ИНФОРМАТИКА

2008 **январь-март** **№ 1**

УДК 621.382

А.А. Иванюк, А.В. Степанов

ОПРЕДЕЛЕНИЕ ТЕСТОВЫХ НАБОРОВ ДЛЯ ОБНАРУЖЕНИЯ НЕИСПРАВНОСТЕЙ АДРЕСНЫХ ЛИНИЙ ОЗУ

Предлагается подход к определению тестовых наборов для обнаружения неисправностей адресных линий оперативных запоминающих устройств (ОЗУ). Приведенные в статье теоретические и экспериментальные результаты показывают, что данный подход позволяет существенно сократить количество тестовых наборов, необходимых для обнаружения всех неисправностей адресных линий ОЗУ, по сравнению с существующими маршевыми алгоритмами.

Введение

В настоящее время такая область электроники, как встраиваемые системы и системы на кристалле (СнК) развивается стремительными темпами. Связано это в первую очередь со значительным прогрессом в области технологий и быстрым удешевлением элементной базы. Такой быстрый прогресс позволяет значительно расширить круг применения встраиваемых систем. Действительно, если раньше в какой-либо из областей использование встраиваемых систем было экономически неоправданным, то в настоящее время встраиваемые решения приобретают все большую привлекательность [1].

Одним из основных компонентов для построения встраиваемых систем и СнК являются ОЗУ. В среднем доля памяти может составлять около 90 % аппаратных средств системы. В последнее время отрасль производства ОЗУ отличается высокими темпами развития, которые в основном обусловлены сокращением размера и высокой степенью интеграции компонентов, используемых при их производстве, однако уменьшение размеров элементов и их плотная упаковка приводят к проблемам технологического характера: усложнению структуры модулей, а сложность, в свою очередь, – к увеличению вероятности появления сбоев в работе таких устройств. Анализ статистических данных показывает, что ОЗУ относятся к наименее надежным устройствам [2]. Результаты исследований [3] свидетельствуют о том, что отказы ОЗУ составляют до 70 % от общего числа отказов системы. Вследствие этого очень важно обеспечить правильность функционирования ОЗУ.

В данной статье предлагается методика определения минимально необходимого числа тестовых наборов для обнаружения неисправностей адресных линий ОЗУ.

1. Постановка задачи

В последнее время для тестирования массивов запоминающих элементов все чаще стали использовать встроенную аппаратуру самотестирования [4]. Для этого на кристалле располагается дополнительная аппаратура, обеспечивающая генерацию тестовых наборов (ГТН) и обработку результатов тестирования. Данный подход позволяет провести функциональное тестирование кристалла для обнаружения любых типов неисправностей, а также диагностику для выявления места возникновения неисправности, однако при установке исправно функционирующей компоненты памяти на печатной плате (ПП) возможны случаи обрывов и замыканий между линиями межсоединений, среди которых могут быть адресные линии.

В настоящее время для тестирования электронных компонентов, установленных на ПП, применяются два основных подхода:

1. Использование ведущего устройства на ПП. Данный подход основывается на генерировании тестовых наборов ведущим устройством и анализе реакций, формируемых ведомыми устройствами на промежуточных и выходных полюсах. В качестве ведущего устройства может выступать микроконтроллер, микропроцессор или специальное устройство тестирования.

2. Использование технологии граничного сканирования. Основным элементом архитектуры граничного сканирования является специальный дополнительный триггер на каждом вхо-

де и выходе интегральной схемы (ИС), который получил название ячейки граничного сканирования. Все ячейки граничного сканирования объединены в один длинный сдвиговый регистр. Объединяя выход одной ИС с входом другой, можно создать одну длинную цепь сканирования, охватывающую все ИС на печатной плате. Таким образом, можно проверить правильность функционирования установленных компонентов.

Особенно сложной и трудоемкой является задача тестирования межсоединительных линий, которые подключены к контактам ИС ОЗУ. Применение описанных выше подходов для данного случая имеет ряд недостатков:

1. Использование технологии граничного сканирования требует значительных временных затрат. Кроме того, очень часто ИС ОЗУ принадлежит группе схем, не вовлеченных в общую цепь сканирования, что существенно усложняет задачу тестирования.

2. По сравнению с встроенным самотестированием, использование ведущего устройства на ПП не позволяет производить тестирование на внутренней частоте ИС ОЗУ, поэтому временные затраты возрастают даже при применении простейших алгоритмов.

Следовательно, возникает необходимость в создании новых функциональных тестов, которые позволят обнаруживать неисправности адресных линий ОЗУ с минимальными временными затратами. Для этого решим задачу определения минимального числа тестовых наборов, необходимых для обнаружения всех неисправностей адресных линий ОЗУ.

