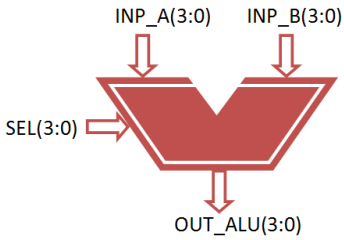
توضیحات آزمایش ۹

هدف از این آزمایش طراحی واحد محاسبه و منطق است که بتواند عملیات های حسابی و منطقی را انجام دهد. واحد محاسبه و منطق طراحی شده باید بتواند عملیات های زیر را انجام دهد.



Selection			Output	Operation
s2	s1	s0		
0	0	0	F = A+1	Increment A
0	0	1	F = A-1	Decrement A
0	1	0	F = A+B	Addition
0	1	1	F = A-B	Subtraction
1	0	0	F = ror A	Rotate Left
1	0	1	F = rol A	Rotate Right
1	1	0	F = A | B	OR
1	1	1	F = A&& B	AND

Defining and using function

```
function function_name (parameter_list) return type is
        declarations
begin
        sequential statements
end function_name;
```

```
4   library ieee;
5   use ieee.std_logic_1164.all;
6   entity function_ex is
7      port(a,b,c:in std_logic;
8           cout:out std_logic);
9   end entity;
10  architecture behavior of function_ex is
11
12  --declaring function of carry
13  function carry(bit1,bit2,bit3:in std_logic) return std_logic is
14  variable result:std_logic;
15  begin
16      result:=(bit1 AND bit2) OR (bit1 AND bit3) OR (bit2 AND bit3);
17  return result;
18  end carry;
19  --end declaring
20
21
22  begin
23  cout<=carry(a,b,c); --(1)
24  end architecture;
```

Packages:

```
package package_name is
        declarations
end package_name;
```

Declarations may typically be any of the following: **type, subtype, constant, file, alias, component, attribute, function, procedure**

```
package DEMO_PACK is
  constant SOME_FLAG : bit_vector := "11111111";
  type STATE is (RESET,IDLE,ACKA);
  component HALFADD
    port(A,B : in bit;
         SUM,CARRY : out bit);
  end component;
end DEMO_PACK;
```

Items declared in a package are visible wherever selected via a **use** clause. For instance, assume DEMO_PACK is analysed into library work:

```
use work.DEMO_PACK.all;
entity DEMO is
port
 (Z: out bit_vector(7 downto 0));
end DEMO;

architecture BEHAVE of DEMO is
begin
  Z <= SOME_FLAG;
end BEHAVE;
```