УДК 621.391.3:681.3.06

В.И. Кубицкий, канд. техн. наук
Всероссийский научно-исследовательский ин-т радиоаппаратуры
(Россия, 125212, Москва, Ленинградское шоссе, 43-а,
тел. +7(499) 1597847, e-mail: vkubitski@mail.ru),
И.А. Жуков, д-р техн. наук
Национальный авиационный университет
(Украина, 03680, Киев, пр-т Космонавта Комарова, 1,
тел. +38(044) 4977257, e-mail: zhuia@ukr.net)

Реализация параллельного метода кодирования кода Лагранжа

Разработаны алгоритмы программ и схемы устройств для реализации параллельного метода кодирования кодов Лагранжа. Определены операционная сложность алгоритмов, а также аппаратурная и временная сложности схем устройств кодирования.

Розроблено алгоритми програм та схеми пристроїв для реалізації паралельного методу кодування кодів Лагранжа. Визначено операційну складність алгоритмів, а також апаратурну та часову складність схем пристроїв кодування.

К л ю ч е в ы е с л о в а: *коды, кодирование, алгоритмы, схемы устройств.*

Для средств и систем, создаваемых на основе ЭВМ и каналов передачи данных, актуальна проблема достоверности информации, которая особенно важна при создании авиационных, космических, транспортных, энергетических и других систем. Искажение информации в таких системах ведет к снижению качества их функционирования, а в системах жизнеобеспечения — к снижению уровня безопасности. Поэтому требования к точности передачи, ввода, обработки и хранения информации в них значительно выше, чем в обычных производственных или коммерческих системах.

В компьютерных системах одной из наиболее актуальных проблем является проблема потерь и задержек пакетов. В системах передачи данных и сетевых приложениях, работающих в режиме реального времени, задержки пакетов равнозначны потерям, так как нет возможности приостановить процесс обработки, передачи и отображения данных в ожидании пакета или его повторной передачи. Кроме того, запросы на пересылку, особенно в топологии «один—многим», приводят к дополнительным

нагрузкам на канал передачи данных, вследствие чего уменьшается объем трафика в сети. При этом механизм обратной связи может дать положительный результат, если ошибки в пакетах или потери пакетов носят случайный характер. В противном случае, при систематических ошибках и потерях пакетов, восстановить потерянные пакеты в отведенное время с использованием обратной связи весьма затруднительно.

Поэтому в современных компьютерных сетях с большими объемами цифровых данных и жесткими ограничениями на максимально допустимую задержку данных в цепи отправитель—получатель необходимо применять методы обмена без использования обратных каналов. Это приводит к необходимости разработки методов защиты от ошибок, которые не предусматривают наличия обратного канала. Такие методы могут быть основаны на применении помехоустойчивого кодирования.

Существует множество кодов (CRC, Хэмминга, БЧХ, Рида—Соломона и др.), предназначенных для контроля и исправления ошибок в процессах передачи и хранения информации. Коды CRC (Cyclic Redundancy Code), применяемые в сетевых приложениях в качестве обнаруживающих кодов, способны обнаруживать как независимые битовые ошибки, так и серии ошибок в последовательно расположенных битах. Корректирующий код Хэмминга используют для исправления отдельных битовых ошибок в пакете, при этом обратный канал не используется.

Однако применение указанных кодов для кодирования пакетов не решает проблему восстановления всего потерянного пакета без использования обратного канала. Не могут решить эту проблему и другие классические помехоустойчивые коды: сверточные коды исправляют отдельные битовые ошибки, многие блоковые коды способны исправлять пачки ошибок в отдельно взятом пакете. Кроме того, рассмотренные коды не приспособлены для защиты данных при обработке в компьютерных системах.

В результате исследований установлено, что перспективными для применения в системах обработки, передачи и хранения информации являются полиномиальные коды Лагранжа [1]. Эти коды имеют параллельную структуру, обладают минимальной избыточностью и высокими свойствами побайтовой коррекции при передаче и хранении информации, а также обеспечивают защиту функциональных преобразований, логических и сдвиговых операций. Такие возможности кодов Лагранжа позволяют создавать подсистемы обеспечения помехозащищенности информации в компьютерной системе, используя меньше различных кодов, что исключает перекодирование при переходе между звеньями системы, имеющими разные коды.

Код Лагранжа можно использовать в качестве внешнего кода при двухуровневом каскадном кодировании в сети с коммутацией пакетов (например, с дейтаграммной передачей по каналу без обратной связи). При

этом внутренним кодом могут быть контрольная сумма, код CRC и другие коды, используемые в сети. В [2] описана общая процедура двухуровневого кодирования передаваемого сообщения. Для такого кодирования пакеты с ошибками при декодировании определяются декодированием внутреннего кода, потерянные и ошибочные пакеты отмечаются как стертые и декодируются внешним кодом. Код Лагранжа представляет собой блоковый код, в котором символы состоят из m бит. Поэтому его можно также использовать в сетях с коммутацией сообщений, кодируя сообщения, разбитые на блоки (элементы длины m).

Покажем, как можно реализовать кодирование кода Лагранжа, используя алгоритмы и схемы устройств кодирования для последовательной и параллельной передачи информации. Параллельная передача предполагает одновременную поэлементную передачу сообщения, т.е. каждый элемент (недвоичный) передается по отдельному каналу. Сложность кодирования определяется следующими параметрами:

1) операционная сложность — число операций (модульных) в конечном поле $GF(2^m)$: сложение ($N_{\oplus}$), умножение ($N_{\otimes}$), инвертирование ($N_{\sqcap}$);

2) аппаратурная сложность схем кодирования (S).

