

2 (99)' 2013

ІНФОРМАЦІЙНО -КЕРУЮЧІ СИСТЕМИ НА ЗАЛІЗНИЧНОМУ ТРАНСПОРТІ

Виходить 6 разів на рік

Видається з 23 квітня 1996 р.

INFORMACIJNO–KERUŮCI SISTEMI NA
ZALIZNICNOMU TRANSPORTI

Зміст – Содержание – Contents

Видання

Української державної
академії залізничного
транспорту

Даренский А.Н.

Жесткость боковых упоров промежуточных
скреплений КБ.....3

Бабанин А.Б., Турчинов Р.В.

Задачи и оптимизация требований к
информационным системам для
диагностирования локомотивов.....5

Раскин Л.Г., Каткова Т.И., Головки В.А.

Нечеткий многомерный дискриминантный
анализ в задаче диагностики состояния.....13

Серая О.В.

Транспортная задача высокой размерности со
стохастическим спросом.....18

Клименко К.С., Клименко Л.А.

Використання марківських кіл для розрахунку
показників функційної безпечності
дубльованих відновлювальних систем
з періодичним контролем.....23

Баранник В.В., Кривонос В.Н., Хаханова А.В.

Структурная модель информативности значимых
компонент трансформант.....26

Межуев В.І., Кудінов М.В., Тимошенко Є.В.

Колективний підхід до моделювання предметних
областей засобами програмної інженерії.....30

Головко А.В. Использование клеточных автоматов для представления массообмена и энергообмена в процессе распространения огня.....	36
Баранник В.В., Рогоза И.Е., Красноруцкий А.А. Обоснование базовой технологии компрессии изображений с заданным качеством визуализации.....	41
Прохорченко А.В. Розробка методики визначення кількості ниток графіку руху спеціалізованих поїздів на залізничному напрямку за умови дотримання технічних і технологічних обмежень.....	46
Ломотько Д.В., Бронза С.Д., Юрчак Н.С., Овчів М.Ж. Розподіл імовірності станів системи обороту вагонів на залізничному вузлі. Загальне рішення. Частина II.....	52
Яновський П.О. Прогнозування транспортних потоків з використанням нелінійних моделей.....	56
Ходаківський О. М. Модель оптимального формування людського ресурсу у системі українського транспортного логістичного центру Укрзалізниці.....	65
Пронина О.И., Пятикоп Е.Е. Реализация модели классификации ситуаций, представленных в четком и нечетком виде.....	71
Шкиль А.С., Сыревич Е.Е., Альмадхоун С., Фастовец Г.П. Структурное и функциональное диагностирование HDL-моделей цифровых устройств в САПР РЭА.....	75
Раисов Ю.А., Бычков Н.И., Бычков И.В. NURBS – интерполяция точечно-заданных кривых.....	83
Бабаев М.М., Гребенюк В.Ю. Проверка модели контроля состояния путевого участка на адекватность.....	91
Блиндюк В.С. Нейромережева кон'юнктивна модель.....	97

ШКИЛЬ А.С., к.т.н., доцент каф. АПВТ,
СЫРЕВИЧ Е.Е., к.т.н., доцент каф. ЭВМ,
АЛЬМАДХОУН С., аспирант каф. АПВТ,
ФАСТОВЕЦ Г.П., к.т.н., доцент каф. АПВТ, (ХНУРЭ)

Структурное и функциональное диагностирование HDL-моделей цифровых устройств в САПР РЭА

Рассмотрена общая структура САПР РЭА на этапе функциональной верификации и поиска ошибок проектировании в моделях цифровых устройств на языках описания аппаратуры. В качестве методов поиска ошибок проектирования рассмотрены функциональный метод на основе таблиц функций неисправностей и структурный метод на основе графовой модели для языковых описаний стиля «поток данных». Приведены оценки эффективности указанных методов.

Ключевые слова: HDL-модель, верификация, методы поиска ошибок проектирования, диагностический эксперимент

Введение и постановка задачи

Входным представлением цифровых устройств (ЦУ) в системах автоматизированного проектирования радиоэлектронной аппаратуры (САПР РЭА) являются описания (модели) на языках описания аппаратуры (Hardware Description Language, HDL). Исходной информацией для построения HDL-модели является спецификация, которая может быть представлена в форме алгоритма функционирования (словесного или формального), таблицы функционирования, временной диаграммы или одним из способов описания цифровых автоматов (таблицей переходов/выходов, содержательным графом переходов, граф-схемой алгоритма).

