

Помехоустойчивое кодирование для субмикронных динамических ОЗУ

К.А. Петров

НИИ системных исследований РАН, Москва, bemsy@mail.ru

Аннотация — Для защиты динамических субмикронных ОЗУ от сбоев наилучшим по критериям количества кристаллов, избыточности и быстродействия кодер-декодера методом с помехоустойчивым кодированием при отсутствии многократных сбоев является метод, использующий кодирование с восстановлением байта данных. При наличии многократных сбоев наилучшим по критерию количества кристаллов является метод, использующий технологию Chipkill с помехоустойчивым кодированием, исправляющим пакетные ошибки.

Ключевые слова — помехоустойчивое кодирование, ДОЗУ, кратные ошибки, временный функциональный отказ.

I. ВВЕДЕНИЕ

Динамические оперативные запоминающие устройства (ДОЗУ), использующиеся в микропроцессорных системах, отличаются от статических запоминающих устройств (СОЗУ) высокой плотностью упаковки и низкой стоимостью. При этом они имеют меньшую боеустойчивость, в том числе к воздействию ядерных частиц, что ограничивает их применение в радиационно-стойких приложениях. Самыми распространенными эффектами, возникающими в ДОЗУ вследствие воздействия ядерных частиц, являются однократные сбои, ведущие к переключению одного хранящегося в кодовом слове бита. Помимо однократных сбоев, возникают также многократные сбои, приводящие к ошибке в нескольких смежных битах кодового слова. Большое значение эти ошибки приобретают при снижении проектных норм производства микросхем ниже 180 нм [1]. Также большое значение имеют временные функциональные отказы, являющиеся следствием борьбы с физическим эффектом зашелки.

Решена задача оценки методов для защиты данных в ДОЗУ при временном функциональном отказе в случае наличия или отсутствия многократных сбоев. Рассматривались ДОЗУ, использующие самые распространенные на сегодняшний день разрядности информационного слова: 32, 64 и 128 бит.

Для борьбы с однократными сбоями широко используются помехоустойчивые коды, исправляющие одну ошибку (SEC – single-error-detection), либо дополнительно обнаруживающие двукратную ошибку (SEC-DED – single-error-detection double-error-correction) в кодовом слове [2]. Также для борьбы с накоплением ошибок в памяти широко применяется метод самовосстановления информации в ОЗУ, известный как техно-

логия scrubbing [3]. Для борьбы с многократными сбоями в одном кодовом слове и с временными функциональными отказами отдельными СБИС ДОЗУ используется технология Chipkill [4], также называемая Chipkill-кодированием. Организация хранения данных при использовании технологии Chipkill-кодирования позволяет переводить многократную ошибку в одном кодовом слове в однократную ошибку в нескольких кодовых словах, которую возможно исправить методами обычного помехоустойчивого кодирования, используемого для исправления одной и обнаружения двух ошибок в кодовом слове. Так как Chipkill-кодирование требует большого количества микросхем (равного количеству битов кодового слова), предложены методы использования помехоустойчивых кодов, исправляющие смежные ошибки кратности два [5] и более [3] вместе с Chipkill-кодированием. Эти методы позволяют при той же помехоустойчивости снизить минимальное количество кристаллов ДОЗУ соответственно в два и более раза. Помимо кодов, исправляющих ошибки, существуют методы восстановления потерянной при временном функциональном отказе информации на базе широко используемых SEC-DED кодов [6].

II. ПОМЕХОУСТОЙЧИВОЕ КОДИРОВАНИЕ

При записи в информации ОЗУ с помощью схемы кодера производится вычисление проверочных битов из информационного слова.

