Технологии проектирования компьютерных систем

Лекция 7. Представление системы в VHDL

Общая структура описания проекта системы

Система предназначена для выполнения заданных преобразований, для чего она должна:

- получить некоторые входные данные от окружающей ее среды;
- выполнить преобразования;
- вывести некоторые данные.

Из анализа приведенного следует, что система должна иметь связи с окружающей ее средой, которые называют интерфейсом.

Интерфейс системы описан в VHDL его сущностью (Entity), которая является основной единицей проекта для любой системы.

Выполнение преобразований осуществляется внутренней частью системы или телом (Body), которая называется архитектурой (Architecture).

Для получения дополнительных возможностей систем применяют пакеты (Package) и библиотеки (Library).

Общая структура описания проекта системы

Структуру проекта в общем случае описывают по шаблону:

-- Context Clauses

-- LIBRARY __library_name; -- Library Clause

-- USE __library_name.__package_name.ALL; -- Use Clause

-- Library Units

-- Package Declaration (optional)

-- Package Body (optional)

-- Entity Declaration

-- Architecture Body

В структуре проекта последовательно описывают применяемые стандартные библиотеки (Library Clause) и входящие в них пакеты (Use Clause), пакеты пользователя (Package Declaration и Package Body), интерфейс объекта (Entity Declaration) и его архитектуру (Architecture Body).

В самом простейшем виде описание может содержать только Entity Declaration и Architecture Body.

Сущность проекта системы

Любой проект системы в VHDL следует начинать с декларации сущности (Entity Declaration), которую описывают следующим образом:

```
ENTITY __entity_name IS
GENERIC ( __parameter_name  : __TYPE := __default_value);
    PORT      (__port            :__mode  __TYPE );
END  ENTITY  __entity_name;
```

Имя сущности (__entity_name), формально названное идентификатором, предназначено для целей документирования. Рекомендуется задавать имя с учетом функций, выполняемых системой.

Сущность проекта системы

Сущность обеспечивает спецификацию интерфейса системы и обычно включает в себя два элемента:

- параметры настройки системы (Generic);
- порты связи (Port), которые передают информацию к системе и от нее (системные вводы и выводы).

Параметр настройки системы (Generic) представляет собой канал статической информации, которая будет сообщена системе из окружающей среды. Он используется для описания постоянных значений. Например, он может задавать разрядность шины данных, адреса, направления счета, модуль счета и т.д.

Сущность проекта системы

Описание параметров настройки состоит из ключевого слова Generic и списка параметров, заключенных в скобки , например:

```
GENERIC   (WIDTH : INTEGER  := 8;
           DEPTH:    INTEGER  := 15;
           INDATAWIDTH : POSITIVE := 8);
```

Если значение не определено и система используется в иерархической структуре, то значение должно быть определено через текущий компонент (Component Instantiation).

Сущность проекта системы

Порт связи (Port) язык VHDL определяет как канал для динамической связи между сущностью и окружающей средой. На практике эти каналы (сигналы) формируют интерфейс системы, поэтому каждый порт должен быть точно определен.

Каждый порт определяется своим предложением порта (Port Clause), например:

```
PORT (SIGNAL  i0      : IN   BIT;     -- входные данные
      SIGNAL  sel     : IN   BIT;     -- выбор адреса
      SIGNAL  y       : OUT  BIT); -- выходные данные
```

Сущность проекта системы

Port Clause состоит из следующих элементов:

- ключевое слово Signal (необязательно);
-имя порта;
- режим работы порта (mode);
- тип данных порта (type);
- начальное значение, которому предшествует символ := (необязательно);
- комментарий, описывающий порт (необязательный, но рекомендуемый).

Режимы порта

Имеется пять доступных режимов: In, Out, Inout, Buffer, Linkage.

Режим In - интерфейсный объект можно только читать (изнутри).

Режим Out - интерфейсный объект можно переустанавливать (изнутри), но не читать.

Режим Inout - интерфейсный объект можно читать и переустанавливать.

Режим Buffer - интерфейсный объект должен переустанавливаться (изнутри) только одним источником. В отличие от интерфейсного объекта с модой Out его можно читать.

Режим Linkage - интерфейсный объект можно читать и переустанавливать, при этом все другие интерфейсные объекты, связанные с ним, также должны иметь режим Linkage.

По умолчанию используется режим In.

Для интерфейсных объектов Constant возможна только режим In.

Сущность, как большинство конструкций VHDL, заканчивается ключевым словом End.

