2η εργαστηριακή άσκηση

Ιωάννης Μπουφίδης 03120162
Φωτεινή Κυριακοπούλου 03120182

Half Adder - HA

Υλοποιούμε το half adder σε περιγραφή ροής δεδομένων ως εξής:

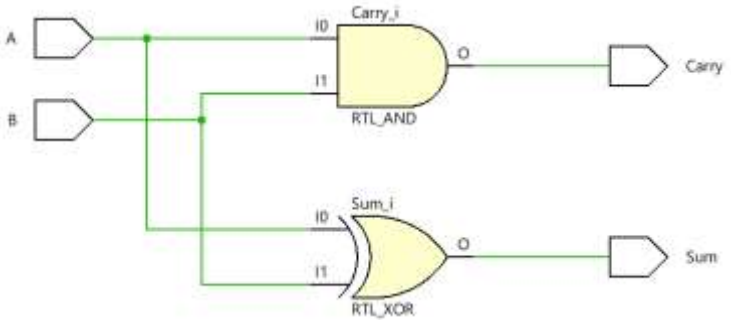
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity half_adder is
   Port (
      A, B: in std_logic;
      Sum, Carry: out std_logic
   );
end half_adder;

architecture half_adder_arch of half_adder is
begin
     Sum <= A xor B;
     Carry <= A and B;
end half_adder_arch;
```

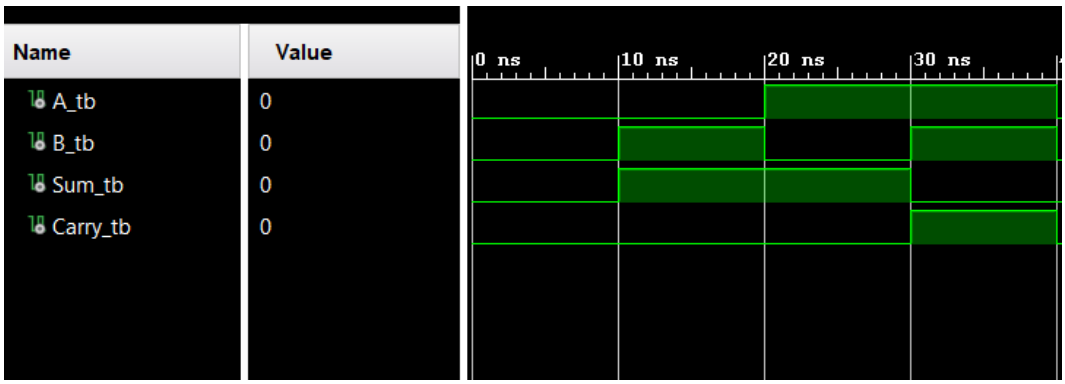
Προκειμένου να προσομοιώσουμε την αρχιτεκτονική, γράφουμε το παρακάτω testbench:

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


entity half_adder_tb is
end half_adder_tb;

architecture half_adder_test of half_adder_tb is
   component half_adder
     Port (
        A, B: in std_logic;
        Sum, Carry: out std_logic
      );
   end component;
   signal A_tb, B_tb, Sum_tb, Carry_tb: std_logic;
begin
  uut: half_adder port map (
    A => A_tb,
    B => B_tb,
    Sum => Sum_tb,
    Carry => Carry_tb
  );
```

```
  check: process
  begin
   A_tb <= '0'; B_tb <= '0';
   wait for 10 ns;

   A_tb <= '0'; B_tb <= '1';
   wait for 10 ns;

   A_tb <= '1'; B_tb <= '0';
   wait for 10 ns;

   A_tb <= '1'; B_tb <= '1';
   wait for 10 ns;
  end process check;
end half_adder_test;
```

Παρακάτω φαίνεται το αποτέλεσμα της προσομοίωσης (κάτω) και το σχηματικό RTL (πάνω):





Το κρίσιμο μονοπάτι του κυκλώματος είναι **From B To Sum** με χρονική καθυστέρηση **5.377**.

