УДК 621.3.049.77

МОДЕЛИРОВАНИЕ НАДЕЖНОСТИ УСТРОЙСТВ ХРАНЕНИЯ И КРИПТОГРАФИЧЕСКОГО ПРЕОБРАЗОВАНИЯ ИНФОРМАЦИИ

Романенко Д. М., Галабурда А. И., Урбанович П. П.

Белорусский государственный технологический университет

Наблюдаемое в последние годы резкое увеличение информационных потоков, широкое распространение персональных ЭВМ и другой вычислительной техники, а также ужесточение требований к целостности передаваемой и обрабатываемой информации резко обострили проблему надежного хранения и передачи двоичной информации. Один из способов решения такой проблемы заключается в использовании помехоустойчивого кодирования данных с помощью избыточных кодов.

Двухмерные итеративные коды, широко применяемые на практике и более известные как HV-коды, являются простейшим примером использования методов комбинирования известных кодов для построения новых и представляют собой прямое произведение кодов простой проверки на четность [1, 2].

Вместе с тем поступательное и все ускоряющееся улучшение основных эксплуатационных параметров (быстродействие, емкость ОЗУ – оперативных запоминающих устройств и др.) персональных компьютеров и других средств вычислительной техники, продиктованное необходимостью внедрения новых информационных технологий, базируется на улучшении адекватных параметров, прежде всего, полупроводниковых устройств хранения и выдачи информации (ОЗУ, ПЗУ, ППЗУ). Принятое за рубежом направление усовершенствования таких ЗУ основывается на создании систем памяти, интегрированных на целой полупроводниковой пластине (WSI – wafer scale integration).

При использовании WSI в качестве ЗУ естественным представляется обобщение итеративных кодов на трехмерный случай. Возможность такого обобщения появляется благодаря тому, что в WSI суммарная емкость системы набирается из однотипных элементов – блоков. Физически такой способ кодирования можно представить в виде куба или параллелепипеда, состоящего из n одинаковых накопителей (отдельных блоков кристалла), «наложенных» друг на друга. Боковые стенки этого куба образуются совокупностью элементов четности n матриц. Верхняя грань фор-

мирует четность всех элементов, содержащихся в каждой матрице. Именно к данному типу относится трехмерный итеративный код с проверочными символами по диагонали [3]. В дальнейшем рассмотрим основные характеристики данного кода, алгоритмы коррекции многократных ошибок для повышения надежности ЗУ, модель криптоалгоритма ГОСТ с реализацией такой дополнительной функции, как обеспечение повышенной функциональной надежности.

Трехмерный итеративный код с проверочными символами по диагонали характеризуется следующими параметрами: избыточностью (r), минимальным кодовым расстоянием (d), скоростью (R), эффективностью коррекции ошибок.

Величина избыточности равна

$$r = 2 \cdot (k_1 + k_2) \cdot k_3 + 2 \cdot (k_1 + k_2), \qquad (1)$$

где k_1 и k_2 – количество строк и столбцов в линейном итеративном коде с проверочными символами по диагонали;

k_3 – количество плоскостей в трехмерном итеративном коде.

Если $k_1 = k_2 = k_3$, то

$$r = 6 \cdot k_m^2, \qquad (2)$$

следовательно,

$$n = k + r = k_m^3 + 6 \cdot k_m^3. \qquad (3)$$

Тогда скорость (R) определяется следующим соотношением:

$$R = \frac{k}{n} = \frac{k_m}{(k_m + 6)}. \qquad (4)$$

Минимальное кодовое расстояние трехмерного итеративного кода, согласно [3], будет $d = 10$, а это значит, что код исправляет до четырех ошибок включительно.

Сейчас известно, что декодирование любого кода является наиболее трудоемкой операцией. Сложности реализации декодеров особенно возрастают при коррекции ошибок, число которых больше единицы. При этом иногда возникают ситуации, когда сложность декодера приводит к таким значительным временным и аппаратурным затратам, что сводит на нет корректирующие способности кода. Поэтому более подробно остановимся на различных алгоритмах коррекции многократных ошибок.

Авторами разработан специальный алгоритм коррекции ошибок кратностью не выше четырех. Данный алгоритм строится на принципах, аналогичных положенным в основу коррекции многократных ошибок дополненным линейным итеративным кодом [3], т. е. с ис-

пользованием hvd классификации с учетом проверок z–рядов (т. е. проверок на четность вдоль оси OZ) [3].