2. Теоретический анализ

Определение факта возникновения дефекта и его классификация представляется весьма трудоемкой задачей, зачастую неразрешимой. Это, прежде всего, объясняется тем, что возникновение дефекта чаще всего можно определить лишь по косвенным признакам, таким, как неправильное функционирование ОЗУ [5]. Классически функциональные неисправности ОЗУ принято разделять на три типа по числу основных блоков функциональной структуры ОЗУ: неисправности дешифраторов адреса, неисправности массива запоминающих элементов и неисправности логики чтения/записи [4]. В настоящей статье рассматриваются вопросы, связанные с обнаружением неисправностей дешифраторов адреса (рис. 1).

Неисправности дешифраторов адреса (AF) можно разделить на четыре основных типа [6, 7]:

AF1 – ни одна из всех ячеек памяти недоступна по заданному адресу;

AF2 – заданная ячейка памяти недоступна;

AF3 – по заданному адресу может осуществляться доступ к нескольким ячейкам памяти;

AF4 – доступ к заданной ячейке памяти может быть осуществлен по нескольким адресам.

Данные неисправности могут проявляться в виде обрыва линии шины адреса или наличия короткого замыкания между линиями.

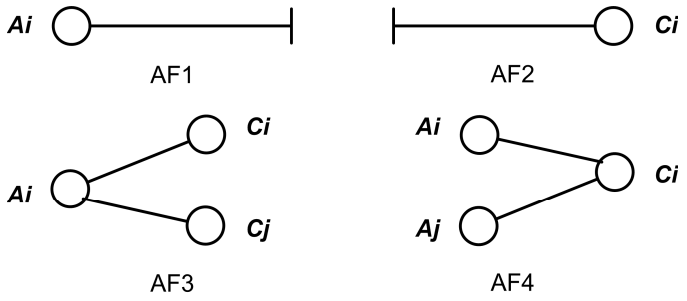

Рис. 1. Основные типы неисправностей дешифраторов адреса:
A – адресное пространство ОЗУ; *C* – массив ячеек памяти

Рассмотрим, каким образом адресные неисправности ОЗУ можно представить при помощи моделей неисправностей сигнальных линий. Одной из универсальных моделей, описывающей дефекты сигнальных линий, является модель мостиковой неисправности. С помощью этой модели можно описать такие неисправности, как обрыв линии, неисправности константного нуля и константной единицы и т. д.

Мостиковой неисправностью принято называть неисправность, которая предполагает наличие проводящего пути между двумя полюсами схемы (рис. 2) [8].

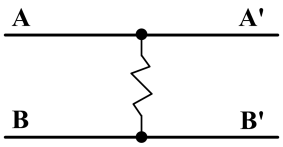

Рис. 2. Мостиковая неисправность

На рис. 3 показаны графические нотации основных моделей мостиковых неисправностей [9].

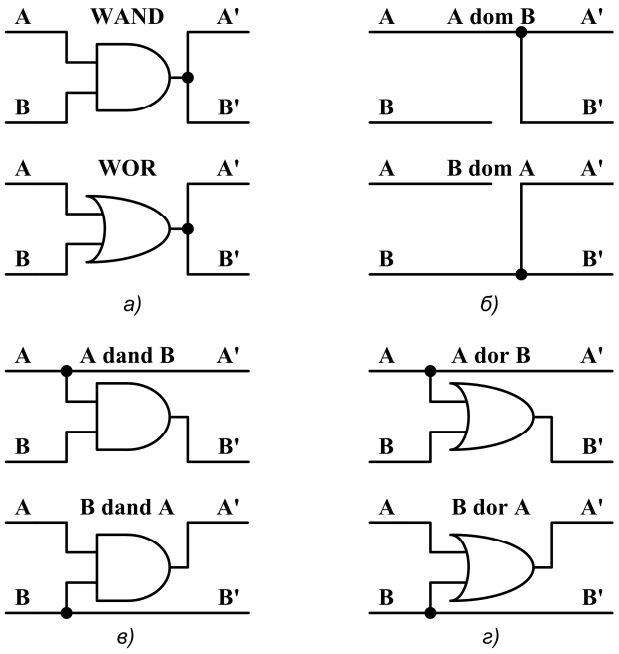

Рис. 3. Модели мостиковых неисправностей: *а)* типа «монтажное-И» и «монтажное-ИЛИ»; *б)* доминантной неисправности; *в)* типа «доминантное-И»; *г)* типа «доминантное-ИЛИ»

