

Compito di linguaggi di descrizione dell'hardware

Esercizio 1

Si realizzi un modello comportamentale in VHDL di una rete sequenziale asincrona con tre ingressi x , y ed e . Compito della rete é verificare che i fronti di salita su x e y si alternino quando $e = 1$. In tale caso, le uscite o, p assumono il valore 00. Se si ha una violazione di tale condizione, le uscite assumono il valore 01 se la transizione che manca é quella di x e 10 se, invece, manca la transizione di y . Tali valori di errore vengono mantenuti fino a quando e non si porta a 0.

Soluzione

Questa soluzione é piú complessa di quella richiesta in quanto usa un meccanismo per consentire alla rete di funzionare anche in presenza di transizione simultanea su x e y .

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

entity check is
  port (x,y,e: in std_logic;
        o,p: out std_logic);
end entity;

architecture behav of check is
begin

  process(x,y,e)
  variable last_transition: natural:=2; -- 0 x, 1 y, 2 both
  variable error: natural:=0; -- 0 no error, 1 error
  begin
    if (e='1') then
      if (rising_edge(x) and (error=0)) then
        if (last_transition/=0) then
          o <= '0';
          p <= '0';
          last_transition:=0;
        else
          o <= '1';
          p <= '0';
          error:=1;
        end if;
      end if;
      if (rising_edge(y) and (error=0)) then
        if (last_transition/=1) then
          o <= '0';
          p <= '0';
          last_transition:=1;
        else
          o <= '0';
          p <= '1';
          error:=1;
        end if;
      end if;
    end if;
  end process;
end architecture;
```

```

    elsif (e='0') then
        o <= '0';
        p <= '0';
        last_transition:=2;
        error:=0;
    end if;
end process;

end architecture;

```

Esercizio 2

Si realizzi un modello comportamentale in VHDL di una rete combinatoria che ha in ingresso una parola di n (passato come generic) bit $a_{n-1..0}$. Compito della rete é verificare se a é una palindroma (es. 101101101), in tale caso, l'uscita é 0 altrimenti é 1. L'esercizio deve essere risolto per un generico valore di n .

Soluzione

```

library IEEE;
use IEEE.std_logic_1164.all;

entity chk is
    generic(n: natural);
    port(a: in std_logic_vector(n-1 downto 0);
         o: out std_logic);
end entity;

architecture behav of chk is
begin
    process(a)
        variable otmp: std_logic;
    begin
        otmp:='0';
        for i in 0 to n/2 loop
            if (a(i)/=a(n-1-i)) then
                otmp:='1';
            end if;
        end loop;
        o <= otmp;
    end process;
end architecture;

```

Esercizio 3

Si consideri il seguente algoritmo:

```

u0:=a*b;          u6:=u1+u2;
u1:=c*d;          u7:=u3*u4;
u2:=e*f;          u8:=u3+u6;
u3:=a+e;          u9:=u7+u7;
u4:=b*c;          u10:=u5+u9;
u5:=u1+u0;

```

si tracci il DFG e si determini uno scheduling a risorse limitate (nell'ipotesi di ciclo

singolo) che minimizzi la latenza. Si supponga che il numero di risorse disponibili sia dato da due adder e un moltiplicatore. Una volta realizzato lo scheduling si minimizzi il numero di registri da utilizzare e si fornisca una descrizione al livello RTL delle operazioni.

Soluzione Metto solo di DFG con lo scheduling:

