

# École Polytechnique de Montréal

Département de génie informatique et génie logiciel

INF1500 – Logique des systèmes numériques

Vendredi le 16 décembre 2011, 13h30

## Examen final

---

### **Directives :**

- Le quiz est sur 20 points;
- La pondération pour la session est de 40%;
- La documentation et la calculatrice (programmable ou non-programmable) ne sont pas permises;
- Ne pas utiliser un crayon rouge pour répondre aux questions;
- Répondre sur le(s) cahier(s) d'examen de l'École et le(s) remettre;
- Répondre sur les pages de droite des cahiers d'examen et réserver celles de gauche pour produire un brouillon qui ne sera pas considéré par le correcteur;
- Glisser dans le cahier d'examen principal les pages du questionnaire prévues pour répondre à certaines questions spécifiques après les avoir détachées du questionnaire;
- Pour les questions à développement, prenez soin d'exprimer clairement vos arguments, car la correction en tiendra compte;
- La durée est de 150 minutes.

## 1. Vrai ou Faux (1.5 points)

Répondre par vrai ou par faux aux questions suivantes. Justifiez votre réponse. Considérez 0.25 points par justification (la bonne réponse sans justification ne compte pour rien).

- Dans une addition de 2 nombres en complément à deux il n'est pas possible d'avoir une retenue tout en ayant un résultat valide. Expliquez en donnant un exemple.
- Le mot clef **port** définit en VHDL l'interface entre le monde extérieur et l'entité. L'expression qu'il introduit énumère les ports du circuit et leur direction seulement.
- En VHDL le modèle d'un circuit numérique par flot de données décrit sa fonction sans nécessairement définir sa structure.
- Un processus doit contenir obligatoirement une liste de sensibilité statique et une instruction explicite d'attente *wait*
- Pour une fonction donnée  $F(A, B, C): \Sigma_{A,B,C}(3,5,6,7) = \Pi_{A,B,C}(1,2,4)$  (réponse F)
- Les compteurs asynchrones ont le même comportement que ceux de type synchrone.

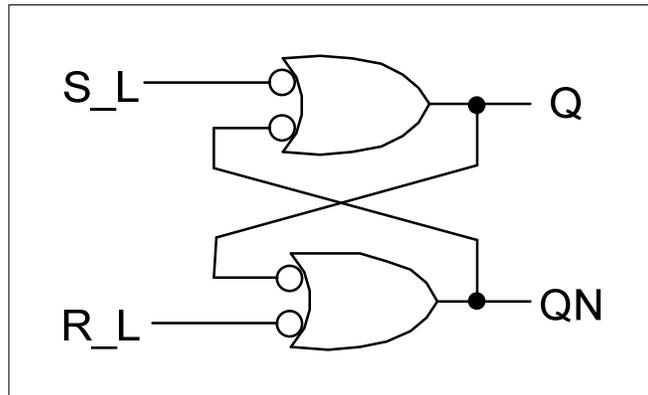
## 2. Circuits combinatoires – MSI (3 points)

- Soit la fonction sous forme canonique  $F(A,B,C,D) = \Sigma_{A,B,C,D} (4, 6, 7, 8, 14, 15)$ .
  - Donnez la table de vérité correspondante (**Utilisez la page prévue qui se situe à la fin du questionnaire et incluez-la dans vos cahiers d'examen**). (0.5 point)
  - Implémentez la fonction à l'aide d'un multiplexeur 4-a-1 et de quelques autres portes de base, où A et B sont les entrées « *select* » Considérez les entrées et la sortie en logique mixte actif-haut. (1 point)
- Soit la fonction sous forme canonique  $f(a,b,c,e) = \Sigma_{a,b,c,e} (1,3,7,9,15)$ .
  - Donnez la table de vérité correspondante (**Utilisez la page prévue qui se situe à la fin du questionnaire et incluez-la dans vos cahiers d'examen**). (0.5 point)

- 2) Implémentez la fonction à l'aide d'un décodeur avec 3 entrées actives au niveau haut (correspondant au « a », « b », et « c » dans la fonction « f »), 8 sorties actives au niveau bas et un signal d'activation de type *enable*, EN, (correspondant à « e » dans la fonction « f ») actif-bas et le plus petit nombre possible de portes NAND (et synonymes). L'utilisation des inverseurs et OOPS est interdite. (1 point).

