

## "گزارشکار آزمایش ۱"

### عنوان VHDL مقدماتی

**مقدمه** توصیف مدارهای پایه آزمایشات و تعریف برخی قطعات.

**هدف** استفاده از قطعات تعریف شده برای ساخت نیم جمع کننده و در نهایت ساخت جمع کننده با استفاده از دو نیم جمع کننده.

شرح آزمایش

الف) OR چهار ورودی:

اولین قدم برای تعریف OR چهار ورودی تعریف OR دو ورودی است:

```
32 entity or_gate is
33     Port ( A_or : in  STD_LOGIC;
34           B_or : in  STD_LOGIC;
35           out_or : out STD_LOGIC);
36 end or_gate;
37
38 architecture Behavioral of or_gate is
39
40 begin
41     out_or <= A_or or B_or;
42 end Behavioral;
```

به صورت مقابل گیت OR

دو ورودی را تعریف میکنیم و

تست بنچ آن ها هم تعریف

میکنیم:

```
81 -- Stimulus process
82 stim_proc: process
83 begin
84     -- hold reset state for 100 ns.
85     -- wait for 100 ns;
86
87     -- wait for <clock>_period*10;
88
89     -- insert stimulus here
90     A_or <='0','1' after 50ns;
91     B_or <='0','1' after 100ns;
92     wait;
93 end process;
94
95 END;
```

حال با استفاده از سه گیت OR دو ورودی گیت OR چهار ورودی را تعریف میکنیم:

```

32 entity or_4_gate is
33     Port ( A : in  STD_LOGIC;
34           B : in  STD_LOGIC;
35           C : in  STD_LOGIC;
36           D : in  STD_LOGIC;
37           OUT_result : out  STD_LOGIC);
38 end or_4_gate;
39
40 architecture Behavioral of or_4_gate is
41 component or_gate is
42     Port ( A_or : in  STD_LOGIC;
43           B_or : in  STD_LOGIC;
44           out_or : out  STD_LOGIC);
45 end component;
46 signal w1,w2: std_logic;
47
48 begin
49 instance0 : or_gate port map(A_or=>A, B_or=>B,out_or=>w1);
50 instance1 : or_gate port map(A_or=>C, B_or=>D,out_or=>w2);
51 instance2 : or_gate port map(A_or=>w1, B_or=>w2,out_or=>OUT_result);
52
53 end Behavioral;