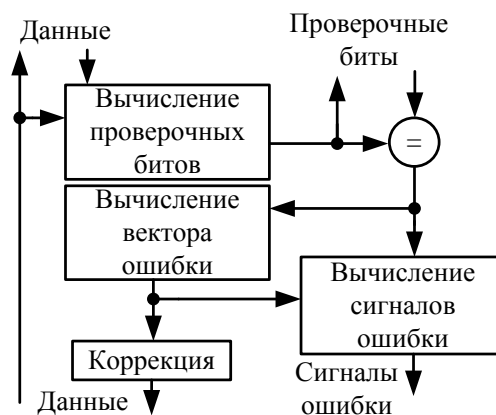


Рис. 1. Функциональная схема кодер-декодера для линейного блочного кода

Линейные блочные SEC-DED коды типа Хэмминга, исправляющие однократные и детектирующие двукратные ошибки в кодовом слове, наиболее часто применяются для повышения надежности ОЗУ. Для борьбы с ошибками кратности больше единицы, такими как многократная ошибка в кодовом слове или стирание данных вследствие временного функционального отказа, использования SEC-DED кодов недостаточно. В связи с тем, что в основе всех оцененных методов лежат линейные коды, используемые кодер-декодеры не имеют значительных отличий, кроме тех, которые оговорены отдельно для каждого конкретного случая. Общая функциональная схема кодер-декодера для всех оцененных методов представлена на рис. 1.

III. ТЕХНОЛОГИЯ CHIPKILL

Для Chipkill-кодирования [4] 72-разрядного кодового слова используется 72 СБИС ДОЗУ (рис. 2). Запись и чтение данных происходит только пакетами по несколько кодовых слов, количество которых равно разрядности отдельной СБИС. В основе приведенного примера Chipkill-кодирования лежит SEC-DED код Хэмминга (72, 64), позволяющий исправлять одну и обнаруживать две ошибки в кодовом слове. Также используется кодер-декодер этого кода без изменений. Основные отличия от ДОЗУ с обычным кодированием заключаются в контроллере ДОЗУ, который в случае применения технологии Chipkill-кодирования использует пакетные обращения к памяти с минимальным количеством слов в пакете равным четырем.

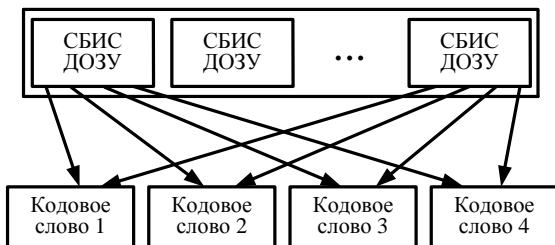


Рис. 2. Функциональная схема применения технологии Chipkill в динамическом ОЗУ

В результате использования Chipkill-кодирования при возникновении однократной ошибки происходит ее исправление декодером SEC-DED кода. При возникновении многократной ошибки она при декодировании представляется в виде однократной ошибки в нескольких кодовых словах, количество которых равно кратности ошибки. Полученные однократные ошибки в свою очередь также исправляются декодером SEC-DED кода. Функциональный отказ представляется при декодировании как наличие однократной ошибки во всех кодовых словах, хранящихся в памяти. При этом в случае возникновения еще одной ошибки, помимо выявленной в отказавшей СБИС, кодовое слово определяется как имеющее неисправимую ошибку.

Основным недостатком Chipkill-кодирования является ограничение снизу по количеству микросхем динамической памяти. Это ограничение несущественно для систем с большим объемом ДОЗУ, где используется близ-

кое к значению разрядности кодового слова количество СБИС, но представляет собой препятствие для микропроцессорных систем, где не требуется или невозможно с точки зрения конструкции использовать большое количество СБИС ДОЗУ. Еще одним недостатком систем динамической памяти с применением технологии Chipkill является необходимость чтения и записи данных блоками минимум по 4 кодовых слова (пакетное обращение к памяти), что может существенно снизить быстродействие для микропроцессорной системы, использующей информационные слова малой длины.

Необходимо отметить, что для систем динамической памяти с 32-разрядным информационным словом и SEC-DED кодом (39, 32) и Chipkill-кодированием требуется соответственно 39 СБИС. Для систем с разрядностью информационного слова 128 бит и SEC-DED кодом (128, 137) использование Chipkill-кодирования потребует соответственно 137 СБИС ДОЗУ.

