

## گزارش تمرین عملی دوم

مهدی حق‌وردی

۲۰ اردیبهشت ۱۴۰۲

# فهرست مطالب

۱	ماژول‌های ساختار جمع	۱
۱	۱.۱ ماژول Half Adder	۱.۱
۱	۱.۱.۱ کد این ماژول	۱.۱.۱
۱	۲.۱.۱ تست‌بنچ ۱ این ماژول	۲.۱.۱
۲	۳.۱.۱ تست‌بنچ ۲ این ماژول	۳.۱.۱
۴	۴.۱.۱ خروجی simulation	۴.۱.۱
۵	۲.۱ ماژول Full Adder	۲.۱
۵	۱.۲.۱ کد این ماژول	۱.۲.۱
۶	۲.۲.۱ تست‌بنچ ۱ این ماژول	۲.۲.۱
۷	۳.۲.۱ تست‌بنچ ۲ این ماژول	۳.۲.۱
۱۰	۴.۲.۱ خروجی simulation	۴.۲.۱
۱۲	۳.۱ Four Bit Full Adder	۳.۱
۱۲	۱.۳.۱ کد این ماژول	۱.۳.۱
۱۳	۲.۳.۱ تست‌بنچ این ماژول	۲.۳.۱
۱۴	۳.۳.۱ خروجی simulation	۳.۳.۱
۱۵	۴.۱ طراحی‌ها	۴.۱
۱۷	ماژول‌های ساختار تفریق	۲
۱۷	۱.۲ ماژول Half Subber	۱.۲
۱۷	۱.۱.۲ کد این ماژول	۱.۱.۲
۱۷	۲.۱.۲ تست‌بنچ ۱ ماژول	۲.۱.۲
۱۸	۳.۱.۲ تست‌بنچ ۲ ماژول	۳.۱.۲
۲۰	۴.۱.۲ خروجی simulation	۴.۱.۲
۲۱	۲.۲ ماژول Full Subber	۲.۲
۲۱	۱.۲.۲ کد این ماژول	۱.۲.۲
۲۳	۲.۲.۲ تست‌بنچ ماژول	۲.۲.۲
۲۴	۳.۲.۲ خروجی simulation	۳.۲.۲
۲۴	۳.۲ ماژول Four Bit Full Subber	۳.۲
۲۴	۱.۳.۲ کد این ماژول	۱.۳.۲
۲۶	۲.۳.۲ تست‌بنچ ماژول	۲.۳.۲
۲۸	۳.۳.۲ خروجی simulation	۳.۳.۲
۲۹	۴.۲ طراحی‌ها	۴.۲

۳۰	۳	ماژول‌های ساختار ضرب
۳۰	۱.۳	Four Bit Multiplier ماژول
۳۰	۱.۱.۳	کد این ماژول
۳۲	۲.۱.۳	تست‌بنچ این ماژول
۳۳	۳.۱.۳	خروجی simulation
۳۴	۲.۳	طراحی‌ها
۳۵	۴	ماژول مالتی پلکسر
۳۵	۱.۴	mux4x1 ماژول
۳۵	۱.۱.۴	کد این ماژول
۳۵	۲.۱.۴	تست‌بنچ این ماژول
۳۷	۳.۱.۴	خروجی simulation
۳۸	۵	ALU ماژول
۳۸	۱.۵	ALU ماژول
۳۸	۱.۱.۵	کد این ماژول
۴۰	۲.۱.۵	تست‌بنچ این ماژول
۴۲	۳.۱.۵	خروجی simulation
۴۳	۲.۵	طراحی‌ها

## فصل ۱

# ماژول‌های ساختار جمع

برای پیاده‌سازی عملیت جمع در این ALU از یک Half adder یک بیتی، یک Full adder یک بیتی استفاده شده که در کنار هم یک Four bit full adder را تشکیل می‌دهند.

### ۱.۱ ماژول Half Adder

#### ۱.۱.۱ کد این ماژول

---

```
1 library ieee;
2 use ieee.std_logic_1164.all;
3
4 entity half_adder is
5     port (
6         a, b : in std_logic;
7         sum, carry : out std_logic
8     );
9 end half_adder;
10
11 architecture arch of half_adder is
12 begin
13     sum <= a xor b;
14     carry <= a and b;
15 end arch;
```

---

[https://github.com/mahdihaghverdi/archproject/blob/main/ALU/src/half\\_adder/half\\_adder.vhdl](https://github.com/mahdihaghverdi/archproject/blob/main/ALU/src/half_adder/half_adder.vhdl)

#### ۲.۱.۱ تست‌بنچ ۱ این ماژول

---

```
1 library ieee;
```

```

2 use ieee.std_logic_1164.all;
3
4
5 entity half_adder_tb is
6 end half_adder_tb;
7
8 architecture tb of half_adder_tb is
9     signal a, b : std_logic; -- inputs
10    signal sum, carry : std_logic; -- outputs
11 begin
12    -- connecting testbench signals with half_adder.vhd
13    UUT : entity work.half_adder port map (
14        a => a,
15        b => b,
16        sum => sum,
17        carry => carry
18    );
19
20    -- inputs
21    -- ba
22    -- 00 at 0 ns
23    -- 01 at 20 ns, as b is 0 at 20 ns and a is changed to 1
24    --    at 20 ns
25    -- 10 at 40 ns
26    -- 11 at 60 ns
27    -- 11 at 80 ns
28    a <= '0', '1' after 20 ns, '0' after 40 ns, '1' after 60
29    ns, '1' after 80 ns;
30    b <= '0', '1' after 40 ns, '1' after 60 ns;
31 end tb ;

```

---

[https://github.com/mahdihaghverdi/archproject/blob/main/ALU/src/half\\_adder/half\\_adder\\_tb.vhdl](https://github.com/mahdihaghverdi/archproject/blob/main/ALU/src/half_adder/half_adder_tb.vhdl)

### ۳.۱.۱ تست‌بنچ ۲ این ماژول

---

```

1 -- half_adder_process_tb.vhd
2
3 library ieee;
4 use ieee.std_logic_1164.all;
5
6 entity half_adder_process_tb is
7 end half_adder_process_tb;

```

```

8
9 architecture tb of half_adder_process_tb is
10     signal a, b : std_logic;
11     signal sum, carry : std_logic;
12 begin
13     -- connecting testbench signals with half_adder.vhd
14     UUT : entity work.half_adder port map (
15         a => a,
16         b => b,
17         sum => sum,
18         carry => carry
19     );
20
21     tb1 : process
22         constant period: time := 20 ns;
23         begin
24             a <= '0';
25             b <= '0';
26             wait for period;
27             assert ((sum = '0') and (carry = '0')) -- expected
                output
28             -- error will be reported if sum or carry is not 0
29             report "test failed for input combination 00"
                severity error;
30
31             a <= '0';
32             b <= '1';
33             wait for period;
34             assert ((sum = '1') and (carry = '0'))
35             report "test failed for input combination 01"
                severity error;
36
37             a <= '1';
38             b <= '0';
39             wait for period;
40             assert ((sum = '1') and (carry = '0'))
41             report "test failed for input combination 10"
                severity error;
42
43             a <= '1';
44             b <= '1';
45             wait for period;
46             assert ((sum = '0') and (carry = '1'))

```

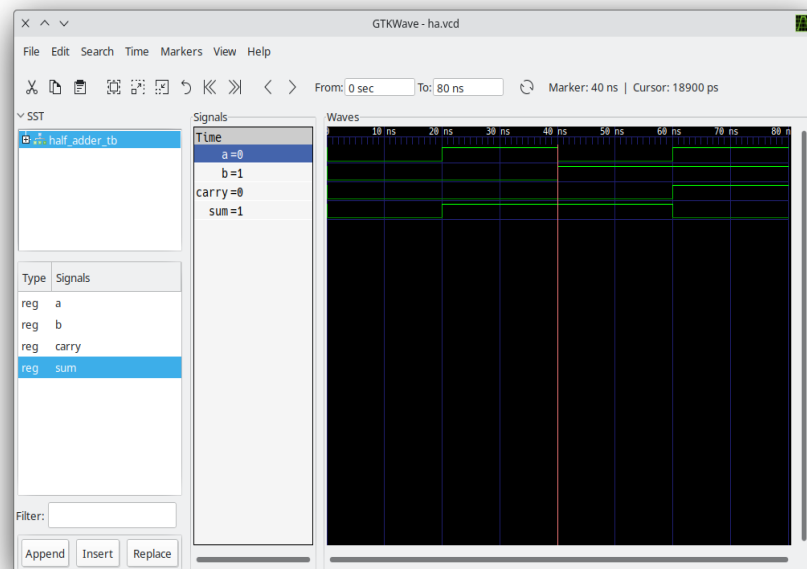
```

47         report "test failed for input combination 11"
           severity error;
48
49         wait; -- indefinitely suspend process
50     end process;
51 end tb;

```

[https://github.com/mahdihaghverdi/archproject/blob/main/ALU/src/half\\_adder/half\\_adder\\_process\\_tb.vhdl](https://github.com/mahdihaghverdi/archproject/blob/main/ALU/src/half_adder/half_adder_process_tb.vhdl)

