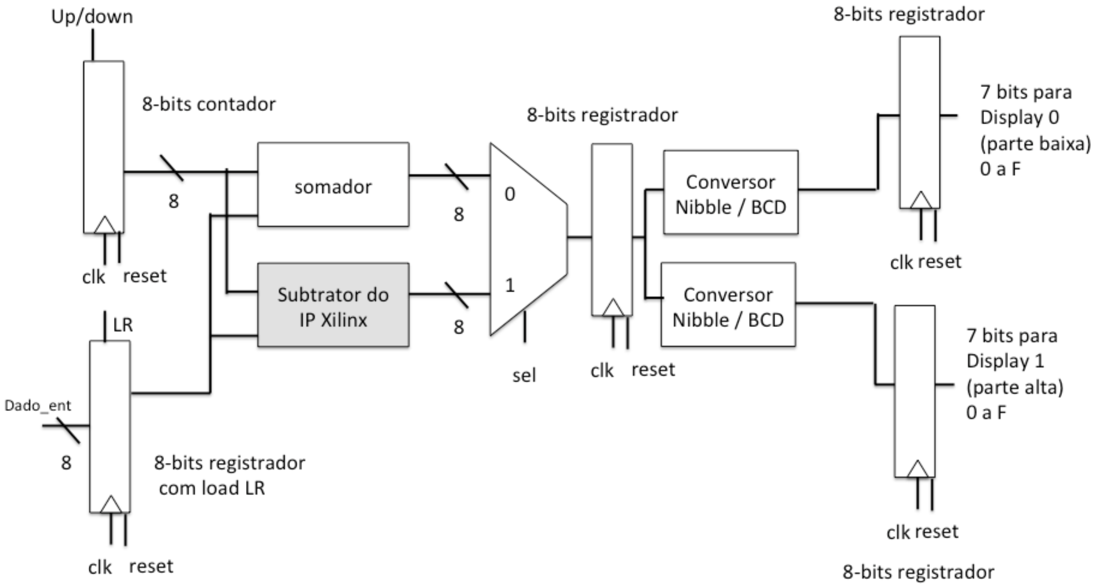
Trabalho 1 – Sistemas Digitais – 2020-2

Nome: Cleiber Gustavo Soares Rodrigues Junior___________________________
Matrícula: 00270139_____________________________

1) Descreva o seguinte circuito digital em VHDL na ferramenta Vivado



No **testbench:**

- o dado de entrada (dado_ent) deve ser os **dois dígitos menos significativos** da matricula descritos em binário (Exemplo: 00221**34** seria 34 em binário).

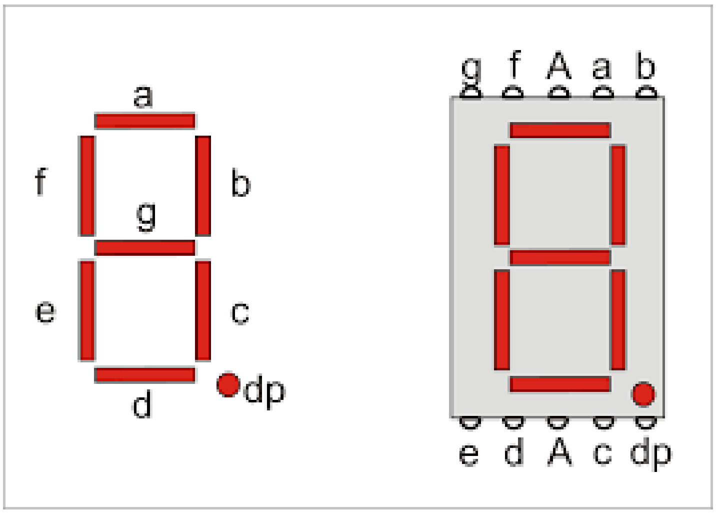
No **VHDL**:

- Registrador contador é inicializado com os **dois dígitos mais significativos** da matricula descritos em binário (Exemplo: **00**22134 seria 00 em binário), quando o reset='1', e pode contar para cima e para baixo (Up/Down).

