А. А. Давыдов, А. Ю. Дрожжина-Лабинская, Исправление байтов ошибок длины 4 и двойных независимых ошибок кодом Боуза — Чоудхури — Хоквингема в полупроводниковых запоминающих устройствах, *Автомат. и телемех.*, 1989, выпуск 11, 135–146

УДК 519.725

ИСПРАВЛЕНИЕ БАЙТОВ ОШИБОК ДЛИНЫ 4 И ДВОЙНЫХ НЕЗАВИСИМЫХ ОШИБОК КОДОМ БОУЗА — ЧОУДХУРИ — ХОКВИНГЕМА В ПОЛУПРОВОДНИКОВЫХ ЗАПОМИНАЮЩИХ УСТРОЙСТВАХ

ДАВЫДОВ А. А., ДРОЖЖИНА-ЛАБИНСКАЯ А. Ю.

(Москва)

Рассмотрено исправление ошибок, характерных для полупроводниковых запоминающих устройств большого объема: байтов ошибок длины 4 и двойных независимых ошибок. Предложены структура проверочной матрицы и алгоритм декодирования двоичного кода Боуза – Чоудхури – Хоквингема с расстоянием 6, позволяющие исправлять оба вида ошибок. Приведены способы построения проверочной матрицы, удобной для реализации на больших интегральных схемах. Даны оценки обнаруживающей способности алгоритма. Приведен пример блок-схемы декодера.

1. Введение

Полупроводниковые запоминающие устройства (ЗУ) большого объема обычно строят [1] из одноразрядных микросхем, которые конструктивно объединены в группы по b разрядов, имеющие общие электрические цепи (часто $b=4$). Поэтому возникающие в ЗУ ошибки подразделяют на независимые ошибки, связанные с одноразрядностью микросхем, и байты ошибок длины b – произвольные искажения в группе из b разрядов.

Для повышения надежности каждое слово ЗУ кодируется корректирующим кодом [2, 3], который должен защищать это слово и от независимых ошибок, и от байтов ошибок. Так как в ЗУ большого объема каждый разряд слова ЗУ содержит сотни (иногда тысячи) микросхем памяти, то одним из основных требований, предъявляемых к указанному коду, является небольшая избыточность (хотя это требование не всегда согласуется со стремлением повысить быстродействие и надежность декодера [4, 5]).

Для исправления одиночных независимых ошибок в ЗУ обычно используют расширенный код Хэмминга с минимальным расстоянием $d=4$ [1–6]. Можно применять и другие коды (см., например, [4, 5, 7, 8]). Код Хэмминга часто модифицируют так, чтобы обнаруживать байты ошибок длины 4 [1, 9–11]. Исправлять одиночные байты ошибок можно кодами из [12–17] (см. также списки литературы в этих работах).

При больших объемах ЗУ (например, в массовой памяти супер-ЭВМ) недостаточно исправлять одиночные байты ошибок. Требуется исправлять двойные независимые ошибки и одиночные байты ошибок длины 4. Обычно предполагается, что независимые ошибки и байт ошибок не происходят одновременно, а вероятность появления двух байтов ошибок мала.

Коды из [12–17], исправляющие одиночные байты ошибок, не исправляют двойных независимых ошибок. Выполнить указанные требования можно, используя двоичный код Боуза — Чоудхури — Хоквингема (БЧХ) с расстоянием $d=6$. Его избыточность близка к минимальной, а в классе линейных кодов, видимо, в точности минимальна. (При увеличении избыточности могут быть использованы и другие коды, например, составной нелинейный (80, 64)-код [18, 19], имеющий на один проверочный символ больше, чем соответствующий код БЧХ.)

В [20—22] показано, что код БЧХ с $d=6$ наряду с исправлением двойных независимых ошибок способен обнаруживать байты ошибок. В [23] рассмотрено исправление байтов ошибок длины 4 и двойных независимых ошибок для укороченного (144, 128)-кода БЧХ с $d=6$.

В данной работе задача исправления байтов ошибок длины 4 и двойных независимых ошибок решается в общем случае: для кодов БЧХ произвольной длины. В предложенном алгоритме одиночные и двойные независимые ошибки исправляются обычным образом [2, 3, 20], а исправление байтов ошибок по сложности сравнимо с исправлением одиночных независимых ошибок. Рассмотрены способы построения проверочных матриц, удобных для реализации на больших интегральных схемах (БИС). При построении матриц использованы идеи и результаты работ [12, 13, 17].

В разделе 2 для кода БЧХ с $d=6$ рассмотрены структура проверочной матрицы и алгоритм декодирования, позволяющие исправлять байты ошибок длины 4, одиночные и двойные независимые ошибки. Приведен пример блок-схемы декодера. В разделе 3 рассмотрены способы построения проверочной матрицы для $(n, n-2m-1)$-кодов произвольной длины $n\leqslant 2^m$. В разделе 4 для укороченных $(n, n-2m)$-кодов БЧХ с $d=6$ указан метод построения проверочной матрицы при четных $m=4$, 6, 8 и получена проверочная матрица (144, 128)-кода. В разделе 5 приведены оценки обнаруживающей способности алгоритма. Алгоритм позволяет обнаружить (т. е. классифицировать как неисправимые) подавляющую часть тройных независимых ошибок и значительную часть четырехкратных независимых ошибок. Для (79, 64)-кода обнаруживается более 99,5% тройных и более 80% четырехкратных независимых ошибок.

Основные результаты работы опубликованы без доказательств в [24].

2. Структура проверочной матрицы и алгоритм декодирования

Проверочную матрицу H расширенного двоичного (n, k)-кода БЧХ с расстоянием $d=6$, $n\leqslant 2^m$, $k=n-2m-1$, $m\geqslant 4$ можно записать в виде [2, 3]:

$$(1)\qquad H=\begin{Vmatrix} h_1 & h_2 & \dots & h_n \\ h_1^3 & h_2^3 & \dots & h_n^3 \\ 1 & 1 & \dots & 1 \end{Vmatrix}=\begin{Vmatrix} H_1 \\ H_3 \\ J \end{Vmatrix},$$

где h_j — локатор j-й позиции кодового слова; H_1 — матрица локаторов; H_3 — матрица кубов локаторов; J — строка из единиц;

$$(2)\qquad h_{j_1}\neq h_{j_2} \text{ при } j_1\neq j_2,\quad j_1, j_2\in\{\overline{1, n}\}.$$

Локаторы и кубы локаторов при декодировании и описании свойств матрицы рассматриваются как элементы поля Галуа $GF(2^m)$. В матрице H записывается m-разрядное двоичное представление этих элементов Суммирование столбцов как элементов поля означает их поразрядное сложение по модулю два.

