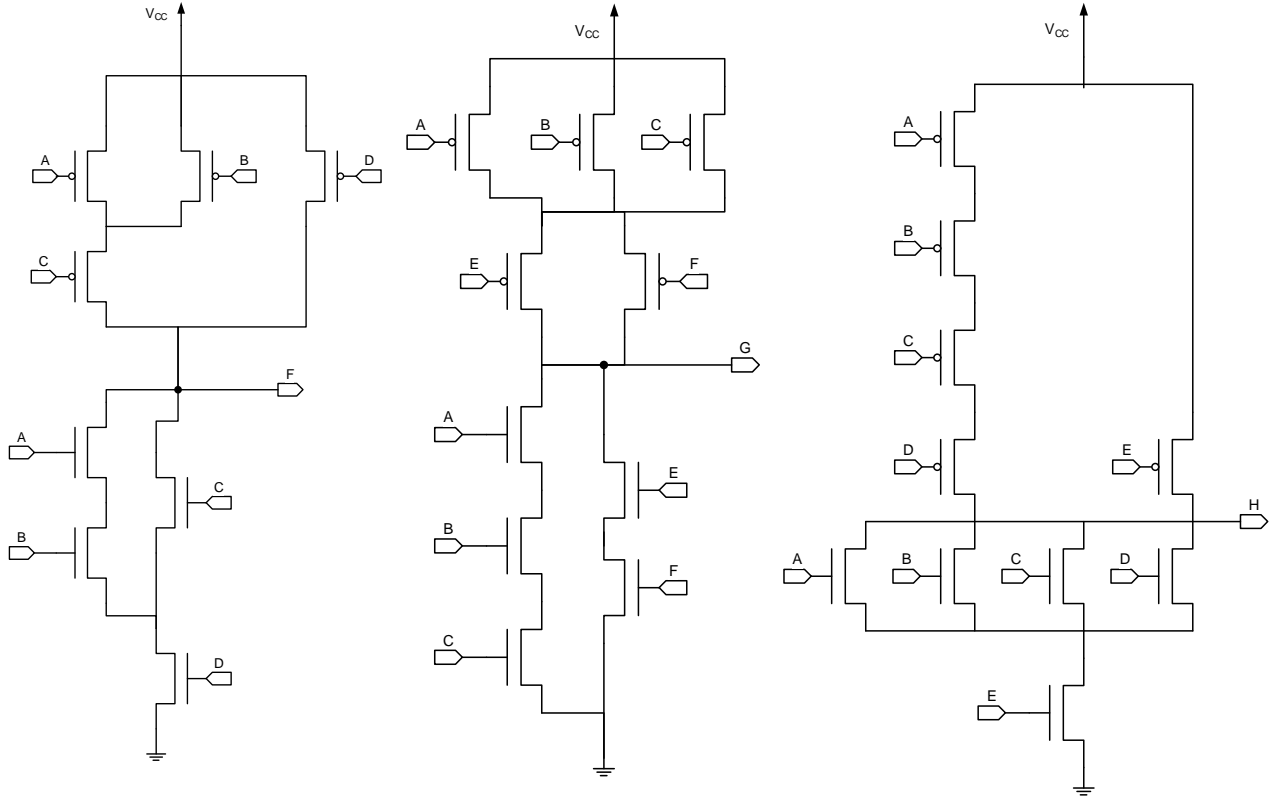


0201 Circuits intégrés numériques – quelques bases

1. Donnez la fonction logique réalisée par les circuits CMOS suivants.



2. Donnez un circuit CMOS pour réaliser la fonction logique T de la table de vérité suivante. Utilisez le moins de transistors possible. Les entrées A, B et C ne sont pas disponibles en version inversée.

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

0202 Mémoires PROM

3. Estimez le nombre de transistors, de diodes et de fusibles programmables utilisés dans une ROM à 16 mots de 8 bits si elle est réalisée avec un circuit CMOS.

4. Programmez le PROM suivant pour implémenter le code VHDL qui suit. Étiquetez correctement les ports d'entrée et de sortie, et placer des boulets (•) aux intersections de lignes à relier ensemble.

