УДК 621.391.251:681.3.06

Кубицкий В.И.

ВЫЧИСЛЕНИЕ СИНДРОМОВ ПРИ ДЕКОДИРОВАНИИ МНОГОКРАТНЫХ ОШИБОК КОДАМИ ЛАГРАНЖА

ГосНИИ «Аэронавигация» (Россия, Москва)

Приведены процедуры вычисления величины синдромов при декодировании многократных ошибок кодами Лагранжа. Определено количество операций в конечных полях и объёмы памяти для хранения постоянных величин при реализации этих процедур. Дана сравнительная оценка рассмотренных процедур

При декодировании кодов Лагранжа необходимо решать ключевые уравнения синдромов. При этом одной из основных задач является вычисление значений синдромов. В [1] выводятся выражения для определения величин синдромов кодов Лагранжа при декодировании ошибок.

В данной работе рассматриваются процедуры вычисления синдромов в соответствии с этими выражениями.

Вычисление величин синдромов

Величины синдромов Q_μ для случаев применения параллельного (А1), последовательного (А2) и параллельно-последовательного (А4) алгоритмов кодирования вычисляются из выражений [1]:

$$Q_\mu = \sum_{\beta_j \in T} R_j \beta_j^\mu, \quad \mu = \overline{0, r-1}, \quad j = \overline{1, r}, \quad (1)$$

где $R_j = \tilde{f}(\beta_j) - f^*(\beta_j)$;

$\tilde{f}(\beta_j)$ – контрольные символы полученного (принятого) кодового сообщения;

$f^*(\beta_j)$ – вычисленные с использованием алгоритмов кодирования контрольные символы;

$T = \{\beta_1, \Lambda, \beta_r\}$ – множество контрольных узлов.

Вычисление величин синдромов Q_μ по формуле (1) можно производить в соответствии со следующими процедурами, в которых для каждого $j = \overline{1, r}$ величины μ принимают значения $\mu = \overline{0, r-1}$.

1. Параллельная (независимая) процедура:

$$Q_\mu = \sum_{\beta_j \in T} R_j B_\mu^{(j)}, \quad (2)$$

где $B_\mu^{(j)} = \beta_j^\mu = const$, $B_0^{(j)} = 1$.

2. Последовательная процедура:

$$Q_\mu = \sum_{\beta_j \in T} Q_\mu^{(j)}, \quad (3)$$

где $Q_\mu^{(j)} = Q_{\mu-1}^{(j)} \beta_j = const$, $Q_0^{(j)} = R_j$.

Количество модульных операций для этих процедур будет следующим:

$$N_\oplus = N_\otimes = r(r-1),$$

где $\oplus$ – модульная операция сложения,

$\otimes$ – модульная операция умножения,

Кроме того, для хранения величин β_j^μ и β_j при вычислении Q_μ с применением алгоритмов А1, А2, А4 необходимо иметь ячеек памяти соответственно:

$r(r-1)$ – для параллельной (независимой) процедуры (2),

r – для последовательной процедуры (3).

Если $\beta_{r-1} = 1$ и $\beta_r = 0$, то имеем:

$$N_\oplus = (r-1)^2,$$

$$N_\otimes = (r-1)(r-2).$$

При этом ячеек памяти для хранения величин β_j^μ и β_j потребуется соответственно в количестве:

$(r-1)(r-2)$ – для параллельной (независимой) процедуры (2),

$(r-2)$ – для последовательной процедуры (3).

