INF3500 : Conception et réalisation de systèmes numériques

Examen intra #1 – 2 octobre 2017

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

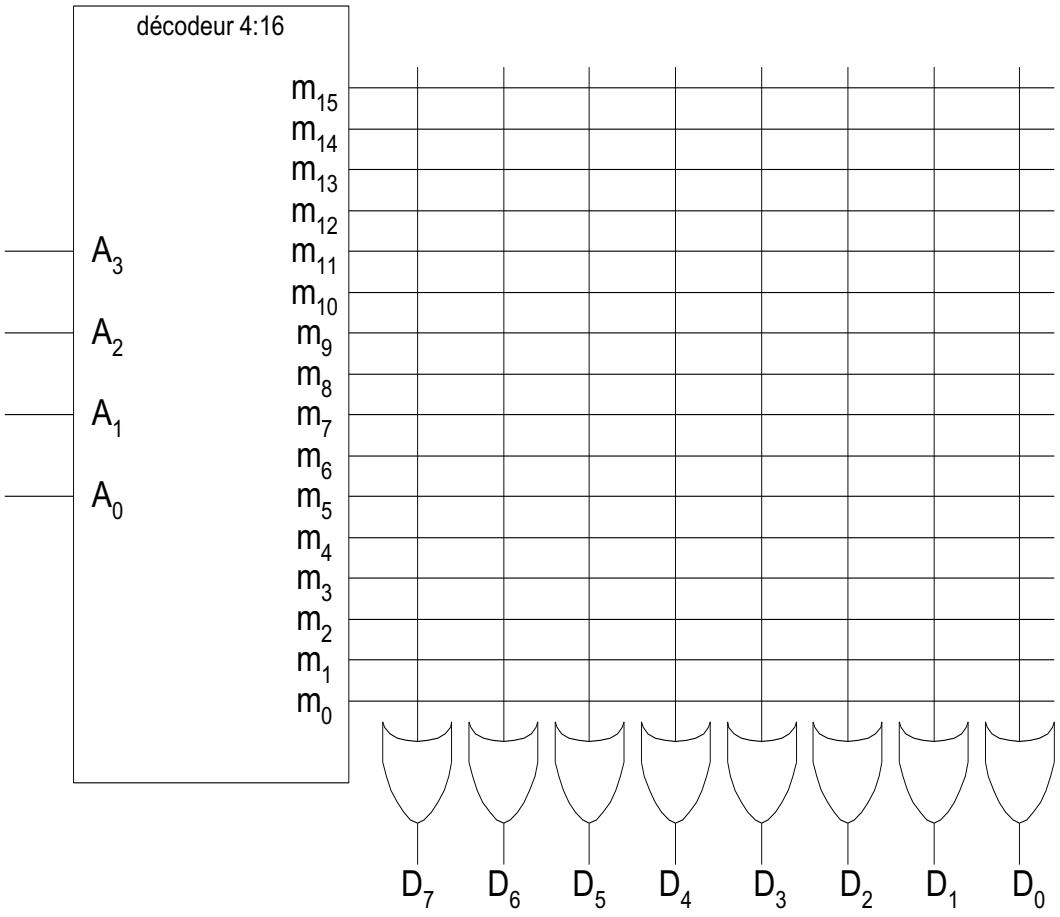
- Ordinateurs interdits. Appareils mobiles interdits.
- Répondre à toutes les questions, la valeur de chaque question est indiquée.
- Répondre sur le questionnaire et le remettre.
- Ne posez pas de questions. En cas de doute sur le sens d'une question, énoncez clairement vos suppositions.

Question 1. (4 points)

