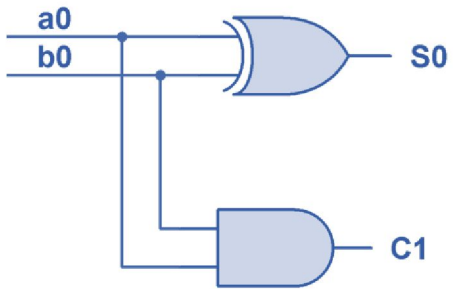


Logigramme :

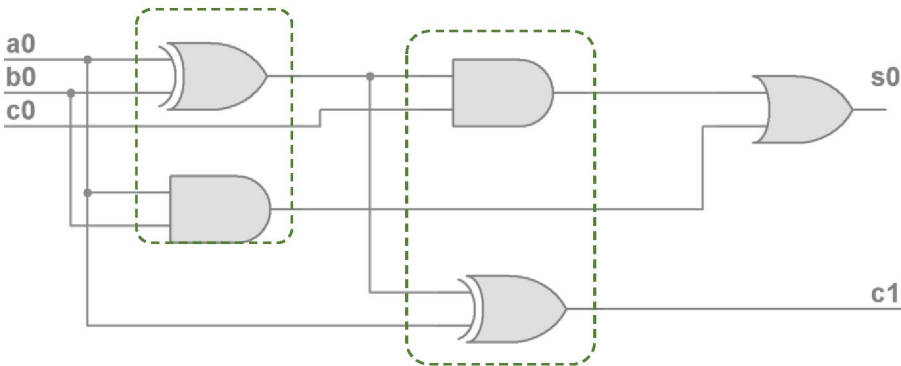


Pour un demi-additionneur :  
S la somme ; C la retenue

S0= .....

C1 = .....

Logigramme :

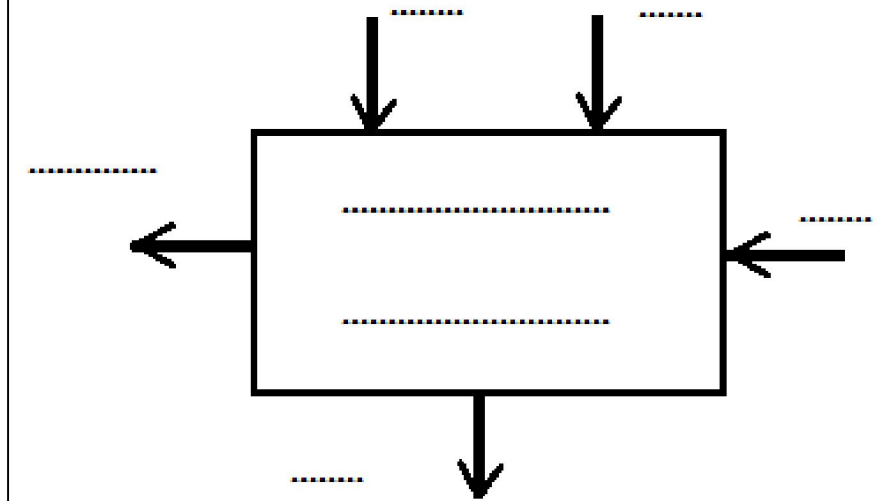


Pour un additionneur :  
S la somme ; C la retenue

S0= .....

C1 = .....

Modélisation de l'additionneur complet :



Il permet d'additionner.....

