Прийнявши $t_i = t_i^0$ і вирішивши друге рівняння системи (6), визначаємо необхідний час: t^0:

$$t^0 = \sum_{i=0}^{N+1} t^0_i$$

Завдання 3. Визначення максимуму ймовірності безвідмовної роботи комплексу $P(t,t)$ при заданому часі налагодження. Запишемо для цього випадку функцію Лагранжа:

$$f_2(t,t,w) = P(t,t) + w(\sum_{i=0}^{N+1} t_i - t_{зад}) \quad (7)$$

Диференціюючи (7) по t_i и w і прирівнявши отримані вирази нулю, отримаємо систему рівнянь:

$$\partial f_2(t,t,w)/\partial t_i = 0,$$

$$\sum_{i=0}^{N+1} t_i - t_{зад} = 0$$

Вирішивши друге рівняння системи, знаходяться значення t_i^0, що підставляються в $P(t,t)$ для визначення максимальної надійності СПК.

Висновки

Розглянуті в роботі моделі дозволяють робити оцінку надійності складних програмних комплексів при відомих показниках надійності складових модулів і експериментально отриманої стохастичної матриці перехідних станів. Вирішення обернених задач дозволяє проводити розрахунок надійності з урахуванням обмежень на ресурси, що залежать від кваліфікації розробника програмного забезпечення. Наведений метод може бути використаний для аналізу та оцінки надійності програмних комплексів в цілому, а також для дослідження надійності комплексу на початковому етапі розробки, що включає етапи налагодження і тестування.

Література

1. Липаев В.В. Качество программного обеспечения / В.В. Липаев. – М.: Финансы и статистика, 1983. – 261 с.

2. Карповский Е.Я. Надежность программной продукции / Е.Я. Карповский, С.А. Чижов. – Киев: Техника, 1990. – 160с.

3. Авижение А.Н. Гарантоспособные вычисления: от идей до реализации в проектах / А.Н. Авижение. – ТИИЭР, 1986. –Т.74. – № 5. – с. 8-21.

4. Musa J.D. Validity of Execution time theory of software reliability // IEEE Trans. On reliability. – 1979. – υ 3. – P.199-205.

5. Баглюк С.И. Надежность функционирования программного обеспечения / Баглюк С.И., Мальцев М.Г., Смагин В.А., Филимонихин Г.В. – С. – Пб.: -1991. – 78с.

6. Гантмахер Ф.Р. Теория матриц / Ф.Р. Гантмахер. – М.: Наука, 1988. – 548 с.

Надійшла 9.8.2011 р.

УДК 681.327

Д.Н. МОАМАР, В.Г. РЯБЦЕВ, Т.Ю. УТКИНА
Черкасский государственный технологический университет

МЕТОД ОЦЕНКИ ДИАГНОСТИЧЕСКИХ СВОЙСТВ ТЕСТОВ МИКРОСХЕМ ПАМЯТИ

В статті розглянуті питання діагностування несправностей мікросхем пам'яті. Наведені результати моделювання несправностей мікросхем пам'яті за допомогою моделей, представлених у вигляді графів цифрових автоматів зі скінченим числом станів. Проведено аналіз особливостей алгоритмів тестів мікросхем пам'яті, визначені діагностичні властивості тестів. На основі отриманих результатів розроблено метод оцінки діагностичних властивостей тестів мікросхем пам'яті.

The questions of fault diagnosis of memory chips are considered in the article. The results of fault simulation of memory chips using the models presented in the form of graphs of digital machines with a finite number of states. The analysis of the features of algorithms tests of memory chips is carried out; the diagnostic properties of tests are defined. On the basis of the received results the method of an estimation of the diagnostic properties of tests of memory chips is developed.

Ключові слова: мікросхеми пам'яті, несправності, діагностичні властивості тестів.

Введение

Имеется несколько очень веских причин, которые объясняют, почему методы диагностирования микросхем памяти заслуживают особого внимания. Во-первых, микросхемы памяти являются жизненно

важными электронными устройствами. Вряд ли найдется цифровая система, которая не включает в себя определенный тип памяти. Во-вторых, как и у других цифровых устройств, плотность и сложность микросхем памяти постоянно увеличивается. Хотя в отличие от большинства цифровых схем, микросхемы памяти имеют регулярные структуры, что первоначально может означать простоту диагностирования, но большое число состояний ячеек памяти не позволяет сформировать их все возможные состояния, что и вызывает сложность тестирования, такую, какой нет при изготовлении комбинационных логических схем. Кроме того, наиболее массовое распространение получили микросхемы памяти двух типов: статического типа (SRAM) и динамического типа (DRAM), что также вносит новые особенности при их функционировании.

