

## АЛГОРИТМЫ ДЕТЕРМИНИРОВАННОГО СИНТЕЗА ТЕСТОВ ЦИФРОВЫХ ВЫЧИСЛИТЕЛЬНЫХ УСТРОЙСТВ

*Шаповалов А.И., Зинченко Ю.Е.*

*Донецкий национальный технический университет, г. Донецк*

Проблема диагностики цифровых вычислительных устройств (далее ЦВУ) всегда была актуальной. Диагностика позволяет выявлять неисправные устройства, как на этапе производства, так и на этапе функционирования. Современные ЦВУ могут содержать миллионы логических вентилях и линий соединения. Поскольку размер пространства поиска, внутри которого производится поиск тестовой последовательности векторов, растет экспоненциально от числа входов ЦВУ и собственных состояний схемы, задача генерации тестов отнесена к классу NP-полных. Для тестирования последовательных схем используются автоматические генераторы тестов.

Общее решение задачи генерации тестов для последовательных схем впервые было предложено и сформулировано Ротом [1], который назвал свой метод - D-алгоритмом. Метод основан на идее многомерной активизации пути. В его основе лежат понятия логических кубов, что делает его удобным для программной реализации. D-алгоритм позволяет строить детерминированные тесты для любых видов константных неисправностей, но он характеризуется большим числом перебором. Число переборов экспоненциально зависит, как от числа входов, так и от числа ветвей схемы.

Дальнейшим развитием идеи D-алгоритма явился алгоритм PODEM (Goel, 1981).[2] Число переборов в нем остается велико и экспоненциально зависит от числа входов схемы. Рассмотрим основную идею данного алгоритма.

Пусть целью процедуры является установка значения  $V$  на некоторой линии. Тогда значение внешнего входа определяется четностью числа элементов инвертирующего типа вдоль выбранного пути. В случае прохождения элемента со многими входами выбирается один из них. При этом используются следующие правила выбора. Прежде всего, определяется, имеется ли один или несколько вариантов присвоения значений входов вентиля, при которых на его выходе устанавливаются требуемое значение. В случае нескольких вариантов выбирается вход, ближайший к внешнему входу или имеющий лучший показатель управляемости, т.е. наиболее перспективный путь. Если имеется только один вариант, то выбирается вход, наиболее далекий от внешних входов или имеющий худший показатель управляемости, а остальные входы заносятся в стек этой процедуры. Это делается с целью скорейшего обнаружения противоречивых ситуаций. Блок-схема этой процедуры представлена на рисунке 1. После того как внешний вход в результате этой процедуры получил определенное значение, выполняется импликация. Если в результате импликации цель достигнута (на "целевой" линии устанавливается требуемое значение  $D$  или  $\bar{D}$ ), то выполняется следующий шаг  $D$  распространения. Иначе выполняется возврат назад.

В методе PODEM при построении теста по аналогии с задачами целочисленного программирования используется метод ветвей и границ. Структура дерева решений указывает на тот факт, что процесс поиска по алгоритму PODEM основан на прямом неявном переборе всех входных векторов схемы. Как в D-алгоритме, поиск заканчивается, если возникла ситуация ошибки, или в ситуации, когда для неисправности тест не существует.