2η εργαστηριακή άσκηση

Full Adder - FA

Υλοποιούμε το full adder με περιγραφή δομής ως εξής:

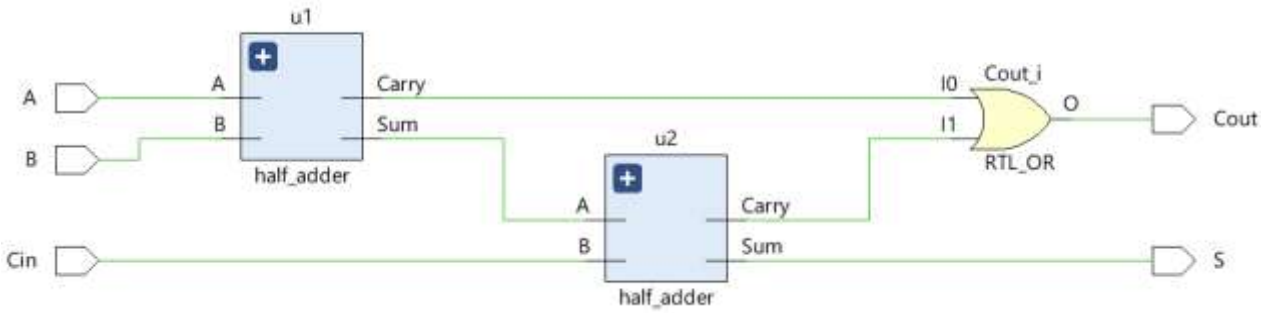
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity full_adder is
    Port (
       A, B, Cin: in std_logic;
       S, Cout: out std_logic
     );
end full_adder;

architecture full_adder_arch of full_adder is
  signal u1_out_carry, u1_out_sum, u2_out_carry: std_logic;
  component half_adder is
    port (
       A, B: in std_logic;
       Sum, Carry: out std_logic
    );
  end component;
begin
  u1: half_adder port map(A=>A, B=>B, Sum=>u1_out_sum, Carry=>u1_out_carry);
  u2: half_adder port map(A=>u1_out_sum, B=>Cin, Sum=>S, Carry=>u2_out_carry);
  Cout <= u1_out_carry or u2_out_carry;
end full_adder_arch;
```

Προκειμένου να προσομοιώσουμε την αρχιτεκτονική, γράφουμε το παρακάτω testbench:

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

entity full_adder_tb is
end full_adder_tb;

architecture full_adder_test of full_adder_tb is
   component full_adder
     Port (
        A, B, Cin: in std_logic;
        S, Cout: out std_logic
      );
   end component;
   signal test_input: std_logic_vector(2 downto 0);
   signal Cout_tb, S_tb: std_logic;
```

```
begin
  uut: full_adder port map (
    A => test_input(2),
    B => test_input(1),
    S => S_tb,
    Cin => test_input(0),
    Cout => Cout_tb
  );

  check: process
  begin
    test_input <= "000"; wait for 10ns;
    test_input <= "001"; wait for 10ns;
    test_input <= "010"; wait for 10ns;
    test_input <= "011"; wait for 10ns;
    test_input <= "100"; wait for 10ns;
    test_input <= "101"; wait for 10ns;
    test_input <= "110"; wait for 10ns;
    test_input <= "111"; wait for 10ns;
  end process check;
end full_adder_test;
```

Παρακάτω φαίνεται το αποτέλεσμα της προσομοίωσης (κάτω) και το σχηματικό RTL (πάνω):



Το κρίσιμο μονοπάτι του κυκλώματος είναι **From A To S** με χρονική καθυστέρηση **5.377**.

