

УДК 681.391

## МЕТОДЫ И АППАРАТУРА КОДИРОВАНИЯ И ДЕКОДИРОВАНИЯ СИСТЕМАТИЧЕСКОГО НЕРЕГУЛЯРНОГО КОДА ПОВТОРЕНИЯ-НАКОПЛЕНИЯ (IRA) ДЛЯ DVB-S2 И DVB-T2 ДЕМОДУЛЯТОРОВ

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

**Ключевые слова:** кодирование, декодирование, низкоплотностной код, демодулятор, метод, алгоритм, граф, матрица.

### Основные понятия

LDPC коды (низкоплотностные коды) были предложены Галлагером в 1962 году. Отличительной особенностью LDPC кода является существование случайной разреженной проверочной  $m \times n$  матрицы  $H$ , т.е. такой матрицы, у которой число ненулевых элементов пренебрежимо мало по сравнению с ее размерами.

Проверочная матрица может быть представлена в виде графа Таннера, изображенного на рис. 1. Граф Таннера является двудольным неориентированным графом, имеющим  $n$  символьных и  $m$  проверочных узлов, причем  $i$ -ый символьный узел соединен с  $j$ -ым проверочным узлом тогда и только тогда, когда  $H = [h_{i,j}]_{m \times n} = 1$ . 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 ( $r$ ). Степени всех символьных узлов ( $c$ ) также одинаковы, и доля ребер в графе Таннера, соединенных с проверочными узлами, в этом случае равна ( $c$ ).

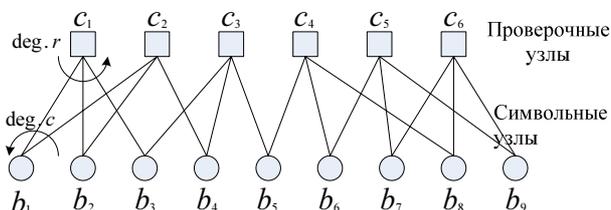


Рис. 1. Граф Таннера двоичного регулярного (2, 3)-LDPC кода

Минимальная кодовая скорость определяется как  $R = 1 - m/n$ . Число ребер в соответствующем двудольном графе - вес Хэмминга  $w(H)$  проверочной матрицы  $H$   $w(H) = nc = mr$ , и плотность определяется, как  $d(H) = w(H)/mn = c/m = r/n$ . Минимальное кодовое расстояние этих кодов увеличивается линейно с увеличением длины кодового слова  $n$  для данной кодовой скорости  $R$  и степеней символьных ( $\text{deg.}c$ ) и про-

Рассмотрена методика конструирования проверочных матриц систематического нерегулярного кода повторения-накопления (IRA) для DVB-S2 и DVB-T2 демодуляторов на основе таблиц (Parity Bit Tables) стандарта, специфицирующих IRA коды с различными кодовыми скоростями. Приводится методика конструирования структурированного ("architecture-aware") LDPC кода из IRA кода. Показано, что полученная конструкция LDPC кода является циклической. Описана архитектура декодера, реализующего "Novel" алгоритм декодирования, который имеет высокую сходимость декодирования и, следовательно, более высокую пропускную способность, а также уменьшенные требования к блоку памяти. Представлены результаты моделирования эффективности декодера LDPC кода длиной 16200 при различных кодовых скоростях в канале с АБГШ для случая двоичной кодовой модуляции.

верочных узлов ( $\text{deg.}r$ ). В нерегулярном LDPC коде (C, R)-LDPC степени символьных ( $\text{deg.}c$ ) и проверочных узлов ( $\text{deg.}r$ ) определяются совокупностями  $R$  и  $C$ , соответственно. Обычно регулярные коды легче кодируются и имеют простые архитектуры декодеров чем нерегулярные, однако последние достигают более высокого выигрыша от кодирования.

### LDPC код для демодулятора DVB-S2

#### Систематические нерегулярные коды повторения-накопления (IRA)

IRA коды [1], разработанные на основе RA кодов [2], используются в DVB-S2 стандарте. Систематический IRA код может быть представлен с помощью графа Таннера с  $m$  проверочных узлов и  $n$  символьных узлов. Символьные узлы  $n$  могут быть разделены на  $k = n - m$  информационных узлов ( $IN$ ) и  $m$  кодовых узлов ( $PN$ ). Информационные узлы представляют собой биты информации кодового слова в то время как кодовые узлы представляют кодовые биты, соответственно. Проверочные узлы ( $CN$ ) отображают проверочные уравнения. Все проверочные узлы соединены с информационными узлами через перемешиватель (permutation), как показано на рис. 2.

Информационные узлы имеют различные степени распределения -  $f[\text{deg.}c^{\max}, \dots, 3, 2]$ . Каждый проверочный узел связан с " $a$ " информационными и двумя кодовыми узлами (принимается во внимание тот факт, что самый первый проверочный узел связан только с одним кодовым узлом).