### ۴.۱.۱ خروجی simulation



## ۲.۱ مازول Full Adder

## ۱.۲.۱ کد این مازول

---

```

1  -- one bit full adder
2
3  library ieee;
4  use ieee.std_logic_1164.all;
5
6  entity full_adder is
7      port (
8          a, b, ci : in std_logic;
9          s, c : out std_logic
10     );
11 end full_adder;
12
13
14 architecture arch of full_adder is
15     component half_adder
16         port (
17             a, b : in std_logic;
18             sum, carry : out std_logic
19         );
20     end component;
21
22     for half_adder_0: half_adder use entity work.half_adder;
23     for half_adder_1: half_adder use entity work.half_adder;
24     signal asb, aab, asbco: std_logic;
25
26 begin
27     half_adder_0: half_adder port map(
28         -- entity-signal-name => local-signal-name
29         a => a,
30         b => b,
31         sum => asb,
32         carry => aab
33     );
34     half_adder_1: half_adder port map(
35         a => asb,
36         b => ci,
37         sum => s,
38         carry => asbco
39     );

```



```

40
41     c <= aab or asbco;
42 end arch;

```

---

[https://github.com/mahdihaghverdi/archproject/blob/main/ALU/src/full\\_adder/full\\_adder.vhdl](https://github.com/mahdihaghverdi/archproject/blob/main/ALU/src/full_adder/full_adder.vhdl)

## ۲.۲.۱ تست‌بنچ ۱ این ماژول

---

```

1  library ieee;
2  use ieee.std_logic_1164.all;
3
4  entity full_adder_tb is
5  end full_adder_tb;
6
7  architecture tb of full_adder_tb is
8      signal a, b, ci : std_logic; -- inputs
9      signal sum, carry : std_logic; -- outputs
10 begin
11     -- connecting testbench signals with full_adder.vhdl
12     UUT : entity work.full_adder port map (
13         a => a,
14         b => b,
15         ci => ci,
16         s => sum,
17         c => carry
18     );
19
20     -- inputs
21     -- ci ba
22     -- 0 00 at 0 ns
23     -- 0 01 at 20 ns
24     -- 0 10 at 40 ns
25     -- 0 11 at 60 ns
26     -- 1 00 at 80 ns
27     -- 1 01 at 100 ns
28     -- 1 10 at 120 ns
29     -- 1 11 at 140 ns
30     -- 1 11 at 160 ns
31
32     a <= '0', '1' after 20 ns, '0' after 40 ns, '1' after 60
        ns, '0' after 80 ns, '1' after 100 ns, '0' after 120
        ns, '1' after 140 ns, '1' after 160 ns;

```

```

33     b <= '0', '1' after 40 ns, '0' after 80 ns, '0' after 120
        ns, '1' after 140 ns, '1' after 160 ns;
34     ci <= '0', '1' after 80 ns, '1' after 120 ns, '1' after
        160 ns;
35 end tb ;

```

[https://github.com/mahdihaghverdi/archproject/blob/main/ALU/src/full\\_adder/full\\_adder\\_tb.vhdl](https://github.com/mahdihaghverdi/archproject/blob/main/ALU/src/full_adder/full_adder_tb.vhdl)

### ۳.۲.۱ تست‌بنچ ۲ این ماژول

```

1  library ieee;
2  use ieee.std_logic_1164.all;
3
4
5  entity full_adder_process_tb is
6  end full_adder_process_tb;
7
8  architecture tb of full_adder_process_tb is
9      signal a, b, ci : std_logic; -- inputs
10     signal sum, carry : std_logic; -- outputs
11 begin
12     -- connecting testbench signals with full_adder.vhdl
13     UUT : entity work.full_adder port map (
14         a => a,
15         b => b,
16         ci => ci,
17         s => sum,
18         c => carry
19     );
20     tb1 : process
21         constant period: time := 20 ns;
22         begin
23             -- a b s c
24             -- 0 0 0 0
25             -- 0 1 1 0
26             -- 1 0 1 0
27             -- 1 1 0 1
28
29             a <= '0';
30             b <= '0';
31             ci <= '0';
32             wait for period;

```

```

33     assert ((sum = '0') and (carry = '0')) -- expected
        output
34     -- error will be reported if sum or carry is not 0
35     report "test failed for input combination 000"
        severity error;
36
37     a <= '0';
38     b <= '1';
39     ci <= '0';
40     wait for period;
41     assert ((sum = '1') and (carry = '0'))
42     report "test failed for input combination 001"
        severity error;
43
44     a <= '1';
45     b <= '0';
46     ci <= '0';
47     wait for period;
48     assert ((sum = '1') and (carry = '0'))
49     report "test failed for input combination 010"
        severity error;
50
51     a <= '1';
52     b <= '1';
53     ci <= '0';
54     wait for period;
55     assert ((sum = '0') and (carry = '1'))
56     report "test failed for input combination 011"
        severity error;
57
58
59     a <= '0';
60     b <= '0';
61     ci <= '1';
62     wait for period;
63     assert ((sum = '1') and (carry = '0')) -- expected
        output
64     -- error will be reported if sum or carry is not 0
65     report "test failed for input combination 100"
        severity error;
66
67     a <= '0';
68     b <= '1';
69     ci <= '1';

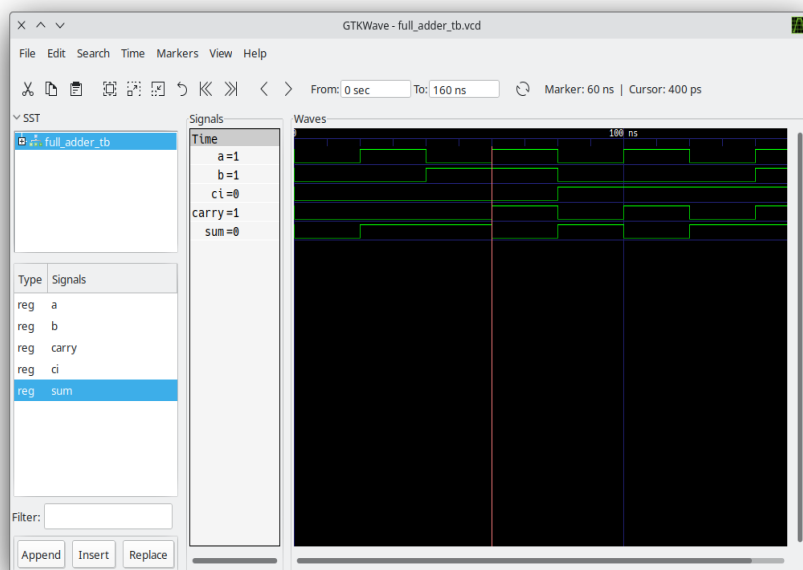
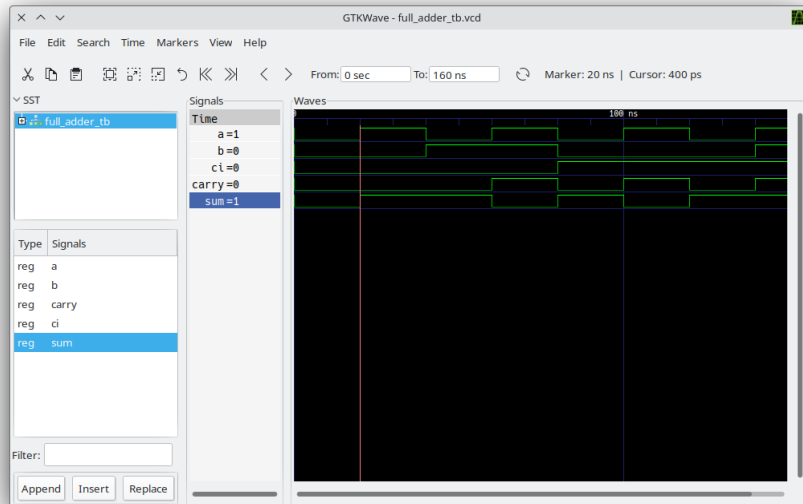
```

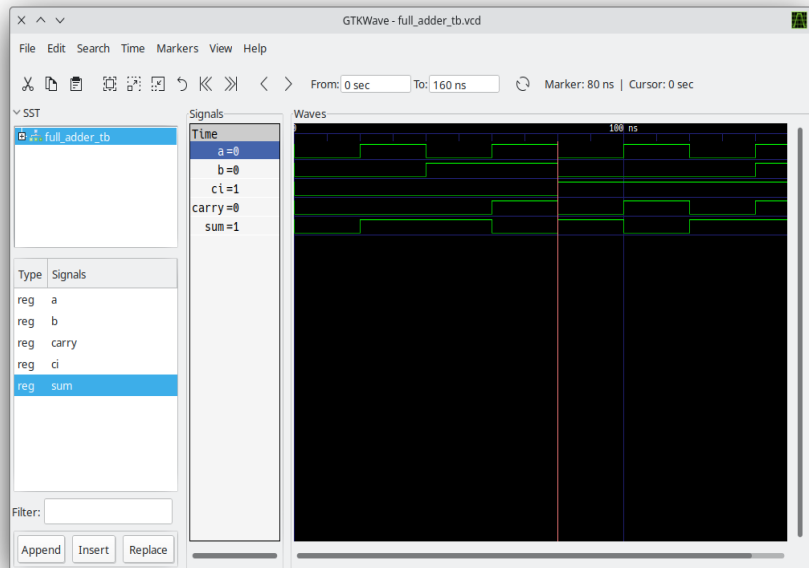
```
70     wait for period;
71     assert ((sum = '0') and (carry = '1'))
72     report "test failed for input combination 101"
       severity error;
73
74     a <= '1';
75     b <= '0';
76     ci <= '1';
77     wait for period;
78     assert ((sum = '0') and (carry = '1'))
79     report "test failed for input combination 110"
       severity error;
80
81     a <= '1';
82     b <= '1';
83     ci <= '1';
84     wait for period;
85     assert ((sum = '1') and (carry = '1'))
86     report "test failed for input combination 111"
       severity error;
87
88
89     assert false report "all tests passed." severity note;
90     wait; -- indefinitely suspend process
91 end process;
92 end tb ;
```

