

Circuits de Calcul Arithmétique

Cours Systèmes Logiques

1^{ère} Année SupNum

Dr. Mohamed Sass Mohamdy

Année universitaire 2024/2025

Plan du Cours

I – Demi-Additionneur

II – Additionneur Complet (à 1 bits)

III - Additionneur parallèle à retenue propagée (à n bits)

IV – Demi-Soustracteur

V – Soustracteur Complet (à 1 bits)

VI - Soustracteur parallèle à propagation de retenu (à n bits)

VII – Circuit Multiplieur

VIII – Exemple d'Unité Arithmétique et Logique

I – Demi-additionneur

- Appelons A, B les deux variables d'entrée représentant les bits à additionner, S la somme et C la retenue (C : appelée "carry" en anglais).
- La table de vérité du demi-additionneur est la suivante :



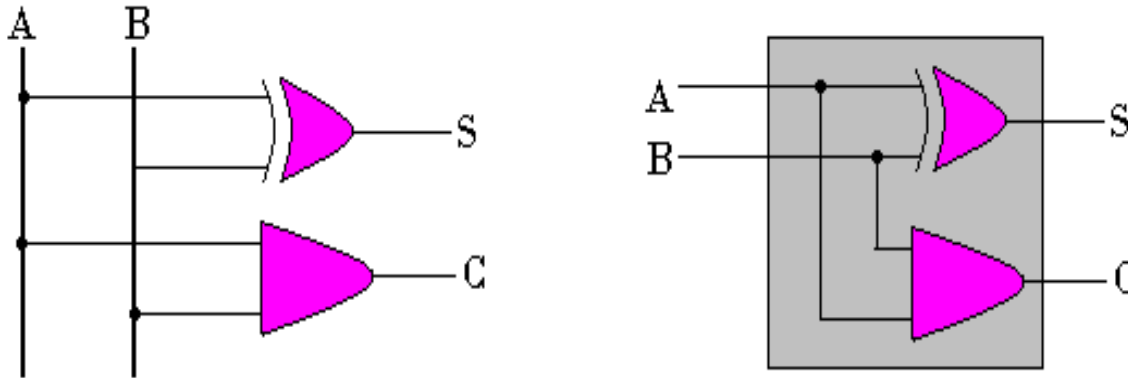
A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

- On déduit alors les expressions logiques de S et de C.

$$S = \overline{A}B + A\overline{B} = A \oplus B$$

$$C = AB$$

- Le circuit logique du demi-additionneur est donné par :



- Le demi-additionneur ne peut faire que l'addition des deux chiffres de plus faible poids, puisqu'il ne peut pas prendre en compte la retenue qui provient d'une addition précédente.
- **Pour résoudre cette difficulté on utilise un circuit à trois entrées : c'est l'additionneur complet (A.C).**

II - Additionneur Complet (A.C)

- C'est un circuit qui fournit la somme et la retenue de deux chiffres codés en binaire pur (en Base 2), et de la retenue du rang précédent.
- **Pour cela l'additionneur complet est un circuit à trois entrées A_n , B_n et C_{n-1} (report précédent) et possédant toujours deux sorties somme S_n et retenue C_n .**
- La table de vérité de l'additionneur complet est donnée par le tableau suivant :

A_n	B_n	C_{n-1}	S_n	C_n
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

- Equations des sorties d'un additionneur complet :**

$$S_n = \overline{A_n} \overline{B_n} C_{n-1} + \overline{A_n} B_n \overline{C_{n-1}} + A_n \overline{B_n} \overline{C_{n-1}} + A_n B_n C_{n-1}$$

$$S_n = \overline{A_n} (\overline{B_n} C_{n-1} + B_n \overline{C_{n-1}}) + A_n (\overline{B_n} \overline{C_{n-1}} + B_n C_{n-1})$$

$$S_n = \overline{A_n} (B_n \oplus C_{n-1}) + A_n (\overline{B_n} \oplus \overline{C_{n-1}})$$

