

Lesson 15

Type and Array

Mohd Saufy Rohmad

Technical Trainer and Consultant

Type

- Type is an enumeration that has an ordered set of values.
- Therefore its easier to call those values whenever the type is declared.
- The package Standard contains declarations of several predefined enumeration types: BIT, BOOLEAN, CHARACTER, SEVERITY_LEVEL, FILE_OPEN_KIND and FILE_OPEN_STATUS.
- Apart from that the package Std_Logic_1164 defines another enumeration type, STD_ULOGIC.
- Some of those predefined enumeration is declared as:

Type

- type severity_level is (note, warning, error, failure);
- type boolean is (false, true); type bit is ('0', '1');
- You can declare your own types also.
- For example, type logic_level is (unknown, low, undriven, high);
- type alu_function is (disable, pass, add, subtract, multiply, divide);
- A literal cant appear in a type twice, for example,
- type NotGood is (X, '0', '1', X); -- illegal
- - is illegal, because X appeared twice.

Subtype

- Subtype distinguishes a subset of values of some type.
- There are two predefined subtypes specified in the package STANDARD: natural and positive. Both are subtypes of the type INTEGER. The package Std_Logic_1164 also contains declarations of subtypes, which are constrained subtypes of the Std_Logic: X01, X01Z, UX01, and UX01Z.
- For example, subtype DIGITS is INTEGER range 0 to 9;
- INTEGER is a predefined type and the subtype DIGITS will constrain the type to ten values only, reducing the size of registers if the specification is synthesized.

Array

- By now, you are probably wondering how to use array in VHDL.
- Any array first have to be declared as a type, for example,
- `type input_data_type is array(1 to 4) of std_logic_vector(7 downto 0);`
- Here, we declared an array called `input_data_type`. The array has 4 elements, declared as (1 to 4). The 4 elements are all `std_logic_vectors` of 8-bits.

Array

- By now, you are probably wondering how to use array in VHDL.
- Any array first have to be declared as a type, for example,
- type input_data_type is array(1 to 4) of std_logic_vector(7 downto 0);
- Here, we declared an array called input_data_type.
- The array has 4 elements, declared as (1 to 4).
- The 4 elements are all std_logic_vectors of 8-bits.

Array Declaration and Access

- We have seen how to declare an array with a type.
- So how do we use the types and arrays in a VHDL entity?
- The types can be declared in the same place where signals and components are declared. For example,

architecture synth of ram_array is

type mem_array is array(7 downto 0) of STD_LOGIC_VECTOR(31 downto 0);

signal mem: mem_array;

signal RDataxDR: STD_LOGIC_VECTOR(31 downto 0);

begin

RDataxDR <= mem(to_integer(RAdrxR));

Array Declaration and Access

- This is a VHDL code snippet that shows how to declare the array.
- The array is declared first as a type where the size of the array (which is 7 down to 0, i.e. 8) and the size of the elements (which is 31 down to 0, i.e. 32 bits) are declared.
- After declaring the array type, we have to assign it to some signal.
- You can think of it as, we created a signal type first, like `std_logic`.
- Then we are assigning a signal with this type.
- Here the signal is `mem`.
- Similar to many other programming language, the elements of the `mem` can be accessed as `mem(1)`, `mem(2)` etc.
- However the output `RDataXDR` and the size of each element of the `mem` (32-bits here) has to match.

Thank You