1)A)

Dezena		Unidade		Display
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	1	1	1	13
0	1	0	0	10
0	1	0	1	11
0	1	1	0	12
1	0	1	1	23
1	0	0	0	20
1	0	0	1	21
1	0	1	0	22
1	1	1	1	33
1	1	0	0	30
1	1	0	1	31
1	1	1	0	32
0	0	1	1	03

B) Tem a função de armadilha.

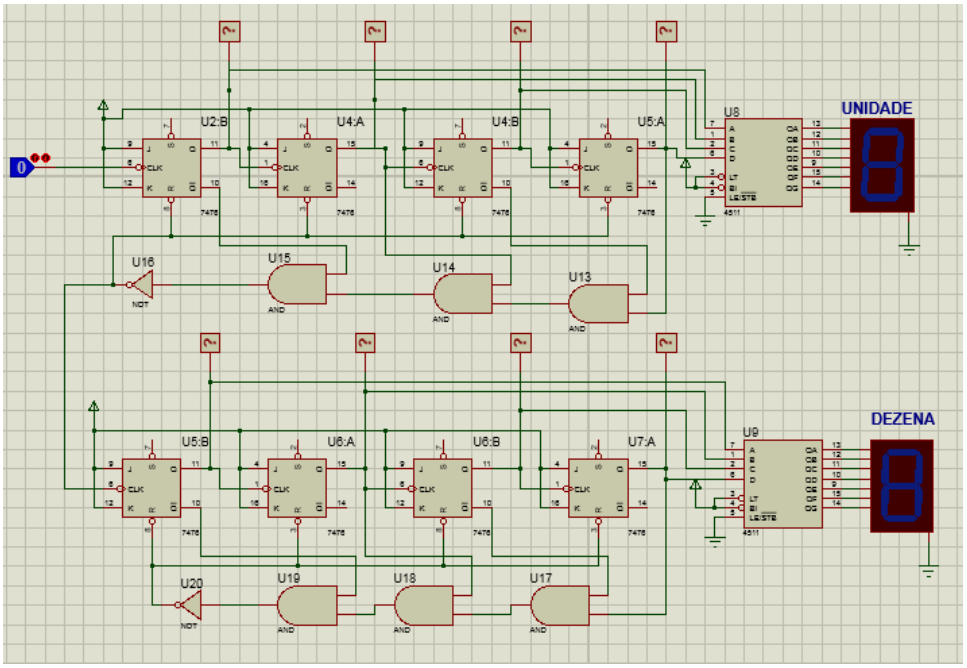
C) Porque com 2 Flip-Flops na unidade só é possível contar até 3. É necessário adicionar mais 3 Flip-Flops para corrigir esse erro para que a contagem consiga ir de 0-9.

D) Decodificador de binário para decimal.

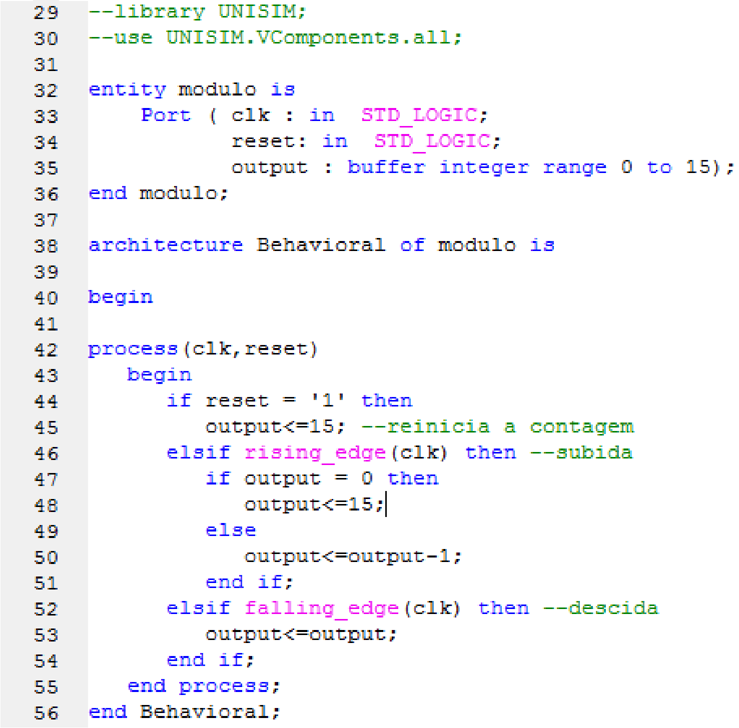
E) Cátodo comum pois o display utilizado esta ligado ao terra, logo ele é cátodo comum.

