Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Εξάμηνο: 8ο Ακ.
Έτος: 2024- 2025

«Ψηφιακά Συστήματα VLSI»

Χαράλαμπος Παπαδόπουλος (ΑΜ: 03120199)
Νικόλαος Παπακωνσταντόπουλος (ΑΜ: 03120069)

Ομάδα: 45

BCD Parallel Adder

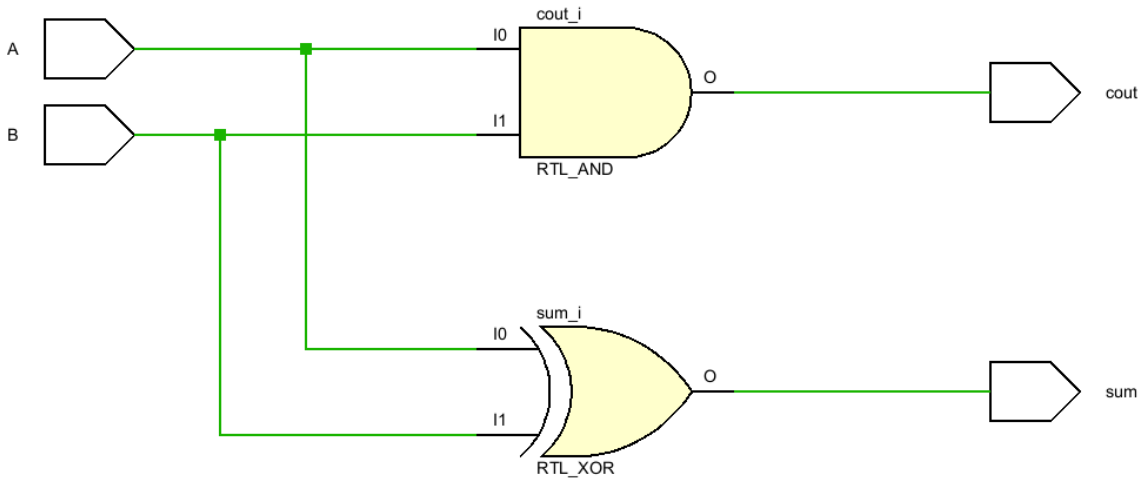
Ζητούμενο αυτής της άσκησης ήταν η υλοποίηση ενός BCD Parallel Adder 4 ψηφίων, ο οποίος παίρνει 2 Binary Coded Decimal αριθμούς των 4 ψηφίων (δηλαδή από 0 έως 9999) και τους αθροίζει κάνοντας διόρθωση στην έξοδο μέσω του c_{out}.

Η άσκηση ακολούθησε μία bottom up λογική, δηλαδή σε κάθε ερώτημα μας ζητούταν να σχεδιάσουμε ένα κύκλωμα το οποίο θα χρησιμοποιούταν ως δομική μονάδα σε ένα πιο περίπλοκο κύκλωμα μέσω structural αρχιτεκτονικής.

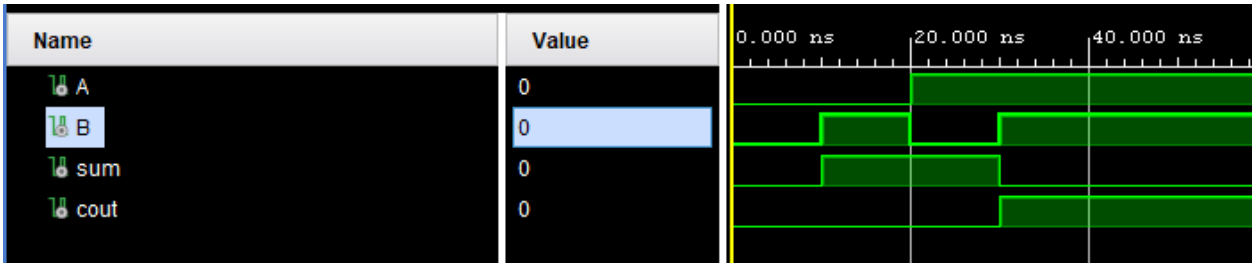
Παρατίθενται οι κώδικες κάθε υποερωτήματος, μαζί με τα αντίστοιχα RTL, testbench και critical paths.

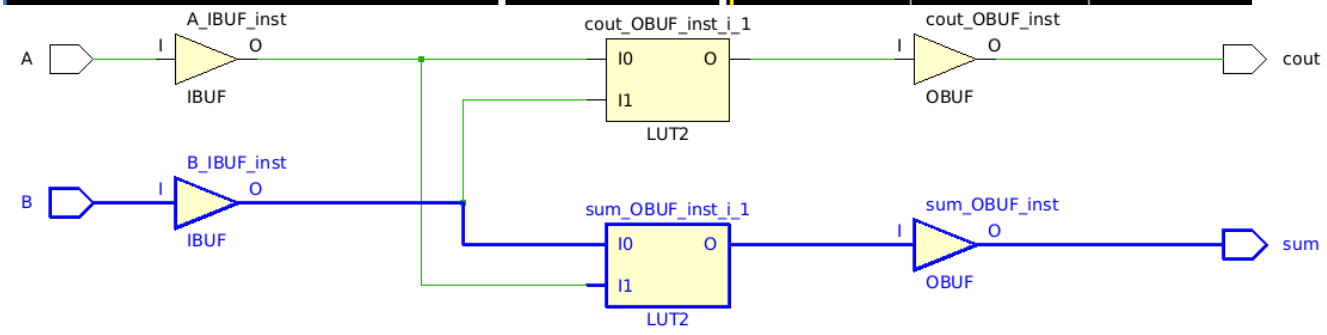
i) *Half Adder*

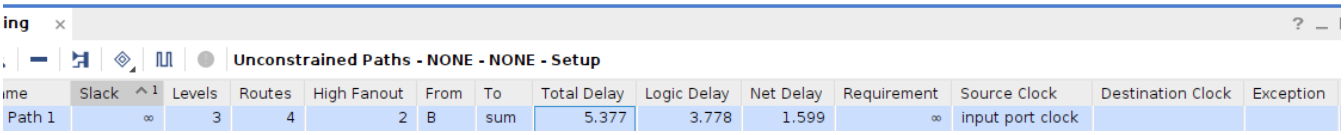
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity half_adder is
  Port (
        A, B        : in  std_logic;
        sum, cout   : out std_logic
        );
end half_adder;

