Sottrattore combinatorio e sequenziale

lunedì 3 gennaio 2022 11:02

COMBINATORIO

SOURCE CODE

```
library IEEE;
use IEEE.std_logic_1164.al;

entity Subtracter is
	generic(nb := integer := 16);
	port ( A : in std_logic_vector (nb-1 downto 0);
		B : in std_logic_vector (nb-1 downto 0);
		sub : out std_logic_vector (nb downto 0);
end Subtracter;

architecture Behav of Subtracter is
signal p, g : std_logic_vector (nb downto 0);
signal c : std_logic_vector (nb+1 downto 0);

begin
	-- A-B = A + not B + 1
	c(0) <= '1';
	p <= (A(nb-1) xor not B(nb-1)) & (A xor not B);
	g <= (A(nb-1) and not B(nb-1)) & (A and not B);
	c(nb+1 downto 1) <= g or (p and c(nb downto 0));
	sub <= p xor c(nb downto 0);
end behav;
```

TESTBENCH

```
entity SimSub is
	generic (nb := integer := 16);
end SimSub;

architecture Behav of SimSub is
component Subtracter is
	generic(nb := integer := 16);
	port ( A : in std_logic_vector (nb-1 downto 0);
		B : in std_logic_vector (nb-1 downto 0);
		sub : out std_logic_vector (nb downto 0);
end component;
signal IA, IB : std_logic_vector(nb-1 downto 0);
signal Osub : out std_logic_vector(nb downto 0);

begin
	CUT: subtracter port map(IA, IB, Osub);
	process
	begin
		for va in -(2**(nb-1)) to (2**(nb-1))-1 loop
			for vb in -(2**(nb-1)) to (2**(nb-1))-1 loop
				IA <= conv_std_logic_vector(va,nb);
				IB <= conv_std_logic_vector(vb,nb);
				wait for 10 ns;
			end loop;
		end loop;
	end process;
end Behav;
```

Questo circuito, se simulato, solleva un errore di temperatura di giunzione superata a causa dell'alta dissipazione di potenza per via della natura combinatoria.

SEQUENZIALE

SOURCE CODE

```
library IEEE;
use IEEE.std_logic_1164.al;

entity Subtracter is
	generic(nb := integer := 16);
	port ( A : in std_logic_vector (nb-1 downto 0);
		B : in std_logic_vector (nb-1 downto 0);
		clk : in std_logic;
		sub : out std_logic_vector (nb downto 0);
end Subtracter;
```

```
architecture Behav of Subtracter is
signal p, g : std_logic_vector (nb downto 0);
signal c : std_logic_vector (nb+1 downto 0);
signal IA, IB : in std_logic_vector (nb-1 downto 0);
signal Osub : out std_logic_vector (nb downto 0);

begin
      -- A-B = A + not B + 1
      c(0) <= '1';
      p <= ( IA(nb-1) xor not IB(nb-1)) & ( IA xor not IB );
      g <= ( IA(nb-1) and not IB(nb-1)) & ( IA and not IB );
      c(nb+1 downto 1) <= g or (p and c(nb downto 0));
      Osub <= p xor c(nb downto 0);
      process
            if rising_edge( clk ) then
                  IA <= A;
                  IB <= B;
                  sub <= Osub;
            end if;
      end process;
end behav;
```

