# گزارشکار آزمایش ۲. ۵

: ordinary_multiplier

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity ordinary_multiplier is
	port(
	A : in Std_logic_vector(3 downto 0);
	B : in Std_logic_vector(3 downto 0);
	M : out Std_logic_vector(7 downto 0)
	);
end ordinary_multiplier;
architecture structure of ordinary_multiplier is
component FullAdder is
	port(
		A : in STD_LOGIC;
		B : in STD_LOGIC;
		Cin : in STD_LOGIC;
		S : out STD_LOGIC;
		Cout : out STD_LOGIC
		);
end component FullAdder;
component HalfAdder is
	Port (
	A : in STD_LOGIC;
	B : in STD_LOGIC;
	S : out STD_LOGIC;
	Cout : out STD_LOGIC
	);
end component HalfAdder;
```

```
signal sAnd : STD_LOGIC_VECTOR(15 downto 1);
signal sC : STD_LOGIC_VECTOR(10 downto 0);
signal sS : STD_LOGIC_VECTOR(6 downto 1);
begin
	M(0) <= A(0) and B(0);
	sAnd(1) <= A(1) and B(0);
	sAnd(2) <= A(2) and B(0);
	sAnd(3) <= A(3) and B(0);
	sAnd(4) <= A(0) and B(1);
	sAnd(5) <= A(1) and B(1);
	sAnd(6) <= A(2) and B(1);
	sAnd(7) <= A(3) and B(1);
	sAnd(8) <= A(0) and B(2);
	sAnd(9) <= A(1) and B(2);
	sAnd(10) <= A(2) and B(2);
	sAnd(11) <= A(3) and B(2);
	sAnd(12) <= A(0) and B(3);
	sAnd(13) <= A(1) and B(3);
	sAnd(14) <= A(2) and B(3);
	sAnd(15) <= A(3) and B(3);

	HalfAdder1 : HalfAdder port map (
		A => sAnd(4),
		B => sAnd(1),
		S => M(1),
		Cout => sC(0)
	);

	FullAdder1 : FullAdder port map (
		A => sAnd(2),
```

```
            B => sAnd(5),
            Cin => sC(0),
            S => sS(1),
            Cout => sC(1)
        );

        FullAdder2 : FullAdder port map (
            A => sAnd(3),
            B => sAnd(6),
            Cin => sC(1),
            S => sS(2),
            Cout => sC(2)
        );

        HalfAdder2 : HalfAdder port map (
            A => sAnd(7),
            B => sC(2),
            S => sS(3),
            Cout => sC(3)
        );
----------

        HalfAdder3 : HalfAdder port map (
            A => sAnd(8),
            B => sS(1),
            S => M(2),
            Cout => sC(4)
        );

        FullAdder3 : FullAdder port map (
```

```
        A => sAnd(9),
        B => sS(2),
        Cin => sC(4),
        S => sS(4),
        Cout => sC(5)
);

FullAdder4 : FullAdder port map (
        A => sAnd(10),
        B => sS(3),
        Cin => sC(5),
        S => sS(5),
        Cout => sC(6)
);

FullAdder5 : FullAdder port map (
        A => sAnd(11),
        B => sC(3),
        Cin => sC(6),
        S => sS(6),
        Cout => sC(7)
);

-------------------------
HalfAdder4 : HalfAdder port map (
        A => sAnd(12),
        B => sS(4),
        S => M(3),
        Cout => sC(8)
);
```

```
	FullAdder6 : FullAdder port map (
		A =>sAnd(13),
		B => sS(5),
		Cin => sC(8),
		S => M(4),
		Cout => sC(9)
	);

	FullAdder7 : FullAdder port map (
		A => sAnd(14),
		B => sS(6),
		Cin => sC(9),
		S => M(5),
		Cout => sC(10)
	);

	FullAdder8 : FullAdder port map (
		A => sAnd(15),
		B => sC(7),
		Cin => sC(10),
		S => M(6),
		Cout => M(7)
	);
end structure;
```

---

: ordinary_multiplier_tb

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