architecture dataflow of half_adder is

begin
    sum  <= A XOR B;
    cout <= A AND B;

end dataflow;
```



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

entity half_adder_tb is
end half_adder_tb;

architecture test of half_adder_tb is
    component half_adder
        Port (
                A, B        :   in  std_logic;
                sum, cout   :   out std_logic
            );
     end component;

     signal A, B        :   std_logic;
     signal sum, cout   :   std_logic;

begin
    uut: half_adder
        port map (
            A    => A,
            B    => B,
            sum  => sum,
            cout => cout
        );

    process
    begin
        A <= '0'; B <= '0'; wait for 10 ns;
        A <= '0'; B <= '1'; wait for 10 ns;
        A <= '1'; B <= '0'; wait for 10 ns;
        A <= '1'; B <= '1'; wait for 10 ns;

    wait;
    end process;

end test;
```





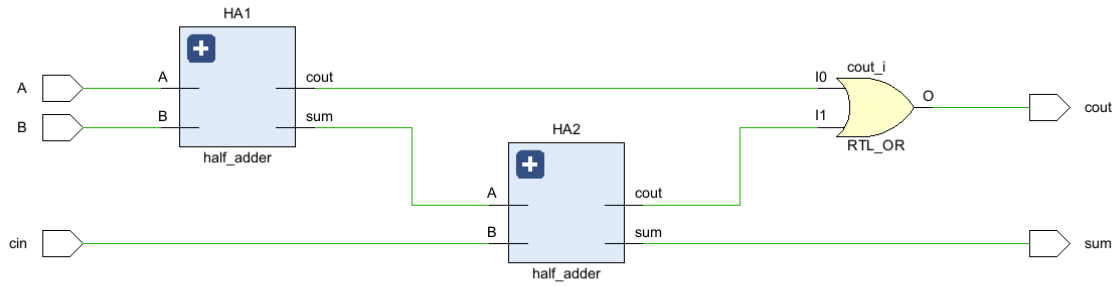


Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Exception
Path 1 ∞	3	4	2	B	sum	5.377	3.778	1.599	∞	input port clock		

Critical Path: $B \rightarrow sum$ 5.377 *ns*

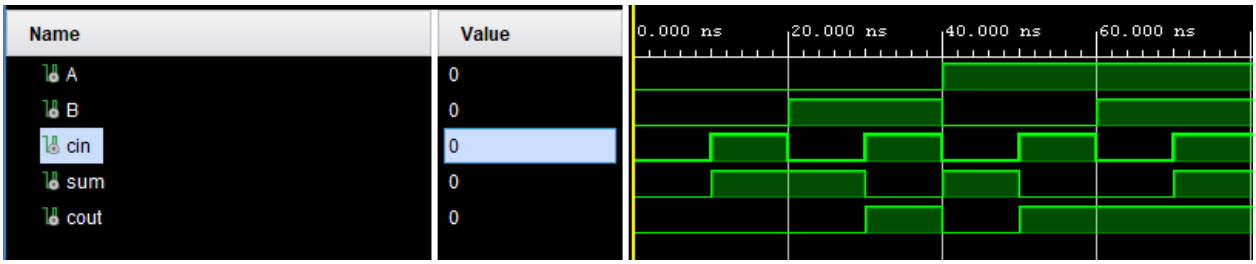
ii) Full adder

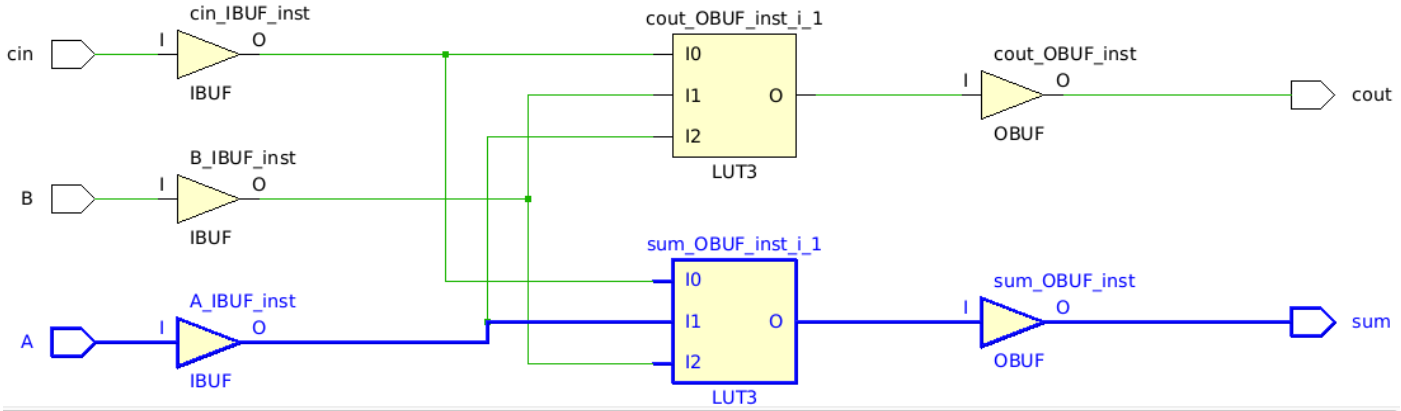
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity full_adder is
  Port (
        A, B, cin   : in  std_logic;
        sum, cout   : out std_logic
       );
end full_adder;