Если ограничить кратность ошибок, которые должны быть исправлены, до $(d-1)/2+1$ (если $d-$ нечетное) и до $(d-2)/2+1$ (если d – четное), где d – минимальное кодовое расстояние трехмерного итеративного кода с проверочными символами по диагонали, то для коррекции ошибок можно использовать следующий алгоритм: в случае возникновения ошибки кратностью $(d-1)/2+1$ или $(d-2)/2+1$ координаты ошибки определяются по Z – рядам, «показавшим» о наличии ошибки; в случае возникновения ошибок кратностью меньше $(d-1)/2+1$ или $(d-2)/2+1$ ошибки исправляются «силами» линейных итеративных кодов с проверочными символами по диагонали. Для реализации данного алгоритма можно использовать урезанную проверочную матрицу, т. е. из проверочной матрицы исключаются паритеты паритетов. В данном случае, хотя и происходит уменьшение кратности корректируемых ошибок, одновременно снижается избыточность кода, а также существенно упрощается алгоритм коррекции. Так, избыточность стандартной проверочной матрицы трехмерного итеративного кода с проверочными символами по диагонали при $k = 1024$ равна $r = 560$; при той же длине информационного слова для усеченной проверочной матрицы $r = 513$. Избыточность уменьшилась на 8.4%. Для усеченной проверочной матрицы величина избыточности будет определяться по следующей зависимости:

$$r = 2 \cdot (k_1 + k_2) \cdot k_3 + k_1 \cdot k_2 + 1), \tag{5}$$

При $k_1 = k_2 = k_3$ величина избыточности будет равна

$$r = 5 \cdot k_m^2 + 1, \tag{6}$$

следовательно,

$$n = k + r = k_m^3 + 5 \cdot k_m^3 + 1. \tag{7}$$

Тогда скорость (R) определяется следующим соотношением:

$$R = \frac{k}{n} = \frac{5 \cdot k_m + 1}{k_m^3 + 5 \cdot k_m + 1}. \tag{8}$$

Данный алгоритм является общим для всех трехмерных кодов, полученных путем прямого (кронекеровского) произведения двух кодов, при условии, что одним из кодов-сомножителей является свертка по модулю два. Вторым сомножителем может быть любой код, например код Хемминга.

Одним из этапов как разработки новых, так и совершенствования старых избыточных кодов является определение оптимальных размеров проверочных матриц, описывающих код. Так как код должен корректировать все ошибки заданной кратности, то основным критерием оптимизации будет выступать величина избыточности, которая должна быть минимальна.

Трехмерный итеративный код с проверочными символами по диагонали, описанный в [3], может использоваться в двух видах: со стандартной и урезанной проверочными матрицами. Для определения оптимальных размеров данных матриц была разработана специальная программа на языке Delphi. Полученные результаты представлены в табл. 1 и табл. 2 (для стандартной матрицы – табл. 1, для урезанной матрицы – табл. 2).

Таблица 1

Оптимальные размеры стандартной проверочной матрицы

Длина информационного слова k	Размерность матрицы $k_1 \times k_2 \times k_3$	Величина избыточности r	Относительная избыточность $r_{отн}$, %
8	2 × 2 × 2	28	350
16	2 × 2 × 4 4 × 2 × 2	44	275.0
64	4 × 4 × 4	96	150.0
512	8 × 8 × 8	352	68.7
4096	16 × 16 × 16	1344	32.8

Таблица 2

Оптимальные размеры стандартной проверочной матрицы

Длина информационного слова k	Размерность матрицы $k_1 \times k_2 \times k_3$	Величина избыточности r	Относительная избыточность $r_{отн}$, %
8	2 × 2 × 2 4 × 2 × 1	21	262.5
16	4 × 4 × 1 4 × 2 × 2	33	206.2
64	4 × 4 × 4 8 × 4 × 2	81	126.6
512	8 × 8 × 8 16 × 8 × 4	321	62.7
4096	16 × 16 × 16 32 × 16 × 8	1281	31.3

Как видно из табл. 1 и табл. 2, в обоих случаях оптимальной является проверочная матрица, близкая к кубической, либо кубиче-

ская (если при заданной длине информационного слова такую матрицу можно построить). При использовании урезанной проверочной матрицы, как правило, существует две оптимальные матрицы. С точки зрения аппаратной реализации и построения алгоритма коррекции целесообразно использовать проверочную матрицу с $k_1 = k_2$, т. е. с квадратной проверочной матрицей для линейного итеративного кода в плоскости.