При использовании последовательного алгоритма А3 вычисление величин Q_μ производится в соответствии с выражениями [1]:

1. Для произвольного (не фиксированного) набора узлов интерполирования

$$Q_\mu = R_j\left[\beta_j^\mu + \sum_{m=1}^{\mu}\phi_m^{(\mu)}\beta_j^{\mu-m}\right] - \sum_{m=1}^{\mu}\phi_m^{(\mu)}Q_{\mu-m},$$

$$\mu = r - j, \qquad (4)$$

где $\phi_o^{(\mu)} = 1$, $\phi_1^{(\mu)} = -\sum_{z=r-\mu+1}^{r}\beta_z$,K ,

$$\phi_\mu^{(\mu)} = (-1)^\mu \prod_{z=r-\mu+1}\beta_z .$$

2. Для фиксированного набора узлов интерполирования

$$Q_\mu = \lambda_j R_j - \sum_{m=1}^{\mu}\phi_m^{(\mu)}Q_{\mu-m}, \quad \mu = r - j, \quad (5)$$

где $\lambda_j = \beta_j^\mu + \sum_{m=1}^{\mu}\phi_m^{(\mu)}\beta_j^{\mu-m} = const$ и $\phi_m^{(\mu)} = const$.

Для синтеза схем устройств вычисления величин Q_μ выражение (4) запишем в виде:

$$\begin{aligned}
Q_0 &= R_r\\
Q_1 &= \left[\beta_{r-1} + \phi_1^{(1)}\right]R_{r-1} - \phi_1^{(1)}Q_0\\
Q_2 &= \left[(\beta_{r-2} + \phi_1^{(2)})\beta_{r-2} + \phi_2^{(2)}\right]R_{r-2} - \left[\phi_1^{(2)}Q_1 + \phi_2^{(2)}Q_0\right]\\
Q_3 &= \left[((\beta_{r-3} + \phi_1^{(3)})\beta_{r-3} + \phi_2^{(3)})\beta_{r-3} + \phi_3^{(3)}\right]R_{r-3} -\\
&\quad -\left[\phi_1^{(3)}Q_2 + \phi_2^{(3)}Q_1 + \phi_3^{(3)}Q_0\right]\\
&\text{M}\\
Q_{r-2} &= [(\text{K}\,((\beta_2 + \phi_1^{(r-2)})\beta_2 + \phi_2^{(r-2)})\beta_2 + \text{K} +\\
&\quad + \phi_{r-3}^{(r-2)})\beta_2 + \phi_{r-2}^{(r-2)}]R_2 -\\
&\quad -\left[\phi_1^{(r-2)}Q_{r-3} + \text{K} + \phi_{r-3}^{(r-2)}Q_1 + \phi_{r-2}^{(r-2)}Q_0\right]\\
Q_{r-1} &= [(\text{K}\,((\beta_1 + \phi_1^{(r-1)})\beta_1 + \phi_2^{(r-1)})\beta_1 + \text{K} +\\
&\quad + \phi_{r-2}^{(r-2)})\beta_1 + \phi_{r-1}^{(r-1)}]R_1 -\\
&\quad -\left[\phi_1^{(r-1)}Q_{r-2} + \text{K} + \phi_{r-2}^{(r-2)}Q_1 + \phi_{r-1}^{(r-1)}Q_0\right].
\end{aligned} \qquad (6)$$

Выражение (5) можно представить следующим образом:

$$\begin{aligned}
Q_0 &= R_r\\
Q_1 &= \lambda_{r-1}R_{r-1} - \phi_1^{(1)}Q_0\\
Q_2 &= \lambda_{r-2}R_{r-2} - \left[\phi_1^{(2)}Q_1 + \phi_2^{(2)}Q_0\right]\\
Q_3 &= \lambda_{r-3}R_{r-3} - \left[\phi_1^{(3)}Q_2 + \phi_2^{(3)}Q_1 + \phi_3^{(3)}Q_0\right]\\
&\text{M}\\
Q_{r-2} &= \lambda_2 R_2 - \left[\phi_1^{(r-2)}Q_{r-3} + \text{K} + \phi_{r-2}^{(r-2)}Q_0\right]\\
Q_{r-1} &= \lambda_1 R_1 - \left[\phi_1^{(r-1)}Q_{r-2} + \text{K} + \phi_{r-1}^{(r-1)}Q_0\right].
\end{aligned} \qquad (7)$$

Вычисления Q_μ в соответствии с выражениями (6) и (7) предполагают, что при определении каждого следующего значения Q известны все предыдущие. Т.е. вычисления производятся согласно последовательной процедуре.