### 3. Loquet et bascule (3.5 points)

En considérant ce circuit :



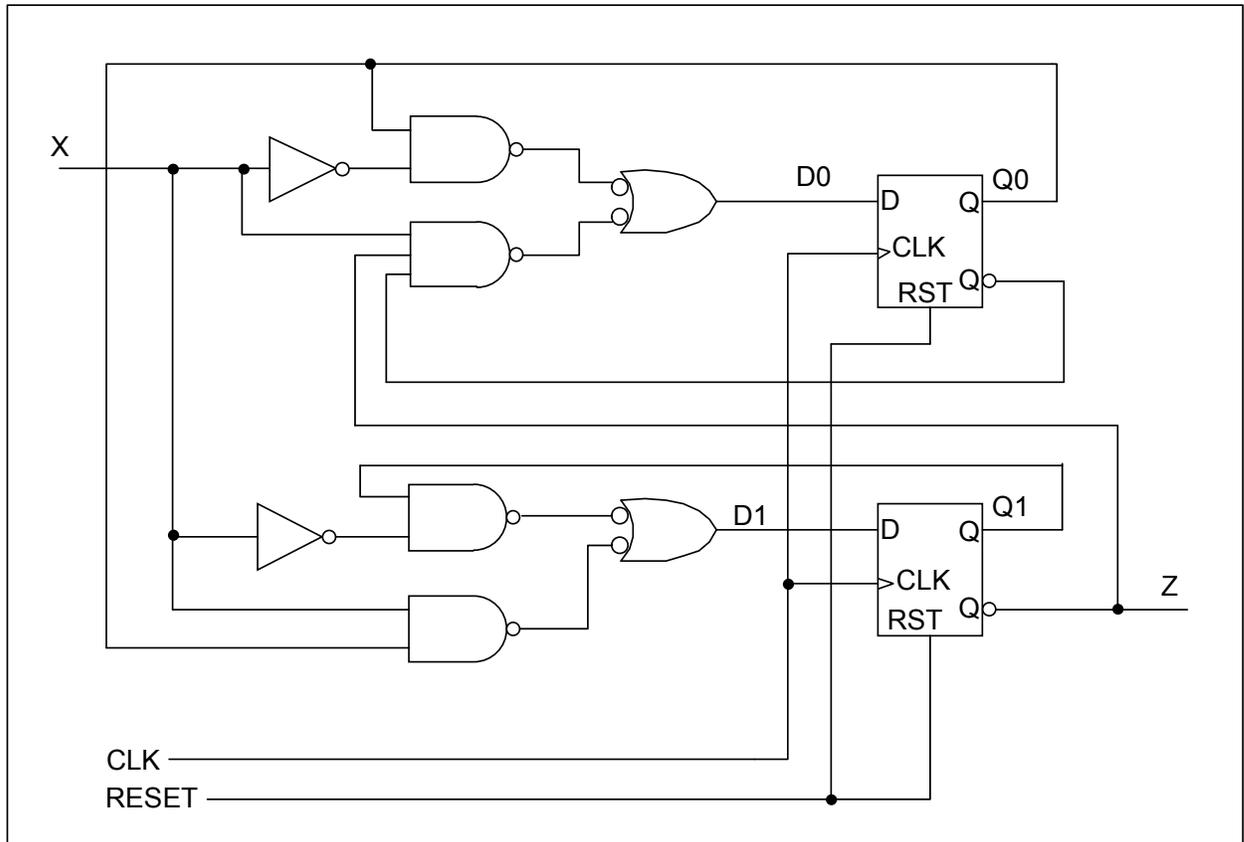
- a) Donnez la table de vérité avec les deux entrées  $S\_L$  et  $R\_L$  et pour les deux sorties  $Q$  et  $QN$ . (1 point)
- b) Cette affirmation est tirée directement des notes de cours :  
 « Afin d'éviter les résultats non prévisibles (non déterministes), les entrées doivent changer une à la fois, laissant suffisamment de temps entre deux entrées successives pour permettre des états stables (c'est de là que proviennent les temps de stabilisation et de maintien). »

