

0 = 01

1 = 00

2 = 10

3 = 11

a1a0	b1b0	Cin	r1r0	cond
1 00	1 00	0	✓ 1 1	0
00	00	1	✓ 3 1 1	0
00	0 1	0	1 00	0
00	0 1	1	✓ 2 1 0	0
00	1 0	0	✓ 3 1 1	0
00	1 0	1	0 0 1	1 ✓
00	1 1	0	0 0 1	1 ✓
00	1 1	1	1 00	1 ✓
0 01	1 00	0	1 00	0
0 1	00	1	✓ 2 1 0	0
0 1	0 1	0	0 0 1	0
0 1	0 1	1	1 00	0
0 1	1 0	0	✓ 2 1 0	0
0 1	1 0	1	✓ 3 1 1	0
0 1	1 1	0	✓ 2 1 1	0
0 1	1 1	1	0 0 1	1 ✓
1 10	1 00	0	✓ 3 1 1	0
1 0	00	1	0 0 1	1 ✓
1 0	0 1	0	✓ 2 1 0	0
1 0	0 1	1	✓ 2 1 1	0
1 0	1 0	0	0 0 1	1 ✓
1 0	1 0	1	1 00	1 ✓
1 0	1 1	0	1 00	1 ✓
1 0	1 1	1	✓ 2 1 0	1 ✓
1 1	1 00	0	0 0 1	1 ✓
1 1	00	1	1 00	1 ✓
1 1	0 1	0	✓ 3 1 1	0
1 1	0 1	1	0 0 1	1 ✓
1 1	1 0	0	1 00	1 ✓
1 1	1 0	1	✓ 2 1 0	1 ✓
1 1	1 1	0	✓ 3 1 1	1 ✓
1 1	1 1	1	✓ 3 1 1	1 ✓

$$r1 = \sum (0, 1, 3, 4, 9, 12, 13, 14, 16, 19, 19, 23, 26, 29, 30, 31)$$

$$r0 = \sum (1, 4, 5, 6, 10, 13, 14, 15, 16, 17, 19, 20, 24, 26, 27, 31)$$

$$cond = \sum (5, 6, 7, 15, 17, 20, 21, 22, 25, 24, 25, 27, 28, 29, 30, 31)$$

$$r1 = \sum m(0, 1, 3, 4, 9, 12, 13, 14, 16, 18, 19, 23, 26, 29, 30, 31)$$

		a	b	b1
	000	001	011	010 110 111 101 100
b0cin	00	1	1	1
	01	1	1	1
	11	1	1	1
	10	1	1	1

$$r1 = \overline{a1} \overline{a0} \overline{b0} cin + \overline{a1} \overline{a0} b1 cin + \overline{a1} a0 b1 \overline{cin} + \overline{a1} a0 \overline{b0} cin +$$

$$+ a1 a0 b0 \overline{cin} + a1 a0 b1 cin + a1 a0 b0 cin + a1 a0 b1 \overline{cin}$$

$$r0 = \sum m(1, 4, 5, 6, 10, 13, 14, 15, 16, 17, 19, 20, 24, 26, 27, 31)$$

		a	a0	b1
	000	001	011	010 110 111 101 100
b0cin	00	1	1	1
	01	1	1	1
	11	1	1	1
	10	1	1	1

$$r0 = \overline{a1} \overline{a0} \overline{b1} \overline{b0} + \overline{a1} \overline{a0} \overline{b0} cin + \overline{a1} a0 b1 cin + \overline{a1} b1 b0 cin +$$

$$+ a0 b1 b0 cin + a1 a0 \overline{b1} cin + a1 a0 b0 cin + a1 a0 b0 cin +$$

$$+ a1 a0 b1 cin$$

$$cout = \sum m(5, 6, 7, 15, 17, 20, 21, 22, 23, 24, 25, 27, 28, 29, 30, 31)$$

		a	a0	b1
	000	001	011	010 110 111 101 100
b0cin	00	1	1	1
	01	1	1	1
	11	1	1	1
	10	1	1	1

$$cout = a1 b1 + a1 b0 b0 + a1 a0 cin + a1 b0 cin + a1 b1 b0 cin +$$

$$+ a1 a0 b1 cin + a1 a0 b1 b0$$

PORTAL 2023/2024

library IEEE;
use IEEE.Std_Logic_1164.All;

entity FA is port (
a, b: in std_logic_vector (1 downto 0);
cin: in std_logic;
r: out std_logic_vector (1 downto 0);
cout: out std_logic);
end FA;