Кодовая скорость систематического IRA кода определяется как [3]:

$$R = a * k / (a * k + m) \quad (1)$$

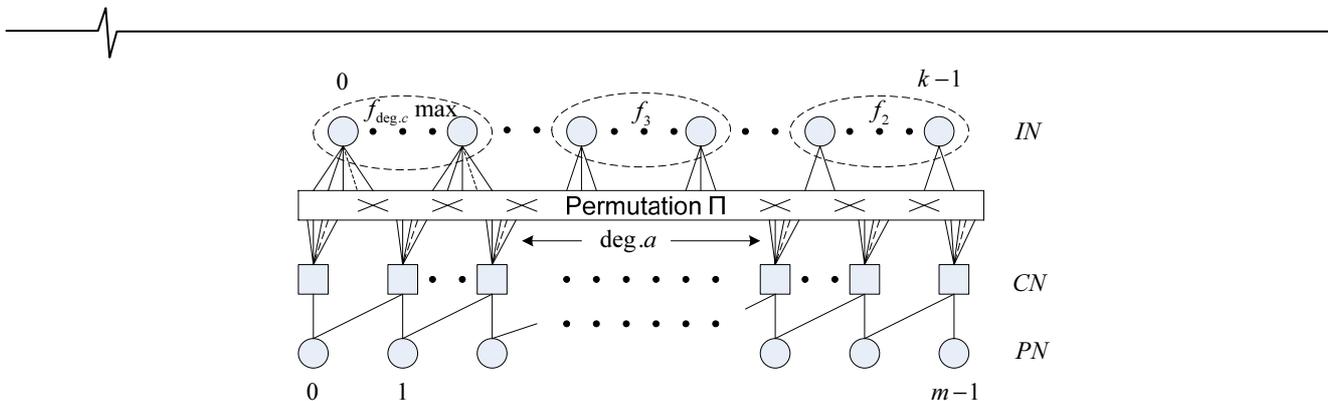


Рис. 2. Граф Таннера IRA кода

**Кодирование IRA кода**

Проверочная матрица систематического IRA кода для DVB-S2 стандарта имеет следующий вид [4]:

$$H = (\Pi | T) = (H_1 | H_2 | T), \quad (2)$$

где  $T$  -  $m \times m$  диагональная матрица принимает вид рис 3.

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & \dots & 0 & 0 & 0 \\ \dots & \dots \\ 0 & 0 & 0 & 0 & \dots & 0 & 1 & 1 \end{bmatrix}$$

Рис. 3. Структура диагональной матрицы

Матрица  $T$  устанавливает связь между проверочными и кодовыми узлами. Каждая строка соответствует одному проверочному узлу, а каждый столбец - одному кодовому узлу. Матрица  $\Pi$  размерностью  $m \times k$  устанавливает связи между информационными и проверочными узлами. Каждая строка матрицы  $\Pi$  имеет одинаковый вес "а" для всех длинных кодов и частично для коротких кодов DVB-S2 стандарта. Вес столбцов зависит от степени распределения символьных узлов. Проверочные матрицы ( $\Pi$ -части) DVB-S2 стандарта специфицируются таблицами [4]. Например, таблица 1 (Parity Bit Accumulators Table) специфицирует короткий код с кодовой скоростью 3/5:

2765 5713 6426 3596 1374 4811 2182 544 3394 2840 4310 771  
 4951 211 2208 723 1246 2928 398 5739 265 5601 5993 2615  
 210 4730 5777 3096 4282 6238 4939 1119 6463 5298 6320  
 4016

4167 2063 4757 3157 5664 3956 6045 563 4284 2441 3412 6334  
 4201 2428 4474 59 1721 736 2997 428 3807 1513 4732 6195  
 2670 3081 5139 3736 1999 5889 4362 3806 4534 5409 6384  
 5809  
 5516 1622 2906 3285 1257 5797 3816 817 875 2311 3543 1205  
 4244 2184 5415 1705 5642 4886 2333 287 1848 1121 3595 6022  
 2142 2830 4069 5654 1295 2951 3919 1356 884 1786 396 4738

Табл. 1.

0	2161 2653	9 266 4878
1	1380 1461	10 4913 3247
2	2502 3707	11 4763 3937
3	3971 1057	12 3590 2903
4	5985 6062	13 2566 4215
5	1733 6028	14 5208 4707
6	3786 1936	15 3940 3388
7	4292 956	16 5109 4556
8	5692 3417	17 4908 4177

Согласно таблице 1,  $\Pi$ -часть матрицы  $H$  состоит из двух частей:  $H_1, H_2$ .

Этот код имеет следующие параметры:

кодовая скорость=3/5,  $n=16200, m=6480, deg.c1=12, deg.c2=3, deg.c3=2, deg.r=11, bitgr1=27, bitgr2=9, bitgr3=9$  (каждая группа включает 360 бит),  $q=18$  ( $q=m/360$ ). На рис. 4 показана структура проверочной матрицы кода.

LDPC кодер кодирует систематически информационный блок размером  $k, i = (i_0, i_1, \dots, i_{k-1})$  в кодовое слово размерности  $n, c = (i_0, i_1, \dots, i_{k-1}, p_0, p_1, \dots, p_{n-k-1})$ . Передача кодового слова начинается в определенном порядке с  $i_0$ , и заканчивается с  $p_{n-k-1}$ .