- Os demais registradores devem ser inicializados em ZERO quando (reset='1').

Conversor:

- Conversor de Nibble para 7seg é uma grande tabela verdade onde cada entrada de 4 bits define os leds do display 7 segmentos que deve acender para desenha o numero em hexadecimal correspondente. O led é ligado com o valor logico 1.



Nibble				7seg						
N3	N2	N1	N0	a	b	c	d	e	f	g
0	0	0	0							
0	0	0	1							
0	0	1	0							
0	0	1	1							
0	1	0	0							
0	1	0	1							
0	1	1	0							
0	1	1	1							
1	0	0	0							
1	0	0	1							
1	0	1	0							
1	0	1	1							
1	1	0	0							
1	1	0	1							
1	1	1	0							
1	1	1	1							

Cole o VHDL completo aqui:

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 06.03.2021 22:30:53
-- Design Name:
-- Module Name: trabalho1 - Behavioral
-- Project Name:
-- Target Devices:
-- Tool Versions:
-- Description:
--
```

```
-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
----------------------------------------------------------------------------------


library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity trabalho1 is
port ( Up_down : IN STD_LOGIC; -- Foi
      Rst, Clk: IN STD_LOGIC; -- Acho que foi tbm
      LR_entrada: in STD_LOGIC; --
      Dado_entrada : in STD_LOGIC_VECTOR(7 DOWNTO 0); --Foi
      seletor : in STD_LOGIC; -- Foi
      out_1 : out STD_LOGIC_VECTOR(7 DOWNTO 0);
      out_2 : out STD_LOGIC_VECTOR(7 DOWNTO 0));
end trabalho1;

architecture Behavioral of trabalho1 is

-- Aqui comecamos a importar os componentes
-- DONE
component reg8bitsLR is
port ( D : IN STD_LOGIC_VECTOR(7 DOWNTO 0) ;
      Resetn, Clock: IN STD_LOGIC ;
      LR: in STD_LOGIC;
      Q : OUT STD_LOGIC_VECTOR(7 DOWNTO 0) );
end component;
--DONE
component cont8bits is
   Port  ( clk : in STD_LOGIC;
         rst : in STD_LOGIC;
         updown : in STD_LOGIC;
         contador : out STD_LOGIC_VECTOR (7 downto 0));
end component;
-- DONE
```

```
component somador8bits is
   Port ( A : in STD_LOGIC_VECTOR (7 downto 0);
        B : in STD_LOGIC_VECTOR (7 downto 0);
        soma : out STD_LOGIC_VECTOR (7 downto 0));
end component;
-- DONE
component sub8bits_xilinx is
   Port ( A : in STD_LOGIC_VECTOR (7 downto 0);
        B : in STD_LOGIC_VECTOR (7 downto 0);
        sub : out STD_LOGIC_VECTOR (7 downto 0));
end component;
-- DONE
component mux2p1 is
   Port ( Somador : in STD_LOGIC_VECTOR (7 downto 0);
        Subtrator : in STD_LOGIC_VECTOR (7 downto 0);
        sel : in STD_LOGIC;
        saida : out STD_LOGIC_VECTOR (7 downto 0));
end component;
-- DONE
component reg8bits is
port ( D : IN STD_LOGIC_VECTOR(7 DOWNTO 0) ;
      Resetn, Clock: IN STD_LOGIC ;
      Q : OUT STD_LOGIC_VECTOR(7 DOWNTO 0) );
end component;
-- IN PROGRESS
component display4saidas is
   Port ( valor_in : in STD_LOGIC_VECTOR (3 downto 0);
        sete_seg : out STD_LOGIC_VECTOR (7 downto 0));
end component;

-- Fim da instanciacao dos componentes

-- Criando os fios que irao estimular o circuito

-- A entrada do registrador vai ser diferente
signal out_op: std_logic_vector(7 downto 0);

