Εθνικό Μετσόβιο Πολυτεχνείο

Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Ψηφιακά Συστήματα **VLSI**

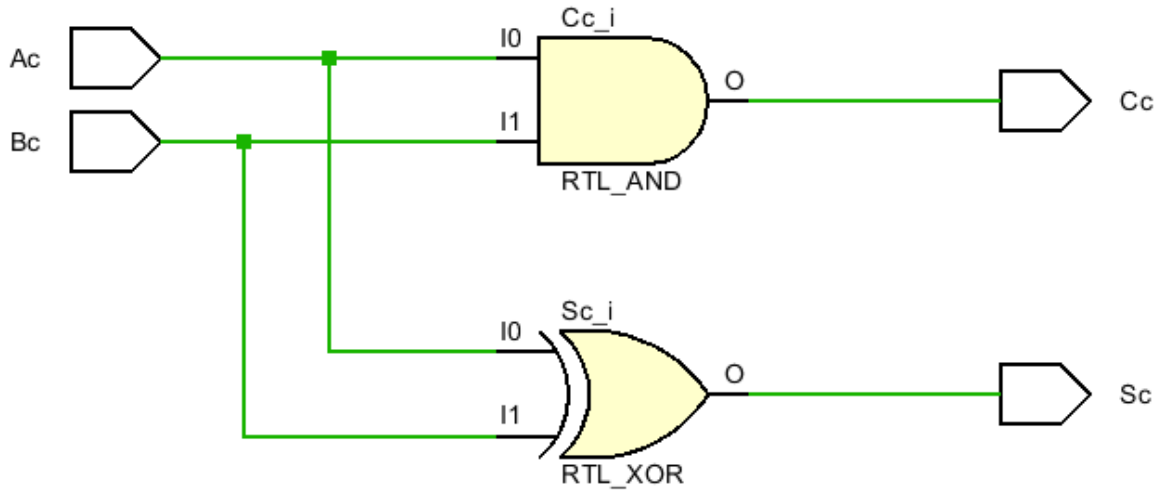
2η Εργαστηριακή Άσκηση

Καραμπίνας Παναγιώτης (03116170)
Αλικάρης Αντώνιος Δημήτριος (03118062)

1. Ημιαθροιστής

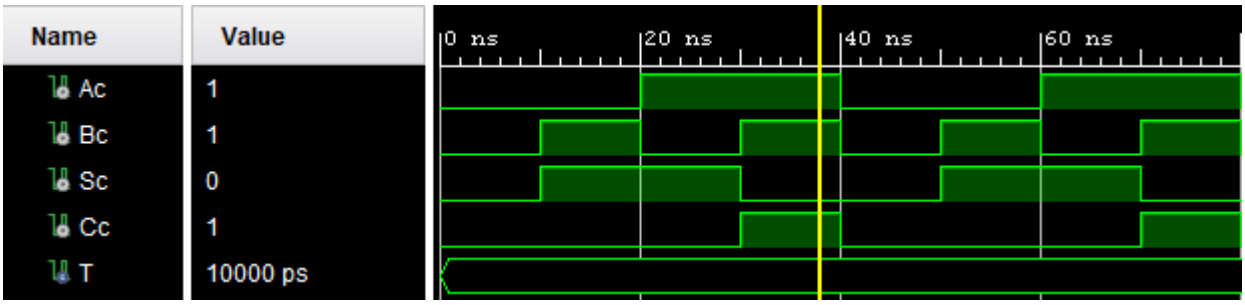
Κώδικας και δομικό διάγραμμα (RTL)

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

entity half_adder is        --Here we declare our entity -to be designed-
    Port ( Ac : in STD_LOGIC;        --declaration of variables as inputs or
    ↪  outputs
           Bc : in STD_LOGIC;
           Sc : out STD_LOGIC;
           Cc : out STD_LOGIC);
end half_adder;

architecture Data_flow of half_adder is

begin
    Sc <= Ac XOR Bc;        --by using the logic diagram of Half Adder as our
    ↪  starting point we initiate our design
    Cc <= Ac AND Bc;

end Data_flow;
```

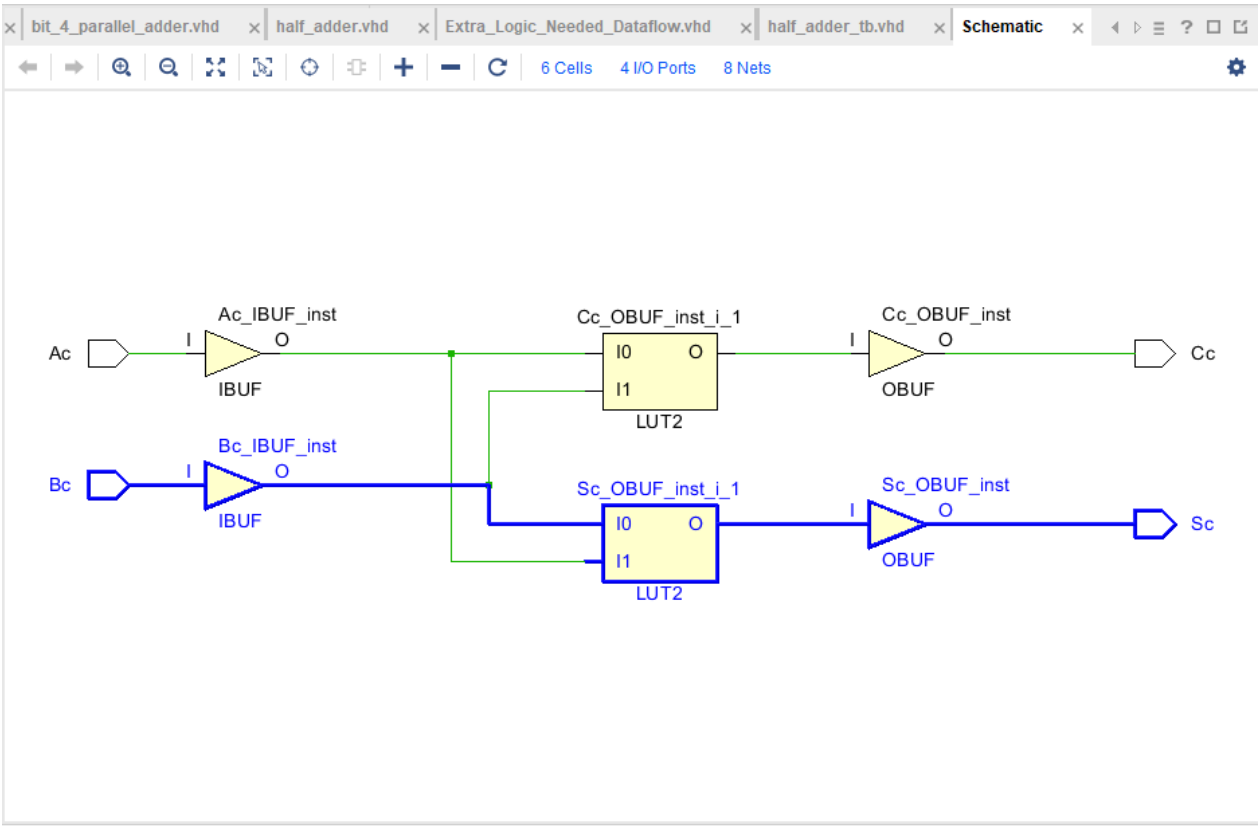


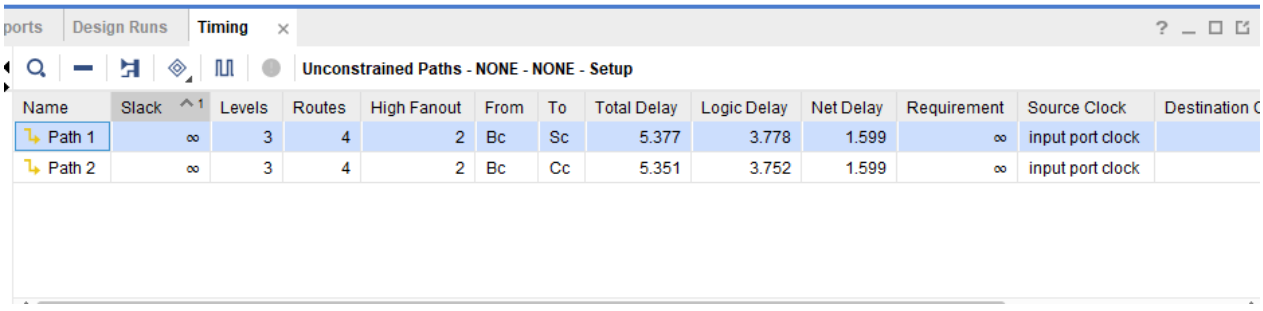
Κώδικας testbench και Κυματομορφή εξόδου

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
USE IEEE.numeric_std.ALL;
USE IEEE.std_logic_unsigned.ALL;

entity half_adder_tb is
--  Port ( );
end half_adder_tb;

architecture Behavioral of half_adder_tb is

COMPONENT half_adder IS
    Port ( Ac : in STD_LOGIC;
           Bc : in STD_LOGIC;
           Sc : out STD_LOGIC;
           Cc : out STD_LOGIC);
end COMPONENT;

SIGNAL Ac, Bc, Sc, Cc : STD_LOGIC;

CONSTANT T : TIME := 10 ns;

begin

 uut: half_adder PORT MAP (Ac => Ac,
                           Bc => Bc,
                           Sc => Sc,
                           Cc => Cc);

 stimuli: PROCESS
 begin
    Ac <= '0';
    Bc <= '0';
    WAIT FOR T;
    Ac <= '0';
    Bc <= '1';
    WAIT FOR T;
    Ac <= '1';
    Bc <= '0';
    WAIT FOR T;
    Ac <= '1';
    Bc <= '1';
    WAIT FOR T;
 end PROCESS;
end Behavioral;
```



