

## Основы языка описания аппаратуры VHDL

77-30569/289296

# 12, декабрь 2011

Ларионцева Е. А.

УДК 004.438

МГТУ им. Н.Э. Баумана

[spesial@mail.ru](mailto:spesial@mail.ru)

### Введение

Быстрый рост степени интеграции и функциональной сложности современных электронных устройств приводит к необходимости совершенствования и развития методов проектирования больших и сверхбольших интегральных схем (БИС и СБИС).

Поскольку современные СБИС содержат миллионы полупроводниковых структур на кристалле, широко использовавшийся ранее метод поэлементного проектирования архитектуры вычислительных схем по восходящей методологии не в состоянии обеспечить бездефектное проектирование сложных систем в строго установленные, приемлемые сроки. Кроме того, данная методология не обеспечивает возможность описания программных средств и учет этого описания при проектировании. Ошибки, выявленные на верхних уровнях представления, приводят к необходимости повторного выполнения этапов маршрута проектирования, начиная с уровня, на котором были допущены просчеты.

Для решения этих проблем в последние десятилетия появился новый подход для проектирования интегральных схем – подход с использованием методологии нисходящего проектирования, свободный от перечисленных недостатков. Проектирование сверху вниз описывается на многоуровневом иерархическом представлении устройств, и разработка ведется в соответствии с системной иерархией от общего описания системы к детальному описанию ее компонент.

На первом этапе разработчик описывает систему при помощи поведенческих или функциональных моделей, отражающих функцию и временные характеристики системы. Затем, на основании исходного описания, происходит компиляция исходного проекта. При этом на каждом этапе рассматриваемого маршрута проектирования с помощью моделирования выполняется верификация описания или проектных решений на соответствие заданной функции

вычислительной системы. Выявленные недостатки устраняют либо с помощью оптимизации, которая выполняется на более высоких уровнях проекта, либо путем выявления и исправления ошибок, допущенных на предыдущих уровнях [1].

Таким образом, нисходящее проектирование позволяет разработчику сосредоточиться на поведенческом и функциональном представлении системы, не отвлекаясь на структуру. Для описания аппаратуры программным способом используют *языки описания*. Языком описания называется набор синтаксических и семантических правил, определяющих формат представления устройств.

В данной статье рассматривается пример такого языка описания аппаратуры – VHDL, его структура и применение при проектировании больших интегральных схем.

## 1. Описание VHDL

Язык *VHDL (VHSIC - Very High Speed Integrated Circuits – Hardware Description Language)* разрабатывался как язык описания аппаратуры для высокоскоростных интегральных схем. Первоначальное назначение языка заключалось в обеспечении обмена проектами между различными соисполнителями работ по созданию сверхскоростных интегральных схем. Однако позже с учетом предложений и рекомендаций известных специалистов в области вычислительных систем язык был усовершенствован и стандартизован Институтом инженеров по электротехнике и радиоэлектронике (IEEE), в результате чего в 1987 году был утвержден стандарт IEEE Standard 1076 VHDL.

Язык VHDL обеспечивает высокоуровневую абстракцию описания аппаратных средств благодаря наличию как множества predefined типов данных, так и возможности создавать пользовательские иерархически организованные типы данных на основе базовых, заложенных в языке.

VHDL является формальной записью, предназначенной для описания функции и логической организации цифровой системы. Функция системы определяется как преобразование значений на входах в значения на выходах. Причем время в этом преобразовании задается явно. Организация системы задается перечнем связанных компонентов.

Язык предназначен для моделирования преимущественно на вентиляльном уровне, уровне регистровых передач и корпусов микросхем, а также успешно используется и при синтезе устройств. VHDL - не единственный язык проектирования радиоэлектронной аппаратуры ( в аббревиатура HDL в переводе с английского означает «язык проектирования устройств» - Hardware Design Language). Так, для проектирования интегральных схем широко применяется Verilog,

находят применение языки Cupl, Palasm, Ample и др., но для сквозного проектирования на всех требуемых для интегральных микросхем уровнях (выше схемотехнического) предназначены только VHDL и Verilog. Преимущества VHDL по сравнению с Verilog заключаются в следующем:

- промышленный стандарт IEEE;
- не зависит от конкретной технологии или производителя;
- код может использоваться многократно [2]

## 2. Лексические элементы

Текст описания на языке VHDL состоит из одного или более файлов проекта. Файл проекта представляет собой последовательность лексических элементов, каждый из которых составлен из строго определенного набора символов. Текст каждого модуля проекта является последовательностью отдельных лексических элементов. Каждый лексический элемент может быть ограничителем, идентификатором (который может являться служебным словом), абстрактным, символьным, строковым или битово-строковым литералом, либо комментарием.

