Хмельницький: ХНУ, 2011 - 206 с.

11. ISO/IEC 25010:2011. Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- System and software quality models

12. В.А.Благодатских, В.А.Волнин, К.Ф.Поскакалов. Стандартизация разработки программных средств: Учебное пособие. - М.: Финансы и статистика, 2005. - 288 с.

13. Software Requirement Specifications [Full Version] // [Electronic resource] - Access mode: http://findebookee.com/s/software-requirement-specifications

References

1. Mishchenko V.O., Pomorova O.V., Hovorushchenko T.A. CASE-otsenka kriticheskih programmnih sistem. Tom 1. Kachestvo / Pod red. Kharchenko V.S. - Kharkov: NAU "KhAI", 2012 - 201 s. [in Russian]

2. IEEE 830-1998. Recommended Practice for Software Requirements Specifications - New York: IEEE, 1998

3. R.T.Futrell, D.F.Shafer, L.I.Shafer. Quality Software Project Management - New York: Prentice Hall PTR, 2003.

4. Steve McConell. Code Complete - New York: Microsoft Press, 2013

5. Pomorova O.V., Hovorushchenko T.O. Suchasni problemi otsinuvannya yakosti programnogo zabezpechennya // Radioelektronni i komp'uterni sistemi - Kharkiv: NAU "KhAI", 2013 - № 5, c.319-327 [in Ukrainian]

6. IEEE 1031-2011: IEEE Guide for the Functional Specification // [Electronic resource] - Access mode: https://standards.ieee.org/findstds/standard/1031-2011.html

7. Boehm B.W. Software Engineering; R&D trends and defense needs. – In: Research. Directions in Software Technology (P.Wegner, ed.). – Cambridge, MA: MIT Press, 1979. – 543 pp.

8. Skripkin K.G. Ekonomicheskaya effektivnost informatsionnih sistem - M.: Radio i svyaz, 2003 - 156 s.

9. Kovalevskaya E.V. Materiali k kursu "Metrologiya, kachestvo i sertifikatsiya PO" - M: MGUESI, 2002 - 38 s. // [Electronic resource] - Access mode: http://bookinist.net/books/bookid-333504.html [in Russian]

10. Pomorova O.V., Hovorushchenko T.O. Proektuvannya interfeisiv koristuvacha: Navchalniy posibnik - Khmelnitskiy,: KhNU, 2011 - 206 s. [in Ukrainian]

11. ISO/IEC 25010:2011. Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- System and software quality models

12. V.A.Blagodatskih, V.A.Volnin, K.F.Poskakalov. Standartizatsiya razrabotki programmnih sredstv: Uchebnoe posobie - M.: Finansi i statistika, 2005 - 288 s. [in Russian]

13. Software Requirement Specifications [Full Version] // [Electronic resource] - Access mode: http://findebookee.com/s/software-requirement-specifications

Рецензія/Peer review : 18.10.2013 р. Надрукована/Printed :24.11.2013 р.
Рецензент: Шалапко Ю. І., д.т.н., професор кафедри ІМ, ХНУ,

УДК 004.3:681.518

Д.Е. ИВАНОВ
Институт прикладной математики и механики НАН Украины, г.Донецк

МЕТОД ПОСТРОЕНИЯ ТЕСТОВ ЦИФРОВЫХ УСТРОЙСТВ С ПОДТВЕРЖДЕНИЕМ СОСТОЯНИЙ НА ОСНОВЕ ГЕНЕТИЧЕСКОГО АЛГОРИТМА

В статье описывается метод построения тестов для цифровых последовательностных устройств на логическом уровне описания. Он основан на двухуровневом подходе, а особенностью является реализация нижнего уровня метода. Целью метода данного уровня является построение теста одной выбранной неисправности, а реализация состоит из двух фаз. Первая фаза выполняет структурное распространение влияния неисправности внутри одного такта модельного времени, при этом формируются ограничения на исходное состояние устройства. Подтверждение данного состояния выполняется с помощью разработанного генетического алгоритма. Применение комбинированного подхода построения тестов должно увеличить эффективность такой процедуры для сложных современных СБИС.

Ключевые слова: цифровое устройство, генерация тестов, генетический алгоритм, достижение состояния.

