

1. Introduction

- Dans un circuit combinatoire l'état des sorties est uniquement fonction de l'état des variables d'entrée. Cet état reste stable tant que l'état des variables d'entrée n'est pas modifié.



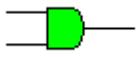



La sortie d'un circuit combinatoire est une fonction booléenne de ses entrées

Objectif de TP

- ✓ Apprendre les circuits combinatoires arithmétiques
- ✓ Apprendre la structure de quelques circuits combinatoires souvent utilisés.
- ✓ Apprendre comment utiliser des circuits combinatoires pour concevoir d'autres circuits plus complexes
- ✓ réaliser des opérations arithmétiques de base (addition, comparaison, soustraction,...), et cela avec l'implantation dans le digital traîné

Et pour ce TP on utilise :

- Logique simulateur Crocodile Physic
- ✓ Symboles du logiciel :

<i>Port</i>	<i>Symboles</i>
<i>Entrées logique</i>	
<i>Sorties logique</i>	
<i>AND</i>	
<i>XOR</i>	
<i>OR</i>	
<i>NOT</i>	

II. Manipulation

1. Compareurs

Le comparateur est un circuit arithmétique permettant de comparer deux nombres binaires A et B. A et B doivent avoir la même longueur (nombre de bits). On cherche à savoir Si $A > B$, $A < B$ ou $A = B$. On comprend donc que le circuit répond à une question à trois choix.

⊙ Réalisation d'un comparateur de 2 nombres de 1 bit (a, b) :

Soit 2 entrées a et b, et 3 sorties $\{S_0, S_1, S_2\}$.

$$: \text{ Avec : } \begin{cases} S_0 = 1, & \text{Si } a = b \\ S_1 = 1, & \text{Si } a > b \\ S_2 = 1, & \text{Si } a < b \end{cases}$$

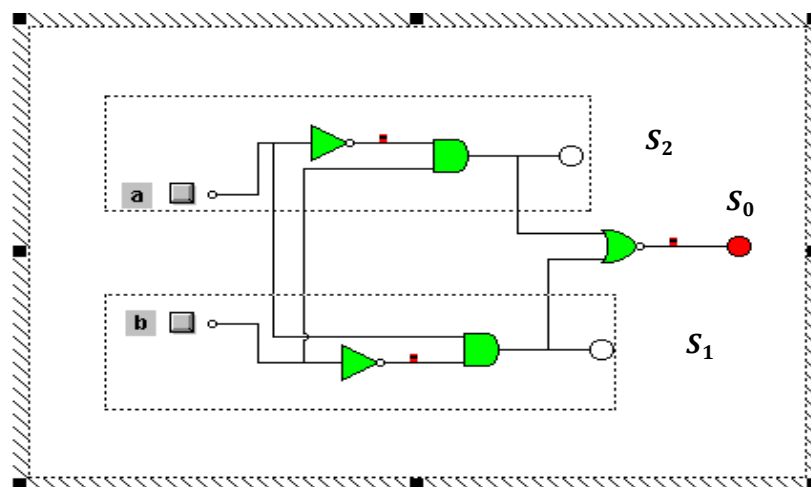
On peut dresser la table de vérité de ce circuit. On a 2 entrées alors est possibilité $\{2^2\}$ des résultats, résumé on table ci-dessous.

a	b	S_0	S_1	S_2
0	0	1	0	0
0	1	0	0	1
1	0	0	1	0
1	1	1	0	0

A partir cette table de vérité on peut déduire les expressions de ces circuits logique :

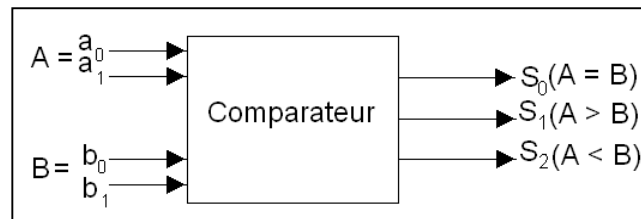
$$\begin{cases} S_0 = \overline{a \oplus b} = \overline{a\bar{b} + \bar{a}b} = \overline{S_1 + S_2} \\ S_1 = a\bar{b} \\ S_2 = \bar{a}b \end{cases}$$

❖ Schéma de circuit comparateur :



Figure_1 Schéma de circuit logique comparateur 2 bits

⊙ Réalisation d'un Comparateur de 2 nombres de 2 bit (A, B) :



1^{ere}. On a quelques conditions en ce circuit

$$\text{Soit } \begin{cases} A = a_0, a_1 \\ B = b_0, b_1 \end{cases} \cdot \text{ Est } \begin{cases} S_0 = 1, & \text{Si } A = B \\ S_1 = 1, & \text{Si } a_1 > b_1 \text{ ou } a_1 = b_1 \text{ et } a_0 > b_0 \\ S_2 = 1, & \text{Si } a_1 < b_1 \text{ ou } a_1 = b_1 \text{ et } a_0 < b_0 \end{cases}$$

