

## گزارشکار آزمایش ۴

نکته : در این گزارشکار، کد مربوط به **FullAdder** و **HalfAdder** و گیت ها به دلیل اینکه در گزارشکارهای قبل بودند، آورده نشده اند.

: RippleAdder4Bits

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

entity RippleAdder4Bits is
port(
    a, b : in STD_LOGIC_VECTOR(3 downto 0);
    cin : in STD_LOGIC;
    s: out STD_LOGIC_VECTOR(3 downto 0);
    cout : out STD_LOGIC
);
end RippleAdder4Bits;

architecture structure of RippleAdder4Bits is
    component FullAdder is
        port(
            i0, i1, cin: in STD_LOGIC;
            Sum, Carry : out STD_LOGIC
        );
    end component FullAdder;

    signal c : STD_LOGIC_VECTOR(3 downto 0);

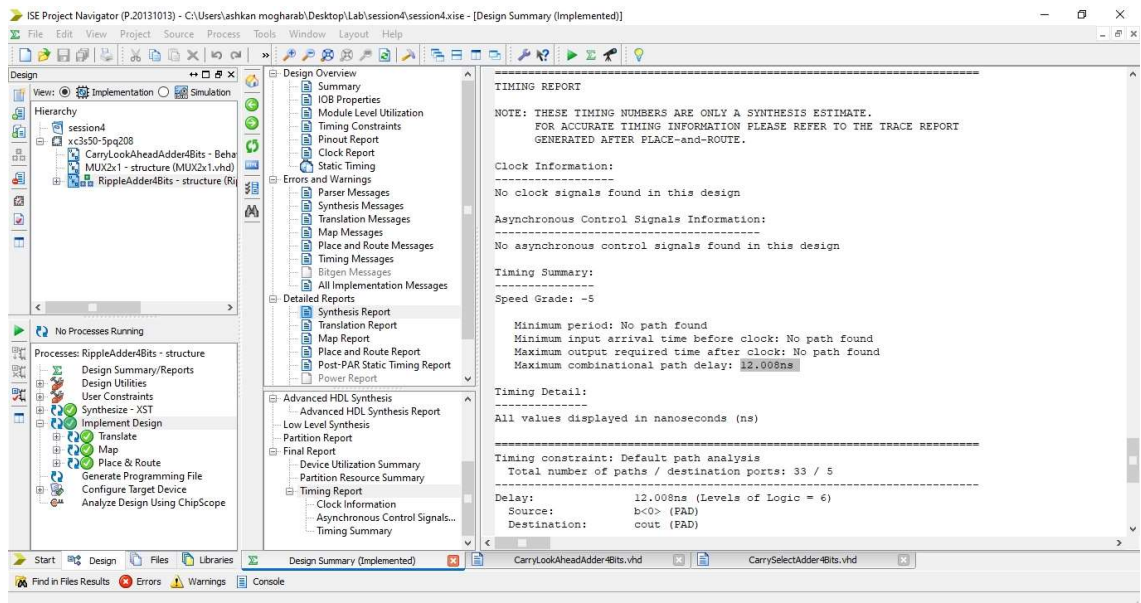
begin
    c(0) <= cin;

    FullAdder1 : FullAdder port map (
        i0 => a(0),
        i1 => b(0),
        cin => c(0),
        Sum => s(0),
        Carry => c(1));
```

```
FullAdder2 : FullAdder port map (  
    i0 => a(1),  
    i1 => b(1),  
    cin => c(1),  
    Sum => s(1),  
    Carry => c(2)  
);  
FullAdder3 : FullAdder port map (  
    i0 => a(2),  
    i1 => b(2),  
    cin => c(2),  
    Sum => s(2),  
    Carry => c(3)  
);  
FullAdder4 : FullAdder port map (  
    i0 => a(3),  
    i1 => b(3),  
    cin => c(3),  
    Sum => s(3),  
    Carry => cout  
);  
end structure;
```

---

## : RippleAdder4Bits مربوط به TIMING REPORT



## : RippleAdder4Bits\_tb