F) Neste caso o set e o reset não estão sendo utilizados, logo para deixá-los desabilitados eles foram conectados em nível lógico alto (Vcc).

G)

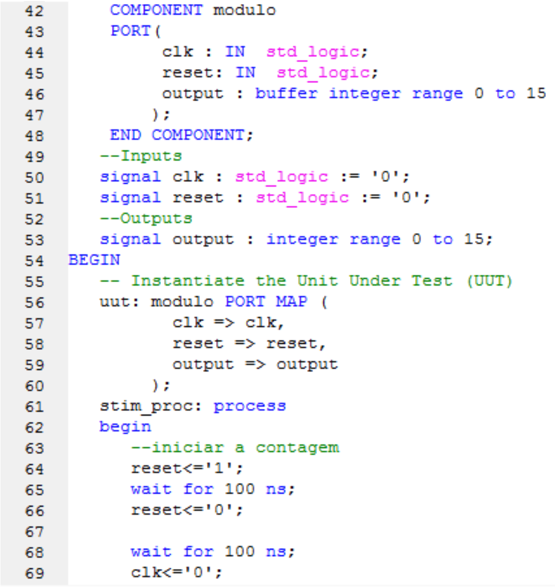


2) Modulo:

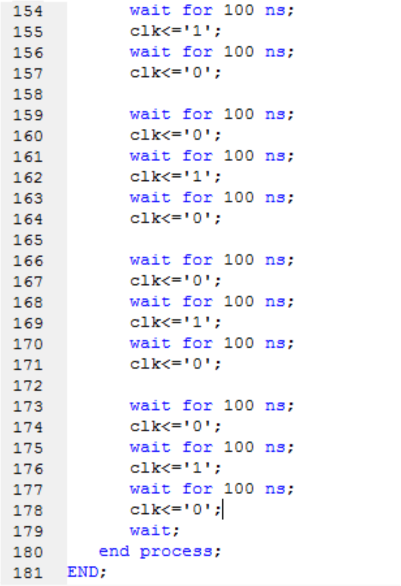


```
--library UNISIM;
--use UNISIM.VComponents.all;

entity modulo is
    Port ( clk : in  STD_LOGIC;
           reset: in  STD_LOGIC;
           output : buffer integer range 0 to 15);
end modulo;

architecture Behavioral of modulo is

begin

process(clk,reset)
   begin
      if reset = '1' then
         output<=15; --reinicia a contagem
      elsif rising_edge(clk) then --subida
         if output = 0 then
            output<=15;
         else
            output<=output-1;
         end if;
      elsif falling_edge(clk) then --descida
         output<=output;
      end if;
   end process;
end Behavioral;
```

Teste:

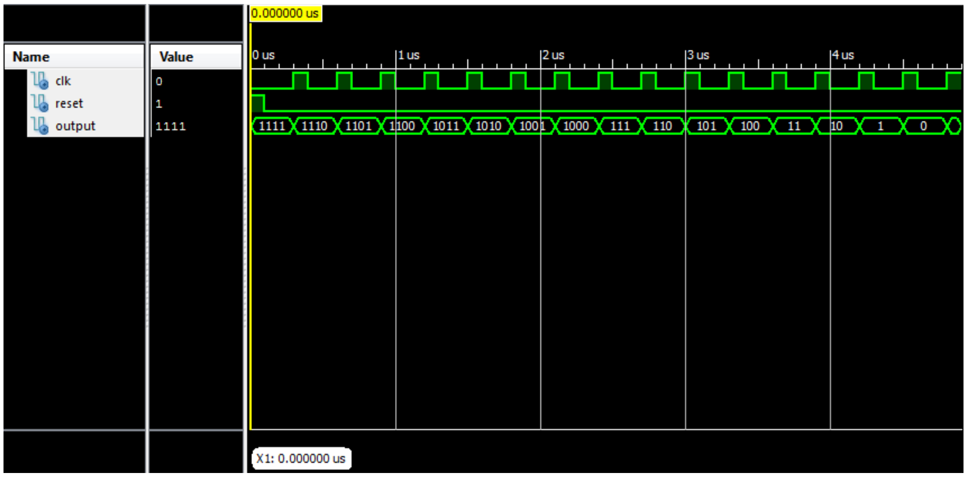


```
    COMPONENT modulo
    PORT(
         clk : IN  std_logic;
         reset: IN  std_logic;
         output : buffer integer range 0 to 15
        );
    END COMPONENT;
   --Inputs
   signal clk : std_logic := '0';
   signal reset : std_logic := '0';
   --Outputs
   signal output : integer range 0 to 15;
BEGIN
   -- Instantiate the Unit Under Test (UUT)
   uut: modulo PORT MAP (
          clk => clk,
          reset => reset,
          output => output
        );
   stim_proc: process
   begin
      --iniciar a contagem
      reset<='1';
      wait for 100 ns;
      reset<='0';

      wait for 100 ns;
      clk<='0';
```



```
      wait for 100 ns;
      clk<='1';
      wait for 100 ns;
      clk<='0';

      wait for 100 ns;
      clk<='0';
      wait for 100 ns;
      clk<='1';
      wait for 100 ns;
      clk<='0';

      wait for 100 ns;
      clk<='0';
      wait for 100 ns;
      clk<='1';
      wait for 100 ns;
      clk<='0';

      wait for 100 ns;
      clk<='0';
      wait for 100 ns;
      clk<='1';
      wait for 100 ns;
      clk<='0';
      wait;
   end process;
END;
```

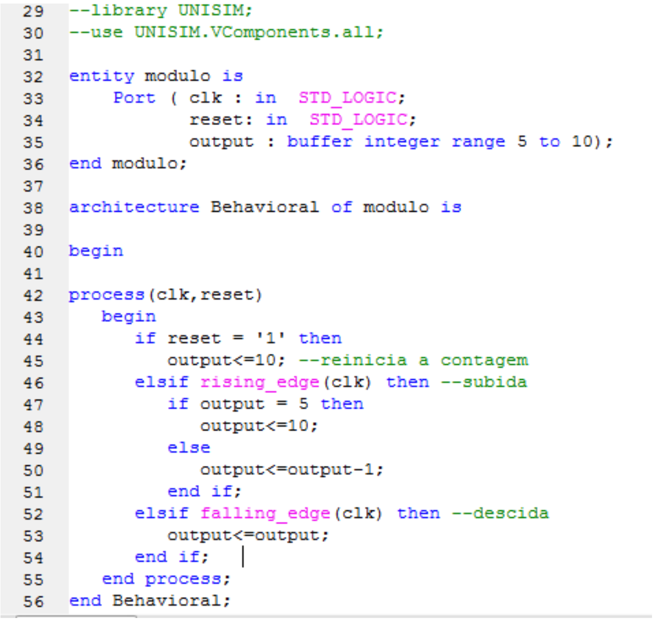
Gráfico:



Name	Value
clk	0
reset	1
output	1111

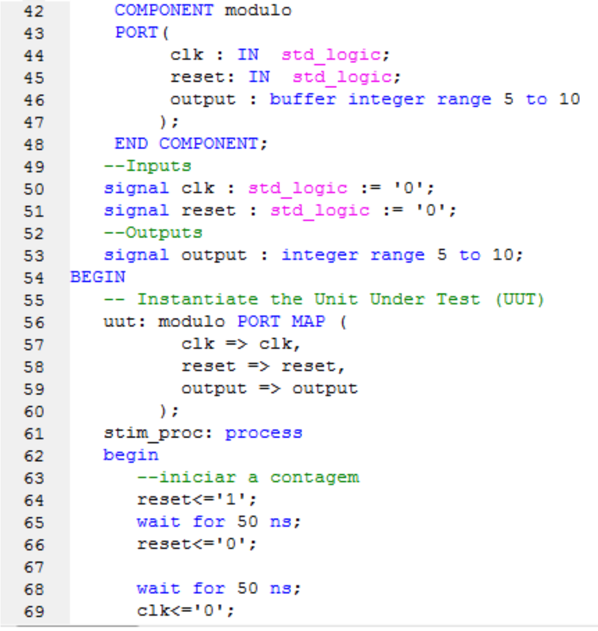
Proposto:

