



Технологии проектирования компьютерных систем

Лекция 5. Имена



Формы имен

При любом объявлении объекта определяют идентификатор, который допускает ссылку на такой объект. Однако не всегда возможно указать объект или часть его, используя только его идентификатор. Более общая форма ссылки к объектам - *ИМЯ*.

Существуют шесть форм имен.

name ::=

- simple_name** -- простое имя;
- | **operator_symbol** -- символ оператора;
- | **selected_name** -- селективное имя;
- | **indexed_name** -- индексное имя;
- | **slice_name** -- вырезка имени;
- | **attribute_name** -- имя атрибута.

Простое имя

Простое имя - это идентификатор. Имя может быть создано:

- **посредством объявления;**
- **при использовании метки;**
- **при создании цикла;**
- **при создании псевдонима объекта.**

Формат описания простого имени.

simple_name ::= identifier

identifier ::= letter {[_] letter | digit}

Примеры описания простых имен

ARCHITECTURE behavel_shifter **OF** shifter **IS** -- behavel_shifter
- простое имя.

SIGNAL reg_com: **BIT_VECTOR** (0 **TO** 15); -- reg_com - простое
имя.

proc_shifter: **PROCESS BEGIN** -- proc_shifter - имя поименованного
процесса.

FOR k **IN** 1 **TO** 12 **LOOP** -- k - имя переменной цикла.

VARIABLE e: **BIT_VECTOR** (0 **TO** 3); -- создание псевдонима.
ALIAS f : **BIT_VECTOR** (0 **TO** 3) **is** e;

Символ оператора

Символ оператора представляет собой имена только
предопределенных операторов в их изначальном или перегруженном
смысле.

operator_symbol ::= string_literal

Примеры описания символа оператора:

FUNCTION "+"(L, R : BOOLEAN) RETURN BOOLEAN -- "+" -
символ (имя) предопределенного оператора плюс.

Селективное имя

Селективное имя служит для указания объекта, объявленного внутри другого объекта или в библиотеке.

Формат описания селективного имени.

selected_name ::= prefix . suffix

prefix ::= name | function_call

suffix ::= simple_name | character_literal | operator_symbol | all

Селективное имя состоит из префикса, который является названием составного объекта, точки (.) и суффикса, который может быть простым именем, символьной константой, символом оператора или зарезервированным словом **all**. В последнем случае имя обращается ко всем объектам, объявленным в составном объекте.

Примеры описания селективных имен

WORK.N_GATES
PAC_OPER."*"
NMOS.GATES.all

DATA_RECORD.DAY
STD_ULOGIC.'X'

Первое имя дает доступ к пакету N_Gates в библиотеке Work. Второе обращается к объекту DAY составного типа Data_Record. Заметим, что оба имени созданы одинаковым образом и их правильная интерпретация возможна только в контексте. Третье имя определяет оператор * пакета PAC_OPER, а четвертое - значение типа данных STD_ULOGIC. Последнее имя определяет все объекты, объявленные в пакете NMOS.GATES.

Префикс может также иметь форму селективного имени (всего с двумя точками), делая ссылку более сложной.

A_LIB.PAC_OPER."+"

Комплексное селективное имя дает доступ к оператору "+", который определен в пакете PAC_OPER библиотеки A_LIB.

Индексное имя

Индексное имя указывает элемент массива, который обозначен списком выражений.

Формат описания индексного имени:

`indexed_name ::= prefix (expression { , expression })`

Число выражений в списке должно соответствовать размерности массива, а значения выражений должны находиться в пределах соответствующих индексных диапазонов.

Для приведенных выше переменных индексными именами являются:

`REGISTER_ARRAY (5)` - элемент одномерного массива

`MEMORY_CELL (1024,7)` - элемент двухмерного массива

Индексное имя имеет ту же самую форму, что и вызов функции. В таком случае интерпретация имени зависит от контекста. Если интерпретация такого имени невозможна, то имя неоднозначно.

Вырезка имени

Вырезки имени позволяют манипулировать с частями одномерных массивов, то есть префикс здесь должен иметь тип одномерного массива.

Формат описания вырезки имени:

slice_name ::= prefix (discrete_range)

Префикс определяет название массива, а дискретный диапазон - элементы объекта, к которым обращаются через вырезку имени. Дискретный диапазон должен быть совместим с диапазоном массива и иметь направление, указанное в объявлении объекта.

Примеры:

signal R15: BIT_VECTOR (0 to 31);

constant DATA: BIT_VECTOR (31 downto 0);

R15 (0 to 7).

DATA (24 downto 1).

Имя атрибута

Атрибут - это значение, связанное с поименованным объектом. В языке существует ряд predefined атрибутов для таких объектов, как массивы, блоки, сигналы, типы (Array, Block, Signal, Type). Кроме этого, можно объявить, а затем специфицировать (связать с поименованными объектами и задать значения) свои (пользовательские) атрибуты.

Формат описания имени атрибута;

attribute_name ::= prefix ' attribute_designator [(expression)]

attribute_designator ::= attribute_simple_name

Пример описания имени атрибута:

TYPE COLOUR IS (white, blue, red);

Значение выражения **COLOUR'VAL(2)** вернет red.

Видимость и область действия имен



Каждому месту в тексте программы соответствует некоторое видимое множество имен, которое должно быть непротиворечивым - однозначно воспринимаемым. И наоборот, введение (объявление) имени в том или ином месте текста определяет область в тексте программы, где это объявление действительно.

Правила видимости и установки границ действия имен в VHDL те же, что и в большинстве языков программирования.

