

INF3500 : Conception et réalisation de systèmes numériques

Examen intra #1 – mercredi 3 février 2016

Durée: 1 heure.

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

Pondération: 10%.

Calculatrice: Programmable permise.

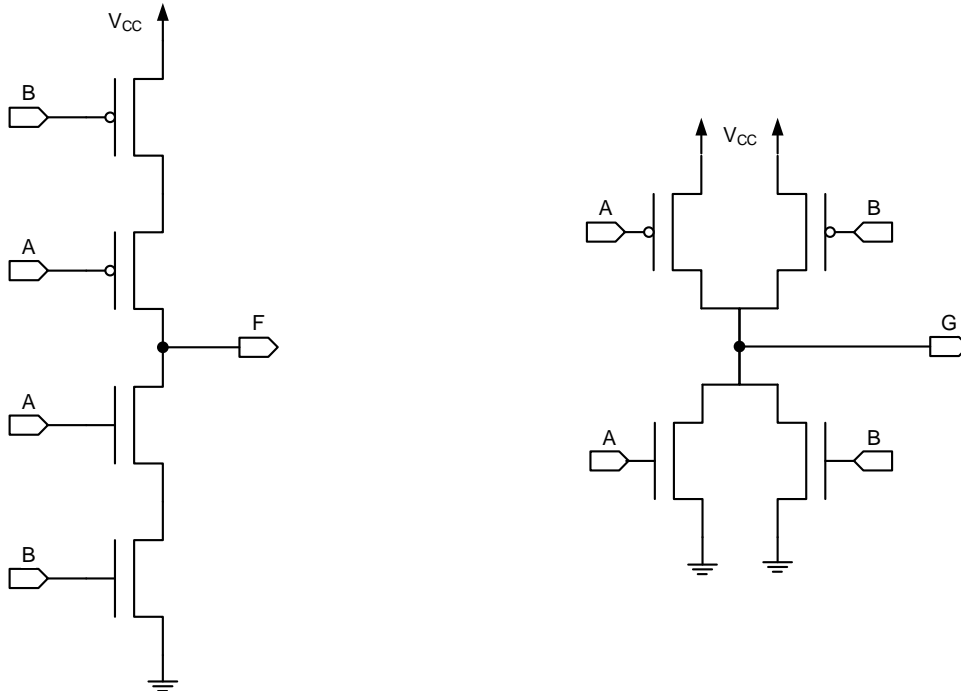
Directives particulières:

- 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. (1 point)

Les circuits suivants CMOS suivants ne respectent pas la règle de complémentarité pour leurs réseaux PMOS et NMOS. L'un deux a quand même un fonctionnement utile, l'autre non.

Quelles sont les valeurs des signaux F et G, selon le type STD_LOGIC de VHDL, quand $A \text{ xor } B = 1$? Justifiez votre réponse.



Réponse pour F :

Réponse pour G :

Question 2. (3 points)

On est en juillet 2016. Vous travaillez chez FPGA Solutions Consultants Inc. Tous les employés, sauf vous et la présidente, sont partis en vacances. La présidente vous parle du nouveau produit d’un client: un système de vision artificielle pour une voiture autonome. Un prototype est fonctionnel, et le client veut maintenant procéder à un premier déploiement dans 500 voitures en pré-production.

Le système doit traiter des données provenant de plusieurs caméras, pour un débit total de 750×10^6 résultats par seconde. On peut l’implémenter avec un processeur unique qui produit un résultat par cycle d’horloge. Ce processeur occupe l’équivalent de 3500 blocs logiques, et sa fréquence maximale dépend de la technologie utilisée. Si la fréquence d’opération est insuffisante pour supporter le débit nécessaire, il est possible d’instancier plusieurs processeurs en parallèle pour multiplier le débit (par exemple, 2 processeurs avec une fréquence de 400 MHz auraient un débit de 800×10^6 résultats/sec et occuperaient 7000 blocs logiques). Cependant, il est essentiel que tout le système puisse être implémenté sur une seule puce.

La présidente vous demande de l’aider à choisir parmi trois technologies dont les données de comparaison sont présentées au tableau suivant. Laquelle des trois technologies privilégiez-vous? Énoncez clairement toutes les suppositions raisonnables que vous faites et montrez tous vos calculs.

Technologie	Frais fixes de développement	Coût par puce	Fréquence d’horloge maximale	Ressources de calcul (blocs logiques)
A. Logique fixe	\$875 000	\$24	2.00 GHz	Max. 25 000
B. FPGA moyenne gamme	\$12 000	\$1 350	0.50 GHz	7 500
C. FPGA bas de gamme	\$12 000	\$49	0.25 GHz	1 750

Utilisez le verso si nécessaire

Question 3. (3 points)

Considérez le module décrit par le code VHDL suivant. Montrez comment implémenter ce module avec le FPGA suivant. Utilisez le moins de ressources possible. Étiquetez clairement les ports d'entrée et de sorties. Indiquer directement sur le dessin où chaque signal se situe ainsi que les interconnexions 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. Utilisez le verso si nécessaire.

