взаимодействует с анализатором XML для получения информации о разметке.

Полная схема взаимодействия при десериализации выглядит следующим образом. Код приложения устанавливает связь между XML Parser и блоком создания и конфигурации объектов (Factory). После этого XML Parser и Factory взаимодействуют напрямую. Анализируя заданный документ, XML Parser посылает сообщения Factory. Factory, принимая обработанные элементы разметки текста, получает информацию о необходимых действиях из XML/RTTI Binding, который хранит отображение во внутренних структурах данных, полученных при обработке Binding файла на начальном этапе инициализации системы. Получив необходимые данные для данного элемента разметки, Factory обращается к модулю RTTI Access для создания или конфигурации объекта.

При сериализации модуль Serializer получает информацию о сохраняемом объекте через модуль RTTI Access. Элементы разметки текста, соответствующие классу данного объекта, Serializer получает из блока связывания и формирует вывод в формате XML.

Как показано на рис.2, XML Parser и Serializer формируют уровень поддержки формата (Format support layer) и только эти два блока обрабатывают все операции, связанные с внутренним представлением потока сериализации. Такая структура позволяет локализовать изменения, связанные с форматом, на уровне одной подсистемы, не затрагивая остальной код программы. Кроме того, изменения в представлении предметной области вызывают только модификацию классов данных и файлов привязки. Следовательно, изменения формата или классов данных локализованы на уровне подсистем, т.е. не влекут за собой изменения в других подсистемах. Следует отметить, что предложенная архитектура позволяет легко менять формат сериализации. Такой переход осуществляется заменой классов в блоках XML Parser и Serializer на альтернативные, реализующие те же интерфейсы, а также заменой внешних файлов связывания на альтернативные файлы в новом формате. При наличии нескольких реализаций таких унифицированных интерфейсов и их динамическом связывании на этапе выполнения смена формата может осуществляться баз перекомпиляции системы и даже во время ее выполнения. Таким образом, предложенная модель может служить основой для построения систем с гетерогенным информационным обменом, при котором общение с различными источниками информации осуществляется в разных форматах, а ядро системы позволяет подключать обработчики новых форматов и не зависит от формата внешнего представления данных.

Литература: 1. *Bosak J.* XML, Java,and the Future of the Web // XML.com. XML: Principles, Tools and Techniques. Edited by. Dan Connolly and Rohit Khare, 1997. Oct 02. 2. *Manola F.* Towards a Web Object Model, Object Services and Consulting, Inc. (OBJS). 1998. 3. Capturing the State of Distributed Systems with XML by Rohit Khare and Adam Rifkin (XML special issue of the World Wide Web Journal, Autumn 1997. Vol. 2, № 4. P. 207-218.

Поступила в редколлегию 04.12.1998

Рецензент: д-р техн. наук Смеляков С.В.

Евсюков Александр Юрьевич, аспирант филиала кафедры ИИИС ХТУРЭ. Научные интересы: объектно-ориентированные технологии, компонентные модели, Internet технологии. Адрес: Украина, 310145, Харьков, ул. Новгородская, 20, кв. 54, тел. 45-51-32, e-mail: evsukov@email.com

---

УДК 681.324:519.713

МЕТОДЫ МОДЕЛИРОВАНИЯ ЦИФРОВЫХ СТРУКТУР. ТРЕХТАКТНЫЙ АВТОМАТ

ХАХАНОВ В.И., СКВОРЦОВА О.Б., ХАНЬКО В.В., БЕДРАТЫЙ Р.В.

Описываются методы моделирования цифровых устройств (ЦУ), ориентированные на верификацию проектов относительно существования опасных состязаний сигналов, гонок, рисков сбоев, обусловленных практической реализацией дискретных устройств в виде интегральных микросхем, плат. Показываются направления развития методов цифрового логического моделирования по пути повышения их адекватности.

1. Формализация задач логического анализа цифровых устройств

Моделирование можно рассматривать как процесс определения неизвестных компонентов тетрады <M,D,T,R> – <модель, дефекты, стимулы, реакция> на основе применения правил логического вывода. Если принять во внимание, что каждый из упомянутых компонентов тетрады может принимать значения: 0 – отсутствие; 1– наличие, то множество всех возможных значений <M,D,T,R>, заданных в двоичном алфавите, можно представить в виде табл.1, где столбец Y задает компоненты тетрады, которые могут быть определены в результате выполнения соответствующих алгоритмов логического анализа. Запись по табл.1 дизъюнктивной нормальной формы для вычисления каждого из компонентов формализует четыре основные проблемы логического анализа, которые дифференцируются в классические задачи технической диагностики.

Таблица 1

MDTR	Y
0000	-
0001	M
0010	-
0011	M
0100	-
0101	M
0110	-
0111	M
1000	DTR
1001	TD
1010	RD
1011	D
1100	TR
1101	T
1110	R
1111	-

1. Проектирование модели цифрового изделия при наличии или отсутствии неисправностей:

$M = R \vee DR \vee TR \vee DTR = R(1 \vee \vee D \vee T \vee DT)$.