Critical Path και Συνολική Καθυστέρηση



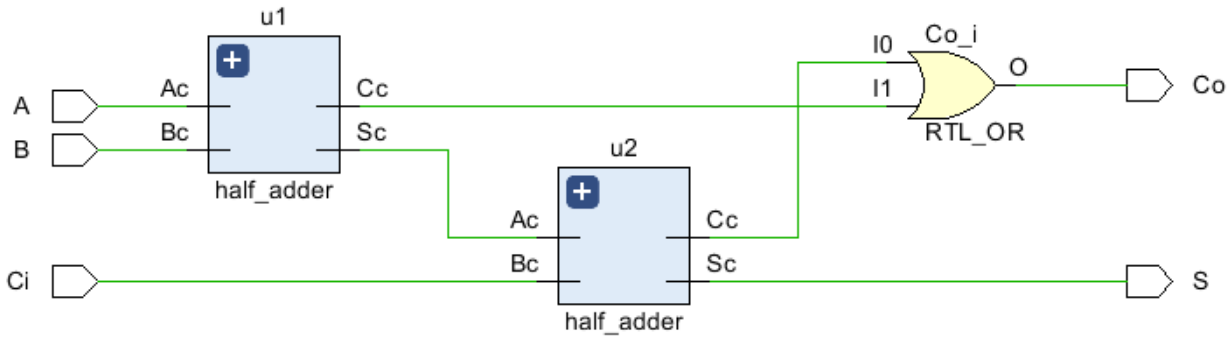


Όπως φαίνεται από την παραπάνω εικόνα, το critical path είναι από το Bc στο Sc και η συνολική καθυστέρηση είναι 5.377ns.

2. Πλήρης Αθροιστής

Κώδικας και δομικό διάγραμμα (RTL)

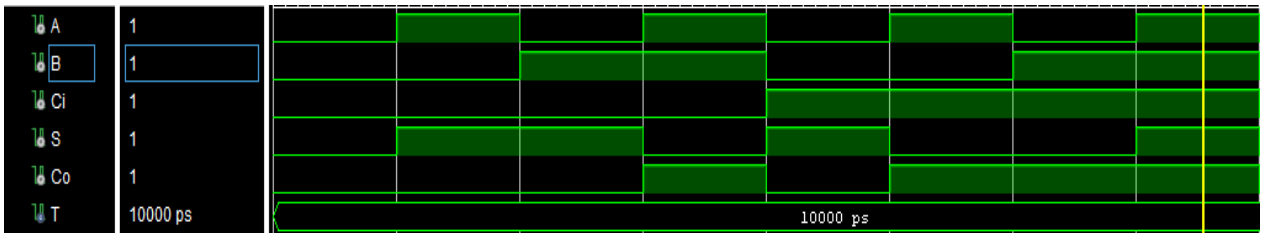
```vhdl
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity full_adder is
    Port ( A : in STD_LOGIC;
           B : in STD_LOGIC;
           Ci : in STD_LOGIC;
           S : out STD_LOGIC;
           Co : out STD_LOGIC);
end full_adder;

architecture Structural of full_adder is

SIGNAL w1 : STD_LOGIC ;     --We need 3 signals, because by looking at the logic diagram of the Full Adder the are three intermediate paths
SIGNAL w2 : STD_LOGIC ;     --that can't be described as inputs or outputs.
SIGNAL w3 : STD_LOGIC ;

COMPONENT half_adder
    Port ( Ac : in STD_LOGIC;
           Bc : in STD_LOGIC;
           Sc : out STD_LOGIC;
           Cc : out STD_LOGIC);
end COMPONENT;

begin
    --Below we see the structural use of 2 Half Adders, followed by a the use of a (simple) OR gate.
    --The use of OR as a component was deemed excessive so the final code may be described as Mixed(and not really structural)
    u1: half_adder PORT MAP (Ac => A,
                             Bc => B,
                             Sc => w1,
                             Cc => w2);

    u2: half_adder PORT MAP (Ac => w1,
                             Bc => Ci,
                             Sc => S,
                             Cc => w3);

    Co <= w3 OR w2;

end Structural;
```