❖ On va dresser la table de vérité

A		B		S ₁	S ₂	S ₀
a ₀	a ₁	b ₀	b ₁			
0	0	0	0	0	0	1
0	0	0	1	0	1	0
0	0	1	0	0	1	0
0	0	1	1	0	1	0
0	1	0	0	1	0	0
0	1	0	1	0	0	1
0	1	1	0	1	0	0
0	1	1	1	0	1	0
1	0	0	0	1	0	0
1	0	0	1	0	1	0
1	0	1	0	0	0	1
1	0	1	1	0	1	0
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	0	0	1

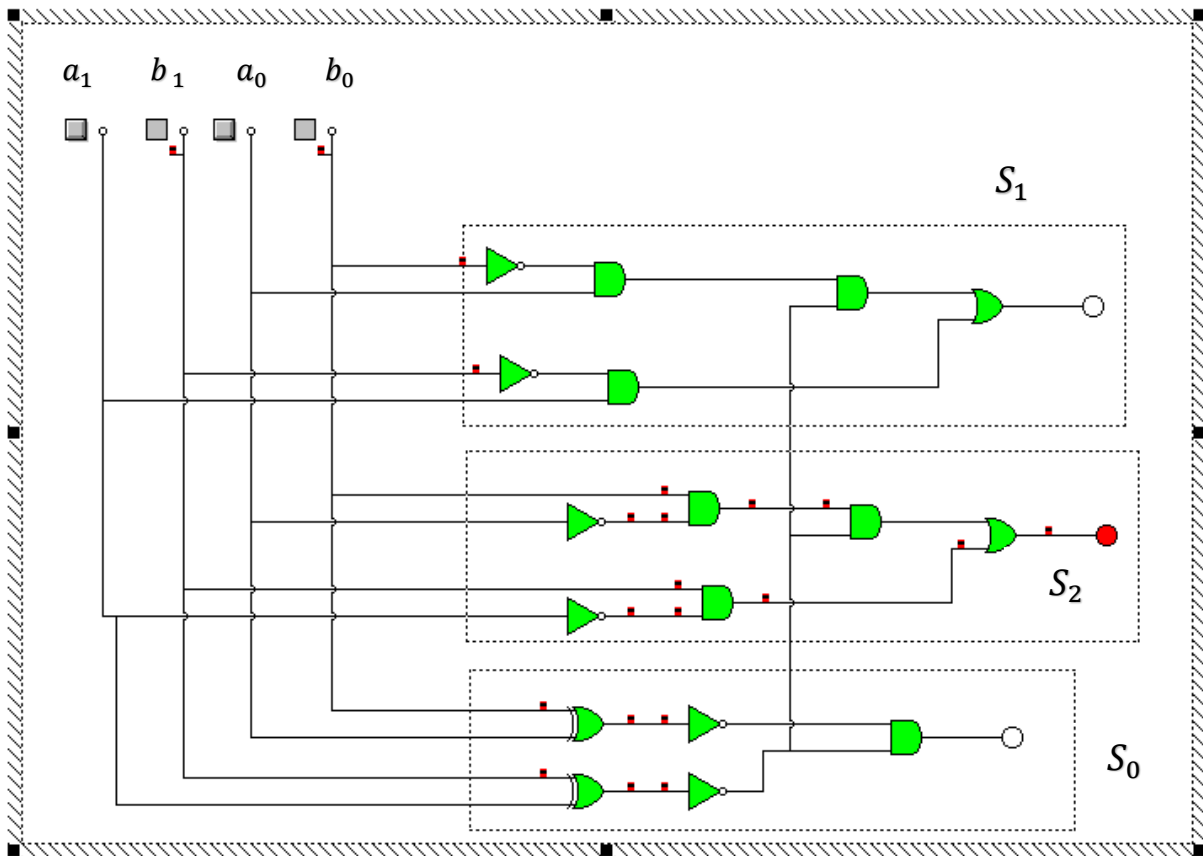
⊙ Donc on peut déduire les expressions logiques

$$S_1 (A > B): \begin{cases} a_1 > b_1 \\ \text{ou} \\ a_0 > b_0 \text{ et } a_1 = b_1 \end{cases} \quad \text{danc soit la fonction : } S_1 = a_1 \bar{b}_1 + (\overline{a_1 \oplus b_1})(\bar{b}_0 a_0)$$

$$S_2 (A < B): \begin{cases} a_1 < b_1 \\ \text{ou} \\ a_0 < b_0 \text{ et } a_1 = b_1 \end{cases} \quad \text{On, a la fonction : } S_2 = b_1 \bar{a}_1 + (\overline{a_1 \oplus b_1})(\bar{a}_0 b_0)$$

$$S_0 (A = B): \begin{cases} a_1 = b_1 \\ \text{et} \\ a_0 = b_0 \end{cases} \quad \text{On a soit la fonction : } S_0 = (\overline{a_1 \oplus b_1})(\overline{a_0 \oplus b_0})$$

