Summary:

Entity Declaration:

ENTITY entity_name IS
  PORT ( signal_name, signal_name, ... : signal_mode signal_type;
         signal_name, signal_name, ... : signal_mode signal_type;
         ...
         signal_name, signal_name, ... : signal_mode signal_type);
END entity_name;

Signal types discussed in book (p.687, etc.)
  BIT, BIT_VECTOR, STD_LOGIC, STD_LOGIC_VECTOR,
  STD_ULOGIC, SIGNED, UNSIGNED, INTEGER, BOOLEAN,
  ENUMERATION.

Signal modes: (Table A2, p. 695)
  • IN: input signal only
  • OUT: output signal only; cannot be used on the right side of the
    simple signal assignment operator <=
  • INOUT: a signal that is both an input and an output.
  • BUFFER: an output signal. However, it can be used on the right
    side of the simple signal assignment operator <=
Statement in the Architecture body can be **structural** or **behavioral**.

**PORT MAP** statement is the only structural type statement in VHDL, used to instantiate a component that has been declared using a component declaration.

Simple **signal assignment**: `signal_name <= expression;`

Selected **signal assignment**: **WITH-SELECT-WHEN** statement

Conditional **signal assignment**: **WHEN-ELSE** statement

**PROCESS** statement (contains **sequential** statements)
- Simple signal assignment statement `<=`
- Variable assignment statement `:=`
- **IF-THEN-ELSE** statement
- **CASE-WHEN** statement
- **FOR-LOOP** and **WHILE-LOOP** statements
LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY mux4to1Equations IS
  PORT ( w0, w1, w2, w3 : IN STD_LOGIC;
          s  : IN STD_LOGIC_VECTOR(1 DOWNTO 0);
          f : OUT STD_LOGIC );
END mux4to1Equations;

ARCHITECTURE Behavior OF mux4to1Equations IS
BEGIN
  f <=
    (NOT s(1) AND NOT s(0) AND w0)
    OR (NOT s(1) AND  s(0) AND w1)
    OR ( s(1) AND NOT s(0) AND w2)
    OR ( s(1) AND  s(0) AND w3);
END Behavior;
LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY mux4to1 IS
  PORT (    w0, w1, w2, w3 : IN   STD_LOGIC;
          s : IN   STD_LOGIC_VECTOR(1 DOWNTO 0);
          f : OUT STD_LOGIC );
END mux4to1;

ARCHITECTURE Behavior OF mux4to1 IS
BEGIN
  WITH s SELECT
    f <= w0 WHEN "00",
          w1 WHEN "01",
          w2 WHEN "10",
          w3 WHEN OTHERS;
END Behavior;

Figure 6.28 VHDL code for a 4-to-1 multiplexer
LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY mux4to1When IS
  PORT ( w0, w1, w2, w3 : IN STD_LOGIC;
         s     : IN STD_LOGIC_VECTOR(1 DOWNTO 0);
         f     : OUT STD_LOGIC );
END mux4to1When;

ARCHITECTURE Behavior OF mux4to1When IS
BEGIN
  f <= w0 WHEN s ="00" ELSE w1 WHEN s ="01" ELSE w2 WHEN s ="10" ELSE w3;
END Behavior;
LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY dec2to4Equations IS
  PORT ( w : IN STD_LOGIC_VECTOR(1 DOWNTO 0);
         En : IN STD_LOGIC;
         y : OUT STD_LOGIC_VECTOR(0 TO 3) );
END dec2to4Equations;

ARCHITECTURE Behavior OF dec2to4Equations IS
  SIGNAL Enw : STD_LOGIC_VECTOR(2 DOWNTO 0);
BEGIN
  y(0) <= En AND NOT w(1) AND NOT w(0);
  y(1) <= En AND NOT w(1) AND w(0);
  y(2) <= En AND w(1) AND NOT w(0);
  y(3) <= En AND w(1) AND w(0);
END Behavior;
LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY dec2to4When IS
  PORT ( w : IN STD_LOGIC_VECTOR(1 DOWNTO 0);
         En : IN STD_LOGIC;
         y  : OUT STD_LOGIC_VECTOR(0 TO 3) );
END dec2to4When;

ARCHITECTURE Behavior OF dec2to4When IS
BEGIN
  y <= "1000" WHEN En = '1' AND W="00" ELSE
       "0100" WHEN En = '1' AND W="01" ELSE
       "0010" WHEN En = '1' AND W="10" ELSE
       "0001" WHEN En = '1' AND W="11" ELSE
       "0000";
END Behavior;
LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY dec2to4 IS
    PORT ( w : IN STD_LOGIC_VECTOR(1 DOWNTO 0);
           En : IN STD_LOGIC;
           y : OUT STD_LOGIC_VECTOR(0 TO 3))
END dec2to4;

ARCHITECTURE Behavior OF dec2to4 IS
    SIGNAL Enw : STD_LOGIC_VECTOR(2 DOWNTO 0);
BEGIN
    Enw <= En & w;
    WITH Enw SELECT
    y <= "1000" WHEN "100",
         "0100" WHEN "101",
         "0010" WHEN "110",
         "0001" WHEN "111",
         "0000" WHEN OTHERS;
END Behavior;
LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY priority IS
  PORT ( w : IN STD_LOGIC_VECTOR(3 DOWNTO 0);
        y : OUT STD_LOGIC_VECTOR(1 DOWNTO 0);
        z : OUT STD_LOGIC );
END priority;

ARCHITECTURE Behavior OF priority IS
BEGIN
  y <= "11" WHEN w(3) = '1' ELSE
       "10" WHEN w(2) = '1' ELSE
       "01" WHEN w(1) = '1' ELSE
       "00";
  z <= '0' WHEN w = "0000" ELSE '1';
END Behavior;

Figure 6.32. VHDL code for a priority encoder.