

# Extra: Array in VHDL

## 6.1 Array

In VHDL, an `array` is a **collection of elements** that share the same data type. You can think of an `array` as a variable that holds many elements of the same type, and these elements are indexed to be accessed. 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, three-dimensional, and so on). Two-dimensional arrays are often used to represent tables or matrices.

## 6.2 Type

A `type` is a definition used to **declare a new data type** in VHDL. A `type` can be used to define complex data types, such as arrays or records, or as a type 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 VHDL libraries are called "built-in types," while types that we define ourselves are called "user-defined types."

Here is an example of using type and array to create a bank of 8-bit registers:

```
type RegisterArray is array (0 to 7) of std_logic_vector(7 downto 0);  
signal registers : RegisterArray := (others => (others => '0'));
```

In the example above, we are defining a structure that could be used in an entity like a `RegisterBank` which has eight 8-bit registers. These registers are represented by the `registers` array, which has 8 elements, each with a length of 8 bits.

---

Revision #2

Created 2025-09-15 16:24:49 UTC by RE

Updated 2025-09-16 10:24:25 UTC by RE