```
ENTITY ordinary_multiplier_tb IS
END ordinary_multiplier_tb;

ARCHITECTURE test OF ordinary_multiplier_tb IS

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

   COMPONENT ordinary_multiplier
   PORT(
        A : IN  std_logic_vector(3 downto 0);
        B : IN  std_logic_vector(3 downto 0);
        M : OUT  std_logic_vector(7 downto 0)
       );
   END COMPONENT;
  --Inputs
  signal A : std_logic_vector(3 downto 0);
  signal B : std_logic_vector(3 downto 0);
          --Outputs
  signal M : std_logic_vector(7 downto 0);
BEGIN
          -- Instantiate the Unit Under Test (UUT)
  uut: ordinary_multiplier PORT MAP (
        A => A,
        B => B,
        M => M
       );
       A <= "0001" ,  "0101" after 100 ns ,  "0101" after 200 ns , "0110" after 300 ns;
       B <= "0010" ,  "0100" after 100 ns , "0111" after 200 ns , "0110" after 300 ns ;
END;
```

---

خروجی شبیه سازی ordinary_multiplier_tb :


: array_multiplier

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity array_multiplier is
Port ( A, B: in  STD_LOGIC_VECTOR(3 downto 0);
                        RES : out  STD_LOGIC_VECTOR(7 downto 0));
end array_multiplier;

architecture structure of array_multiplier is
component 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 component CarryLookAheadAdder4Bits;
```

```
signal Level0, Level1, Level2, p0, p1, p2, p3, p4 ,p5: STD_LOGIC_VECTOR(3 downto 0);
        signal cout1, cout0: STD_LOGIC;
begin
p0(0) <= A(0) and B(1) ;
 p0(1) <= A(0) and B(2) ;
 p0(2) <= A(0) and B(3) ;
 p0(3) <= '0';
 p1(0) <= A(1) and B(0) ;
 p1(1) <= A(1) and B(1) ;
 p1(2) <= A(1) and B(2) ;
 p1(3) <= A(1) and B(3) ;
 p2(0) <= A(2) and B(0) ;
 p2(1) <= A(2) and B(1) ;
 p2(2) <= A(2) and B(2) ;
 p2(3) <= A(2) and B(3) ;
 p3(0) <= A(3) and B(0) ;
 p3(1) <= A(3) and B(1) ;
 p3(2) <= A(3) and B(2) ;
 p3(3) <= A(3) and B(3) ;

p4(2 downto 0) <= Level0(3 downto 1);
 p4(3) <= cout0;
 p5(2 downto 0) <= Level1(3 downto 1);
 p5(3) <= cout1;

CLA0: CarryLookAheadAdder4Bits port map(a => p0, b => p1, cin =>'0',s => Level0, cout =>
cout0);
CLA1: CarryLookAheadAdder4Bits port map(a => p2, b => p4, cin =>'0',s => Level1, cout =>
cout1);
CLA2: CarryLookAheadAdder4Bits port map(a => p3, b=>  p5, cin =>'0',s => Level2, cout =>
RES(7));
```

```
        RES(0) <= A(0) and B(0);
        RES(1) <= Level0(0);
        RES(2) <= Level1(0);
  RES(6 downto 3) <= Level2(3 downto 0);
end structure;
```