Пример. Проверочная матрица (16, 7)-кода БЧХ с $m=4$ представлена на рис. 1, *а*. Использована таблица поля $GF(2^4)$ из [2, с. 182].

$$
T = \left\| w_0 \,\vdots\, W_3 \right\| = \begin{array}{c} \quad w_0 \qquad\qquad \overbrace{\qquad\qquad\qquad\qquad\qquad\qquad}^{W_3} \\ \left\|\begin{array}{cccccccc} 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{array}\right\| \end{array}
$$

$$
\sum \begin{array}{cccc} 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{array}
$$

$$
H = \left\|\begin{array}{cccccccc}
00 & 00 & 00 & 11 & 00 & 11 & 11 & 11 \\
00 & 00 & 11 & 00 & 11 & 11 & 11 & 00 \\
00 & 11 & 00 & 00 & 11 & 00 & 11 & 11 \\
01 & 01 & 01 & 01 & 01 & 01 & 01 & 01 \\
00 & 11 & 11 & 11 & 00 & 11 & 11 & 11 \\
00 & 01 & 10 & 01 & 00 & 00 & 01 & 11 \\
00 & 01 & 01 & 11 & 00 & 01 & 00 & 10 \\
01 & 01 & 00 & 01 & 11 & 00 & 00 & 10 \\
11 & 11 & 11 & 11 & 11 & 11 & 11 & 11
\end{array}\right\|
\begin{array}{l}
\left.\begin{array}{c} \\ \\ \\ \\ \end{array}\right\} H_1 \\
\left.\begin{array}{c} \\ \\ \\ \\ \end{array}\right\} H_3 \\
J
\end{array}
$$

а

$$
f_1 = \left\|\begin{array}{c} 0 \\ 1 \\ 1 \\ 0 \end{array}\right\| = \alpha^5; \quad f_2 = \left\|\begin{array}{c} 0 \\ 0 \\ 1 \\ 1 \end{array}\right\| = \alpha^4; \quad f_3 = \left\|\begin{array}{c} 0 \\ 0 \\ 1 \\ 0 \end{array}\right\| = \alpha; \quad f_4 = \left\|\begin{array}{c} 0 \\ 1 \\ 1 \\ 1 \end{array}\right\| = \alpha^{10}
$$

б

$$
T = \left\|\begin{array}{cccccc} 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{array}\right\|
$$

$$
\sum \begin{array}{ccc} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{array}
$$

в

$$
H = \left\|\begin{array}{cccccc}
00 & 11 & 00 & 11 & 11 & 11 \\
11 & 00 & 00 & 11 & 11 & 00 \\
00 & 00 & 11 & 00 & 11 & 11 \\
01 & 01 & 01 & 01 & 01 & 01 \\
11 & 11 & 11 & 11 & 11 & 11 \\
10 & 01 & 01 & 00 & 01 & 11 \\
01 & 11 & 01 & 01 & 00 & 10 \\
00 & 01 & 01 & 00 & 00 & 10
\end{array}\right\|
\begin{array}{l}
\left.\begin{array}{c} \\ \\ \\ \\ \end{array}\right\} H_1 \\
\left.\begin{array}{c} \\ \\ \\ \\ \end{array}\right\} H_3
\end{array}
$$

$$
f_1 = \left\|\begin{array}{c} 0 \\ 0 \\ 1 \\ 1 \end{array}\right\| = \alpha^4; \quad f_2 = \left\|\begin{array}{c} 0 \\ 1 \\ 0 \\ 1 \end{array}\right\| = \alpha^8; \quad f_3 = \left\|\begin{array}{c} 0 \\ 1 \\ 1 \\ 1 \end{array}\right\| = \alpha^{10}
$$

г

Рис. 1. Проверочные матрицы и указатели байтов ошибок кодов БЧХ с локаторами из поля $GF\ (2^4)$. *а* – вспомогательная матрица T с несовпадающими суммами пар столбцов для (16, 7)-кода БЧХ и проверочная матрица H (16, 7)-кода БЧХ (W_3 – проверочная матрица циклического (7, 4)-кода Хэмминга, $N=8$, $m-1=3$, Σ – суммы пар столбцов); *б* – указатели байтов ошибок (16, 7)-кода БЧХ (α – примитивный элемент поля); *в* – матрица T для (12, 4)-кода БЧХ ($N=6$, $m-1=3$); *г* – проверочная матрица и указатели байтов ошибок (12, 4)-кода БЧХ (строка J, совпадающая с верхней строкой матрицы H_3, удалена)

Введем обозначения: $a' = (a_1', a_2', \ldots, a_n')$ — декодируемое слово; $a_j' \in \in \{0, 1\}$, $j = \overline{1, n}$; штрих указывает, что символы декодируемого слова могут оказаться искаженными; $\mu = \left\lfloor \frac{n}{4} \right\rfloor$; $\lfloor x \rfloor$ — целая часть x.

Любое искажение декодируемого слова $a' = (a_1', \ldots, a_n')$ в пределах позиций с номерами $4i-3$, $4i-2$, $4i-1$, $4i$ ($i \in \overline{\{1, \mu\}}$) называют i-м байтом ошибок длины 4. Если искажены все четыре символа a_{4i-3}', a_{4i-2}', a_{4i-1}', a_{4i}', то имеет место байт ошибок длины 4 веса 4 (сплошной байт ошибок). Если искажены любые три из этих символов (например, a_{4i-3}', a_{4i-1}', a_{4i}'), а четвертый символ (a_{4i-2}') не искажен, то произошел байт ошибок длины 4 веса 3. Одиночные и двойные ошибки в пределах байта не будем отличать от независимых ошибок.

Сумму кубов локаторов, соответствующих i-му байту ошибок, назовем указателем i-го байта ошибок и обозначим f_i (см. рис. 1, *б*):

$$f_i = h_{4i-3}^3 + h_{4i-2}^3 + h_{4i-1}^3 + h_{4i}^3, \qquad i \in \{\overline{1, \mu}\}, \qquad \mu = \left\lfloor \frac{n}{4} \right\rfloor.$$

Обозначим $F = \{f_1, f_2, \ldots, f_\mu\}$ — совокупность всех указателей.