3) временная сложность схем кодирования (T) .

Параллельный метод кодирования кода Лагранжа и его реализация. Код Лагранжа определяется интерполяционной формулой Лагранжа [1]

$$f(\beta_j) = \sum_{i=0}^{s} f_i L_S^{(i)}(\beta_j), \quad j = \overline{1, r}, \tag{1}$$

где $f(\beta_j)$ — значения кодового полинома $f(x)$ в контрольных узлах, которые вычисляются с учетом значений этого полинома только в информационных узлах; f_i — значения кодового полинома $f(x)$ в информационных узлах; $L_S^{(i)}(\beta_j)$ — фундаментальные полиномы Лагранжа в контрольных узлах,

$$L_S^{(i)}(\beta_j) = -\prod_{l=1, l\neq j}^{r} \frac{x_i - \beta_l}{\beta_j - \beta_l} = -L_T^{(j)}(x_i);$$

$S = \{x_0, ..., x_s\}$ — множество информационных узлов мощности k; $T = \{\beta_1, ..., \beta_r\}$ —множество контрольных узлов мощности r.

Метод вычисления контрольных символов в соответствии с формулой (1) назван параллельным методом, а процедура вычисления — параллельным алгоритмом кодирования кода Лагранжа [3]. Рассмотрим два случая кодирования.

1. Необходимо вычислять коэффициенты Лагранжа (при $L^{(i)}(x) \neq \text{const}$). Такая необходимость возникает, например, при исправлении стираний, когда заранее неизвестно, в каких интерполяционных узлах потребуется вычислять значения кодового полинома. Поэтому коэффициенты $L^{(i)}(x)$ вычисляются только после определения местоположения стираний.

2. Узлы интерполирования фиксированы. Тогда значения $L^{(i)}(x)$ могут быть вычислены заранее и храниться в памяти как постоянные величины, которые можно использовать в процессе вычислений. В этом случае [4] выражение для вычисления значений полинома в контрольных узлах принимает вид

$$f(\beta_j) = \sum_{i=0}^{s} f_i A_{ij}, \quad j = \overline{1, r}, \tag{2}$$

где $A_{ij} = L_S^{(i)}(\beta_j) = \text{const}$.

Запишем формулу (1) в следующем виде:

$$f(\beta_1) = f_{s+1} = -\frac{1}{\prod_{l=2}^{r}(\beta_1 - \beta_l)} \sum_{i=0}^{s} f_i \prod_{l=2}^{r}(x_i - \beta_l) =$$

$$= -\frac{1}{\prod_{l=2}^{r}(\beta_1 - \beta_l)} \sum_{i=0}^{s} f_i \; a_{i2} \; a_{i3} \ldots a_{ir},$$

$$f(\beta_2) = f_{s+2} = -\frac{1}{\prod_{l=1,\; l\neq 2}^{r}(\beta_2 - \beta_l)} \sum_{i=0}^{s} f_i \prod_{l=1,\; l\neq 2}^{r}(x_i - \beta_l) =$$

$$= -\frac{1}{\prod_{l=1, l\neq 2}^{r}(\beta_2 - \beta_l)} \sum_{i=0}^{s} f_i \; a_{i1} \; a_{i3} \ldots a_{ir}, \tag{3}$$

. . . . . .

$$f(\beta_r) = f_{s+r} = -\frac{1}{\prod_{l=1}^{r-1}(\beta_r - \beta_l)} \sum_{i=0}^{s} f_i \prod_{l=1}^{r-1}(x_i - \beta_l) =$$

$$= -\frac{1}{\prod_{l=1}^{r-1}(\beta_r - \beta_l)} \sum_{i=0}^{s} f_i \; a_{i1} \; a_{i2} \ldots a_{i,r-1},$$

где $a_{i1} = x_i - \beta_1$; $a_{i2} = x_i - \beta_2$, ...; $a_{i,\,r-1} = x_i - \beta_{r-1}$; $a_{ir} = x_i - \beta_r$.

Введем обозначения:

$$-\frac{1}{\prod_{l=2}^{r}(\beta_1 - \beta_l)} = \gamma_1, \; -\frac{1}{\prod_{l=1, l\neq 2}^{r}(\beta_2 - \beta_l)} = \gamma_2, \ldots, \; -\frac{1}{\prod_{l=1}^{r-1}(\beta_r - \beta_l)} = \gamma_r.$$

На рис. 1 приведен следующий алгоритм кодирования, позволяющий определять значения кодового полинома в контрольных узлах в соответствии с формулами (3):

1. Для каждого $i = \overline{0, s}$:

а) задавая $l = \overline{1, r}$, вычисляем величины $a_{il} = x_i - \beta_l$ и запоминаем их;

б) для каждого $j = \overline{1, r}$ определяем произведение $\alpha_{ji} = f_i \prod_{l\neq j} a_{il}$ и сумму $\sum_{i=0}^{s} f_i \prod_{l\neq j} a_{il}$.

2. Для каждого $j = \overline{1, r}$ вычисляем $\gamma_j = -1/\prod_{l=1, l\neq j}^{r} (\beta_j - \beta_l)$ и f_{s+j}.