2η εργαστηριακή άσκηση

4-bit Parallel Adder – 4-bit PA

Υλοποιούμε το 4-bit parallel adder με περιγραφή δομής ως εξής:

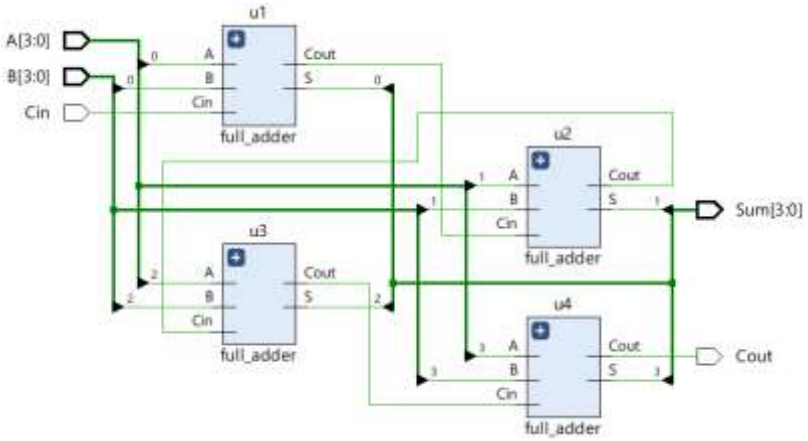
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity parallel_adder is
   port (
      A, B: in std_logic_vector(3 downto 0);
      Cin: in std_logic;
      Sum: out std_logic_vector(3 downto 0);
      Cout: out std_logic
   );
end parallel_adder;

architecture parallel_adder_arch of parallel_adder is
  signal u1_out_carry, u2_out_carry, u3_out_carry: std_logic;
  component full_adder is
    Port (
       A, B, Cin: in std_logic;
       S, Cout: out std_logic
     );
  end component;
begin
  u1: full_adder port map(A=>A(0), B=>B(0), Cin=>Cin, S=>Sum(0), Cout=>u1_out_carry);
  u2: full_adder port map(A=>A(1), B=>B(1), Cin=>u1_out_carry, S=>Sum(1), Cout=>u2_out_carry);
  u3: full_adder port map(A=>A(2), B=>B(2), Cin=>u2_out_carry, S=>Sum(2), Cout=>u3_out_carry);
  u4: full_adder port map(A=>A(3), B=>B(3), Cin=>u3_out_carry, S=>Sum(3), Cout=>Cout);
end parallel_adder_arch;
```

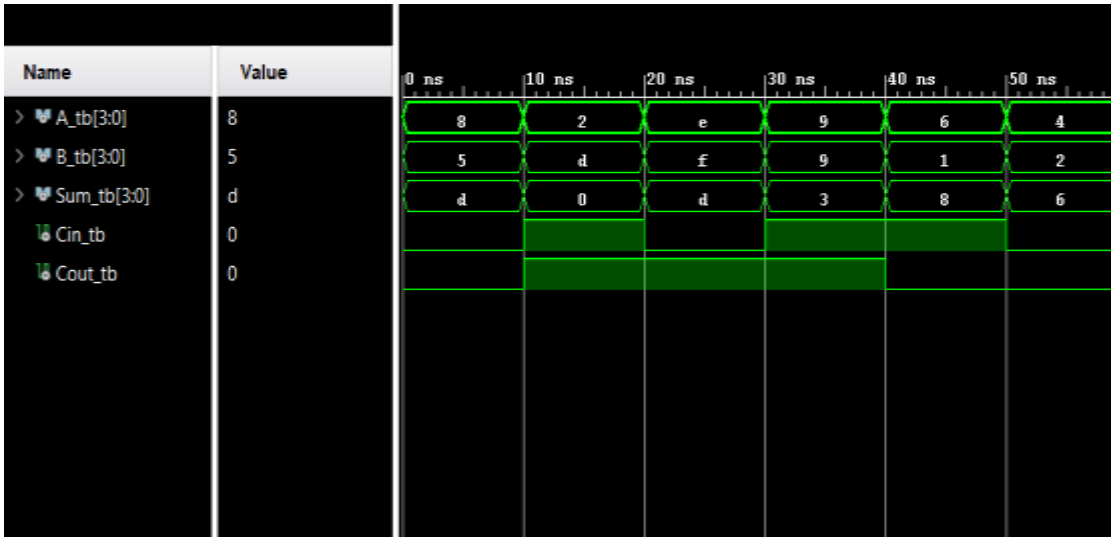
Προκειμένου να προσομοιώσουμε την αρχιτεκτονική, γράφουμε το παρακάτω testbench:

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

entity parallel_adder_tb is
end parallel_adder_tb;

architecture parallel_adder_test of parallel_adder_tb is
   component parallel_adder
     Port (
        A, B: in std_logic_vector(3 downto 0);
        Cin: in std_logic;
        Sum: out std_logic_vector(3 downto 0);
        Cout: out std_logic
      );
   end component;
   signal A_tb, B_tb, Sum_tb: std_logic_vector(3 downto 0);
   signal Cin_tb, Cout_tb: std_logic;
```

```
begin
  uut: parallel_adder port map (
    A => A_tb,
    B => B_tb,
    Sum => Sum_tb,
    Cin => Cin_tb,
    Cout => Cout_tb
  );

  check: process
  begin
    Cin_tb <= '0'; A_tb <= "1000"; B_tb <= "0101"; wait for 10ns;
    Cin_tb <= '1'; A_tb <= "0010"; B_tb <= "1101"; wait for 10ns;
    Cin_tb <= '0'; A_tb <= "1110"; B_tb <= "1111"; wait for 10ns;
    Cin_tb <= '1'; A_tb <= "1001"; B_tb <= "1001"; wait for 10ns;
    Cin_tb <= '1'; A_tb <= "0110"; B_tb <= "0001"; wait for 10ns;
    Cin_tb <= '0'; A_tb <= "0100"; B_tb <= "0010"; wait for 10ns;
  end process check;
end parallel_adder_test;
```