VHDL поддерживает три различных стиля для описания аппаратных архитектур. Первый из них — структурное описание (*structural description*), в котором архитектура представляется в виде иерархии связанных компонентов.

Второй — потоковое описание (*data-flow description*), в котором архитектура представляется в виде множества параллельных регистровых операций, каждая из которых управляется вентильными сигналами.

Третий – поведенческое описание (*behavioral description*), в котором преобразование описывается последовательными программными предложениями, которые похожи на имеющиеся в любом современном языке программирования высокого уровня [2].

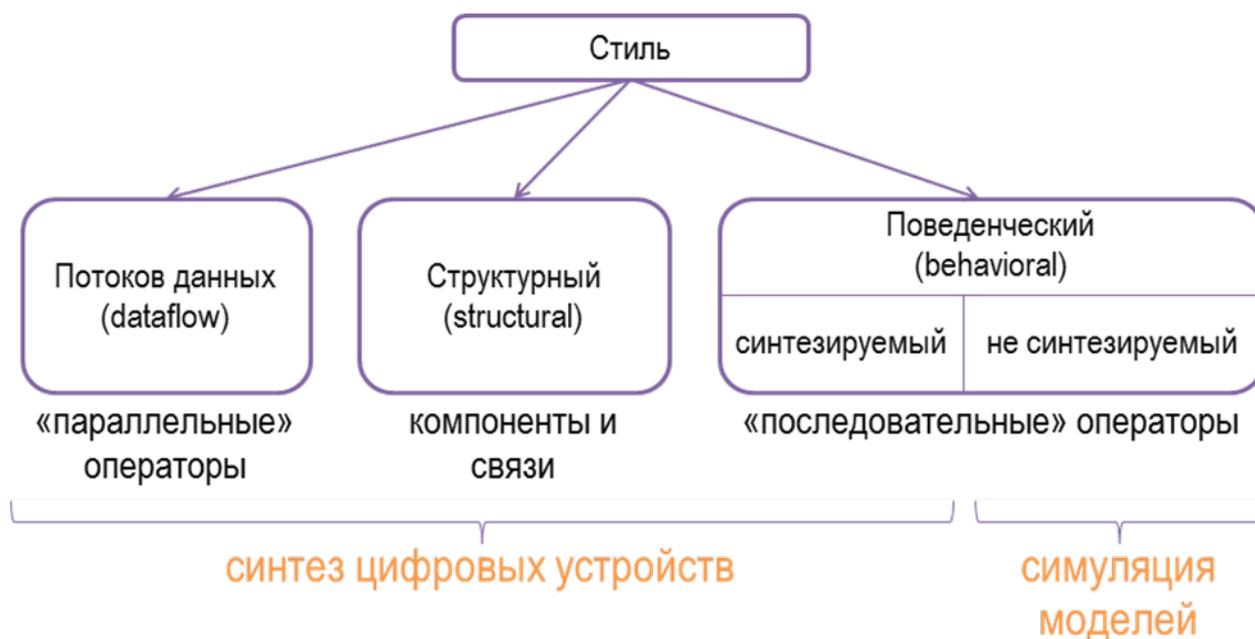


Рисунок 1. Основные стили языка.

Основными в языке VHDL являются следующие понятия.

- **ОБЪЕКТ ПРОЕКТА** (*entity*) представляет собой описание компоненты проекта, имеющей чётко заданные входы и выходы и выполняющей чётко определённую функцию. Объект проекта может представлять всю проектируемую систему, некоторую подсистему, устройство, узел, стойку, плату, кристалл, макроячейку, логический элемент и т. п. В описании объекта проекта можно использовать компоненты, которые, в свою очередь, могут быть описаны как самостоятельные объекты проекта более низкого уровня.
  - **ОБЪЯВЛЕНИЕ ОБЪЕКТА ПРОЕКТА** (*entity declaration*) описывает интерфейс и определяет только входы и выходы объекта проекта.
    - Для описания поведения объекта или его структуры служит **АРХИТЕКТУРНОЕ ТЕЛО** (*architecture body*). Чтобы задать, какие объекты проекта использованы для создания полного проекта, используется **ОБЪЯВЛЕНИЕ КОНФИГУРАЦИИ** (*configuration declaration*).
    - В языке VHDL предусмотрен механизм пакетов для часто используемых описаний, констант, типов, сигналов. Эти описания помещаются в **ОБЪЯВЛЕНИИ ПАКЕТА** (*package declaration*).
    - Если пользователь использует нестандартные операции или функции, их интерфейсы описываются в объявлении пакета, а тела содержатся в **ТЕЛЕ ПАКЕТА** (*package body*).