Рассмотрим подробнее представленные модели:

1) модель мостиковой неисправности типа «монтажное-И» иногда упоминается как 0-доминантная мостиковая неисправность (WAND) (рис. 3, *а*);

2) модель мостиковой неисправности типа «монтажное-ИЛИ» иногда упоминается как 1-доминантная мостиковая неисправность (WOR) (рис. 3, *а*);

3) модель доминантной мостиковой неисправности («A dom B» или «B dom A») (рис. 3, *б*);

4) модель мостиковой неисправности типа «доминантное-И» («A dand B» или «B dand A») (рис. 3, *в*);

5) модель мостиковой неисправности типа «доминантное-ИЛИ» («A dor B» или «B dor A») (рис. 3, *г*).

Рассмотрим неисправности адресных линий ИС ОЗУ для случая, когда имеет место доминантная мостиковая неисправность (рис. 4). ИС ОЗУ, установленная на печатной плате, имеет встроенную аппаратуру самотестирования, которая позволяет обнаружить наличие неисправностей ОЗУ. Предположим, что при установке ИС ОЗУ на печатную плату произошло замыкание адресных линий, в результате образовалась доминантная мостиковая неисправность. Поэтому относительно других компонент печатной платы исправная ИС ОЗУ функционирует некорректно, однако встроенная аппаратура самотестирования не позволяет обнаружить неисправности такого типа.

В соответствии с рассмотренной выше классификацией основных типов неисправностей дешифраторов адреса наличие доминантной мостиковой неисправности образует тип адресных неисправностей AF2 и AF4. Так, например, в данном случае ячейка памяти C_2 недоступна, а доступ к ячейке C_3 может быть осуществлен по нескольким адресам (001 и 011).

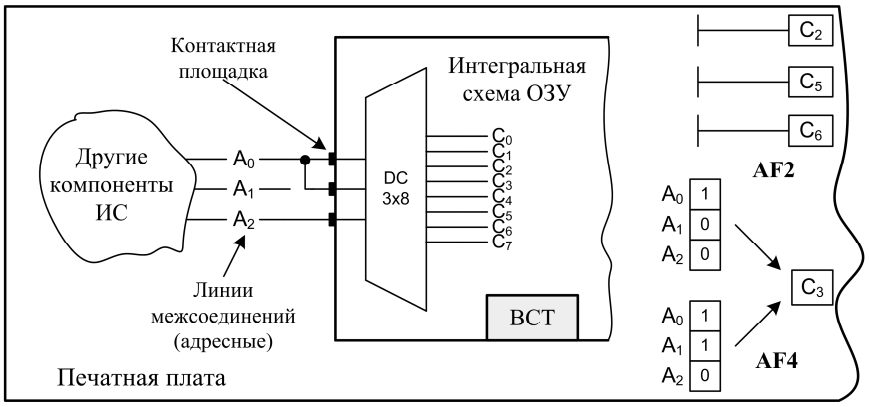

Рис. 4. Пример неисправности адресных линий ИС ОЗУ

Поведение адресных линий в случае наличия одной из моделей мостиковых неисправностей представлено в табл. 1 [8]. Различия при правильном и некорректном поведении отображены серым цветом. Так, например, мостиковая неисправность «A dand B» проявит себя только при подаче тестового набора {0, 1} в виде искажения логического значения на линии *B'*.

Таблица 1

Поведение адресных линий в случае наличия мостиковых неисправностей

Корректное поведение		WAND		WOR		«A dom B»		«B dom A»		«A dand B»		«A dor B»	
A	*B*	*A′*	*B′*	*A′*	*B′*	*A′*	*B′*	*A′*	*B′*	*A′*	*B′*	*A′*	*B′*
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	**0**	**1**	1	0	**0**	**1**	1	0	**0**	0	1
1	0	**0**	0	1	**1**	1	**1**	**0**	0	1	0	1	**1**
1	1	1	1	1	1	1	1	1	1	1	1	1	1

Утверждение. *Для обеспечения условия проявления любой из указанных мостиковых неисправностей (см. рис. 3) необходимо и достаточно установить на замкнутых полюсах (A и B) значения парафазных наборов {0, 1} и {1, 0}.*

Доказательство. Рассмотрим адресную шину, состоящую из двух адресных линий (рис. 5).

