

Architecture des ordinateurs

Chapitre 2 : C. Circuits Arithmétiques

L2 Génie Industriel

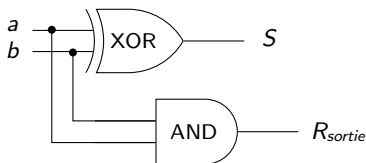
Circuits arithmétiques de base

- L'additionneur / Le soustracteur
- L'incrémenteur / Le décrémenteur
- Le décaleur
- L'Unité Arithmétique et Logique (UAL)

Demi-additionneur

- **Entrées** : les 2 bits à additionner a et b
- **Sorties** :
 - ▶ la somme $S = a + b$
 - ▶ la retenue de sortie R_{sortie}
- **Rôle** : Additionner a et b en conservant la retenue

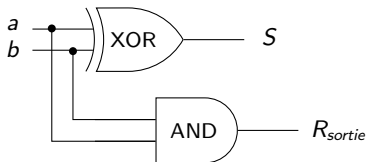
| a | b | S | R_{sortie} |
|-----|-----|-----|--------------|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |



Demi-additionneur

- **Entrées** : les 2 bits à additionner a et b
- **Sorties** :
 - ▶ la somme $S = a + b$
 - ▶ la retenue de sortie R_{sortie}
- **Rôle** : Additionner a et b en conservant la retenue

| a | b | S | R_{sortie} |
|-----|-----|-----|--------------|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |



Problème : si plusieurs additions successives ?

Additionneur complet (1/2)

- Entrées :

- ▶ les 2 bits à additionner a et b
- ▶ la retenue d'entrée R_{entree}

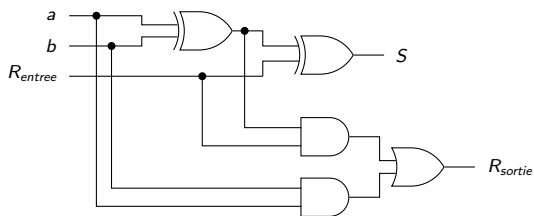
- Sorties :

- ▶ la somme $S = a + b + R_{entree}$
- ▶ la retenue de sortie R_{sortie}

- Rôle : Additionner a et b en prenant en compte la retenue d'entrée R_{entree} et en conservant la retenue de sortie R_{sortie}

Additionneur complet (2/2)

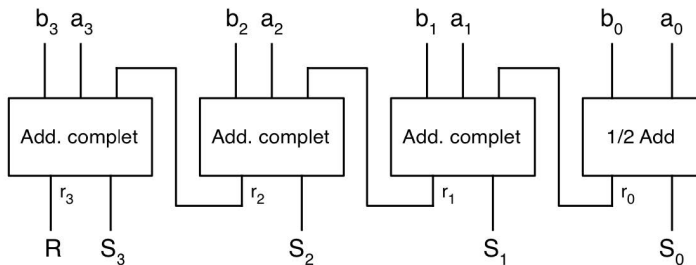
| a | b | R_{entree} | S | R_{sortie} |
|-----|-----|--------------|-----|--------------|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |