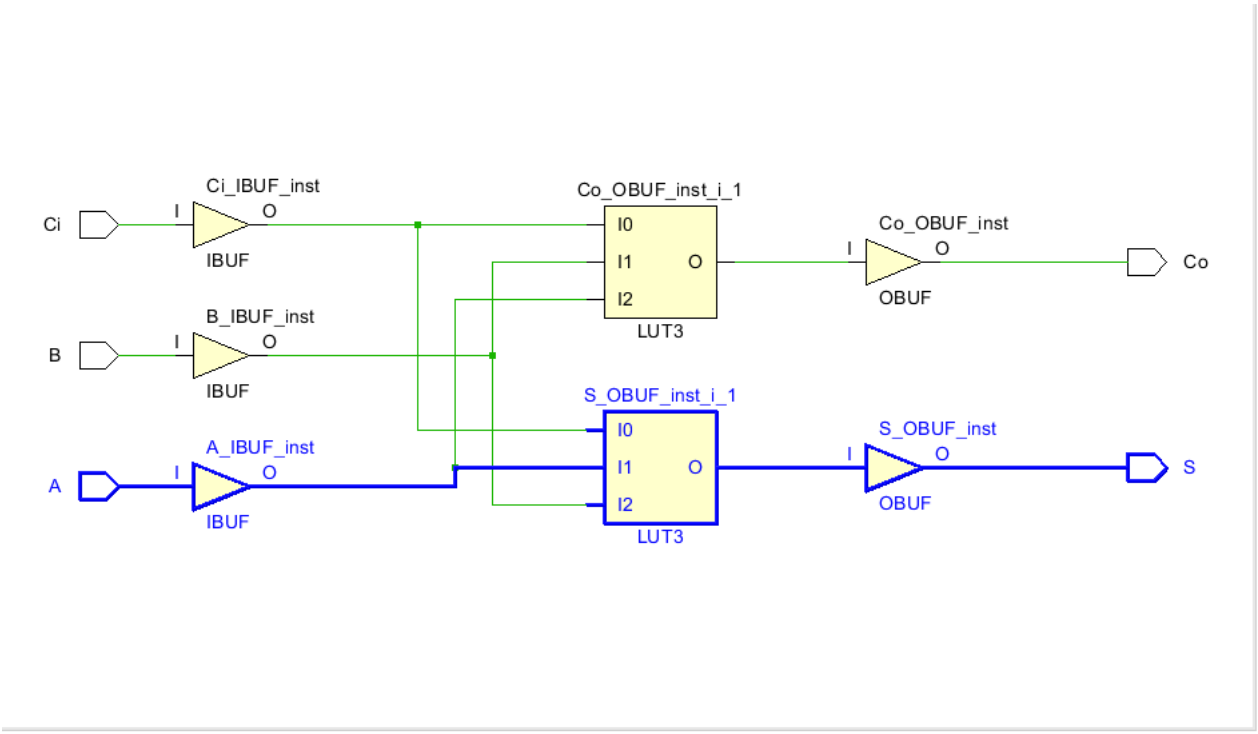
Κώδικας testbench και Κυματομορφή εξόδου

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


entity full_adder_tb is
--  Port ( );
end full_adder_tb;

architecture Behavioral of full_adder_tb is

COMPONENT full_adder IS
    Port ( A : in STD_LOGIC;
           B : in STD_LOGIC;
           Ci : in STD_LOGIC;
           S : out STD_LOGIC;
           Co : out STD_LOGIC);
end COMPONENT;

SIGNAL A, B, Ci, S, Co : STD_LOGIC;

CONSTANT T : TIME := 10 ns;

begin

 uut: full_adder PORT MAP (A => A,
                           B => B,
                           Ci => Ci,
                           S => S,
                           Co => Co);

 stimuli: PROCESS
 begin
```

```
33      A <= '0';
34      B <= '0';
35      Ci <= '0';
36      WAIT FOR T;
37      A <= '1';
38      WAIT FOR T;
39      A <= '0';
40      B <= '1';
41      WAIT FOR T;
42      A <= '1';
43      WAIT FOR T;
44      A <= '0';
45      B <= '0';
46      Ci <= '1';
47      WAIT FOR T;
48      A <= '1';
49      WAIT FOR T;
50      A <= '0';
51      B <= '1';
52      WAIT FOR T;
53      A <= '1';
54      WAIT FOR T;
55   end PROCESS;
56  end Behavioral;
```



Critical Path και Συνολική Καθυστέρηση



ports | Design Runs | Timing

Unconstrained Paths - NONE - NONE - Setup

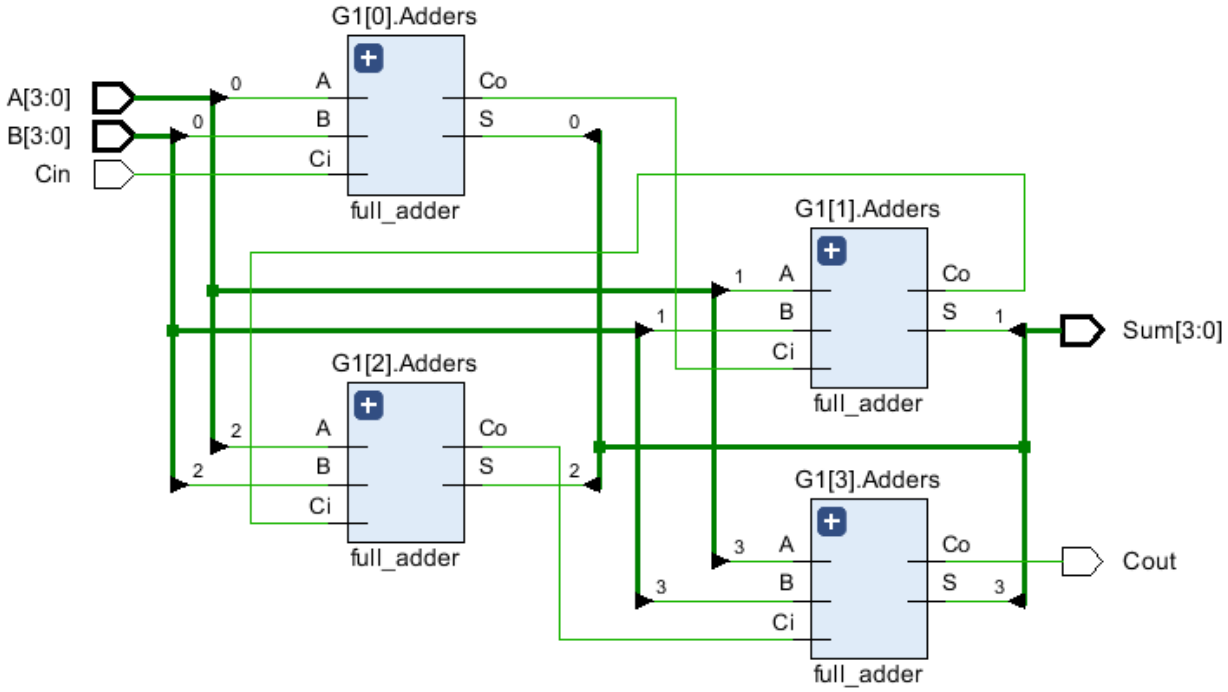
Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destination C
Path 1	∞	3	4	2	A	S	5.377	3.778	1.599	∞	input port clock	
Path 2	∞	3	4	2	A	Co	5.351	3.752	1.599	∞	input port clock	

Όπως φαίνεται από την παραπάνω εικόνα, το critical path είναι από το A στο S και η συνολική καθυστέρηση είναι 5.377ns.

3. Παράλληλος Αθροιστής 4 bits

Κώδικας και δομικό διάγραμμα (RTL)

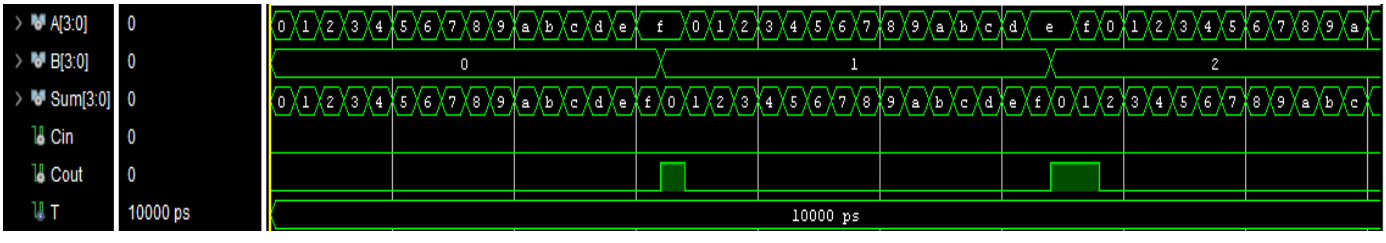
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;


entity bit_4_parallel_adder is
    Port ( A : in STD_LOGIC_VECTOR (3 downto 0);     --The use of Logic_Vectors was made due to the results in the behavioral simulation ->
           B : in STD_LOGIC_VECTOR (3 downto 0);     -- -> So the program prints the results and inputs as a whole binary number.
           Cin : in STD_LOGIC;
           Cout : out STD_LOGIC;
           Sum : out STD_LOGIC_VECTOR (3 downto 0));
end bit_4_parallel_adder;

architecture Structural of bit_4_parallel_adder is


SIGNAL Carry : STD_LOGIC_VECTOR (4 DOWNTO 0) ;

COMPONENT full_adder
    Port ( A : in STD_LOGIC;
           B : in STD_LOGIC;
           Ci : in STD_LOGIC;
           S : out STD_LOGIC;
           Co : out STD_LOGIC);
end COMPONENT;

begin
    --The Logic diagram of The 4 bit Parallel Adder consists of 4 consecutive full adders
    --so for our convenience we used a GENERATE Loop so the code is easier to read.
    Carry(0) <= Cin;
    G1: FOR i IN 0 TO 3 GENERATE
        Adders: full_adder PORT MAP (A => A(i),
                                     B => B(i),
                                     Ci => Carry(i),
                                     S => Sum(i),
                                     Co => Carry(i+1));
        end GENERATE;
    Cout <= Carry(4);

end Structural;
```

