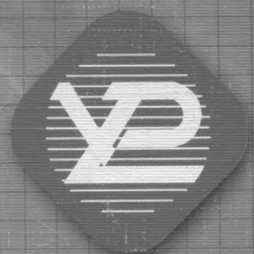
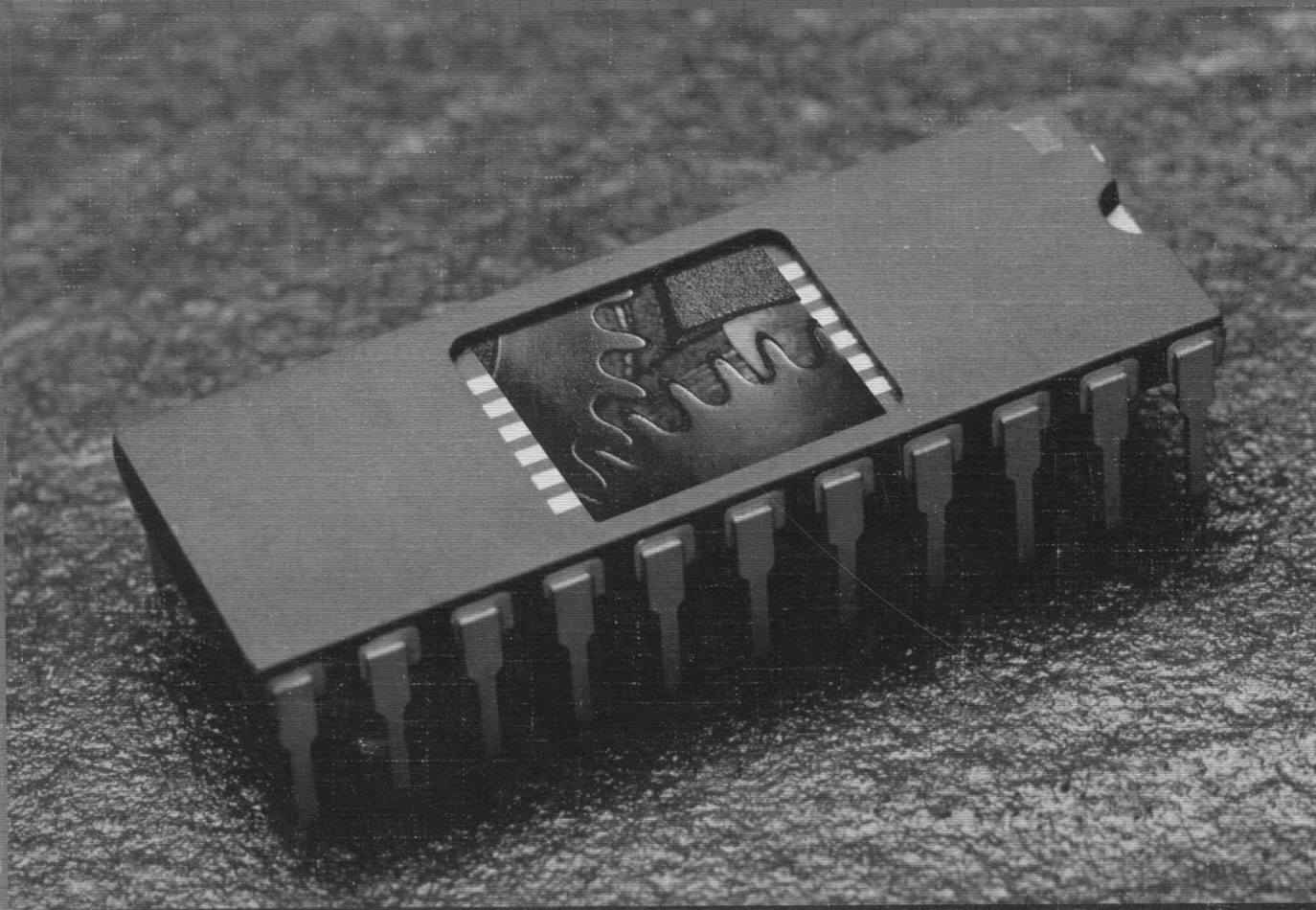