Exemple : Additionneur 4 bits

$$A = a_3 a_2 a_1 a_0 = 1 1 0 1$$

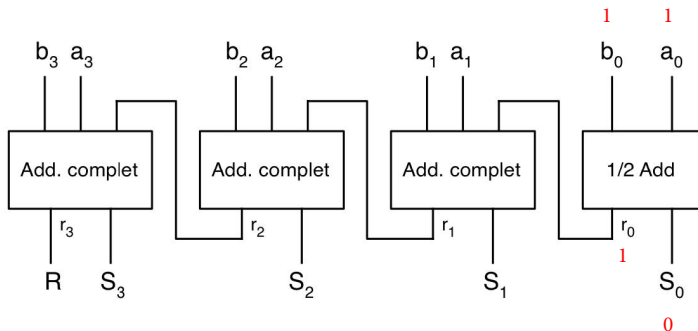
$$B = b_3 b_2 b_1 b_0 = 1 0 1 1$$



Exemple : Additionneur 4 bits

$$A = a_3 a_2 a_1 a_0 = 1 1 0 1$$

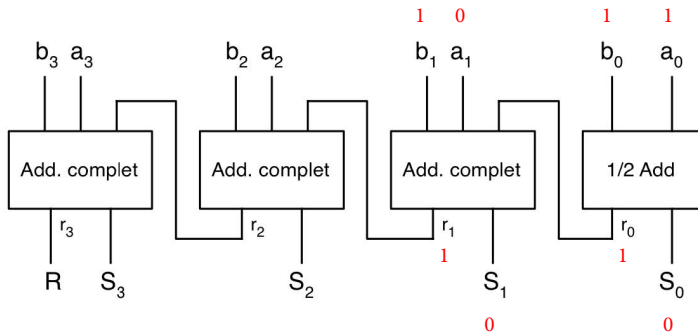
$$B = b_3 b_2 b_1 b_0 = 1 0 1 1$$



Exemple : Additionneur 4 bits

$$A = a_3 a_2 a_1 a_0 = 1 1 0 1$$

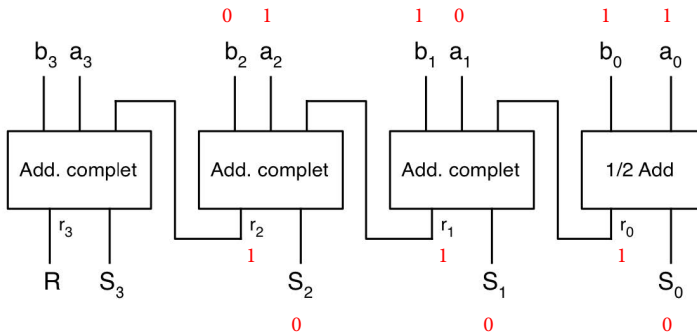
$$B = b_3 b_2 b_1 b_0 = 1 0 1 1$$



Exemple : Additionneur 4 bits

$$A = a_3 a_2 a_1 a_0 = 1 1 0 1$$

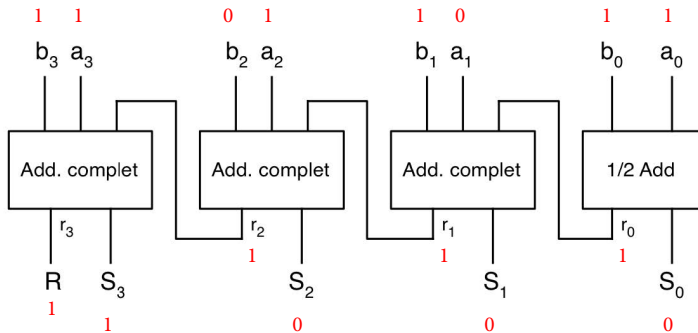
$$B = b_3 b_2 b_1 b_0 = 1 0 1 1$$



Exemple : Additionneur 4 bits

$$A = a_3 a_2 a_1 a_0 = 1 1 0 1$$

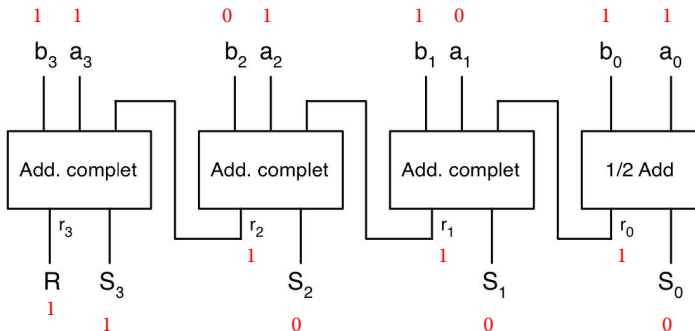
$$B = b_3 b_2 b_1 b_0 = 1 0 1 1$$



Exemple : Additionneur 4 bits

$$A = a_3 a_2 a_1 a_0 = 1 1 0 1$$

$$B = b_3 b_2 b_1 b_0 = 1 0 1 1$$

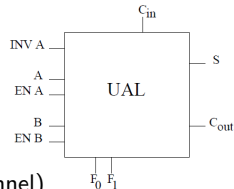


$$S = A + B = S_3 S_2 S_1 S_0 = 1 0 0 0 \text{ avec retenue } R = 1$$

Unité Arithmétique et Logique (UAL) (1/2)

● Entrées :

- ▶ A et B : les variables (données)
- ▶ F_0 et F_1 : bits de choix du signal d'activation
- ▶ R_{entree} : la retenue d'entrée
- ▶ $EN A$ et $EN B$: les bits inhibiteurs de A et B (optionnel)
- ▶ $INV A$: pour obtenir \bar{A} (optionnel)



