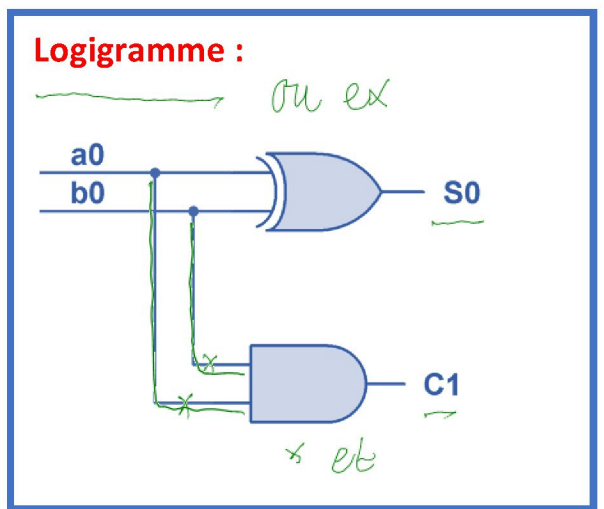
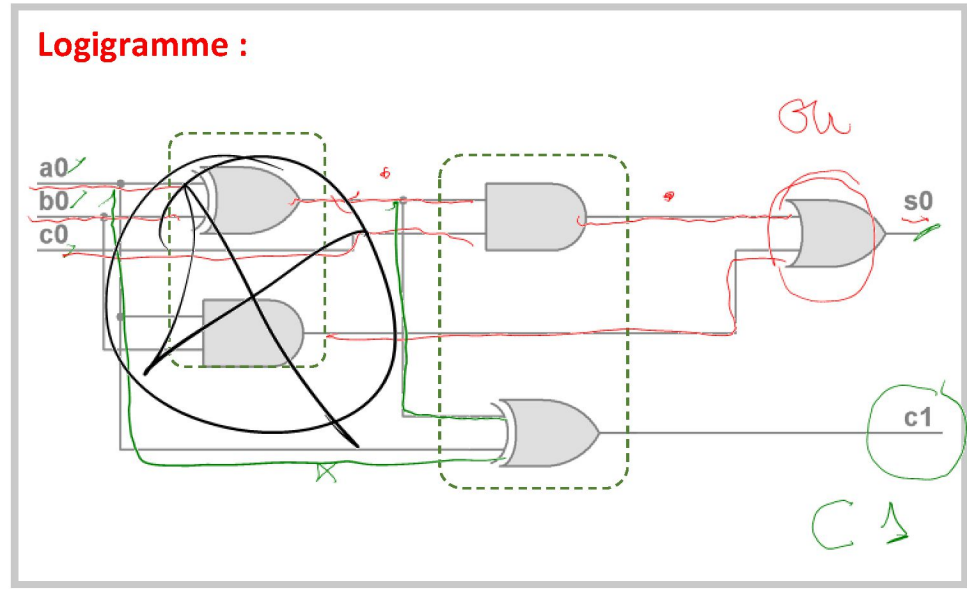
