Universidade Federal de Santa Catarina
Centro Tecnológico – CTC
Departamento de Engenharia Elétrica

“EEL7020 – Sistemas Digitais”

Prof. Eduardo Augusto Bezerra

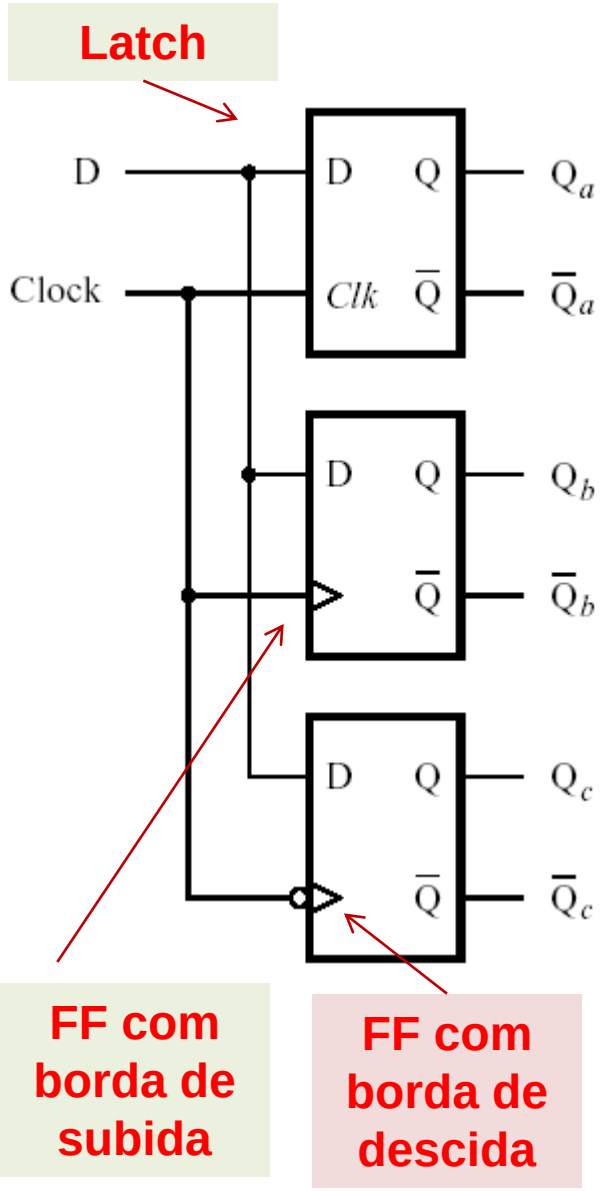
Eduardo.Bezerra@eel.ufsc.br

Florianópolis, março de 2010.

