Appendix-II

VHDL Code

-- VHDL program for source flow rate control (Chapter-3)
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
-- in1 shows node1's data, in2 shows node2's data, in3 shows node3's data.
-- clk = 25MHz for XC3E100, CP132 xilinxFPGA board.
-- divclk1, divclk2 and divclk3 shows reduced source data rate
-- status of a shows then number of transmitting nodes in the network
entity new1 is
  Port ( a : in STD_LOGIC_vector(1 downto 0);
         divclk1,divclk2,divclk3,divclk4,clk: in STD_LOGIC;
         in1 : inout STD_LOGIC_vector(3 downto 0);
         in2 : inout STD_LOGIC_vector(3 downto 0);
         in3 : inout STD_LOGIC_vector(3 downto 0);
         buffer1 : inout STD_LOGIC_vector(11 downto 0);
         myclk : inout STD_LOGIC);
end new1;
architecture Behavioral of new1 is
begin
in1 <= "1011";-- node1
in2 <= "0100";-- node2
in3 <= "1100";-- node3
-- Process to divide the source clock frequency 12.5MHz
Process(clk)
variable counter1: integer range 0 to 1;
begin
if clk'event and clk = '1' then
counter1 := counter1 + 1;
if counter1 = 1 then
divclk1 <= not(divclk1);
counter1 := 0;
end if;
end if;
end process;
-- Process to divide the source clock frequency 6.255MHz
Process(divclk1)
variable counter2: integer range 0 to 1;
begin
if divclk1'event and divclk1 = '1' then
counter2 := counter2 + 1;
if counter2 = 1 then
divclk2 <= not(divclk2);
counter2 := 0;
end if;
end if;
end process;
end if;
end if;
end process;

-- Process to divide the source clock frequency 6.255MHz
Process(divclkl)
variable counter2: integer range 0 to 1;
begn
if divclkl'event and divclkl = '1' then
counter2 := counter2 + 1;
if counter2 = 1 then
divclk2 <= not(divclk2);
counter2 := 0;
end if;
end if;
end if;
end process;

-- Process to divide the source clock frequency 3.127MHz
Process(divclk2)
variable counter3: integer range 0 to 1;
begn
if divclk2'event and divclk2 = '1' then
counter3 := counter3 + 1;
if counter3 = 1 then
divclk3 <= not(divclk3);
counter3 := 0;
end if;
end if;
end if;
end process;
-- Process to select particular source rate according
-- to the number of transmitting nodes in the network
process(clk)
begin
if clk'event and clk = '1' then
if a = "00" then
myclk <= divclk1;
elseif a = "01" then
myclk <= divclk2;
elseif a = "10" then
myclk <= divclk3;
elseif a = "11" then
myclk <= divclk4;
end if;
end if;
end process;
-- Process to queue data into the buffer
process(myclk)
variable j: integer range 0 to 2;
variable i: integer range 0 to 7;
begin
if myclk'event and myclk = '1' then
if i = 0 then
buffer1(i) <= in1(i);
buffer1(i+1) <= in2(i);
buffer1(i+2) <= in3(i);
i := i+3;
elsi if i = 3 then
  buffer1(i) <= in1(i-2);
  buffer1(i+1) <= in2(i-2);
  buffer1(i+2) <= in3(i-2);
i := i+3;
elsi if i = 6 then
  buffer1(i) <= in1(i-4);
  buffer1(i+1) <= in2(i-4);
  buffer1(i+2) <= in3(i-4);
i := i+3;
elsi if i = 9 then
  buffer1(i) <= in1(i-6);
  buffer1(i+1) <= in2(i-6);
  buffer1(i+2) <= in3(i-6);
i := 0;
end if;
end if;
--end loop;
--end loop;
end process;
end Behavioral;
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

-- in1, in2 and in3 shows data set of node1, node2 and node3 respectively.
-- clk = 25MHz for XC3E100, CP132 xilinx FPGA board.
-- divclk1, divclk2 and divclk3 shows reduced source data rate.
entity congestion2 is
  Port (ini : input STD_LOGIC_vector(3 downto 0);
in2 : inout STD_LOGIC_vector(3 downto 0);
in3 : inout STD_LOGIC_vector(3 downto 0);
a : inout STD_LOGIC_vector(2 downto 0);
divclk1, divclk2, divclk3, myclk, congestion, admission:
inout std_logic;
buffer1 : inout STD_LOGIC_vector(11 downto 0);
clk : in STD_LOGIC);
end congestion2;
architecture Behavioral of congestion2 is
begin
ini <= "1011";-- node1
in2 <= "0100";-- node2
in3 <= "1100";-- node3
-- Process to divide the source clock frequency 12.5MHz
Process(clk)
variable counter1 : integer range 0 to 1;
begin
if clk'event and clk = '1' then
  counter1 := counter1 + 1;
  if counter1 = 1 then
    divclk1 <= not(divclk1);
    counter1 := 0;
  end if;
end if;
end process;

-- Process to divide the source clock frequency 6.255MHz
Process(divclk1)
variable counter2: integer range 0 to 1;
begin
if divclk1'event and divclk1 = '1' then
  counter2 := counter2 + 1;
  if counter2 = 1 then
    divclk2 <= not(divclk2);
    counter2 := 0;
  end if;
end if;
end process;

-- Process to divide the source clock frequency 6.255MHz
Process(divclk1)
variable counter2: integer range 0 to 1;
begin
if divclk1'event and divclk1 = '1' then
counter2 := counter2 + 1;
if counter2 = 1 then
divclk2 <= not(divclk2);
counter2 := 0;
end if;
end if;
end process;
-- Process to divide the source clock frequency 3.127MHz
Process(divclk2)
variable counter3: integer range 0 to 1;
begin
if divclk2'event and divclk2 = '1' then
counter3 := counter3 + 1;
if counter3 = 1 then
divclk3 <= not(divclk3);
counter3 := 0;
end if;
end if;
end if;
end process;
-- Process to select particular source rate according
-- to the number of transmitting nodes in the network
process(clk)
begin
if clk'event and clk = '1' then
if a = "00" then
myelk <= divclk1;
end if;
end if;
end process;
elsif a = "01" then
myclk <= divclk2;
elsif a = "10" then
myclk <= divclk3;
elsif a = "11" then
myclk <= divclk4;
end if;
end if;
end process;

-- logic to grant/reject admission of new node
process(myclk)
variable buflength: integer range 0 to 15;
variable bandwidth: integer range 0 to 24;
variable N : natural := 2;-- number of transmitting nodes
begin
if myclk'event and myclk = '1' then
bandwidth := 25 - (2*N);
buflength := N*4;
if bandwidth < 2 or buflength > 12 or congestion = '1' then
admission <= '0';
else
admission <= '1';
end if;
end if;
end process;
end Behavioral;