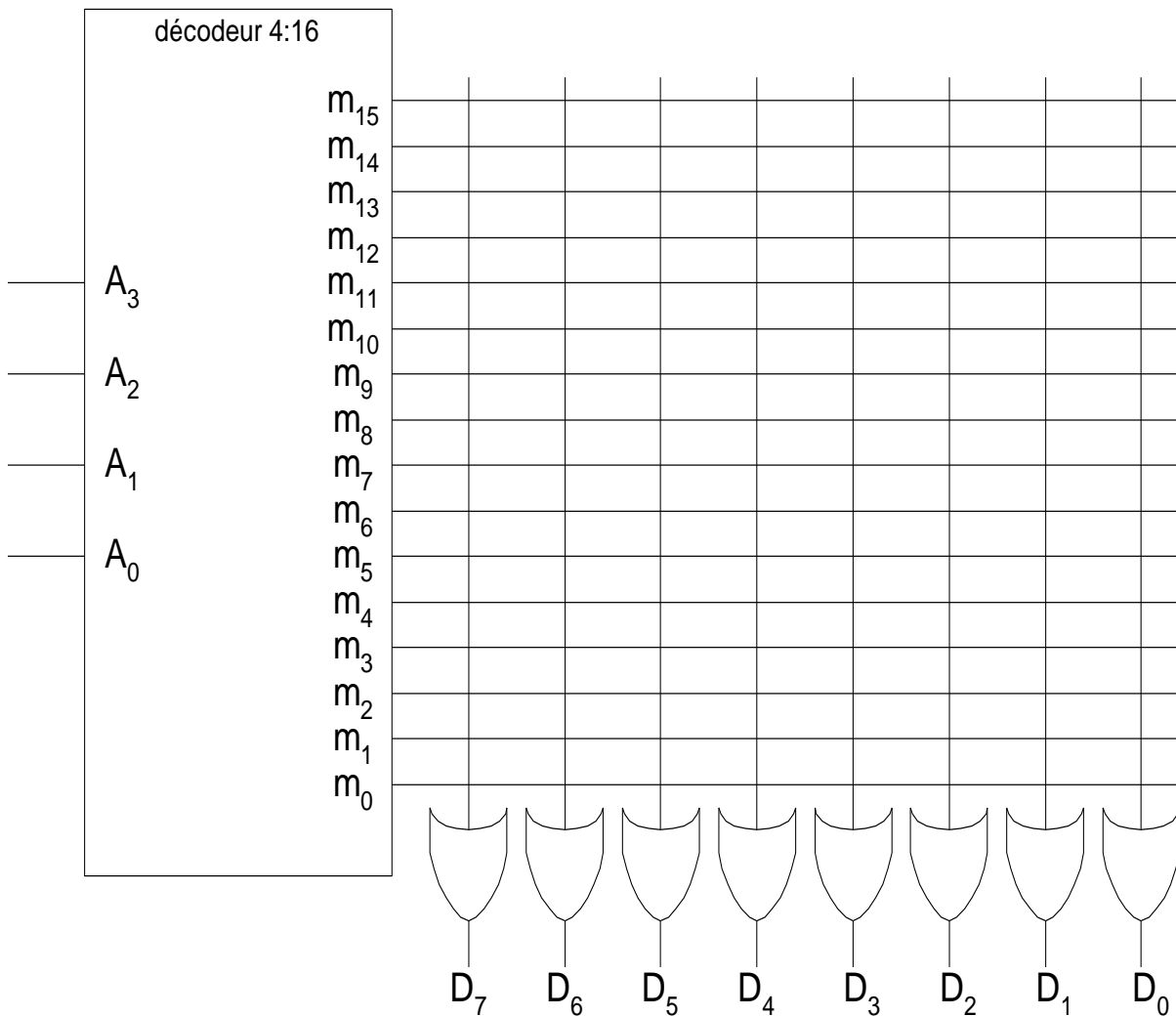
```

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

-- multiplie le nombre en entrée par 3

entity fois3 is
port (
    n : in unsigned(2 downto 0);
    nfois3: out unsigned(4 downto 0)
);
end fois3;

architecture arch1 of fois3 is
signal temp : natural range 0 to 21;
begin
    with to_integer(n) select
        temp <=
            0 when 0,
            3 when 1,
            6 when 2,
            9 when 3,
            12 when 4,
            15 when 5,
            18 when 6,
            21 when 7,
            0 when others;
    nfois3 <= to_unsigned(temp, 5);
end arch1;
    
```



0203 PLA, PAL, GAL et CPLD

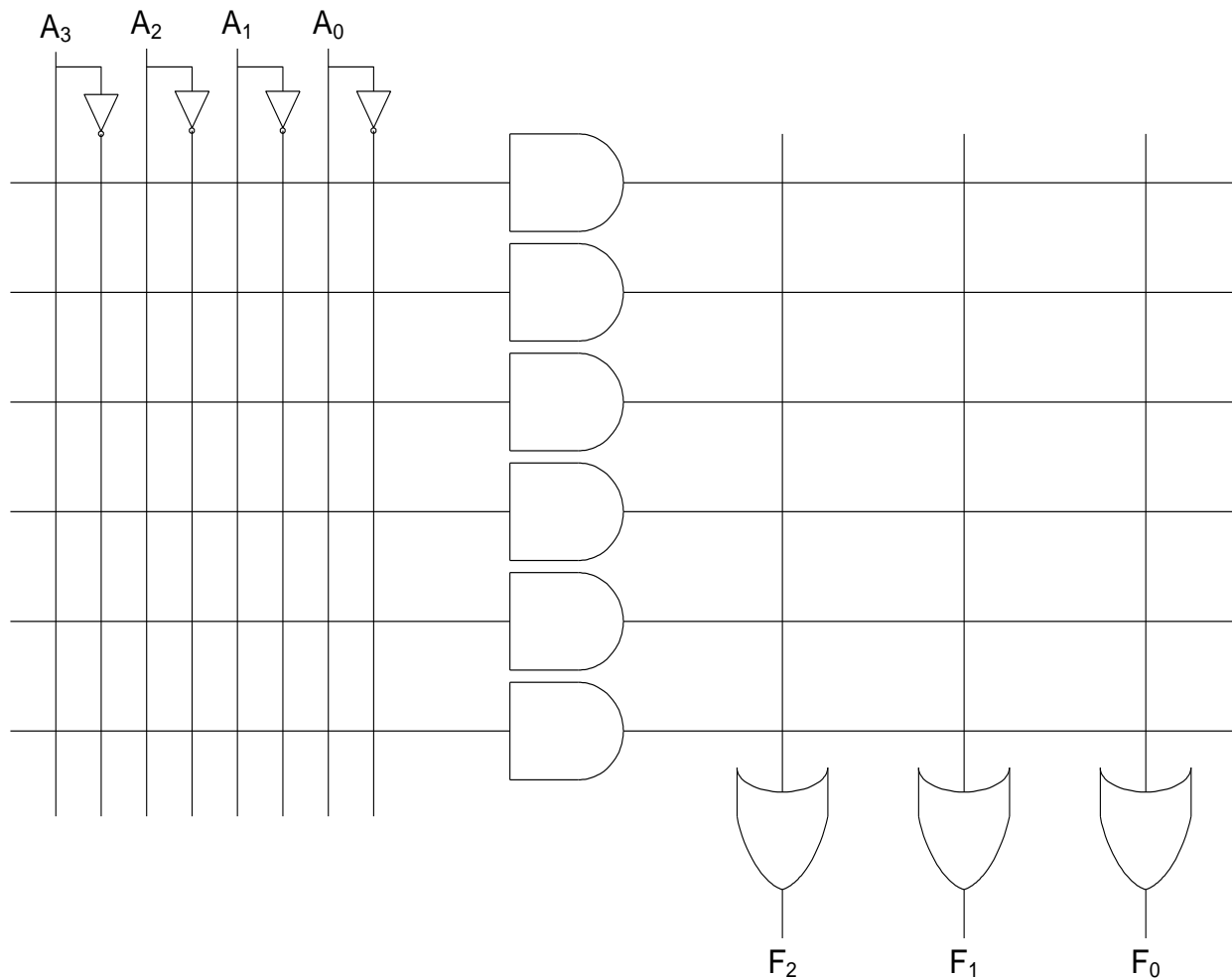
5. Quelles sont les similarités architecturales entre :
 - a. Une PROM et un PLA?
 - b. Un PLA et un PAL?
 - c. Un PAL et un CPLD?

