4.1)

Not is a higher precedence than the other logical operators.

And or nand nor xor xnor have the same precedence.

Defined in std\_logic, they are overloaded from std\_ulogic.

The associative logical operators are and, or, xor, xnor.

4.3)

x <= (not a) and (not b) and (not c) and (not d) ;

4.4)

The logical operations are performed by position, left to right, instead of index. The two compared arrays must be of the same length, same with their output.

4.6)

Concurrent signal assignment using boolean expressions, as well as selected and conditional signal assignments

4.8)

Attached

4.12)

Arch  
begin

with fuel select  
 led <= “0000” when “1100” | “1101” | “1110” | “1111”;  
 “1000” when “1000” | “1001” | “1010” | “1011” ;  
 “1100” when “0100” | “0101” | “0110” | “0111”;  
 “1110” when “0001” | “0010” | “0011”;  
 “1111” when others;

4.14)

It is evaluated bitwise starting with the MSB.

4.15)

Entity  
 portmap(a,b,c : in std\_logic;  
 x,y : out std\_logic);

Arch  
begin  
 x<= ‘1’ when (a = ‘0’ and ((b = “0” and c = “1”) or (b = “1” and c = “0”)) or a = “1” and  
 ((b = “0” and c = “0”)) or (b = ”1” and c = ”1”))) else ‘0’;  
 y <= ‘1’ when (a = ‘1’ and (( b = ‘0’ and c = ‘1’) or (b = ’1’ and c =’0’)) or a = ’0’ and  
 b = ‘0’ and c = ‘0’) else ‘0’ ;

end

4.17) A comparator has two 4-bit inputs a and b and two outputs. Output equal is asserted only if the two inputs are equal. Output cmpl is asserted only if the two inputs are the complement of each other. Write an entity declaration for the comparator. Write an architecture named selected that implements each output with a separate selected signal assignment statement. Write an alternative architecture named conditional that implements each output with a separate conditional signal assignment statement.

Entity

Portmap(a,b : in std\_logic\_vector(3 downto 0);

Equal, cmpl : out std\_logic);

Architecture conditional is

Begin

Equal <= ‘1’ when a = b else ‘0’;  
 cmpl <= ‘1’ when ((a and not b) = “1111”) else ‘0’;

End conditional;

4.21)

Arch  
begin  
 with i3 & i2 & i1 & i0 select  
 a<= “00” when “0000” | “0001” | “0010” | “0011” | “0100” | “0101” | “0110” | “0111”;  
 “01 when “1000” | “1001” | “1010” | “1011” ;  
 “10” when “1100” | “1101” ;  
 “11” when “1110” | “1111”;  
 “XX” when others;  
 end arch;

4.26)

Arch selected therm2bin is  
 begin  
 with in selected  
 bin <= “000” when “0000000”;  
 “001” when “0000001”;  
 “010” when “000001-“;  
 “011” when “00001—“;  
 “100” when “0001---“;  
 “101” when “001----“;  
 “110” when “01-----“;  
 “111” when “1------“;  
 “XXX” when others;  
 end arch  
  
 Arch conditional therm2bin is  
 begin  
 bin<= “111” when (in(7) = 1) else,  
 “110” when ((in(7) = 0) and (in(6) = 1)) else,  
 “101” when ((in(7 downto 6) = “00”) and (in(5) = 1)) else,  
 “100” when ((in(7 downto 5) = “000”) and (in(4) = 1)) else,  
 “011” when ((in(7 downto 4) = “0000”) and (in(3) = 1)) else,  
 “010” when ((in(7 downto 3) = “00000”) and (in(2) = 1)) else,  
 “001” when ((in(7 downto 2) = “000000”) and (in(1) = 1)) else,  
 “000” when ((in(7 downto 1) = “0000000”) and (in(0) = 1)) else “XXX”;

End arch;