Для получения возможности исправлять байты ошибок длины 4 переставим локаторы в матрице H_1 так, чтобы выполнялись соотношения

$$h_{4i-3} + h_{4i-2} + h_{4i-1} + h_{4i} = 0, \quad i = \overline{1, \mu}, \tag{3}$$

$$f_i \neq f_j \text{ при } i \neq j, \quad i, j \in \{\overline{1, \mu}\}. \tag{4}$$

В разделе 3 показано, как осуществить такую расстановку локаторов.

Условие (3) позволяет при декодировании отличить байт ошибок длины 4 веса 4 от двойных независимых ошибок. Условие (4) для байтов ошибок играет ту же роль, что условие (2) для независимых ошибок. Благодаря условию (4) значение f_i позволяет определить местоположение ошибочного байта.

Алгоритм декодирования A

Для декодируемого слова a' вычисляется синдром $S = (S_1, S_3, P) = a'H^T$, где T — знак транспонирования, $S_1 = a'H_1^T$, $S_3 = a'H_3^T$, $P = a'J^T$. Решение принимается следующим образом.

1. Если $P = 0$, $S_1 = S_3 = 0$, то ошибок нет.

2. Если $P = 1$, $S_3 = S_1^3$; $S_1 \in \{h_1, \ldots, h_n\}$, то произошла одна ошибка на позиции с локатором S_1.

3. Если $P = 0$, $S_1 \neq 0$; $y_1, y_2 \in \{h_1, \ldots, h_n\}$, где y_1, y_2 — корни многочлена $y^2 + S_1 y + S_3/S_1 + S_1^2$, то произошли две независимые ошибки на позициях с локаторами y_1, y_2.

4. Если $P = 0$, $S_1 = 0$, $S_3 \in F$, то имеет место сплошной байт ошибок длины 4 веса 4 с указателем S_3.

5. Если $P = 1$, $S_3 \neq S_1^3$, $S_3 + S_1^3 = f_i \in F$ и локатор S_1 входит в байт с указателем f_i (т. е. $S_1 = h_{4i-j}$, $j \in \{0, 1, 2, 3\}$), то имеет место байт ошибок длины 4 веса 3 с указателем f_i. Позиция с локатором S_1 не искажена, остальные три позиции байта искажены.

6. Если не имеют места ситуации 1—5, обнаружена неисправимая ошибка.

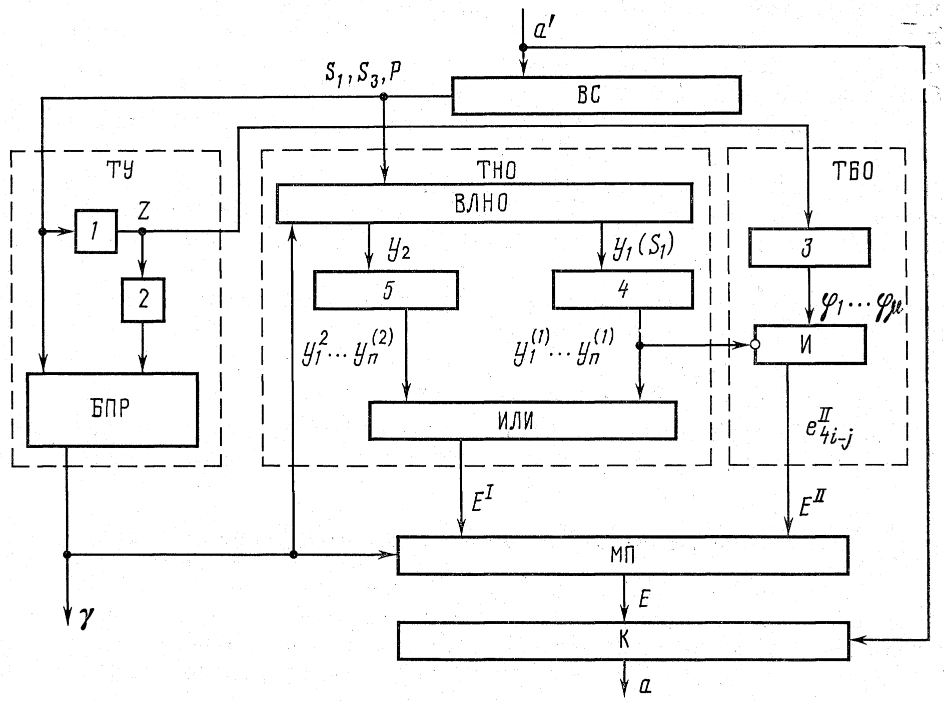


Рис. 2. Пример блок-схемы декодера кода БЧХ, исправляющего байты ошибок длины 4 и двойные независимые ошибки (1 – формирователь суммы $Z=S_3+S_1^3$, 2 – схема сравнения с нулем, 3–5 – дешифраторы, ВС – вычислитель синдрома, ТУ – тракт управления, ТНО – тракт независимых ошибок, ТБО – тракт байтов ошибок, ВЛНО – вычислитель локаторов независимых ошибок, БПР – блок принятия решения, МП – мультиплексор, К – корректор, $a'=(a_1',\ldots,a_n')$ – декодируемое слово, $e^{\mathrm{II}}_{4i-j}=\varphi_i\bar{y}^{(1)}_{4i-j}$, $E^{\mathrm{I}}=(e_1{}^{\mathrm{I}},\ldots,e_n{}^{\mathrm{I}})$, $E^{\mathrm{II}}=(e_1{}^{\mathrm{II}},\ldots,e_n{}^{\mathrm{II}})$, $E=(e_1,\ldots,e_n)$ – вектор ошибок, $a=a'+E=(a_1,\ldots,a_n)$ – исправленное слово, γ – тип и количество ошибок)

В алгоритме шаги 1—3 стандартные [3, с. 93; 20, с. 65]. Для шагов 4, 5 существенно, что все байты ошибок имеют разные указатели (см. (4)). Шаг 4 основан на том, что для байта ошибок длины 4 веса 4 имеем $S_1=0$ (см. (3)), а при двух независимых ошибках – $S_1\neq 0$ (см. (2)). При выполнении шага 5 подразумевается, что байт ошибок длины 4 веса 3 более вероятен, чем три независимых ошибки.