1.1. Определение модели цифрового устройства в виде совокупности всех его возможных реакций на функциональные входные наборы:

$M = f^{1,1}(T,D,R)|_{D=\varnothing;\ T=\varnothing}$.

1.2. Создание модели цифрового устройства в виде совокупности всех его возможных реакций на функциональные входные наборы при наличии неисправностей заданного класса (словарь неисправностей):

$M = f^{1,2}(T,D,R)|_{T=\varnothing}$.

1.3. Построение модели цифрового устройства в виде входных набо-

ров и соответствующих им реакций – таблица истинности (переходов):

$M = f^{1,3}(T,D,R)|_{D=\varnothing}$.

1.4. Создание модели цифрового устройства в виде входных тестовых наборов и соответствующих им реакций при наличии заданных неисправностей – таблица функций неисправностей:

$M = f^{1,4}(T,D,R)$.

2. Определение множества потенциальных, фактических или проверяемых дефектов в цифровом изделии:

$D = M \vee MT \vee MR \vee MTR = M(1 \vee T \vee R \vee TR)$.

2.1. Генерация списков неисправностей цифрового изделия по его модели в виде константных или функциональных дефектов:

$D = f^{2,1}(M,T,R)|_{T=\varnothing;\ R=\varnothing}$.

2.2. Моделирование заданных неисправностей цифрового изделия на входных тестовых наборах в целях определения качества теста:

$D = f^{2,2}(M,T,R)|_{R=\varnothing}$.

2.3. Определение фактических неисправностей цифрового изделия на функциональных наборах по экспериментальным реакциям – обратное прослеживание дефектов:

$D = f^{2,3}(M,T,R)|_{T=\varnothing}$.

2.4. Определение фактических неисправностей цифрового изделия на тестовых наборах по экспериментальным реакциям – безусловное диагностирование:

$D = f^{2,4}(M,T,R)$.

3. Построение теста верификации модели, проверка исправности, обнаружение заданных неисправностей:

$T = M \vee MD \vee MR \vee MDR = M(1 \vee D \vee R \vee DR)$.

3.1. Проектирование теста (проверка исправности) цифрового изделия по его модели:

$T = f^{3,1}(M,D,R)|_{D=\varnothing;\ R=\varnothing}$.

3.2. Проектирование теста (проверяющего, диагностирования) цифрового изделия по его модели и заданным неисправностям:

$T = f^{3,2}(M,D,R)|_{R=\varnothing}$.

3.3. Проектирование теста (проверка исправности) цифрового изделия по его модели исправного поведения и заданным реакциям – обратная импликация:

$T = f^{3,3}(M,D,R)|_{D=\varnothing}$.

3.4. Проектирование теста (проверяющего, диагностирования) – определение состояний входных переменных цифрового изделия по его модели, заданным дефектам и реакциям – обратная импликация:

$T = f^{3,4}(M,D,R)$.

4. Определение реакции цифрового изделия – прямая импликация:

$R = M \vee MD \vee MT \vee MDT = M(1 \vee D \vee T \vee DT)$.

4.1. Определение состояний невходных линий цифрового изделия как реакции модели на входные последовательности:

$R = f^{4,1}(M,D,T)|_{D=\varnothing;\ T=\varnothing}$.

4.2. Определение состояний невходных линий цифрового изделия как реакции модели на входные последовательности при наличии заданных неисправностей:

$R = f^{4,2}(M,D,T)|_{T=\varnothing}$.

4.3. Определение состояний невходных линий цифрового изделия как реакции модели на входные тестовые последовательности:

$R = f^{4,3}(M,D,T)|_{D=\varnothing}$.

4.4. Определение состояний невходных линий цифрового изделия как реакции модели на входные тестовые последовательности при наличии заданных дефектов – моделирование неисправностей заданного класса: $R = f^{4,4}(M,D,T)|$.

Таблица 2

Компоненты	Определение модели цифрового устройства
{f}	Функциональная синхронная модель не учитывает временных задержек и структуры объекта
{t}	Временная модель учитывает время прохождения сигналов от входов к выходам без учета структуры и функций
{h}	Структурная модель определяется графом, который не учитывает функций и временных параметров примитивных элементов структуры
{f,t}	Функциональная асинхронная модель (формирование значений выходов связано с временными задержками элемента)
{f,h}	Структурно-функциональная синхронная модель есть совокупность взаимосвязанных элементов без учета временных параметров
{h,t}	Структурно-временная модель имеет структуру в виде графа элементов, для которых определены временные задержки прохождения сигналов
{f,t,h}	Структурно-функциональная асинхронная модель учитывает все параметры и является наиболее адекватной объекту среди всех упомянутых

В формулировках проблем $f^{i,j}$ (i=1,4; j=1,4) есть система логических преобразований (методы и алгоритмы), предназначенная для рационального решения конкретной задачи анализа или проектирования.