Tarefa I: circuito com dois flip-flops e um latch

Arquivo: lab3_VHDL.pdf
parte IV

Tarefa I: Circuito com dois FFs e um latch

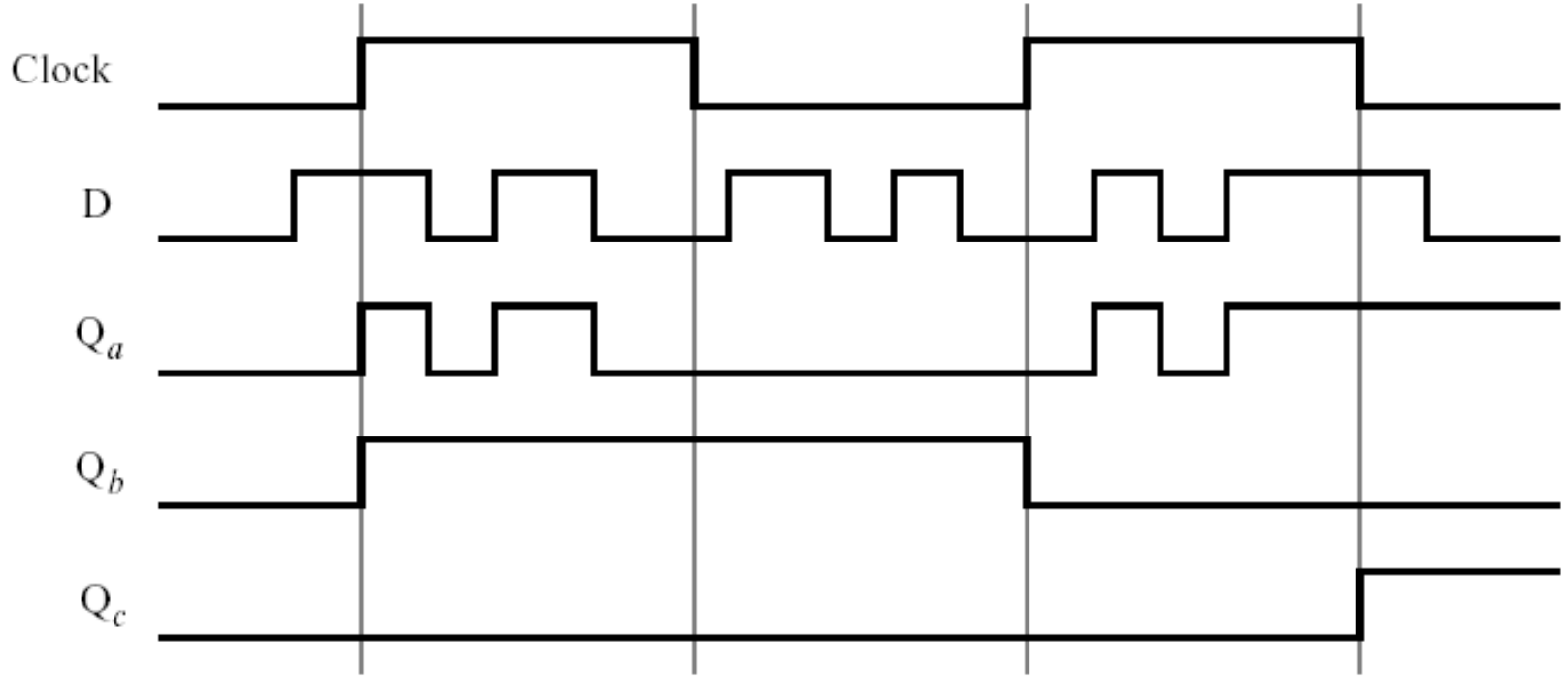


Etapas:

1. **Implementar o circuito em VHDL, usando o Quartus II, e realizar a síntese.**
2. **Utilizar a ferramenta em *Tools -> Netlist Viewers -> Technology Map Viewer*, e examinar o circuito gerado pela síntese.**
3. **Verificar que o latch é implementado em uma lookup table, e para os flip-flops são aproveitados os flip-flops existentes no FPGA da DE2.**
4. **Realizar a simulação funcional de acordo com os dados descritos no diagrama de formas de onda do próximo slide.**
5. **Observar as diferenças no comportamento dos três elementos de armazenamento implementados.**

Tarefa I: Circuito com dois FFs e um latch

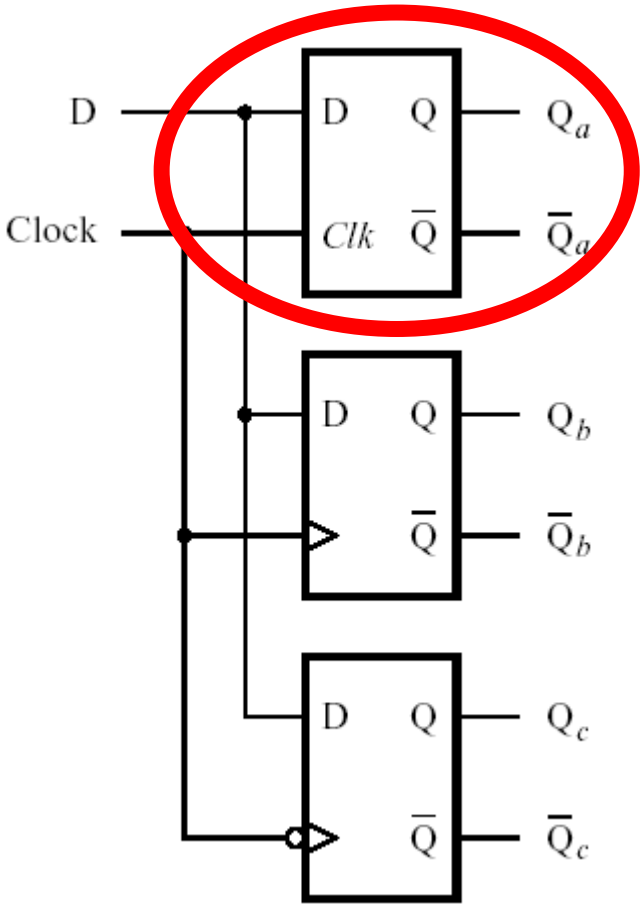
Resultado esperado da simulação funcional.



