

ИССЛЕДОВАНИЕ МЕТОДОВ ГЕНЕРАЦИИ МНОГОСВЯЗНЫХ МАРКОВСКИХ ЦЕПЕЙ В ЗАДАЧЕ СЕРТИФИКАЦИИ МИКРОПРОЦЕССОРНЫХ КОМПОНЕНТОВ

Введение

Сложность современных микропроцессорных устройств, более широкий класс возможных неисправностей сверхбольших интегральных схем (СБИС) резко обостряют проблему их сертификации [4,5] и контроля [1,2]. Во-первых, значительная часть неисправностей микропроцессорных СБИС проявляется лишь на высоких частотах, в то время как на низких частотах микропроцессор функционирует правильно. Во-вторых, кроме функциональных неисправностей механизмов обработки, управления, хранения для микропроцессоров характерны так называемые "нелюбимые коды", т.е. неисправности типа "чувствительность к определенным последовательностям команд". Указанные особенности обуславливают необходимость широкого применения методов функционального контроля, при котором отвлекаются от конкретной вентиляльной реализации микропроцессорной СБИС и рассматривают лишь выполняемые схемой функции.

К настоящему времени сложилось два подхода к функциональному тестированию микропроцессорных СБИС. Первый основан на построении регулярных (детерминированных) тестов для заданного класса неисправностей СБИС. Основным его недостатком является сложность построения тестов, так как число неисправностей для современных СБИС огромно. Второй подход связан с использованием псевдослучайных тестов, для построения которых не требуется явного перечисления неисправностей. Однако было обнаружено, что формируемые псевдослучайные тесты сильно избыточны и содержат запрещенные коды и их комбинации, которые могут привести к непредсказуемому поведению объекта контроля. Устранить указанные недостатки можно путем управления процессом генерации, т.е. изменением вероятности порождения тестовых воздействий в зависимости от особенностей конкретного объекта контроля.

1. Псевдослучайный контроль МП

Процесс контроля микропроцессора (МП) заключается в следующем. На МП подается детерминированная последовательность команд, которая устанавливает МП в исходное состояние: загружает регистры, сбрасывает признаки результата, устанавливает счетчик команд и указатель стека и т.д. Далее на входы МП подается псевдослучайная последовательность команд. Отклики проверяемого МП анализируются путем сравнения либо с откликами эталонного МП, либо с заранее вычисленными характеристиками (сигнатуры, контрольные суммы, число переходов 0-1 и др.). Следует подчеркнуть, что рассматриваемая методика контроля ориентирована на проверку функций, реализуемых микропроцессором, независимо от его внутренней структуры.

Очевидно, что с целью обнаружения неисправностей, характерных для микропроцессорных СБИС, метод псевдослучайной генерации должен удовлетворять следующим требованиям:

- генерация тест-программ должна осуществляться на частоте не ниже рабочей частоты контролируемого микропроцессора;
- получаемые псевдослучайные тест-программы должны обладать заданной полнотой контроля.

Как известно, полнота контроля обеспечивается соблюдением условий загрузки, проявлением и транспортировкой неисправностей, что возможно лишь при управлении стохастическими характеристиками генерируемых тест-программ. Под термином "управление" подразумевается возможность задания условных либо безусловных вероятностей порождения отдельных команд соответствующей тест-программы.

В простейшем случае псевдослучайные последовательности могут состоять из команд, вероятности появления которых фиксированы и не зависят от того, какие команды поступали на МП в предыдущих тактах. К сожалению, получаемые тест-программы характеризуются низкой полнотой контроля [1].

Повысить полноту контроля можно за счет изменения вероятностей соответствующих команд в зависимости от того, какие команды поступали на МП в предыдущих тактах. В качестве математического аппарата управления процессом формирования тестовых программ используются многосвязные марковские цепи. Стохастические свойства порождаемых тест-программ задаются с помощью

соответствующей матрицы переходных вероятностей, что обеспечивает требуемое качество и эффективность контроля.