Полный цикл диагностирования HDL-модели ЦУ на этапе функциональной верификации, начиная от его описания на языках описания аппаратуры и заканчивая поиском места возникновения ошибки проектирования, включает в себя следующие этапы: анализ HDL-кода и при необходимости его преобразование (проверка на синтаксис, декомпозиция, преобразование во внутреннюю модель), построение списка ошибок проектирования, генерацию тестов, получение эталонов по спецификации или с помощью внешней моделирующей программы, моделирование тестовых воздействий на HDL-модели в специальной среде моделирования и получение выходных реакций, сравнение результатов моделирования с эталонами, построение алгоритмов поиска ошибок проектирования и проведение диагностического эксперимента (ДЭ).

На основании спецификации строится HDL-код модели и для него составляется список ошибок проектирования. Под ошибкой проектирования понимается ошибка в HDL-операторе, не относящаяся к классу синтаксических и нарушающая алгоритм функционирования модели устройства, заданный спецификацией. В общем случае для HDL-моделей стиля «поток данных» рассматриваются ошибки проектирования типа «замена оператора», а для HDL-моделей стиля «автоматный шаблон» рассматриваются ошибки проектирования типа «неправильный переход в графе переходов автомата».

Структура HDL-модели ЦУ зависит от формы спецификации и выбранного стиля языкового описания модели. Используя дуализм HDL-кода, который одновременно является программным кодом и описанием аппаратной реализации ЦУ, строится графовая модель, состоящая из информационного и управляющего графов. На основании этой модели строятся тесты, реализующие активизацию путей в информационном графе или обход условий управляющего графа. Полученные тесты моделируются и результаты сравниваются с эталонами.

Если HDL-код корректен (результат моделирования совпал с эталоном), HDL-модель передается на последующие этапы проектирования (синтез и имплементация). Если результат моделирования не совпадал с эталоном, выполняется этап поиска ошибок проектирования в HDL-коде. При этом применяются функциональные или структурные алгоритмы поиска ошибок проектирования.

После нахождения ошибочного оператора HDL-код корректируется, и моделирование тестов повторяется.

На рисунке 1 показана общая структура САПР РЭА на этапе функциональной верификации и корректировки ошибок проектирования в HDL-коде.

