УДК 004.423.24

В.К. Воропаев, А.Ю. Медведков, Ж.Б. Садыков
V.K. Voropaev, A.Ju. Medvedkov, Zh.B. Sadykov
Омский государственный технический университет, г. Омск, Россия
Omsk State Technical University, Omsk, Russia

VHDL И VERILOG-HDL-ЯЗЫКИ ОПИСАНИЯ ЦИФРОВОЙ АППАРАТУРЫ

VHDL AND VERILOG HDL-DESCRIPTION LANGUAGES DIGITAL EQUIPMENT

В статье рассмотрена общая информация о языках описания цифровой аппаратуры Verilog (Verilog-HDL) и VHDL, их достоинства и недостатки.

General information had been considered about digital hardware description languages Verilog (Verilog-HDL) and VHDL, their advantage and disadvantage in the article.

Ключевые слова: *Verilog, VHDL, проектирование*

Keywords: Verilog, *VHDL, design*

Введение.

Развитие вычислительной техники привило к разработке языков описания цифровой аппаратуры. На сегодняшний день одними из основных языков описания цифровой аппаратуры являются Verilog (Verilog-HDL) и VHDL.

Исторически сложилось так, что в первой половине восьмидесятых годов XX века по инициативе Министерства обороны США был разработан язык спецификации проектов (VHDL). Стандарт ANSI/IEEE STD 1076-1987 на версию языка бал утвержден в 1987 году. В последующем стандарты языка менялись, к примеру: IEEE STD 1076-1993, IEEE STD 1076-1999, IEEE STD 1076-2002, IEEE STD 1076-2008.

В 1985 году фирмой Gateway Design Automaton был разработан язык моделирования аппаратуры Verilog-HDL. Первый стандарт языка был утвержден в 1995 году (IEEE 1364-1995)[1].

На сегодняшний день сложно представить описания цифровых схем без них. В данной статье рассмотрена общая информация о языках описания цифровой аппаратуры Verilog (Verilog-HDL) и VHDL, их достоинства и недостатки.

Язык VHDL.

Структура проекта в VHDL.

Основной структурной единицей VHDL является ENTITY. Декларация ENTITY определяет имя проекта и, необязательно, его интерфейс, т. е. порты и параметры настройки. Пакет – это набор объявлений вводимых пользователем типов, переменных, констант, подпрограмм и т. п. Архитектурные тела представляют содержательное описание проекта [2, с.395]. В VHDL представление сигналов реализовано в библиотеке std_logic_1164. Архитектурное тело представляет описание функций и процедур.

Типы данных.

Как и практически во всех языках программирования в VHDL используются типы данных. VHDL является строго типизированным языком. Любой единице информации в VHDL присваивается имя и определяется тип.

Типы данных:

1. *integer* – целый;
2. *real* – действительный;

3. *bit* - представляет один логический бит (значение ‘0’ либо ‘1’);
4. *boolean* - объекты этого типа принимают значения true либо false;
5. *character* - объединяет все символы;
6. *time* – используется для задания задержек;
7. *severity_level* – служит для управления работой компилятора;
8. *file_open_status и file_open_kind* – обеспечивают возможность контроля процедур между программой и файловой системой компьютера;
9. *string и bit_vector* – массив символов и битов соответственно.

Объекты VHDL.

Объекты выполняют роль хранения различных значений. Идентификаторы обязательно должны начинаться с буквы. Объекты должны быть объявлены перед использованием, за исключением переменной цикла в операторе ***for,*** которая объявляется по умолчанию.

Классы объектов:

- *Constant* – константы.
- *Variable* – переменные.
- *Signal* – сигналы, представляют значения, передаваемые по проводам.

Синтаксис объявления объектов:

Constant {name [, name]}: Type [(index_range [, index_range])] := initial_value;
Variable {name [, name]}: Type [(index_range [, index_range])] [:= initial_value];
Signal {name [, name]}: Type [(index_range)];

Операторы VHDL.

Операторы в VHDL делятся на два типа:

- Последовательные.
- Параллельные.

Последовательные операторы включены в операторы процессов. В одной строке можно размещать несколько операторов. Все операторы в VHDL оканчиваются точкой с запятой.

Последовательные операторы включают в себя:

- Оператор присваивания – выполняет присваивание переменной или сигналу результата выражения.

\результат\:=\выражение\ – присваивания переменной.
\результат\<=\выражение\ – присваивание сигнала[3, с.52].

- Оператор ожидания события ***wait .***

Синтаксис оператора:

\ wait \::= wait[onname signal\ {\name signal**}]**

- Оператор ***if –*** оператор цепочки последовательных событий;

\ if \::= if \условие **\ then**
{\последовательный оператор**}**
end if

- Оператор **case** – оператор разрешения выполнения последовательных операторов.

