TP4 EN 1B CHRONOMETRE 2021-2022

Oliver BELLIARD & Fahd RAFIK

La préparation se trouve en annexe.

Réalisation 9 :

“ChronoEnContinu.vhd” se trouve en annexe. Le code a été testé et vérifié par l’enseignant. Nous avons en revanche ajusté la valeur de qui divise “TICK_1ms” pour que la durée d’une seconde soit le plus proche possible d’une vraie seconde mesurée à l’œil nu à l'aide des chronomètres de nos portables. D’où le “N=>25” en tant que paramètre pour “ms”.

Réalisation 10 :

Vous trouverez “Chrono_SW.vhd” en annexe.

Réalisation 11 :

Vous trouverez notre implémentation de DFM en annexe.

Lors de la séance nous avons perdu beaucoup de temps pour le débogage car nous n’avons pas remappé les boutons que nous utilisions lors des tests. C’est pourquoi nous n’avons pas pu aboutir à la fin du TP, notamment à “chrono_BTN”.

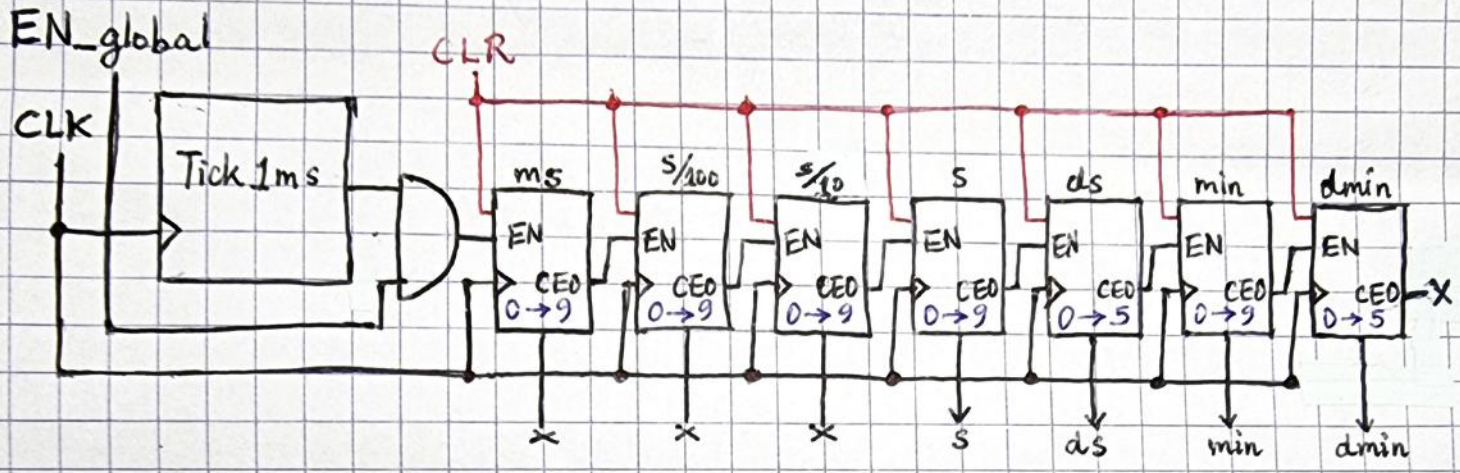
Vous trouverez les annexes à partir de la page suivante.

Annexe 1 :