2. Генератор многосвязных Марковских цепей

В задачах технической диагностики и функционального контроля наиболее часто используются марковские цепи с дискретным временем и конечным числом состояний [3]. Рассмотрим общий случай многосвязной однородной Марковской цепи. Будем рассматривать единственно возможные и несовместимые события

$$A_1, A_2, \dots, A_n$$

в неограниченном ряде испытаний с номерами $0, 1, 2, \dots$, которые соединяются в γ -членные звенья с номерами $0, 1, 2, \dots$, придавая номер h звену, объединяющему испытания с номерами

$$h, h+1, h+2, \dots, h+\gamma-1$$

Однородной γ -связной Марковской цепью будем называть такую последовательность испытаний, при которых вероятность появления события A_β в испытании с номером $k > \gamma$ равна

$$P_{\alpha_1 \alpha_2 \dots \alpha_\gamma \beta},$$

когда установлено появление событий

$$A_{\alpha_1}, A_{\alpha_2}, \dots, A_{\alpha_\gamma} \quad (1)$$

в γ предшествующих испытаниях, т.е. в испытаниях с номерами $k-\gamma, k-\gamma+1, k-\gamma+2, \dots, k-1$ (или в звене $k-\gamma$), при этом испытания с номерами $k+1, k+2$ и т. д. предполагаются еще не осуществленными и результаты их неопределенными и таким образом не влияют на результат.

Такое определение однородной многосвязной марковской цепи с конечным числом событий и дискретным временем (ММЦ) позволяет обобщить понятие МЦ. Например, «обычные» марковские цепи можно рассматривать как многосвязные со связностью 1. Независимые испытания, вероятность событий которых не зависят от предыдущих состояний системы, рассматриваются как ММЦ связности 0.

Законом цепи называется матрица порядка n^γ , составленная из переходных вероятностей $P_{\alpha_1 \alpha_2 \dots \alpha_\gamma \beta}$. Данные переходные вероятности полностью определяют характер поведения ММЦ.

Задание переходных вероятностей возможно несколькими способами. Все примеры приведены для ММЦ связности $n=2$ с числом состояний $\gamma=3$.

1 способ. Задание переходных вероятностей определяется квадратной матрицей размерности n^γ , горизонтальными и вертикальными координатами в которой выступают звенья. Однако как видно из примера на рис. 1 матрица получается сильно разреженной ввиду наличия большого числа несовместных событий, и размерность матрицы значительно возрастает с ростом n или γ , что ведет к значительным непродуктивным затратам памяти при машинной реализации.

		Звено $t-1$							
		11	12	13	21	22	23	31	3
Звено $t-2$	11	P_{111}	P_{112}	P_{113}	0	0	0	0	0
	12	0	0	0	P_{121}	P_{122}	P_{123}	0	0
	13	0	0	0	0	0	0	P_{131}	P
	21	P_{211}	P_{212}	P_{213}	0	0	0	0	0
	22	0	0	0	P_{221}	P_{222}	P_{223}	0	0
	23	0	0	0	0	0	0	P_{231}	P
	31	P_{311}	P_{312}	P_{313}	0	0	0	0	0
	32	0	0	0	P_{321}	P_{322}	P_{323}	0	0
	33	0	0	0	0	0	0	P_{331}	P

Рис. 1

2 способ. Задание переходных вероятностей определяется прямоугольной матрицей размерности $n \times n^2$, где по горизонтали располагаются состояния, а по вертикали звенья. Приведенный на рис.2 пример данной матрицы показывает избыточность данного способа представления данных.

		Состояние t		
		1	2	3
Звено $t-2$	11	P_{111}	P_{112}	P_{11}
	12	P_{121}	P_{122}	P_{12}
	13	P_{131}	P_{132}	P_{13}
	21	P_{211}	P_{212}	P_{21}
	22	P_{221}	P_{222}	P_{22}
	23	P_{231}	P_{232}	P_{23}
	31	P_{311}	P_{312}	P_{31}
	32	P_{321}	P_{322}	P_{32}
	33	P_{331}	P_{332}	P_{33}

Рис. 2

Аналогичным является представление матрицы переходных вероятностей в виде многомерного массива вида `var P:array[1..3,1..3,1..3]` в Pascal – нотификации.

