

Array and Types in VHDL

Array

In VHDL, an array is a collection of elements that have the same data type. You can think of an array as a variable that has many elements with the same data type, and these elements are indexed for access. The index can be a number or another indexable type, such as integer, natural, or `std_logic_vector`. Arrays can have one dimension (one-dimensional array) or more (two-dimensional array, three-dimensional array, and so on). Two-dimensional arrays are often used to represent tables or matrices.

Types

In VHDL, a type is a definition used to declare a new data type. Types can be used to define complex data types, such as arrays or records, or as types used to declare variables, ports, or signals. Types can also be used to describe the properties and structure of data. VHDL has predefined data types, such as `std_logic`, `std_logic_vector`, `integer`, and others, but we can also create our own custom data types. Types that are predefined or embedded in the VHDL library are called "built-in types," while types that we define ourselves are called "derived types."

Example

```
-- Custom type definition
type color is (red, green, blue, yellow, black, white);

-- Variable declaration using custom type
signal primary_color: color;

-- Array declaration using custom type
type color_array is array (natural range <>) of color;

-- Array instantiation
signal color_table: color_array(0 to 3);
```

In the example above, we define a custom type `color` with specific values. We then declare a signal `primary_color` using this custom type. We also define an array type `color_array` that can hold elements of type `color`, and we instantiate an array `color_table` with a range of 0 to 3 using this array type.

Revision #1

Created 23 September 2024 16:03:48 by GI

Updated 23 September 2024 16:03:59 by GI