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.

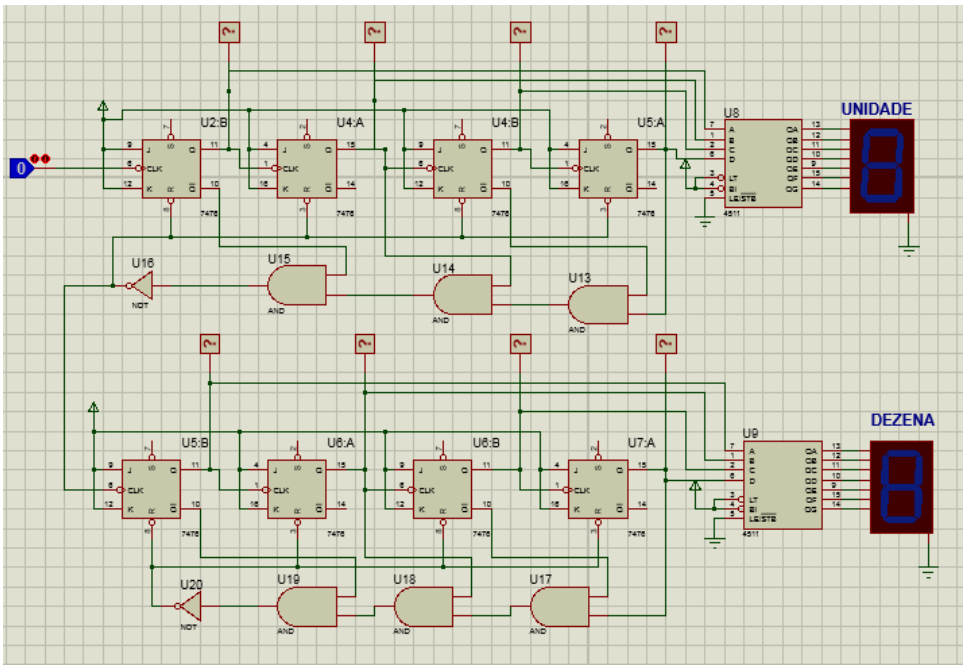
C) Porque com 2 Flip-Flops na unidade só é possível contar até 3. É necessário adicionar mais 3 Flip-Flops para corrigir esse erro para que a contagem consiga ir de 0-9.

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:

```
29  --library UNISIM;
30  --use UNISIM.VComponents.all;
31
32  entity modulo is
33      Port ( clk : in  STD_LOGIC;
34             reset: in  STD_LOGIC;
35             output : buffer integer range 0 to 15);
36  end modulo;
37
38  architecture Behavioral of modulo is
39
40  begin
41
42  process(clk,reset)
43     begin
44        if reset = '1' then
45           output<=15; --reinicia a contagem
46        elsif rising_edge(clk) then --subida
47           if output = 0 then
48              output<=15;
49           else
50              output<=output-1;
51           end if;
52        elsif falling_edge(clk) then --descida
53           output<=output;
54        end if;
55     end process;
56  end Behavioral;
```

Teste:

```
42      COMPONENT modulo
43      PORT(
44           clk : IN  std_logic;
45           reset: IN  std_logic;
46           output : buffer integer range 0 to 15
47          );
48      END COMPONENT;
49     --Inputs
50     signal clk : std_logic := '0';
51     signal reset : std_logic := '0';
52     --Outputs
53     signal output : integer range 0 to 15;
54  BEGIN
55     -- Instantiate the Unit Under Test (UUT)
56     uut: modulo PORT MAP (
57            clk => clk,
58            reset => reset,
59            output => output
60          );
61     stim_proc: process
62     begin
63        --iniciar a contagem
64        reset<='1';
65        wait for 100 ns;
66        reset<='0';
67
68        wait for 100 ns;
69        clk<='0';
154       wait for 100 ns;
155       clk<='1';
156       wait for 100 ns;
157       clk<='0';
158
159       wait for 100 ns;
160       clk<='0';
161       wait for 100 ns;
162       clk<='1';
163       wait for 100 ns;
164       clk<='0';
165
166       wait for 100 ns;
167       clk<='0';
168       wait for 100 ns;
169       clk<='1';
170       wait for 100 ns;
171       clk<='0';
172
173       wait for 100 ns;
174       clk<='0';
175       wait for 100 ns;
176       clk<='1';
177       wait for 100 ns;
178       clk<='0';
179       wait;
180    end process;
181 END;
```

Gráfico:

Proposto:

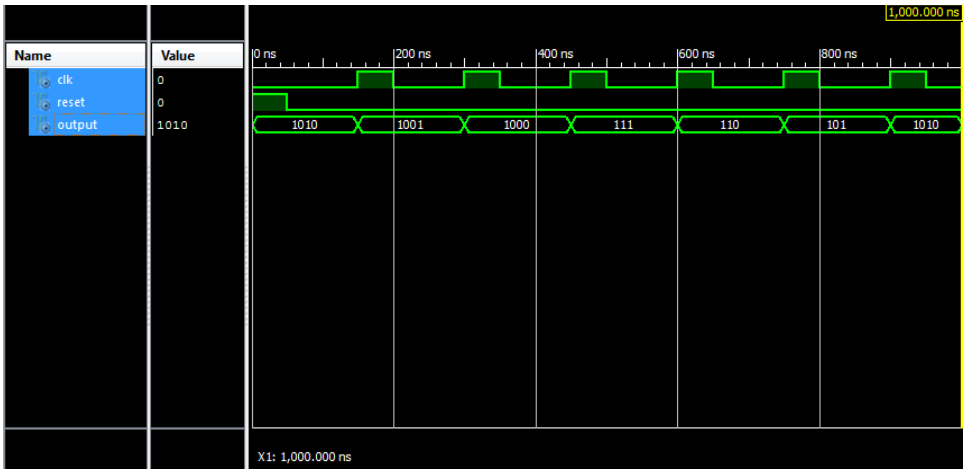
Modulo:

```
--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
reset
output
0
0
1010
0 ns
200 ns
400 ns
600 ns
800 ns
1,000.000 ns
1010
1001
1000
111
110
101
1010
X1: 1,000.000 ns