Csc343:
Assignment for 9/30/05.

a). I did a two inputs (each input) with 4 bits multiplexor, you are responsible for
   1. a two inputs (each input) with 32 bits multiplexor.
   2. a three inputs (each input) with 32 bits multiplexor.
   3. a four inputs (each input) with 32 bits multiplexor.
   4. a thirty-two inputs (each input) with 32 bits multiplexor.
b). get to understand the 2-4 decoder and design your 3-8 decoder.
c). write a report and print the wave forms and discuss what you have learned from the assignment.

Example of multiplexors:

Example of decoders:
library ieee;
use ieee.std_logic_1164.all;

entity Multiplexor is
port(   I1: in std_logic_vector(3 downto 0);
       I0: in std_logic_vector(3 downto 0);
       S:  in std_logic;
       O:  out std_logic_vector(3 downto 0)
    );
end Multiplexor;

architecture behv of Multiplexor is
begin
    process(I1,I0,S)
    begin
        case S is
        when '0' =>         O <= I0;
        when '1' =>          O <= I1;
        when others=>        O <="ZZZZ";
        end case;
    end process;
end behv;

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use work.all;

entity test_multiplexor is 
end test_multiplexor;
end test_multiplexor;

architecture TB of test_multiplexor is

  component multiplexor
    port(
      I1: in std_logic_vector(3 downto 0);
      I0: in std_logic_vector(3 downto 0);
      S: in std_logic;
      O: out std_logic_vector(3 downto 0)
    );
  end component;

  signal T_I1: std_logic_vector(3 downto 0):="0000";
  signal T_I0: std_logic_vector(3 downto 0):="0000";
  signal T_O: std_logic_vector(3 downto 0);
  signal T_S: std_logic;

  begin

    U_multiplexor: multiplexor port map (T_I1, T_I0, T_S, T_O);

    process

      variable err_cnt: integer :=0;

      begin

        T_I1 <= "0101";
        T_I0 <= "1111";

        -- case s ='0'
        wait for 30 ns;
        T_S <= '0';
        wait for 1 ns;


end TB;
assert (T_O="1111") report "Error Case 0" severity error;
  if (T_O/="1111") then
    err_cnt := err_cnt+1;
  end if;

-- case s = '1'
wait for 30 ns;
T_S <= '1';
wait for 1 ns;
assert (T_O="0101") report "Error Case 1" severity error;
  if (T_O/="0101") then
    err_cnt := err_cnt+1;
  end if;

-- case S = 'U'
  wait for 30 ns;
  T_S <= 'U';

-- summary of all the tests
if (err_cnt=0) then
  assert (false)
  report "Testbench of multiplexor completed sucessfully!"
  severity note;
else
  assert (true)
  report "Something wrong, try again!"
  severity error;
end if;

wait;

end process;

end TB;
configuration CFG_TB of test_multiplexor is
   for TB
   end for;
end CFG_TB;

library ieee;
use ieee.std_logic_1164.all;

entity decoder is
  port( I: in std_logic_vector(1 downto 0);
       O: out std_logic_vector(3 downto 0)
  );
end decoder;

architecture behv of decoder is
begin
process (I)
begin

    case I is
        when "00" =>      O <= "0001";
        when "01" =>      O <= "0010";
        when "10" =>      O <= "0100";
        when "11" =>      O <= "1000";
        when others =>    O <= "XXXX";
    end case;

end process;
end behv;

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use work.all;

entity test_decoder is
end test_decoder;

architecture TB of test_decoder is

    component decoder
        port(I: in std_logic_vector(1 downto 0);
        O: out std_logic_vector(3 downto 0))
    );
    end component;

    signal T_I: std_logic_vector(1 downto 0):="00";
    signal T_O: std_logic_vector(3 downto 0);

begin

    U_decoder: decoder port map (T_I, T_O);

process

    variable err_cnt : integer := 0;

begin

    -- case I="00"
    wait for 20 ns;
    T_I <= "00";
    wait for 1 ns;
    assert (T_O="0001") report "Error Case 0" severity error;
    if (T_O/="0001") then
        err_cnt := err_cnt + 1;
    end if;

    -- case I="01"
    wait for 20 ns;
    T_I <= "01";
    wait for 1 ns;
    assert (T_O="0010") report "Error Case 1" severity error;
    if (T_O/="0010") then
        err_cnt := err_cnt + 1;
    end if;

    -- case I="10"
    wait for 20 ns;
    T_I <= "10";
    wait for 1 ns;
    assert (T_O="0100") report "Error Case 2" severity error;
    if (T_O/="0100") then
        err_cnt := err_cnt + 1;
    end if;

    -- case I="11"
wait for 20 ns;
T_I <= "11";

wait for 1 ns;
assert (T_O="1000") report "Error Case 3" severity error;
if (T_O="/1000") then
    err_cnt := err_cnt + 1;
end if;

-- case I="UU"
wait for 20 ns;
T_I <= "UU";

-- summary of all the tests
if (err_cnt=0) then
    assert false
    report "Testbench of decoder completed successfully!"
    severity note;
else
    assert true
    report "Something wrong, try again"
    severity error;
end if;

wait;

end process;

end TB;

configuration CFG_TB of test_decoder is
    for TB
    end for;
end CFG_TB;