INF3500 : Conception et réalisation de systèmes numériques

Examen intra

21 octobre 2008

Durée: 2h.
Pondération: 20%.
Documentation: Une feuille recto-verso 8.5”×11” ou A4 manuscrite permise.
Calculatrice: Programmable permise.
Directives particulières: Ordinateurs interdits, répondre à toutes les questions, la valeur de chaque question est indiquée.
Ne posez pas de questions. En cas de doute sur le sens d’une question, énoncez clairement toute supposition que vous faites.

Question 1. (3 points)

La fonction majorité accepte un certain nombre d'entrées, les inspecte, et indique si le nombre de bits en entrée avec une valeur '1' est supérieur au nombre de bits en entrée avec une valeur de '0'. Par exemple, pour le vecteur d'entrée "11010", la sortie serait '1'. Considérez la déclaration d'entité suivante en VHDL:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity majorite is
    generic (
        W : positive := 5 -- le nombre de bits d'entrée
    );
    port (
        I : in std_logic_vector(W - 1 downto 0);
        F : out std_logic
    );
end majorite;
```

Donnez une architecture synthétisable pour cette entité. (Barème de correction : pour un maximum de 2 points sur 3, donnez une architecture synthétisable qui est valide uniquement pour W = 5. Pour un maximum de 3 points sur 3, donnez une architecture synthétisable qui est valide pour toute valeur de W.)

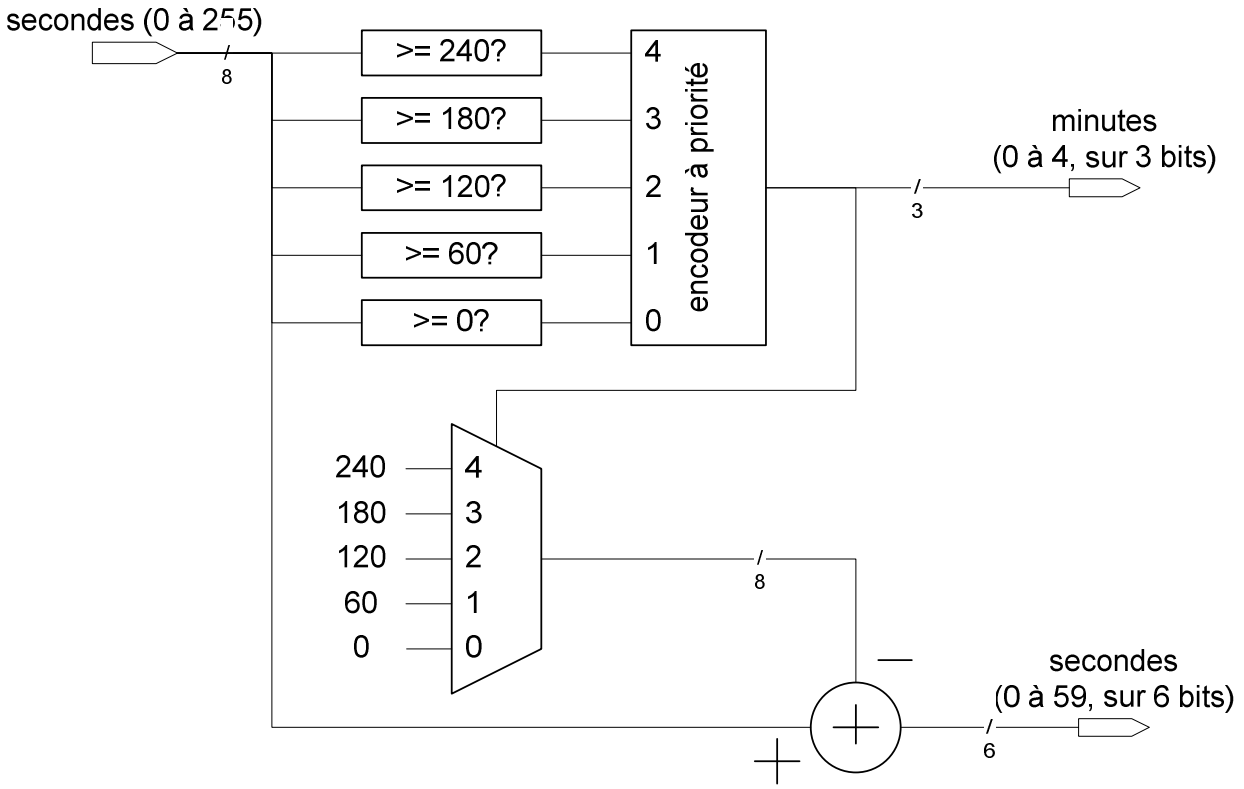
Solution :