En vous inspirant de l'affirmation précédente, redessinez le circuit (plus d'une fois si la situation demande une explication qui le justifie) et montrez deux transitions arrivant au même moment sur les entrées  $S\_L$  et  $R\_L$  et expliquez leurs effets sur les sorties  $Q$  et  $QN$ . Utiliser une notation  $0 \rightarrow 1$  ou  $1 \rightarrow 0$  pour désigner les transitions en question. (1.5 point).

- c) Quelle imperfection physique du circuit fera éventuellement que les valeurs de sorties pourraient se stabiliser de façon difficilement prévisible dans la situation décrite en b). (1 point).

#### 4. Analyse de circuit séquentiel (4.5 points)

En considérant le circuit suivant :



- Donnez les équations de transitions et de sortie du circuit. (1 point)
- Donnez la table des transitions et sortie du circuit. (1 point)
- Dites s'il s'agit d'une machine de Moore ou de Mealy en justifiant votre réponse. (0.5 point)
- Donnez la table symbolique des états et dites combien d'états forment cette machine et pourquoi. (1 point)
- Tracez le diagramme des états du circuit et dites combien d'états forment cette machine. (1 point)

## 5. Synthèse de circuit séquentiel (5 points)

Concevez une machine de Moore respectant le comportement suivant :

- Un signal «reset», actif au niveau haut, doit initialiser le système.
- On désire détecter que les signaux A, B et C sont, chacun à tour de rôle, à 1 sur trois cycles d'horloge successifs. Si oui, la sortie F prend la valeur 1 avant de pouvoir recommencer une autre séquence de détection. Dans tous les autres cas, F est à zéro.
- Lorsque A, B ou C doit être égale à 1 pour respecter la séquence à un cycle d'horloge précis, les deux autres signaux du trio peuvent prendre n'importe quelle autre valeur.

- Donnez le diagramme des états (2 points)
- Donnez la table de transition des états, avant assignation (1.5 point)
- Donnez la table de transition des états après une assignation des états de type simple en vous basant sur la table obtenue en b). (0.5 point)
- Tracez un circuit avec des bascules D en représentant les circuits combinatoires en entrées et en sortie sous forme d'ovales mais sans préciser leur équation (ce qui serait trop long ici si on devait faire toutes les simplifications). Assurez-vous simplement d'identifier les entrées potentielles de ces fonctions. (1 point)

## 6. Le langage VHDL et les laboratoires (2.5 points)

Donnez le diagramme temporel obtenu suite à l'exécution du code VHDL suivant implémentant une machine à états finis. **(Utilisez la page prévue qui se situe à la fin du questionnaire et incluez-la dans vos cahiers d'examen)**

```

library ieee;
  use ieee.std_logic_1164.all;

entity MSAExamen is
  Port (
    -- Inputs
    -- horloge et reset globals
    clk          : in std_logic;
    rst          : in std_logic;

    X            : in std_logic; -- signal d'entree

    -- Outputs
  
```

```

        Z          : out std_logic_vector(2 downto 0)
    );
end entity MSAExamen;

architecture Behav of MSAExamen is

    signal etat_present , etat_suivant : std_logic_vector(2 downto 0);

begin

    MSA_synch : process(rst , clk)
    begin
        if (rst = '1') then
            etat_present <= "000";
        elsif clk'event and clk = '1' then
            etat_present <= etat_suivant;
        end if;
    end process;

    MSA_asynch : process(etat_present, X)
    begin
        case etat_present is
            when "000" =>
                etat_suivant <= "001";
                Z <= (others => '0');

            when "001" =>
                if (X = '0') then
                    etat_suivant <= "010";
                else
                    etat_suivant <= "011";
                end if;
                Z <= etat_present;

            when "010" =>
                if (X = '0') then
                    etat_suivant <= "011";
                else
                    etat_suivant <= "110";
                end if;
                Z <= "010";

            when "011" =>
                if (X = '0') then
                    etat_suivant <= "110";
                else
                    etat_suivant <= "010";
                end if;
        end case;
    end process;
end architecture;

