

INF3500 : Conception et réalisation de systèmes numériques

Examen intra #1 – 27 septembre 2018

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.

Q1	
Q2	
Q3	
Q4	
Total	

Question 1. (1 point)

Un additionneur de deux bits est un circuit qui accepte en entrée deux bits X et Y, qui les additionne et qui exprime la somme sous la forme d'une retenue R et d'une somme S. Sa table de vérité est montrée ici.

Donnez le diagramme d'un circuit CMOS qui implémente un additionneur de deux bits. Identifiez clairement toutes les entrées et sorties. Utilisez le moins de transistors possible.

X	Y	R	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Question 2. (3 points)

Considérez le code VHDL suivant, qui modélise une unité logique à deux bits et un multiplexeur 2:1. Montrez comment l'implémenter sur le circuit PLA donné. Respectez les entrées et sorties déjà identifiées. Indiquez par un boulet rempli (•) les connexions à faire dans la matrice d'interconnexions.

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;

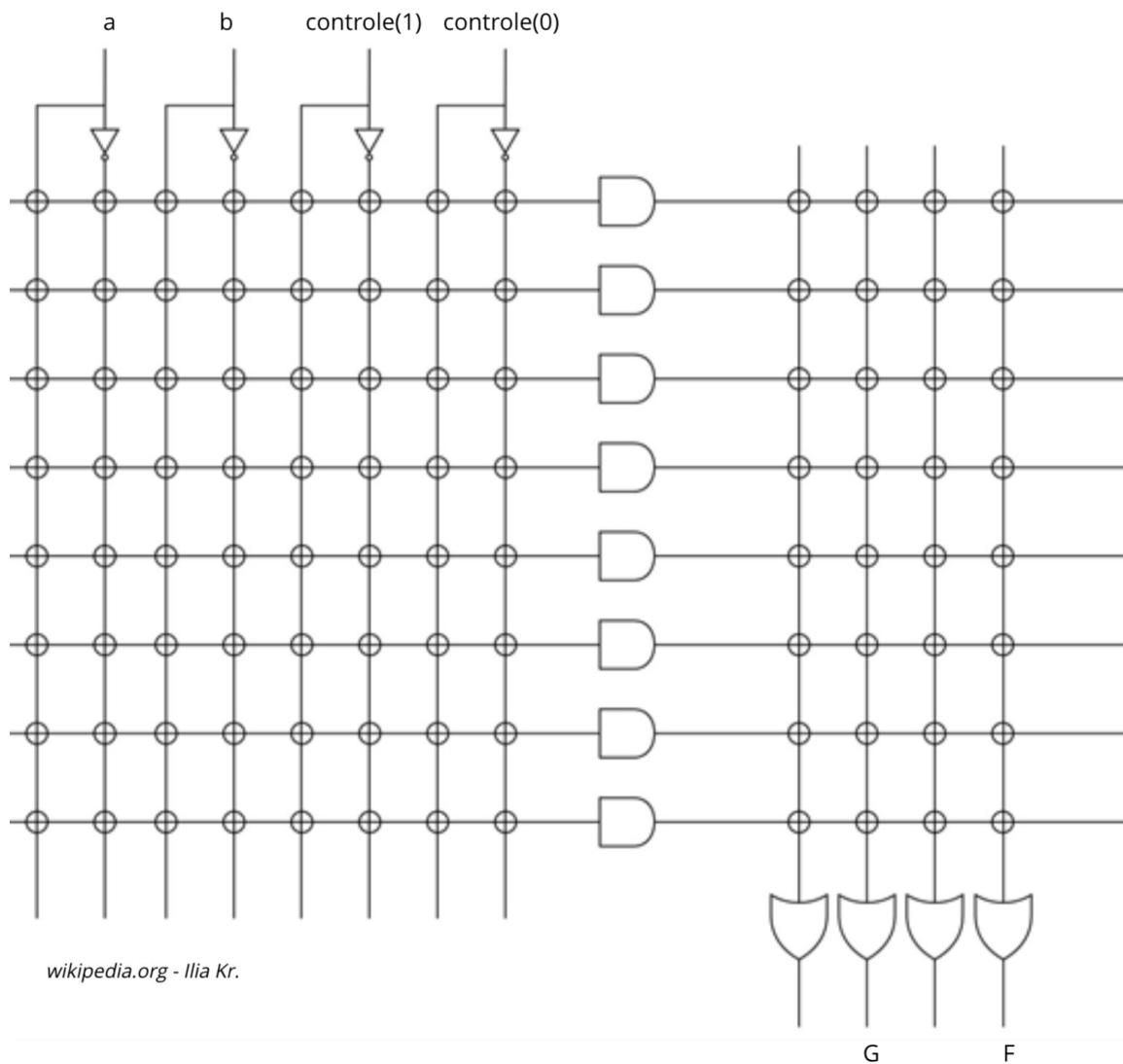
entity module201803C1Q2 is
  port (
    a, b : in std_logic;
    controle : in unsigned(1 downto 0);
    F, G : out std_logic
  );
end module201803C1Q2;

architecture arch of module201803C1Q2 is
begin
  assert a = b report "a et b sont différents"
  severity warning;

```