Modulo:

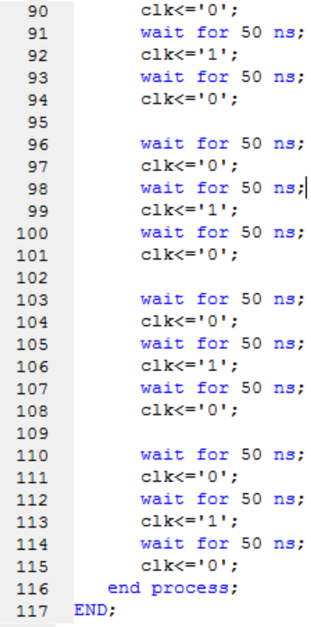


```
--library UNISIM;
--use UNISIM.VComponents.all;

entity modulo is
    Port ( clk : in  STD_LOGIC;
           reset: in  STD_LOGIC;
           output : buffer integer range 5 to 10);
end modulo;

architecture Behavioral of modulo is

begin

process(clk,reset)
   begin
      if reset = '1' then
         output<=10; --reinicia a contagem
      elsif rising_edge(clk) then --subida
         if output = 5 then
            output<=10;
         else
            output<=output-1;
         end if;
      elsif falling_edge(clk) then --descida
         output<=output;
      end if;
   end process;
end Behavioral;
```

Teste:

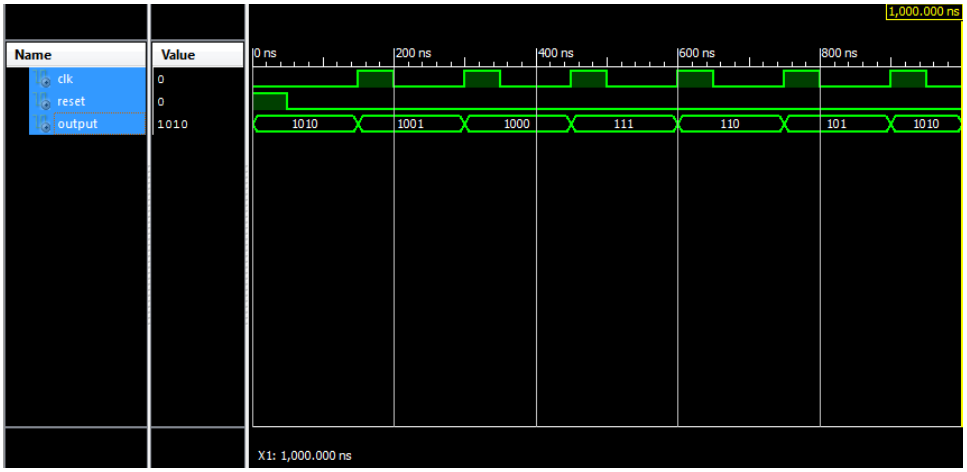


```
    COMPONENT modulo
    PORT(
         clk : IN  std_logic;
         reset: IN  std_logic;
         output : buffer integer range 5 to 10
        );
    END COMPONENT;
   --Inputs
   signal clk : std_logic := '0';
   signal reset : std_logic := '0';
   --Outputs
   signal output : integer range 5 to 10;
BEGIN
   -- Instantiate the Unit Under Test (UUT)
   uut: modulo PORT MAP (
          clk => clk,
          reset => reset,
          output => output
        );
   stim_proc: process
   begin
      --iniciar a contagem
      reset<='1';
      wait for 50 ns;
      reset<='0';

      wait for 50 ns;
      clk<='0';
```



```
      clk<='0';
      wait for 50 ns;
      clk<='1';
      wait for 50 ns;
      clk<='0';

      wait for 50 ns;
      clk<='0';
      wait for 50 ns;
      clk<='1';
      wait for 50 ns;
      clk<='0';

      wait for 50 ns;
      clk<='0';
      wait for 50 ns;
      clk<='1';
      wait for 50 ns;
      clk<='0';

      wait for 50 ns;
      clk<='0';
      wait for 50 ns;
      clk<='1';
      wait for 50 ns;
      clk<='0';
   end process;
END;
```

Gráfico:



Name	Value
clk	0
reset	0
output	1010