```

```

        Z <= "011";

        when "110" =>
            if (X = '0') then
                Z <= "100";
            else
                Z <= etat_present;
            end if;
            etat_suivant <= "111";

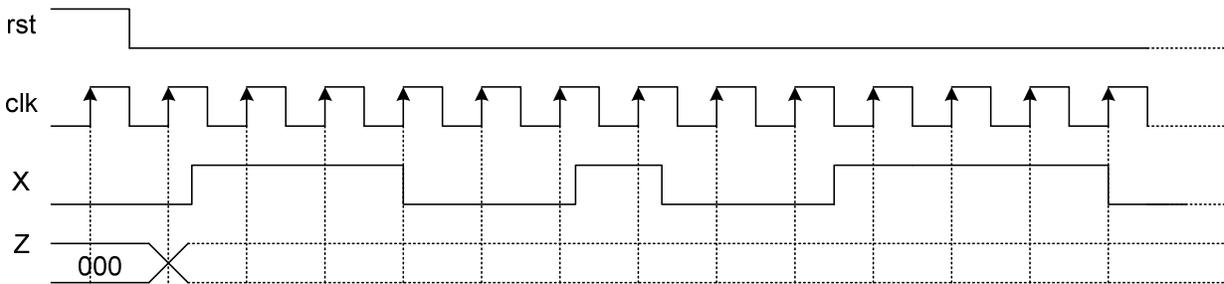
        when "111" =>
            if (X = '0') then
                Z <= "101";
            else
                Z <= "111";
            end if;
            etat_suivant <= (others => '0');

        when others =>
            etat_suivant <= "000";
            Z <= (others => '0');

    end case;
end process;

end architecture Behav;

```





**INF1500 – Logique des systèmes numérique – Automne 2011**

**NOM :** \_\_\_\_\_ **Prénom :** \_\_\_\_\_

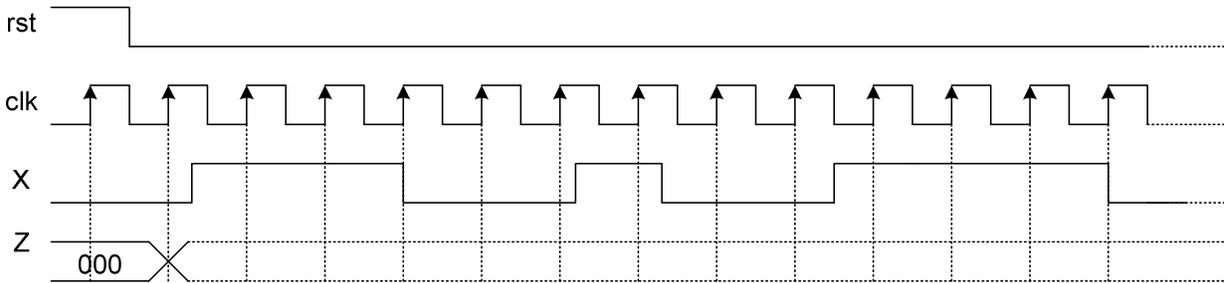
**Page pour répondre à la question 2 b)**

