АТРИБУТЫ

Атрибуты - скаляры, отражающие некоторые свойства объектов, используемых в программных модулях. Например, атрибуты типа предназначены для сжатого представления информации о множестве значений, объединенных типом, а атрибуты сигнала - для представления временных свойств сигнала. Атрибуту присваивается имя и тип. Имя является обычной переменной в выражениях того типа, который присвоен атрибуту. Имя атрибута записывается следующим образом:

<имя атрибута> ::=

<имя атрибутируемого объекта>'<определитель атрибута> [(<выражение>)].

Определитель атрибута определяет свойство объекта, представляемое атрибутом. Необязательное выражение может задавать дополнительные данные для вычисления значения атрибута.

Бывают предопределённые атрибуты и атрибуты, вводимые программистом. Рассмотрим только представлением наиболее употребительных предопределенных атрибутов.

Предопределенные атрибуты типов

В таблице 1 приведены предопределенные атрибуты типов. Здесь T - имя типа, N - целое, X - "вспомогательное" выражение, тип которого совпадает с типом T. Тип перечисленных атрибутов, кроме T'pos и T'image, совпадает с атрибутируемым типом. Атрибут T'pos принимает целое значение, а T'image - строка.

Таблица 1

Вид атрибута	Вычисляемое значение	Атрибутируемый тип
T'left	Левая граница значений T	Любой скалярный
T'right	Правая граница значений T	Любой скалярный
T'low	Нижняя граница значений T	Численный, физический
T'high.	Верхняя граница значений T	Численный, физический
T'image(X)	Строка символов, представляющая значение X	Любой
T'pos(X)	Позиция значения X в наборе значений T	Перечислимый

T'val(N)	Значение элемента в позиции N в наборе значений T .	Перечислимый, физический, целый
T'leftof(X)	Значение в наборе значений T, записанное в позиции слева от X	Перечислимый, физический, целый
T'rightof(X)	Значение в наборе значений T, записанное в позиции справа от X	Перечислимый, физический, целый
T'pred(X)	Значение в наборе значений T на одну позицию меньшее X	Перечислимый, физический, целый
T'succ(X)	Значение в наборе значений T на одну позицию большее X	Перечислимый, физический, целый

Например, именем атрибутируемого объекта может быть тип std_logic – перечисляемый тип, содержащий следующие перечисляемые значения: ‘U’, ‘X’, ‘0’, ‘1’, ‘Z’, ‘W’, ‘L’, ‘H’, ‘-’. Тогда справедливо:

std_logic’low = ‘U’;
std_logic’pos (‘1’) = 3;
std_logic’val (7) = ‘H’;
std_logic’succ (‘Z’) = ‘W’;

Пусть совокупность возможных состояний некоторого устройства объявлена типом

TYPE state **IS** :=(s0, s1, s2, s3, s4, s5),

а текущее состояние представлено сигналом current_state. Рассмотрим поведение фрагмента, представленного процессом:

```
PROCESS
IF current_state = state'right THEN
    current_state<=state'left;
ELSE current_state <=state'succ (current_state);
WAIT 100 nS;
END IF;
END PROCESS;
```

Устройство поочередно принимает состояния в порядке записи в списке значений от s0 до s5, а из s5 выполняется переход в начальное состояние s0.

Для атрибутов целых типов позиция совпадает с фактическим значением вспомогательного выражения, а для атрибутов физических типов - с числом базовых единиц в значении вспомогательного выражения.

Пусть требуется определять заряд Q, передаваемый по некоторой цепи постоянным током I за время T. Тогда следует объявить соответствующие типы и переменные:

```
TYPE current IS RANGE integer'low TO integer'high;
  UNITS uA; -- микроампер;
   mA= 1000 uA; -- миллиампер;
   A=1000 mA; -- ампер;
  END UNITS current;
TYPE charge IS RANGE integer'low TO integer'high
  UNITS pQ; -- пикокулон;
   uQ= 1000 pQ;
  END UNITS charge;
VARIABLE I : current;
VARIABLE Q : charge;
VARIABLE T : time;
```

Тогда оператор вычисления заряда можно записать следующим образом:

```
Q:= charge'val ( current'pos(I) * time'pos( T) * E-9);
```

Множитель E-9 согласует размерности единиц измерения: 10^{-6} (А) × 10^{-15} (с) = 10^{-21} (Кл) = $10^{-9} \times 10^{-12} = 10^{-9}$ (пКл).

Предопределенные атрибуты массивов

Приведенные в табл. 2 атрибуты упрощают запись подпрограмм и описаний настраиваемых модулей. Они, в частности, позволяют записывать границы обработки безотносительно к фактическому размеру массива. В таблице A - имя типа массива, а N— порядковый номер измерения многомерного массива. Для одномерного массива N=1, но можно выражение в скобках при записи атрибута вообще опускать. Тип результата всегда совпадает с типом индекса. Смысл определителей left, low, right, high такой же, как у определителей типов.

