LAB 5 – VERZIJA 2

(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) <= not x(1) after 10 ns;
  y(0) <= 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) <= ( 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 arch;
```

Full Adder

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

ENTITY FA IS PORT(
   a: in std_logic_vector (1 downto 0);
   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

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

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

   r(0) <= ( (a(0) and b(1) and not b(0) and cin) or (not a(0) and b(1) and b(0) and cin) or (a(1) and
   a(0) and not b(1) and not b(0)) or (a(1) and not a(0) and not b(1) and b(0)) or (not a(1) and a(0)
   and not b(0) and not cin) or (not a(1) and not a(0) and b(0) and not cin) or (a(1) and a(0) and
   b(1) and b(0) and not cin) or (not a(1) and a(0) and not b(1) and b(0) and cin) or (a(1) and not
   a(0) and b(1) and not b(0) and not cin) or (not a(1) and not a(0) and not b(1) and not b(0) and
   cin) ) 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 : in std_logic;
   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
   b1kompl : ENTITY work.b1kompl PORT MAP (b, i);
   dmux: ENTITY work.dmux PORT MAP (b, i, oper, j);
   FA: ENTITY work.FA PORT MAP (a, j, cin, r, cout);
END arch;
```

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 arch OF zbrajalo IS

  SIGNAL c : std_logic_vector (2 downto 0);

BEGIN

  prim1 : ENTITY work.primitiv PORT MAP (a(1 downto 0), b(1 downto 0), oper, oper, r(1 downto 0), c(0));
  prim2 : ENTITY work.primitiv PORT MAP (a(3 downto 2), b(3 downto 2), c(0), oper, r(3 downto 2), c(1));
  prim3 : ENTITY work.primitiv PORT MAP (a(5 downto 4), b(5 downto 4), c(1), oper, r(5 downto 4), c(2));
  prim4 : ENTITY work.primitiv PORT MAP (a(7 downto 6), b(7 downto 6), c(2), oper, r(7 downto 6), cout);

END arch;
```