3 способ. Все вышеперечисленные способы в машинной реализации являются применимыми при заранее известной размерности матрицы (то есть определены связность n и число состояний γ). В том случае, если данные параметры заранее неизвестны, или матрица переходных вероятностей является настолько разреженной, что не представляет смысла хранить ее в виде массива, предлагается представление в виде связанного дерева, как показано на рис. 3.

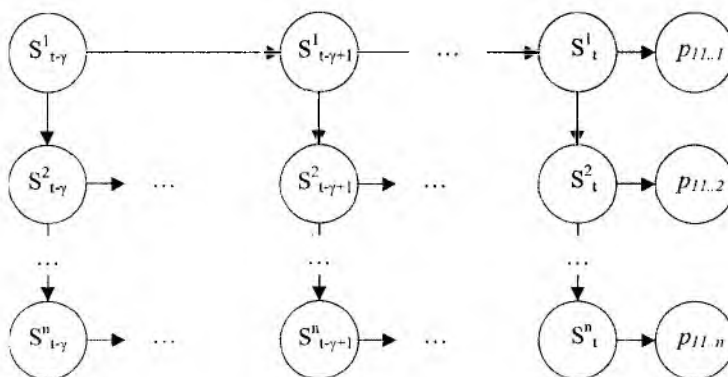


Рис. 3

Данная структура является универсальной с точки зрения программной реализации, позволяя строить матрицы переходных вероятностей для ММЦ с любыми n и γ .

Доказано[2], что наиболее благоприятным условием для генерации псевдослучайных тестовых программ является случай $H(P) \rightarrow 0$, где $H(P)$ -энтропия матрицы переходных вероятностей. В то же время данное условие является противоречивым по отношению к сущности метода псевдослучайного тестирования, так как при $H(P) = 0$ каждая строка матрицы переходных вероятностей содержит только одну единицу и все остальные нули. В данном случае можно утверждать, что генерируемая тестовая программа будет *детерминированной* и постоянной. Разрешить данное противоречие можно с помощью рационального формирования матрицы переходных вероятностей. В то же время из условия $H(P) \rightarrow 0$ можно сделать вывод, что матрица переходных вероятностей будет разреженной. В таком случае наиболее рациональным с точки зрения затрат памяти при больших n и γ является именно способ представления матрицы переходных вероятностей в виде связанного дерева.

3. Аппаратные и программные способы генерации ММЦ

Как известно, в общем случае для генерации случайной цепи с дискретными событиями необходимо произвести соответствие равномерно распределенной в диапазоне $[0,1]$ случайной величины $x=R[0,1]$ будущему состоянию S_i с помощью соотношений:

$$S = \begin{cases} S_0 & \text{если } x \leq F(S_0) \\ S_1 & \text{если } F(S_0) < x \leq F(S_1) \\ \dots & \dots \\ S_n & \text{если } x > F(S_{n-1}) \end{cases},$$

где $F(S)$ – функция распределения.

Представим теперь алгоритм генерации марковской последовательности произвольной связности как состоящий из 2-х этапов. На первом этапе по сформированной случайной величине x и функции распределения $F(S)$ определяется будущее состояние цепи. На втором этапе производится модификация массива предыстории, в соответствии с ним формируется новая $F(S)$ и изменяется модельное время. Таким образом, для программной генерации многосвязных марковских последовательностей предлагается обобщенный алгоритм, представленный на рис. 4.



Рис. 4

В данном алгоритме выделенные вершины являются вариантными по отношению к размерности генерируемой цепи. Рассмотрим более подробно функции каждой вершины. В вершине «изменение массива предыстории events» происходит модификация занесением будущего состояния, представленная на рис. 5.