⊙ Alors le schéma logigramme de ce circuit sera en la figure 2 ci-dessous :



.Figure_2 Schéma de circuit logique comparateur 2 mots 2 bits

2. Additionner :

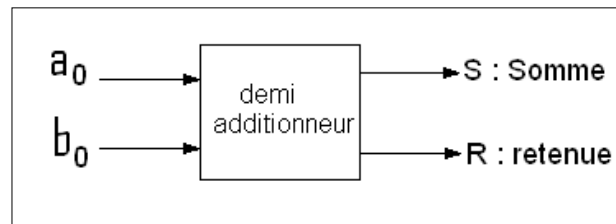
Un additionneur est un circuit capable de faire l'addition de deux nombre de n bits. Une addition génère deux résultats : la somme et la retenue

Commençons par demi-additionner.

⊙ Demi-additionner

Ce circuit, qui permettrait d'effectuer l'addition des deux bits de plus bas poids est appelé demi-additionneur.

⊙ Réalisation d'un Demi additionneur de 2 nombres de 1 bit (a, b)



✓ Ecrivons la table de vérité de celui-ci :

a_0	b_0	S	R
0	0	0	0
0	1	1	0
1	0	1	0
1	1	1	1

✓ On déduire alors les expressions du {S, R}

$$\begin{cases} S = \overline{a_0} b_0 + a_0 \overline{b_0} = a_0 \oplus b_0 \\ R = a_0 b_0 \end{cases}$$

Ce qui peut être réalisé par le circuit schématisé sur le logigramme de la figure_3 celui-ci :

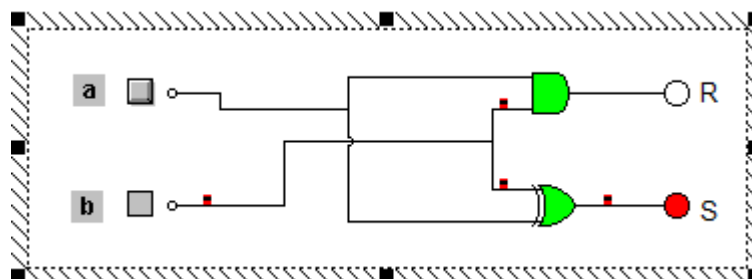
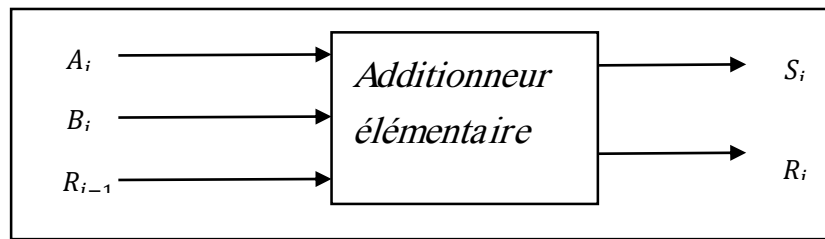


Figure _3. Schéma de circuit semi-additionner

- ⊙ Réalisation d'un additionneur de 2 bits de même poids en tenant compte de la retenue précédente.



- ✓ Ecrivons la table de vérité de celui-ci :

A_i	B_i	R_{i-1}	S_i	R_i
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Il y a deux méthodes pour obtenir les équations de ce circuit

- ✓ Méthode I
 ✓ Analytique :

Alors l'expression de la somme S_i est :

$$\begin{aligned}
 S_i &= \overline{A_i} \cdot \overline{B_i} \cdot R_{i-1} + \overline{A_i} \cdot B_i \cdot \overline{R_{i-1}} + A_i \cdot \overline{B_i} \cdot \overline{R_{i-1}} + A_i \cdot B_i \cdot R_{i-1} \\
 &= \overline{A_i} \cdot (B_i \cdot \overline{R_{i-1}} + \overline{B_i} \cdot R_{i-1}) + A_i \cdot (\overline{B_i} \cdot \overline{R_{i-1}} + B_i \cdot R_{i-1}) \\
 &= \overline{A_i} \cdot (B_i \oplus R_{i-1}) + A_i \cdot (\overline{B_i} \oplus \overline{R_{i-1}}) = A_i \oplus B_i \oplus R_{i-1}
 \end{aligned}$$

Et l'expression de la retenue R_i :

$$\begin{aligned}
 R_i &= \overline{A_i} \cdot B_i \cdot R_{i-1} + A_i \cdot \overline{B_i} \cdot R_{i-1} + A_i \cdot B_i \cdot \overline{R_{i-1}} + A_i \cdot B_i \cdot R_{i-1} \\
 &= (\overline{A_i} \cdot B_i + A_i \cdot \overline{B_i}) R_{i-1} + A_i \cdot B_i = (A_i \oplus B_i) R_{i-1} + A_i \cdot B_i
 \end{aligned}$$

