**BC/NW 2016 № 2 (29):11.1**

**РАЗРАБОТКА МОДУЛЯ РАСПОЗНАВАНИЯ СЛОВ ЯЗЫКА VHDL ДЛЯ ТРАНСЛЯТОРА С ЯЗЫКА VHDL НА ЯЗЫК SYSTEMC»**

Кольцов А.А.

Целью работы является разработка и реализация алгоритма лексического анализа моделей цифровых устройств на языке VHDL.

Основой алгоритма лексического анализа является прочтение исходного текста построчно, определение в строке слов языка VHDL и определение категории в которую эти слова входят.

Алгоритм прикладной программы был реализован в среде разработки Borland Delphi 7.

**Проект и структура описания объектов в VHDL.**

***Структура проекта***

Проект в VHDL определяется как совокупность связанных проектных пакетов. Проектными пакетами (design unit) называются независимые (external) фрагменты описаний, которые можно независимо анализировать компилятором и помещать в рабочую библиотеку проекта (Work).

Проектными пакетами могут быть:

• объявление интерфейса объекта проекта (entity);

• объявление архитектуры (architecture);

• объявление конфигурации (configuration);

• объявление интерфейса пакета (package);

• объявление тела пакета (package body).

Можно выделить две категории модулей проекта: первичные и вторичные. К первичным относятся объявления пакета, объекта проекта, конфигурации, к вторичным — объявление архитектуры, тела пакета. Файл, в котором размещаются один или несколько модулей проекта, называется файлом проекта (design file).

Все проанализированные модули помещаются в библиотеку проекта (design library) и становятся библиотечными модулями (library unit). Существует два класса библиотек проекта: рабочие библиотеки и библиотеки ресурсов. Рабочая библиотека — это библиотека Work, с которой в данном сеансе работает пользователь и в которую помещается пакет, полученный в результате анализа пакета проекта. Библиотека ресурсов — это библиотека, содержащая библиотечные модули, используемые в анализируемом модуле проекта. В каждый момент времени пользователь работает с одной рабочей библиотекой и произвольным количеством библиотек ресурсов.

***Описание объектов проекта***

Полное описание модели объекта проекта состоит из следующих частей:

а) описание интерфейса объекта проекта (entity), включающее:

— Port (списки входных и выходных сигналов);

— Generic (настраиваемые параметры модели);

б) описание архитектуры объекта проекта (ARCHITECTURE), включающее:

— объявление переменных и дополнительных (внутренних) сигналов;

— операторную часть, представляющую собой описание объекта проекта на структурном или поведенческом уровне;

в) (только для структурной формы описания) описание конфигурации (configuration), задающей подключение библиотеки моделей элементов и выборку их в качестве компонентов структуры.

Интерфейс объектов проекта описывается при помощи ключевого слова entity.

В VHDL существуют два основных уровня описания архитектуры объектов — поведенческий и структурный

В данном курсовом проекте будет рассматриваться только содержимое файлов .vhd , являющихся моделями цифровых устройств.

Поскольку в данном курсовом проекте рассматривается только модуль распознавания слов анализатора кода, то код модели или псевдокод языка будет разбит на следующие лексические категории:

* Служебные слова языка VHDL;
* Типы данных;
* Структурные единицы кода (переменные, константы и функции);
* Комментарии

**Разработка программы**

***Пользовательский интерфейс***

Программы написана с помощью языка Pascal в среде разработки Borland Delphi 7 и представляет собой исполняемый файл Windows Forms, в котором реализованы диалог открытия файлов .vhd , лексический анализ открытого файла с выводом результата на экран, а так же возможность сохранить результат анализа в файле .txt посредством диалога сохранения файла. Каждая из перечисленных функций реализуется путем нажатия на соответсвующую кнопку.