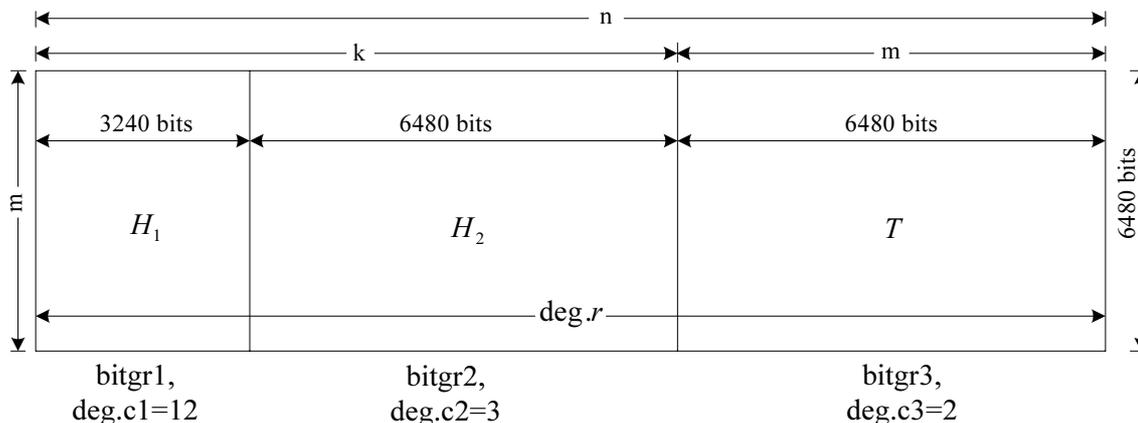


Рис. 4. Структура проверочной матрицы короткого IRA кода с кодовой скоростью равной 3/5.

В соответствии с методом, предложенном в [4], кодирование осуществляется следующим образом (пример для IRA кода с кодовой скоростью  $R=3/5$ ):

1. Инициализация  $P_0 = P_1 = \dots = P_{n-k-1} = 0$ .

2. Суммировать первый информационный бит  $i_0$  с информацией, содержащейся в соответствующих адресах кодовых бит, указанных в первой строке таблицы 1.

$$\begin{aligned}
 P_{2765} &= P_{2765} + i_0 & P_{2182} &= P_{2182} + i_0 \\
 P_{5713} &= P_{5713} + i_0 & P_{544} &= P_{544} + i_0 \\
 P_{6426} &= P_{6426} + i_0 & P_{3394} &= P_{3394} + i_0 \\
 P_{3596} &= P_{3596} + i_0 & P_{2840} &= P_{2840} + i_0 \\
 P_{1374} &= P_{1374} + i_0 & P_{4310} &= P_{4310} + i_0 \\
 P_{4811} &= P_{4811} + i_0 & P_{771} &= P_{771} + i_0
 \end{aligned}$$

3. Для следующих 359 бит информации,  $i_m$ ,  $m = 1, 2, \dots, 359$  суммировать  $i_m$  бит с информацией, содержащейся в соответствующих адресах кодовых бит  $\{x + m \bmod 360 \times q\} \bmod (n - k)$ , где  $x$  обозначает адреса кодовых бит, вычисленных в соответствии с первым битом, а  $q$  является кодовой константой. Так, продолжая пример, для информационного бита  $i_1$  выполняются следующие операции:

$$\begin{aligned}
 P_{2783} &= P_{2783} + i_1 & P_{2200} &= P_{2200} + i_1 \\
 P_{5731} &= P_{5731} + i_1 & P_{562} &= P_{562} + i_1 \\
 P_{6444} &= P_{6444} + i_1 & P_{3412} &= P_{3412} + i_1 \\
 P_{3614} &= P_{3614} + i_1 & P_{2858} &= P_{2858} + i_1 \\
 P_{1392} &= P_{1392} + i_1 & P_{4329} &= P_{4329} + i_1 \\
 P_{4828} &= P_{4828} + i_1 & P_{789} &= P_{789} + i_1
 \end{aligned}$$

4. Для 361 информационного бита адреса кодовых бит приведены во второй строке таблицы 1. В аналогичном порядке адреса кодовых бит для следующих 359 бит информации,  $m = 361, 362, \dots, 719$  получены с помощью формулы  $\{x + (m \bmod 360) \times q\} \bmod (n - k)$ , где  $x$  обозначает адрес кодовых бит, вычисленных в соответствии с информационным битом  $i_{360}$ , т.е. позиции во второй строке таблицы 1.

5. В аналогичном порядке, для каждой группы из 360 новых информационных бит, новая строка в таблице 1 используется для поиска адресов кодовых бит.

После обработки последней строки из таблицы 1 для каждого информационного бита известны все адреса кодовых бит, накапливающих/суммирующих этот бит.

Основываясь на этой информации, могут быть определены адреса информационных бит, вовлеченных в вычисление каждого кодового бита. По сути дела адреса информационных бит есть адреса "1" в строке проверочной матрицы (П-часть). Такое представление проверочной матрицы обычно называют "alist format".

Таблица 2 иллюстрирует часть проверочной матрицы  $H = (\Pi | T)$  короткого LDPC кода с кодовой скоростью  $= 3/5$ .

Таблица 2.

3	908	1202	2308	3121	3218	3240	4181	6569	9720	0
445	721	1716	2049	2198	3014	3480	3600	9488	9720	9721
698	729	857	2430	2926	3191	3960	6067	9107	9721	9722

Схема кодера представлена на рис. 5. Кодер можно рассматривать как генератор проверочных узлов, основанный на проверочной матрице и рекурсивном сверточном кодере с кодовой скоростью 1.