Число операций в конечном поле $GF(2^m)$ для параллельного алгоритма кодирования при $L^{(i)}(x) \neq \text{const}$ составляет:

$$N_{\oplus} = \begin{cases} 2r(n-1) - r^2 & \text{при } r > 1, \\ n-2 & \text{при } r = 1, \end{cases} \quad N_{\otimes} = r(r-1)(n-r+1), \; N_{\boxed{\text{г}}} = r.$$

Здесь $n = k + r$ — длина кодового сообщения. Число операций определялось с учетом наличия r регистров памяти для хранения величин a_{il}. С учетом наличия памяти для хранения n узлов интерполирования необходимо иметь $(n + r)$ регистров.

На рис. 2 приведены схемы устройств кодирования при последовательном и параллельном поступлении информации на устройства кодирования. На рис. 2, *а*, ключи в исходном состоянии замкнуты на контакты *а*. После поступления последнего информационного символа и соответствующих вычислений ключи размыкаются и поочередно с приходом каждого значения β_j переключаются на контакты b_j. Далее выполняется определение коэффициентов γ_j и окончательное вычисление контрольных символов

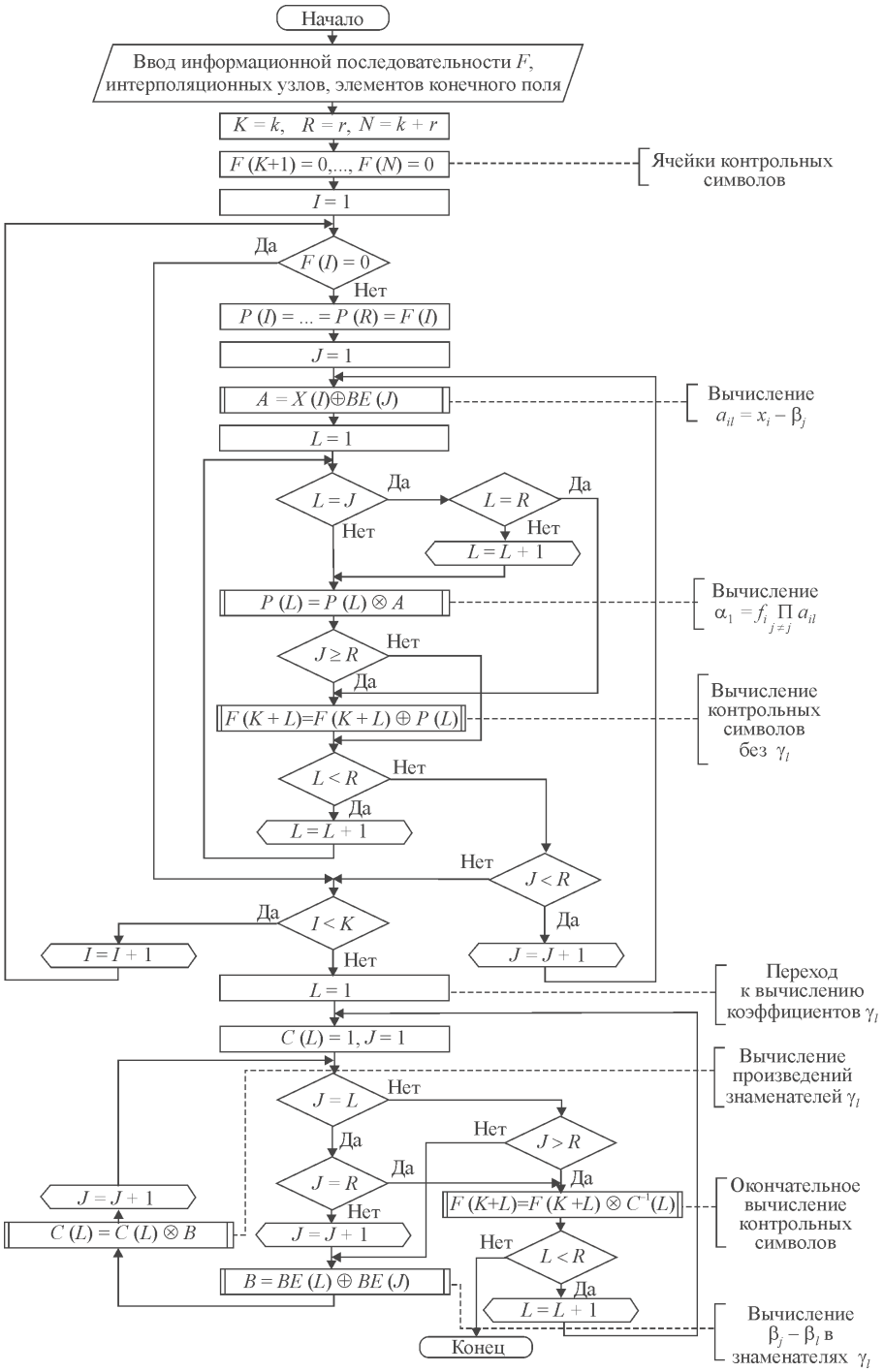


Рис. 1. Блок-схема параллельного алгоритма кодирования при $L^{(i)}(x) \neq \text{const}$