```
architecture flotdedonnees of majorite is
-- *********************************
-- valide uniquement pour W = 5 !!!!!
-- *********************************
begin
    with I select
        F <= '1' when "00111" | "01011" | "01101" | "01110" |
                      "01111" | "10011" | "10101" | "10110" |
                      "10111" | "11001" | "11010" | "11011" |
                      "11100" | "11101" | "11110" | "11111",
             '0' when others;
end flotdedonnees;

architecture comportementale of majorite is
begin
    process (I)
    variable compte : integer := 0;
    begin
        compte := 0;
        for k in W - 1 downto 0 loop
            if I(k) = '1' then
                compte := compte + 1;
            end if;
        end loop;

        if (compte > W / 2) then
            F <= '1';
        else
            F <= '0';
        end if;

    end process;
end comportementale;
```

Question 2. (3 points)

Faites la conception d'un circuit numérique combinatoire qui accepte en entrée un vecteur de 8 bits représentant un nombre positif de secondes. Votre circuit doit avoir deux sorties binaires donnant le nombre de minutes et secondes correspondantes. Donnez un diagramme de votre circuit, ne donnez pas de code VHDL. Indiquez la largeur des ports de sortie en bits. Vous pouvez utiliser des opérations logiques et arithmétiques (mais pas la division ni le modulo), la comparaison, des multiplexeurs, décodeurs et encodeurs, et toute autre porte logique de base. Votre circuit doit être purement combinatoire, il ne doit pas inclure d'éléments à mémoire.

Solution :



Question 3. (4 points)

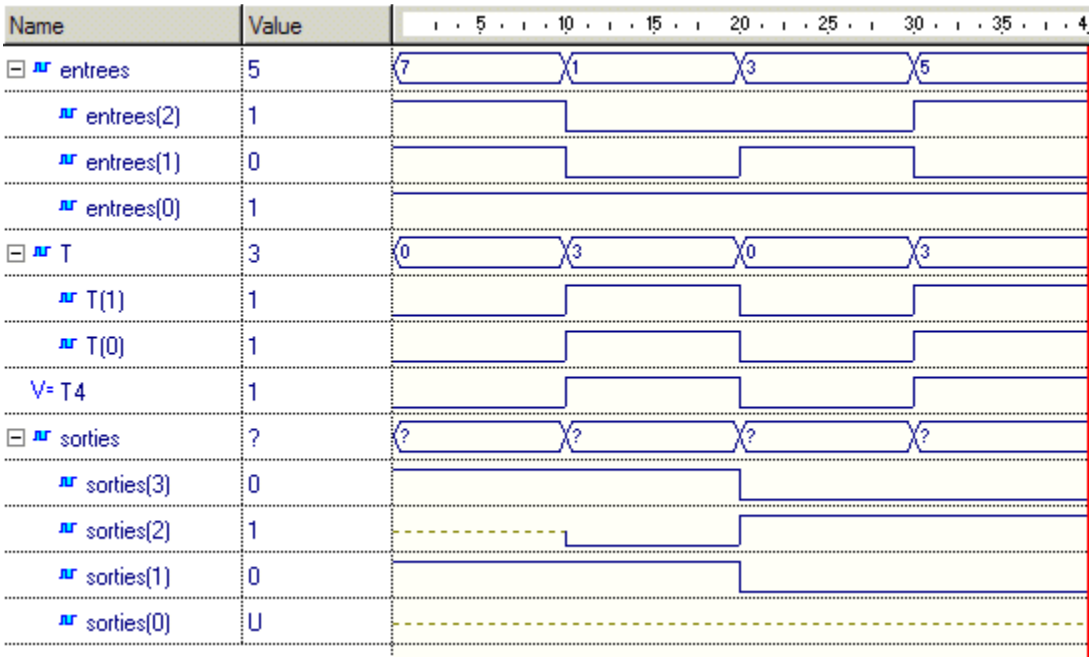
Considérez le code VHDL suivant pour un module combinatoire et son banc de test associé.

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;

entity sim is
    port (
        entrees : in std_logic_vector(2 downto 0);
        sorties : out std_logic_vector(3 downto 0)
    );
end sim;

architecture arch of sim is
signal T : std_logic_vector(1 downto 0);
begin

    T(1) <= entrees(1) xor entrees(0);
    sorties(3) <= entrees(2) xor T(1);

    process(entrees)
    begin
        T(0) <= entrees(1) xor entrees(0);
        sorties(2) <= entrees(2) xor T(0);
    end process;

    process(entrees)
    variable T4 : std_logic;
    begin
        T4 := entrees(1) xor entrees(0);
        sorties(1) <= entrees(2) xor T4;
    end process;

    process(entrees(0))
    begin
        sorties(0) <= entrees(2) xor T(0);
    end process;

end arch;
```