$$S_n = A_n \oplus (B_n \oplus C_{n-1})$$

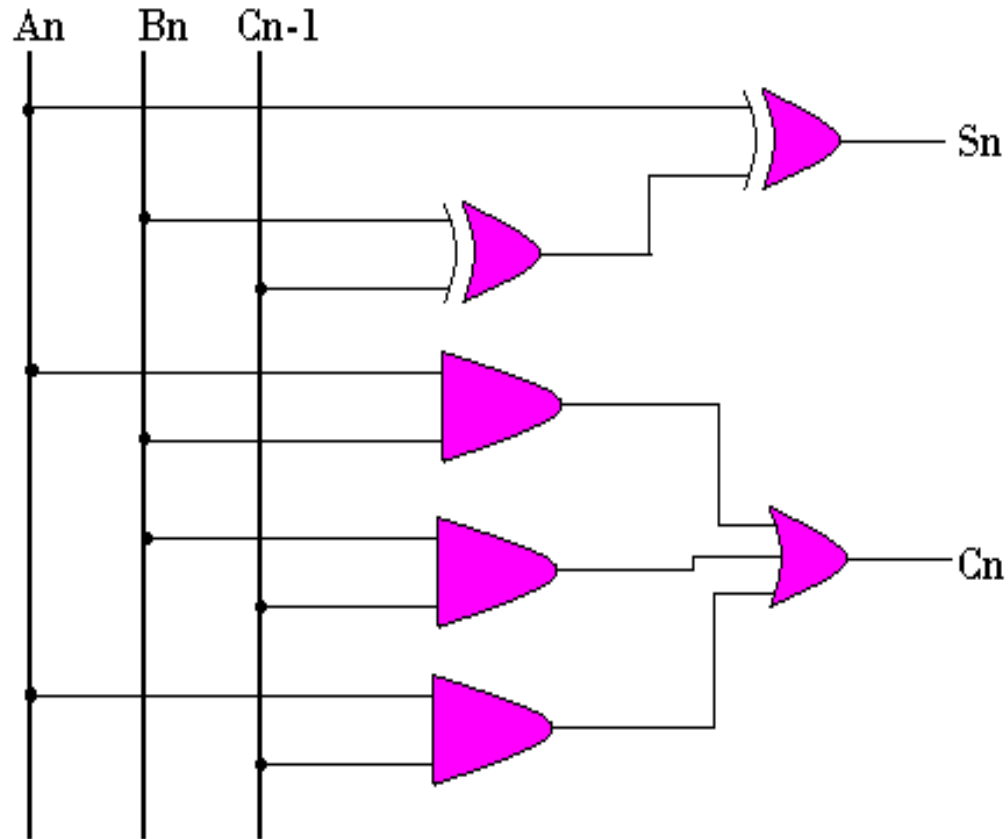
- Simplification de la fonction C_n :**

C_n

	$\overline{C_{n-1}}$	C_{n-1}
$\overline{A_n} \overline{B_n}$	0	0
$\overline{A_n} B_n$	0	1
$A_n \overline{B_n}$	1	1
$A_n B_n$	0	1

$$C_n = A_n B_n + B_n C_{n-1} + A_n C_{n-1}$$

- Le logigramme d'un additionneur complet :



III - Additionneur parallèle à retenue propagée (à n bits)

A partir des additionneurs complets (A.C), il est facile d'effectuer l'addition de deux nombres binaires à N bits.

Cette addition est réalisée par la mise en cascade de N additionneurs complets, comme le montre le montage de la figure ci-dessous, Il s'agit d'un additionneur parallèle, parce qu'on additionne toutes les colonnes en même temps.