-- Fios para o registrador LR
signal reglr_data_out: std_logic_vector(7 downto 0);

-- Fios para o somador, subtrator, mux, regA
signal sum_out, sub_out, mux_out, regA_out, regB_out, regC_out:
std_logic_vector(7 downto 0);

--Fios para os nibbles
signal nibA_out, nibB_out: std_logic_vector(7 downto 0);

begin
```

```
cont: cont8bits
Port map( clk => Clk,
        rst => Rst,
        updown => Up_Down,
        contador => out_op); --Fio que sai do contador

regLR: reg8bitsLR
port map( D => Dado_entrada,
        Resetn => Rst,
        Clock => Clk,
        LR => LR_entrada,
        Q => reglr_data_out );

somador: somador8bits
Port map ( A => out_op,
         B => reglr_data_out,
         soma => sum_out);

subtractor: sub8bits_xilinx
Port map ( A => out_op,
         B => reglr_data_out,
         sub => sub_out);

mux: mux2p1
Port map ( Somador => sum_out,
         Subtrator => sub_out,
         sel => seletor,
         saida => mux_out);

regA: reg8bits
Port map ( D => mux_out ,
         Resetn => Rst,
         Clock => Clk,
         Q => regA_out);

nibbleA: display4saidas
Port map ( valor_in => regA_out(7 downto 4),
         sete_seg => nibA_out);

nibbleB: display4saidas
Port map ( valor_in => regA_out(3 downto 0),
         sete_seg => nibB_out);

regB: reg8bits
Port map ( D => nibA_out ,
         Resetn => Rst,
         Clock => Clk,
         Q => regB_out);
```

```
regC: reg8bits
Port map ( D => nibB_out ,
        Resetn => Rst,
        Clock => Clk,
        Q => regC_out);

out_1 <= nibA_out;
out_2 <= nibB_out;

end Behavioral;


library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity cont8bits is
   Port ( clk : in STD_LOGIC;
        rst : in STD_LOGIC;
        updown : in STD_LOGIC;
        contador : out STD_LOGIC_VECTOR (7 downto 0));
end cont8bits;

architecture Behavioral of cont8bits is

signal cont : std_logic_vector(7 downto 0);

begin

process(clk, rst)
begin
-- Se reset = 0 entao zera o contador
if (rst='0') then
  cont <= (others=>'0'); -- <= "00000000";
-- Quando vem a borda de subida, devemos salvar o conteudo no registrador
elsif rising_edge(clk) then
    CASE updown is
     when '0' => cont <= std_logic_vector(unsigned(cont)+1);
     when others => cont <= std_logic_vector(unsigned(cont)-1);
    end case;
end if;
end process;

-- Aqui o contador parte pra saida
contador <= cont;

end Behavioral;


library ieee ;
use ieee.std_logic_1164.all ;

entity reg8bitsLR is
port ( D : IN STD_LOGIC_VECTOR(7 DOWNTO 0) ;
    Resetn, Clock: IN STD_LOGIC ;
    LR: in STD_LOGIC;
    Q : OUT STD_LOGIC_VECTOR(7 DOWNTO 0) );
end reg8bitsLR;

architecture Behavior of reg8bitsLR is

begin

process ( Resetn, Clock )
begin
if Resetn = '0' then Q <= "00000000";
-- Here we check if register was enable to receive the input
elsif Clock'event and Clock = '1' and LR ='1' then Q <= D;
end if;
end process;

end Behavior;

----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 06.03.2021 14:26:49
-- Design Name:
-- Module Name: somador8bits - Behavioral
-- Project Name:
-- Target Devices:
-- Tool Versions:
-- Description:
--
-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
```

```
--
----------------------------------------------------------------------------------


library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity somador8bits is
   Port ( A : in STD_LOGIC_VECTOR (7 downto 0);
          B : in STD_LOGIC_VECTOR (7 downto 0);
          soma : out STD_LOGIC_VECTOR (7 downto 0));
end somador8bits;

architecture Behavioral of somador8bits is

begin

soma <= std_logic_vector(signed(A) + signed(B));

end Behavioral;


----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 06.03.2021 15:35:23
-- Design Name:
-- Module Name: sub8bits_xilinx - Behavioral
-- Project Name:
-- Target Devices:
-- Tool Versions:
-- Description:
--
-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
```

```
----------------------------------------------------------------------------------


library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity sub8bits_xilinx is
   Port ( A : in STD_LOGIC_VECTOR (7 downto 0);
          B : in STD_LOGIC_VECTOR (7 downto 0);
          sub : out STD_LOGIC_VECTOR (7 downto 0));
end sub8bits_xilinx;

architecture Behavioral of sub8bits_xilinx is

-- subtrator XILINX
COMPONENT c_addsub_0
  PORT (
    A : IN STD_LOGIC_VECTOR(7 DOWNTO 0);
    B : IN STD_LOGIC_VECTOR(7 DOWNTO 0);
    S : OUT STD_LOGIC_VECTOR(7 DOWNTO 0)
  );
END COMPONENT;

begin

-- mapeamento pra uso
your_instance_name : c_addsub_0
  PORT MAP (
    A => A,
    B => B,
    S => sub
  );


end Behavioral;


----------------------------------------------------------------------------------
-- Company:
-- Engineer:
```

```
--
-- Create Date: 06.03.2021 15:54:09
-- Design Name:
-- Module Name: mux2p1 - Behavioral
-- Project Name:
-- Target Devices:
-- Tool Versions:
-- Description:
--
-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
----------------------------------------------------------------------------------


library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity mux2p1 is
   Port ( Somador : in STD_LOGIC_VECTOR (7 downto 0);
          Subtrator : in STD_LOGIC_VECTOR (7 downto 0);
          sel : in STD_LOGIC;
          saida : out STD_LOGIC_VECTOR (7 downto 0));
end mux2p1;

architecture Behavioral of mux2p1 is

begin

-- Condicoes do Mux 2 para 1
saida <= Somador when sel='0' else
       Subtrator when sel='1';

end Behavioral;


library ieee ;
```

```
use ieee.std_logic_1164.all ;

entity reg8bits is
port ( D : IN STD_LOGIC_VECTOR(7 DOWNTO 0) ;
      Resetn, Clock: IN STD_LOGIC ;
      Q : OUT STD_LOGIC_VECTOR(7 DOWNTO 0) );
end reg8bits;

architecture Behavior of reg8bits is

begin

process ( Resetn, Clock )
begin
if Resetn = '0' then Q <= "00000000";
elsif Clock'event and Clock = '1' then Q <= D;
end if;
end process;

end Behavior;

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;



-- Definicao do display com 4 saidas
-- Ajeitando isso para que se tenha um comportamento BCD + REG
entity display4saidas is
   Port ( valor_in : in STD_LOGIC_VECTOR (3 downto 0);
         sete_seg : out STD_LOGIC_VECTOR (7 downto 0));
end display4saidas;

architecture Behavioral of display4saidas is

--  Tentando adaptar isso para BCD
signal dado : STD_LOGIC_VECTOR (3 downto 0);
signal out_7_seg : STD_LOGIC_VECTOR (7 downto 0);

begin
-- Aqui esta o nibble BCD
Nibble : process(dado)
begin

case dado is
--Lembrando que o display eh 7 segmentos
   when "0000" => out_7_seg <= "11111100";
   when "0001" => out_7_seg <= "01100000";
   when "0010" => out_7_seg <= "11011010";
   when "0011" => out_7_seg <= "11110010";
   when "0100" => out_7_seg <= "01100110";
   when "0101" => out_7_seg <= "10110110";
   when "0110" => out_7_seg <= "10111110";
   when "0111" => out_7_seg <= "11100000";
   when "1000" => out_7_seg <= "11111110";
   when "1001" => out_7_seg <= "11110110";
   when "1010" => out_7_seg <= "11101110";
   when "1011" => out_7_seg <= "00111110";
   when "1100" => out_7_seg <= "10011100";
   when "1101" => out_7_seg <= "01111010";
   when "1110" => out_7_seg <= "10011110";
   when "1111" => out_7_seg <= "10001110";
   when others => out_7_seg <= "11111100";

end case;

end process Nibble;

sete_seg <= out_7_seg;

end Behavioral;