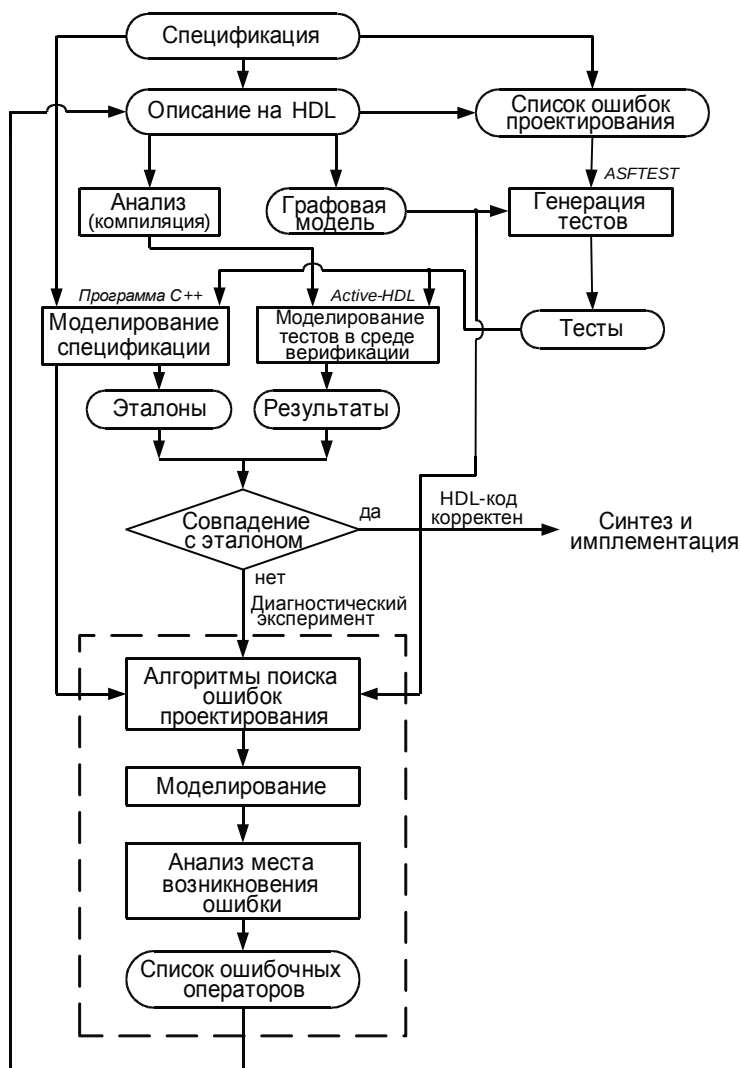


Рисунок 1 – Функциональная верификация HDL-моделей цифровых устройств в САПР РЭА

В [1] и [2] были предложены внутренняя модель и методы генерации тестов для тестирования HDL-моделей, а в [3] и [4] функциональный и структурный методы поиска дефектов. В качестве ошибки проектирования рассматривались ошибки типа «замена оператора» из списка predetermined операторов языка VHDL. В качестве тестов использовались различающие последовательности (РПС), а HDL-модель рассматривалась как совокупность сигналов и связей между ними. При диагностировании HDL-

моделей выделяют два типа контрольных точек (КТ), используемых при локализации места ошибки. КТ первого рода – сигнал (переменная) модели, эталонные значения которых известны из спецификации. КТ второго рода – сигнал (переменная) модели, значения которых наблюдаемы, но до начала ДЭ неизвестны или не определены.

Целью данной работы является разработка методов поиска (локализации) ошибок проектирования в HDL-моделях цифровых устройств без глобальных

обратных связей, и разработка методов проведения диагностического эксперимента над HDL-моделями.

Метод поиска ошибок проектирования на основе таблиц функций неисправностей

Для поиска ошибок проектирования в HDL-коде используется безусловный метод проведения диагностического эксперимента, в котором выбор последовательности реализации элементарных проверок (ЭП) не зависит от результатов предыдущих проверок. Универсальной математической моделью для представления диагностической информации является таблица функций неисправностей (ТФН), в которой строки соответствуют допустимым элементарным проверкам (входным воздействиям на объект диагностирования), а столбцы – техническим состояниям объекта диагностирования (ошибкам проектирования) [3]. Традиционно для построения алгоритмов поиска дефектов аппаратуры используют таблицы неисправностей (ТН), но в данной работе предположено использовать именно фрагмент ТФН, связанный с выходными переменными модели.

Применение ТФН объясняется тем, что при проведении ДЭ необходимо найти тот столбец (ошибку проектирования), с которым совпадает реакция, полученная на выходе ЦУ. К тому же классическую ТН использовать нельзя, так как она содержит только двучисленные неисправности («работает / не работает»). В нашем же случае, ошибки проектирования являются многозначными. Таким образом, при проведении ДЭ используется ТФН, где P – множество допустимых элементарных проверок (тестовых наборов); P_i – элементарная проверка в форме РПС для каждого функционального элемента (ФЭ); E – множество ошибок проектирования; e – исправное состояние объекта диагностирования; e_j – неисправное состояние объекта диагностирования (замена одного оператора другим, встречающимся в коде); R – множество результатов элементарных проверок; R_i^j – результат элементарной проверки P_i в техническом состоянии e_j .

Множество P обладает свойством обнаружения любой ошибки проектирования из множества E , т.е. для любой ошибки $e_j \in E$ найдётся хотя бы одна элементарная проверка P_i такая, что $R_i^{испр.} \neq R_i^j$, а также свойством различения всех ошибок из множества E , т.е. для каждой пары ошибок (e_j^i, e_j^k)

найдётся хотя бы одна элементарная проверка $P_i \in P$ такая, что $R_i^j \neq R_i^k$.

