

В. П. Семеренко, канд. техн. наук
(Винницкий государственный технический университет,
Украина)

Разработка универсального кодера — декодера циклических кодов

(Статью представил канд. техн. наук
В. В. Мохор)

Рассмотрены кубическое представление циклических кодов и новая интерпретация циклического кодирования на основе теории линейной последовательностной машины. Предложена систолическая реализация универсального кодера — декодера, способного перенастраиваться на заданные алгоритмы кодообработки в реальном масштабе времени.

Розглянуто кубічне представлення циклічних кодів та нова інтерпретація циклічного кодування на основі теорії лінійної послідовнісної машини. Запропоновано систолічну реалізацію універсального кодера-декодера, здатного перебудовуватись на задані алгоритми кодообробки в реальному масштабі часу.

К л ю ч е в ы е с л о в а: циклические коды, кодеры, декодеры, линейная последовательностная машина, систолическая структура.

Выбор методов кодирования информации является сложной задачей и зависит от ряда факторов: требуемой верности передаваемых сообщений, длительности передачи, параметров канала связи, секретности передаваемых данных [1]. Эти факторы непрерывно изменяются, что приводит к необходимости оперативного изменения характеристик кодов и, следовательно, к смене средств кодообразования. Решить эту проблему можно как поиском новых алгоритмов кодирования — декодирования, чем занимается теория адаптивного кодирования и декодирования [2], так и созданием новых средств кодообработки традиционных кодов.

Смена алгоритмов кодирования — декодирования возможна, если обработка информации выполняется программно на микропроцессорах. Однако универсальные программные средства кодообразования значительно уступают по скорости преобразования специализированным аппаратным средствам [3]. Поэтому актуальной является задача создания универсальных и быстродействующих кодеров и декодеров, способных перенастраиваться на реализацию заданного алгоритма в реальном масштабе времени.

Такая задача может быть решена, если кодеры и декодеры будут построены на основе универсальной программируемой логической интегральной схемы (ПЛИС).

Рассмотрим задачи кодирования и декодирования только для наиболее распространенных на практике циклических кодов.

Способы описания циклических кодов. Наиболее распространены три способа описания циклического (n, k) -кода [1, 4, 5]: — полиномиальное описание с помощью порождающего полинома $P(x)$ степени r ($r = n - k$):

$$P(x) = p_0 \oplus p_1 x \oplus \dots \oplus p_r x^r; \quad (1)$$

матричное описание с помощью порождающей $(k \times n)$ -матрицы \mathbf{G} или проверочной $(r \times n)$ -матрицы \mathbf{H} ; описание на основе теории линейной последовательностной машины (ЛПМ).

Первые два способа описания, несмотря на наглядность представления, требуют громоздких аналитических преобразований при кодировании и декодировании циклических кодов. Третий способ описания используется в основном при рассмотрении схемной реализации циклического кодирования в виде цепей регистров сдвига. Такие схемы также часто называют фильтрами [4] или линейными переключательными схемами [5].

Рассмотрим новую интерпретацию циклического кодирования с помощью теории ЛПМ, покажем ее взаимосвязь с традиционными представлениями и предложим удобные алгоритмы кодирования и декодирования.

Описание циклических кодов на основе теории ЛПМ. В общем виде n -мерная ЛПМ над полем Галуа $GF(q)$ задается функцией состояний (переходов)

$$\mathbf{S}(t+1) = \mathbf{A} \cdot \mathbf{S}(t) + \mathbf{B} \cdot \mathbf{U}(t), \quad GF(q),$$

и функцией выходов

$$\mathbf{Y}(t) = \mathbf{C} \cdot \mathbf{S}(t) + \mathbf{D} \cdot \mathbf{U}(t), \quad GF(q),$$

где $\mathbf{S}(t)$, $\mathbf{U}(t)$, $\mathbf{Y}(t)$ — соответственно векторы состояний, входной и выходной; $\mathbf{A} = \|a_{ij}\|_{r \times r}$ — основная характеристическая матрица ЛПМ; $\mathbf{B} = \|b_{ij}\|_{r \times l}$, $\mathbf{C} = \|c_{ij}\|_{m \times r}$, $\mathbf{D} = \|d_{ij}\|_{m \times l}$ — характеристические матрицы ЛПМ.

Для описания процессов в традиционном r -разрядном регистре сдвига достаточно ограничиться представлением ЛПМ в виде модели автомата Мура над полем $GF(2)$

$$\mathbf{S}(t+1) = \mathbf{A} \cdot \mathbf{S}(t) \oplus \mathbf{B} \cdot \mathbf{U}(t),$$

$$\mathbf{Y}(t) = \mathbf{S}(t),$$

где $\mathbf{A} = \|a_{ij}\|_{r \times r}$, $\mathbf{B} = \|b_i\|_r$.

