OVERLOADING

Overloading in VHDL is of two types.

- Subprogram Overloading
- Operator Overloading

Subprogram Overloading

- Sometimes it is convenient to have two or more subprograms with the same name. In such a case, the subprogram name is said to be *overloaded. For example, consider the following two declarations.*
- **function COUNT (ORANGES: INTEGER) return INTEGER;**
- **function COUNT (APPLES: BIT) return BIT;**

Subprogram Overloading Cont..

- Both functions are overloaded since they have the same name, COUNT, and have different parameter types. When a call to either function is made, it is easily possible to identify the exact function to which the call was made from the type of the actual parameters passed. For example, the function call

COUNT(20)

refers to the first function since 20 is of type INTEGER, while the function call

COUNT('1')

refers to the second function, since the type of actual parameter is BIT.

Subprogram Overloading Cont..

- It is also possible for two subprograms to have the same parameter types and result types but have a different number of parameters.
- Here is an example of such a set of functions that determine the smallest value from a set of 2,4, or 8 integers.

```
function SMALLEST (A1, A2: INTEGER)
   return INTEGER;
function SMALLEST (A1, A2, A3, A4:
   INTEGER) return INTEGER;
function SMALLEST (A1, A2, A3, A4, A5, A6,
   A7, A8: INTEGER)
return INTEGER;
```

Subprogram Overloading Cont..

A call such as

. . . SMALLEST (4, 5) . . .

refers to the first function, while the function call

. . . SMALLEST (20, 45,52,1,89,67,91,22)...

refers to the third function

Operator Overloading

- Operator overloading is one of the most useful features in the language.
- When a standard operator symbol is made to behave differently based on the type of its operands, the operator is said to be overloaded.
- The need for operator overloading arises from the fact that the predefined operators in the language are defined for operands of certain predefined types.

Operator Overloading cont..

The operator in the expression

S1 and S2

where S1 and S2 are of type MVL, would then refer to the and operation that was defined by the model writer as a function. The operator in the expression

CLK1 **and CLK2**

where CLK1 and CLK2 are of type BIT, would refer to the predefined and operator.

Operator Overloading cont..

- Function bodies are written to define the behaviour of overloaded operators. Such a function has, at most, two parameters; the first one refers to the left operand of the operator and the second parameter, if present, refers to the second operand. Here are some examples of function declarations for such function bodies.
- **type MVL is ('U', '0', '1', 'Z');**
- **function "and" (L, R: MVL) return MVL;**
- **function "or" (L, R: MVL) return MVL;**
- **function "not" (R: MVL) return MVL;**