---

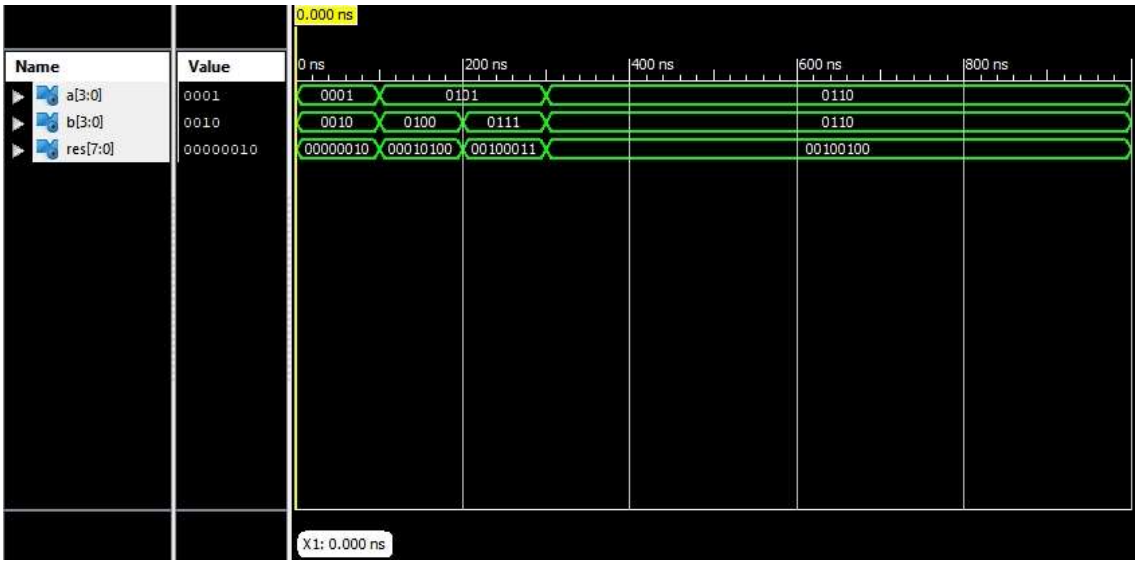
: array_multiplier_tb

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
ENTITY array_multiplier_tb IS
END array_multiplier_tb;
ARCHITECTURE test OF array_multiplier_tb IS
   -- Component Declaration for the Unit Under Test (UUT)
   COMPONENT array_multiplier
   PORT(
       A : IN  std_logic_vector(3 downto 0);
       B : IN  std_logic_vector(3 downto 0);
       RES : OUT  std_logic_vector(7 downto 0)
      );
   END COMPONENT;
  --Inputs
  signal A : std_logic_vector(3 downto 0) := (others => '0');
  signal B : std_logic_vector(3 downto 0) := (others => '0');
        --Outputs
  signal RES : std_logic_vector(7 downto 0);
BEGIN
        -- Instantiate the Unit Under Test (UUT)
  uut: array_multiplier PORT MAP (
       A => A,
```

