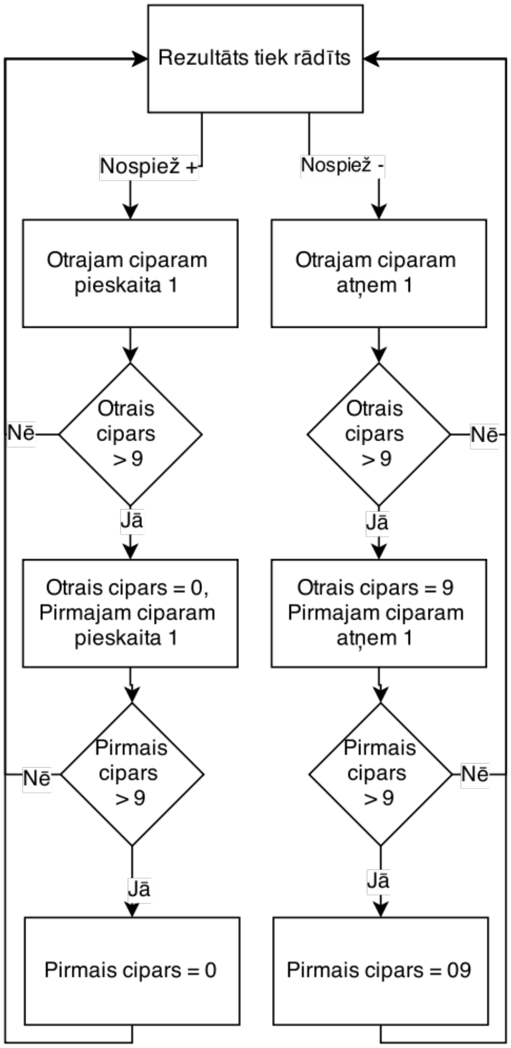
Linarda Jukmaņa

2. Laboratorijas darbs
Funkcionālo un loģisko shēmu modelēšanā
“BCD skaitītājs”

Laboratorijas darba ietvaros tika izstrādāts divciparu decimāls skaitītājs. Autors piedāvā darbības blokshēmu un izmantoto kodu.



```
entity TEST is

     port
     (
          plus       : in std_logic;
          minus      : in std_logic;
          clock      : in std_logic;
          data       : out std_logic_vector(7
downto 0);
          dig_leds   : out std_logic_vector(7
downto 0);
          dig1_gnd : out std_logic;
          dig2_gnd : out std_logic
     );

end entity;

architecture rtl of TEST is

     type DECODE_ARRAY is array (0 to 15) of
std_logic_vector(0 to 7);
     constant segment_values :
DECODE_ARRAY := ( X"3F", X"06", X"5B", X"4F",
X"66", X"6D", X"7D", X"07", X"7F", X"6F",
X"77", X"7C", X"39", X"5E", X"79", X"71");

     shared variable bcd0             :
std_logic_vector(3 downto 0) := X"0";
     shared variable bcd1             :
std_logic_vector(7 downto 4) := X"0";
     shared variable tmp_dig_leds  :
std_logic_vector(7 downto 0) := X"00";
     shared variable act_dig1         : std_logic := '0';

     signal clocker                   : integer := 0;
     signal clocker2                  : integer := 0;
begin

     process(plus, minus, clock)

     begin
          if(rising_edge(clock)) then
               if(clocker < 5000000) then
                    clocker <= clocker + 1;
               else
                    clocker <= 0;
                    if(plus = '0') then
                    bcd0 := bcd0 + 1;
                elsif(minus = '0') then
                    bcd0 := bcd0 - 1;
                end if;

                if(bcd0 > X"09") then
                    if(plus = '0') then
                        bcd1 := bcd1 + 1;
                        bcd0 := X"0";
                    else
                        bcd1 := bcd1 - 1;
                        bcd0 := X"9";
                    end if;
                end if;

                if(bcd1 > X"09") then
                    if(plus = '0') then
                        bcd1 := X"0";
                    else
                        bcd1 := X"9";
                    end if;
                end if;
            end if;
        end if;
    end process;

    process(clock)
        variable temp : integer := 0;

        begin
        if(rising_edge(clock)) then
            if(clocker2 < 50000) then
                clocker2 <= clocker2 + 1;
            else
                clocker2 <= 0;

                act_dig1 := not act_dig1;
                if(act_dig1 = '1') then
                    temp := to_integer(unsigned(bcd1));
                else
                    temp := to_integer(unsigned(bcd0));
                end if;

                tmp_dig_leds := segment_values(temp);
            end if;
        end if;

    end process;

    dig1_gnd <= not act_dig1;
    dig2_gnd <= act_dig1;
    dig_leds <= not tmp_dig_leds;   -- Common anode
    data <= bcd1 & bcd0;

end rtl;
```