---

[https://github.com/mahdihaghverdi/archproject/blob/main/ALU/src/full\\_adder/full\\_adder\\_process\\_tb.vhdl](https://github.com/mahdihaghverdi/archproject/blob/main/ALU/src/full_adder/full_adder_process_tb.vhdl)

## ۴.۲.۱ خروجی simulation





## Four Bit Full Adder ۳.۱

## ۱.۳.۱ کد این مازول

---

```

1  library ieee;
2  use ieee.std_logic_1164.all;
3
4  entity FourBitFullAdder is
5      port (
6          input_a, input_b : in std_logic_vector (3 downto 0);
7          sum : out std_logic_vector (3 downto 0);
8          carry : out std_logic
9      );
10 end FourBitFullAdder;
11
12 architecture arch of FourBitFullAdder is
13     component full_adder
14         port (
15             a, b, ci : in std_logic;
16             s, c : out std_logic
17         );
18     end component;
19
20     for full_adder_0: full_adder use entity work.full_adder;
21     for full_adder_1: full_adder use entity work.full_adder;
22     for full_adder_2: full_adder use entity work.full_adder;
23     for full_adder_3: full_adder use entity work.full_adder;
24
25     signal carry_0_1, carry_1_2, carry_2_3 : std_logic;
26 begin
27     full_adder_0: full_adder port map (
28         a => input_a(0),
29         b => input_b(0),
30         ci => '0',
31         s => sum(0),
32         c => carry_0_1
33     );
34     full_adder_1: full_adder port map (
35         a => input_a(1),
36         b => input_b(1),
37         ci => carry_0_1,
38         s => sum(1),
39         c => carry_1_2

```

```

40 );
41 full_adder_2: full_adder port map (
42     a => input_a(2),
43     b => input_b(2),
44     ci => carry_1_2,
45     s => sum(2),
46     c => carry_2_3
47 );
48 full_adder_3: full_adder port map (
49     a => input_a(3),
50     b => input_b(3),
51     ci => carry_2_3,
52     s => sum(3),
53     c => carry
54 );
55
56 end arch;
```

---

[https://github.com/mahdihaghverdi/archproject/blob/main/ALU/src/four\\_bit\\_full\\_adder/four\\_bit\\_full\\_adder.vhdl](https://github.com/mahdihaghverdi/archproject/blob/main/ALU/src/four_bit_full_adder/four_bit_full_adder.vhdl)

### ۲.۳.۱ تست‌بنچ این ماژول

---

```

1  library ieee;
2  use ieee.std_logic_1164.all;
3
4
5  entity FourBitFullAdderTB is
6  end FourBitFullAdderTB;
7
8  architecture tb of FourBitFullAdderTB is
9      signal a, b: std_logic_vector (3 downto 0); -- inputs
10     signal sum : std_logic_vector (3 downto 0); -- outputs
11     signal carry : std_logic;
12
13 begin
14     -- connecting testbench signals with 4bitfa.vhdl
15     UUT : entity work.FourBitFullAdder port map (
16         input_a => a,
17         input_b => b,
18         sum => sum,
19         carry => carry
20     );
```



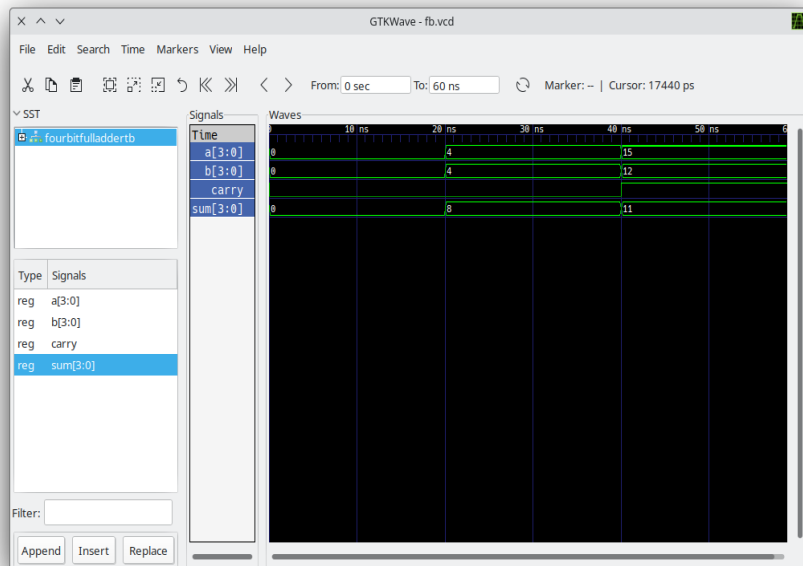
```

21
22 -- inputs and outs dec
23
24 -- bb aa          c ss
25 -- 0 0 at 0 ns -> 0 0
26 -- 4 4 at 20 ns -> 0 8
27 -- 12 15 at 40 ns -> 1 11
28
29 -- inputs and outs bin
30 -- 0000 0000 -> 0 0000
31 -- 0100 0100 -> 0 1000
32 -- 1100 1111 -> 1 1011
33
34 a <= "0000", "0100" after 20 ns, "1111" after 40 ns,
    "1111" after 60 ns;
35 b <= "0000", "0100" after 20 ns, "1100" after 40 ns,
    "1100" after 60 ns;
36 end tb ;

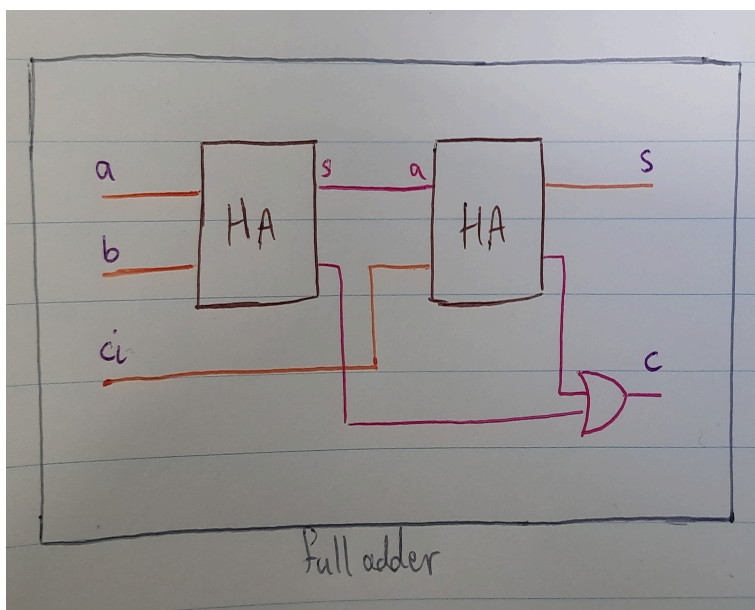
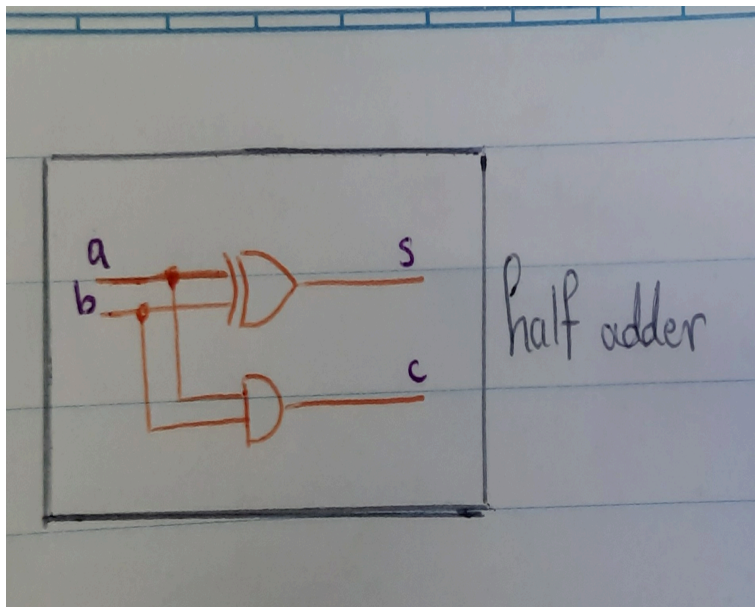
```

