

УДК 681.391

## СНИЖЕНИЕ СЛОЖНОСТИ ДЕКОДИРОВАНИЯ НИЗКОПЛОТНОСТНОГО КОДА

Кравченко А.Н., к.т.н., alexander.kravchenko@thomson.net

**Ключевые слова:** декодирование, низкоплотностное, функция правдоподобия, цифровая связь, эффективность декодера.

**Введение**

Низкоплотностные (LDPC) коды [1] вследствие их превосходной корректирующей способности находят широкое применение в современных цифровых системах связи. Например, коды со средней пропускной способностью используются в стандартах DVB-S2, DVB-T2, DVB-C2, WiMax(IEEE 802.16e) и WLAN(IEEE 802.11n). Более того, LDPC коды с высокой пропускной способностью используются в стандарте mmWPAN (IEEE 802.15.3c). В связи с возрастающей популярностью LDPC кодов требуются специальные методы конструирования этих кодов, а также методы и аппаратура их декодирования.

Как известно, проектирование декодера тесно привязано к структуре кода и его алгоритму декодирования. В связи с этим при проектировании LDPC декодера решаются следующие основные задачи:

1. Обеспечение необходимой эффективности декодирования (ЭД) и заданной пропускной способности декодера.
2. Выбор алгоритма декодирования, обеспечивающего необходимую скорость сходимости декодирования. Как правило, алгоритм декодирования определяет архитектуру декодера и влияет на пропускную способность.
3. Минимизация аппаратных затрат (требования к памяти, вычислительным блокам, организация межблочных связей).

Для разработки оптимального декодера необходимо оптимизировать все параметры, что является сложной задачей. В настоящей работе рассматривается оптимизация двух наиболее важных параметров. Это ЭД и аппаратные затраты. Целью оптимизации является снижение аппаратных затрат без значительных потерь в ЭД.

Наиболее трудоёмким блоком при проектировании

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}$$

Рис. 1 Проверочная матрица регулярного LDPC кода

Рассмотрено проектирование блока вычисления внешних логарифмических отношений функций правдоподобия в LDPC декодере, который существенно влияет на эффективность декодирования и является наиболее трудоёмким при проектировании декодера. Приводятся некоторые приближенные алгоритмы и методы, которые упрощают вычисления внешних логарифмических отношений функций правдоподобия без значительных потерь в эффективности декодирования, а также приводят к значительному снижению аппаратных затрат при практической реализации декодера.

декодера является блок вычисления внешних логарифмических отношений функций правдоподобия (ЛОФП), который связан с вычислением гиперболических функций, требующих при реализации значительных вычислительных затрат. Некоторые упрощения при вычислении этих функций приводят к значительным потерям в ЭД, но при этом получается выигрыш в аппаратной реализации декодера.

В данной работе приводятся некоторые методы и алгоритмы, которые упрощают вычисления внешних ЛОФП без значительных потерь в ЭД и приводят к снижению аппаратных затрат при практической реализации декодера.

**Представление LDPC кода**

LDPC код [1] специфицируется разреженной проверочной  $m \times n$  матрицей  $\mathbf{H}$ , показанной на рис. 1. Если код специфицируется проверочной матрицей, то кодовое слово  $\mathbf{c} = [c_0, c_1, \dots, c_{n-1}]$  должно удовлетворять условию  $\mathbf{c}\mathbf{H}^T = 0$ . Проверочная матрица может быть представлена в виде графа Таннера, изображенного на рис. 2. Граф Таннера является двудольным неориентированным графом, имеющим  $n$  символьных и  $m$  проверочных узлов, причем  $i$ -й символьный узел соединен с  $j$ -м проверочным узлом тогда и только тогда, когда  $\mathbf{H} = [H_{ij}]_{m \times n} = 1$ . Ребра в графе Таннера задают систему связей между символьными и проверочными узлами. Степень узла определяют число ребер выходящих из узла.  $(C, R)$ -LDPC код называется регулярным, если степени  $r$  всех проверочных узлов одинаковы, доля ребер в графе Таннера, соединенных с символьными узлами, в этом случае равна  $r$ .

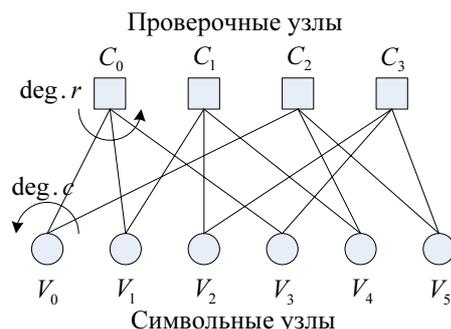
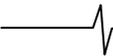


Рис. 2 Граф Таннера двоичного регулярного (2, 3)-LDPC кода



Степени всех символьных узлов  $s$  также одинаковы. Доля ребер в графе Таннера, соединенных с проверочными узлами, в этом случае равна  $s$ .

В нерегулярном LDPC коде  $(C, R)$ -LDPC степени символьных  $deg.c$  и проверочных  $deg.r$  узлов определяются совокупностями  $R$  и  $C$  соответственно. Обычно в литературе [2] для спецификации символьных и проверочных узлов используются полиномы, описывающие распределение степеней узлов.