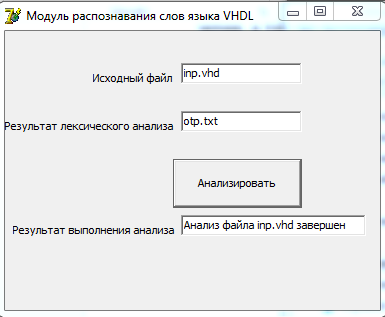


Рис.1 . Внешний вид окна интерфейса программы.

Таблица 1 Элементы управления программы.

|  |  |  |
| --- | --- | --- |
| Имя элемента управления на экране | Имя элемента управления в коде | Назначение |
| «Анализировать» | Analyze | Начало обработки файла, введенного в поле VHDLFile и вывод результата анализа в файл, введенный в поле ResultFIle |

По нажатию кнопки «Анализировать» проверяется существование файла, указаного в верхней текстовой форме. Путь к файлу загружается автоматически, но пользователь может изменить этот путь вручную.

В случае если файл не существует, то выдается сообщение об этом. В случае если файл существует, начинается процедура анализа и по завершении анализа выведено сообщение.

***Анализ файла***

В верхнем уровне алгоритма анализа файла происходит чтение файла по строкам и вызов процедуры анализа строки для каждой из считанных строк.

По достижению конца файла в результат анализа добавляется запись о завершении анализа файла с полным путем к файлу.

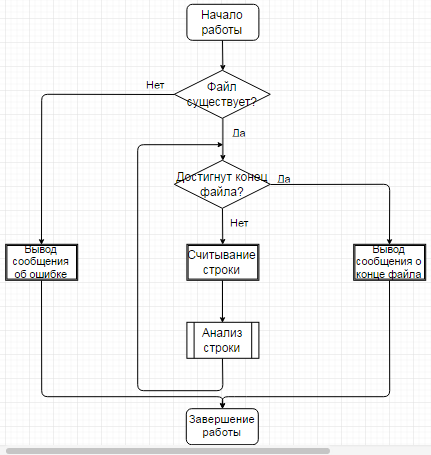


Рис. 2 Схема алгоритма анализа файла

***Анализ строки***

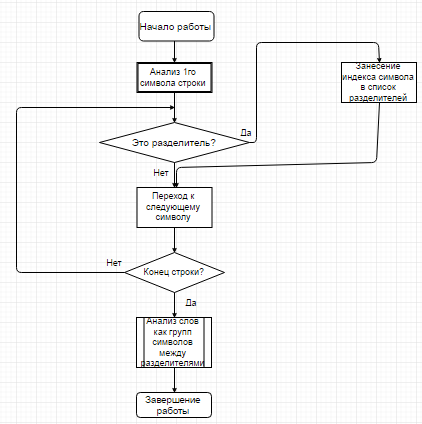


Рис. 3. Схема алгоритма обработки строки

Анализируемая строка считывается посимвольно и производит проверку символа на принадлежность его к списку символов окончания слова до тех пор, пока не произойдет одна из ситуаций:

* + 1. Встретится символ окончания слова. В эти символы входят пунктуационные символы, которые не могут входить в слово языка VHDL. По нахождению такого символа порядковый номер символа запоминается в массиве, и по завершении просмотра строки, проводится анализ ее слов как группы символов между символами окончания слова.
    2. Cтрока завершится. Последний символ строки добавляется к слову, если он не является символом окончания слова, и слово отправляется на анализ.