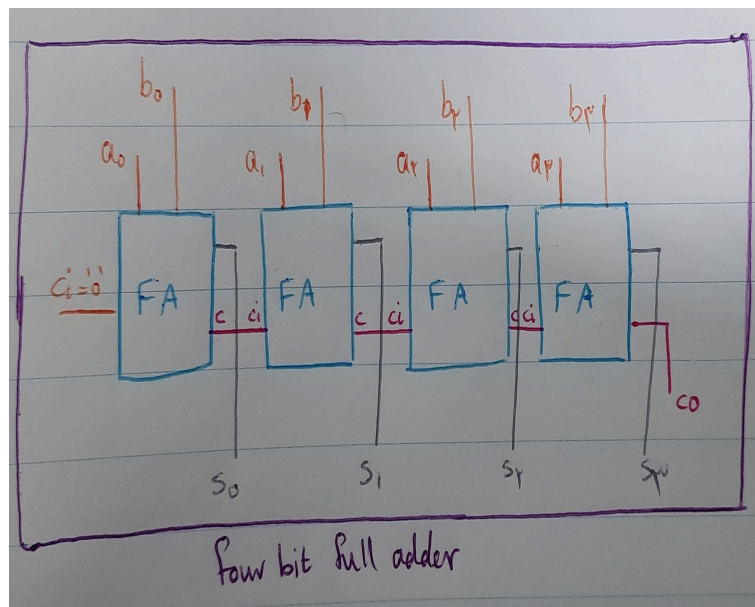
[https://github.com/mahdihaghverdi/archproject/blob/main/ALU/src/four\\_bit\\_full\\_adder/four\\_bit\\_full\\_adder\\_tb.vhdl](https://github.com/mahdihaghverdi/archproject/blob/main/ALU/src/four_bit_full_adder/four_bit_full_adder_tb.vhdl)

### ۳.۳.۱ خروجی simulation



## ۴.۱ طراحی‌ها





## فصل ۲

# ماژول‌های ساختار تفریق

### ۱.۲ ماژول Half Subber

#### ۱.۱.۲ کد این ماژول

---

```
1 library ieee;
2 use ieee.std_logic_1164.all;
3
4 entity half_subber is
5     port(
6         a, b: in std_logic;
7         diff, borrow : out std_logic
8     );
9 end half_subber;
10
11 architecture arch of half_subber is
12 begin
13     diff <= a xor b;
14     borrow <= not(a) and b;
15 end arch;
```

---

[https://github.com/mahdihaghverdi/archproject/blob/main/ALU/src/half\\_subber/half\\_subber.vhdl](https://github.com/mahdihaghverdi/archproject/blob/main/ALU/src/half_subber/half_subber.vhdl)

#### ۲.۱.۲ تست‌بنچ ۱ ماژول

---

```
1 library ieee;
2 use ieee.std_logic_1164.all;
3
4
5 entity half_subber_tb is
```

```

6 end half_subber_tb;
7
8 architecture tb of half_subber_tb is
9     signal a, b : std_logic; -- inputs
10    signal diff, borrow : std_logic; -- outputs
11 begin
12    -- connecting testbench signals with half_subber.vhdl
13    UUT : entity work.half_subber port map (
14        a => a,
15        b => b, diff => diff,
16        borrow => borrow
17    );
18
19    -- inputs
20    -- ba          db
21    -- 00 at 0 ns -> 00
22    -- 01 at 20 ns -> 11
23    -- 10 at 40 ns -> 10
24    -- 11 at 60 ns -> 00
25    -- 11 at 80 ns -> 00
26
27    a <= '0', '1' after 20 ns, '0' after 40 ns, '1' after 60
        ns, '1' after 80 ns;
28    b <= '0', '1' after 40 ns, '1' after 60 ns;
29 end tb ;

```

---

[https://github.com/mahdihaghverdi/archproject/blob/main/ALU/src/half\\_subber/half\\_subber\\_tb.vhdl](https://github.com/mahdihaghverdi/archproject/blob/main/ALU/src/half_subber/half_subber_tb.vhdl)

### ۳.۱.۲ تست‌بنچ ۲ ماژول

---

```

1 -- half_subber_process_tb.vhd
2
3 library ieee;
4 use ieee.std_logic_1164.all;
5
6
7 entity half_subber_process_tb is
8 end half_subber_process_tb;
9
10 architecture tb of half_subber_process_tb is
11     signal a, b : std_logic;
12     signal diff, borrow : std_logic;

```

```

13 begin
14     -- connecting testbench signals with half_subber.vhd
15     UUT : entity work.half_subber port map (
16         a => a,
17         b => b,
18         diff => diff,
19         borrow => borrow
20     );
21
22     tb1 : process
23         constant period: time := 20 ns;
24         begin
25             a <= '0';
26             b <= '0';
27             wait for period;
28             assert ((diff = '0') and (borrow = '0')) --
                expected output
29             -- error will be reported if diff or borrow is not 0
30             report "test failed for input combination 00"
                severity error;
31
32             a <= '0';
33             b <= '1';
34             wait for period;
35             assert ((diff = '1') and (borrow = '1'))
36             report "test failed for input combination 01"
                severity error;
37
38             a <= '1';
39             b <= '0';
40             wait for period;
41             assert ((diff = '1') and (borrow = '0'))
42             report "test failed for input combination 10"
                severity error;
43
44             a <= '1';
45             b <= '1';
46             wait for period;
47             assert ((diff = '0') and (borrow = '0'))
48             report "test failed for input combination 11"
                severity error;
49
50             assert false report "all tests passed." severity
                note;

```

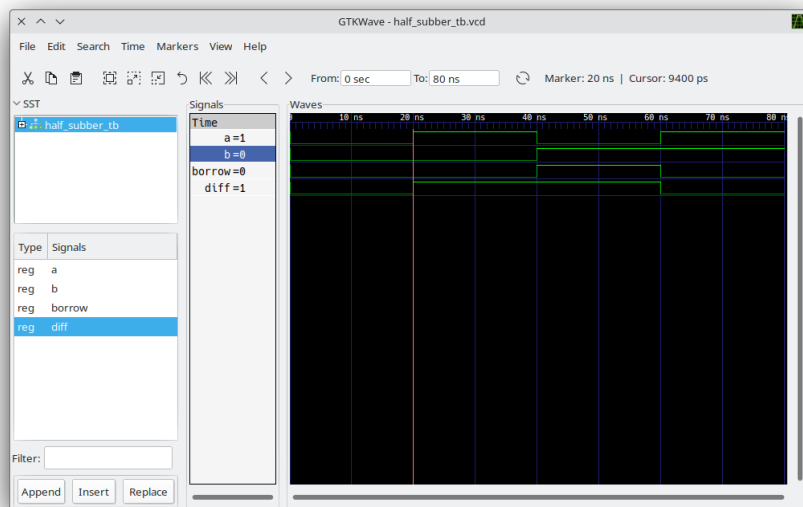
```

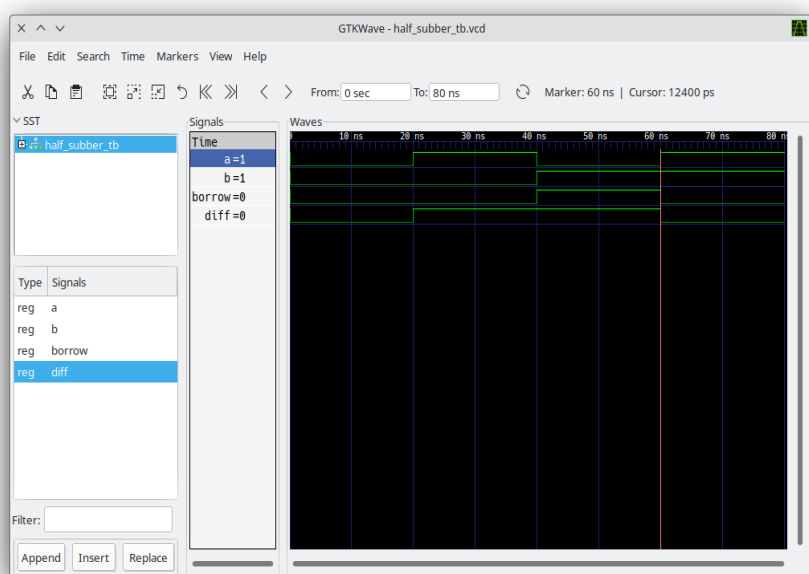
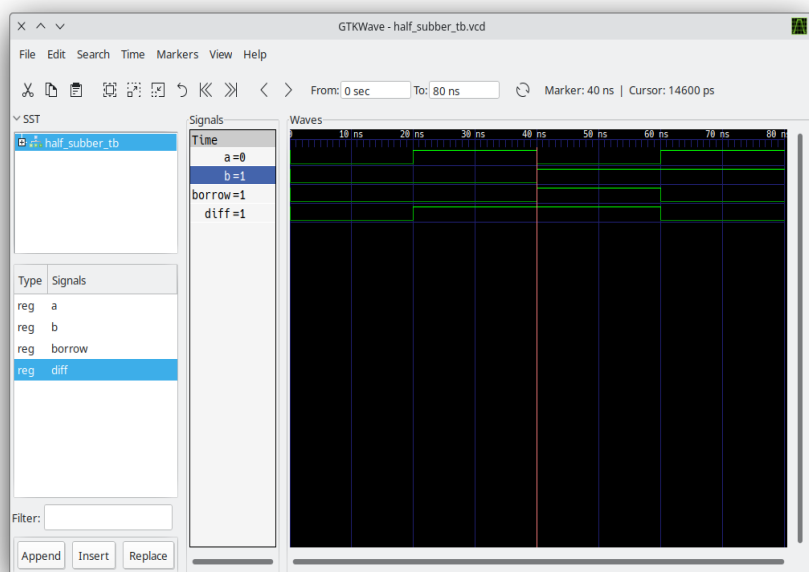
51         wait; -- indefinitely suspend process
52     end process;
53 end tb;