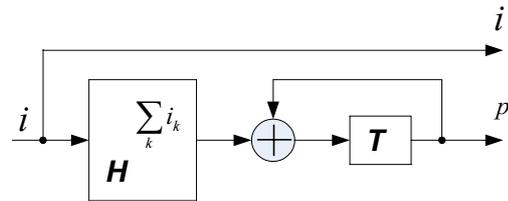


Рис. 5. Схема кодера IRA кода

Кодер суммирует (по модулю 2) все биты информации в соответствии с позициями "1" в проверочной матрице  $H$  (П-часть), и результат подается на сверточный кодер. В свою очередь сверточный кодер вычисляет кодовые биты.

Расчет кодового бита 0 ( $p_0$ ) это особый случай. Бит  $p_0$  имеет позицию 9720 и вычисляется в соответствии с алгоритмом:

```
for(k=0; k<deg.r-2; ++)
```

$$[9720]=[3]+[908]+[1202]+[2308]+[3121]+[3218]+[3240]+[4181]+[6569];$$

Следующие кодовые биты вычисляются в соответствии с алгоритмом:

```
for(k=0; k<deg.r-1; ++)
```

$$[9721]=[445]+[721]+[1716]+[2049]+[2198]+[3014]+[3480]+[3600]+[9488]+[9720];$$

$$[9722]=[698]+[729]+[857]+[2430]+[2926]+[3191]+[3960]+[6067]+[9107]+[9721]; \text{ и т.д.}$$

### Декодирование

Декодирование IRA кода можно свести к процедуре декодирования структурированного ("architecture-aware" LDPC (AA-LDPC) кода [5].

Как известно [5], применение AA-LDPC кодов при проектировании LDPC декодера, позволяет решить проблемы, связанные с внутренними межузловыми связями между вычислительными элементами, уменьшить требования к памяти, увеличить скорость сходимости декодирующего алгоритма, а также решить проблему масштабируемости, ассоциированную с LDPC декодером.

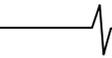
#### Построение AA-LDPC кодов

AA-LDPC код описывается "блочной" матрицей, состоящей из  $B \times D$  подматриц (блоков), где каждый блок - двоичная  $S \times S$  подматрица, формируемая по вектору кодонезависимых параметров  $S$ .

Каждая подматрица должна удовлетворять следующим двум условиям: каждый столбец имеет одну 1, и каждый строка - одну 1. Подматрицы, составляющие блочную матрицу должны быть только трех видов:

1. Подматрицы, включающие только нули (*all-zero matrix*);
2. Еденичные подматрицы (identity matrix,  $I^0$ );
3. Перестановочные матрицы (permutation matrix,  $I^x$ ).

$S \times S$  перестановочная матрица получается из единичной подматрицы либо путем случайной перестановки рядов (идентично столбцов), либо путем циклического сдвига единичной подматрицы в соответствии с индексом сдвига  $x$ , при этом матрица обозначается как  $I^x$ . Регуляр-



ный (C, R)-LDPC код должен иметь  $D$  подматриц в столбце и  $B$  подматриц в строке базисной матрицы соответственно. Регулярный AA-LDPC код имеет длину  $n = SB$ , и кодовую скорость  $R >= 1 - D/B$ .

**Проектирование "AA-LDPC" кода из IRA кода для DVB-S2 стандарта**

Так как линейные блочные коды обладают свойствами коммутативности, то в процессе декодирования можно изменить порядок строк матрицы  $H$ .

Из алгоритма кодирования, предложенного в [4], можно видеть, что он обрабатывает группы, каждая из которых равна 360 бит. Поэтому имеет смысл разбить проверочную матрицу на подматрицы размерностью  $360 \times 360$ . Кроме того, имеет смысл переупорядочить проверочные уравнения и сгруппировать их в группы проверочных узлов следующим образом:

Проверочные узлы:  $0, q, 2 \times q, 3 \times q, \dots, 359 \times q$  формируют группу проверочных узлов 0 (CNG0).

Проверочные узлы:  $1, 1 + q, 1 + 2 \times q, 1 + 3 \times q, \dots, 1 + 359 \times q$  формируют группу проверочных узлов 1 (CNG1).

Проверочные узлы:  $2, 2 + q, 2 + 2 \times q, 2 + 3 \times q, \dots, 2 + 359 \times q$  формируют группу проверочных узлов 2 (CNG2).

Проверочные узлы:  $q - 2, q - 2 + q, q - 2 + 2 \times q, q - 2 + 3 \times q, \dots, q - 2 + 359 \times q$  формируют группу проверочных узлов  $q - 2$  (CNG $q - 2$ ).

Проверочные узлы:  $q - 1, q - 1 + q, q - 1 + 2 \times q, q - 1 + 3 \times q, \dots, q - 1 + 359 \times q$  формируют группу проверочных узлов  $q - 1$  (CNG $q - 1$ ).

Дополнительно каждые 360 символьных узлов группируются в группы символьных узлов.

Символьные узлы:  $0, 1, 2, 3, \dots, 359$  формируют группу символьных узлов 0 (BNG0)

Символьные узлы:  $0 + 360, 1 + 360, 2 + 360, 3 + 360, \dots, 359 + 360$  - группа символьных узлов 1 (BNG1).