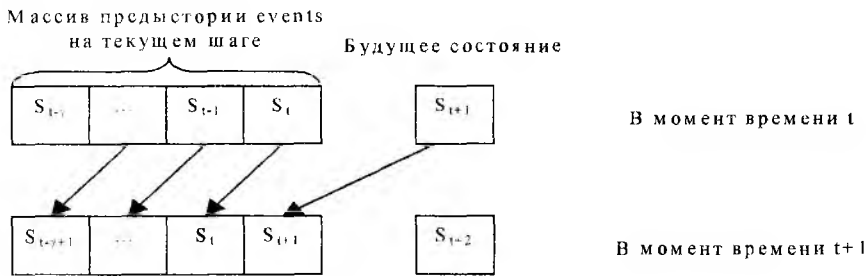


Рис. 5

В вершине «модификация функции распределения» происходит формирование новой функции распределения в соответствии с измененным массивом предыстории. Более детальные функции данной вершины сильно зависят от формы представления матрицы переходных вероятностей. Наиболее удобно данный алгоритм заложить в рамки объекта, реализующего функции загрузки матрицы переходных вероятностей из файла или базы данных, начальной инициализации, формирования случайной величины x , определения будущего состояния, изменения массива предыстории и получения соответствующей функции распределения. В целях исследования эффективности данного алгоритма составлена программа на языке C++ с применением библиотеки STL, содержащая объявление описанного выше класса, реализующего все 3 способа хранения матрицы переходных вероятностей и связанной с этим функциональности.

Для ускорения процесса формирования тестовых воздействий и разгрузки процессора возможно применение аппаратно реализованного генератора многосвязных марковских последовательностей. Структура данного формирователя представлена на рис. 6.

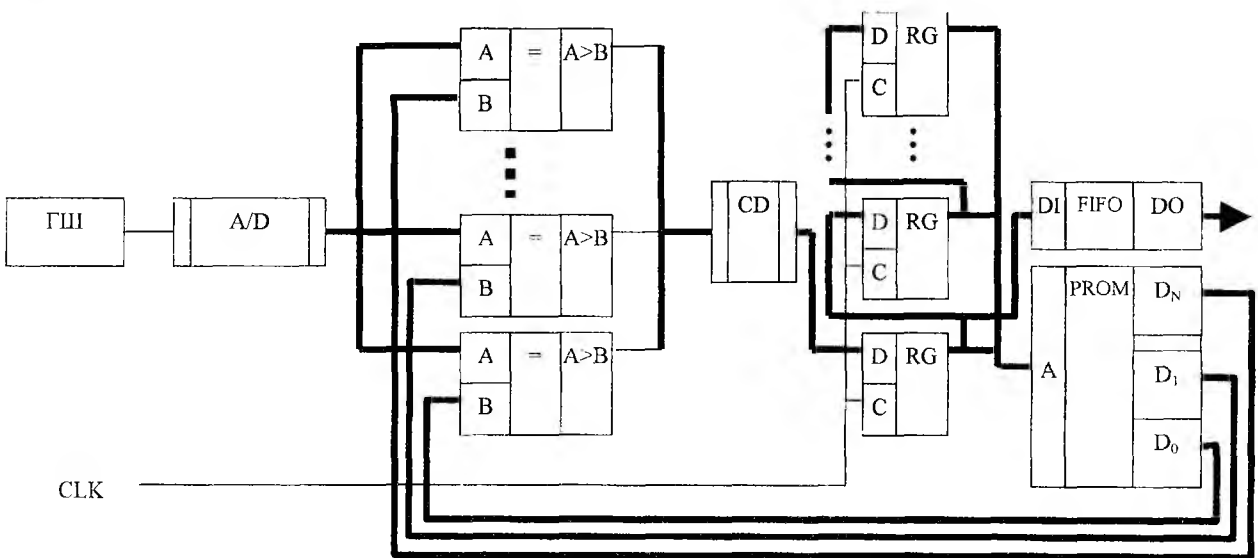


Рис.6

В структуре данного формирователя можно выделить следующие основные блоки:

- ГШ (генератор шума) и A/D (аналого-цифровой преобразователь), выполняющие функцию генерации случайного числа x .
- Блок цифровых компараторов, на вход A поступает случайная величина x , а на вход B – код $F(S_t)$ функции распределения, хранящейся в PROM.
- Приоритетный шифратор CD, формирующий код состояния на основе информации с выходов цифровых компараторов.
- Блок из y регистров, реализующий функции хранения предыстории процесса. Выходы блока регистров подаются на адресный вход PROM, формируя таким образом соответствующую данной предыстории процесса функцию распределения $F(p)$.
- Буфер FIFO для хранения и воспроизведения формируемой последовательности.

Как видно из структуры данного формирователя, в нем используется второй способ представления матрицы переходных вероятностей, что предопределяет невозможность динамического изменения размерности матрицы и параметров цепи n и γ .