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;



-- Definicao do display com 4 saidas
-- Ajeitando isso para que se tenha um comportamento BCD + REG
entity display4saidas is
   Port ( valor_in : in STD_LOGIC_VECTOR (3 downto 0);
          sete_seg : out STD_LOGIC_VECTOR (7 downto 0));
end display4saidas;

architecture Behavioral of display4saidas is

--  Tentando adaptar isso para BCD
signal dado : STD_LOGIC_VECTOR (3 downto 0);
signal out_7_seg : STD_LOGIC_VECTOR (7 downto 0);

begin
-- Aqui esta o nibble BCD
Nibble : process(dado)
begin

case dado is
```

```
--Lembrando que o display eh 7 segmentos
   when "0000" => out_7_seg <= "11111100";
   when "0001" => out_7_seg <= "01100000";
   when "0010" => out_7_seg <= "11011010";
   when "0011" => out_7_seg <= "11110010";
   when "0100" => out_7_seg <= "01100110";
   when "0101" => out_7_seg <= "10110110";
   when "0110" => out_7_seg <= "10111110";
   when "0111" => out_7_seg <= "11100000";
   when "1000" => out_7_seg <= "11111110";
   when "1001" => out_7_seg <= "11110110";
   when "1010" => out_7_seg <= "11101110";
   when "1011" => out_7_seg <= "00111110";
   when "1100" => out_7_seg <= "10011100";
   when "1101" => out_7_seg <= "01111010";
   when "1110" => out_7_seg <= "10011110";
   when "1111" => out_7_seg <= "10001110";
   when others => out_7_seg <= "11111100";

end case;

end process Nibble;

sete_seg <= out_7_seg;

end Behavioral;
```

2) Sintetize no VIVADO o VHDL circuito e informe a frequência e a utilização dos recursos

FPGA utilizado: xa7a12tcpg238-2I
Numero de LUTs: 9
Numero de flip-flops: 16

Quantos ciclos de relógio demora para aparecer o numero correto na saída do circuito quando o contador é atualizado? 5

3) Descreva um testbench que estimule alguns casos de operação do circuito, por exemplo que o contador conte para cima e para baixo e que haja somas e subtrações. A representação dos números é em complemento de 2.

Copie e cole aqui o Testbench:

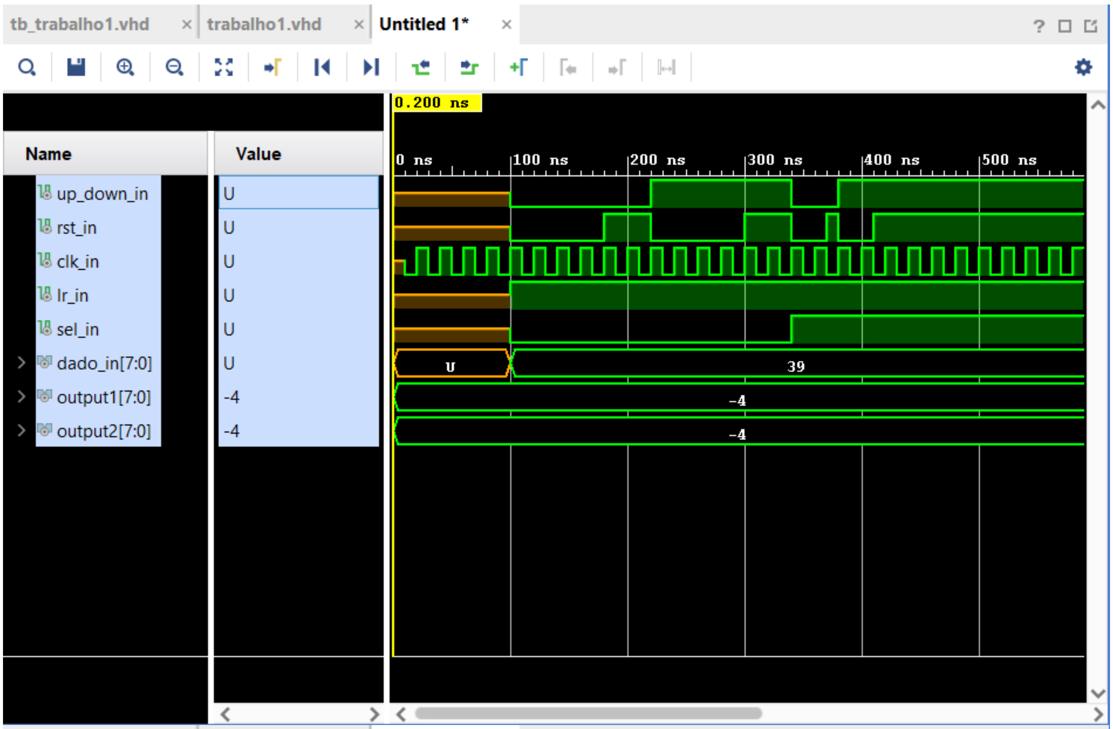
```
--------------------------------------------------------------------------------
```