architecture PORTAL of FA is

BEGIN

r(0) <= (((not a(1) and not a(0) and b(1) and not b(0)) or (not c(1) and not a(0)
and not b(0) and cin) or (not a(1) and a(0) and b(1) and cin) or
(not a(1) and b(1) and b(0) and not cin) or (a(0) and not b(1) and b(0) and
not cin) or (a(1) and a(0) and not b(1) and not cin) or (a1 and a0
and b0 and cin) or (a1 and not a(0) and not b(0) and not cin) or
(a(1) and not a(0) and not b(1) and cin))) after 10ns;

r(1) <= (((not a(1) and not a(0)) and ((not b(0) and not cin) or (not b(1) and cin)))
or ((not a(1) and a(0)) and ((b(1) and not cin) or (not b(0) and cin))) or
((a(1) and a(0)) and ((b(0) and not cin) or (b(1) and cin))) or
((a(1) and not a(0)) and ((b(0) and cin) or (not b(1) and not cin))))
after 10ns;

$cout \leq ((a(1) \text{ and } (b(1) \text{ or } (a(0) \text{ and not } b(0))) \text{ or } (a(0) \text{ and } cin) \text{ or } (\text{not } b(0) \text{ and } cin))) \text{ or } (\text{not } a(1) \text{ and } ((b(1) \text{ and } b(0) \text{ and } cin) \text{ or } (\text{not } a(0) \text{ and } b(1) \text{ and } cin) \text{ or } (\text{not } a(0) \text{ and } b(1) \text{ and } b(0)))) \text{ after } 10 \text{ ns};$

end ponasajna;

3-komplement

$x_1 x_0$	$y_1 y_0$
01	11
00	10
10	00
11	01

$$y_1 = \overline{x_1}$$

$$y_0 = x_0$$

library IEEE;

use IEEE.std_logic_1164.ALL;

entity b1kompl is port (
 x: in std_logic_vector (1 downto 0);
 y: out std_logic_vector (1 downto 0);
 end b1kompl;

architecture ponasajna of b1kompl is

BEGIN

 y(1) <= not x(1) after 10 ns;

 y(0) <= x(0) after 10 ns;

end ponasajna;

2-2-2-2-2-2-2-2

$x_1 x_0 y_1 y_0 s$	$z_1 z_0$
0 0 1 0 0	0 0
0 0 1 0 1	1 0
0 1 1 1 0	0 1
0 1 1 1 1	1 1
1 0 0 0 0	1 0
1 0 0 0 1	0 0
1 1 0 1 0	1 1
1 1 0 1 1	0 1

$$z_1 = (x_1 \cdot \bar{s}) + (y_1 \cdot s)$$

$$z_0 = (x_0 \cdot \bar{s}) + (y_0 \cdot s)$$

library IEEE;

use IEEE.std_logic_1164.ALL;

entity dmux is port (
 x, y: in std_logic_vector (1 downto 0);
 s: in std_logic_vector;
 z: out std_logic_vector (1 downto 0);
 end dmux;

architecture ponasajna of dmux is

BEGIN

 z(1) <= ((x(1) and not s) or (y(1) and s)) after 10 ns;

 z(0) <= ((x(0) and not s) or (y(0) and s)) after 10 ns;

end ponasajna;

PRIMITIV

library IEEE;

use IEEE.std_logic_1164.ALL;

entity PRIMITIV is port (
 a, b: in std_logic_vector (1 downto 0);
 cin, oper: in std_logic;
 r: out std_logic_vector (1 downto 0);
 cout: out std_logic;
 end primitiv;

architecture STRUKTURA of primitiv is

 Signal m, n: std_logic_vector (1 downto 0);

BEGIN

 kompl: entity work.b1kompl port map (b, n);

 mux: entity work.dmux port map (b, n, oper, m);

```
fa: entity work.fa port map (a, m, cm, r, cout);  
end STRUKTURNA;
```

