Kodovi za 5. labos – inačica 7

B-1 Komplement

```
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 arch OF b1kompl IS
BEGIN
y(1) <= x(1) after 10 ns;
y(0) <= not x(0) after 10 ns;
END arch;
```

Dvostruki multipleksor

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

ENTITY dmux IS PORT (
x: IN std_logic_vector (1 downto 0);
y: IN std_logic_vector (1 downto 0);
s: IN std_logic;
z: OUT std_logic_vector (1 downto 0));
END dmux;

ARCHITECTURE arch OF dmux IS
BEGIN
z(1)<= ((not s and x(1)) OR (s and y(1))) after 10 ns;
z(0)<= ((not s and x(0)) OR (s and y(0))) after 10 ns;
END arch;
```

Potpuno zbrajalo

```
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 arch OF FA IS
BEGIN
r(1) <= (cin and a(1) and a(0) and b(1) and b(0)) or (cin and not a(1) and a(0) and not b(1) and b(0)) or
(not cin and a(1) and not a(0) and b(1) and not b(0)) or (not cin and not a(1) and not a(0) and not b(1)
and not b(0)) or (not cin and not a(1) and a(0) and b(1)) or (not cin and a(1) and a(0) and not b(1)) or
(not a(1) and not a(0) and b(1) and b(0)) or (a(1) and not a(0) and not b(1) and b(0)) or (cin and not
a(1) and b(1) and not b(0)) or (cin and a(1) and not b(1) and not b(0)) after 10 ns;
r(0) <= (not cin and a(0) and b(0)) or (cin and not a(0) and b(0)) or (cin and a(0) and not b(0)) or (not
cin and not a(0) and not b(0)) after 10 ns;
cout <= (cin and not a(1) and not a(0)) or (not a(1) and not a(0) and b(1)) or (cin and not b(1) and not
b(0)) or (a(1) and not b(1) and not b(0)) or (a(1) and a(0) and b(1) and b(0)) or (not a(0) and not b(1)
and not b(0)) or (cin and not a(0) and b(1) and b(0)) or ( cin and a(1) and a(0) and not b(0)) after 10
ns;
END arch;
```

Primitiv

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

ENTITY primitiv IS PORT(
a: IN std_logic_vector (1 downto 0);
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 arch OF primitiv IS
signal i,j:std_logic_vector (1 downto 0);
BEGIN
sklop1: ENTITY work.b1kompl PORT MAP(b(1 downto 0),i(1 downto 0));
sklop2: ENTITY work.dmux PORT MAP (b(1 downto 0),i(1 downto 0),oper,j(1 downto 0));
sklop3: ENTITY work.FA PORT MAP (a(1 downto 0),j(1 downto 0),cin,r(1 downto 0),cout);
END arch;
```

4-znamenkasto zbrajalo/oduzimalo

```
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 arch OF zbrajalo IS
signal i1,i2,i3: std_logic;
BEGIN
c0: ENTITY work.primitiv PORT MAP (A(1 downto 0),B(1 downto 0),oper,oper,R(1 downto 0),i1);
c1: ENTITY work.primitiv PORT MAP (A(3 downto 2),B(3 downto 2),i1,oper,R(3 downto 2),i2);
C2: ENTITY work.primitiv PORT MAP (A(5 downto 4),B(5 downto 4),i2,oper,R(5 downto 4),i3);
c3: ENTITY work.primitiv PORT MAP (A(7 downto 6),B(7 downto 6),i3,oper,R(7 downto 6),cout);
END arch;
```

Kodiranje znamenaka za inačicu 7 u trenutku pisanja ovog dokumenta:

Znamenka	Kodna riječ
1	**01**
2	**10**
3	**11**
4	**00**