Многообразие моделей цифрового изделия представлено следующей триадой компонентов [1]:

$$F = \langle f, t, h \rangle;$$

здесь f, t, h – дискретные параметры функциональной, временной, структурной адекватности: $f=(f_1,...,f_i,...,f_k)$; $t=(t_1,...,t_i,...,t_m)$; $h=(h_1,...,h_i,...,h_n)$, где k, m, n – мощности незамкнутых множеств известных способов формализации для описания функций, времени, структур. Построение пространства соотношения адекватностей моделей есть экспериментальная работа, сопряженная со сравнением существующих способов формализации задания параметров <f,t,h>. Для анализа цифровых объектов при проектировании диагностического обеспечения интерес представляет множество всех подмножеств универсума F, задающее отображенную в табл.2 классификацию моделей цифровых объектов.

Компонент f отображает многообразие аналитических, табличных и графических форм представления моделей. Относительно t классификация моделей определяет синхронные (не учитывающие временные параметры объектов и примитивов), асинхронные (использующие реальные или модельные задержки элементов), дельта-троичные (учитывающие модельные задержки примитивных элементов (ПЭ) и разброс времени переключения входных сигналов объекта) и с нарастающей неопределенностью (момент переключения входных сигналов элементов учитывает разброс параметров задержки прохождения сигналов от внешних входов к выходам). Параметр h классифицирует степень подробности задания структуры и идентифицирует: функциональную или автоматную модель для представления поведения комбинационного или последовательностного ПЭ; структурно-функциональную или итеративную, а также чистую структуру, представляющую собой орграф.

Развитие методов моделирования традиционно осуществляется по двум направлениям, связанным с отсутствием и наличием дефектов в схеме [2,3]. В первом случае речь идет об анализе переходных процессов в целях выявления и последующего устранения опасных состязаний на невходных линиях схемы, которые идентифицируются символами X в фактических состояниях линий, что может быть следствием конкретной структурной реализации [4,5]. Во втором – моделирование служит инструментом для оценки качества теста относительно одиночных константных неисправностей (ОКН) [6,7]. Многообразие тех и других методов моделирования ставится в зависимость от алфавита описания технического состояния линий, форм представления моделей обьектов и алгоритмов их обработки [8-10].

2. Прямая импликация на кубическом покрытии

Алгоритм выполнения прямой импликации

$$(X^{(t-1,t)}, Y^{t-1}) \rightarrow Y^{t}$$

по модели примитива определяется способом его описания. Если это аналитическая запись, в том числе и на алгоритмических языках, то для определения выходной реакции примитива на входное слово необходимо иметь универсальные или специальные решатели – трансляторы или компиляторы, дорогостоящие, но имеющие высокое быстродействие. Недостаток аналитических моделей – их высокая стоимость, определяемая сложностью написания и отладки фактических программ моделирования примитивов. Кроме того, такие модели не подлежат автоматическому анализу или синтезу в целях проектирования общих структур данных устройства или декомпозиции последнего на составные части, а также их нельзя использовать для решения других задач технической диагностики. Графовое представление поведения примитива может быть оформлено в аналитическую запись и тогда для нее будут характерны все те преимущества и недостатки, которые названы выше.

Табличная форма описания модели есть явная запись системы отношений на множестве входных (внутренних) и выходных переменных, благодаря чему имеет при одном недостатке (большой объем структур данных в сравнении с аналитическим, что требует значительных временных затрат) такие преимущества:

1. Универсальность и простота алгоритмов анализа и синтеза табличных моделей.

2. Возможность декомпозиции таблицы на элементы с генерацией для последних собственных таблиц меньшей размерности.

3. Композиция таблиц отдельных примитивов для создания единой таблицы истинности цифрового устройства.

4. Универсализм табличных моделей для решения задач прямой и обратной импликации, моделирования неисправностей и исправного поведения, генерации тестов, контроля и поиска дефектов заданного класса.

Учитывая высокую производительность современных компьютеров и практическую безграничность информационных емкостей памяти ЭВМ, недостаток табличных форм не является существенным.

Процедура анализа таблицы [1,9] заключалась в сравнении входного (выходного) слова со строками, и если такое имело место, то состояние выходов (входов) элемента определялось подстановкой значений соответствующих координат из рассматриваемой строки таблицы. D-исчисление Рота [2] также ничего не добавило к упомянутой процедуре анализа, хотя уже появилась логическая избыточность алфавита в виде символов {0,1,X}. Методы генерации тестов на основе использования кубических (вырожденных) покрытий (КП) [5] ориентированы на решение импликативных задач без рассмотрения алгоритмов моделирования исправного поведения и неисправностей.

Введение избыточности, предложенной Ротом, в алфавит послужило развитием более сложных процедур анализа, поскольку метод подстановки в чистом виде уже не обеспечивал правильного решения из-за наличия символа X. Следующие доказательства показывают возможность построения процедур для адекватного моделирования цифровых объектов.

Определение 1. Пересечение векторов в замкнутом теоретико-множественном алфавите является непустым (непротиворечивым), если результирующий вектор не содержит символов пустого множества.