✓ Méthode II

✓ Tables de Karnaugh :

Pour S_i :

$R_{i-1} \backslash A_i B_i$	00	01	11	10
0	0	1	0	1
1	1	0	1	0

Pour R_i :

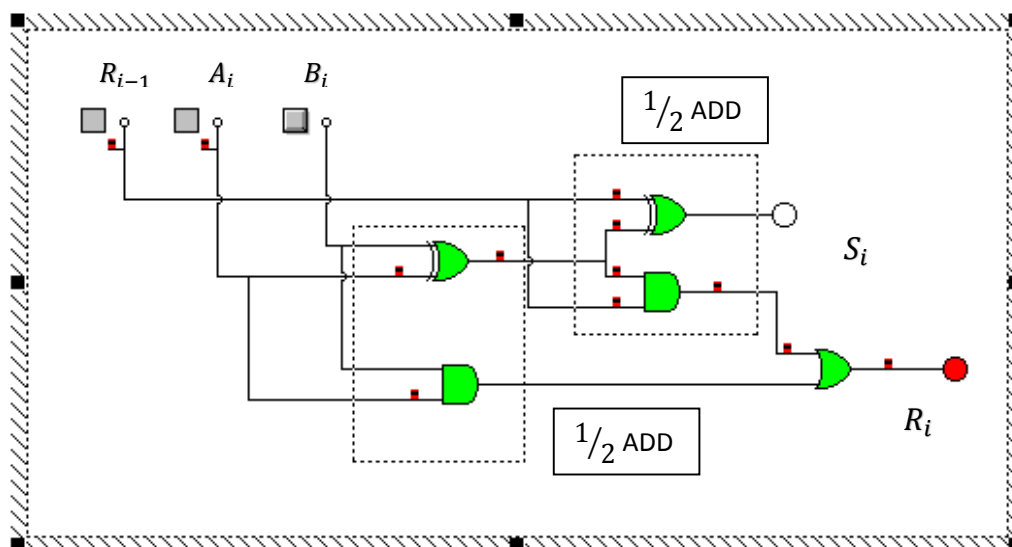
$R_{i-1} \backslash A_i B_i$	00	01	11	10
0	0	0	1	0
1	0	1	1	1

Alors on a directement après simplification les fonctions suivant :

$$S_i = A_i \oplus B_i \oplus R_{i-1} \quad . \quad R_i = a_i b_i + R_{i-1}(a_i \oplus b_i)$$

✓ logigramme

Maintenu peut réaliser logigramme de circuit additionneur-élémentaire en la figure_4 :

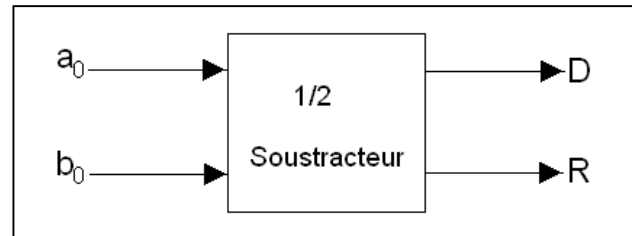


Figure_4 .schéma de circuit additionné élémentaire

3. Le Soustracteur

Il n'y a pas de circuit soustracteur dans un processeur parce que l'on peut implémenter la soustraction à l'aide de l'additionneur avec des modifications mineures. Pour ce faire, on exploite les propriétés du complément à 2 et le fait que le bit de poids faible de l'additionneur n'a pas de retenue d'entrée

⊙ Réalisation d'un Demi-soustracteur de deux bits :



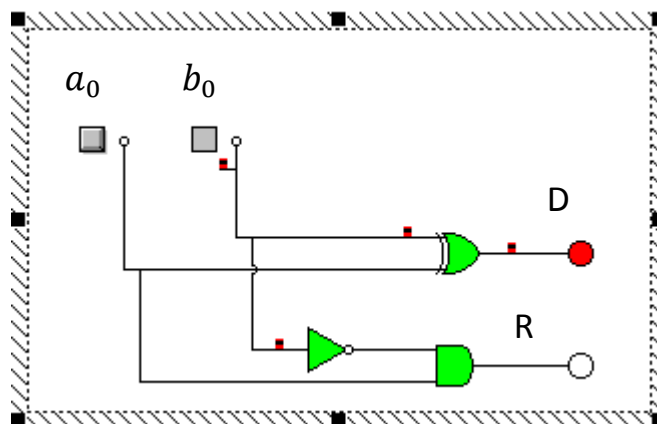
✓ Ecrivons la table de vérité de celui-ci :

b_0	a_0	D	R
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

✓ Alors on peut facilement déduire l'équation de ce circuit :

