

"به نام یزدان پاک"

گزارش کار آزمایش تفریق کننده

اعضای گروه:

کیانا آقا کثیری 9831006

محمد چوپان 9831125

سارا تاجرنیا 9831016

نویسنده گزارش : کیانا آقا کثیری

تاریخ آزمایش : 1400/02/07

تاریخ تحویل گزارش : 1400/02/14

شرح آزمایش:

۱. یک نیم تفریق گر را در سطح گیت پیاده سازی کنید.
۲. یک تمام تفریق گر را با استفاده از ماژول نیم تفریق گر که در بخش ۱ پیاده سازی شده است، پیاده سازی کنید.
(کل طرح در سطح گیت پیاده سازی نشود از ماژول بخش ۱ استفاده کنید).
۳. با استفاده از یک جمع کننده ۴ بیتی یک جمع کننده/تفریق کننده در سطح ماژول بسازید.
(از $A-B = A+B'+1$ استفاده کنید).

شکل 1: شرح آزمایش

HS:

x	y	(تفریق ریاضی)	
		borrow	sub
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

Diagram of a Half Subtractor (HS) block. Inputs: x, y. Outputs: borrow = $x'.y$, sub = $x \oplus y$.

$\text{delay (sub)} = d$
 $\text{delay (borrow)} = 2d$
 $\text{Cost} = 3g$

شکل 2: طراحی HS

Behavior of HS:

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

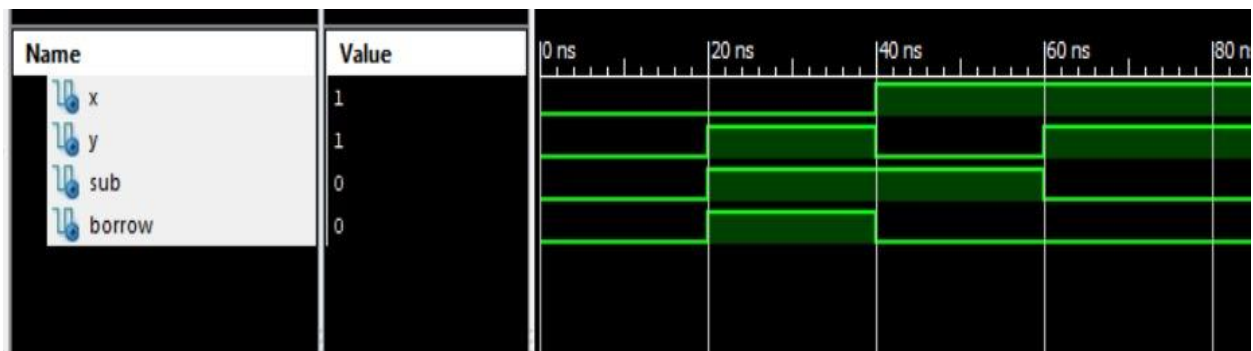
entity HS is
    Port ( x : in  STD_LOGIC;
          y : in  STD_LOGIC;
          sub : out STD_LOGIC;
          borrow : out STD_LOGIC);
end HS;