Определение 2. Объединение векторов в замкнутом теоретико-множественном алфавите будет непустым (пустым), если каждый из них не содержит символов (содержит символы) пустого множества.

Определение 3. Пересечение векторов, один из которых имеет символ пустого множества (U), равно пустому множеству.

Теорема. Непустое объединение непротиворечивых результатов пересечений входного вектора с кубами покрытия определяет состояния всех переменных объекта в замкнутом теоретико-множественном алфавите.

Если все результаты пересечения вектора E с каждым кубом покрытия $C = \{C_1, C_2, \ldots, C_i, \ldots, C_n\}$ равны пустому множеству, то их объединение равно U. Будем считать, что исходные векторы E и C_i свободны от символов U. Отсюда следует отсутствие в n-мерном векторном пространстве общей области у покрытия C и вектора E. Это может быть лишь в случае задания неполного кубического покрытия для описания логических функций объекта, или противоречивости определения системы отношений между входными и выходными переменными вектора E. Например, для КП ПЭ 2И-НЕ, имеющего три куба: {0X1, X01, 110}, выполнение пересечения вектора E=(000) с каждым кубом покрытия определяет

результат, равный U. Если же предположить отсутствие в упомянутом КП последнего куба, то пересечение вектора E=110 с каждым кубом неполного покрытия будет пустым. Это значит, что пустым будет и объединение – свидетельство некорректности задания входного вектора E или неполноты КП.

Если же существует хотя бы одно непустое пересечение между E и C_i, то согласно определениям 1 и 2 их объединение будет непустым. Оно идентифицирует логические состояния всех линий объекта как общую область n-мерного пространства для E и C, которая есть максимально возможный общий результат реакции объекта на входное слово E.

Следствие 1. Для выполнения прямой импликации по КП примитивного элемента необходимо априорное определение невходных координат вектора E символами X.

Максимальная неопределенность невходных координат позволяет получить максимально возможное множество непротиворечивых результатов пересечения между C и E, формирующих самое общее состояние переменных. Любой другой вариант предварительного задания невходных линий лишь подтвердит существование частного решения или опровергнет его в случае получения пустого объединения результатов пересечений.

Следствие 2. Реакция комбинационного объекта на двоичный вектор не может быть троичной. Это следует из функционирования конечного автомата, когда двоичному входному слову не могут быть поставлены в соответствие две различные реакции, в противном случае он переходит в класс недетерминированных автоматов, которые здесь не рассматриваются. Следовательно, даже при наличии нескольких непустых пересечений между E и C результат должен быть только двоичным по всем линиям ПЭ, в противном случае покрытие не принадлежит к КП комбинационного цифрового автомата.

Следствие 3. Реакция комбинационной схемы на троичный входной вектор может быть троичной.

При существовании хотя бы одного символа X в векторе E фактически можно говорить о компактной записи двух двоичных векторов. Каждый из них может иметь собственную реакцию выходов, которые не всегда совпадают, что и формирует при их объединении троичную реакцию на невходных линиях. Если вектор E с одним символом X определяет на выходе значение неопределенности, то данная входная переменная существенна или активна на данном векторе E.

Избыточность однотактного алфавита кубического исчисления, определяемая двумя символами {0,U}, один из которых не значащий, привела к усложнению процедуры анализа КП посредством введения двух векторных операций пересечения и объединения. Однако компактность получаемых моделей функционально сложных комбинационных устройств можно считать результатом, превосходящим понесенные издержки [1,5].

Избыточность двухтактного алфавита кубического исчисления не изменяет сущности теоретико-множественного анализа КП, но позволяет оформить модели сложных последовательностных, а также комбинационных схем в КП приемлемых размеров. Далее предлагаются способы анализа ЦУ и их примитивов, основанные на использовании двухтактных покрытий в общем случае, где КП комбинационных схем есть частный случай описания цифрового автомата [1,2,5].

3. Автомат описания функций примитивов

Проектирование процедур и алгоритмов анализа исправного поведения цифровых устройств на основе использования двухтактного кубического исчисления предполагает исследование функциональных особенностей составляющих компонентов. Классификация разнообразия примитивов и их структур определяется следующей совокупностью качественных характеристик, которые следует учитывать при создании системы моделирования:

1. Многовыходовой комбинационный или последовательностный элемент, в котором каждый выход функционально зависим от всех входов. Для такого примитива определяется полное КП, задающее все возможные переходы на пространстве входных состояний. Примерами таких ПЭ могут служить логические элементы, сумматоры, коммутаторы, дешифраторы, преобразователи кодов, триггеры, счетчики, управляющие конечные автоматы.

2. Многовыходовой элемент, представленный КП, в котором состояние выходной переменной определяется подмножеством кубов на ограниченном множестве существенных для данного выхода линий. Значения других выходов на упомянутых кубах покрытия могут не определяться благодаря наличию на них символов Z. Примеры таких примитивов – микросхемы, составленные из нескольких, несвязанных между собой логических, комбинационных или последовательностных элементов; многофункциональные регистры, где отдельные операции не требуют участия всех переменных при их описании.

