## گزارش تمرین عملی دوم

مهدي حقوردي

۲۰ اردیبهشت ۲۰

## فهرست مطالب

١	ماژولهای ساختار جمع	١
١	۱.۱ ماژول Half Adder ماژول	
١	۱۰۱۰ کد این ماژول	
١	۲۰۱۰۱ تستبنچ ۱ این ماژول	
۲	۳۰۱۰۱ تست منتج ۲ این ماژول	
۴	۴۰۱۰۱ خروجی simulation	
۵	۲.۱ ماژول Full Adder ماژول	
۵	۱۰۲۰۱ کد این ماژول	
۶	۲۰۲۰۱ تستبنچ ۱ این ماژول	
٧	۳۰۲۰۱ تستبنچ ۲ این ماژول	
۰ (	۴۰۲۰۱ خروجی simulation	
۱۲	· · · · · · · · · · · · · · · · · · ·	
۱۲	۱۰۳۰۱ کد این ماژول	
۱۳		
۱۴		
۱۵	۴.۱ طراحیها	
		J
17	ماژولهای ساختار تفریق مرکز بازی سرماله TT-16 Galla	١
١٧		
١٧	۱۰۱۰۲ کد این ماژول	
١٧		
١٨	۳۰۱۰۲ تستبنچ ۲ ماژول	
۲۰	۴۰۱۰۲ خروجی simulation	
۲۱	۲.۲ ماژول Full Subber	
۲۱	۱۰۲۰۲ کد این ماژول	
74	(* 1	
74		
74		
74		
78		
۲۸	۳.۳.۲ خروجی simulation	
49	۴.۲ طراحیها	

ب	فهرست مطالب
ب	فهرست مطالب

٣	۳ ماژولهای ساختار ضرب	٣.
	۱۰۳ ماژول Four Bit Multiplier	٣٠
	۱۰۱۰۳ کد این ماژول	٣٠
	۲۰۱۰۳ تستبنچ این ماژول	44
	۳۰۱۰۳ خروجی simulation	٣٣
	۲۰۳ طراحیها ۲۰۰۰ میراند	44
۴	۴ ماژول مالتی پلکسر	٣۵
	۱.۴ ماژول mux4x1 ماژول ۱.۴	٣۵
	۱۰۱۰۴ کد این ماژول	
	۲۰۱۰۴ تستبنچ این ماژول ۲۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰	٣۵
	۳۰۱۰۴ خروجی simulation	
۵	۵ ماژول ALU	٣٨
	۱۰۵ ماژول ALU ماژول	٣٨
	۱۰۱۰۵ کد این ماژول	٣٨
	۲۰۱۰۵ تستبنچ این ماژول ۲۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰	40
	۳۰۱۰۵ خروجی simulation	
	۲.۵ طراحیها	44

# فصل ۱ ماژولهای ساختار جمع

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

#### ماژول Half Adder 1.1

۱۰۱۰۱ کد این ماژول

```
1 library ieee;
2 use ieee.std_logic_1164.all;
3
4
   entity half_adder is
       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;</pre>
14
       carry <= a and b;</pre>
15 end arch;
```

https://github.com/mahdihaghverdi/archproject/blob/main/ALU/sr c/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;
8 architecture tb of half_adder_tb is
9
       signal a, b : std_logic; -- inputs
       signal sum, carry : std_logic; -- outputs
10
11 begin
12
       -- connecting testbench signals with half adder.vhd
       UUT : entity work.half_adder port map (
13
14
              a \Rightarrow a
15
              b \Rightarrow b,
16
              sum => sum,
17
              carry => carry
           );
18
19
20
      -- inputs
       -- ba
21
22
       -- 00 at 0 ns
23
       -- 01 at 20 ns, as b is 0 at 20 ns and a is changed to 1
          at 20 ns
24
       -- 10 at 40 ns
25
       -- 11 at 60 ns
26
       -- 11 at 80 ns
       a <= '0', '1' after 20 ns, '0' after 40 ns, '1' after 60
           ns, '1' after 80 ns;
       b <= '0', '1' after 40 ns, '1' after 60 ns;
28
29 end tb;
```

https://github.com/mahdihaghverdi/archproject/blob/main/ALU/src/half\_adder/half\_adder\_tb.vhdl

#### ۳.۱.۱ تستبنچ ۲ این ماژول

