

3. Горбунов В. П., Горбунов В. М., Завьялкин Ф. М., Квасница М. С. Влияние усреднения измеряемой характеристики изделия в поле зрения детектора на выбор радиометрического устройства // Дефектоскопия.— 1976.— № 2.
4. Вычислительная оптика: Справочник/Под ред. М. М. Русипова, А. И. Грамматина, П. Д. Иванова и др.— Л.: Машиностроение, 1984.
5. CT/T Technology Continuum // General Electric, Co.— USA, 1978.
6. Informationen zur Computer — Tomography // Meseigenschaften des SOMATOM2.— BRD: Siemens AU, 1979.
7. Harding G. Dose rate control tomography—a study of image quality from a transfer function standpoint // Radiologia Diagnostica.— 1979.— V. 4, N 20.— P. 581—586.
8. McCullough E. C., Payne I. T. X-ray-transmission computed tomography // Medical Physics.— 1977.— V. 4, N 2.— P. 85—98.
9. Reimers P., Goebels I. New possibilities of nondestructive evaluation by X-ray computed tomography // Mater. Evaluation.— 1983.— V. 41.— P. 732—737.

*Поступила в редакцию 22 апреля 1985 г.*

УДК 681.3.06 : 800.02

**С. О. СТЕПАНИН**

*(Ереван)*

### **ЯЗЫК АЛГОРИТМИЧЕСКОГО ЗАДАНИЯ ТЕСТОВ (АЗАТ)**

В последние годы для тестового контроля цифровых схем все чаще применяются функциональные тесты (ФТ). Эта тенденция объясняется в основном следующими причинами: во-первых, традиционные автоматизированные системы генерации тестов плохо справляются с возросшей сложностью современных цифровых схем [1, 2]; во-вторых, возможности генерации теста по структурному описанию схемы теоретически ограничены [3, 4]; в-третьих, с повышением быстродействия элементного базиса возрастает роль и значение динамического контроля, для которого пригодны только ФТ [2, 5], и, наконец, в-четвертых, функциональное описание схемы, как правило, всегда проще, понятнее и доступнее, чем описание ее структуры [2, 6, 7].

Для упрощения задачи составления ФТ разработано множество языков описания тестов (ЯОТ) [8—18], распространению которых способствовала, в частности, возможность обеспечения программной поддержки ЯОТ непосредственно в исполнительных контрольно-диагностических системах (КДС), построенных на базе мини- и микроЭВМ [7]. Специфика ЯОТ усматривается, прежде всего, в эффективном решении таких вопросов, как обработка информационных объектов, не обязательно кратных по длине формату машинного слова базовой ЭВМ [19], и совмещение механизмов формирования тестовых последовательностей во времени (на одном контакте для многих тактов) и в пространстве (на многих контактах для одного такта). Применение с этой целью конструкций универсальных языков программирования (например, Паскаля [16, 17]) обычно приводит к громоздкой и неаглядной форме необходимого описания ФТ. В этом смысле предлагаемый язык алгоритмического задания тестов (АЗАТ), разработанный для КДС «Севан-2», обладает рядом отличительных особенностей, раскрываемых ниже.

Прежде всего язык АЗАТ ориентируется на контролируемую схему, а не КДС. С этой целью из номенклатуры команд, унифицированной для ЯОТ [18, 19], исключены команды непосредственного управления аппаратурой КДС и периферийными установками ввода — вывода. В итоге исходная программа (ИП), написанная на языке АЗАТ, задает только порядок формирования стимулов и ожидаемых реакций контролируемой схемы и ничего более этого. Текст ИП складывается из предложений, каждое из которых содержит базовую команду языка или выражение. Предусмотрены всего семь различных базовых команд, часть которых выполняет стандартные функции управления последователь-

ностью действий. Специфические команды ТАКТ и ЖДАТЬ семантически определяются как команды синхронизации, чтобы не связываться с техническими и функциональными особенностями исполнительной аппаратуры. В КДС «Севап-2» команды ТАКТ воспринимаются как сигналы о завершении формирования отдельных тест-наборов и возбуждают обращение к исполнительным средствам КДС как к подпрограмме. Команда ЖДАТЬ <предел> выполняется аналогично, но с программируемым допустимым временем ожидания правильной реакции, поэтому она используется при контроле схем в асинхронном режиме. Множество базовых команд языка АЗАТ может произвольным образом паразитировать пользователем. Вновь вводимые команды среди подпрограмм, но вызываются по простому упоминанию в тексте ИП. Выражения в языке АЗАТ представляют единственную допустимую форму обработки данных. Операции в выражениях не имеют приоритетности и выполняются слева направо в заданном порядке следования.