АВТОМАТИЗИРОВАННЫЕ СИСТЕМЫ УПРАВЛЕНИЯ И ПРИБОРЫ АВТОМАТИКИ



'2001
124

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ

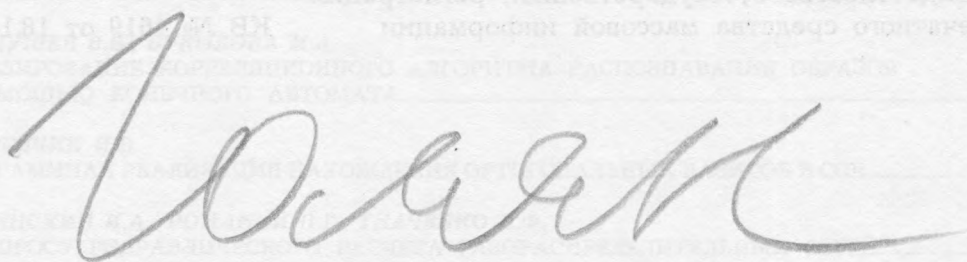
ХАРЬКОВСКИЙ НАЦИОНАЛЬНЫЙ
УНИВЕРСИТЕТ РАДИОЭЛЕКТРОНИКИ

АВТОМАТИЗИРОВАННЫЕ СИСТЕМЫ УПРАВЛЕНИЯ И ПРИБОРЫ АВТОМАТИКИ

Всеукраинский межведомственный
научно-технический сборник

Основен в 1965 г.