### Итеративные алгоритмы декодирования

LDPC код может декодироваться различными методами, а именно: мажоритарным методом (majority-logic), методом с переворачиванием бита (bit-flipping), взвешенным методом с переворачиванием бита, а также итеративными вероятностными методами, обладающими наиболее высокой ЭД по сравнению с другими методами. В данной работе рассматриваются только итеративные вероятностные методы.

#### Принцип декодирования по вероятностям: распространение доверия

Галлагер [1] разработал алгоритм декодирования, вычисляющий итеративно распределение вероятностей в граф-ориентированной модели, который известен в литературе под разными именами, например: the sum-product algorithm (SPA), the belief propagation algorithm (BPA), the message passing algorithm (MPA).

Термин "message passing" обычно относится ко всем итеративным алгоритмам, включая SPA и BPA и их аппроксимации. В дальнейшем для удобства обозначим MPA-алгоритм как алгоритм передачи сообщений (АПС), а BPA-алгоритм как алгоритм распространения доверия (АРД).

АПС-алгоритм, основанный на графе Таннера, вычисляет сообщения итеративно. Каждая итерация включает в себя две фазы вычислений. В первой половине итерации (первая фаза) каждый символьный узел обрабатывает его входные сообщения и передает результирующие выходные сообщения (идущие вверх) к смежным проверочным узлам. Этот процесс отображен на рис. 3,а для сообщения  $q_{00}(b)$  от символьного узла  $V_0$  к проверочному узлу  $C_0$  ( $b \in \{0, 1\}$ ). Следует отметить, что информация, проходящая к рассматриваемому проверочному узлу - вся информация, находящаяся в распоряжении символьного узла. Это канальная информация и информация от всех смежных проверочных узлов, исключая информацию от рассматриваемого проверочного узла. Для матрицы  $\mathbf{H}$  это канальная информация  $y_0$  и информация от проверочного узла  $C_2$ , проходящие через символьный узел  $V_0$  к проверочному узлу  $C_0$ . Информация  $q_{ji}(b)$  вычисляется для каждой связанной  $ji$ -пары в каждой фазе.

Во второй половине итерации (вторая фаза) каждый проверочный узел обрабатывает входные сообщения и передает результирующие выходные сообщения (идущие вниз) к смежным символьным узлам. Этот процесс отображен на рис. 3,б для сообщения  $r_{00}(b)$  от проверочного узла  $C_0$  к символьному узлу  $V_0$ . Следует отметить, что информация, проходящая к рассматриваемому символьному узлу, - это вся информация, находящаяся в распоряжении проверочного узла. Указанная

информация поступает от всех смежных символьных узлов, исключая информацию от рассматриваемого символьного узла. В случае матрицы  $\mathbf{H}$  - это информация от символьных узлов  $V_1, V_3$ , проходящая через проверочный узел  $C_0$  к символьному узлу  $V_0$ . Такая внешняя информация  $r_{ij}(b)$  вычисляется для каждой связанной  $ij$ -пары в каждой фазе.

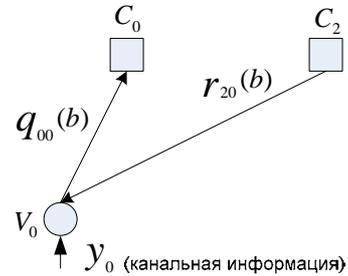


Рис. 3,а. Подграф графа Таннера, соответствующего матрице  $\mathbf{H}$ , для вычисления сообщения от символьного узла  $V_0$  к проверочному узлу  $C_0$ .

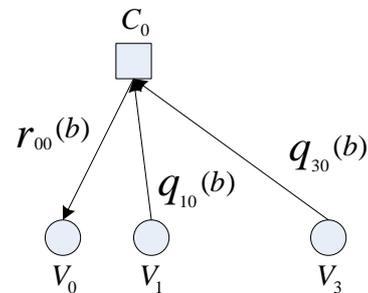


Рис. 3,б. Подграф графа Таннера, соответствующего матрице  $\mathbf{H}$ , для вычисления сообщения от проверочного узла  $C_0$  к символьному узлу  $V_0$ .

После назначенного числа итераций или после некоторого критерия остановки алгоритма, декодер определяет жесткое решение бит  $\hat{c}_i$  кодового слова на основе вычисленных сообщений (апостериорных вероятностей).

#### АРД декодирующий алгоритм, основывающийся на апостериорной вероятности

Описание алгоритма производится в соответствии с работой [3]. При описании алгоритма вводятся следующие обозначения:

- 1)  $N(i) = \{i : H_{ij} = 1\}$  - группа номеров символьных узлов, которые участвуют в проверочном уравнении  $i$ .
- 2)  $N(i) \setminus j = \{j' : H_{ij'} = 1, j' \neq j\}$  - группа номеров символьных узлов, исключая номер  $j$ .
- 3)  $M(j) = \{j : H_{ij} = 1\}$  - группа номеров проверочных узлов, соединенных с символьным узлом  $V_j$ .
- 4)  $M(j) \setminus i = \{i' : H_{i'j} = 1, i' \neq i\}$  - группа номеров проверочных узлов, соединенных с символьным узлом  $V_j$ , исключая номер  $i$ .
- 5)  $r_{ij}(b)$  - есть вероятность того, что проверочное уравнение  $i$  удовлетворяется, когда  $c_i = b$  ( $b \in \{0, 1\}$ ) и остальные символы имеют распределение вероятностей, заданных величинами  $q_{ij}$ ,  $j \in N(i) \setminus j$ .
- 6)  $q_{ij}(b)$  - есть вероятность того, что  $c_i = b$ , вычисленной на основе информации, полученной от проверочных узлов  $M(j)$ , исключая номер  $i$ .