Определим этапы построения алгоритма поиска ошибок проектирования.

1. HDL-модель ЦУ представляется внутренней моделью в виде композиции двух графов (I-графа и С-графа).

2. Для I-графа строятся тесты РПС, идентифицирующие отличия всех типов функциональных элементов графовой модели друг от друга.

3. Строится ТФН путем внесения в HDL-код ошибок проектирования из списка (ошибки типа «замена операторов») и вычисления соответствующих реакций.

Проведение ДЭ заключается в подаче РПС на диагностируемый код HDL-модели и сравнении полученного результата на выходе ЦУ последовательно с каждым столбцом ТФН. Если произошло совпадение с каким-либо столбцом, то можно сделать вывод не только о том, что в HDL-модели есть ошибка, но и точно определить какой ФЭ и на какой был ошибочно заменен.

Область применения разработанного метода ограничивается количеством строк HDL-кода, для которого строится фрагмент ТФН. Её размерность вычисляется:

$$D = i \times j, \quad (1)$$

где i – количество строк ТФН (количество тестовых наборов РПС), j – количество столбцов ТФН (число ошибок проектирования R плюс исправное состояние).

Количество ошибок проектирования R вычисляется по формуле:

$$R = \sum_{m=1}^n ((A_n^k / n) \cdot l_m), \quad (2)$$

где A_n^k – размещение из n элементов по k ; n – число

групп логических элементов, k – число соединений, l_m – число операторов в каждой группе.

Проанализировав формулу (2), можно сделать вывод, что между размерностью ТФН и количеством ошибок проектирования существует нелинейная зависимость.

Использование ТФН как формы представления информации при построении и реализации алгоритмов диагностирования часто невозможно по причине высокой размерности ТФН. А значит, целесообразнее использовать предложенный метод только для HDL-

моделей, содержащих ограниченное количество операторов. Если рассматривать разработанный фрагмент ТФН с точки зрения реализации его обработки в программе на языке C++, то занимать такая таблица будет приблизительно 80 байт, если использовать тип данных языка C++ `std::vector<bool>`. Такой тип данных является битовым массивом, который реализован в виде специализации основного шаблона `std::vector<T>`.

Разработанный метод поиска ошибок позволил определить структуру тестового окружения в системах моделирования HDL-моделей. При этом предлагается использовать известные методы свертки двоичных последовательностей (например, на основе сигнатурного анализатора). В качестве примера рассмотрим фрагмент ТФН (рис.2), которая служит основой для функционального метода.

Тест				G17_out	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	1	0	0	1	1	1	-	-	0	1	1	1	1	1	1	-	-	-	1	-	-	
0	1	0	0	1	1	1	1	0	0	1	1	1	1	1	1	1	-	0	1	-	-	
0	0	1	1	0	0	1	1	0	1	1	0	1	0	1	0	0	0	0	0	1	1	

Рисунок 2 – Фрагмент ТФН с неопределенными ячейками

Под символом «-» в ячейке понимается неопределенное значение «X». Если использовать последовательный сигнатурный анализатор для получения свертки, то неопределенные значения необходимо заменить на значение «0», т.к. нулевое

значение не влияет на очередное значение в сдвиговом регистре. Доопределенная ТФН будет выглядеть следующим образом (рис.3):

Тест				G17_out	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	1	0	0	1	1	1	0	0	0	1	1	1	1	1	1	1	0	0	0	1	0	0
0	1	0	0	1	1	1	1	0	0	1	1	1	1	1	1	1	0	0	1	0	0	
0	0	1	1	0	0	1	1	0	1	1	0	1	0	1	0	0	0	0	0	1	1	

Рисунок 3 – Фрагмент доопределенной ТФН

Технология создания тестовой среды для верификации и диагностирования может быть автоматизирована при наличии автоматического генератора РПС (их число и вид заранее известен), сформированного списка ошибок проектирования (который строится на основе композиционной графовой модели), HDL-модели проверяемого устройства, HDL-модели сигнатурного генератора и модели памяти (для формирования базы неисправных состояний). Все элементы могут быть реализованы в виде модулей в едином тестовом окружении (рис. 4). Блоки 2 и 3 на рисунке 4 показывают техническую реализацию блока сравнения с эталоном САПР РЭА (рис.1).