При выполнении шага 6 обнаруживаются (т. е. классифицируются как неисправимые) те конфигурации тройных независимых ошибок, синдромы которых не совпадают с синдромами байтов ошибок длины 4 веса 3. Обнаруживаются также неисправимые конфигурации четырехкратных независимых ошибок, синдромы которых не совпадают с синдромами двойных независимых ошибок и с синдромами байтов ошибок длины 4 веса 4.

Из сказанного вытекает следующее утверждение.

Утверждение 1. Для кода БЧХ с минимальным расстоянием $d=6$, заданного проверочной матрицей H вида (1)—(4), алгоритм декодирования A позволяет исправлять одиночные и двойные независимые ошибки и байты ошибок длины 4 при условии, что независимые ошибки и байты ошибок не происходят одновременно. При декодировании обнаруживается часть тройных и четырехкратных независимых ошибок.

Пример. Основные блоки и связи декодера показаны на рис. 2. Блок 1 вычисляет сумму $Z=S_3+S_1^3$. Равенство $Z=0$, проверяемое схемой 2, означает, что $S_3=S_1^3$. Эта информация необходима для принятия решения о наличии одиночной ошибки. С другой стороны, Z является указателем байта ошибок и подается на дешифратор 3, выход φ_i которого соответствует i-му байту ошибок: если $Z=f_i$, то $\varphi_i=1$.

Локаторы независимых ошибок y_1, y_2 поступают в дешифраторы 4, 5, имеющие выходы $y_i^{(1)}$ и $y_i^{(2)}$, $i=\overline{1, n}$. Если $P=1$, то $y_1=S_1$. Если $y_1=h_v$, $y_2=h_s$, то $y_v^{(1)}=1$, $y_s^{(2)}=1$. Объединение выходов дешифраторов 4, 5 создает вектор $E^{\mathrm{I}}=(e_1^{\mathrm{I}},\dots,e_n^{\mathrm{I}})$ с единицами в позициях, соответствующих независимым ошибкам. Выходы дешифратора 4 используются и для получения вектора $E^{\mathrm{II}}=(e_1^{\mathrm{II}},\dots,e_n^{\mathrm{II}})$ с единицами в позициях, соответствующих байту ошибок. На шаге 5 алгоритма позиция с локатором S_1 считается не искаженной и для i-го байта ошибок:

$$e^{\mathrm{II}}_{4i-j}=\varphi_i\bar{y}^{(1)}_{4i-j},\qquad j=0, 1, 2, 3;\qquad i=\overline{1, \mu}.$$

Выход мультиплексора $E=(e_1,\dots,e_n)$ совпадает с вектором E^{I} (или E^{II}), если принято решение о наличии независимых ошибок (или байтов ошибок). Исправленное слово a получается в корректоре: $a=a'+E$.

3. Построение проверочной матрицы требуемой структуры

Построение проверочной матрицы H сводится к построению матрицы локаторов H_1, удовлетворяющей условиям (2)—(4). В данном разделе построение такой матрицы H_1 в свою очередь сводится к формированию вспомогательной матрицы T, обладающей определенными свойствами, и для матрицы T приводятся методы построения. Эти методы используют идеи и результаты работ [12, 13, 17].

Обозначим через $T=\|t_1t_2\dots t_N\|$ матрицу размера $(m-1)\times N$, обладающую следующими свойствами: N — четное; $N=n/2$; каждый столбец t_i является $(m-1)$-разрядным представлением элемента поля $GF(2^{m-1})$ и

(5) $\qquad t_k\neq t_l$ при $k\neq l$; $k, l\in\{\overline{1, N}\}$,

(6) $\qquad t_{2i-1}+t_{2i}\neq t_{2j-1}+t_{2j}$ при $i\neq j$; $i, j\in\{\overline{1, N/2}\}$.

Из матрицы T построим матрицу H_1 размера $m\times n$ (где $n=2N$):

$$H_1=\left\|\begin{array}{cccc|cccc|c|cccc} t_1 & t_1 & t_2 & t_2 & t_3 & t_3 & t_4 & t_4 & \multirow{2}{*}{\dots} & t_{N-1} & t_{N-1} & t_N & t_N\\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & & 0 & 1 & 0 & 1\end{array}\right\|. \tag{7}$$

При построении матрицы H_1 каждый столбец t_i записывается дважды. К первой записи снизу приписывается ноль, ко второй — единица.

Утверждение 2. Матрица H_1 вида (7) удовлетворяет условиям (2)—(4). Доказательство утверждения 2 приведено в приложении.

Пример. Для $m=4$, $N=8$ и $N=6$ матрицы T, H_1 см. на рис. 1, *а* и 1, *в*, *г*.

Итак, чтобы построить матрицу H_1, достаточно найти матрицу T, столбцы которой различны и суммы последовательно расположенных пар столбцов (t_1+t_2; t_3+t_4; t_5+t_6; ...) также различны (см. (5) и (6)).

Обозначим H_2^* матрицу, которая содержит столбцы только нечетного веса и является проверочной матрицей кода, исправляющего одиночные байты ошибок длины 2. Матрица H_2^*, из которой удалена любая строка, удовлетворяет условиям (5), (6) [12, 13, 17]. Следовательно, из $m\times 2^{m-1}$-матриц H_2^*, построенных в [13, 17], можно получить $(m-1)\times 2^{m-1}$-матрицы T. Практические требования к проверочной матрице разнообразны и противоречивы [1, 4, 5, 6, 11, 20]. Для того чтобы обеспечить разработчику возможность выбора матрицы из большего числа вариантов, приве-

$$B=\|\,01\,\|.\quad W_2=\begin{Vmatrix}0&1&1\\1&0&1\end{Vmatrix}.\qquad T=\begin{Vmatrix}V_{20}&V_{21}&V_{22}&V_{23}\\B&B&B&B\end{Vmatrix}=\begin{Vmatrix}00&01&11&10\\00&10&01&11\\01&01&01&01\end{Vmatrix}$$