Из вышеописанного можно сделать вывод о целесообразности использования данного кода для аппаратных реализаций криптоалгоритмов.

Далее проанализируем один из вариантов аппаратной реализации криптоалгоритма ГОСТ на предмет устойчивости к сбоям оборудования. Сбои оборудования являются следствием сложных физических процессов, происходящих в аппаратуре преобразования данных, а именно в таких компонентах криптосхем, как логические, арифметические элементы, элементы памяти и другие сбои. Следствием влияния сбоев и отказов в криптографических схемах на достоверность преобразуемой информации будем считать ошибки. Структурная схема простейшего устройства, реализующего шифрование информации в режим простой замены, представлена на рис. 1.

Схема содержит:

– два 32-разрядных накопителя N_1 и N_2;

– ключевое запоминающее устройство, состоящее из восьми 32-разрядных накопителей (на схеме представлен только один из них);

– 32-разрядный сумматор по модулю 2^{32} ($СМ_1$);

– 32-разрядный сумматор по модулю 2 ($СМ_2$);

– модуль циклического сдвига на одиннадцать шагов в сторону старшего разряда (R);

– блок подстановки.

Криптографический алгоритм ГОСТ является блочным. Это означает, что данные для шифрования разбиваются на блоки, каждый из которых в дальнейшем подвергается многократным преобразованиям. Рассмотрим алгоритм шифрования блока данных. Блок данных $T = (a_1, a_2, \ldots, a_{32}, b_1, b_2, \ldots, b_{32})$, состоящий из 64 бит, разбивается на два 32-битных блока. Далее производится ввод информации в накопители N_1 и N_2 таким образом, что значение a_1 вводится в 1-й разряд N_1, a_2 – во 2-й и т. д., а значение b_1 вводится в 1-й разряд N_2, b_2 – во 2-й и т. д. В результате получаем состояние (a_{32}, $a_{31}, \ldots, a_2, a_1$) накопителя N_1 и состояние ($b_{32}, b_{31}, \ldots, b_2, b_1$) накопителя N_2. Алгоритм зашифрования 64-разрядного блока открытых данных в режиме простой замены состоит из 32 циклов.

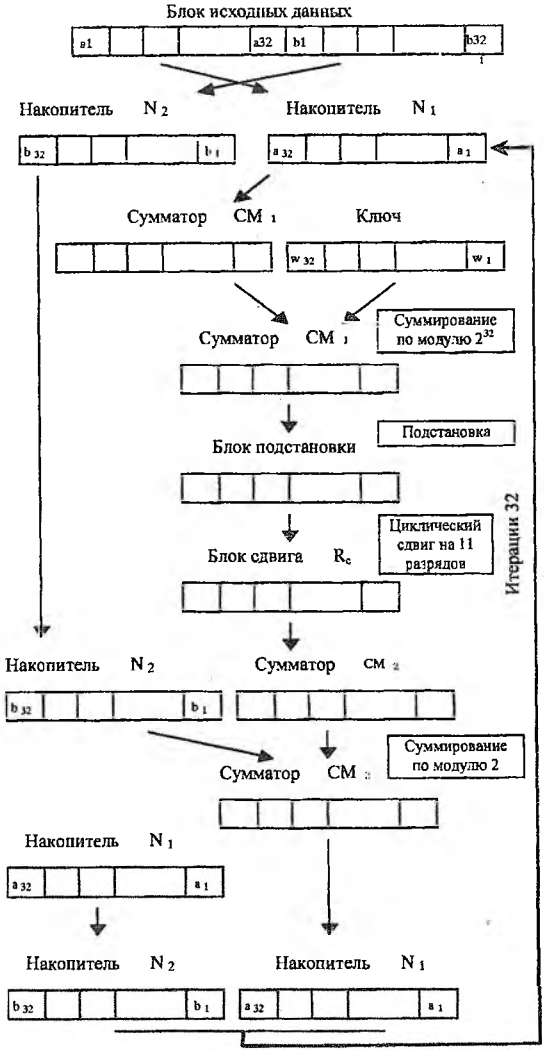

Рис. 1. Структурная схема устройства шифрования

В первом цикле начальное заполнение накопителя N_1 суммируется в сумматоре $СМ_1$ с заполнением накопителя X_0 (ключевая информация), при этом содержимое накопителя сохраняется. Результат суммирования преобразуется в блоке подстановки, и полученный вектор поступает на вход регистра R_c, где циклически сдви-

гается на 11 разрядов в сторону старших разрядов. Результат сдвига суммируется поразрядно по модулю 2 с заполнением накопителя N_2. Полученный в $СМ_2$ результат записывается в N_1, при этом старое заполнение N_1 переписывается в N_2. На этом первый цикл заканчивается. Последующие циклы осуществляются аналогично.