Покажем, как можно производить параллельное (независимое) вычисление величин синдрома Q_μ для фиксированного набора узлов интерполирования.

Для этого в (7) вместо $Q_{\mu-m}$ сделаем соответствующие подстановки и, выполнив преобразования, получим:

$$\begin{aligned}
Q_0 &= R_r\\
Q_1 &= \lambda_r^{(1)}R_r + \lambda_{r-1}^{(1)}R_{r-1}\\
Q_2 &= \lambda_r^{(2)}R_r + \lambda_{r-1}^{(2)}R_{r-1} + \lambda_{r-2}^{(2)}R_{r-2}\\
&\text{M}\\
Q_{r-2} &= \lambda_r^{(r-2)}R_r + \lambda_{r-1}^{(r-2)}R_{r-1} + \text{K} + \lambda_2^{(r-2)}R_2\\
Q_{r-1} &= \lambda_r^{(r-1)}R_r + \lambda_{r-1}^{(r-1)}R_{r-1} + \text{K} + \lambda_1^{(r-1)}R_1
\end{aligned} \qquad (8)$$

или

$$Q_\mu = \sum_{h=0}^{\mu}\lambda_{r-h}^{(\mu)}R_{r-h},$$

где коэффициенты $\lambda_{r-h}^{(\mu)} = g(\beta_r, \text{K}\,, \beta_{r-h})$ являются функциями от значений контрольных узлов, могут быть вычислены заранее и храниться в памяти как постоянные коэффициенты, используемые в процессе вычислений.

Вычисление Q_μ в соответствии с (7) и (8) при использовании алгоритма А3 требует выполнения модульных операций в количестве:

$$N_\oplus = r(r-1)/2,$$

$$N_{\otimes} = (r+2)(r-1)/2 .$$

Для хранения коэффициентов $\lambda_{r-h}^{(\mu)}$ (λ_j и $\phi_m^{(\mu)}$) необходимо иметь $(r+2)(r-1)/2$ ячеек памяти.

Из (7) видно, что при $\beta_r = 0$ произведение $\phi_{\mu}^{(\mu)} Q_0 = 0$, такт как $\phi_{\mu}^{(\mu)} = (-1)^{\mu} \prod_{z=r-\mu+1} \beta_z = 0$. Тогда при $\beta_{r-1} = 1$ и $\beta_r = 0$:

$$N_{\oplus} = (r-1)(r-2)/2 ,$$

$$N_{\oplus} = (r+1)(r-2)/2 .$$

При этом количество ячеек памяти для хранения коэффициентов $\lambda_{r-h}^{(\mu)}$ равно $(r+1)(r-1)/2$.

Если строить схемы устройств для реализации вычисления величин Q_{μ} в соответствии с (2), (3), (6), (7), (8), то они легко адаптируемы к количеству ошибок. Для этого необходимо с изменением количества ошибок (и невязок R_j) исключить или добавить соответствующее количество функциональных блоков. При этом вычисленные ранее величины β_j^{μ}, $\lambda_{r-h}^{(\mu)}$, λ_j и $\phi_m^{(\mu)}$ сохраняют свои значения.

Вычисление невязок

При вычислении значений синдромов Q_{μ} необходимо также вычислять величины невязок:

$$R_j = \tilde{f}(\beta_j) - f^*(\beta_j), \quad j = \overline{1,r} ,$$

где $\tilde{f}(\beta_j)$, $f^*(\beta_j)$ – принятые и вновь вычисленные контрольные символы соответственно.

Величины $f^*(\beta_j)$ определяются с использованием алгоритмов кодирования А1÷А4 [1].

а. *Параллельный алгоритм А1*:

$$f^*(\beta_j) = -\sum_{i=0}^{s} \tilde{f}_i \, L_S^{(i)}(\beta_j), \quad j = \overline{1,r} ,$$

где $\tilde{f}_i$ – информационные символы принятой кодовой последовательности;

$L_S^{(i)}(\beta_j) = -\prod_{\substack{l=1 \\ l \neq j}}^{r} \frac{x_i - \beta_l}{\beta_j - \beta_l}$ – фундаментальные полиномы Лагранжа в контрольных узлах;

$S = \{x_o, \Lambda, x_s\}$ – множество информационных узлов.