architecture structural of full_adder is

  component half_adder
    Port (
          A, B        : in  std_logic;
          sum, cout   : out std_logic
         );
  end component;

  signal sum1, c1, c2 : std_logic;

begin

  HA1: half_adder
       port map (
            A    => A,
            B    => B,
            sum => sum1,
            cout   => c1
       );

  HA2: half_adder
       port map (
            A       => sum1,
            B       => cin,
            sum     => sum,
            cout    => c2
       );

  cout <= c1 OR c2;

end structural;
```




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

```
entity full_adder_tb is
end full_adder_tb;
architecture testbench of full_adder_tb is

  component full_adder
    Port (
          A, B, cin  : in  std_logic;
          sum, cout  : out std_logic
         );
  end component;

  signal A, B, cin : std_logic := '0';
  signal sum, cout : std_logic;

begin

  uut: full_adder
       port map (
            A    => A,
            B    => B,
            cin  => cin,
            sum  => sum,
            cout => cout
       );

  process
  begin
       A <= '0'; B <= '0'; Cin <= '0'; wait for 10 ns;
       A <= '0'; B <= '0'; Cin <= '1'; wait for 10 ns;
       A <= '0'; B <= '1'; Cin <= '0'; wait for 10 ns;
       A <= '0'; B <= '1'; Cin <= '1'; wait for 10 ns;
       A <= '1'; B <= '0'; Cin <= '0'; wait for 10 ns;
       A <= '1'; B <= '0'; Cin <= '1'; wait for 10 ns;
       A <= '1'; B <= '1'; Cin <= '0'; wait for 10 ns;
       A <= '1'; B <= '1'; Cin <= '1'; wait for 10 ns;
  wait;
  end process;
end testbench;
```





Unconstrained Paths - NONE - NONE - Setup

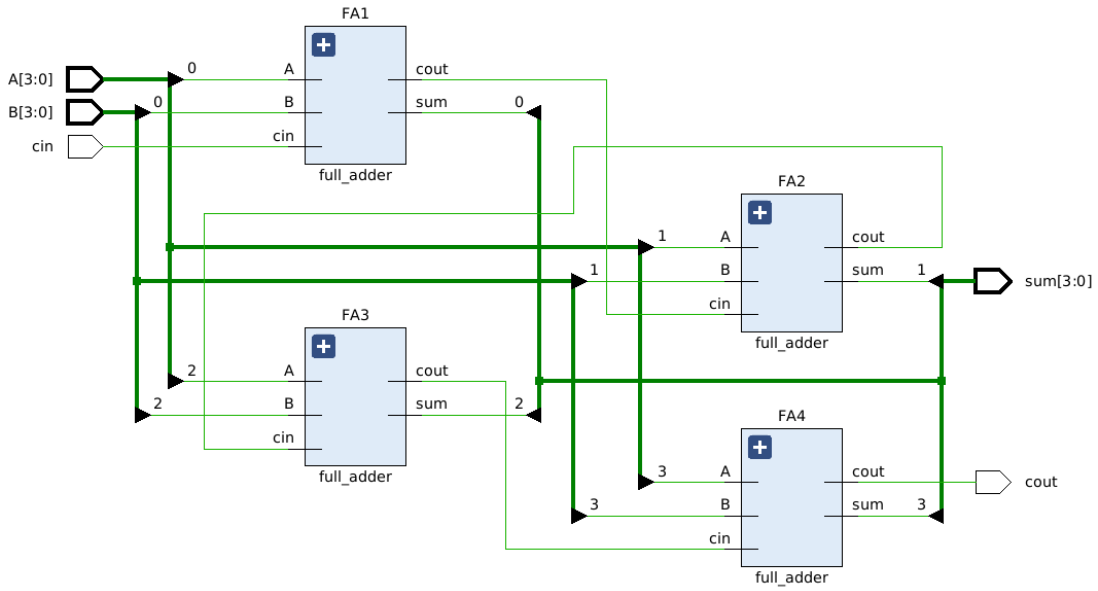
Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Exc
Path 1	∞	3	4	2	A	sum	5.377	3.778	1.599	∞	input port clock		

Critical Path: $A \rightarrow sum$ 5.377 *ns* (ίσο με πριν, από είσοδο ως sum)

iii) Parallel Adder

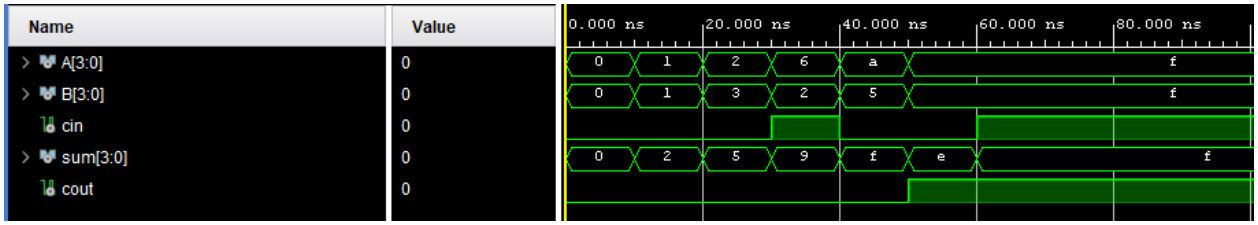
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity parallel_adder is
    Port (
        A, B        : in  std_logic_vector(4-1 downto 0);
        cin         : in  std_logic;
        sum         : out std_logic_vector(4-1 downto 0);
        cout        : out std_logic
    );
end parallel_adder;

architecture structural of parallel_adder is

    component full_adder is
        Port (
            A, B, cin   : in  std_logic;
            sum, cout   : out std_logic
        );
    end component;
    signal c1, c2, c3    : std_logic;
begin
    FA1: full_adder
            port map(
                A    => A(0),
                B    => B(0),
                cin  => cin,
                cout => c1,
                sum  => sum(0)
            );

    FA2: full_adder
            port map(
                A    => A(1),
                B    => B(1),
                cin  => c1,
                cout => c2,
                sum  => sum(1)
            );
```