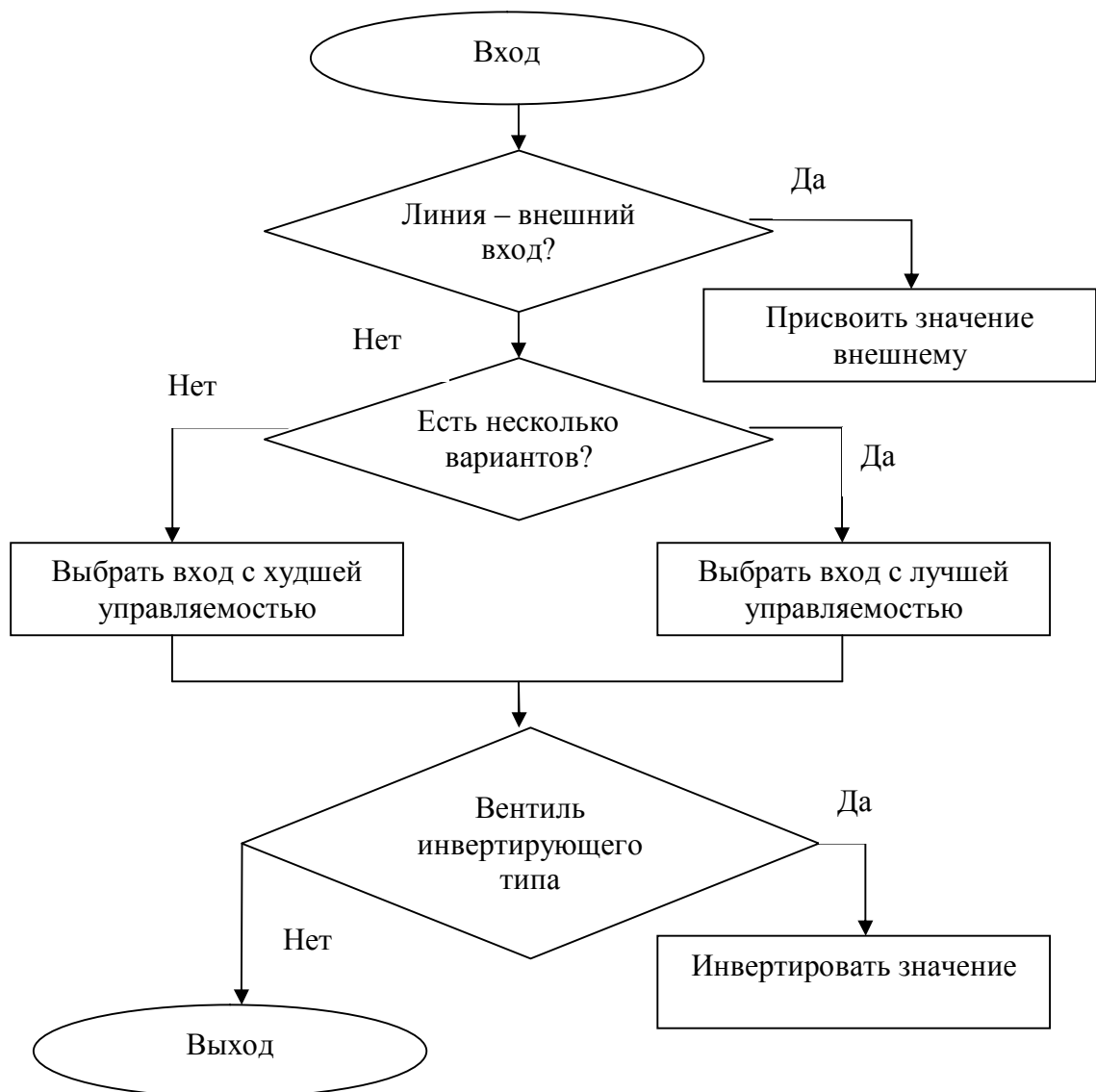


Рисунок 1 – блок-схема алгоритма PODEM

Для минимизации числа переборov в дереве решений алгоритма PODEM был разработан алгоритм FAN (Fujiwara, Shimono, 1983)[2]. В алгоритме FAN на этапе доопределения алгоритм может остановиться на внутренних ветвях, а не на входах (как в алгоритме PODEM). К тому же в алгоритме FAN используется процедура с множественным возвратом, которая пытается удовлетворять набор функций.

На базе алгоритма FAN разработано много моделей и усовершенствований, в частности Kirkland и Mercer (1987 г.); Cheng (1988 г.); Убар (1988 г.); Schulz и Auth (1989.); Cheng и Davidson (1989 г.); Lee и Ha (1993 г.); Kelsey и Saluja (1993 г.); Teramoto (1993г.); Silva и Sakallah (1994 г.); Ривин И., Chakradhar S.T. (1994 г.); Евтушенко, Лебедев и Петренко (1994 г.); Gläser и Vierhaus (1995 г.); Гессель и Согомонян (1996 г.); Куфарева (2000 г.).

### Литература

- [1] Y. Chang, Fault diagnosis of digital systems, Wiley-interscience, New-York, 1970
- [2] M. Abramovici, Digital systems testing and testable design, Wiley-interscience, 1998