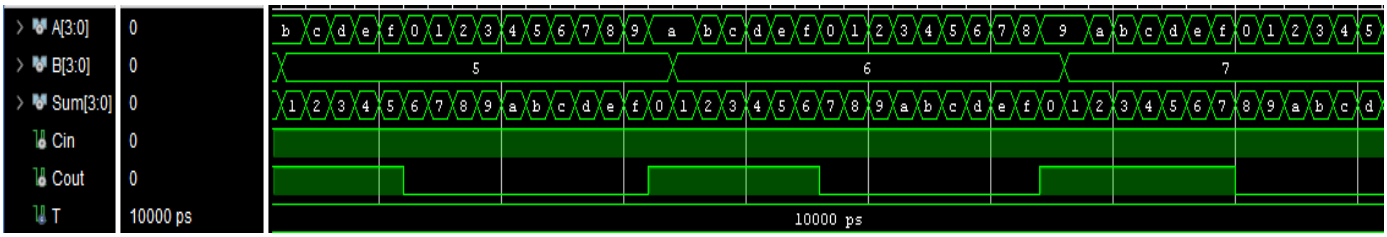


Κώδικας testbench και Κυματομορφή εξόδου

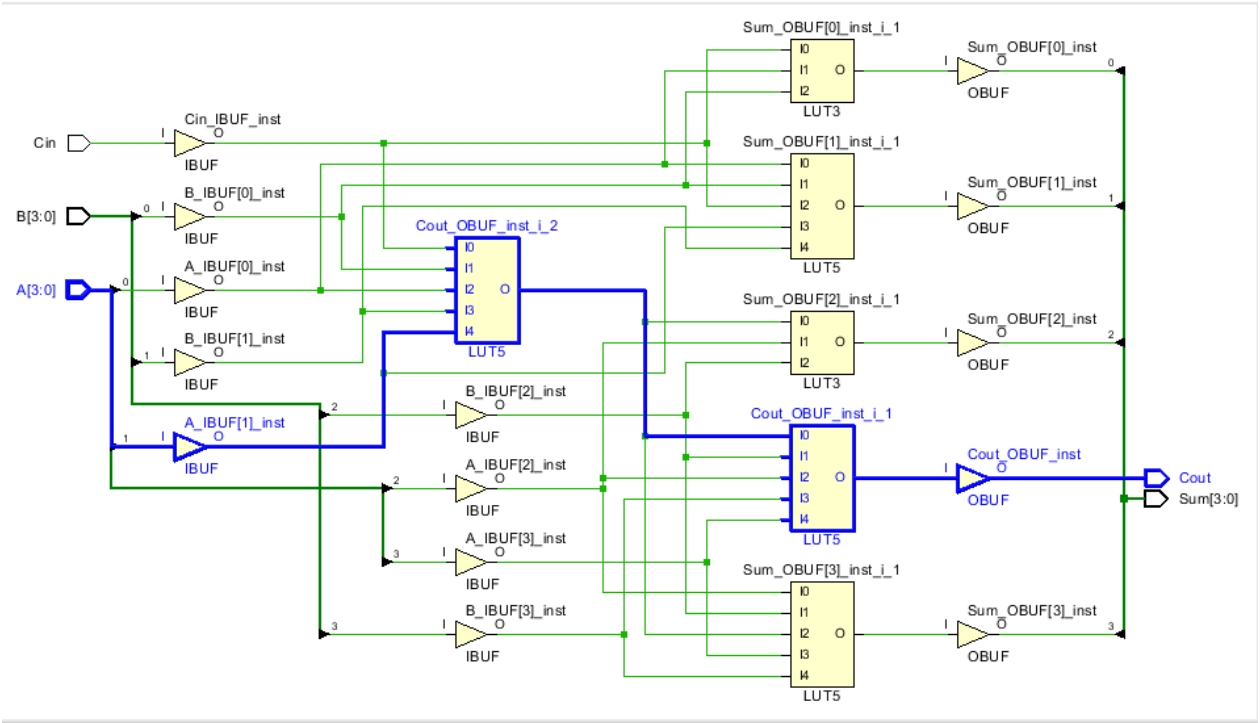
```
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3  USE IEEE.numeric_std.ALL;
4  USE IEEE.std_logic_unsigned.ALL;
5
6  entity bit_4_parallel_adder_tb is
7  --  Port ( );
8  end bit_4_parallel_adder_tb;
9
10 architecture Bench of bit_4_parallel_adder_tb is
11
12 COMPONENT bit_4_parallel_adder is
13     Port ( A : in STD_LOGIC_VECTOR (3 downto 0);
14            B : in STD_LOGIC_VECTOR (3 downto 0);
15            Cin : in STD_LOGIC;
16            Cout : out STD_LOGIC;
17            Sum : out STD_LOGIC_VECTOR (3 downto 0));
18 end COMPONENT;
19
20 SIGNAL A, B, Sum : STD_LOGIC_VECTOR(3 downto 0);
21 SIGNAL Cin : STD_LOGIC := '0';
22 SIGNAL Cout : STD_LOGIC;
23
```