```
LIBRARY ieee;

USE ieee.std_logic_1164.ALL;

ENTITY RippleAdder4Bits_tb IS

END RippleAdder4Bits_tb;

ARCHITECTURE test OF RippleAdder4Bits_tb IS

    -- Component Declaration for the Unit Under Test (UUT)

    COMPONENT RippleAdder4Bits

    PORT(

        a : IN  std_logic_vector(3 downto 0);

        b : IN  std_logic_vector(3 downto 0);

        cin : IN  std_logic;

        s : OUT std_logic_vector(3 downto 0);

        cout : OUT std_logic

    );

    END COMPONENT;

    --Inputs

    signal a : std_logic_vector(3 downto 0) ;

    signal b : std_logic_vector(3 downto 0) ;

    signal cin : std_logic := '0';

    --Outputs

    signal s : std_logic_vector(3 downto 0);

    signal cout : std_logic;

    BEGIN

        -- Instantiate the Unit Under Test (UUT)

        uut: RippleAdder4Bits PORT MAP (

            a => a,
```

```

b => b,

cin => cin,

s => s,

cout => cout

);

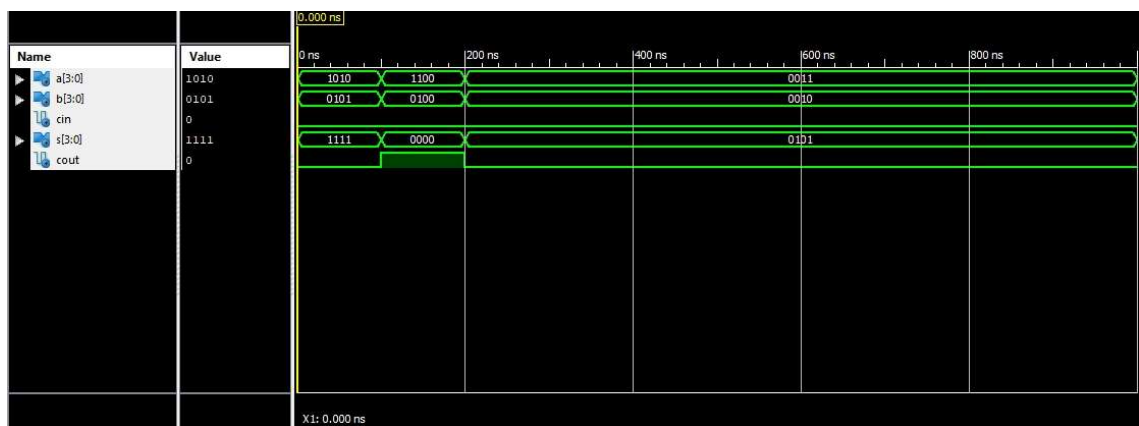
a <="1010" , "1100" after 100 ns , "0011" after 200 ns;

b <="0101" , "0100" after 100 ns , "0010" after 200 ns;

END;

```

خروجی شبیه سازی : RippleAdder4Bits\_tb



## : CarryLookAheadAdder4Bits

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

entity CarryLookAheadAdder4Bits is
port(
    a, b : in STD_LOGIC_VECTOR(3 downto 0);
    cin : in STD_LOGIC;
    s: out STD_LOGIC_VECTOR(3 downto 0);
    cout : out STD_LOGIC
);
end CarryLookAheadAdder4Bits;

architecture Behavioral of CarryLookAheadAdder4Bits is
    signal G : STD_LOGIC_VECTOR(3 downto 0);
    signal P : STD_LOGIC_VECTOR(3 downto 0);
    signal C : STD_LOGIC_VECTOR(3 downto 0);
    signal H : STD_LOGIC_VECTOR(3 downto 0);

begin
    G(0) <= a(0) and b(0);
    G(1) <= a(1) and b(1);
    G(2) <= a(2) and b(2);
    G(3) <= a(3) and b(3);

    P(0) <= a(0) xor b(0);
    P(1) <= a(1) xor b(1);
    P(2) <= a(2) xor b(2);
    P(3) <= a(3) xor b(3);

    C(0) <= cin;
```