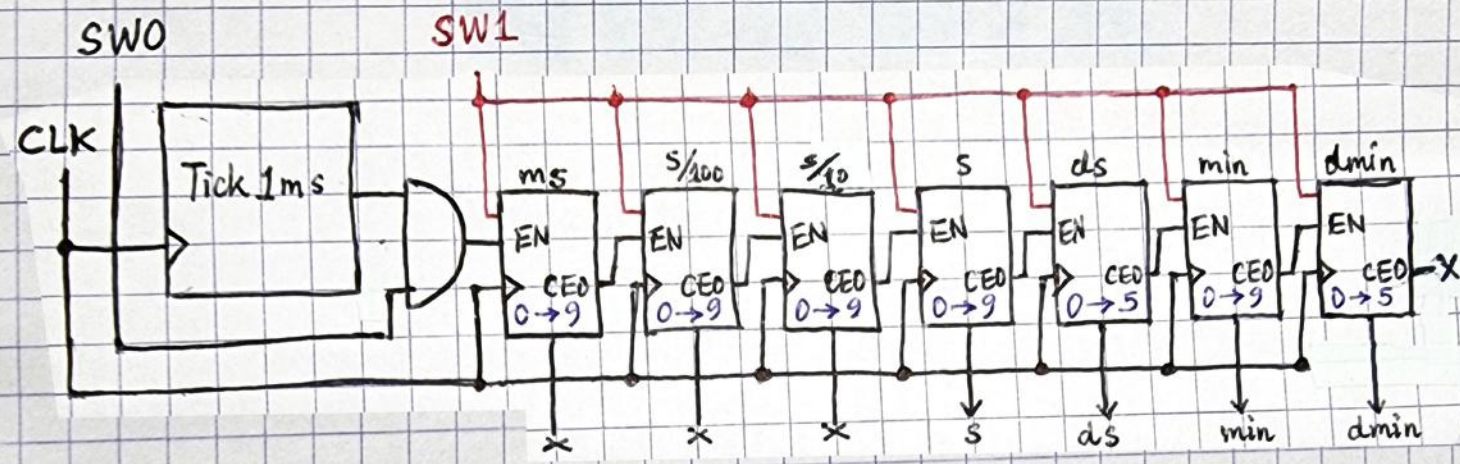
Préparation

EN-TP4

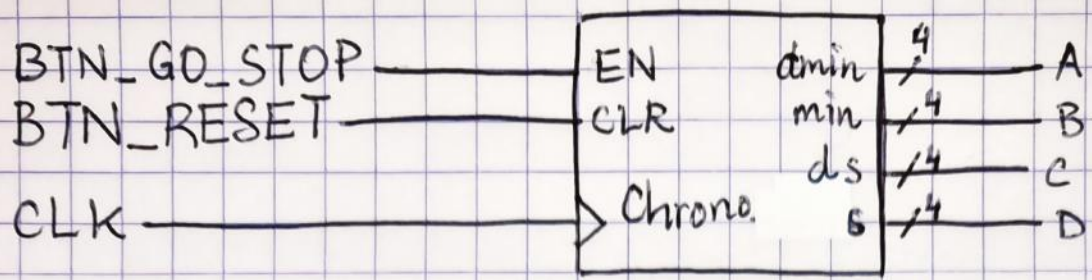
Prepa 9.A:



Prépa 10.A:

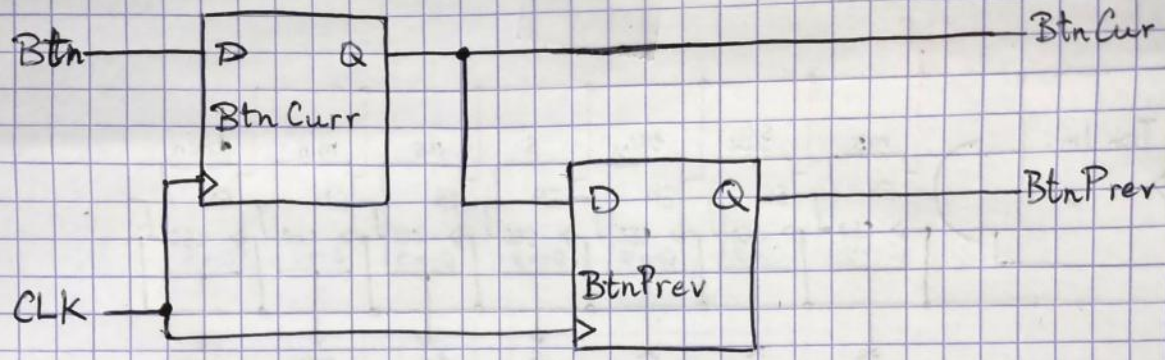


Prép12.A:

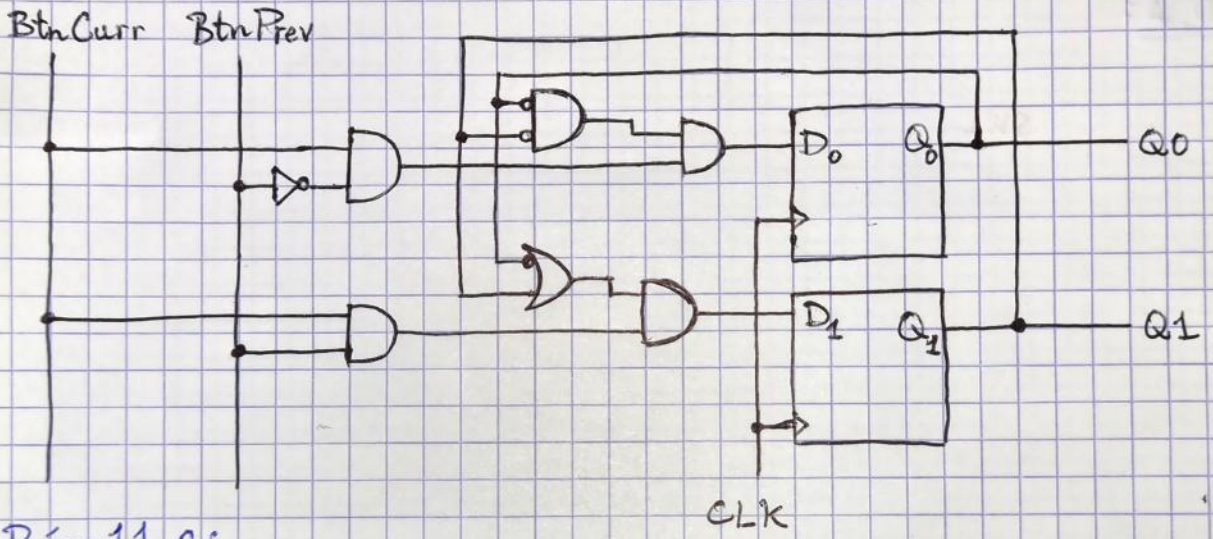


Prép12.B:

Prépa 11.A:



Prépa 11.B:



Prépa 11.C:

$$D1 = (\overline{Q1} \cdot Q0) + ((Q1 \cdot \overline{Q0})(BtnCurr + BtnPrev))$$

$$D0 = (\overline{Q1} \cdot \overline{Q0}) \cdot (BtnCurr \cdot \overline{BtnPrev})$$

Q1	Q0	BtnCurr	BtnPrev	D1	D0
0	0	0	X	0	0
0	0	1	1	0	0
0	0	1	0	0	1
0	1	X	X	1	0
1	0	1	X	1	0
1	0	X	1	1	0
1	0	0	0	0	0

On peut se retrouver dans l'état Q1Q0 = 00 après un temps d'inactivité, suite à l'appui sur un bouton.

Annexe 2 :

ChronoEnContinu.vhd

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 04.01.2022 09:08:18
-- Design Name:
-- Module Name: ChronoEnContinu - 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;
use IEEE.NUMERIC_STD.ALL;

entity ChronoEnContinu is
    Port ( CLK : in STD_LOGIC;
           EN : in STD_LOGIC;
           CLR : in STD_LOGIC;
           S : out STD_LOGIC_VECTOR (6 downto 0);
           ENAN : out STD_LOGIC_VECTOR (3 downto 0));
end ChronoEnContinu;

architecture Behavioral of ChronoEnContinu is

    signal EN_1ms: STD_LOGIC;
    signal EN_ticks: STD_LOGIC_VECTOR(7 downto 0);
    signal Temps: STD_LOGIC_VECTOR (31 downto 0);

begin

    disp : entity work.Disp4D
           port map (A=>Temps(15 downto 12),
                     B=>Temps(19 downto 16),
                     C=>Temps(23 downto 20),
                     D=>Temps(27 downto 24),
                     CLK=>CLK,
                     S=>S,
                     ENAN=>ENAN);

    tick : entity work.TICK_1ms
           port map (CLK=>CLK, Tick=>EN_1ms);

    ms : entity work.Cnt0toN_EN generic map (N=>25)
          port map (CLK=>CLK, EN=>EN_1ms, CLR=>CLR, Cout=>EN_ticks(0), Q=>Temps(3 downto 0));

    csu : entity work.Cnt0toN_EN generic map (N=>10)
          port map (CLK=>CLK, EN=>EN_ticks(0), CLR=>CLR, Cout=>EN_ticks(1), Q=>Temps(7 downto 4));

    csd : entity work.Cnt0toN_EN generic map (N=>6)
          port map (CLK=>CLK, EN=>EN_ticks(1), CLR=>CLR, Cout=>EN_ticks(2), Q=>Temps(11 downto 8));

    su : entity work.Cnt0toN_EN generic map (N=>10)
          port map (CLK=>CLK, EN=>EN_ticks(2), CLR=>CLR, Cout=>EN_ticks(3), Q=>Temps(15 downto 12));

    sd : entity work.Cnt0toN_EN generic map (N=>6)
         port map (CLK=>CLK, EN=>EN_ticks(3), CLR=>CLR, Cout=>EN_ticks(4), Q=>Temps(19 downto 16));

    mu : entity work.Cnt0toN_EN generic map (N=>10)
         port map (CLK=>CLK, EN=>EN_ticks(4), CLR=>CLR, Cout=>EN_ticks(5), Q=>Temps(23 downto 20));

    md : entity work.Cnt0toN_EN generic map (N=>6)
         port map (CLK=>CLK, EN=>EN_ticks(5), CLR=>CLR, Cout=>EN_ticks(6), Q=>Temps(27 downto 24));