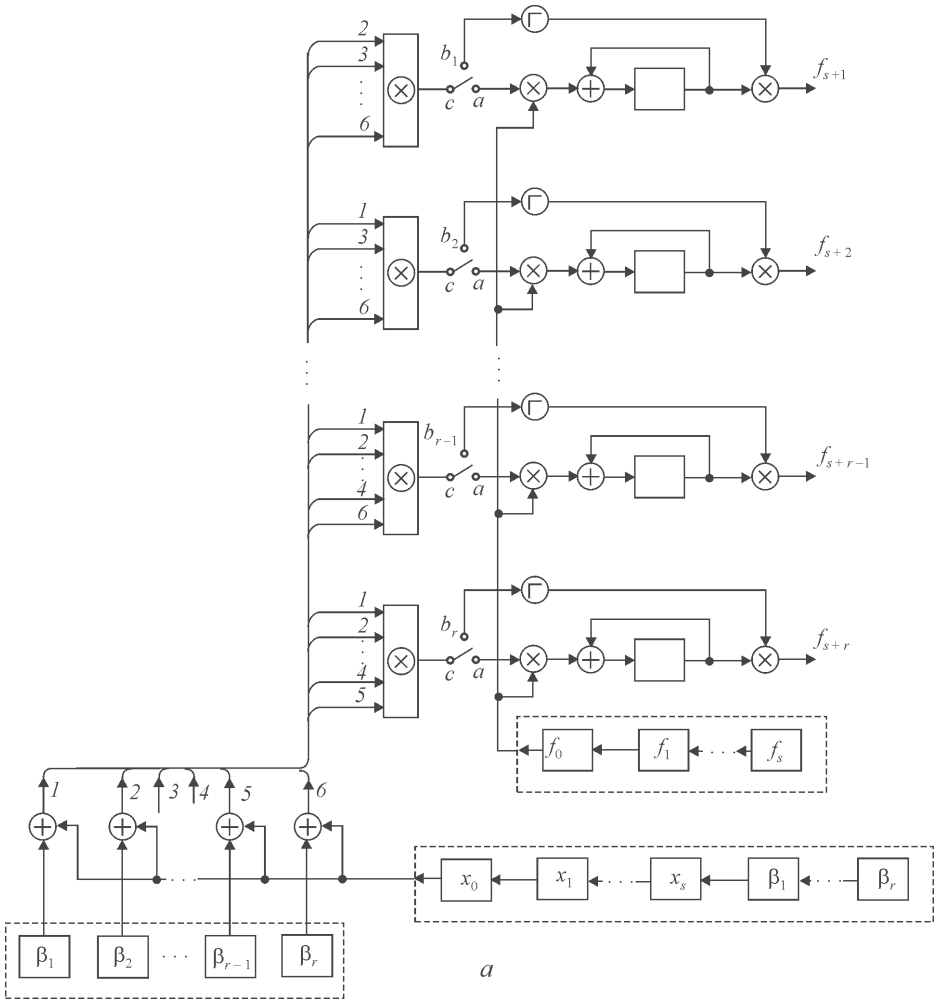


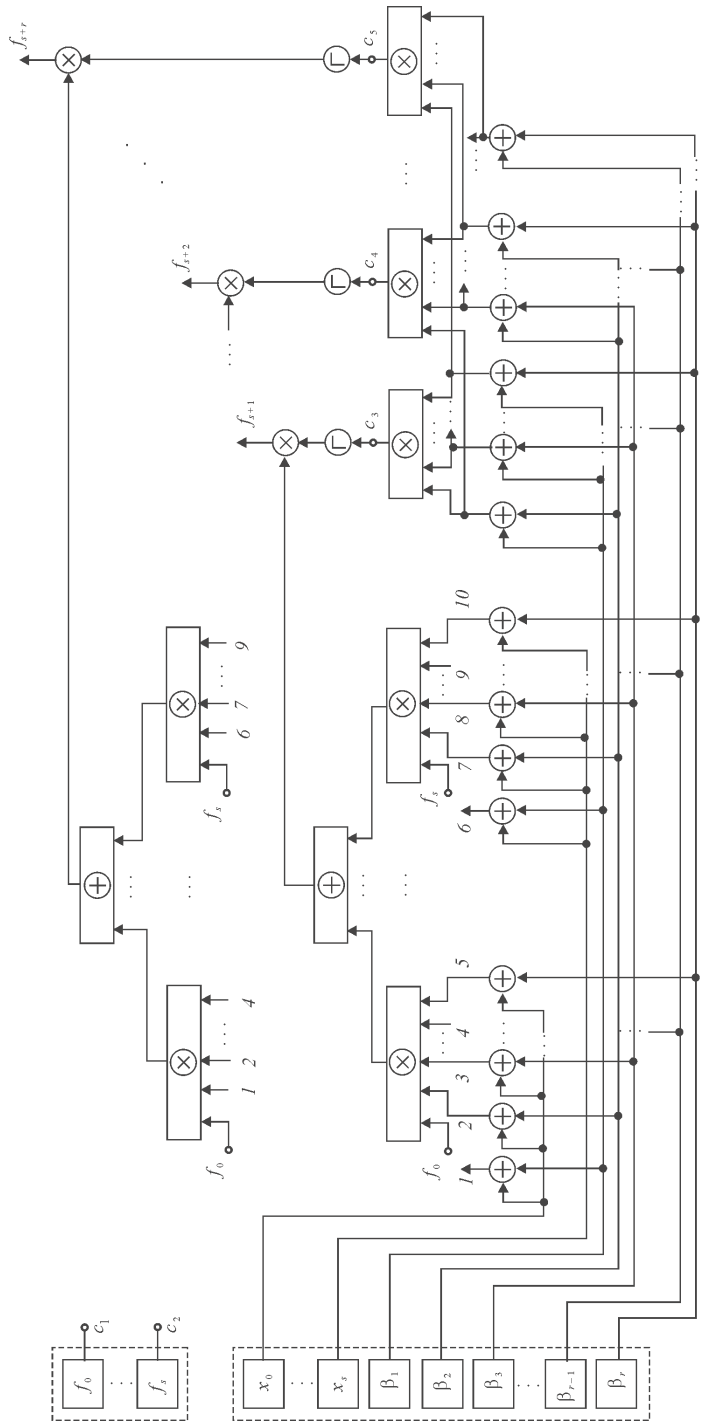
Рис. 2. Схемы устройства кодирования для параллельного алгоритма при последовательном (*а*) и параллельном (*б*) поступлении информации ($L^{(i)}(x) \neq \text{const}$) (см. также с. 74)