Παρακάτω φαίνεται το αποτέλεσμα της προσομοίωσης (κάτω) και το σχηματικό RTL (πάνω):





Το κρίσιμο μονοπάτι του κυκλώματος είναι **From B[0] To Cout** με χρονική καθυστέρηση **5.970**.

2η εργαστηριακή άσκηση

BCD Full Adder – BCD FA

Υλοποιούμε το BCD full adder με περιγραφή δομής ως εξής:

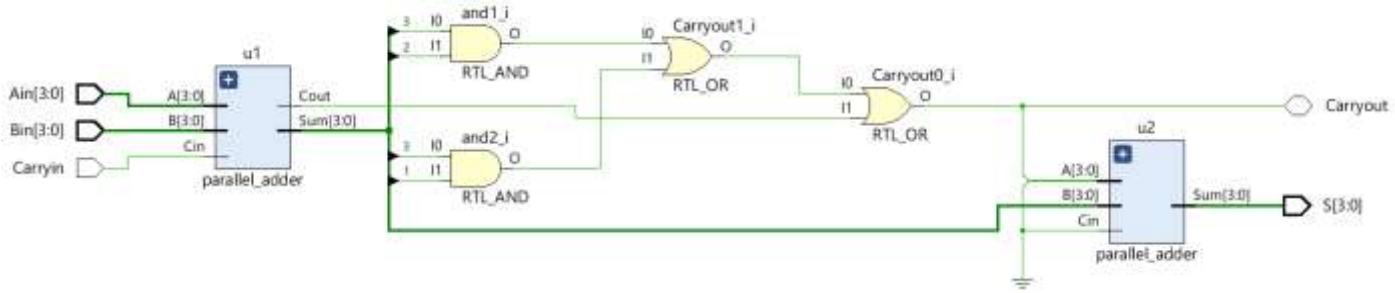
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity bcd_full_adder is
   Port (
     Ain, Bin: in std_logic_vector(3 downto 0);
     Carryin: in std_logic;
     S: out std_logic_vector(3 downto 0);
     Carryout: inout std_logic
   );
end bcd_full_adder;