Некоторые авторы делают предположения, что предлагаемая ими последовательность операций приводит к возникновению отказов или сбоев микросхем памяти и предлагают тесты для их локализации [1, 2]. Но трудно определить вероятность проявления таких неисправностей. Для оценки диагностических свойств тестов выполняют тестирование партии микросхем памяти, но при этом трудно определить неисправности каких типов определяют тесты [3]. «Жесткие» отказы микросхем памяти обнаруживают все тесты, а «мягкие» отказы проявляются редко, поэтому оценивать диагностические свойства тестов можно лишь косвенно по количеству забракованных микросхем, но также трудно определить какой тип неисправности обнаружил тот или иной тест.

Целью работы является разработка метода оценки диагностических свойств тестов по результатам исследования моделей микросхемы памяти, содержащих неисправности и представленных в виде графов цифровых автоматов с конечным числом состояний.

1. Моделирование неисправностей микросхем памяти

Неисправности могут возникать в различных частях микросхемы: в массиве запоминающих ячеек и в электронике обрамления. Последняя состоит из дешифраторов адреса, драйверов записи, усилителей считывания и регистров. Хотя, кажется, что электронику обрамления можно тестировать как обычную логическую схему, но любой отказ логической схемы можно фиксировать, как ответную реакцию на тест ячеек памяти. Таким образом, тестирование массива ячеек должно также включать тестирование электроники обрамления.

Работоспособная микросхема памяти должна обеспечивать хранение 1 и 0 в каждой ячейке и изменение состояния любой ячейки из 1 в 0 и наоборот, кроме того каждая ячейка должна сохранять свое значение после чтения.

Набор используемых функциональных моделей неисправностей микросхем памяти включает константные и перемежающие неисправности. Однако этих моделей не достаточно для представления всех типов отказов и сбоев в микросхеме памяти.

Константные неисправности (SAFs – stuck-at faults) возникают, если ячейки памяти ведут себя, как будто они постоянно хранят только 1 или 0. Для обнаружения константных неисправностей необходимо записать 1 (0) в ячейки, прочитать и сравнить с эталонным значением результат операции. К константным неисправностям следует отнести отсутствие доступа к ячейке (SOFs – stuck-open faults).

Частным случаем константных неисправностей являются неисправности перехода (TF – transition faults). Они происходят тогда, когда одна из ячеек не может сделать переход от 0 до 1 (↑) или от 1 до 0 (↓). Обозначения переходов верхними и нижними стрелками в предложено Ван де Гором в 1998 году. Для обнаружения этих неисправностей необходимо, чтобы при выполнении теста каждая ячейка изменяла свое состояние из 0 в 1 и из 1 в 0 и чтобы при этом выполнялась проверка полученных результатов.

Из регулярности структуры микросхемы памяти может происходить изменение состояния одной ячейки из-за изменения состояния в другой ячейке. Такие неисправности называются сопряженными (CF – coupling fault). Такие неисправности могут возникать, например, из-за наличия паразитных емкостных связей между ячейками. Различают сопряженную инверсию (CF inversion) и зависимый переход (CF idempotent). Неисправности типа сопряженная инверсия (CFins) возникают, когда смена состояния одной ячейки вызывает инверсию состояния в другой ячейке. То есть при выполнении перехода из 0 в 1 состояния ячейки i состояние ячейки j инвертируется.

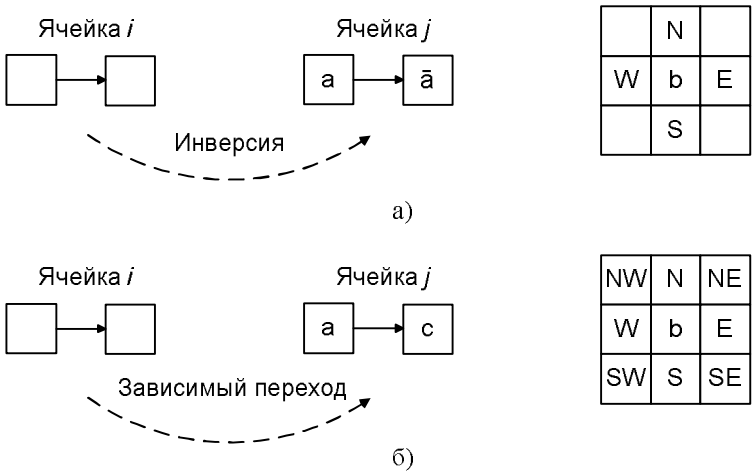

Рис. 1. Представления неисправностей:
а) – сопряженные неисправности; б) – NPSF неисправности

Две смежные ячейки, которые вызывают сопряженные неисправности можно рассматривать как частный случай k-мерного множества взаимодействующих ячеек. Однако исследовать взаимодействия множества ячеек большой размерности является слишком сложной задачей. Обычно рассматривают две

ячейки, влияющие, например, на угловые ячейки, показанные на рис. 1.