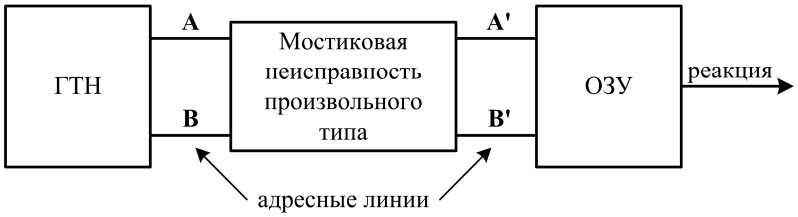

Рис. 5. Адресная шина при наличии неисправности произвольного типа

Опишем мостиковую неисправность в виде нотации $F(A, B; A', B')$, которая имеет два входных (A, B) и два выходных (A', B') значения. Рассмотрим использование данной нотации для случаев наличия различных типов мостиковых неисправностей на линиях A и B:

1) мостиковые неисправности отсутствуют. Для данного случая нотация примет следующий вид: $F(A, B; A, B)$;

2) мостиковая неисправность типа «монтажное-И». $Fwand = F(A, B; A \wedge B, A \wedge B)$;

3) мостиковая неисправность типа «монтажное-ИЛИ». $Fwor = F(A, B; A \vee B, A \vee B)$;

4) доминантная мостиковая неисправность:

а) «A dom B». $Fdom = F(A, B; A, A)$;

б) «B dom A». $Fdom = F(A, B; B, B)$;

5) мостиковая неисправность типа «доминантное-И»:

а) «A dand B». $Fdand = F(A, B; A, A \wedge B)$;

б) «B dand A». $Fdand = F(A, B; A \wedge B, B)$;

6) мостиковая неисправность типа «доминантное-ИЛИ»:

а) «A dor B». $Fdor = F(A, B; A, A \vee B)$;

б) «B dor A». $Fdor = F(A, B; A \vee B, B)$.

Данная нотация может быть использована для описания не только мостиковых неисправностей, но и константных неисправностей и неисправностей типа «обрыв».

Константные неисправности:

а) $F_0 = F(A, B; A, 0)$ – неисправность типа «константный нуль» на линии B;

б) $F_1 = F(A, B; 1, B)$ – неисправность типа «константная единица» на линии A.

Обрыв адресной линии может быть описан следующим образом:

$Fz = F(A, B; Z, B)$ – обрыв линии A, где Z – высокоимпедансное значение выходной линии.

Рассмотрим мостиковую неисправность типа «доминантное-И». Для двух сигнальных линий A и B можно выделить две конфигурации неисправностей: $F(A, B; A, A \wedge B)$ и $F(A, B; A \wedge B, B)$, которые при использовании парафазных входных значений преобразуются к видам $F(A, \neg A; A, 0)$ и $F(A, \neg A; 0, \neg A)$ соответственно. Первая конфигурация $F(0, 1; 0, 0)$ проявляется при тестовом наборе $\{0, 1\}$, который, однако, не проявляет вторую конфигурацию $F(0, 1; 0, 1)$. Использование инверсного тестового набора $\{1, 0\}$ позволяет проявиться второй конфигурации $F(1, 0; 0, 0)$, но не проявляет первую $F(1, 0; 1, 0)$. Таким образом, чтобы обнаружить наличие мостиковой неисправности для рассматриваемого случая, необходимо и достаточно подать на вход тестовые наборы $\{0, 1\}$ и $\{1, 0\}$. Аналогичными рассуждениями можно доказать необходимость и достаточность этих наборов для проявления остальных типов мостиковых неисправностей.

Следовательно, для общего случая, когда адресная шина имеет разрядность N, необходимо и достаточно, чтобы для любой из пары линий были применены тестовые наборы $\{0, 1\}$ и $\{1, 0\}$. Например, таким свойством обладает исчерпывающая последовательность, которая содержит 2^N различных тестовых наборов и применяется в классических маршевых алгоритмах. Целью же данного исследования является нахождение минимального количества тестовых наборов, удовлетворяющих приведенному выше утверждению.

Рассмотрим получение тестовых наборов при использовании кольцевого счетчика. Для случая, когда разрядность адресной линии равна восьми, а начальное состояние счетчика «10000000», имеем следующую матрицу E:

$$E = \begin{bmatrix} 10000000 \\ 01000000 \\ \dots \\ 00000010 \\ 00000001 \end{bmatrix}.$$

Данная матрица является единичной. Строки представляют собой наборы, которые генерирует алгоритм «Бегущая диагональ». Как показывает исследование [10], использование дан-

ного метода позволяет не только обнаружить адресные неисправности, но и произвести их диагностику. Для случая, когда необходимо выполнить только проверку на наличие неисправностей, данный набор тестовых векторов является избыточным.