<b>a</b>	<b>b</b>	<b>c</b>	<b>e</b>	<b>f</b>

**INF1500 – Logique des systèmes numérique – Automne 2011**

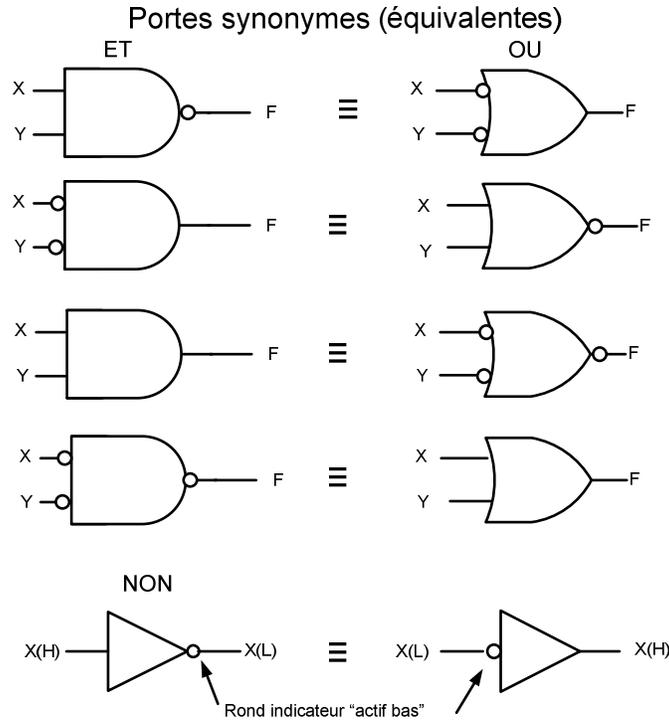
**NOM :** \_\_\_\_\_ **Prénom :** \_\_\_\_\_

**Page pour répondre à la question 6**



Zinit=000

# Annexes



## Théorèmes de Boole

**Table 4-1**  
Switching-algebra  
theorems with one  
variable.

(T1)	$X + 0 = X$	(T1')	$X \cdot 1 = X$	(Identities)
(T2)	$X + 1 = 1$	(T2')	$X \cdot 0 = 0$	(Null elements)
(T3)	$X + X = X$	(T3')	$X \cdot X = X$	(Idempotency)
(T4)	$(X')' = X$			(Involution)
(T5)	$X + X' = 1$	(T5')	$X \cdot X' = 0$	(Complements)

**Table 4-2** Switching-algebra theorems with two or three variables.

(T6)	$X + Y = Y + X$	(T6')	$X \cdot Y = Y \cdot X$	(Commutativity)
(T7)	$(X + Y) + Z = X + (Y + Z)$	(T7')	$(X \cdot Y) \cdot Z = X \cdot (Y \cdot Z)$	(Associativity)
(T8)	$X \cdot Y + X \cdot Z = X \cdot (Y + Z)$	(T8')	$(X + Y) \cdot (X + Z) = X + Y \cdot Z$	(Distributivity)
(T9)	$X + X \cdot Y = X$	(T9')	$X \cdot (X + Y) = X$	(Covering)
(T10)	$X \cdot Y + X \cdot Y' = X$	(T10')	$(X + Y) \cdot (X + Y') = X$	(Combining)
(T11)	$X \cdot Y + X' \cdot Z + Y \cdot Z = X \cdot Y + X' \cdot Z$			(Consensus)
(T11')	$(X + Y) \cdot (X' + Z) \cdot (Y + Z) = (X + Y) \cdot (X' + Z)$			

**Table 4-3** Switching-algebra theorems with  $n$  variables.

(T12)	$X + X + \dots + X = X$	(Generalized idempotency)
(T12')	$X \cdot X \cdot \dots \cdot X = X$	
(T13)	$(X_1 \cdot X_2 \cdot \dots \cdot X_n)' = X_1' + X_2' + \dots + X_n'$	(DeMorgan's theorems)
(T13')	$(X_1 + X_2 + \dots + X_n)' = X_1' \cdot X_2' \cdot \dots \cdot X_n'$	
(T14)	$[F(X_1, X_2, \dots, X_n, +, \cdot)]' = F(X_1', X_2', \dots, X_n', \cdot, +)$	(Generalized DeMorgan's theorem)
(T15)	$F(X_1, X_2, \dots, X_n) = X_1 \cdot F(1, X_2, \dots, X_n) + X_1' \cdot F(0, X_2, \dots, X_n)$	(Shannon's expansion theorems)
(T15')	$F(X_1, X_2, \dots, X_n) = [X_1 + F(0, X_2, \dots, X_n)] \cdot [X_1' + F(1, X_2, \dots, X_n)]$	