D.E. IVANOV
Institute of Applied Mathematics and Mechanics, National Academy of Sciences of Ukraine, Donetsk

TEST GENERATION METHOD FOR DIGITAL DEVICES WITH STATE SUBSTANTIATION BASING ON GENETIC ALGORITHM

Abstract – The aim of this paper is developing a test generation method for sequential digital devices at the logical level of description. Its main feature is the use of two phases for the construction of test of one selected fault.

The first phase of method performs structural fault propagation in one clock cycle of model time, which well developed for this time. At end of this phase are formed the restrictions to the initial state of the fault and fault-free devices. Substantiation of this state is performed by using the new developed genetic algorithm.

The use of such combined approach of test generation increases the effectiveness of this procedure for modern complex VLSI, which contains both data processing parts and controllers.

Keywords: digital device, test generation, genetic algorithm, state substantiation.

Введение

Согласно Международной Дорожной Карте [1] число транзисторов в SOC-системах потребительского уровня увеличится в 17 раз к 2024 году. С другой стороны, длина входных последовательностей для тестирования

одиночных константных неисправностей (ОКН) увеличится с нынешних 30 тысяч входных векторов в 16 раз к указанному времени. Видно, что задача построения тестов с высокой полнотой остаётся одной из центральных в технической диагностике цифровых устройств (ЦУ), что связано как с ростом размерности проектируемых СБИС, так и с увеличением их сложности.

При построении тестов современных проектируемых ЦУ часто используется двухуровневая модель [2], суть которой заключается в итеративном построении тестовой последовательности для каждой из неисправности в списке. При это метод заканчивает работу при выполнении некоторого критерия: достигнута необходимая полнота теста, исчерпан лимит времени и т.п.

В детерминированных алгоритмах построения тестов [3, 4] каждая неисправность первоначально должна быть активизирована. Активизацией неисправности называется случай, когда исправное и неисправное ЦУ находятся в различных состояниях. Затем влияние неисправности необходимо транспортировать на внешние выходы анализируемого ЦУ. При этом в последовательностных ЦУ это может потребовать от устройства находиться в некотором определённом состоянии для того, чтобы такая активация/распространение были возможны. Это требует разработки процедур обратного распространения, которые усложняются двумя факторами:

- обратное распространение может потребоваться для нескольких тактов модельного времени;
- для многих логических элементов возникает проблема неоднозначности значений входов, заставляющая рассматривать множество вариантов.

Дополнительные ограничения могут возникать из технологических особенностей: частичное или полное использование сканирующих регистров для установки начального состояния, предопределённость начального состояния, ограничение на число тактов для достижения состояния и т.д.

В противоположность обратному распространению в подходах идентификации, основанных на моделировании [5], влияние неисправности всегда распространяется только в прямом направлении. Сюда же относятся и генетические алгоритмы (ГА) построения тестов [6, 2]. При разработке таких методов обычно применяется следующий подход. Активизация неисправности происходит с помощью генерации псевдослучайных последовательностей, а её распространение с помощью генетического алгоритма. При рассмотрении двухуровневой схемы генерации тестов на нижнем уровне, фактически, реализуется ГА-метод верификации эквивалентности двух заданных ЦУ: исправного и неисправного. Один из вариантов такого ГА описан, например, в [7].

Детерминированные алгоритмы используются в схемах-контроллерах, а подходы основанные на моделировании – для устройств обработки данных. Развитие сложных комплексных СБИС потребовало объединения этих двух направлений. Например, в [8] метод начинает работу с ГА. Если же в течении определённого времени не происходит улучшения ситуации, то алгоритм переключается на детерминированный метод построения тестов.

Другим способом объединения двух указанных подходов является подтверждение состояний ЦУ при активизации неисправностей в детерминированных методах, которое реализуется, в свою очередь, основанным на моделировании алгоритмом [9]. Таким алгоритмом подтверждения состояния может быть соответствующий ГА-метод [10].

Кроме иллюстрации технической возможности такого конструирования данный метод имеет и практические предпосылки, которые заключаются в следующем:

- детерминированная фаза гарантирует распространение влияния неисправности на внешние выходы ЦУ, если оно осуществимо;
- применение ГА в фазе активизации неисправности исключает сложные процедуры обратного распространения для нескольких тактов модельного времени, которые требуют построения больших стеков возвратов рассмотрения альтернатив.

Таким образом, целью данной работы является разработка двухуровневого гибридного метода построения тестов последовательностных ЦУ, который на нижнем уровне объединяет детерминированный метод распространения влияния неисправности и ГА-метод достижения заданного состояния.