3. Многовыходовой ПЭ, имеющий двунаправленные линии, которые должны быть как входами, так и выходами при описании КП примитива. При проектировании структур данных ЦУ такие переменные должны быть отнесены ко входам.

4. Многовыходовой примитив, имеющий неполное покрытие, которое формирует отдельные функции ПЭ на векторе существенных переменных.

5. Структура элементов, имеющих полные КП, нагруженных на выходы одного потенциала – монтажная логика И (ИЛИ).

6. Совокупность элементов с неполными КП (которые формируют отдельные операции функционального элемента), имеющих выходы одного потенциала – объединение выходов для определения их значений посредством анализа всех нагруженных на них примитивов.

7. Функционалы, реализующие арифметические операции при использовании в качестве входов и выходов одних и тех же линий.

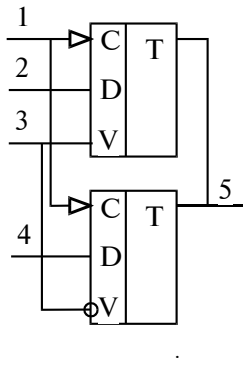
8. ПЭ, имеющие на входах и выходах многоразрядные шины и управляющие передачей информации от A к B, и наоборот.

Триггеры и КП представлены на рис. 1.

Упомянутое многообразие примитивов и схемных соединений предполагает модификацию автомата первого рода к так называемому U-автомату (Universimom) $U=<X,Y,f>$, ориентированному на анализ цифрового объекта и определяемому функцией выходов

$$Y(t+1)=f[X(t-1),X(t),Y(t)]$$

на множестве входных и выходных (невходных) состояний. Автомат может не иметь инициирующего начального состояния, но решение задачи установки



1	2	3	5
E	0	1	0
E	1	1	1
F	X	1	S
X	X	0	Z

1	4	3	5
E	0	0	0
E	1	0	1
F	X	0	S
X	X	1	Z

Рис. 1. Функциональная модель структуры

схемы по всем линиям в наперед заданное достижимое двоичное состояние за конечное число входных наборов должно иметь положительный результат. Отсутствие классификации линий на внутренние и выходные связано с необходимостью отображения логического состояния обьекта (включающего понятие состояние автомата), которое является условием для моделирования очередного перехода объекта при подаче входного набора. Однако существует необходимость выделения на множестве невходных переменных ЦУ наблюдаемых линий, которые имеют гальванические связи с контактами внешнего разъема, что нужно для реализации алгоритмов генерации тестов, моделирования неисправностей, контроля и поиска дефектов. В соответствии с предложенным автоматом можно определить структуру куба покрытия примитива:

	X					Y				
t-1: M0	0	1	X	1	0					
t : M1	1	1	0	0	X	0	1	X	0	1
t+1: M2						1	Z	0	U	X

(1)

Система отношений между входными и выходными переменными конечного автомата определена на трех тактах. Это позволяет создавать модели примитивов, которые смогут модифицировать не только линии Y, но, что самое важное, и значения переменных X без использования дополнительных псевдопеременных. Отсюда следуют возможности:

1. Описание входных условий в двух автоматных тактах с целью повысить адекватность и компактность моделей примитивов, что на практике эквивалентно: исключению необходимости введения дополнительных переменных, которым нельзя поставить в соответствие физическую линию; записи наличия или отсутствия переднего или заднего фронтов на синхровходах, задаваемых символами {E,F,H,L} [1]; формированию минимальных условий по отдельным входным линиям в такте t-1 или t с помощью {0,1,X,G,T,K}; заданию таких переходов на входных координатах, которые не вызывают установку автомата в непредсказуемое состояние; определению изменений входных сигналов, инициирующих опасные состязания в ПЭ.

2. Задание условий и направления устойчивого перехода автомата на физически существующих выходных (невходных) переменных в тактах t и t+1 или его установки в неопределенность.

3. Формирование отношений для функций, которые используют одни и те же физические линии в качестве входных и выходных {A=A+1, A=A-1}; {A=B, B=A}.

4. Автомат моделирования

ЦУ в виде конечного автомата, использующего структуру примитива (1), адекватно моделируется в трех автоматных тактах <t-1, t, t+1>, которым соответствуют поля M0,M1,M2:

	X					Y				
M0	0	1	X	1	0	1	1	0	0	X
M1	1	1	0	0	X	0	1	X	0	1
M2	U	U	U	U	U	U	U	U	U	U

(2)