```
        B => B,
        RES => RES
      );
        A <= "0001" ,  "0101" after 100 ns ,  "0101" after 200 ns , "0110" after 300 ns;
        B <= "0010" ,  "0100" after 100 ns , "0111" after 200 ns , "0110" after 300 ns ;
END;
```

---

خروجی شبیه سازی array_multiplier_tb :




: carry_save_multiplier

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity carry_save_multiplier is
Port ( A, B: in  STD_LOGIC_VECTOR(3 downto 0);
      RES : out  STD_LOGIC_VECTOR(7 downto 0));
end carry_save_multiplier;

architecture structure of carry_save_multiplier is
component FullAdder is
```

```
        port(
                A : in STD_LOGIC;
                B : in STD_LOGIC;
                Cin : in STD_LOGIC;
                S : out STD_LOGIC;
                Cout : out STD_LOGIC
                );
end component FullAdder;
component HalfAdder is
        Port (
        A : in STD_LOGIC;
        B : in STD_LOGIC;
        S : out STD_LOGIC;
        Cout : out STD_LOGIC
        );
end component HalfAdder;
  signal Level0, Level1, Level2: STD_LOGIC_VECTOR(4 downto 0);
        signal andd: STD_LOGIC_VECTOR(15 downto 0);
        signal Level3: STD_LOGIC_VECTOR(1 downto 0);
begin
        -- create all and gates
        -- layer 1
        andd(0) <= A(0) and B(0);
        andd(1) <= A(0) and B(1);
        andd(2) <= A(0) and B(2);
        andd(3) <= A(0) and B(3);
        -- layer 2
        andd(4) <= A(1) and B(0);
        andd(5) <= A(1) and B(1);
        andd(6) <= A(1) and B(2);
```

```
        andd(7) <= A(1) and B(3);
        -- layer 3
        andd(8) <= A(2) and B(0);
        andd(9) <= A(2) and B(1);
        andd(10) <= A(2) and B(2);
        andd(11) <= A(2) and B(3);
        -- layer 4
        andd(12) <= A(3) and B(0);
        andd(13) <= A(3) and B(1);
        andd(14) <= A(3) and B(2);
        andd(15) <= A(3) and B(3);

        -- calcute level 0 :
        RES(0) <= andd(0);

        -- calculate level 1 :
half_adder_instance0: halfAdder port map(A =>andd(1), B => andd(4),  Cout => Level0(0), S
=> RES(1));
half_adder_instance1: halfAdder port map(A => andd(2), B => andd(5), Cout => Level0(2), S
=> Level0(1));
half_adder_instance2: halfAdder port map(A => andd(3), B => andd(6), Cout => Level0(4), S
=> Level0(3));

        -- calculate level 2 :
full_adder_instance0: fullAdder port map(A => andd(8), B => Level0(0), Cin => Level0(1), S =>
RES(2), Cout => Level1(0));
full_adder_instance1: fullAdder port map(A => andd(9), B => Level0(2), Cin => Level0(3), S =>
Level1(1), Cout => Level1(2));
full_adder_instance3: fullAdder port map(A => andd(10),B => andd(7), Cin => Level0(4), S =>
Level1(3), Cout => Level1(4));

        -- calculate level 3 :
```

```
full_adder_instance4: fullAdder port map(A => andd(12), B => Level1(0), Cin => Level1(1), S
=> RES(3),    Cout => Level2(0));

full_adder_instance5: fullAdder port map(A => andd(13), B => Level1(2), Cin => Level1(3), S
=> Level2(1), Cout => Level2(2));

full_adder_instance6: fullAdder port map(A => andd(14), B => andd(11) , Cin => Level1(4), S
=> Level2(3), Cout => Level2(4));


        -- calculate final level :

half_adder_instance3: halfAdder port map(A => Level2(0), B => Level2(1), Cout => Level3(0),
S => RES(4));

full_adder_instance7: fullAdder port map(A => Level2(2), B => Level2(3), Cin => Level3(0), S
=> RES(5), Cout => Level3(1));

full_adder_instance8: fullAdder port map(A => andd(15), B => Level2(4), Cin => Level3(1), S
=> RES(6), Cout => RES(7));

end structure;
```

---

## : carry_save_multiplier_tb

```
LIBRARY ieee;

USE ieee.std_logic_1164.ALL;

ENTITY carry_save_multiplier_tb IS

END carry_save_multiplier_tb;

ARCHITECTURE test OF carry_save_multiplier_tb IS
   -- Component Declaration for the Unit Under Test (UUT)


   COMPONENT carry_save_multiplier
   PORT(
       A : IN  std_logic_vector(3 downto 0);
       B : IN  std_logic_vector(3 downto 0);
       RES : OUT  std_logic_vector(7 downto 0)
      );
   END COMPONENT;
```

```
  --Inputs
  signal A : std_logic_vector(3 downto 0);
  signal B : std_logic_vector(3 downto 0);

        --Outputs
  signal RES : std_logic_vector(7 downto 0);

BEGIN
        -- Instantiate the Unit Under Test (UUT)
  uut: carry_save_multiplier PORT MAP (
       A => A,
       B => B,
       RES => RES
      );
       A <= "0001" ,  "0101" after 100 ns ,  "0101" after 200 ns , "0110" after 300 ns;
       B <= "0010" ,  "0100" after 100 ns , "0111" after 200 ns , "0110" after 300 ns ;

END;
```

---

خروجی شبیه سازی carry_save_multiplier_tb :