

Compito di linguaggi di descrizione dell'hardware

Esercizio 1

Si realizzi un modello comportamentale in VHDL di una piccola memoria contenente 4 bit. La memoria sincrona, ha un ingresso dati D , due ingressi a_1a_0 che danno l'indirizzo del dato da scrivere e un comando di abilitazione WE . In uscita (segnale Q) si ha sempre il valore del dato indirizzato da a_1a_0 . Il dato D viene campionato sui fronti di salita del segnale di clock (ulteriore ingresso) in maniera simile a un FF di tipo D. L'esercizio é risolubile anche utilizzando il VHDL strutturale.

Soluzione

L'esercizio é simulabile (con un test bench minimale da espandere) al seguente indirizzo <https://www.edaplayground.com/x/2vA2>

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

entity memory is
  generic(tw,tr: time);
  port(d: in std_logic;
       a: in std_logic_vector(1 downto 0);
       we: in std_logic;
       clk: in std_logic;
       q: out std_logic);
end entity memory;

architecture behav of memory is
  signal s: std_logic_vector(0 to 3);
begin
  process(a,we,clk) -- update the state
  begin
    if ((rising_edge(clk)) and (we='1')) then
      case a is
        when "00" => s(0) <= d after tw;
        when "01" => s(1) <= d after tw;
        when "10" => s(2) <= d after tw;
        when "11" => s(3) <= d after tw;
        when others => s <= "XXXX";
      end case;
    end if;
  end process;
  process(a,s) -- output
  begin
```

```

    case a is
    when "00" => q <= s(0) after tr;
    when "01" => q <= s(1) after tr;
    when "10" => q <= s(2) after tr;
    when "11" => q <= s(3) after tr;
    when others => q <= 'X';
    end case;
end process;
end architecture;

```

Esercizio 2 Si realizzi la descrizione comportamentale di una rete combinatoria che riceve in ingresso due parole $a_{7..0}$ e $b_{3..0}$ che rappresentano due interi con segno A e B codificati in complemento a 2. Compito della rete é produrre in uscita $o_{8..0}$ il valore di $A + 2 * B$ codificato nello stesso modo degli ingressi. Le variazioni dell'uscita devono avvenire con un ritardo di 2 ns rispetto a quelle dell'ingresso.

Soluzione

Anche in questo caso l'esercizio é simulabile all'indirizzo <https://www.edaplayground.com/x/3ELs>.

```

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

entity prova is
    generic(t: time);
    port (a: in std_logic_vector(7 downto 0);
          b: in std_logic_vector(3 downto 0);
          o: out std_logic_vector(8 downto 0));
end entity prova;

architecture behav of prova is
    signal as, bs, os: signed(8 downto 0);
begin
    as <= signed(a(7) & a);
    bs <= signed(b(3) & b(3) & b(3) & b(3) & b & '0');
    os <= as+bs;
    o <= std_logic_vector(os) after t;
end architecture;

```

Esercizio 3

Si consideri il seguente algoritmo:

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

si tracci il DFG e si determini poi uno scheduling che fissate le risorse a 1 adder e 1 moltiplicatore, minimizzi la latenza. Si minimizzi il numero di registri utilizzati per tale scheduling e si descrivano sinteticamente al livello RTL le operazioni svolte da tale rete.