$$\sum\begin{matrix}0&1&0&1\\0&1&1&0\\1&1&1&1\end{matrix}$$

а

$$T'=\begin{Vmatrix}\overbrace{00\;\;00\;\;00\;\;00}^{V_{80}}&\overbrace{01\;\;01\;\;01\;\;01}^{V_{81}}&\overbrace{11\;\;11\;\;11\;\;11}^{V_{82}}&\overbrace{10\;\;10\;\;10\;\;10}^{V_{83}}\\00\;\;00\;\;00\;\;00&10\;\;10\;\;10\;\;10&01\;\;01\;\;01\;\;01&11\;\;11\;\;11\;\;11\\00\;\;01\;\;11\;\;10&00\;\;01\;\;11\;\;10&00\;\;01\;\;11\;\;10&00\;\;01\;\;11\;\;10\\00\;\;10\;\;01\;\;11&00\;\;10\;\;01\;\;11&00\;\;10\;\;01\;\;11&00\;\;10\;\;01\;\;11\\01\;\;01\;\;01\;\;01&01\;\;01\;\;01\;\;01&01\;\;01\;\;01\;\;01&01\;\;01\;\;01\;\;01\end{Vmatrix}$$

$$\sum\begin{matrix}0&0&0&0&1&1&1&1&0&0&0&0&1&1&1&1\\0&0&0&0&1&1&1&1&1&1&1&1&0&0&0&0\\0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1\\0&1&1&0&0&1&1&0&0&1&1&0&0&1&1&0\\1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\end{matrix}$$

б

Рис. 3. Итеративное построение вспомогательных матриц T при $Q=2$. *а* – $N=2^3$ (Σ – суммы пар столбцов); *б* – $N=2^5$

дем гибкий итеративный метод построения регулярных матриц T, удобных для реализации на БИС. При построении используются, как и в [12, 13, 17], некоторые свойства проверочной матрицы кода Хэмминга.

Обозначим через $W_R=\|w_1w_2\ldots w_U\|$ проверочную матрицу циклического $(U,\ U-R)$-кода Хэмминга длины $U=2^R-1$ [2, с. 246; 3, с. 192]. Столбцы w_i матрицы W_R являются двоичным представлением элементов поля $GF(2^R)$: $w_i=\alpha^{i-1}$, где α – примитивный элемент поля. Известно [2, 3], что

(8) $\qquad w_i\neq w_j$ при $i\neq j,\quad i,j\in\{\overline{1,U}\}$,

(9) $\qquad w_i+w_{i+1}\neq w_j+w_{j+1}$ для $i\neq j,\quad i,j\in\{\overline{1,U}\},\ w_{U+1}=w_1$.

Примеры. Матрицы W_2 и W_3 показаны соответственно на рис. 3, *а* и 1, *а*.

Для $N=2^{m-1}$ матрица T строится на основе обладающей нужными свойствами матрицы B меньшего размера.

Пусть B – матрица размера $L\times s$, удовлетворяющая условиям (5), (6); $s=2^L$; $m-3\geqslant L\geqslant 1$; $Q=m-1-L$; $U=2^Q-1$; V_{s0} – нулевая матрица размера $Q\times s$. Из $Q\times U$-матрицы $W_Q=\|w_1w_2\ldots w_U\|$ получим $Q\times s$-матрицы V_{si}, где

$$V_{si}=\|\underbrace{v_i\ldots v_i}_{s/2\text{ раз}}\|,\quad v_i=\|w_iw_{i+1}\|,\quad i=\overline{1,U},\quad w_{U+1}=w_1,$$

т. е. матрица V_{si} состоит из $s/2$ одинаковых пар столбцов w_iw_{i+1}. Матрица T имеет вид

$$T=\begin{Vmatrix}V_{s0}&V_{s1}&V_{s2}\ldots V_{sU}\\B&B&B\ldots B\end{Vmatrix}\tag{10}$$

и в соответствии с (8), (9) удовлетворяет условиям (5), (6).

Полученную матрицу T можно снова использовать как матрицу B.

Пример. Пусть $N=2^3$, $m-1=3$, $L=1$, $s=2$, $B=\|01\|$. Тогда $Q=2$, $U=3$. Матрицы W_2, T показаны на рис. 3, *а*. Положим $N=2^5$, $m-1=5$, $L=3$, $s=8$, $Q=2$ и возьмем матрицу T на рис. 3, *а* в качестве матрицы B. Новая матрица T' показана на рис. 3, *б*.

При фиксированном значении $m-1$, меняя параметр L и вид матрицы B, можно получать различные матрицы T, которые, как правило, не совпадают с матрицами, полученными выбрасыванием строки из H_2^*. (Но в частных случаях такое совпадение возможно, например, при $L=1$, $B=\|01\|$ – см. [17].)

Отметим, что матрица W_R, как следует из (8), (9), удовлетворяет условиям (5), (6). Поэтому матрицу T при $N<U=2^{m-1}-1$ можно строить непосредственно из матрицы W_{m-1}, исключив $U-N$ лишних столбцов. Вначале исключаем из W_{m-1} любой столбец w_{2j-1} с нечетным номером и получаем матрицу T вида

$$T=\|w_1\ldots w_{2j-3}w_{2j-2}w_{2j}w_{2j+1}\ldots w_U\|,\quad j\in\{1, 2, \ldots, 2^{m-2}\},\tag{11}$$

затем из T исключаем $(U-N-1)/2$ лишних пар столбцов.

Если $N=U+1=2^{m-1}$, то для получения матрицы T можно удлинить матрицу W_{m-1}, добавив нулевой столбец w_0. При этом следует учитывать четность числа x, где $\alpha^x=\alpha+1$ [17, 25]. Для фиксированного значения v поля $GF(2^v)$, построенные по разным примитивным полиномам, могут вести себя по отношению к четности числа x неодинаково [17]. Если x – нечетное число (см., например, таблицы полей $GF(2^{m-1})$ $m-1=3$ [3, с. 114] (рис. 1, *а*), $m-1=7$ [20, с. 54], $m-1=8$ [26, с. 564]), то, как следует из результатов [17], в этом и только в этом случае столбец w_0 можно добавить, не меняя порядок столбцов в W_{m-1}. Тогда получим

$$T=\|w_0w_1w_2w_3w_4w_5\ldots w_U\|,\quad U=2^{m-1}-1.\tag{12}$$

Если x – четное число (см., например, таблицы полей $GF(2^{m-1})$ для $m-1=4$ [2, с. 182], $m-1=5$, $m-1=6$ [3, с. 114]), то можно показать, что достаточно переставить местами столбцы w_1 и w_2. Тогда

$$T=\|w_0w_2w_1w_3w_4w_5\ldots w_U\|,\quad U=2^{m-1}-1.\tag{13}$$

4. Уменьшение избыточности при укорочении кода

Если число информационных символов $k<2^m-2m-1$, то код укорачивают, исключая из проверочной матрицы $2^m-2m-1-k$ столбцов. После этого в проверочной матрице может появиться нулевая строка, линейно зависимые или одинаковые строки. Удаление одной из указанных строк приводит к уменьшению избыточности на один символ [3, с. 573; 23, 27].