Символьные узлы:  $0 + 2 \cdot 360, 1 + 2 \cdot 360, 2 + 2 \cdot 360, 3 + 2 \cdot 360, \dots, 359 + 2 \cdot 360$  - группа символьных узлов 2 (BNG2).

Символьные узлы:  $0 + n360, 1 + n360, 2 + n360, 3 + n360, \dots, 359 + n360$  - группа символьных узлов  $n$  (BNG $n$ ), где максимальное значение параметра  $n$  для длинного кода равно  $n_{\max} = 179 - q$  и для короткого кода  $n_{\max} = 44 - q$  соответственно.

Результирующая переупорядоченная проверочная матрица может быть записана следующим образом:

$$H = \begin{bmatrix} H(0,0) & H(0,1) & \dots & H(0,44) \\ H(1,0) & H(1,1) & \dots & H(1,44) \\ \vdots & \vdots & \vdots & \vdots \\ H(q,0) & H(q,1) & \dots & H(q,44) \end{bmatrix}$$

Обозначим  $H(m,n)$  как ненулевую подматрицу переупорядоченной проверочной матрицы  $H$ , соответствующей группе проверочных узлов  $m$ , и группе символьных узлов  $n$ .

Для строки  $n$  из таблицы 1 может быть вычислена группа подматриц, соответствующих группе символьных узлов  $n$ .

Для данного значения  $k$  в строке  $n$ , соответствующая

группа проверочных узлов есть  $m = k \bmod q$ , индекс сдвига равен  $x = \lfloor k/q \rfloor$ , и соответствующая перестановочная подматрица записывается как  $H(m,n) = I^{(x)}$ .

Например, для первой строки таблицы 1  
2765 5713 6426 3596 1374 4811 2182 544 3394 2840 4310 771

соответствующие значения перестановочных подматриц:

$$\begin{array}{ll} 2765: H(11,0) = I^{153} & 2182: H(4,0) = I^{121} \\ 5713: H(7,0) = I^{317} & 544: H(4,0) = I^{30} \\ 6426: H(0,0) = I^{357} & 3394: H(10,0) = I^{188} \\ 3596: H(14,0) = I^{199} & 2840: H(14,0) = I^{157} \\ 1374: H(6,0) = I^{76} & 4310: H(8,0) = I^{239} \\ 4811: H(5,0) = I^{267} & 771: H(15,0) = I^{43} \end{array}$$

Перестановочные подматрицы для следующих строк из таблицы 1 должны рассчитываться по тому же пути. Полученные таким образом перестановочные подматрицы удовлетворяют условиям, предъявляемым к AA-LDPC коду.

**"Novel" Алгоритм декодирования**

"Novel" алгоритм декодирования был предложен в работе [6] для декодирования AA-LDPC кода в демодуляторе DVB-S2. В других публикациях этот алгоритм известен также как "layered" [7], "staggered" [8].

Этот алгоритм декодирования обладает определенными преимуществами по сравнению с другими [9, 10, 11, 12]:

1. Декодер, спроектированный при использовании этого алгоритма, требует меньшего объема памяти, чем стандартный декодер, реализующий стандартный двухфазный метод декодирования - "сумма-произведение" (two-phase message-passing, TPMP algorithm).

2. Скорость сходимости алгоритма существенно выше, чем у "TPMP" (алгоритм выполняет меньше итераций для достижения равной эффективности декодирования).

При описании алгоритма воспользуемся метрикой, называемой логарифмическим отношением функций правдоподобия (log-likelihood ratio - LLR) [13], которая определяется выражением.

$$LLR(x|y) = \ln \left[ \frac{p(y|x=0)}{p(y|x=1)} \right] \quad (3)$$

Пусть  $x = [x_1, x_2, \dots, x_n]$  - кодовое слово, которое модулируется при использовании двоичной фазовой модуляции, и модулированные значения  $x$  передаются по каналу с белым аддитивным гауссовым шумом (АБГШ).

Входную последовательность принятых сигналов (символов) обозначим, как  $y = [y_1, y_2, \dots, y_n]$ . Демодулятор принимает входную последовательность сигналов и вычисляет соответствующие LLR значения:

$$\text{For } j = 1, 2, \dots, n \quad \lambda_j = LLR(y_j|x_j) = \ln \frac{p(y_j|x_j=0)}{p(y_j|x_j=1)} \quad (4)$$

Обычно при двоичной bipolarной передаче по каналу LLR значения вычисляются как

$$\lambda_j = \frac{2}{\sigma^2} * y_j, \tag{5}$$

где  $\sigma^2$  - дисперсия АБГШ.

При описании алгоритма воспользуемся следующими обозначениями:

$\lambda_j$  - "llr" значение принятого  $j$ -го символа,  $R_{ij}[k]$  - внешние (extrinsic) "llr" значения,  $Q_{ij}[k]$  - внутренние (intrinsic) "llr" значения,  $\Lambda_j$  - апостериорные "llr" значения,  $R[i]$  - индексы символьных узлов, вовлеченных в вычисление  $i$ -го проверочного узла.

