INF3500 : Conception et réalisation de systèmes numériques

Examen intra #1 – 26 septembre 2019

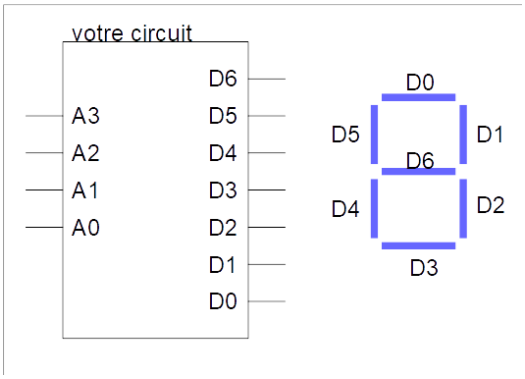
Q1	
Q2	
Q3	
Q4	
Total	

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

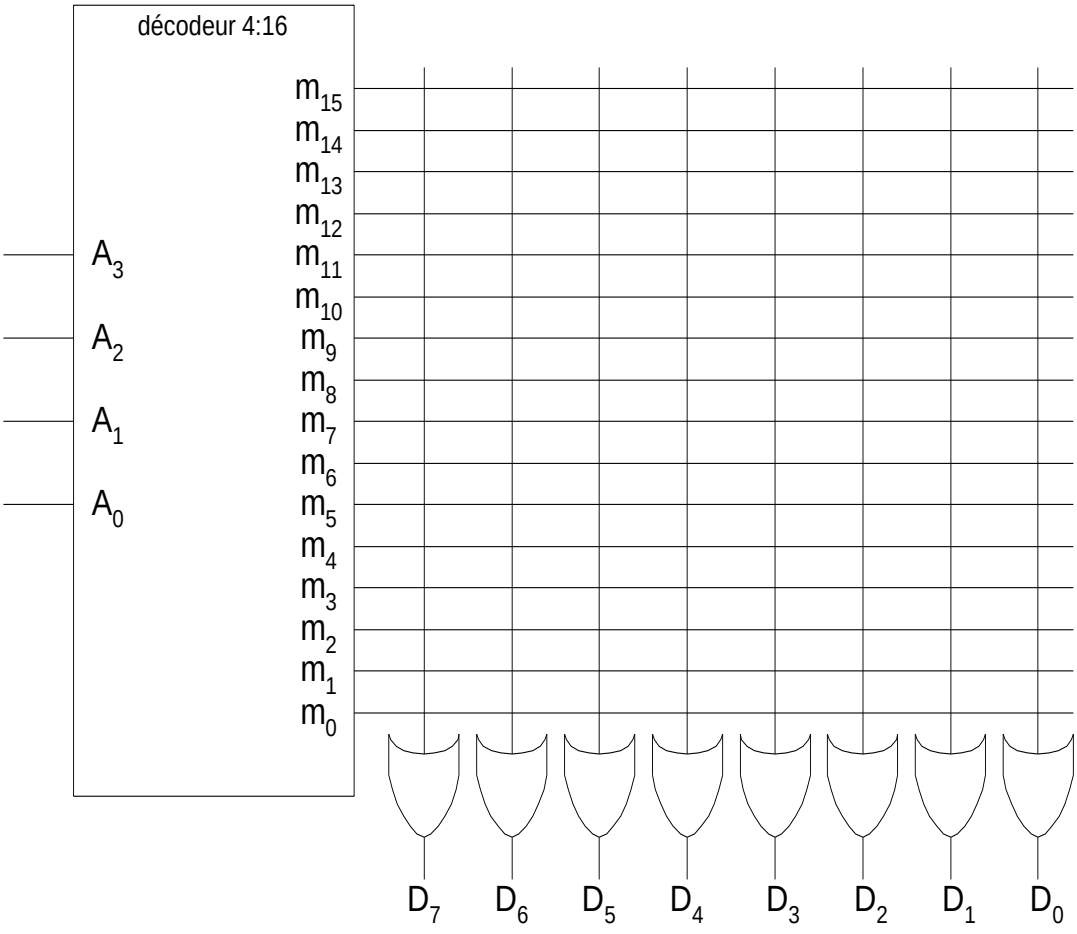
- Ordinateurs interdits. Appareils mobiles interdits.
- Répondre à toutes les questions, la valeur de chaque question est indiquée.
- Répondre sur le questionnaire et le remettre.
- Ne posez pas de questions. En cas de doute sur le sens d’une question, énoncez clairement vos suppositions.