ИП на языке АЗАТ составляется в виде линейного списка независимых модулей с унифицированной структурой. Первый модуль ИП называется тест-примером и содержит описание конкретного, завершеного ФТ данной схемы. Последующие модули ИП содержат описание дополнительных команд пользователя, если таковые применяются в тест-примере. Каждый модуль состоит из четырех обязательных частей (заголовка, декларативной зоны, процедурной зоны, хвоста), которые выделяются специальными ключами (ТЕСТ-ПРИМЕР или КОМАНДА, ПОЛАГАТЬ, НАЧАЛО, КОНЕЦ). В заголовке модуля тест-примера указываются четыре параметра (номиналы частоты контроля и задержки считывания реакции, тип и число внешних разъемов), характеризующие схему как объект контроля, а в заголовке модуля команды пользователя — формальные параметры. Передача параметров при обращении к команде пользователя производится по принципу позиционного соответствия и по ссылке [20].

В декларативной зоне модуля заранее описываются переменные, используемые в процедурной зоне. Всего предусмотрено четыре типа переменных: «слово», «контакт», «сигнал» и «указатель»; их значения всегда трактуются как двоичный код целого числа. Формат всех типов переменных может объявляться произвольным в пределах от 1 до 256 бит, по умолчанию устанавливается формат машинного слова (16 бит).

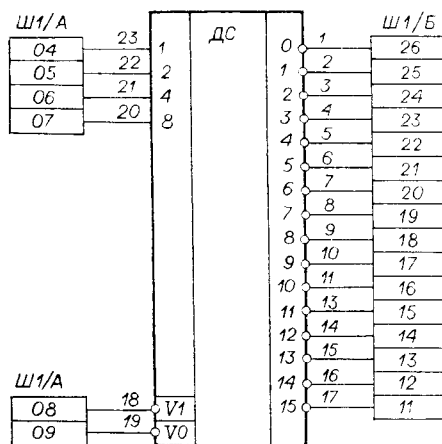
Переменные типа «слово» вводятся исключительно для программистских целей (хранения промежуточных результатов, создания счетчиков циклов и т. п.), типа «сигнал» — для описания временных диаграмм логических сигналов «0» и «1», которые предполагается потактно подавать на определенные контакты схемы или считывать с них, а типа «контакт» — для привязки тест-примера к конкретной схеме. Привязка осуществляется традиционным способом [12], однако контакты внешнего разъема схемы указываются в привычных конструкторских, а не в условных обозначениях. Расшифровка этих обозначений производится на основе данных о типе и числе разъемов схемы, приведенных в заголовке тест-примера. В отличие от других типов переменная типа «контакт» обладает не только значением, но и двумя различными состояниями «вход — выход» и «наблюдаем — ненаблюдаем», которые управляются путем присвоения переменной специальных констант ВХОД или ВЫХОД и ОТКРЫТО или ЗАКРЫТО соответственно. Для КДС эти присвоения означают переключение соответствующих контролируемых каналов на передачу — прием и разрешение — запрет анализа правильности реальных сигналов схемы. Переменная типа «указатель» является универсальной в том смысле, что по команде <указатель> НА <переменная> может замещать любую переменную, присваивая себе ее тип, формат и значение. Числовые константы записываются на языке АЗАТ непосредственно. Использование указателей и явных констант представляется более наглядным и удобным, чем прямое определение

в языке скрытых в них механизмов косвенной и непосредственной адресации. В частности, формальные параметры команд пользователя допускаются только типа «указатель», что избавляет от необходимости переключения сред параметров и копирования фактических значений в момент обращения к этим командам.