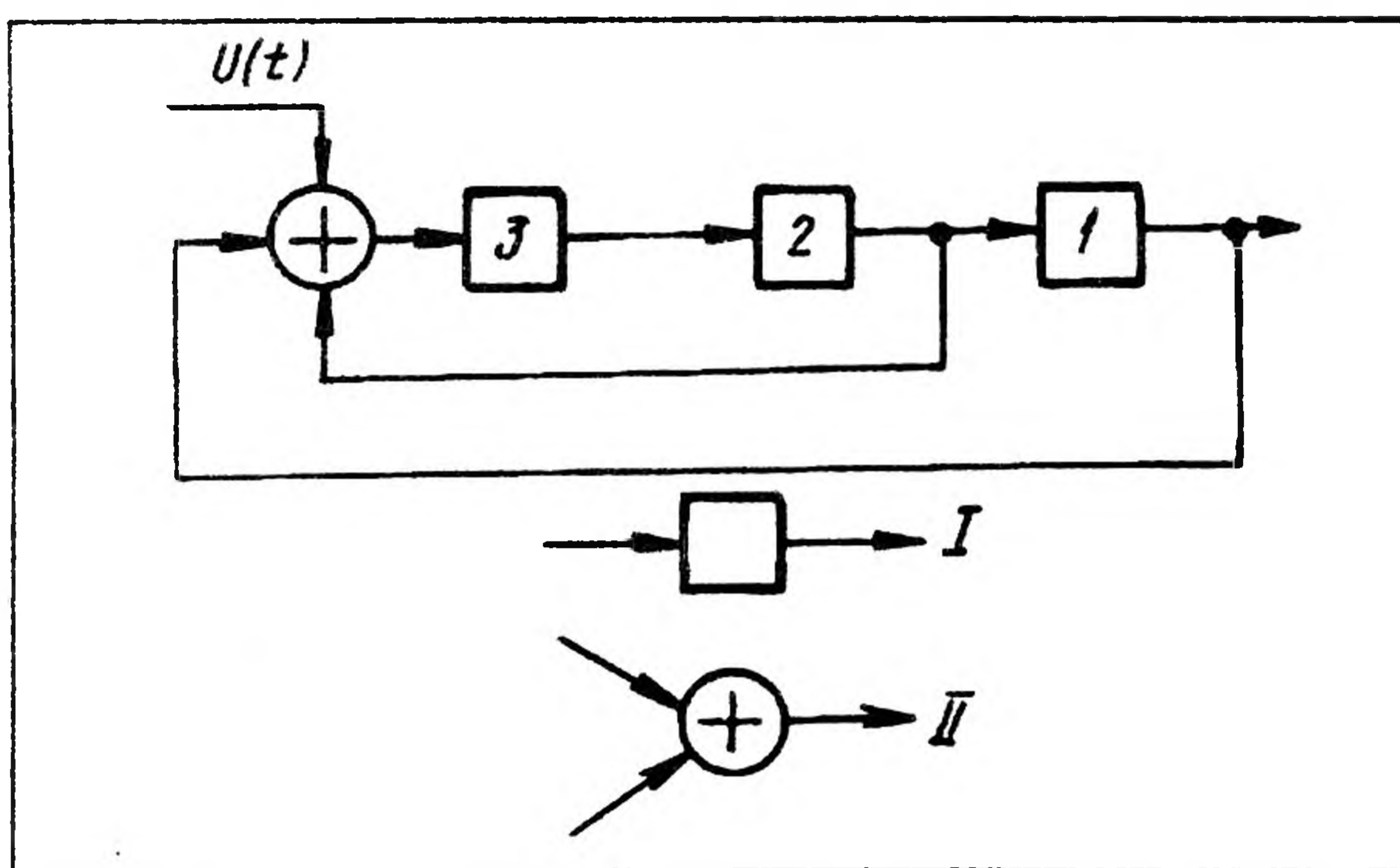


Рис. 1. Схема реализации ЛПМ в виде односумматорного регистра сдвига для порождающего полинома $P(x) = 1 \oplus x^2 \oplus x^3$: I — элемент задержки (триггер); II — сумматор по модулю два