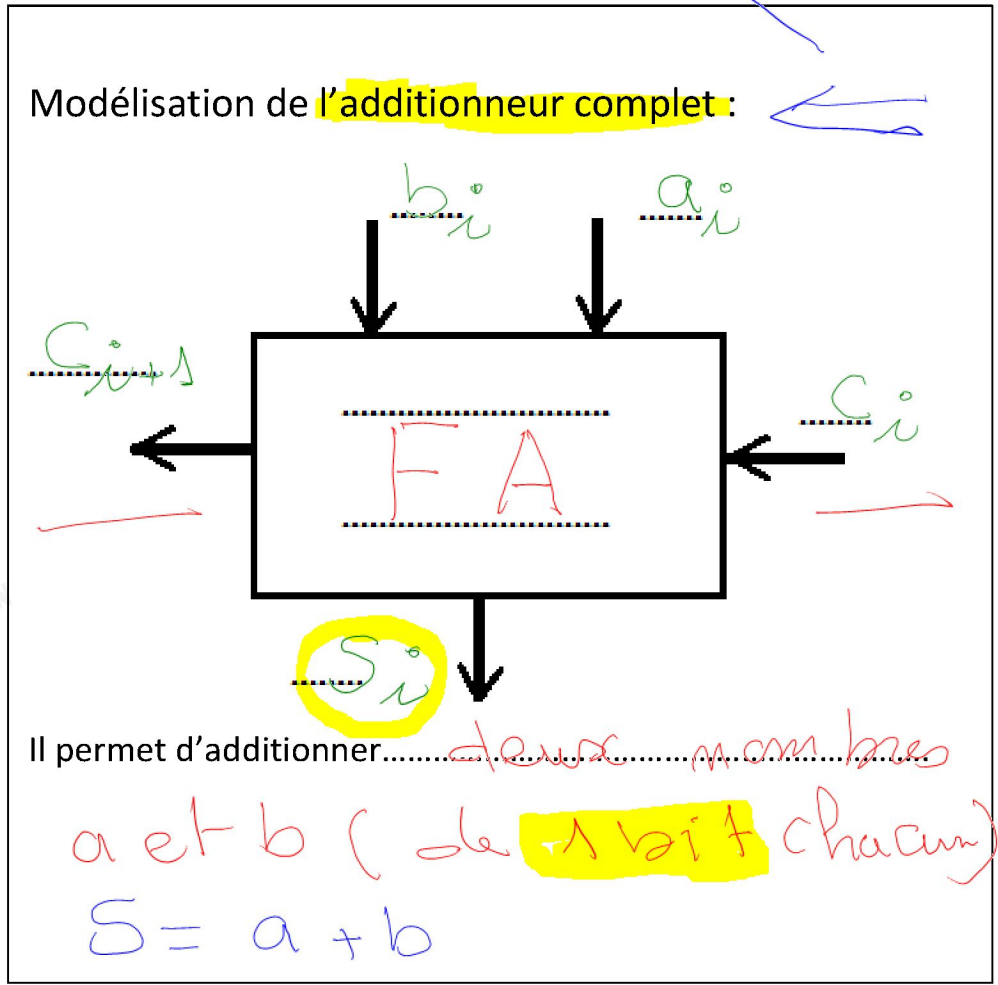