Покажем конструктивно построение такого ГА-метода генерации тестов.

ГА-метод достижения заданного состояния в ЦУ

Пусть рассматривается синхронное последовательностное ЦУ A_0, работа которого моделируется в трёхзначном алфавите $E_3 = \{0,1,u\}$. Пусть Z – множество всех определённых состояний, $z_u = (uu...u)$ – начальное полностью неопределённое состояние.

Задача заключается в переводе ЦУ A_0 из заданного начального состояния $z_{нач}$ в конечное состояние $z_{кон}$. В том случае, когда в $z_{кон}$ не все линии имеют определённое значение $0,1 \in E_3$ (т.е. имеются линии с неопределённым значением сигнала $u \in E_3$), то, вообще говоря, $z_{кон}$ определяет некоторое множество состояний устройства.

Подробно разработанный авторами ГА построения последовательностей достижения состояния (ПДС) описан в [2, 10]. Здесь мы приведём его краткое описание, достаточное для понимания места в разрабатываемом методе построения тестов.

В ГА построения идентифицирующих последовательностей (ИдП) каждое потенциальное решение-последовательность S называется особью. Каждый входной вектор особи-последовательности соответствует одному такту модельного времени работы ЦУ. Набор особей образует популяцию. Основной характеристикой особи является оценочная функция, которая показывает качество решения поставленной задачи.

Выполним анализ и построение оценочной функции. Семантически в нашей задаче оценочная функция показывает: насколько близко текущее состояние $z_{тек}$ моделируемого ЦУ A_0, стартующего из состояния $z_{нач}$ и получающего на вход последовательность S, находится от требуемого состояния $z_{кон}$. Поскольку мы рассматриваем наиболее общий случай, когда граф переходов ЦУ A_0 не известен, то для формализации оценки будем использовать расстояние по Хэммингу для двоичного представления $z_{тек}$ и $z_{кон}$, показывающее число совпадений элементов состояний. Таким образом, для заданного ЦУ A_0 и заданной входной последовательности S оценка вычисляется на основании результатов исправного моделирования: $z_{тек} = \delta(z_{нач}, S)$, где $\delta(Z,S)$ – функция перехода устройства A_0. Тогда можно записать:

$$O(A_0, S, z_{нач}, z_{кон}) = O(z_{тек}(A_0, S, z_{нач}), z_{кон}) = \sum_{Z_{тек}[i]=Z_{кон}[i]} 1, \quad (1)$$

где $z[i]$ показывает значение i-го элемента состояния, $i = \overline{1,m}$, m – число линий состояний в A_0.

Оценка особи-последовательности в виде (1) содержит только одну компоненту, прямо показывающую число совпадений на линиях состояний ЦУ. Данную оценку можно обобщить путём добавления компонент, которые показывают активность ЦУ A_0 при работе на входной последовательности S, а также длину особи, что должно улучшить качество оценки [11]. Такие компоненты, очевидно, должны иметь меньший вес в оценочной функции.

Реализация вычисления оценочной функции представлена ниже.

Алгоритм А1

```
ОценитьОсобь( A0 , S , zнач , zкон )
{
  ИнициализацияНачальногоСостояния( A0 , zнач );
  SV =МоделированиеРаботыИсправногоЦУ( A0 , S );
  zтек =ОпределитьДостигнутоеСостояние( SV );
  // сравнить достигнутое состояние с требуемым –формула (1)
  return Оценка= O(A0, zтек, zкон);
}
```

Состояние $z_{нач}$ необходимо для инициализации ЦУ A_0 всякий раз перед моделированием, которое выполняется для текущей особи-последовательности S с целью её оценки. Состояние $z_{тек}$ является результатом такого моделирования и определяется из массива SV, в котором хранятся значения сигналов для всех линий устройства.

Непосредственно ГА-метода решения задачи генерации ПДС строится по одноуровневой схеме [2]. Данная схема определяет эволюцию популяций потенциальных решений. Общая структура такой эволюции неоднократно описывалась и мы не будем приводить её реализацию. При эволюции потенциальных решений, представленных входными двоичными последовательностями, используются определённые над ними проблемно-ориентированные эволюционные операции: скрещивание и мутация [2].

Критерием остановки эволюции является совпадение текущего состояния $z_{тек}$ и финального $z_{кон}$. При этом численно оценка в виде (1) будет равняться m.