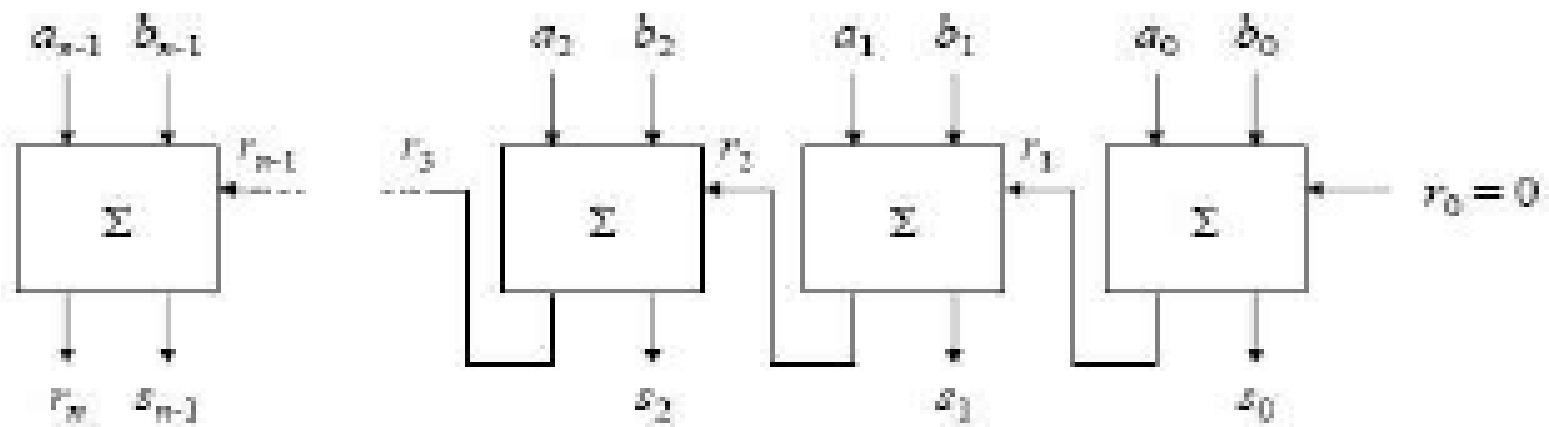
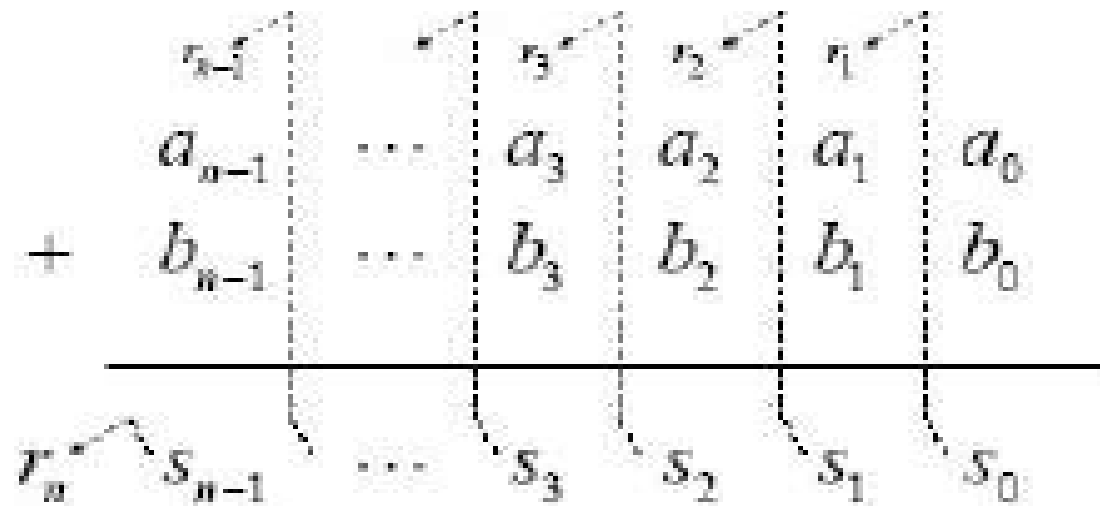