Выпуск 124



Харьков

2003

СОДЕРЖАНИЕ

НАТАЛУХА Ю.В., АФАНАСЬЕВ В.А., КЛИМОВА Н.П. ОБ ОДНОМ КРИТЕРИИ ПРОВЕРКИ ЛИНЕЙНОЙ НЕЗАВИСИМОСТИ ФУНКЦИЙ.....	4
ЛЕЩИНСКАЯ Е.Л. ИДЕНТИФИКАЦИЯ СЕМАНТИЧЕСКИХ ОШИБОК ПРИ СИНТАКСИЧЕСКОМ АНАЛИЗЕ.....	8
ТИХОНОВ В.А., ЗУБКОВ О.В., СИДОРОВ Г.И. ГЕНЕРИРОВАНИЕ КВАДРАТУРНЫХ СОСТАВЛЯЮЩИХ СЛУЧАЙНОГО УЗКОПОЛОСНОГО АНАЛИТИЧЕСКОГО СИГНАЛА НА НЕСУЩЕЙ ЧАСТОТЕ ПО ЗАДАННЫМ ПАРАМЕТРАМ СПЕКТРАЛЬНОЙ ПЛОТНОСТИ МОЩНОСТИ.....	18
ДОРОВСКОЙ В.А. ЭНТРОПИЙНАЯ МОДЕЛЬ ДИАЛОГОВОГО ПОВЕДЕНИЯ ПАРТНЕРОВ.....	23
ГОЛЯН В.В. VHDL-МОДЕЛЬ ПРОЕКТИРОВАНИЯ ПРЕОБРАЗОВАТЕЛЕЙ КОДОВ.....	27
КАНЦЕДАЛ С.А., КОСТИКОВА М.В. О ДВУХ АЛГОРИТМАХ ДЛЯ РЕШЕНИЯ ОБЩЕЙ ЗАДАЧИ ТЕОРИИ РАСПИСАНИЙ.....	31
СОРОЧАН М.В. МОДЕЛИРОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ ПРИ ПОСТРОЕНИИ ТРЕБОВАНИЙ К ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ.....	37
ГВОЗДИНСКИЙ А.Н., ЯКИМОВА Н.А. ЛОГИКО-АЛГЕБРАИЧЕСКИЕ МОДЕЛИ СИСТЕМ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ.....	43
АЛИПОВ Н.В., АЛИПОВ И.Н., ХИЛЬ М.И., РЕВЕЗЮК Л.Н. ПРИМЕРЫ ПОСТРОЕНИЯ ПОСЛЕДОВАТЕЛЬНЫХ АЛГОРИТМОВ ПОИСКА ТОЧКИ С ХАРАКТЕРНЫМ ПРИЗНАКОМ, ПОМЕХОУСТОЙЧИВЫХ К СИММЕТРИЧНЫМ НЕРЕГУЛЯРНЫМ ВИРТУАЛЬНЫМ ПОСЛЕДОВАТЕЛЬНОСТЯМ.....	47
ДЮВКО Г.Ф., ХАРЧЕНКО С.Л. КОМПОНЕНТНАЯ ТЕХНОЛОГИЯ ПРОГРАММИРОВАНИЯ ДЛЯ ОПИСАНИЯ МОДЕЛЕЙ ЦИФРОВЫХ УСТРОЙСТВ.....	55
СИНОТИН А.М. ИССЛЕДОВАНИЕ ВЛИЯНИЯ ЭЛЕМЕНТОВ РАДИОЭЛЕКТРОННОГО АППАРАТА НА ЭФФЕКТИВНУЮ ТЕПЛОПРОВОДНОСТЬ В ОСНОВНЫХ НАПРАВЛЕНИЯХ.....	61
СИТНИКОВ Д.Э., ВИЛЬЧИНСКАЯ О.С., КРАВЕЦ Н.С. АЛГЕБРАИЧЕСКИЙ МЕТОД НАХОЖДЕНИЯ АППРОКСИМАЦИЙ ПРИБЛИЖЕННЫХ МНОЖЕСТВ И ФОРМИРОВАНИЕ НА ИХ ОСНОВЕ ЛОГИЧЕСКИХ ПРАВИЛ.....	66
ДЮВКО Г.Ф., ПОЛУНИН Р.А. РАЗРАБОТКА БАЗ ЗНАНИЙ ДЛЯ ПРОВЕДЕНИЯ ЛОГИЧЕСКОГО КОНТРОЛЯ В КОМПЬЮТЕРНЫХ ОБУЧАЮЩИХ СИСТЕМАХ.....	72
КЛИМУШЕВ В.В., БУКОЛОВА М.А. МОДЕЛИРОВАНИЕ КОРРЕЛЯЦИОННОГО АЛГОРИТМА РАСПОЗНАВАНИЯ ОБРАЗОВ С ПОМОЩЬЮ КОНЕЧНОГО АВТОМАТА.....	80
КУПРЕЙЧИК И.В. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ НАХОЖДЕНИЯ ОРТОГОНАЛЬНЫХ БАЗИСОВ В СОК.....	87
БОЖИНСКИЙ И.А., РОМАНИЙ П.Г., ТКАЧЕНКО В.Ф. К ВОПРОСУ ГИДРАВЛИЧЕСКОГО РАСЧЕТА ГАЗОРАСПРЕДЕЛИТЕЛЬНЫХ СЕТЕЙ.....	91
ПОЛОНСКИЙ А.Д. ИНВАРИАНТНАЯ АЛГЕБРА РАНГОВЫХ ПРЕДИКАТОВ ДЛЯ СИНТЕЗА КЛАССИФИКАТОРОВ АНАЛОГОВЫХ СИГНАЛОВ.....	99
ГОЛОВАШИЧ С.А., ПОНОМАРЕВ Д.В. ПОТОЧНЫЙ ШИФР VMGL И ОСОБЕННОСТИ ЕГО ПРОГРАММНОЙ РЕАЛИЗАЦИИ.....	107
АНПИЛГОВ Е.М., ДЗЮНДЗЮК В.В., СТЫЦЕНКО Т.Е. ИНТЕРПРЕТАЦИЯ И АНАЛИЗ НАСЛЕДУЕМЫХ РАЗЛИЧНЫХ ФАКТОРОВ НА ОПЕРАЦИЯХ ИЗГОТОВЛЕНИЯ СИСТЕМЫ.....	113
РЕФЕРАТИ	116

КОМПОНЕНТНАЯ ТЕХНОЛОГИЯ ПРОГРАММИРОВАНИЯ ДЛЯ ОПИСАНИЯ МОДЕЛЕЙ ЦИФРОВЫХ УСТРОЙСТВ

Рассматриваются проблемы автоматизации процесса проектирования сложных цифровых устройств на основе применения компонентной технологии проектирования, базирующейся на анализе модели объекта, представленной в виде иерархии декомпозиций моделей до неделимого компонента; язык формальной записи полученной модели цифрового устройства, пригодной для преобразования в код программы VHDL, описывающей структурную и поведенческую модель цифрового устройства; механизмы синтаксического и семантического анализа формальной записи модели объекта на основе применения базы знаний.

