TP N°3

Circuits Arithmétiques : Demi-Additionneur et Additionneur complet

1. But

L'objectif de ce TP est d'analyser et de concevoir un circuit arithmétique basé sur des portes logiques, en prenant l'additionneur comme cas d'étude. L'étudiant devra :

☐Comprendre la différence entre un demi-additionneur et un additionneur complet.

☐Établir leurs tables de vérité, exprimer leurs fonctions logiques, les simplifier, les réaliser, puis comparer les résultats obtenus avec un additionneur intégré.

2. Matériels utilisés

- Plaque d'essai,
- Fils de connexion,
- Circuits intégrés (7404, 7408, 7432, 7486 et 7483).

3. Additionneur numérique

Les opérations arithmétiques (+, −, ×, /) en électronique numérique, comme dans une calculatrice ou un microprocesseur, sont effectuées en système binaire. Elles impliquent des opérations entre deux bits ('0' et/ou '1').

3.1. Demi-additionneur

Prenons l'exemple de l'addition : cette opération binaire présente quatre cas possibles, que l'on peut organiser sous forme d'une table de vérité comme suit :

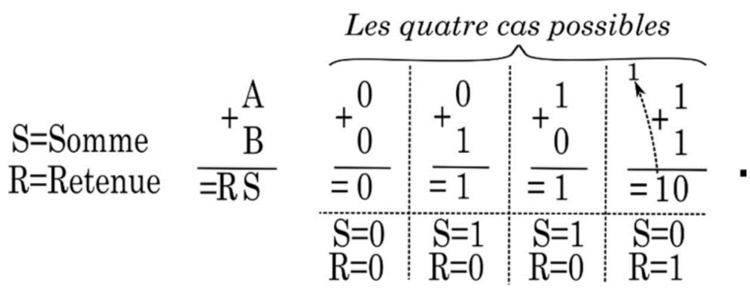


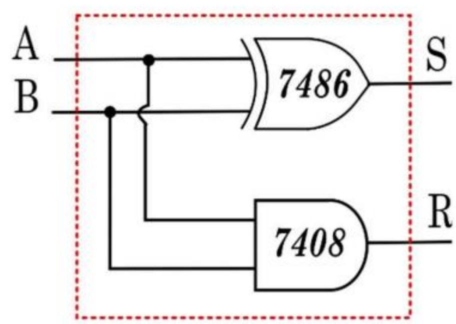
Table de vérité

A	B	S	R	
0	0	0	0	
0	1	1	0	$\overline{A}B$
1	0	1	0	$A\overline{B}$
1	1	0	1	AB

- On se basant sur la table de vérité de demi-additionneur on **déduit** les équations logiques de deux variables de sortie S et R puis on **trace** leur logigramme.

$$\begin{cases} \mathbf{S} = \overline{\mathbf{A}}\mathbf{B} + \mathbf{A}\overline{\mathbf{B}} = \mathbf{A}\oplus\mathbf{B} \\ \mathbf{R} = \mathbf{A.B} \end{cases}$$

Leur logigramme

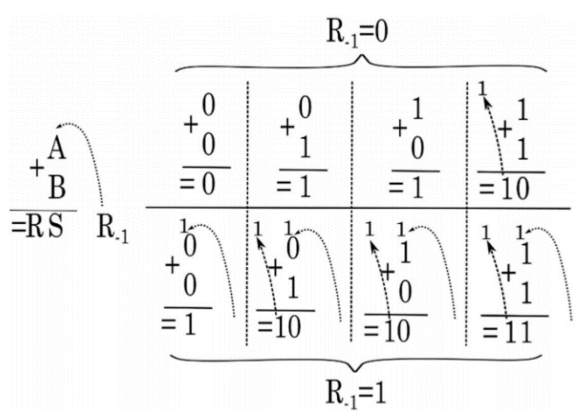


Demi-additionneur

Le circuit à l'intérieur du rectangle pointu s'appelle **un demi-additionneur**, il fait l'addition de deux bits sans tenir en compte la retenue précédente.

3.2. ***Additionneur complet***

- **Un additionneur complet** fait l'addition de trois bits :
- Deux bits (A et B) et le troisième bit qui est la retenue de l'étage précédent R_{-1}



4. Travail à préparer

- On se basant sur la table de vérité de l'additionneur complet, on **déduire** les équations logiques de deux variables de sortie S et R puis tracer leur logigramme.

$$\begin{cases} \mathbf{S} = \cdots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots .. \\ \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots .. \\ \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots .. \\ \mathbf{R} = \cdots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \\ \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \\ \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots . \end{cases}$$

A	B	R_{-1}	S	R
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

- Leur logigramme

Additionneur complet

5. Travail à réalisé

Partie 01 : Additionneur à base des portes logiques

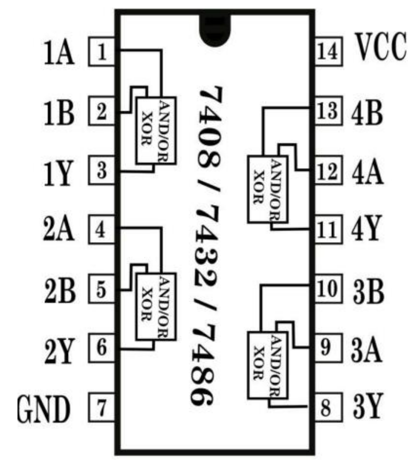
- Réaliser à l'aide **des portes logiques** le circuit d'un **demi-additionneur** et d'un **additionneur complet** et remplir leurs tableaux de mesure suivantes :