Question 1. (2 points)

Faites la conception d’un circuit combinatoire qui prend en entrée un nombre binaire A exprimé sur 4 bits $A_3A_2A_1A_0$ et qui mène un afficheur à 7 segments D6, D5, … D0 pour afficher la représentation du nombre dans l’intervalle [0, 9]. Pour les autres valeurs du nombre, l’afficheur doit être éteint. Chaque segment Dk s’allume quand il reçoit un ‘1’ logique. On veut avoir les représentations suivantes pour les 10 chiffres:



Implémentez votre circuit sur la ROM donnée ici. Respectez les entrées et sorties indiquées.



Question 2. (2 points)

Considérez le module VHDL suivant qui décrit un multiplexeur 2:1.

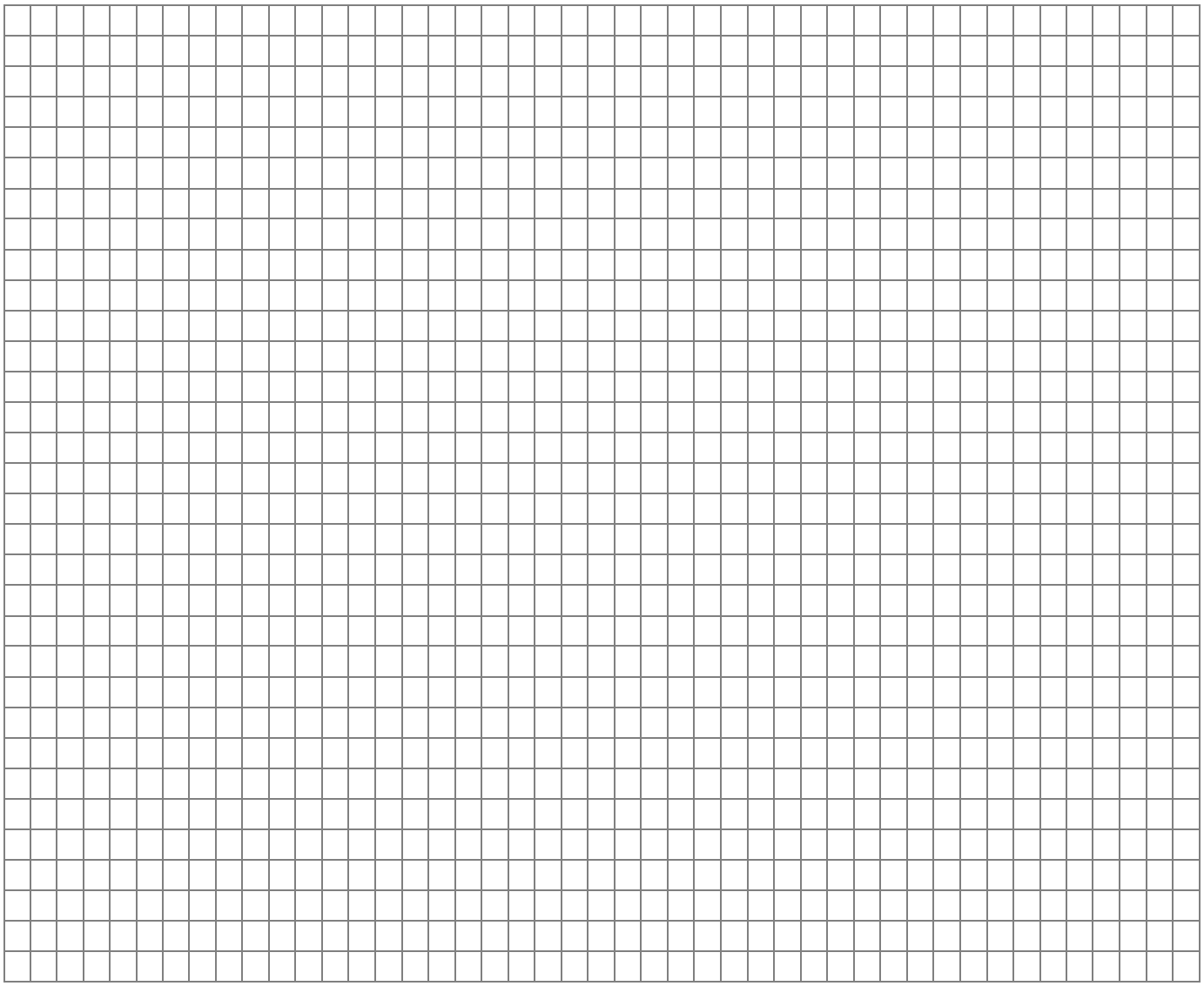
a. Complétez la table de vérité donnée.

b. Implémentez le module avec un circuit CMOS en utilisant le moins de transistors possible. Les entrées ne sont disponibles qu'en forme non complémentée.