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;

entity simTB is
end simTB;

architecture arch of simTB is

component sim
port (
    entrees : in std_logic_vector(2 downto 0);
    sorties : out std_logic_vector(3 downto 0)
);
end component;

signal entrees : std_logic_vector(2 downto 0);
signal sorties : std_logic_vector(3 downto 0);

begin

    UUT : sim port map(entrees, sorties);
    entrees <= "111" after 0 ns,
        "001" after 10 ns,
        "011" after 20 ns,
        "101" after 30 ns;

end arch;
```

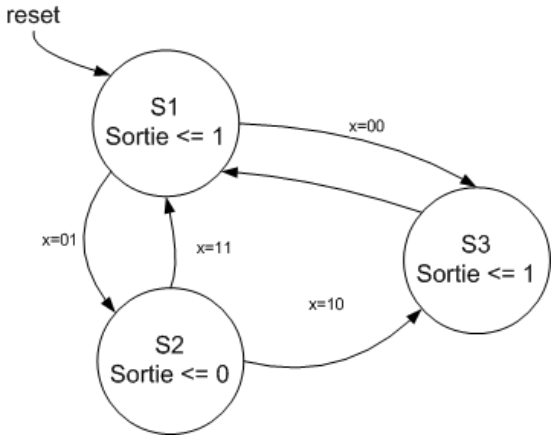
Donnez la valeur des signaux sorties(3), sorties(2), sorties(1) et sorties(0) aux temps de simulation 1 ns, 11 ns, 21 ns et 31 ns.

Solution :



Question 4. (4 points)

Considérez le diagramme d'états et la déclaration d'entité suivants. Les transitions ayant pour origine et destination le même état ne sont pas indiquées. Donnez le modèle VHDL correspondant en décrivant une architecture pour cette entité.




```
library IEEE;
use IEEE.std_logic_1164.all;
entity machineetats is
    port (
        reset, CLK : in STD_LOGIC;
        x : in STD_LOGIC_VECTOR(1 downto 0);
        sortie : out STD_LOGIC
    );
end machineetats;
```

Solution:

```
architecture arch of machineetats is
type type_etat is (S1, S2, S3);
signal etat : type_etat := S1;
begin
    process(CLK, reset) is
    begin
        if (reset = '0') then
           etat <= S1;
        elsif (rising_edge(CLK)) then
           case etat is
              when S1 =>
                 if x = "00" then
                    etat <= S3;
                 elsif x = "01" then
                    etat <= S2;
                 end if;
              when S2 =>
                 if x = "10" then
                    etat <= S3;
                 elsif x = "11" then
                    etat <= S1;
                 end if;
              when S3 =>
                 etat <= S1;
           end case;
        end if;
    end process;
    process(etat)
    begin
        case etat is
           when S1 | S3 =>
              sortie <= '1';
           when S2 =>
              sortie <= '0';
        end case;
    end process;
end arch;
```

Question 5. (3 points)

Un compteur Johnson est particulier parce que tous les '1' du compte sont toujours placés en un seul groupe qui se déplace. Par exemple, pour un compteur Johnson à quatre bits, la séquence serait : 0000, 0001, 0011, 0111, 1111, 1110, 1100, 1000, 0000, 0001, 0011, etc. On observe que le bit inséré en position la moins significative est égal à l'inverse du bit le plus significatif.

Donnez une architecture synthétisable pour la déclaration d'entité suivante en VHDL de façon à modéliser un compteur Johnson.

(Barème de correction : pour un maximum de 2 points sur 3, donnez une architecture synthétisable qui est valide uniquement pour W = 5. Pour un maximum de 3 points sur 3, donnez une architecture synthétisable qui est valide pour toute valeur de W.)

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;
use ieee.numeric_std.all;

entity compteurJohnson is
    generic (
        W : positive := 5
    );
    port (
        reset, clk: in std_logic;
        compte : out std_logic_vector(W - 1 downto 0)
    );
end compteurJohnson;
```

Solution

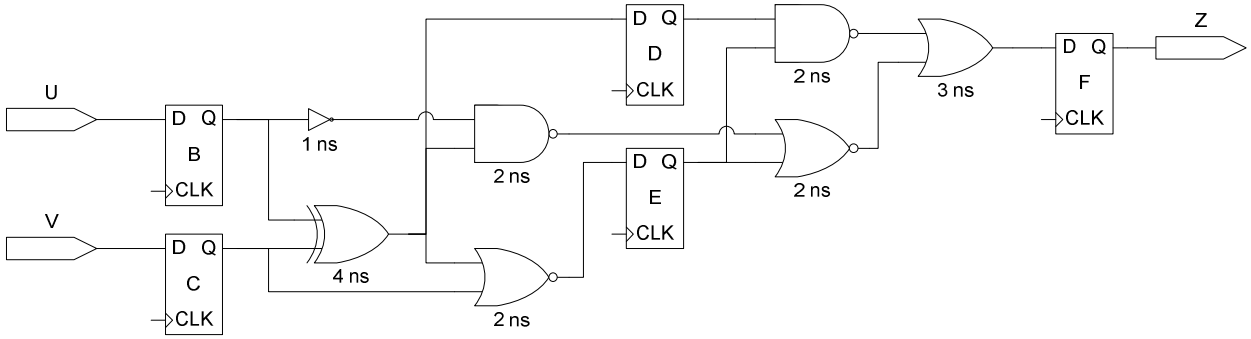
```
architecture arch1 of compteurJohnson is
begin

    process(clk, reset)
    variable c : std_logic_vector(W - 1 downto 0) := (others => 'X');
    begin
        if (rising_edge(clk)) then
            if reset = '1' then
                c := (others => '0');
            else
                c := c(W - 2 downto 0) & not(c(W - 1));
            end if;
        end if;
        compte <= c;
    end process;

end arch1;
```

Question 6. (3 points)

a. (1 point) Donnez deux causes possibles du déphasage d'horloge dans un système numérique.

b. (1 point) Considérez le circuit suivant. Les délais des portes logiques combinatoires sont indiqués. Les bascules ont un temps de propagation de 2 ns, un temps de préparation de 1 ns, et un temps de maintien de 0.5 ns. Toutes les bascules sont alimentées par le même signal d'horloge. Donnez le chemin critique en indiquant les composantes sur ce chemin, et donnez la fréquence maximale d'horloge.

c. (1 point) Un concepteur veut accélérer le circuit en plaçant un registre de pipeline sur le fil entre la porte NON-ET et la porte NON-OU. Cette approche est-elle valide? Quel en est l'effet?



Solution :

a. charges inégales sur le réseau de distribution, longueurs inégales dans le chemin de distribution, contrôle d'horloge par portes logiques (clock gating)

b. Chemin critique : B-xor-net-nou-ou-F : 2+4+2+2+3+1 = 14 ns -> fmax = 71.4 MHz

c. L'approche n'est pas valide et aurait pour effet de changer la fonction du circuit. Pour accélérer le circuit par pipelinage au point mentionné, il faudrait aussi insérer une bascule supplémentaire juste après la bascule D et juste après la bascule E.