CPE462 - VHDL: Simulation and Synthesis - Fall'11

Prof. Nuno Alves (nalves@wne.edu), College of Engineering

Homework Assignment #3

Due Date: Monday, September 26th 2011

1.

part a & b

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_unsigned.all;

entity ex03 is
end entity;

architecture myarch of ex03 is
      type array1 is array (7 downto 0) of std_logic;
      type array2 is array (3 downto 0, 7 downto 0) of std_logic;
      type array3 is array (3 downto 0) of array1;

      signal a: bit; -- scalar
      signal b: std_logic; -- scalar
      signal x: array1; -- 1D
      signal y: array2; -- 2D
      signal w: array3; -- 1D x 1D
      signal z: std_logic_vector(7 downto 0); -- 1D
begin

      a <= '1';
      b <= 'X';
      x <= "00000000";
      y <= (
      ('0','0','0','0','0','0','0','0'),
      ('0','0','0','0','0','0','0','0'),
      ('0','0','0','0','0','0','0','0'),
      ('0','0','0','0','0','0','0','0')
      );
      z <= "XXXXXXXZ";

end architecture;
```

c.

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_unsigned.all;

entity ex03 is
end entity;

architecture myarch of ex03 is
     type array1 is array (7 downto 0) of std_logic;
     type array2 is array (3 downto 0, 7 downto 0) of std_logic;
     type array3 is array (3 downto 0) of array1;

     signal a: bit; -- scalar
     signal b: std_logic; -- scalar
     signal x: array1; -- 1D
     signal y: array2; -- 2D
     signal w: array3; -- 1D x 1D
     signal z: std_logic_vector(7 downto 0); -- 1D
begin

     -- scalar <= scalar ; illegal (bit <= std_logic)
     --a <= x(2);

     -- scalar <= scalar ; legal (std_logic <= std_logic)
     b <= x(2);

     -- scalar <= scalar ; legal (std_logic <= std_logic)
     b <= y(3,5);

     -- scalar <= scalar ; illegal (row 5 does not exist)
     --b <= w(5)(3);

     -- scalar <= scalar ; illegal incorrect y index notation
     --y(1)(0) <= z(7);

     -- scalar <= scalar ; legal (same scale and datatype)
     x(0) <= y(0,0);

     -- 1D <= 1D ; legal assignment
     x<="1110000";

     -- 2D <= 1D ; ilegal assignment (different scales)
     --y<="0000000";

     -- 2D row <= 1D ; illegal with incompatible types
     --y(1) <= x;

     -- 1D array row <= 2D array ; illegal with incompatible types
     --w(0) <= y;
end architecture;
```

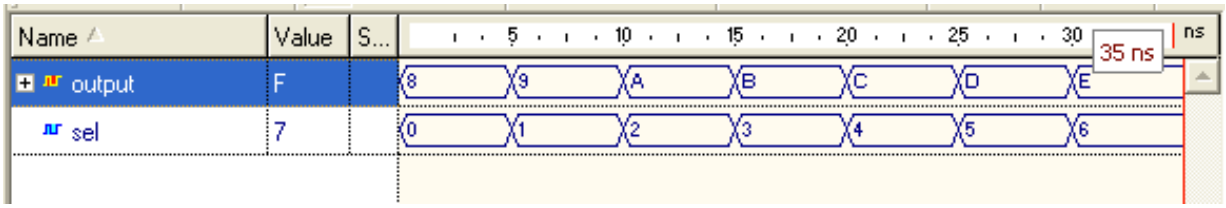
2.

main code

```
library ieee;
use ieee.std_logic_1164.all;

entity test is
     port (sel  : in integer;
            output : out std_logic_vector(3 downto 0)
     );
end entity;

architecture myarch of test is
type rom is array (7 downto 0) of std_logic_vector(3 downto 0);
constant my_rom: rom := ("0000", "0001", "0010", "0011",
                                  "0100", "0101","0101", "0110");
begin

     output<=my_rom(sel);

end architecture;
```

testbench

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_unsigned.all;

entity testrom is
end;

architecture bench of testrom is
  component romblock
     port (sel  : in integer;
            output : out std_logic_vector(3 downto 0));
  end component;

  signal sel : integer := 0;
  signal output : std_logic_vector(3 downto 0);

begin
 sel   <= 0 ,
 1 after 5 ns  , 2 after 10 ns ,
 3 after 15 ns , 4 after 20 ns ,
 5 after 25 ns , 6 after 30 ns,
 7 after 35 ns ;

 m: romblock port map (sel,output);

end bench;
```



3.

main code with STD_LOGIC_VECTOR data types

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
use ieee.std_logic_arith.all;

entity adder1 is
      port (a, b : in STD_LOGIC_VECTOR (3 downto 0);
            sum    : out STD_LOGIC_VECTOR (3 downto 0));
end entity;

architecture adder1 of adder1 is
begin

      sum <= a + b;

end architecture;
```