$A (10101)$
 $B (11111)$
 5 bits



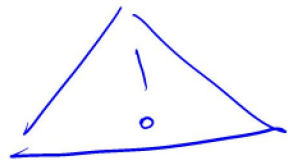
Pour un demi-additionneur :
 S la somme ; C la retenue
 $S_0 = a_0 \oplus b_0$
 $C_1 = a_0 \cdot b_0$



Pour un additionneur :
 S la somme ; C la retenue
 $S_0 = (a_0 \oplus b_0) \oplus c_0$
 $C_1 = (a_0 \oplus b_0) \cdot c_0 + a_0 \cdot b_0$

Comment réaliser une opération d'addition en binaire :

	S	C
$0 + 0 =$		
$0 + 1 =$		
$1 + 0 =$		
$1 + 1 =$		
$1 + 1 + 1 =$		



On veut additionner les deux mots binaire A et B .

A(1100) B(1110)

$$A + B = S$$

$$\begin{aligned} (1)_2 &\rightarrow (1)_{10} \\ + (1)_2 &\rightarrow (1)_{10} \end{aligned}$$

$$\begin{aligned} (10)_2 & \quad (2)_{10} \end{aligned}$$

$$\begin{array}{r} C \ 1 \\ 1100 \\ + 1110 \\ \hline 11010 \\ S \end{array}$$



في دارك... إتهنوني على قرابتك إصغارك



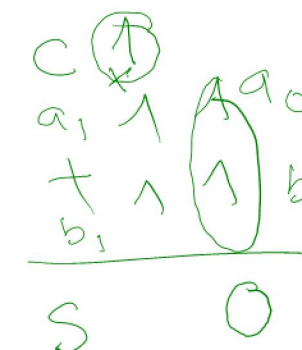
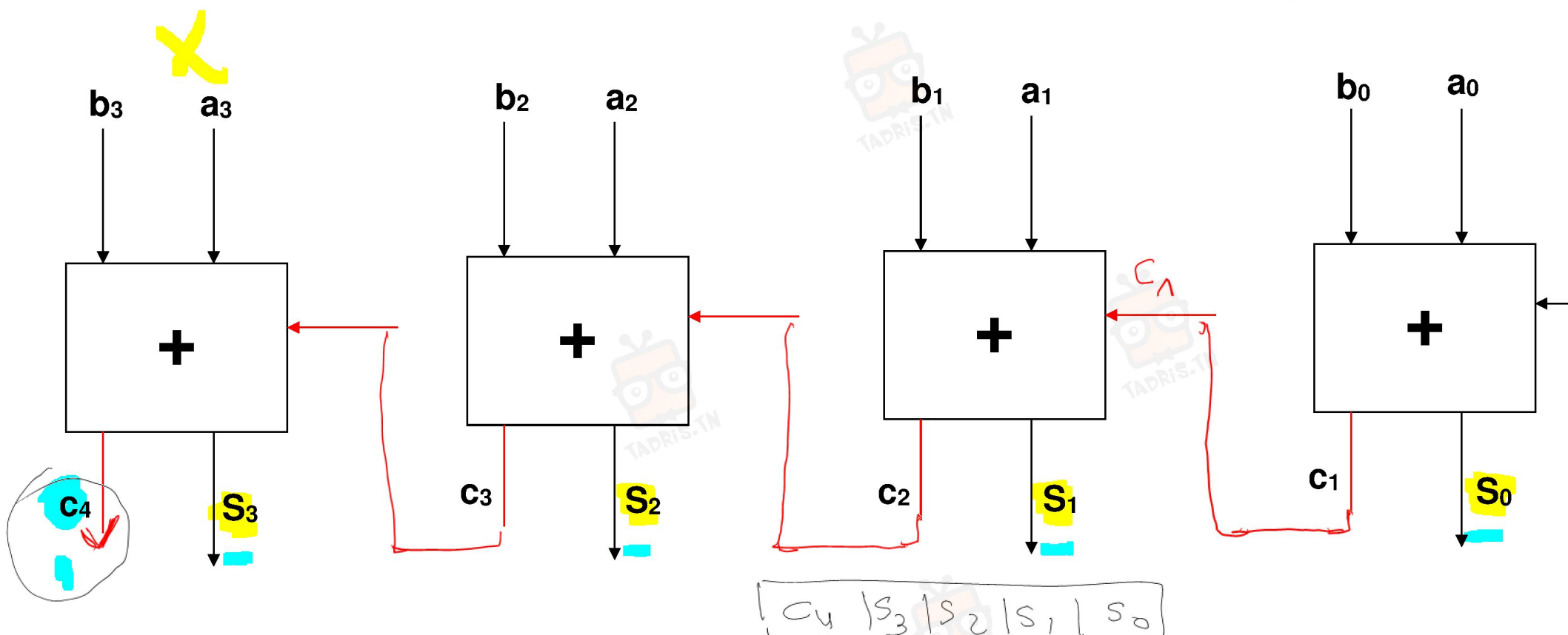
Mise en cascade de plusieurs additionneurs

Ce montage permet d'

additionner les nombres binaires A et B (de 4 bits chacun)

$$A (a_3 a_2 a_1 a_0)$$

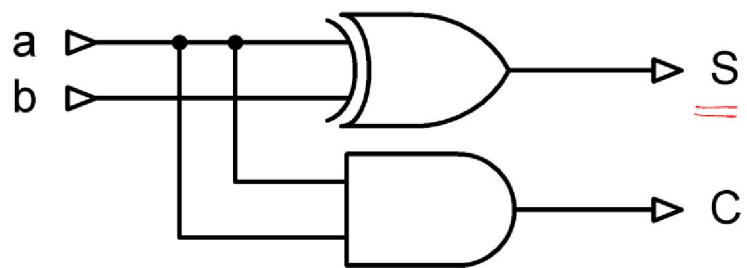
$$B (b_3 b_2 b_1 b_0)$$



$C_4 | S_3 | S_2 | S_1 | S_0$

Exercice 1 :