Tableau de mesure d'un demi-additionneur

A	B	R	S
0	0		
0	1		
1	0		
1	1		

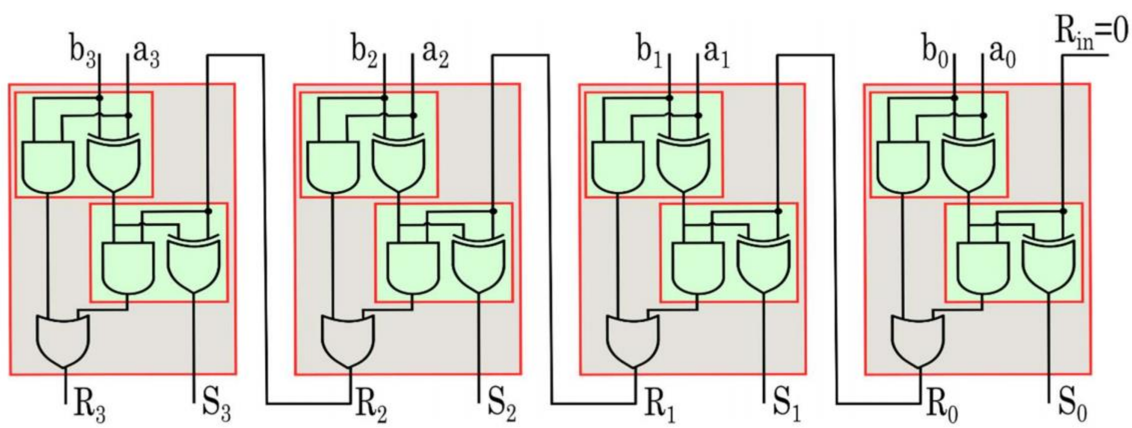
Tableau de mesure d'un additionneur complet

A	B	R_{-1}	R	S
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		



Partie 02 : Additionneur à circuit intégré :

En réalité, les nombres binaires sont composés de plus d'un bit, à cet effet, pour faire la somme de deux nombres (A=a3a2a1a0 et B=b3b2b1b0) de 4 bits chacun, il nous faut 4 additionneurs complets, qui seront cablés de la manière suivante :



Le schéma de l'additionneur de deux nombres à 4 bits, décrit ci-dessus représente le schéma interne de l'additionneur à 4 bits à circuit intégré 7483, dont son brochage et son câblage est le suivant :

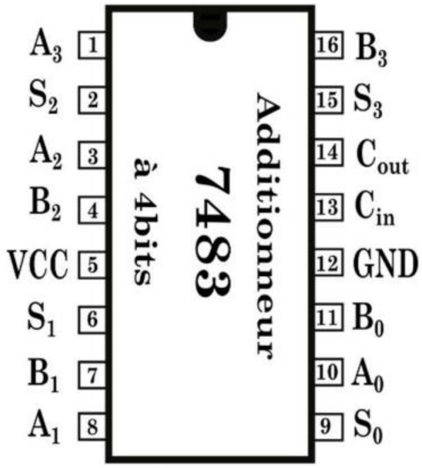


Schéma de brochage

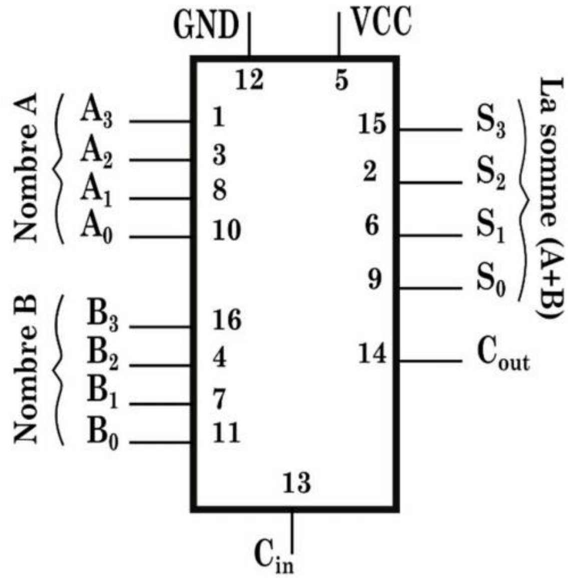


Schéma de câblage

Selon la valeur d'une variable de sélection C_{in} , on voudrait utiliser l'additionneur 7483 comme **additionneur-soustracteur** . Pour **C_{in} =1**, il s'agit d'une soustraction (A-B), pour **C_{in}= 0** c'est une addition.

- Utiliser ce circuit (7483) pour réaliser un montage qui vous permettra de calculer la somme de deux nombres A et B (à 4 bits) donnés dans le tableau suivant :

A				B				S				
A_3	A_2	A_1	A_0	B_3	B_2	B_1	B_0	C_{out}	S3	S2	S1	S0
0	0	1	0	0	0	1	1					
0	1	0	0	0	1	1	0					
1	0	1	0	0	1	1	0					
1	0	1	0	1	0	1	0					
1	1	1	1	1	1	1	1					