

ИНФОРМАТИКА

УДК 681.326.7

ПОВЫШЕНИЕ ДОСТОВЕРНОСТИ ТЕКУЩЕГО КОНТРОЛЯ ОЗУ

А.А. ИВАНЮК, С.Б. МУСИН

*Белорусский государственный университет информатики и радиоэлектроники
П. Бровка, 6, Минск, 220013, Беларусь**Поступила в редакцию 15 декабря 2004*

В данной статье рассматриваются обнаруживающая и диагностическая способность эффективного метода текущего контроля ОЗУ — адаптивного сигнатурного анализа (АСА). Исследуются способы повышения достоверности данного метода, а также предлагается новый подход для обнаружения с помощью метода АСА четырехкратных ошибок в ОЗУ.

Ключевые слова: текущий контроль ОЗУ, адаптивный сигнатурный анализ.

Введение

Системы обработки и хранения информации, вычислительные и управляющие комплексы — компьютерные системы — нашли широкое применение в самых различных областях. В настоящее время много внимания уделяется безопасности таких систем. При этом выделяют два аспекта:

- обеспечение целостности и особенно конфиденциальности хранимых данных;
- обеспечение надежности функционирования, устойчивости к возникновению ошибок и сбоев в работе.

Требование безопасности системы — это в первую очередь требование безопасности ее функциональных узлов. Неотъемлемой составляющей любой компьютерной системы является оперативное запоминающее устройство (ОЗУ). ОЗУ — один из самых ненадежных функциональных узлов компьютерных систем [1]. Ошибка в данных, хранимых ОЗУ, может повлечь как выдачу системой неправильного результата, так и нарушение работы всей системы вплоть до получения к ней несанкционированного доступа [2]. Вот почему сегодня одной из важных задач является контроль правильности функционирования ОЗУ.

Тестирование и контроль ОЗУ

Хранение информации в ОЗУ, как правило, осуществляется в виде двоичных кодов с символами 0 и 1. Любое искажение этих кодов называется ошибкой. Под кратностью ошибки понимается количество искаженных информационных символов.

Причиной ошибки может быть неисправность либо сбой в работе ОЗУ. Для выявления неисправностей в процессе эксплуатации могут применяться методы тестирования, а для обнаружения ошибочных данных — методы текущего контроля.

При тестировании (рис. 1) предполагается наличие дополнительного устройства, которое служит для генерации тестовых воздействий и обработки результатов тестирования. Тестирование осуществляется во время простоя системы либо после подачи внешнего сигнала, по

которому ОЗУ переключается в соответствующий режим. К достоинствам данного подхода можно отнести возможность проведения исчерпывающего тестирования для выявления любых типов неисправностей, к недостаткам — тот факт, что ошибки, возникшие в результате воздействия внешних факторов (альфа-частицы естественной радиации окружающей среды), в промежутках между тестированием не могут быть обнаружены.

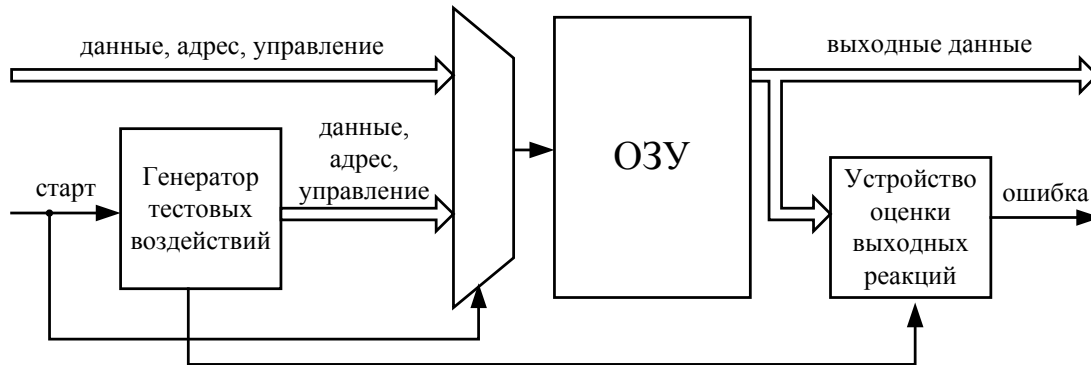


Рис. 1. Схема тестирования ОЗУ

Текущий контроль (рис. 2) в отличие от тестирования не нарушает процесс нормального функционирования ОЗУ.