Тесты, обнаруживающие неисправности зависимого перехода (CFids), могут также обнаружить неисправности сопряженной инверсии (CFins).

Мостиковые неисправности (BF – bridging fault) обусловлены тем, что две или более линий непреднамеренно могут касаться друг друга. Замкнутые линии могут вести себя так, как будто они объединены операциями AND или OR над ними.

Другой тип неисправностей ячеек памяти обусловлен взаимным влиянием массива смежных ячеек памяти, что приводит к кодо-чувствительным неисправностям (PSFs – pattern-sensitive faults). Эти неисправности обусловлены в первую очередь влиянием нежелательных помех между ячейками в связи с их высокой плотностью.

На стадии разработки тестов, вероятно, более практичным является изучение повреждений ячеек расположенных близко друг к другу. Они называются соседствами ячеек, в которых возникают неисправности типа PSFs (NPSFs). Можно рассматривать соседства ячеек пятого порядка, в котором на базовую ячейку b воздействуют смежные ячейки $E, W, N, S,$ как показано на рис. 1, б. Можно также рассматривать соседство ячеек девятого порядка.

Неисправности типа NPSFs делятся на три категории: активные, пассивные и статические. При этом учитывается характер изменения состояний в соседних ячейках. Базовые ячейки могут (1) изменять состояния (active fault), (2) оставаться в фиксированном состоянии, независимо от попытки изменить его (passive fault), или (3) изменять конкретное значение (static fault). Модель неисправностей типа NPSF охватывает многие другие модели неисправностей микросхем памяти. Тем не менее, в связи с тем, что эти неисправности не являются общими, они требуют выполнения тестов очень высокой сложности. Задача построения тестов для обнаружения кодо-чувствительных неисправностей не нашла в настоящее время удовлетворительного решения. Для исследования диагностических свойств, проектируемых тестов, целесообразно использовать модели микросхем памяти.

Впервые модель оперативного запоминающего устройства предложена Хайесом С.П. (Hayes S.P.) [4] и представлена в виде цифрового автомата, содержащего множество запоминающих ячеек $C_i \in C$, которые подвергаются воздействию операторов: $O_i \in \{W_i,\ V_i, R_i\}$, где W_i – оператор записи единицы в ячейку C_i; V_i – оператор записи 0 в ячейку C_i; R_i – оператор считывания состояния ячейки C_i.

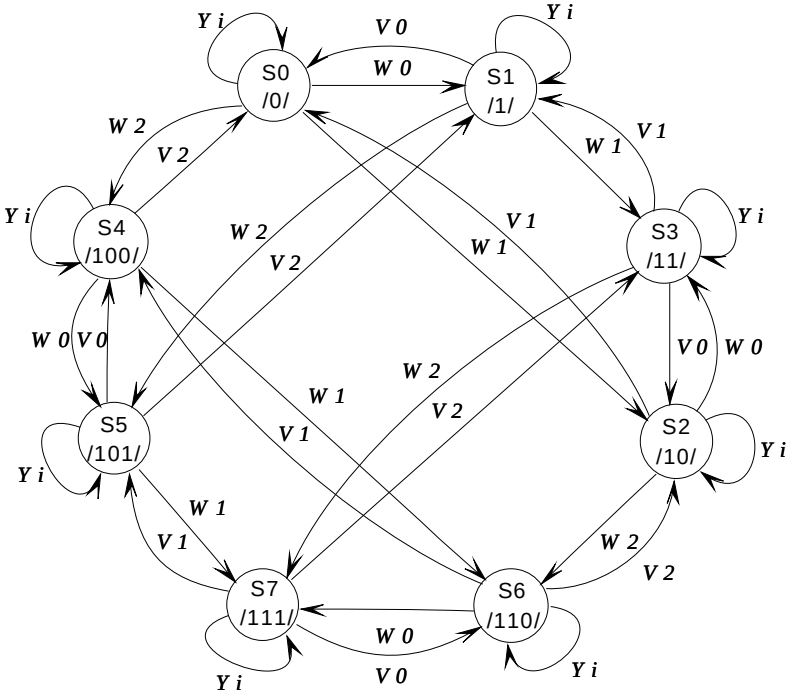

Рис. 2. Модель микросхемы памяти в работоспособном состоянии

Обычно емкость памяти в такой модели ограничивают двумя-тремя ячейками, что существенно сокращает продолжительность исследований. Модель микросхемы памяти в работоспособном состоянии, содержащая 3 ячейки памяти, приведена на рис. 2.