***Анализ слов в строке***

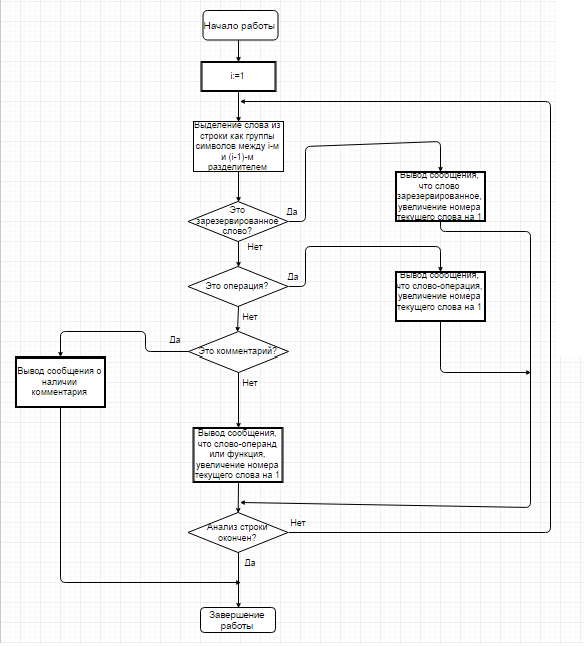


Рис 4 Схема алгоритма обработки полученных слов в строке

Производится разбиение строки на отдельные слова. При анализе строки были выделены символы, которые являются символами окончания слова(разделителями).Из строки циклически извлекаются группы символов между i-м и (i-1)-м разделителями до тех пор, пока не будет пройден весь список разделителей, а соответственно, не будут обработаны все слова в строке. Все слова анализируемой строки проходят проверку на принадлежность к одной из следующих категорий:

Таблица 2.Категории слов

|  |  |
| --- | --- |
| Категория | Признак |
| Комментарий | Поднят флаг «комментарий» |
| Служебное слово | Слово начинается с символа «#» или содержится в списке служебных слов |
| Операция | Слово содержится в списке операций |
| Операнд/Метод | Слово не подходит ни под одну иную категорию |

**Тестирование программы**

Таблица 3. План тестирования работоспособности прикладной программы.

|  |  |  |
| --- | --- | --- |
| № | Действие | Ожидаемый результат |
| 1 | Введен неправильный путь к файлу | Вывод сообщения «Указан неверный путь к файлу» |
| 2 | Введен пустой файл | Вывод только сообщений о начале и окончании анализа |
| 3 | Введен непустой файл, не содержащий слов | Вывод только сообщений о начале анализа файла, начале анализа каждой из строк и окончании анализа |
| 4 | Введен файл содержащий служебные слова | Служебные слова распознаны |
| 5 | Введен файл содержащий слова-типы данных | Слова-типы данных распознаны |
| 6 | Введен файл содержащий слова-операции | Слова-операции распознаны |
| 7 | Введен файл содержащий комментарии | Комментарии распознаны |
| 8 | Введен файл содержащий комбинацию всех видов слов языка VHDL | Все слова распознаны. |

***Тест №1.***

В форму ввода пути к файлу введен несуществующий путь и нажата кнопка «Анализировать»

Результат:

Появилось надпись с сообщением «Указан неверный путь к файлу»

***Тест №2.***

В форму ввода пути к файлу введен путь к пустому файлу.

Результат:

В результат анализа занесены следующие записи:

Начало анализа кода из файла inp.txt

Конец анализа кода

***Тест №3.***

В форму ввода пути к файлу введен путь к файлу, содержащему следующую строку:

;;; ;;;(,)

Результат:

В результат анализа занесены следующие записи:

Начало анализа кода из файла inp.txt

Строка номер1

Конец анализа кода

***Тест №4***.

В форму ввода пути к файлу введен путь к файлу, содержащему следущую строку:

#include SC\_METHOD #ifndef

Результат:

В результат анализа занесены следущие записи:

Начало анализа кода из файла inp.txt

Строка номер1

library зарезервированное слово

of зарезервированное слово

if зарезервированное слово

Конец анализа кода

***Тест №5.***

В форму ввода пути к файлу введен путь к файлу, содержащему следущую строку:

bool sc\_uint sc\_bv

Результат:

В результат анализа занесены следущие записи:

Начало анализа кода из файла inp.txt

Строка номер1

std\_logic тип данных

bool тип данных

int тип данных

Конец анализа кода

Тест №6.