При текущем контроле ОЗУ с применением корректирующих кодов совокупность запоминающих элементов рассматривается как канал передачи информации, в котором информация передается не в пространстве, а во времени. Данные, поступающие на хранение в ОЗУ, предварительно обрабатываются кодером, который вносит избыточность с целью обнаружения и/или исправления ошибок определенной кратности. При считывании происходит декодирование: на основании информационных и контрольных символов рассчитывается синдром. Значение синдрома определяется конфигурацией ошибок, оно не зависит от информационных символов. Таким образом, на основании синдрома ошибка может быть обнаружена и, если возможно, исправлена. Обнаруживающая способность данного подхода зависит от характеристик используемого корректирующего кода. Существенным недостатком текущего контроля ОЗУ с применением корректирующих кодов являются значительные аппаратные затраты на реализацию.

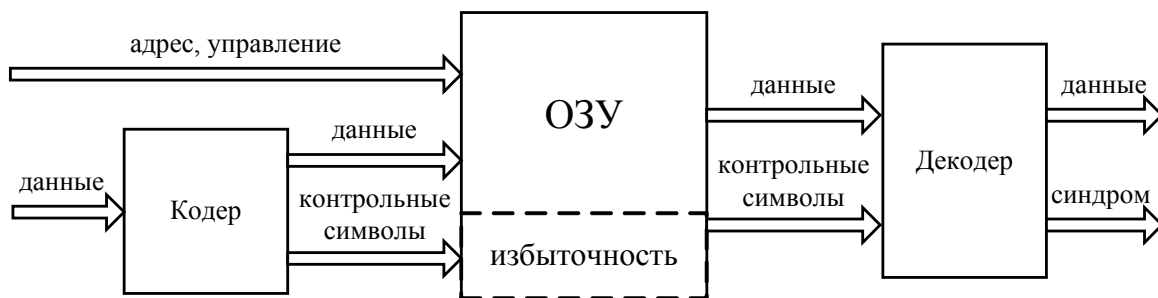


Рис. 2. Схема контроля ОЗУ

Адаптивный сигнатурный анализ

В [3] был предложен подход, основанный как на принципе тестирования, так и на принципе текущего контроля — адаптивный сигнатурный анализ (АСА). Затраты аппаратуры на реализацию АСА значительно меньше, чем при использовании традиционного подхода текущего контроля ОЗУ с использованием корректирующих кодов.

В основу АСА положен метод неразрушающего тестирования ОЗУ, предложенный Ко-неманом в 1986 г. [4]. Данный метод основан на свойстве линейности сигнатурного анализатора.

Для проведения неразрушающего тестирования содержимое ОЗУ сжимается на сигнатурном анализаторе в соответствии с фазами чтения неразрушающего алгоритма тестирования. Полученная эталонная сигнатура сохраняется. Далее подаются тестовые воздействия, а получаемые выходные реакции снова сжимаются на сигнатурном анализаторе. Полученная сигнатура сравнивается с эталонной. Если сигнатуры различны, то произошла ошибка в работе ОЗУ. Главным недостатком такого подхода является необходимость вычисления эталонной сигнатуры каждый раз после изменения содержимого ОЗУ.

Метод АСА позволяет производить коррекцию эталонной сигнатуры, не выполняя сжатие всего содержимого ОЗУ, что позволяет использовать сигнатурный анализатор в качестве инструмента текущего контроля. Основное отличие метода АСА от классического сигнатурного анализа – сжатие на сигнатурном анализаторе вместо информационных символов их адресов.

Процедуру проведения текущего контроля ОЗУ с применением АСА можно описать следующим образом (рис. 3).

