Nom : _____ Matricule : _____

INF3500 : Conception et réalisation de systèmes numériques

Examen intra #1 – jeudi 5 février 2015

Durée: 1 heure. Documentation: Une feuille recto verso 8.5"×11" ou A4 permise.
Pondération: 10%. Calculatrice: Programmable permise.
Directives particulières:

- Ordinateurs interdits. Appareils mobiles interdits.
- Répondre à toutes les questions, la valeur de chaque question est indiquée.
- Répondre sur le questionnaire et le remettre.
- Ne posez pas de questions. En cas de doute sur le sens d'une question, énoncez clairement toute supposition que vous faites.

Question 1. (2 points) Réponses brèves

a. (0.5 point) Donnez 3 exemples de types scalaires et 2 exemples types composés du langage VHDL.

Types scalaires	Types composés

b. (0.5 point) Donnez 5 valeurs possibles d'un signal de type `std_logic` en VHDL, et expliquez la signification de ces cinq valeurs dans un circuit électrique.

Valeur	Signification dans un circuit électrique

c. (0.5 point) Quelle est la différence entre la simulation d'un module VHDL avant sa synthèse et après son implémentation?

d. (0.5 point) Dans le laboratoire #1, quelle(s) information(s) contenait le fichier *.ucf ?

Question 2. (2 points)

Donnez un circuit CMOS pour réaliser la fonction logique T de la table de vérité suivante.

Les entrées A, B et C ne sont pas disponibles en version inversée.

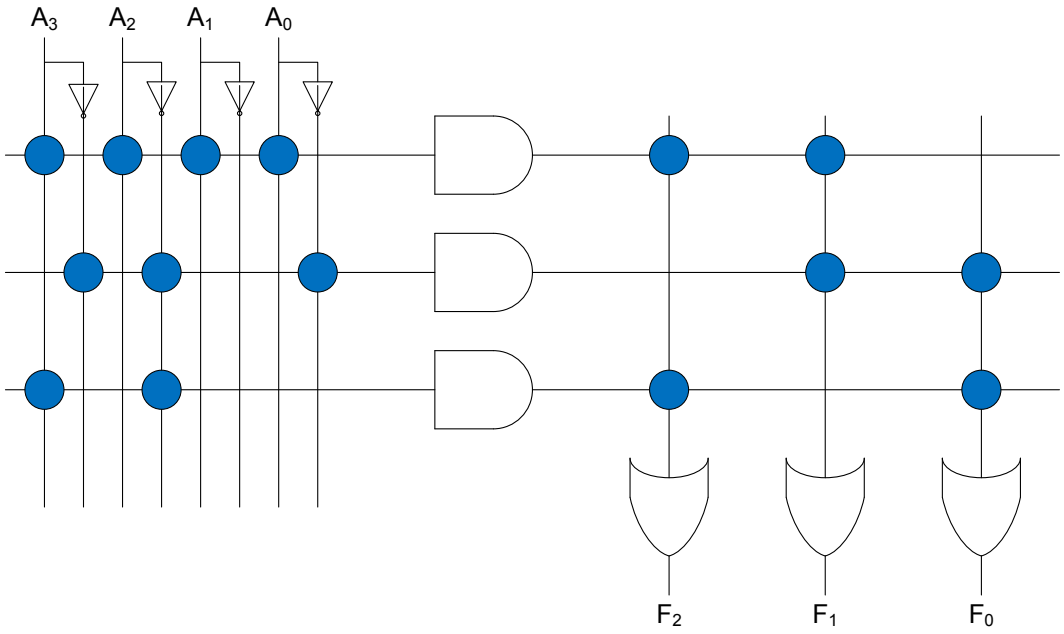
Utilisez le moins de transistors possible.

A	*B*	*C*	*T*
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

(*Utilisez le verso si nécessaire*)

Question 3. (3 points)

Donnez une description en VHDL synthétisable qui correspond au circuit implémenté sur le PLA suivant.



(*Utilisez le verso si nécessaire*)

Question 4. (3 points)

Les compresseurs sont utilisés dans les circuits arithmétiques pour accélérer l'addition de plusieurs nombres. Un compresseur M:N accepte en entrée un vecteur V de M bits et indique le nombre de bits de valeur '1' de V, sur un mot de N bits. Par exemple, pour M = 3 et N = 2, et le vecteur V = [110], le compresseur donnerait la sortie 10 puisque 2 des bits de V sont à 1. On doit avoir $2^N - 1 \geq M$. Donnez une architecture synthétisable pour l'entité suivante selon ces spécifications.

Note : la fonction `to_unsigned(entier, N)` donne la représentation d'un entier sur N bits avec le type `unsigned`.

