1.11 VHDL Examples

This is an alphabetical list of all keywords that support synthesis:

- A (abs, all, alias, and, architecture, array, attribute)
- B (begin, block, body, buffer)
- C (case, component, configuration, constant)
- D (downto)
- E (else, elsif, end, entity, exit)
- F (for, function)
- G (generate, generic, group)
- I (if, in, inout, is)
- L (library, literal, loop)
- M (map, mod)
- N (name, next, nor, not, null)
- O (of, or, others, out)
- P (package, port, procedure, process)
- R (range, record, rem, return, rol, ror)
- S (select, signal, sla, slr, srl, subtype)
- T (then, to, type)
- U (until, use)
- V (variable)
- W (wait, when, while, with)
- Z (xor, xnor)

This is a list of all supported operators:

1. Logical (and, or, NAND, NOR, XOR, XNOR)
2. Relational (=, /=, <, <=, >, >=)
3. Shift (SLL, SLR, SLA, SRA, SRL, XOR)
4. Adding (+, -)
5. Unary signing (+, -)
6. Multiplying (*, /, mod, rem)
7. Miscellaneous (**, abs, not)

Note: Without parentheses, operators in group 7 have the highest precedence, followed by
operators in group 6, etc. down to the lowest precedence or group 1. Within each group there is
no operator precedence, and precedence must be established by parentheses. In an
expression without parentheses, operators are applied left to right.

The following are examples of design styles in an architecture declaration that you should know
how to use with single signals. The design entity is for a simple 2-input OR gate with std_logic
data type (or scalar data type) for the inputs A and B, and the output P as shown in Figure V1. A
simulation waveform that shows the correct functionality of the simple 2-input OR gate is also
included.
Library example:

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

Entity example:

entity BB is port(
    A, B, C: in std_logic;
    D: inout std_logic_vector(0 to 3)
); end BB;

Note: the if statement is the sequential equivalent of the when-else statement.

Case statement:
- place the case statement between begin and end process
  case A is
    when '00' => D := "0001";
    when '01' => D := "0010";
    when '10' => D := "0100";
    when '11' => D := "1000";
    when others => null;
  end case.

Note: the case statement is the sequential equivalent of the with-select-when statement.

Component declaration:

component DEC is port(
    s0, s1, s2, s3: in std_logic;
    o0, o1, o2, o3: out std_logic
); end component;

Component instantiation:

C1: DEC port map (s0 => A(0), s1 => A(1), s2 => D(0), o1 => D(1), o2 => D(2), o3 => D(3));

The waveforms shown in Waveform V2 represent the correct functionality of the design of the
sample 2-to-4 Decoder.

Library example:

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

Entity example:

entity BB is port(
    A, B, C: in std_logic;
    D: inout std_logic_vector(7 downto 0);
    E: out std_logic_vector(0 to 3)
); end BB;