Число возможных состояний модели микросхемы памяти в работоспособном состоянии определяется по формуле:

$$S_m = 2^n,$$

где n – емкость микросхемы памяти.

Для $n = 3$ число состояний $S_m = 2^3 = 8.$

Количество операций, при помощи которых можно изменить состояния запоминающих ячеек модели микросхемы в работоспособном состоянии, определяется по формуле:

$$O_m = n \cdot 2^n.$$

Для трех запоминающих ячеек число возможных операций, изменяющих состояния ячеек памяти равно 24.

Для оценки диагностических свойств тестов предлагается модифицировать модель памяти Хайеса и

получать модели, которые описывают функционирование микросхем, содержащих неисправности. Константные неисправности микросхемы памяти обычно выявляют все тесты, поэтому такие неисправности в работе не рассматриваются.

Рассмотрим модель микросхемы памяти, содержащей сопряженную неисправность. Например, при переходе из состояния S_2 в S_6 произошла дополнительная запись единицы в ячейку C_0. Граф состояний модели микросхемы, содержащей приведенную выше неисправность, приведен на рис. 3.

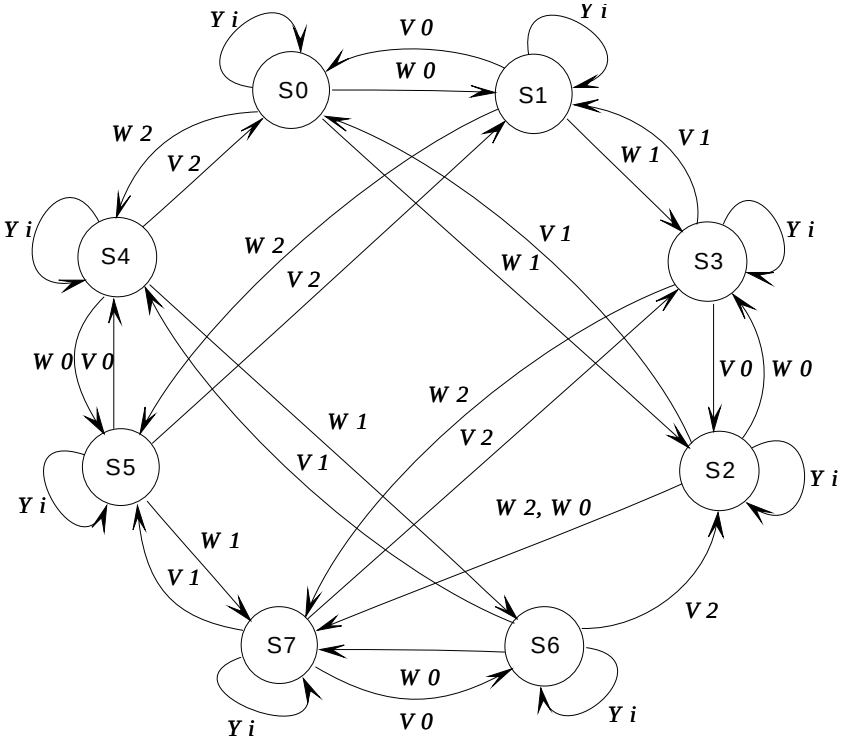

Рис. 3. Граф состояний модели микросхемы, содержащей неисправность

2. Исследование диагностических свойств тестов March_x и March_c

Для отображения операций диагностирования запоминающих устройств используется операторная форма записи алгоритмов [5, 6]. Операция обращения для записи нуля в ячейку памяти с адресом a отображает оператор $V(a)$, записи единицы – $W(a)$, считывания данных – $R(a)$.