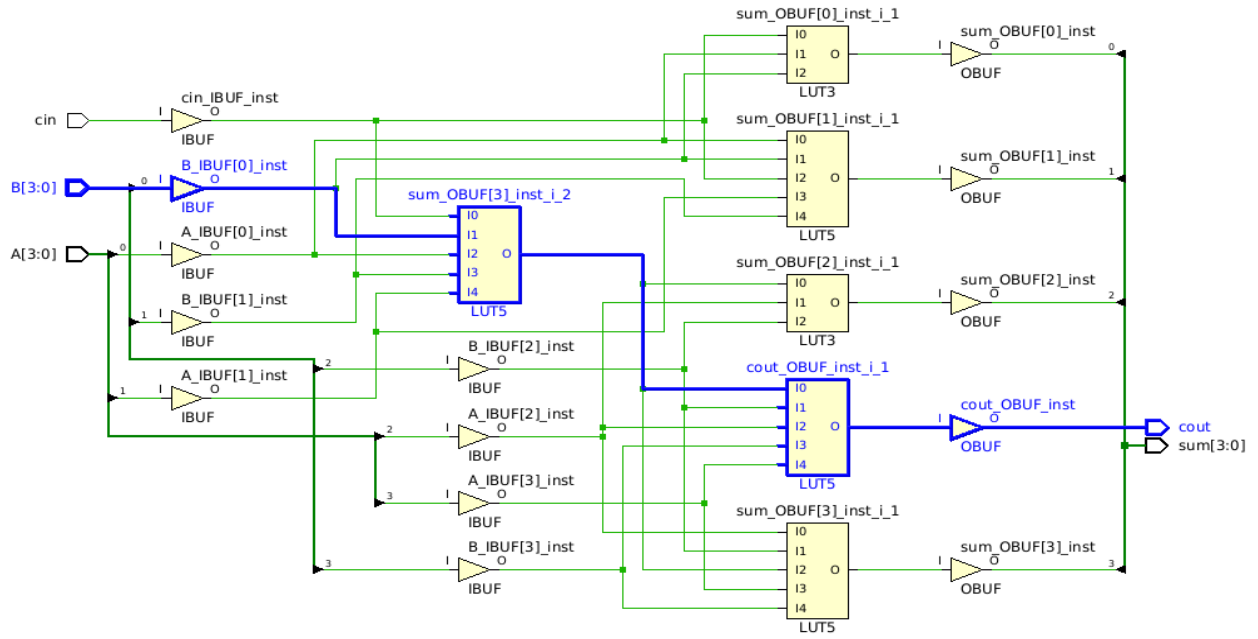
```
    FA3: full_adder
            port map(
                A    => A(2),
                B    => B(2),
                cin  => c2,
                cout => c3,
                sum  => sum(2)
            );

    FA4: full_adder
            port map(
                A    => A(3),
                B    => B(3),
                cin  => c3,
                cout => cout,
                sum  => sum(3)
            );
end structural;
```




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

entity parallel_adder_tb is
end parallel_adder_tb;

architecture testbench 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, B  : std_logic_vector(3 downto 0);
    signal cin   : std_logic;
    signal sum   : std_logic_vector(3 downto 0);
    signal cout  : std_logic;
```

```
begin
    uut: parallel_adder
        port map (
            A    => A,
            B    => B,
            cin  => cin,
            sum  => sum,
            cout => cout
        );

    process
    begin
        A <= "0000"; B <= "0000"; cin <= '0'; wait for 10 ns;    -- 0 + 0
        A <= "0001"; B <= "0001"; cin <= '0'; wait for 10 ns;    -- 1 + 1
        A <= "0010"; B <= "0011"; cin <= '0'; wait for 10 ns;    -- 2 + 3
        A <= "0110"; B <= "0010"; cin <= '1'; wait for 10 ns;    -- 6 + 2 + 1
        A <= "1010"; B <= "0101"; cin <= '0'; wait for 10 ns;    -- 10 + 5
        A <= "1111"; B <= "1111"; cin <= '0'; wait for 10 ns;    -- 15 + 15
        A <= "1111"; B <= "1111"; cin <= '1'; wait for 10 ns;    -- 15 + 15 + 1