Подобно процессу декодирования турбокодов, где вычисляются апостериорные вероятности (АВ) для всех информационных бит, при декодировании LDPC кода используется принцип максимума АВ для определения вероятности того, что данный бит в переданном кодовом слове  $\mathbf{c}=[c_0, c_1, \dots, c_{n-1}]$  равен 1, вычисленная при условии, что последовательность символов  $\mathbf{y}=[y_0, y_1, \dots, y_{n-1}]$  была принята. АВ вычисляется как

$$P_i = \Pr(c_i = 1 | y) \quad (1)$$

Галлагер в его работе [1] ввел следующие определения:

1) Вероятность того, что последовательность двоичных чисел  $\mathbf{a}=(a_0, a_1, \dots, a_m)$  содержит четное число единиц равна

$$1/2 + 1/2 \prod_{k=1}^m (1 - 2p_k), \quad (2)$$

где  $p_k = \Pr(a_k)$ .

2) Вероятность того, что последовательность двоичных чисел  $\mathbf{a}$  содержит нечетное число единиц равна

$$1/2 - 1/2 \prod_{k=1}^m (1 - 2p_k), \quad (3)$$

Учитывая определение (2), вместе с соотношением  $p_k \leftrightarrow q_{ji}(1)$  можно записать

$$r_{ij}(0) = \frac{1}{2} + \frac{1}{2} \prod_{j \in N(i) \setminus j} (1 - 2q_{ji}(1)), \quad (4)$$

$r_{ij}(0)$  определяет вероятность того, что проверочное уравнение  $i$  удовлетворено, если  $c_j = 0$  и вероятность того, что другие биты кодового слова  $\{c_j : j \in N(i) \setminus j\}$  содержат четное число единиц, равна 1.

Очевидно, что  $r_{ij}(1) = 1 - r_{ij}(0)$ . (5)

Декодирующий алгоритм включает следующие основные шаги.

1. Инициализация: АД алгоритм (для вычисления АВ величин) инициализируется при следующих начальных значениях:  $q_{ji}(b) = \Pr(c_j = b | y_j)$  для всех  $j$  и  $i$ , при условии что  $H_{ij} = 1$ , где  $y_j$  - канальная информация. В случае аддитивного белого гауссова шума (АБГШ) инициализировать априорные вероятности символьных узлов, равными:

$$q_{ji}(0) = 1 - P_j = \Pr(c_j = 0 | y_j) = \frac{1}{1 + e^{-2y_j/\delta}} \quad (6)$$

$$q_{ji}(1) = P_j = \Pr(c_j = 1 | y_j) = \frac{1}{1 + e^{2y_j/\delta}} \quad (7)$$

2. Для всех  $j$  и  $i$  вычислить  $r_{ij}(0)$  и  $r_{ij}(1)$  в соответствии с выражениями (4) и (5).

3. Для всех  $j$  и  $i$  вычислить  $q_{ij}(0)$  и  $q_{ij}(1)$  в соответствии с выражениями

$$q_{ji}(0) = (1 - P_j) \prod_{i \in M(j) \setminus i} r_{ij}(0),$$

$$q_{ji}(1) = P_j \prod_{i \in M(j) \setminus i} r_{ij}(1) \quad (8)$$

и нормировать с множителем  $k_{ji} = 1/(q_{ji}(0) + q_{ji}(1))$ ,

$$q_{ji}(0) = k_{ji}(1 - P_j) \prod_{i \in M(j) \setminus i} r_{ij}(0),$$

$$q_{ji}(1) = k_{ji}P_j \prod_{i \in M(j) \setminus i} r_{ij}(1) \quad (9)$$

4. Для всех  $j$  вычислить апостериорные вероятности

$$Q_j(0) = (1 - P_j) \prod_{i \in M(j)} r_{ij}(0), \quad Q_j(1) = P_j \prod_{i \in M(j)} r_{ij}(1) \quad (10)$$

и нормировать множителем  $K_j = 1/(Q_j(0) + Q_j(1))$ ,

$$Q_j(0) = K_j(1 - P_j) \prod_{i \in M(j)} r_{ij}(0), \quad Q_j(1) = K_jP_j \prod_{i \in M(j)} r_{ij}(1) \quad (11)$$

5. Проверить кодовое слово  $\hat{\mathbf{c}} = [\hat{c}_0, \dots, \hat{c}_{n-1}]$ , где  $\hat{c}_j = 1$ , если  $Q(1)_j > Q(0)_j$ , иначе  $\hat{c}_j = 0$ , на предмет принадлежности коду, т.е.  $\hat{\mathbf{c}}\mathbf{H}^T = 0$ . При выполнении этого условия производится остановка алгоритма. Если это условие не выполняется, то переход к шагу 2, до тех пор, пока не будет превышено некоторое предельное число итераций.