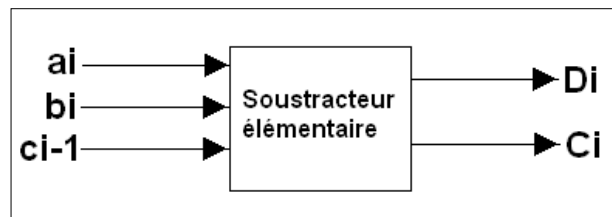
$$\begin{cases} D = \overline{a_0} \cdot b_0 + a_0 \overline{b_0} = a_0 \oplus b_0 \\ R = a_0 \overline{b_0} \end{cases}$$

Ce qui peut être réalisé par le circuit schématisé sur le logigramme de figure_5 ci-dessous :



Figure_5.schéma de circuit 1/2 soustracteur

- ⊙ Réalisation d'un soustracteur de deux mots de même poids en tenant compte de la retenue précédente :



✓ Table de vérité

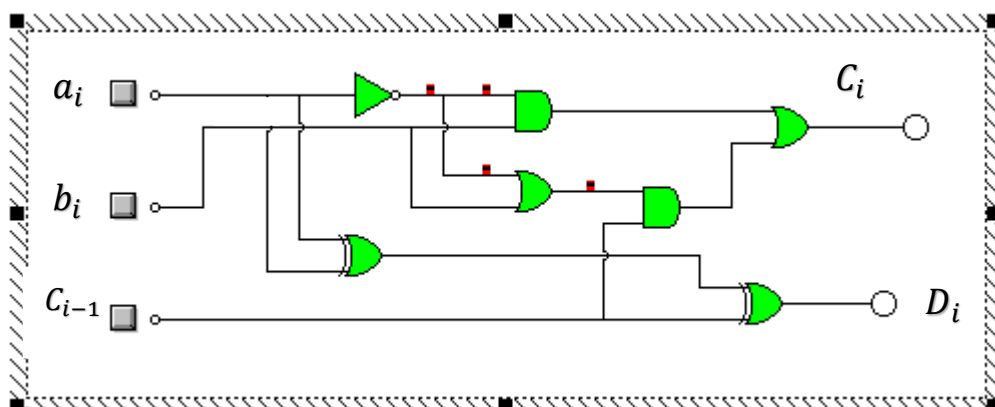
1^{ère} Ecrivons la table de vérité de celui-ci :

C_{i-1}	b_i	a_i	D_i	C_i
0	0	0	0	0
0	0	1	1	0
0	1	0	1	1
0	1	1	0	0
1	0	0	1	1
1	0	1	0	0
1	1	0	0	1
1	1	1	1	1

✓ Les équations :

$$\begin{cases} D_i = (a_i \oplus b_i) \oplus C_{i-1} \\ C_i = \bar{a}_i \cdot b_i + C_{i-1} (\bar{a}_i + b_i) \end{cases}$$

✓ le logigramme de circuit dans la figure_6 :



Figure_6 schéma de circuit soustracteur élémentaire

4. Le circuit V.A :

✓ *Table de vérité* : soit les entrées $\langle A, B, C \rangle$ et les sorties $\langle L1, L2 \rangle$

Comme nous avons vu et essayé dans laboratoire, nous obtenu le table suivant :

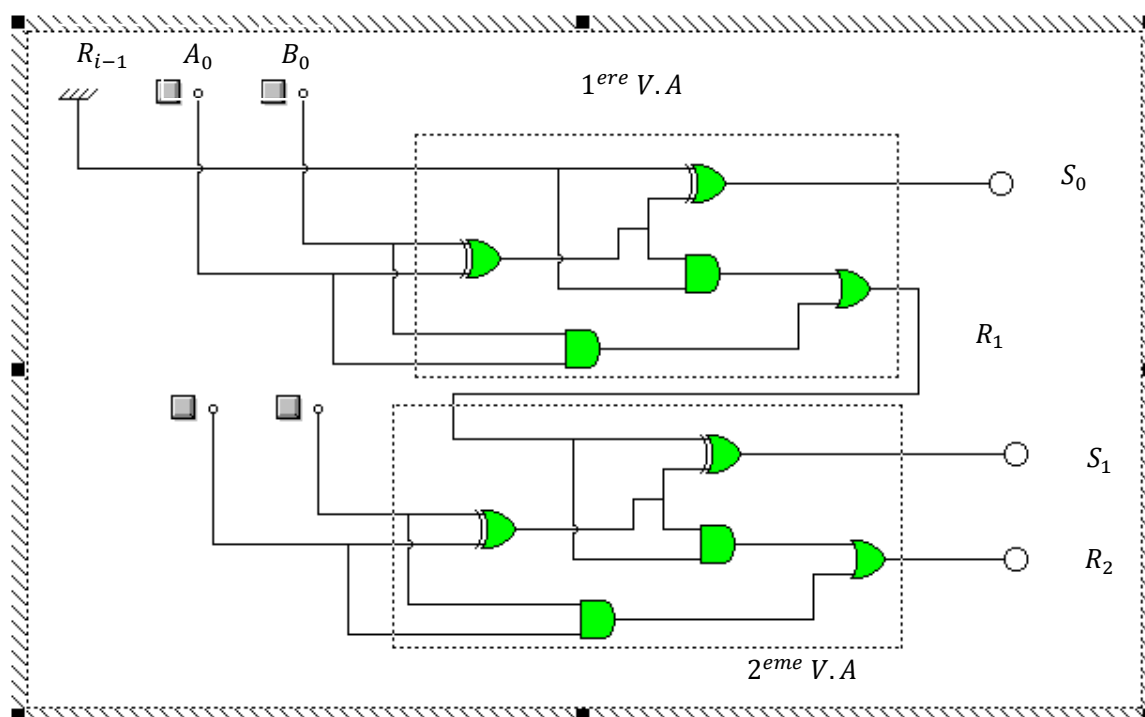
A	B	C	L_1	L_2
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