architecture bcd_full_adder_arch of bcd_full_adder is
  component parallel_adder is
    port (
        A, B: in std_logic_vector(3 downto 0);
        Cin: in std_logic;
        Sum: out std_logic_vector(3 downto 0);
        Cout: out std_logic
    );
  end component;
  signal u1_out_sum, u2_in: std_logic_vector(3 downto 0);
  signal u1_out_cout, u2_out_cout, and1, and2: std_logic;
begin
  u1: parallel_adder port map (A=>Ain, B=>Bin, Cin=>Carryin, Sum=>u1_out_sum, Cout=>u1_out_cout);
  and1 <= u1_out_sum(3) and u1_out_sum(2);
  and2 <= u1_out_sum(3) and u1_out_sum(1);
  Carryout <= and1 or and2 or u1_out_cout;
  u2_in <= '0'&Carryout&Carryout&'0';
  u2: parallel_adder port map (A=>u2_in, B=>u1_out_sum, Cin=>'0', Sum=>S, Cout=>u2_out_cout);

end bcd_full_adder_arch;
```

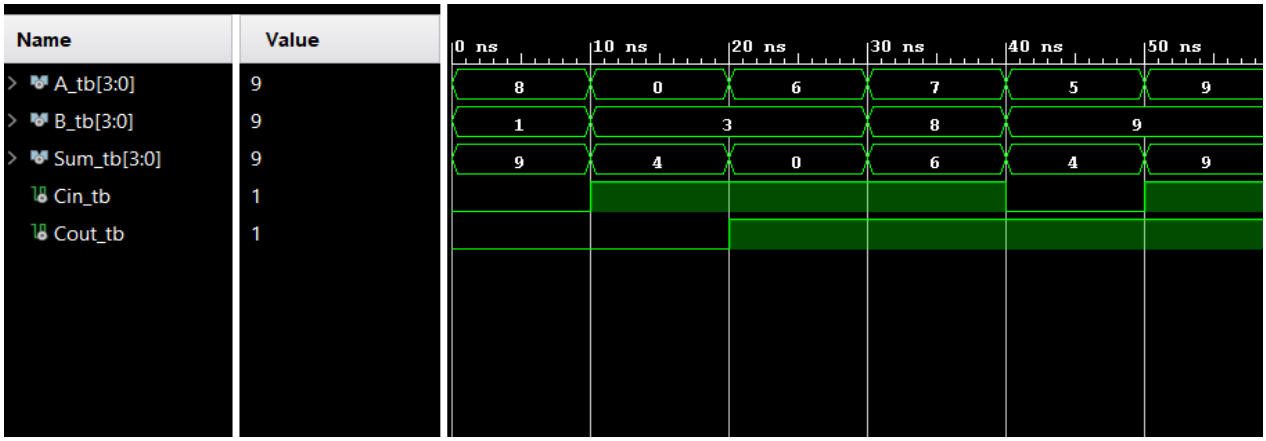
Προκειμένου να προσομοιώσουμε την αρχιτεκτονική, γράφουμε το παρακάτω testbench:

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

entity bcd_full_adder_tb is
end bcd_full_adder_tb;

architecture bcd_full_adder_test of bcd_full_adder_tb is
   component bcd_full_adder
     Port (
        Ain, Bin: in std_logic_vector(3 downto 0);
        Carryin: in std_logic;
        S: out std_logic_vector(3 downto 0);
```

```
        Carryout: inout std_logic
      );
   end component;
   signal A_tb, B_tb, Sum_tb: std_logic_vector(3 downto 0);
   signal Cin_tb, Cout_tb: std_logic;
begin
  uut: bcd_full_adder port map (
    Ain => A_tb,
    Bin => B_tb,
    S => Sum_tb,
    Carryin => Cin_tb,
    Carryout => Cout_tb
  );

  check: process
  begin
    Cin_tb <= '0'; A_tb <= "1000"; B_tb <= "0001"; wait for 10ns;
    Cin_tb <= '1'; A_tb <= "0000"; B_tb <= "0011"; wait for 10ns;
    Cin_tb <= '1'; A_tb <= "0110"; B_tb <= "0011"; wait for 10ns;
    Cin_tb <= '1'; A_tb <= "0111"; B_tb <= "1000"; wait for 10ns;
    Cin_tb <= '0'; A_tb <= "0101"; B_tb <= "1001"; wait for 10ns;
    Cin_tb <= '1'; A_tb <= "1001"; B_tb <= "1001"; wait for 10ns;
  end process check;
end bcd_full_adder_test;
```

