Lesson 20

Package and Function

Mohd Saufy Rohmad
Technical Trainer and Consultant

Package and Function

- Package is used in VHDL language to have a modular VHDL design.
- It has many similarities with the headers of C language.
- Package can be used to declare a type, subtype, constant, file, alias, component, attribute, function etc and use them in different VHDL files of a same project.
- I show you an example here of a VHDL package, which includes some constants and datatypes.

```
package Package_MF is
    constant sync_read: integer:= 0;
    constant sync_all: integer:= 0;
    constant sync_selector: integer:= 0;
    constant N: integer:= 32; --16;
    constant INPUTSIZE: integer := 160;
    ---------- load register types ----------
    type loadN is array (1 to 16) of std_logic;
    type load_yMF_type is array(1 to 16) of std_logic;
    type load_Lreg_type is array(1 to 16, 1 to 16) of std_logic;
end Package_MF;
```

Package and Function

- If we include this package in any VHDL files, we can use constants like sync_read or sync_all without declaring them again and again.
- This helps us to write an error free and readable code.
- The same goes for the types of datas, we can use the loadN type of data which has 16 separate 1-bit load values.
- To call this header in another VHDL file, we have to use,
- use work.package_MF.ALL;
- We have seen in the last hour that the components can also be placed in the package.
- If you place the components in a package, you dont have to declare them again, you can just mention their port maps.

```
------------------------------------------------------------
-- package for datapath and controller
------------------------------------------------------------
library ieee ;
use ieee.std_logic_1164.all ;
use work.averager_types.all ;
package averager_components is
    component datapath
    port (
    a, b, c, d : in num ;
    sum : out num ;
    sel : in std_logic_vector (1 downto 0) ;
    load, clear, clk : in std_logic
    ) ;
 end component ;
    component controller
    port (
    update : in std_logic ;
    sel : out std_logic_vector (1 downto 0) ;
    load, clear : out std_logic ;
    clk : in std_logic
    ) ;
    end component ;
end averager_components ;
```

```
------------------------------------------------------
-- Top Entity-----------------------------------------
------------------------------------------------------
library ieee ;
use ieee.std_logic_1164.all ;
use ieee.std_logic_arith.all ;
use work.averager_types.all ;
use work.averager_components.all ;
entity averager is port (
a, b, c, d : in num ;
sum : out num ;
update, clk : in std_logic ) ;
end averager ;
architecture rtl of averager is
    signal sel : std_logic_vector (1 downto 0) ;
    signal load, clear : std_logic ;
-- other declarations (e.g. components) here
begin
    d1: datapath port map ( a, b, c, d, sum, sel, load,
    clear, clk ) ;
    c1: controller port map ( update, sel, load,
    clear, clk ) ;
end rtl ;
```

Package

- Several packages can be used to have a nice VHDL design.
- You can see two packages averager_types and averager_components being used in the top entity in this example.

Function

- As VHDL is a hardware description language, a simple function can be a very powerful statement to express digital logic.
- Just like other languages, the VHDL functions are used to increase reusability for parts of the code.
- I would suggest to use functions for small amount of codes to represent combinational logic, for example, calculating the multiplication or the addition etc.
- Function always has to return a statement and the return value has to be assigned to something. The functions cannot use a wait statement.
- An example of a function to compute the maximum of two values is,

```
function max (n : integer; m : integer) return integer is
begin
    if n > m then
      return n;
    else
      return m;
    end if;
end max;
```

Thank You