**АРД декодирующий алгоритм, основывающийся на логарифмическом отношении функций правдоподобия (ЛОФП-АРД)**

Определим следующие значения ЛОФП [4]:

$$R_{ij} = \log(r_{ij}(0)/r_{ij}(1)), \quad Q_{ji} = \log(q_{ji}(0)/q_{ji}(1))$$

Допустим, что есть случайная двоичная величина  $x \in \{0, 1\}$ , тогда ЛОФП величины  $x$  равно

$$L(x) = \log \frac{P(x=0)}{P(x=1)} = \log \frac{p_0}{p_1}, \quad (12)$$

где

$$p_0 = \frac{e^{L(x)}}{1 + e^{L(x)}} \text{ и } p_1 = \frac{1}{1 + e^{L(x)}}$$

Учитывая определения (2), (3), вместе с соотношением  $p_k \leftrightarrow Q_{ji}$ , можно записать

$$R_{ij} = \log \left( \frac{1 + \prod_{j \in N(i) \setminus j} (1 - 2Q_{ji})}{1 - \prod_{j \in N(i) \setminus j} (1 - 2Q_{ji})} \right) \quad (13)$$

Используя факт, что

$$\tanh \left[ \frac{1}{2} L(x) \right] = p_0 - p_1 = 1 - 2p_1, \quad (13) \text{ можно записать как}$$

$$R_{ij} = \log \left( \frac{1 + \prod_{j \in N(i) \setminus j} \tanh(Q_{ji}/2)}{1 - \prod_{j \in N(i) \setminus j} \tanh(Q_{ji}/2)} \right) \quad (14)$$

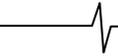
Используя далее, что

$$\log \frac{1+x}{1-x} = 2 \tanh^{-1}, \quad (14) \text{ можно записать как}$$

$$R_{ij} = 2 \tanh^{-1} \left( \prod_{j \in N(i) \setminus j} \tanh(Q_{ji}/2) \right) \quad (15)$$

Декодирующий алгоритм включает следующие основные шаги.

1. Инициализация: для всех  $j = 0, 1, \dots, n-1$  инициализировать символьные узлы равными логарифми-



ческому отношению априорных вероятностей  $Q_{ji}$

$$Q_{ji} = l_j = 2y_j / \sigma^2,$$

где  $\sigma^2$  - дисперсия АБГШ.

2. Для всех  $i$  и  $j$  вычислить  $R_{ij}$  в соответствии с выражением (15).

3. Для всех  $j$  и  $i$  вычислить  $Q_{ji}$  в соответствии с выражением

$$Q_{ji} = l_j + \sum_{i \in M(j) \setminus i} R_{ij} \quad (17)$$

4. Вычислить логарифмические отношения апостериорных вероятностей в соответствии с выражением

$$P_j = l_j + \sum_{i \in M(j)} R_{ij} \quad (18)$$

5. Проверить кодовое слово  $\hat{c} = [\hat{c}_0, \dots, \hat{c}_{n-1}]$ , где  $\hat{c}_j = 1$ , если  $P_j \leq 0$ , иначе  $\hat{c}_j = 0$ .

Если  $\hat{c}H^T = 0$  или число итераций равно максимальному лимиту, то производится остановка алгоритма, иначе переход к шагу 2.

АРД, а также ЛОФП-АРД декодирующие алгоритмы требуют выполнения операций умножения, которые значительно сложнее при аппаратной реализации, чем операции сложения. Кроме того, операции умножения вносят большую погрешность в вычисления, чем операции сложения при использовании арифметики с фиксированной запятой. Исходя из этого, операции сложения более предпочтительны при вычислениях, чем операции умножения. В дальнейшем будут рассматриваться алгоритмы декодирования, использующие только операции сложения, а также упрощенные методы вычисления правой части уравнения (15).

#### Упрощенные алгоритмы декодирования, основывающиеся на ЛОФП

1. ЛОФП-АРД1 декодирующий алгоритм, основанный на исключении операции умножения комплексной функции  $\tanh$  (параллельная схема).

В правой части уравнения (15) содержится операция умножения комплексной функции  $\tanh$ . Для исключения этого в работе [1] используется следующее.

Фактор  $Q_{ji} = \alpha_{ji} \beta_{ji}$  записывается как произведение знака и величины (абсолютного значения):

$$\alpha_{ji} = \text{sign}(Q_{ji}), \quad \beta_{ji} = |Q_{ji}|.$$

С учетом этих обозначений, можно записать

$$\prod_{j \in N(i) \setminus j} \tanh(Q_{ji}/2) = \prod_{j \in N(i) \setminus j} \alpha_{ji} \cdot \prod_{j \in N(i) \setminus j} \tanh(\beta_{ji}/2).$$

После некоторых преобразований получается следующее равенство:

$$\begin{aligned} R_{ij} &= \prod_j \alpha_{ji} \cdot 2 \tanh^{-1} \left( \prod_{j \in N(i) \setminus j} \tanh(\beta_{ji}/2) \right) \\ &= \prod_j \alpha_{ji} \cdot 2 \tanh^{-1} \log^{-1} \log \left( \prod_{j \in N(i) \setminus j} \tanh(\beta_{ji}/2) \right) \\ &= \prod_j \alpha_{ji} \cdot 2 \tanh^{-1} \log^{-1} \sum_j \log(\tanh(\beta_{ji}/2)) \\ &= \prod_{j \in N(i) \setminus j} \alpha_{ji} \cdot \phi \left( \sum_{j \in N(i) \setminus j} \phi(\beta_{ji}) \right), \end{aligned} \quad (19)$$

$$\text{где } \phi(x) = -\log[\tanh(x/2)] = \log \left( \frac{e^x + 1}{e^x - 1} \right) \quad (20)$$

с учетом того, что  $\phi^{-1}(x) = \phi(x)$ , когда  $x > 0$ .