end Behavioral;
```

Annexe 3 :

Chrono_SW.vhd

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 04.01.2022 11:58:00
-- Design Name:
-- Module Name: chrono_SW - 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;
use IEEE.NUMERIC_STD.ALL;

entity chrono_SW is
    Port ( CLK : in STD_LOGIC;
           EN : in STD_LOGIC;
           CLR : in STD_LOGIC;
           S : out STD_LOGIC_VECTOR (6 downto 0);
           ENAN : out STD_LOGIC_VECTOR (3 downto 0));
end chrono_SW;

architecture Behavioral of chrono_SW is

    signal EN_1ms: STD_LOGIC;
    signal EN_ticks: STD_LOGIC_VECTOR(7 downto 0);
    signal Temps: STD_LOGIC_VECTOR (31 downto 0);
    signal EN_SW: STD_LOGIC;

begin

    disp : entity work.Disp4D
           port map (A=>Temps(15 downto 12),
                     B=>Temps(19 downto 16),
                     C=>Temps(23 downto 20),
                     D=>Temps(27 downto 24),
                     CLK=>CLK,
                     S=>S,
                     ENAN=>ENAN);

    tick : entity work.TICK_1ms
           port map (CLK=>CLK, Tick=>EN_1ms);

    ms : entity work.Cnt0toN_EN generic map (N=>25)
          port map (CLK=>CLK, EN=>EN_SW, CLR=>CLR, Cout=>EN_ticks(0), Q=>Temps(3 downto 0));

    csu : entity work.Cnt0toN_EN generic map (N=>10)
          port map (CLK=>CLK, EN=>EN_ticks(0), CLR=>CLR, Cout=>EN_ticks(1), Q=>Temps(7 downto 4));

    csd : entity work.Cnt0toN_EN generic map (N=>6)
          port map (CLK=>CLK, EN=>EN_ticks(1), CLR=>CLR, Cout=>EN_ticks(2), Q=>Temps(11 downto 8));

    su : entity work.Cnt0toN_EN generic map (N=>10)
          port map (CLK=>CLK, EN=>EN_ticks(2), CLR=>CLR, Cout=>EN_ticks(3), Q=>Temps(15 downto 12));

    sd : entity work.Cnt0toN_EN generic map (N=>6)
         port map (CLK=>CLK, EN=>EN_ticks(3), CLR=>CLR, Cout=>EN_ticks(4), Q=>Temps(19 downto 16));

    mu : entity work.Cnt0toN_EN generic map (N=>10)
         port map (CLK=>CLK, EN=>EN_ticks(4), CLR=>CLR, Cout=>EN_ticks(5), Q=>Temps(23 downto 20));

    md : entity work.Cnt0toN_EN generic map (N=>6)
         port map (CLK=>CLK, EN=>EN_ticks(5), CLR=>CLR, Cout=>EN_ticks(6), Q=>Temps(27 downto 24));

    EN_SW <= EN and EN_1ms;
end Behavioral;
```

Annexe 4 :

DFM

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 04.01.2022 11:43:52
-- Design Name:
-- Module Name: DFM - 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;
use IEEE.NUMERIC_STD.ALL;

entity DFM is
    Port ( CLK : in STD_LOGIC;
           Btn : in STD_LOGIC;
           Niveau : out STD_LOGIC;
           Detect : out STD_LOGIC);
end DFM;

architecture Behavioral of DFM is

    signal BtnCurr,BtnPrev : STD_LOGIC:= '0';
    signal Q,D : STD_LOGIC_VECTOR(1 downto 0):= "00";
    signal EF,EP : STD_LOGIC:= '0';

begin
    --PARTIE 1
    process(CLK) begin
        if RISING_EDGE (CLK)then
        BtnCurr <= Btn;
        BtnPrev <= BtnCurr;
        end if;
    end process;

    --PARTIE 2 début
    D(1) <= (NOT(Q(1)) and Q(0)) or ((Q(1) and NOT(Q(0)))
    and (BtnCurr or BtnPrev));

    D(0) <= (NOT(Q(1)) and NOT(Q(0))) and(BtnCurr and
    NOT(BtnPrev));

    process(CLK) begin
        if RISING_EDGE (CLK)then
        Q(1) <= D(1);
        Q(0) <= D(0);
        end if;
    end process;
    -- PARTIE 2 fin

    -- PARTIE 3
    EF <= EP XOR Q(0);

    process(CLK) begin
        if RISING_EDGE (CLK)then
        EP <= EF;
        end if;
    end process;

    Detect <= Q(0);
    Niveau <= EP;

end Behavioral;
```