4-BITNO ZBRAJALO

```
library IEEE;  
use IEEE.std_logic_1164.all;
```

```
entity ZBRAJALO is port(  
  a: in std_logic_vector (7 downto 0);  
  b: in std_logic_vector (7 downto 0);  
  oper: in std_logic;  
  r: out std_logic_vector (7 downto 0);  
  cout: out std_logic);  
end ZBRAJALO
```

```
architecture STRUKTURNA of zbrajalo is  
  signal c1, c2, c3, c4: std_logic;  
  BEGIN
```

```
    prim1: entity work.primitiv port map (a (1 downto 0), b (1 downto 0),  
                                             oper, oper, r (1 downto 0), c1);  
    prim2: entity work.primitiv port map (a (3 downto 2), b (3 downto 2),  
                                             c1, oper, r (3 downto 2), c2);  
    prim3: entity work.primitiv port map (a (5 downto 4), b (5 downto 4),  
                                             c2, oper, r (5 downto 4), c3);  
    prim4: entity work.primitiv port map (a (7 downto 6), b (7 downto 6), c3,  
                                             oper, r (7 downto 6), cout);
```

```
  end STRUKTURNA;
```

2. Zadatak

Zadatak ove vježbe je izgradnja 4-znamenkastog zbrajala/oduzimala brojeva u bazi 4, ovisno o upravljačkom ulazu *oper*. Ako je *oper*=0, potrebno je obaviti zbrajanje, a ako je *oper*=1, potrebno je obaviti oduzimanje. Da biste mogli riješiti zadatak, potrebno je još definirati način kodiranja znamenaka. Na kraju ovog dokumenta nalazi se tablica s propisanim kodiranjem znamenaka, ovisno o vašem JMBAG-u, kao i naputak kako otkriti koju tablicu kodiranja trebate koristiti.

2.1. Dodijeljeni način kodiranja

U tablicu u nastavku čitko prepisite Vama dodijeljeni kod.

Znamenka	Kodna riječ
0	01
1	00
2	10
3	11

2.2. Izrada potpunog zbrajala

Projektirajte sklop potpuno zbrajalo. Konkretno, promatrajte izlaze c_{out} , r_1 i r_0 kao funkcije od a_1 , a_0 , b_1 , b_0 te c_{in} (lokalne oznake ulaza sklopa FA). Poslužite se tabličnim prikazom. Kako glasi minimalni oblik tih funkcija? Upišite ih u sljedeću tablicu.

$$c_{out}(a_1, a_0, b_1, b_0, c_{in}) = a_1 b_1 + a_1 a_0 \overline{b_0} + a_1 a_0 c_{in} + a_1 \overline{b_0} c_{in} + a_1 b_1 b_0 c_{in} + a_1 \overline{a_0} b_1 c_{in} + a_1 a_0 \overline{b_1} b_0$$

$$r_1(a_1, a_0, b_1, b_0, c_{in}) = \overline{a_1} \overline{a_0} \overline{b_0} c_{in} + \overline{a_1} \overline{a_0} b_1 c_{in} + \overline{a_1} a_0 b_1 c_{in} + \overline{a_1} a_0 \overline{b_0} c_{in} + a_1 a_0 \overline{b_0} c_{in} + a_1 a_0 b_1 c_{in} + a_1 \overline{a_0} b_0 c_{in} + a_1 \overline{a_0} b_1 c_{in}$$

$$r_0(a_1, a_0, b_1, b_0, c_{in}) = \overline{a_1} \overline{a_0} b_1 \overline{b_0} + \overline{a_1} \overline{a_0} \overline{b_0} c_{in} + \overline{a_1} a_0 b_1 c_{in} + \overline{a_1} b_1 \overline{b_0} c_{in} + a_0 \overline{b_1} \overline{b_0} c_{in} + a_1 a_0 \overline{b_1} c_{in} + a_1 a_0 b_0 c_{in} + a_1 \overline{a_0} \overline{b_0} c_{in} + a_1 \overline{a_0} b_1 c_{in}$$

Na zaseban list papira napišite VHDL model ovog sklopa. Sklop nazovite *FA*. U sučelju koristite signale *a*, *b* (vektor od 1 do 0), *cin* (skalar), *r* (vektor od 1 do 0) te *cout* (skalar) – upravo tim redosljedom. Svi izlazi trebaju kasniti 10 ns.

2.3. Izrada sklopa za izračun (B-1)-komplementa

Projektirajte sklop koji računa 3-komplement. Ulaze sklopa označite s x_1 i x_0 , izlaze s y_1 i y_0 . Izlaze promatrajte kao funkcije od ulaza, i poslužite se tabličnim prikazom. Kako glase minimalni oblici izlaznih funkcija? Upišite ih u tablicu u nastavku.

$$y_1(x_1, x_0) = \overline{x_1}$$

$$y_0(x_1, x_0) = x_0$$

Na zaseban list papira napišite VHDL model ovog sklopa. Sklop nazovite *b1kompl*. U sučelju koristite signal x (vektor od 1 do 0) te y (vektor od 1 do 0) – upravo tim redosljedom. Svi izlazi trebaju kasniti 10 ns.

