CprE 381: Computer Organization and Assembly Level Programming

Lab Week 3 VHDL

Henry Duwe
Electrical and Computer Engineering
Iowa State University

Model Truth Table Part I

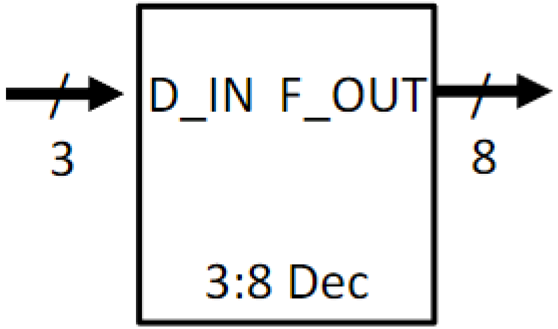
Use with select… when statement (FreeRange VHDL Listing 4.13 for full file)

```
entity my_4t1_mux is
   port (D3,D2,D1,D0 : in std_logic;
         SEL : in std_logic_vector(1 downto 0);
         MX_OUT : out std_logic);
end my_4t1_mux;
architecture mux4t1 of my_4t1_mux is
begin
   with SEL select
   MX_OUT <= D3 when “11”,
             D2 when “10”,
             D1 when “01”,
             D0 when “00”,
             ‘0’ when others;
end mux4t1;
```

Model Truth Table Part II

Use with select … when statement
(FreeRange VHDL Listing 13.10 for more information)

```
with D_IN select
   F_OUT <=  “00000001” when “000”,
             “00000010” when “001”,
             “00000100” when “010”,
             “00001000” when “011”,
             “00010000” when “100”,
             “00100000” when “101”,
             “01000000” when “110”,
             “10000000” when “111”,
            “00000000” when others;
```



Others

Use with others assignment (FreeRange VHDL 10.2 for more information)

```
with SEL select
   s_mux_result <= D0 when ‘0’,
                   D1 when ‘1’,
                   (others => ‘0’) when others
```

- The model for the 2:1 MUX uses the terminology (others => '0'). This is a short-hand terminology for assigning all of the outputs to '0'.

- The neat part about this instruction is that you do not need to know how many 0's you need to write. **If the width of the associated bundle were to change, this particular line of code would not need to be modified.**

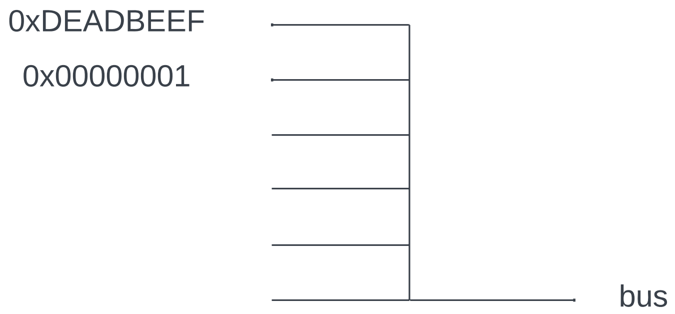
User-Defined Types

Use Custom types for simplification and organization (FreeRange VHDL 11.6 for more information)

```
type bus_t is array(3 downto 0) of std_logic_vector(31 downto 0);

Signal mux_bus : bus_t;

mux_bux(0) <= x”DEADBEEF”;
Mux_bus(1) <= x”00000001”;
```



- This syntax is handy for large groups of signals and can reduce clutter within your code. **Can be especially powerful when paired with generate loops to individually connect multiple components.**

Packages Part I

my_package.vhd

```
package my_package is
    constant four_bit_zero : std_logic_vector := "0000";
    type t_bus_2x32 is array (0 to 1) of std_logic_vector(31 downto 0);
end package my_package;
```

- Similar to header file in C, you can define custom data types, set constants, or even define functions

- This file will need to be referenced in your subsequent component files to be useful **and it always needs to be compiled before the dependent files**.

User-Defined Types

Mux32b2t1.vhd

```
use work.my_package.all;
use IEEE.numeric.std. all;

entity big_mux is
      port( data : in t_bus_2x32;
              sel   : in std_logic_vector(0 downto 0);
              out  : out std_logic_vector(31 downto 0));
end big_mux;

architecture dataflow of big_mux is
begin
   out <=data(to_integer(unsigned(sel)));
end dataflow
```

***Notice we had to import our package use the syntax "use work.my_package.all;".**