Таким образом, при описании цифровых схем на языке VHDL, возможно использование пяти различных типов описаний: объявление объекта проекта, архитектурное тело, объявление конфигурации, объявление пакета и тело пакета. Каждое из описаний является самостоятельной конструкцией языка VHDL, может быть независимо проанализирована анализатором и поэтому получило название “МОДУЛЬ ПРОЕКТА” (*design unit*) [1].

Средства VHDL для отображения структур цифровых систем базируются на представлении о том, что описываемый объект **entity** представляет собой структуру из компонент **component** соединяемых друг с другом линиями связи. (Компонент часто называют «черным ящиком» - он является аналогом объявления интерфейса объекта, реализованного в библиотеке) Каждая компонента, в свою очередь, является объектом и может состоять из компонент низшего уровня (иерархия объектов). Взаимодействуют объекты путем передачи сигналов **signal** по линиям связи. Линии связи подключаются к входным и выходным портам компонент. В VHDL сигналы отождествляются с линиями связи. писание структуры объекта строится как описание связей конкретных компонент, каждая из которых имеет имя, тип и карты портов. Карта портов **port map** определяет соответствие портов компонент поступающим на них сигналам, можно интерпретировать карту портов как разъем, на который приходят сигналы и в который вставляется объект-компонента.

Архитектура любого проекта, написанного на языке VHDL, включает в себя описание одного или нескольких параллельных процессов. Описание процесса состоит из последовательности операторов, отображающих действия по переработке информации. Все операторы внутри процесса выполняются последовательно. Процесс может находиться в одном из двух состояний- либо пассивном, когда процесс ожидает прихода сигналов запуска или наступления соответствующего момента времени, либо активном- когда процесс исполняется. Процессы взаимодействуют путем обмена сигналами [1].

Кроме описанных выше средств языка, в программах нередко используются массивы. Массив представляет собой упорядоченную структуру однотипных данных. Массив имеет диапазон индексов, который может быть возрастающим либо убывающим. На любой элемент массива можно сослаться, используя его индекс.

Объявление ограниченного типа "массив" имеет вид:

```
TYPE имя_типа IS
    ARRAY (диапазон_индексов [ , диапазон_индексов] )
    OF тип_элемента;
```

*диапазон\_индексов* может определяться двумя способами:

1) явным заданием границ диапазона

$$m \text{ TO } n$$
$$n \text{ DOWNTO } m$$

где  $m, n$  – целочисленные константы,  $m \leq n$ ;

2) с использованием идентификатора ограниченного подтипа. В этом случае значения границ подтипа являются значениями границ индекса массива.

Все описанные элементы являются примером использования поведенческого стиля описания [2].

### 3. Методология проектирования на языке VHDL

Широкое применение язык VHDL нашел при проектировании устройств на так называемых программируемых интегральных микросхемах (сокращенно ПЛИС).

На данный момент на рынке ПЛИС представлены микросхемы FPGA- и CPLD-структур. FPGA (Field Programmable Gate Array) – это микросхема, конфигурируемая пользователем. Она состоит из ячеек, часть из которых отвечает за реализацию элементарных функций (Configurable Logic Blocks CLB) а часть – за внутренние соединения (Programmable Switch Matrices PSM). Схема FPGA представлена на рисунке 2.

