

**NOM:****Code permanent:**

---

**IFT1227 – Architecture des ordinateurs I**  
**Examen Intra – Automne 2014**

**Chargée de cours :** Alena Tsikhanovich**Durée :** 2 heures**Directives :**

- Toute documentation est permise,
  - Tous les appareils électroniques (cellulaire etc.) sont interdits à l'exception d'une calculatrice,
  - Inscrivez votre nom et code permanent sur l'entête du questionnaire et du cahier d'examen,
  - Répondez aux questions directement sur la feuille du questionnaire et/ou dans le cahier d'examen,
  - Veillez à la clarté de vos réponses et justifiez-les,
  - Si une question vous semble ambiguë, faites une interprétation, expliquez-la et continuez.
- 

**Note :**

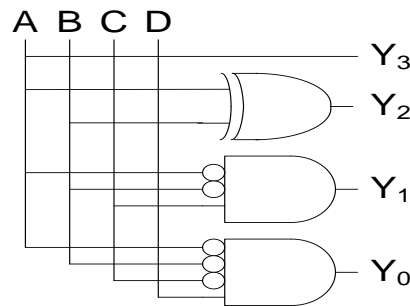
Question 1	/ 25
Question 2	/ 25
Question 3	/ 25
Question 4	/ 25
<b>Total</b>	<b>/ 100</b>

NOM: \_\_\_\_\_

Code permanent: \_\_\_\_\_

**Question 1. Circuits combinatoires, simplification (25 pts)**

- a) (5 pts) Écrire les équations booléennes implémentant le circuit présenté plus bas.

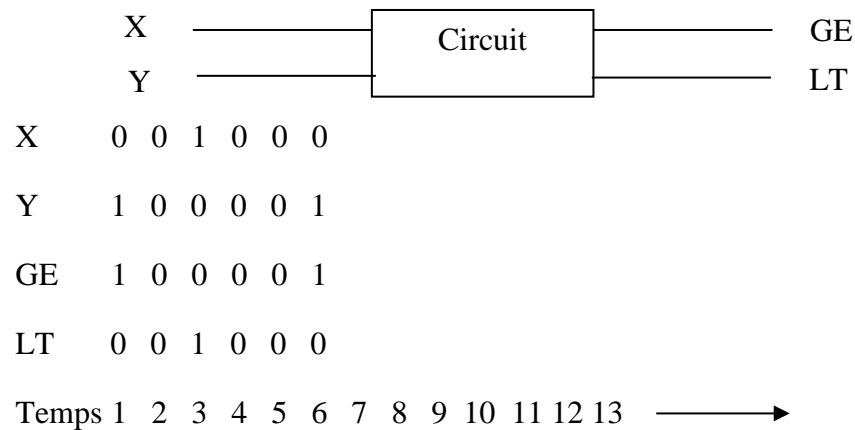


- b) (5 pts) Donner une implémentation de la fonction  $Y_0$  en utilisant un multiplexeur 8-à-1.
- c) (5 pts) Donner une implémentation de la fonction  $Y_2$  en utilisant un décodeur et une porte OU logique.
- d) (10 pts) La fonction  $Y$  est définie par la table de vérité suivante. Simplifiez la fonction  $Y$  en utilisant la méthode de Quine-McCluskey.

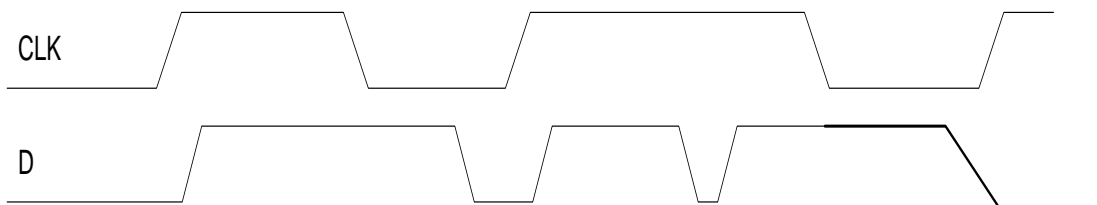
A	B	C	D	Y
0	0	0	0	1
0	0	0	1	X
0	0	1	0	1
0	0	1	1	X
0	1	0	0	0
0	1	0	1	X
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	X
1	0	1	0	1
1	0	1	1	X
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

**NOM:****Code permanent:****Question 2. Circuits séquentiels (25 pts)**

- a) (15) Dessinez un diagramme d'états de la machine à états finis d'un circuit qui trie deux mots binaires X et Y entrés en parallèle en série les bits les plus significatifs d'abord. Ces deux mots doivent apparaître sur deux sorties GE et LT. Sur la sortie GE doit apparaître le mot le plus grand ou égal de deux et le plus petit sur la sortie LT. Voici l'illustration du comportement de ce circuit.



- b) (5 pts) Dessiner un chronogramme du signal Q (wave) en supposant que Q est une sortie de verrou (latch) et ensuite une sortie de bascule D (flipflop).