        wait;
    end process;
end testbench;
```





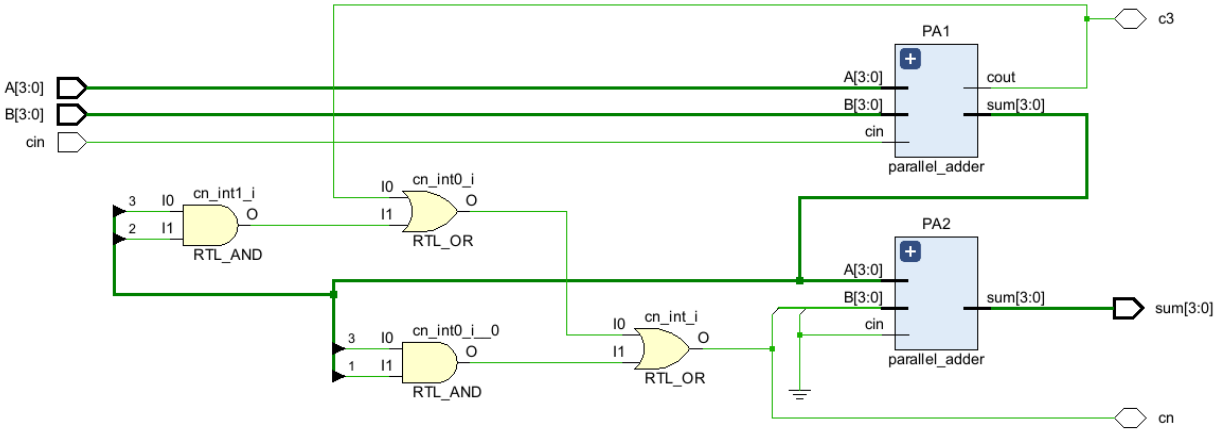
Unconstrained Paths - NONE - NONE - Setup

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destina
Path 1	∞	4	5	3	B[0]	cout	5.970	3.904	2.066	∞	input port clock	

Critical Path: $B \rightarrow c_{out}$ 5.970 *ns*

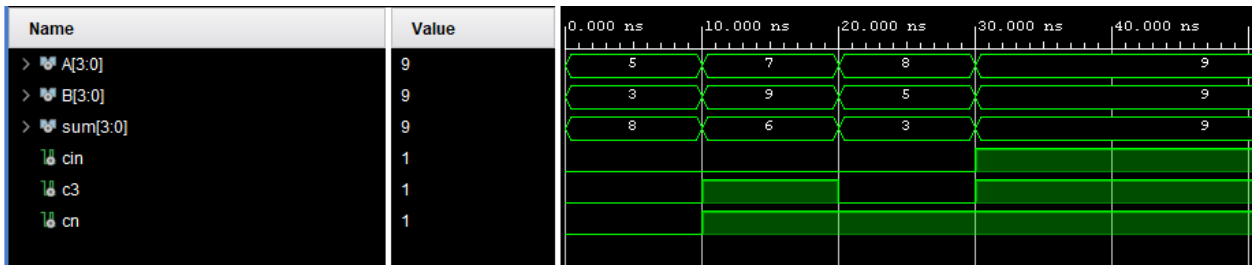
iv) BCD Full Adder

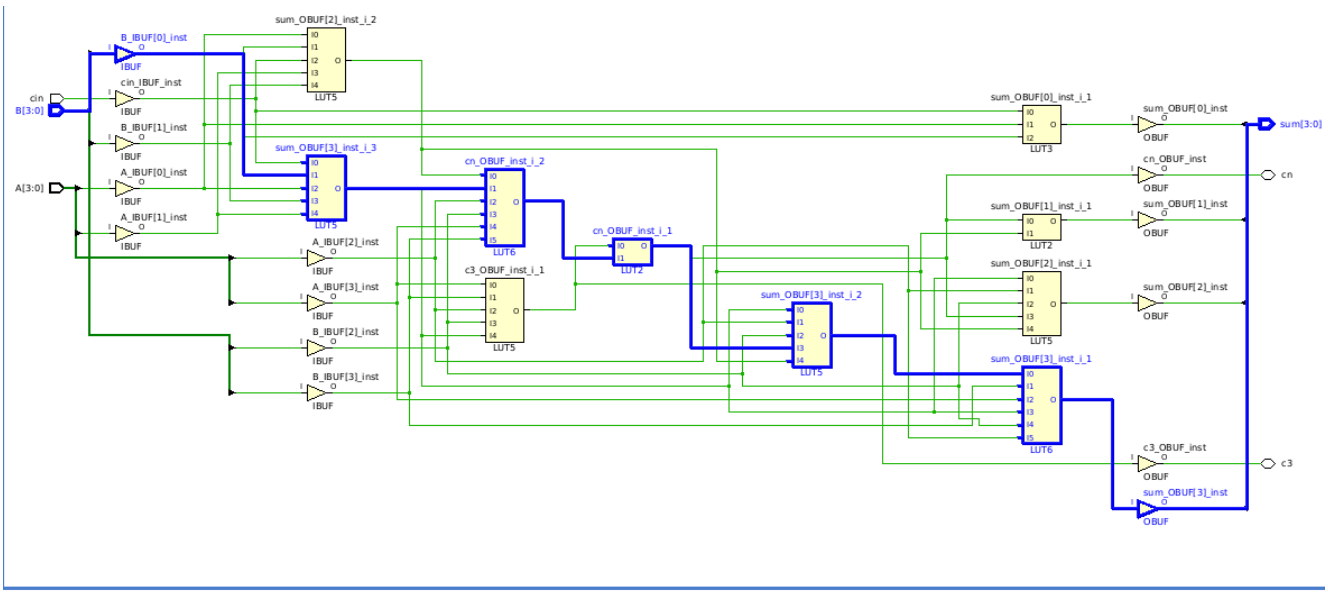
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity BCD_full_adder is
    Port (
        A, B        : in  std_logic_vector(4-1 downto 0);
        cin         : in  std_logic;
        sum         : out std_logic_vector(4-1 downto 0);
        c3, cn    : inout std_logic
    );
end BCD_full_adder;

architecture structural of BCD_full_adder is

    signal sum1, B1 : std_logic_vector(3 downto 0);
    signal cn_int   : std_logic;
    signal or_input : std_logic_vector(2 downto 0);

    component parallel_adder is
        Port (
            A, B        : in  std_logic_vector(4-1 downto 0);
            cin         : in  std_logic;
            sum         : out std_logic_vector(4-1 downto 0);
            cout        : out std_logic
        );
    end component;

begin

    PA1: parallel_adder
            port map(
                A    => A,
                B    => B,
                sum  => sum1,
                cin  => cin,
                cout => c3
            );

    B1 <= ('0' & cn_int & cn_int & '0');     -- second input of second PA

    PA2: parallel_adder
            port map(
                A    => sum1,
                B    => B1,
                sum  => sum,
                cin  => '0'
            );

     cn_int <= ( c3 OR (sum1(3) AND sum1(2))OR (sum1(3) and sum1(1)) );
     cn <= cn_int;

end structural;
```