1. Введение

Разработка современных цифровых устройств, из которых состоят вычислительные машины различного назначения, не обходится без применения систем автоматизации проектирования. При этом решаются задачи разработки и отладки алгоритмов функционирования сложных цифровых устройств (ЦУ), отладки программного и микропрограммного обеспечения на моделях, разработки и отладки регистровых схем, отладки поведения проектируемой системы в окружающей среде. В настоящее время разработано множество языков описания ЦУ, которые позволяют моделировать различные аспекты структуры, свойств и функционирования ЦУ. Одним из них является VHDL-язык описания аппаратуры. VHDL поддерживается соответствующей системой моделирования, благодаря чему модель, представленная на VHDL, может использоваться для тестирования работоспособности создаваемой системы. Так как VHDL является, по сути, языком программирования, описывающим структуру объекта и алгоритмы его функционирования, разработка программ на этом языке имеет те же проблемы, что и в языках программирования для разработки программного обеспечения общего назначения. Построение программ на VHDL является сложной и трудоёмкой задачей для программистов. Поэтому есть необходимость разрабатывать системы, позволяющие в определенной мере автоматизировать создание кода на VHDL.

2. Актуальность проблемы

Сокращение сроков разработки и уменьшение затрат рабочего времени на цикл «проектирование – тестирование – внедрение» важно как с точки зрения сокращения финансовых затрат, так и для продвижения продукции на рынке товаров и услуг.

3. Постановка задачи

Рассмотрим подход к автоматизации создания кода на языке описания аппаратуры с точки зрения объектно-ориентированной и компонентной технологий.

При проектировании ЦУ разработчики используют понятие иерархии абстрактных представлений. Абстрактное представление может относиться к структуре, предусматривающей описание компонента как совокупность взаимосвязанных компонентов более низкого уровня, или к поведению, которое описывается компонентами по зависимостям вход/выход при помощи некоторой процедуры. Таким образом, весьма естественно можно описать наследование (если оно существует) и построить компонентную технологию на базе стандартных компонентов. Для компонентной технологии формулируется набор правил, которые позволяют строить программы на VHDL из компонентов,

определённых визуально. Реализация компонентной технологии осуществляется с помощью интеллектуальных средств: уровень представления модели ЦУ может быть преобразован в другой уровень посредством баз знаний в форме продуктов, дизъюнктов, графовых представлений, процедур.

Современной тенденцией в разработке прикладного программного обеспечения является переход к «сборке» прикладных программ из готовых и относительно универсальных фрагментов-компонентов, которые функционируют под управлением компонентных сред. Автоматическая генерация кода при этом осуществляется средствами визуального программирования и специализированными трансляторами. Компонентная среда обеспечивает оптимальное использование ресурсов компьютера и сетевого оборудования, взаимодействие прикладных объектов с различными операционными системами, быстрое создание гибких и эффективных интерфейсов с пользователями.

Компонент-это развитие концепции объекта, вернее класса, который и является его формальным определением. Для компонента характерно наличие объявлений и реализаций интерфейсов, определяющих функциональность и применимость компонентов в рамках рассматриваемой компонентной модели.

Последняя определяется как набор правил, позволяющих строить программы из компонентов. Компонентные модели являются основой визуального программирования. Существующие компонентные модели привязаны либо к конкретным программным продуктам (DELPHI, C++ BUILDER), либо к операционным системам (WINDOWS-ACTIVE X), либо к языку программирования (JAVA BEANS и EJB).

Любой проектировщик ЦУ сталкивается в своей работе с большим количеством информации и рискует «утонуть» в деталях. Чтобы избежать такой опасности, необходимо ограничивать количество информации, анализируемой в конкретный момент времени, работая на соответствующем уровне абстракции. Типичные уровни абстракции для ЦУ: процессоры-памяти-коммутаторы, микросхемный, регистровый, вентильный, схемный, кремниевый. Каждый из этих уровней описывается структурно и поведенчески. Структура представляется набором и указанием соединений простейших элементов (примитивов). В качестве примитивов могут выступать как набор простейших, не разлагающихся на более мелкие единицы элементов, так и структуры.