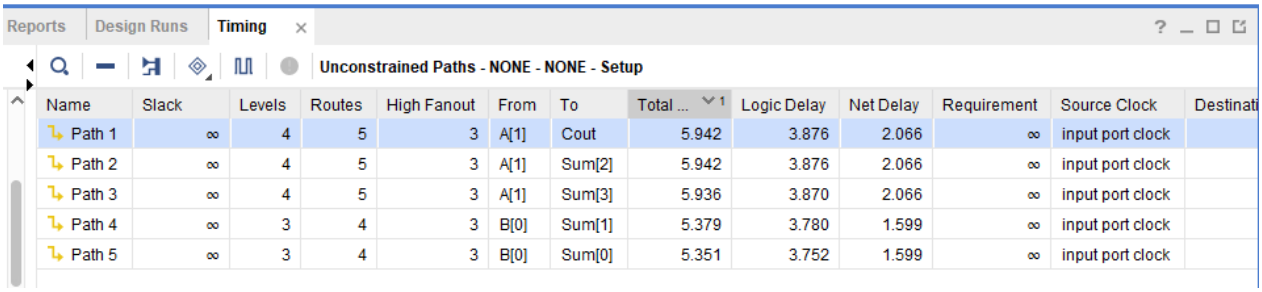
```
24
25  CONSTANT T : TIME := 10 ns;
26
27  begin
28
29  uut: bit_4_parallel_adder PORT MAP (A => A,
30                                      B => B,
31                                      Cin => Cin,
32                                      Cout => Cout,
33                                      Sum => Sum);
34
35  stimuli: PROCESS
36   begin
37      A <= "0000";
38      B <= "0000";
39      WAIT FOR T;
40
41      FOR j IN 1 TO 15 LOOP
42          FOR i IN 1 TO 15 LOOP
43              A <= A + 1;
44              WAIT FOR T;
45          end LOOP;
46          B <= B + 1;
47          WAIT FOR T;
48      END LOOP;
49
50     IF Cin = '0' THEN
51      Cin <= '1';
52     ELSE
53      Cin <= '0';
54     END IF;
55
56   end PROCESS;
57
58  end Bench;
```





Critical Path και Συνολική Καθυστέρηση





Name	Slack	Levels	Routes	High Fanout	From	To	Total ...	Logic Delay	Net Delay	Requirement	Source Clock	Destinati
Path 1	∞	4	5	3	A[1]	Cout	5.942	3.876	2.066	∞	input port clock	
Path 2	∞	4	5	3	A[1]	Sum[2]	5.942	3.876	2.066	∞	input port clock	
Path 3	∞	4	5	3	A[1]	Sum[3]	5.936	3.870	2.066	∞	input port clock	
Path 4	∞	3	4	3	B[0]	Sum[1]	5.379	3.780	1.599	∞	input port clock	
Path 5	∞	3	4	3	B[0]	Sum[0]	5.351	3.752	1.599	∞	input port clock	

Όπως φαίνεται από την παραπάνω εικόνα, το critical path είναι από το A[1] στο Cout και η συνολική καθυστέρηση είναι 5.942ns.

4. BCD Πλήρης Αθροιστής

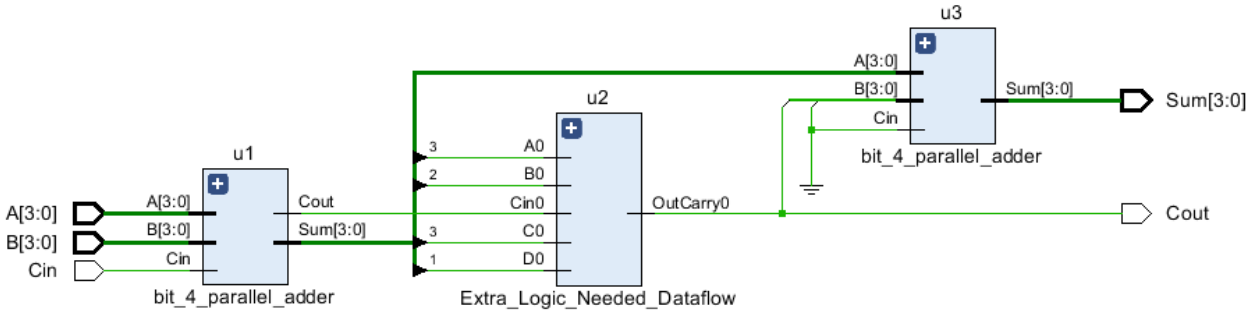
Κώδικας και δομικό διάγραμμα (RTL)

```vhdl
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity Extra_Logic_Needed_Dataflow is
  port(
       A0,B0,C0,D0,Cin0 : in  std_logic;
       OutCarry0    : out std_logic
      );
end entity; -- extra logic

architecture dataflow_arch of Extra_Logic_Needed_Dataflow is

    signal L1,L2,L3 : std_logic;

begin

  L1<=A0 and B0;
  L2<=C0 and D0;
  L3<=(Cin0 or L1 or L2);
  OutCarry0<= L3;

end architecture ; -- arch
```

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

entity bcd_full_adder is
    Port ( A : in STD_LOGIC_VECTOR (3 downto 0);
           B : in STD_LOGIC_VECTOR (3 downto 0);
           Sum : out STD_LOGIC_VECTOR (3 downto 0);
           Cin : in STD_LOGIC;
           Cout : out STD_LOGIC);   --Cout is the 5th bit of our BCD number,
                                    --where:
                                    --Cout&Sum(3)&Sum(2)&Sum(1)&Sum(0) is our
                                    ↪  whole BCD output number
                                    -- '&' is concatenation
end bcd_full_adder;

architecture Structural of bcd_full_adder is
--We use 2 Components one is the 4bit Parallel Adder and the other one is
↪  Extra_Logic
--which is defined so the resulting code is designed with structural
↪  architecture.
```

```
COMPONENT bit_4_parallel_adder
    Port ( A : in STD_LOGIC_VECTOR (3 downto 0);
           B : in STD_LOGIC_VECTOR (3 downto 0);
           Cin : in STD_LOGIC;
           Cout : out STD_LOGIC;
           Sum : out STD_LOGIC_VECTOR (3 downto 0));
end COMPONENT;