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

entity BCD_full_adder_tb is
end BCD_full_adder_tb;

architecture test of BCD_full_adder_tb is

    component BCD_full_adder
        Port (
            A, B  : in  std_logic_vector(3 downto 0);
            cin   : in  std_logic;
            sum   : out std_logic_vector(3 downto 0);
            c3, cn: inout std_logic
        );
    end component;

    signal A, B, sum : std_logic_vector(3 downto 0);
    signal cin, c3, cn : std_logic;

begin
    uut: BCD_full_adder
        port map (
            A   => A,
            B   => B,
            cin => cin,
            sum => sum,
            c3  => c3,
            cn  => cn
        );

    process
    begin
        A   <= "0101";  -- 5
        B   <= "0011";  -- 3
        cin <= '0';
        wait for 10 ns;

        A   <= "0111";  -- 7
        B   <= "1001";  -- 9
        cin <= '0';
        wait for 10 ns;

        A   <= "1000";  -- 8
```

```
        B   <= "0101";  -- 5
        cin <= '0';
        wait for 10 ns;

        A   <= "1001";  -- 9
        B   <= "1001";  -- 9
        cin <= '1';
        wait for 10 ns;

        wait;
    end process;

end test;
```





Timing

Unconstrained Paths - NONE - NONE - Setup

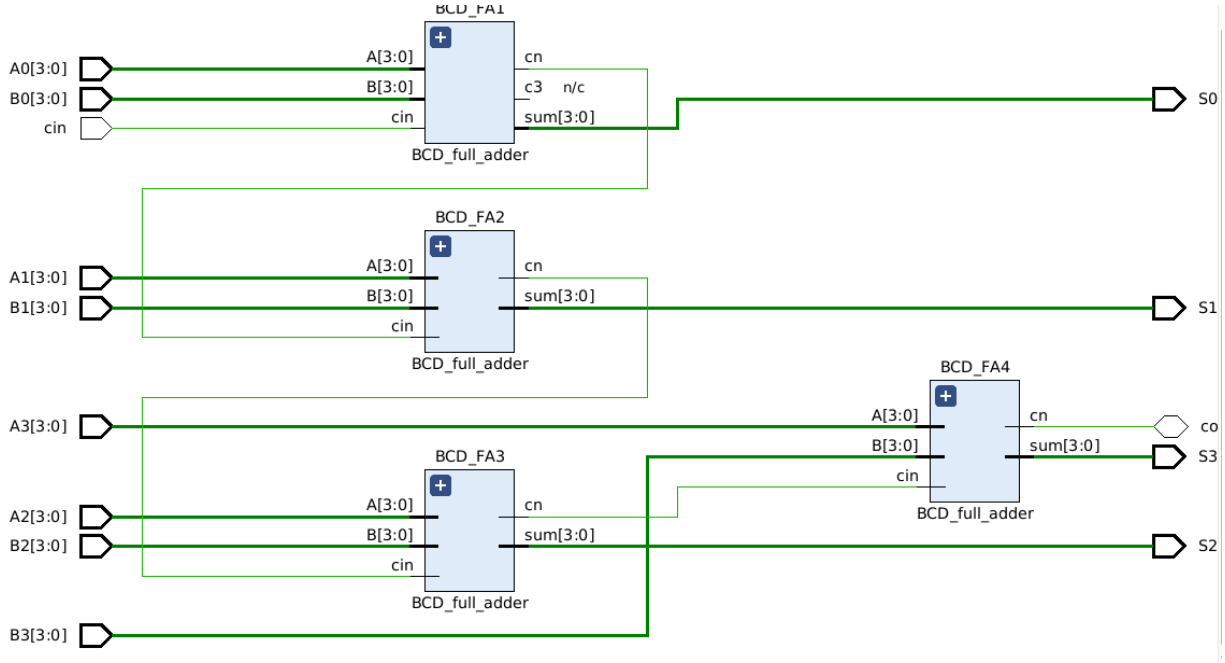
Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Exception
Path 1	∞	7	8	5	B[0]	sum[3]	7.717	4.270	3.447	∞	input port clock		

Critical Path: $B \rightarrow sum$ 7.717 ns

v) BCD Parallel Adder

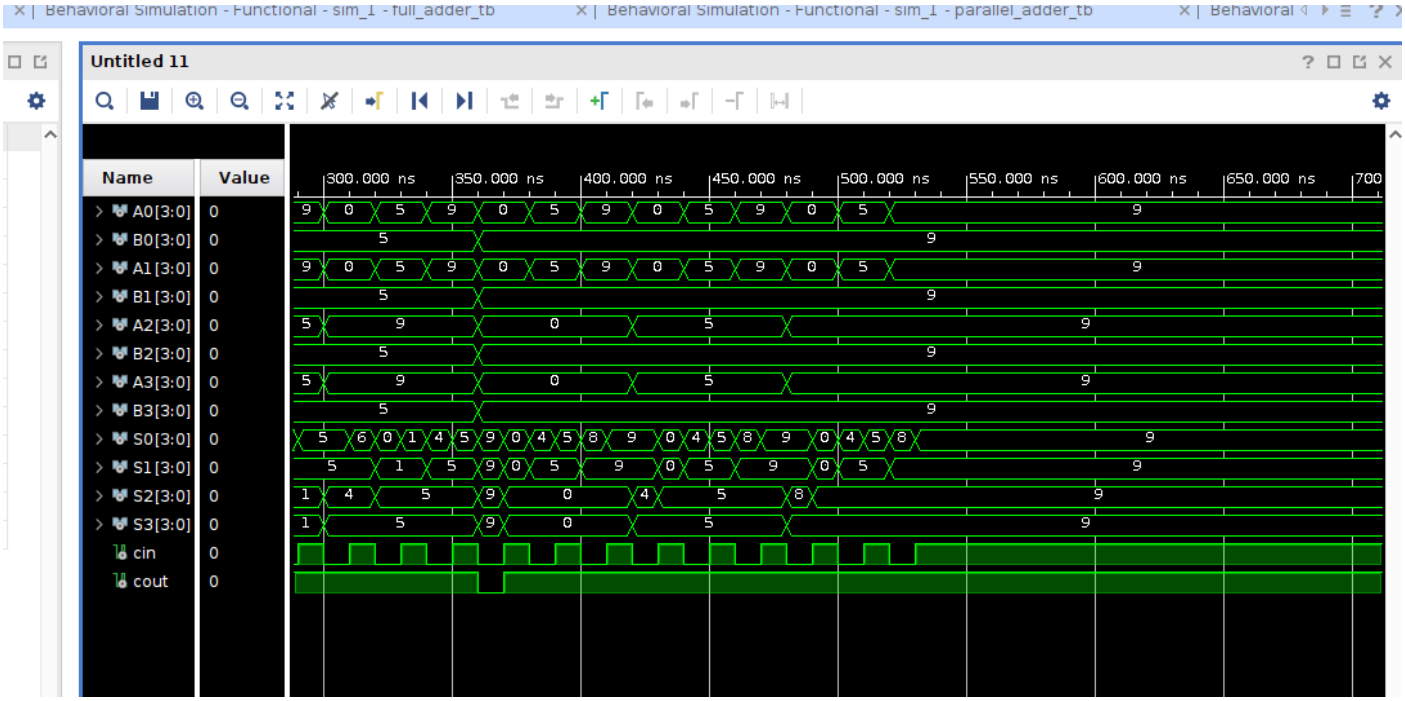
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity BCD_parallel_adder is
    Port (
        A0, B0          : in  std_logic_vector(4-1 downto 0);
        A1, B1          : in  std_logic_vector(4-1 downto 0);
        A2, B2          : in  std_logic_vector(4-1 downto 0);
```