On donne le d montage suivant :



1- Que représente :

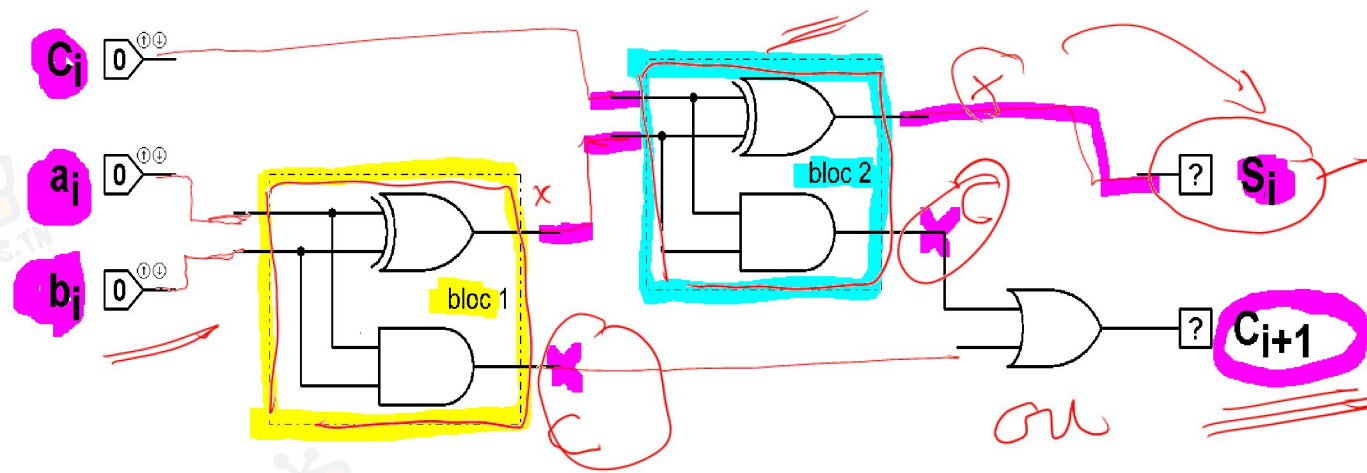
* La sortie **S** : $a \oplus b$

* La sortie **C** : $a \cdot b$

* Donner le nom du montage : un demi-additionneur

\oplus ou ex

2- Compléter le logigramme de l'additionneur complet



a- Identifier les blocs 1 et 2 : ... demi-additionneur

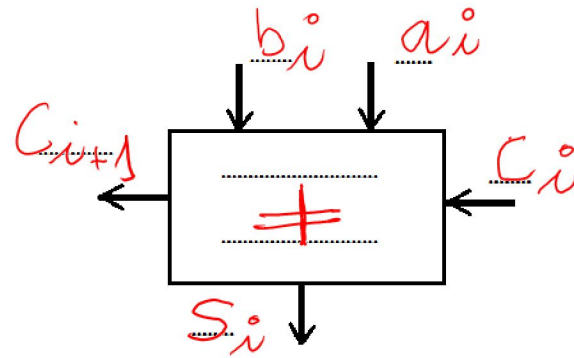
b- Donner l'équation de Si en fonction de ai bi et ci.

$S_i = (a_i \oplus b_i) \oplus c_i$

c- Donner l'équation de Ci+1 en fonction de ai bi et ci.

$C_{i+1} = (a_i \cdot b_i) + (c_i \cdot (a_i \oplus b_i))$

d- Modéliser le montage :



3- Le montage étudié permet-il d'additionner deux nombres de 4bits, justifier.

non; car sa capacité est de 1 bit.