<u>Sugestão de implementação VHDL</u>: criar um componente LATCH e um componente FLIP-FLOP e, na descrição VHDL principal (top), utilizar esses componentes para criar o LATCH e os dois FFs, e realizar o mapeamento dos pinos, conectando os diversos sinais com PORT MAP.

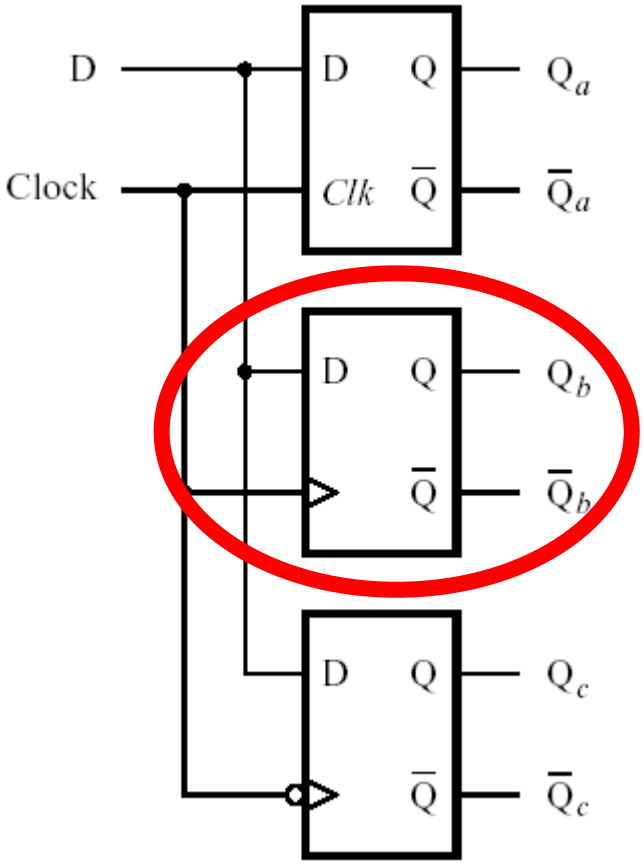
Tarefa I: Sugestão de VHDL

```
library ieee;
use ieee.std_logic_1164.all;
entity D_latch is port (
            C:  in std_logic;
            D:  in std_logic;
            Q: out std_logic
          );
end D_latch;
architecture behv of D_latch is
begin
            process(C, D)
            begin
                if (C = '1') then
                    Q <= D;
                end if;
            end process;
end behv;
```



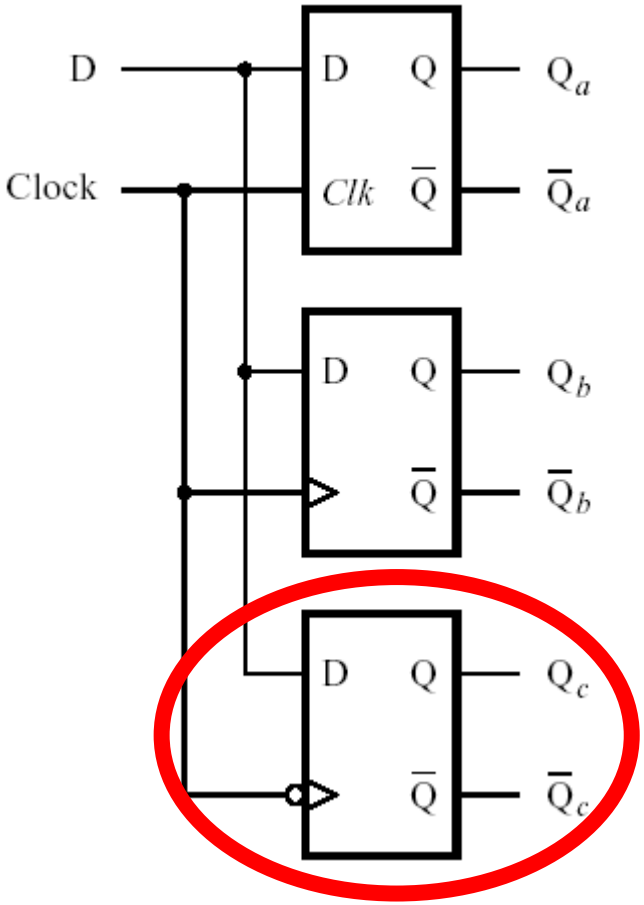
Tarefa I: Sugestão de VHDL

```
library ieee;
use ieee.std_logic_1164.all;
entity D_FF is port (
            CLK:  in std_logic;
            D:  in std_logic;
            Q: out std_logic
          );
end D_FF;
architecture behv of D_FF is
begin
    process(CLK, D)
    begin
        if (CLK'event and CLK = '1') then
                    Q <= D;
        end if;
    end process;
end behv;
```