Структурированный процесс автоматизированного проектирования ЦУ осуществляется с помощью инструментальных средств проектирования, одним из которых является VHDL [1]. VHDL предлагает тщательно разработанные конструкции для структурного (статического) и поведенческого (функционального) представления проекта, а также средства для документирования проектов.

Традиционный подход к проектированию ЦУ представляет собой анализ проектировщиком сочетания словесных описаний, блок-схем, временных диаграмм, логических схем и формул. Автоматизированный подход предполагает ввод проекта в автоматизированное рабочее место проектировщика в виде файла на языке описания аппаратуры или в виде некоторой схемы, или в комбинированном варианте. В традиционном подходе проектировщик относительно легко создавал схемы, словесные описания, формулы, диаграммы. Основные затраты труда возникали при ручном моделировании и анализе исходной информации. При автоматизированном подходе создание описаний на VHDL оказалось весьма трудоемкой, зачастую рутинной задачей. Если учесть, что конструкции VHDL проектировщик получает из традиционной входной информации, то возникает естественное желание автоматизировать создание кода на VHDL, исходя из схем, диаграмм, словесных описаний, формул.

4. Методология проектирования ЦУ с применением компонентной технологии

Ниже предлагается подход к построению кода на VHDL с использованием компонентной технологии, реализованной средствами искусственного интеллекта.

На рис.1 представлена общая схема преобразования описательных конструкций проекта в код VHDL.



Рис. 1. Общая схема получения кода на VHDL

Рассмотрим элементы этой схемы. Исходные данные и цели проекта представляются посредством описаний проекта на языке разработчика. Отдельные части описания можно трактовать как компоненты, обладающие своим интерфейсом и внутренним содержанием. Компонентам входного языка соответствуют выходные компоненты, представленные конструкциями языка VHDL. Компонент на VHDL имеет два типа описаний: описание интерфейса и описание архитектурных тел.

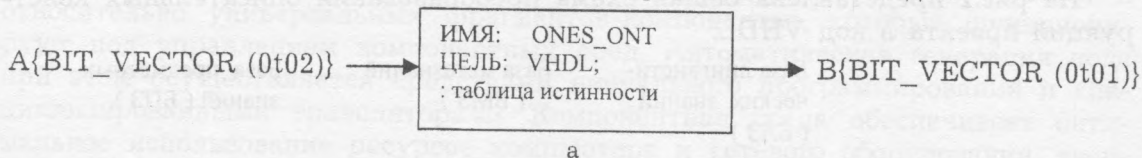
Чтобы преобразовать вход в выход, на рис. 1 используются три блока. Распознавание входных языковых конструкций осуществляется методами трансляции [2] с учетом лингвистических знаний, находящихся в БЛЗ. Традиционного подхода с применением контекстно-свободных грамматик здесь недостаточно, необходимо использовать специальные типы грамматик, которые превращают систему продукций в объектную базу знаний, а вывод в грамматике – в вывод на знаниях. Результатом работы блока распознавания является промежуточная структура, представляющая собой синтаксическое дерево (или семантическую сеть), которое задает в явном виде входные компоненты, связи между ними, а также их характеристики.

Синтаксическое дерево обрабатывается следующим блоком, чтобы преобразовать содержащиеся в нём знания из объектной формы в логическую сформировать существующие факты и цели в логической форме. Преобразования в этом блоке происходят под управлением знаний из БМЗ, которые представляются в продукционной форме [3,4].

Последним блоком служит блок логического вывода [3]. Его знания представлены в виде хорновских дизъюнктов, литералы которых могут иметь термы, выраженные процедурами. Такое представление знаний позволяет соединить декларативные и процедурные знания в рамках логической модели, но накладывает специальные ограничения на процедуру логического вывода. Код на VHDL возникает как результат унификации при логическом выводе.

Продемонстрируем на примере, как работает схема, представленная на рис. 1. Пусть имеется объект проекта (компонент), который производит подсчёт числа единиц во входном битовом векторе, имеющем длину, равную 3. Выходом этого объекта является битовый вектор длины 2.

