алгоритма коррекции целесообразно использовать проверочную матрицу с $k_1=k_2$, т.е. с квадратной проверочной матрицей для линейного итеративного кода в плоскости.

Литература.

1. Урбанович П.П., Романенко Д.М. Свойства и алгоритмы аппаратной реализации нового вида итеративных кодов для систем памяти / / Новые информационные технологии: третья международная конференция NITE'2000, т. 2 – Мн.: БГЭУ,2000.– с. 159–164.

АЛГОРИТМЫ КОРРЕКЦИИ МНОГОКРАТНЫХ ОШИБОК ТРЕХМЕРНЫМ ИТЕРАТИВНЫМ КОДОМ С ПРОВЕРОЧНЫМИ СИМВОЛАМИ ПО ДИАГОНАЛИ

Д.М. Романенко

(БГТУ, г. Минск)

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 является совокупностью элементов четности всех элементов, содержащихся в каждой матрице. В дальнейшем на основе общих характеристик кодов прямого произведения получим основные характеристики кода и рассмотрим его техническую реализацию применительно к WSI.

В [1] описан метод получения трехмерного итеративного кода, а также подробно описаны его свойства. Так трехмерный итеративный код с проверочными символами по диагонали характеризуется следующими параметрами: избыточность **(r)**, минимальное кодовое расстояние **(d)**, скорость **(R)**, эффективность коррекции ошибок.

Величина избыточности равна

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

где $\mathbf{k_1}$ и $\mathbf{k_2}$ – количество строк и столбцов в линейном итеративном коде с проверочными символами по диагонали;

$\mathbf{k_3}$ – количество плоскостей в трехмерном итеративном коде.

Если $\mathbf{k_1 = k_2 = k_3}$, то

$$\mathbf{r=6*k_m^2,} \tag{2}$$

следовательно,

$$\mathbf{n=k+r=k_m^3+5*k_m^2.} \tag{3}$$

Тогда скорость **(R)** определяется следующим соотношением:

$$\mathbf{R=k/n=k_m/(k_m+5).} \tag{4}$$

Величина относительной избыточности $\mathbf{(r_{отн})}$ равна

$$\mathbf{r_{отн}=r/k=5/k_m.} \tag{5}$$

Минимальное кодовое расстояние трехмерного итеративного кода, согласно теореме 3 [1], будет **d**=10, а это значит, что код исправляет до четырех ошибок включительно.

Теорема [1] задает способ построения кода, но не дает алгоритмов декодирования, в то же время известно, что декодирование любого кода является наиболее трудоемкой операцией. Поскольку кодирование информации производится тривиальным перемножением проверочной матрицы на транспонированный вектор передаваемого сообщения (можно использовать алгоритмы построения проверочных матриц для дополненного итеративного кода), в дальнейшем сосредоточимся именно на декодировании. 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.

В докладе представляется разработанный специальный алгоритм коррекции ошибок кратностью не выше четырех. Данный алгоритм строится на принципах, аналогичных положенным в основу коррекции многократных ошибок дополненным линейным итеративным кодом [1], т. е. с использованием **hvd** классификации с учетом проверок **z**–рядов (т.е. проверок на четность вдоль оси **OZ**) [1].

Если ограничить кратность ошибок, которые должны быть исправлены до **(d – 1) / 2 + 1**, если **d** – нечетное, и **(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:

$$\mathbf{r = 2*(k1+k2)*k3 + k1*k2 + 1.} \qquad (6)$$

При $\mathbf{k_1 = k_2 = k_3}$ величина избыточности будет равна

$$\mathbf{r = 5* k_m^2 + 1,} \qquad (7)$$

следовательно,

$$\mathbf{n=k+r=k_m^3+5*k_m^2 +1.} \qquad (8)$$

Тогда скорость **(R)** определяется следующим соотношением:

$$\mathbf{R=k/n=5* k_m^2 + 1/(k_m^3+5*k_m^2 +1).} \qquad (9)$$

Величина относительной избыточности ($\mathbf{r_{отн}}$) равна

$$\mathbf{r_{отн}=r/k=k_m5* k_m^2 + 1 / k_m^3.} \qquad (10)$$

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.

Литература.

1. Урбанович П.П., Романенко Д.М. Свойства и алгоритмы аппаратной реализации нового вида итеративных кодов для систем памяти / / Новые информационные технологии: третья международная конференция NITE'2000, т. 2 – Мн.: БГЭУ,2000.– с. 159–164.

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

ИСПОЛЬЗОВАНИЕ ГИПЕРБОЛИЧЕСКИХ БРАУЗЕРОВ ДЛЯ ВИЗУАЛИЗАЦИИ БОЛЬШИХ ИЕРАРХИЧЕСКИХ СТРУКТУР

В.С. Скращук, В.М. Волчкевич, Д.Н.Кортелев

(ГрГУ, г. Гродно)

В работе предлагается метод создания динамически настраиваемых эффективных средств для графического представления иерархических структур, основанный на технологии “фокус+контекст”.

Рост объема информации, производимой обществом, заставил обратить внимание на факт отсутствия эффективных средств для ее визуального представления, исследования и поиска. В связи с этим вырос интерес к аспектам информационного исследования средств и методов визуализации, цель которого состоит в том, чтобы обнаружить и развить пути усиления человеческого познания.

Один из способов усиления познания видится в том, чтобы увеличить количество информации, которая может быть помещена в центр внимания пользователей.

Среди новых подходов в области создания средств поиска для систем управления данными выделяется метод визуальных моделей, основанный на технологии «фокус+контекст»[1]. Метод «фокус+контекст» - один из информационных методов визуализации, направленных на увеличение количества информации, которая может быть показана пользователю за счет того, что интересующие его данные выводятся на передний план и в то же время сохраняется структура даже очень больших наборов данных.