Учитывая важность для практики (144, 128)-кода с локаторами из поля $GF(2^8)$ [3, 23, 27], рассмотрим случай четного m. Для четного m существует [3, с. 573] укороченный (n_0, n_0-2m)-код БЧХ с $d=6$ длины $n_0=2^{m-1}+2^{m/2}$. Проверочную матрицу такого кода, удовлетворяющую условиям (2)–(4), для $m=4$, 6, 8 построим следующим образом.

В двоичном представлении кубов элементов поля $GF(2^m)$ фиксируем разряд, значение которого совпадает для b^3 и $(b+1)^3$, где b – произвольный элемент поля. При $m=4$ это старший разряд куба (5-я строка матрицы H, рис. 1, *а*); при $m=6$ [3, с. 114] – также старший разряд; при $m=8$ [26, с. 564] – третий, считая от старшего, разряд. Отмечаем n_0 столбцов матрицы H, содержащих в зафиксированном разряде: а) при $m=4$, 8 –

единицу; б) при $m=6$ — ноль. Остальные (не отмеченные) столбцы исключаем из матрицы, тем самым укорачивая код.

Из укороченной матрицы удаляем: а) для $m=4$, 8 последнюю строку (она совпадает со строкой из всех единиц, соответствующей зафиксированному разряду), см. рис. 1, *г*; б) для $m=6$ нулевую строку.

Локаторы укороченной матрицы разбиваем на пары вида b, $b+1$. Двоичные представления локаторов из одной пары отличаются только в младшем разряде. Отбросив этот разряд, получим $n_0/2$ столбцов длины $m-1$, из которых составляем матрицу T, удовлетворяющую условиям (5), (6) (см. рис. 1, *в*). Затем используем конструкцию (7).

Пример. Для (144, 128)-кода с $m=8$ матрица T имеет вид $T=$ $=\|$6 108 3 21 4 102 52 87 72 95 11 119 9 96 10 23 13 24 12 30 58 67 20 115 16 117 49 74 26 126 33 46 36 60 71 94 25 69 28 53 40 98 90 123 31 75 70 114 42 116 32 113 65 100 15 64 93 111 81 124 43 109 44 97 79 127 35 103 45 104 7 54$\|$. Столбец t_i заменен десятичным числом (например, $(0001001)^T$ заменяется на 9).

5. Оценка обнаруживающей способности алгоритма декодирования

Способность алгоритма обнаруживать (т. е. классифицировать как неисправимые) независимые ошибки можно охарактеризовать величиной $\Delta_i=M_i/Y_i$, где M_i — число обнаруживаемых конфигураций i-кратных независимых ошибок, Y_i — общее число i-кратных независимых ошибок (Δ_i — доля обнаруживаемых конфигураций i-кратных независимых ошибок от общего числа таких ошибок).

Обозначим: b_i — число конфигураций байтов ошибок длины 4 веса i в коде длины n; g_i — число необнаруживаемых конфигураций i-кратных независимых ошибок. Очевидно, $M_i=Y_i-g_i$ и

(14) $\qquad b_3=n, \quad b_4=n/4, \quad Y_i=C_n^i-b_i, \quad \Delta_i=1-g_i/Y_i, \quad i=3, 4.$

Утверждение 3. Доля тройных независимых ошибок, обнаруживаемых алгоритмом A при декодировании кода БЧХ длины $n\leqslant 2^m$, составляет

(15) $\qquad \Delta_3=1-g_3/(C_n^3-n).$

Здесь

(16) $$g_3\leqslant\begin{cases} b_3(1/3(2^{m-1}-1)-1), & \text{если } m \text{ — нечетное,}\\ 4K(D_1-1)+(b_3-4K)(D_2-1), & \text{если } m \text{ — четное,}\end{cases}$$

(17) $\qquad D_1={}^1/_3(2^{m-1}-(-2)^{m/2}-1); \quad D_2={}^1/_3(2^{m-1}-(-2)^{m/2-1}-1);$

K — число кубов в F (т. е. K — число указателей байтов ошибок, являющихся кубом некоторого элемента поля $GF(2^m)$). Для $n=2^m$ в (16) имеет место равенство.

Доказательство утверждения 3 приведено в приложении.

Соотношения (15), (16) позволяют строго оценить величину Δ_3 снизу. Для $n=2^m$ оценка (16) точна, но при $n<2^m$ оценка Δ_3 с помощью (15), (16) является несколько заниженной. Хорошее практическое приближение для произвольного n дает соотношение

(18) $\qquad \Delta_3\approx 1-n/2^{2m}.$

Соотношение (18) вытекает из (15), (16) при $n=2^m$ (m — нечетное) и подтверждается численными экспериментами для четных и нечетных значений m. Так, для $n=79$ ($m=7$) из (18) имеем $\Delta_3=0{,}9952$, а подсчеты на ЭВМ для матриц из раздела 3 дают $\Delta_3=0{,}9954\div 0{,}9957$.

Таким образом, величина Δ_3 весьма близка к единице, т. е. алгоритм A обнаруживает подавляющую часть тройных независимых ошибок.

При анализе величины Δ_4, используя сведения о количестве слов веса 6 в коде БЧХ с $d=6$ [3, 28] и применяя те же методы, что при доказательстве утверждения 3, можно показать, что основной «вклад» в g_4 вносит совпадение синдромов двойных независимых ошибок и четырехкратных независимых ошибок. Приближенная оценка, подтверждаемая расчетами на ЭВМ для матриц из раздела 3, имеет вид

$$(19)\qquad \Delta_4\approx 1-\frac{n^2}{2^{2m+1}}.$$

В ЗУ обычно $n=2^{m-1}+2m+1$ и алгоритм A обнаруживает значительную часть четырехкратных независимых ошибок. Для $n=79-\Delta_4\approx 0{,}8095$.