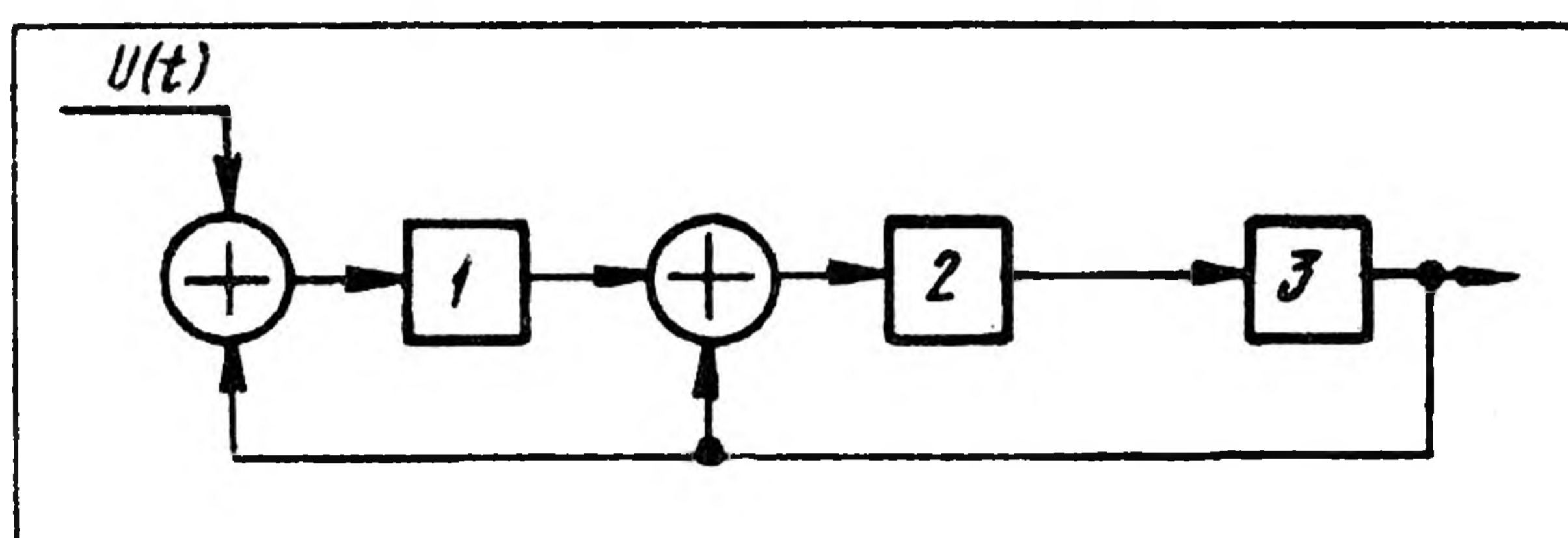


Рис. 2. Схема реализации ЛПМ в виде многосумматорного регистра сдвига для порождающего полинома $P(x) = 1 \oplus x^2 \oplus x^3$

На практике удобно пользоваться следующими формулами:

$$S(t+1) = \begin{cases} \mathbf{A} \cdot \mathbf{S}(t) \oplus \mathbf{B}, & \text{если } U(t) = 1, \\ \mathbf{A} \cdot \mathbf{S}(t), & \text{если } U(t) = 0. \end{cases} \quad (2)$$

В поле $GF(2)$ умножение матрицы \mathbf{M} на вектор \mathbf{N} заменяется сложением по модулю 2 тех столбцов матрицы \mathbf{M} , номера которых совпадают с номерами единичных компонент вектора \mathbf{N} .

При схемной реализации ЛПМ матрица \mathbf{A} задает структуру связей между элементами задержек (триггеров), а матрица \mathbf{B} — структуру входных связей. Для получения регистров сдвига известны два основных варианта реализации матрицы \mathbf{A} [6]. Если матрица \mathbf{A} представляет собой сопровождающую матрицу вида

$$\mathbf{A} = \begin{vmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 \\ p_0 & p_1 & p_2 & \dots & p_{r-1} \end{vmatrix}, \quad (3)$$

то получается регистр сдвига с одним многовходным сумматором по модулю 2. Если же матрица \mathbf{A} представляет собой транспонированную сопровождающую матрицу вида

$$\mathbf{A} = \begin{vmatrix} 0 & 0 & \dots & 0 & p_0 \\ 1 & 0 & \dots & 0 & p_1 \\ 0 & 1 & \dots & 0 & p_2 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 & p_{r-1} \end{vmatrix}, \quad (4)$$

то получается многосумматорный регистр сдвига. В обеих реализациях значение элемента $a_{ij} = 1$ ($a_{ij} = 0$) матрицы \mathbf{A} показывает наличие (отсутствие) связи между выходом j -го триггера с входом i -го триггера. Элементы последней строки матрицы (3) и элементы последнего столбца матрицы (4) представляют собой коэффициенты полинома (1).

Для одновходовой ЛПМ матрица \mathbf{B} представляет собой вектор, в котором значение элемента $b_i = 1$ ($b_i = 0$) показывает наличие (отсутствие) связи входа ЛПМ с входом i -го триггера. Для первой и второй реализаций матрицы \mathbf{A} соответствующие матрицы \mathbf{B} имеют вид

$$\mathbf{B} = \begin{vmatrix} 0 \\ \dots \\ 0 \\ 1 \end{vmatrix}; \quad (5)$$

$$\mathbf{B} = \begin{vmatrix} 1 \\ 0 \\ \dots \\ 0 \end{vmatrix}. \quad (6)$$

На рис. 1 и 2 изображены две реализации ЛПМ соответственно для матриц (3), (5) и (4), (6) на основе полинома $P(x) = 1 \oplus x^2 \oplus x^3$.

Кодирование циклических кодов. При традиционном систематическом кодировании циклических кодов для формирования n -разрядного кодового вектора $\mathbf{C}(x)$ k -разрядный информационный вектор $\mathbf{I}(x)$ умножается на одночлен x^r и к этому произведению добавляется остаток $\mathbf{R}(x)$, полученный в результате деления произведения $\mathbf{C}(x) \cdot x^r$ на порождающий полином $\mathbf{P}(x)$.

Рассмотрим смысл операции кодирования при использовании математического аппарата ЛПМ.