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)**



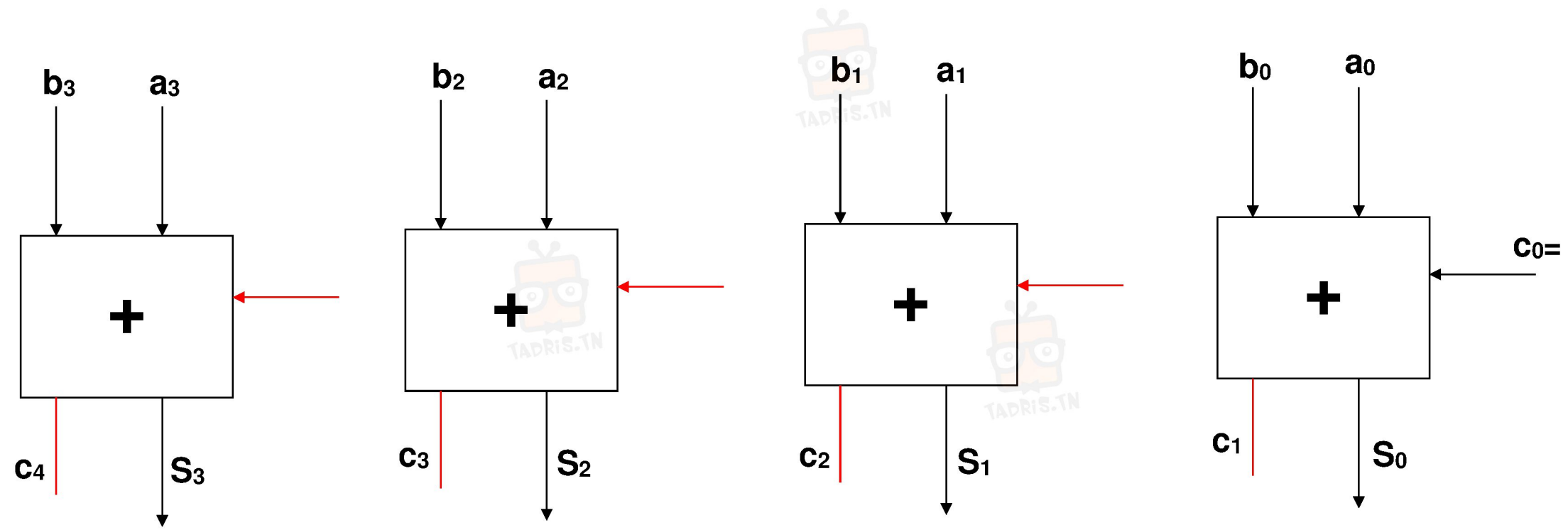
في دارك... إتهنوني على قرابتك إصغارك





### Mise en cascade de plusieurs additionneurs

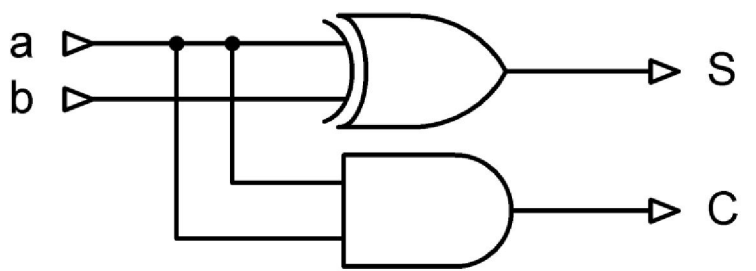
Ce montage permet d'.....





**Exercice 1 :**

On donne le d montage suivant :



1- Que représente :

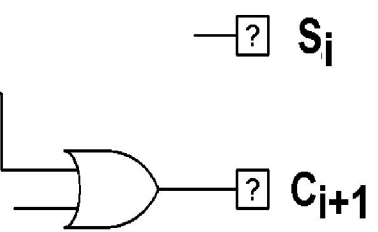
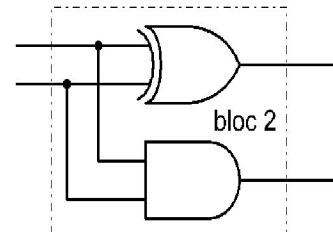
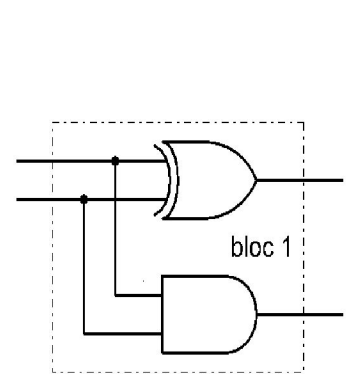
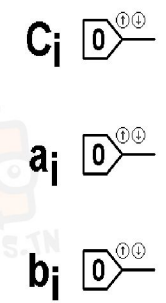
\* La sortie **S** : .....