```
-- half_adder_process_tb.vhd

library ieee;

use ieee.std_logic_1164.all;

entity half_adder_process_tb is

end half_adder_process_tb;
```

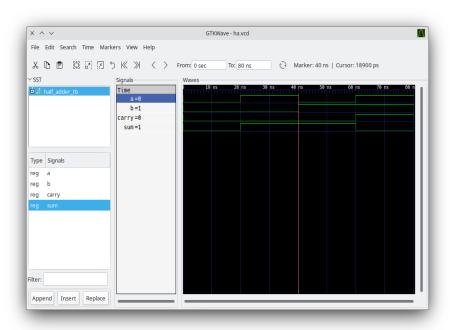
```
architecture tb of half_adder_process_tb is
9
       signal a, b : std_logic;
10
       signal sum, carry : std_logic;
11
12
   begin
13
       -- connecting testbench signals with half_adder.vhd
14
       UUT : entity work.half_adder port map (
15
           a \Rightarrow a
           b \Rightarrow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

```
report "test failed for input combination 11"
severity error;

wait; -- indefinitely suspend process
end process;
end tb;
```

https://github.com/mahdihaghverdi/archproject/blob/main/ALU/src/half\_adder/half\_adder\_process\_tb.vhdl

#### ۴۰۱۰۱ خروجی ۴۰۱۰۱



### ۲.۱ ماژول ۲۰۱

#### ۱.۲.۱ کد این ماژول

```
1 -- one bit full adder
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;
               sum, carry : out std_logic
18
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
       half_adder_0: half_adder port map(
27
           -- entity-signal-name => local-signal-name
28
29
           a \Rightarrow a
30
           b \Rightarrow b,
31
           sum => asb,
32
           carry => aab
33
34
       half_adder_1: half_adder port map(
35
           a \Rightarrow asb,
           b \Rightarrow ci,
36
37
           sum => s,
           carry => asbco
38
39
       );
```

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

https://github.com/mahdihaghverdi/archproject/blob/main/ALU/src/full\_adder.vhdl

### ۲.۲.۱ تستبنچ ۱ این ماژول

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

#### ٣٠٢.١ تستبنچ ٢ اين 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
       signal sum, carry : std_logic; -- outputs
10
11
12
       -- connecting testbench signals with full_adder.vhdl
13
       UUT : entity work.full_adder port map (
14
           a \Rightarrow a
15
           b \Rightarrow 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
           -- 0 0 0 0
24
           -- 0 1 1 0
25
26
           -- 1 0 1 0
           -- 1 1 0 1
27
28
           a <= '0';
29
           b <= '0';
30
31
           ci <= '0';
32
           wait for period;
```

```
فصل ۱. ماژولهای ساختار جمع
```

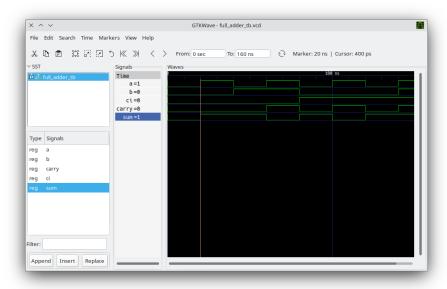
```
٨
```

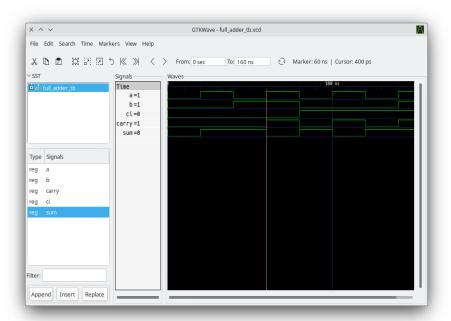
```
assert ((sum = '0') and (carry = '0')) -- expected
33
              output
           -- error will be reported if sum or carry is not 0
34
          report "test failed for input combination 000"
35
              severity error;
36
37
          a <= '0';
          b <= '1';
38
           ci <= '0';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