В практических применениях  $\phi(x)$  функция реализуется в виде поисковой таблицы.

2. ЛОФП-АРД2 декодирующий алгоритм, основанный на исключении операции умножения комплексной функции  $\tanh$  (последовательная схема).

Данный метод позволяет уменьшить аппаратные затраты при конструировании блока вычисления внешних значений ЛОФП по сравнению с выше описанным, за счет последовательного вычисления ЛОФП.

Рассматривается частный проверочный узел  $i$ , связанный с символьными узлами  $N(i) = (j_{q1}, j_2, \dots, j_{\text{deg},r})$  и соответствующими входными сообщениями, равными  $Q_{ij_{q1}}, Q_{ij_{q2}}, \dots, Q_{ij_{\text{deg},r}}$ . Выходные сообщения  $R_{ij}$  могут вычисляться с использованием алгоритма „вперед-назад“ [5]. Определяются вспомогательные случайные двоичные переменные  $f_1 = x_{j_{q1}}, f_2 = f_1 \oplus x_{j_{q2}}, \dots, f_{\text{deg},r} = f_{\text{deg},r-1} \oplus x_{j_{\text{deg},r}}$  и  $b_{\text{deg},r} = x_{j_{\text{deg},r}}, b_{\text{deg},r-1} = b_{\text{deg},r} \oplus x_{j_{\text{deg},r-1}}, \dots, b_1 = b_2 \oplus x_{j_{q1}}$ , где  $\oplus$  обозначает двоичную XOR операцию.

Значения ЛОФП двоичной суммы определяемой выражением

$$L(A_2) = L(x_1 \oplus x_2) = \log \left( \frac{1 + e^{L(x_1)} e^{L(x_2)}}{e^{L(x_1)} + e^{L(x_2)}} \right), \quad (21)$$

которое может быть преобразовано к виду [5]:

$$L(x_1 \oplus x_2) = \text{sign}(L(x_1)) \text{sign}(L(x_2)) \min[|L(x_1)|, |L(x_2)|] + \log(1 + e^{|x_1+x_2|}) - \log(1 + e^{-|x_1-x_2|}) \quad (22)$$

Используя (22) повторно, можно получить  $L(f_1), L(f_2), \dots, L(f_{\text{deg},r})$  и  $L(b_1), L(b_2), \dots, L(b_{\text{deg},r})$  рекурсивным способом, основанном на знании  $Q_{ij_{q1}}, Q_{ij_{q2}}, \dots, Q_{ij_{\text{deg},r}}$ . При использовании проверочного уравнения  $p_i = x_{j_{q1}} \oplus x_{j_{q2}} \oplus \dots \oplus x_{j_{\text{deg},r}}$  проверочного узла  $i$  получается, что  $x_j = f_{j-1} \oplus b_{j+1}$  для каждого  $j \in j_{q2}, j_{q3}, \dots, j_{\text{deg},r-1}$ . Поэтому  $R_{ij}$  сообщения, выходящие из проверочного узла, могут просто выразиться как

$$R_{ij} = L(f_{j-1} \oplus b_{j+1}), \quad j = j_{q2}, j_{q3}, \dots, j_{\text{deg},r-1},$$

$$R_{ij_{q1}} = L(b_2),$$

$$R_{ij_{\text{deg},r}} = L(f_{\text{deg},r-1}). \quad (23)$$

Полное вычисление включает прямое рекурсивное вычисление  $L(f_j)$ , обратное рекурсивное вычисление  $L(b_j)$  и окончательное попарное вычисление  $L(f_{j-1} \oplus b_{j+1})$ , которое требует  $3(\text{deg},r-2)$  ключевых операций (22) на один проверочный узел.

Вычисление функции  $g(y) = \log(1 + e^{-|y|})$ , входящей в выражение (22), может реализовано в виде поисковой таблицы. В таблице 1 показано грубое квантование функции  $g(y)$ . Максимальная ошибка аппроксимации не больше чем 0.05. Функция  $g(y)$  также может быть аппроксимирована кусочно-линейной функцией. Результат аппроксимации показан в таблице 2.

Таблица 1.

$ y $	$\log(1 + e^{- y })$	$ y $	$\log(1 + e^{- y })$
(0, 0.196)	0.65	(1.05, 1.508)	0.25
(0.0196, 0.433)	0.55	(1.508, 2.252)	0.15
(0.433, 0.71)	0.45	(2.252, 4.5)	0.05
(0.71, 1.05)	0.35	(4.5, +∞)	0.0

Таблица 2.

$ y $	$\log(1 + e^{- y })$	$ y $	$\log(1 + e^{- y })$
(0, 0.5)	$- y  * 2^{-1} + 0.7$	(2.2, 3.2)	$- y  * 2^{-1} + 0.7$
(0.5, 1.6)	$- y  * 2^{-2} + 0.575$	(3.2, 4.4)	$- y  * 2^{-1} + 0.7$
(1.6, 2.2)	$- y  * 2^{-3} + 0.375$	(4.4, +∞)	$- y  * 2^{-1} + 0.7$

3. Алгоритм декодирования, основанный на минимальной сумме (АМС)

При исследовании функции (20) можно заметить, что результат вычислений существенно зависит от маленьких значений аргумента и, следовательно, при вычислениях можно обойтись небольшим числом этих значений. Эта особенность лежит в основе АМС алгоритма.