On remarque que est le même table de circuit additionner-élémentaire.

Alors le rôle de ce circuit V.A est le même de circuit additionner-élémentaire.

✓ Réaliser un circuit capable de faire la somme de 2 mots de pois 2 bits avec 2 circuits V.A :

Soit 2 mots $\langle A \text{ et } B \rangle$ de pois 2 bits. $\begin{Bmatrix} a_0 & a_1 \\ b_0 & b_1 \end{Bmatrix}$. Et en utilisons 2 circuit logique V.A pour ça .comme le schéma dans la figure suivant :



Figure_7.schéma de 2 circuits logique V.A

5. Travail théorique

✓ Additionner de 2 mots avec 3 bits :

Un additionneur sur 3 bits est un circuit qui permet de faire l'addition de deux nombres A et B de 3 bits chacun :

$$\begin{cases} A = A_0A_1A_2 \\ B = B_0B_1B_2 \end{cases}$$

Et bien sûr en plus il tient en compte de la retenue entrante.

Alors on a ici 6 entrées plus retenue donc on a 7 entrées en total.

Et en sorties on va avoir le résultat sur 3 bits ainsi que la retenue '4 bits de sorties'

Alors on a un circuit de 7 entrées et 4 sorties.

Pour obtenir l'équation de ce circuit on peut utiliser autre facile méthode (pas table de vérité par ce que on a $2^7 = 128$!!!)

Il faut trouver une solution plus facile et plus efficace pour concevoir ce circuit.