```
-- unité logique
with to_integer(controle) select
F <=
  a and b when 0,
  a or b when 1,
  a xor b when 2,
  not(a) when others;

-- multiplexeur 2:1
process(a, b, controle) is
begin
  if controle(0) = '1' then
    G <= a;
  else
    G <= b;
  end if;
end process;
end arch;
```



Question 4. (3 points)

Donnez le code VHDL synthétisable d'un circuit combinatoire qui indique l'indice du '1' le plus significatif dans un vecteur V de 6 bits. Par exemple, si V = "000010" alors indice = "001"; si V = "000111" alors indice = "010"; si V = "010001" alors indice = "100"; etc. Si V ne comporte aucun '1' (donc si V = "0000000") alors la valeur de la sortie indice est sans importance mais un signal erreur doit prendre la valeur '1'. Si V comporte au moins un '1', alors erreur doit prendre la valeur '0'. Donnez une architecture pour l'entité suivante. Donnez du code de la meilleure qualité possible.

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;

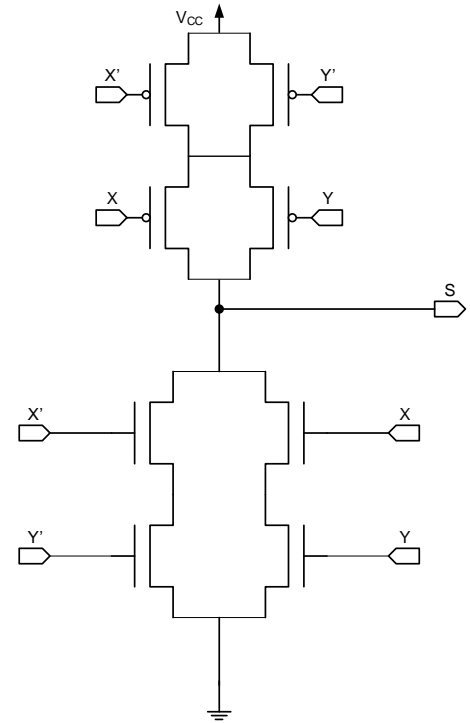
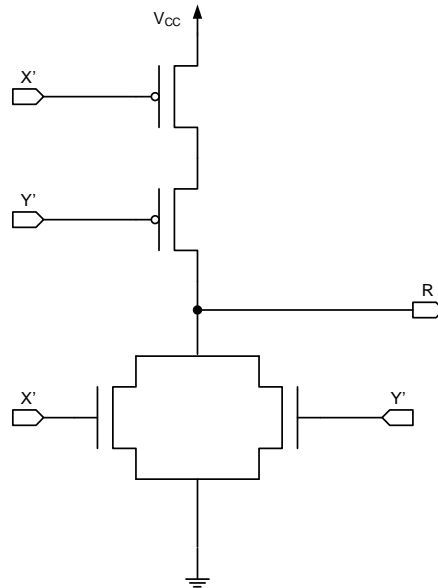
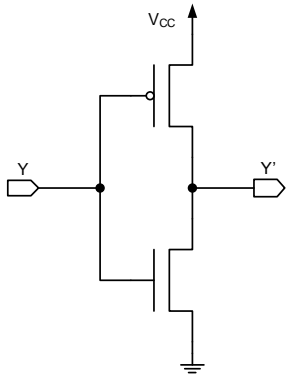
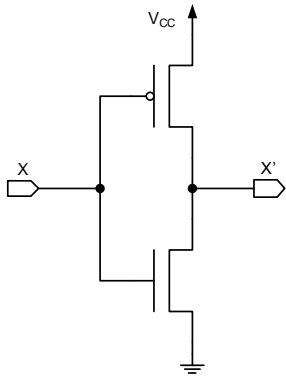
entity module201803C1Q4 is
    port (
        N : in unsigned(5 downto 0);
        indice : out unsigned(2 downto 0);
        erreur : out std_logic
    );
end module201803C1Q4;
```

Utilisez le verso si nécessaire)