Авторы благодарят В. Н. Дынькина и Ю. Л. Сагаловича за конструктивные и полезные замечания.

ПРИЛОЖЕНИЕ

Доказательство утверждения 2. Для матрицы H_1 вида (7) выполнение условий (2), (3) вытекает из (5), (7). Из (7) видно, что $h_{4i-3}+h_{4i-2}=1$, где 1 – единичный элемент поля, имеющий двоичное представление $(0\ldots01)$; h_j–j-й столбец матрицы H_1. Если $a, b, c, d\in GF(2^m)$, $a+b+c+d=0$ и $a+b=1$, то [2, 3]: $a^3+b^3+c^3+d^3=(a+c)^2+(a+c)$. Значит, $f_i=\psi_i^2+\psi_i$, где $\psi_i=h_{4i-3}+h_{4i-1}$, $i=\overline{1,\mu}$. Если $a, b\in GF(2^m)$, то [3, с. 272] $a^2+a=b^2+b$ тогда и только тогда, когда $a=b$ или $a=b+1$. Из (6), (7) вытекает $\psi_i\neq\psi_j$, $\psi_i\neq\psi_j+1$ при $i\neq j$, $i, j\in\{\overline{1,\mu}\}$. Следовательно, матрица H_1 вида (7) удовлетворяет и условию (4).

Доказательство утверждения 3. При $n=2^m$ число тройных ошибок с одинаковым синдромом $(S_1, S_3, 1)=(h_{4i-j}, f_i+h^3_{4i-j}, 1)$ (см. шаг 5 алгоритма A) равно $D^{(ij)}$, где $D^{(ij)}$ – число решений системы

$$(\text{П}.1)\qquad \begin{aligned} &x_1+x_2+x_3=S_1=h_{4i-j},\\ &x_1^3+x_2^3+x_3^3=S_3=f_i+h^{4}_{4i-j},\quad i\in\{\overline{1,\mu}\},\quad j\in\{\overline{0,3}\}. \end{aligned}$$

Одно решение соответствует байту ошибок с указателем f_i, остальные – необнаруживаемым конфигурациям тройных независимых ошибок. Значит,

$$(\text{П}.2)\qquad g_3=\sum_{i=1}^{\mu}\sum_{j=0}^{3}(D^{(ij)}-1),\quad \mu=\frac{n}{4}=\frac{b_3}{4}.$$

В [28, с. 432] показано: при фиксированном x_3 система (П.1) имеет решение относительно x_1, x_2, если выражение $\Lambda_{ij}=f_i/(h_{4i-j}+x_3)^3$ имеет нулевой след (определение следа см. [3, с. 119; 28, с. 119]). Если m – нечетное, то кубы всех элементов поля $GF(2^m)$ различны [3]. Поочередно присваивая переменной x_3 различные значения из $GF(2^m)$ (кроме $x_3=h_{4i-j}$), получим (2^m-1) различных ненулевых значений Λ_{ij}, из которых ровно $2^{m-1}-1$ имеют нулевой след [3, с. 119]. Каждая тройка (x_1, x_2, x_3) при таком переборе появится три раза (с перестановкой компонент). Поэтому для любой пары (ij) при нечетном m верно: $D^{(ij)}={}^1/_3(2^{m-1}-1)$.

Для четного m в [28, с. 432–434, 446] показано, что независимо от значения h_{4i-j} справедливо: если f_i – куб, то $D^{(ij)}=D_1$, если f_i – не куб, то $D^{(ij)}=D_2={}^1/_3(2^{m-1}-(-2)^{m/2-1}-1)$.

С учетом (П.2) сказанное доказывает равенство в (16) при $n=2^m$. Если $n<2^m$, то часть тройных независимых ошибок соответствует позициям, исключенным при укорочении кода, и в (16) имеем неравенство.

ЛИТЕРАТУРА

1. *Chen C. L., Hsiao M. Y.* Error – correcting codes for semiconductor memory applications: a state-of-art review // IBM J. Res. Develop. 1984. V. 28. № 2. P. 124–134.
2. *Питерсон У., Уэлдон Э.* Коды, исправляющие ошибки. М.: Мир, 1976.
3. *Мак-Вильямс Ф. Дж., Слоэн Н. Дж. А.* Теория кодов, исправляющих ошибки. М.: Связь, 1979.