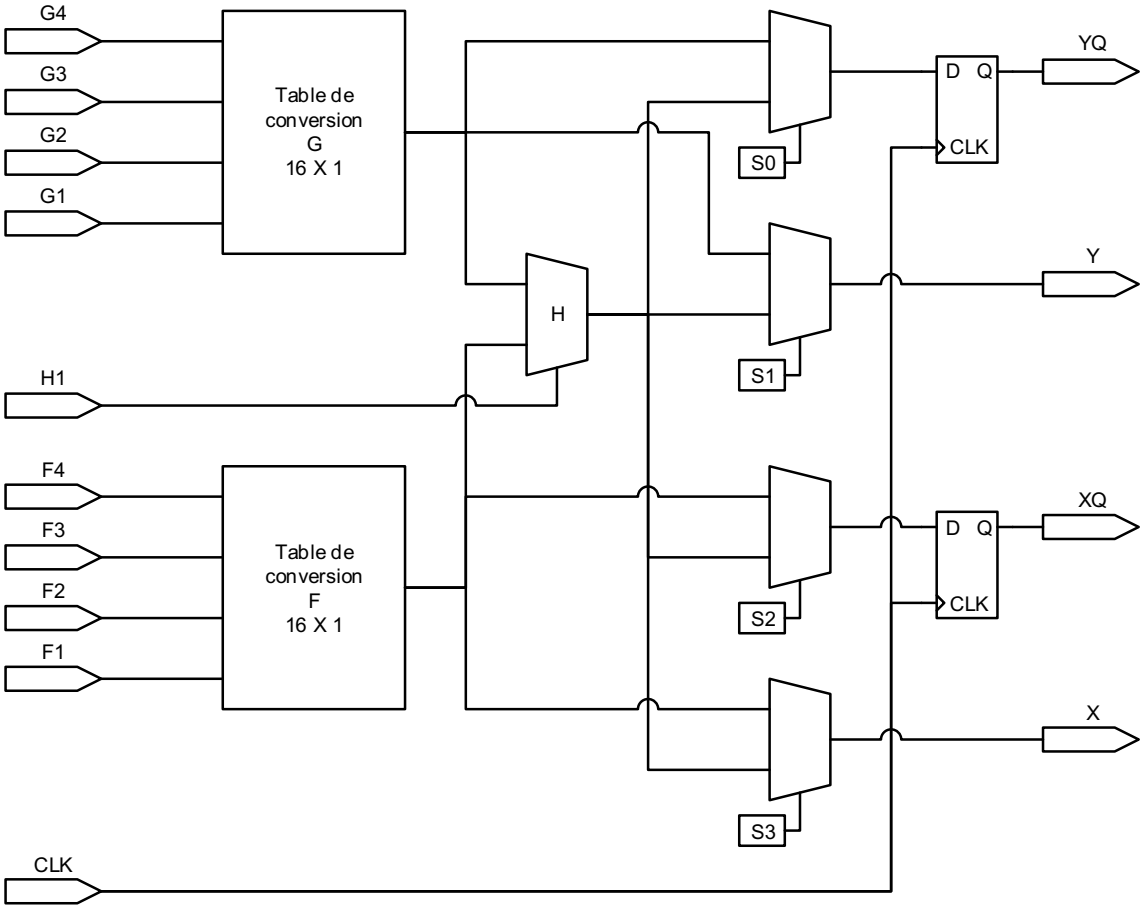
a. (2 points) Considérez le code VHDL suivant. Montrez comment l'implémenter sur la mémoire ROM donnée.

```
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;
```



b. (2 points) Pour le code de la partie a., montrez comment implémenter les sorties F(0) et F(1) sur le modèle de FPGA suivant. Utilisez le moins de ressources possible. Étiquetez clairement les ports d'entrée et de sortie. Indiquer directement sur le dessin où chaque signal se situe ainsi que les interconnections entre les blocs. Indiquer clairement et sans ambiguïté, dans les tables de vérité fournies, le contenu de chacune des tables de conversion (LUT) que vous utilisez.



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	

Question 2. (3 points)

Vous travaillez pour une compagnie qui œuvre dans le domaine des systèmes autonomes interconnectés. Le projet de votre équipe concerne l'implémentation des algorithmes de traitement et décision pour ces systèmes. Vous devez concevoir l'unité qui assurera l'apprentissage automatique profond. Il faut déterminer si cette unité sera implémentée en logique fixe ou sur FPGA. Les métriques à considérer sont : le coût, le temps de développement et la consommation en puissance. Votre client vous commandera 500 ou bien 2000 unités, on ne sait pas encore.

Pour la solution FPGA, le temps de conception est estimé à 6 mois pour 2 ingénieur/es; le temps sera le même pour la logique fixe mais il faudra un/e ingénieur/e de plus. Dans les deux cas, un/e ingénieur/e coûte 100 k$/an. Les licences d'outils de design coûtent 5 k$ par poste pour le FPGA et 20 k$ par poste pour la logique fixe. Pour la logique fixe, les frais fixes de la fonderie s'élèvent à 250 k$, puis les puces coûtent 10 $ l'unité, et les premières puces seraient livrées 4 mois après la commande. Le FPGA qui accommoderait votre design coûte 250 $ et peut être livré dans 4 semaines. Les frais de production et de distribution sont les mêmes pour les deux technologies.

