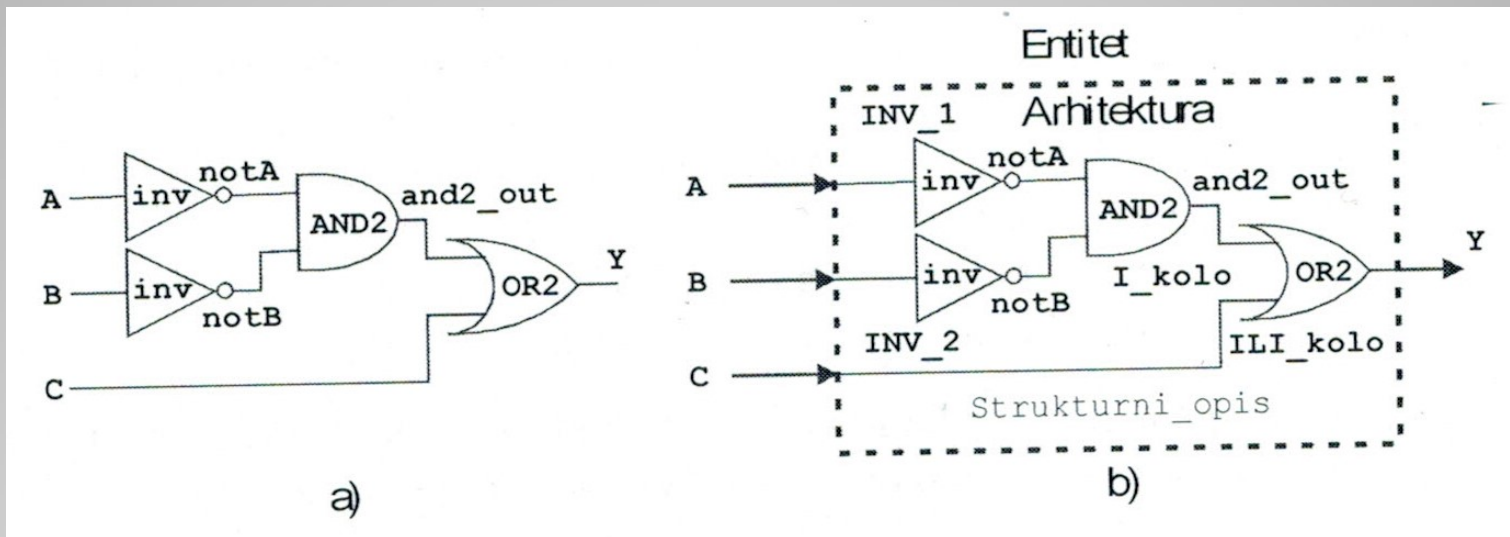


Na slici je prikazana strukturna šema kola koje je potrebno isprojektovati. Koristiti strukturni stil opisa, i kao pomoć koristiti sliku na desnoj strani.



Najpre je potrebno opisati par entitet-arhitektura za svaku komponentu, zatim deklarirati odgovarajuće parove entitet-arhitektura komponentama u okviru arhitekture projekta u kome će se koristiti i na kraju te komponente instancirati (uneti) ih u opis arhitektur projekta.

Tabela istinitosti za kolo koje je potrebno isporojektovati:

A	B	C	Y
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Najpre je potrebno opisati kao zasebne vhd module svaki par entitet-arhitektura koji će se kasnije koristiti kao komponente. Dakle, potrebno je napisati vhd module za Invertor, logičko I i logičko ILI kolo.

INV kolo:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity INV is
    Port ( i : in  STD_LOGIC;
          o : out STD_LOGIC);
end INV;

architecture Behavioral of INV is

begin
    o<=NOT i;

end Behavioral;
```

AND kolo:

```
library IEEE;  
use IEEE.STD_LOGIC_1164.ALL;  
entity AND2 is  
    Port ( i1 : in  STD_LOGIC;  
          i2 : in  STD_LOGIC;  
          y  : out STD_LOGIC);  
end AND2;  
  
architecture Behavioral of AND2 is  
begin  
y<='1' when i1='1' and i2='1' else '0';  
  
end Behavioral;
```

OR kolo:

```
library IEEE;  
use IEEE.STD_LOGIC_1164.ALL;  
entity OR2 is  
    Port ( i1 : in  STD_LOGIC;  
           i2 : in  STD_LOGIC;  
           y : out  STD_LOGIC);  
end OR2;
```

architecture Behavioral of OR2 is

```
begin  
y<='1' when i1='1' or i2='1' else '0';
```

```
end Behavioral;
```

Na kraju je potrebno napisati strukturni opis glavnog projekta na osnovu šeme date na prvom slajdu:

```
library IEEE;  
use IEEE.STD_LOGIC_1164.ALL;  
use work.INV;  
USE work.and2;  
use work.or2;  
entity Projekat is  
    Port ( A : in  STD_LOGIC;  
          B : in  STD_LOGIC;  
          C : in  STD_LOGIC;  
          Y : out STD_LOGIC);  
end Projekat;
```

```
architecture Behavioral of Projekat is  
    component INV  
    port ( i : in  STD_LOGIC;  
          o : out STD_LOGIC);  
    end component INV;
```

```

component OR2
  Port ( i1 : in  STD_LOGIC;
         i2 : in  STD_LOGIC;
         y : out  STD_LOGIC);
end component OR2;
component AND2
  Port ( i1 : in  STD_LOGIC;
         i2 : in  STD_LOGIC;
         y : out  STD_LOGIC);
end component AND2;

signal notA, notB, and2_out: std_logic;
begin
INV_1: INV port map
                                (i=>A,
                                 o=>notA);

INV_2: INV port map
                                (i=>B,
                                 o=> notB);

I_kolo: AND2 port map
                                (i1 => not A, i2=> not B, y=> and2_out);

ILI_kolo: OR2 port map
                                (i1=> and2_out, i2=> C, y=> Y);

end Behavioral;

```