

Compito di linguaggi di descrizione dell'hardware

Esercizio 1

Si realizzi un modello comportamentale in VHDL di un D-latch trasparente con ingressi d , clk e uscita q . Il latch é nello stato di sample quando $clk = 1$. Il modello ha anche i seguenti parametri passati come generic: i) tempo di setup tsu ; ii) tempo di risposta tr ; iii) tempo di decadimento dell'uscita tq . Il terzo parametro rappresenta il tempo massimo per cui il latch puó rimanere nello stato di hold senza che l'uscita si degradi portandosi a X (é un fenomeno presente nei latch dinamici).

Soluzione

```
library IEEE;
use IEEE.std_logic_1164.all;

entity dlatch is
  generic (tsu, tr, tq: time);
  port (d,clk: in std_logic;
        q: out std_logic);
end entity dlatch;

architecture behav of dlatch is
begin
  process (d,clk)
  begin
    if (clk='1') then
      q <= d after tr;
    elsif ((clk'event) and (clk='0')) then
      q <= 'X' after tq;
      if (not (d' stable(tsu))) then
        q <= 'X' after tr;
      end if;
    end if;
  end process;
end architecture;
```

Esercizio 2

Si realizzi la descrizione comportamentale di una rete combinatoria che riceve in ingresso una parola (in std logic) $a_{7..0}$ rappresenta un numero naturale $0 \leq A \leq 99$ nella codifica BCD (i.e. $a_{7..4}$ codifica in binario la cifra corrispondente alle decine d e $a_{3..0}$ codifica le unità u). Compito della rete é produrre in uscita il numero binario corrispondente senza usare moltiplicazioni. A questo riguardo di utilizzi la seguente

relazione: $A = 10 \times d + u = 8 \times d + 2 \times d + u$ (pt. 5.0).

Soluzione

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

entity bcd_to_bin is
port (a: in std_logic_vector(7 downto 0);
       b: out std_logic_vector(6 downto 0));
end entity bcd_to_bin;

architecture behav of bcd_to_bin is
begin
  process (a)
    variable: tmp,d,u: unsigned(6 downto 0);
  begin
    tmp:="0000000";
    d:=unsigned("000" & a(7 downto 4));
    u:=unsigned("000" & a(3 downto 0));

    d:=d(5 downto 0) & "0";
    tmp:=tmp+d;
    d:=d(4 downto 0) & "00";
    tmp:=tmp+d;
    tmp:=tmp+u;

    b <= std_logic_vector(tmp);
  end process;
end architecture;
```

Esercizio 3

Si consideri il seguente algoritmo:

```
u0:=a+b;
u1:=c*d;
u2:=e*f;      u7:=u5+m;
u3:=g+h;      u8:=u7*u6;
u4:=i+1;      u9:=u0+u8;
u5:=u1*u2;
u6:=u3*u4;
```

si tracci il DFG e si determini uno scheduling a latenza minima che minimizzi il numero di risorse sotto l'ipotesi di ciclo singolo. Si supponga poi di rimuovere l'ipotesi

di ciclo singolo in quanto il ritardo massimo dei moltiplicatori é superiore al periodo di clock che si intende utilizzare. In particolare si supponga che il periodo di clock sia pari a 5 ns e che il ritardo degli adder sia pari a 1.1 ns mentre quello dei moltiplicatori sia pari a 9.2 ns . Quale tecnica puó essere utilizzata per consentire al circuito di lavorare anche sotto queste condizioni? Si utilizzi lo scheduling calcolato nel ciclo di singolo per determinare uno scheduling consistente con queste ipotesi (pt. 5.0).

Soluzione