On vous demande de préparer un rapport pour aider l'équipe dans sa prise de décision.

(*Utilisez le verso si nécessaire*)

Question 3. (3 points)

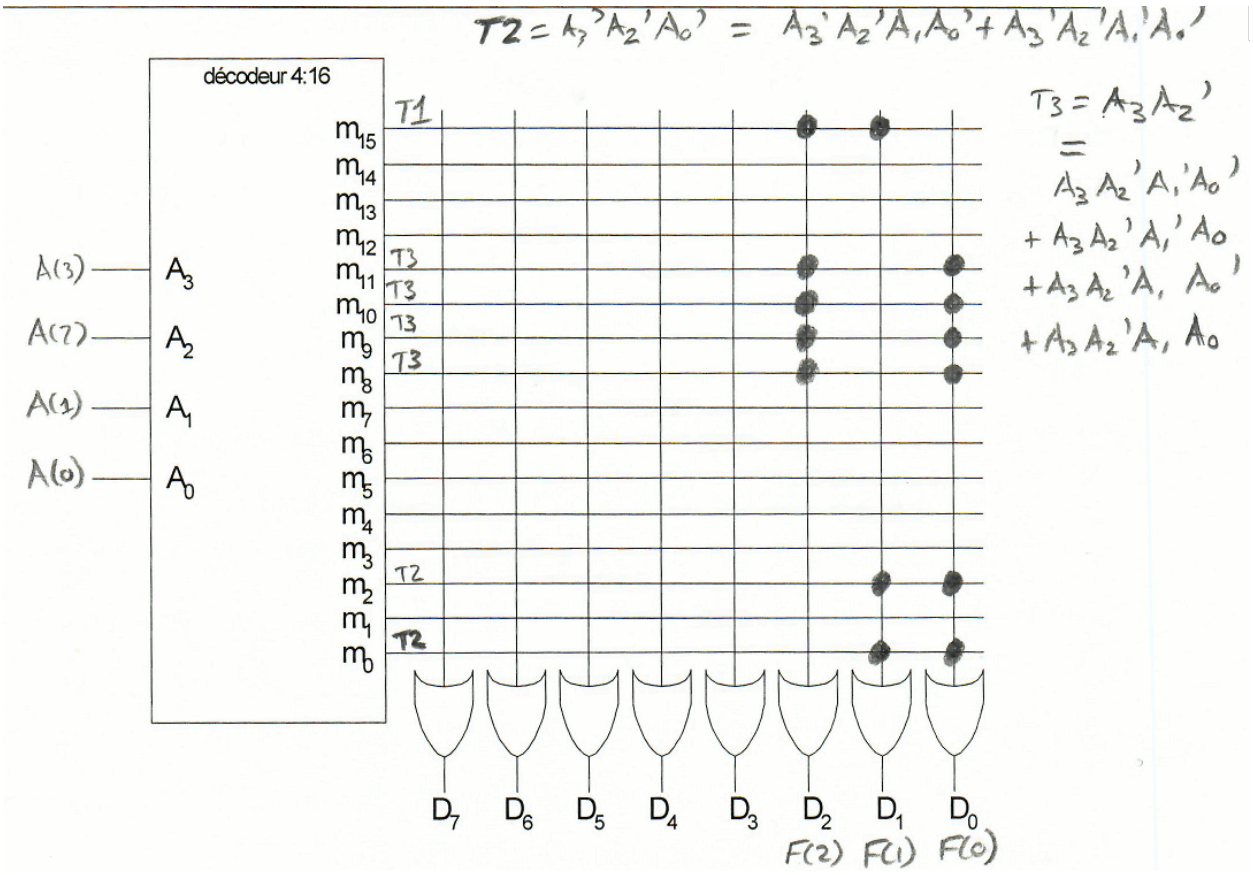
Fony Starc, riche entrepreneur en robotique, possède un vaste manoir situé sur une des îles de la Rivière-des-Milles-Iles. Le manoir est entouré d'une haute clôture surveillée par un réseau de caméras intelligentes, numérotées 0, 1, 2, 3, etc. Les caméras sont capables de détecter automatiquement la présence d'un intrus, jour et nuit. Quand un intrus est détecté, une caméra active un signal d'alerte sur un fil qui lui est dédié. M. Starc a besoin d'une boîte d'interface pour relier toutes ces caméras. La boîte doit activer un signal d'alarme quand au moins une caméra détecte un intrus. Elle doit aussi donner le numéro de la caméra qui détecte un intrus par un nombre binaire. M. Starc, craint pas ses ennemis, s'attend à ce qu'il n'y ait jamais plus d'un intrus à la fois; mais, comme tout bon ingénieur, il veut aussi un signal d'erreur qui s'active si jamais plus d'une caméra rapportait un intrus. Donnez une déclaration d'entité et une architecture en VHDL synthétisable pour la boîte d'interface de M. Starc.