Архитектура проекта системы

В VHDL архитектура проекта системы (Architecture Body) может быть описана как:

```
ARCHITECTURE architecture_name OF entity_name IS
    architecture_declarations
BEGIN
concurrent_statements
END [ ARCHITECTURE ] [ architecture_name ];
```

Архитектура описывает внутренние отношения между портами ввода и вывода объекта. Она состоит из двух частей: объявления и параллельных утверждений.

Первая (декларативная) часть архитектуры может содержать объявления типов, подтипов, сигналов, констант, подпрограмм (функций и процедур), атрибутов, компонент и групп.

Архитектура проекта системы

Параллельные утверждения в теле архитектуры определяют отношения (связи) между входами и выходами. Эти отношения могут быть определены, используя различные типы предложений и стили: структурный, поведенческий или смешанный.

Структурное описание основано на конкретизации компонентов и линий связи между ними и интерфейсом. Оно позволяет создавать иерархические проекты: от простых схем до очень сложных компонентов, описывающих полные подсистемы.

Поведенческое описание определяет алгоритм преобразования входных сигналов в выходные.

Архитектура может содержать утверждения, которые одновременно содержат структурное и поведенческое описание. Такое описание архитектуры называют смешанным.

Предложения VHDL

Рассмотрим рекомендуемую структуру проекта с учетом применяемых основных параллельных и последовательных предложений и места их расположения.

Порядок выполнения параллельных предложений не связан с порядком их появления внутри архитектурного тела. Параллельные предложения активизируются сигналами, которые употребляются для связи параллельных предложений.

Последовательные предложения (Sequential Statements) выполняются в порядке их появления в VHDL-коде.

Предложения VHDL

Package (optional)
Entity (I/O)
Architecture
- **Concurrent Statements**
  - Signal Declaration
  - Component Instantiation Statement
  - Conditional Signal Assignment Statement
  - Selected Signal Assignment Statement
  - Generate Statement
  - Process Statement
    - **Sequential Statements**
      - Variable Declaration
      - Signal Assignment
      - Variable Assignment
      - Procedure Call
      - If, Case, Loop, Next, Exit, Return
      - Wait Statement

Предопределенные атрибуты

В языке VHDL на некоторые характеристики объектов могут быть даны ссылки в выражениях в удобной и компактной форме, называемой записью атрибута (attribute notation).

Значение атрибута можно получить, если указать после имени объекта апостроф и имя атрибута по формату:

< имя объекта>'< имя атрибута>.

В языке VHDL декларированы 36 атрибутов, которые делят на шесть видов. Это атрибуты для:

- всех типов данных;
- скалярных типов данных;
- дискретных типов физических величин;
- массивов;
- сигналов;
- сущностей.

Атрибуты для скалярных типов данных

Атрибут	Тип результата	Результат
T'left	T	Самое левое значение типа
T'right	T	Самое правое значение типа
T'low	T	Наименьшее возможное значение типа
T'high	T	Наибольшее возможное значение типа
Примечание. T является именем скалярного типа		

Отметим, что для нарастающего диапазона типа данных выполняется условие T'left = T'low, T'rigth = T'high, а для спадающего диапазона - T'left = T'high, T'right = T'low.

Атрибуты для скалярных типов данных

Существуют два предопределенных подтипа целого типа, которые используют атрибут high в их выражениях для диапазонов:

SUBTYPE NATURAL IS INTEGER RANGE 0 TO INTEGER'HIGH;
SUBTYPE POSITIVE IS INTEGER RANGE 1 TO INTEGER'HIGH;

Значение атрибута может быть использовано при объявлении подтипа, в задании параметров цикла, в назначении.

Атрибуты для массивов

Атрибут	Результат
A'left(n)	Левая граница индекса n-й размерности массива
A'right(n)	Правая граница индекса n-й размерности массива
A'low(n)	Нижняя граница индекса n-й размерности массива
A'high(n)	Верхняя граница индекса n-й размерности массива
A' length(n)	Разрядность вектора n-й размерности массива
A' range(n)	Диапазон изменения индекса n-й размерности массива
Примечание. Атрибут range можно использовать только в задании параметров цикла.	

Атрибуты для сигналов

Атрибут	Результат
S'event	True, если произошла смена сигнала S в текущем цикле моделирования, False – в противоположном случае

Атрибут event применяют для выделения переднего или заднего фронтов сигналов синхронизации при описании последовательных устройств.