

Elettronica Digitale

1° Progetto: Carry Select a 8 bit.

Studente: Tassone Roberto

Matricola: 149351

Descrizione del circuito e dei componenti

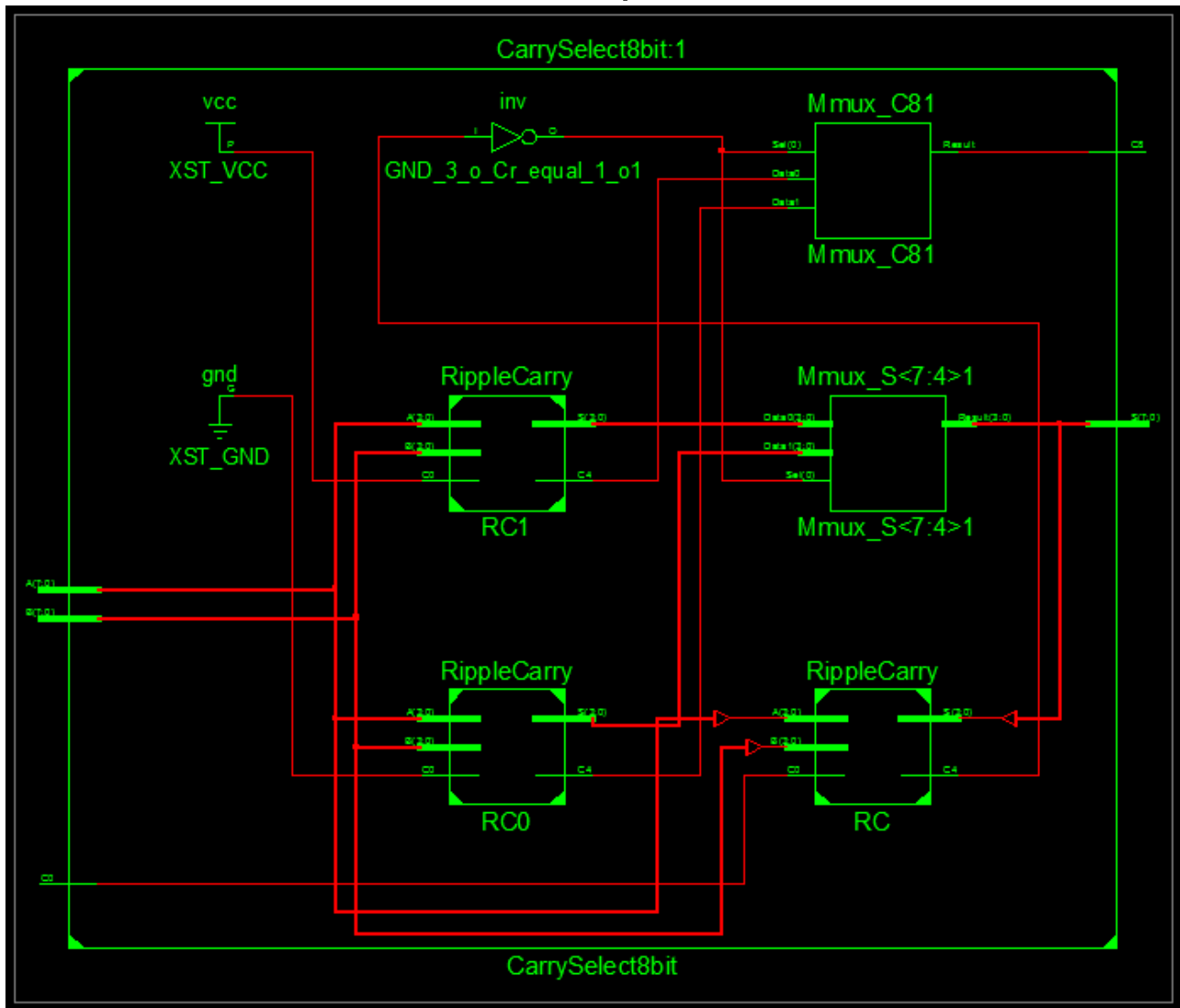
Si vuole progettare, utilizzando il linguaggio VHDL, un Carry Select a otto bit. Questo ha tre porte in ingresso e due in uscita. Nelle prima due porte di ingresso gli vengono passati, attraverso dei bus a otto bit, i due addendi da sommare. Nella terza porta invece gli viene passato un eventuale riporto nel caso in cui questo componente sia utilizzato per un altro circuito. Per quanto riguarda le uscite abbiamo un bus a otto bit dal quale si ottiene la somma ed una porta che coincide con il riporto di questa somma. Utilizzare il Carry Select è conveniente in quanto permette di scomporre la somma in più parti e di conseguenza si riduce il tempo di calcolo.