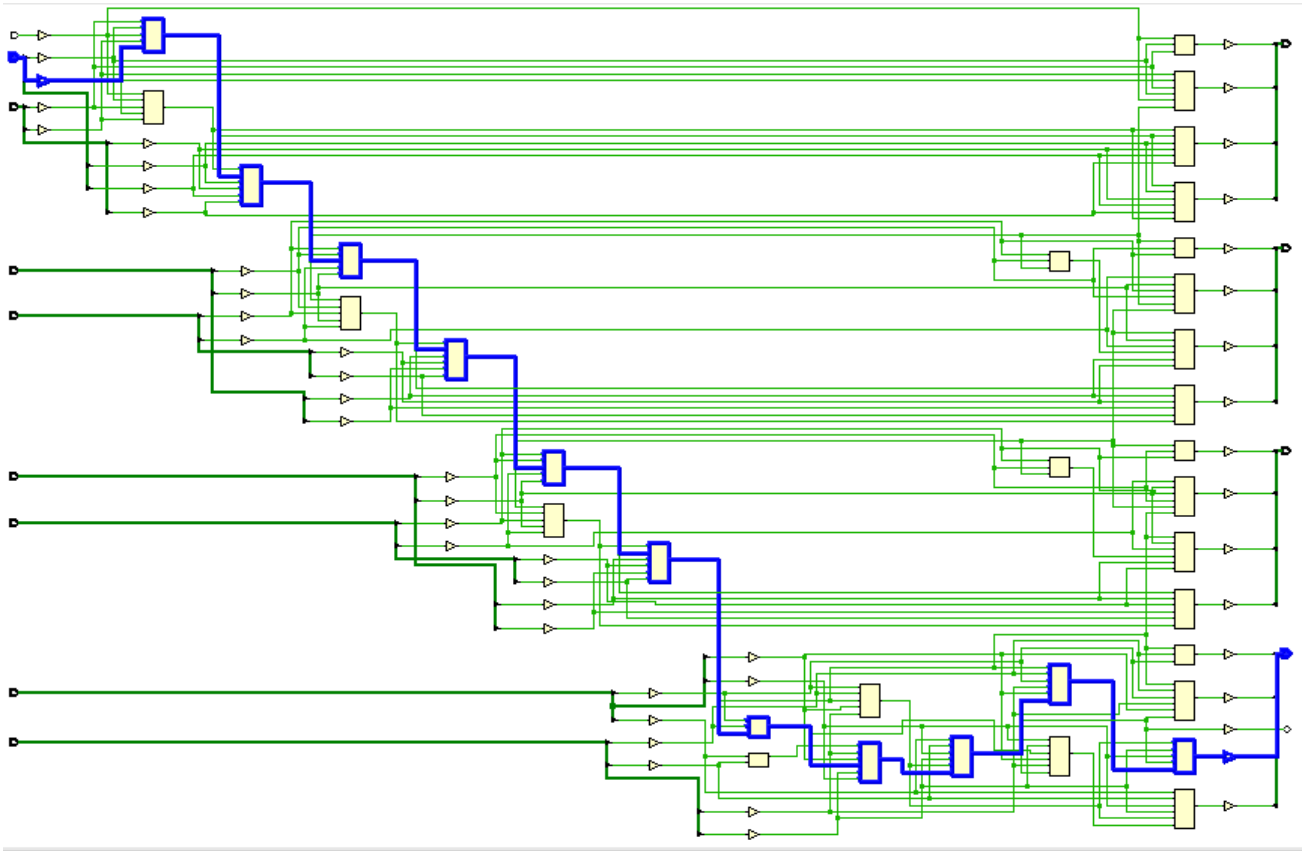
```
        A3, B3        : in  std_logic_vector(4-1 downto 0);
        cin          : in  std_logic;
        S0          : out std_logic_vector(4-1 downto 0);
        S1          : out std_logic_vector(4-1 downto 0);
        S2          : out std_logic_vector(4-1 downto 0);
        S3          : out std_logic_vector(4-1 downto 0);
        cout     : inout std_logic
    );
end BCD_parallel_adder;

architecture Behavioral of BCD_parallel_adder is
    signal cout_0, cout_1, cout_2 : std_logic;

    component BCD_full_adder is
        Port (
            A, B         : in  std_logic_vector(4-1 downto 0);
            cin          : in  std_logic;
            sum          : out std_logic_vector(4-1 downto 0);
            c3, cn     : inout std_logic
        );
    end component;

begin

    BCD_FA1: BCD_full_adder
        port map(
            A    => A0,
            B    => B0,
            sum  => S0,
            cin  => cin,
            cn   => cout_0
        );

    BCD_FA2: BCD_full_adder
        port map(
            A    => A1,
            B    => B1,
            sum  => S1,
            cin  => cout_0,
            cn   => cout_1
        );

    BCD_FA3: BCD_full_adder
        port map(
            A    => A2,
            B    => B2,
            sum  => S2,
            cin  => cout_1,
            cn   => cout_2
        );

    BCD_FA4: BCD_full_adder
        port map(
            A    => A3,
            B    => B3,
            sum  => S3,
            cin  => cout_2,
            cn   => cout
        );


end Behavioral;
```




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

entity BCD_parallel_adder_tb is
end BCD_parallel_adder_tb;

architecture test of BCD_parallel_adder_tb is

    component BCD_parallel_adder is
        Port (
            A0, B0         : in  std_logic_vector(4-1 downto 0);
            A1, B1         : in  std_logic_vector(4-1 downto 0);
            A2, B2         : in  std_logic_vector(4-1 downto 0);
            A3, B3         : in  std_logic_vector(4-1 downto 0);
            cin          : in  std_logic;
            S0           : out std_logic_vector(4-1 downto 0);
            S1           : out std_logic_vector(4-1 downto 0);