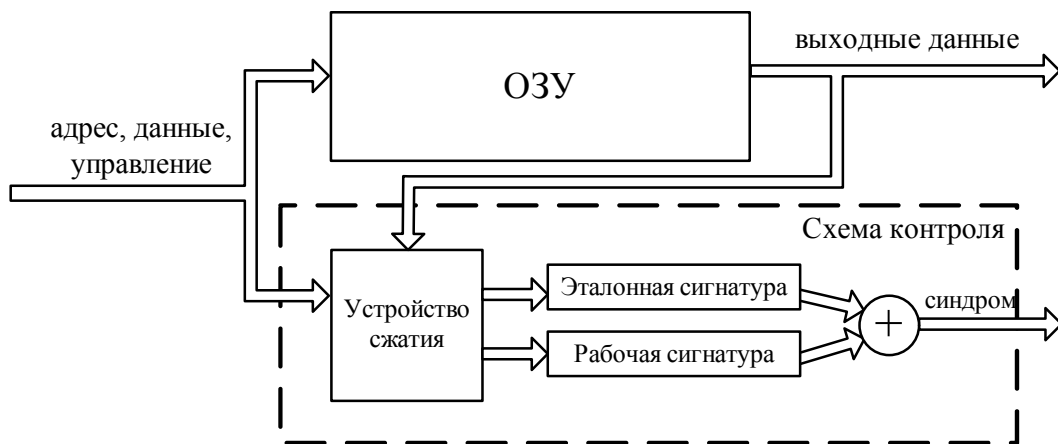


Рис. 3. Схема контроля ОЗУ с использованием АСА

Изначально значения рабочей и эталонной сигнатур равны нулю. При изменении содержимого ОЗУ текущее значение эталонной сигнатуры суммируется по модулю два со значениями адресов ячеек памяти, поменявших свое значение. Если необходимо провести контроль хранящихся данных, функционирование ОЗУ приостанавливается. Рабочая сигнатура вычисляется как сумма по модулю два ячеек ОЗУ со значением 1. Рабочая и эталонная сигнатуры сравниваются. Факт несовпадения эталонной и рабочей сигнатур говорит о том, что ОЗУ содержит ошибочные данные. Адресация ячеек не может начинаться с нулевого адреса, так как каждый адрес, сжимаемый на анализаторе, должен изменять значение сигнатуры.

Обнаруживающая и диагностическая способность метода АСА

Рассмотрим бит-ориентированное ОЗУ, состоящее из массива запоминающих элементов размерности 2^N , N — разрядность шины адреса. Совокупность адресов рассматриваемого ОЗУ является векторным пространством $A = \{a_1, a_2, \dots, a_i, \dots, a_{2^N}\}$, где $a = \{\alpha_0, \alpha_1, \dots, \alpha_j, \dots, \alpha_{N-1}\}$, $\alpha \in GF(2)$.

Пусть x_i — значение, хранящееся по адресу a_i в A , $x_i \in GF(2)$; SL — значение эталонной сигнатуры; SW — значение рабочей сигнатуры; S — синдром; \oplus — оператор суммирования в $GF(2)$.

Тогда при каждом изменении значения x_i на новое значение x'_i коррекция эталонной сигнатуры проводится по следующей формуле:

$$SL' = SL \oplus a_i, \forall x_i \neq x'_i, \text{ где } SL - \text{старое, } SL' - \text{новое значения сигнатуры.}$$

Рабочая сигнатура вычисляется как

$$SW = \bigoplus_{i=1}^N a_i, \forall x_i = 1.$$

Если содержимое ОЗУ корректно, то $SL = SW$ или $S = SW \oplus SL = 0$.

Утверждение 1. Ошибка не может быть обнаружена, если после ее возникновения $S = 0$.

Пусть произошла ошибка кратности $k=1$: значение ячейки по адресу a_{e_1} изменилось.

Тогда:

$$1) \quad SW_{e_1} = \bigoplus_{i=1}^N a_i \oplus a_{e_1}, \forall x_i = 1, x'_{e_1} = 1, x_{e_1} = 0, \quad S = SW \oplus SL = a_{e_1} \neq 0,$$

$$2) \quad SW_{e_1} = \bigoplus_{i=1}^{N-1} a_i, \forall x_i = 1, x'_{e_1} = 0, x_{e_1} = 1, \quad S = SW \oplus SL = a_{e_1} \neq 0.$$

Следовательно, однократные ошибки обнаруживаются и исправляются.