Fig. Additionneur parallèle de deux nombres binaires à N bits

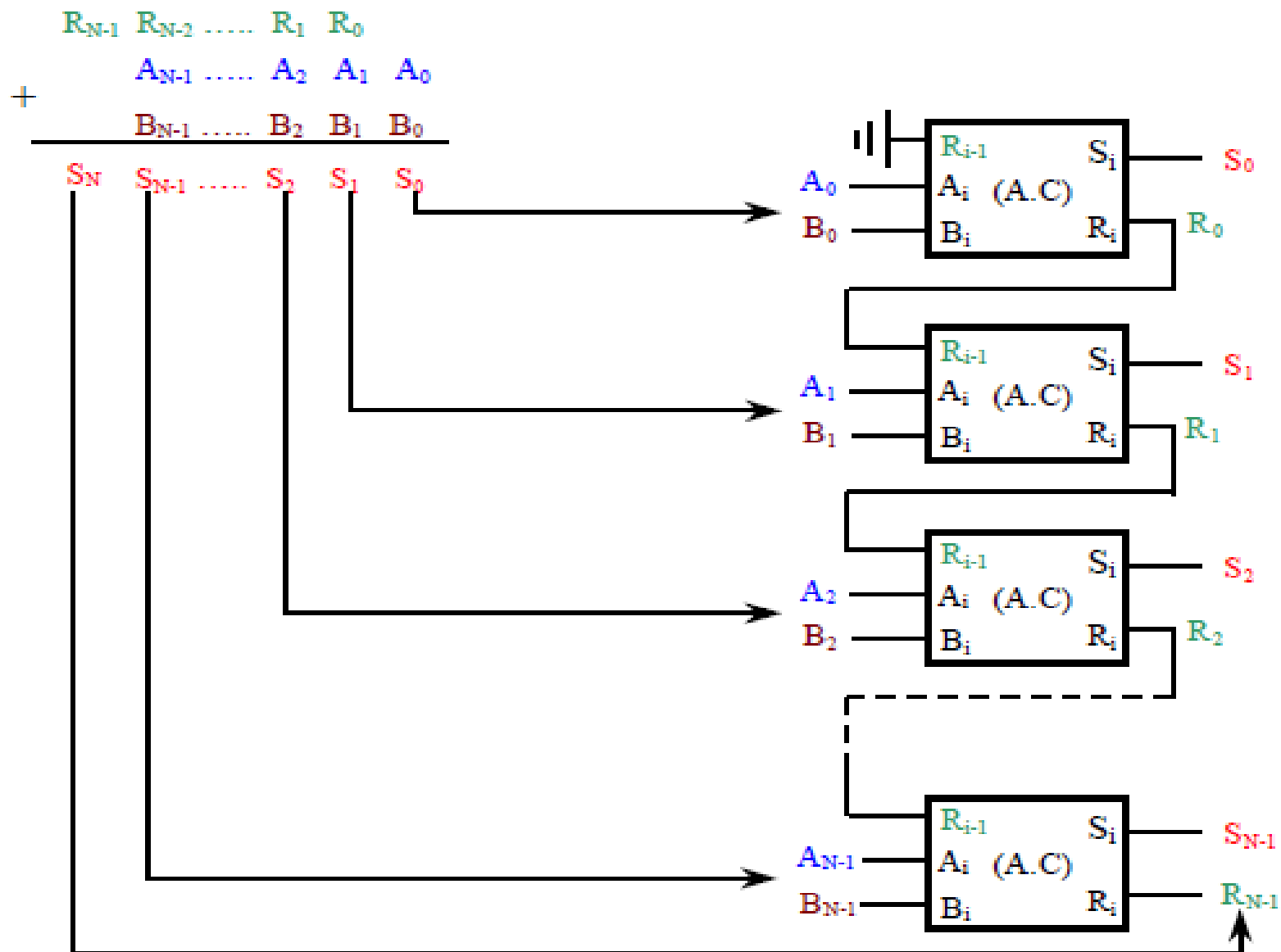


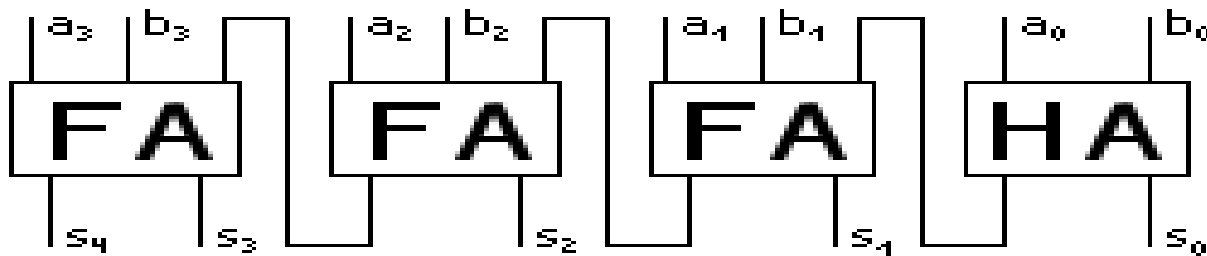
Fig. Additionneur parallèle de deux nombres binaires à N bits

▪ Additionneur parallèle de deux nombres binaires à 4 bits :

L'additionneur suivant, additionne les deux chiffres codés en binaire :

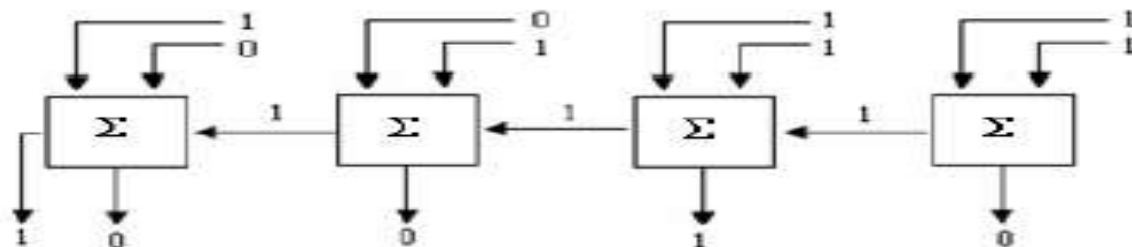
$a_3a_2a_1a_0$ et $b_3b_2b_1b_0$:

Le résultat de la somme est : $s_4s_3s_2s_1s_0$



Exemple :