```

[https://github.com/mahdihaghverdi/archproject/blob/main/ALU/src/half\\_subber/half\\_subber\\_process\\_tb.vhdl](https://github.com/mahdihaghverdi/archproject/blob/main/ALU/src/half_subber/half_subber_process_tb.vhdl)

## ۴.۱.۲ خروجی simulation





## ۲.۲ مازول Full Subber

### ۱.۲.۲ کد این مازول



---

```

1  library ieee;
2  use ieee.std_logic_1164.all;
3
4  entity full_subber is
5      port(
6          a , b, borrow_in: in std_logic;
7          diff, borrow_out : out std_logic
8      );
9  end full_subber;
10
11 architecture arch of full_subber is
12     component half_subber is
13         port(
14             a, b : in std_logic;
15             diff, borrow : out std_logic
16         );
17     end component;
18
19     for half_subber_0: half_subber use entity work.half_subber;
20     for half_subber_1: half_subber use entity work.half_subber;
21     signal adb, anab, anabi : std_logic;
22
23 begin
24     half_subber_0: half_subber port map (
25         a => a,
26         b => b,
27         diff => adb,
28         borrow => anab
29     );
30     half_subber_1: half_subber port map (
31         A => adb,
32         B => borrow_in,
33         diff => diff,
34         borrow => anabi
35     );
36
37     borrow_out <= anab or anabi;
38 end arch;

```

---

[https://github.com/mahdihaghverdi/archproject/blob/main/ALU/src/full\\_subber/full\\_subber.vhdl](https://github.com/mahdihaghverdi/archproject/blob/main/ALU/src/full_subber/full_subber.vhdl)

## ۲.۲.۲ تست‌بنچ مازول

---

```

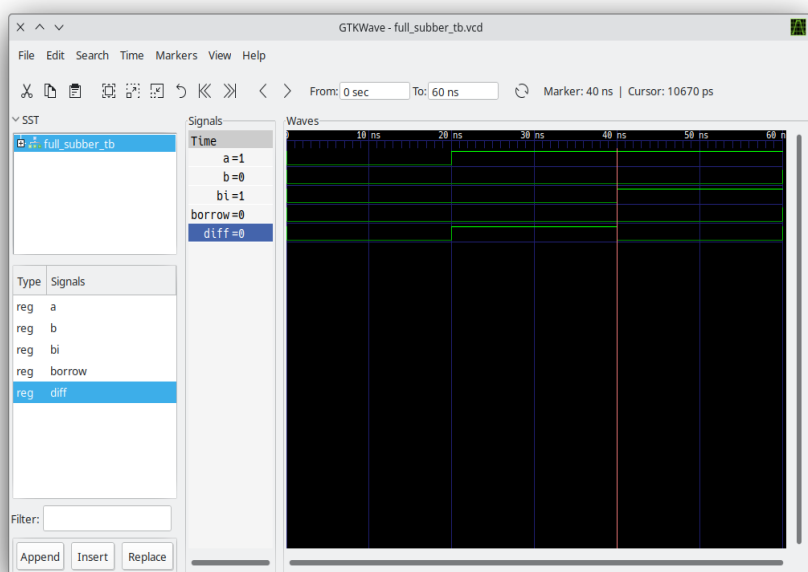
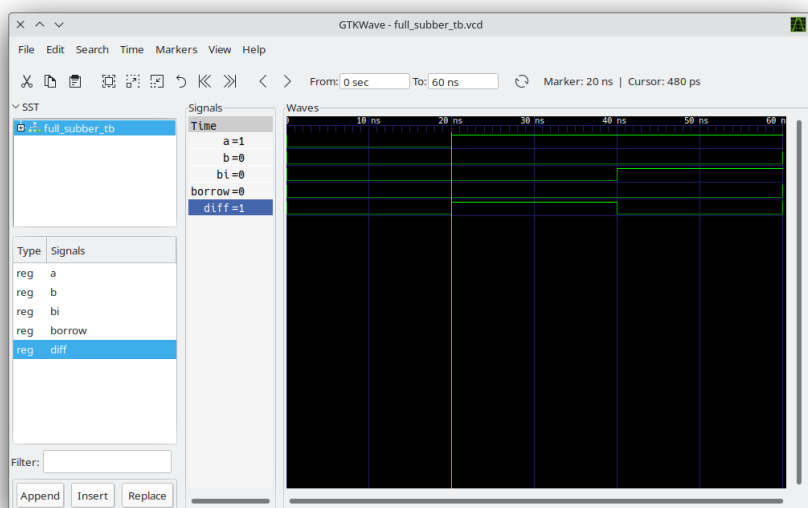
1  library ieee;
2  use ieee.std_logic_1164.all;
3
4
5  entity full_subber_tb is
6  end full_subber_tb;
7
8  architecture tb of full_subber_tb is
9      signal a, b, bi : std_logic; -- inputs
10     signal diff, borrow : std_logic; -- outputs
11 begin
12     -- connecting testbench signals with full_subber.vhdl
13     UUT : entity work.full_subber port map (
14         a => a,
15         b => b,
16         borrow_in => bi,
17         diff => diff,
18         borrow_out => borrow
19     );
20
21     -- inputs
22     -- bi ba          db
23     -- 0 00 at 0 ns -> 00
24     -- 0 01 at 20 ns -> 11
25     -- 1 01 at 40 ns -> 01
26     -- 1 11 at 60 ns -> 11
27
28     bi <= '0', '1' after 40 ns, '1' after 60 ns;
29     b <= '0', '1' after 60 ns;
30     a <= '0', '1' after 20 ns, '1' after 60 ns;
31 end tb ;

```

---

[https://github.com/mahdihaghverdi/archproject/blob/main/ALU/src/full\\_subber/full\\_subber\\_tb.vhdl](https://github.com/mahdihaghverdi/archproject/blob/main/ALU/src/full_subber/full_subber_tb.vhdl)

## ۳.۲.۲ خروجی simulation



## ۳.۲ مازول Four Bit Full Subber

## ۱.۳.۲ کد این مازول

---

```

1  library ieee;
2  use ieee.std_logic_1164.all;
3
4  entity FourBitFullSubber is
5      port (
6          input_a, input_b : in std_logic_vector (3 downto 0);
7          diff : out std_logic_vector (3 downto 0);
8          borrow : out std_logic
9      );
10 end FourBitFullSubber;
11
12 architecture arch of FourBitFullSubber is
13     component full_subber
14         port(
15             a , b, borrow_in: in std_logic;
16             diff, borrow_out : out std_logic
17         );
18     end component;
19
20     for full_subber_0: full_subber use entity work.full_subber;
21     for full_subber_1: full_subber use entity work.full_subber;
22     for full_subber_2: full_subber use entity work.full_subber;
23     for full_subber_3: full_subber use entity work.full_subber;
24
25     signal borrow_0_1, borrow_1_2, borrow_2_3 : std_logic;
26     begin
27         full_subber_0: full_subber port map (
28             a => input_a(0),
29             b => input_b(0),
30             borrow_in => '0',
31             diff => diff(0),
32             borrow_out => borrow_0_1
33         );
34         full_subber_1: full_subber port map (
35             a => input_a(1),
36             b => input_b(1),
37             borrow_in => borrow_0_1,
38             diff => diff(1),
39             borrow_out => borrow_1_2
40         );
41         full_subber_2: full_subber port map (
42             a => input_a(2),
43             b => input_b(2),

```

```

44         borrow_in => borrow_1_2,
45         diff => diff(2),
46         borrow_out => borrow_2_3
47     );
48     full_subber_3: full_subber port map (
49         a => input_a(3),
50         b => input_b(3),
51         borrow_in => borrow_2_3,
52         diff => diff(3),
53         borrow_out => borrow
54     );
55 end arch;

```

---

[https://github.com/mahdihaghverdi/archproject/blob/main/ALU/src/four\\_bit\\_full\\_subber/four\\_bit\\_full\\_subber.vhdl](https://github.com/mahdihaghverdi/archproject/blob/main/ALU/src/four_bit_full_subber/four_bit_full_subber.vhdl)