Παρακάτω φαίνεται το σχηματικό RTL, ενώ πιο κάτω το αποτέλεσμα της προσομοίωσης:





Το κρίσιμο μονοπάτι του κυκλώματος είναι **From Bin[0] To S[3]** με χρονική καθυστέρηση **7.717**.

2η εργαστηριακή άσκηση

4-BCD Parallel Adder – 4-BCD PA

Υλοποιούμε το 4-BCD parallel adder με περιγραφή δομής ως εξής:

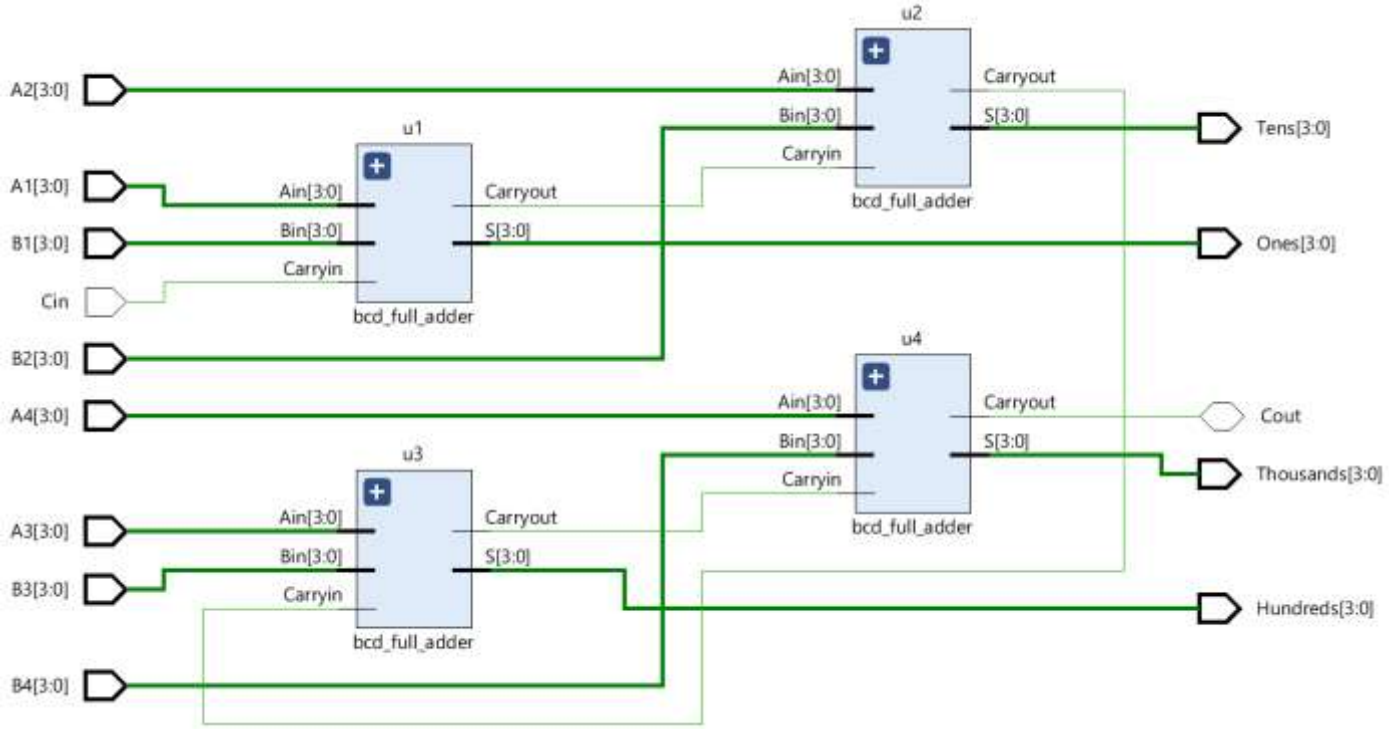
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;