б. *Последовательный алгоритм А2*:

$$f^*(\beta_j) = [\sum_{i=0}^{s} \tilde{f}_i + \sum_{\substack{h=1 \\ h<j}}^{j-1} f^*(\beta_h)] \, L_{S_{j-1}}^{(i)}(\beta_j) ,$$

где $\tilde{f}_i$ – информационные символы принятой кодовой последовательности;

$f^*(\beta_h)$ – вычисленные значения полинома в предыдущих (j-1)-ых контрольных узлах;

$L_{S_{j-1}}^{(i)}(\beta_j) = -\prod_{l=j+1}^{r} \frac{x_i - \beta_l}{\beta_j - \beta_l}$ – фундаментальные полиномы Лагранжа в контрольных узлах;

$$S_{j-1} = S \, \mathrm{Y} \, \{\beta_1, \Lambda, \beta_{j-1}\}; \quad x_i \in S_{j-1} .$$

в. *Последовательный алгоритм А3*:

$$f^*(\beta_j) = \sum_{i=0}^{s+j-1} \tilde{f}_i \, L_{S_{j-1}}^{(i)}(\beta_j) ,$$

где $\tilde{f}_i$ – символы принятой кодовой последовательности, включающие информационные символы и принятые значения $\tilde{f}(\beta_{j-1})$ в (j-1)-ых контрольных узлах.

г. *Параллельно-последовательный алгоритм А4*:

$$f^*(\beta_j) = -\sum_{i=0}^{s} \tilde{f}_i \, L_S^{(i)}(\beta_j), \quad j = \overline{1, r-1} ,$$

$$f^*(\beta_r) = -[\sum_{i=0}^{s} \tilde{f}_i + \sum_{j=1}^{r-1} f^*(\beta_j)] .$$

Очевидно, что для вычисления R_j количество модульных операций умножения такое же, как для соответствующих алгоритмов кодирования, а количество

операций модульного сложения больше на величину r.

Если узлы интерполирования фиксированы, то величины $L^{(i)}(x)$ могут быть вычислены заранее и храниться в памяти как постоянные коэффициенты, которые можно использовать в процессе вычислений.

В этом случае количество модульных операций для вычисления значений невязок R_j будет следующим:

а. При использовании параллельного алгоритма А1

$$N_{\oplus} = (n-r)\, r\,,$$

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

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

б. При использовании последовательного алгоритма А2

$$N_{\oplus} = (2n-r-1)\, r/2\ ,$$

$$N_{\otimes} = (2n-r-2)(r-1)/2\ .$$

Для хранения коэффициентов $L^{(i)}(x)$ нужно $(r-1)(2n-r-2)/2$ регистров.

в. При использовании последовательного алгоритма А3

$$N_{\oplus} = (2n-r-1)\, r/2\ ,$$

$$N_{\otimes} = (2n-r-2)(r-1)/2\ .$$

Для хранения коэффициентов $L^{(i)}(x)$ нужно $(r-1)(2n-r-2)/2$ регистров.

г. При использовании параллельно-последовательного алгоритма А4

$$N_{\oplus} = r(n-r)-1$$

$$N_{\otimes} = (r-1)(n-r)\,.$$

Для хранения коэффициентов $L^{(i)}(x)$ необходимо иметь $(r-1)(n-r)$ регистров.

Определим общее количество модульных операций для процедур вычисления синдромов Q_{μ} при фиксированных узлах интерполирования (т.е. при $L^{(i)}(x) = const$).