COMPONENT Extra_Logic_Needed_Dataflow
    Port (
            A0,B0,C0,D0,Cin0 : in  std_logic;
            OutCarry0    : out std_logic
           );
end COMPONENT;

SIGNAL half_output : STD_LOGIC_VECTOR (3 DOWNTO 0) ;
SIGNAL D1 : STD_LOGIC;
SIGNAL D2 : STD_LOGIC;
SIGNAL D3 : STD_LOGIC;
SIGNAL ign: STD_LOGIC;
SIGNAL Y1 : STD_LOGIC;

begin
    --First Parallel Adder
    u1: bit_4_parallel_adder PORT MAP (A => A,
                                       B => B,
                                       Cin => Cin,
                                       Cout => D1,
                                       Sum => half_output);

    --Extra Logic
    u2: Extra_Logic_Needed_Dataflow PORT MAP (
                                      A0 => half_output(3),
                                      B0 => half_output(2),
                                      C0 => half_output(3),
                                      D0 => half_output(1),
                                      Cin0 => D1,
                                      OutCarry0 =>  Y1
                                         );

    --Second Parallel Adder
    u3: bit_4_parallel_adder PORT MAP (A => half_output,
                                       B(0) => '0',
                                       B(1) => Y1,
                                       B(2) => Y1,
                                       B(3) => '0',
                                       Cin => '0',
                                       Cout => ign,
```

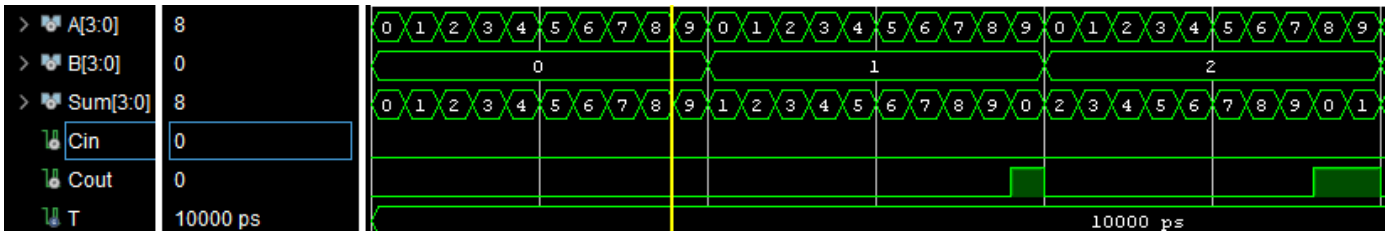
```
                                        Sum => Sum);

    Cout <= Y1;

end Structural;
```

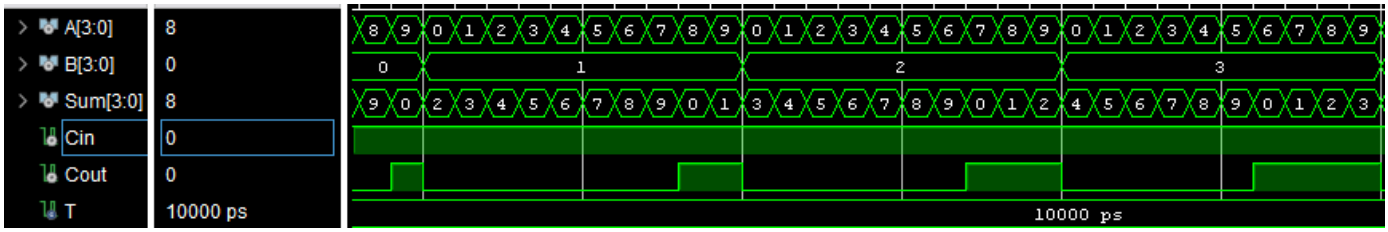


Κώδικας testbench και Κυματομορφή εξόδου

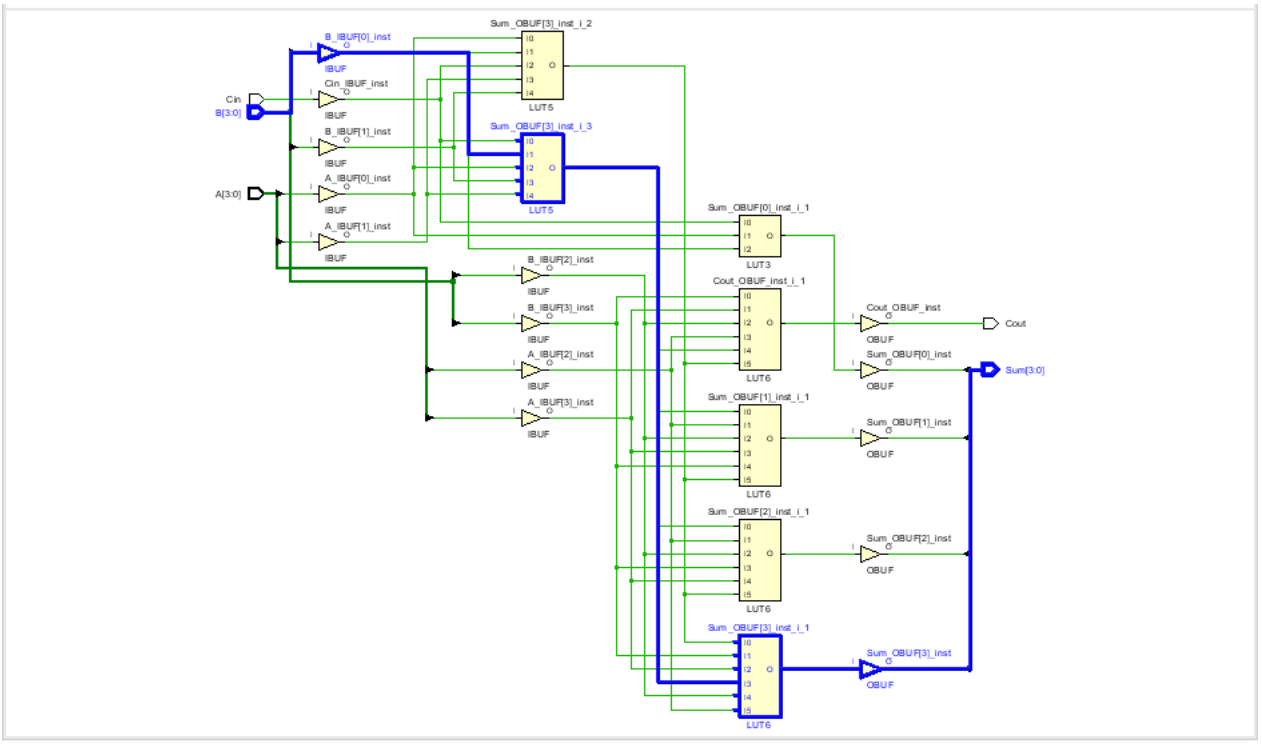
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
USE IEEE.numeric_std.ALL;
USE IEEE.std_logic_unsigned.ALL;

entity bcd_full_adder_tb is
--  Port ( );
end bcd_full_adder_tb;

architecture Bench of bcd_full_adder_tb is

    COMPONENT bcd_full_adder is
        Port ( A : in STD_LOGIC_VECTOR (3 downto 0);
               B : in STD_LOGIC_VECTOR (3 downto 0);
               Sum : out STD_LOGIC_VECTOR (3 downto 0);
               Cin : in STD_LOGIC;
               Cout : out STD_LOGIC);
    end COMPONENT;

    SIGNAL A, B, Sum : STD_LOGIC_VECTOR(3 downto 0);
    SIGNAL Cin : STD_LOGIC := '0';
    SIGNAL Cout : STD_LOGIC;


    CONSTANT T : TIME := 10 ns;

begin

    uut: bcd_full_adder PORT MAP (A => A,
```

```
                                    B => B,
                                    Cin => Cin,
                                    Cout => Cout,
                                    Sum => Sum);

    stimuli: PROCESS
     begin
        A <= "0000";
        B <= "0000";


        FOR j IN 1 TO 9 LOOP
        WAIT FOR T;
            FOR i IN 1 TO 9 LOOP
                A <= A + 1;
                WAIT FOR T;
            end LOOP;
            A <= "0000";
            B <= B + 1;
        END LOOP;

       IF Cin = '0' THEN
        Cin <= '1';
       ELSE
        Cin <= '0';
       END IF;

     end PROCESS;

end Bench;
```