### ۲.۳.۲ تست‌بنچ ماژول

---

```

1  library ieee;
2  use ieee.std_logic_1164.all;
3
4
5  entity FourBitFullSubberTB is
6  end FourBitFullSubberTB;
7
8  architecture tb of FourBitFullSubberTB is
9      signal a, b: std_logic_vector (3 downto 0); -- inputs
10     signal diff : std_logic_vector (3 downto 0); -- outputs
11     signal borrow : std_logic;
12
13 begin
14     -- connecting testbench signals with 4bitfa.vhdl
15     UUT : entity work.FourBitFullSubber port map (
16         input_a => a,
17         input_b => b,
18         diff => diff,
19         borrow => borrow
20     );
21
22     -- inputs and outs dec
23
24     -- b a bi          d b
25     -- 0 0 0 at 0 ns -> 0 0

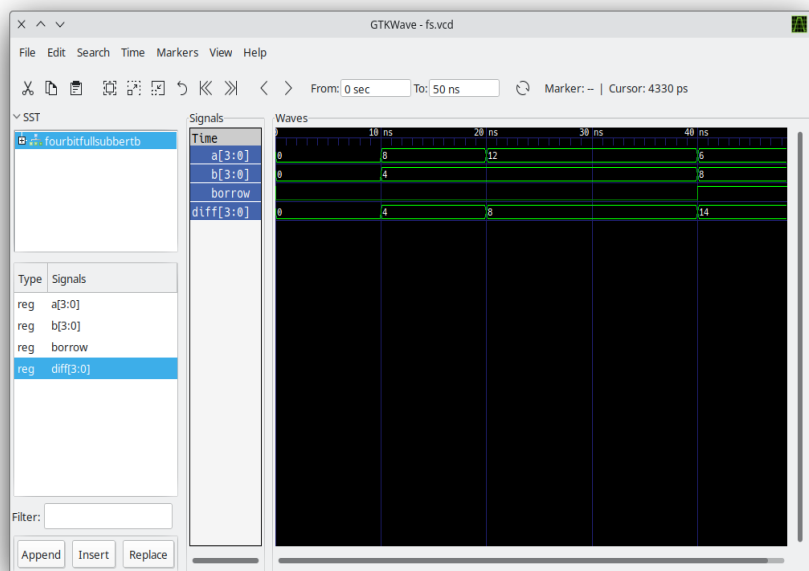
```

```
26      -- 0 0 1 at 20 ns -> 1 1
27      -- 0 1 0 at 40 ns -> 1 1
28      -- 0 1 1 at 60 ns -> 0 1
29      -- 1 0 0 at 80 ns -> 1 0
30      -- 1 0 1 at 100 ns -> 0 0
31      -- 1 1 0 at 120 ns -> 0 0
32      -- 1 1 1 at 140 ns -> 1 1
33
34      -- inputs and outs bin
35      -- 1000 0100 -> 0 0100
36      -- 1100 0100 -> 0 1000
37      -- 0110 1000 -> 1 1110
38
39      a <= "0000",
40          "1000" after 10 ns,
41          "1100" after 20 ns,
42          "1100" after 30 ns,
43          "0110" after 40 ns,
44          "0110" after 50 ns;
45
46      b <= "0000",
47          "0100" after 10 ns,
48          "0100" after 20 ns,
49          "0100" after 30 ns,
50          "1000" after 40 ns,
51          "1000" after 50 ns;
52  end tb ;
```

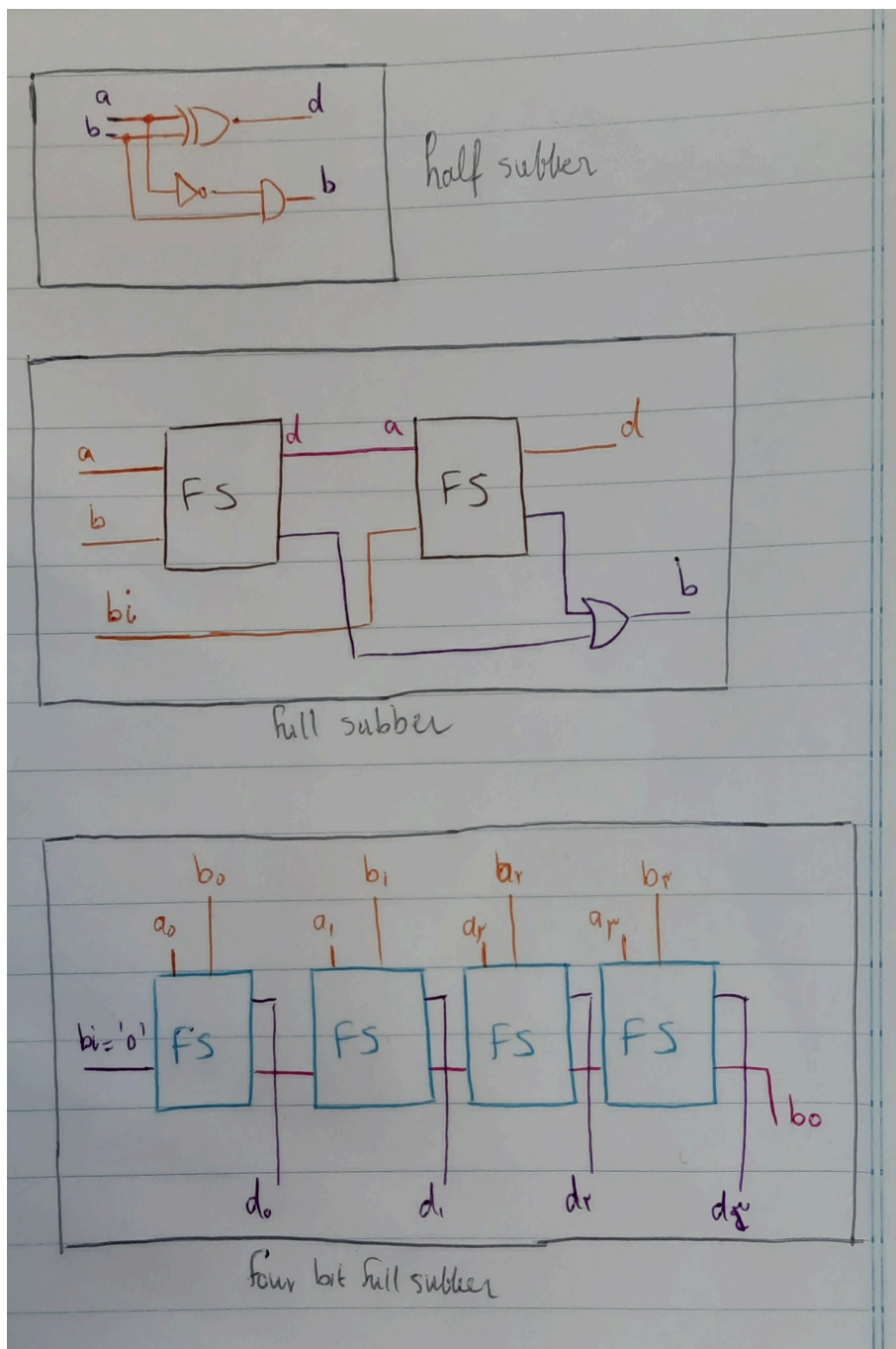
---

[https://github.com/mahdihaghverdi/archproject/blob/main/ALU/src/four\\_bit\\_full\\_subber/four\\_bit\\_full\\_subber\\_tb.vhdl](https://github.com/mahdihaghverdi/archproject/blob/main/ALU/src/four_bit_full_subber/four_bit_full_subber_tb.vhdl)

## ۳.۳.۲ خروجی simulation



۴.۲ طراحی‌ها





## فصل ۳

# ماژول‌های ساختار ضرب

### ۱.۳ ماژول Four Bit Multiplier

۱.۱.۳ کد این ماژول

---

```
1 library ieee;
2 use ieee.std_logic_1164.all;
3
4 entity FourBitMultiplier is
5     port (
6         input_a, input_b : in std_logic_vector (3 downto 0);
7         output: out std_logic_vector (7 downto 0)
8     );
9 end FourBitMultiplier;
10
11 architecture arch of FourBitMultiplier is
12     component FourBitFullAdder
13     port (
14         input_a, input_b : in std_logic_vector (3 downto 0);
15         sum : out std_logic_vector (3 downto 0);
16         carry : out std_logic
17     );
18 end component;
19
20 for four_bit_full_adder_0: FourBitFullAdder use entity
21     work.FourBitFullAdder;
22 for four_bit_full_adder_1: FourBitFullAdder use entity
23     work.FourBitFullAdder;
```

```

24  signal a0b1, a0b2, a0b3 : std_logic;
25  signal input_a_0, input_b_0, input_a_1, input_b_1,
    input_a_2, input_b_2 : std_logic_vector (3 downto 0);
26  signal a1b0, a1b1, a1b2, a1b3 : std_logic;
27  signal a2b0, a2b1, a2b2, a2b3 : std_logic;
28  signal a3b0, a3b1, a3b2, a3b3 : std_logic;
29  signal sum0, sum1, sum2 : std_logic_vector (3 downto 0);
30  signal carry_0_1, carry_1_2, carry_2_3 : std_logic;
31
32  begin
33      a0b1 <= input_a(0) and input_b(1);
34      a0b2 <= input_a(0) and input_b(2);
35      a0b3 <= input_a(0) and input_b(3);
36      input_a_0 <= ('0', a0b3, a0b2, a0b1);
37      input_b_0 <= (a1b3, a1b2, a1b1, a1b0);
38
39      a1b0 <= input_a(1) and input_b(0);
40      a1b1 <= input_a(1) and input_b(1);
41      a1b2 <= input_a(1) and input_b(2);
42      a1b3 <= input_a(1) and input_b(3);
43
44      four_bit_full_adder_0: FourBitFullAdder port map (
45          input_a => input_a_0,
46          input_b => input_b_0,
47          sum => sum0,
48          carry => carry_0_1
49      );
50      a2b0 <= input_a(2) and input_b(0);
51      a2b1 <= input_a(2) and input_b(1);
52      a2b2 <= input_a(2) and input_b(2);
53      a2b3 <= input_a(2) and input_b(3);
54
55      input_a_1 <= (carry_0_1, sum0(3), sum0(2), sum0(1));
56      input_b_1 <= (a2b3, a2b2, a2b1, a2b0);
57
58      four_bit_full_adder_1: FourBitFullAdder port map (
59          input_a => input_a_1,
60          input_b => input_b_1,
61          sum => sum1,
62          carry => carry_1_2
63      );
64
65      a3b0 <= input_a(3) and input_b(0);
66      a3b1 <= input_a(3) and input_b(1);