Векторы M0X, M1X – координаты инициирующих сигналов на входах объекта; M2 – поле задания координат, состояния которых должны быть определены в процессе анализа кубических покрытий; M1Y – поле-идентификатор состояния объекта в момент t; M0Y – определяет значения невходных линий в момент t-1, которые в совокупности с полем M1Y формируют входные сигналы для моделирования внутренних примитивов схемы. Поскольку для схемы, как и для примитива, нет смысла делить невходные переменные на внутренние и выходные, определение конечного S-автомата (Simulation) имеет вид

$$Y(t+1)=f[X(t-1),X(t),Y(t-1),Y(t)],$$

где Y(t+1) – поле модификации входных или выходных координат. Его основное назначение состоит в определении поведения ЦУ на множестве реальных эквипотенциальных линий схемы, которая имеет глобальные обратные связи, фронтальную синхронизацию, двунаправленные линии, элементы памяти. Для таких устройств важно знать только входные линии и то лишь в целях выставления на них инициирующих воздействий, а разделение невходных переменных на внутренние и выходные нужно, как и в случае с примитивом, лишь для идентификации последних в качестве наблюдаемых линий при работе алгоритмов проектирования диагностической информации. Отсюда следует, что U-автомату примитива в формате (2) будет соответствовать фрагмент:

	X					Y				
M0	0	1	X	1	0					
M1	1	1	0	0	X	0	1	X	0	1
M2	U	U	U	U	U	U	U	U	U	U

Здесь состояние Y(t+1) может определяться любой координатой из M2.

Такая структура объясняет поведение функций, где переменные аргументов и результата представлены одними и теми же эквипотенциальными линиями. Например, КП функций {A=A+1, A=A-1}; {A=B, B=A} невозможно реализовать на основе концепции классического автомата без использования дополнительных переменных или элементов памяти.

5. Процедуры анализа цифровых объектов

Отход от классического задания конечного автомата в двух временных фреймах и введение трехтактного автомата есть результат принятия структурно-функциональной модели устройства без псевдоразрыва глобальных обратных связей и введения дополнительных псевдопеременных с целью повысить адекватность моделирования исправного поведения цифрового объекта. Для эффективной обработки такой модели следует учитывать концепции введенных U-, S-автоматов, разнообразие используемых структур ПЭ и схемных соединений, значность

алфавитов моделирования и описания переходов в целях задания оптимальных и технологичных операций и массивов, чтобы реализовать программные средства моделирования исправного поведения.

1. Векторы моделирования (2) предназначены для формирования входного слова в троичном алфавите и хранения состояний всех линий как результатов анализа КП элементов схемы. Текущее значение входов, для которых необходимо найти реакцию невходных линий, заносится в поле M1X, при этом поле M0 по всем координатам определяет состояния всех переменных автомата в момент t-1; M1Y – уже определенные значения линий устройства в момент t. Поле M2 есть вычисляемые состояния всех переменных автомата в момент t+1 в результате анализа кубических покрытий на заданных значениях M0, M1 по входам и M1 – по выходам. Поле M2X может быть модифицировано относительно M1X только в случае наличия в схеме функциональных элементов, имеющих входные и выходные переменные, отмеченные одинаковыми номерами линий, которые при построении структур данных относятся к входным переменным модели ЦУ.

Идентичность полей M1, M2 по невходным координатам является признаком окончания итеративного процесса при моделировании очередного входного набора. Если такого не происходит, то при достижении максимального, наперед заданного числа итераций всем изменяющимся на невходных полях M1, M2 координатам присваивается символ X, после чего цикл вычислений повторяется. Такая процедура управления простыми итерациями гарантирует сходимость алгоритма моделирования при наличии в схеме опасных состязаний или генераторных режимов. Исходное состояние трех векторов до начала моделирования должно быть равно символу U (здесь он эквивалентен значению X) по всем координатам. Далее, перед обработкой очередного активного примитива или группы ПЭ, нагруженных на одни и те же выходы, последним должны быть присвоены символы U в поле M2. Такая избирательность обусловлена событийным характером алгоритма анализа схемы, когда ПЭ будет моделироваться в случае его активности – наличия изменений на входных линиях примитива в полях M0, M1, что является условием для формирования в позиции вектора активности VA, соответствующей текущему элементу, признака 1.

2. Вектор объединенных выходов VF определяется числом линий схемы, где содержимое позиции задает количество выходов ПЭ, нагруженных на линию с данным номером:

X			Y			Z		
0	0	1	1	1	2	1	3	1
1	2	3	4	5	6	7	8	9

Символ 1 в позиции 3 свидетельствует о двунаправленности линии 3, на которую нагружен один выход ПЭ; линии 6 и 8 объединяют 2 и 3 примитива соответственно. Очевидно, переменные 1 и 2 являются только входами; 4,5,7,9 – выходами, соединенными со входами предшествующих элементов. Наличие символа >0 на входной координате или >1 – на внутренней или выходной служит условием для обработки по сквозному (безусловному) алгоритму всех элементов, связанных с отмеченными линиями, чтобы получить на них адекватное реальному поведению решение, поскольку только все примитивы, нагруженные на объединенные выходы, создают полную картину их состояний.