При поступлении на вход ЛПМ вектора $\mathbf{I}(x)$ произойдет переход ЛПМ из начального нулевого состояния $\mathbf{S}(0)$ в некоторое состояние $\mathbf{S}(k)$. Далее на вход ЛПМ необходимо подать такую r -разрядную входную последовательность $\mathbf{U}(x)$, чтобы ЛПМ снова перешла в состояние $\mathbf{S}(0)$. Из теории ЛПМ известно, что для любой сильносвязной r -уп-

руляемой ЛПМ существует входная последовательность $U(r)$ длиной r , переводящая ЛПМ из состояния $S(i)$ в состояние $S(j)$ [6]. Линейная последовательностная машина является r -управляемой, если ранг матрицы $L_r = ||A^{r-1} \cdot B, A^{r-2} \cdot B, \dots, A \cdot B, B||$ равен r . Для матриц (3), (5) и (4), (6) соответствующие матрицы L'_r и L''_r имеют вид

$$L'_r = \left\| \begin{array}{cccccc} 1 & 0 & \dots & 0 & 0 \\ l_{21} & 1 & \dots & 0 & 0 \\ l_{31} & l_{21} & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ l_{r1} & l_{(r-1)1} & \dots & l_{21} & 1 \end{array} \right\|, \quad l_{ij} = \{0, 1\}; \quad (7)$$

$$L''_r = \left\| \begin{array}{ccccc} 0 & 0 & \dots & 0 & 1 \\ 0 & 0 & \dots & 1 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 1 & 0 & \dots & 0 & 0 \end{array} \right\|. \quad (8)$$

Поскольку ранг матриц (7) и (8) равен r , можно определить искомую последовательность $U(r)$, которая переводит ЛПМ из состояния $S(k)$ в состояние $S(0)$ по известной формуле

$$S(0) \oplus A^r \cdot S(k) = L_r \cdot U(r). \quad (9)$$

Левую часть равенства (9) можно записать в виде

$$S(0) \oplus A^r \cdot S(k) = A^r \cdot S(k) = S(k+r) = S(n). \quad (10)$$

Как следует из (10), ЛПМ из состояния $S(k)$ в состояние $S(n)$ перейдет после подачи на ее вход r -разрядного нулевого вектора $O(x) = ||00 \dots 0||$. Тогда для матрицы вида (8) согласно равенству (9)

$$U(r) = \left\| \begin{array}{c} u(1) \\ u(2) \\ \dots \\ u(r) \end{array} \right\| = S(n) = \left\| \begin{array}{c} s_r(n) \\ s_{r-1}(n) \\ \dots \\ s_1(n) \end{array} \right\|, \quad (11)$$

а для матрицы вида (7) разряды вектора $U(r)$ определяются из следующей системы уравнений:

$$\begin{aligned} u(1) &= s_1(n); \\ u(2) &= s_2(n) \oplus u(1) \cdot l_{21}; \\ \dots & \\ u(r) &= s_r(n) \oplus u(1) \cdot l_{r1} \oplus \dots \oplus u(r-1) \cdot l_{21}; \end{aligned}$$

или

$$\begin{aligned} u(1) &= s_1(n); \\ u(2) &= s_2(n) \oplus s_1(n) \cdot l_{21}; \\ u(3) &= s_3(n) \oplus s_2(n) \cdot l_{21} \oplus s_1(n) \cdot (l_{21} \oplus l_{31}); \\ u(4) &= s_4(n) \oplus s_3(n) \cdot l_{21} \oplus s_2(n) \cdot (l_{21} \oplus l_{31}) \oplus \\ &\quad \oplus s_1(n) \cdot (l_{21} \oplus l_{41}); \end{aligned} \quad (12)$$

Разряды вектора $U(r)$ являются по сути проверочными символами кода, т.е. вектор $U(r)$ совпадает с остатком $R(x)$ при традиционном кодировании. Поэтому можно записать $R(x) = ||U(r)||^{TP} = ||u(1)u(2) \dots u(r)||$.

В канал передачи данных разряды кодового вектора $C(x) = I(x)R(x)$ поступают, начиная со старших (левых) разрядов $I(x)$ и $R(x)$.

Таким образом, алгоритм кодирования циклических кодов на основе теории ЛПМ будет следующим.

Алгоритм 1. 1. Определение по формулам (2) кода состояния $S(k)$, в которое перейдет ЛПМ из состояния $S(0)$ при подаче на ее вход вектора $I(x)$.

2. Определение по формулам (2) кода состояния $S(n)$, в которое перейдет ЛПМ из состояния $S(k)$ при подаче на ее вход вектора $O(x)$.

3. Вычисление по формулам (11) или (12) остатка $R(x)$.

4. Формирование кодового вектора $C(x) = I(x)R(x)$.

5. Конец.

Пример 1. Для (7,4) — кода с порождающим полиномом $P(x) = 1 \oplus x^2 \oplus x^3$ матрицы ЛПМ имеют вид

$$A = \left\| \begin{array}{ccc} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \end{array} \right\|; \quad B = \left\| \begin{array}{c} 0 \\ 0 \\ 1 \end{array} \right\|;$$

$$L = \left\| \begin{array}{ccc} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{array} \right\|; \quad S(0) = \left\| \begin{array}{c} 0 \\ 0 \\ 0 \end{array} \right\|.$$

Для вектора $I(x) = 1010$ необходимо получить кодовый вектор $C(x)$. Согласно п.1 алгоритма 1 находим

$$S(1) = A \cdot S(0) \oplus B = \left\| \begin{array}{c} 0 \\ 0 \\ 1 \end{array} \right\|;$$

$$S(2) = A \cdot S(1) = \left\| \begin{array}{c} 0 \\ 1 \\ 1 \end{array} \right\|;$$

$$S(3) = A \cdot S(2) \oplus B = \left\| \begin{array}{c} 1 \\ 1 \\ 0 \end{array} \right\|;$$

$$S(4) = A \cdot S(3) = \left\| \begin{array}{c} 1 \\ 0 \\ 1 \end{array} \right\|.$$