Exercice 2 :

Réaliser les opérations suivantes

$1001_{(2)} + 1000_{(2)}$

$$\begin{array}{r} 1001 \\ + 1000 \\ \hline 10001 \end{array}$$

$1101_{(2)} + 1000_{(2)}$

$$\begin{array}{r} 1101 \\ + 1000 \\ \hline 10101 \end{array}$$

$11011101_{(2)} + 10001000_{(2)}$

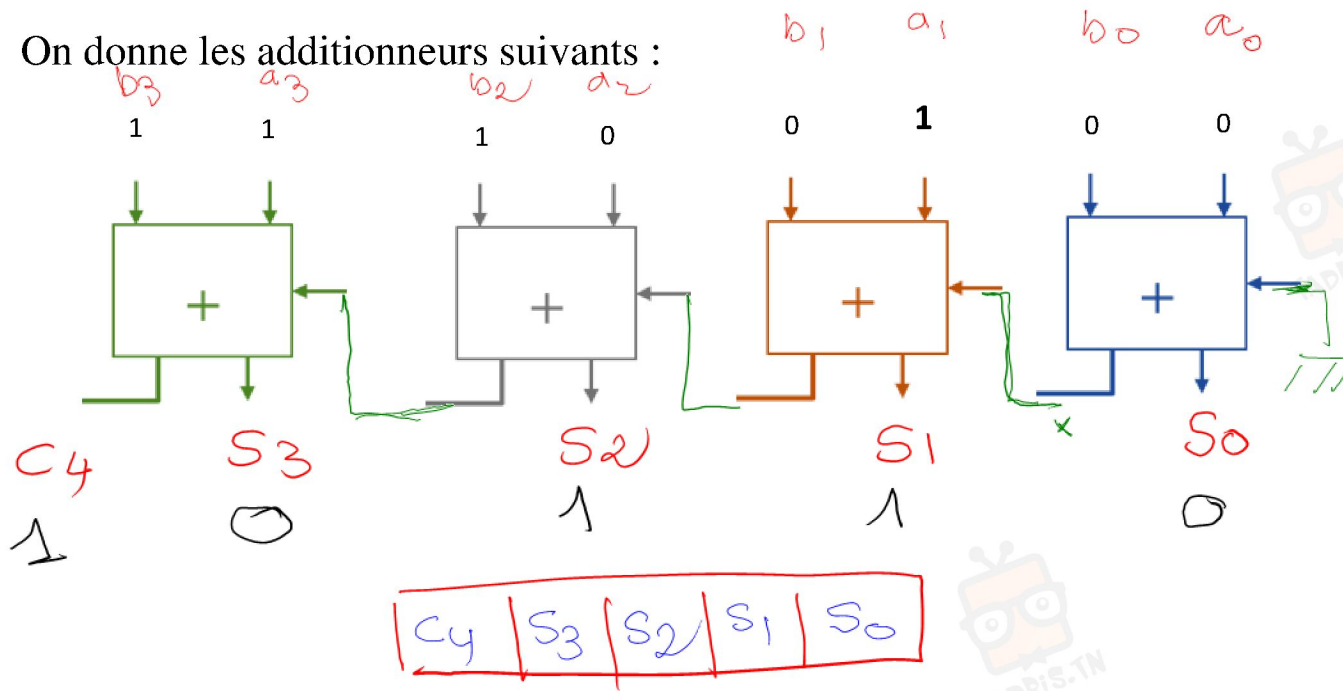
$$\begin{array}{r} 11011101 \\ + 10001000 \\ \hline 10110101 \end{array}$$

$10111011_{(2)} + 10001000_{(2)}$

$$\begin{array}{r} 10001000 \\ + 10111011 \\ \hline 10100011 \end{array}$$

Exercice 3 :

On donne les additionneurs suivants :



1- Compléter le montage

2- déterminer les valeurs de :

A (A₃A₂A₁A₀) = ... 1 0 1 0 ...

B (B₃B₂B₁B₀) = ... 1 1 0 0 ...