Critical Path και Συνολική Καθυστέρηση



ports | Design Runs | **Timing**

Unconstrained Paths - NONE - NONE - Setup

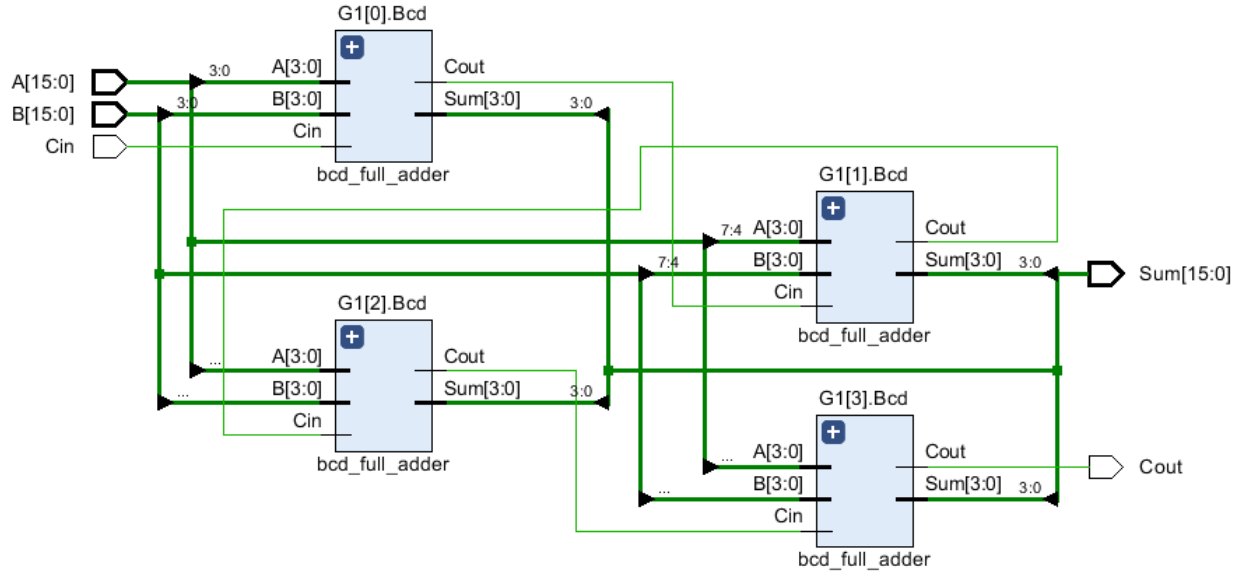
Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destinati
Path 1	∞	4	5	4	B[0]	Sum[3]	5.976	3.904	2.072	∞	input port clock	
Path 2	∞	4	5	4	B[1]	Cout	5.948	3.876	2.072	∞	input port clock	
Path 3	∞	4	5	4	B[1]	Sum[1]	5.948	3.876	2.072	∞	input port clock	
Path 4	∞	4	5	4	B[1]	Sum[2]	5.948	3.876	2.072	∞	input port clock	
Path 5	∞	3	4	3	B[0]	Sum[0]	5.351	3.752	1.599	∞	input port clock	

Όπως φαίνεται από την παραπάνω εικόνα, το critical path είναι από το B[0] στο Sum[3] και η συνολική καθυστέρηση είναι 5.976ns.

5. Παράλληλος BCD Αθροιστής 4ων ψηφίων

Κώδικας και δομικό διάγραμμα (RTL)

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

entity bcd_4_parallel_adder is
    Port ( A : in STD_LOGIC_VECTOR (15 downto 0);
           B : in STD_LOGIC_VECTOR (15 downto 0);
           Cin : in STD_LOGIC;
           Sum : out STD_LOGIC_VECTOR (15 downto 0);
           Cout : out STD_LOGIC);   --Cout is the 17th bit of our BCD number
                                    --where:
                                    --Cout&Sum(15...0) is our whole BCD output
                                    ↪  number
                                    -- '&' is concatenation
end bcd_4_parallel_adder;

architecture Structural of bcd_4_parallel_adder is
--The Code of this structure is similar to the 4 bit Parallel Adder we designed
↪  in 3rd subquestion
--the only difference is the use of the bcd_full_adder here as the component.
 COMPONENT bcd_full_adder
    Port ( A : in STD_LOGIC_VECTOR (3 downto 0);
           B : in STD_LOGIC_VECTOR (3 downto 0);
           Sum : out STD_LOGIC_VECTOR (3 downto 0);
           Cin : in STD_LOGIC;
           Cout : out STD_LOGIC);
end COMPONENT;

SIGNAL Carry : STD_LOGIC_VECTOR (4 DOWNTO 0) ;
begin

Carry(0) <= Cin;

G1: FOR i IN 0 TO 3 GENERATE
    Bcd: bcd_full_adder PORT MAP (A => A(4*i + 3 DOWNTO 4*i),
                                  B => B(4*i + 3 DOWNTO 4*i),
                                  Sum => Sum(4*i + 3 DOWNTO 4*i),
                                  Cin => Carry(i),
                                  Cout => Carry(i+1));


   end GENERATE;