Пусть произошла ошибка кратности $k=2$: значения ячеек по адресам a_{e_1}, a_{e_2} изменились. Тогда:

$$S = SW \oplus SL = a_{e_1} \oplus a_{e_2},$$

$$a_{e_1} \oplus a_{e_2} = 0, \text{ если } a_{e_1} = a_{e_2} \text{ (свойство суммирования в } GF(2)), \text{ причем}$$

$a_i \neq a_j, \forall a_i \in A, a_j \in A$ (свойство векторного пространства). Следовательно, двукратные ошибки обнаруживаются.

Повышение обнаруживающей и диагностической способности метода АСА

Для обнаружения ошибок более высокой кратности ($k > 2$) в [5, 6] было предложено сжимать не адреса ОЗУ, а некоторые бинарные векторы, которые ставятся в соответствие каждому реальному адресу.

Пусть произошла ошибка кратности $k=3$: значения ячеек по адресам $a_{e_1}, a_{e_2}, a_{e_3}$ изменились. Тогда:

$$S = SW \oplus SL = a_{e_1} \oplus a_{e_2} \oplus a_{e_3},$$

$a_{e_1} \oplus a_{e_2} \oplus a_{e_3} = 0$, если $a_{e_1} \oplus a_{e_2} = a_{e_3}$, т.е. для обнаружения трех ошибок необходимо преобразовать исходное векторное пространство таким образом, чтобы сумма любых двух векторов из данного пространства не равнялась третьему вектору, принадлежащему этому пространству.

Теорема. Пусть $a \in A$ и $a = \{\alpha_0, \dots, \alpha_x, \dots, \alpha_{N-1}\}$, $b \in A$ и $b = \{\beta_0, \dots, \beta_x, \dots, \beta_{N-1}\}$, $a \neq b \neq 0$. $a \oplus b < a$ тогда и только тогда, когда $\exists x: \alpha_x = 1$ и $\{\beta_0, \dots, \beta_{x-1}, \beta_x\} = \{0, \dots, 0, 1\}$.

Доказательство. Необходимость. $a \oplus b < a$, или

$$\{\alpha_0, \dots, \alpha_x, \dots, \alpha_{N-1}\} \oplus \{\beta_0, \dots, \beta_x, \dots, \beta_{N-1}\} < \{\alpha_0, \dots, \alpha_x, \dots, \alpha_{N-1}\},$$

$$\{\alpha_0 \oplus \beta_0, \dots, \alpha_x \oplus \beta_x, \dots, \alpha_{N-1} \oplus \beta_{N-1}\} < \{\alpha_0, \dots, \alpha_x, \dots, \alpha_{N-1}\}.$$

Следовательно, существуют позиции в a , где $\alpha_x = 1$, и в $a \oplus b$, где $\alpha_x \oplus \beta_x = 0$.

Достаточность. $a = \{\alpha_0, \dots, 1, \dots, \alpha_{N-1}\}$, $b = \{0, \dots, 1, \dots, \beta_{N-1}\}$. Тогда

$$a \oplus b = \{\alpha_0 \oplus 0, \dots, 1 \oplus 1, \dots, \alpha_{N-1} \oplus \beta_{N-1}\} = \{\alpha_0, \dots, 0, \dots, \alpha_{N-1} \oplus \beta_{N-1}\}.$$

$$\{\alpha_0, \dots, 0, \dots, \alpha_{N-1} \oplus \beta_{N-1}\} < \{\alpha_0, \dots, 1, \dots, \alpha_{N-1}\}. \text{ Следовательно } a \oplus b < a.$$

Для обнаружения АСА ошибок кратности $k=3$ закодируем адреса ОЗУ следующим образом: увеличим размерность вектора на одну координату:

$$a_1 = \{1, 0, \dots, 0, 1\} \text{ или } \{0, \dots, 0, 1, 0, \dots, 0, 1\},$$

$$a_2 = \{1, 0, \dots, 1, 0\} \text{ или } \{0, \dots, 0, 1, 0, \dots, 1, 0\},$$

$$a_i = \{1, \alpha_0, \dots, \alpha_{N-2}, \alpha_{N-1}\} \text{ или } \{0, \dots, 0, 1, \alpha_0, \dots, \alpha_{N-2}, \alpha_{N-1}\}.$$

Покажем, что любые ошибки кратности $k=3$ обнаруживаются. Возьмем два произвольных вектора a_i и a_j таких, что $a_i = \{1, \alpha_0, \dots, \alpha_{N-1}\}$ и $a_j = \{1, \beta_0, \dots, \beta_{N-1}\}$, тогда из теоремы следует, что $a_i \oplus a_j < a_1$. Поэтому ошибки третьей кратности обнаруживаются.