Еще одна особенность языка АЗАТ связана с наличием команды сокрытия ИСПОЛЬЗОВАТЬ, предназначенной для динамического расширения семантики команд синхронизации. Идея заключается в том, чтобы исключить необходимость явной записи повторяющихся действий, выполнение которых обязательно при формировании каждого такта ФТ. Если, например, диаграмма переменной К типа «контакт» предопределена значением переменной С типа «сигнал» или типовой процедурой П(X), оформленной как команда пользователя, то достаточно заранее объявить об этом посредством предложений ИСПОЛЬЗОВАТЬ С(К) или ИСПОЛЬЗОВАТЬ П(К). В этом случае после каждого выполнения команды ТАКТ или ЖДАТЬ переменной К будет автоматически присваиваться очередное тактовое значение сигнала, взвешенное из переменной С или вычисленное путем обращения к П(X).

Отмеченные особенности языка АЗАТ иллюстрируются простым примером ФТ для контроля микросхемы типа 13ЗИДЗ (рисунок). ФТ составлен с учетом того обстоятельства, что при переборе кодовых комбинаций на входах исправной схемы с соблюдением условия  $V0 \& V1 = 1$  на ее выходах формируется «бегущий ноль», а при  $V0 \& V1 = 0$  наблюдаются сплошные единицы.

Программная поддержка языка



ФТ контроля микросхемы 13ЗИДЗ:

«=» — знак присвоения, «<» — условие «не равно», «>» — операция циклического сдвига влево, «Ш» — префикс двоично-шестнадцатеричной константы, «\*» — скобки комментариев

\*ТЕСТ ФУНКЦИОНАЛЬНОГО КОНТРОЛЯ МИКРОСХЕМЫ 13ЗИДЗ\*

ФТ13ЗИДЗ: ТЕСТ-ПРИМЕР (4 МГц, 50 нс, ГРПМ90, 1);

ПОЛАГАТЬ: АДРЕС: КОНТАКТ = Ш1/А,4/7;

ОТВЕТ: КОНТАКТ = Ш1/Б,11/26;

VV: КОНТАКТ = Ш1/А,8,9;

НАЧАЛО: АДРЕС, VV = ВХОД,ЗАКРЫТО; \*НАЧАЛЬНЫЕ УСТАНОВКИ\*

ОТВЕТ = ВЫХОД,ОТКРЫТО;

АДРЕС, VV = 0; ОТВЕТ = .ШFFFFE;

ИСПОЛЬЗОВАТЬ СЧЕТ(АДРЕС), БЕГ(ОТВЕТ);

ЦИКЛ: \*ИСЧЕРПЫВАЮЩЕГО ПЕРЕБОРА ВХОДНЫХ ЗНАЧЕНИЙ\*

ТАКТ; ПРИ АДРЕС < 0 ИДТИ ЦИКЛ;

VV = 1; ИСПОЛЬЗОВАТЬ СЧЕТ ( VV);

ОТВЕТ = .ШFFFFF; \*ОЖИДАЕМАЯ РЕАКЦИЯ ПРИ VV < 0\*

ТАКТ 3 \* РАЗА \* ;

СТОП;

КОНЕЦ ФТ13ЗИДЗ;

\*КОМАНДА ИСЧЕРПЫВАЮЩЕГО ПЕРЕБОРА КОДОВ\*

СЧЕТ: КОМАНДА (X);

ПОЛАГАТЬ: X: УКАЗАТЕЛЬ;

НАЧАЛО: X = X + 1; СТОП;

КОНЕЦ СЧЕТ;

\*КОМАНДА ФОРМИРОВАНИЯ КОДОВ 'БЕГУЩИЙ 0' И 'БЕГУЩАЯ 1'\*

БЕГ: КОМАНДА (Y) ;

ПОЛАГАТЬ: Y : УКАЗАТЕЛЬ;

НАЧАЛО: Y = Y << 1; СТОП;

КОНЕЦ БЕГ;

АЗАТ содержит три компонента: компилятор, интерпретатор и посткомпилятор. Компилятор переводит ИИ на язык промежуточного уровня ЯИУ — АЗАТ, образуя интерпретируемую объектную программу (ИОИ). Интерпретатор обеспечивает пробный прогон ИОИ на программно моделируемом 256-разрядном процессоре ЯИУ — АЗАТ в стартовом и пошаговом режимах и дает возможность выявлять скрытые смысловые ошибки, недоступные для обнаружения на этапе компиляции. Посткомпилятор преобразует ИОИ в выполняемую объектную программу, которая непосредственно воспринимается исполняемыми средствами целевой КДС.