Применение АМС алгоритма декодирования приводит потенциально к уменьшению сложности декодера и соответственно площади кристалла. Сложность декодера уменьшается вследствие исключения функции  $\phi(x) = -\log[\tanh(x/2)]$ , которая реализуется при помощи поисковых таблиц, а также вследствие уменьшения размера памяти, необходимого для запоминания и считывания (символьный узел)/(проверочный узел сообщений). Недостатком алгоритма является включение устройства сортировки, которое ищет сообщение с минимальным абсолютным значением и сортирует оставшиеся сообщения по мере возрастания их абсолютных величин. Этот недостаток приводит к уменьшению пропускной способности декодера. Главная разница между ЛОФП-АРД и АМС алгоритмами лежит в вычислении  $R_{ij}$  значений. Вместо вычислений (19) используется его приближение

$$\phi\left(\sum_j \phi(\beta_{ji})\right) \cong \phi\left(\phi\left(\min_j\right)\right) = \min_{j \in N(i) \setminus j} \beta_{ji}$$

или с учетом знака

$$R_{ij} \cong \prod_{j \in N(i) \setminus j} \alpha_{ji} \cdot \min_{j \in N(i) \setminus j} \beta_{ji} \quad (24)$$

В работе [6], на основе свойств алгебры логарифмических функций правдоподобия, получено приближенное выражение для вычисления  $R_{ij}$ . Показано, что для двух независимых случайных величин  $x_1$  и  $x_2$  с ЛОФП, равными

$$L(x_1) = \log \frac{P(x_1 = 0)}{P(x_1 = 1)} = \log \frac{p_{10}}{p_{11}} \quad \text{и}$$

$$L(x_2) = \log \frac{P(x_2 = 0)}{P(x_2 = 1)} = \log \frac{p_{20}}{p_{21}},$$

значение их суммы равно

$$L(x_1 \oplus x_2) = \log \left( \frac{1 + e^{L(x_1)} e^{L(x_2)}}{e^{L(x_1)} + e^{L(x_2)}} \right); \quad (25)$$

$$L(x_1 \oplus x_2) \approx \text{sign}(L(x_1)) \cdot \text{sign}(L(x_2)) \cdot \min(|L(x_1)|, |L(x_2)|). \quad (26)$$

В работе [6] введен символ  $\boxplus$  для обозначения сложения, определенного как

$$L(x_1) \boxplus L(x_2) = L(x_1 \oplus x_2), \quad (27)$$

Для множества  $x$ , размером  $J$ , сумма вычисляется как

$$\sum_{j=1}^J \boxplus L(x_j) = L\left(\sum_{j=1}^J \otimes x_j\right) = \log \left( \frac{\prod_{j=1}^J (e^{L(x_j)} + 1) + \prod_{j=1}^J (e^{L(x_j)} - 1)}{\prod_{j=1}^J (e^{L(x_j)} + 1) - \prod_{j=1}^J (e^{L(x_j)} - 1)} \right). \quad (28)$$

Используя соотношение  $\tanh(x/2) = (e^x - 1)/(e^x + 1)$ , получено выражение

$$\sum_{j=1}^J \boxplus L(x_j) = \log \left( \frac{1 + \prod_{j=1}^J \tanh(L(x_j)/2)}{1 - \prod_{j=1}^J \tanh(L(x_j)/2)} \right) = 2 \tanh^{-1} \left( \prod_{j=1}^J \tanh(L(x_j)/2) \right). \quad (29)$$

С учетом аппроксимации (26) выражение (29) приобретает вид

$$\sum_{j=1}^J \boxplus L(x_j) = L\left(\sum_{j=1}^J \otimes x_j\right) \approx \left(\prod_{j=1}^J \text{sign}(L(x_j))\right) \cdot \min_{j=1..J} |L(x_j)|. \quad (30)$$

Как видно из выражения (30), надежность суммы определяется в основном наименьшей надежностью из всех составляющих, входящих в сумму.

На основании полученного результата (30), выражение (15) в ЛОФП-АРД алгоритме декодирования можно заменить упрощенным выражением

$$R_{ij} = L\left(\sum_{j \in N(i) \setminus j} Q_{ji}\right) = \prod_{j \in N(i) \setminus j} \text{sign}(Q_{ji}) \cdot \min_{j \in N(i) \setminus j} |Q_{ji}|. \quad (31)$$

При практической реализации АМС алгоритма сложность блока вычисления внешних ЛОФП определяется в основном структурой блока сортировки, который сортирует внутренние ЛОФП значения в порядке увеличения их абсолютных значений, а также логикой обработки знаков. Сложность сортировки определяется числом уровней компараторов, которое пропорционально  $\log_2(\text{deg}.r)$ . В различных модификациях АМС алгоритма в основном используются первый и второй минимум, а также индекс первого минимума.

Известно, что АМС алгоритм декодирования, использующий упрощенное выражение (31) для вычисле-

ния  $R_{ij}$ , приводит к потерям в ЭД. Потери в ЭД регулярных LDPC кодов достигают 1 дБ для кодов со степенью проверочных узлов больше 6 [7].

В дальнейшем будут рассматриваться методы, повышающие эффективность АМС алгоритма декодирования.