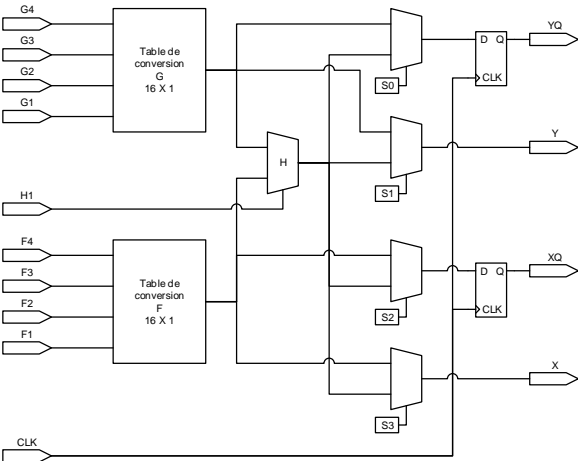
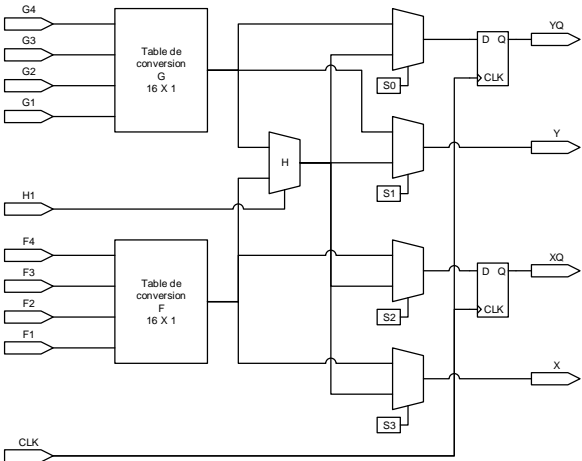
```

library ieee;
use ieee.std_logic_1164.all;

entity combinatoire2 is
  port (
    x1, x2, x3 : in std_logic;
    y1, y2 : out std_logic
  );
end combinatoire2;
    
```

```

architecture arch of combinatoire2 is
  signal p3, p4 : std_logic;
begin
  y2 <= p3 or p4;
  p3 <= not x1 and not x2;
  p4 <= x1 and not x2 and not x3;
  y1 <= x1 xor x3;
end arch;
    
```



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	


Solutions


1. Réponses


La fonction F est en haute impédance quand $A \text{ xor } B = 1$ ou 'Z' en VHDL.

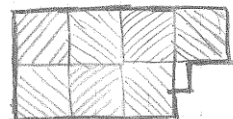
La fonction G est un court-circuit quand $A \text{ xor } B = 1$, ou 'X' en VHDL.

2. Réponses

 1 processeur
3500 blocs logique


 FPGA bas de gamme
1750 blocs logiques = 1/2 processeur $f_{\max} = 250 \text{ MHz}$


 FPGA moyenne gamme, $f_{\max} = 500 \text{ MHz}$
7500 blocs logiques $\approx 2,1$ processeurs


 Logique fixe
25000 blocs logiques $\approx 7,1$ processeurs
 $f = 2,0 \text{ GHz}$

A. En logique fixe, 1 processeur @ 2.0 GHz \rightarrow 2 G-résultats/seconde $> 750 \text{ M rés/sec}$ ok
1 processeur = 3500 blocs logiques < 25000 ok

B. FPGA moyenne gamme

 3500 blocs @ 500 MHz = 500 M rés/s

 3500 blocs @ 500 MHz = $\frac{500 \text{ M rés/s}}{1000 \times 10^6 \text{ rés/s}} > 750 \times 10^6 \text{ rés/sec}$ ok

 7000 blocs < 7500 blocs sur FPGA \rightarrow 1 seule puce ok

C. FPGA bas de gamme
1 processeur (3500 blocs) n'entre même pas dans le FPGA (1750 blocs)
(contrainte: une seule puce)

A. Logique Fixe :

Débit : un seul processeur 750 MHz < 2 GHz ok

Espace : 3500 blocs < 25000 ok

Coût : $\$875 \text{ K} + 500 \text{ unités} \times \$24 = \$887 \text{ K}$.