Крайняя малочисленность элементов программы и типов данных обуславливают простоту и легкость усвоения языка АЗАТ. Одновременно язык обладает достаточно широкими функциональными возможностями и естественным образом поддерживает модульный принцип разработки ФТ. Отсутствие команд для непосредственного управления аппаратурой КДС и принятая схема двухступенчатой компиляции обеспечивают портативность языка. Применение команд сокрытия повторяющихся действий и предусмотренная возможность произвольного наращивания переходного множества базовых команд сообщают гибкость языку и способствуют получению компактных и обзорных ИИ.

Описанная версия языка АЗАТ реализована в автоматизированной КДС «Севан-2», предназначенной для контроля и диагностики цифровых схем четвертого поколения. Этапы составления, компиляции, отладки и эксплуатации ФТ, написанных на языке АЗАТ, замкнуты в единый технологический цикл. Общий объем средств программной поддержки языка составил примерно 35 Кбайт.

#### ЛИТЕРАТУРА

1. Степанян С. О. Обеспечение тестируемости цифровых схем // Измерения, контроль, автоматизация: Научн.-техн. сб. обзоров/ЦНИИТЭИ приборостроения.— М., 1985.— Вып. 1.
2. Степанян С. О. Тенденции развития автоматизированных систем генерации тестов для цифровых схем.— Ереван, 1983.— Рукопись деп. в ВИНТИ, № 5616-83.
3. Fujiwara H., Toida S. The complexity of fault detection problems for combinational logic circuits // IEEE Trans. on Computers.— 1982.— V. C-31, N 6.— P. 555—560.
4. Miszo A. The sequential ATPG: a theoretical limit // Int. Test. Conf.— N. Y.: Cherry Hill, 1983.
5. Bradley R. S. A three-stage approach to LSI board testing // Electr. Engineering.— 1981.— V. 53, N 651.— P. 83—91.
6. Thatte S. M., Abraham J. A. Test generation for microprocessors // IEEE Trans. on Computers.— 1980.— V. C-29, N 6.— P. 429—441.
7. Степанян С. О., Папян Г. Г. Методы и средства автоматизированного контроля и диагностики современных цифровых схем // Зарубеж. радиоэлектрон.— 1983.— № 11.
8. Белов В. М., Буровцев В. А., Ибрагимов К. Ш., Подзин А. Е. Специализированный входной язык системы контроля БИС ЗУ // Автометрия.— 1975.— № 4.
9. Подзин А. Е., Ибрагимов К. Ш., Корня И. Х., Кирпленко Б. З. Проблемно ориентированный язык ТЕСТ для автоматизированных систем контроля электронной аппаратуры // Упр. сист. и ман.— 1979.— № 4.
10. Глух Р. Е., Клецов-Надеев А. А., Кузнецов С. В. Специализированный язык контроля цифровых устройств // Обмен опытом в радиопром-сти.— 1981.— Вып. 12.
11. Борисова Я. И., Липатова Л. В., Лыщенко В. И., Трускова С. М. Язык ЭЛЕГОН Ф для описания процесса контроля БИС // Электрон. пром-сть.— 1984.— Вып. 7.
12. Соколов В. А., Орешкин М. И. Использование языка системы тестового контроля при проверке цифровых узлов // Обмен опытом в радиопром-сти.— 1983.— Вып. 4.
13. Heiser J. E. On-line incremental ATLAS processing // Autotestcon'78.— San Diego, Ca., 1978.
14. Wickham G. S. PLT: a test language for new products // Eur. Conf. on Electr. Design Automation.— Brighton, La., 1981.
15. Stevens A. K. Structured programming and the IC test engineer // Int. Test Conf.— Silver Spring, Md., 1982.