Метод построения тестов с подтверждением состояний

Постановка задачи является традиционной. Пусть задано исправное синхронное последовательностное ЦУ A_0. Также задано множество неисправностей $F = \{f_1, ..., f_n\}$, причём $|F| = n < \infty$. Неисправности множества F порождают, соответственно, множество неисправных ЦУ $A = \{A_1, ..., A_n\}$. В качестве модели неисправности может выступать любая такая модель, для которой разработаны методы её моделирования. Для ясности изложения будем рассматривать множество ОКН. Множество неисправных ЦУ не строится явно, а порождается путём внесения влияния неисправностей в процессе моделирования ЦУ.

Требуется определить исправность заданного ЦУ $A_i \in A$, т.е. проверить соответствует ли его поведение A_0.

Метод решения задачи строится по двухуровневой схеме [2]. На верхнем уровне метода из списка непроверенных неисправностей F' выбирается целевая неисправность $f_{цел}$, для которой на нижнем уровне будет построен тест. В качестве целевой выбирается любая непроверенная неисправность $f_{цел} = f' \in F'$, а стратегия такого выбора должна быть обоснована на основании машинных экспериментов.

Для иллюстрации места ГА в методе построения теста одной неисправности нижнего уровня рассмотрим рис.1. Здесь изображены три такта времени работы ЦУ: $t-1$, t, $t+1$. Полностью поведение комбинационного блока (КБ) в произвольный момент времени определяют значения на его входах X и псевдовходах Z. Для КБ в

момент времени модельного t это векторы x_t и z_t. Начальным на рисунке является момент времени t, когда происходит внесение влияния $f_{цел}$. В данном такте времени t реализуется метод существенного пути построения тестов [12], состоящий из двух фаз.

Обратное распространение выполняется исходя из условия проявления неисправности: на линии в месте неисправности необходимо обеспечить различные значения в устройствах A_0 и A_f. При этом будут получены значения (возможно частично определённые) для входов и псевдовходов ЦУ. Для примера на рисунке имеем: $x_t = (1,u,u,1)$ и $z_t = (0,1,u,u,1)$. Реализация фазы обратного распространения осуществляется любым соответствующим структурным методом. Поскольку обработка ЦУ производится в условиях частичной неопределенности в алфавите E_3, то может быть применён метод для частично определённых булевых функций, например, из [13].

Продвижение вперёд: необходимо выполнить продвижение различия поведения исправного и неисправного ЦУ на внешние выходы/псевдовыходы. Для нашего примера имеем: $x_{t+1} = (u,u,u,0/1)$ и $z_{t+1} = (u,u,1/0,u,u)$. Здесь 0/1 обозначает значение сигнала на линии в паре устройств A_0 и A_f.

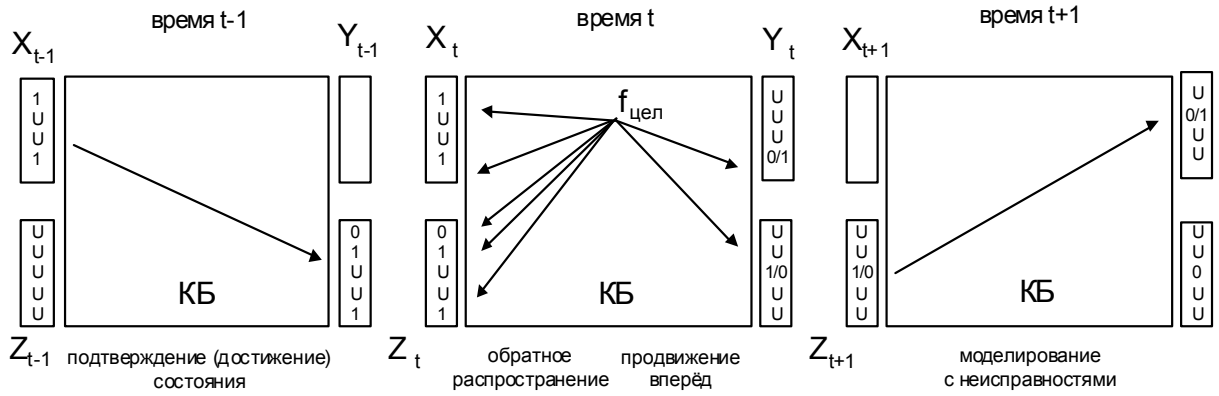

Рис.1. Принцип работы метода построения тестов с подтверждением состояний.