```

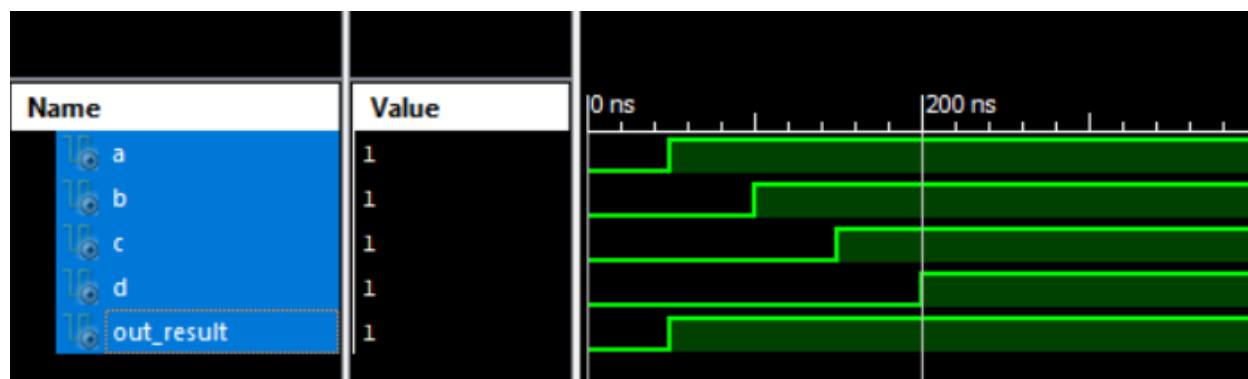
تست بنچ آن هم به این صورت تعریف میشود:

```

89     begin
90         -- hold reset state for 100 ns.
91         wait for 100 ns;
92         --
93         wait for <clock>_period*10;
94         --
95         -- insert stimulus here
96         A <='0','1' after 50ns;
97         B <='0','1' after 100ns;
98         C <='0','1' after 150ns;
99         D <='0','1' after 200ns;
100         wait;
101     end process;
102
103 END;

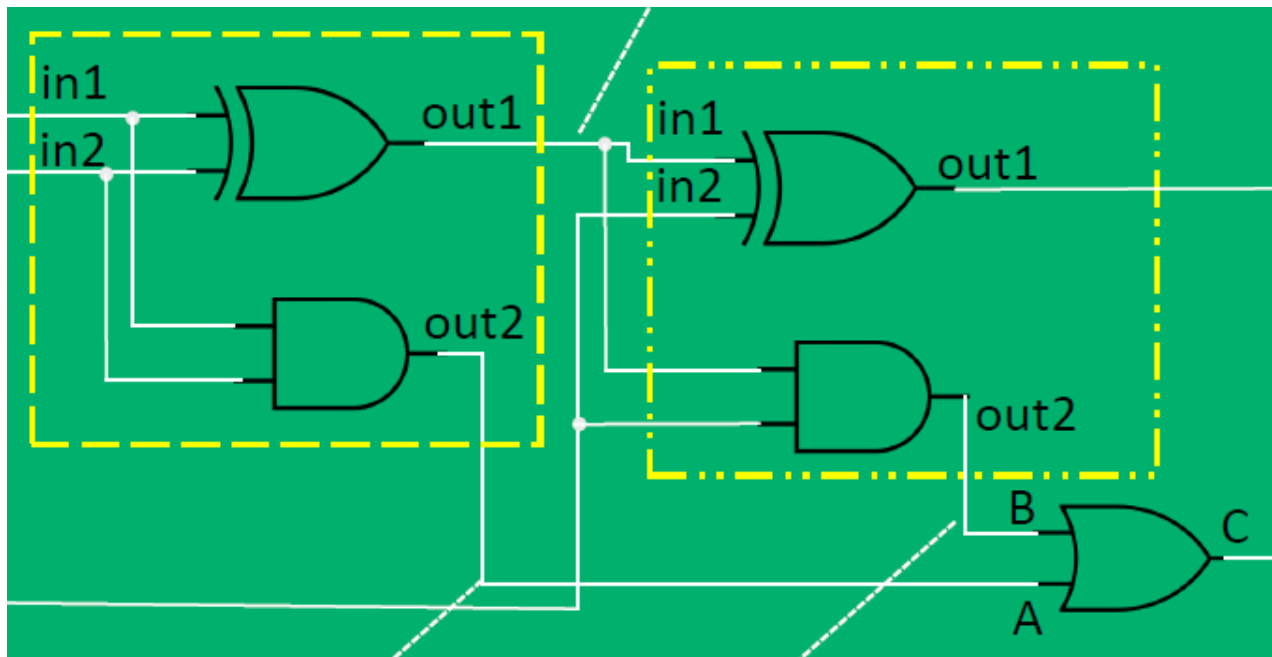
```

همانطور که مشاهده میشود گیت OR چهار ورودی تعریف شده بصورت زیر عمل میکند:



### ب) جمع کننده :

برای تعریف جمع کننده باید ابتدا گیت های AND, XOR دو ورودی را تعریف کنیم، سپس با استفاده از آنها یک نیم جمع کننده تعریف کرده و با استفاده از دو نیم جمع کننده و یک گیت OR تعریف جمع کننده را تکمیل کنیم.



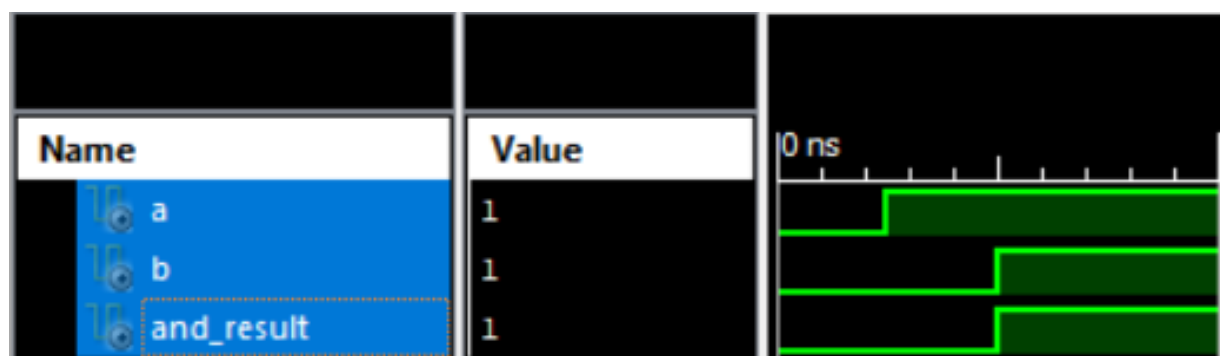
## ۱- تعریف گیت AND دو ورودی:

```
32 entity and_gate is
33 port(
34   A : in std_logic;
35   B : in std_logic;
36   and_result : out std_logic
37 );
38 end and_gate;
39
40 architecture Behavioral of and_gate is
41 begin
42   and_result <= A and B;
43 end Behavioral;
```

