

П.П. Урбанович, профессор; Д.М. Романенко, аспирант; А.В. Орлов, ассистент

## ОСОБЕННОСТИ РЕАЛИЗАЦИИ ТРЕХМЕРНОГО ИТЕРАТИВНОГО КОДА ДЛЯ СИСТЕМ ПАМЯТИ СБИЕ С ИНТЕГРАЦИЕЙ НА ПОЛУПРОВОДНИКОВОЙ ПЛАСТИНЕ

The article are considered some aspects of realizing a new type of the code – three-dimensional iterative code, that will allow to reach the minimal redundancy and the maximal speed in systems of semi-conductor memory with wafer scale integration.

Развитие информационных технологий базируется, в основном, на улучшении эксплуатационно-технических параметров компьютеров и их комплектующих. Современные винчестеры на основе магнитных дисковых накопителей обладают высокой информационной емкостью, однако скорость их работы очень низкая. Перспективным направлением улучшения быстродействия современных жестких дисков является переход от магнитных носителей к полупроводниковым. Это предполагает создание сверхбольших систем памяти, интегрированных на полупроводниковых СБИС-пластинах (Wafer Scale Integration).

Технология СБИС-пластин позволяет получить значительно более высокую плотность упаковки памяти и уменьшить время доступа к ней. Среднее время доступа к «полупроводниковому диску» в десятки раз меньше среднего времени доступа обычных накопителей на магнитных дисках.

В современных ЗУ, отличающихся большой емкостью, доля ошибок в общем количестве ошибок ЗУ из-за нарушения функциональности в элементах логики не превышает несколько процентов. Поэтому основное внимание необходимо уделить нейтрализации ошибок, возникающих в процессе хранения информации в ячейках памяти. Для этого необходимы корректирующие коды, позволяющие корректировать не только одиночные, но многократные ошибки.

Для защиты информации в таких системах памяти предлагается использовать трехмерный линейный итеративный код (ТИК) [1].

Достоинства ТИК:

- высокая скорость кодирования и декодирования;
- простота алгоритмов коррекции ошибок;
- эффективность для больших объемов информации;
- схожесть с архитектурой полупроводниковой памяти (в частности, с архитектурой «полупроводниковых пластин»).

Недостатки ТИК:

- высокая избыточность (особенно при малых длинах информационных слов).

Для нивелирования высокой избыточности трехмерного линейного итеративного кода необходимо знать следующее:

- корректирующий код защищает не отдельные информационные слова, а некоторое информационное пространство, представляющее собой совокупность кристаллов (секций кристаллов) одной либо разных пластин (предполагаемое расположение информационных пространств на пластине представлено на рисунке).



Рисунок. Схема расположения информационных пространств на пластине

Количество информационных пространств ( $N_{инф.пр.}$ ) на пластине равно

$$N_{инф.пр.} = N_{h-сек} \cdot k, \quad (1)$$

где  $N_{h-сек}$  – количество горизонтальных секций (при использовании 8-разрядных микросхем емкостью 256 Мбит с размером секции 64 x 64 бит  $N_{h-сек} = 8192$ );  $k$  – количество групп микросхем;

- при накоплении в информационном пространстве трех ошибок они должны исправляться, так как появление четвертой ошибки может привести к потере информации;
- вышеописанная особенность приводит к необходимости создания некоторого дополнительного модуля, включающего в себя алгоритмы формирования проверочных символов и коррекции ошибок, который либо постоянно, либо в определенные моменты времени будет сканировать память и исправлять накопившиеся ошибки;
- процессы записи и чтения информации в целом изменений не претерпевают (построчная запись информации);
- при записи информации в память происходит корректировка соответствующих проверочных символов (все проверочные символы не пересчитываются);
- при необходимости считывания информации проверяется плоскость на наличие ошибок итеративным кодом. Если итеративный код показывает на наличие ошибок, то запускается модуль коррекции, а затем дается разрешение на считывание информации. В противном случае информацию можно считывать без запуска модуля коррекции ошибок;
- записывать информацию целесообразно в минимальное количество информационных пространств (по возможности в минимальное количество информационных плоскостей);
- после считывания информации данные из ячеек памяти не удаляются (чтобы не пересчитывать проверочные символы).

Вышеотмеченные особенности позволяют: реализовать основное достоинство трехмерных итеративных кодов – схожесть со структурой памяти, что должно существенно упростить алгоритмы формирования проверочных символов; более эффективно использовать информационное пространство.

В таблице приведены значения относительной избыточности для ТИК.

Таблица

Длина информационного слова, $k$	Размерность матрицы, $k_1 \times k_2 \times k_3$	Относительная избыточность, $r_{отн}$
$2^3$	$2 \times 2 \times 2$	70,3
$2^6$	$4 \times 4 \times 4$	48,8
$2^9$	$8 \times 8 \times 8$	29,8
$2^{12}$	$16 \times 16 \times 16$	13,1
$2^{18}$	$64 \times 64 \times 64$	4,54
$2^{21}$	$128 \times 128 \times 128$	2,31

Как видно из таблицы, при увеличении длины информационного слова  $k$  относительная избыточность  $r_{отн}$  резко уменьшается. При длине информационного слова  $k=2^{18}$  составляет менее 5 %. Таким образом, для систем памяти целесообразно использовать код с проверочной матрицей не менее  $64 \times 64 \times 64$ .

### Вывод

В современной полупроводниковой памяти используются микросхемы емкостью от 16 до 256 Мбит. Так, фирма Legacy electronics для полупроводниковой памяти сверхбольшой емкости использует  $(64 \text{ Мбит}) \times 4$  микросхемы памяти.

Для систем памяти СБИЕ с интеграцией на полупроводниковой пластине целесообразно использовать 8- и более разрядные микросхемы. При использовании трехмерного итеративного кода с проверочной матрицей  $64 \times 64 \times 64$  избыточность памяти не будет превышать 6%. Быстродействие систем памяти будет существенно зависеть от схемной реализации вышеописанного.

### 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. – Мн.: БГЭУ, 2000. – Т. 2. – С. 159–164.