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;

entity mux21 is
   port(
      a, b, c: in std_logic;
      F : out std_logic
   );
end mux21;
```

```
architecture flotDeDonnees of mux21 is

begin
   process(a, b, c) is
   begin
      if c = '0' then
         F <= a;
      else
         F <= b;
      end if;
   end process;

end flotDeDonnees;
```

a	b	c	F
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	



Question 3. (3 points)

Alors que vous débutez votre stage à l'été 2020, votre collègue vous propose plutôt de l'aider à construire un Processeur pour Miner des Polycoins (PMP), une nouvelle cryptomonnaie. Elle a trouvé une firme d'investissement qui n'attend qu'à être convaincue du rendement potentiel de votre PMP.

Le rôle du PMP est de résoudre des problèmes cryptographiques 24 h sur 24 afin d'obtenir des Polycoins. Pour une solution FPGA, vous allez relier ensemble 30 cartes de développement, chacune contenant un PMP opérant à une fréquence d'horloge de 100 MHz. À cette fréquence, vous estimez qu'un PMP génère des revenus de 235 $ par mois en Polycoins. Chaque carte coûterait 375 $ à l'achat et consommerait 17 $ d'électricité par mois. Pour la solution en logique fixe, vous pourriez intégrer 30 MP sur une seule puce avec une fréquence d'horloge de 1.5 GHz. La puce au complet ne coûterait que 65 $ par mois d'électricité à opérer, mais les frais de la fonderie seraient de 215 k$ pour l'obtenir et la monter sur une carte. Le temps de conception serait de trois mois pour le FPGA et de 6 mois pour la logique fixe. De plus, pour la logique fixe, vous auriez besoin d'une 3^{e} personne pour aider à la conception. Chaque personne de l'équipe aurait un salaire de 4 k$ par mois.

Le système aura une durée utile finie : la nature des calculs requis pour les Polycoins sera changée le 1er septembre 2021. Pour le FPGA, votre système serait en fonction et commencerait à générer des revenus dès le 1er septembre 2020. À cause des délais de livraison, la solution en logique fixe ne pourrait fonctionner qu'à partir du 1er juin 2021. Que conseillez-vous à la firme d'investissement ? Justifiez complètement votre réponse. Énoncez clairement toutes vos suppositions et montrez tous vos calculs.