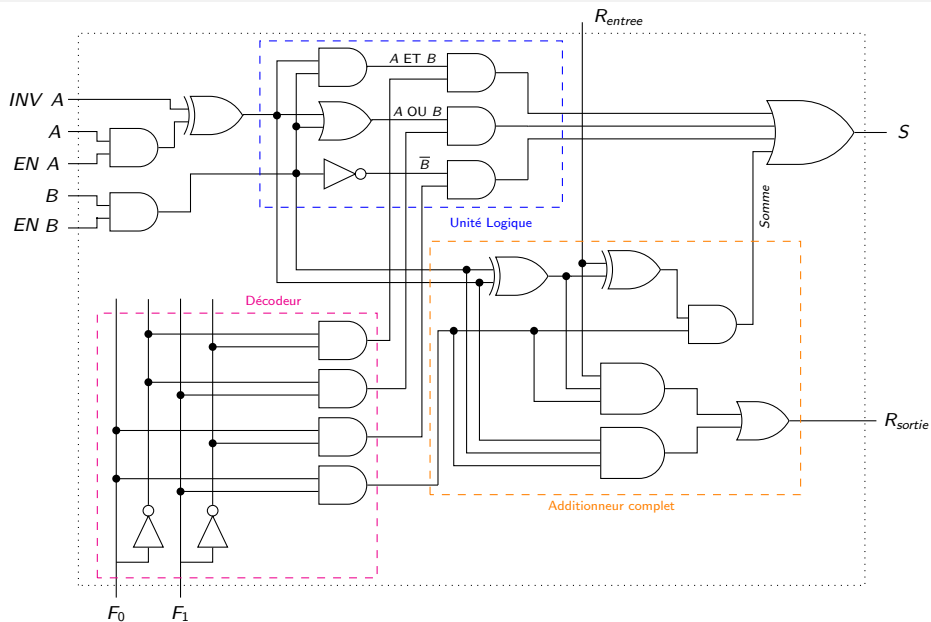
● Sorties :

- ▶ S : le résultat de l'opération
- ▶ R_{sortie} : la retenue de sortie

● Rôle : Faire l'une des 4 opérations (en fonction des bits d'activation choisis) :

- ▶ A ET B
- ▶ A OU B
- ▶ \bar{B}
- ▶ $A + B + R_{entree}$

Unité Arithmétique et Logique (UAL) (2/2)



UAL n bits

- Pour 2 bits d'entrée, l'UAL est un circuit qui a peu d'intérêt ...
- En connectant les retenues de n UALs, on obtient une UAL n bits telle que :
 - ▶ les opérations logiques sont des opérations bit à bit
 - ▶ les opérations arithmétiques sont effectuées sur des entiers en complément à 2 sur n bits

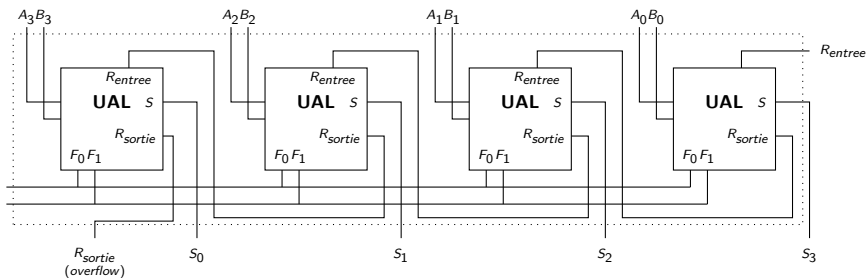
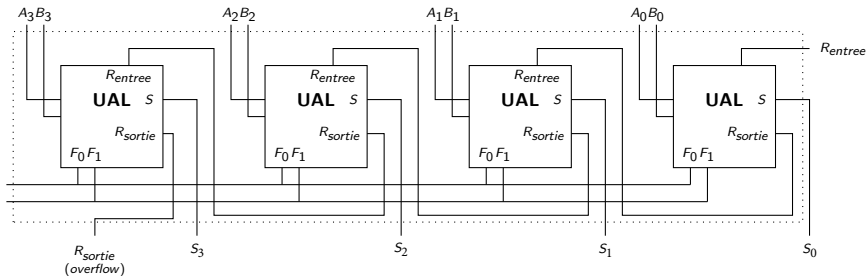