2.4. Izrada dvostrukog multipleksora

Projektirajte dvostruki multipleksor. Podatkovne ulaze sklopa označite s x_1 , x_0 , y_1 te y_0 , selekcijski ulaz sa s , a izlaze sa z_1 i z_0 . Izlaze promatrajte kao funkcije od ulaza, i poslužite se tabličnim prikazom. Kako glase minimalni oblici izlaznih funkcija? Upišite ih u tablicu u nastavku.

$$z_1(x_1, x_0, y_1, y_0, s) = (x_1 \cdot \overline{s}) + (y_1 \cdot s)$$

$$z_0(x_1, x_0, y_1, y_0, s) = (x_0 \cdot \overline{s}) + (y_0 \cdot s)$$

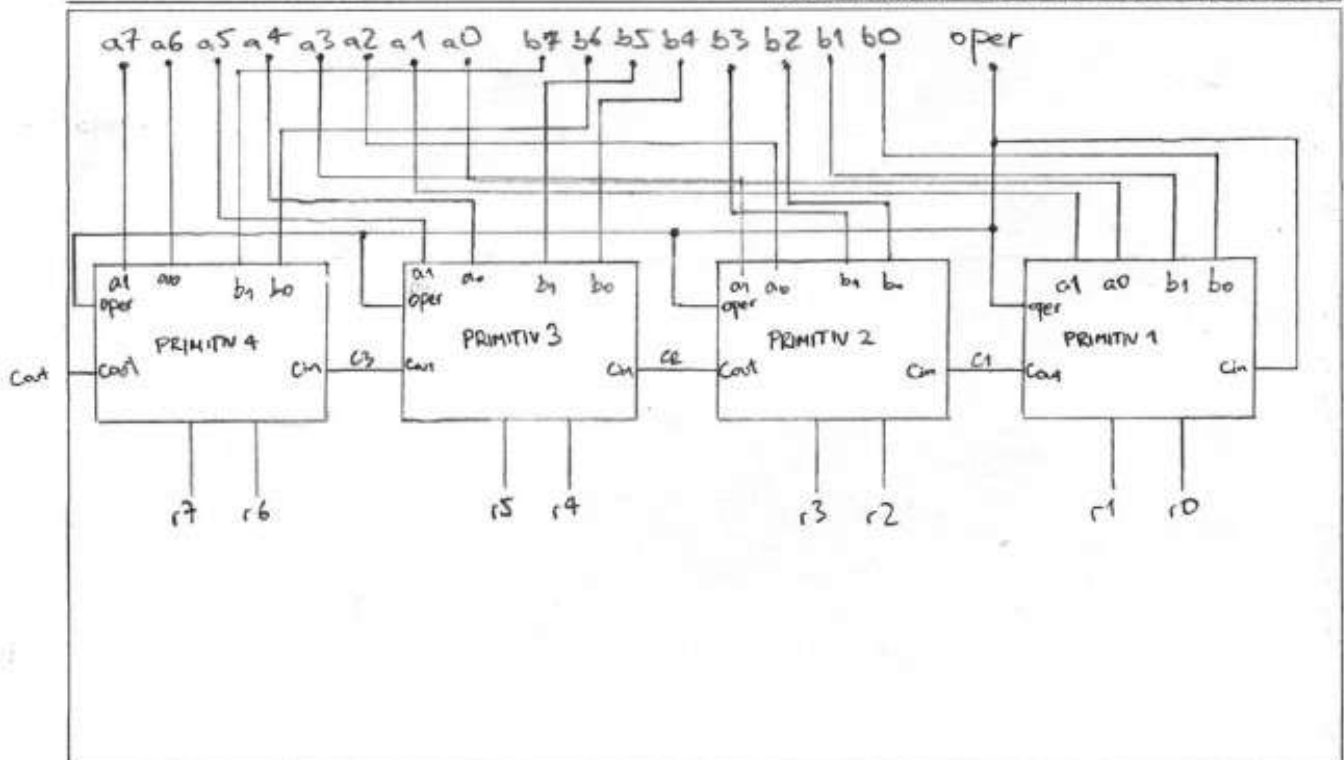
Na zaseban list papira napišite VHDL model ovog sklopa. Sklop nazovite *dmux*. U sučelju koristite signal x (vektor od 1 do 0), y (vektor od 1 do 0), s (skalar) te z (vektor od 1 do 0) – upravo tim redosljedom. Svi izlazi trebaju kasniti 10 ns.