Analyse des coûts

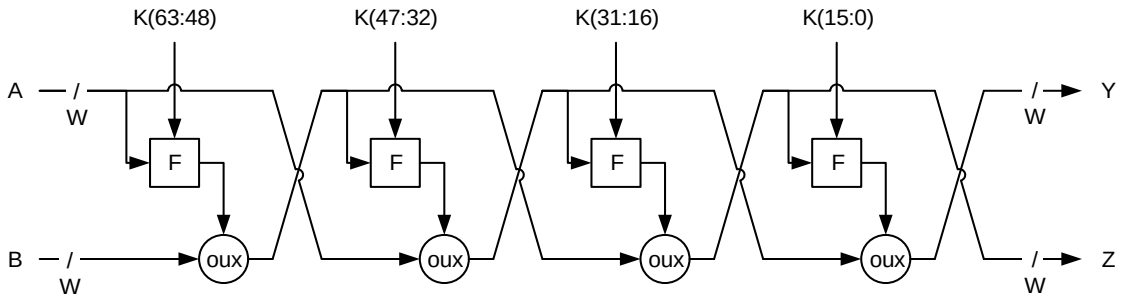
Critère	FPGA	Logique fixe

Argumentation

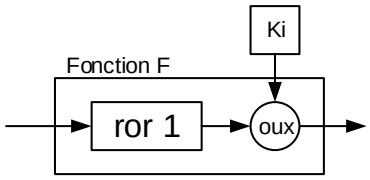
(Utilisez le verso si nécessaire)

Question 4. (3 points)

Le diagramme suivant illustre un réseau de Feistel à quatre étages. Le réseau de Feistel est utilisé par plusieurs algorithmes cryptographiques, dont DES, Blowfish et RC5, pour lesquels il diffère principalement dans le nombre d'étages et dans la nature de la fonction F. Le message à chiffrer est décomposé en un flux de nombres non signés appliqués aux entrées A et B du réseau, et les sorties Y et Z sont le message chiffré. À chaque étage, la fonction F applique une opération au signal du haut avec une portion de la clé secrète K. On effectue ensuite un ou-exclusif bit à bit avec le signal du bas. À la fin de l'étage, les signaux du haut et du bas sont interchangés.



Pour cette question, supposez que la fonction F consiste à appliquer une rotation de l'entrée de 1 bit vers la droite (`ror 1` en VHDL) puis à effectuer l'opération ou-exclusif bit à bit avec la clé K_i exprimée sur 16 bits.



Considérez la déclaration d'entité suivante pour le diagramme du réseau de Feistel. Donnez une architecture en VHDL synthétisable correspondant à cette entité et au diagramme.

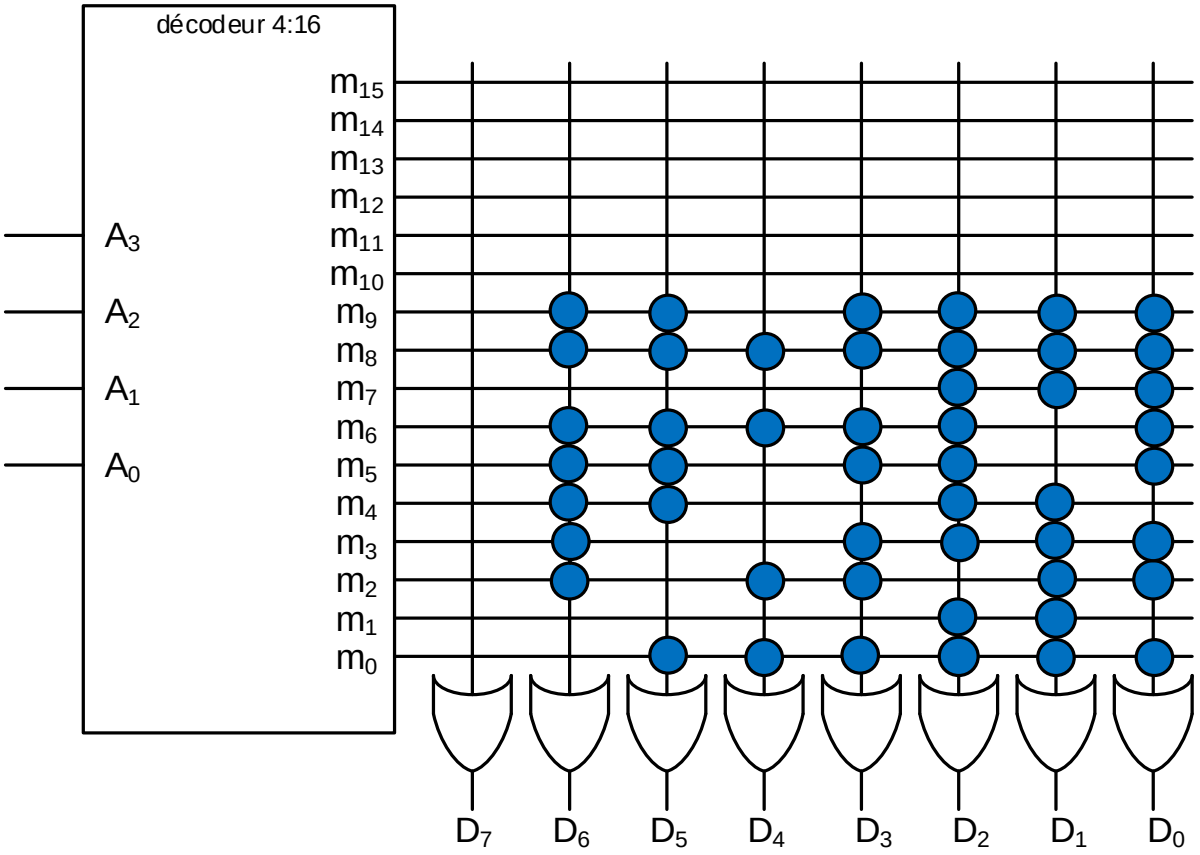
Pour un maximum de 2 points, donnez une architecture qui fonctionne pour le cas $N = 4$ étages. Pour un maximum de 3 points, donnez une architecture qui fonctionne pour un nombre arbitraire d'étages $N > 4$.

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

entity feistelcomb is
    generic (
        W : positive := 16; -- largeur des mots
        N : positive := 4 -- nombre d'étages
    );
    port (
        A, B : in unsigned(W - 1 downto 0);
        K : in unsigned(N * W - 1 downto 0);
        Y, Z : out unsigned(W - 1 downto 0)
    );
end feistelcomb;
```