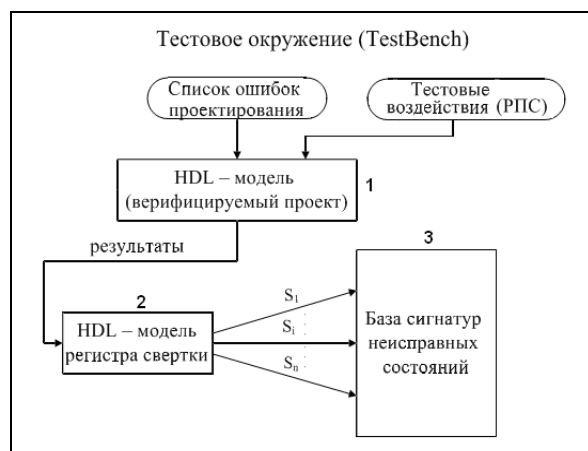


Рисунок 4 – Тестовое окружение автоматизированной системы верификации

HDL-модель сигнатурного генератора (регистра свертки) приведена на рис.5.

```
entity SA is
  port ( X: in STD_LOGIC; C: in STD_LOGIC; R: in STD_LOGIC;
        Q: out STD_LOGIC_VECTOR (1 to 16));
end SA;
architecture SA of SA is
  signal Qint: std_logic_vector(1 to 16);
begin
  Signatura: process (R,C) begin
    if (R='1') then Qint <= "0000000000000000";
    elsif (C'event and C='1') then Qint <= (Qint (7)xor Qint (9)xor Qint (12)xor Qint
      (16)xor X) & Qint (1 to 15);
    end if;
  end process;
  Q <= Qint;
end SA;
```

Рисунок 5 – HDL-модель сигнатурного регистра свертки

Предложенный подход позволяет значительно сократить объем хранимых эталонных реакций и время, затрачиваемое на сравнение полученных в ходе ДЭ реакций с эталонами.

Структурный проектирования	метод	поиска	ошибок
----------------------------	-------	--------	--------

Основной принцип, лежащий в основе структурного алгоритма поиска ошибок проектирования следующий. Если в очередной КТ первого рода (на выходе ФЭ) результат ЭП отрицателен (полученная сигнатура не совпала с эталонной), то в область подозреваемых ошибок на очередном шаге алгоритма входят сам ФЭ и все его предшественники в I-графе. Если результат ЭП положителен (полученная сигнатура совпала с эталонной), то все предшественники предполагаются исправными, а подозреваемая ошибка произошла среди остальных ФЭ области подозреваемых ошибок предыдущего шага алгоритма. Условный алгоритм поиска ошибок проектирования строится при условии минимизации максимальной длины последовательности ЭП, что реализуется с применением стратегии половинного деления.			
--	--	--	--

HDL-код представляется в виде графовой модели, которая позволяет выполнять генерацию тестов. Построение теста выполняется путем активизации многоразрядных путей от ФЭ первого ранга до внешних выходов или определенных выходных КТ в I-графе. В качестве элементарных тестов для ФЭ используются предварительно вычисленные различающие последовательности [2]. Выбор сделан в пользу РПС, а не стандартных проверяющих тестов по нескольким причинам. Основной из них является то,

что в основе построения проверяющих тестов лежат понятия активизации пути, по которому неисправность транспортируется на внешний выход схемы. А это не всегда представляется возможным сделать, так как в HDL-коде функциональные неисправности могут быть замаскированы дальнейшими вычислениями таким образом, что ошибка не будет наблюдаться на внешнем выходе. Для декомпозиции исходного графа используются КТ, аналогичные дополнительным КТ при генерации тестов, которые позволяют «разбить» путь активизации и определить границы подграфов. Выбранные КТ будут выходами каждого из подграфов. Таким образом, ошибка проектирования транспортируется на внешний выход каждого подграфа.