C(1) <= G(0) or H(0) ;

C(2) <= G(1) or H(1);

C(3) <= G(2) or H(2);

H(0) <= C(0) and P(0);

H(1) <= C(1) and P(1);

H(2) <= C(2) and P(2);

H(3) <= C(3) and P(3);

s(0) <= P(0) xor C(0);

s(1) <= P(1) xor C(1);

s(2) <= P(2) xor C(2);

s(3) <= P(3) xor C(3);

cout <= G(3) or H(3);

end Behavioral;

## : CarryLookAheadAdder4Bits مربوط به TIMING REPORT

ISE Project Navigator (P.20131013) - C:\Users\ashkan mogharab\Desktop\Lab\session4\session4.xise - [Design Summary (Implemented)]

Design Overview

- Summary
- IOB Properties
- Module Level Utilization
- Timing Constraints
- Pinout Report
- Clock Report
- Static Timing
- Errors and Warnings
- Parser Messages
- Synthesis Messages
- Translation Messages
- Map Messages
- Place and Route Messages
- Timing Messages
- Bitgen Messages
- All Implementation Messages
- Detailed Reports
  - Synthesis Report
  - Translation Report
  - Map Report
  - Place and Route Report
  - Post-PAIR Static Timing Report
  - Power Report

Design Summary (Implemented)

Timing Report

NOTE: THESE TIMING NUMBERS ARE ONLY A SYNTHESIS ESTIMATE.  
FOR ACCURATE TIMING INFORMATION PLEASE REFER TO THE TRACE REPORT  
GENERATED AFTER PLACE-and-ROUTE.

Clock Information:

No clock signals found in this design

Asynchronous Control Signals Information:

No asynchronous control signals found in this design

Timing Summary:

Speed Grade: -S

Minimum period: No path found  
Minimum input arrival time before clock: No path found  
Maximum output required time after clock: No path found  
Maximum combinational path delay: 12.008ns

Timing Detail:

All values displayed in nanoseconds (ns)

Timing constraint: Default path analysis  
Total number of paths / destination ports: 33 / 5



: CLA4Bits\_tb

```
LIBRARY ieee;

USE ieee.std_logic_1164.ALL;

ENTITY CLA4Bits_tb IS
END CLA4Bits_tb;

ARCHITECTURE test OF CLA4Bits_tb IS

    -- Component Declaration for the Unit Under Test (UUT)

    COMPONENT CarryLookAheadAdder4Bits

    PORT(

        a : IN  std_logic_vector(3 downto 0);
        b : IN  std_logic_vector(3 downto 0);
        cin : IN  std_logic;
        s : OUT  std_logic_vector(3 downto 0);
        cout : OUT  std_logic

    );

    END COMPONENT;

    --Inputs

    signal a : std_logic_vector(3 downto 0);
    signal b : std_logic_vector(3 downto 0);
    signal cin : std_logic := '0';

    --Outputs

    signal s : std_logic_vector(3 downto 0);
    signal cout : std_logic;

    BEGIN

        -- Instantiate the Unit Under Test (UUT)
```

ut: CarryLookAheadAdder4Bits PORT MAP (

a => a,

b => b,

cin => cin,

s => s,

cout => cout

);