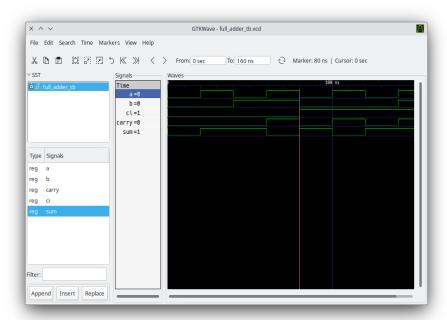
```
٩
                                               فصل ۱. ماژولهای ساختار جمع
70
          wait for period;
71
          assert ((sum = '0') and (carry = '1'))
          report "test failed for input combination 101"
72
              severity error;
73
          a <= '1';
74
75
          b <= '0';
          ci <= '1';
76
77
          wait for period;
          assert ((sum = '0') and (carry = '1'))
78
          report "test failed for input combination 110"
79
              severity error;
80
          a <= '1';
81
82
          b <= '1';
          ci <= '1';
83
          wait for period;
84
          assert ((sum = '1') and (carry = '1'))
85
          report "test failed for input combination 111"
86
              severity error;
87
88
89
          assert false report "all tests passed." severity note;
          wait; -- indefinitely suspend process
90
91
       end process;
```

92 end tb;

### ۴.۲.۱ خروجی ۴.۲.۱







#### Four Bit Full Adder 7.1

#### ۱.٣.١ كد اين ما ژول

```
1 library ieee;
2 use ieee.std_logic_1164.all;
3
   entity FourBitFullAdder is
4
       port (
5
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
       component full_adder
13
14
           port (
15
              a, b, ci : in std_logic;
               s, c : out std_logic
16
17
           );
18
       end component;
19
20
       for full_adder_0: full_adder use entity work.full_adder;
       for full_adder_1: full_adder use entity work.full_adder;
21
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
       full_adder_0: full_adder port map (
27
           a => input_a(0),
28
29
           b \Rightarrow input_b(0),
           ci => '0',
30
31
           s \Rightarrow sum(0),
           c => carry_0_1
32
33
       );
       full_adder_1: full_adder port map (
34
35
           a => input_a(1),
           b => input_b(1),
36
37
           ci => carry_0_1,
           s \Rightarrow sum(1),
38
           c => carry_1_2
39
```

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

https://github.com/mahdihaghverdi/archproject/blob/main/ALU/src/four\_bit\_full\_adder.vhdl

