

University of Bridgeport

INTRO TO VLSI DESIGN  
CPE 448

VHDL  
Tutorial-V

Scalar Data Types and Operations

For:

By: Prakash S. Thapa

The concept of type is very important when describing data in a VHDL model. The type of a data object defines the set of values that the object defines the set of values that the object can assume, as well as the set of operations that can be performed on those values. A scalar type consists of single, indivisible values.

1.    **CONSTANTS and VARIABLES:**

Both constants and variables need to be declared before they can be used in a model. A declaration simply introduces the name of the object, defines its type and may give it an initial value.

```
constant number_of_bytes : integer := 4;  
constant number_of_bits : integer := 8*number_of_bytes;  
constant e : real := 2.718281828;  
constant prop_delay : time := 3 ns;
```

```
variable index: integer := 0;  
variable start, finish : time := 0 ns;
```

**Example 1:**

```
architecture sample of ent is  
    constant pi : real : 3.14159  
begin  
    process is  
        variable counter : integer;  
    begin  
        ■ do anything  
    end process;  
end architecture sample;
```

One restriction on where a variable declaration may occur is that it may not be placed so that the variable would be accessible to more than one process. The exception to this rule is if a variable is declared specially as *shared* variable.

2. SCALAR TYPES:

A scalar type is one whose value are indivisible.

```
type apples is range 0 to 100;  
type oranges is range 0 to 100;
```

Example 1:

```
package int_types is  
    type small_int is range 0 to 255;  
end package int_types;  
  
use work.int_types.all;  
entity smaller_adder is  
    port (    a, b :    in small_int;  
             s    :    out small_int);  
end entity small_adder;
```

3. INTEGER TYPES:

A predefined type integer is included, which includes all whole numbers representable on a particular host computer.

```
type day_of_month is range 0 to 31;
```

```
type year is range 0 to 2100;
```

```
variable today: day_of_month := 19;
```

```
variable start_year: year := 2002;
```

It is illegal to make the assignment:

```
start_year := today;
```

4. FLOATING-POINT TYPES:

**type** *INPUT\_LEVEL* **is range** -10.0 to 10.0;

**type** *probability* **is range** 0.0 to 1.0;

**variable** input\_A: input\_level;

## 5. PHYSICAL TYPES:

### Example 1:

```
type resistance is range 0 to 1E9
  units
    ohm;
    kohm      = 1000ohm;
    Mohm      = 1000kohm;
end units resistance;
```

We can use them as

5 ohm      22 ohm      4kohm

A predefined physical type *time* is very important in VHDL.

```
type time is range implementation defined
  units
    fs;
    ps=1000fs;
    ns=1000ps;
    us=1000ns;
    ms=1000us;
    sec=1000ms;
    min=60sec;
    hr=60min;
end units time;
```

6. ENUMERATION TYPES:

```
type alu_function is (disable, pass, add, subtract, multiply, divide);  
type octal_digit is ( '0', '1', '2', '3', '4', '5', '6', '7');
```

```
variable alu_op: alu_function;  
variable last_digit: octal_digit := '0';
```

and make assignments to them:

```
alu_op      := subtract;  
last_digit  := '7';
```