Фаза продвижения вперёд может потребовать несколько тактов модельного времени для распространения различия на внешние выходы. Дополнительного продвижения вперёд в тактах времени $t+1$ и далее может не потребоваться в том случае, если в момент времени t различие в поведении исправного ЦУ A_0 и неисправного A_f распространилось на внешние выходы. На рисунке $y_{t+1} = (u,u,u,0/1)$ и, следовательно, нет необходимости выполнять дополнительное распространение вперёд для времени $t+1,...$ Если бы в векторе y_{t+1} не было различных значений для A_0 и A_f, то дополнительное продвижение вперёд было бы необходимо (как это показано на рис.1).

Центральным моментом в методе является подтверждение состояния z_t, полученного в фазе обратного распространения: стартуя из $z_u = (uu...u)$ необходимо получить состояние z_t в паре устройств A_0 и A_f.

Построить последовательность S, переводящую устройства A_0 и A_f в состояние z_t из z_u можно с помощью процедуры обратного распространения, применяя её итеративно для тактов времени $t-1$, $t-2$ и т.д. до выполнения условия $z_{t-i} = z_u$. Однако сложность заключается именно в том, что обратное распространение выполняется через несколько тактов модельного времени. Структурные методы в этом случае не позволяют обрабатывать большие ЦУ.

С другой стороны, задача подтверждения состояния в рассматриваемой постановке фактически является задачей построения ПДА. В этом случае параметрами ГА-метода, описанного в предыдущем разделе, являются: $z_{нач} = z_u$, $z_{кон} = z_t$. Таким образом, в рассматриваемом методе построения тестов для достижения одной цели, определённой на верхнем уровне, вызываются два различных метода на нижнем уровне: метод существенного пути и ГА-метод достижения состояния. Поэтому описываемый метод мы называем гибридным двухуровневым.

Для описания метода построения тестов будем считать, что у нас имеется процедура, реализующая метод существенного пути: для заданного устройства A_0 и неисправности $f_{цел}$ порождаются входные наборы x_t и $z_t \neq z_u$, обеспечивающие распространение влияния неисправности на внешние выходы устройства. Известно [14], что в методах данного класса при выполнении условий транспортировки влияния неисправности возможно формирование нескольких альтернатив для значений сигналов на линиях ЦУ. С другой стороны, при выполнении процедуры обратного распространения возможны противоречия в формировании таких значений, которые необходимы для выполнения условий активизации и распространения. В этом случае в методе происходит откат до места выбора последней альтернативы. Таким образом, результатом рассмотрения каждой такой альтернативы являются входной набор x_t и состояние $z_t \neq z_u$ в том случае, если удалось произвести распространение влияния

неисправности на внешние выходы ЦУ. В противном случает метод существенного пути возвратит признак непроверяемости неисправности. Считаем, что реализация метода существенного пути формирует множество альтернатив, которые доступны для рассмотрения далее. Тогда в описываемом методе построения тестов на нижнем уровне формируется цикл по всем таким альтернативам: (x_t, z_t).

Работу метода на нижнем уровне можно представить в виде следующего псевдокода.

Алгоритм А2

```
Гибридное_ПостроениеТеста( A0 , fцел )
{
  МетодСущественногоПути( A0 , fцел );
  while( есть нерассмотренные состояния  zt ≠ zu  )
  {
    // вызов ГА-метода построения ПДС
    ГА_Достижение_Состояния( A0 , zu , zt );
    if( последовательность построена )
    {
      S =СформироватьТест();
      ДобавитьВТест( S );
      return; // тест построен–возврат на верхний уровень
    }
    else
      continue; // переход к выбору другой альтернативы
  }//конец есть варианты в методе существенного пути
  ОтметитьКакНепроверяемую( fцел );
  return;
} // конец алгоритма
```

Процедура *МетодСущественногоПути()* выполняет в нашем методе фазу продвижения вперёд и обратного распространения для заданной неисправности, а также формирует множество альтернатив (x_t, z_t).