IV. МЕТОДЫ С ИСПОЛЬЗОВАНИЕМ ПОМЕХОУСТОЙЧИВЫХ КОДОВ, ИСПРАВЛЯЮЩИХ СМЕЖНЫЕ ОШИБКИ

Для снижения количества СБИС ДОЗУ следует использовать класс кодов Хэмминга, известный как DAEC (double-adjacent-error-correction) коды, исправляющие двукратные смежные ошибки [5]. При этом DAEC коды, имеющие одинаковое с SEC-DED кодами количество проверочных битов, являются SEC-DAEC (single-error-correction double-adjacent-error-correction) кодами, то есть не обнаруживают всех двукратных ошибок. В случае возникновения двукратной несмежной ошибки существует вероятность того, что декодер не обнаружит ее, как неисправимую, а определит, как двукратную смежную и произведет неверное исправление с ошибочным выставлением сигнала исправимой ошибки. Необходимо также отметить, что при адаптации этих SEC-DAEC кодов к рассматриваемой задаче необходимо исправлять не все двукратные смежные ошибки, а только те из них, которые физически относятся к сбоям в одной микросхеме памяти. После адаптации количество не обнаруживаемых кодом двукратных несмежных ошибок снижается с 34 до 17%. Результатом использования таких кодов будет снижение устойчивости ДОЗУ к двукратным несмежным ошибкам, а также снижение вдвое минимального количества СБИС – до 36.

Для детектирования всех двукратных ошибок необходимо добавить к кодовому слову дополнительные проверочные биты. Получившийся в результате добавления проверочных битов SEC-DED-DAEC (single-error-correction double-error-detection double-adjacent-error-correction) код (76, 64) отличается от SEC-DAEC кода (72, 64) возможностью обнаружения всех несмежных двукратных ошибок. Для использования такого кода с Chipkill-кодированием потребуется 38 СБИС ДОЗУ.

Для систем с 32-разрядным информационным словом и SEC-DAEC кодом (40, 32) и Chipkill-кодированием требуется соответственно 20 СБИС, а для SEC-DED-DAEC кода (42, 32) требуется 21 СБИС.

Также существует широкий класс кодов, исправляющих пакетные ошибки [3]. S4EC-DED (single-4-bit-error-correction double-error-detection) код (76, 64) требует для своей реализации с Chipkill-кодированием 19 микросхем.

Кодер-декодеры для кодов, исправляющих смежные или пакетные ошибки, являются менее быстродействующими и требуют больших аппаратных затрат по сравнению с кодер-декодерами для обычных SEC-DED кодов. Основные отличия кодер-декодеров заключаются в количестве элементов в схеме вычисления проверочных битов вследствие большего количества самих проверочных битов. Также значительно увеличиваются аппаратные затраты на схему вычисления вектора ошибки, так как количество различных исправляемых кодов ошибок увеличивается. Вследствие увеличения физического размера декодера и времени его работы вероятность возникновения в нем сбоя в комбинационной логике увеличивается, что необходимо учитывать при разработке микропроцессорной системы. В случае, когда снижение надежности кодер-декодера является значительным, решением может служить использование сбоеустойчивых библиотечных элементов при проектировании кодер-декодера или модульное резервирование кодер-декодера. В связи с большим количеством проверочных битов, большими аппаратными затратами и низким быстродействием не рассматриваются коды, исправляющие ошибки кратности более четырех.

S4EC-DED код (44, 32) для 32-разрядного слова требует для своей реализации с Chipkill-кодированием 11 микросхем. А в случае использования S4EC-D4ED кода (144, 128) для ДОЗУ с 128-разрядным информационным словом количество СБИС составит 26 штук.

V. МЕТОД С ИСПОЛЬЗОВАНИЕМ ПОМЕХОУСТОЙЧИВОГО КОДА, ВОССТАНАВЛИВАЮЩЕГО БАЙТ В КODOVOM СЛОВЕ

В случае системы динамической памяти, состоящей из нескольких СБИС ДОЗУ, временный функциональный отказ представляет собой временный отказ одной из СБИС с полной потерей всех хранящихся в ней данных. В этом случае в каждом кодовом слове происходит стирание одного «байта» информации, где размер «байта» равен разрядности шины данных СБИС ДОЗУ, из которых состоит динамическая память.

Одной из наиболее перспективных альтернатив технологии Chipkill является метод, использующий SbER коды (single-byte-error-repair), восстанавливающие байт данных [6]. Проверочная матрица кода (72, 64), восстанавливающего байт, составляется на основе проверочной матрицы SEC-DED кода таким образом, что любой из девяти байтов кодового слова можно вычислить из остальных восьми также как вычисляется проверочный байт при кодировании информационного слова при записи в память. Таким образом, при временном отказе одной из СБИС ДОЗУ существует возможность перезаписи данных, ранее хранящихся в ней. Необходимо отметить, что для реализации кодирования, восстанавливающего байт, для системы памяти с 64-разрядным информационным словом требуется 9 СБИС ДОЗУ, что

является стандартом для сбоеустойчивых запоминающих устройств на СБИС динамической памяти, использующих обычное SEC-DED-кодирование.