Код с выхода аналого-цифрового преобразователя, представляющий собой случайную величину x , параллельно поступает на входы блока цифровых компараторов, на вторые входы которых подается коды соответствующей предыстории функции распределения. Выходы компараторов подаются на вход приоритетного шифратора, формирующего код будущего состояния по позиции старшей единицы. Код будущего состояния с выхода приоритетного шифратора по переднему фронту тактового импульса сохраняется в младшем регистре блока регистров, реализующего функции массива предыстории. Одновременно с этим происходит сдвиг значений в остальных регистрах по направлению к старшему регистру, реализуя приведенное на рис. 5 преобразование. Код с младшего регистра сохраняется в буфере FIFO, что дает возможность чтения сформированной последовательности блоками в управляющий компьютер и воспроизведения последовательности. Так как выходы блока регистров составляют адресный вход для PROM, изменения в блоке регистров приводят к изменениям на выходах данных $D_0 \dots D_n$, представляющих значения функции распределения для состояний $S_0 \dots S_n$. Измененная функция распределения подается на вторые входы блока компараторов, завершая цикл формирования одного состояния. Данная структура реализована на базе программируемых ИМС AL-TERA FLEX 8000 в САПР MAX + PLUS II и показала высокое быстродействие. В качестве элементов памяти использовались высокоскоростные микросхемы Motorola MCM 6806 BRJ-6 с временем доступа 6 нс. К преимуществам данной структуры можно отнести высокое быстродействие, реализуемое за счет параллельного сравнения и сдвига массива предыстории. К недостаткам следует отнести специфические требования к организации памяти и связанную с этим невысокую точность представления матрицы переходных вероятностей. Как видно из структуры формирователя, матрица переходных вероятностей хранится в параллельном виде. Это определяет, например, при связности $n=2$ с числом состояний $\gamma=3$ и восьмибитовом представлении вероятностей следующие требования к организации памяти: разрядность 16 бит, глубина 9. Можно сделать вывод, что данная структура более применима в случаях с небольшим числом состояний γ .

Для исследования алгоритмов формирования многосвязных марковских последовательностей была написана исследовательская программа с использованием пакета Inprise C Builder 4. Входными данными для программы являются количество состояний, связность МЦ, матрица переходных вероятностей и начальное состояние. Все исходные данные хранятся в базе данных типа DBF, доступ к которой осуществляется через библиотеку BDE. Программа генерирует γ -связные ММЦ и ведет статистики появления заданного состояния или цепочки состояний. Управление программным обеспечением комплекса диагностики осуществляется с помощью технологии OLE, для чего необходимо изначально зарегистрировать сервер OLE в системе. Тестирование программы показало высокое быстродействие предлагаемого метода генерации МЦ.

Заключение

В данной статье рассмотрены вопросы сертификации микропроцессорных компонентов с точки зрения псевдослучайного тестирования. Особое внимание было уделено проблеме высокочастотной генерации тестовых воздействий на основе математического аппарата многосвязных марковских последовательностей. В качестве результатов исследования выступает разработанный способ представления матрицы переходных вероятностей в виде связного дерева, обобщенный алгоритм и структура аппаратного формирователя многосвязных марковских последовательностей. Тестирование с помощью разработанных программных модулей и платы прототипирования показало высокую эффективность предлагаемых методов.

Список литературы: 1. Хаханов В.И. Техническая диагностика элементов и узлов персональных компьютеров. ХТУРЭ.-К.: ИСМО, 1997. 308 с. 2. Клисторин И.Ф., Гремальский А.А. Функциональный контроль микропроцессорных устройств. Минск: Знание, 1990. 90 с. 3. Дыкин Е.Б. Марковские процессы. М.: Физматгиз, 1963. 860 с. 4. Анип Б.Б. Защита компьютерной информации. СПб.: BHV, 2000. VIII. 368с.:ил. 5. Программно аппаратные средства обеспечения информационной безопасности / В.Г. Проскурин и др. // Защита в ОС. М.: Радио и связь, 2000. 166с.

Харьковский государственный технический университет радиотехники

Поступила в редколлегию 5.04.2001