Для процедур (2) и (3) с применением параллельного алгоритма А1 будем иметь:

$$N_{\oplus} = r(n-1)\,,$$

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

Если $\beta_{r-1} = 1$ и $\beta_r = 0$, то количество операций для обеих процедур (2) и (3) следующее:

$$N_{\oplus} = r(n-2)+1\,,$$

$$N_{\otimes} = r(n-3)+2\,.$$

Общее количество ячеек памяти для хранения $L^{(i)}(x)$, β_j^{μ} (β_j) равно:

а) при $\beta_{r-1} \neq 1$ и $\beta_r \neq 0$

$r(n-1)$ – для параллельной (независимой) процедуры (2),

$r(n-r+1)$ – для последовательной процедуры (3);

б) при $\beta_{r-1} = 1$ и $\beta_r = 0$

$r(n-3)+2$ – для параллельной (независимой) процедуры (2),

$r(n-r+1)-2$ – для последовательной процедуры (3).

Вычисление величин Q_{μ} в соответствии с процедурами (2) и (3) для последовательного алгоритма А2 требует следующего количества модульных операций:

$$N_{\oplus} = r[(n-1)+(r-1)/2]\,,$$

$$N_{\otimes} = (r-1)[n+(r-2)/2]\,.$$

При $\beta_{r-1} = 1$, $\beta_r = 0$:

$$N_{\oplus} = r(n-1)+(r-1)(r-2)/2\,,$$

$$N_{\otimes} = n(r-1)+(r-1)(r-6)/2.$$

Общее количество ячеек памяти для хранения $L^{(i)}(x)$, β_j^{μ} (β_j) составляет:

а) при $\beta_{r-1} \neq 1$ и $\beta_r \neq 0$

$(r-1)[2(n-1)+r]/2$ – для параллельной (независимой) процедуры (2),
$(r-1)(2n-r)/2+1$ – для последовательной процедуры (3);

б) при $\beta_{r-1}=1$ и $\beta_r=0$
$(r-1)(2n+r-6)/2$ – для параллельной (независимой) процедуры (2),
$(r-1)(2n-r)/2-1$ – для последовательной процедуры (3).

Общее количество модульных операций при вычислении Q_μ с применением параллельно-последовательного алгоритма А4 в соответствии с процедурами (2) и (3) следующее:

$$N_\oplus = nr-1,$$

$$N_\otimes = n(r-1).$$

При $\beta_{r-1}=1$, $\beta_r=0$:

$$N_\oplus = r(n-1),$$

$$N_\otimes = (r-1)(n-2).$$

Общее количество ячеек памяти для хранения $L^{(i)}(x)$, β_j^μ (β_j) равно:

а) при $\beta_{r-1}\neq 1$ и $\beta_r\neq 0$
$n(r-1)$ – для параллельной (независимой) процедуры (2),
$n(r-1)-r(r-2)$ – для последовательной процедуры (3);

б) при $\beta_{r-1}=1$ и $\beta_r=0$
$(r-1)(n-2)$ – для параллельной (независимой) процедуры (2),
$n(r-1)-r(r-2)-2$ – для последовательной процедуры (3).

Общее количество модульных операций при вычислении Q_μ с применением последовательного алгоритма А3 в соответствии с (7) и (8) равно:

$$N_\oplus = r(n-1),$$

$$N_\otimes = n(r-1).$$

При $\beta_{r-1}=1$, $\beta_r=0$:

$$N_\oplus = r(n-2)+1,$$

$$N_\otimes = (r-1)(n-1)-1.$$

Общее количество ячеек памяти для хранения $L^{(i)}(x)$, $\lambda_{r-h}^{(\mu)}$ равно:

а) $n(r-1)$ – при $\beta_{r-1}\neq 1$ и $\beta_r\neq 0$;

б) $(r-1)(n-1)-1$ – при $\beta_{r-1}=1$ и $\beta_r=0$.

Стандартный алгоритм вычисления величин синдромов

Рассмотрим ещё один способ вычисления значений синдромов Q_μ, заключающийся в вычислении в соответствии с выражением [1]:

$$Q_\mu = \sum_{x_i\in SYI'} \widetilde{f}_i x_i^\mu, \qquad \mu=\overline{0,r-1}. \quad (9)$$

Вычисление в соответствии с (9) назовём стандартным алгоритмом вычисления величин синдромов кода Лагранжа.