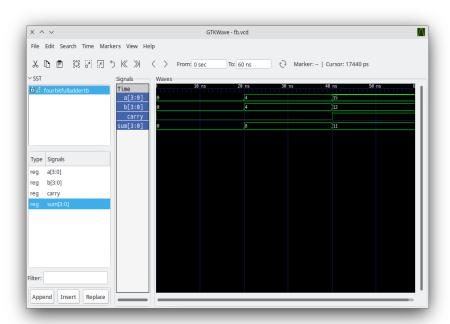
#### ۲.۳.۱ تستبنچ این ماژول

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

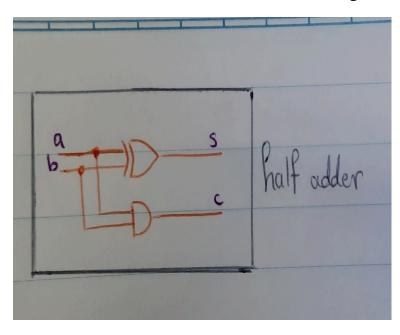
```
21
22
       -- inputs and outs dec
23
24
       -- bb aa
       -- 0 0 at 0 ns -> 0 0
25
       -- 4 4 at 20 ns -> 0 8
26
27
       -- 12 15 at 40 ns -> 1 11
28
29
       -- inputs and outs bin
      -- 0000 0000 -> 0 0000
30
       -- 0100 0100 -> 0 1000
31
      -- 1100 1111 -> 1 1011
32
33
 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

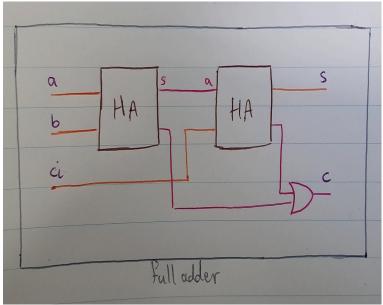
https://github.com/mahdihaghverdi/archproject/blob/main/ALU/src/four\_bit\_full\_adder/four\_bit\_full\_adder\_tb.vhdl

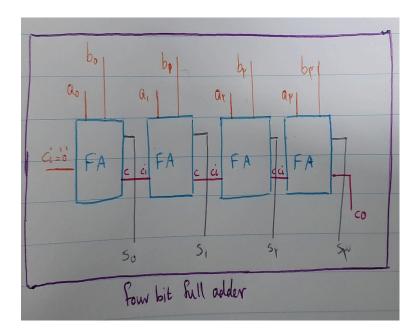
#### simulation خروجی ۳.۳.۱



## ۴.۱ طراحیها







# فصل ۲ ماژولهای ساختار تفریق

۱۰۲ 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
       );
   end half_subber;
9
10
11 architecture arch of half_subber is
12 begin
13
       diff <= a xor b;</pre>
       borrow <= not(a) and b;</pre>
14
15 end arch;
```