entity bcd_parallel_adder is
   Port (
      A1, A2, A3, A4: in std_logic_vector(3 downto 0);
      B1, B2, B3, B4: in std_logic_vector(3 downto 0);
      Cin: in std_logic;
      Cout: inout std_logic;
      Ones, Tens, Hundreds, Thousands: out std_logic_vector(3 downto 0)
   );
end bcd_parallel_adder;

architecture bcd_parallel_adder_arch of bcd_parallel_adder is
  component bcd_full_adder is
     Port (
       Ain, Bin: in std_logic_vector(3 downto 0);
       Carryin: in std_logic;
       S: out std_logic_vector(3 downto 0);
       Carryout: inout std_logic
     );
  end component;
  signal c1_out, c2_out, c3_out: std_logic;
begin
  u1: bcd_full_adder port map (Ain=>A1, Bin=>B1, Carryin=>Cin, S=>Ones, Carryout=>c1_out);
  u2: bcd_full_adder port map (Ain=>A2, Bin=>B2, Carryin=>c1_out, S=>Tens, Carryout=>c2_out);
  u3: bcd_full_adder port map (Ain=>A3, Bin=>B3, Carryin=>c2_out, S=>Hundreds, Carryout=>c3_out);
  u4: bcd_full_adder port map (Ain=>A4, Bin=>B4, Carryin=>c3_out, S=>Thousands, Carryout=>Cout);
end bcd_parallel_adder_arch;
```

Προκειμένου να προσομοιώσουμε την αρχιτεκτονική, γράφουμε το παρακάτω testbench:

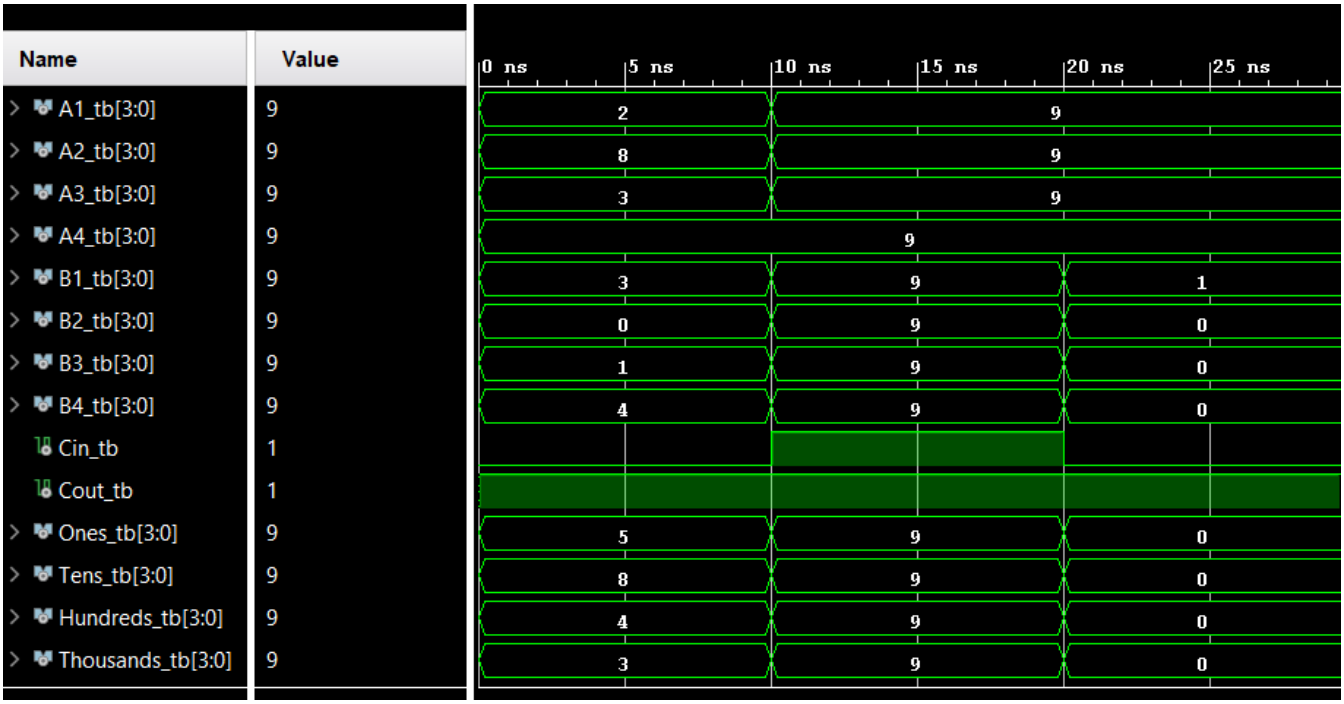
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity bcd_parallel_adder_tb is
end bcd_parallel_adder_tb;

architecture bcd_parallel_adder_test of bcd_parallel_adder_tb is
   component bcd_parallel_adder
    Port (
      A1, A2, A3, A4: in std_logic_vector(3 downto 0);
      B1, B2, B3, B4: in std_logic_vector(3 downto 0);