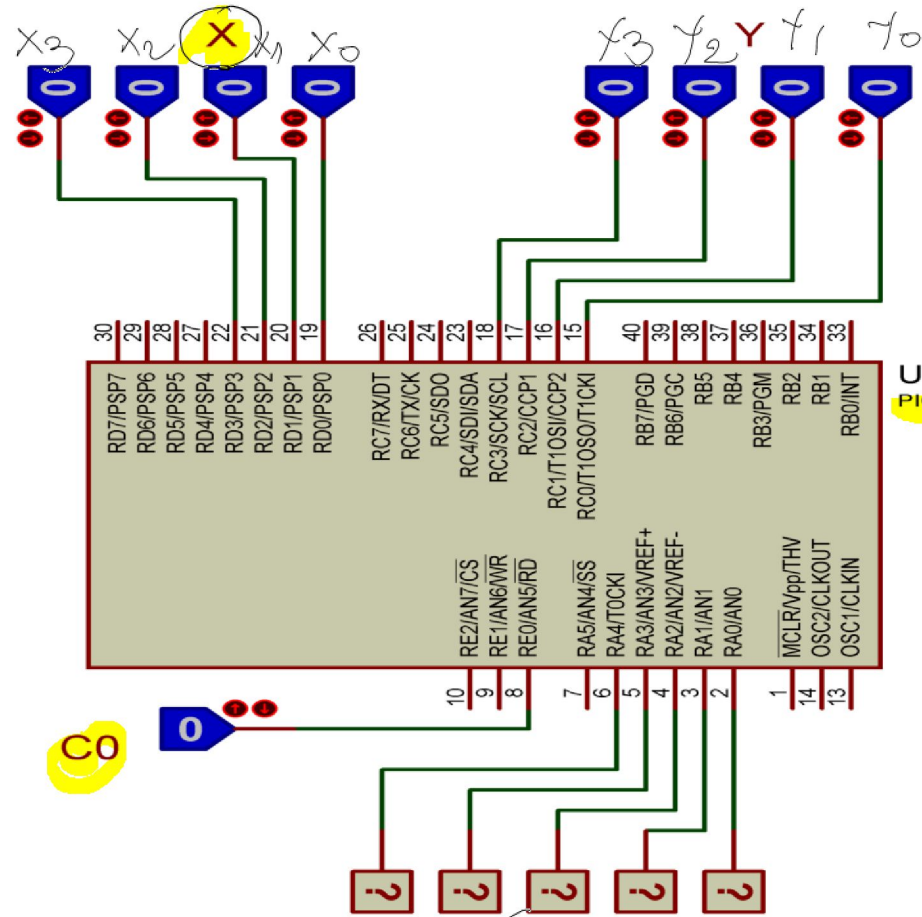
S (S₃S₂S₁S₀) = ... 0 1 1 0 ... et C₄ = ... 1 ... avec C₀ = 0

3- Donner le nom de ce montage.....

c'est un additionneur à 4 bits

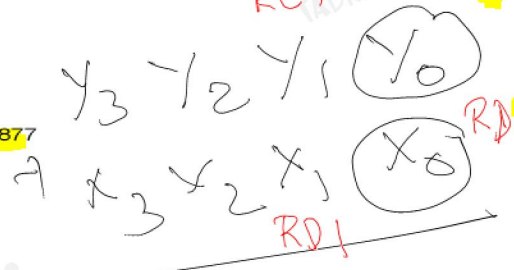
$$\begin{array}{r}
 1\ 1\ 0\ 1\ 0 \\
 +\ 1\ 1\ 0\ 0 \\
 \hline
 1\ 0\ 1\ 1\ 0
 \end{array}$$

Exercice 4 :



$S = X + Y$

adresse non
utilisées sera
considérée comme
une entrée



Compléter le programme en mikroC **8 bits**

Algorithme

Déclarer 4 variables du type octet :
X et Y Co représente la retenue
précédente et S représente la somme.
Mot clé du début du programme

- Configurer le port D
- Configurer le port C
- Configurer le port A
- Configurer le port E

Port A libéré en numérique

Initialiser S à 0
Boucle TANT QUE (vraie)
DÉBUT
X ← port D
Y ← port C
Co ← RE0
S ← X + Y + Co
Port A ← S
Fin du programme
FIN