Операции записи единиц во все ячейки при возрастании кода адреса записываются в виде последовательности операторов

$$\mathop{\mathbf{P}}_{a=0}^{n-1} W(a).$$

Здесь символ P определяет последовательность выполняемых операций, нижний индекс определяет начальный адрес ячейки памяти, а верхний индекс – конечный адрес запоминающих ячеек. Последовательность операций записи нулей во все ячейки при уменьшении кода адреса отображает следующая последовательность операторов:

$$\mathop{\mathbf{P}}_{a=n-1}^{0} V(a).$$

Операции считывания данных при возрастании кода адреса отображает запись:

$$\mathop{\mathbf{P}}_{a=0}^{n-1} R(a),$$

а при уменьшении кода адреса – $\mathop{\mathbf{P}}_{a=n-1}^{0} R(a).$

Рассмотрим особенности алгоритма теста March_x, последовательность операций которого имеет вид:

$$\pi_{MARCH\_X} = \mathop{\mathbf{P}}_{a=0}^{n-1}(V(a))\ \mathop{\mathbf{P}}_{a=0}^{n-1}(R(a)W(a))\ \mathop{\mathbf{P}}_{a=0}^{n-1}(R(a)W(a))\ \mathop{\mathbf{P}}_{n-1}^{a=0}(R(a)V(a))\ \mathop{\mathbf{P}}_{n-1}^{a=0}(R(a)).$$

При диагностировании данным тестом микросхемы емкостью 3 бита формируется следующая последовательность смены состояний, приведенная на рис. 4.

Количество состояний, которые формируются при выполнении теста March_x, вычисляется по формуле:

$$S_{m\_x} = 2^{n-1}.$$

Для микросхемы, содержащей 3 ячейки, число состояний, сформированных тестом March_x – $S_{m\_x} = 2^2 = 4.$ Столько же состояний из числа возможных не формируется. Из каждого несформированного тестом состояния S_2, S_4, S_5, S_6 в микросхеме возможны дополнительные переходы, которые могут образоваться при возникновении сопряженных неисправностей. Так как при выполнении теста March_x состояния S_2, S_4, S_5, S_6 не формируются, то неисправности, которые отображаются дополнительными переходами из данных несформированных состояний, данным тестом не выявляются. Количество неисправностей, невыявляемых тестом, пропорционально количеству несформированных им состояний в модели микросхемы памяти.

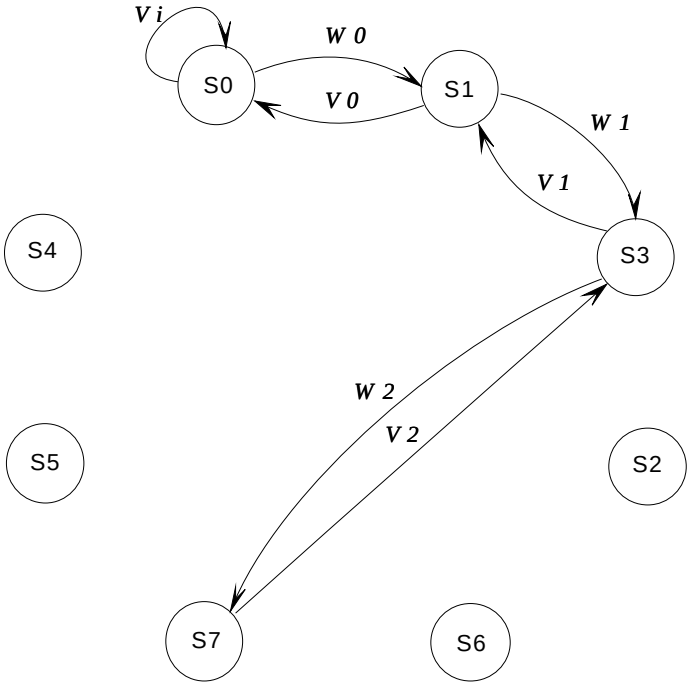

Рис. 4. Состояния, возникающие при выполнении теста March_x

Таким образом, диагностические свойства тестов можно оценивать при помощи коэффициента покрытия неисправностей, который вычисляется по формуле:

$$k_{eff} = 1 - \frac{(S_m - S_t)\cdot n}{O_m},$$

где S_t – число состояний модели микросхемы памяти, сформированных при выполнении теста.

Для теста March_x и $n = 3$ получаем значение коэффициента покрытия неисправностей:

$$k_{eff\_x} = 1 - \frac{(2^n - 2^{n-1})\cdot n}{n\cdot 2^n} = 1 - \frac{8-4}{8} = 0{,}5.$$

Рассмотрим особенности алгоритма теста March_c, последовательность операций которого имеет вид:

$$\pi_{MARCH\_C} = \prod_{a=0}^{n-1} (V(a)) \prod_{a=0}^{n-1} (R(a)W(a)) \prod_{a=0}^{n-1} (R(a)V(a)) \prod_{n-1}^{a=0} (R(a)W(a)) \prod_{n-1}^{a=0} (R(a)V(a)) \prod_{a=0}^{n-1} (R(a)).$$

При выполнении теста March_c и емкости микросхемы памяти 3 бита формируется следующая последовательность смены состояний, приведенная на рис. 5.