"Novel" алгоритм описывается следующим образом:

1. Инициализация:

$$R_{ij}^0 = 0, \Lambda_j^0 = \lambda_j \tag{6}$$

2. Горизонтальное сканирование (check node update rule):

$$Q_{ji}^{(k)} = \Lambda_j^{(k-1)} - R_{ij}^{(k-1)}; \tag{7}$$

$$R_{ij}^{(k)} = \prod_{j \in R(i) \setminus j} \text{sign}(Q_{ji}^{(k)}) * \{ \text{MIN}_{j \in R(i) \setminus j} | Q_{ji}^{(k)} | \}; \tag{8}$$

$$\Lambda_j^{(k)} = Q_{ji}^{(k)} + R_{ij}^{(k)}. \tag{9}$$

**Архитектура LDPC декодера**

Центром архитектуры декодера, реализующего "Novel" алгоритм декодирования, является вычислительный блок [6], изображенный на рис. 6. Блок вычисляет внешние "llr" и апостериорные "llr" значения, используя априорные "llr" значения, поступающие на вход блока. Декодер включает 360 вычислительных блоков (проблема масштабируемости в данной работе не рассматривается), что соответствует размеру группы символьных узлов (BNG).  $Degr * Q_{ji}$  - число внутренних "llr" значений вычисляется путем вычитания  $R_{ij}$  из последовательно поступающих значений  $\Lambda_j$ . Вычисленные внутренние "llr" значения запоминаются в FIFO элементе и одновременно в блоке вычисления внешних "llr" величин (Check Node Update Unit) для дальнейшего вычисления новых внешних "llr" значений.

Апостериорные "llr" значения вычисляются путем сложения внутренних величин с выхода FIFO элемента, с новыми внешними "llr" значениями, поступающими с выхода блока вычисления внешних величин.

Архитектура декодера представлена на рис. 7. Де-

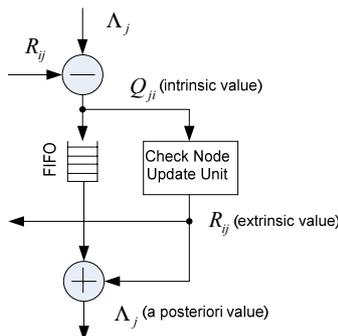


Рис. 6. Схема вычислительного блока

кодер включает следующие блоки:

1. Конфигурирующий блок памяти (Configuration ROM);
  2. Блок памяти апостериорных "llr" значений (Lambda memory);
  3. Два сдвигающих регистра (barrel shifters);
  4. Блоки памяти внешних "llr" значений (R memories).
- Конфигурирующий блок памяти содержит адреса групп символьных узлов в блоке памяти апостериорных "llr" значений, а также сдвигающие индексы для соответствующих групп символьных узлов.

Блок памяти апостериорных "llr" значений используется для запоминания "llr" значений с выхода демодулятора ( $\lambda_j$ ), а также для записи апостериорных "llr" значений с выходов вычислительных блоков ( $\Lambda_j$ ).

Сдвиговые регистры выполняют циклический сдвиг соответствующих символьных групп в соответствии с их индексами сдвига.

Блоки памяти внешних "llr" значений предназначены для записи и чтения внешних "llr" значений в процессе декодирования.

На рис. 8 представлен алгоритм декодирования, включающий следующие блоки и операции.

1. Счетчик групп проверочных узлов (k), счетчик групп символьных узлов (i), счетчик итераций, указатели адреса записи и чтения в R блоках памяти (read\_ptr, write\_ptr) и счетчик llr\_word's (j) обнуляются, кодовое слово с выхода демодулятора запоминается в "Lambda" блоке памяти.
2. Для всех групп символьных узлов, принадлежащих текущей группе проверочных узлов (старт  $k = 0$ ), вычисляются внутренние "llr" значения, вычисленные на основе априорных "llr" значений, считанных из "Lambda" блока памяти.
3. Deg.r внешних "llr" значений вычисляются в каждом "Check Node Update Unit" блоке.
4. Используя внутренние "llr" значения, считанные из FIFO's, и внешние "llr" значения, считанные из "Check Node Update Unit" блоков, вычисляются апостериорные "llr" значения в каждом вычислительном блоке, которые запоминаются в "Lambda" блоке памяти.

Шаги 2-4 повторяются до тех пор, пока не будет выполнено условие  $k=q-1$ .

Если это условие удовлетворено, выполняется шаг 5.