Barème de correction. Pour un maximum de 2 points sur 3, donnez un code valide uniquement pour 4 caméras. Pour un maximum de 3 points sur 3, donnez un code valide pour un nombre arbitraire de caméras.

Utilisez le verso si nécessaire)

Solutions

1.a Réponse



1 b. réponse: F(0) provient de Y, et F(1) provient de X

A3	A2	A1	A0	F(0)
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

A3	A2	A1	A0	F(1)
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

2. Réponse

Analyse des coûts

Quoi	FPGA	Logique fixe
Main d'oeuvre	2 × 6 / 12 × $100 k / an = $100 k	3 × 6 / 12 × $100 k/ an = $150 k
Licenses	2 × $5 k	3 × $20 k
Autres frais	$0	(fonderie) $250 k
Total des frais fixes	$110 k	$460 k
Par puce	$250	$10
Commande #1 : 500 systèmes	$110 k + 500 × $0.25 k = $235 k	$460 k + 500 × $0.01 k = $465 k
Commande #2 : 2000 systèmes	$110 k + 2000 × $0.25 k = $610 k	$460 k + 2000 × $0.01 k = $480 k

Le FPGA est préférable pour la commande de 500 unités, et la logique fixe est préférable pour la commande de 2000 unités.

Le facteur temps peut être important aussi. La logique fixe a un délai de 4 mois. Les risques sont grands. Si on découvre un bogue, il faut attendre à nouveau 4 mois et repayer les frais fixes de fonderie pour faire une nouvelle puce. Avec le FPGA, les puces seraient disponibles dans un mois. Mais si un bogue était découvert, il ne serait pas nécessaire de faire une nouvelle commande, il faudrait seulement reprogrammer la puce.

La consommation d'énergie est plus élevée pour le FPGA que la logique fixe.

3. Solution, il y a plusieurs réponses possibles.

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

entity museeSecurite3 is
   generic (
      N : positive := 6
   );
   port (
      alerte_Camera : in std_logic_vector(N - 1 downto 0) ;
      alarme_Intrus : out std_logic;
      erreur : out std_logic;
      code_Camera : out unsigned(integer(ceil(log2(real(N)))) - 1 downto 0)
   );
end;

architecture arch of museeSecurite3 is
begin
   assert N = 4 report "architecture valable uniquement pour N = 4" severity failure;
   alarme_Intrus <= alerte_Camera(3) or alerte_Camera(2) or alerte_Camera(1) or
alerte_Camera(0);

   with alerte_Camera select
   code_Camera <=
   "00" when "0001",
   "01" when "0010",
   "10" when "0100",
   "11" when "1000",
   "00" when others;

   erreur <= (alerte_Camera(3) and alerte_Camera(2))
          or (alerte_Camera(3) and alerte_Camera(1))
          or (alerte_Camera(3) and alerte_Camera(0))
          or (alerte_Camera(2) and alerte_Camera(1))
          or (alerte_Camera(2) and alerte_Camera(0))
          or (alerte_Camera(1) and alerte_Camera(0));

end arch;

architecture archflexible of museeSecurite3 is
begin
   process(alerte_Camera)
   variable compte : natural range 0 to N;
   begin
      compte := 0; -- les variables sont persistentes,
                -- il faut les réinitialiser à chaque itération
      alarme_Intrus <= '0'; -- il faut une valeur par défaut
      for k in N - 1 downto 0 loop
         if alerte_Camera(k) = '1' then
            code_Camera <= to_unsigned(k, code_Camera'length);
            alarme_Intrus <= '1';
            compte := compte + 1;
         end if;
      end loop;
      if compte >= 2 then
         erreur <= '1';
      else
         erreur <= '0';
      end if;
   end process;

end archflexible;
```