```

```

67     a3b2 <= input_a(3) and input_b(2);
68     a3b3 <= input_a(3) and input_b(3);
69
70     input_a_2 <= (carry_1_2, sum1(3), sum1(2), sum1(1));
71     input_b_2 <= (a3b3, a3b2, a3b1, a3b0);
72     four_bit_full_adder_2: FourBitFullAdder port map (
73         input_a => input_a_2,
74         input_b => input_b_2,
75         sum => sum2,
76         carry => carry_2_3
77     );
78     output <= (
79         carry_2_3,
80         sum2(3), sum2(2), sum2(1), sum2(0),
81         sum1(0),
82         sum0(0),
83         input_a(0) and input_b(0)
84     );
85 end arch;

```

---

[https://github.com/mahdihaghverdi/archproject/blob/main/ALU/src/four\\_bit\\_multiplier/four\\_bit\\_multiplier.vhdl](https://github.com/mahdihaghverdi/archproject/blob/main/ALU/src/four_bit_multiplier/four_bit_multiplier.vhdl)

### ۲.۱.۳ تست‌بنچ این ماژول

---

```

1  library ieee;
2  use ieee.std_logic_1164.all;
3
4
5  entity FourBitMultiplierTB is
6  end FourBitMultiplierTB;
7
8  architecture tb of FourBitMultiplierTB is
9      signal a, b: std_logic_vector (3 downto 0); -- inputs
10     signal output : std_logic_vector (7 downto 0); -- outputs
11
12 begin
13     -- connecting testbench signals with 4bitfa.vhdl
14     UUT : entity work.FourBitMultiplier port map (
15         input_a => a,
16         input_b => b,
17         output => output
18     );

```

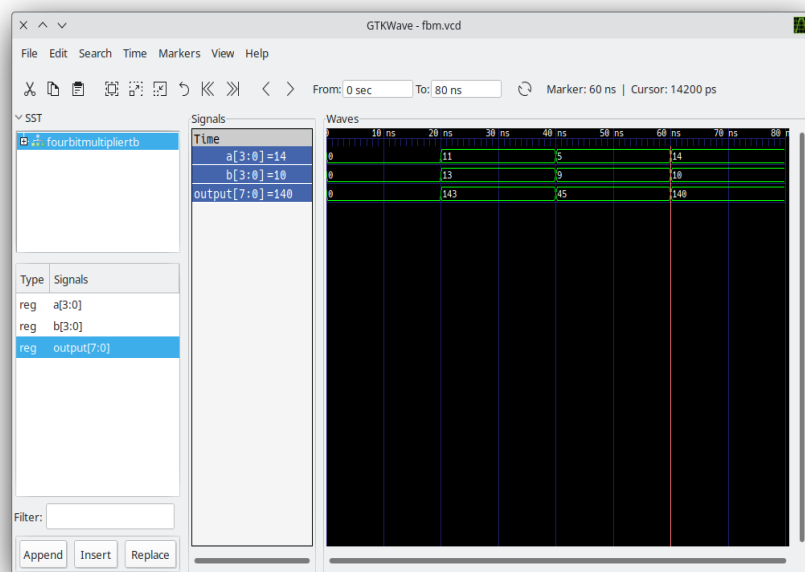
```

19
20      -- bbbb aaaa  oooooooooo
21      -- 1101 1011 -> 10001111
22      -- 1001 0101 -> 00101101
23      -- 1010 1110 -> 10001100
24
25      a <= "0000", "1011" after 20 ns, "0101" after 40 ns,
          "1110" after 60 ns, "1110" after 80 ns;
26      b <= "0000", "1101" after 20 ns, "1001" after 40 ns,
          "1010" after 60 ns, "1010" after 80 ns;
27  end tb ;

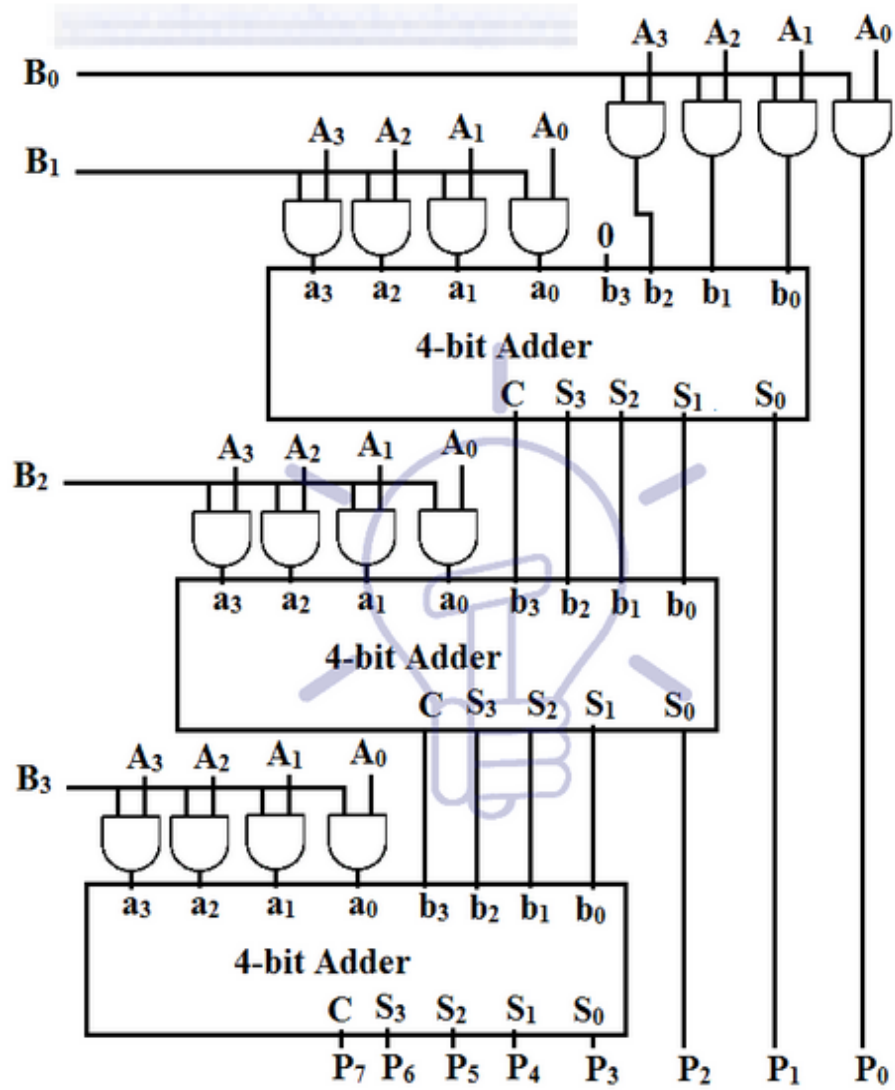
```

[https://github.com/mahdihaghverdi/archproject/blob/main/ALU/src/four\\_bit\\_multiplier/four\\_bit\\_multiplier\\_tb.vhdl](https://github.com/mahdihaghverdi/archproject/blob/main/ALU/src/four_bit_multiplier/four_bit_multiplier_tb.vhdl)

### ۳.۱.۳ خروجی simulation



### ۲.۳ طراحی‌ها



## فصل ۴

# ماژول مالتی پلکسر

۱.۴ ماژول mux4x1

۱.۱.۴ کد این ماژول

---

```
1 library ieee;
2 use ieee.std_logic_1164.all;
3
4 entity mux4x1 is
5     port (
6         i0, i1, i2, i3 : in std_logic_vector (7 downto 0);
7         sel : in std_logic_vector (1 downto 0);
8         output: out std_logic_vector (7 downto 0)
9     );
10 end mux4x1;
11
12 architecture arch of mux4x1 is
13     begin
14         with sel select output <=
15             i0 when "00",
16             i1 when "01",
17             i2 when "10",
18             i3 when others;
19 end arch;
```

---

<https://github.com/mahdihaghverdi/archproject/blob/main/ALU/src/mux4x1/mux4x1.vhdl>