3. Координатные операции конкатенации, пересечения, объединения [1] определяют технологию анализа КП примитивов, сочетающую свойства простоты метода и максимальных функциональных возможностей, которые можно выжать из синхронного метода моделирования в пятизначном алфавите {0,1,X,Z,U}. Операция конкатенации (*) предназначена для получения двухтактного символа кодирования состояния линии в фреймах (t-1,t) или на полях (M0,M1) с целью выполнить последующее пересечение двухтактных кубов покрытий с аналогичными состояниями координат упомянутых полей векторов моделирования. Операция ($\cap$) формирует результат пересечения куба КП с исходными для рассматриваемого ПЭ значениями соответствующих входных, выходных линий, полученных конкатенацией состояний упомянутых переменных, заданных полями M0,M1. Для повышения быстродействия анализа КП операция пересечения формирует результат в виде состояния линии в момент t+1 по схеме, представленной на рис. 2.

Выполнение $\cap$-операции формирует символ двухтактного алфавита, в данном случае E, который может быть разложен к виду двух однотактных, а далее #-операция вычисляет однотактную составляющую в момент t, которая переносится Δ-операцией во фрейм t+1. Бинарная операция пересечения заменяет собой последовательность действий: $\cap, \rightarrow, =, \rightarrow, \#, \rightarrow, \Delta$ (см. рис.2) без потери адекватности моделирования. Естественно, $\cup$-операция, предназначенная для объединения непротиворечивых результатов пересечений векторов моделирования и кубов покрытия, определена на символах однотактного алфавита, что делает ее компактной. Результирующие значения в ней представлены пятью символами, которые идентифицируют: {0,1} – устойчивое двоичное состояние; X – переход линии в неопределенное значение; Z – обозначение высокого импеданса для линий, которые могут иметь три устойчивых состояния; U – неопределяемое значение переменной, связанное с отсутствием для моделируемого входного набора системы отношений, формирующих состояние линии. Появление символа U есть следствие неполного КП или наличия в схеме номера линии, которым не отмечена ни одна переменная в множестве примитивов объекта. Чтобы повысить быстродействие выполнения упомянутых операций, символы двухтактного алфавита кодируются однобайтными десятичными числами от 0 до 22.

Отличие таблично-кодовой организации вычисления операций от варианта последовательности условных операторов заключается в реализации за один программный такт любого пересечения, объединения, конкатенации путем нахождения содержимого ячейки n-мерного массива по его индексам. Таким образом, программные средства оперируют кодами символов, но при вводе и выводе необходимо иметь преобразователи "символ-код" и "код-символ" для удобства восприятия информации пользователем.

4. Буферный вектор моделирования предназначен для формирования по выходам объединения непус-

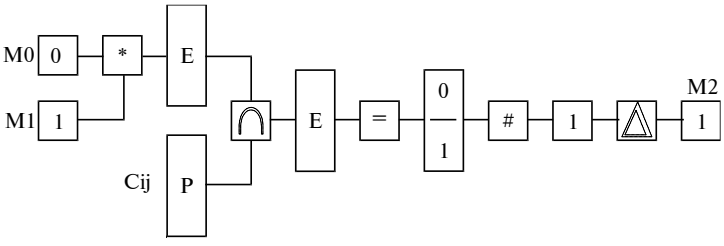


Рис. 2. Формирование результатов пересечения

тых результатов пересечений конкатенации значений векторов моделирования M0,M1 с каждым кубом покрытия по линиям, соответствующим обрабатываемому ПЭ.

Рассмотрим схему анализа куба примитива и формирования выходных значений элемента в буферном векторе моделирования VB.

Каждая строка КП имеет множество входных и невходных (внутренних и выходных) координат. Чтобы она участвовала в формировании состояния выходов, необходима ее непротиворечивость по входным, а также по невходным координатам в момент t (вектор M1), которые определяют предыдущее состояние автомата. Процедура анализа куба по всем координатам:

1. Конкатенация состояний очередной входной координаты ПЭ, выбранных из векторов моделирования.

2. Выполнение операции пересечения между результатом конкатенации и входной координатой рассматриваемого куба.

3. Если пересечение пусто, выполняется переход к анализу следующего куба (i=i+1). В противном случае рассматривается следующая входная координата (j=j+1) анализируемой строки.

4. Если по всем входным координатам куба зафиксировано непустое пересечение, выполняется переход к анализу невходных координат, для чего реализуется конкатенация состояния очередной невходной линии $M1_j$ с символом пустого множества, который здесь равен X.

5. Пересечение конкатенации с очередной невходной координатой анализируемого куба покрытия. Результат непустого пересечения заносится в соответствующую позицию буфера VB_j.

6. Если результат предыдущей операции пуст, осуществляется переход к анализу очередного куба по входным линиям (i=i+1). Иначе – анализ невходной координаты (j=j+1).

7. Если по всем состояниям невходных линий зафиксировано непустое пересечение, выполняется покоординатное объединение полученных в буфере значений внутренних и выходных переменных ПЭ с состояниями соответствующих линий вектора моделирования M2 с записью полученного результата в упомянутый вектор.

Иллюстрацией выполнения процедуры анализа кубов покрытия примитивов (см. рис.1) может служить результат моделирования отдельных входных наборов, представленных в табл. 3.