architecture Behavioral of HS is

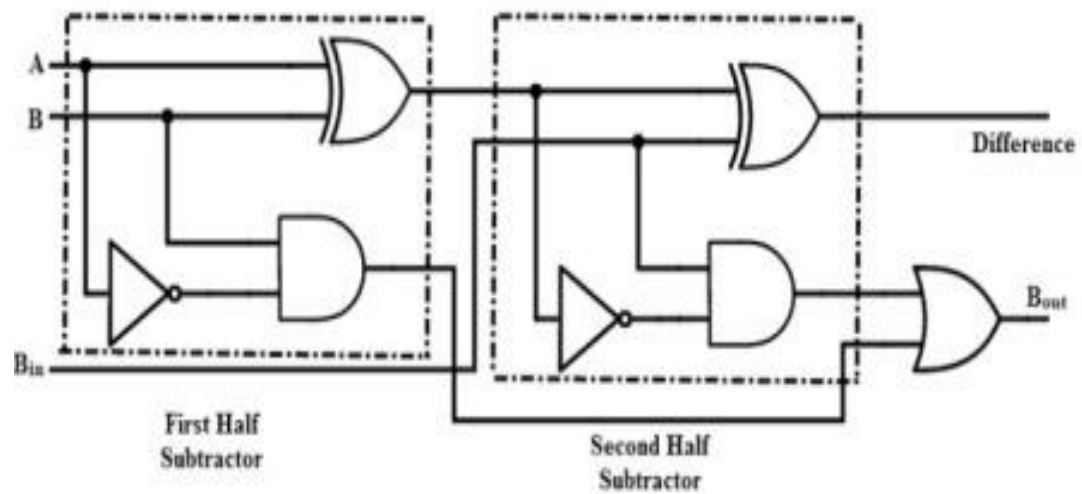
begin
    sub<=x xor y;
    borrow<=(not x) and y ;
end Behavioral;
```

شکل 3: توصیف HS

Result of Simulation in isim :



شکل 4: نتایج شبیه سازی



شکل 5: طراحی FS

Behavior of FS:

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity FSub is
    Port ( x : in  STD_LOGIC;
          y : in  STD_LOGIC;
          z : in  STD_LOGIC;
          sub : out STD_LOGIC;
          borrow : out STD_LOGIC);
end FSub;

architecture Behavioral of FSub is
    component HS is
        Port ( x : in  STD_LOGIC;
              y : in  STD_LOGIC;
              sub : out STD_LOGIC;
              borrow : out STD_LOGIC);
    end component;

    signal temp_sub:std_logic;
    signal temp_borrow:std_logic;
    signal temp_borrow1:std_logic;

begin
    HS_1 : HS port map (x=>x,y=>y,sub=>temp_sub,borrow=>temp_borrow);
    HS_2 : HS port map (x=>temp_sub,y=>z,sub=>sub,borrow=>temp_borrow1);
    borrow<=temp_borrow1 or temp_borrow;

```

شکل 6: توصیف FS

Result of simulation in isim:



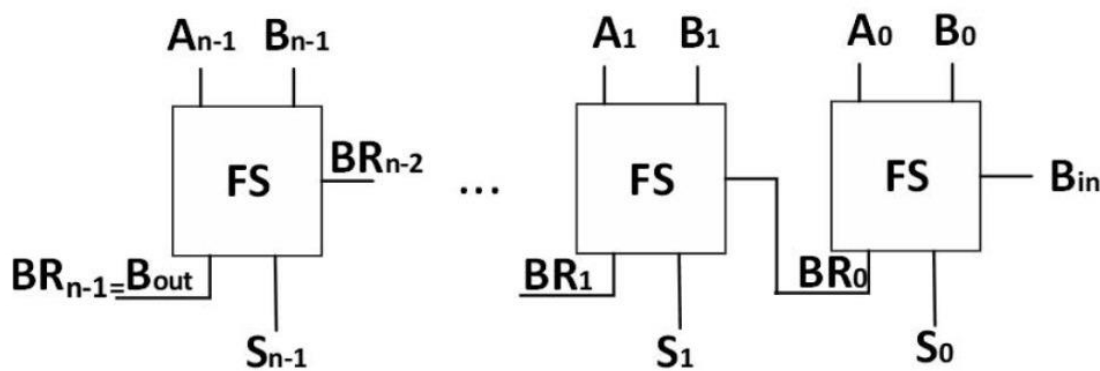
شکل 7: نتایج شبیه سازی

توضیح اول آزمایش:

در این آزمایش هدف پیاده سازی نیم تفریق گر، تمام تفریق گر و تفریق گر 4 بیتی می باشد که با طراحی آن توسط گیت های پایه انجام می شود.