Как было показано в [11], для формирования тестовых векторов, позволяющих обнаружить описанные типы мостиковых неисправностей, можно использовать матрицу $T^* = \left[\frac{T}{T'}\right]$, где матрица T может быть построена наподобие проверочной матрицы кода Хэмминга, а матрица T' получена путем поэлементной инверсии матрицы T.

Код Хэмминга позволяет получить тестовый набор, состоящий из $[\log_2 N]$ векторов, в котором значения всех столбцов уникальны, а остальные $[\log_2 N]$ вектора получаются путем инверсии (рис. 6).

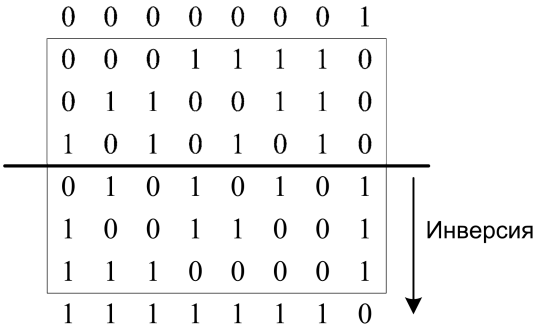

Рис. 6. Код Хэмминга ($N = 8$)

Рассмотрим возможность формирования матрицы тестовых наборов, эквивалентных строкам матрицы T^*, с использованием кольцевого счетчика. Для того чтобы получить минимальный набор тестовых векторов для разрядности адресной шины N, необходимо иметь такое начальное состояние кольцевого счетчика, при котором путем циклического сдвига $[\log_2 N]$ раз получается матрица размерности $N \times [\log_2 N]$, в которой значения всех столбцов различны, а остальные $[\log_2 N]$ вектора получаются в результате инверсии строк полученной матрицы.

Так как любой столбец матрицы, полученный путем циклического сдвига, состоит из $[\log_2 N]$ последовательных элементов начального состояния счетчика, а значения всех столбцов должны быть различны, можно ввести ограничение на значение начального состояния счетчика: пусть начальное состояние представляет собой последовательность, состоящую из N элементов, тогда любое подмножество, состоящее из $[\log_2 N]$ последовательных элементов искомой последовательности, должно быть уникальным.

Приведем алгоритм получения значения начального состояния для случая использования кольцевого счетчика.

Алгоритм

n – разрядность адресной шины;

m – размерность уникального подмножества, $m = [\log_2 n]$;

P – искомая последовательность, $P = (p_0, p_1, \ldots, p_{n+m})$, где $p_i \in \{0, 1\}$, $i = \overline{0, n+m}$;

t – индекс текущего элемента последовательности P, $t = \overline{0, n+m}$;

Gj – подмножество искомой последовательности P, $Gj = (p_{t-m+1}, p_{t-m+2}, \ldots, p_{t-1}, j)$, где $p_i \in P, j \in \{0, 1\}, i = \overline{t-m+1, t-1}$.

1. Первым m элементам последовательности P присваивается значение ‘1’ ($p_i = 1$, $\forall i = \overline{0, m-1}$, $t = m$).

2. Так как все подмножества размерности m должны быть уникальны, следовательно, следующему элементу последовательности P присваивается значение ‘0’ ($p_m = 0$, $t = t + 1$).

3. Рекурсивный поиск последующих значений элементов последовательности P:

а) если подмножество $G1$ уже существует ($G1 \subset P$) или произошел возврат из рекурсии, то переход к п. б), иначе элементу присваивается значение ‘1’ ($p_t = 1$, $t = t + 1$);

б) если подмножество $G0$ уже существует ($G0 \subset P$), то возврат к предыдущему элементу ($t = t - 1$), иначе элементу присваивается значение ‘0’ ($p_t = 0$, $t = t + 1$).

Данный шаг повторяется до тех пор, пока не будут получены значения всех элементов последовательности P.

4. Первые n элементов искомой последовательности P и являются значениями начального состояния кольцевого счетчика.

Данный алгоритм позволяет получить начальное значение кольцевого счетчика для произвольной разрядности адресной шины. В табл. 2 приведены значения начального состояния кольцевого счетчика, полученные предложенным алгоритмом.