Согласно п.2 алгоритма 1

$$\mathbf{S}(5) = \mathbf{A} \cdot \mathbf{S}(4) = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix};$$

$$\mathbf{S}(6) = \mathbf{A} \cdot \mathbf{S}(5) = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix};$$

$$\mathbf{S}(7) = \mathbf{A} \cdot \mathbf{S}(6) = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}.$$

По формулам (12) определяем разряды остатка $\mathbf{R}(x)$:

$$u(1) = s_1(7) = 0;$$

$$u(2) = s_2(7) \oplus s_1(7) \cdot l_{21} = 0 \oplus 0 \cdot 1 = 0;$$

$$u(3) = s_3(7) \oplus s_2(7) \cdot l_{21} \oplus s_1(7) \cdot (l_{21} \oplus l_{31}) = 1 \oplus 0 \cdot 1 \oplus 0 \cdot (1 \oplus 1) = 1.$$

Следовательно, $\mathbf{R}(x) = ||001||$ и тогда вектор $\mathbf{C}(x) = ||1010001||$.

Так как все операции в алгоритме 1 сводятся к действиям с булевыми матрицами и векторами, они могут быть легко программно или аппаратно реализованы. Как показано ниже, главное преимущество предлагаемого метода — возможность распараллеливания вычислений.

Декодирование циклических кодов. В задачу декодирования входит обнаружение и

исправление ошибок. С позиций теории ЛПМ смысл этих операций следующий.

При подаче на вход ЛПМ кодового вектора $\mathbf{C}(x)$, в котором отсутствуют ошибки, произойдет переход ЛПМ из начального нулевого состояния $\mathbf{S}(0)$ снова в состояние $\mathbf{S}(0)$. При наличии ошибки в кодовом векторе, который обозначим $\mathbf{C}'(x)$, ЛПМ перейдет в некоторое ненулевое состояние $\mathbf{S}^i(n)$. В дальнейшем для именованного кода состояния, в которое переходит ЛПМ под воздействием кодового вектора, будем использовать общепринятый термин «синдром».

Теорема 1. Ошибка в i -м разряде вектора $\mathbf{C}'(x)$ приводит к появлению синдрома $\mathbf{S}^i(n)$, для перехода из которого в состояние $\mathbf{S}(0)$ необходим i -разрядный вектор

$$||\mathbf{U}(t)||^{\text{TP}} = ||u(1) u(2) \dots u(i)||,$$

где $u(1) = u(2) = \dots = u(i-1) = 0, u(i) = 1, i = 1, \dots, n$.

Теорема 2. При подаче на вход ЛПМ i -разрядного вектора $\mathbf{U}(t)$

$$||\mathbf{U}(t)||^{\text{TP}} = ||u(1) u(2) \dots u(i)||,$$

где $u(1) = u(2) = \dots = u(i-1) = 0, u(i) = 1, i = 1, \dots, n$, будет получен синдром

$\mathbf{S}^i(n)$, соответствующий вектору $\mathbf{C}'(x)$ с ошибкой в i -м разряде. Начальное состояние ЛПМ — состояние $\mathbf{S}(0)$.

При полиномиальном представлении циклических кодов теоремам 1 и 2 соответствует теорема Меггитта [1].

Для использования метода синдромного декодирования с целью исправления одиночных ошибок нет необходимости хранить все синдромы, достаточно только одного из них, например $\mathbf{S}^1(n)$. Поскольку синдром $\mathbf{S}^1(n)$ соответствует вектору $\mathbf{C}'(x)$ с ошибкой в первом разряде, то согласно теореме 1 из состояния $\mathbf{S}^1(n)$ в состояние $\mathbf{S}(0)$ существует путь единичной длины $u(1) = 1$. Из теоремы 1 следует также, что кодовым векторам с ошибкой в соседних разрядах i и $i+1$ соответствуют соседние синдромы $\mathbf{S}^i(n)$ и $\mathbf{S}^{i+1}(n)$. Можно предложить следующий алгоритм исправления одиночной ошибки в векторе $\mathbf{C}'(x)$, используя один известный синдром $\mathbf{S}^1(n)$.

Алгоритм 2. 1. Положить $i = 1, j = 0$. Вычисление по формулам (2) для кодового вектора синдрома $\mathbf{S}(n+j)$.

2. Если $\mathbf{S}(n+j) = \mathbf{S}(0)$, то переход к п.6.

3. Если $\mathbf{S}(n+j) = \mathbf{S}^1(n)$, то переход к п.5.

4. Вычисление $\mathbf{S}(n+j+1) = \mathbf{A} \cdot \mathbf{S}(n+j)$; $i = i+1, j = j+1$. Переход к п.3.

5. Исправление ошибки в i -м разряде кодового вектора.

6. Конец.