Utilisez le verso si nécessaire)

Solutions

#1 PROM



#2 Circuit CMOS

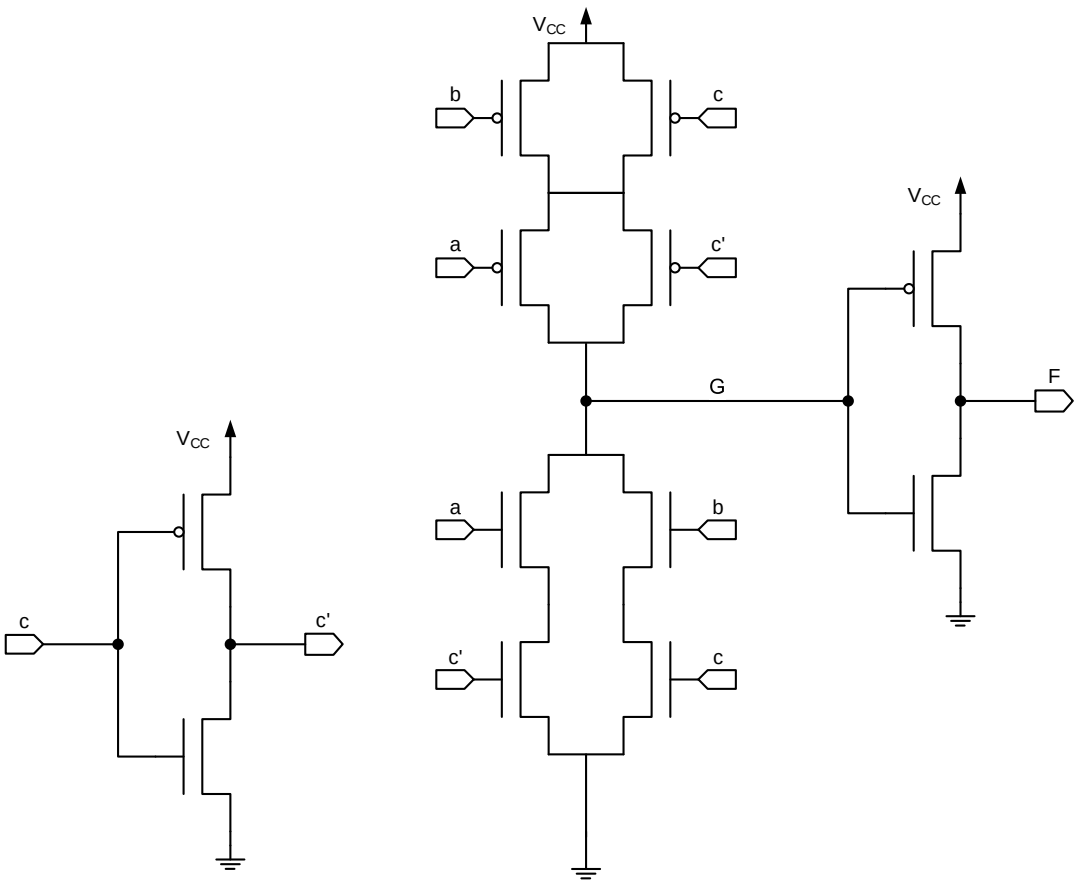
Table de vérité :

a	b	c	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

Il y a plusieurs solutions.