Таблица 2

Значения начального состояния кольцевого счетчика

Разрядность адресной шины	Начальное значение счетчика, представленное в двоичном виде
2	10
4	1100
8	11101000
16	1111011001010000
20	11111011100110101000

Пример. Генерирование тестовых наборов ($N = 8$). Общая последовательность действий процесса генерирования тестовых наборов при использовании кольцевого счетчика представлена на рис. 7.

	n-1	n-2		...		1	0	
$\mathbf{S_0}$	1	1	1	0	1	0	0	0 - начальное состояние (INV=0)
$\overline{\mathbf{S_0}}$	0	0	0	1	0	1	1	1 - инвертирование (INV=1)
$\mathbf{S_1}$	0	1	1	1	0	1	0	0 - циклический сдвиг вправо (INV=0)
$\overline{\mathbf{S_1}}$	1	0	0	0	1	0	1	1 - инвертирование (INV=1)
$\mathbf{S_2}$	0	0	1	1	1	0	1	0 - циклический сдвиг вправо (INV=0)
$\overline{\mathbf{S_2}}$	1	1	0	0	0	1	0	1 - инвертирование (INV=1)

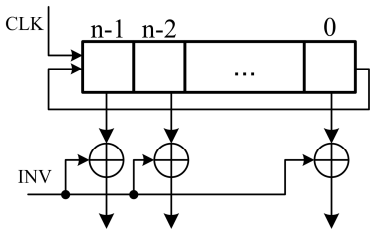

Рис. 7. Генерирование тестовых наборов при помощи кольцевого счетчика

Таким образом, применение кольцевого счетчика при соответствующих начальных значениях позволяет минимизировать число тестовых наборов, необходимых для обнаружения различных типов адресных неисправностей ОЗУ, с N до $2*[\log_2 N]$ (см. рис. 6).

3. Экспериментальная часть

В предыдущем разделе аналитически было получено минимальное число тестовых наборов, необходимых для обнаружения адресных неисправностей ОЗУ. Теперь определим число тестовых наборов экспериментальным способом и произведем сравнение с аналитическими данными.

Для эксперимента использовались следующие исходные данные:

– программная модель бит-ориентированного ОЗУ;

– разрядность адресной шины от 2 до 20 бит;

– методы генерирования тестовых наборов: кольцевой счетчик, линейный сдвиговый регистр с обратной связью или Linear Feedback Shift Register(LFSR), счетчик Джонсона, код Хэмминга и «жадный» алгоритм.

Выбор методов осуществлялся на основании следующих критериев: простота реализации алгоритма и минимальные временные и аппаратурные затраты при реализации алгоритма.

Исключение составляет «жадный» алгоритм, который выбран только исходя из того, что теоретически он является наиболее предпочтительным для определения минимально необходимого числа тестовых наборов.

Под «жадным» алгоритмом будем понимать алгоритм, включающий следующую последовательность шагов:

1. Поиск тестовых наборов, которые имеют наибольшую покрывающую способность для первоначального набора неисправностей.

2. Рекурсивный поиск последующих тестовых наборов, которые имеют наибольшую покрывающую способность с учетом текущей конфигурации набора неисправностей, т. е. расчет покрывающей способности набора выполняется исходя из впервые обнаруженных неисправностей. Данный шаг повторяется до тех пор, пока не будут обнаружены все неисправности.

3. Сравнение полученного количества тестовых наборов с текущим минимальным набором и при необходимости выполнение корректировки.

4. Шаги 2 и 3 повторяются для всех наборов, полученных на шаге 1.

Механизм определения минимального тестового набора для случая использования вышеперечисленных методов имеет следующий вид:

1) на вход генератора подается значение из диапазона от 1 до $2^N - 1$, где N – разрядность адресной шины;

2) генерируется тестовый набор;

3) переход на шаг 2, если есть необнаруженные неисправности;

4) полученное количество тестовых наборов сравнивается с текущим минимальным набором и при необходимости производится корректировка.

Результаты проведения эксперимента приведены в табл. 3.

Таблица 3

Минимальное количество тестовых наборов

Метод	Разрядность адресной шины				
	2	4	8	16	20
Кольцевой счетчик	2	4	7	8	8
LFSR	2	4	5	7	8
Счетчик Джонсона	–	–	6	7	8
Код Хэмминга	2	4	6	8	10
«Жадный» алгоритм	2	4	6	7	9

На рис. 8 представлена сравнительная оценка тройки лучших ГТН.