6. Soit le code VHDL suivant. Compléter le circuit PLA donné pour implémenter la fonctionnalité du module `combinatoire2`. Étiqueter correctement les ports d'entrée et de sortie, et placer des boulets (•) aux intersections des lignes à relier ensemble.

```
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;
```



0204 Les FPGA: structure, fonctionnement et utilisation

7. À propos des FPGA

- Donnez le diagramme général d'un FPGA; identifiez et décrivez brièvement les trois types de composantes principales.
- Donnez le diagramme général d'une tranche de type L d'un FPGA Artix-7 et identifiez les quatre composantes principales. Il n'est pas nécessaire de montrer toutes les interconnexions possibles.
- Énumérez et décrivez brièvement cinq composantes secondaires d'un FPGA.

8. Implémentez le module décrit par le code VHDL suivant avec un circuit CMOS, la ROM donnée, le PLA donné, le PAL donné et le FPGA donné.

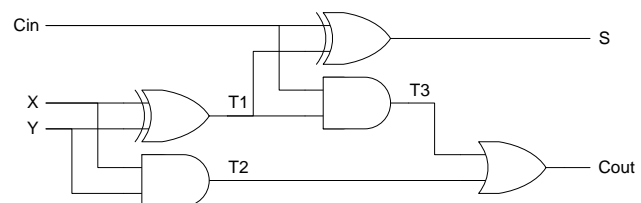
Étiquetez correctement les ports d'entrée et de sortie. Pour la ROM, le PLA et le PAL, placez des boulets (•) aux intersections de lignes à relier ensemble.