На рисунке 6 представлена HDL-модель ЦУ, на примере которой покажем применение структурного алгоритма поиска ошибок в HDL-коде.

```
Entity SCH is
  Port (X1,X2,X3,X4,X5,X6: in bit;
        y1, y2: out bit);
End;
Architecture BEN of SCH is
Begin
  x6<='0';
  y1<=X1 nor (x2 nor x3);
  y2<=x6 nand (x5 nor (x4 and (x2
    nor x3)));
End;
```

Рисунок 6 – HDL-модель цифрового устройства

Рассмотрим этапы построения структурного алгоритма поиска ошибок проектирования.

1. HDL-модель ЦУ представляется в виде I-графа и выполняется его ранжирование путём нумерации выходов каждого ФЭ (рис. 7). На рис.7 номера ФЭ помечены красным цветом.

Приведенный граф обладает следующими особенностями:

- все операторные вершины (G1 – G11) являются КТ первого рода, т.е. эталоны в них известны;

- в графе выделено три выходных КТ (G8, G10, G11), которые являются внешними выходами при транспортировании РПС, что определено на этапе построения тестов;

- активизацию к вершине G11 невозможно выполнить из-за наличия константного нуля (операндовая вершина G6='0') на одном из входов ФЭ5 (оператор pand).

2. На основании ранжированного I-графа строится граф связей, вершинами которого являются операндовые вершины, а линии графа определяют связи между ними (рис. 8а).

Здесь и далее буква G применительно к номеру операндовой вершины не пишется, а указывается только ее номер. Двойной окружностью на графе а) помечены выходы транспортирования РПС, которые разбивают исходный граф на подграфы б) и в), приведенные на рис.8.

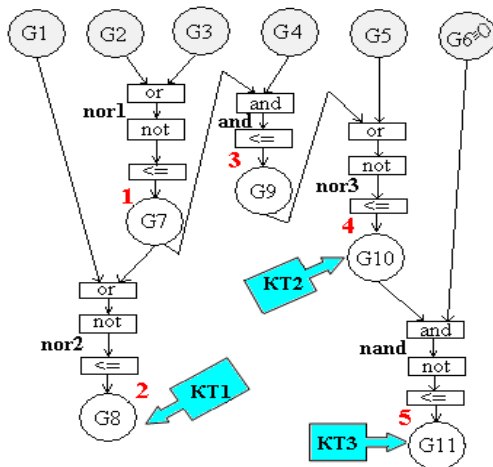


Рисунок 7 – Представление цифрового устройства в виде I-графа

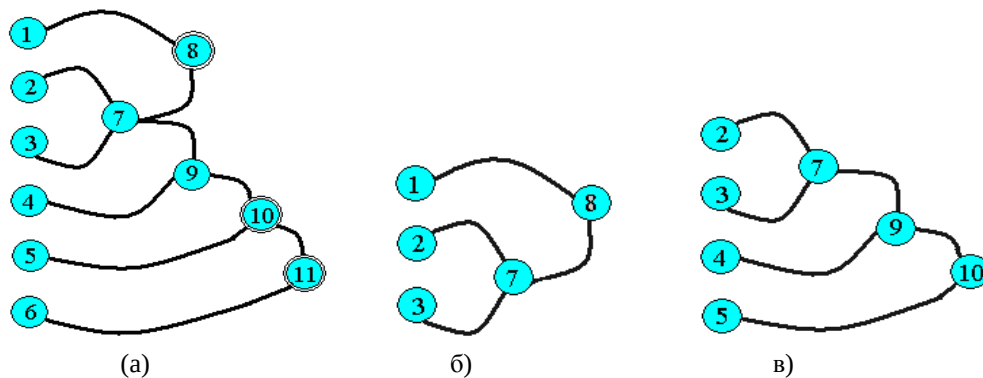


Рисунок 8 – Граф связей HDL-модели и его разбиение на подграфы

3. Для каждого из подграфов строится матрица достижимостей (МД), строки которой соответствуют предшественникам, а столбцы приемникам некоторой операндовой вершины главной диагонали. При построении МД единицами заполняется главная диагональ и те клетки в строках, которые являются предшественниками для операндовых вершин главной диагонали.

4. Для построения оптимального алгоритма поиска дефекта (относительно минимизации

максимальной последовательности ЭП) используется стратегия половинного деления и для нее вычисляется функция предпочтения при выборе очередной ЭП по формуле [3]:

$$f = \min |D_i \& M_j - D_i / 2|, \quad (3)$$

где D_i - текущее подмножество подозреваемых ошибок, M_j это j -я строка МД.