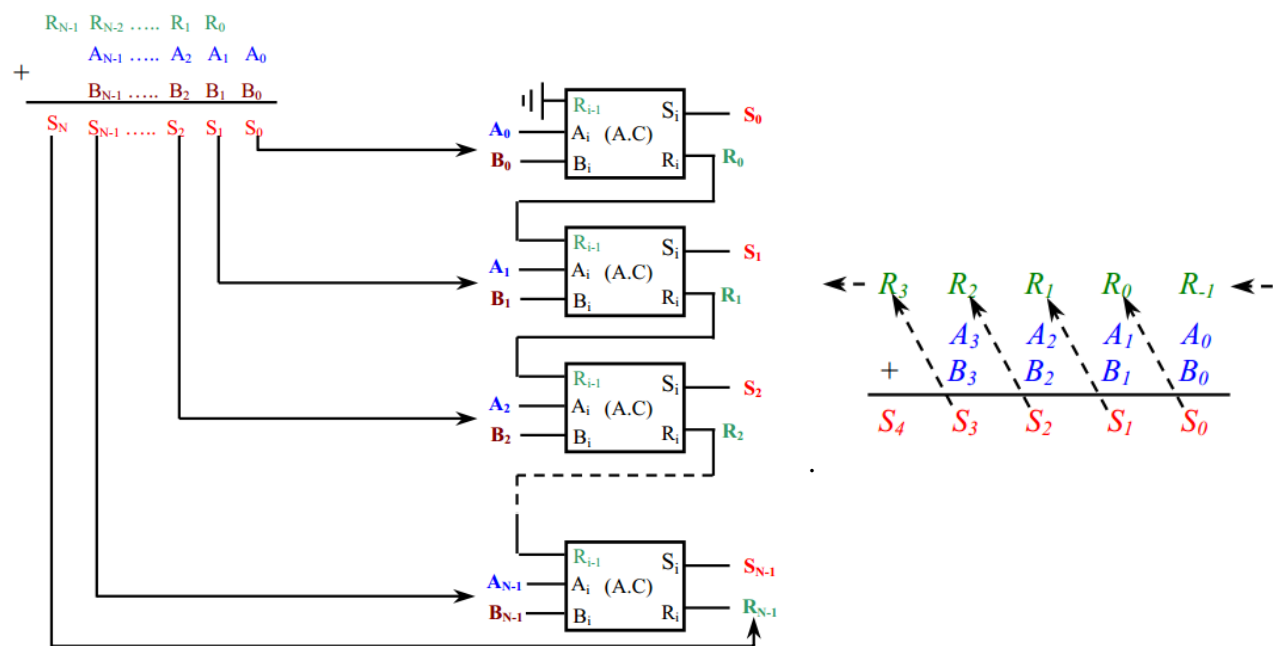
Solution

Lorsque on fait l'addition en binaire, on additionne bit par bit en commençant à partir du poids faible (LSB) et à chaque fois on propage la retenue sortante au bit du rang supérieur

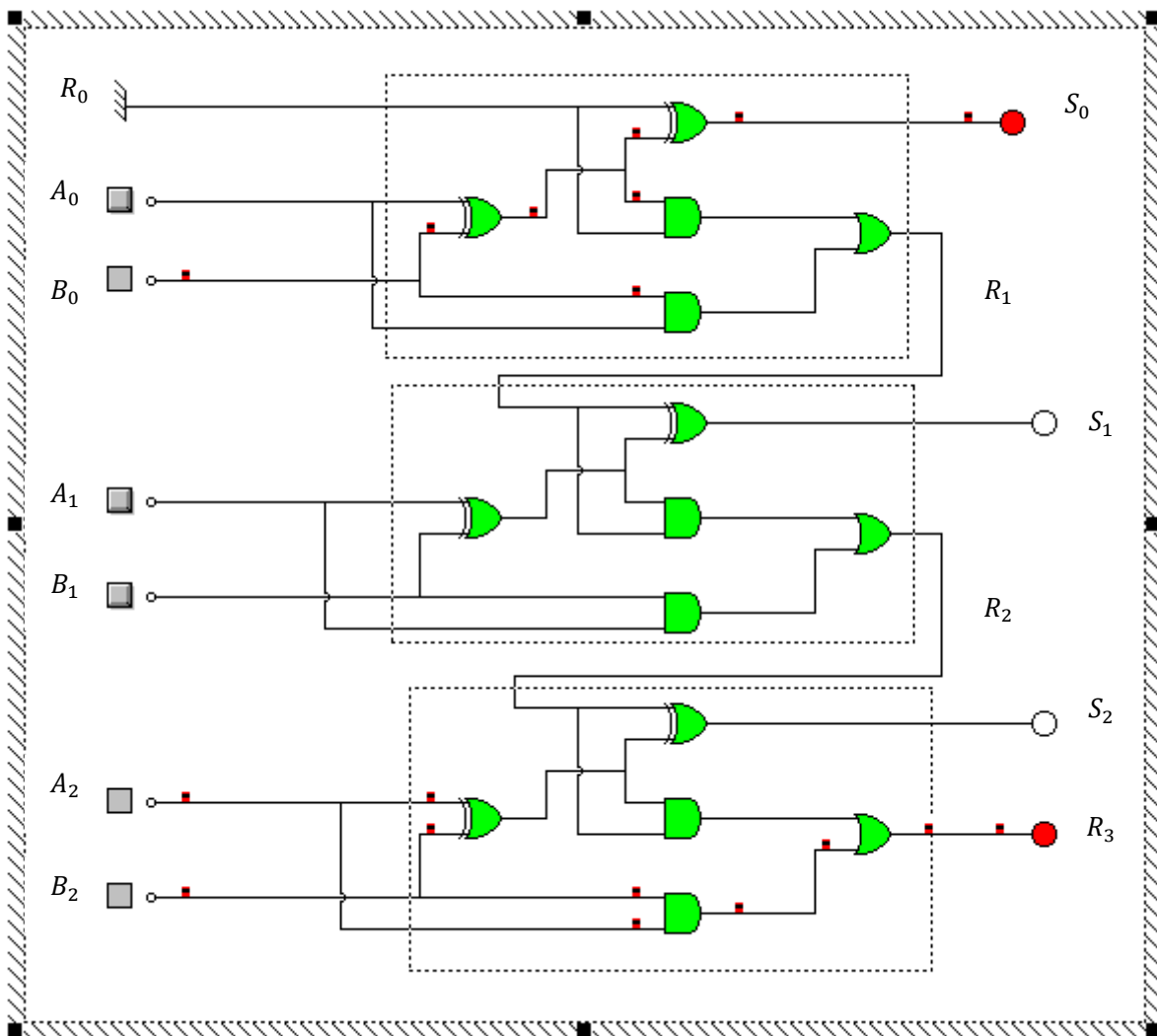
L'addition sur un bit peut se faire par un additionneur complet sur 1 bit.