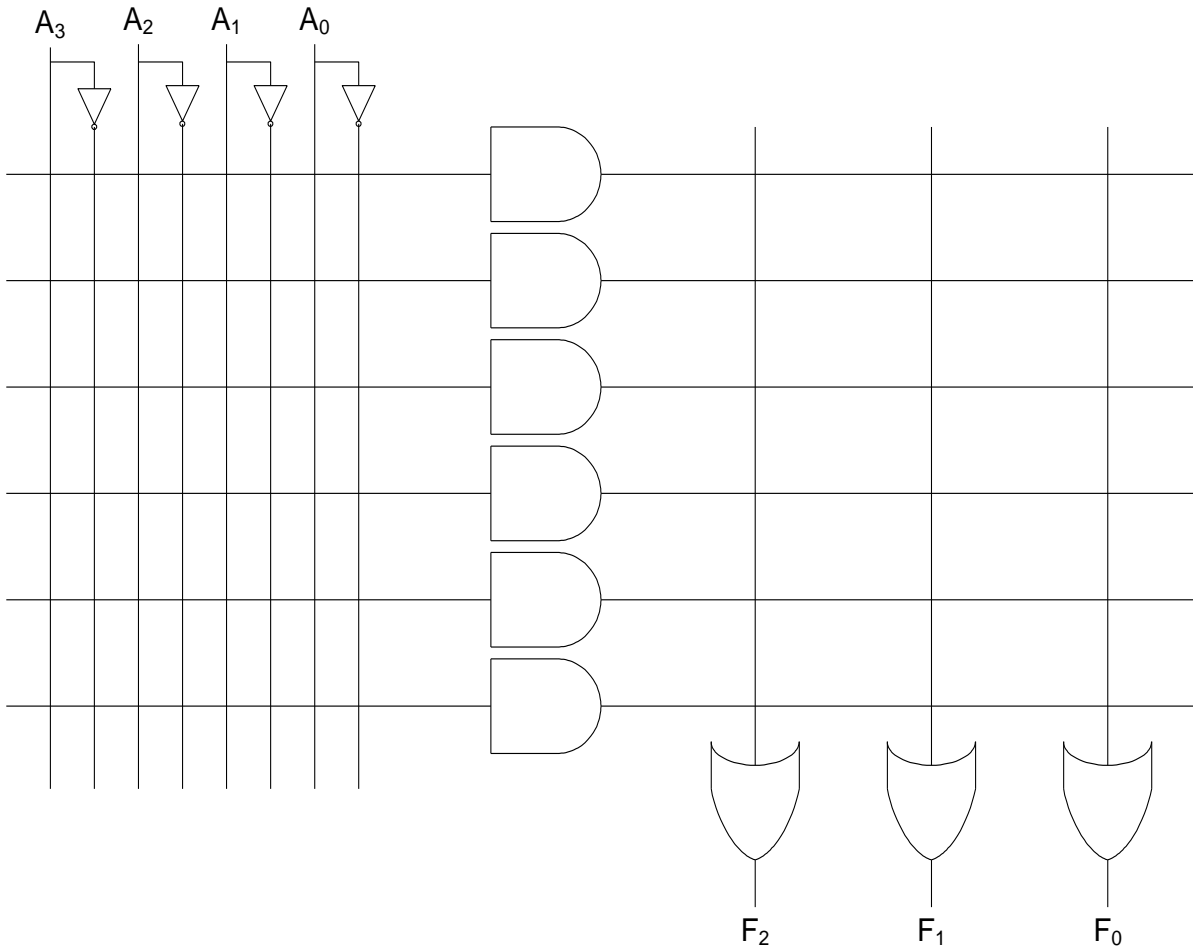
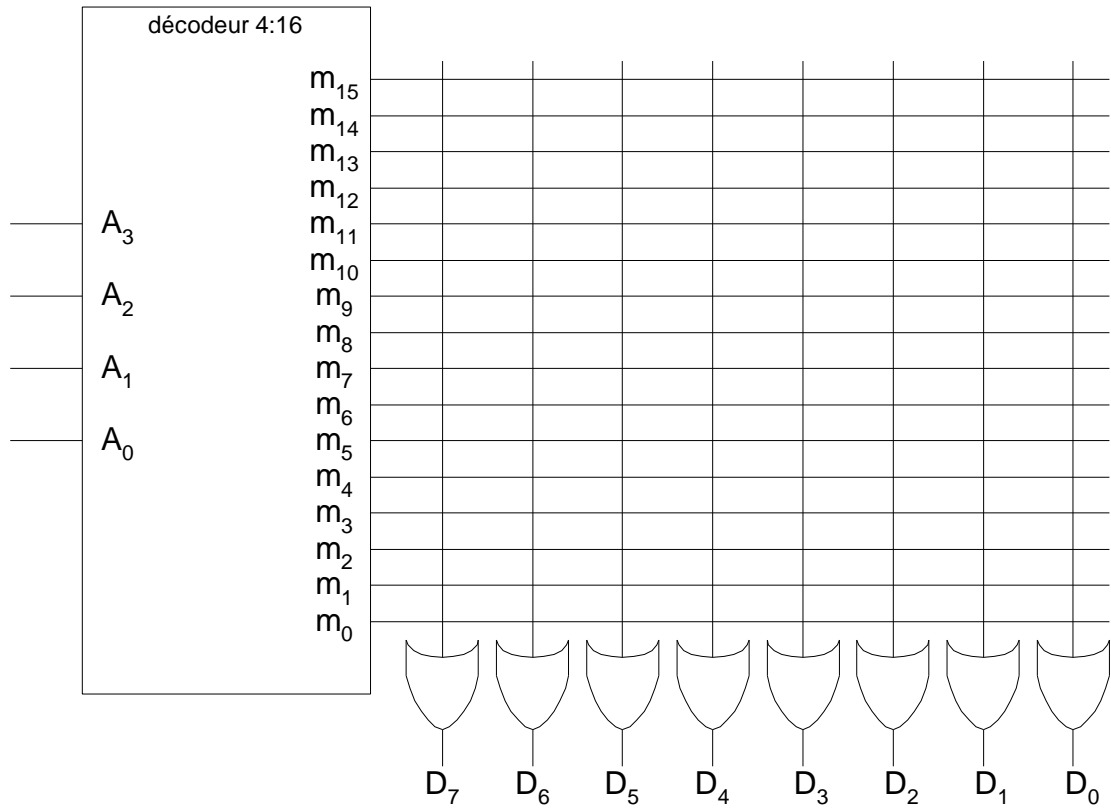
Pour le FPGA, indiquer directement sur le dessin où chaque signal se situe ainsi que les interconnexions entre les blocs. Indiquer dans les tables de vérité fournies le contenu de chacune des tables de conversion que vous utilisez.

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

entity add3bits is
  port (
    Cin, X, Y : in std_logic;
    Cout, S : out std_logic
  );
end add3bits;

architecture flotdonnees of add3bits is
  signal T1, T2, T3 : std_logic;
begin
  S <= T1 xor Cin;
  Cout <= T3 or T2;
  T1 <= X xor Y;
  T2 <= X and Y;
  T3 <= Cin and T1;
end flotdonnees;
```





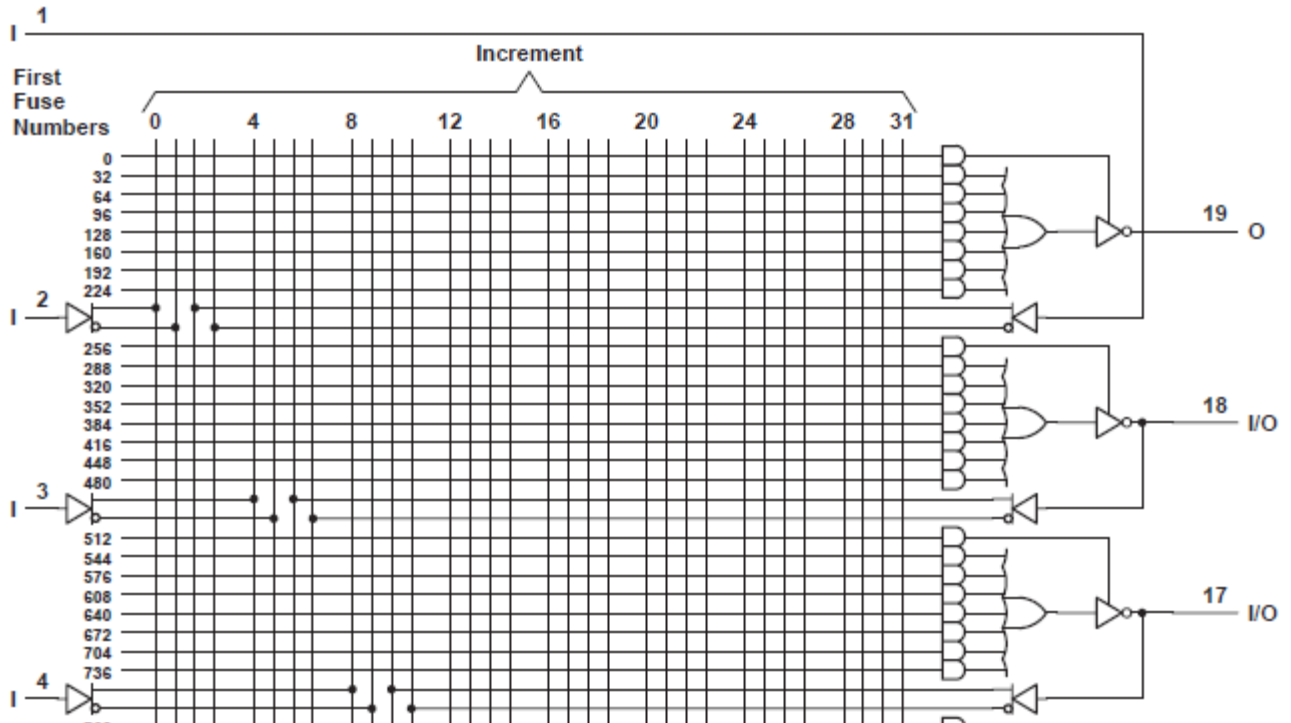
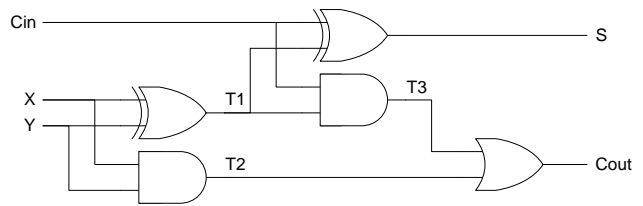
```