Cout <= Carry(4);

end Structural;
```

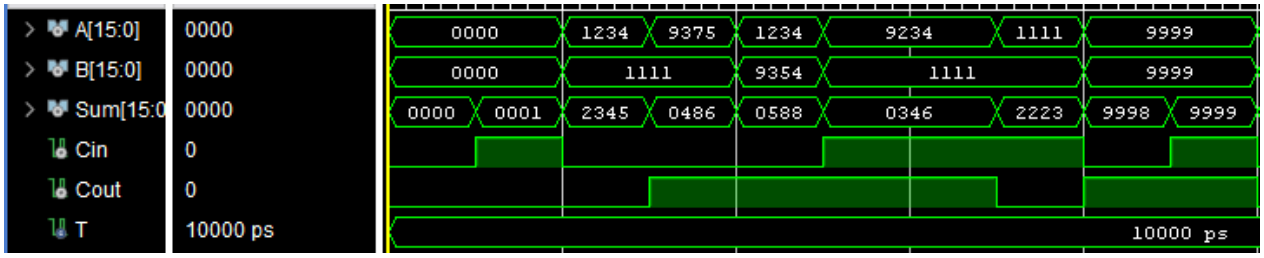


Κώδικας testbench και Κυματομορφή εξόδου

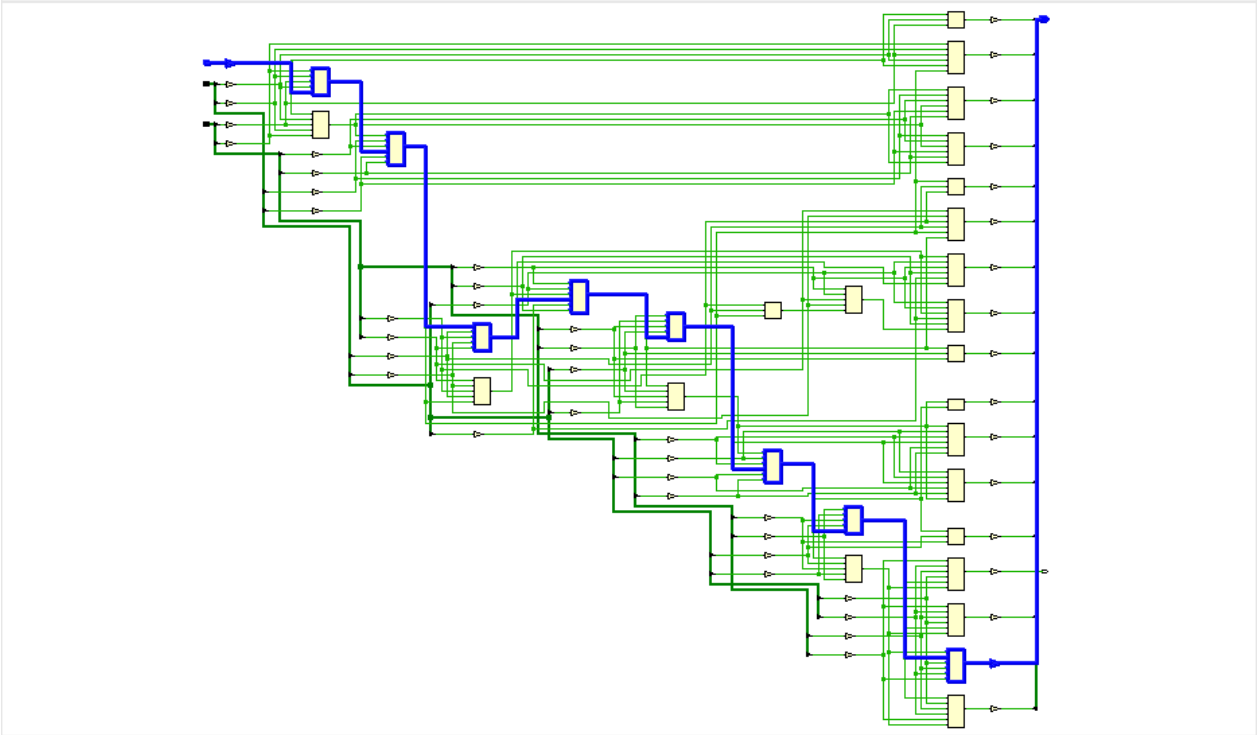
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;


entity bcd_4_parallel_adder_tb is
end entity;

architecture bench of bcd_4_parallel_adder_tb is

COMPONENT bcd_4_parallel_adder IS
    Port ( A : in STD_LOGIC_VECTOR (15 downto 0);
           B : in STD_LOGIC_VECTOR (15 downto 0);
           Cin : in STD_LOGIC;
           Sum : out STD_LOGIC_VECTOR (15 downto 0);
           Cout : out STD_LOGIC);
end COMPONENT;

SIGNAL A, B, Sum : STD_LOGIC_VECTOR (15 downto 0);
SIGNAL Cin, Cout : STD_LOGIC;

CONSTANT T : TIME := 10 ns;


begin

    uut: bcd_4_parallel_adder PORT MAP (A => A,
                                        B => B,
```

```
28                                          Cin => Cin,
29                                          Sum => Sum,
30                                          Cout => Cout);
31
32      stimuli: PROCESS
33      BEGIN
34          A <= "0000000000000000";  --0000
35          B <= "0000000000000000";  --0000
36          Cin <= '0';
37          WAIT FOR T;
38
39          A <= "0000000000000000";  --0000
40          B <= "0000000000000000";  --0000
41          Cin <= '1';
42          WAIT FOR T;
43
44          A <= "0001001000110100";  --1234
45          B <= "0001000100010001";  --1111
46          Cin <= '0';
47          WAIT FOR T;
48
49          A <= "1001001101110101";  --9375
50          B <= "0001000100010001";  --1111
51          WAIT FOR T;
52
53          A <= "0001001000110100";  --1234
54          B <= "1001001101010100";  --9354
55          WAIT FOR T;
56
57          A <= "1001001000110100";  --9234
58          B <= "0001000100010001";  --1111
59          Cin <= '1';
60          WAIT FOR T;
61
62          A <= "1001001000110100";  --0000
63          B <= "0001000100010001";  --1111
64          Cin <= '1';
65          WAIT FOR T;
66
67          A <= "0001000100010001";  --1111
68          B <= "0001000100010001";  --1111
69          Cin <= '1';
70          WAIT FOR T;
71
72          A <= "1001100110011001";  --9999
73          B <= "1001100110011001";  --9999
74          Cin <= '0';
75          WAIT FOR T;
```

```
76
77          A <= "1001100110011001";  --9999
78          B <= "1001100110011001";  --9999
79          Cin <= '1';
80          WAIT FOR T;
81      end PROCESS;
82
83  end bench;
```



Critical Path και Συνολική Καθυστέρηση



orts | Design Runs | Timing

Unconstrained Paths - NONE - NONE - Setup

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Des
Path 1	∞	10	11	6	Cin	Sum[14]	9.513	4.620	4.893	∞	input port clock	
Path 2	∞	10	11	6	Cin	Cout	9.507	4.614	4.893	∞	input port clock	
Path 3	∞	10	11	6	Cin	Sum[13]	9.507	4.614	4.893	∞	input port clock	
Path 4	∞	10	11	6	Cin	Sum[15]	9.507	4.614	4.893	∞	input port clock	
Path 5	∞	9	10	6	Cin	Sum[12]	8.916	4.496	4.420	∞	input port clock	
Path 6	∞	9	10	6	Cin	Sum[9]	8.910	4.490	4.420	∞	input port clock	

Όπως φαίνεται από την παραπάνω εικόνα, το critical path είναι από το Cin στο Sum[14] και η συνολική καθυστέρηση είναι 9.513ns.