Анализ двух вариантов реализации ЛПМ показывает, что декодирование для них осуществляется одинаково, а процесс кодирования более трудоемкий для ЛПМ первого типа (на основе матриц (3) и (5)). Однако для аппаратного воплощения универсального кодера — декодера, как показано ниже, более предпочтительно использование ЛПМ первого типа.

Систолическая реализация универсального кодера — декодера. Создать универсальный кодер — декодер с фиксированной конфигурацией связей возможно при условии только программного изменения алгоритмов обработки информации.

Эффективным решением этой задачи является использование нового класса программируемых схем — программируемых систолических структур (ПСС) [7]. Для реализации на основе ПСС любого вычислительного узла необходимо лишь определить кубические покрытия, которые определяют функционирование данного узла, и записать их в ПСС.

Если анализировать схему r -разрядной ЛПМ на основе матриц (3) и (5), легко видеть, что алгоритм функционирования такой

ЛПМ определяется только конфигурацией связей выходов всех триггеров со входом r -го триггера. Следовательно, с помощью ПСС можно вычислять функцию возбуждения только для r -го триггера, а остальные триггеры ЛПМ образуют обычный регистр сдвига. При использовании ЛПМ на основе матриц (4) и (6) необходимо предусмотреть вычисление функций возбуждения для всех триггеров с помощью r ПСС.

Аппаратные затраты для обоих типов ЛПМ можно значительно уменьшить, если использовать иную форму покрытий, чем в [7].

Введем такие кубические покрытия, которые были бы эквивалентны полиномиальной нормальной форме (ПНФ) булевой алгебры логических функций. Назовем полиномиальным D^n -покрытием кубической функции $Z()$ такое покрытие, в котором каждый куб соответствует одной импликанте ПНФ изоморфной ей логической функции $F()$ таким образом, что единичное (нулевое) значение компоненты куба соответствует прямому (инверсному) значению переменной в импликанте ПНФ, причем число компонент куба равно числу переменных логической функции $F()$. Аналогично можно определить полиномиальное R^n -покрытие, соответствующее инверсной логической функции $F()$.

Пример 2. Полиномиальной нормальной форме логической функции возбуждения третьего триггера схемы на рис.1 $F = u \oplus s_1 \oplus s_3$ соответствует

$$D^n = \begin{vmatrix} u & s_1 & s_2 & s_3 \\ 1 & x & x & x \\ x & 1 & x & x \\ x & x & x & 1 \end{vmatrix}. \quad (13)$$

Поскольку для ЛПМ каждый куб D^n -покрытия всегда будет иметь одну единицу, а остальные значения — x , можно записать сокращенное D^n -покрытие в виде одной строки, которая получается в результате операции пересечения всех его кубов. Для предыдущего примера

$$D_c^n = |1 \ 1 \ x \ 1|. \quad (14)$$

По аналогии с [7] выведем правило определения значений кубической функции на основе заданных полиномиальных покрытий. Входной набор W_i принадлежит к множеству наборов, на которых функция $Z(W_i)$ принимает значение 1 (0), если имеется нечетное (четное) число h непустых пересечений компонент набора W_i с компонентами D_c^n -покрытия этой функции:

$$Z(W_i) = \begin{cases} 1, & \text{если } w_j \cap d_j \neq 0 \text{ для} \\ & \text{нечетного числа } h, h = 1, 3, \dots; \\ 0, & \text{если } w_j \cap d_j \neq 0 \text{ для} \\ & \text{четного числа } h, h = 0, 2, 4, \dots, \end{cases} \quad (15)$$

где $W_i = \{w_1(i) \dots w_{r+1}(i)\}$, $D_c^n = \{d_1 \dots d_{r+1}\}$ $j = 1, \dots, r+1$.

Здесь операция пересечения кубов, в отличие от [7], определяется согласно таблице. Если символ x и символ \emptyset пустого пересечения закодировать нулем, то такая операция пересечения кубов может быть реализована с помощью логического элемента И. При систематической реализации ЛПМ набор W_i удобно представлять в формате $\{u(i+1) s_r(i) \dots s_1(i)\}$.

Пример 3. Определим очередное состояние $S(1)$, в которое перейдет ЛПМ (см. рис.1) из начального нулевого состояния $S(0)$ после подачи на ее вход воздействия $U(1) = 1$. В формате $\{u(1) s_3(0) s_2(0) s_1(0)\}$ набор W_1 равен: $W_1 = \{1 \ 0 \ 0 \ 0\}$. Тогда результаты операций пересечения набора W_1 с покрытием (14)

$$\begin{array}{r} 1 \ 0 \ 0 \ 0 \\ \cap \\ 1 \ 1 \ x \ 1 \\ \hline 1 \ 0 \ 0 \ 0 \end{array}$$

Поскольку имеется одно непустое пересечение, функция $Z(W_1)$ возбуждения третьего триггера равна единице. Значения выходов других триггеров определяются в результате операции сдвига соседних разрядов регистра. В итоге получаем искомое состояние, совпадающее с $S(1)$ в примере 1:

$$S(1) = \begin{vmatrix} s_1(1) \\ s_2(1) \\ s_3(1) \end{vmatrix} = \begin{vmatrix} 0 \\ 0 \\ 1 \end{vmatrix}.$$