В примере моделирования вектор M01 есть конкатенация двух предыдущих по всем координатам; буферные векторы Bi (VBi) получаются в результате пересечения куба C_i с вектором M01 по правилам операции пересечения; M12 равен вектору M1 по входам и множеству M2 вычисленных истинных значений – по выходам. Если на линию нагружено несколько выходов ПЭ, результат определяется объединением состояний данной выходной переменной в векторе M2, полученных при анализе соответствующих покрытий. Объединение результатов моделирования дает:

Таблица 3

Анализ КП 5									
Ci	1	2	3	5	Ci	1	4	3	5
1	E	0	1	0	1	E	0	0	0
2	E	1	1	1	2	E	1	0	1
3	F	X	1	S	3	F	X	0	S
4	X	X	0	Z	4	X	X	1	Z
M0	0	0	0	1	M0	0	0	0	1
M1	1	0	0	U	M1	1	0	0	U
M01	E	Q	Q	B	M01	E	Q	Q	B
B1			U		B1	1	0	0	0
B2		U			B2		U		
B3			U		B3	U			
B4	1	0	0	Z	B4			U	
M12	1	0	0	Z	M12	1	0	0	0

$$\begin{array}{cccc} & 1 & 0 & 0 & Z \\ \cup & 1 & 0 & 0 & 0 \\ \hline M12 & 1 & 0 & 0 & 0 \end{array}$$

Описанные процедуры анализа примитивов и схем ориентированы на максимальное приближение проектируемых моделей к схемотехническим особенностям микросхем и цифровых объектов в целях нахождения рынка пользователей программными средствами моделирования в среде разработчиков радиоэлектронной аппаратуры.

При моделировании неисправностей алгоритм ориентируется на наблюдаемые линии, которые могут быть как входными, так и внутренними или выходными. Идентификация наблюдаемых переменных необходима также для выполнения процедур генерации тестов, организации и проведения диагностического эксперимента. Естественно, чем больше таких линий, тем проще процедуры анализа ЦУ, но дороже устройство из-за увеличения числа выводов на внешнем разъеме.

Литература: 1. *Хаханов В.И.* Техническая диагностика элементов и узлов персональных компьютеров. К.: ІЗМН. 1997. 308 с. 2. *Основы* технической диагностики / Под. ред. П.П.Пархоменко. М.: Энергия, 1976. 460с. 3. *Breuer M.A., Friedman A.D.* Diagnosis and reliable design of digital system. Woodlound wills, Computerscience press inc., 1976. 308p. 4. *Автоматизированное* проектирование цифровых устройств / С.С.Бадулин, Ю.М.Барнаулов и др./ Под ред. С.С. Бадулина. М.: Радио и связь, 1981. 240с. 5. *Баранов С.И., Майоров С.А., Сахаров Ю.П., Селютин В.А.* Автоматизация проектирования цифровых устройств. Л.: Судостроение, 1979. 264с. 6. *Автоматизация* диагностирования электронных устройств / Ю.В.Малышенко и др./ Под ред.В.П.Чипулиса.М.: Энергоатомиздат, 1986. 216с. 7. *Беннеттс Р.Д.* Проектирование тестопригодных логических схем: Пер.с англ. Дербуновича Л.В. М.: Радио и связь, 1990. 176с. 8. *Богомолов А.М., Сперанский Д.В.* Аналитические методы в задачах контроля и анализа дискретных устройств. Саратов: Издво Сарат. ун-та, 1986. 240с. 9. *Байда Н.П., Кузьмин И.В., Шпилевой В.Т.* Микропроцессорные системы поэлементного диагностирования. М.: Радио и связь, 1987. 256с. 10. *Courtois B.* CAD and testing of ICs and systems. Where are we going?– TIMA, France. 1995. 250 p.

Поступила в редколлегию 12.11.1998

Рецензент: д-р техн. наук, проф. Кривуля Г.Ф.

Хаханов Владимир Иванович, д-р техн. наук, профессор кафедры АПВТ ХТУРЭ. Научные интересы: техническая диагностика вычислительных устройств, систем, сетей. Хобби: баскетбол, горные лыжи. Адрес: Украина, 310726, Харьков, пр. Ленина, 14, тел. (0572) 40-93-26.

Скворцова Ольга Борисовна, аспирантка кафедры АПВТ ХТУРЭ. Научные интересы: техническая диагностика вычислительных устройств. Хобби: аэробика, музыка, иностранные языки. Адрес: Украина, 310726, Харьков, пр. Ленина, 14, тел. (0572) 40-93-26.

Ханько Вадим Викторович, аспирант кафедры АПВТ ХТУРЭ. Научные интересы: диагностика вычислительных систем и сетей. Адрес: Украина, Донецкая обл., г. Мариуполь, ул. Новороссийская, 14, кв. 87, тел. (0629) 35-21-33, 37-70-93.

Бедратый Роман Витальевич, аспирант кафедры АПВТ ХТУРЭ. Научные интересы: техническая диагностика вычислительных устройств. Хобби: тяжелая атлетика, культуризм. Адрес: Украина, 310726, Харьков, пр. Ленина, 14, тел. (0572) 40-93-26.