\* La sortie **C** : .....

\* Donner le nom du montage : .....



2- Compléter le logigramme de l'additionneur complet



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

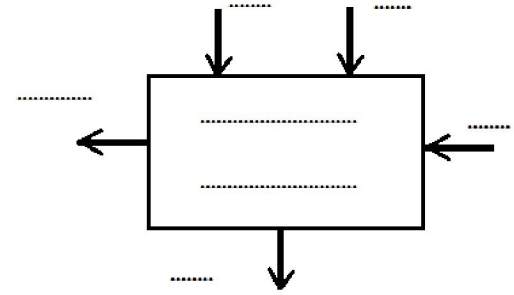
b- Donner l'équation de  $S_i$  en fonction de  $a_i$ ,  $b_i$  et  $c_i$ .

c- Donner l'équation de  $C_{i+1}$  en fonction de  $a_i$ ,  $b_i$  et  $c_i$ .





d- Modéliser le montage :



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

.....

**Exercice 2 :**

Réaliser les opérations suivantes

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

.....  
.....  
.....  
.....

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

.....  
.....  
.....

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

.....  
.....  
.....

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

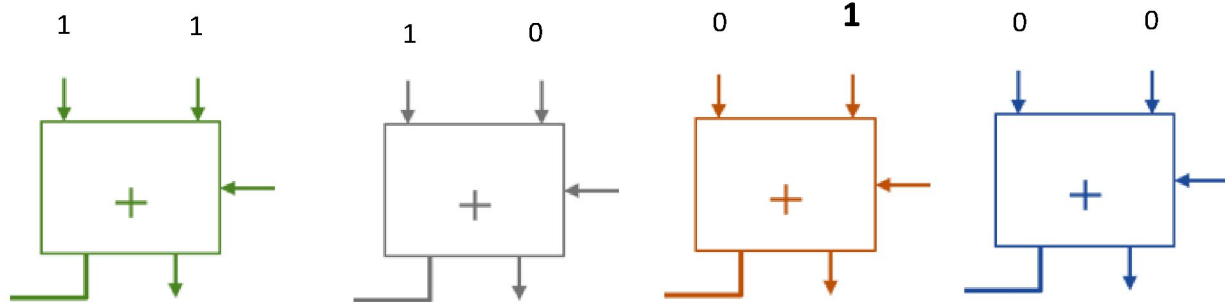
.....  
.....  
.....





**Exercice 3 :**

On donne les additionneurs suivants :



1- Compléter le montage

2- déterminer les valeurs de :