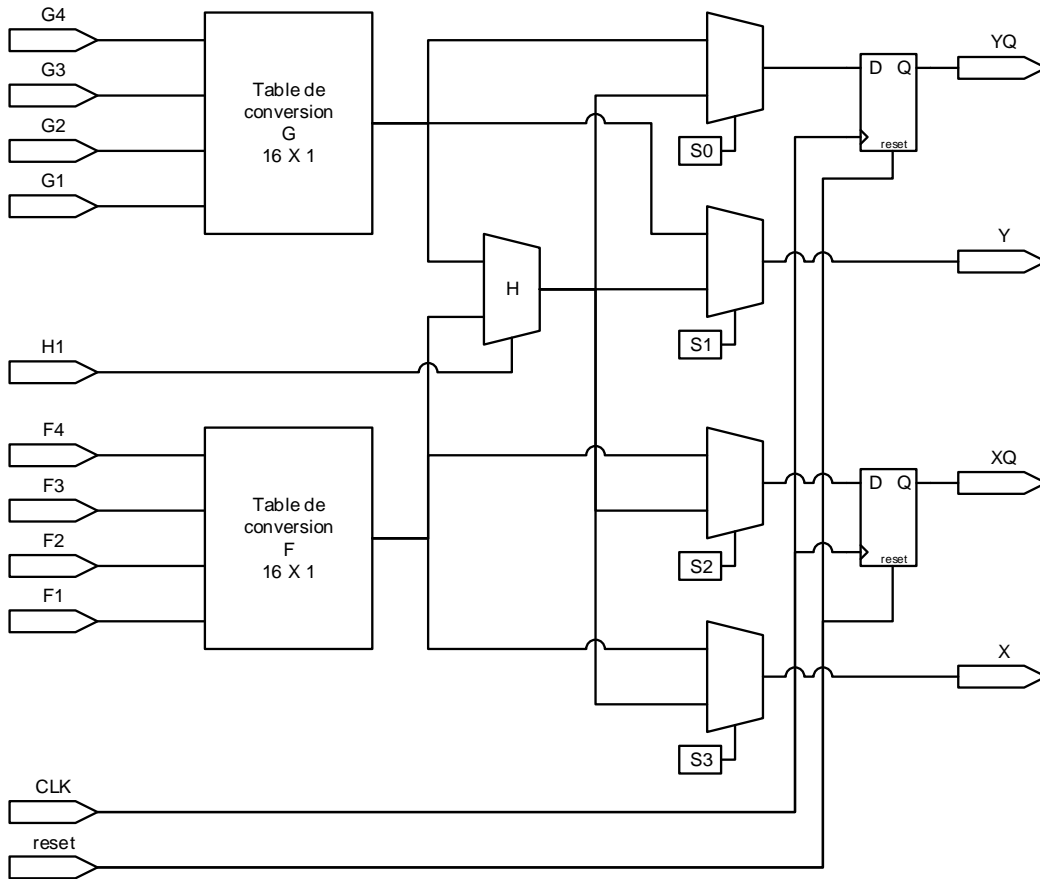
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity add3bits is
    port (
        Cin, X, Y : in std_logic;
        Cout, S : out std_logic
    );
end add3bits;