Tarefa I: Sugestão de VHDL

```
library ieee;
use ieee.std_logic_1164.all;
entity D_FFd is port (
            CLK:  in std_logic;
            D:  in std_logic;
            Q: out std_logic
          );
end D_FFd;
architecture behv of D_FFd is
begin
    process(CLK, D)
    begin
        if (CLK'event and CLK = '0') then
                    Q <= D;
        end if;
    end process;
end behv;
```



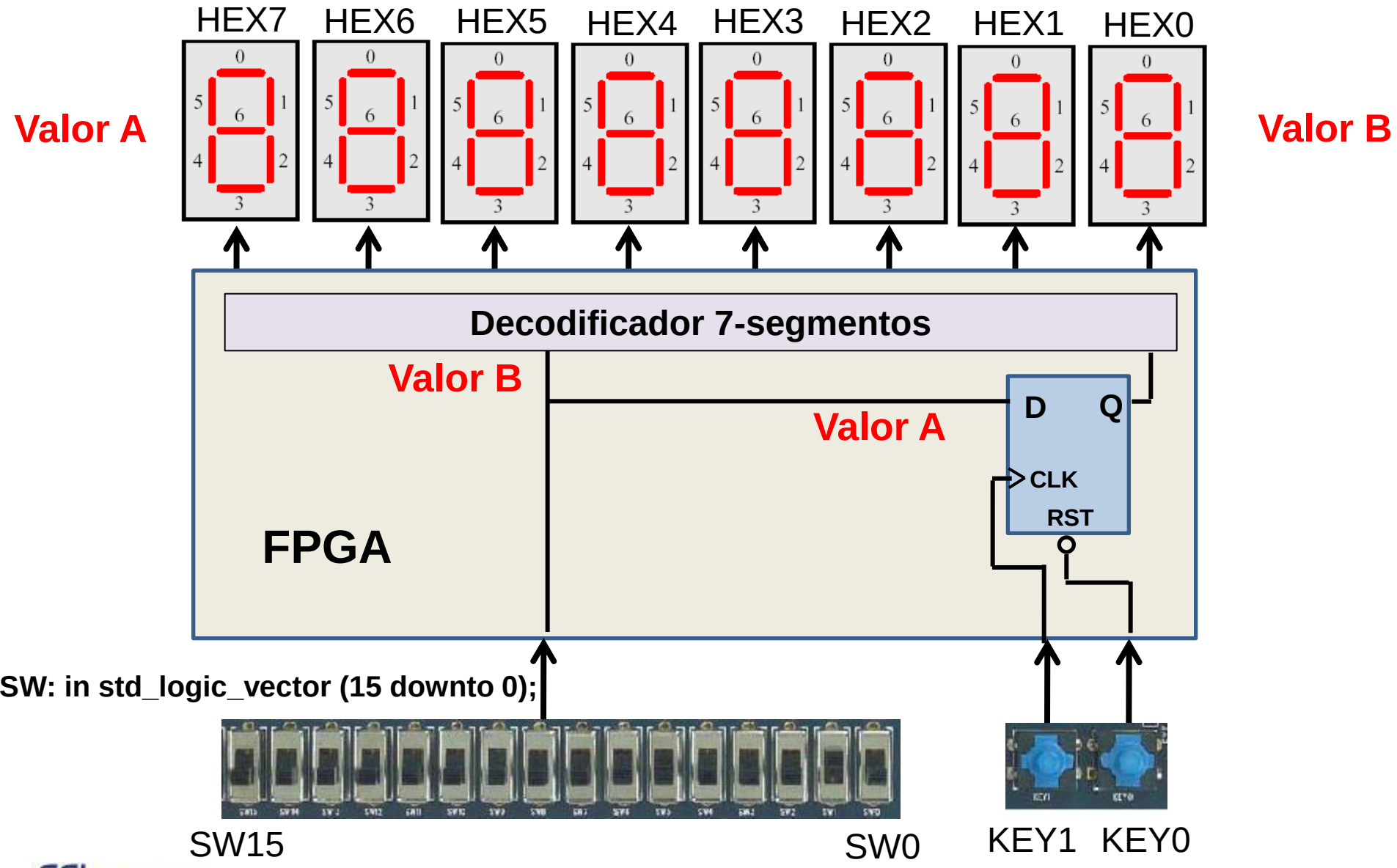
Tarefa I: Sugestão de VHDL

```
library  ieee;
use  ieee.std_logic_1164.all;
entity lab7 is
   port (D : IN STD_LOGIC;
         CLK : IN STD_LOGIC;
         QA : OUT STD_LOGIC;
         QB : OUT STD_LOGIC;
         QC : OUT STD_LOGIC
       );
end lab7;

architecture top of lab7 is

   component D_latch
      port (          C:  in std_logic;
                      D:  in std_logic;
                      Q: out std_logic
           );
   end component;

   component D_FF
    port (  CLK:  in std_logic;
            D:  in std_logic;
            Q: out std_logic
          );
   end component;

  component D_FFd
    port (  CLK:  in std_logic;
            D:  in std_logic;
            Q: out std_logic
          );
   end component;

begin   -- início da architecture
            L0: D_latch port map (CLK, D, QA);
            FS: D_FF port map (CLK, D, QB);
            FD: D_FFd port map (CLK, D, QC);
end top; -- fim da architecture
```