تست بنچ:

```
83 begin
84   -- hold reset state for 100 ns.
85   -- wait for 100 ns;
86
87   -- wait for <clock>_period*10;
88
89   -- insert stimulus here
90   A <='0','1' after 50ns;
91   B <='0','1' after 100ns;
92   wait;
93 end process;
94
95 END;
```

عملکرد:



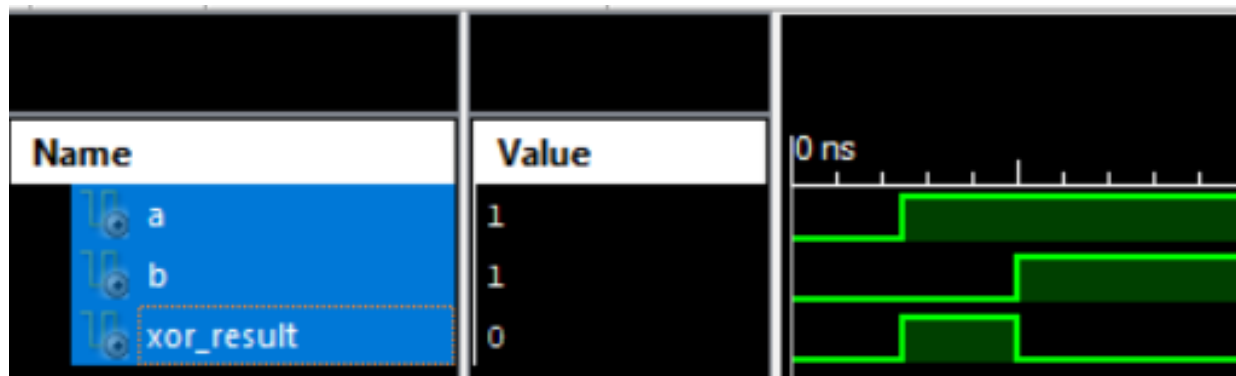
## ۲- تعریف گیت XOR دو ورودی:

```
32 entity xor_gate is
33 port (
34   A : in std_logic;
35   B : in std_logic;
36   xor_result : out std_logic
37 );
38 end xor_gate;
39
40 architecture Behavioral of xor_gate is
41 begin
42   xor_result <= A xor B;
43 end Behavioral;
```

تست بنچ:

```
83   begin
84       -- hold reset state for 100 ns.
85       --   wait for 100 ns;
86
87       --   wait for <clock>_period*10;
88
89       -- insert stimulus here
90   A <='0','1' after 50ns;
91   B <='0','1' after 100ns;
92       wait;
93   end process;
94
95 END;
```

عملکرد:



۳- نیم جمع کننده :

```

32 entity HA is
33     Port ( in1 : in  STD_LOGIC;
34           in2 : in  STD_LOGIC;
35           out1 : out STD_LOGIC;
36           out2 : out STD_LOGIC);
37 end HA;
38
39 architecture Behavioral of HA is
40     component and_gate is
41     port(
42     A : in std_logic;
43     B : in std_logic;
44     and_result : out std_logic
45     );
46 end component;
47     component xor_gate is
48     port(
49     A : in std_logic;
50     B : in std_logic;
51     xor_result : out std_logic
52     );
53 end component;
54     signal w1,w2: STD_LOGIC;
55 begin
56     instanceAnd : and_gate port map(A=>in1,B=>in2,and_result=>out2);
57     instanceXor : xor_gate port map(A=>in1,B=>in2,xor_result=>out1);
58
59 end Behavioral;

