

ПСЕВДОИСЧЕРПЫВАЮЩЕЕ ТЕСТИРОВАНИЕ ОЗУ ВСТРАИВАЕМЫХ СИСТЕМ

Леванцевич В. А.

Кафедра программного обеспечения информационных технологий, Белорусский государственный университет информатики и радиоэлектроники

Минск, Республика Беларусь

E-mail: lvn@bsuir.by

Анализируются методы тестирования современных запоминающих устройств, в том числе оперативных запоминающих устройств (ОЗУ) встраиваемых систем. Обосновывается использование многократных тестов ОЗУ с изменяемым начальным состоянием для псевдоисчерпывающего тестирования ОЗУ. Приводятся оценки минимальной, средней кратности многократного теста для обеспечения исчерпывающего множества комбинаций для заданного числа ячеек ОЗУ

ВВЕДЕНИЕ

В кодочувствительных неисправностях (Pattern Sensitive Faults – PSF) логическое состояние или изменение логического состояния одного запоминающего элемента ОЗУ, зависит от содержимого (0/1) или логических переходов 0/1 или 1/0 влияющих запоминающих элементов [1]. На практике используют модель граничных кодочувствительных неисправностей (Neighborhood Pattern Sensitive Faults – NPSF), для которой количество элементов k участвующих в неисправности, не превышает 10. Так как логическая топология ОЗУ, как правило, не совпадает с физической, то влияющими ячейками, участвующими в кодочувствительной неисправности, могут быть любые ячейки ОЗУ. Для обнаружения таких неисправностей процедура тестирования должна сформировать в любых k из N ячеек ОЗУ все возможные 2^k двоичные комбинации.

1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

Исчерпывающее тестирование характеризуется максимальной эффективностью обнаружения неисправностей и позволяет сформировать в N ячейках ОЗУ все 2^N возможные комбинации. Однако, исчерпывающие тесты имеют слишком большую временную сложность, что ограничивает их применение. Поэтому на практике используют аппроксимацию исчерпывающего тестирования в виде псевдоисчерпывающих тестов [2]. Псевдоисчерпывающий тест формирует в любых k из N ($k < N$) ячейках все возможные 2^k комбинации, т. е. является «локально исчерпывающим» тестом. Управляемые вероятностные тесты, так же являются аппроксимацией исчерпывающих тестов, однако, такие тесты имеют значительную вычислительную сложность, обусловленную необходимостью перебора потенциальных кандидатов в тестовые наборы, и принятия решения о включении или не включении их в вероятностный тест [5]. Для уменьшения сложности формирования управляемых вероятностных тестов и повышения их эффективности приме-

няют многократные управляемые вероятностные тесты [3]. При этом для повышения эффективности многократных тестов используют два подхода: на каждой итерации многократного теста изменяют либо порядок следования адресов, либо начальные состояния ячеек ОЗУ.

II. РЕАЛИЗАЦИЯ

Проанализируем возможность использования многократных тестов с изменяемыми начальными состояниями и фиксированным порядком следования адресов, для псевдоисчерпывающего тестирования ОЗУ. Введем понятие орбиты, которая представляет собой количество двоичных векторов, формируемых одной фазой маршевого теста. Конкретный набор векторов, составляющих орбиту, будет зависеть от двух основных факторов [3]. Прежде всего, используемый маршевый тест однозначно определяет структуру орбиты, так как он формулирует правила, по которым генерируются двоичные векторы в ячейках запоминающего устройства. Начальное состояние запоминающего устройства является вторым аргументом, который принимается во внимание при формировании конкретной орбиты. Для примера рассмотрим биториентированное ОЗУ, состоящее из $k = 3$ ячеек. На рисунке 1 показан результат многократного выполнения простейшего маршевого теста $\{\uparrow(r0, w0)\}$, выполняющего инвертирование содержимого ячеек памяти, при различных начальных состояниях ОЗУ и фиксированном порядке следования адресов.

O_0	O_1	O_2	O_3	O_4	O_5	O_6	O_7
0 0 0 0 0 1 0 1 0	0 1 0 1 0 0 1 1 0	0 1 1 1 0 0 1 0 0	0 1 1 1 0 0 1 0 0	1 0 0 1 0 1 0 1 0	1 0 1 1 0 0 1 1 0	1 1 1 0 0 1 1 1 0	1 1 1 1 0 1 1 1 0
0 0 1 0 0 0 0 1 1 0	1 0 1 0 1 0 1 0 1 0	1 0 1 0 1 0 1 0 1 0	1 1 1 1 1 0 1 1 1 0	1 1 1 1 0 1 0 1 0 0	1 1 1 1 0 1 0 1 0 0	1 1 1 1 0 1 0 1 0 0	1 1 1 1 0 1 0 1 0 0
1 1 1 1 1 1 0 1 0 1	1 0 1 0 1 1 0 0 0 0	1 1 0 0 0 1 1 0 1 1	1 1 0 0 0 1 1 0 1 1	1 0 1 0 1 0 1 0 1 0	1 0 1 0 1 0 1 0 1 0	1 0 1 0 1 0 1 0 1 0	1 0 1 0 1 0 1 0 1 0

Рис. 1 – Орбиты для разных начальных состояний ОЗУ

Анализ рисунка позволяет сделать следующие выводы. Во первых: для k ячеек памяти существует всего 2^k возможных орбит. Во вторых:

в каждой орбите существует $k + 1$ двоичный вектор.

Рассмотрим два классических маршевых теста MATS++ $\{\uparrow(w0); \uparrow(r0, w1); \downarrow(r1, w0, r0)\}$ и MarchC- $\{\uparrow(w0); \uparrow(r0, w1); \uparrow(r1, w0); \downarrow(r0, w1); \downarrow(r1, w0); \uparrow(r0)\}$. Можно отметить, что одна итерация теста MATS++ формирует две орбиты, содержащие $k + 1$ различных двоичных вектора, а одна итерация теста March C- формирует четыре орбиты, содержащие $2k$ различных векторов.

Если в идеальном случае, все последующие орбиты теста MATS++ будут включать по $k + 1$ новому двоичному вектору, ранее не использованному в предыдущих орбитах, а все последующие орбиты теста MarchC- будут включать $2k$ новых двоичных векторов, то минимальное количество орбит, необходимое для формирования всех возможных комбинаций в любых k из N двоичных комбинаций для теста MATS++ должно удовлетворять неравенству [3]:

$$Q_{max} \geq \lceil \frac{2^k}{k+1} \rceil$$

а для теста March- неравенству

$$Q_{max} \geq \lceil \frac{2^{k-1}}{k} \rceil$$

где $k = 1, 2, 3, \dots$

Так как начальные состояния ячеек памяти, в случае вероятностных тестов, представляют собой случайные равномерно распределенные значения, то для определения среднего количества орбит, необходимых для формирования в любых k из N ячейках ОЗУ, всех 2^k возможных комбинаций, можно применить известную в комбинаторике задачу собирателя купонов [3]. В этом случае среднее количество орбит можно определить по выражению:

$$Q_{ave} = 1 + \frac{2^k}{2^k - 1} + \frac{2^k}{2^k - 2} + \dots + \frac{2^k}{2} + 2^k = 2^k \times \sum_{n=1}^{2^k} \frac{1}{n}$$

Учитывая количество различных двоичных векторов, формируемых одной итерацией тестов MATS++ и MarchC-, среднее количество орбит $Q_{ave}(\text{MATS}++)$ для теста MATS++ можно определить по формуле:

$$Q_{ave}(\text{MATS}++) = \frac{Q_{ave}}{k+1}$$

а среднее количество орбит для теста $Q_{ave}(\text{MarchC-})$ определяется выражением:

$$Q_{ave}(\text{MarchC-}) = \frac{Q_{ave}}{2k}$$

На рис. 2 и 3 представлены теоретические и экспериментальные оценки кратности для двух множеств многократных маршевых тестов с изменяемыми начальными состояниями MATS++

и MarchC- для произвольных k ячеек памяти. Экспериментальное значение величины Q_{ave} представляет собой среднее значение количества итераций тестов в процентах, полученное на основании 100 000 экспериментальных значений.

k	2	3	4	5	6	7	8	9	10
2^k	4	8	16	32	64	128	256	512	1024
Теор. Q_{ave}	2,77	5,44	10,82	21,64	43,37	86,93	174,20	349,00	699,03
Эксп. Q_{ave}	-	4,44	9,42	19,54	40,90	83,57	170,36	341,10	680,86
Эксп. Q_{min}	-	2	4	7	17	41	96	201	413
Эксп. Q_{max}	-	16	31	60	110	201	366	706	1405

Рис. 2 – Теоретические и экспериментальные значения $Q_{ave}(\text{MATS}++)$

k	2	3	4	5	6	7	8	9	10
2^k	4	8	16	32	64	128	256	512	1024
Теор. Q_{ave}	2,77	3,62	6,76	12,98	25,30	49,67	97,98	193,89	384,47
Эксп. Q_{ave}	-	2,33	4,42	9,34	19,55	40,55	82,62	169,32	339,48
Эксп. Q_{min}	-	2	2	4	7	18	37	86	213
Эксп. Q_{max}	-	7	17	28	65	113	207	391	769

Рис. 3 – Теоретические и экспериментальные значения $Q_{ave}(\text{MarchC-})$

III. ВЫВОДЫ

Анализ полученных данных показывает, что средние значения количества орбит, полученные по предложенным аналитическим выражениям, приближены к значениям, полученным экспериментально. Из рисунков видно, что среднее количество орбит для получения всех исчерпывающих комбинаций в любых k из N ячейках ОЗУ для теста MATS++ почти в два раза больше, чем для теста March C-. Это связано в первую очередь со структурой входящих в тест маршевых элементов. Учитывая, что одна итерация теста MATS++ генерирует две орбиты, а одна итерация теста March C- четыре орбиты, несложно подсчитать количество итераций многократных тестов. Таким образом, предложенные подходы могут быть использованы для построения многократных тестов, обнаруживающих сложные кодочувствительные неисправности ОЗУ, в том числе и ОЗУ встроенных систем.

СПИСОК ЛИТЕРАТУРЫ

1. Barzilai, Z. Exhaustive Generation of Bit Pattern with Application to VLSI Self-Testing / Z. Barzilai, D. Coppersmith, A. Rozenberg // IEEE Transactions on Computers. - 1983. - Vol. C.31, no. 2. - P. 190-194.
2. Das, D. Exhaustive and Near-Exhaustive Memory Testing Techniques and their BIST Implementations / D. Das, M.G. Karpovsky // Journal of Electronic Testing. - 1997. - Vol. 10. - P. 215-229.
3. Ярмолик, В.Н. Псевдоисчерпывающее тестирование ОЗУ / В.Н. Ярмолик, И. Мрозек, В.А. Леванцевич // Информатика. - 2017. - №2(54). - С. 58-69.
4. Yarmolik, S.V. The Syntheses of Probability Tests with a Small Number of Kits / S. V. Yarmolik, V.N. Yarmolik // Automatic Control and Computer Science. - 2011. - Vol. 45, no.3. - P. 133-141.
5. Ярмолик, С.В. Многократные неразрушающие маршевые тесты с изменяемыми адресными последовательностями / С.В. Ярмолик, В.Н. Ярмолик // Автоматика и телемеханика. - 2007. - №4. - С. 126-137.