При рассмотрении каждой из альтернатив происходит вызов ГА-метода построения ПДС. Условием его вызова является тот факт, что для активизации неисправности необходимы определённые значения в векторе состояния ЦУ z_t. Это выполняется при условии $z_t \neq z_u$. Отличием данной процедуры от описанного выше ГА-метода построения ПДС является то, что построенная последовательность должна обеспечивать необходимые состояния как в исправном A_0, так и в неисправном A_f ЦУ.

Если ГА построил ПДС для z_t, то для $f_{цел}$ в функции *СформироватьТест()* компонуется тестовая последовательность S, состоящая из:

- последовательности достижения состояния для z_t;
- входного набора z_t;
- последовательности распространения влияния неисправности.

Если рассмотрены все альтернативы и тест для $f_{цел}$ не построен, либо множество альтернатив было пусто, то данная неисправность отмечается как непроверяемая.

В том случае, если метод не построил тестовую последовательность для некоторой неисправности f_i, это не говорит о том, что такая последовательность не существует. Поскольку нижний уровень метода фактически состоит из двух алгоритмов, то каждый из них может стать причиной отрицательного результата:

- метод существенного пути не позволил выполнить распространение влияния неисправности ввиду собственных ограничений;
- ГА не построил ПДС ввиду собственных ограничений.

Если произвести сравнение данного метода построения тестов и классической схемы применения ГА [6, 2], то хорошо видно место соответствующих ГА-методов построения последовательностей. Рис. 1. можно рассматривать как изображение обобщённого подхода к решению задачи построения теста одной неисправности $f_{цел}$ для последовательностных ЦУ, который включает фазы активизации неисправности $f_{цел}$ и распространения её влияния на внешние выходы.

В двухуровневом ГА-методе построения тестов [6, 12] происходит псевдослучайная генерация последовательностей, активизирующих неисправность. Т.е. такая генерация должна построить состояние z_{t+1} (в такте времени t на рисунке) с различными значениями в исправном и неисправном ЦУ, что выполняется в такты времени до t включительно. Дальнейшее распространение влияния происходит с помощью процедуры ГА в тактах времени $t+1$ и т.д.

В гибридном методе, который разрабатывается в данной статье, активизация и распространение влияния неисправности происходит с помощью метода существенного пути: такты времени t, $t+1$ и т.д. Тогда как ГА

необходим для достижения состояния z_t и работает по такт времени $t-1$ включительно.

В качестве замечаний о дальнейших направлениях исследования можно отметить следующее. Описанный метод использует 3-значный алфавит моделирования E_3 и, следовательно, одномерную активизацию неисправности. Ещё большей эффективности метода можно достичь при использовании многозначных алфавитов [14]. Использование шестизначного алфавита T_6 позволит проводить многомерную активизацию путей распространения неисправностей, например, с помощью D-алгоритма. Таким образом, использование многозначных логик является одним из перспективных направлений при построении подобных методов.

Выводы

В работе разработан гибридный метод генерации тестовых последовательностей для синхронных последовательностных ЦУ. Его особенностью является реализация уровня построения теста одной неисправности, которая заключается в совместном применении как детерминированного так и вероятностного подходов. Детерминированный метод обеспечивает распространения влияния неисправности на внешние выходы ЦУ. В качестве вероятностного метода применён ГА-метод построения последовательностей достижения состояний, целью которого является перевод исправного и неисправного устройств из начального полностью неопределённого состояния в заданное. Объединение двух альтернативных подходов построения тестов повышает эффективность такой процедуры для сложных современных СБИС.

В качестве дальнейших исследований можно отметить применение многозначных логик, что позволит выполнять многомерную активизацию путей распространения неисправностей.

Литература

1. ITRS 2010 technology roadmap (1.09.2013) http://www.itrs.net/Links/2010ITRS/Home2010.htm.

2. Д.Е. Иванов Генетические алгоритмы построения входных идентифицирующих последовательностей цифровых устройств / Иванов Д.Е. – Донецк. – 2012. – 240с.

3. Niermann T. HITEC: A Test Generation Package for Sequential Circuits / T. Niermann, J.H. Patel. – Proc. European Design Automation Conf. – 1991. – P.214-218.

4. Lee D.H. A new test generation method for sequential circuits / D.H. Lee, S.M. Reddy. – Proc. Int. Conf. Computer-Aided design. – 1991. – P.446-449.

5. Wunderlich H.-J. Multiple distributions for biased random test patterns / H.-J. Wunderlich. – IEEE Trans. Comp. Aided Design. – 1990. – Vol.9, №6. – P.584-593.