Рассмотрим сначала вариант, где поведение этого объекта описывается чистым сопоставлением выхода входу. Предположим, что рассматриваемый объект проекта представлен описательной конструкцией на рис. 2. Здесь изображен интерфейс объекта в виде визуального компонента, описание поведения этого объекта дано в форме таблицы истинности. Указана также цель проектирования - получение компонента, представленного кодом VHDL.



ONES_CNT:	A[8,3] (0,0,0,	B[8,2] (0,0,
	0,0,1,	0,1,
	0,1,0,	0,1,
	0,1,1,	1,0,
	1,0,0,	0,1,
	1,0,1,	1,0,
	1,1,0,	1,0,
	1,1,1),	1,1);

б

Рис. 2. Описание счётчика единиц на языке разработчика: а - имя, цель, интерфейсы компонента; б - поведение компонента

В БПЗ знания о преобразовании описания в код выражены одним хорновским дизъюнктом:

код_ти(f(Имя,Имя1,Имя2,Тип1,Тип2)):

-vhdl_ти(Имя),вход(Имя1,Тип1),выход(Имя2,Тип2) (1)

где f - процедура формирования кода VHDL из таблицы истинности по заданным параметрам. Литералы дизъюнкта (1) выражают: vhdl_ти - необходимо сформировать код по таблице истинности и имя кодового компонента есть Имя; вход - входной сигнал компонента с именем Имя1 типа Тип1; выход - выходной сигнал с именем Имя2 типа Тип2; код_ти - представить проектируемый компонент на VHDL посредством функции f.

БЛЗ представлена продукциями грамматики. Поскольку рассматриваемый случай простой, используется контекстно-свободная грамматика. Для экономии места закодируем нетерминальные символы грамматики целыми числами в соответствии с приведенной ниже таблицей.

Система продукций для рассматриваемого случая имеет вид:

1.<1> → <2><3>

2.<2> → <4><5><6><7>

3.<4> → <8><9>

4.<5> → <13>

5.<6> → <13>

6.<7> → <10><9>

7.<8> → <13>

8.<9> → BIT_VECTOR(<11>)

9.<10> → <13>

10.<11> → <12>to<12>

11.<12> → 0|1|2|3

12.<13> → ONES_CNT | VHDL_TI | A | B

13.<3> → <14><15>

14.<14> → <8>[<12>,<12>](<16>)

15.<15> → <10>[<12>,<12>](<16>)

16.<16> → <12> | <12>,<16>

Нотация нетерминала	Что обозначает нетерминал	Код нетерминала
<описание>	Определение компонента и цели	<1>
<интерфейс>	Связи компонента	<2>
<поведение>	Поведение компонента	<3>
<вход>	Входной сигнал компонента	<4>
<имя>	Имя компонента	<5>
<цель>	Как производится преобразование компонента и в какой выход	<6>
<выход>	Выходной сигнал компонента	<7>
<имя вх>	Имя входного сигнала	<8>
<описание интер.>	Тип и параметры сигналов интерфейса	<9>
<имя вых>	Имя выходного сигнала	<10>
<индексы>	Параметры сигналов интерфейса	<11>
<число>	Числовое значение параметра сигнала интерфейса	<12>
<идентификатор>	Имена составных частей проекта	<13>
<вход1>	Тип и значение входа	<14>
<выход1>	Тип и значение выхода	<15>
<последовательность чисел>	Матрицы значений входа и выхода	<16>

Выполнив синтаксический анализ языковой конструкции, представленной на рис. 1, получим синтаксическое дерево.

С продукциями 14 и 15 должна быть связана семантическая процедура, формирующая двумерные массивы таблицы истинности и механизмы доступа к ним.

БМЗ должна иметь следующие правила:

- если значение нетерминала <цель> есть VHDL_TI, то формировать предикат « не код_ti(X)»;
- если значение нетерминала <имя> - а, формировать предикат “vhdl_ti(a)”;
- если значение нетерминала <вход1> - a[p,q]..., формировать предикат «вход(a, array)»;
- если значение нетерминала <выход1> - a[p,q]...,формировать предикат “выход(a,array)”.

Как результат работы БМЗ получаются следующие предикаты:

- цель «не код_ti (X)», где X - искомая переменная, т.е. программа на VHDL;
- факты «vhdl_ti (ONES_CNT)», «вход (A,array)», «выход (B, array)».