5. Проверка условия:  $x \otimes H^T = 0$

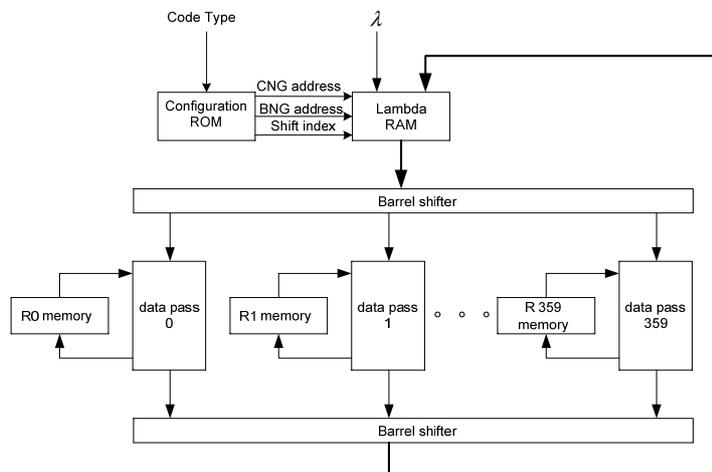


Рис. 7. Архитектура декодера

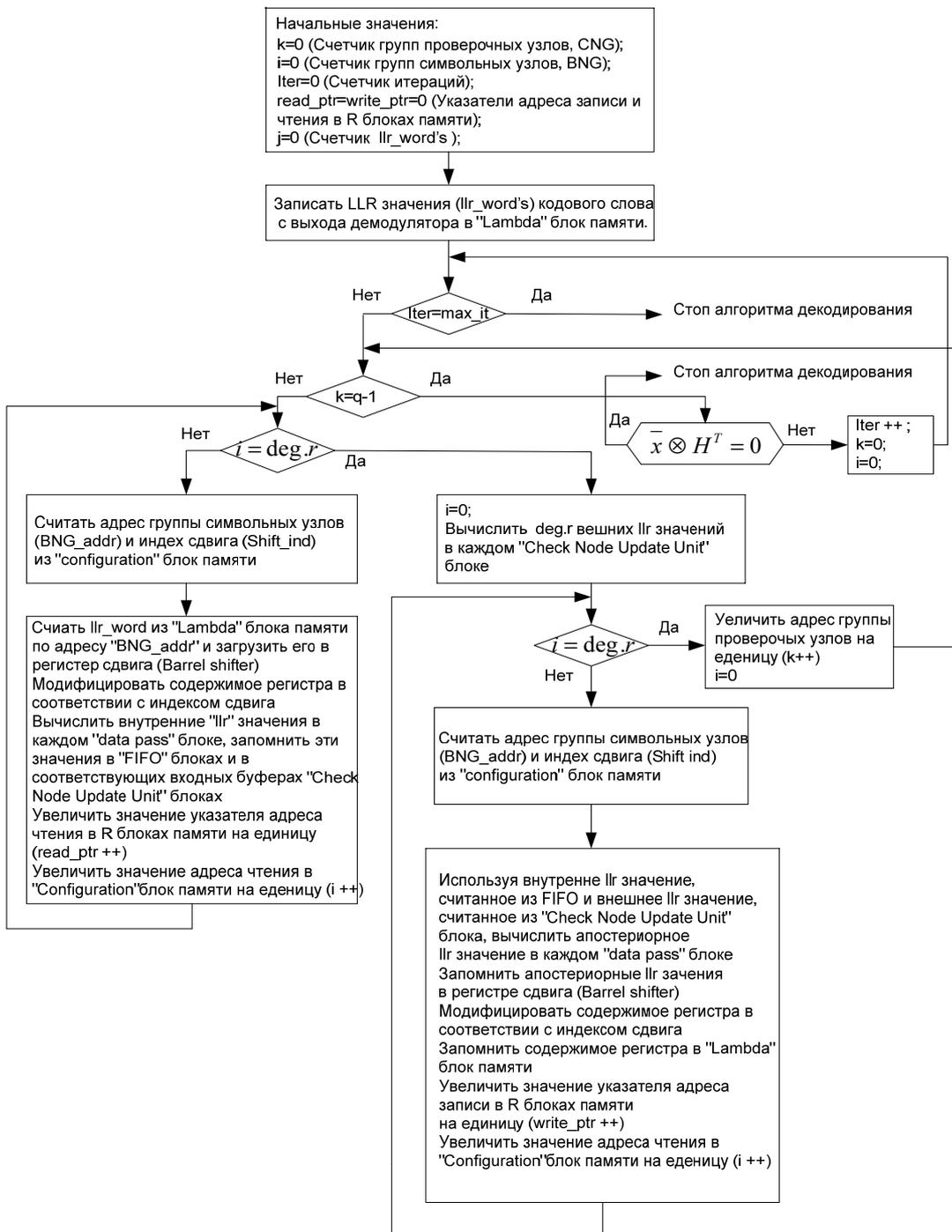


Рис. 8. Алгоритм декодирования

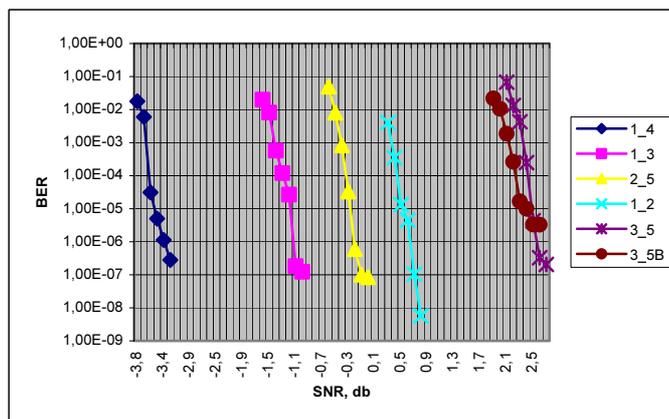


Рис. 9. Эффективность декодера для кодовых скоростей - 1/4, 1/3, 2/5, 1/2, 3/5, 3/5B