https://github.com/mahdihaghverdi/archproject/blob/main/ALU/sr c/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
       signal a, b : std_logic; -- inputs
9
       signal diff, borrow : std_logic; -- outputs
10
11 begin
12
       -- connecting testbench signals with half_subber.vhdl
       UUT : entity work.half_subber port map (
13
14
          a \Rightarrow a
15
          b => b, diff => diff,
16
          borrow => borrow
17
       );
18
19
       -- inputs
20
      -- ba
      -- 00 at 0 ns -> 00
21
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
       a <= '0', '1' after 20 ns, '0' after 40 ns, '1' after 60
27
          ns, '1' after 80 ns;
       b <= '0', '1' after 40 ns, '1' after 60 ns;
28
29 end tb;
```

https://github.com/mahdihaghverdi/archproject/blob/main/ALU/src/half\_subber\_tb.vhdl

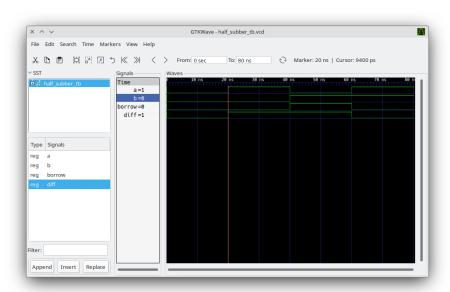
#### ۳.۱.۲ تستبنچ ۲ ماژول

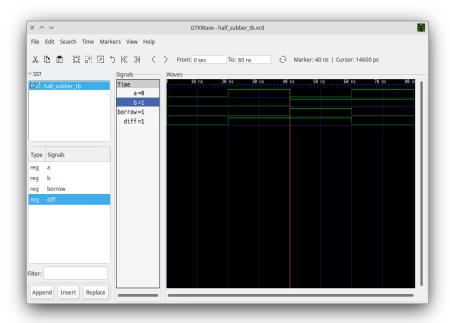
```
1 -- half_subber_process_tb.vhd
2
3 library ieee;
4 use ieee.std_logic_1164.all;
5
6
7 entity half_subber_process_tb is
8 end half_subber_process_tb;
9
10 architecture tb of half_subber_process_tb is
11 signal a, b : std_logic;
12 signal diff, borrow : std_logic;
```

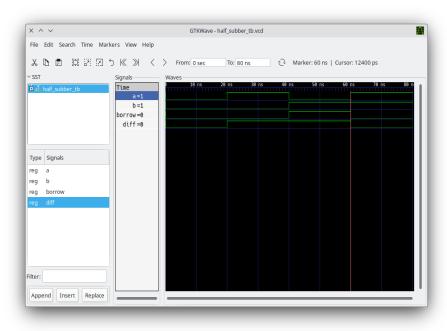
```
13 begin
14
       -- connecting testbench signals with half_subber.vhd
       UUT : entity work.half_subber port map (
15
16
           a \Rightarrow a
           b \Rightarrow b,
17
18
           diff => diff,
19
           borrow => borrow
20
       );
21
22
       tb1 : process
23
           constant period: time := 20 ns;
24
           begin
25
              a <= '0';
              b <= '0';
26
              wait for period;
27
              assert ((diff = '0') and (borrow = '0')) --
28
                  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
              a <= '0';
32
33
              b <= '1';
34
              wait for period;
              assert ((diff = '1') and (borrow = '1'))
35
36
              report "test failed for input combination 01"
                  severity error;
37
              a <= '1';
38
              b <= '0';
39
40
              wait for period;
              assert ((diff = '1') and (borrow = '0'))
41
              report "test failed for input combination 10"
                  severity error;
43
44
              a <= '1';
              b <= '1';
45
              wait for period;
46
47
              assert ((diff = '0') and (borrow = '0'))
              report "test failed for input combination 11"
48
                  severity error;
49
50
              assert false report "all tests passed." severity
                  note;
```

https://github.com/mahdihaghverdi/archproject/blob/main/ALU/src/half\_subber/half\_subber\_process\_tb.vhdl

#### ۴.۱.۲ خروجی ۴.۱.۲







۲۰۲ ماژول ۲۰۲

۱۰۲.۲ کد این ماژول

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

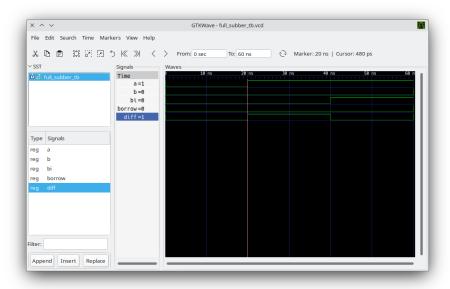
https://github.com/mahdihaghverdi/archproject/blob/main/ALU/src/full\_subber.vhdl

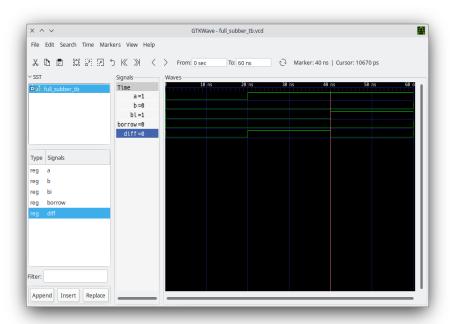
### ۲.۲.۲ تستبنچ ماژول

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

https://github.com/mahdihaghverdi/archproject/blob/main/ALU/src/full\_subber\_tb.vhdl

#### ۳.۲.۲ خروجی ۳.۲.۲





Four Bit Full Subber ماژول ۳.۲ ۱.۳.۲ کد این ماژول

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

```
borrow_in => borrow_1_2,
44
45
               diff => diff(2),
               borrow_out => borrow_2_3
46
47
           );
           full_subber_3: full_subber port map (
48
49
               a \Rightarrow input_a(3),
               b \Rightarrow input_b(3),
50
               borrow_in => borrow_2_3,
51
               diff => diff(3),
52
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

