

Конструирование знаний для автоматизации разработки тестов цифровых устройств

Рассматривается организация знаний для разработки тестов цифровых устройств, обсуждается их иерархическая структура. Предлагается подход к конструированию знаний для автоматизации разработки диагностического тестового обеспечения на основе конкурентного метода моделирования неисправностей.

Разработка цифровых устройств как интегрированная задача проектирования и тестирования является предметной областью систем, основанных на знаниях. Накоплен определенный опыт применения экспертных систем для проектирования цифровых схем [1] и СБИС [2, 3], а также диагностирования электронных схем различной сложности [4–6]. Хотя немногие из существующих систем автоматизируют процесс получения знаний, существует тенденция к оформлению в искусственном интеллекте самостоятельного направления — системы приобретения знаний [7]. Однако в приобретении знаний для экспертных систем имеется ряд серьезных трудностей. С одной стороны, это чисто технологические и технические ограничения, с другой, — это большое влияние субъективного фактора: несоответствие представления знаний экспертом и формализации последних в базах знаний; неспособность, а порой и нежелание эксперта выразить и передать знания. Поэтому среди возможных способов получения знаний предпочтительнее те, в которых эксперт непосредственно не участвует [8], т. е. — из книг, справочников, инструкций.

В настоящей статье предлагается подход к конструированию знаний в инструментальных средствах автоматизации разработки тестового обеспечения цифровых устройств.

Средства автоматизации разработки тестов аппаратуры содержат знания справочного и нормативного характера, традиционные для баз данных САПР: временные и функциональные характеристики, электрические номиналы, спецификацию структуры и т. п. для каждого из типов компонент элементной базы. Уровень моделирования в рассматриваемых инструментальных средствах — поэлементный. Части компоненты, реализующие одинаковую функцию, будем называть подсхемами; вырожденный случай представляют собой компоненты из одной подсхемы.

Декларативные знания об элементной базе (понятия «тип_компоненты», «структура_компоненты», «структура_подсхемы», «вывод») представлены соответствующими фреймами класса. Для разработки тестов достаточно в фрейме классов ТИП_КОМПОНЕНТЫ иметь слоты ИМЯ_ТИПА, СТРУКТУРА, ПОВЕДЕНИЕ и МЕТОД_ТЕСТИРОВАНИЯ, ПОВЕДЕНИЕ и МЕТОД_ТЕСТИРОВАНИЯ описывают соответственно алгоритм функционирования подсхемы компоненты конкретного типа и метод получения тестов для нее. Понятие СТРУКТУРА конструируется иерархически: фреймом класса СТРУКТУРА КОМПОНЕНТЫ описывается наличие ПОДСХЕМ в фрейме класса СТРУКТУРА_ПОДСХЕМЫ указывается список ВЫВОДОВ; в фрейме класса ВЫВОД содержатся слоты ОБОЗНАЧЕНИЕ (имя), НОМЕР (цоколевка), НАЗНАЧЕНИЕ (вход/выход/двухнаправленный выход и т. д.).

Множество допустимых типов компонент элементной базы представляют фреймы-шаблоны. В этих фреймах по сравнению с соответствующими фреймами класса для каждого типа компоненты означаются отдельные слоты, значения которых будут наследоваться в дальнейшем фреймами-экземплярами, а также определяются слоты для характеристик экземпляров компонент данного типа. Так, в фрейме-шаблоне КОМПОНЕНТА_ТИПА_K531ЛН1 фиксируются значения

слов ИМЯ_ТИПА, СТРУКТУРА и ПОВЕДЕНИЕ и вводят слоты ИДЕНТИФИКАТОР, НОМЕР, ПРИНАДЛЕЖНОСТЬ_ПЛАТЕ конкретной компоненты.

Настройку проекта цифрового устройства на конкретную элементную базу отражают фреймы-экземпляры. Они соответствуют экземплярам компонент каждого типа.

Описанная фреймовая структура составляет первый уровень знаний о тестируемой цифровой схеме.

Второй уровень иерархии знаний образует информация о связи между компонентами в схеме. Подобным образом в системе VMES [5] организованы знания в семантической сети SNPrs. В описываемых инструментальных средствах вершины семантической сети представляют такие ОБЪЕКТЫ предметной области, как «выводы компонент» тестируемой схемы и «память для хранения значений» на этих выводах, а также СИТУАЦИЮ «соединение». Дуги семантической сети отражают следующие отношения между вершинами-объектами и вершиной-ситуацией: ИСТОЧНИК для вершины типа «выход компоненты»; ПРИЕМНИК для вершины типа «вход компоненты»; ХРАНЕНИЕ_ЗНАЧЕНИЯ для вершины типа «память значений».

Названные уровни знаний для генератора тестов аппаратуры первичны, на них базируется модель правильно работающей схемы проектируемого устройства. Для разработки проверяющих тестов этих знаний достаточно.

Диагностическое тестовое обеспечение разрабатывается различными способами: алгоритмическим (от D- и РОДЕМ- до FAN-алгоритма) и путем моделирования неисправностей [9]. Среди методов моделирования наименее трудоемкий — конкурентное моделирование [10]. Кроме того, применение названного метода позволяет автоматизировать процесс получения тестов с помощью ЭВМ на основе имитатора правильно работающей схемы. Основа конкурентного метода моделирования — список неисправностей, обнаруживаемых данным входным набором и транспортируемых к выходам схемы. Знания, составляющие правила транспортирования неисправностей через компоненту конкретного типа, могут быть получены из первичных знаний о структурном и поведенческом аспектах компоненты этого типа.