Programme

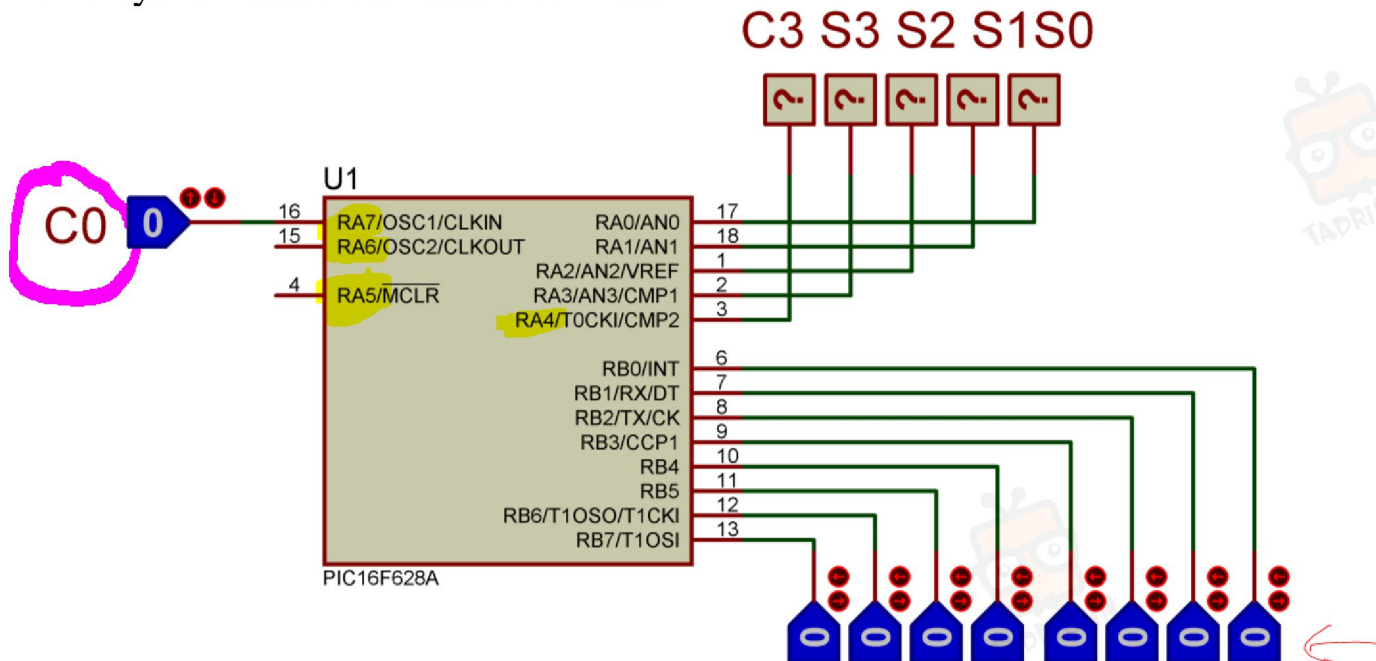
```

char X, Y, S, Co;
void main()
{
    trisD = ob11111111;
    trisC = ob11111111;
    trisA = ob11100000;
    trisE = ob11111111;

    INTCON = 0x06;
    S = 0;
    while(1)
    {
        X = portD;
        Y = portC;
        Co = portE & B0;
        S = X + Y + Co;
        PortA = S;
    }
}
    
```

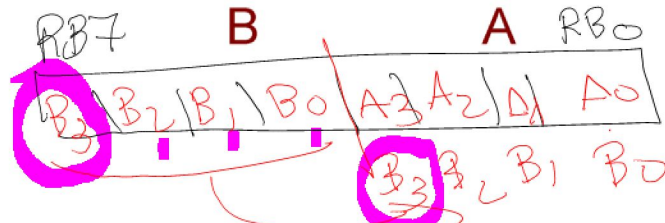

Exercice 5 :

Le schéma du circuit illustre un additionneur de deux nombres binaires A et B ayant chacun un format de 4 bits :



Compléter le programme en mikroC

$$B = B \gg 4$$



Algorithme

Déclarer 4 variables du type octet : char A; B; Co; S;
 A et B Co représente la retenue précédente et S représente la somme.