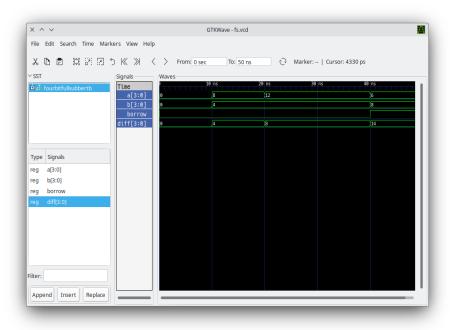
#### ۲.۳.۲ تستبنچ ماژول

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

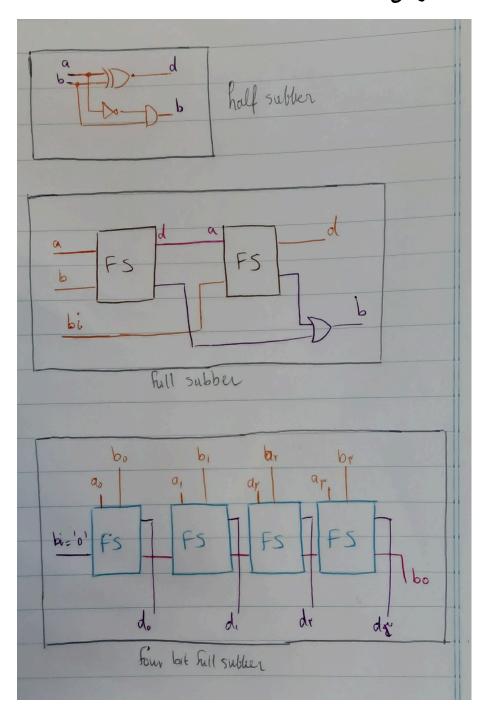
```
26
       -- 0 0 1 at 20 ns -> 1 1
27
       -- 0 1 0 at 40 ns -> 1 1
28
       -- 0 1 1 at 60 ns -> 0 1
       -- 1 0 0 at 80 ns -> 1 0
29
30
      -- 1 0 1 at 100 ns -> 0 0
      -- 1 1 0 at 120 ns -> 0 0
31
32
       -- 1 1 1 at 140 ns -> 1 1
33
34
      -- inputs and outs bin
      -- 1000 0100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

https://github.com/mahdihaghverdi/archproject/blob/main/ALU/src/four\_bit\_full\_subber/four\_bit\_full\_subber\_tb.vhdl

#### ۳.۳.۲ خروجی ۳.۳.۲



## ۴۰۲ طراحیها



# فصل ۳ ماژولهای ساختار ضرب

۱.۳ ماژول ۱۰۳ ۱.۱.۳ کد این ماژول

```
1 library ieee;
2 use ieee.std_logic_1164.all;
3
4
   entity FourBitMultiplier is
       port (
5
          input_a, input_b : in std_logic_vector (3 downto 0);
6
7
          output: out std_logic_vector (7 downto 0)
8
   end FourBitMultiplier;
9
10
   architecture arch of FourBitMultiplier is
11
       component FourBitFullAdder
12
13
          port (
          input_a, input_b : in std_logic_vector (3 downto 0);
14
15
          sum : out std_logic_vector (3 downto 0);
16
          carry : out std_logic
17
          );
       end component;
18
19
       for four_bit_full_adder_0: FourBitFullAdder use entity
20
          work.FourBitFullAdder;
       for four_bit_full_adder_1: FourBitFullAdder use entity
21
          work.FourBitFullAdder;
       for four_bit_full_adder_2: FourBitFullAdder use entity
22
          work.FourBitFullAdder;
23
```