Tarefa II: display de 7-segmentos com registrador

Arquivo: lab3_VHDL.pdf
parte V

Tarefa II: Sugestão de circuito



Tarefa II: Sugestão de VHDL (Prof. Joni Fraga)

```
library ieee;
use ieee.std_logic_1164.all;
entity quartus1 is
        port (
                KEY:  in std_logic_vector(1 downto 0);
                SW:  in std_logic_vector(15 downto 0);
                HEX0 : out std_logic_vector(6 downto 0);
                HEX1 : out std_logic_vector(6  downto 0);
                HEX2 : out std_logic_vector(6 downto 0);
                HEX3 : out std_logic_vector(6  downto 0);
                HEX4 : out std_logic_vector(6 downto 0);
                HEX5 : out std_logic_vector(6  downto 0);
                HEX6 : out std_logic_vector(6 downto 0);
                HEX7 : out std_logic_vector(6  downto 0);
                LEDR:  out std_logic_vector(15 downto 0)
    );
end quartus1;
```

Tarefa II: Sugestão de VHDL (Prof. Joni Fraga)

```
architecture behv of quartus1 is
        signal RST, CLK   : std_logic;
        signal D, Q       : std_logic_vector(15 downto 0);
        component deco7
                port (    d: in std_logic_vector(3 downto 0);
                          f: out std_logic_vector(6 downto 0)
                     );
        end component;
begin
        D <= SW;
        RST <= KEY(0);
        CLK <= KEY(1);
        DEC0:  deco7 port map(Q(3 downto 0),HEX0(6 downto 0));
        DEC1:  deco7 port map(Q(7 downto 4),HEX1(6 downto 0));
        DEC2:  deco7 port map(Q(11 downto 8),HEX2(6 downto 0));
        DEC3:  deco7 port map(Q(15 downto 12),HEX3(6 downto 0));
        DEC4:  deco7 port map(SW(3 downto 0),HEX4(6 downto 0));
        DEC5:  deco7 port map(SW(7 downto 4),HEX5(6 downto 0));
        DEC6:  deco7 port map(SW(11 downto 8),HEX6(6 downto 0));
        DEC7:  deco7 port map(SW(15 downto 12),HEX7(6 downto 0));
```

Tarefa II: Sugestão de VHDL (Prof. Joni Fraga)

```
process(CLK,  RST, D)
    begin
            if (RST = '0') then
                    Q <= "0000000000000000";
            elsif (CLK'event and CLK = '1') then
                    Q <= D;
            end if;
    end process;
end behv;
```

Tarefa II: Sugestão de VHDL (Prof. Joni Fraga)

```
library ieee;
use ieee.std_logic_1164.all;
-- decodificador 7 segmentos
entity deco7 is
  port (
     d : in std_logic_vector(3 downto 0);
     f : out std_logic_vector(6 downto 0)
  );
end deco7;
```

```
architecture dec_beh of deco7 is
begin
  f <=      "1111110" when d="0000" else
            "0110000" when d="0001" else
            "1101101" when d="0010" else
            "1111001" when d="0011" else
            "0110011" when d="0100" else
            "1011011" when d="0101" else
            "1011111" when d="0110" else
            "1110000" when d="0111" else
            "1111111" when d="1000" else
            "1110011" when d="1001" else
            "1110111" when d="1010" else
            "0011111" when d="1011" else
            "1001110" when d="1100" else
            "0111101" when d="1101" else
            "1001111" when d="1110" else
            "1000111";
end dec_beh;
```