На рис. 3 представлена функциональная схема системы динамической памяти с кодированием, позволяющим восстановить байт данных. К каждой из девяти СБИС ДОЗУ добавляется контроллер питания, содержащий датчик тока, регистрирующий повышение потребления вследствие эффекта защелки и производящий в этом случае отключение питания микросхемы. Это приводит к временному функциональному отказу устройства вместо возможного постоянного отказа вследствие повреждения СБИС ДОЗУ. После обратного включения питания производится последовательный перебор всех адресов ДОЗУ с помощью схемы самовосстановления. Для каждого кодового слова производится вычисление байта, который хранился во временно отказавшем СБИС ДОЗУ. В результате перебора происходит восстановление данных.

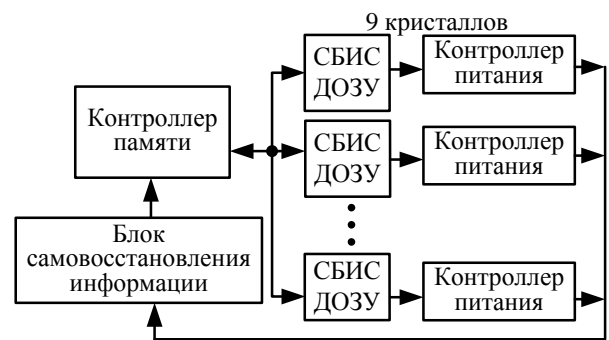


Рис. 3. Функциональная схема системы динамической памяти со SbER-кодированием

Помимо достоинств, таких как малое количество СБИС ДОЗУ и отсутствие необходимости пакетного обращения к памяти, у метода SbER-кодирования существует ряд недостатков.

Во-первых, в случае возникновения временного функционального отказа необходимо время для восстановления данных, в течение которого система динамической памяти будет неработоспособна. В случае других методов такое время желательно, но не необходимо. Отличие в том, что в случае возникновения дополнительной ошибки после временного функционального отказа при использовании метода с Chipkill-кодированием будет обнаружена неисправимая ошибка, в случае использования метода с кодированием, позволяющим восстанавливать байт данных, ошибка обнаружена не будет.

Во-вторых, отсутствие защиты от многократных сбоев, не вызывающего отключения питания микросхемы. В случае отсутствия эффекта защелки контроллер питания не регистрирует ошибки, и она не будет исправлена кодер-декодером. Для борьбы со вторым недостатком авторами метода предлагается использовать микросхемы памяти от производителя, гарантирующего отсутствие многократных сбоев.

В-третьих, контроллеры питания целесообразно применять только после анализа амплитуд скачков токов потребления, характерных для каждой конкретной системы динамической памяти, чтобы избежать ложных срабатываний.

Кодер-декодер для SbER-кодирования представляет собой кодер-декодер SEC-DED кода, дополненный схемой вычисления любого из байтов кодового слова. В обычном режиме работы используется только основная часть кодер-декодера, что делает его быстродействие равным быстродействию кодер-декодера обычного SEC-DED кода. Дополнительная схема используется при временном функциональном отказе и ее быстродействие сравнимо с быстродействием схемы вычисления проверочных битов. Площадь, занимаемая на кристалле дополнительной схемой, значительно превышает площадь схемы вычисления проверочных битов в связи с тем, что возможно вычисление любого из девяти байтов кодового слова.

Для ДОЗУ с 32-разрядным информационным словом является возможным уменьшить проверочную матрицу кода (72, 64) до проверочной матрицы кода (40, 32), сохранив все необходимые свойства кода. Такая система с SbER-кодированием потребует 5 СБИС. В случае применения SbER-кодирования на удвоенном коде Хсяо (72, 64) для системы с 128-разрядным информационным словом и СБИС ДОЗУ с 16-битной шиной данных количество СБИС остается равным 9, как и для систем с 64-разрядным кодовым словом.