Число состояний, формируемых тестом $S_{t\_c} = 2\cdot n$, а количество операций, изменяющих состояние модели микросхемы $O_{m\_c} = 4\cdot n.$

Для теста March_c и $n = 3$ получаем значение коэффициента покрытия неисправностей:

$$k_{eff\_c} = 1 - \frac{(2^n - 2\cdot n)\cdot n}{n\cdot 2^n} = 1 - \frac{8-6}{8} = 0{,}75.$$

Таким образом, диагностические свойства теста March_c по обнаружению сопряженных неисправностей выше, чем теста March_x.

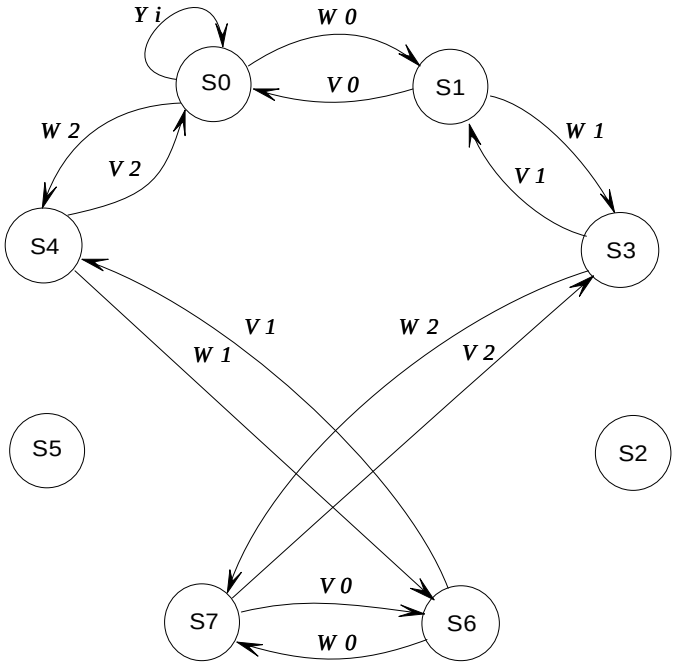

Рис. 5. Состояния, возникающие при выполнении теста March_c

3. Особенности алгоритма теста March_DSS

Исследуем диагностические свойства теста March_DSS по обнаружению сопряженных неисправностей. Данный тест является одним из последних достижений в области диагностирования запоминающих устройств и на него получен патент США [7]. Продолжительность теста составляет $46n$ операций обращения к микросхеме памяти и его алгоритм содержит 16 циклов.

Цикл 1. В первом цикле операции записи нулей во все ячейки:

$$\pi_{DSS\_1} = \prod_{a=0}^{n-1} V(a)$$

выполняются в состоянии S_0, приведенном на модели исправной микросхемы.

Цикл 2. Во втором цикле выполняется набор следующих операций:

$$\pi_{DSS\_2} = \prod_{a=0}^{n-1} (R(a)R(a)V(a)R(a)W(a)).$$

Первые 4 операции данного набора выполняются в состоянии S_0, а при выполнении четвертой операции осуществляется смена состояний $S_0 \to S_1; S_1 \to S_3; S_3 \to S_7$.

Цикл 3. В третьем цикле выполняется набор следующих операций:

$$\pi_{DSS\_3} = \prod_{a=0}^{n-1} (R(a)W(a)V(a)).$$

Первые 2 операции выполняются в состоянии S_7, а для выполнения третьей операции осуществляется смена состояний $S_7 \to S_6; S_6 \to S_4; S_4 \to S_0$.

Цикл 4. В четвертом цикле выполняется набор следующих операций:

$$\pi_{DSS\_4} = \prod_{a=n-1}^{0} (R(a)V(a)W(a)).$$

Первые 2 операции выполняются в состоянии S_0, а для выполнения третьей операции осуществляется смена состояний $S_0 \to S_4; S_4 \to S_6; S_6 \to S_7$.

Цикл 5. В данном цикле выполняется набор из 6-ти операций.

$$\pi_{DSS\_5} = \prod_{a=n-1}^{0} (R(a)W(a)R(a)V(a)V(a)R(a)).$$

Первые 3 операции выполняются в состоянии S_7, а для выполнения 4-й и 5-й операций осуществляется смена состояний $S_7 \to S_3; S_3 \to S_1; S_1 \to S_0$. Шестая операция выполняется в состоянии S_0.

