

```
1  library IEEE;-- chama as bibliotecas
2  use IEEE.std_logic_1164.all;
3  use IEEE.std_logic_arith.all;
4  use IEEE.std_logic_unsigned.all;
5  use work.bcd_decoder.all;--chama o decoder de 7seg no outro arquivo
6
7  entity calculadora is
8      port (
9          SW: in unsigned( 9 downto 0 );-- define as chaves como binarios sem sinal
10         KEY: in std_logic_vector (3 DOWNT0 0);--define os push-buttons
11         HEX0 :OUT STD_LOGIC_VECTOR (6 downto 0);--disp1
12         HEX1 :OUT STD_LOGIC_VECTOR (6 downto 0);--disp2
13         HEX2 :OUT STD_LOGIC_VECTOR (6 downto 0);--disp3
14         HEX3 :OUT STD_LOGIC_VECTOR (6 downto 0)--disp4
15     );
16 end calculadora;
17
18 architecture operacoes of calculadora is
19     signal soma: integer range -7 to 49; -- soma é um inteiro entre -7 e 49
20     signal S: integer range -7 to 49;
21     signal as,bs: integer range 0 to 7;
22     signal res,diga,digb: std_LOGIC_VECTOR (20 downto 0);
23     signal Sel: std_logic_vector (1 DOWNT0 0);
24     signal sinal: std_logic_vector (6 DOWNT0 0);
25     signal R : integer;
26     signal rest : std_LOGIC_VECTOR (20 downto 0);
27 begin
28     Sel<= conv_std_logic_vector (SW(5 DOWNT0 4), 2);-- sel é as chaves 5 e 4 ↵
29     convertidas para vetor
30
31     op: process (Sel,SW,as,bs,soma,S,R)
32     Begin
33         as <= conv_integer (SW(9 DOWNT0 7));-- as é chaves 9 à 7 convertidas pra inteiro
34         bs <= conv_integer (SW(2 DOWNT0 0));-- as é chaves 2 à 0 convertidas pra inteiro
35         if Sel="00" THEN -- se Sel for 00 entao soma é as + bs
36             soma<= (as) + (bs);
37             S<= soma;-- S é soma
38         elsif Sel="01" THEN -- se Sel for 01 entao soma é as - bs
39             soma<= (as) - (bs);
40             S<= soma;
41         elsif Sel="10" THEN -- se Sel for 10 entao soma é as * bs
42             soma<= (as) * (bs);
43             S<= soma;
44         elsif Sel="11" THEN --se Sel for 11 entao
45             if bs /= 0 then --e se bs for diferente de entao
46                 soma<= (as) / (bs);-- soma é as/bs
47                 S<= soma; --S é soma
48                 R<= as rem bs; --R é o resto da divisão as/bs
49             end if;
50         end if;
51     end process;
52
53     sig: process (Sel,SW) -- process pra definir o sinal da operacoes
54     Begin
55         if Sel="00" THEN -- se Sel for 00 entao sinal é o vetor
56             sinal <= "0011010";
57         elsif Sel="01" THEN
58             sinal <= "0111111";
59         elsif Sel="10" THEN
60             sinal <= "0001001";
61         elsif Sel="11" THEN
62             sinal <= "0101101";
63         end if;
64     end process;
65
```

```
66      disp: process (KEY(3), Sel, res, rest, S, R, as, bs, diga, sinal, digb) -- process para exibir
no display
67      Begin
68          if KEY(3)='0' THEN -- se a key(3) for pressionada
69              if Sel="11" Then -- e se Sel for 11(divisao)
70                  if bs /= 0 Then --e se bs for diferente de 0
71                      res <= conviseq(S); --res é o vetor S convertido pela função conviseq
72                      rest <= conviseq(R); --rest é o vetor R convertido pela função conviseq
73                      HEX0 <= rest(6 DOWNTO 0);-- disp 1 é os 7 primeiros digitos de rest
74                      HEX1 <= "0101111";--disp 2 é "r"
75                      HEX2 <= res(6 DOWNTO 0);-- disp 3 é os 7 primeiros digitos de res
76                      HEX3 <= "0011000";--disp 4 é "q"
77                  else --senao(bs=0)
78                      HEX0 <= "0100011"; --disp 1 "o"
79                      HEX1 <= "0101111"; --disp 2 "r"
80                      HEX2 <= "0101111"; --disp 3 "r"
81                      HEX3 <= "0000110"; --disp 4 "E"
82                  end if;
83
84              else--senao(Sel diferente de 11)
85                  res <= conviseq(S);--res é o vetor S convertido pela função conviseq
86                  HEX0 <= res(6 DOWNTO 0);-- disp 1 é os 7 primeiros digitos de res
87                  HEX1 <= res(13 DOWNTO 7);-- disp 2 é os digitos de 14 a 8 de res
88                  HEX2 <= res(20 DOWNTO 14);-- disp 3 é os digitos de 21 a 15 de res
89                  HEX3 <= "0001000";--disp 4 é "A"
90              end if;
91
92              else--senao(Key(3) liberada)
93                  diga <= conviseq(as);-- diga é o vetor as convertido pela função conviseq
94                  digb <= conviseq(bs);-- diga é o vetor bs convertido pela função conviseq
95                  HEX3 <= "1000110";-- disp 4 é "C"
96                  HEX2 <= diga(6 DOWNTO 0);-- disp 3 e é os 6 primeiros digitos de diga
97                  HEX1 <= sinal(6 DOWNTO 0);--disp 2 é o vetor em sinal
98                  HEX0 <= digb(6 DOWNTO 0);-- disp 1 e é os 6 primeiros digitos de digb
99              end if;
100          end process;
101      end operacoes;
102
```