Функционирование универсального кодера — декодера. Универсальный кодер — декодер циклического (n, k) -кода удобно рас-

d_1	W_1	Результат
1	0	\emptyset
1	1	1
x	0	\emptyset
x	1	\emptyset

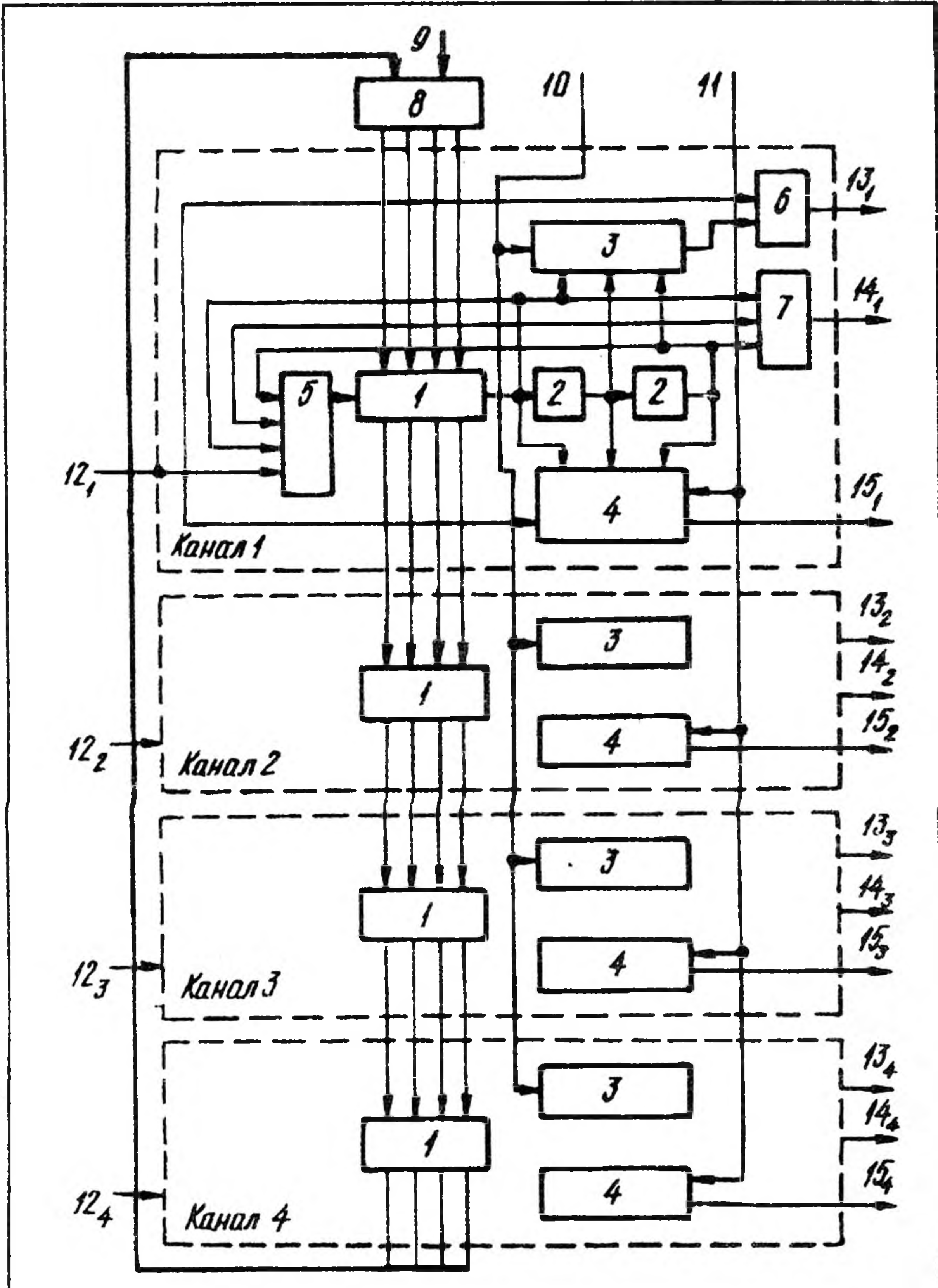


Рис. 3. Схема универсального кодера — декодера циклических кодов

смаивать в виде $r + 1$ -канального устройства (рис. 3). В каждом канале, содержащем ячейку 1, сдвиговый регистр 2, блок кодирования 3, блок декодирования 4 и переключатели 6—8, происходит обработка одного информационного или кодового вектора. Связывает каналы между собой информационный поток через схему ИЛИ 5 и ячейки 1, которые образуют одномерную ПСС.

Кодер — декодер может работать в трех режимах: программирования, кодирования и декодирования. В режиме программирования в ячейки 1 через группу входов 9 записывается D_c^n -покрытие, причем в каждую ячейку 1 записывается одна его компонента. Если число m компонент D_c^n -покрытия меньше, чем $r + 1$, то к нему добавляется $r + 1 - m$ -компонент, состоящих из значений x .

Все каналы устройства работают одинаково, только все вычисления в $i + 1$ -м канале происходят со сдвигом во времени на один цикл относительно вычислений в i -м канале.