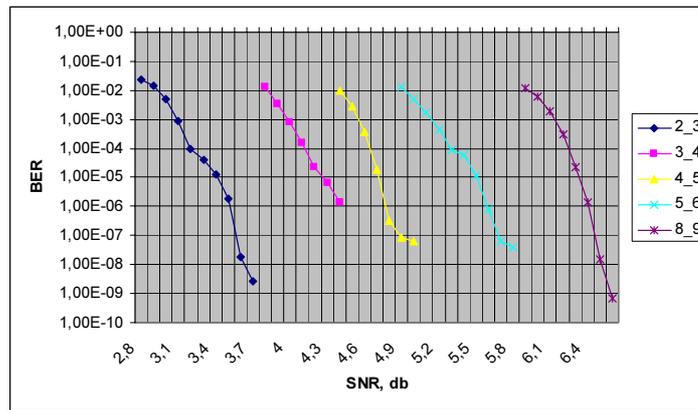


Рис. 10. Эффективность декодера для кодовых скоростей – 2/3, 3/4, 4/5, 5/6, 8/9

Если это условие удовлетворено, то декодирование завершается, декодированный бинарный вектор  $x$  передается к "BCH" декодеру и алгоритм декодирования переходит к шагу 1. Если это условие не удовлетворено, то счетчик итераций увеличивается на единицу, и алгоритм декодирования переходит к шагу 2.

На рис. 9, 10 представлены оценки эффективности декодера LDPC кода длиной 16200 при различных кодовых скоростях в канале с АБГШ для случая двоичной фазовой модуляции.

При моделировании декодера использовалась арифметика с фиксированной запятой. Для представления мягких решений (LLR значений) демодулятора и сообщений внутри декодера использовалась прямой код с числом разрядов  $b=6$ .

#### Литература

1. D. Divsalar, H. Jin, and R. McEliece. Coding theorems for turbo-like codes. Proc. 36th Annual Allerton Conf. on Comm., Control, and Computing, Sept. 1998, pp. 201-210.
2. H. Jin, A. Khandekar, and R. McEliece. Irregular repeat-accumulate codes. Proc. 2nd. Int. Symp. on Turbo Codes and Related Topics, Brest, France, Sept. 2000, pp. 1-8.
3. Mohammad M. Mansour. High-Performance Decoders for Regular and Irregular Repeat-Accumulate Codes. IEEE Communication Society Globecom, 0-7803-8794-5/04/\$20.00 © 2004 IEEE, pp. 2583-2588.
4. European Telecommunications Standards Institute. Digital video broadcasting (DVB); Second generation framing structure, channel coding and modulation systems for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications. DRAFT EN 302 307 (v.1.1.2 06.2006).
5. Mohammad M. Mansour. Architecture-Aware Low-Density Parity-Check Codes. 0-7803-7761-3/03/\$17.00 © 2003 IEEE, pp. 57-60.
6. J. Dielissen, A. Hekstra, and V. Berg. Low cost LDPC decoder for DVB-S2. In Proc. 2006 Design, Automation and Test in Europe (DATE '06), Munich, Germany, Mar. 2006.
7. D. Hecovar. A Reduced Complexity Decoder Architecture via Layered Decoding of LDPC Codes. In IEEE Workshop on Signal Processing Systems, SISAP 2004, 2004, pp 107-112.
8. Mohammad M. Mansour. Unified Decoder Architectures for

Repeat-Accumulate and LDPC Codes. 0-7803-8622-1/04/\$20.00 © 2004 IEEE, pp. 527-531.

9. Овечкин Г. В., Чикин А.В. Архитектура и реализация декодера LDPC кодов для демодулятора DVB-S2. ТРУДЫ НИИР, ВЫПУСК 3, 2008.
10. Ефимов А.Г. Об аппаратной реализации декодеров LDPC-кодов. ВОПРОСЫ ПЕРЕДАЧИ И ЗАЩИТЫ ИНФОРМАЦИИ, Сборник статей под редакцией д.т.н Е. А. Крука, Санкт-Петербург, 2006.
11. F.Kienle, T. Brack, and N. Wehn. A synthesizable IP core for DVB-S2 LDPC code decoding. In IEEE Conference on Design Automation and Test Europe (DATE), 2005.
12. P. Urard, E. Yeo, L. Paumier, P. Georgelin, T. Michel, V. Lebars, E. Lantreibecq, B. Gupta. A 135Mb/s DVB-S2 Compliant Codec Based on 64800b LDPC and BCH Codes. 2005 IEEE International Solid-State Circuits Conference, 0-7803-8904-2/05/\$20.00 ©2005 IEEE.
13. Бернадт Сяляр. Цифровая связь. Теоретические основы и практическое применение. М.: Издательский дом "Вильямс", 2003, 1104 с.

#### METHODS AND THE APPARATUS FOR CODING AND DECODING SYSTEMATIC IRREGULAR REPEAT ACCUMULATE CODE (IRA) OF DVB-S2 AND DVB-T2 DEMODULATORS.

*Kravtchenko A.N.*

Design method of parity check matrixes of systematic irregular repeat accumulate code is proposed for DVB-S2 and DVB-T2 demodulators based on the Parity Bit Tables, which specify IRA codes with different code rates. Scheme of encoder is presented. Method of the design of structural AA-LDPC code from IRA code is suggested. Designed code has the cycle code property. Decoder architecture for AA-LDPC codes is described based on a novel decoding algorithm which has a faster convergence rate and hence a throughput advantage and also the memory overhead problem is reduced over the standard decoding algorithm (TPMP). Interconnect complexity problem in the decoder is mitigated by using the AA-LDPC codes. Result of performance simulation of the decoder for 16200 bits code words with different code rates over AWGN channel is presented.