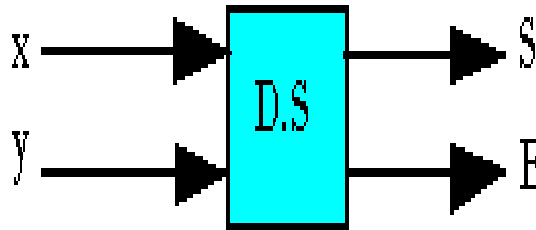
$$\begin{array}{r}
 A \quad + \quad 1 \quad 0 \quad 1 \quad 1 \\
 B \quad \quad 0 \quad 1 \quad 1 \quad 1 \\
 \hline
 1 \quad 0 \quad 0 \quad 1 \quad 0
 \end{array}$$



IV - Demi-Soustracteur

Un demi-soustracteur est un circuit capable de faire la soustraction de deux nombre binaires d'un bit chacun. Le circuit aura deux entrées x , y et deux sorties S et B .

S : Sortie du bit de soustraction, B : Retenue (Borrow).



Ainsi, la table de vérité de cette fonction logique $(x - y)$ est la suivante :

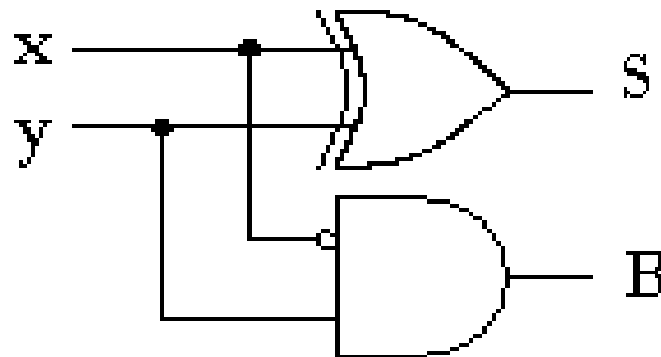
X	Y	S	B
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

- Et, par Conséquent, Les équations de sortie, sont :

$$S = \overline{x}y + x\overline{y} = x \oplus y$$

$$B = \overline{x}y$$

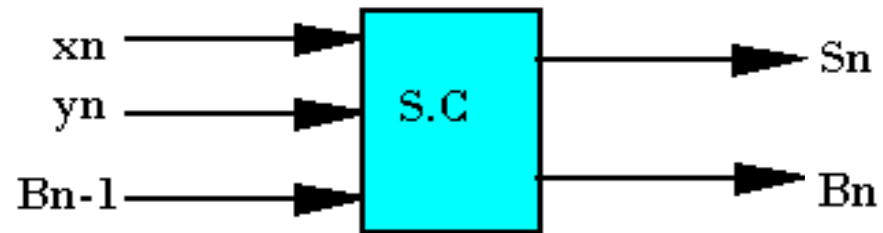
- Le circuit logique du demi-soustracteur est donné par :**



V - Soustracteur Complet à un bit

Un soustracteur complet est un circuit capable de faire la soustraction de deux bits de rang n , $(x_n - y_n)$ tout en tenant compte de la retenue B_{n-1} provenant de la soustraction des bits de rang directement inférieur.

On aura deux sorties S_n et B_n .



x_n	y_n	B_{n-1}	S_n	B_n
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

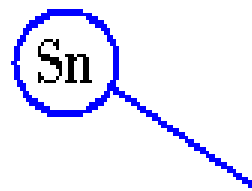
Table de vérité d'un soustracteur complet

- Equations de sortie :

$$S_n = \overline{x_n}\overline{y_n}B_{n-1} + \overline{x_n}y_n\overline{B_{n-1}} + x_n\overline{y_n}\overline{B_{n-1}} + x_ny_nB_{n-1}$$

$$B_n = \overline{x_n}\overline{y_n}B_{n-1} + \overline{x_n}y_n\overline{B_{n-1}} + x_n\overline{y_n}B_{n-1} + x_ny_nB_{n-1}$$

- Simplification de la fonction S_n :



	$\overline{x_n}\overline{y_n}$	$\overline{x_n}y_n$	$x_n\overline{y_n}$	x_ny_n
$\overline{B_{n-1}}$	0	1	0	1
B_{n-1}	1	0	1	0

Pas de regroupement

$$S_n = \overline{x_n}\overline{y_n}B_{n-1} + \overline{x_n}y_n\overline{B_{n-1}} + x_n\overline{y_n}\overline{B_{n-1}} + x_ny_nB_{n-1}$$

$$S_n = \overline{x_n}(\overline{y_n}B_{n-1} + y_n\overline{B_{n-1}}) + x_n(\overline{y_n}\overline{B_{n-1}} + y_nB_{n-1})$$