```
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);
       signal a1b0, a1b1, a1b2, a1b3 : std_logic;
26
       signal a2b0, a2b1, a2b2, a2b3 : std_logic;
27
       signal a3b0, a3b1, a3b2, a3b3 : std_logic;
28
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
           a0b1 \le input a(0) and input b(1);
33
34
           a0b2 <= input_a(0) and input_b(2);
35
           a0b3 <= input_a(0) and input_b(3);
           input_a_0 <= ('0', a0b3, a0b2, a0b1);
36
           input_b_0 <= (a1b3, a1b2, a1b1, a1b0);
37
38
           a1b0 <= input_a(1) and input_b(0);
39
           a1b1 <= input_a(1) and input_b(1);</pre>
40
           a1b2 <= input_a(1) and input_b(2);</pre>
41
           a1b3 <= input_a(1) and input_b(3);
42
43
44
           four_bit_full_adder_0: FourBitFullAdder port map (
              input_a => input_a_0,
45
46
              input_b => input_b_0,
47
              sum => sum0,
48
              carry => carry_0_1
49
           );
           a2b0 <= input_a(2) and input_b(0);</pre>
50
51
           a2b1 <= input_a(2) and input_b(1);
52
           a2b2 <= input_a(2) and input_b(2);</pre>
53
           a2b3 <= input_a(2) and input_b(3);</pre>
54
           input_a_1 <= (carry_0_1, sum0(3), sum0(2), sum0(1));
55
56
           input_b_1 <= (a2b3, a2b2, a2b1, a2b0);
57
           four_bit_full_adder_1: FourBitFullAdder port map (
58
59
              input_a => input_a_1,
              input_b => input_b_1,
60
              sum => sum1,
61
              carry => carry_1_2
62
           );
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
           input_a_2 <= (carry_1_2, sum1(3), sum1(2), sum1(1));
70
           input_b_2 <= (a3b3, a3b2, a3b1, a3b0);
71
           four_bit_full_adder_2: FourBitFullAdder port map (
72
73
              input_a => input_a_2,
74
              input_b => input_b_2,
75
              sum => sum2,
76
              carry => carry_2_3
77
           );
          output <= (
78
79
              carry_2_3,
              sum2(3), sum2(2), sum2(1), sum2(0),
80
              sum1(0),
81
82
              sum0(0),
              input_a(0) and input_b(0)
83
84
          );
85
   end arch;
```

https://github.com/mahdihaghverdi/archproject/blob/main/ALU/src/four\_bit\_multiplier.vhdl

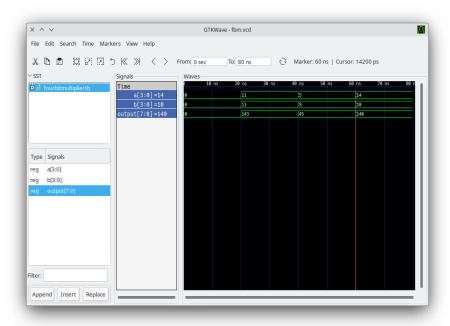
#### ۲.۱.۳ تستبنچ این ماژول

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

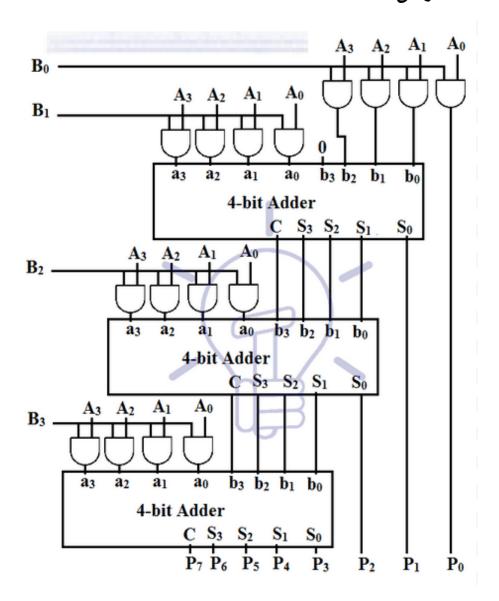
```
19
20
      -- bbbb aaaa ooooooo
21
      -- 1101 1011 -> 10001111
       -- 1001 0101 -> 00101101
22
      -- 1010 1110 -> 10001100
23
24
25
       a <= "0000", "1011" after 20 ns, "0101" after 40 ns,
          "1110" after 60 ns, "1110" after 80 ns;
       b \le "0000", "1101" after 20 ns, "1001" after 40 ns,
26
          "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\_multplier\_tb.vhdl

#### ۳۰۱۰۳ خروجی ۳۰۱۰۳



۲.۳ طراحیها



# فصل ۴ ماژول مالتی پلکسر

۱.۴ ماژول ۱۰۴

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);
          sel : in std_logic_vector (1 downto 0);
          output: out std_logic_vector (7 downto 0)
8
9
       );
10
   end mux4x1;
11
12
   architecture arch of mux4x1 is
13
       with sel select output <=
14
15
          i0 when "00",
          i1 when "01",
16
17
          i2 when "10",
18
          i3 when others;
19 end arch;
```

https://github.com/mahdihaghverdi/archproject/blob/main/ALU/sr c/mux4x1/mux4x1.vhdl

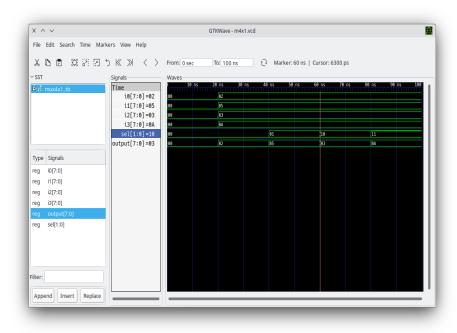
۲.۱.۴ تستبنچ این ماژول

1 library ieee;