VI. ЗАКЛЮЧЕНИЕ

Сравнительный анализ методов повышения надежности ДОЗУ, обеспечивающих стойкость к кратным ошибкам в данных и временному отказу микросхем, представлен в табл. 1.

Таблица 1

Сравнительный анализ методов повышения стойкости ДОЗУ к кратным ошибкам в данных и временному отказу микросхемы

Вид кодирования	№1	№2	№3	№4	№5
Количество СБИС	72	36	38	19	9
Пакетное чтение и запись	Да	Да	Да	Да	Нет
Обнаружение всех двукратных ошибок	Да	Нет	Да	Да	Да
Стойкость к многократным сбоям	Да	Да	Да	Да	Нет

Примечание: №1 – Chipkill, №2 – Chipkill+SEC-DAEC, №3 – Chipkill+SEC-DED-DAEC, №4 – Chipkill+S4EC-DED, №5 – SbER.

Результаты сравнительного анализа кодов с исправлением байта (SbEC), кодов с исправлением смежных ошибок (DAEC) при использовании технологии Chipkill, а также кодов с восстановлением байта (SbER) позволяют сделать вывод о том, что для защиты субмикронных ДОЗУ от сбоев наилучшим по критериям аппаратурной

избыточности, количества СБИС ДОЗУ и быстродействия (на уровне широко применяемого кода Хсяо [7]) методом при отсутствии многократных сбоев является метод, использующий коды с восстановлением байта.

При наличии многократных сбоев наилучшим по критерию количества СБИС ДОЗУ является метод, использующий технологию Chipkill с кодом, исправляющим пакетные ошибки кратности четыре (Chipkill+S4EC-DED). Он также является наилучшим по критерию сбоеустойчивости (на уровне значительно более сбоеустойчивых, чем коды Хэмминга, БЧХ-кодов [8]), если не учитывать большой физический размер и малое быстродействие кодер-декодера, что может оказать влияние на общую надежность этого метода.

Таким образом, при разработке устойчивого к временному функциональному отказу одного из кристаллов динамического ОЗУ необходимо либо снижать стойкость к многократным сбоям, либо увеличивать количество кристаллов. Нахождение оптимума является задачей, решаемой при разработке системы динамической памяти.

Перспективным направлением являются использование каскадного кодирования для компенсации недостатков SbER-кодов, а также использование помехоустойчивых кодов для исправления пакетных ошибок с учетом повышения надежности схем более сложных кодер-декодеров. Также важным является более точное сравнение схем кодер-декодеров для реализации различных методов по критериям быстродействия, аппаратурной избыточности и сбоеустойчивости.

ЛИТЕРАТУРА

- [1] Borucki L., Schindlbeck G., Slayman C. Comparison of accelerated DRAM soft error rates measured at component and system level // IEEE 46th Annual International Reliability Physics Symp. Phoenix, 2008. P. 482–487.
- [2] Блейхут Р. Быстрые алгоритмы цифровой обработки сигналов: Пер. с англ. М.: Мир, 1989. 448 с.
- [3] Fujiwara E., Pradhan D. K. Error-control coding in computers // IEEE Computer. 1990. P. 63–72.
- [4] Timothy J. A white paper on the benefits of chipkill correct ECC for PC server main memory [IBM Microelectronics Division]. URL: http://ece.umd.edu/courses/enee759h.S2003/references/ibm_chipkill.pdf (дата обращения: 09.10.2011).
- [5] Dutta A., Touba N. A. Multiple bit upset tolerant memory using a selective cycle avoidance based SEC-DED-DAEC code // 25th IEEE VLSI Test Symp., 2007. P. 349–354.
- [6] Pontarelli S., Cardarilli G. C., Re M., Salsano A. Error correction codes for SEU and SEFI tolerant memory systems // DFT., 2009. P. 425–430.
- [7] Hsiao M. Y. A class of optimal minimum odd-weight-column SEC-DED codes // IBM J. Res. Develop. 1970. Vol. 14. P. 395–401.
- [8] Naseer R., Draper J. DEC ECC design to improve memory reliability in sub-100 nm technologies // Proc. 15th IEEE Int'l Conf. on Electronics, Circuits and Systems. 2008. P. 586–589.