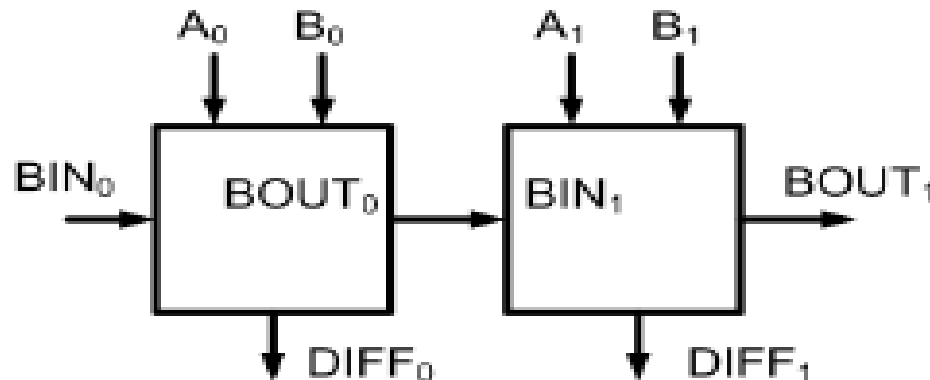
$$S_n = \overline{x_n}(y_n \oplus B_{n-1}) + x_n(\overline{y_n} \oplus \overline{B_{n-1}})$$

$$S_n = x_n \oplus (y_n \oplus B_{n-1})$$

VI - Soustracteur parallèle à propagation de retenu (à n bits)

Pour réaliser un soustracteur à n bits, on peut connecter des soustracteurs complets ensemble comme avec les additionneurs.

Et c'est la chaîne de retenue (d'emprunt) qui risque de prendre beaucoup de temps...



Soustracteur réalisé à partir d'un additionneur :

L'autre option pour la réalisation d'un soustracteur, par l'utilisation d'un complément à 2.

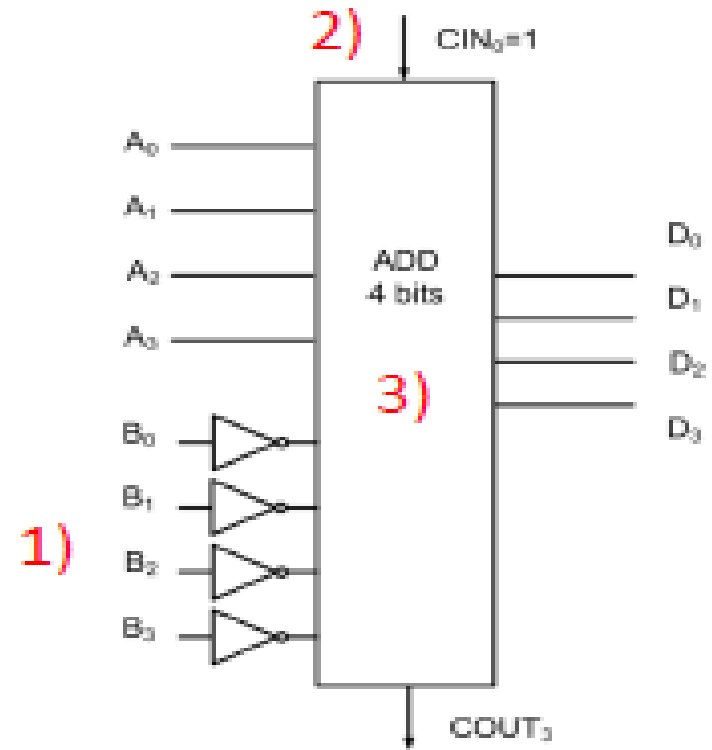
- Imaginons qu'on veuille faire $A - B$:
- On pourrait prendre $-B$ et l'additionner à A
- $-B$ devrait être en complément à 2
- Comment mettre $-B$ en complément à 2?
- On inverse les bits.
- Et on additionne 1.
- Ça s'implante bien avec un additionneur...

L'idée est donc d'additionner A et $(-B)$:

1) Pour avoir $-B$, on fait le complément à 1 (inversion).

2) On additionne 1 au complément à 1 pour avoir un complément a 2 ($C_{IN0} = 1$).

3) Et on additionne normalement...