Так как знания БПЗ таковы, что позитивный литерал имеет функцию в качестве терма, необходимо модифицировать процесс логического вывода в БПЗ. К моменту применения резолюции к позитивному литералу все его переменные должны быть конкретизованными, что требует первоначального применения резолюций ко всем литералам правой части дизъюнкта (1). Поэтому по цели «неС» ищется дизъюнкт C:-L₁,...,L_k, где полностью конкретизованные L₁,...,L_k находятся в фактах. Но резольвента по C и “неС» не создаётся, а откладывается до тех пор, пока не будут получены все резольвенты по L₁,...,L_k. Если этот процесс окажется успешным, унифицируются C и «неС» с вычислением значения функции f. Теперь несложно составить алгоритм, реализующий функцию f и выдающий в качестве результата код счётчика единиц на VHDL в виде:

```
entity CNT_ONS is
  port (A: in BIT_VECTOR (0 to 2); B: out BIT_VECTOR (0 to 1));
end CNT_ONES;
architecture POVEDENIE of CNT_ONES is
begin
  process (A)
```

```
variable NUM: INTEGER range 0 to 3;
```

```
begin
```

```
NUM:=0;
```

```
for k in 0 to 2 loop
```

```
if A(k)='1' then
```

```
NUM:=NUM+1
```

```
end if;
```

```
end loop;
```

```
case NUM is
```

```
when 0 => B <='00';
```

```
when 1 => B <='01';
```

```
when 2 => B <='10';
```

```
when 3 => B <='11';
```

```
ende case;
```

```
end process.
```

Пусть теперь проектирование счётчика единиц рассматривается с уровня логических формул. Применив карты Карно для обоих выходов B[0] и B[1], проектировщик получает:

$$\begin{aligned} B[1] &= A[1]*A[0]+A[2]*A[0]+A[2]*A[1], \\ B[0] &= A[2]*\text{not}A[1]*\text{not}A[0]+\text{not}A[2]*\text{not}A[1]*A[0]+A[2]*A[1]*A[0]+\text{not}A[2]*A[1]*\text{not}A[0]. \end{aligned} \quad (2)$$

Предположим, что в этом случае проект описан языковой конструкцией, приведенной на рис. 3.

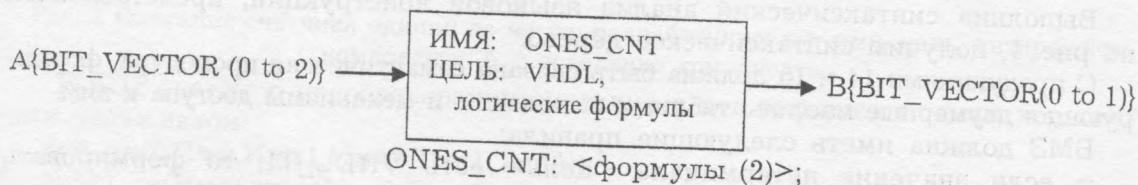


Рис. 3. Описание проекта логическими формулами

Преобразовав формулы (2) во время трансляции этой языковой конструкции в польскую инверсную запись, несложно составить процедуру f, которая создаст код VHDL:

```
entity ONES_CNT is
```

```
port(A:in BIT_VECTOR(0 to 2);B:out BIT_VECTOR(0 to 1));
```

```
end ONES_CNT;
```

```
architecture LOG_POVED of ONES_CNT is
```

```
begin
```

```
B[1]<=(A(1) and A(0))or (A(2) and A(0)) or (A(2) and A(1));
```

```
B[0]<=(A(2) and not A(1) and not A(0)) or
```

```
(not A(2) and not A(1) and A(0)) or
```

```
(A(2) and A(1) and A(0)) or
```

```
(not A(2) and A(1) and not A(0)) ;
```

```
end LOG_POVED;
```

5. Выводы

Внедрение компонентной технологии при проектировании цифровых систем позволит сократить сроки разработки в два раза. Это достигается путем снижения трудоемкости работ на всех этапах проектирования и реализации системы. Компонентный подход позволит локализовать и устранить противоречия в техническом задании, исключить влияние человеческого фактора на конечный код VHDL, обеспечить высокую надежность проектируемой системы с помощью автоматического анализа и тестирования предполагаемых схемных решений на каждом уровне декомпозиции системы.