На рисунке 9 представлены матрицы достижимости с функциями предпочтения для подграфов б) и в) рисунка 8 на первом шаге построения алгоритмов поиска дефектов. Отметим, что

первоначальная оценка области подозреваемых ошибок для подграфа 8б) будет $D_i / 2 = 5 / 2 = 2,5$, а для подграфа 8в) $D_i / 2 = 7 / 2 = 3,5$ соответственно.

	1	2	3	7	8	f
1	1					1,5
2		1				1,5
3			1			1,5
7		1	1	1		0,5
8	1	1	1	1	1	2,5

а)

	2	3	4	5	7	9	10	f
2	1							2,5
3		1						2,5
4			1					2,5
5				1				2,5
7	1	1			1			0,5
9	1	1	1		1	1		1,5
10	1	1	1	1	1	1	1	3,5

б)

Рисунок 9 – Матрицы достижимости и функции предпочтения для подграфов б) и в)

На рисунке 10 представлены результирующие деревья поиска ошибок проектирования для подграфов б) и в) рисунка 8 соответственно. Отмеченные строки в

матрицах достижимости (рис.9) являются корневыми вершинами деревьев.

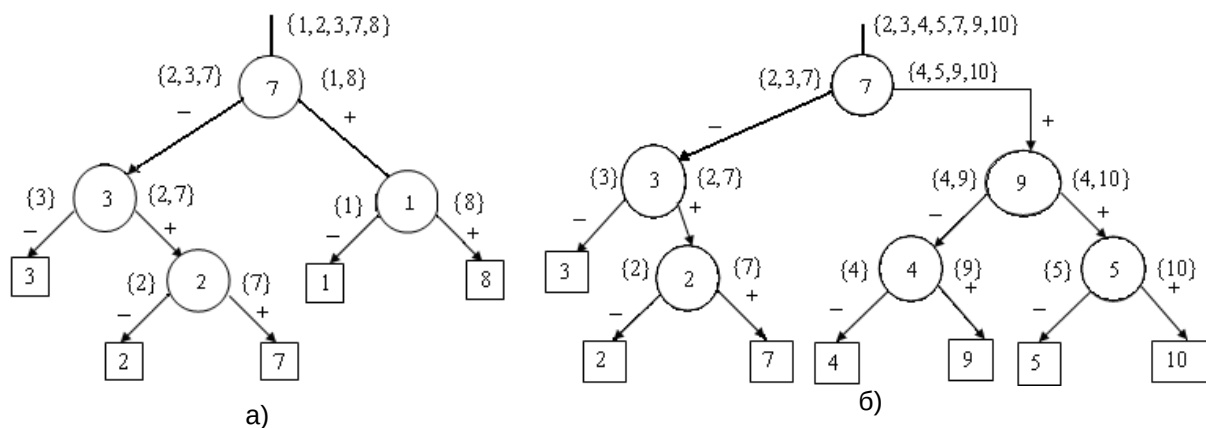


Рисунок 10 – Деревья поиска ошибок проектирования для подграфов б) и в)

ДЭ будет проводиться над каждым подграфом, на выходе которого была обнаружена ошибка. Например, если ошибка произошла внутри подграфа б), то, анализируя его, не нужно будет анализировать другие подграфы. Тем самым сокращается время проведения ДЭ и уменьшается длина алгоритма диагностирования. В дальнейшем предполагается пересечение МД всех подграфов для уменьшения временных затрат на поиск ошибки проектирования.

Область применения разработанного метода ограничивается количеством строк HDL-кода, для которого строится МД. Её размерность вычисляется $D = n \times (n + 1)$, где n – количество матрицы строк (число операторов в HDL-коде); $(n + 1)$ – количество столбцов (на 1 больше числа строк за счёт столбца функции предпочтения).