      Cin: in std_logic;
      Cout: inout std_logic;
```

```
      Ones, Tens, Hundreds, Thousands: out std_logic_vector(3 downto 0)
     );
   end component;
  signal A1_tb, A2_tb, A3_tb, A4_tb: std_logic_vector(3 downto 0);
  signal B1_tb, B2_tb, B3_tb, B4_tb: std_logic_vector(3 downto 0);
  signal Cin_tb, Cout_tb: std_logic;
  signal Ones_tb, Tens_tb, Hundreds_tb, Thousands_tb: std_logic_vector(3 downto 0);

begin
  uut: bcd_parallel_adder port map (
    A1 => A1_tb, A2 => A2_tb, A3 => A3_tb, A4 => A4_tb,
    B1 => B1_tb, B2 => B2_tb, B3 => B3_tb, B4 => B4_tb,
    Cin => Cin_tb, Cout => Cout_tb,
    Ones => Ones_tb, Tens => Tens_tb, Hundreds => Hundreds_tb, Thousands => Thousands_tb
  );

  check: process
  begin
   A1_tb <= "0010"; A2_tb <= "1000"; A3_tb <= "0011"; A4_tb <= "1001"; Cin_tb <= '0';
   B1_tb <= "0011"; B2_tb <= "0000"; B3_tb <= "0001"; B4_tb <= "0100"; wait for 10 ns;

   A1_tb <= "1001"; A2_tb <= "1001"; A3_tb <= "1001"; A4_tb <= "1001"; Cin_tb <= '1';
   B1_tb <= "1001"; B2_tb <= "1001"; B3_tb <= "1001"; B4_tb <= "1001"; wait for 10 ns;

   A1_tb <= "1001"; A2_tb <= "1001"; A3_tb <= "1001"; A4_tb <= "1001"; Cin_tb <= '0';
   B1_tb <= "0001"; B2_tb <= "0000"; B3_tb <= "0000"; B4_tb <= "0000"; wait for 10 ns;
  end process check;
end bcd_parallel_adder_test;
```

Παρακάτω φαίνεται το σχηματικό RTL, ενώ πιο κάτω το αποτέλεσμα της προσομοίωσης:



2η εργαστηριακή άσκηση



Το κρίσιμο μονοπάτι του κυκλώματος είναι **From B1[1] To Thousands[2]** με χρονική καθυστέρηση **11.262**.