f_{s+j} поочередно, начиная с f_{s+1} (после поступления β_1) и заканчивая f_{s+r} (после поступления β_r). Вычисление контрольных символов f_{s+j} (см. рис. 2, *б*) выполняется одновременно.

Аппаратурная сложность $S_{\text{пр}}$ схем кодирования ($L^{(i)}(x) \neq \text{const}$) при параллельном методе следующая:

для схемы, представленной на рис. 2, *а* ,

$$S_{\text{пр}} = (2r+n)\, s_{\text{я}} + 2rs_{\text{с}} + r^2 s_{\text{у}} + rs_{\text{и}} + rs_{\text{к}};$$



б

Рис. 2. Окончание

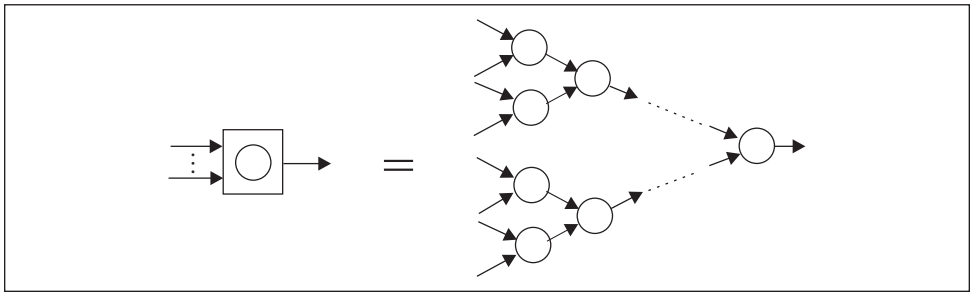


Рис. 3. Схема соединения модулей в блоках сумматора и умножителя

для схемы, представленной на рис. 2, *б*,

$$S'_{\text{пр}} = n\, s_{\text{я}} + [4n - 3(r+1)]\, rs_{\text{с}}/2 + (r-1)(n-r+1)\, rs_{\text{у}} + rs_{\text{и}}.$$

Здесь $s_{\text{с}}$ — сложность модуля сложения двух величин; $s_{\text{у}}$ — сложность модуля умножения двух величин; $s_{\text{я}}$ — сложность ячейки памяти; $s_{\text{и}}$ — сложность модуля инвертирования; $s_{\text{к}}$ — сложность ключа.

Время кодирования (при $L^{(i)}(x) \neq \text{const}$) составляет:

для схемы, представленной на рис. 2, *а*,

$$T_{\text{пр}} = (2n - r - 1)\, t_{\text{я}} + (2n - r)\, t_{\text{с}} + n\,(\lceil \log_2(r-1) \rceil + 1)\, t_{\text{у}} + rt_{\text{и}} + nt_{\text{к}};$$

для схемы, представленной на рис. 2, *б*,

$$T'_{\text{пр}} = (\lceil \log_2 r \rceil + 1)\, t_{\text{у}} + (\lceil \log_2(n-r) \rceil + 1)\, t_{\text{с}} \text{ при } \lceil \log_2 k \rceil\, t_{\text{с}} \geq t_{\text{и}},$$

$$T'_{\text{пр}} = (\lceil \log_2 r \rceil + 1)\, t_{\text{у}} + t_{\text{с}} + t_{\text{и}} \text{ при } \lceil \log_2 k \rceil\, t_{\text{с}} \leq t_{\text{и}}.$$

Здесь $t_{\text{с}}$ — время сложения двух величин; $t_{\text{у}}$ — время умножения двух величин; $t_{\text{и}}$ — время инвертирования; $t_{\text{я}}$ — время задержки ячейки памяти; $t_{\text{к}}$ — время задержки ключа; $\lceil A \rceil$ — ближайшее к A целое число, большее или равное A.

Следует заметить, что каждый из блоков сумматоров и умножителей имеет древовидную структуру соединения модулей (рис. 3), что уменьшает время вычисления в этих блоках по сравнению с последовательной схемой соединения модулей. Время выполнения операций в блоке составляет $\lceil \log_2 m \rceil\, t$, где m — число входов в блок, t — время выполнения операции в модуле.

Блок-схема параллельного алгоритма кодирования при $L^{(i)}(x) = \text{const}$, реализующая выражение (2), приведена на рис. 4, а блок-схема алгоритма вычисления величин $L^{(i)}(x)$ — на рис. 5.