$A (A_3 A_2 A_1 A_0) = \dots\dots\dots$

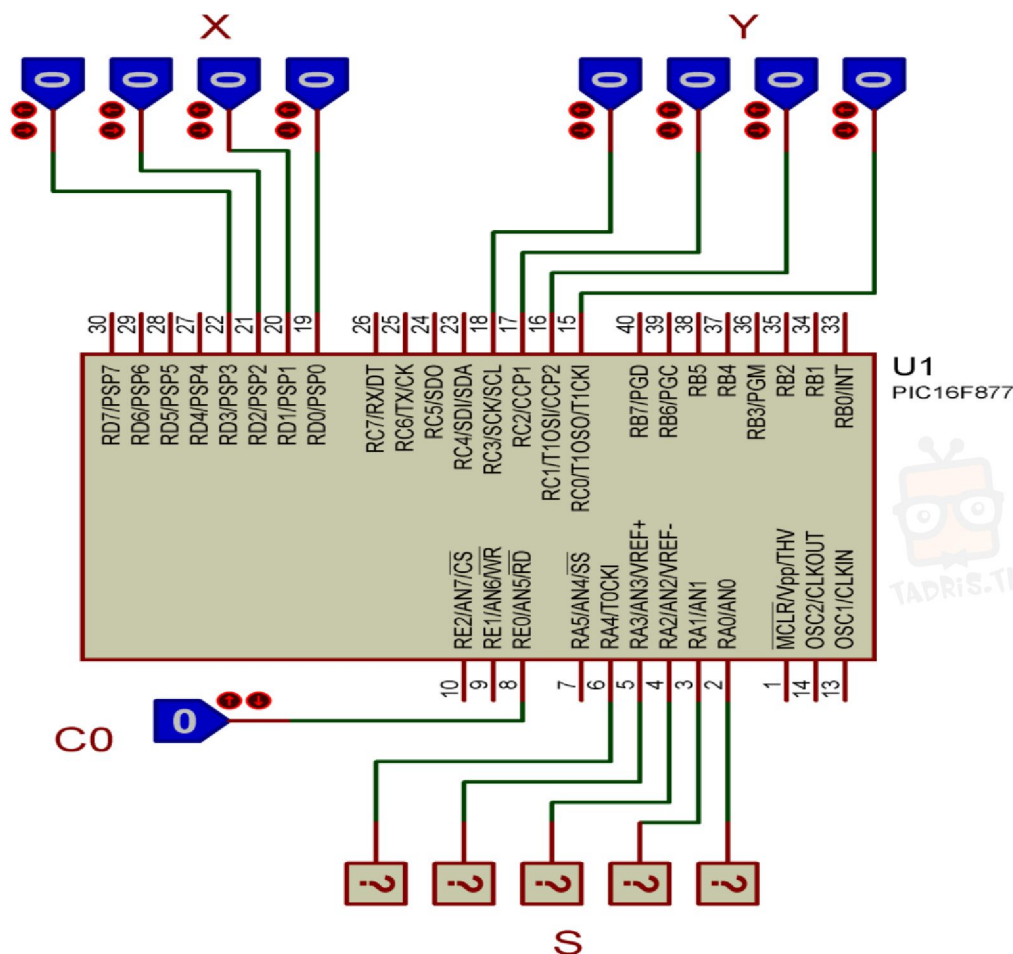
$B (B_3 B_2 B_1 B_0) = \dots\dots\dots$

$S (S_3 S_2 S_1 S_0) = \dots\dots\dots$  et  $C_3 = \dots\dots$  avec  $C_0 = 0$

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



### Exercice 4 :



Compléter le programme en mikroC

#### 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

PortA 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 ..., ..., ....., ..... ;

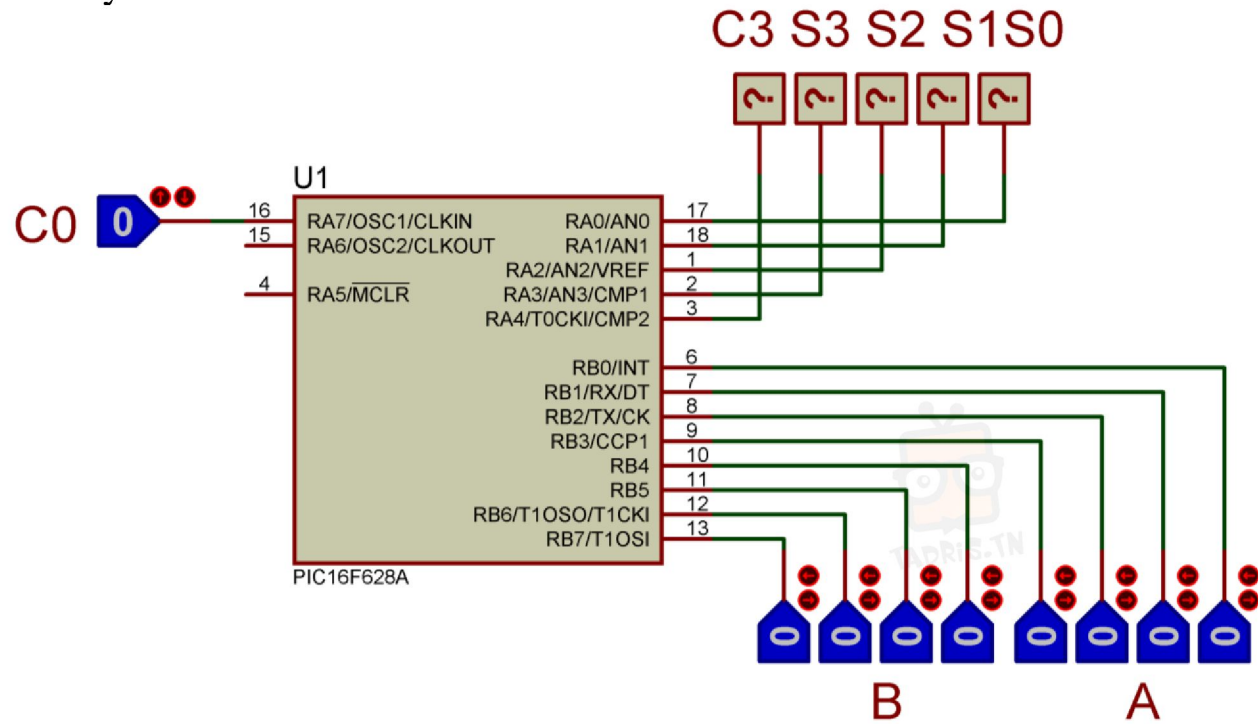
.....()
{
  trisD = ob.....;
  trisC = ob.....;
  trisA = ob.....;
  trisE = ob.....;