```

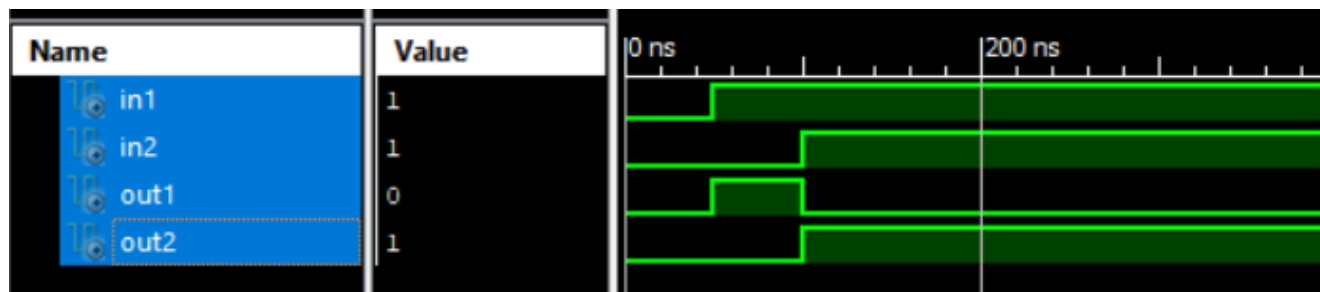
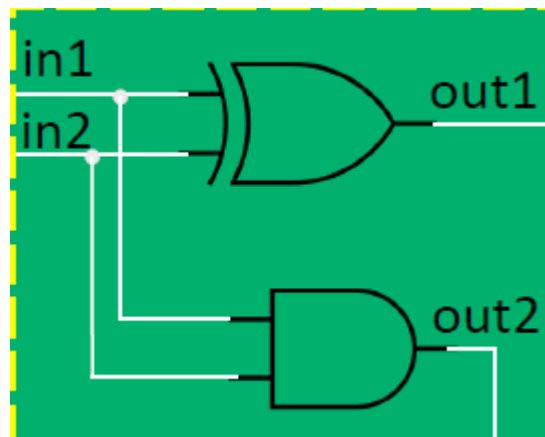
```

86     begin
87         -- hold reset state for 100 ns.
88         -- wait for 100 ns;
89         --
90         -- wait for <clock>_period*10;
91
92         -- insert stimulus here
93         in1 <='0','1' after 50ns;
94         in2 <='0','1' after 100ns;
95         wait;
96     end process;
97
98 END;

```

تست بنچ:

عملکرد:



ثمین مهدی پور

۹۸۳۹۰۳۹

۴- جمع کننده:

```
32 entity FA is
33     Port ( I1 : in  STD_LOGIC;
34           I2 : in  STD_LOGIC;
35           Cin : in  STD_LOGIC;
36           S : out  STD_LOGIC;
37           Cout : out STD_LOGIC);
38 end FA;
```

```
40 architecture Behavioral of FA is
41 component HA is
```

```
42     Port ( in1 : in  STD_LOGIC;
43           in2 : in  STD_LOGIC;
44           out1 : out STD_LOGIC;
45           out2 : out STD_LOGIC);
46 end component;
```

```
47 component or_gate is
48     Port ( A_or : in  STD_LOGIC;
49           B_or : in  STD_LOGIC;
50           out_or : out STD_LOGIC);
51 end component;
```

```
52 signal Internal_signal0, Internal_signal1, Internal_signal2: STD_LOGIC;
```

```
53 begin
```

```
54 instanceHA1 : HA port map(in1=>I1,in2=>I2,out1=>Internal_signal0,out2=>Internal_signal1);
```

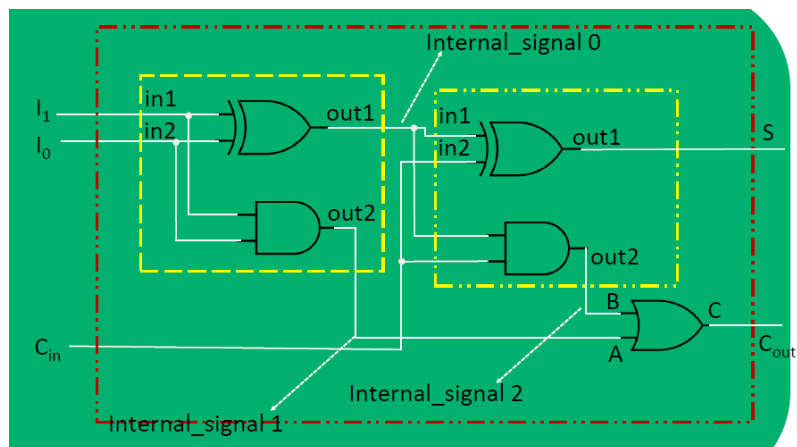
```
55 instanceHA2 : HA port map(in1=>Internal_signal0,in2=>Cin,out1=>S,out2=>Internal_signal2);
```

```
56 instandeOr : or_gate port map(A_or=>Internal_signal1, B_or=>Internal_signal2,out_or=>Cout);
```

```
57
```

```
58
```

```
59 end Behavioral;
```



تست بنچ:

```
89 begin
90     -- hold reset state for 100 ns.
91     -- wait for 100 ns;
92     --
93     -- wait for <clock>_period*10;
94     --
95     -- insert stimulus here
96     Cin<='0' ;
97     I1<='0','1' after 50ns;
98     I2<='0','1' after 100ns;
99     wait;
100     end process;
101
102 END;
```

عملکرد:

