Nom : _____Matricule : _____

INF3500 : Conception et réalisation de systèmes numériques
Examen intra - jeudi 18 octobre 2012

Durée: 2h; Pondération: 20%.

Documentation: Une feuille recto verso 8.5"×11" ou A4 permise.

Ordinateurs interdits. Calculatrice: programmable permise.

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. (4 points)

a. Donnez les micro-opérations implémentées par les chemins de données représentés à la Figure 1.

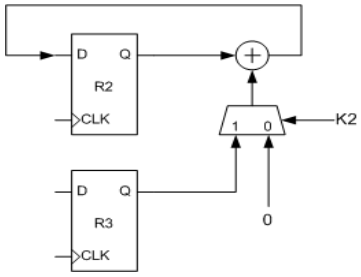


Figure 1.1

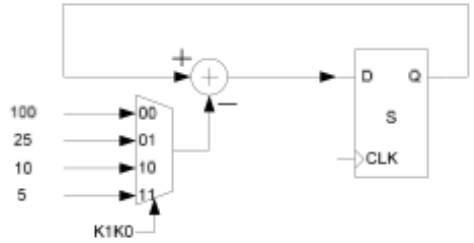


Figure 1.2

b. Donnez le circuit spécifié par le code suivant :

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity E1 is
	generic (
		n : positive := 3;
		valeurActive : std_logic := '1'
	);
	port(
		A : in std_logic_vector(n - 1 downto 0);
		F: out std_logic_vector(2 ** n - 1 downto 0)
	);
end E1;

architecture comportementale of E1 is

begin

	process(A)
	begin
		F <= (others => not(valeurActive));
		F(to_integer(unsigned(A))) <= valeurActive;
	end process;

end comportementale;
```

Question 2. (4 points)

a. (1.5 point)

i. Donnez le chronograme des ports G1, Q1, G2, Q2 pour le circuit suivant. Considérez les entrées D et CLK illustrées dans la Figure 2.

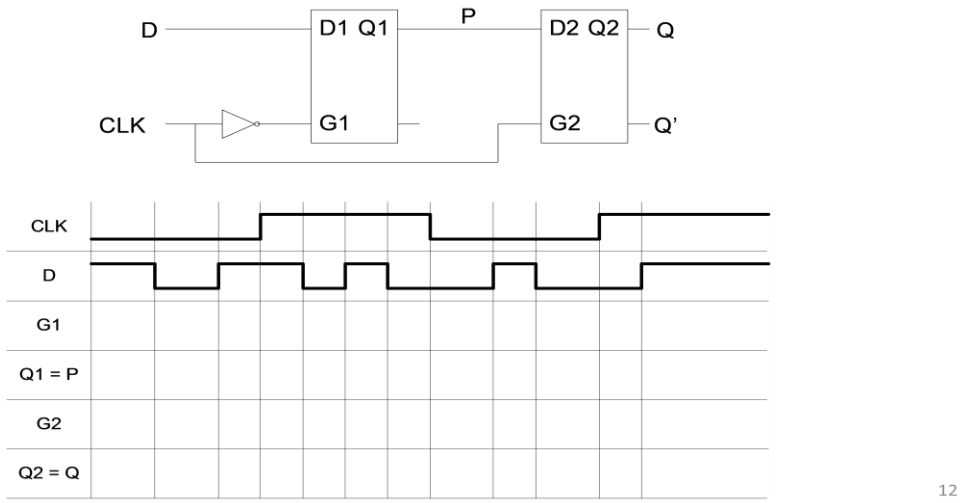


Figure 2

ii. Le circuit considéré au point i. représente l'implémentation d'un élément de mémoire de base. Spécifiez le type de cet élément mémoire.

b. (2.5 points) Donnez le circuit implémenté par le code donné à la Figure 3.

```
library IEEE;
use IEEE.std_logic_1164.all;

entity cctsequentielex1 is
        port (
                reset : in STD_LOGIC;
                CLK : in STD_LOGIC;
                X : in STD_LOGIC;
                Z : out STD_LOGIC
        );
end cctsequentielex1;

architecture arch1 of cctsequentielex1 is

signal A : STD_LOGIC; -- bascule A (en haut)
signal B : STD_LOGIC; -- bascule B (en bas)

begin

        process(CLK, reset) is
        begin
          if (reset = '0') then
                        A <= '0';
                        B <= '0';
          elsif (rising_edge(CLK)) then
                        A <= A et B;
                        B <= x xor not(B);
          end if;
        end process;