INTCON = 0x06 ;
  ..... = .....;
  ..... (1)
  {
    X = .....;
    Y = .....;
    Co = .....;
    S = .....;
    PortA = .....;
    .....
    .....
  }
```



### 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



### Algorithme

Déclarer 4 variables du type octet : char ..., ..., ....., ..... ;  
 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  
 Désactivé 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

```

.....()
{
  trisB = ob.....;
  trisA = ob.....;
  CMCON=ox07;
  .....=.....;

  ..... (1)
  {
    A = PortB;
    A=A & ob.....;
    B = .....;
    B = B .....;
    Co =
    .....;
    S = .....;
    PortA = .....;
    .....
    .....
  }

```

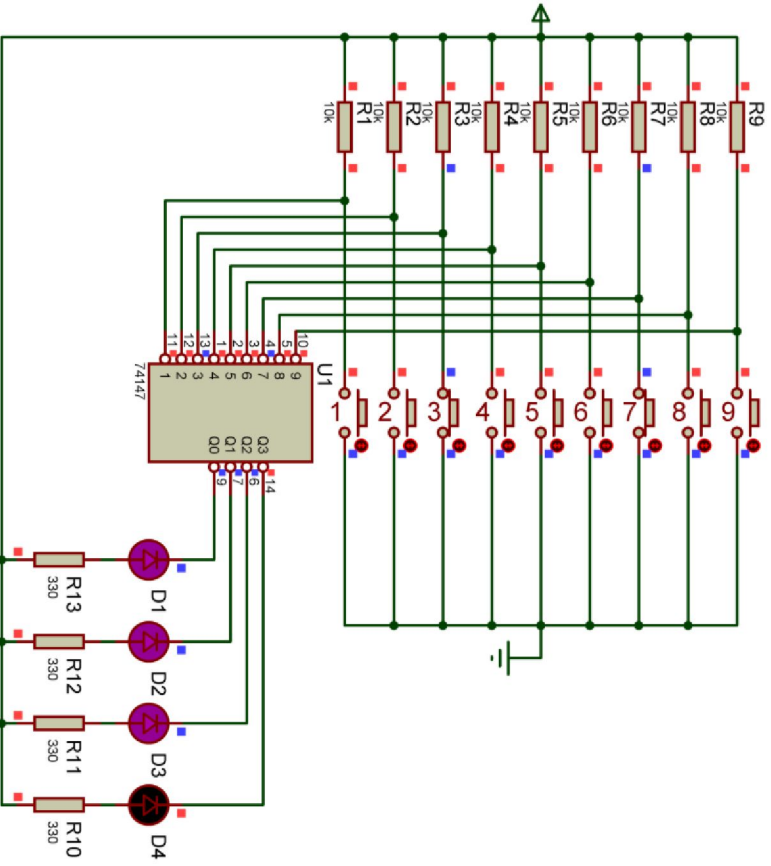






**Exercice 6 : Fonction codage décimal/BCD**

- 1- Donner les noms des entrées et des sorties du circuit 74147 ?.....
- 2- Compléter le tableau en se référant au montage



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

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