Список литературы: 1. Армстронг Дж. Р. Моделирование цифровых систем на языке VHDL: Пер. с англ. / М.: Мир, 1992. 175с. 2. Хантер Робин. Основные концепции компиляторов: Пер. с англ. М.: Издательский дом «Вильямс», 2002. 256с. 3. Тейз, Грибомон П., Луи Ж., Снийерс Д., Водон П., Гоше П., Грегуар Э., Санчес Э., Дельсарт Ф. Логический подход к искусственному интеллекту: от классической логики к логическому программированию: Пер. с франц. М.: Мир, 1990. 432с. 4. Базы знаний интеллектуальных систем / Т.А. Гаврилова, В.Ф. Хорошковский. СПб: Питер, 2000. 384с.

Поступила в редколлегию 17.05.2003

Дюбко Геннадий Федорович, канд. техн. наук, профессор кафедры ПОЭВМ ХНУРЭ. Адрес: Украина, 61046, Харьков, пр. Гагарина, 38, кв.70, тел. раб.: 70-21-446, тел. дом.: 27-13-26.

Харченко Сергей Леонидович, ассистент кафедры ПОЭВМ ХНУРЭ. Адрес: Украина, 61045, Харьков, ул. Аральская, 5, кв.2, тел. 70-21-446.

УДК 658.51.011.56

А.М. СИНОТИН

ИССЛЕДОВАНИЕ ВЛИЯНИЯ ЭЛЕМЕНТОВ РАДИОЭЛЕКТРОННОГО АППАРАТА НА ЭФФЕКТИВНУЮ ТЕПЛОПРОВОДНОСТЬ В ОСНОВНЫХ НАПРАВЛЕНИЯХ

Исследуется влияние радиоэлектронных элементов, распределенных в пространстве в определенной последовательности (системы дальнего порядка), на значение эффективных теплопроводностей в основных направлениях радиоэлектронного аппарата.

1. Актуальность

Современные технические устройства различного назначения все более насыщаются радиоэлектронной аппаратурой. Это ведет к увеличению числа входящих в нее элементов, усложнению РЭА, к необходимости дальнейшего совершенствования элементной базы при постоянном стремлении максимально снизить габаритные размеры аппаратуры.

Энергетический коэффициент полезного действия радиоэлементов, как правило, невелик, и значительная доля энергии питания (до 90%) превращается в тепловую энергию, что приводит к перегреву аппаратуры. Поэтому разработчику очень важно еще на стадии проектирования выбрать такие конструктивные параметры, которые смогли бы обеспечить прибору нормальный тепловой режим в процессе его эксплуатации. Но для расчета температурного поля аппарата необходимо знание эффективной теплопроводности в основных направлениях. Существующие методы [1,2] не учитывают влияния элементов на общую эффективную теплопроводность, что приводит к значительным погрешностям расчета температурного поля аппарата. Да и сам радиоэлектронный элемент зачастую представляет сложную систему, состоящую из различных материалов по теплопроводности, значения эффективной теплопроводности которого не описываются в техническом паспорте и соответствующей литературе.

2. Цель исследования

При анализе температурных полей сложных систем (например, радиоэлектронный аппарат) приходится иметь дело с анизотропными телами, тепловые свойства которых зависят от направления. На рис. 1 представлена простейшая тепловая модель аппарата (нагретая зона (ядро) 1, зазор 2, корпус 3), где под нагретой зоной понимается зона, занимаемая деталями (рис. 2).

Целью исследования является выявление влияния элементов радиоэлектронного аппарата на общую эффективную теплопроводность в основных направлениях.

Для преподавателей университетов,
научных работников, аспирантов

Тематика:

- **Компьютерная инженерия**
- **Математическое моделирование**
- **Оптимизация и процессы управления**
- **Автоматизация проектирования и диагностика**
- **Информационные интеллектуальные системы**
- **Проектирование интегральных схем и микросистем**
- **Компьютерные технологии в образовании**

Адрес редакции: 61166, Харьков, просп. Ленина, 14,
ХНУРЭ, тел. 40-93-26, e-mail: ri@kture.kharkov.ua