4. *Сагалович Ю. Л., Щербаков Н. С.* Выбор системы кодирования для защиты запоминающих устройств от ошибок // Пробл. передачи информ. 1984. Т. 20. Вып. 1. С. 19–27.
5. *Горшков В. Н.* Надежность оперативных запоминающих устройств ЭВМ. Л.: Энергоатомиздат, 1987.
6. *Конопелько В. К., Лосев В. В.* Надежное хранение информации в полупроводниковых запоминающих устройствах. М.: Радио и связь, 1986.
7. *Панченко В. И.* Об оптимизации линейных кодов с расстоянием 4 // VIII Всесоюз. конф. по теории кодирования и передачи информ. Тез. докл. Ч. II. Теор. кодирования. М.– Куйбышев, 1981. С. 132–134.
8. *Давыдов А. А., Томбак Л. М.* Альтернатива кодам Хэмминга при коррекции однократных ошибок в ЗУ супер-ЭВМ // II Всесоюз. конф. по актуальным проблемам информатики и вычислительной техники – Информатика-87. Тез. докл. Ереван, 1987. С. 23–25.
9. *Kaneda S.* A class of odd weight-column SEC-DED-SbED codes for memory system applications // IEEE Trans. Comput. 1984. V. C-33. № 8. P. 737–739.
10. *Kaneda S.* A class of SEC-DED-SbED codes for semiconductor memory systems // Systems and Computers in Japan. 1985. V. 16. № 5. P. 88–96.
11. *Бояринов И. М., Давыдов А. А., Шабанов Б. М.* Исправление ошибок в основной памяти высокопроизводительной ЭВМ // АиТ. 1987. № 7. С. 152–165.
12. *Hong S. J., Patel A. M.* A general class of maximal codes for computer applications // IEEE Trans. Comput. 1972. V. C-21. № 12. P. 1322–1331.
13. *Fujiwara E.* Odd-weight-column b-adjacent error correcting codes // Trans. IECE of Japan. 1978. V. E61. № 10. P. 781–787.
14. *Kaneda S., Fujiwara E.* Single byte error correcting double byte error detecting codes for memory systems // IEEE Trans. Comput. 1982. V. C-31. № 7. P. 596–602.
15. *Зайцев Г. В., Зиновьев В. А., Семаков Н. В.* Коды с минимальной плотностью проверок для исправления байтов ошибок, стираний или дефектов // Пробл. передачи информации. 1983. Т. 19. Вып. 3. С. 29–37.
16. *Chen C. L.* Byte-oriented error-correcting codes for semiconductor memory systems // IEEE Trans. Comput. 1986. V. C-35. № 7. P. 646–648.
17. *Сагалович Ю. Л.* Об одном свойстве кода Хэмминга // Пробл. передачи информ. 1988. Т. 24. Вып. 1. С. 102–105.
18. *Анохин А. В., Бояринов И. М., Давыдов А. А. и др.* Исправление двойных и обнаружение тройных ошибок в полупроводниковой памяти ЭВМ // Вопросы кибернетики. Проблемы создания высокопроизводительных ЭВМ. ВК-103. М.: Науч. совет по комплексной проблеме «Кибернетика» АН СССР, 1984. С. 50–66.
19. *Анохин А. В., Бояринов И. М.* Параллельный алгоритм исправления двойных независимых и одиночных байтов ошибок в полупроводниковой памяти супер-ЭВМ // I Всесоюз. конф. «Проблемы создания супер-ЭВМ, супер-систем и эффективность их применения». Тез. докл. Ч. 2. Минск, 1987. С. 115–117.
20. *Анохин А. В., Бояринов И. М., Давыдов А. А.* Методы параллельного декодирования кодов БЧХ, исправляющих двойные и обнаруживающих тройные ошибки // Вопросы кибернетики. Проблемы организации высокопроизводительных ЭВМ. ВК-97. М.: Науч. совет по комплексной проблеме «Кибернетика» АН СССР, 1984. С. 48–92.
21. *Гарбузов Н. И., Паращук Л. Н., Шарапов А. П.* Запоминающее устройство с коррекцией ошибок: А. с. 1127012 СССР // Б. И. 1984. № 4. С. 85.
22. *Анохин А. В., Бояринов И. М., Давыдов А. А.* Устройство для декодирования кода: А. С. 1190525 СССР // Б. И. 1985. № 41. С. 89.
23. *Chen C. L.* Double error correction-triple error detection code for a memory: Europ. pat. A20107038. 02.05.1984.
24. *Давыдов А. А., Дрожжина-Лабинская А. Ю.* Исправление байтов ошибок длины 4 и двойных независимых ошибок кодом БЧХ в полупроводниковых ЗУ супер-ЭВМ // I Всесоюз. конф. «Проблемы создания супер-ЭВМ, супер-систем и эффективность их применения». Тез. докл. Ч. 2. Минск, 1987. С. 137–139.
25. *Варшамов Р. Р., Геворкян Д. Н., Климиашвили М. А.* Построение оптимальных кодов, исправляющих пачки ошибок // 2-я Венгерская конф. по выч. техн. Труды. Ч. 2. Будапешт, 1977. С. 905–909.
26. *Lin S., Costello D. J.* Error control coding: fundamental and applications. New Jersey: Prentice-Hall, 1983.
27. *Chen C. L.* On shortened finite geometry codes // Information and Control. 1972. V. 20. № 3. P. 216–221.
28. *Берлекэмп Э.* Алгебраическая теория кодирования. М.: Мир, 1971.

Поступила в редакцию
18.III.1988

CORRECTION OF ERROR BYTES OF LENGTH 4 AND DOUBLE INDEPENDENT ERRORS BY THE BOSE — CHAUDHURI — HOCQUENJHEM CODE IN SEMI-CONDUCTING MEMORY UNITS

DAVYDOV A. A., DROZHZHINA-LABINSKAYA A. Yu.

The article is concerned with correction of errors that are characteristic of semi-conducting memory units of a large size, error bytes of length 4 and double independent errors. A structure of the checking matrix and an algorithm for decoding a binary Bose — Chaudhuri — Hocquenjhem code with distance 6 whereby both kinds of errors are corrected. Techniques are described for design of a checking matrix which can be easily implemented in large-scale integration circuits. The detecting ability of the algorithm is estimated. A decoder block-diagram is provided.

УДК 681.326.74:517.987

ФУНКЦИОНАЛЬНОЕ ДИАГНОСТИРОВАНИЕ НЕСТАЦИОНАРНЫХ ДИНАМИЧЕСКИХ СИСТЕМ

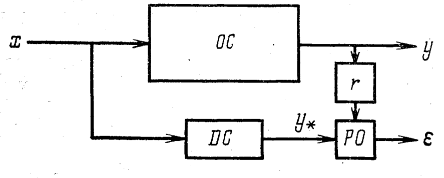
ЖИРАБОК А. Н., ШУМСКИЙ А. Е.

(Владивосток)

Рассматривается задача функционального диагностирования динамических систем, описываемых нелинейными дифференциальными уравнениями с переменными коэффициентами.

1. Введение

Теория функционального диагностирования (ФД) разработана достаточно подробно для непрерывных линейных стационарных систем [1]. В [2] предложено решение задачи ФД для стационарных нелинейных динамических систем, в основе которого лежит использование специального математического аппарата — алгебры функций. В настоящей работе предложенный в [2] подход распространяется на случай нелинейных нестационарных непрерывных динамических систем, что позволяет сформировать для последних процедуру синтеза средств ФД. Эта процедура может быть использована также и для линейных нестационарных систем. Распространение предложенного в [2] подхода на новый класс объектов стало возможным благодаря дальнейшему развитию аппарата алгебры функций.



2. Постановка задачи ФД

На рисунке представлена блок-схема ФД. Здесь ОС — основная (диагностируемая) система, ДС — диагностирующая система, РО — решающий орган, осуществляющий сравнение векторов выхода ОС $y(t)$ и ДС $y_*(t)$ в каждый момент времени t и вырабатывающий сигнал ошибки $\varepsilon \neq 0$ в случае их несовпадения, x — вектор входа ОС.

Основная система задана в пространстве состояний[1] дифференциаль-

[1] Описание технических объектов в пространстве состояний может быть получено в соответствии с методами, изложенными, например, в [3].