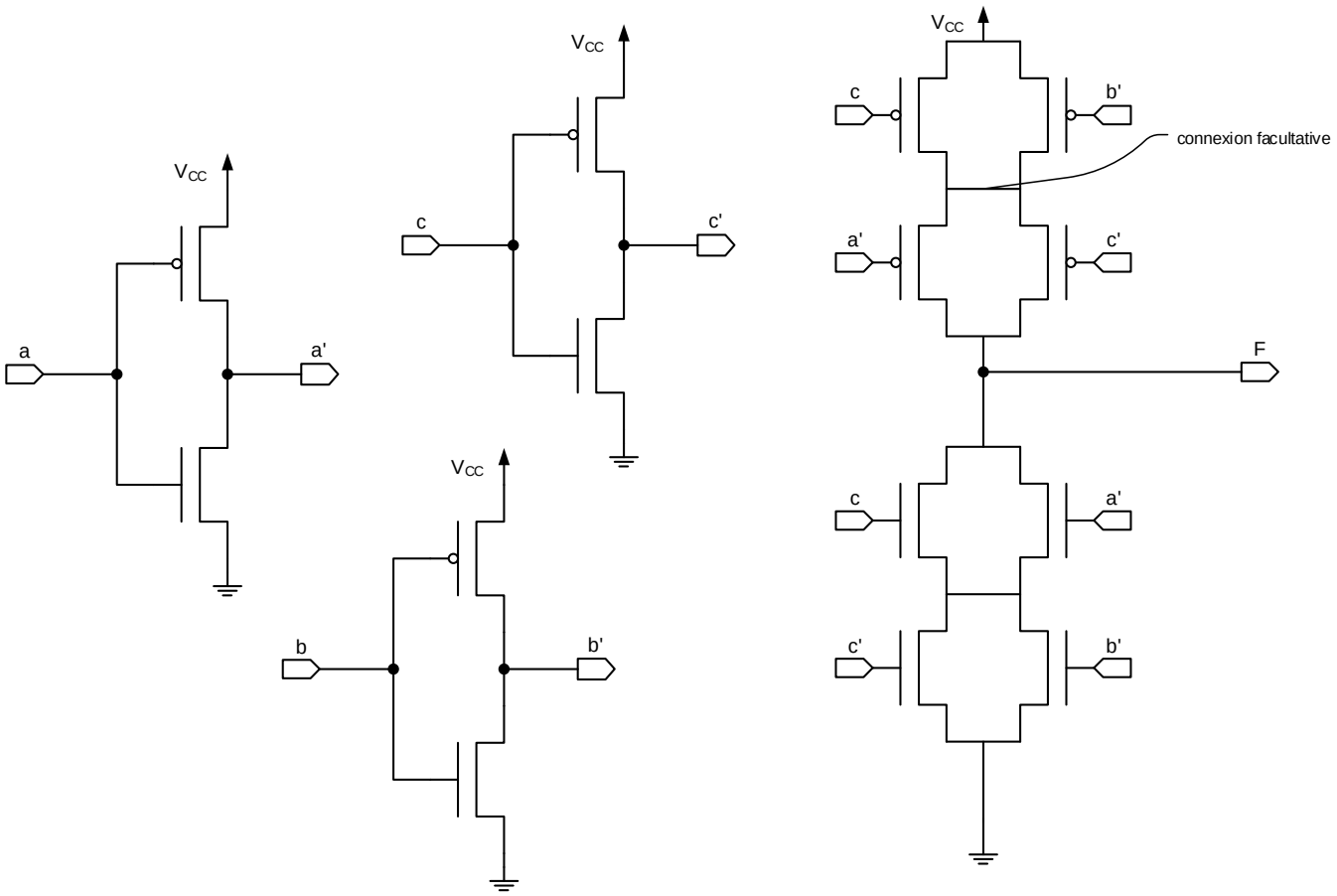
<u>Solution 1 avec 12 transistors</u>

Équation : F = ac' + bc
On construit le circuit CMOS pour G = F', puis on inverse cette fonction. L'avantage c'est qu'on n'a besoin que de deux inverseurs : un pour l'entrée c et un pour la sortie.