Цикл 6. Все три операции данного цикла

$$\pi_{DSS\_6} = \prod_{a=n-1}^{0} (R(a)V(a)R(a))$$

выполняются в состоянии S_0.

Цикл 7. Для выполнения операций данного цикла

$$\pi_{DSS\_7} = \prod_{a=n-1}^{0} (R(a)W(a)R(a))$$

осуществляется смена состояний $S_0 \to S_4; S_4 \to S_6; S_6 \to S_7$.

Цикл 8. Все три операции данного цикла

$$\pi_{DSS\_8} = \prod_{a=n-1}^{0} (R(a)W(a))$$

выполняются в состоянии S_7.

Цикл 9. Для выполнения операций данного цикла

$$\pi_{DSS\_9} = \prod_{a=n-1}^{n-1} (R(a)V(a)R(a))$$

осуществляется смена состояний $S_7 \to S_6; S_6 \to S_4; S_4 \to S_0$.

Цикл 10. Для выполнения операций данного цикла

$$\pi_{DSS\_10} = \prod_{a=n-1}^{n-1} (R(a)W(a)R(a))$$

осуществляется смена состояний $S_0 \to S_1; S_1 \to S_3; S_3 \to S_7$.

Цикл 11. Для выполнения операций данного цикла

$$\pi_{DSS\_11} = \prod_{a=0}^{n-1} (R(a)V(a))$$

выполняется смена состояний $S_7 \to S_6; S_6 \to S_4; S_4 \to S_0$.

Цикл 12. Операции данного цикла

$$\pi_{DSS\_12} = \prod_{a=n-1}^{0} (R(a)V(a))$$

выполняются в состоянии S_0.

Цикл 13. Для выполнения операций данного цикла

$$\pi_{DSS\_13} = \prod_{a=0}^{n-1} (R(a)W(a))$$

выполняется смена состояний $S_0 \to S_1; S_1 \to S_3; S_3 \to S_7$.

Цикл 14. Все три операции данного цикла

$$\pi_{DSS\_14} = \prod_{a=n-1}^{0} (R(a)W(a)R(a))$$

выполняются в состоянии S_7.

Цикл 15. Для выполнения операций данного цикла

$$\pi_{DSS\_15} = \prod_{a=n-1}^{0} (R(a)V(a))$$

выполняется смена состояний $S_7 \to S_3; S_3 \to S_1; S_1 \to S_0$.

Цикл 16. Операции данного цикла

$$\pi_{DSS\_16} = \prod_{a=0}^{n-1} R(a)$$

выполняются в состоянии S_0.

Последовательность смены состояний трех запоминающих ячеек при выполнении теста March_DSS аналогична смене состояний, которые формируются тестом March_c.

Выводы

На основании анализа алгоритма теста March_DSS приходим к выводу, что число различающихся операций, которые изменяют состояния модели микросхемы памяти, данного теста и теста March_c, одинаково, также одинаково число несформированных состояний из числа возможных состояний модели микросхемы. Полученные сведения о свойствах теста March_DSS позволяют сделать вывод о том, что данный тест также как и тест March_c не обеспечивает обнаружение сопряженных неисправностей, которые на модели микросхемы отображаются дополнительными переходами из несформированных данными тестами состояний, например, из S_2 или S_5 (рис. 5). При тестировании микросхем памяти статического типа циклы 6, 8, 12 и 14, не изменяющие состояния ячеек памяти, можно исключить из алгоритма теста March_DSS. При этом общая продолжительность данного теста составит $36n$ операций обращения к микросхеме памяти.

Литература

1. Hamdioui S. March SS: A Test for All Static Simple RAM Faults / S. Hamdioui, A. J. Van de Goor, M. Rodgers // Proceedings of IEEE International Workshop on Memory Technology, Design and Test, 2002. – P. 95-100.

2. An Efficient March Test for Detection of All Two-Operation Dynamic Faults from Subclass Sav / G. Harutyunyan, H. Avetisyan, V. Vardanian, Y. Zorian // Proc. of East-West & Test International Workshop (EWDTW'09): (18-21 Sep. 2009, Moscow, Russia). – Kharkiv: KHNURE, 2009. – P. 175– 178.

3. Memory test experiment: industrial results and data / S. Hamdioui, A. J. Van de Goor, D. J. Reyes and others // IEEE Proc. – Computer. Digit. Tech.: (10 Jan. 2006). – Vol. 153. – № 1. – P. 1– 8.

4. Hayes S. P. Detection of pattern-sensitive fault in random-access memories / S. P. Hayes // IEEE Trans. Comp., 1975. – № 2. – P. 150– 157.

