\boldsymbol{u}^T + \left(\boldsymbol{-ET}^{-1}\boldsymbol{B} + \boldsymbol{D}\right)\boldsymbol{p}_1^T = \boldsymbol{0} \qquad (3)$$

Define $\phi := \boldsymbol{-ET}^{-1}\boldsymbol{B} + \boldsymbol{D}$ and when we use the parity check matrix as indicated appendix we can get $\boldsymbol{\phi} = \boldsymbol{I}$. Then from (4) we conclude that

$$\boldsymbol{p}_1^T = \left(\boldsymbol{-ET}^{-1}\boldsymbol{A} + \boldsymbol{C}\right)\boldsymbol{u}^T \qquad (5)$$

and

$$\boldsymbol{p}_2^T = \boldsymbol{T}^{-1}\left(\boldsymbol{A}\boldsymbol{u}^T + \boldsymbol{B}\boldsymbol{p}_1^T\right). \qquad (6)$$

As a result, the encoding procedures and the corresponding operations can be summarized below and illustrated in Fig. 1.

Encoding procedure

Step 1) Compute $\boldsymbol{A}\boldsymbol{u}^T$ and $\boldsymbol{C}\boldsymbol{u}^T$.

Step 2) Compute $\boldsymbol{ET}^{-1}\left(\boldsymbol{A}\boldsymbol{u}^T\right)$.

Step 3) Compute $\boldsymbol{p}_1^T$ by $\boldsymbol{p}_1^T = \boldsymbol{ET}^{-1}\left(\boldsymbol{A}\boldsymbol{u}^T\right) + \boldsymbol{C}\boldsymbol{u}^T$.

Step 4) Compute $\boldsymbol{p}_2^T$ by $\boldsymbol{T}\boldsymbol{p}_2^T = \boldsymbol{A}\boldsymbol{u}^T + \boldsymbol{B}\boldsymbol{p}_1^T$.

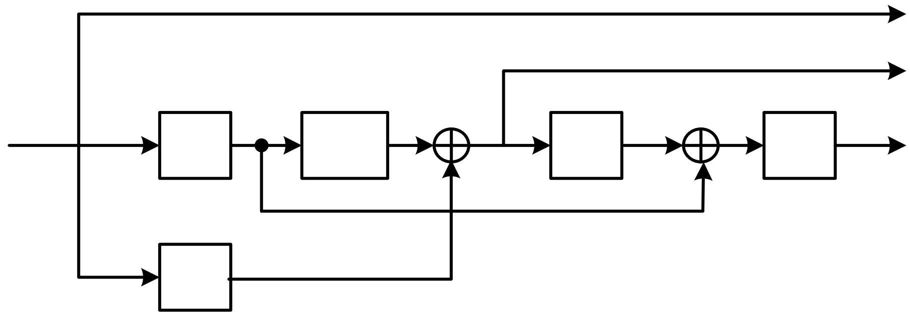


Fig. 2 Block diagram of the encoder architecture for the block LDPC code.

A

u

C