2.5. Izrada primitiva

Na zaseban list papira napišite VHDL model sklopa *primitiv*. Napišite strukturni model koristeći se prethodno razvijenim sklopovima.

2.6. Izrada 4-znamenkastog zbrajala/oduzimala

Koristeći pojednostavljeni simbol primitiva prikazan na početku upute, u prazno polje u nastavku nacrtajte shemu 4-znamenkastog zbrajala/oduzimala (gdje je točna operacija definirana ulazom *oper*). Na shemi jasno označite nazive primjeraka primitiva koje ćete koristiti pri strukturnom modeliranju, kao i nazive internih signala koje ćete koristiti.



Na zaseban list papira napišite VHDL model ovog sklopa. Sklop nazovite *zbrajalo*. U sučelju koristite signal *a* (vektor od 7 do 0), *b* (vektor od 7 do 0), *oper* (skalar) te *r* (vektor od 7 do 0) i *cout* (skalar) – upravo tim redoslijedom. Uočite da ovaj sklop nema ulaz *c_{in}*. – zašto? Razmislite!

Prije dolaska na laboratorijske vježbe potrebno je riješiti sve prethodno navedene zadatke (projektiranje sklopova te njihov opis u jeziku VHDL), kao i popuniti tablicu u poglavlju 2.8. Na laboratorijskim vježbama provjerite još i sljedeće:

- prilikom zbrajanja brojeva, u kojem slučaju dolazi do najdužeg kašnjenja do pojave ispravnog rezultata (tzv. vrijeme stabilizacije rezultata)?
- koliko iznosi to vrijeme?

2.7. Priprema

Na samu vježbu potrebno je donijeti ovu uputu u kojoj sva polja za unos rješenja trebaju biti popunjena, papire s modelima traženih sklopova opisanih jezikom VHDL te papire korištene tijekom projektiranja samih sklopova (gdje su vidljive tražene tablice i sam postupak minimizacije). Sva rješenja (uključivo VHDL kod) trebaju biti napisana rukom, običnom ili kemijskom olovkom. Na vrhu svake stranice potrebno je kemijskom olovkom napisati prezime, ime te matični broj studenta.

Alternativno, umjesto pisanja VHDL modela rukom na papiru, moguće je kompletnu vježbu prije dolaska na sam termin vježbe napraviti kroz vhdllab koristeći Vaše korisničko ime i zaporku s Ferka (*naglasak na čitavu vježbu zajedno s ispitnim sklopovima i simulacijama; u tom slučaju asistent će pogledati vježbu koju ste tako napravili*).

2.8. Provjera rada sklopa

Za Vama dodijeljeni kod, što će biti na izlazu 4-znamenkastog zbrajala/oduzimala za ulaze iz sljedeće tablice? Popunite tablicu prije dolaska na vježbu!

<i>oper</i>	<i>a</i> _{3 2 1 0}	<i>b</i> _{3 2 1 0}	<i>r</i> _{3 2 1 0}	<i>cout</i>
0	11100100	01000010	11110011	0
0	11100100	01001010	11111011	0
1	11100100	01000010	11011011	1
1	11100100	01001010	11010011	1

2.9. Pitanja za vježbu

1. Kako izgleda sklop *primitiv* ako se radi o zbrajanju/oduzimanju binarnih brojeva (dakle, kada je B=2)? U što tada degenerira multipleksor i sklop za izračun komplementa?
2. Ako razmatramo znamenke baze 10, konkretno, BCD kod, kako tada možemo ostvariti sklop FA? Isplati li se tada sklop projektirati tablično ili imamo bolje rješenje? Kako tada izgleda sklop za izračun 9-komplementa?
3. Ako razmatramo znamenke baze 10, konkretno, Excess-3 kod, kako tada možemo ostvariti sklop FA? Isplati li se tada sklop projektirati tablično ili imamo bolje rješenje? Kako tada izgleda sklop za izračun 9-komplementa?
4. Usporedite rješenja 2. i 3. zadatka. Što daje jednostavnije sklopovlje?

Literatura:

- [1] Peruško, Glavinić: *Digitalni sustavi*. Školska knjiga, 2005.
- [2] Čupić, *Digitalna elektronika i Digitalna logika. Zbirka riješenih zadataka*. Kigen, 2006.