4. АМС алгоритм декодирования с корректирующим фактором.

1) Нормализованный АМС алгоритм декодирования.

Эффективность декодирования может быть улучшена за счет введения масштабирующего фактора  $\alpha$  (большее единицы) в выражение (31) [7,8]:

$$R_{ij} = \prod_{j \in N(i) \setminus j} \text{sign}(Q_{ji}) \cdot \frac{\min_{j' \in N(i) \setminus j} |Q_{ji'}|}{\alpha} \quad (32)$$

Фактор  $\alpha$  зависит от отношения сигнал-шум и вычисляется следующим образом [8].

$$\alpha = E(|L_2|) / E(|L_1|) \quad (33)$$

$E(|L_2|)$  и  $E(|L_1|)$  вычисляются в соответствии с выражениями:

$$E(|L_1|) = E\left(2 \tanh^{-1}\left(\prod_{j=1}^W \tanh(|X_j|/2)\right)\right) \quad (34)$$

$$E(|L_2|) = E(\min(|X_1|, |X_2|, \dots, |X_W|)) \quad (35)$$

где  $X_1, X_2, \dots, X_W$  - независимые идентично распределенные случайные переменные:  $\{X_j : j = 1, 2, \dots, W\} = \{R_{ij} : j' \in N(i) \setminus j\}$  и  $W = \text{deg}.r - 1$ , где  $\text{deg}.r$  определяет степень проверочного узла. Функция плотности вероятности переменных  $X_j$  зависит от отношения сигнал-шум.

2) АМС алгоритм декодирования с корректирующей константой.

Эффективность декодирования может быть улучшена за счет введения корректирующей константы [8]:

$$R_{ij} = \prod_{j' \in N(i) \setminus j} \text{sign}(Q_{ji'}) \cdot \max\left\{\min_{j' \in N(i) \setminus j} |Q_{ji'}| - \beta, 0\right\} \quad (36)$$

Константа  $\beta$  подбирается с учетом достижения максимальной ЭД при заданном отношении сигнал-шум и кодовой скорости применяемого кода. Отмечается, что АМС алгоритм, нормализованный АМС алгоритм и АМС алгоритм декодирования с корректирующей константой не требуют знания характеристик шума ( $\sigma^2$ ) и работают непосредственно с величинами, поступающими на вход блока.

Оценки сложности блоков вычисления внешних значений ЛОФП для всех рассмотренных выше алгоритмов, представлены в таблице 3. Эффективность различных методов декодирования иллюстрирует рис. 4.

Таблица 3

Алгоритм	Умножения	Сложения	Компараторы	Специальные Операции
ЛОФП-АРД, прямой метод, (15)	$\text{deg}.r-1$			$\tanh(x)$ $\tanh^{-1}(x)$
ЛОФП-АРД1, таблич. метод, (19)		$2\text{deg}.r+1$		$2\text{deg}.r$ поисковых таблиц, логика обработки знака
ЛОФП-АРД2, таблич. метод, (23) Ключевая операция (22) Число ключевых операций		5	1	2 поисковые таблицы  $3(\text{deg}.r-2)$
АМС, (31)			$\log_2(\text{deg}.r)$	сортировка, поиск минимума логика обработки знака
Нормализованный АМС, (32)	1		$\log_2(\text{deg}.r)$	сортировка, поиск минимума логика обработки знака
АМС с коррект. константой (36)		1	$\log_2(\text{deg}.r) + 1$	сортировка, поиск минимума логика обработки знака

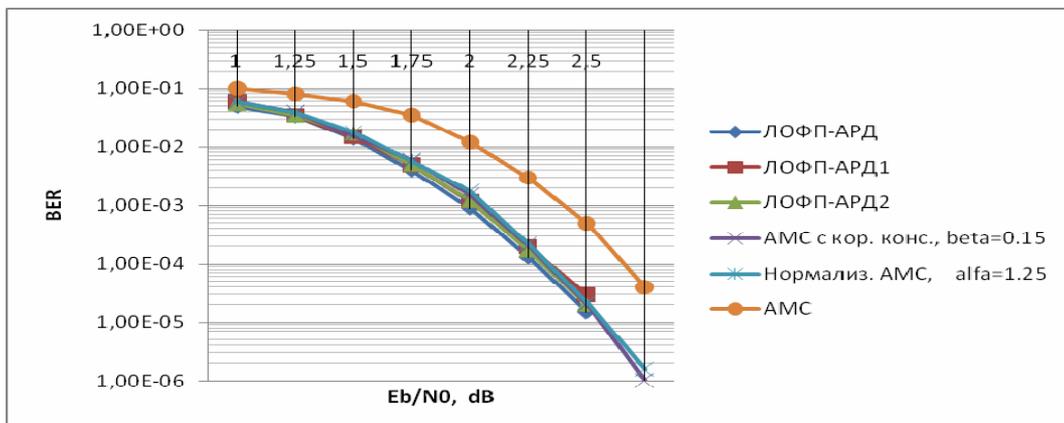


Рис. 4. Эффективности декодирования различных алгоритмов декодирования (1008, 504) регулярного LDPC кода с  $(\text{deg}.c, \text{deg}.r) = (3, 6)$ .

Примеры аппаратурной реализации LDPC декодеров, использующих приближенные методы вычисления внешних ЛОФП, можно найти в [9-12].