architecture flotdonnees of add3bits is
    signal T1, T2, T3 : std_logic;
begin
    S <= T1 xor Cin;
    Cout <= T3 or T2;
    T1 <= X xor Y;
    T2 <= X and Y;
    T3 <= Cin and T1;
end flotdonnees;

```





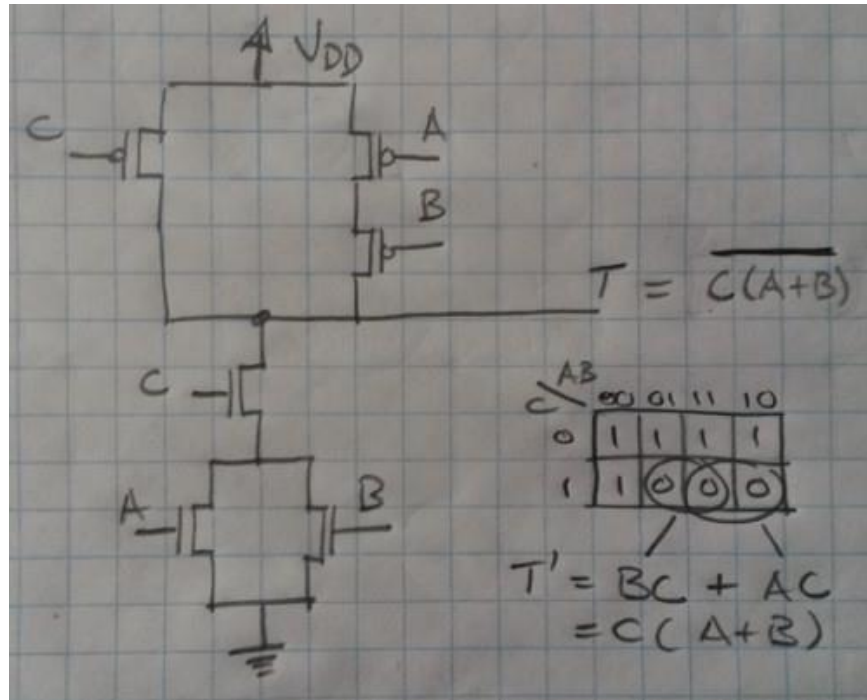
G4	G3	G2	G1	G
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

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 : $F = ((AB + C) \cdot D)'$; $G = (ABC + EF)'$; $H = ((A + B + C + D) \cdot E)'$

2. Réponse : 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.



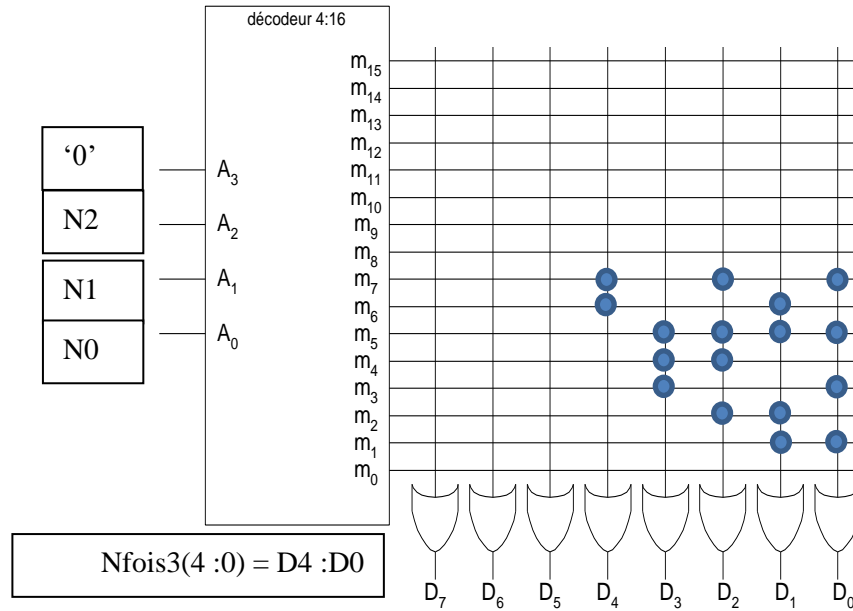
3. Réponse

Réponse (estimation de 1^{er} ordre)

$16 \times 8 = 128$ points de programmation, donc 128 diodes et fusibles

16 portes ET à 4 entrées + 4 inverseurs = $16 \times 10 + 4 \times 2 = 168$ transistors