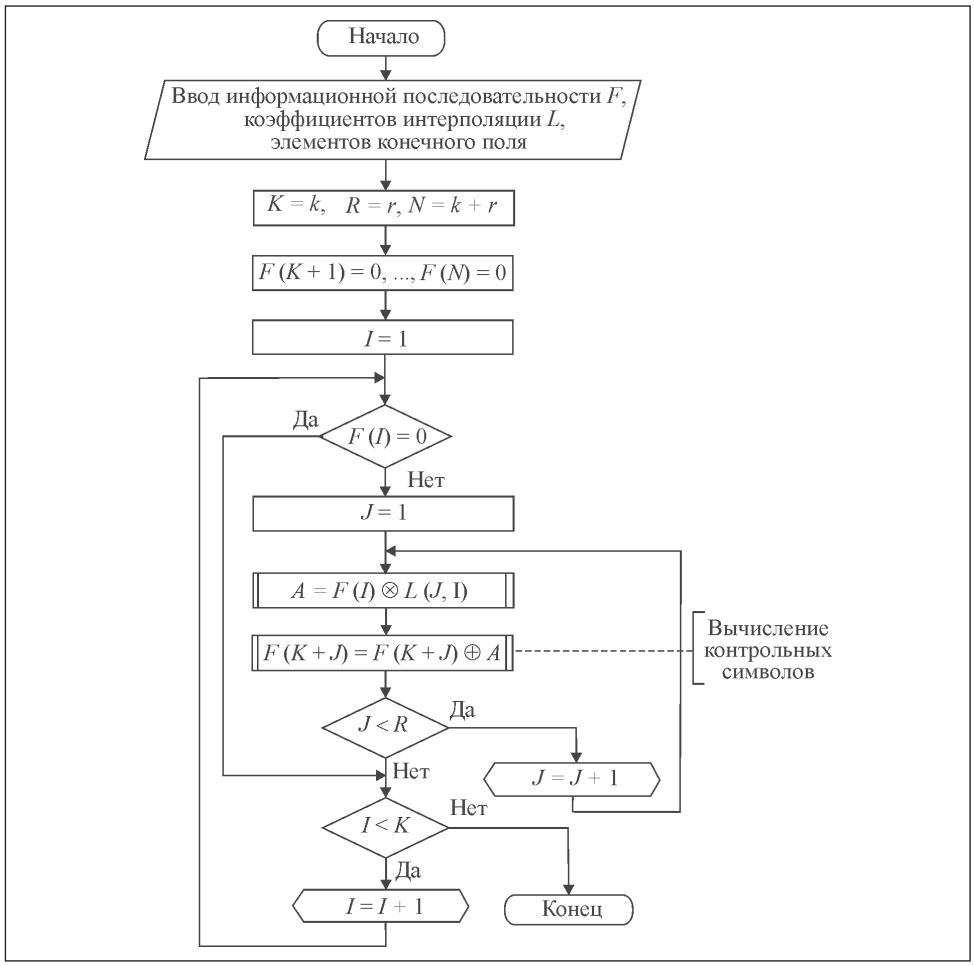


Рис. 4. Блок-схема параллельного алгоритма кодирования при $L^{(i)}(x)=\text{const}$

Число операций в поле $GF(2^m)$ для параллельного алгоритма кодирования при $L^{(i)}(x)=\text{const}$ следующее:

$$N_{\oplus}=(n-r-1)\,r,\; N_{\otimes}=\begin{cases}(n-r)\,r & \text{при r}>1,\\ 0 & \text{при r}=1.\end{cases}$$

Для хранения коэффициентов $L^{(i)}(x)$ требуется $r\,(n-r)$ регистров.

Схемы устройств кодирования при последовательном и параллельном поступлении информации представлены на рис. 6. Вычисление контрольных символов f_{s+j} на этих схемах выполняется одновременно.