5. Аль Мади М.К. Алгоритмы тестового диагностирования полупроводниковых запоминающих устройств / М. К. Аль Мади, Д. Н. Моамар, В. Г. Рябцев. – К.: "Корнійчук", 2008. – 220 с.

6. Almadi Mudar. New Infrastructure for Memory Testing Design. / Mudar Almadi, Diaa Moamar and V. G. Ryabtsev // Proc. of the First International Workshop Critical Infrastructure Safety and Security. Kharkiv, National Aerospace University named N. E. Zhukovsky "KhAI", 2011. – P. 434– 440.

7. Pat. N20090199057A1 USA, Int.Cl.G11 C29/00. March DSS: Memory Diagnostic Test. – Publ. 06.08.2009.

Надійшла 28.8.2011 р.

УДК 004.056.52

К.В. КОЛЕСНІКОВ, А.О. ЛАВДАНСЬКИЙ
Черкаський державний технологічний університет

СПОСІБ ДИНАМІЧНОЇ БІОМЕТРИЧНОЇ ІДЕНТИФІКАЦІЇ

Важко переоцінити важливість задач захисту інформації в нашому житті. Одним з перспективних напрямків у цій задачі виступає біометрична ідентифікація особистості. В роботі розглянуті основні методи біометричної ідентифікації, а також запропоновано новий спосіб зчитування почерку (підпису) для методу рукописного вводу.

It is difficult to overestimate the importance of information security problems in our lives. One of the promising directions in this problem is the biometric identification. In work describes the main methods of biometric identification, and proposed a new way to read handwriting (signature) for the method of handwriting.

Ключові слова: біометрична ідентифікація, метод рукописного вводу.

Вступ

Засоби біометричної ідентифікації не вимагають запам'ятовування складних парольних фраз або логінів. Об'єкти, за якими відбувається ідентифікація, унікальні для конкретної особистості і їх дуже важко (в деяких випадках навіть неможливо) підробити. На сьогодні усі біометричні технології є імовірнісними, жодна з них не здатна гарантувати повну відсутність помилок FAR/FRR. Досліджень та методів гарантованої верифікації в наш час бракує. Тому актуальність методів, спрямованих на підвищення рівня достовірної ідентифікації поза всяким сумнівом.

Постановка завдання

Метою роботи є узагальнення методів біометричної ідентифікації, їх короткий огляд; більш детальне дослідження методу ідентифікації за динамікою рукописного вводу; пошук альтернативних способів вводу для описаного методу.

Методи біометричної ідентифікації

Слід розрізняти дві основні групи методів біометричної ідентифікації: ідентифікація за статичними ознаками особистості (такі, що не змінюються з часом) та динамічні методи ідентифікації (підсвідомі дії людини).

Методи статичної ідентифікації ґрунтуються на фізіологічних особливостях людини (ДНК, зображення обличчя, відбитки пальців, райдужна оболонка ока, сітківка ока, геометрія кисті руки, розташування вен на тильній стороні долоні та інші). Вони не змінюються протягом життя людини.

Методи динамічної ідентифікації ґрунтуються на поведінковій характеристиці людини (особливості голосу, динаміка рукописного\клавіатурного почерку та інші) [1].

Розглянемо методи статичної та динамічної ідентифікації більш детально.

Найбільш поширеним на даний момент методом є розпізнавання відбитків пальців. Даний метод ґрунтується на унікальності папілярного малюнка відбитку пальця. В процесі його роботи відбувається зчитування малюнку відбитку пальця за допомогою спеціального сканера, його аналіз і збереження інформаційної частини у базі даних. До недоліків даного методу слід віднести легкість пошкодження папілярного малюнку пальця, а також можливість підробки малюнку.

Методи ідентифікації за сітківкою і райдужною оболонкою ока ґрунтуються на унікальності кровоносних судин очного дна (для сітківки ока), і унікальності малюнка райдужної оболонки ока (для райдужної оболонки). Описані методи є найбільш надійними на даний момент. До недоліків можна віднести високу вартість впровадження.

Метод ідентифікації по зображенню обличчя ґрунтується на унікальності обличчя людини. Аналіз може проводитися як по двовимірному, так і по тривимірному зображенню обличчя людини. Зауважимо, що більш надійним є метод при використанні тривимірного зображення, але при цьому більш складним у реалізації. Як перевагу даного методу можна виділити ідентифікацію на відстані, об'єкту ідентифікації не обов'язково перебувати у безпосередній близькості до датчиків, а також контактувати з ними.

При використанні методу ідентифікації по венозному малюнку руки за допомогою інфрачервоної