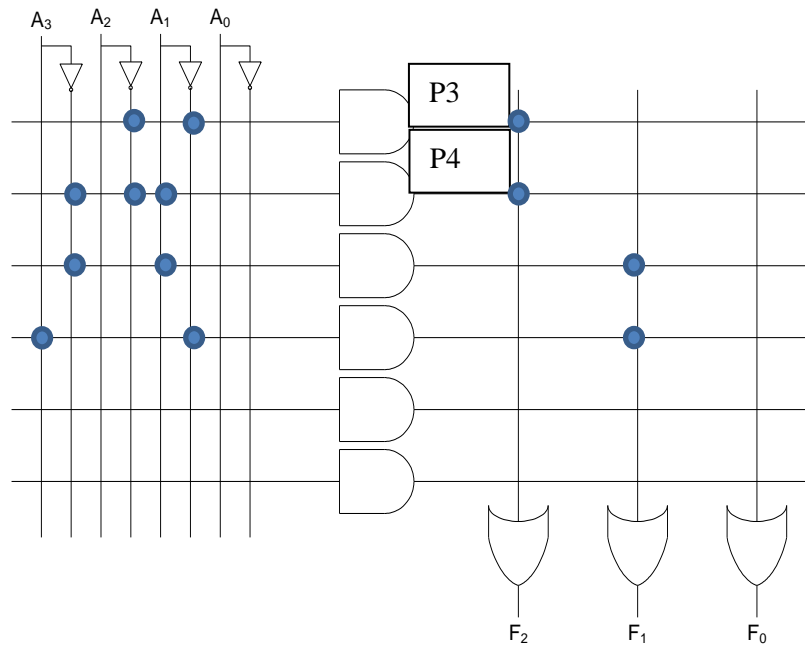
4. Réponse



5. a. Une PROM réalise (ou décode) tous les produits de termes (mintermes) formés à partir des entrées, alors qu'un PLA ne peut pas les réaliser tous. Les produits à décoder sont programmables dans le PLA.
- b. Dans un circuit PAL, on peut aussi, comme dans le PLA, programmer les mintermes à décoder. Par contre, le réseau OU (qui effectue la somme des produits) est fixe.
- c. Un CPLD est effectivement composé de plusieurs circuits PALs, reliés entre eux par un réseau d'interconnexions.

6. Réponse :

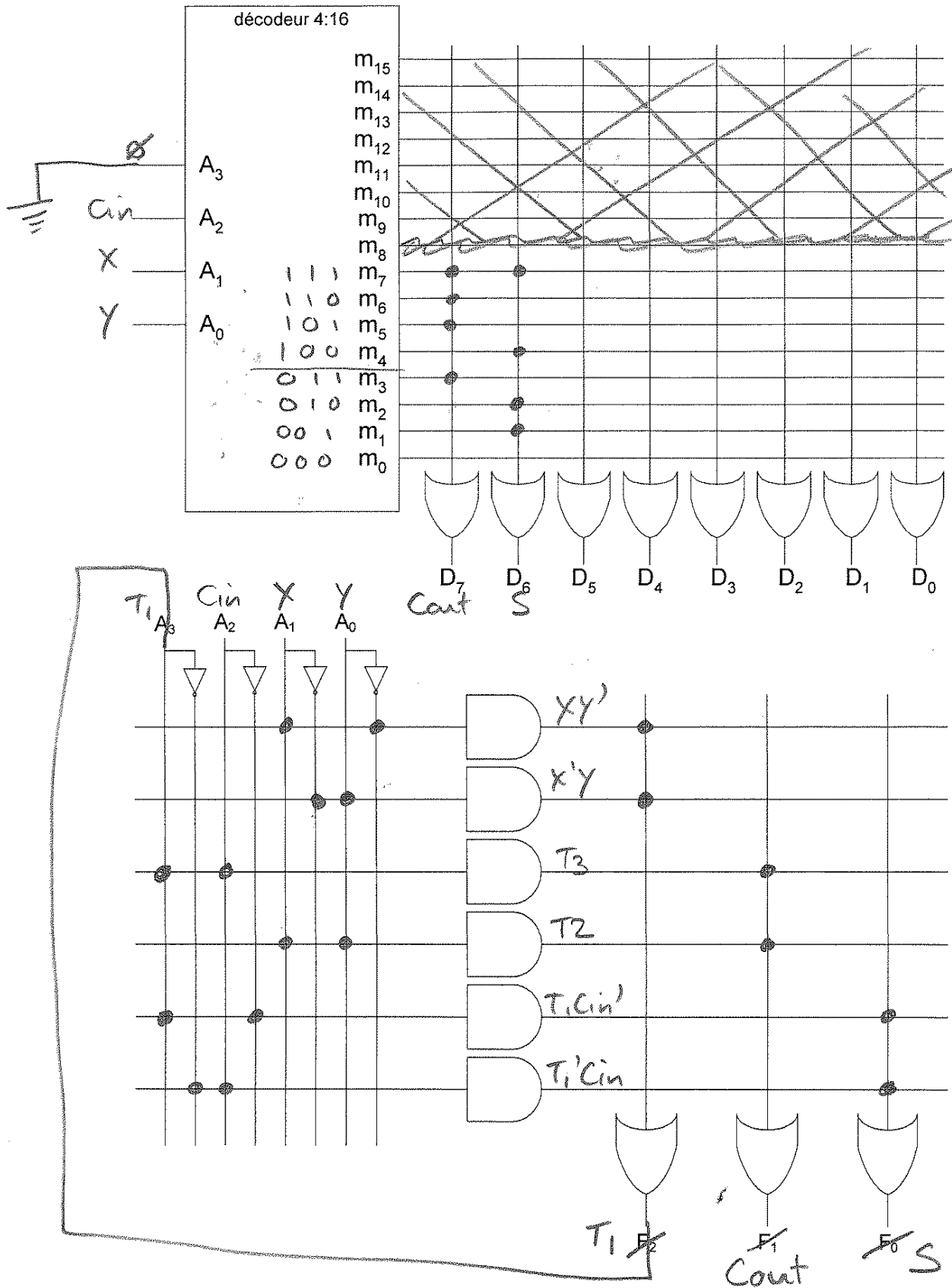
Les ports $x(3:1)$ sont reliés à $A(3:1)$. Les ports $y(2:1)$ sont reliés à $F(2:1)$. La fonction OUX doit être réalisée en sommes de produits : $x_1 \text{ xor } x_3 = x_1'x_3 + x_1x_3'$

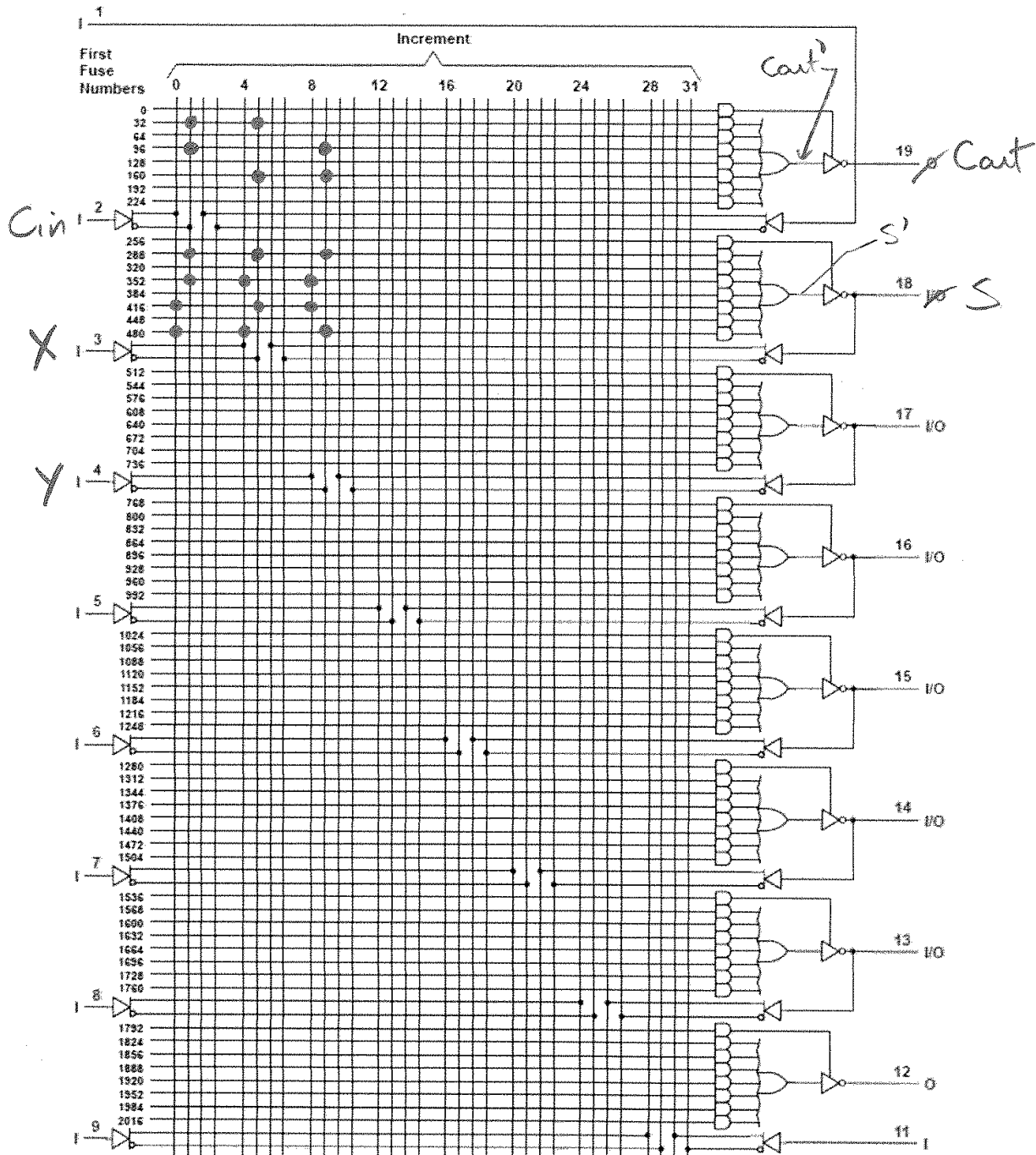