Et à partir des Additionneurs Complètes ou bien élémentaire (AC), il Est facile d'effectuer l'ajout de deux de

Nombres binaires à n bits. Comme le Montre le montage de la méthode ci-Dessous



✓ Circuit d'implantation du additionner 2 mots de 3 bits



figure_8

Exemple utilisé :

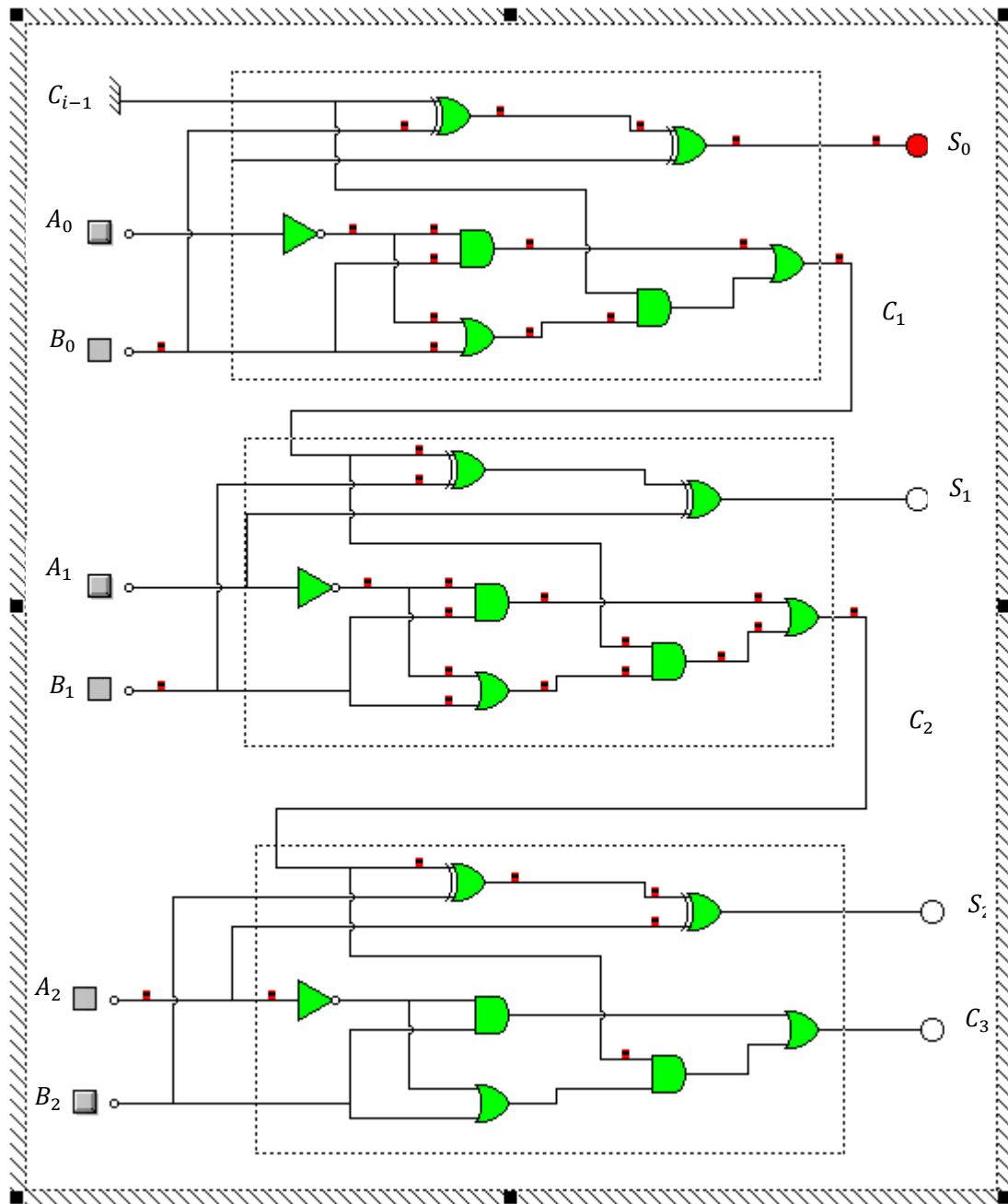
$A = 1\ 0\ 0, B = 1\ 0\ 1$

$S = A + B = 1\ 0\ 0\ 1$ dans ce case $R_3 = S_3$ "bit N°4"

Nous remarquons que il y a 3 additionner ou bien {V.A} dans ce circuit donc en peut obtenir un additionner de 2 mots de 3 bits avec 2 VA circuit.

✓ Soustracteur de 2 mots avec 3 bits :

Dans ce circuit on peut utiliser la même méthode d'additionner et simplement en va obtenir le circuit dans la figure_9 ci-dessous :



Figure_9

Exemple utilisé :

$$A = 100, \quad B = 011, \quad S = A - B. \quad S = 001$$