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

entity compresseurMN is
 generic (
   M : positive := 4;
   N : positive := 3
 );
 port (
   V: in std_logic_vector(M - 1 downto 0);
   nombre1 : out unsigned(N - 1 downto 0)
 );
end compresseurMN;
```

Barème de correction: code qui fonctionne pour M = 4 et N = 3, 1 point; code qui fonctionne pour toute valeur conforme de M et N, 2 points ; vérification de la conformité de M et N avec message d'erreur lors de la simulation ou synthèse, +0.5 point ; syntaxe correcte de VHDL, +0.5 point.

(*Utilisez le verso si nécessaire*)

Solutions

1.

a. scalaires : boolean, bit, character, integer, positive, real, std_logic, etc.

composés: std_logic_vector, unsigned, signed, bit_vector, string, etc.

b. ‘U’ : valeur inconnue, pas initialisée - pas de sens physique, seulement utile en simulation

‘X’ : valeur inconnue forcée: court-circuit

‘0’ : 0 forcé: mise à la masse

‘1’ : 1 forcé: relié à l'alimentation

‘Z’ : haute impédance, pas connecté

‘W’ : inconnu faible: entre la masse et l'alimentation avec des résistances

‘L’ : 0 faible: relié à la masse par une résistance

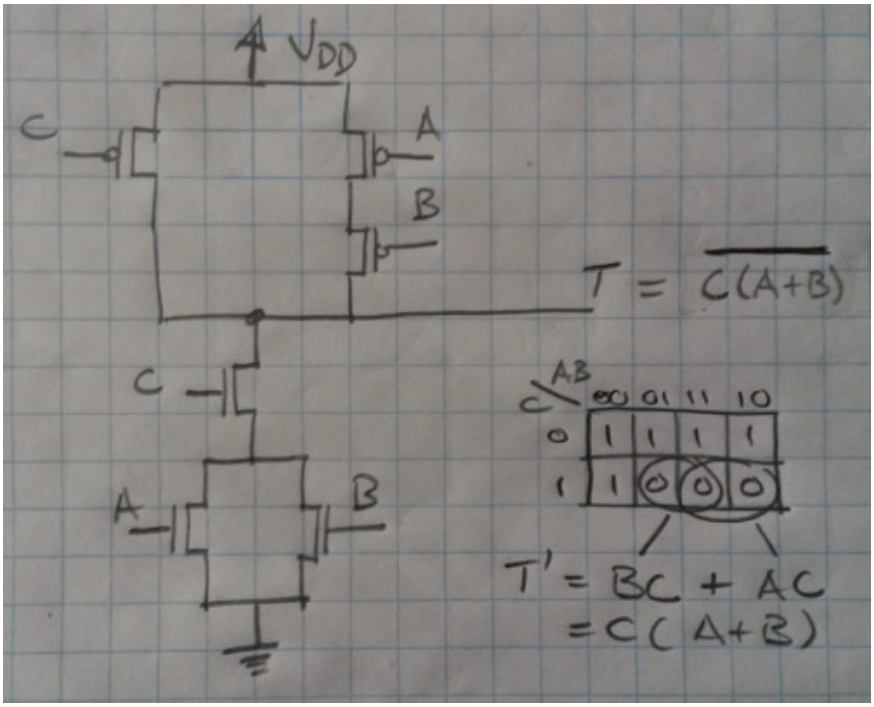
‘H’ : 1 faible: relié à l'alimentation par une résistance

‘-‘ : peu importe (don’t care) - pas de sens physique

c. Avant la synthèse : simulation fonctionnelle seulement; après la synthèse : simulation avec une estimation des délais précise, qui reflète les choix de ressources, leur localisation sur la puce, et les interconnections entre elles.

d. Le fichier *.ucf indiquait à quelles pattes du FPGA relier les ports d’entrée et de sortie du module.

2. On a T = A’B’ + C’ = [(A + B)C]’. Il faut donc 6 transistors. Le réseau NMOS a A et B en parallèle, avec la paire en série avec C. Le réseau PMOS a C en parallèle avec la paire A et B qui est en série.



3. Solution

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

entity combinatoire8 is
   port (
      A : in std_logic_vector(3 downto 0);
      F : out std_logic_vector(2 downto 0)
   );
end combinatoire8;

architecture arch of combinatoire8 is
signal T1, T2, T3 : std_logic;
begin

   T1 <= A(3) and A(2) and A(1) and A(0);
   T2 <= not(A(3)) and not(A(2)) and not(A(0));
   T3 <= A(3) and not(A(2));

   F(2) <= T1 or T3;
   F(1) <= T1 or T2;
   F(0) <= T2 or T3;

end arch;
```

4. Solution

```
architecture arch of compresseurMN is
begin

   assert 2 ** N - 1 >= M report "valeurs incorrectes de M et N" severity failure;

   process(V)
   variable compte : integer range 0 to M := 0;
   begin
      compte := 0;
      for k in 0 to M - 1 loop
         if V(k) = '1' then
            compte := compte + 1;
         end if;
      end loop;
      nombre1 <= to_unsigned(compte, N);
   end process;

end arch;

-- notes:
-- 1. L'énoncé assert devrait être placé à l'extérieur du processus
-- 2. Il faut absolument réinitaliser la variable compte au début de chaque processus;
-- l'initaliser lors de sa déclaration ne suffit pas;
-- les valeurs des variables sont persistantes entre différents appels d'un processus.
-- 3. Il est important de définir une gamme de valeurs pour la variable compte.
-- Le synthétiseur s'en sert pour allouer juste la bonne quantité de bascules
-- pour la contenir.
```