8404129 Laitteistonkuvauskielet/ Hardware Description Languages
Tentti/ Exam
18.12.2003

Vastaa kaikkiin kysymyksiin samalla kielellä, joko suomeksi tai englanniksi. /
Use the same language, either english or finnish, through the exam.

1. Ovatko seuraavat väittämät oikein vai väärin? (oikea vastaus 1 piste, ei vastausta 0 pistettä, väärä vastaus -0,5 pistettä)
   Are the following statements true or false? (right answer 1 point, no answer 0 point, wrong answer -0,5 point)
   a) Verilogia voidaan käyttää vain täysin synkronisten piirien kuvaamiseen. /
   Verilog can be used only when describing fully synchronous circuits.
   b) VHDL:n if-lause suoritetaan sekventiaalisesti. /
   if statement is executed sequentially in VHDL.
   c) std_logic- ja std_ulogic-tyypeille on määritelty resoluutiofunktio. /
   Types std_logic and std_ulogic have a resolution function.
   d) VHDL:n signaali ja muuttuja ovat käytännössä sama rakenne/
   There is no difference between VHDL's signal and variable.
   e) Verilogin always-lauseella kuvataan prosessia. /
   always statement describes a process in Verilog.
   f) Verilogissa ei voida määrittää viiveitä komponenteille ja signaaleille. /
   Delays can not be defined for components and signals in Verilog.

2. a) Selitä VHDL-kielen prosessi. / Explain what is a VHDL process.
   (3 pistettä / 3 points)
   b) VHDL:n procedure- ja function-rakenteet. Mitä yhtäläisyyksiä ja eroja niillä on? /
   VHDL's procedure and function. What are the similarities and differences?
   (3 pistettä / 3 points)

3. Mikä on testipenkki? Mitkä ovat hyvän testipenkin ominaisuudet? /
   What is a test bench? What are the properties of a good test bench?
   (6 pistettä / 6 points)

4. Kirjoita kolme synkronista lohkoa asynkronisella resetillä. Lohkot toteuttavat AND operaation data sisäänmenoille. Kirjoita yksi lohko jokaisella käsitellyllä kielellä (VHDL, Verilog ja SystemC).
   Write three synchronous blocks with asynchronous reset. The blocks implements AND operation to data inputs. Write one block with each language (VHDL, Verilog and SystemC).
   (2 pistettä / kieli, 2 points / language)

5. Seuraavan sivun VHDL-koodi toteuttaa erään toiminnon, minkä? Kerro miten kyseinen koodi muodostaa toiminnon. Käytä rivinumeroita apuna vastauksessa. /
   The VHDL in the next page implements a certain functionality. What is that functionality? Using the line numbers explain how the code does the job.
   (6 pistettä / 6 points)

```
1  --------------------------------------------------------------------------------
2  --  Designer:
3  --
4  --  Generics: noname    : ## CENCORED
5  --  Inputs:     input1 :  ## CENCORED
6  --                  input2 : ## CENCORED
7  --                  input3 : ## CENCORED
8  --                  input4 : ## CENCORED
9  --  Outputs:  output1 : ## CENCORED
10 --                 output2 : ## CENCORED
11 --
12 --------------------------------------------------------------------------------
13 library IEEE;
14 use IEEE.std_logic_1164.all;
15 use IEEE.std_logic_arith.all;
16
17 entity sandman is
18    generic(noname   : integer:= 5);
19    port (input1 : in   std_logic;
20             input2 : in   std_logic;
21             input3 : in   std_logic;
22             input4 : in   std_logic;
23            output1 : out  std_logic;
24            output2 : out  std_logic_vector(noname-1 downto 0));
25 end sandman;
26
27 architecture enter of sandman is
28   type status is (aaa,bbb,ccc);
29   signal CurrentState : status;
30   signal CurrentBitPos : natural range 0 to noname+1;
31   signal NextOut : std_logic_vector(noname-1 downto 0);
32 begin
33
34   process (input2,input4)
35     begin
36       if input4 = '0' then
37           output2 <= (others=> '0');
38           CurrentState <= aaa;
39           CurrentBitPos <= 0;
40           NextOut <= (others => '0');
41
42      elsif input2'event and input2 = '1' then
43        Case CurrentState is
44        when aaa =>
45            output1 <= '0';
46            NextOut <= (others => '0');
47            if input3 ='1' then CurrentState <= bbb;
48            else CurrentState <= aaa;
49            end if;
50        when bbb =>
51            output1 <= '0';
52            if CurrentBitPos = noname then CurrentState <= ccc;
53            else NextOut(CurrentBitPos) <= input1;
54                 currentbitpos <= currentbitpos +1;
55            end if;
56        when ccc =>
57            CurrentState <= aaa;
58            output1 <= '1';
59            CurrentBitPos <= 0;
60            output2 <= NextOut;
61        end Case;
62      end if;
63    end process;
65 end enter;
```