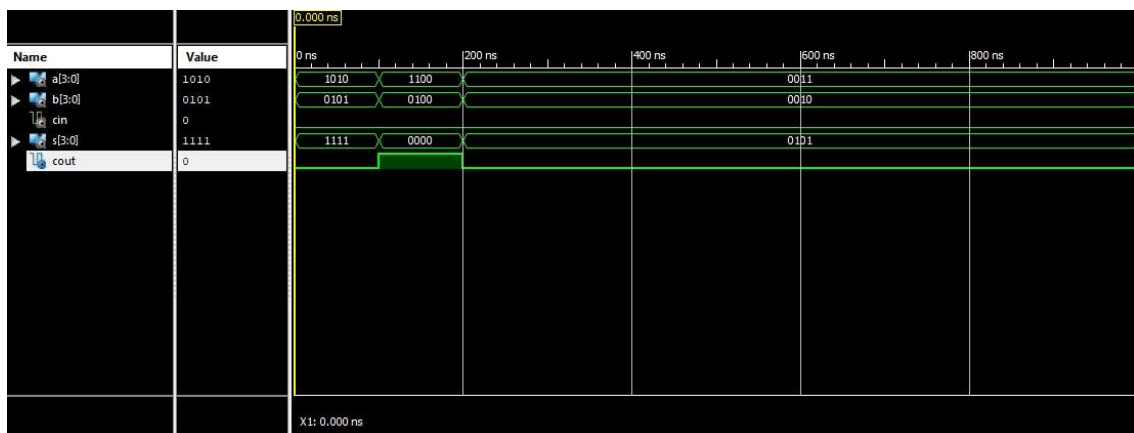
a <="1010" , "1100" after 100 ns , "0011" after 200 ns;

b <="0101" , "0100" after 100 ns , "0010" after 200 ns;

END;

---

## خروجی شبیه سازی CLA4Bits\_tb :



: MUX2x1

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

entity MUX2x1 is
    port(
        i : in STD_LOGIC_VECTOR(1 downto 0);
        s : in STD_LOGIC;
        o : out STD_LOGIC
    );
end entity MUX2x1;

architecture structure of MUX2x1 is
    signal sig1, sig2 : STD_LOGIC;
begin
    sig1 <= not s and i(0);
    sig2 <= s and i(1);
    o <= sig1 or sig2;
end structure;
```

---

: CarrySelectAdder4Bits

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

entity CarrySelectAdder4Bits is
    port(
        a, b : in STD_LOGIC_VECTOR(3 downto 0);
        cin : in STD_LOGIC;
        s: out STD_LOGIC_VECTOR(3 downto 0);
        cout : out STD_LOGIC
    );
end CarrySelectAdder4Bits;
```

architecture structure of CarrySelectAdder4Bits is

component RippleAdder4Bits is

```
    port(  
        a, b : in STD_LOGIC_VECTOR(3 downto 0);  
        cin : in STD_LOGIC;  
        s: out STD_LOGIC_VECTOR(3 downto 0);  
        cout : out STD_LOGIC  
    );
```

end component RippleAdder4Bits;

component MUX2x1 is

```
    port(  
        i : in STD_LOGIC_VECTOR(1 downto 0);  
        s : in STD_LOGIC;  
        o : out STD_LOGIC  
    );
```

end component MUX2x1;

signal s1, s2 : STD\_LOGIC\_VECTOR(3 downto 0); --sum row 1 and 2

signal cout1, cout2 : STD\_LOGIC;

signal sumMUX1 : STD\_LOGIC\_VECTOR(1 downto 0);

signal sumMUX2 : STD\_LOGIC\_VECTOR(1 downto 0);

signal sumMUX3 : STD\_LOGIC\_VECTOR(1 downto 0);

signal sumMUX4 : STD\_LOGIC\_VECTOR(1 downto 0);

signal FinalMux : STD\_LOGIC\_VECTOR(1 downto 0);

begin

RippleAdder4Bits\_1 : RippleAdder4Bits port map (

a => a,

b => b,

```

        cin => '0',
        s => s1,
        cout => cout1
    );

    RippleAdder4Bits_2 : RippleAdder4Bits port map (
        a => a,
        b => b,
        cin => '1',
        s => s2,
        cout => cout2
    );