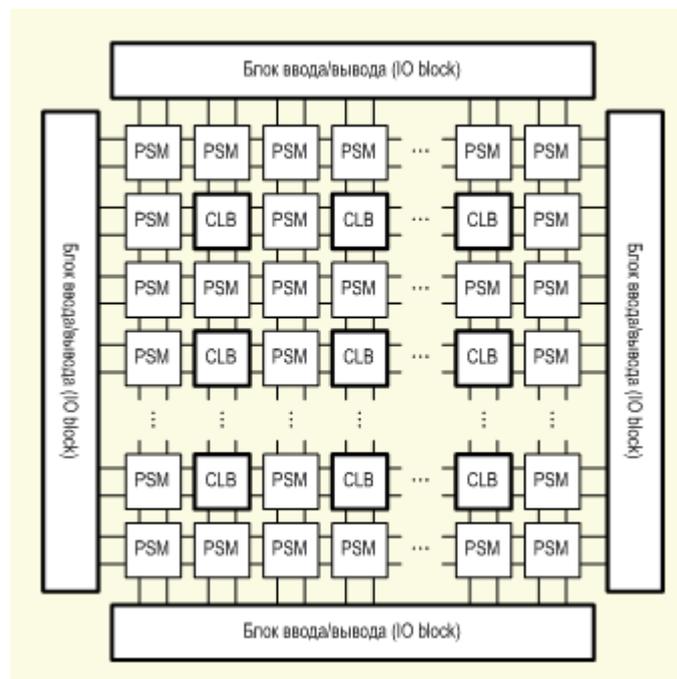


Рисунок 2. FPGA структура

ПЛИС на FPGA-архитектуре основаны на ОЗУ и могут быть перепрограммированы бесконечное число раз. Но из-за этого у таких ПЛИС появляется существенный недостаток: при отключении питания программа теряется и для предотвращения ее потери нужен специальный конфигурактор. У ПЛИС с CPLD-структурой отсутствует проблема потери программы при отключении питания, т.к. они основаны на флэш-памяти. Поэтому программа может храниться постоянно, к тому же она может быть защищена битом секретности. Но вместе с тем у таких ПЛИС появляется другой недостаток: микросхема может быть перепрограммируема ограниченное число раз [3].

Процесс проектирования устройства является многоступенчатым (рисунок 3).

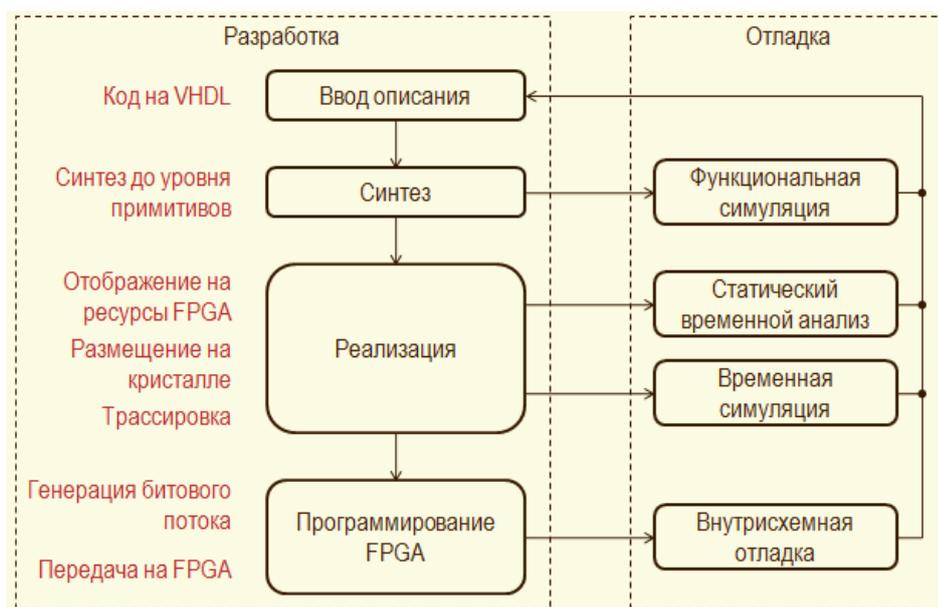


Рисунок 3. Процесс проектирования устройства

Разработку интегральных схем можно выполнять, опираясь на различные уровни абстрактного представления проекта. Наиболее высокий уровень абстракции можно представить алгоритмическими выражениями, описывающими логику работы устройства. Кроме того, стандарт IEEE Standard 1076 VHDL позволяет компиляторам, реализующим VHDL, допускать встраивание описаний различных подпрограмм на языках программирования высокого уровня (Си, Паскаль и т.д.) в качестве реализации функций и процедур. В этом стиле удобно разрабатывать описание устройств высокого уровня сложности (ЭВМ, процессоры, контроллеры различных устройств, сложные последовательные схемы и т.д.). При поведенческом описании компоненты могут вводиться на любых иерархических уровнях, что позволяет пользоваться библиотечными наборами готовых логических и архитектурных решений. Немаловажное значение для качества проектирования имеет возможность оптимизации исходного поведенческого описания. Так, например, введение неоправданно большого количества

дополнительных, промежуточных сигналов и переменных способно привести к совершенно неработоспособному проекту [1].