Mot clé du début du programme

Configurer le port B
 Configurer le port A
 Désactiver les comparateurs analogiques
 Initialiser S à 0

Boucle TANT QUE (vraie)
 DÉBUT
 A ← port B
 Masquer 4 bits (B7 ...B4)
 B ← port B
 Décalage à droite de 4 bits
 Co ← RA07

S ← A + B + Co
 Port A ← S
 Fin du programme
 FIN

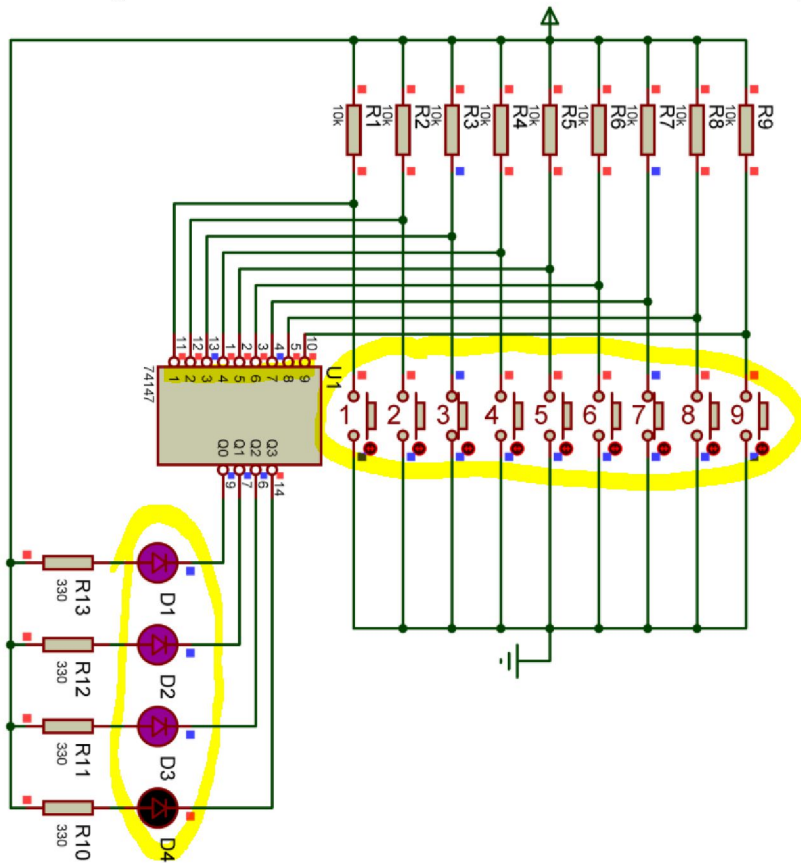
Programme

```

void main()
{
    trisB = ob...11111111;
    trisA = ob...10000000;
    CMCON = 0x07;
    S = 0;
    while (1)
    {
        A = PortB; // A reçoit la valeur de B
        A = A & ob...00001111; // Masquer les bits B7-B4
        B = PortB;
        B = B >> 4; // Décalage à droite de 4 bits
        Co = PortA & 0x07; // Co ← RA07
        S = A + B + Co;
        PortA = S;
    }
}
    
```


Exercice 6 : Fonction codage décimal/BCD

- Donner les noms des entrées et des sorties du circuit 74147 (0-9) (A B C D).....
- Compléter le tableau en se référant au montage



$$(0 \rightarrow 9)_{10} \Rightarrow (x_3 x_2 x_1 x_0)_{BCD}$$

9	8	7	6	5	4	3	2	1	D	C	B	A
0	0	0	0	0	0	0	0	0	0	0	0	0
1	X	X	X	X	X	X	X	X	1	0	0	1
0	1	X	X	X	X	X	X	0	1	0	0	0
0	0	1	X	X	X	X	X	0	0	1	1	1
0	0	0	1	X	X	X	X	0	0	1	1	0
0	0	0	0	1	X	X	X	0	0	1	0	1
0	0	0	0	0	1	X	X	0	1	0	0	0
0	0	0	0	0	0	1	X	0	0	1	1	1
0	0	0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	0	0	0	0	1	0	0	0	1

BCD
4 bits

0000
0001
0010
0011
0100
0101
0110
0111
1000
1001

- Si deux entrées sont activées en même temps que se passe il ?

la valeur la plus grande sera affichée

27