6. Corno F. Experiences in the use of evolutionary techniques for testing digital circuits / F. Corno, M. Sonza Reorda, M. Rebaudengo. – Proc. of Conf. Applications and science of neural networks, fuzzy systems, and evolutionary computation, San Diego CA. – 1998. – P.128-139.

7. Д.Е. Иванов Генетический подход проверки эквивалентности последовательностных схем / Д.Е. Иванов. – «Радіоелектроніка. Інформатика. Управління» . – Запоріжжя, ЗНТУ. – 2009. – №1(20). – С.118-123.

8. Saab D.G. Iterative [Simulation-Based+Deterministic Techniques]=Complete ATPG / D.G. Saab, Y.G. Saab, J. Abraham. – Proc. Int. Conf. on Computer Aided Design. – 1994. – P.40-43.

9. Rudnick E.M Combining deterministic and genetic approaches for sequential circuit test generation / E. M. Rudnick and J. H. Patel. – Proc. Design Automation Conf. – 1995. – P.183-188.

10. Иванов Д.Е. Алгоритмы достижения состояний в цифровых устройствах и их применение в задачах диагностики / Д.Е. Иванов. – Вісник Хмельницького національного університету. Технічні науки.- Хмельницький. – 2012. – №3(189). – С.104-110.

11. Иванов Д.Е. Применение информации структурного уровня в алгоритмах построения идентифицирующих последовательностей / Д.Е. Иванов. – Известия ЮФУ. Технические науки. – 2013. – №1. – С.149-160.

12. Убар Р. Проектирование контролепригодных дискретных систем / Р.Убар. – Таллин:Из-во Таллиннского политехнического института. – 1988. – 68с.

13. Черемисинова Л.Д. Проверка схемной реализации частичных булевых функций / Л.Д. Черемисинова, Д.Я. Новиков. – Вестник Томского государственного университета, Управление, вычислительная техника и информатика. – 2008.- № 4(5). – С.102-111.

14. Скобцов Ю.А. Логическое моделирование и тестирование цифровых устройств / Ю.А. Скобцов, В.Ю. Скобцов. – Донецк:ИПММ НАНУ, ДонНТУ. – 2005. – 436с.

References

1. ITRS 2010 technology roadmap (1.09.2013) http://www.itrs.net/Links/2010ITRS/Home2010.htm.

2. Ivanov D.E. Geneticheskie algoritmy postroeniya vxodnyx identificiruyushhix posledovatel'nostej cifrovyx ustrojstv / D.E. Ivanov – Doneck. – 2012. – 240s. [in Russian]

3. Niermann T. HITEC: A Test Generation Package for Sequential Circuits / T. Niermann, J.H. Patel. – Proc. European Design Automation Conf. – 1991. – P.214-218.

4. Lee D.H. A new test generation method for sequential circuits / D.H. Lee, S.M. Reddy. – Proc. Int. Conf. Computer-Aided design. – 1991. – P.446-449.

5. Wunderlich H.-J. Multiple distributions for biased random test patterns / H.-J. Wunderlich. – IEEE Trans. Comp. Aided Design. – 1990. – Vol.9, №6. – P.584-593.

6. Corno F. Experiences in the use of evolutionary techniques for testing digital circuits / F. Corno, M. Sonza Reorda, M. Rebaudengo. – Proc. of Conf. Applications and science of neural networks, fuzzy systems, and evolutionary computation, San Diego CA. – 1998. – P.128-139.

7. Ivanov D.E Geneticheskij podxod proverki e'kvivalentnosti posledovatel'nostnyx sxem / D.E. Ivanov. – «Radioelektronika. Informatyka. Upravlinnia».- Zaporizhzhia, ZNTU. – 2009. – №1(20). – S.118-123. [in Russian]

8. Saab D.G. Iterative [Simulation-Based+Deterministic Techniques]=Complete ATPG / D.G. Saab, Y.G. Saab, J. Abraham. – Proc. Int. Conf. on Computer Aided Design. – 1994. – P.40-43.

9. Rudnick E.M Combining deterministic and genetic approaches for sequential circuit test generation / E. M. Rudnick and J. H. Patel. – Proc. Design Automation Conf. – 1995. – P.183-188.