Рассмотрим подробнее функционирование i -го канала ($i = 1, \dots, r + 1$). Ячейка 1, где вычисляется функция $Z(W_i)$ согласно формуле (15), и сдвиговый регистр 2 образуют ЛПМ первого типа. Перед началом режима кодирования в блок 3 по входу 11 также за-

писывается информация об элементах матрицы L'_r в виде вектора $Q(x)$:

$$Q(x) = \begin{pmatrix} 1 \\ l_{21} \\ l_{21} \oplus l_{31} \\ l_{21} \oplus l_{41} \\ \dots \end{pmatrix}.$$

В режиме кодирования в течение k макроциклов на вход 10_i поступают k разрядов информационного вектора $I_i(x)$, а затем в течение r макроциклов — r -разрядный нулевой вектор $O(x)$. В течение первых $r + 1$ циклов каждого макроцикла переключатели 6 и 7 находятся в положении Π_1 и на вход ячейки 1 поступают сигналы с последнего триггера регистра 2. В положении Π_2 переключатель 6 устанавливается в $r + 2$ -м цикле, когда на вход 10_i поступает очередной разряд векторов $I_i(x)$ или $O(x)$, а переключатель 7 — в $r + 3$ -м цикле каждого макроцикла. В течение макроцикла происходит циклическое перемещение по вертикали между ячейками 1 компонент D_c^n -покрытия и в итоге в конце $r + 2$ -го цикла в ячейке 1 вычисляется функция $Z(W_i)$. В последнем, $r + 3$ -м, цикле каждого макроцикла значение $Z(W_i)$ поступает на вход регистра 2 и блока 3, где вычисляются проверочные разряды кода согласно (12).

На выход 13_i устройства через переключатель 8 в течение k макроциклов передается исходный вектор $I_i(x)$, а затем в течение r макроциклов — проверочные разряды с выхода блока 3.

Перед началом режима декодирования, одновременно с записью в ПСС D_c^n -покрытия, в блок 4 по входу 12 записывается синдром ошибки $S^1(n)$. В режиме декодирования на входы блока 4 поступают кодовый вектор $C(x)$ и значение функции $Z(W_i)$. Если через n макроциклов будет получен нулевой синдром, тогда единичный сигнал на выходе 14_i свидетельствует об отсутствии ошибок в $C(x)$ и на выход 13_i через переключатель 8 вектор $C(x)$ передается без изменений. Для исправления одиночной ошибки на вход 10_i в течение последующих n макроциклов подается нулевой вектор $O(x)$, а с выхода 13_i поступает исправленный кодовый вектор.

В устройстве могут обрабатываться кодовые векторы с изменяющимися параметрами n , k и r .

Достоинства разработанного универсального кодера — декодера на основе кубического представления циклических кодов — это сочетание универсализма и аппаратной парал-

лельной обработки информации на уровне компонент кубов, т.е. на битовом уровне.

Изменение разрядности кодового вектора, увеличение или уменьшение корректирующей способности кода влекут за собой изменение порождающего полинома и соответствующего ему кубического покрытия. Запись этого покрытия в вычислительные ячейки устройства и представляет по сути настройку кодера — декодера к требуемым условиям передачи данных. Перепрограммирование устройства происходит очень быстро, поскольку не требуется его демонтажа и какого-либо физического воздействия (типа прожига связей в ЛПМ). В ПСС можно использовать и другие варианты кубических покрытий.

Предлагаемое устройство при меньших аппаратных затратах более гибкое и производительное, чем известный матричный декодер [8].

Самостоятельное значение имеет рассмотренная новая интерпретация циклического кодирования. Математический аппарат ЛПМ позволяет легко выбирать оптимальные алгоритмы кодообработки и их простую программную и аппаратную реализацию.

The cubic representation of cyclic codes and new interpretation of cyclic coding are considered on the base of the theory of linear sequential machine. The systolic implementation of universal coder — decoder is proposed that is able to re-adjust itself to given algorithms of code processing in real time.

1. Блейхут Р. Теория и практика кодов, контролирующих ошибки / Пер. с англ. — М.: Мир, 1986. — 576 с.
2. Шувалов В. П. Прием сигналов с оценкой их качества. — М.: Связь, 1979. — 200 с.
3. Жураковский Ю. П. Передача информации в ГАП: Учеб. пособие. — К.: Вища шк., 1991. — 216 с.
4. Питерсон У., Уэлдон Э. Коды, исправляющие ошибки / Пер. с англ. — М.: Мир, 1976. — 596 с.
5. Колесник В. Д., Мирончиков Е. Т. Декодирование циклических кодов. — М.: Связь, 1968. — 252 с.
6. Гилл А. Линейные последовательностные машины / Пер. с англ. — М.: Наука, 1974. — 288 с.
7. Семеренко В. П. Систолическая реализация кубических функций. // Электрон. моделирование. — 1992. — №1. — С. 21—25.
8. Муттер В. М., Петров Г. А., Маринкин В. И., Степанов В. С. Микропроцессорные кодеры и декодеры. — М.: Радио и связь, 1991. — 184 с.

Поступила 28.02.94

СЕМЕРЕНКО Василий Петрович, канд. техн. наук, доцент Винницкого государственного технического университета. В 1976 г. окончил Винницкий политехнический институт. Область научных исследований — параллельная обработка данных в различных системах: передачи данных, искусственного интеллекта, технической диагностики и др.