В инструментальных средствах автоматизации разработки тестов первичные знания о функционировании компонент (слот ПОВЕДЕНИЕ) имеют процедурное представление. Аргументами этих процедур являются вершины-объекты типа «память значений», обслуживающие прием-передачу сообщений и реакцию на него. Конструктор вторичных знаний создает для каждого типа компонент элементной базы присоединенную процедуру транспортируемых неисправностей — значение слота МЕТОД_ТЕСТИРОВАНИЯ соответствующего фрейма-шаблона.

Процедура транспортирования неисправностей описывает алгоритм конкурентного моделирования одиночных константных неисправностей для подсхемы компоненты. k -константная неисправность на линии a обозначается a/k . В инструментальных средствах линиям сопоставлены отношения ИСТОЧНИК (ПРИЕМНИК) семантической сети, поэтому обозначение линии записывается как

ИДЕНТИФИКАТОР компоненты . НОМЕР вывода

и извлекается из соответствующих слотов. Процедура МЕТОД_ТЕСТИРОВАНИЯ имеет следующий список параметров: идентификатор компоненты; номер подсхемы; множество элементов памяти, содержащих значения на выводах; указатель списка транспортируемых неисправностей.

Алгоритм конструирования процедуры транспортируемых неисправностей следующий. Для каждой подсхемы формируются извлечение значения слота НОМЕР для всех выводов данной подсхемы $Nx_k, Nx_{k+1}, \dots, Nx_i, \dots, Ny_j$, для каждого выхода u_j ; 1) определение значения V_{u_j} на выходе для правильно работающей подсхемы с помощью присоединенной процедуры

ПОВЕДЕНИЕ, 2) последовательное инвертирование значения V_{x_j} каждого из входов x_j этой подсхемы с восстановлением исходного значения на остальных; входах и вычисление значения на выходе, 3) определение транспортируемых неисправностей.

Неисправностями, транспортируемыми через данную подсхему, являются те, которые устанавливаются на выходе y_j значение, отличное от V_{y_j} . Всегда через подсхему транспортируется неисправность на линии, соответствующей рассматриваемому выходу, со значением, инверсным истинному:

ИДЕНТИФИКАТОР компоненты. N_{y_j} / V_{y_j}

Список транспортируемых неисправностей данной подсхемы конкатенируется со списком неисправностей, обнаруживаемых на линиях, соответствующих входам x_i .

Так, используя знания о структуре компоненты конкретного типа и библиотеку процедур функционирования компонент элементной базы, конструктор вторичных знаний создает библиотеку процедур для транспортируемых неисправностей. Одна из возможностей оптимизации метода конкурентного моделирования заключается в сочетании базы вторичных знаний с эвристиками, сокращающими множество перебора комбинаций входных и выходных сигналов. Указанные эвристики, в свою очередь, могут расширить базу первичных знаний.

В качестве языка моделирования цифровых устройств в инструментальных средствах разработки тестов предлагается язык СИ, обладающий лингвистическими средствами работы с битовыми данными, а также имеющий достаточно эффективные компиляторы. Этот язык выбран также и для реализации собственно инструментальных средств и конструктора знаний. Такой выбор определяется не только распространенностью и мобильностью языка, но и опытом его успешного применения как в САПР БИС [11], так и в экспертных системах [8].

Работа выполнена в Донецком политехническом институте.

Список литературы

1. Moskowitz L. Knowledge-based circuit design // Proc. Auto-testcon' 85. 1985. P. 69—74.
2. Subrahmanyam P. A. Synapse: an expert system for VLSI design // Computer. 1986. Vol. 19. № 7. P. 78—89.
3. Построение экспертных систем / Под ред. Ф. Хейеса, Рота, Д. Уотермана, Д. Лената: Пер. с англ. Ю. И. Крюкова, Н. Д. Смольянинова, С. Б. Трубициной / Под ред. В. Л. Стефанюка. М.: Мир, 1987. 441 с.
4. Ramsey I. D. Diagnosis: using automatic test equipment and artificial intelligence expert system // Proc. NAECON, 1985. Vol. 2. P. 1369—1374.
5. Tail M. R., Geller /., Srihari S. N., Shapiro S. C. Knowledge based modeling of circuit boards // Proc. Annu. Rel. and Maintainab. Symp. 1987. P. 422—427.
6. Varma P., Tohma Y. PROTEAN a knowledge based test generator // Proc. IEEE Custom. Integr. Circuits Conf. 1987. P. 78—81.
7. Gaines B. R. Second generation knowledge acquisition systems // GMD-Stud. 1988. N2 143. P. 17/1 — 17/14.
8. Искусственный интеллект: применение в химии / Под ред. Т. Пирса, Б. Хони: Пер. с англ. А. Ю. Батыря, В. Н. Петрова / Под ред. В. Л. Стефанюка. М.: Мир, 1988. 430 с.
9. Киносита К., Асада К., Карацу О. Логическое проектирование СБИС: Пер. с японск. Д. А. Ковтунова, Л. В. Поспелова / Под ред. Л. В. Поспелова. М.: Мир, 1988. 308 с.
10. Rogers W. L., Guzolek I. K., Abraham I. A. Concurrent hierarchical fault simulation: a performance model and two optimizations // IEEE Trans. Comput. — Aid. Des. Integr. Circuits and Systems. 1987. Vol. 6. N:5. P. 848—862.
11. Feldman S. I. The circuit design language Xi // Proc. ICCD' 83. 1983. P. 652—655.