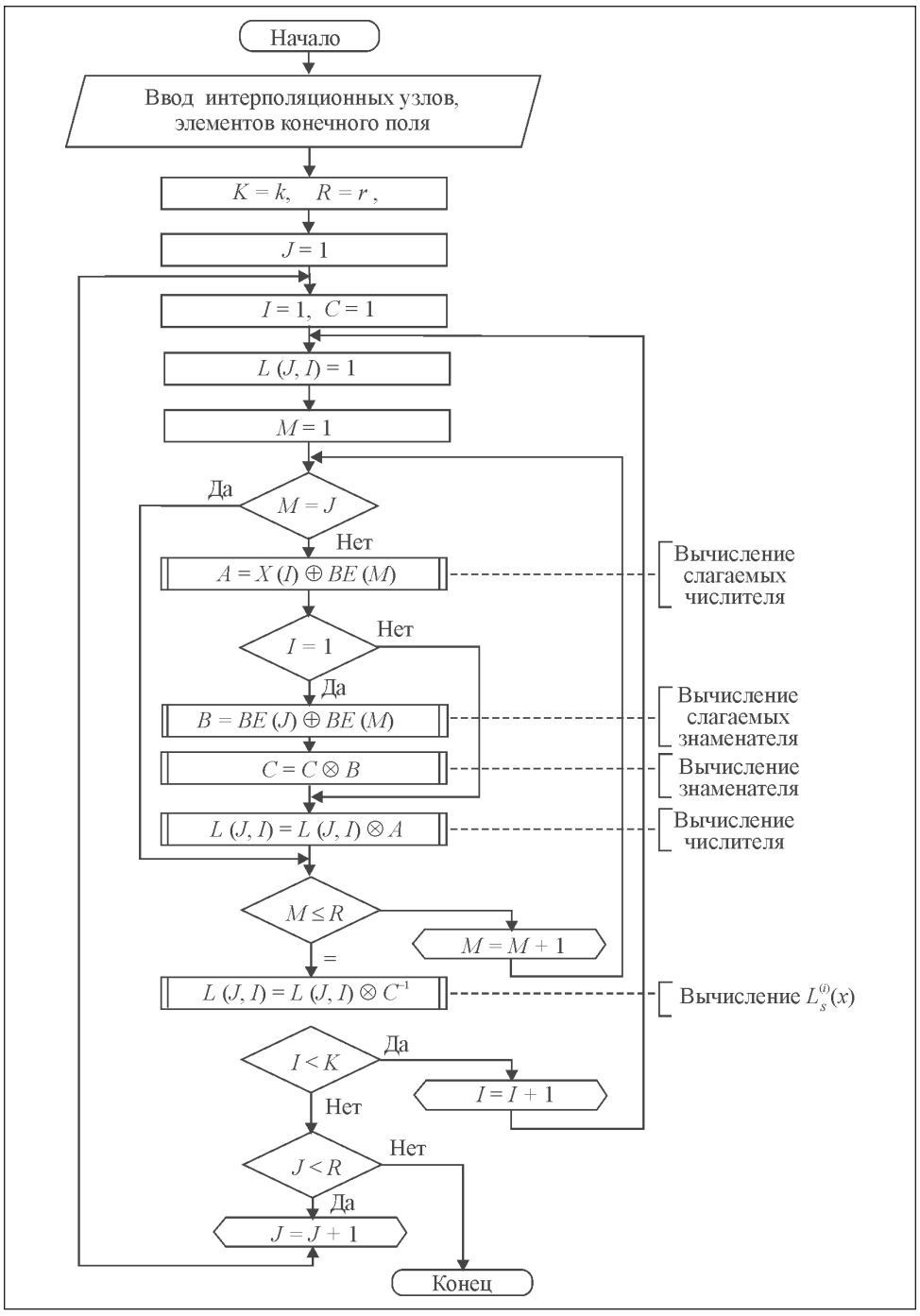


Рис. 5. Блок-схема алгоритма вычисления $L^{(i)}(x)$ при параллельном методе кодирования

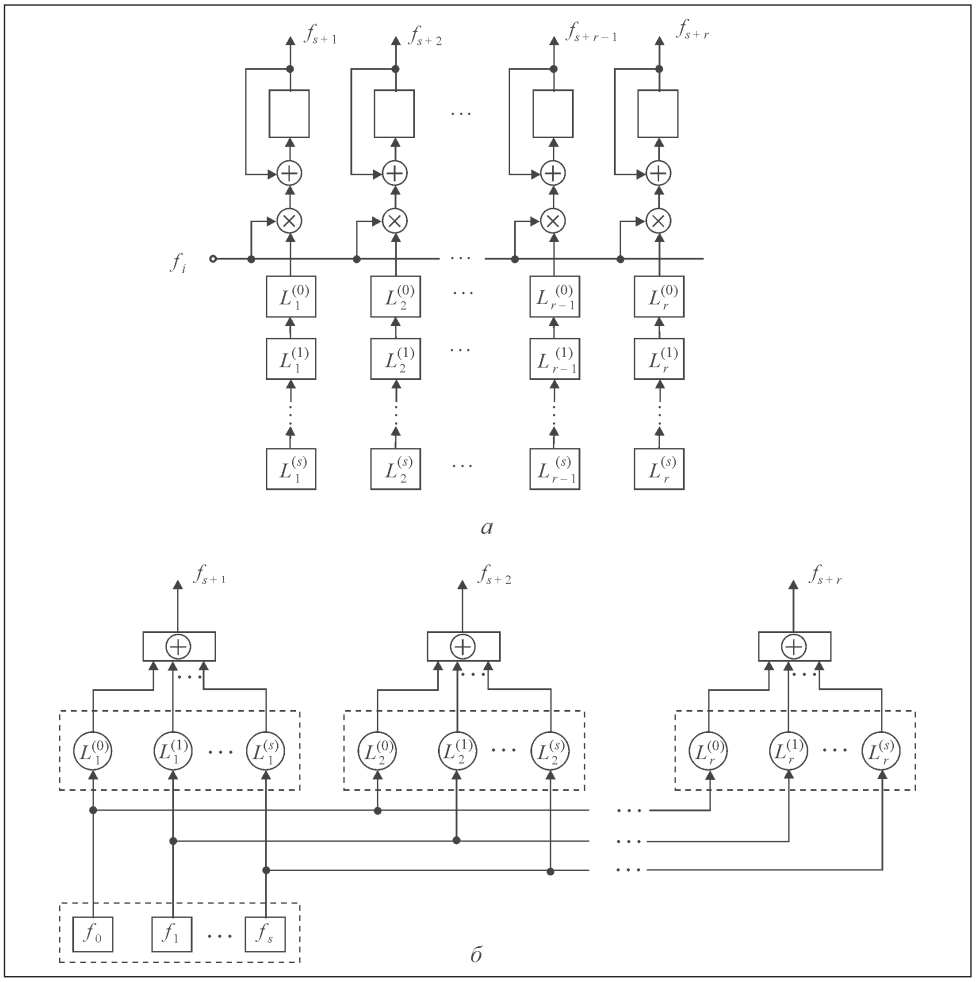


Рис. 6. Схемы устройств кодирования параллельным методом при последовательном (*а*) и параллельном *(б)* поступлении информации ($L^{(i)}(x)=\text{const}$)