        -- signal de sortie
        z <= A et B;
end arch1;
```

Figure 3

Question 3. (4 points)

a. Dans certains cas, les circuits CPLD offrent une alternative intéressante aux FPGA. Donnez ces cas.
b. i. Donnez les fonctions de sortie (F0, F1, F2) du circuit programmable illustré dans la Figure 4.
   ii. Spécifiez le type de circuit programmable représenté à la Figure 4.

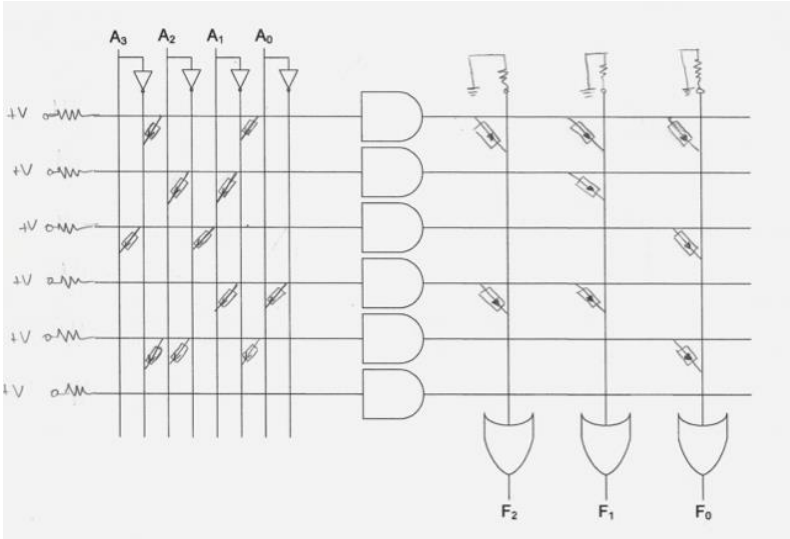


Figure 4

Question 4. (3 points)

Lors du dernier laboratoire, nous avons vu de quelle manière il est possible d'implémenter la division en matériel à l'aide d'une *look-up table*. En effet, pour effectuer l'opération a/b, nous mettions en mémoire morte (ROM) les résultats de 1/b afin de faire la multiplication a*(1/b).

a) Quels sont les nécessités en terme de ressources FPGA (slices, multiplieurs, mémoire) afin d'implémenter la table comportant les résultats de 1/b? (b est encodé sur 8 bits et le résultat de 1/b comporte 8 bits de partie entière et 2 bits de partie fractionnaire, en virgule fixe.
b) Discutez de la plus grande limitation de cette approche. Donnez une autre méthode pour faire le calcul et discutez ses avantages et inconvénients.
c) En utilisant le code VHDL suivant, que vaut le signal sortie ?

```
        ……
component frac is
        generic(
                Went : integer := 8;
                Wfrac : integer := 4
        );
        port (
                a : in  unsigned(Went - 1 downto 0);
                b : in  unsigned(Went - 1 downto 0);
                f : out unsigned(Went + Wfrac - 1 downto 0)
        );
end component frac;

signal a,b : unsigned(7 downto 0);

begin

frac1:  frac
        GENERIC MAP(Went  => 8, --nombre bit de la partie entière
                        Wfrac => 2) –-nombre de bit de la partie fractionnaire
                PORT MAP(a =>num,    --numérateur
                        b =>den,    --dénominateur
                        f =>res);   --résultat de la division