Для этого алгоритма можно предложить две процедуры вычисления синдромов, аналогичные процедурам (2) и (3).

1. Параллельная (независимая) процедура:

$$Q_\mu = \sum_{x_i\in SYI'} \widetilde{f}_i X_\mu^{(i)}, \quad (10)$$

где $X_\mu^{(i)} = x_i^\mu = const$, $X_0^{(i)}=1$.

2. Последовательная процедура:

$$Q_\mu = \sum_{x_i\in SYI'} Q_\mu^{(i)}, \quad (11)$$

где $Q_\mu = Q_{\mu-1}^{(i)} x_i = const$, $Q_0^{(i)} = \widetilde{f}_i$.

Количество операций, требуемое для вычисления величин Q_μ в соответствии с процедурами (10) и (11), следующее:

$$\begin{aligned} N_\oplus &= r(n-1) \\ N_\otimes &= n(r-1). \end{aligned} \quad (12)$$

Для процедуры (10) необходимо иметь $n(r-1)$ ячеек памяти для хранения

значений x_i^{μ}, для процедуры (11) – n ячеек.

Это количество операций и ячеек памяти указано для любого набора интерполяционных узлов, то есть и для такого, в котором отсутствуют узлы $x_{i_1}=1$ и $x_{i_2}=0$.

Если узлы интерполирования выбрать так, чтобы были узлы со значениями $x_{i_1}=1$ и $x_{i_2}=0$, то получим меньшее количество операций:

$$\begin{aligned} N_{\oplus} &= r(n-2)+1 \\ N_{\otimes} &= (r-1)(n-2). \end{aligned} \qquad (13)$$

При этом уменьшится также количество ячеек памяти для хранения значений x_i^{μ} и составит:

а) $(r-1)(n-2)$ – для параллельной процедуры (10);

б) $(n-2)$ – для последовательной процедуры (11).

Вычисление суммы (9) можно производить, используя схему Горнера и принимая $x_i=\alpha^{i+1}$ (где α – примитивный элемент конечного):

$$\begin{aligned} Q_0 &= \sum_{i=0}^{n-1} \widetilde{f}_i \\ Q_{\mu} &= \{[(\widetilde{f}_0\alpha^{\mu}+\widetilde{f}_1)\alpha^{\mu}+\widetilde{f}_2]\alpha^{\mu}+\mathrm{K}+ \\ &\quad +\widetilde{f}_{n-3}\}\alpha^{\mu}+\widetilde{f}_{n-2}, \quad \mu=\overline{1,r-1}\ . \end{aligned} \qquad (14)$$

При этом для хранения значений α^{μ} требуется $(r-1)$ ячеек памяти при параллельном вычислении величин Q_{μ} в соответствии с процедурой (10) и 1 ячейка – при последовательном вычислении в соответствии с процедурой (11). Количество операций определяется из (13), так как здесь также можно выбрать узлы $x_{n-2}=1$ и $x_{n-1}=0$.

Для набора интерполяционных узлов, в котором значения $x_{n-2}\neq 1$ и $x_{n-1}\neq 0$, вычисление синдромов Q_{μ} с использованием схемы Горнера производится по формуле (для $n<q-1$):

$$\begin{aligned} Q_{\mu} &= \{[(\widetilde{f}_0\alpha^{\mu}+\widetilde{f}_1)\alpha^{\mu}+\widetilde{f}_2]\alpha^{\mu}+\mathrm{K}+ \\ &\quad +\widetilde{f}_{n-1}\}\alpha^{\mu}, \quad \mu=\overline{0,r-1}\ . \end{aligned} \qquad (15)$$

В данном случае количество операций определяется из соотношений (12), а количество ячеек памяти составляет $(r-1)$ в соответствии с процедурой (10) и 1 ячейка – при последовательном вычислении в соответствии с процедурой (11).

Сравнение процедур вычисления синдромов

Проведём сравнение рассмотренных процедур вычисления синдромов многократных ошибок для кодов Лагранжа, а также сравним эти процедуры с аналогичными процедурами для PC-кодов.