## Заключение

В настоящее время язык VHDL используется для работы с вычислительными системами всех уровней сложности. Различные этапы проектами могут выполняться различными группами разработчиков. Поведенческое и потоковое описание устройства могут служить коммерческим результатом проекта. Эти данные после передачи заказчику могут быть использованы для дальнейшей проработки и структурной конкретизации проекта.

Одним из примеров применения VHDL при проектировании реальных устройств является использование его при программировании ПЛИС Altera MAX II, предназначенная для изучения и освоения работы схем на базе микросхемы программируемой логики. Данная микросхема является частью отладочной платы Cyclon II FPGA Starter Board, обеспечивающей пользователей всеми необходимыми средствами для создания и тестирования проектов без необходимости использования дополнительных средств (микроконтроллеров, контроллеров памяти и так далее).

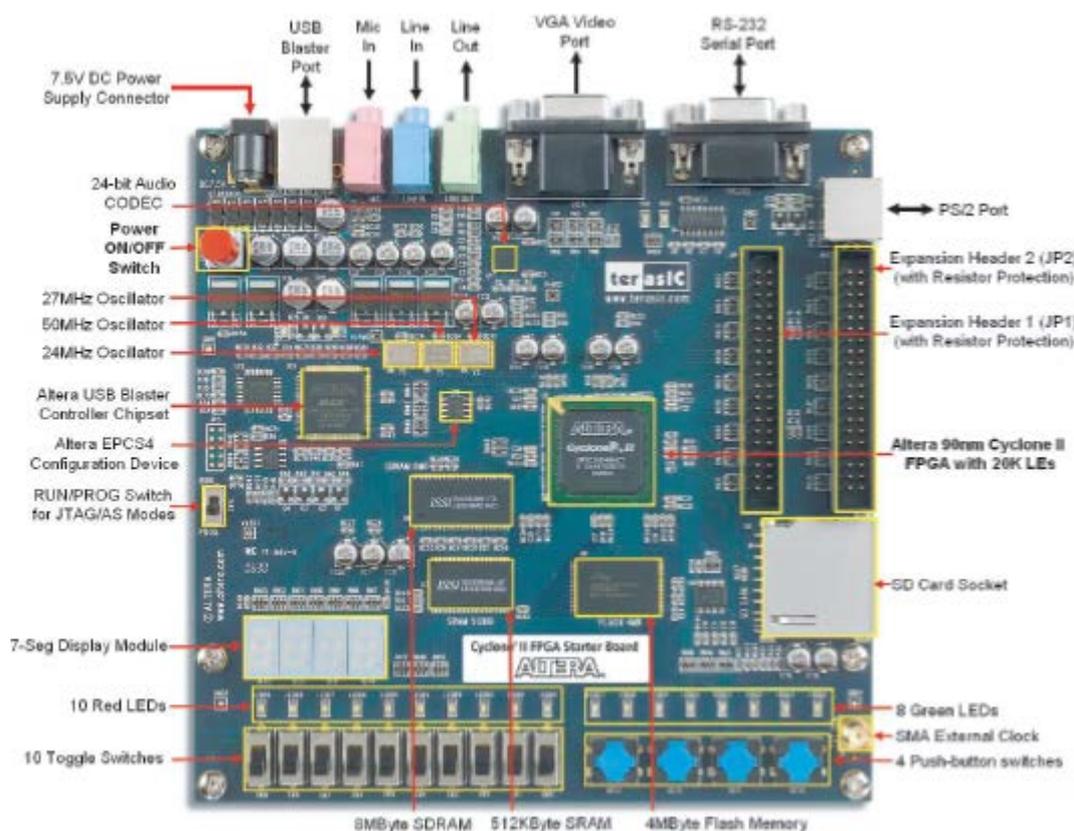


Рисунок 4. Cyclon II FPGA Starter Board

Применение языка VHDL на различных этапах проектирования устройств значительно повышает эффективность проектирования, позволяет создавать понятные модели проектируемых устройств и эффективно выполнять соответствующее имитационное моделирование, обладая при необходимости независимостью от конкретных аппаратных структур [3].

### Список литературы

- 1) Ивченко В. Г. Применение языка VHDL при проектировании специальных СБИС. Т.: ТГРУ, 2000. с. 1-15, 45-47.
- 2) Сергиенко А. М. VHDL для проектирования вычислительных устройств. М.: DiaSoft, 2003. с. 6-9, 25-30
- 3) Cyclone II FPGA Starter Development Board. Reference manual. San-Jose.:Altera, 2006. с. 1-6.