7. Réponses :

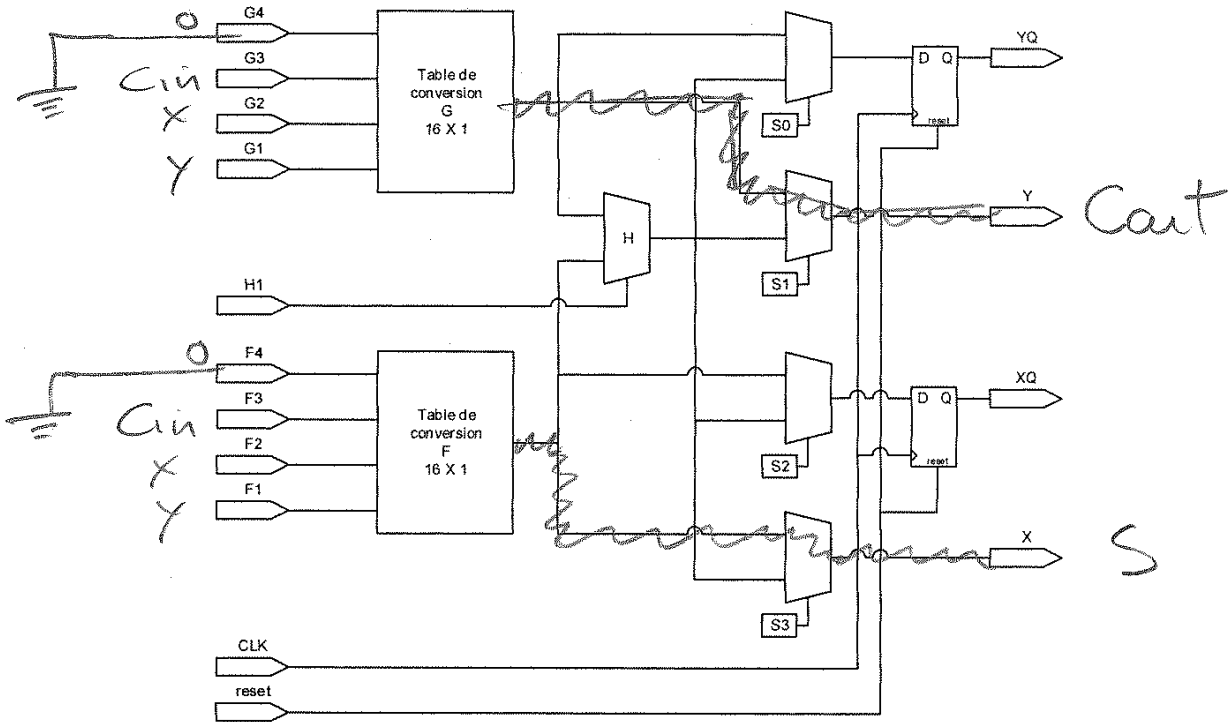
- Voir les diapositives du cours. On note les blocs de logique configurable, les blocs d'entrées-sorties et le réseau d'interconnexions.
- Voir les diapositives du cours. On note les tables de correspondance, les portes logiques pour l'addition rapide, les multiplexeurs pour router les signaux et les éléments à mémoire.
- Blocs de mémoire intégrée, fonctions arithmétiques avancées, modules de génération d'horloge, réseaux de distribution d'horloges et microprocesseurs fixes.

8. Réponses





On a $Cout' = Cin'X' + Cin'Y' + X'Y'$
 $S' = Cin'X'Y' + Cin'XY + CinX'Y + CinXY'$



G4	G3	G2	G1	G
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

F4	F3	F2	F1	F
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1