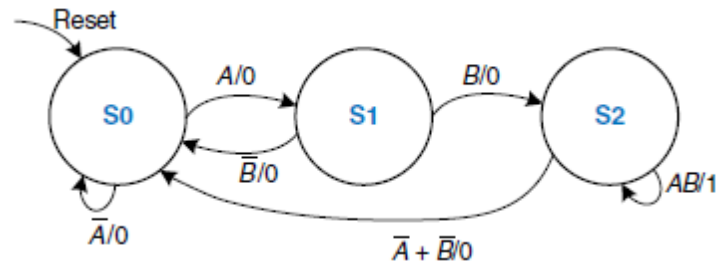


Q (verrou/latch)

Q (bascule/flipflop)

**NOM:****Code permanent:**

- c) (5 pts) Considérer la machine à états finis suivante. Donner une table de transition d'états. Mettre à côté des noms d'états symboliques les deux encodages des états : encodage binaire et encodage « one hot ».



### Question 3. VHDL (25 pts)

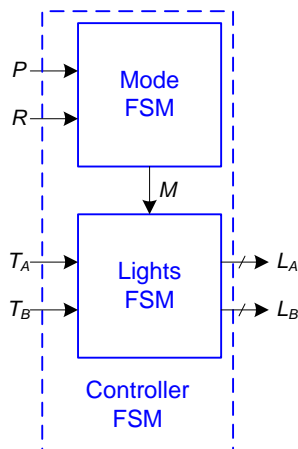
- a) (15 pts) Supposons que les implémentations de deux modules `modFSM` et `lightsFSM` existent déjà.

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

```
entity modFSM is
  port (P,R: in STD_LOGIC;
        clk: in STD_LOGIC;
        M:   out STD_LOGIC);
end;
```

```
...
entity lightsFSM is
  port (Ta,Tb, M:      in STD_LOGIC;
        clk:          in STD_LOGIC;
        La, Lb       : out STD_LOGIC_VECTOR (2 downto 0));
end;
```

Écrire un modèle structurel du module `controllerFSM`. Le schéma de ce module est présenté plus bas.



**NOM:****Code permanent:**

- b) (10 pts) Codez un module VHDL comportemental d'un circuit implémentant les trois fonctions logiques définies par les équations booléennes suivantes.

$$F_1 = A \oplus B$$

$$F_2 = ABC + \bar{A}\bar{B}D$$

$$F_3 = ABCD + \bar{A}\bar{B}CD$$

**Question 4. FSM en VHDL (25 pts)**

Considérez le code VHDL de la machine à états finis suivant :

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

entity fsm1 is
port (clk, reset: in STD_LOGIC;
      a, b:      in STD_LOGIC;
      z :       out STD_LOGIC);
end;

architecture synth of fsm1 is
type statetype is (S0, S1, S2, S3);
signal state, nextstate: statetype;
signal ab: STD_LOGIC_VECTOR(1 downto 0);
begin
-- state register
process (clk, reset) begin
    if reset = '1' then state <= S0;
        elsif clk'event and clk = '1' then
            state <= nextstate;
        end if;
end process;
-- next state logic
ab <= a & b;
process (all) begin
    case state is
        when S0 =>
            case (ab) is
                when "00" => nextstate <= S0;
                when "01" => nextstate <= S0;
                when "10" => nextstate <= S3;
                when "11" => nextstate <= S1;
                when others => nextstate <= S0;
            end case;
        when S1 =>
            case (ab) is
                when "00" => nextstate <= S0;
                when "01" => nextstate <= S2;
```

**NOM:****Code permanent:**

---

```
        when "10" => nextstate <= S3;
        when "11" => nextstate <= S1;
        when others => nextstate <= S0;
    end case;
when S2 =>
    case (ab) is
        when "00" => nextstate <= S0;
        when "01" => nextstate <= S2;
        when "10" => nextstate <= S3;
        when "11" => nextstate <= S1;
        when others => nextstate <= S0;
    end case;
when S3 =>
    case (ab) is
        when "00" => nextstate <= S0;
        when "01" => nextstate <= S2;
        when "10" => nextstate <= S3;
        when "11" => nextstate <= S1;
        when others => nextstate <= S0;
    end case;
    when others => nextstate <= S0;
end case;
end process;

-- output logic
process(all) begin
    case state is
        when S0 =>
            if ab = "11" then z <= '1';
            else z <= '0';
            end if;
        when S1 =>
            if ab = "00" then z <= '0';
            else z <= '1';
            end if;
        when S2 =>
            if ab = "11" then z <= '1';
            else z <= '0';
            end if;
        when S3 =>
            if ab = "00" then z <= '0';
            else z <= '1';
            end if;
        when others => z <= '0';
    end case;
end process;
end;
```

**NOM:****Code permanent:**

---

- a) (5 pts) La machine de quel type est codée (Moore ou Melay, justifier votre réponse) ?
- b) (15 pts) À partir du code dessiner un automate représentant cette machine.
- c) (5 pts) Dans le bout de code ici-bas identifier le mécanisme de synchronisation du signal reset. Montrer le code avec la synchronisation différente de celle trouvée.

```
process (clk, reset) begin
    if reset = '1' then state <= S0;
        elsif clk'event and clk = '1' then
            state <= nextstate;
        end if;
end process;
```