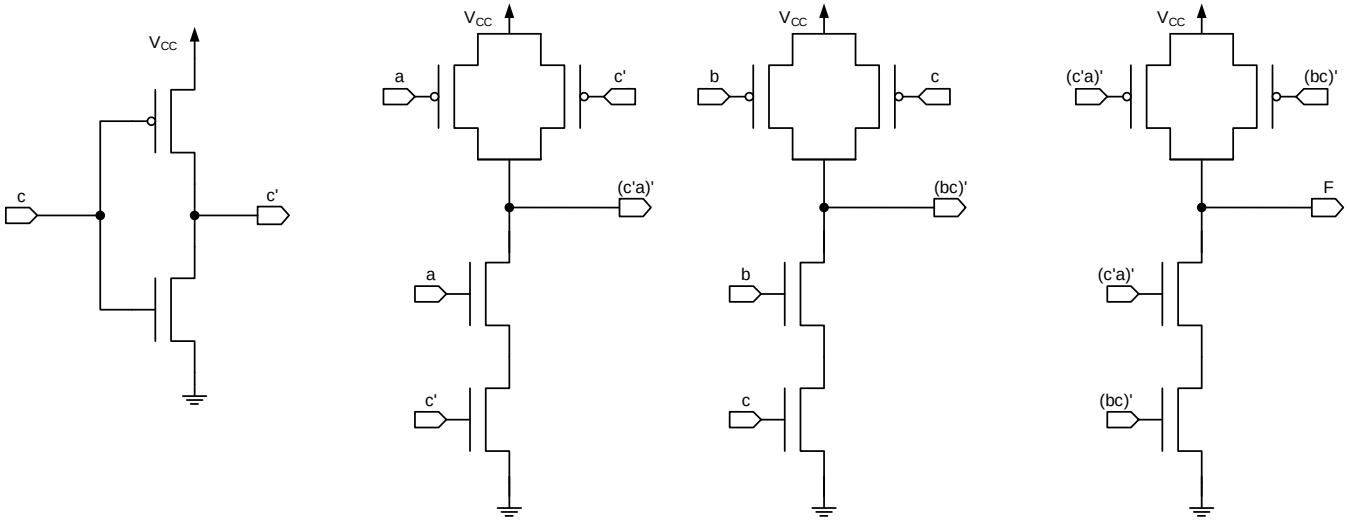
Solution 2 avec 14 transistors

Équation : F = c'a + cb; F' = (c + a') • (c' + b')



Solution 3 avec 14 transistors

Équation F = c'a + cb = [c'a + cb]" = [(c'a)' • (cb)']'



#3 Analyse des coûts

Conception	FPGA	fixe
Npersonnes	2	3
Mois	3	6
Taux	4000	4000
freq horloge (MHz)	100	1500
	FPGA	fixe
salaires	24000	72000
matériel	11250	215000
électricité/mois/puce	17	65
électricité	6120	195
Coût total	41370	287195
nPMP	30	30
revenus/mois/100MHz/PMP	235	235
nMois	12	3
Revenus	84600	317250
profits (revenus – coût total)	43230	30055

En conclusion, les deux solutions sont semblables mais le FPGA générerait des profits de 13 k$ plus élevés pour cette période. En plus, le risque associé à la logique fixe n'en vaudrait pas la peine.

#4. Code VHDL, il y a plusieurs réponses possibles.

```
architecture arch1 of feistelcomb is

signal A0, A1, A2, A3, A4, B0, B1, B2, B3, B4 : unsigned(W - 1 downto 0);

begin

    assert N = 4 report "cette architecture ne fonctionne que pour N = 4" severity failure;

    A0 <= A;
    B0 <= B;

    B1 <= A0;
    A1 <= (A0 ror 1) xor K(63 downto 48) xor B0;

    B2 <= A1;
    A2 <= (A1 ror 1) xor K(47 downto 32) xor B1;

    B3 <= A2;
    A3 <= (A2 ror 1) xor K(31 downto 16) xor B2;

    B4 <= A3;
    A4 <= (A3 ror 1) xor K(15 downto 0) xor B3;

    Y <= A4;
    Z <= B4;

end arch1;


architecture arch2 of feistelcomb is

begin

    assert N > 3 report "cette architecture ne fonctionne que pour N > 3" severity failure;

    process (A, B, K)

    variable haut, bas, temporaire : unsigned(W - 1 downto 0);

    begin

        haut := A;
        bas := B;

        for etage in N - 1 downto 0 loop
            temporaire := haut;
            haut := (haut ror 1) xor K(W * (etage + 1) - 1 downto W * etage) xor bas;
            bas := temporaire;
        end loop;

        Y <= haut;
        Z <= bas;

    end process;

end arch2;
```