Si è scelto di comporre il Carry Select utilizzando dei Ripple Carry a quattro bit. Questa scelta è dovuta al fatto che riducendo la dimensione dei Ripple Carry (ad esempio a due bit) aumenta il numero dei componenti da utilizzare e di conseguenza cresce il costo e la grandezza del circuito, nonostante si riduca il tempo di calcolo. Le porte del Ripple Carry sono identiche a quelle del Carry Select ma differiscono solo per la dimensione dei bus dati che scende a quattro bit dato che deve gestire la somma di due numeri a quattro bit. Si utilizzano esattamente tre Ripple Carry: il primo somma i quattro bit meno significativi dei due numeri a otto bit, gli altri due sommano i restanti quattro bit nelle ipotesi che il riporto generato dal primo Ripple Carry sia zero oppure uno. Tramite un MUX si selezionerà il Ripple Carry adeguato usando come segnale di controllo proprio il riporto del primo Ripple Carry.

Ognuno dei Ripple Carry è formato da quattro Full Adder collegati in sequenza. Il Full Adder somma due singoli bit più il riporto scaturito dalla somma precedente e restituisce due bit, uno per la somma ed un altro per il riporto. Si è scelto di utilizzare il Full Adder che sfrutta un multiplexer per eseguire la somma.

Struttura e codice VHDL dei componenti

Schema di CarrySelect8bit



VHDL di CarrySelect8bit

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity CarrySelect8bit is
    Port ( A, B: in  STD_LOGIC_VECTOR (7 downto 0);
          C0: in  STD_LOGIC;
          S: out STD_LOGIC_VECTOR (7 downto 0);
          C8: out STD_LOGIC);
end CarrySelect8bit;
```

architecture Structural of CarrySelect8bit is

```
component RippleCarry is
    Port ( C0: in  STD_LOGIC;
          A, B: in  STD_LOGIC_VECTOR (3 downto 0);
```

```

    S : out STD_LOGIC_VECTOR(3 downto 0);
    C4 : out STD_LOGIC);
end component;

signal Cr, Cr0, Cr1 : STD_LOGIC;
signal S0, S1 : STD_LOGIC_VECTOR (3 downto 0);

begin
RC:RippleCarry port map(C0, A(3 downto 0), B(3 downto 0), S(3 downto 0), Cr);
RC0:RippleCarry port map('0', A(7 downto 4), B(7 downto 4), S0(3 downto 0), Cr0);
RC1:RippleCarry port map('1', A(7 downto 4), B(7 downto 4), S1(3 downto 0), Cr1);
with Cr select
    S(7 downto 4) <= S0 (3 downto 0)when '0',
                    S1 (3 downto 0)when '1',
                    "XXXX" when others;

with Cr select
    C8 <= Cr0 when '0',
        Cr1 when '1',
        'X' when others;
end Structural;

```

VHDL di RippleCarry

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity RippleCarry is
    Port ( C0 : in  STD_LOGIC;
          A, B : in  STD_LOGIC_VECTOR (3 downto 0);
          S : out STD_LOGIC_VECTOR(3 downto 0);
          C4 : out STD_LOGIC);
end RippleCarry;

architecture Structural of RippleCarry is

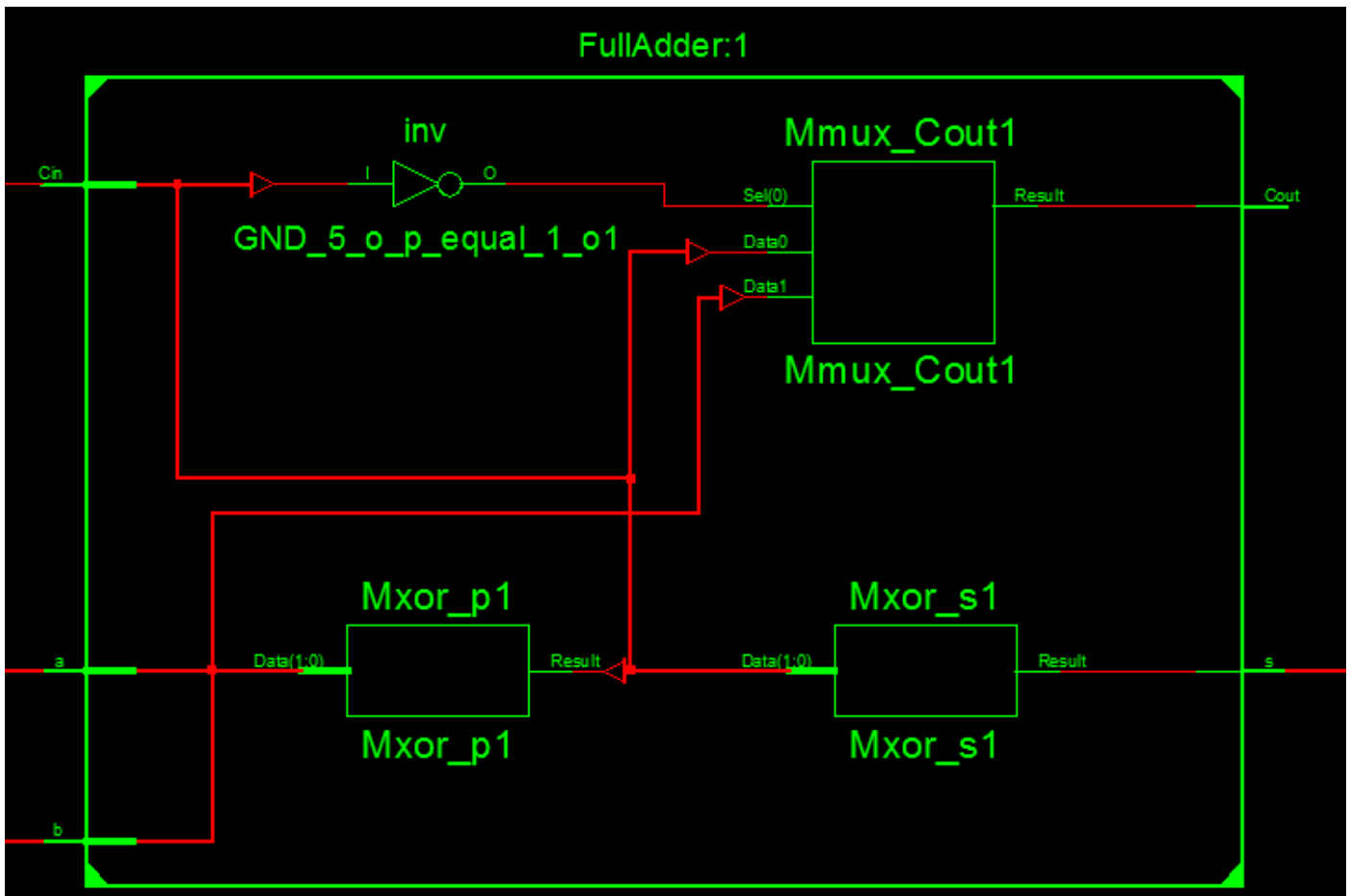
    component FullAdder
        Port ( a, b, Cin : in  STD_LOGIC;
              s, Cout: out STD_LOGIC);
    end component;

    signal C1, C2, C3 : STD_LOGIC;

begin
    FA1:FullAdder port map(A(0), B(0), C0, S(0), C1);
    FA2:FullAdder port map(A(1), B(1), C1, S(1), C2);
    FA3:FullAdder port map(A(2), B(2), C2, S(2), C3);
    FA4:FullAdder port map(A(3), B(3), C3, S(3), C4);
end Structural;