ما در ابتدا یک نیم تفریق گر را با استفاده از گیت های پایه XOR و NOT و AND طراحی کرده ایم . سپس یک تمام تفریق گر را با استفاده از component نیم تفریق گر و اتصالات موجود در تصویر در تصویر port map کرده و یک تمام تفریق گر را شبیه سازی میکنیم.

در نهایت نیز با تعریف component تمام تفریق گر یک تفریق گر 4 بیتی ابشاری را مانند تصویر شبیه سازی میکنیم.



شکل 8: تصویر تفریق کننده ساخته شده با FS

Behavior of ripple_sub:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity ripple_sub is
    Port ( A : in  STD_LOGIC_vector (3 downto 0);
          B : in  STD_LOGIC_vector (3 downto 0);
          Bin : in  STD_LOGIC;
          sub : out STD_LOGIC_vector (3 downto 0);
          Bout : out STD_LOGIC);
end ripple_sub;

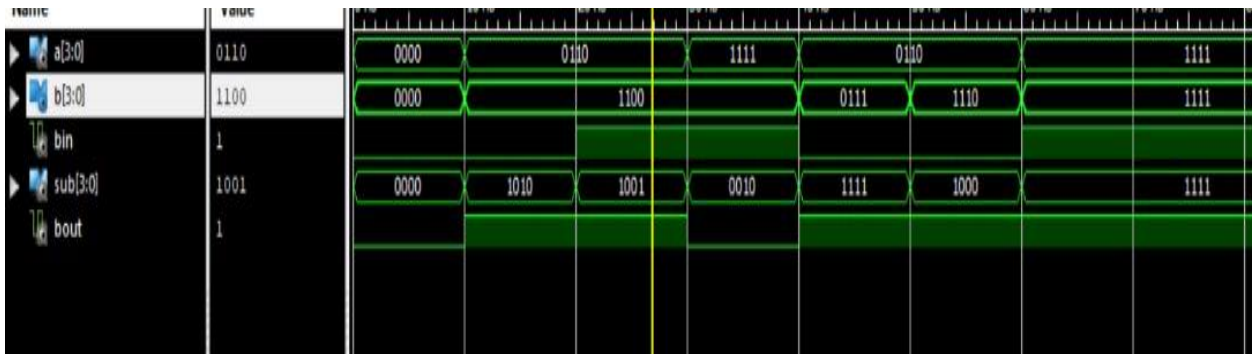
architecture Behavioral of ripple_sub is
    component FSub is
        Port ( x : in  STD_LOGIC;
              y : in  STD_LOGIC;
              z : in  STD_LOGIC;
              sub : out STD_LOGIC;
              borrow : out STD_LOGIC);
    end component ;
    signal temp_borrow:std_logic_vector (2 downto 0);

begin
    Full_subtractor_1: FSub port map (x=>A(0),y=>B(0),z=>Bin,sub=>sub(0),borrow=>temp_borrow(0));
    Full_subtractor_2: FSub port map (x=>A(1),y=>B(1),z=>temp_borrow(0),sub=>sub(1),borrow=>temp_borrow(1));
    Full_subtractor_3: FSub port map (x=>A(2),y=>B(2),z=>temp_borrow(1),sub=>sub(2),borrow=>temp_borrow(2));
    Full_subtractor_4: FSub port map (x=>A(3),y=>B(3),z=>temp_borrow(2),sub=>sub(3),borrow=>Bout);