16. Mahoney R. C. A common PASCAL test language: reality or pipedream // Ibid.
17. Downey A. L. Test program optimization technique for a high-speed performance VLSI tester // Int. Test Conf.—N. Y.: Cherry Hill, 1983.
18. Okamoto T., Shibata H., Kinoshita K. Design of high level test language for digital LSI // Ibid.
19. Подзин А. Е. Организация управления аппаратурными средствами автоматизированных систем контроля // Автометрия.— 1978.— № 4.
20. Пратт Т. Языки программирования: разработка и реализация.— М.: Мир, 1979.

Поступила в редакцию 11 сентября 1985 г.

УДК 62.595 : 519.24

К. В. ПСАЕВ  
(Ростов-на-Дону)

### ОБ ОДНОМ МЕТОДЕ РЕГУЛЯРИЗАЦИИ ЗАДАЧ ОБРАБОТКИ НАБЛЮДЕНИЙ

**Введение.** Для задач обработки наблюдений (экспериментальных данных) характерна следующая постановка.

Пусть векторные случайные величины (наблюдения)  $X_1, X_2, \dots, X_N$  связаны с оцениваемым векторным параметром  $c$  соотношениями

$$X_i = f_i(c) + Y_i, \quad i = 1, 2, \dots, N, \quad (1)$$

где  $f_i(c)$  — заданные вектор-функции;  $Y_1, Y_2, \dots, Y_N$  — последовательность независимых в совокупности одинаково распределенных случайных векторов с плотностью распределения вероятностей (ПРВ)  $\varphi(y|b)$ , определенной с точностью до векторного параметра  $b$ . Требуется по реализации  $x^{(N)} \equiv \{x_1, x_2, \dots, x_N\}$  последовательности наблюдений  $X^{(N)} \equiv \{X_1, X_2, \dots, X_N\}$  найти оценку  $c^*$  параметра  $c$ .

Математически корректные методы решения задач оценивания параметра  $c$  можно связать с байесовским подходом [1, 2], при котором параметры  $c$  и  $b$  интерпретируются как случайные векторы  $C$  и  $B$  с совместной априорной ПРВ  $p_0(c, b|\alpha)$ , зависящей от некоторого (в общем случае векторного) параметра  $\alpha$ , смысл которого будет уточнен ниже. В соответствии с формулой Байеса совместная условная (апостериорная) ПРВ параметров  $C$  и  $B$  относительно реализации наблюдений  $x^{(N)}$  определяется формулой

$$p(c, b|x^{(N)}, \alpha) = k(x^{(N)}, \alpha) p_0(c, b|\alpha) \varphi^{(N)}(x^{(N)}|c, b), \quad (2)$$

где

$$\varphi^{(N)}(x^{(N)}|c, b) = \prod_{i=1}^N \varphi(x_i - f_i(c)|b) \quad (3)$$

— условная (относительно  $c$  и  $b$ ) ПРВ последовательности  $X^{(N)}$ ;  $k(x^{(N)}, \alpha)$  — нормирующий множитель, совпадающий с величиной, обратной значению безусловной ПРВ реализации  $x^{(N)}$ . Наиболее общий из методов оценивания — метод максимума апостериорной вероятности (МАН) [3] — сводится к максимизации плотности (2) или, что то же самое, логарифма этой плотности по паре  $(c, b)$ . С учетом соотношения (3) полученная (зависящая от  $\alpha$ ) МАН-оценка этой пары имеет вид

$$(c_{\text{МАН}}(\alpha), b_{\text{МАН}}(\alpha)) = \arg \max_{(c,b)} \left[ \ln p_0(c, b|\alpha) + \sum_{i=1}^N \ln \varphi(x_i - f_i(c)|b) \right]. \quad (4)$$

Метод МАН тесно связан с методами регуляризации [4] задач обработки (аппроксимации) наблюдений (первое слагаемое в правой части формулы (4) можно рассматривать как стабилизирующую добавку с параметром регуляризации  $\alpha$ ). Вопросы выбора этой добавки и опре-