```

Schema di FullAdder



VHDL di FullAdder

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity FullAdder is
    Port ( a, b, Cin : in  STD_LOGIC;
           s, Cout :out STD_LOGIC);
end FullAdder;

architecture Behavioral of FullAdder is

    signal p : STD_LOGIC;

begin
    p <= a xor b;
    s <= p xor Cin;
    with p select
    Cout <= a when '0',
           Cin when '1',
           'X' when others;
end Behavioral;
    
```

Test bench

Per verificare il corretto funzionamento del circuito si utilizza ulteriormente il codice VHDL. Si simula quindi di avere degli ingressi, scelti arbitrariamente, e si controlla il valore in uscita.

VHDL del test bench

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
ENTITY SimulationCarrySelect8bit IS
END SimulationCarrySelect8bit;

ARCHITECTURE behavior OF SimulationCarrySelect8bit IS

    COMPONENT CarrySelect8bit
    PORT( A, B : IN  std_logic_vector(7 downto 0);
          C0 : IN  std_logic;
          S : OUT std_logic_vector(7 downto 0);
          C8 : OUT std_logic
        );
    END COMPONENT;

    signal A, B, S : std_logic_vector(7 downto 0);
    signal C0 : std_logic;
    signal C8 : std_logic;

BEGIN
    uut: CarrySelect8bit PORT MAP ( A, B, C0, S, C8 );
    process
    begin
        A <= "00000000";
        B <= "00000000";
        C0 <= '1';
        wait for 10 ns;
        A <= "00000000";
        B <= "00000001";
        C0 <= '1';
        wait for 10 ns;
        A <= "00000000";
        B <= "00011111";
        C0 <= '0';
        wait for 10 ns;
        A <= "00000000";
        B <= "11111111";
        C0 <= '0';
        wait for 10 ns;
        A <= "00000001";
        B <= "00000001";
        C0 <= '0';
```

```
end process;  
END;
```

```
wait for 10 ns;  
A <= "00001111";  
B <= "00111000";  
C0 <= '1';  
wait for 10 ns;  
A <= "00001111";  
B <= "00111111";  
C0 <= '1';  
wait for 10 ns;  
A <= "00111111";  
B <= "11110100";  
C0 <= '0';  
wait for 10 ns;  
A <= "11111111";  
B <= "11111111";  
C0 <= '0';  
wait for 10 ns;  
A <= "11111111";  
B <= "11111111";  
C0 <= '1';  
wait for 10 ns;
```

Schema dei segnali di input e output