\case \::= case ** повторное выражение **\ is
when \альтернативы\ **=>{**\последовательный оператор**}**
{when \альтернативы\ **=>{**\последовательный оператор**}}**
end case

- Пустой оператор **null** – не выполняет никаких действий**.**
- Оператор повторения – выполняет повторение последовательных операторов.

\оператор повторения \ ::=
[<метка оператора повторения> **:] [**<итерационная схема> **:] loop**
<оператор> « <оператор > »
end loop [<метка оператора повторения> **][2];**

Параллельные операторы.

Параллельные операторы выполняются при любом изменении сигнала. К параллельным операторам относятся[2]:

- Оператор параллельного присваивания;

- Параллельный вызов процедуры;
- Оператор процесса;
- Оператор блока;
- Оператор вхождения компонента;
- Оператор генерации;
- Параллельный оператор проверки;

Язык Verilog-HDL(Verilog).

Структура проекта.

Основной единицей языка является модуль. Элементами модуля являются декларации и операторы. Структура проекта выглядит следующим образом:

module <имя модуля> (<порт>**« , <порт > »**);

« <декларация> | <параллельный оператор> »

endmodule

Типы данных.

По сравнению с VHDL,в Verilog используется меньшее количество типов данных. Данные в Verilog могут принимать одно из четырех состояний:

- 1 – представляет логическую 1 или значение «истинно» (true)
- 0 – представляет логический 0 или значение «ложно» (false)
- z – представляет состояние высокого импеданса
- x – представляет неизвестное логическое состояние

Verilog включает две группы типов данных:

- Регистры (reg);
- Цепи (wire);

Регистры предназначены для сохранения состояний. Цепи осуществляют передачу состояний между моделируемыми объектами.

Типы цепей:

- *Wire, tri* – соединение элементов
- Wand/triand, wor/trior – моделирование цепей с разрешающей логикой
- Trireg – сохраняет свое состояние(емкость) .
- Tri1, tri0, supply1,supply0 . Цепи Tri1, tri0 – моделирование цепи с резистивной подтяжкой, а supply1,supply0 – моделируют источники питания подключенные к этой цепи.

Операторы Verilog.

Операторы в Verilog, как и в VHDL, делятся на два вида: последовательные и параллельные.

Оператор *initial* запускается единожды при начале моделирования и задает начальное состояние устройства. Синтаксис данного оператора выглядит следующим образом:

< операция инициализации > ::= ***initial*** < оператор >

Оператор *always* – оператор повторения. Запускается в начале программы так же как и *initial*,но повторяется каждый раз после завершения вложенного оператора. Синтаксис оператора имеет вид:

< операция постоянного повторения> ::= ***always*** < оператор >

Операторы принятия решений – выбор одного из путей выполнения алгоритма. К операторам принятия решений относятся:

- Оператор условия

< оператор условия >: ::=

if (<выражение>)<оператор>

- Оператор варианта

<оператор варианта>::=

<определитель оператора варианта>(<ключевое выражение>)

« <вариант>**«, < вариант > » :<оператор> »**

Endcase

Заключение.

В отличии от языка VHDL, Verilog более прост с точки зрения освоения. Он схож с языком Си, а это огромный плюс. В Verilog используется меньшее количество типов данных, меньшее количество служебных слов. VHDL обладает большой универсальностью, но из-за этого он проигрывает в эффективности языку Verilog. Следовательно, человеку, который только начинает свое знакомство с языками описания цифровой аппаратуры будет легче освоить язык Verilog.

Оба языка поддерживаются в качестве стандартов большим количеством программных продуктов. Языки Verilog и VHDL используются при проектировании цифровых устройств, с применением САПР ведущих фирм в области ПЛИС. На сегодняшний день сложно представить описания цифровых схем без них.

Библиографический список

1. Поляков, А. К. Языки VHDL и VERILOG в проектировании цифровой аппаратуры. – М. : СОЛОН-Пресс, 2003. – 320 с. : ил. – (Серия «Системы проектирования»).

2. Грушвицкий, Р. И. Проектирование систем на микросхемах с программируемой структурой / Р. И. Грушвицкий, А. Х. Мурсаев, Е. П. Угрюмов. – 2-е изд., перераб. и доп. – СПб. : БХВ-Петербург, 2006. – 736 с. : ил.

3. Сергеенко, А. М. VHDL для проектирования вычислительных устройств – / А. М. Сергеенко.– М. : ООО «ТИД «ДС», 2003. – 208 с.