```
2 use ieee.std_logic_1164.all;
3
4 entity mux4x1_tb is
   end mux4x1_tb;
5
6
7
   architecture arch of mux4x1_tb is
      signal i0, i1, i2, i3, output : std_logic_vector (7 downto
8
         0):
      signal sel : std_logic_vector (1 downto 0);
9
10
   begin
      UUT : entity work.mux4x1 port map (
11
12
         i0 => i0,
13
         i1 => i1,
14
         i2 => i2,
         i3 => i3,
15
16
         sel => sel,
         output => output
17
      );
18
19
20
21
      -- i3333333 i2222222 i11111111 i0000000 sel oooooooo
      22
23
      -- 00001010 00000011 00000101 00000010 01 00000101
      24
      -- 00001010 00000011 00000101 00000010 11 00001010
25
26
27
      i0 <= "00000000", "00000010" after 20 ns, "00000010" after
         40 ns, "00000010" after 60 ns, "00000010" after 80 ns;
      i1 <= "00000000", "00000101" after 20 ns, "00000101" after
28
         40 ns, "00000101" after 60 ns, "00000101" after 80 ns;
29
      i2 \le "000000000", "000000011" after 20 ns, "000000011" after
         40 ns, "00000011" after 60 ns, "00000011" after 80 ns;
      i3 <= "00000000", "00001010" after 20 ns, "00001010" after
30
         40 ns, "00001010" after 60 ns, "00001010" after 80 ns;
      sel <= "00", "00" after 20 ns, "01" after 40 ns, "10"
31
         after 60 ns, "11" after 80 ns, "11" after 100 ns;
32
33
   end arch;
```

https://github.com/mahdihaghverdi/archproject/blob/main/ALU/src/mux4x1/mux4x1\_tb.vhdl

### ۳.۱.۴ خروجی ۳.۱.۴



## فصل ۵ ماژول **ALU**

۱.۵ ماژول ALU ۱.۱.۵ کد این ماژول

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

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

```
68
           input_b => alu_input_b,
69
           diff => full_subber_to_i1_t,
           borrow => alu_carry_borrow_out
70
71
           );
72
       full_subber_to_i1 <= "0000" & full_subber_to_i1_t;</pre>
73
74
       inner_multiplier : entity work.FourBitMultiplier port map (
75
           input_a => alu_input_a,
           input_b => alu_input_b,
76
77
          output => multiplier_to_i2
       );
78
79
       inner_mux : entity work.mux4x1 port map (
80
           i0 => full_adder_to_i0,
81
82
          i1 => full_subber_to_i1,
83
          i2 => multiplier_to_i2,
           i3 => "111111111",
84
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);
      signal carry_borrow_out : std_logic;
11
      signal sel
                           : std_logic_vector (1 downto 0);
12
13
14 begin
15
      UUT : entity work.alu port map (
16
          alu_input_a => input_a,
```

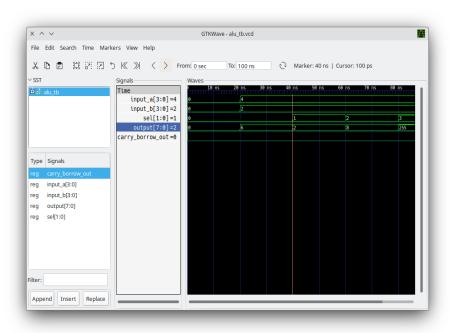
ALUفصل  $\Omega$ . ماژول

```
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;</pre>
      input_b <= "0000", "0010" after 20 ns;</pre>
35
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

فصل ۵. ماژول ALU

#### simulation خروجی ۳.۱.۵



## ۲.۵ طراحیها