segnali in uscita dai registri

→ 1 iniziale da sommare

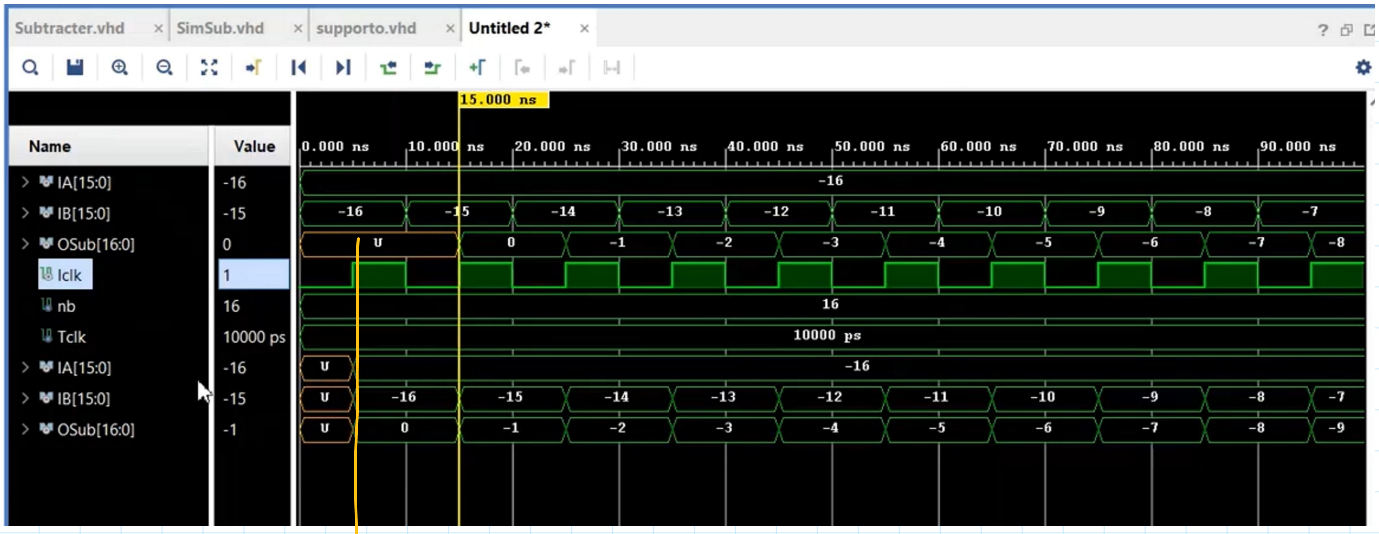
sincronizzazione delle uscite

TESTBENCH

```
entity SimSub is
      generic (nb := integer := 16);
end SimSub;

architecture Behav of SimSub is
component Subtracter is
      generic(nb := integer := 16);
      port ( A : in std_logic_vector (nb-1 downto 0);
             B : in std_logic_vector (nb-1 downto 0);
            clk : in std_logic;
             sub : out std_logic_vector (nb downto 0);
end component;
signal IA, IB : std_logic_vector(nb-1 downto 0);
signal Osub : out std_logic_vector(nb downto 0);
signal Iclk : std_logic := '0';
            -- constant Tclk : time:= 5 ns;  COMMENTATO

begin
      CUT: subtracter port map(IA, IB, Osub);

      process
      begin
            wait for Tclk+100 ns;
            for va in -(2**(nb-1)) to (2**(nb-1))-1 loop
                  for vb in -(2**(nb-1)) to (2**(nb-1))-1 loop
                        IA <= conv_std_logic_vector(va,nb);
                        IB <= conv_std_logic_vector(vb,nb);
                        wait for 10 ns;
                  end loop;
            end loop;
      end process;

      process
      begin
            wait for 5 ns;
            -- potrei scriverlo anche come costante constant Tclk : time:= 5 ns;
            Iclk <= not Iclk;
end Behav;
```

→ sincronizzo col ritardo del clock che per 100 ns fa un reset

→ periodo del clock



disallineamento / latenza del circuito

ottengo il primo risultato in uscita con del ritardo

Route report timing summary

```
------------------------------------------------------------------------------------------------
| Design Timing Summary
| ---------------------
------------------------------------------------------------------------------------------------

    WNS(ns)      TNS(ns)  TNS Failing Endpoints  TNS Total Endpoints      WHS(ns)      THS(ns)  THS Failing Endpoints  THS
    -------      -------  ---------------------  -------------------      -------      -------  ---------------------  ---
      0.341        0.000                      0                   17        0.198        0.000                      0

All user specified timing constraints are met.
```

essendo >0 vol dire che posso lavorare con un periodo maggiore di clock