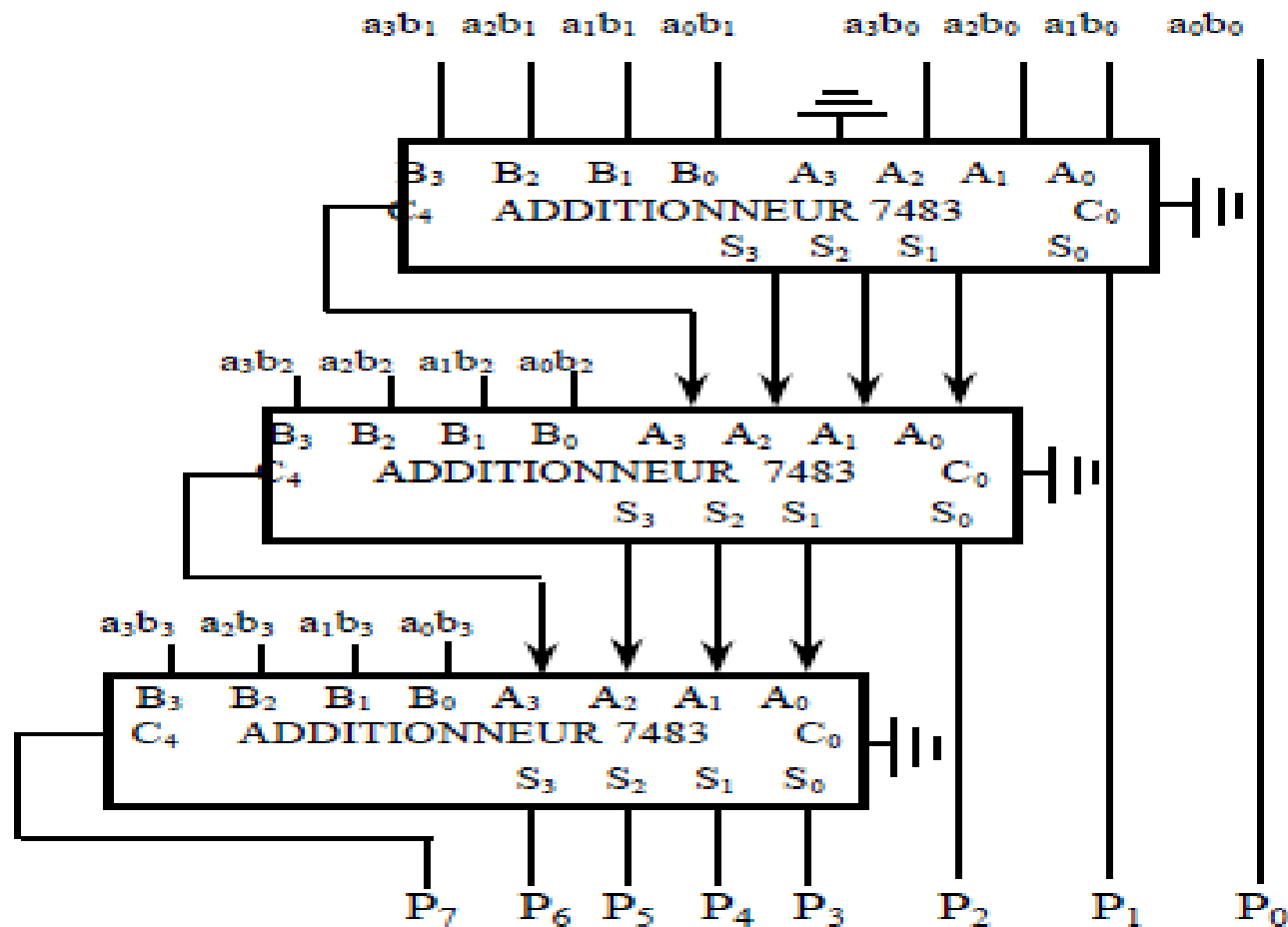


VII - Multiplieur

- Le processus de la multiplication en binaire est illustré par un exemple de multiplication de deux nombres binaires à 4 bits suivant :