```
-- Company:
-- Engineer:
--
-- Create Date: 07.03.2021 15:31:54
-- Design Name:
-- Module Name: tb_trabalho1 - Behavioral
-- Project Name:
-- Target Devices:
-- Tool Versions:
-- Description:
--
-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
----------------------------------------------------------------------------------


library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity tb_trabalho1 is
--  Port ( );
end tb_trabalho1;

architecture Behavioral of tb_trabalho1 is

component trabalho1 is
port ( Up_down : IN STD_LOGIC; -- Foi
    Rst, Clk: IN STD_LOGIC; -- Acho que foi tbm
    LR_entrada: in STD_LOGIC; --
    Dado_entrada : in STD_LOGIC_VECTOR(7 DOWNTO 0); --Foi
    seletor : in STD_LOGIC; -- Foi
    out_1 : out STD_LOGIC_VECTOR(7 DOWNTO 0);
    out_2 : out STD_LOGIC_VECTOR(7 DOWNTO 0));
end component;

-- Definindo os estimulos
```

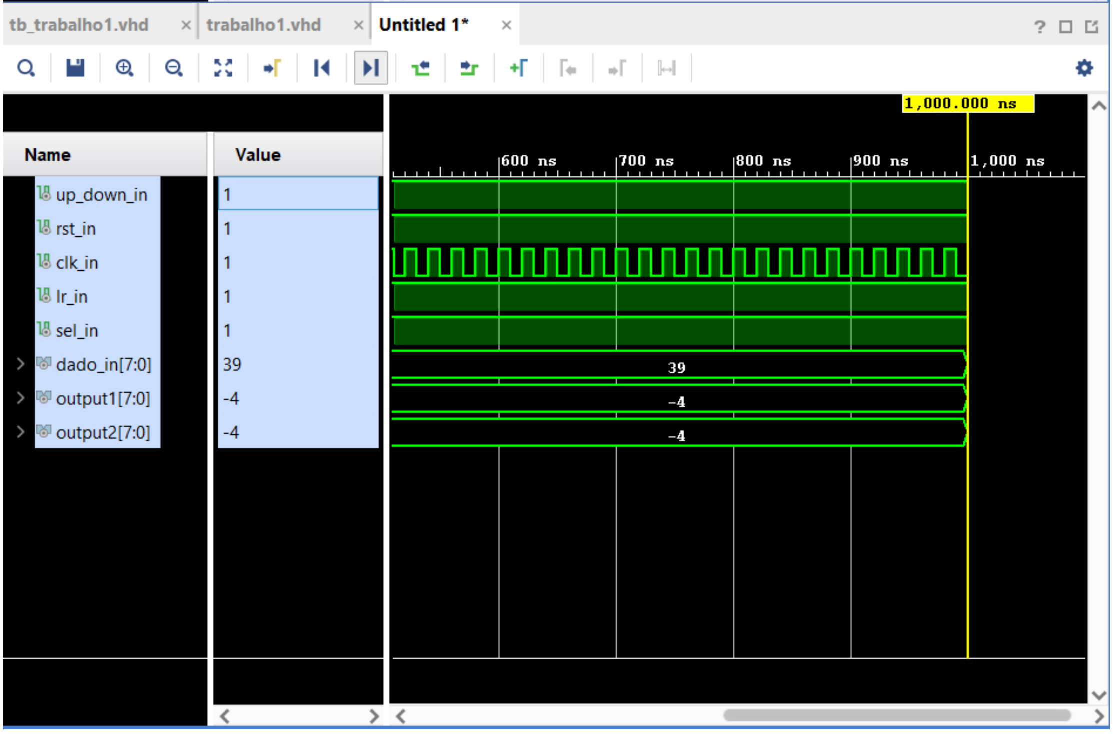
```
signal up_down_in, rst_in, clk_in, lr_in, sel_in : std_logic;
signal dado_in, output1, output2 : std_logic_vector(7 downto 0);

begin

t1: trabalho1
port map ( Up_down => up_down_in, -- Foi
        Rst => rst_in,
        Clk => clk_in, -- Acho que foi tbm
        LR_entrada => lr_in,
        Dado_entrada => dado_in, --Foi
        seletor => sel_in, -- Foi
        out_1 => output1,
        out_2 => output2);

process
begin
wait for 10ns;
clk_in <= '0';
wait for 10ns;
clk_in <= '1';
end process;

process
begin
wait for 100ns;
-- Somador, contador up
up_down_in <= '0';