end Behavioral;
```

شکل 9: توصیف تفریق کننده ساخته شده با FS

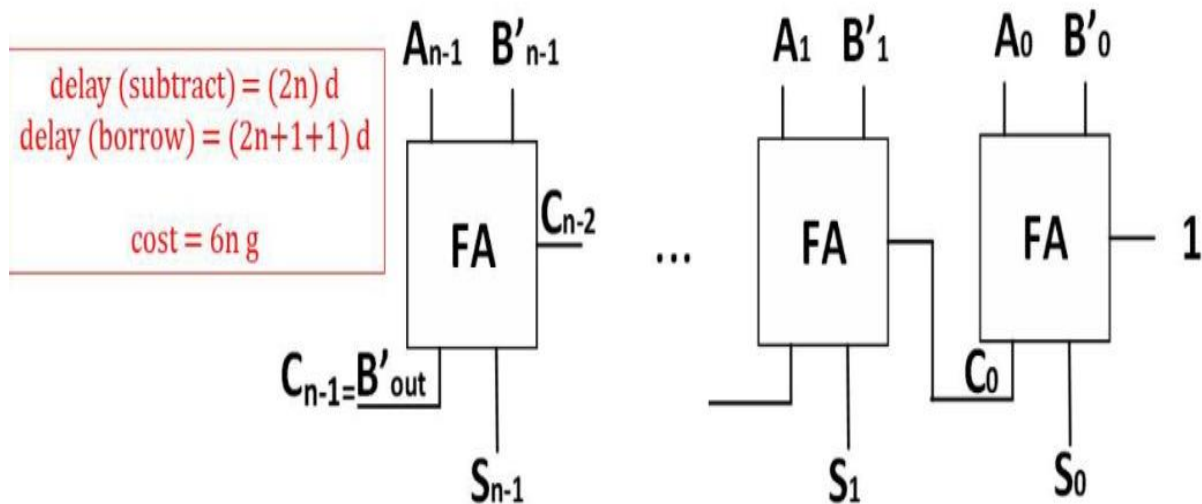
Result of simulation in isim:



شکل 10: نتایج شبیه سازی

در مرحله بعدی نیز یک تفریق گر 4 بیتی را با استفاده از جمع کننده 4 بیتی به روش زیر شبیه سازی میکنیم که برای استفاده از این روش میدانیم:

$$A-B=A+B'+1$$



شکل 11: توصیف تفریق کننده ساخته شده با جمع کننده

Behavior of ripple_adder:

```

1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3
4  entity Ripple_Adder is
5  Port (  A : in STD_LOGIC_VECTOR (3 downto 0);
6        B : in STD_LOGIC_VECTOR (3 downto 0);
7        Sub : out STD_LOGIC_VECTOR (3 downto 0);
8        Bout : out STD_LOGIC);
9  end Ripple_Adder;
10
11  architecture Behavioral of Ripple_Adder is
12
13
14  component FA
15  Port (  A : in STD_LOGIC;
16        B : in STD_LOGIC;
17        Cin : in STD_LOGIC;
18        S : out STD_LOGIC;
19        Cout : out STD_LOGIC);
20  end component;
21
22  signal c1,c2,c3,c4: STD_LOGIC;
23
24  begin
25
26  -- Port Mapping Full Adder 4 times
27  FA1: FA port map( A=>A(0), B=>B(0),Cin=> '1',S=> Sub(0),Cout=> c1);
28  FA2: FA port map( A=>A(1), B=>B(1), Cin=>c1, S=> Sub(1),Cout=> c2);
29  FA3: FA port map( A=>A(2), B=>B(2), Cin=>c2, S=> Sub(2), Cout=>c3);
30  FA4: FA port map( A=>A(3), B=>B(3), Cin=>c3, S=> Sub(3), Cout=>c4);
31  Bout<=not c4;
32
33  end Behavioral;
34
35

```

شکل 12: توصیف تفریق کننده ساخته شده با جمع کننده

Result of simulation in isim:



شکل 13: نتایج شبیه سازی

نتیجه گیری:

هدف نهایی آزمایش : در این آزمایش با نحوه عملکرد تفریق کننده ها نحوه عملکرد آن ها و انواع آن ها را بررسی کرده و شبیه سازی کردیم.

اینکار با شبیه سازی ماژول های نیم تفریق گر تمام تفریق گر و تفریق کننده های 4 بیتی آبشاری و تفریق کننده 4 بیتی آبشاری به وسیله جمع کننده انجام شده است.