Пусть произошла ошибка кратности $k > 3$ и k – нечетно: значение ячейки по адресам $a_{e1}, a_{e2}, \dots, a_{ek}$ изменились.

Базис. Пусть $k=3$. Тогда из теоремы следует, что все ошибки обнаруживаются.

Индукция. Пусть верно, что $\bigoplus_k a_k \neq 0$, где $k \geq 3$, k — нечетно. Покажем, что верно $\bigoplus_k a_k \oplus a_{k+1} \oplus a_{k+2} \neq 0$. Пусть $s = \bigoplus_k a_k$, тогда из теоремы следует, что $s \oplus a_{k+1} \oplus a_{k+2} \neq 0$.

Пусть произошла ошибка кратности $k > 3$ и k — четно: значение ячейки по адресам $a_{e1}, a_{e2}, a_{e3}, a_{e4}$ изменились. Тогда:

$$S = SW \oplus SL = a_{e1} \oplus a_{e2} \oplus a_{e3} \oplus a_{e4},$$

$a_{e1} \oplus a_{e2} \oplus a_{e3} \oplus a_{e4} = 0$, если $a_{e1} \oplus a_{e2} \oplus a_{e3} = a_{e4}$, т.е. результат суммирования любых трех адресов не должен равняться четвертому адресу.

Определение. Вектор g является образующим вектором векторного пространства G размерности r ($r \geq 0$), если

$$g = \{\gamma_0, \dots, \gamma_{r-2}, \gamma_{r-1}\} \text{ и } \{\gamma_0, \dots, \gamma_{r-2}\} = \{0, \dots, 0\}, \gamma_{r-2} = 1.$$

Например: $r = 3$, $G = (001, 010, 011, 100, 101, 110, 111)$, $g = 100$.

Утверждение 2. $\bigoplus_r g_r \neq 0, \forall r$

В векторном пространстве X размерности N можно выбрать N образующих векторов. Из утверждения 2 следует, что сумма любого количества этих векторов не дает образующий вектор. Если кодировать векторы из A образующими векторами, то можно обнаруживать ошибки любой кратности. Однако для того чтобы закодировать $2N$ векторов, длина кодовых векторов должна быть $2N$. Если ограничиться обнаружением ошибок четвертой кратности, то длину кода можно уменьшить.

Будем последовательно выбирать пары векторов из векторного пространства X . Если пара векторов — образующие векторы, то данные векторы являются и кодовыми. Пусть сумма образующих векторов тоже является кодовым вектором, тогда все векторы, которые дают в результате суммирования этот вектор или уже выбранный, не входят в код. Следующий не образующий вектор, не дающий в сумме с уже выбранным вектором кодовый вектор, также является кодовым. Приведем алгоритм построения кода для обнаружения четырехкратных ошибок.

Алгоритм построения кода:

X — исходное множество векторов, $n = |X|$
 G — множество образующих векторов, $G \subset X$, $g_k \in G$, $g_k = 2^k$
 C — множество кодовых векторов, $C \subset X$, $c_i \in C$, $m = |C|$
 $k = [0 \dots n]$, $i = [0 \dots m]$