```

-----

```

    sumMUX1(0) <= s1(0);
    sumMUX1(1) <= s2(0);
    MUX2x1_1 : MUX2x1 port map (
        i => sumMUX1,
        s => cin,
        o => s(0)
    );

    sumMUX2(0) <= s1(1);
    sumMUX2(1) <= s2(1);
    MUX2x1_2 : MUX2x1 port map (
        i => sumMUX2,
        s => cin,
        o => s(1)
    );

    sumMUX3(0) <= s1(2);
    sumMUX3(1) <= s2(2);
    MUX2x1_3 : MUX2x1 port map (

```

```

        i => sumMUX3,
        s => cin,
        o => s(2)
    );
    sumMUX4(0) <= s1(3);
    sumMUX4(1) <= s2(3);
    MUX2x1_4 : MUX2x1 port map (
        i => sumMUX4,
        s => cin,
        o => s(3)
    );
    -----
    FinalMux(0) <= cout1;
    FinalMux(1) <= cout2;
    Final : MUX2x1 port map (
        i => FinalMux,
        s => cin,
        o => cout
    );
end structure;

```

---

## : CarrySelectAdder4Bits مربوط به TIMING REPORT

The screenshot displays the Xilinx ISE Project Navigator interface. The left pane shows the project hierarchy with 'CarrySelectAdder4Bits - structure' selected. The middle pane lists various reports, with 'Timing Report' highlighted. The right pane displays the content of the 'Timing Report'.

**TIMING REPORT**

NOTE: THESE TIMING NUMBERS ARE ONLY A SYNTHESIS ESTIMATE.  
FOR ACCURATE TIMING INFORMATION PLEASE REFER TO THE TRACE REPORT  
GENERATED AFTER PLACE-and-ROUTE.

**Clock Information:**  
No clock signals found in this design

**Asynchronous Control Signals Information:**  
No asynchronous control signals found in this design

**Timing Summary:**  
Speed Grade: -5

Minimum period: No path found  
Minimum input arrival time before clock: No path found  
Maximum output required time after clock: No path found  
Maximum combinational path delay: 10.530ns

**Timing Detail:**  
All values displayed in nanoseconds (ns)

Timing constraint: Default path analysis  
Total number of paths / destination ports: 49 / 5

Delay: 10.530ns (Levels of Logic = 5)  
Source: cin (PAD)



: CSA4Bits\_tb

```
LIBRARY ieee;
```

```
USE ieee.std_logic_1164.ALL;
```

```
ENTITY CSA4Bits_tb IS
```

```
END CSA4Bits_tb;
```

```
ARCHITECTURE test OF CSA4Bits_tb IS
```

```
-- Component Declaration for the Unit Under Test (UUT)
```

```
COMPONENT CarrySelectAdder4Bits
```

```
PORT(
```

```
    a : IN  std_logic_vector(3 downto 0);
```

```
    b : IN  std_logic_vector(3 downto 0);
```

```
    cin : IN  std_logic;
```

```
    s : OUT  std_logic_vector(3 downto 0);
```

```
    cout : OUT  std_logic
```

```
);
```

```
END COMPONENT;
```

```
--Inputs
```

```
signal a : std_logic_vector(3 downto 0) ;
```

```
signal b : std_logic_vector(3 downto 0) ;
```

```
signal cin : std_logic ;
```

```
--Outputs
```

```
signal s : std_logic_vector(3 downto 0);
```

```
signal cout : std_logic;
```

```
BEGIN
```

```

-- Instantiate the Unit Under Test (UUT)

uut: CarrySelectAdder4Bits PORT MAP (

    a => a,

    b => b,

    cin => cin,

    s => s,

    cout => cout

);

    cin <= '0' , '1' after 100 ns , '0' after 200 ns ;

    a <= "1010" , "1100" after 100 ns , "0011" after 200 ns ;

    b <= "0101" , "0100" after 100 ns , "0010" after 200 ns ;

END;

```

خروجی شبیه سازی CSA4Bits\_tb :