            S2           : out std_logic_vector(4-1 downto 0);
            S3           : out std_logic_vector(4-1 downto 0);
            cout     : inout std_logic
        );
    end component;

    signal A0, B0, A1, B1, A2, B2, A3, B3, S0, S1, S2, S3 : std_logic_vector(4-1
downto 0);
    signal cin, cout : std_logic;

begin
    uut: BCD_parallel_adder
        port map (
            A0    => A0,
            B0    => B0,
            A1    => A1,
            B1    => B1,
            A2    => A2,
            B2    => B2,
            A3    => A3,
            B3    => B3,
```

```
            cin  => cin,
            S0   => S0,
            S1   => S1,
            S2   => S2,
            S3   => S3,
            cout => cout
        );

    process
        type test_array is array(0 to 2) of integer range 0 to 15;
        constant tests : test_array := (
            0, 5, 9
        );
    begin
        for i in 0 to 2 loop
            for x in 0 to 2 loop
                for y in 0 to 2 loop

                                B3 <= std_logic_vector(to_unsigned(tests(i), 4));
                                B2 <= std_logic_vector(to_unsigned(tests(i), 4));
                                B1 <= std_logic_vector(to_unsigned(tests(i), 4));
                                B0 <= std_logic_vector(to_unsigned(tests(i), 4));

                                A3 <= std_logic_vector(to_unsigned(tests(x), 4));
                                A2 <= std_logic_vector(to_unsigned(tests(x), 4));
                                A1 <= std_logic_vector(to_unsigned(tests(y), 4));
                                A0 <= std_logic_vector(to_unsigned(tests(y), 4));

                                cin <= '0';
                                wait for 10 ns;

                                cin <= '1';
                                wait for 10 ns;

                end loop;
            end loop;
        end loop;
        wait;
    end process;

end test;
```





Unconstrained Paths - NONE - NONE - Setup

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source
Path 1	∞	13	14	7	B0[1]	S3[2]	11.262	4.992	6.270	∞	input

$Critical\ Path: B0 \rightarrow S3\ \ 11.262ns$