۲.۱.۴ تست بنچ این ماژول

---

```
1 library ieee;
```

```

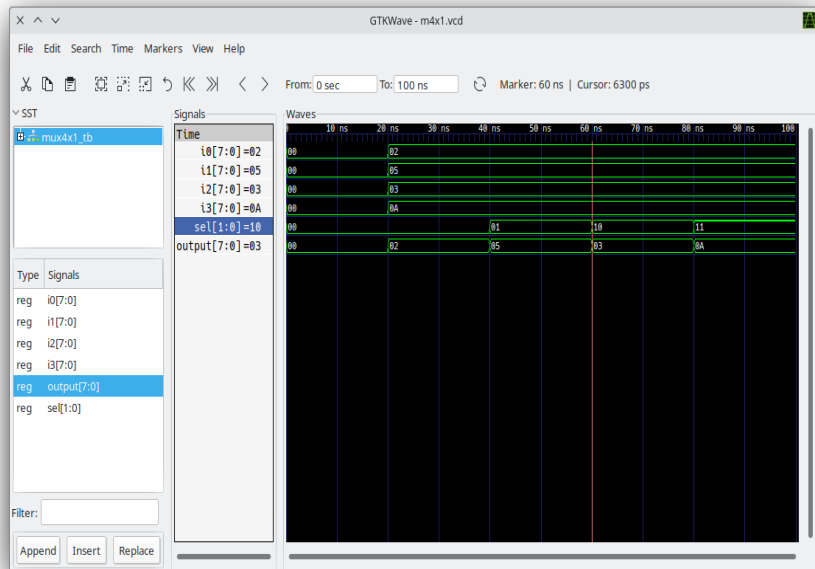
2  use ieee.std_logic_1164.all;
3
4  entity mux4x1_tb is
5  end mux4x1_tb;
6
7  architecture arch of mux4x1_tb is
8      signal i0, i1, i2, i3, output : std_logic_vector (7 downto
9          0);
10     signal sel : std_logic_vector (1 downto 0);
11 begin
12     UUT : entity work.mux4x1 port map (
13         i0 => i0,
14         i1 => i1,
15         i2 => i2,
16         i3 => i3,
17         sel => sel,
18         output => output
19     );
20
21     -- i3333333 i2222222 i1111111 i0000000 sel ooooooooo
22     -- 00001010 00000011 00000101 00000010 00 00000010
23     -- 00001010 00000011 00000101 00000010 01 00000101
24     -- 00001010 00000011 00000101 00000010 10 00000011
25     -- 00001010 00000011 00000101 00000010 11 00001010
26
27     i0 <= "00000000", "00000010" after 20 ns, "00000010" after
28         40 ns, "00000010" after 60 ns, "00000010" after 80 ns;
29     i1 <= "00000000", "00000101" after 20 ns, "00000101" after
30         40 ns, "00000101" after 60 ns, "00000101" after 80 ns;
31     i2 <= "00000000", "00000011" after 20 ns, "00000011" after
32         40 ns, "00000011" after 60 ns, "00000011" after 80 ns;
33     i3 <= "00000000", "00001010" after 20 ns, "00001010" after
34         40 ns, "00001010" after 60 ns, "00001010" after 80 ns;
35     sel <= "00", "00" after 20 ns, "01" after 40 ns, "10"
36         after 60 ns, "11" after 80 ns, "11" after 100 ns;
37
38 end arch;

```

---

[https://github.com/mahdihaghverdi/archproject/blob/main/ALU/src/mux4x1/mux4x1\\_tb.vhdl](https://github.com/mahdihaghverdi/archproject/blob/main/ALU/src/mux4x1/mux4x1_tb.vhdl)

## ۳.۱.۴ خروجی simulation





## فصل ۵

# ماژول ALU

۱.۵ ماژول ALU

۱.۱.۵ کد این ماژول

---

```
1 library ieee;
2 use ieee.std_logic_1164.all;
3
4 entity alu is
5 port (
6   alu_input_a, alu_input_b: in std_logic_vector (3 downto 0);
7   alu_out           : out std_logic_vector (7 downto 0);
8   alu_carry_borrow_out : out std_logic;
9   alu_select        : in std_logic_vector (1 downto 0)
10 );
11 end alu;
12
13 architecture arch of alu is
14   -- we need a mux, four_bit_full_adder/subber/multiplier
15   -- 00 -> full_adder
16   -- 01 -> full_subber
17   -- 10 -> full_multiplier
18   -- 11 -> all one :)
19
20   component mux4x1
21     port (
22       i0, i1, i2, i3: in std_logic_vector (3 downto 0);
23       sel           : in std_logic_vector (1 downto 0);
24       output        : out std_logic_vector (3 downto 0)
25     );
26   end component;
```

```

27
28     component FourBitFullAdder
29         port (
30             input_a, input_b: in std_logic_vector (3 downto 0);
31             sum           : out std_logic_vector (3 downto 0);
32             carry         : out std_logic
33         );
34     end component;
35
36     component FourBitFullSubber
37         port (
38             input_a, input_b: in std_logic_vector (3 downto 0);
39             diff           : out std_logic_vector (3 downto 0);
40             borrow         : out std_logic
41         );
42     end component;
43
44     component FourBitMultiplier
45         port (
46             input_a, input_b: in std_logic_vector (3 downto 0);
47             output          : out std_logic_vector (7 downto 0)
48         );
49     end component;
50
51     signal full_adder_to_i0_t, full_subber_to_i1_t :
52         std_logic_vector(3 downto 0);
53     signal before_alu_out : std_logic_vector(3 downto 0);
54     signal multiplier_to_i2, full_adder_to_i0,
55         full_subber_to_i1 : std_logic_vector (7 downto 0);
56     signal after_alu_out : std_logic_vector (7 downto 0);
57     begin
58         inner_full_adder : entity work.FourBitFullAdder port map (
59             input_a => alu_input_a,
60             input_b => alu_input_b,
61             sum => full_adder_to_i0_t,
62             carry => alu_carry_borrow_out
63         );
64         full_adder_to_i0 <= "0000" & full_adder_to_i0_t;
65
66         inner_full_subber : entity work.FourBitFullSubber port map
67             (
68                 input_a => alu_input_a,

```

```

68     input_b => alu_input_b,
69     diff => full_subber_to_i1_t,
70     borrow => alu_carry_borrow_out
71 );
72 full_subber_to_i1 <= "0000" & full_subber_to_i1_t;
73
74 inner_multiplier : entity work.FourBitMultiplier port map (
75     input_a => alu_input_a,
76     input_b => alu_input_b,
77     output => multiplier_to_i2
78 );
79
80 inner_mux : entity work.mux4x1 port map (
81     i0 => full_adder_to_i0,
82     i1 => full_subber_to_i1,
83     i2 => multiplier_to_i2,
84     i3 => "11111111",
85     sel => alu_select,
86     output => alu_out
87 );
88 end arch;

```

<https://github.com/mahdihaghverdi/archproject/blob/main/ALU/src/alu.vhdl>

## ۲.۱.۵ تست‌بنچ این ماژول

```

1  library ieee;
2  use ieee.std_logic_1164.all;
3
4  -- A testbench has no ports.
5  entity alu_tb is
6  end alu_tb;
7
8  architecture tb of alu_tb is
9      signal input_a, input_b : std_logic_vector (3 downto 0);
10     signal output           : std_logic_vector (7 downto 0);
11     signal carry_borrow_out : std_logic;
12     signal sel              : std_logic_vector (1 downto 0);
13
14 begin
15     UUT : entity work.alu port map (
16         alu_input_a => input_a,

```

```

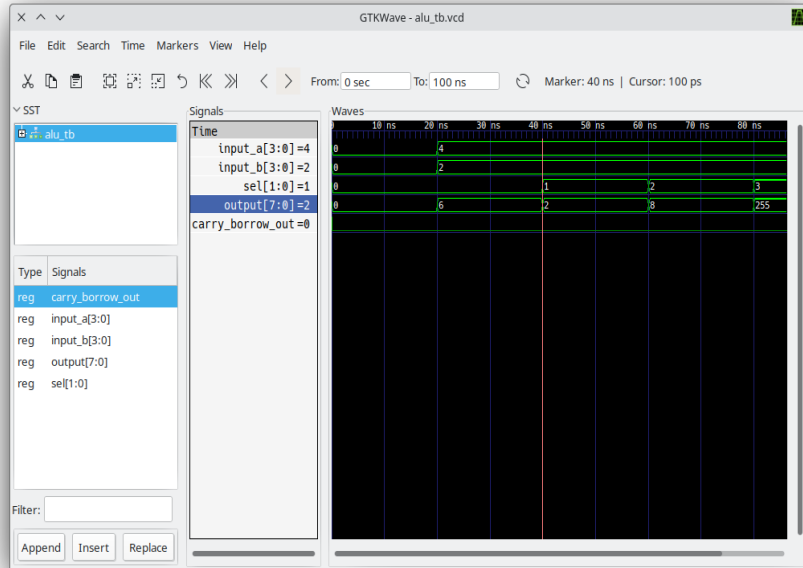
17     alu_input_b => input_b,
18     alu_out => output,
19     alu_carry_borrow_out => carry_borrow_out,
20     alu_select => sel
21 );
22
23 -- aaaa bbbb ss ++++ ---- ***** `~~~`
24 -- 0000 0000 00 0000
25 -- 0000 0000 01 0000
26 -- 0000 0000 10 00000000
27 -- 0000 0000 10 1111
28
29 -- 0100 0010 00 0110
30 -- 0100 0010 01 0010
31 -- 0100 0010 10 00001000
32 -- 0100 0010 11 1111
33
34 input_a <= "0000", "0100" after 20 ns;
35 input_b <= "0000", "0010" after 20 ns;
36 sel <= "00", "00" after 20 ns, "01" after 40 ns, "10"
    after 60 ns, "11" after 80 ns, "11" after 100 ns;
37 end tb;

```

---

[https://github.com/mahdihaghverdi/archproject/blob/main/ALU/src/alu\\_tb.vhdl](https://github.com/mahdihaghverdi/archproject/blob/main/ALU/src/alu_tb.vhdl)

## ۳.۱.۵ خروجی simulation



## ۲.۵ طراحی‌ها