B. FPGA de moyenne gamme :

Débit : deux processeurs nécessaires @ 500 MHz chacun, total 1000 MHz > 750 MHz requis ok

Espace : deux processeurs, total 7000 blocs logiques, donc une seule puce de 7500 blocs logiques ok

Coût : $\$12 \text{ K} + 500 \text{ unités} \times \$1350 = \$687 \text{ K}$

C. FPGA bas de gamme :

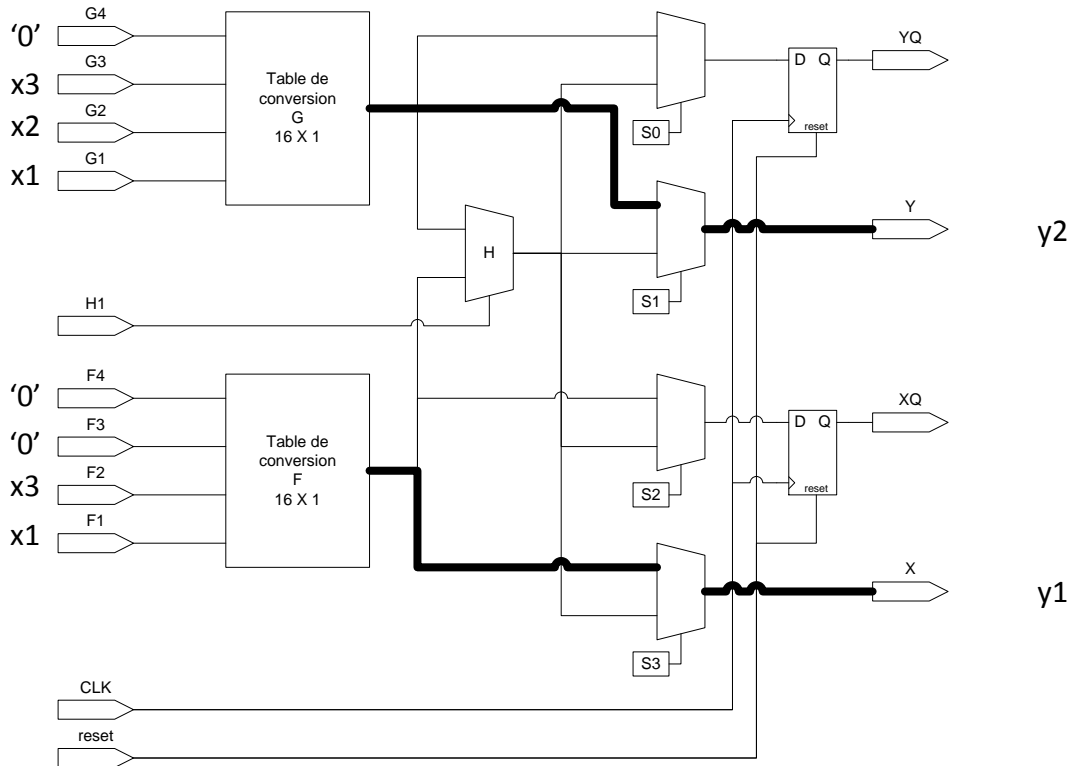
Débit : trois processeurs nécessaires @ 250 MHz chacun, total 750 MHz = 750 MHz requis ok

Espace : trois processeurs, total 10,5 K blocs logiques, donc 6 puces. Cette solution n'est pas acceptable.

Pour ces conditions, on privilégierait donc la solution B, le FPGA de moyenne gamme.

3. Solution

Il y a plusieurs réponses possibles.



G4 (0)	G3 (x3)	G2 (x2)	G1 (x1)	G (y2)
0	0	0	0	1
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
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	0

F4 (‘0’)	F3 (‘0’)	F2 (x3)	F1 (x1)	F (y1)
0	0	0	0	0
0	0	0	1	1
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	0

4. Solution : 2 réponses parmi la multitude de possibilités

```

architecture arch of museeSecurite is
begin

    alarme <=
        (senseursA(2) and senseursA(1))
        or (senseursA(1) and senseursA(0))
        or (senseursA(2) and senseursA(0))

        or (senseursB(3) and senseursB(2))
        or (senseursB(3) and senseursB(1))
        or (senseursB(3) and senseursB(0))
        or (senseursB(2) and senseursB(1))
        or (senseursB(2) and senseursB(0))
        or (senseursB(1) and senseursB(0));

end arch;

architecture archflexible of museeSecurite is
signal alarmeA, alarmeB : std_logic;
begin

    alarme <= alarmeA or alarmeB;

    process(senseursA)
    variable compte : natural range 0 to senseursA'length;
    begin
        compte := 0;
        for k in senseursA'range loop
            if senseursA(k) = '1' then
                report integer'image(k);
                compte := compte + 1;
            end if;
        end loop;
        if compte >= 2 then
            alarmeA <= '1';
        else
            alarmeA <= '0';
        end if;
    end process;

    process(senseursB)
    variable compte : natural range 0 to senseursB'length;
    begin
        compte := 0;
        for k in senseursB'range loop
            if senseursB(k) = '1' then
                compte := compte + 1;
            end if;
        end loop;
        if compte >= 2 then
            alarmeB <= '1';
        else
            alarmeB <= '0';
        end if;
    end process;

end archflexible;

```