Таким образом, использование матрицы достижимости, как формы представления информации

при построении и реализации алгоритмов диагностирования, часто невозможно по причине высокой размерности матрицы. А значит, целесообразнее использовать предложенный метод только для HDL-моделей, содержащих ограниченное количество операторов. Для HDL-кода, содержащего 500 операторов, МД будет приблизительно 30 Кбайт. Отсюда можно сделать вывод, что обработка фрагментов кода, содержащего до 3000 операторов вполне реальна, а значит, разработанный метод приемлем для практического применения.

Выводы

В ходе преобразования спецификации на проектируемое ЦУ в модель на языке описания аппаратуры разработчик может внести ошибки неалгоритмического характера, связанные с «человеческим фактором». Методы поиска дефектов (ошибок проектирования) на этапе верификации HDL-модели позволяют не только обнаружить факт существования ошибки проектирования, но и точно определить место её возникновения (локализовать дефект) в HDL-коде. Результаты применения функционального и структурного методов поиска ошибок проектирования к HDL-моделям, преобразованным в графовые модели, показали следующее.

1) Функциональный метод в ходе верификации позволяет обнаружить место возникновения ошибки проектирования в неструктурированном коде путём сравнения полученных результатов с заранее полученными реакциями на все возможные ошибки проектирования. Метод ограничен размерностью ТФН и позволяет анализировать HDL-модели до 1000 операторов.

2) Структурный метод локализации ошибок проектирования на основе матриц достижимости, построенных по графовой модели, применим для одновыходовых моделей в качестве самостоятельного метода диагностирования, либо как следующий шаг увеличения глубины диагностирования для моделей многovýchодовых ЦУ. Метод ограничен размерностью матриц достижимости и позволяет анализировать HDL-модели до 3000 операторов.

Литература

1. Шкиль А.С. Модели описаний цифровых устройств для диагностирования / А.С. Шкиль, Е.Е. Сыревич, С. Альмадхоун // Вестник Херсонского государственного технического университета. – 2010. – №2 (38). – С. 258-265.
2. Шкиль А.С. Тестовая верификация поведенческих языковых моделей цифровых устройств / А.С. Шкиль, Е.Е. Сыревич, А.Л. Карасев,

Д.И. Чегликов // Автоматизированные системы управления и приборы автоматики. – 2006. – Вып. 134. – С. 4 – 12.

3. Шкиль А.С. Методы поиска ошибок проектирования в HDL-коде / А.С. Шкиль, Е.Е. Сыревич, Д.Е. Кучеренко, Г.П. Фастовец // Радиоэлектроника и информатика. – 2008 – №. 3. – С. 47–53.
4. Сыревич Е.Е. Применение методов поиска дефектов при верификации моделей цифровых устройств на VHDL / Е.Е. Сыревич, В.Б. Таранов, Д.И. Чегликов, Д.Е. Зинченко, А.Л. Карасев // Вестник Херсонского государственного технического университета. – 2008. – № 1(30). – С.335-341.

Шкиль О.С., Сыревич Е.Ю., Альмадхоун С., Фастовец Г.П. Структурне та функціональне діагностування HDL-моделей цифрових пристроїв в САПР РЕА. Розглянуто загальну структуру САПР РЕА на етапі функціональної верифікації та пошуку помилок проектування в моделях цифрових пристроїв на мовах опису апаратури. Розглянуто такі методи пошуку помилок проектування : функціональний метод на основі таблиць функцій несправностей та структурний метод на основі графової моделі для мовних описів стилю «потік даних». Приведені оцінки ефективності вказаних методів.

Ключові слова : HDL-модель, верифікація, методи пошуку помилок проектування, діагностичний експеримент

Shkil Al. Ser., Syrevitch Yev.Yef., Almadhoun S., Fastovets G.P. Structural and functional diagnostics of HDL-models of digital devices in CAD of Radioelectronic Hardware.

The general structure of CAD of Radioelectronic Hardware at the stage of functional verification and design error localization in HDL-models of digital devices in languages of apparatus description has been considered. Functional method on the basis of fault functions tables and structural method based on graph model for language descriptions of “data flow” style have been considered as the methods of design error localization. Performance evaluation of the given methods has been presented.

Key words: HDL-model, design error localization methods, diagnostic experiment

Рецензент д.т.н., профессор Алёшин Г.В. (УкрГАЗТ)

Поступила 18.03.2013г.