Сравнивая процедуры вычисления синдромов Q_{μ}, использующие алгоритмы А1 и А2, имеем:

– при использовании алгоритма А2 выполняется на $r(r-1)/2$ операций сложения больше, операций умножения будет меньше при $r<(1+\sqrt{8n-7})/2$;

– на вычисление Q_{μ} с использованием алгоритма А1 затрачивается меньше времени, чем на вычисление с применением алгоритма А2. Это вызвано тем, что при вычислении каждого значения $f^*(\beta_j)$ по алгоритму А1 декодеру не нужно ждать определения величин $f^*(\beta_j)$ в предыдущих контрольных узлах, так как используются при этом значения $\widetilde{f}_i$ принятой последовательности.

Сравнение процедур вычисления синдромов Q_{μ} при применении последовательного алгоритма А3 с процедурами, использующими алгоритмы А1 и А2, дает следующее:

– при использовании алгоритма А3 операций сложения и умножения меньше, чем для алгоритма А2, на величины $r(r-1)/2$ и $(r-1)(r-2)/2$ соответственно;

– вычисление с использованием алгоритма А3 требует сложений одинаковое

– количество, а умножений – на величину $(n-r)$ меньше, чем вычисление с алгоритмом А1;

– на вычисление Q_μ с использованием алгоритма А3 затрачивается меньше времени, чем на вычисление с применением алгоритма А2. Это вызвано тем, что при вычислении каждого значения $f^*(\beta_j)$ по алгоритму А3 декодеру не нужно ждать определения величин $f^*(\beta_j)$ в предыдущих контрольных узлах, так как используются при этом значения $\tilde{f}(\beta_j)$ принятой последовательности.

При вычислении Q_μ с использованием алгоритма А4 умножений выполняется одинаковое количество, а сложений – на величину (r–1) больше по сравнению с вычислением, использующим алгоритм А3 (лучший из алгоритмов А1, А2 и А3 по количеству операций в конечном поле).

Количество операций в конечных полях для последовательного (А3) и стандартного алгоритмов одинаково и меньше, чем для остальных алгоритмов.

Сравнение объёмов памяти для хранения постоянных величин (количество ячеек памяти) показывает:

– для параллельной (2) и последовательной (3) процедур количество ячеек памяти при вычислении синдромов с использованием алгоритма А2 меньше, чем при вычислении с использованием алгоритма А1, при $r<(1+\sqrt{8n-7})/2$;

– для параллельной (2) и последовательной (3) процедур количество ячеек памяти при вычислении синдромов с использованием алгоритма А3 меньше, чем при вычислении с использованием алгоритма А1, при $n>r$ и $r<(1+\sqrt{4n+1})/2$ соответственно;

– для параллельной процедуры (2) количество ячеек памяти при вычислении синдромов с использованием алгоритма А3 меньше, чем при вычислении с использованием алгоритма А2, на величину $(r-1)(r-2)/2$;

– для последовательной процедуры (3) количество ячеек памяти при вычислении синдромов с использованием алгоритма А2 меньше, чем при вычислении с использованием алгоритма А3, на величину $(r+1)(r-2)/2$;

– для параллельной (2) и последовательной (3) процедур количество ячеек памяти при вычислении синдромов с использованием алгоритма А4 меньше, чем при вычислении с использованием алгоритма А1, при $n>r$;

– для параллельной (2) и последовательной (3) процедур количество ячеек памяти при вычислении синдромов с использованием алгоритма А4 меньше, чем при вычислении с использованием алгоритма А2, на величину $(r-1)(r-2)/2$;

– для параллельной процедуры (2) количество ячеек памяти при вычислении синдромов с использованием алгоритмов А4 и А3 одинаково;

– для последовательной процедуры (3) количество ячеек памяти при вычислении синдромов с использованием алгоритма А4 меньше, чем при вычислении с использованием алгоритма А3, на величину $r(r-2)/2$;

– для параллельной процедуры (2) количество ячеек памяти при вычислении синдромов с использованием алгоритмов А3, А4 и стандартного алгоритма одинаково;

– для последовательной процедуры (3) количество ячеек памяти при вычислении синдромов с использованием стандартного алгоритма меньше, чем при вычислении с использованием алгоритмов А3 и А4, на величины $n(r-2)$ и $(n-r)(r-2)$;

– для стандартного алгоритма при вычислении по схеме Горнера количество ячеек памяти меньше, чем для вычислений с использованием остальных рассмотренных алгоритмов.