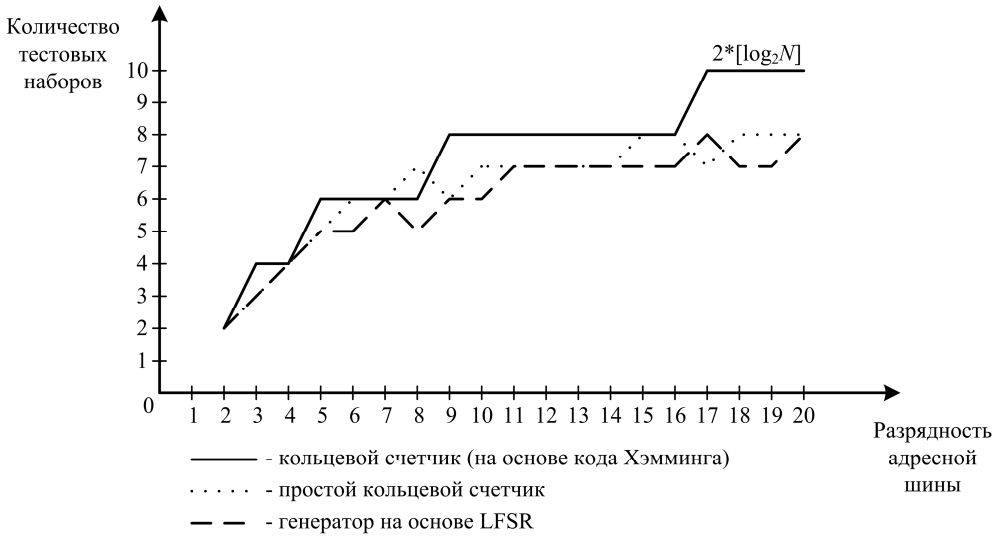

Рис. 8. Сравнительная оценка ГТН

Проведенный эксперимент показал, что наилучшим методом генерации тестовых наборов является метод LFSR. Экспериментально было доказано, что среди всего многообразия полиномов n-й степени минимальным числом тестовых наборов обладает полином вида $\varphi(x) = x^n \oplus 1$ с определенным начальным состоянием. В табл. 4 приведены значения начального состояния для случая использования генератора на основе LFSR.

Таблица 4

Соответствие начального значения LFSR разрядности адресной шины

Разрядность адресной шины, N	Начальное значение, представленное в десятичном виде
2	1 или 2
4	2 или 5
8	230
16	9814
20	78422

Пример. Генерирование тестовых наборов ($N = 8$). В качестве порождающего полинома возьмем $\varphi(x) = x^8 \oplus 1$, а начальное состояние генератора «01100111». Полученные тестовые наборы можно представить в виде строк матрицы T:

$$T = \begin{bmatrix} 01100111 \\ 10110011 \\ 01011001 \\ 10101100 \\ 11010110 \end{bmatrix}.$$

Рассмотрим, каким образом данные тестовые наборы могут быть применены для обнаружения адресных неисправностей.

Алгоритм

1. Производится операция записи ‘0’ во все ячейки памяти.

2. На адресную шину ИС подаются полученные тестовые наборы. Производится изменение содержимого ОЗУ путем записи ‘1’ в ячейки памяти по заданным тестовым адресам.

3. Считывается содержимое ячеек памяти, и полученные значения сравниваются с ‘0’. В том случае, если содержимое ячейки памяти не равно ‘0’, тест останавливается и считается, что ИС содержит адресные неисправности. Данный шаг выполняется только для тех ячеек памяти, адреса которых не принадлежат тестовому набору.

Данная последовательность действий легко может быть реализована программным или аппаратным способом. Генератор тестовых наборов может быть применен как в случае использования специального устройства тестирования, так и в случае использования технологии граничного сканирования.

Сложность приведенного алгоритма в случае использования одного из трех предложенных ГТН (простой кольцевой счетчик, кольцевой счетчик на основе кода Хэмминга, генератор на основе LFSR) оценивается как

$$L \le N + 2*(2*[\log_2[\log_2 N]]) + (N - 2*[\log_2[\log_2 N]]) = 2N + 2*[\log_2[\log_2 N]],$$

где N – информационная емкость ОЗУ в битах.

В то же время сложность минимального классического разрушающего маршевого теста ($\Updownarrow (w0)$; $\Updownarrow (r0, w1)$), который обнаруживает неисправности адресных линий, равна $3N$ [6].

Пример. Время доступа равно 100 нс, где данная величина представляет собой усредненное время, необходимое процессорному блоку для осуществления операций чтения и записи с блоком ОЗУ. На рис. 9 представлены результаты сравнения времени тестирования ОЗУ в слу-

чае использования классического теста и алгоритма, предложенного в данной статье. Штриховой линией показано время тестирования для случая, когда в качестве ГТН для предложенного алгоритма используется метод LFSR.