Таблица 2

Имя атрибута	Результат
A'left(N)	Левая граница диапазона индексов *N*-й координаты массива A
A'right(N)	Правая граница диапазона индексов *N*-й координаты массива A
A'low(N)	Нижняя граница диапазона индексов *N*-й координаты массива A
A'high(N)	Верхняя граница диапазона индексов *N*-й координаты массива A
A'range(N)	Диапазон индексов *N*-й координаты массива A
A'reverse_range(N)	Обратный диапазон индексов *N*-й координаты массива A
A'length(N)	Диапазон индексов *N*-й координаты массива A

Пусть тип type A_BYTE есть массив убывающего диапазона (7 **DOWNTO** 0). Тогда

A_BYTE'left= A_BYTE'high=7

A_BYTE'low= A_BYTE'right=0

В качестве примера следующий листинг представляет декларацию функции glue, аргумент которой есть битовый массив типа arr_type, который определен в вызывающей программе, а возвращаемое значение - вектор, объединяющий четыре старших и четыре младших бита аргумента.

```
FUNCTION glue (x:arr_type) return bit_vector (7 DOWNTO 0) IS
BEGIN
RETURN ( x(arr_type'right DOWNTO arr_type'right-3)
        & x(arr_type'left+3 DOWNTO arr_type'left));
END glue;
```

Замечание.

Определитель range возвращает не одно значение, а множество значений "от- до". Например, для типа массива

TYPE byte **IS ARRAY** (7 **DOWNTO** 0) **OF** bit;

атрибут byte'range принимает значение "7 DOWNTO 0",

атрибут byte' reverse_range - значение "0 TO 7",

а атрибут byte'length - значение 8.

Ниже приведен тест для определения значений атрибутов типа vector, декларированного как массив бит убывающего диапазона:

```
TYPE vector IS ARRAY (7 DOWNTO 0) OF bit;
ENTITY test_attr_vector IS
END test_attr_vector;
ARCHITECTURE beh OF test_attr_vector IS
TYPE vector IS ARRAY (7 DOWNTO 0) OF bit;
SIGNAL x: vector;
SIGNAL A, B, C, D : integer;
SIGNAL E: boolean;
SIGNAL F, G, H : integer;
BEGIN
A <= vector'left;
B <= vector'right;
C <= vector'low;
D <= vector'high;
E <= vector'ascending;
F <= vector'range;
G <= vector'reverse_rangc;
H <= vector'length;
END beh;
```

В результате моделирования получим: A=7, B=0, C=0, D=7, E=false, F=7, G=0, H=8.

Атрибуты сигналов являются эффективным средством анализа поведения сигнала во времени. В таблице 3 символ S означает имя сигнала.

Например, в конструкции

IF current_state'event **THEN** <оператор>;

вложенный оператор будет выполняться, только если в момент инициализации исполнения этой конструкции сигнал current state меняет свое значение.

Таблица 3

Имя атрибута	Тип атрибута	Значение
S'delayed (T)	То же, что у S	Значение S, существовавшее на время T перед вычислением атрибута

S'event	BOOLEAN	Сигнализирует об изменении сигнала
S'stable	BOOLEAN	S'STABLE = not S'EVENT
S'active	BOOLEAN	TRUE, если присвоение сигналу выполнено, но значение еще не изменено (не закончен временной интервал, заданный выражением AFTER)
S' quiet	BOOLEAN	S'ACTIVE = not S'QUIET
S'last_event	TIME	Время от момента вычисления атрибута до последнего перед этим изменения сигнала
S'last_active	TIME	Время от момента вычисления атрибута до последнего присвоения значения сигналу (не совпадает с last_event при наличии слова AFTER в определяющем выражении)

Пример.

```
ENTITY signal_ex IS
END signal_ex;
ARCHITECTURE beh OF signal_ex IS
SIGNAL ex, y1, y3 : bit;
SIGNAL y2 : boolean;
BEGIN
ex <= '0' AFTER 20 ns;
'1' AFTER 50 ns;
'0' after 60 ns;
'1' after 80 ns;
y1 <= ex'transaction;
y2 <= ex'event;
y3 <= ex'last_value;
END beh;
```

Временные диаграммы сигналов после выполнения представленного кода изображены на рис. 1. В момент времени 20ns произошла транзакция сигнала ex (назначение сигналу '0' after 20ns), поэтому атрибут ex'transaction изменил свое значение, следующее изменение

произошло в момент времени 50ns, т. е. тогда, когда произошло следующее изменение сигнала ex, и т. д.

Атрибут ex'event (сигнал y2) один раз изменил свое значение, а именно, после того, как первый раз изменилось значение сигнала ex.

Атрибут ex'last_value (сигнал y3) показывает предыдущее значение сигнала ex.

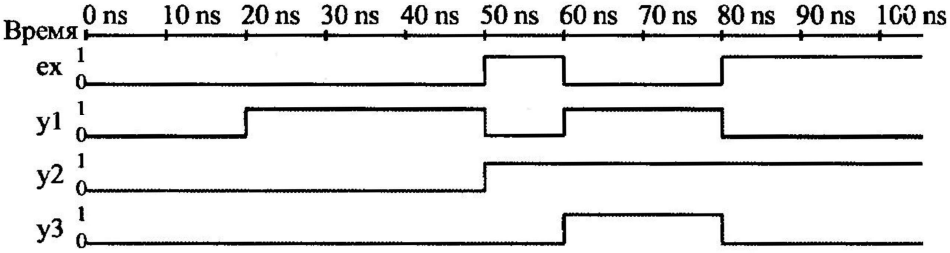

Рис. 1