В связи с ограниченным объемом публикации в данной работе не рассмотрены методы оптимизации корректирующих факторов, нацеленных на поиск максимальной эффективности декодирования LDPC декодера, такие как "Density Evolution" [13] или "EXIT chart" [14] методы.

#### Литература

1. R. Gallager. Low-density parity-check codes. IRE Trans. Information Theory, pp. 21-28, Jan. 1962.
2. T. Richardson, A. Shokrollahi, and R. Urbanke. Design of capacity-approaching irregular low-density parity-check codes. IEEE Trans. On Information Theory, vol. 47, pp. 619-637, Feb. 2001.
3. William E. Ryan. An Introduction to LDPC Codes. The University of Arizona, Box 210104, Tucson, AZ 85721, Aug. 2003.
4. Бернад Складар. Цифровая связь. Теоретические основы и практическое применение. М.: Издательский дом "Вильямс", 2003, 1104 с.
5. X-Y Hu, E. Eleftherior, D-M. Arnold, and A. Dholaki. Efficient implementation of the Sum-product algorithm for decoding LDPC codes, Proc. 2001 IEEE Globe Com Conf., pp. 1036-1036E, Nov. 2001.
6. J. Hagenauer, E. Offer, and L. Papke. Iterative Decoding of Binary Block and Convolutional Codes. IEEE Trans. On Information Theory, vol. 42, pp. 429-445, March 1996.
7. J. Chen, M. P. C. Fossorier. Near Optimum Universal Belief Propagation Based Decoding of Low-Density Parity Check Codes. IEEE Trans. On Communications, vol. 50, pp. 406-414, March 2002.
8. J. Chen, A. Dholakia, E. Eleftherior, M. P. C. Fossorier, and Xiao-Y. Hu. Reduced-Complexity Decoding of LDPC Codes. IEEE Trans. On Communications, vol. 53, pp. 1288-1298, August 2005.
9. Z. Wangy, Qin. Jia, Low Complexity, High Speed Decoder

- Architecture for Quasi-Cyclic LDPC Codes, in Proc. ISCAS, 2005, pp. 5586-5789.
10. J. Sha, M. Gao, Z. Zhang, L. Li, and Z. Wang. A Memory Efficient FPGA Implementation of Quasi-Cyclic LDPC Decoder. Proceedings of the 5<sup>th</sup> WSEAS Int. Con. On Instrumentation, Measurement, Circuits and System, Hangzhou, China, pp. 218-223, 16-18 Apr. 2006.
  11. J. Su, K. Liu, and H.Min. Hardware Efficient Decoding of LDPC Codes Using Partial-min Algorithms. IEEE Transaction on Consumer Electronics, vol. 52, pp. 1463-1468, Nov. 2006.
  12. E. Boutillon, J. Castura and F. R. Kschischang. Decoder-First Code Design. In Proceeding of the 2<sup>nd</sup> International Symposium in Turbo Codes and Related Topics, pp. 459-462, 2000.
  13. T. J. Richardson and R. L. Urbanke. The Capacity of LDPC Codes Under Message-Passing Decoding. IEEE Trans. On Information Theory, vol. 47, pp. 599-618, Feb. 2001.
  14. St. ten Brink. Convergence Behavior of Iterative Decoded Parallel Cocatenated Codes. IEEE Trans. On Communications, Vol. 49, Oct. 2001.

## REDUCED-COMPLEXITY DECODING LDPC CODE

*Kravtchenko A.N.*

Performance and complexity of LDPC decoder depends on the decoding algorithm, in particular on the check-node update unit which calculates extrinsic information in decoder.

Approximate representations of the decoding computations are shown to achieve a reduction in complexity by simplifying the check-node update unit, without loss of performance. In particular, several approaches for simplified check-node updates are presented that are based on the so-called min-sum approximation.

### ИНФОРМАЦИОННОЕ ПИСЬМО

*Уважаемый коллега!*

*Приглашаем Вас принять участие в работе 13-ой Международной конференции «Цифровая обработка сигналов и ее применение – DSPA'2011», которая состоится в Москве в ИИУ РАН марте-апреле 2011 года*

#### ОРГАНИЗАТОРЫ:

- Российское научно-техническое общество радиотехники, электроники и связи им. А.С. Попова
- IEEE Signal Processing Society
- Российская секция IEEE
- Институт радиотехники и электроники РАН
- Институт проблем управления РАН
- Институт проблем передачи информации РАН
- Московский научно-исследовательский телевизионный институт
- Компания AUTEK Ltd. (АВТЭКС)

#### ОСНОВНЫЕ НАПРАВЛЕНИЯ РАБОТЫ КОНФЕРЕНЦИИ:

- Теория сигналов и систем
- Теория и методы ЦОС
- Цифровая обработка многомерных сигналов
- Цифровая обработка речевых и звуковых сигналов
- Цифровая обработка изображений
- ЦОС в системах телекоммуникаций
- ЦОС в радиотехнических системах
- ЦОС в системах управления и робототехники
- Цифровая обработка измерительной информации
- Нейрокомпьютерная обработка сигналов и изображений
- Цифровое телевидение и телевидение
- Цифровое радиовещание
- ЦОС в системах защиты информации
- Проектирование и техническая реализация систем ЦОС
- ЦОС в открытых системах
- Проблемы подготовки специалистов в области ЦОС