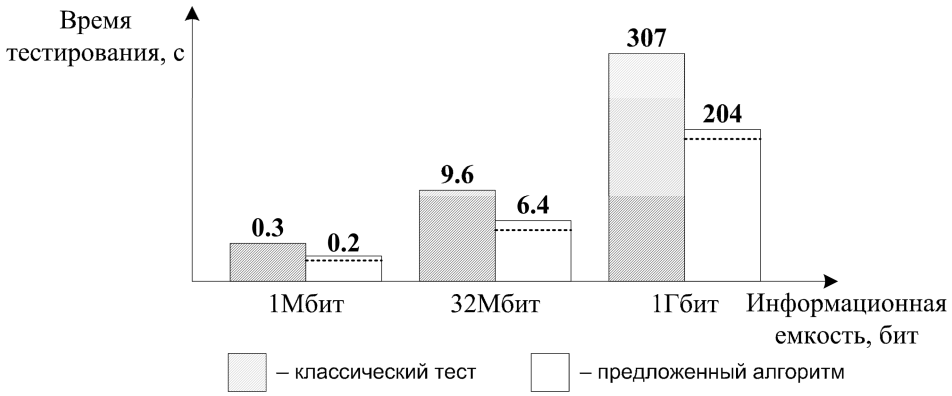

Рис. 9. Сравнительный анализ тестов

Как видно из приведенной диаграммы, применение нового алгоритма позволяет ускорить процесс тестирования ОЗУ на предмет наличия неисправностей адресных линий приблизительно на 30 %.

Заключение

В данной статье предложена методика определения тестовых наборов для обнаружения неисправностей адресных линий ОЗУ. Проведение экспериментальных исследований показало, что использование генератора на основе LFSR при соответствующих начальных значениях позволяет значительно сократить количество тестовых наборов, необходимых для обнаружения неисправностей адресных линий. Предложен также алгоритм, который позволяет применить полученные тестовые наборы для обнаружения неисправностей адресных линий ОЗУ. Полученные результаты могут быть использованы для решения задачи обнаружения неисправностей линий данных и управления.

Список литературы

1. Сигаев, А. Операционные системы для встраиваемых применений / А. Сигаев. – М.: Компоненты и технологии, 2000. – 168 с.

2. Cockburn, B. Tutorial on Semiconductor Memory Testing / B. Cockburn // JETTA. – 1994. – Vol. 5, № 4. – P. 321–336.

3. Goor, A.G. Memory Tests and Their Fault Coverage into a New Perspective, Resulting into a New Test / A.G. Goor // Proc. of SEMICON/Korea's Semiconductor Technical Symposium on Test Technology. – Seoul, Korea, 1996. – P. 67–75.

4. Проектирование самотестируемых СБИС: научное издание. В 2 т. / В.Н. Ярмолик [и др.] – Минск: БГУИР, 2001. – Т. 2. – 163 с.

5. Неразрушающее тестирование запоминающих устройств / В.Н. Ярмолик [и др.]. – Минск: Бестпринт, 2005. – 230 с.

6. Van de Goor, A.J. Testing Semiconductor Memories: Theory and Practice / A.J. van de Goor. – New York: John Wiley & Sons, 1991. – 512 p.

7. Van de Goor, A.J. Opens and Delay Faults in CMOS RAM Address Decoders / A.J. van de Goor // IEEE Transactions on Computers. – 2006. – Vol. 55, № 11. – P. 1630–1639.

8. Stroud, E.A. Designer's Guide to Built-In Self-Test / E.A. Stroud. – Kluwer Academic Publishers, 2002. – 344 p.

9. Ma, S.A Comparison of Bridging Fault Simulation Methods / S. Ma, I. Shaik, R. Fetherston // Proc. IEEE International Test Conf. – Atlantic City, USA, 1999. – P. 587–595.

10. Garbolino, T. Detection, Localization and Identification of Interconnection Faults Using MISR Compactor / T. Garbolino, M. Kopec, K. Gucwa // Proc. of the 9th IEEE Workshop on Design & Diagnostics of Electronic Circuits & Systems (DDECS 2006). – Prague, Czech Republic, 2006. – P. 230–231.

11. Иванюк, А.А. Определение тестовых наборов для обнаружения мостиковых неисправностей / А.А. Иванюк // Известия Национальной академии наук Беларуси. Сер. физ.-тех. наук. – 2006. – № 5. – С. 38–40.

Поступила 25.06.07

Белорусский государственный университет
информатики и радиоэлектроники,
Минск, ул. П. Бровки, 6
e-mail: ivaniuk@bsuir.unibel.by

A.A. Ivaniuk, A.V. Stepanov

TECHNIQUE OF TEST PATTERNS ESTIMATION FOR ADDRESS LINES FAULTS OF RAM

The paper proposes a method for definition of test vectors for detection of address lines faults of RAM. It is shown that this method allows to reduce essentially the number of the test vectors to detect all address lines faults.