10. Ivanov D.E Algoritmy dostizheniya sostoyanij v cifrovyx ustrojstvax i ix primenenie v zadachax diagnostiki / D.E. Ivanov. – Visnyk Khmelnytskoho natsionalnoho universytetu. Tekhnichni nauky. – Khmelnytsky. – 2012. – №3(189). – S.104-110. [in Russian]

11. Ivanov D.E Primenenie informacii strukturnogo urovnya v algoritmax postroeniya identificiruyushhix posledovatel'nostej. – Izvestiya YuFU. Texnicheskie nauki. – 2013. – №1. – S.149-160. [in Russian]

12. Ubar R. Proektirovanie kontroleprigodnyx diskretnyx sistem / R.Ubar. – Tallin:Iz-vo Tallinnskogo politexnicheskogo instituta. – 1988. – 68s. [in Russian]

13. Cheremisinova L.D. Proverka sxemnoj realizacii chastichnyx bulevyx funkcij / L.D. Cheremisinova, D.Ya. Novikov. – Vestnik Tomskogo gosudarstvennogo universiteta, Upravlenie, vychislitel'naya texnika i informatika. – 2008.- № 4(5). – S.102-111. [in Russian]

14. Skobcov Yu.A. Logicheskoe modelirovanie i testirovanie cifrovyx ustrojstv / Yu.A. Skobcov, V.Yu. Skobcov. – Doneck:IPMM NANU, DonNTU. – 2005. – 436s. [in Russian]

Рецензія/Peer review : 30.09.2013 р. Надрукована/Printed :24.11.2013 р.
Рецензент: Скобєлєв В.Г., д.ф.-м.н., д.т.н., проф., п.н.с. відділу теорії керуючих систем ІПММ НАН України

УДК 681.3.053

С.В. БЕВЗ, В.В. ВОЙТКО, О.О. СІВЕЦЬ
Вінницький національний технічний університет

МОДЕЛЬ ОБГРУНТОВАНОГО ВИБОРУ МЕТОДІВ ТЕСТОВОГО КОНТРОЛЮ ЗНАНЬ

У статті розглянуто типи тестів, проведено їх порівняльний аналіз, приведені класифікації методів тестового контролю знань, здійснено аналіз цих методів та розроблено модель їх обґрунтованого вибору.

Ключові слова: тест, тестовий контроль знань, методи тестування.

S.V. BEVZ, V.V. VOJTKO, O.O. SIVETS
Vinnitsya national technical university

THE MODEL OF REASONABLE CHOICE OF METHODS OF KNOWLEDGE TESTING

Abstract – The aim of the research is automating the process of selecting methods of knowledge testing through their systematic, classifying and mapping in the model of reasonable choice of methods of knowledge testing.

The article deals with types of tests such as tests focused on criteria and tests focused on the rate , carried out a comparative analysis of them, given classification of methods of the knowledge testing, analysis of this methods and designed the model of their informed choice.

The test control is an important aspect of education activities, it promotes positive reinforcement of training and cognitive process, enhances the quality of knowledge, thus increasing the effectiveness of control measures, leads saving time in the classroom.

Keywords: test, test control of knowledge, methods of testing.

Вступ

Одним із важливих аспектів освітньої діяльності постає система контролю якості знань. Сучасна система освіти потребує розробки та впровадження ефективних засобів моніторингу рівня знань учнів шляхом впровадження системи контрольно-тестових засобів оцінювання якості навчального процесу [1,2]. Комп'ютерне тестування належить до адаптивної моделі педагогічного тестування. То ж актуальною є проблема розробки та впровадження автоматизованих засобів тестового контролю з метою забезпечення швидкої і якісної перевірки знань, активізації суб'єктів навчального процесу та підвищення в цілому ефективності навчання.

Постановка завдання

Метою роботи є автоматизація процесу вибору методів тестового контролю знань шляхом їх систематизації, класифікації та відображення в моделі обґрунтованого вибору засобів тестування. Об'єктом дослідження вбачаємо процеси тестування якості отриманих знань. Предметом дослідження постають засоби тестового контролю. Задачами дослідження є аналіз методів тестування знань та розробка моделі їх обґрунтованого вибору.

Порівняльний аналіз типів тестів

Класично виділяють два базових типи тестів [3]:

- тести, орієнтовані на критерій (критеріально-орієнтовані);
- тести, орієнтовані на норму (нормативно-орієнтовані).

Функціональні характеристики вказаних типів тестів зведені в табл.1.