$g_0 := 1$

while $g_k < n$ **do begin**

$v1 :=$ вектор из X , для которого $(v1 \oplus c_i) \notin C$

добавить $v1$ в C

удалить $v1$, $(v1 \oplus c_i)$ из X

$v2 :=$ следующий вектор из X , для которого $(v1 \oplus v2) \in X$

```

if  $v_1 = g_k$  and  $v_2 = g_{k+1}$  then begin
    добавить  $v_2$  в  $C$ 
    удалить  $v_2, (v_2 \oplus c_i)$  из  $X$ 

    добавить  $(v_1 \oplus v_2)$  в  $C$ 
    удалить  $(v_1 \oplus v_2), (v_1 \oplus v_2 \oplus c_i)$  из  $X$ 

     $g_k := g_{k+1}$ 
end
else if  $v_1 = g_k$  and  $v_2 < g_{k+1}$  then begin
    while  $v_2 < g_{i+1}$  do begin
        добавить  $v_2$  в  $C$ 
        удалить  $v_2, (v_2 \oplus c_i)$  из  $X$ 

         $v_1 := v_2$ 
         $v_2 :=$  следующий вектор из  $X$ , для которого  $(v_1 \oplus c_i) \in X$ 
    end
end

     $g_k := g_{k+1}$ 
end

```

В таблице приведены векторы, выбранные предложенным алгоритмом из 1024 последовательных векторов. Как видно из таблицы, из 1024 последовательных векторов можно выбрать только 30 кодовых векторов. Сумма любой четверки векторов из полученного набора не дает нулевой вектор.

Векторы для кодирования 30 адресов ОЗУ, позволяющие обнаруживать четырехкратные ошибки в ОЗУ с помощью метода АСА

1_{10}	1_2	216_{10}	11011000_2
2_{10}	10_2	237_{10}	11101101_2
3_{10}	11_2	247_{10}	11110111_2
4_{10}	100_2	256_{10}	10000000_2
8_{10}	1000_2	279_{10}	10001011_2
12_{10}	1100_2	297_{10}	10010100_2
16_{10}	10000_2	457_{10}	11100100_2
32_{10}	100000_2	512_{10}	100000000_2
48_{10}	110000_2	537_{10}	100001100_2
64_{10}	1000000_2	558_{10}	100010110_2
85_{10}	1010101_2	594_{10}	1001010010_2
106_{10}	1101010_2	796_{10}	1100011100_2
128_{10}	10000000_2	847_{10}	1101001111_2
150_{10}	10010110_2	872_{10}	1101101000_2
171_{10}	10101011_2	941_{10}	1110101101_2

Заключение

В данной статье были рассмотрены способы повышения достоверности текущего контроля ОЗУ с использованием метода АСА. Добавление одного дополнительного бита к адресам ОЗУ позволяет методу АСА обнаруживать все неисправности нечетной кратности. Для обнаружения неисправностей четной кратности (выше второй) в соответствие каждому адресу ОЗУ должен ставиться кодовый вектор. Был предложен алгоритм, который позволяет получить кодовые векторы для обнаружения ошибок четвертой кратности. Необходимо дальнейшее исследование свойств полученных кодовых векторов с целью эффективной аппаратурной реализации.

IMPROVING THE REABILITY OF CONCURRENT CHECKING TECHNIQUES FOR RAM

A.A. IVANIUK, S.B. MUSIN

Abstract

In this paper we consider error detection and correction ability of effective concurrent checking techniques for RAM – self-adjusting output data compression (SAODC). We investigate the ways of improving this technique, as well as propose the new method for quadruple RAM errors detecting by the SAODC.

Литература

1. Ярмолик В.Н., Калоша Е.П., Быков Ю.В., и др. Проектирование самотестируемых СБИС. Т. 2. Мн., 2001.
2. Sudhakar Govindavajhala, Andrew W. Applet Using Memory Errors to Attack a Virtual Machine // IEEE Symposium on Security and Privacy, 2003.
3. Ярмолик В.Н., Иванюк А.А. Встроенное самотестирование памяти с использованием сигнатурного анализа // Логическое проектирование. Мн., ИТК, 1997.
4. Nicolaidis M. Transparent BIST for RAMs // Proc. of Int. Test Conf., USA, 1992.
5. Ярмолик В.Н., Иванюк А.А. Адаптивный сигнатурный анализ для бит-ориентированной памяти // Материалы 25-й юбилейной Междунар. конф. и дискуссионный клуб IT+SE'98: Сб. тр. конф. Гурзуф, 1998.
6. Иванюк А.А. Методы повышения достоверности адаптивного сигнатурного анализа // М-лы 10-й науч.-техн. конф. "Новые технологии в машиностроении и вычислительной технике": Сб. тр. конф. Ч. 2. Брест, 1998.