Имеются процедуры [2] с асимпто-

тической сложностью операций в $GF(q)$: $cn\ln r + k$, где c – константа. Но эти процедуры предназначены для реализации с помощью универсальных ЭВМ и имеют более сложную логическую организацию, чем для кодов Лагранжа, предлагаемые алгоритмы которых можно реализовать аппаратно.

Сложность операций в $GF(q)$ вычисления синдромов кодов Лагранжа с использованием последовательного алгоритма А3 при определении значений искаженного полинома в контрольных узлах или стандартной процедуры меньше асимптотической сложности быстрого алгоритма РС-кодов при выполнении условий (для $\beta_{r-1} = 1, \quad \beta_r = 0$):

$(k+2)/n < c\ln r - r + 3$ – для умножения,

$(k+1)/n < c\ln r - r + 2$ – для сложения.

Сравним процедуры вычисления синдромов кодов Лагранжа с аналогичными процедурами *РС*-кодов.

Стандартная процедура вычисления синдрома для *РС*-кодов, использующая схему Горнера, требует $r(n\text{-}1)$ операций умножения (сложения) [3]. Это:

1. При $\beta_{r-1} \neq 1, \quad \beta_r \neq 0$:

– больше количества операций умножения, чем при использовании алгоритмов А3, А4 и стандартного на величину $(n-r)$;

– одинаковое количество операций сложения при вычислении с использованием алгоритмов А1, А3 и стандартного;

– одинаковое количество операций умножения с алгоритмом А1;

– меньше количества операций сложения, чем при использовании алгоритмов А2 и А4 на величины $r(r-1)/2$ и $(r-1)$ соответственно;

– меньшее количества операций умножения, чем при использовании алгоритма А2 на величину $(r-n)+(r-1)(r-2)/2$.

2. При $\beta_{r-1} = 1, \quad \beta_r = 0$:

– больше количества операций сложения, чем при использовании алгоритмов А1, А3 и стандартного на величину $(r-1)$;

– больше количества операций умножения, чем при использовании алгоритма А1 на величину $2(r-1)$, алгоритмов А3 на величину $(n+r-1)$, А4 и стандартного на величину $(n+r-2)$;

– одинаковое количество операций сложения при вычислении с использованием параллельно-последовательного алгоритма А4;

– меньше количества операций сложения и умножения, чем при использовании алгоритма А2 на величины $(r-1)(r-2)/2$ и $(r-n)+(r-1)(r-6)/2$ соответственно.

Выводы

Разработанные процедуры позволяют вычислять величины синдромов при декодировании ошибок кодами Лагранжа для различных алгоритмов вычисления контрольных символов. Лучшие из этих процедур по количеству операций в конечных полях имеют преимущество перед стандартными процедурами вычисления синдрома для РС-кодов. Если строить схемы устройств для реализации вычисления величин Q_μ в соответствии с предлагаемыми процедурами, то они легко адаптируемы к количеству ошибок.

Список литературы

1. *Кубицкий В.И.* Декодирование многократных ошибок кодами Лагранжа // Проблеми інформатизації та управління: Зб. наук. пр. – Вип. 4(22). – К.: НАУ, 2007. – С. 86–92.

2. *Афанасьев В.Б.* Исследование сложности реализации кодов Рида–Соломона: Автореф. дис. на соиск. ученой степени канд. техн. наук. – М., 1976. – 19 с.

3. *Miller R.L., Troung T.K., Reed I.S.* Fast algorithm for encoding the (255, 223) Reed–Solomon code over GF(2^8). – Electronics Letters, 1980, v. 16, №6, – P. 222–223.