a <= "00001000";
b <= "00000010";
sortie <= res*to_unsigned(3,10)
```

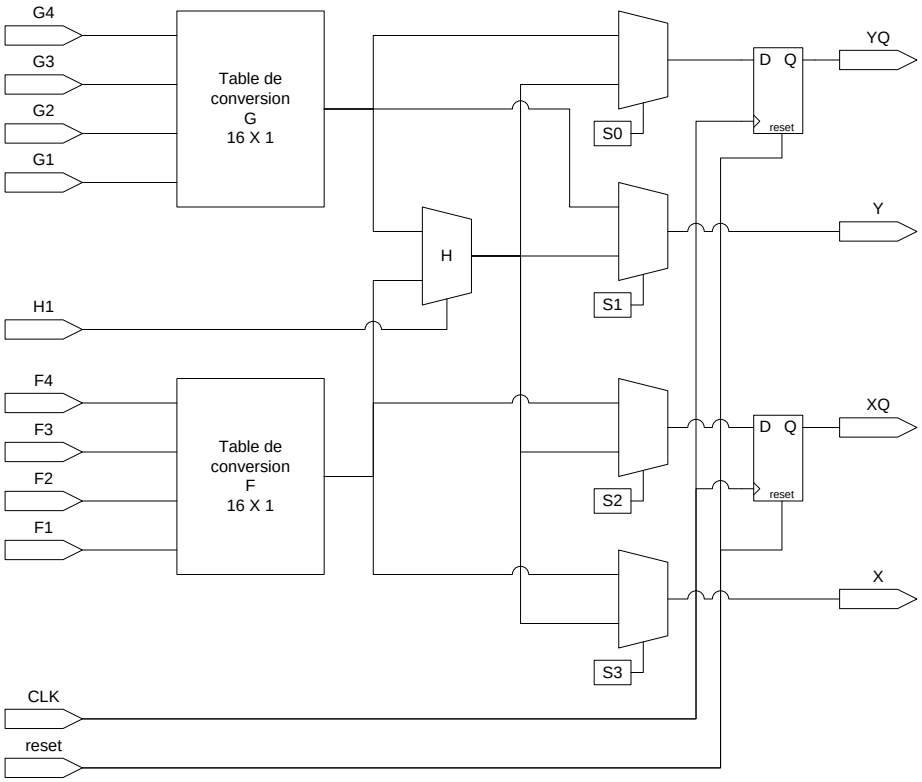
Question 5. (5 points)

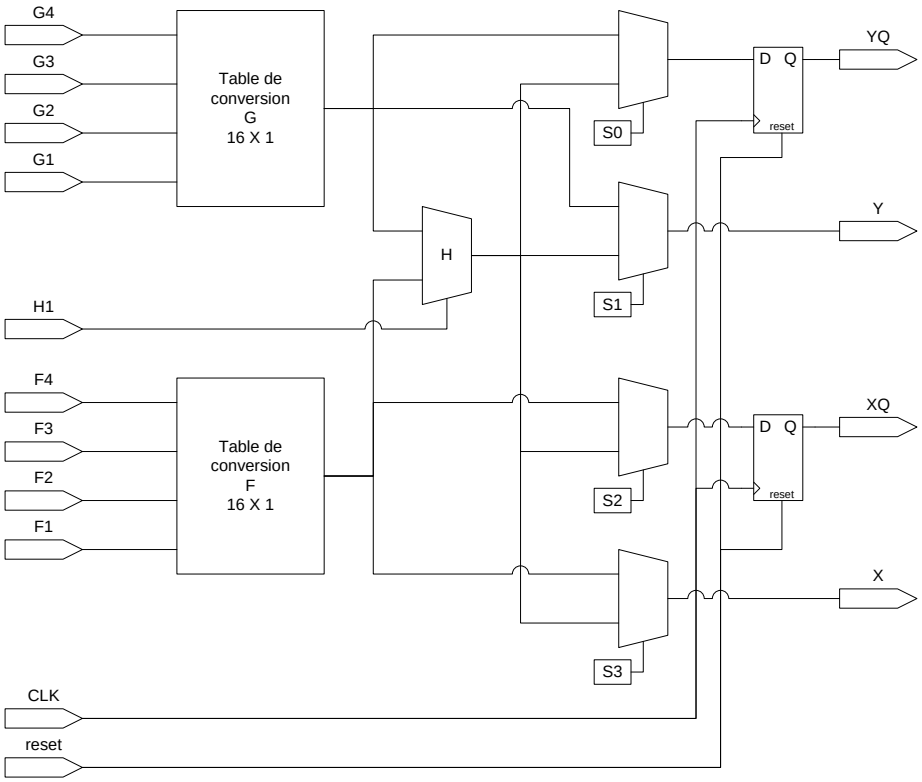
Considérez le code VHDL suivant. Montrez, sur le modèle de FPGA fourni aux pages suivantes, un résultat possible de la synthèse et de l'implémentation de ce code.

a. Indiquez directement sur le dessin où chaque signal se situe ainsi que les interconnections entre les blocs.

b. Indiquez dans les tables de vérité fournies le contenu de chacune des tables de conversion que vous utilisez.

```
library IEEE;
use IEEE.std_logic_1164.all;

entity module2 is
	port (
		CLK, A, B, C, D : in std_logic; F : out std_logic
	);
end module2;

architecture arch of module2 is
signal S : std_logic_vector(1 downto 0);
begin

	process (CLK)
	begin
		if rising_edge(CLK) then
			S(0) <= A or B or C or D;
			S(1) <= A or B or D;
		end if;
	end process;

	process (S, A, C, D)
	begin
		if S(1) = '1' then
			F <= A or B;
		else F <= S(0) or C or D or A;
	end if;
	end process;
end arch;
```





Feuille de réponse pour la question 5 a.

G4	G3	G2	G1	G
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

F4	F3	F2	F1	F
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

G4	G3	G2	G1	G
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

F4	F3	F2	F1	F
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

Feuille de réponse question 5.b