Отдельно следует упомянуть блок замены. Блок состоит из 8 составляющих блоков с памятью на 64 бита каждый. Поступающий на блок подстановки 32-разрядный вектор данных разбивается на 8 блоков по 4 бита. Каждый из этих блоков преобразуется соответствующим узлом замены, представляющим собой матрицу из 16 четырехбитовых строк. Входной вектор определяет адрес строки матрицы, а содержимое строки является выходным вектором. Далее эти выходные вектора последовательно объединяются в 32-разрядный вектор.

Проанализируем работу данной логической схемы на предмет устойчивости к ошибкам.

Будем рассматривать ошибки в следующих блоках схемы:

– сумматор по модулю 2^{32} ($СМ_1$);

– блок подстановки;

– блок циклического сдвига;

– сумматор по модулю 2.

При анализе сумматора по модулю 2^{32} необходимо заметить, что данные могут поступить с искаженными битами или даже целыми группами битов, т. к. данные хранятся в накопителях, а накопители, в свою очередь, могут не содержать механизмов коррекции ошибок. Также сбой может произойти в самом сумматоре. Количество искаженных битов после операции суммирования будет зависеть от положения бита ошибки в 32-битном слове. В самом худшем случае, если бит ошибки находится в младших разрядах, может произойти искажение всех битов 32-битного блока.

Блок подстановки чрезвычайно восприимчив к возможным ошибкам ввиду того, что изменение любого бита в 4 битовом блоке повлечет выемку результата подстановки из другой строки. Что в общем случае приведет к замене всего 4-битового блока неверными значениями. Следует, однако, заметить, что в 10% случаев это не приведет к потере правильных данных [5]. Данные сведения подтверждаются требованиями к узлам подстановки, т. к. одним из основных критериев является обеспечение размножения ошибок.

В блоке циклического сдвига сдвиг на неверную величину приведет к полной потере информации в блоке данных.

В сумматоре по модулю 2 ошибка в любом бите затрагивает лишь бит результата в соответствующем разряде, т. к. в данной операции нет переноса.

Анализируя данные по отказоустойчивости, следует заметить, что любые незначительные однобитовые ошибки будут впоследствии многократно размножены «размешивающими» преобразованиями. Так, например, ошибка в блоке подстановки приведет при циклическом сдвиге на нечетную величину и следующей операции замены к потере 8 битов информации. И это не учитывая влияние операций сложения. Таким образом, операции рассеивания (распространение влияния одного символа блока на все последующие) и перемешивания (операция разрушения статистических характеристик блока), примененные многократно, одновременно с шифрующими действиями способствуют распространению ошибок.

Данные анализа и других исследований [4] свидетельствуют о необходимости применения методов повышения эксплуатационной надежности криптосхем. К ним можно отнести мероприятия по техническому обслуживанию аппаратуры, применение методов понижения влияния шумов и излучений, а также избыточного кодирования данных.

Литература

1. Урбанович П. П., Алексеев В. Ф., Верниковский Е. А. Избыточность в полупроводниковых интегральных микросхемах памяти. – М.: Навука і тэхніка, 1995. – 262 с.

2. Мак-Вильямс Ф., Слоэн Н. Теория кодов, исправляющих ошибки / Под ред. Л. А. Басалыго. – М.: Связь, 1979. – 746 с.

3. Урбанович П. П., Романенко Д. М. Свойства и алгоритмы аппаратной реализации нового вида итеративных кодов для систем памяти // Новые информационные технологии: Материалы третьей международной конференции NITE'2000, Т. 2 – Мн.: БГЭУ, 2000. – С. 159–164.

4. Пацей Н. В., Скачков М. С., Урбанович П. П. Некоторые закономерности распределения ошибок в каналах передачи дискретной информации // IV МНТК Современные средства связи: Материалы конференции. / Известия белорусской инженерной академии. – 1999. – № 1 (7)/1. – С. 18.

5. Пацей Н. В. Метод повышения отказоустойчивости криптографических схем защиты информации // Труды БГТУ. Физико-математич. науки и информатика: Сб. ст. – Мн.: БГТУ, 2000. – Вып. 8.

6. Молдовян А. А., Молдовян Н. А., Советов Б. Я. Криптография. – СПб.: Лань, 2001. – 218 с.