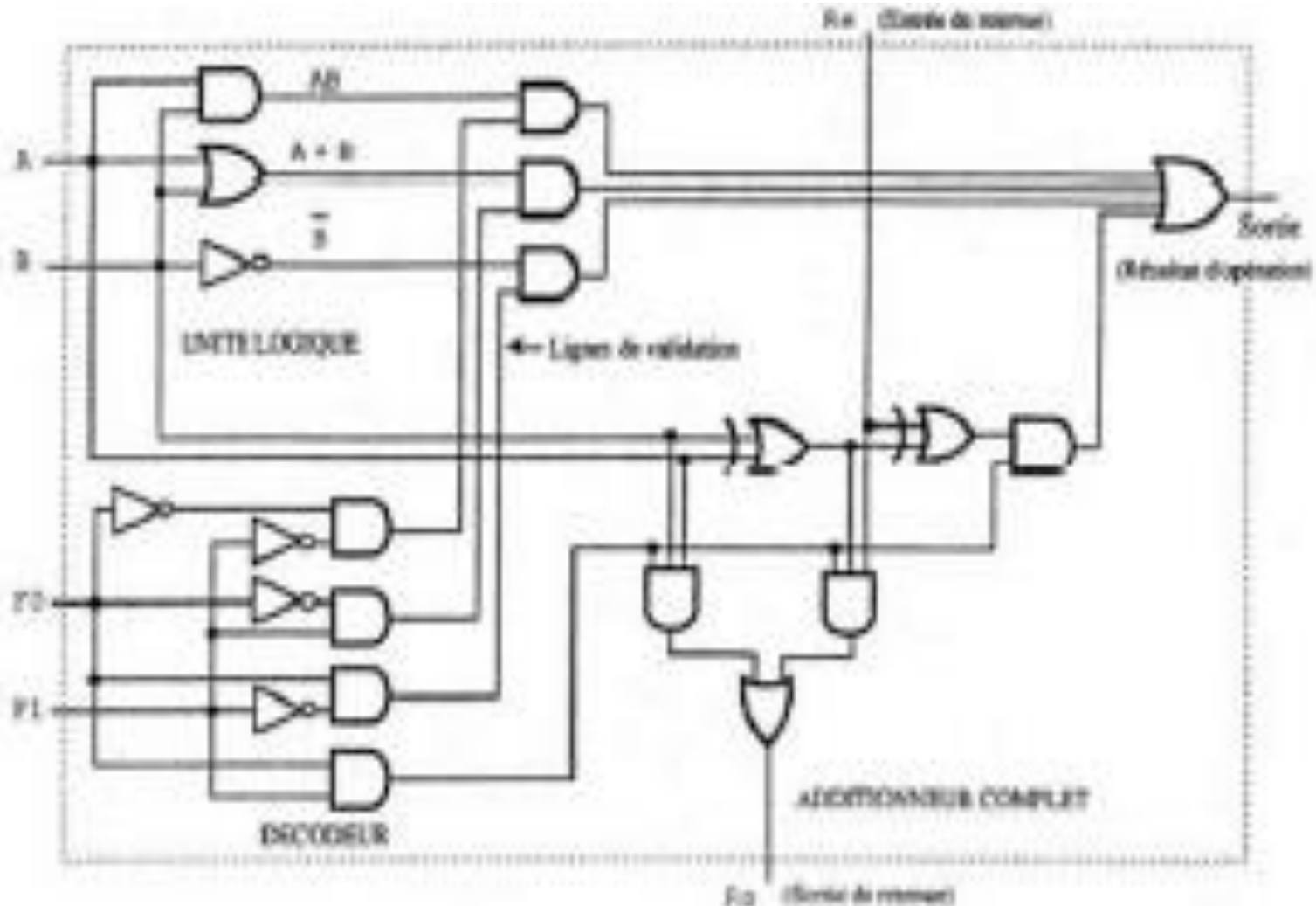
	a ₃	a ₂	a ₁	a ₀	Multiplicande
	b ₃	b ₂	b ₁	b ₀	Multiplicateur
	a ₃ b ₀	a ₂ b ₀	a ₁ b ₀	a ₀ b ₀	1 ^{er} produit partiel
	a ₃ b ₁	a ₂ b ₁	a ₁ b ₁	a ₀ b ₁	2 ^{ème} produit partiel + décalage
	Σ_5^1	Σ_4^1	Σ_3^1	Σ_2^1	Somme partielle Σ^1
	a ₃ b ₂	a ₂ b ₂	a ₁ b ₂	a ₀ b ₂	3 ^{ème} produit partiel + décalage
	Σ_5^2	Σ_4^2	Σ_3^2	Σ_2^2	Somme partielle Σ^2
	a ₃ b ₃	a ₂ b ₃	a ₁ b ₃	a ₀ b ₃	4 ^{ème} produit partiel + décalage
	Σ_5^3	Σ_4^3	Σ_3^3	Σ_2^3	Somme partielle Σ^3
	P ₇	P ₆	P ₅	P ₄	
	P ₃	P ₂	P ₁	P ₀	
					Produit

- La multiplication de deux nombres binaires à 4 bits est effectuée par un multiplieur parallèle utilisant des additionneur complets (A.C) selon le schéma de la figure suivante :



VIII – Exemple d'Unité Arithmétique et Logique

- **Exemple d'UAL simple à 1 bit :** Cette unité effectue les opérations arithmétiques et logiques de base (le code opération détermine la partie du circuit qui va effectuer les opérations).



Merci pour votre attention