rst_in <= '0';
lr_in <= '1';
dado_in <= "00100111";
sel_in <= '0';
wait for 80ns;
-- Clear the registers
rst_in <= not rst_in;
wait for 40ns;
-- Somador, contador down
up_down_in <= '1';
rst_in <= '0';
lr_in <= '1';
dado_in <= "00100111";
sel_in <= '0';
wait for 80ns;
-- Clear the registers
rst_in <= not rst_in;
wait for 40ns;
-- subtrator, contador up
up_down_in <= '0';
rst_in <= '0';
```

```
lr_in <= '1';
dado_in <= "00100111";
sel_in <= '1';
wait for 30ns;
-- Clear the registers
rst_in <= not rst_in;
wait for 10ns;
-- subtrator, contador down
up_down_in <= '1';
rst_in <= '0';
lr_in <= '1';
dado_in <= "00100111";
sel_in <= '1';
wait for 30ns;
-- Clear the registers
rst_in <= not rst_in;
wait for 10ns;
wait;

end process;

end Behavioral;
```

Copie e cole aqui o gráfico das simulações com ZOOM para poder observar os resultados.



Name	Value
up_down_in	U
rst_in	U
clk_in	U
lr_in	U
sel_in	U
dado_in[7:0]	U
output1[7:0]	-4
output2[7:0]	-4



Name	Value
up_down_in	1
rst_in	1
clk_in	1
lr_in	1
sel_in	1
dado_in[7:0]	39
output1[7:0]	-4
output2[7:0]	-4

Comente os resultados:

Tive algumas dificuldades, principalmente no que se refere a juntar todos os componentes para simulação e algumas coisas que não foram muito óbvias na minha percepção. Acredito que tenha algo errado na minha simulação pois acredito que não deveria ter sido este o comportamento correto apresentado, porém eu só consegui dedicar o meu final de semana a este trabalho então deixo aqui minha tentativa.