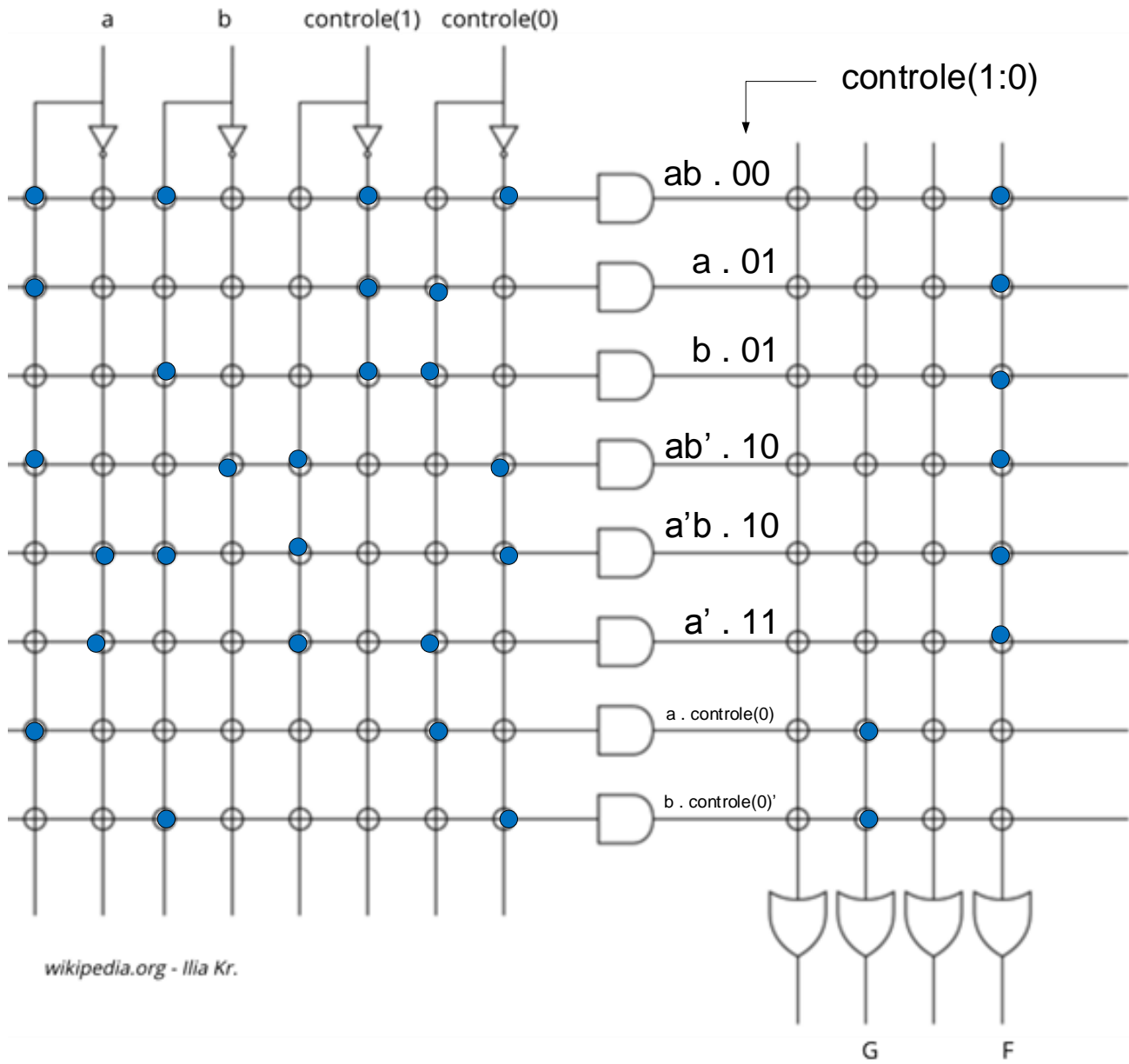
Solutions

#1 circuit CMOS

On a $R = XY = (X' + Y)'$ et $S = X \text{ xor } Y = (X'Y' + XY)'$



#2 PLA



wikipedia.org - Ilija Kr.

#3 Analyse des coûts

On prévoit $(1000 + 800) \times 3 = 5400$ installations.

	FPGA	Logique fixe
Ingénieurs	$3 \times 6 / 12 \times 100 \text{ k\$}$ = 150 k\$	$150 \text{ k\$} \times 1.5$ = 225 k\$
Licences	$3 \times 6 / 12 \times 5 \text{ k\$}$ = 7.5 k\$	$7.5 \text{ k\$} \times 4 \times 1.5$ = 45 k\$
Fonderie	-	400 k\$ (includ 1000 puces)
Total frais fixes	~157.5 k\$	~670 k\$
Total des frais unitaires	75 \$	1000 puces incluses 1 \$ ensuite
Total des frais pour 5400 installations	$157500 + 75 \times 5400$ = 562500 \$	$670000 + 1 \times (5400 - 1000)$ = 674400 \$

Même avec une commande importante de plus de 5000 unités, les frais fixes énoncés font que la solution FPGA reste la plus économique. Il faudrait aussi tenir compte des délais de livraisons, dont les impacts financiers ne sont pas chiffrés dans la question. Il faudrait aussi tenir compte du risque associé à la logique fixe en cas de bogue.

#4. Solution, il y a plusieurs réponses possibles.

```
architecture arch of module201803C1Q4 is
begin
    process(N) is
    begin
        if N(5) = '1' then
            erreur <= '0';
            indice <= to_unsigned(5, indice'length);
        elsif N(4) = '1' then
            erreur <= '0';
            indice <= to_unsigned(4, indice'length);
        elsif N(3) = '1' then
            erreur <= '0';
            indice <= to_unsigned(3, indice'length);
        elsif N(2) = '1' then
            erreur <= '0';
            indice <= to_unsigned(2, indice'length);
        elsif N(1) = '1' then
            erreur <= '0';
            indice <= to_unsigned(1, indice'length);
        elsif N(0) = '1' then
            erreur <= '0';
            indice <= to_unsigned(0, indice'length);
        else
            erreur <= '1';
            indice <= (others => 'X');
        end if;
    end process;
end arch;
```