Аппаратурная сложность схем при $L^{(i)}(x)=\text{const}$ следующая:
для схемы, приведенной на рис. 6, *а*,

$$S_{\text{пр}} = r\left[(n-r+1)\,s_{\text{я}} + s_{\text{у}} + s_{\text{с}}\right];$$

для схемы, приведенной на рис. 6, *б*,

$$S'_{\text{пр}} = r\left[(n-r)(s_{\text{п}} + s_{\text{с}}) - s_{\text{с}}\right],$$

где $s_{\text{п}}$ — сложность модуля умножения на постоянную величину.

Время кодирования составляет:
для схемы, приведенной на рис. 6, *а*,

$$T_{\text{пр}} = (n-r)\,t_{\text{у}} + t_{\text{с}} + 2t_{\text{я}}) - t_{\text{я}};$$

для схемы, приведенной на рис. 6, *б*,

$$T'_{\text{пр}} = t_{\text{п}} + \lceil \log_2(n-r) \rceil\, t_{\text{с}},$$

где $t_{\text{п}}$ — время умножения на постоянную величину.

Установлено, что время кодирования для схем, реализующих параллельный метод, меньше, чем для схем, реализующих последовательный метод кодирования кода Лагранжа.

Выводы

Разработанные алгоритмы и схемы могут быть использованы не только для кодирования информации в компьютерных системах, но и для вычисления контрольных символов при декодировании [5]. Схемы устройств кодирования имеют блочную наращиваемую структуру и могут быть адаптируемы к длине сообщения и количеству ошибок. Предложенные программные алгоритмы и схемы устройств кодирования реализованы с использованием положений арифметики в конечных полях [6, 7].

Полученные аналитические выражения сложности разработанных алгоритмов и схем устройств кодирования, реализующих параллельный метод кодирования кодов Лагранжа, позволят проводить сравнение и выбирать лучшие методы, алгоритмы и схемы кодирования по необходимым параметрам (числу модульных операций, аппаратурным затратам, времени вычисления).

Algorithms of programs and schemes of devices for implementation of the parallel method of encoding the Lagrange codes have been designed. Operational complexity of algorithms as well as the instrumental and temporal complexity of schemes have been determined.

СПИСОК ЛИТЕРАТУРЫ

1. *Амербаев В.М.* Теоретические основы машинной арифметики. — Алма-Ата: Наука, 1976. — 324 с.
2. *Кубицкий В.И.* Восстановление стертых пакетов в компьютерных сетях // Науч. вестн. МГТУ ГА. — 2011. — № 169 (7). — С. 65— 72.
3. *Кубицкий В.И.* Исправление стираний и ошибок кодами Лагранжа // Управление-82: Московская науч.-техн. конф. молодых специалистов и ученых. Март 1982 г.: Материалы конф. Ч. 2. — М. : ВНИИПОУ, 1982. — С. 125— 130.
4. *Кубицкий В.И.* Процедуры кодирования и декодирования для полиномиальных кодов // Эксплуатация программного обеспечения систем реального времени, построенных на базе микро- и мини-ЭВМ: Сб. науч. тр. — Киев: КИИГА, 1989. — С. 67— 71.

5. *Жуков И.А., Кубицкий В.И.* Вычисление синдромов при декодировании кодов Лагранжа // Методи та засоби кодування, захисту й ущілнення інформації: Третя міжнар. наук.-практ. конф., 20—22 квітня 2011 р. Тези допов. — Вінниця: ВНТУ, 2011. — С. 24— 25.
6. *Жуков И.А., Кубицкий В.И., Дрововозов В.И.* Алгоритмы выполнения операций над элементами конечного поля *GF* (2^m) в вычислительных устройствах // Авіа — 2007: VIII Міжнар. наук.-техн. конф., 25—27 квітня 2007 р. Матеріали конф. Т. 1. — Київ : НАУ, 2007.— С. 13.5—13.8.
7. **Пат. № 43629 Украина,** МПК (2009) H03M 7/00. Пристрій для множення елементів скінченних полів *GF* (2^n) / Жуков И.А., Кубицкий В.И., Синельников А.А. — № u 2009. 02754. Опубл. 25.08.2009, Бюл. № 16.

Поступила 26.12.13;
после доработки 05.05.14

КУБИЦКИЙ Валерий Иванович, канд. техн. наук, зам. директора научно-технического центра Всероссийского научно-исследовательского ин-та радиоаппаратуры. В 1973 г. окончил Киевский ин-т инженеров гражданской авиации. Область научных исследований — проектирование, помехоустойчивость компьютерных систем и сетей.

ЖУКОВ Игорь Анатольевич, д-р техн. наук, профессор, зав. кафедрой компьютерных систем и сетей Национального авиационного университета Украины. В 1972 г. окончил Киевский ин-т инженеров гражданской авиации. Область научных исследований — разработка и исследование методов и аппаратно-программных средств повышения производительности вычислительных структур, систем и компьютерных сетей для решения задач большой размерности и систем реального времени.