В форму ввода пути к файлу введен путь к файлу, содержащему следущую строку:

= <= < >

Результат:

В результат анализа занесены следущие записи:

Начало анализа кода из файла inp.txt

Строка номер1

= операция

<= операция

< операция

> операция

Конец анализа кода

***Тест №7.***

В форму ввода пути к файлу введен путь к файлу, содержащему следущую строку:

--= <= < >

Результат:

В результат анализа занесены следущие записи:

Начало анализа кода из файла inp.txt

Строка номер1

--= <= < > комментарий

Конец анализа кода

***Тест №8.***

В форму ввода пути к файлу введен путь к файлу, содержащему описание на языке VHDL D-триггера:

library IEEE;

use IEEE.std\_logic\_1164.all;

entity dff is

port (data, clk : in std\_logic;

q :out std\_logic);

end dff;

architecture behav of dff is

begin

-- комментарий

process (clk) begin

if (clk'event and clk = '1') then

q <= data;

end if;

end process;

end behav;

Результат:

В результат анализа занесены следущие записи:

Начало анализа кода из файла inp.vhd

Строка номер1

library зарезервированное слово

IEEE операнд или функция

Строка номер2

use зарезервированное слово

IEEE.std\_logic\_1164.all операнд или функция

Строка номер3

entity зарезервированное слово

dff операнд или функция

is зарезервированное слово

Строка номер4

port зарезервированное слово

data операнд или функция

clk операнд или функция

: операнд или функция

in зарезервированное слово

std\_logic тип данных

Строка номер5

q операнд или функция

:out операнд или функция

std\_logic тип данных

Строка номер6

end зарезервированное слово

dff операнд или функция

Строка номер7

architecture зарезервированное слово

behav операнд или функция

of зарезервированное слово

dff операнд или функция

is зарезервированное слово

Строка номер8

begin зарезервированное слово

Строка номер9

-- комментарий комментарий

Строка номер10

process зарезервированное слово

clk операнд или функция

begin зарезервированное слово

Строка номер11

if зарезервированное слово

clk'event операнд или функция

and операция

clk операнд или функция

= операция

'1' операнд или функция

then зарезервированное слово

Строка номер12

q операнд или функция

<= операция

data операнд или функция

Строка номер13

end зарезервированное слово

if зарезервированное слово

Строка номер14

end зарезервированное слово

process зарезервированное слово

Строка номер15

end зарезервированное слово

behav операнд или функция

Конец анализа кода

*Вывод о результатах тестирования*.

Результаты тестирования полностью соответствуют ожидаемым результатам, что показывает, что программа полностью и верно реализует разработанный алгоритм.

**Заключение**

В результате выполнения разработки был получен алгоритм лексического анализа кода на языке VHDL. Данный алгоритм был воплощен в виде прикладной программы для компьютеров под управлением операционной системы Microsoft Windows. Работоспособность алгоритма была проверена с использованием уже существующих моделей цифровых устройств, разработанных ранее в рамках курсовых проектов и бакалаврской работы.

Данный алгоритм является базой для дальнейшей разработки анализатора кода VHDL , который будет являться частью транслятора с языка VHDL на язык SystemC.

**Литература**

1. David C. Black, Jack Donovan – “SystemC: From the Ground Up”– Springer, 2010 г. – 291 с.
2. Jack W. Crenshaw – “Let's build a compiler! Part II : Expression parsing” –[http://compilers.iecc.com/crenshaw/tutor2.txt](http://compilers.iecc.com/crenshaw/) – 1988 г. – 14 с.
3. Jack W. Crenshaw – “Let's build a compiler! Part IX : A top view” –[http://compilers.iecc.com/crenshaw/tutor9.txt](http://compilers.iecc.com/crenshaw/) – 1989г. – 31 с.
4. David Gries – “Compiler Construction for Digital Computers.”– John Wiley and Sons, NY – 1971г. – 491 с.
5. Бибило П.Н. Основы языка VHDL: Учебное пособие. Изд. 5-е. — М.: Книжный дом «ЛИБРОКОМ», 2012. — 328 с. — ISBN 978-5-397-02584-3.
6. Свердлов С. З. Языки программирования и методы трансляции. - М.: Питер, 2007