FIGURE: UAL à 4 bits

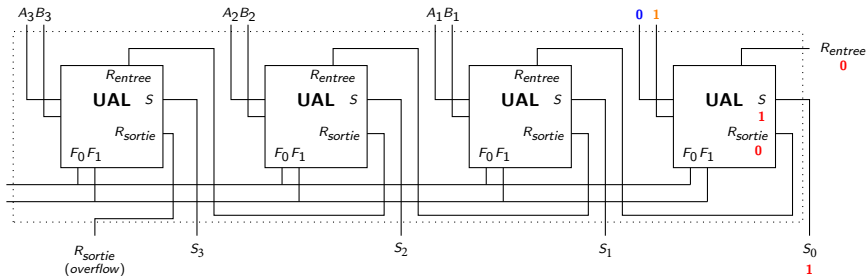
Exemple

- UAL 4 bits
 - On souhaite faire l'addition entre A et B (données) telle que :
 - ▶ A et B sont codés sur 4 bits
 - ▶ $A = 14$ (en base 10) = 1110 (en base 2)
 - ▶ $B = 5$ (en base 10) = 0101 (en base 2)
- ⇒ $A_0 = 0, A_1 = 1, \dots$ et $B_0 = 1, B_1 = 0, \dots$



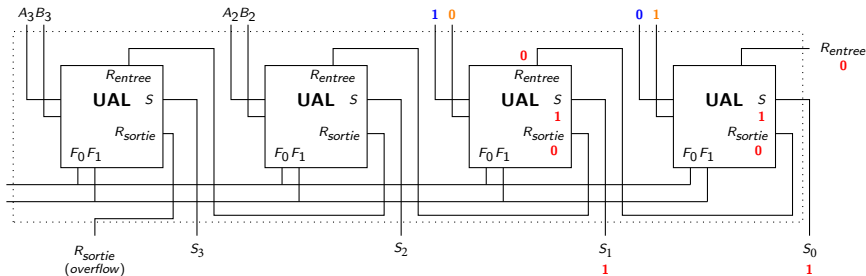
Exemple

- UAL 4 bits
 - On souhaite faire l'addition entre A et B (données) telle que :
 - ▶ A et B sont codés sur 4 bits
 - ▶ $A = 14$ (en base 10) = 1110 (en base 2)
 - ▶ $B = 5$ (en base 10) = 0101 (en base 2)
- ⇒ $A_0 = 0, A_1 = 1, \dots$ et $B_0 = 1, B_1 = 0, \dots$



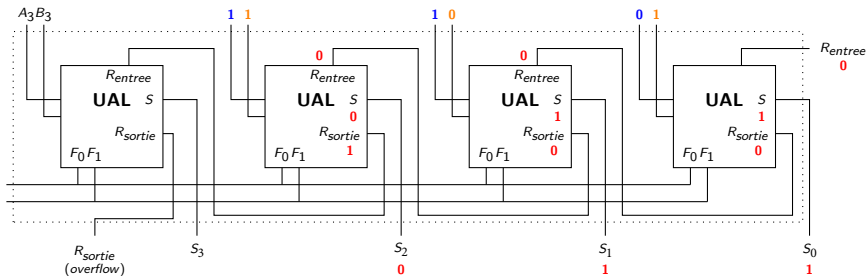
Exemple

- UAL 4 bits
 - On souhaite faire l'addition entre A et B (données) telle que :
 - ▶ A et B sont codés sur 4 bits
 - ▶ $A = 14$ (en base 10) = 1110 (en base 2)
 - ▶ $B = 5$ (en base 10) = 0101 (en base 2)
- ⇒ $A_0 = 0, A_1 = 1, \dots$ et $B_0 = 1, B_1 = 0, \dots$



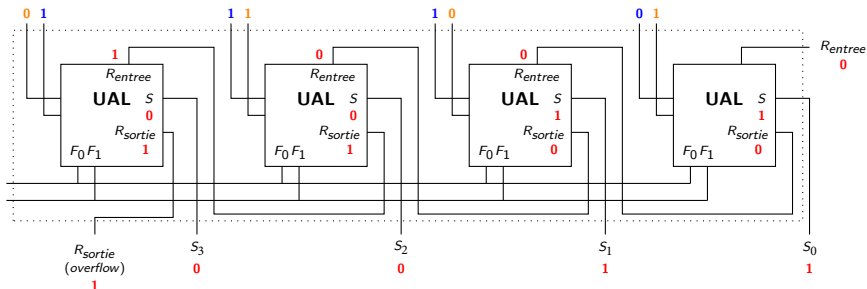
Exemple

- UAL 4 bits
 - On souhaite faire l'addition entre A et B (données) telle que :
 - ▶ A et B sont codés sur 4 bits
 - ▶ $A = 14$ (en base 10) = 1110 (en base 2)
 - ▶ $B = 5$ (en base 10) = 0101 (en base 2)
- ⇒ $A_0 = 0, A_1 = 1, \dots$ et $B_0 = 1, B_1 = 0, \dots$



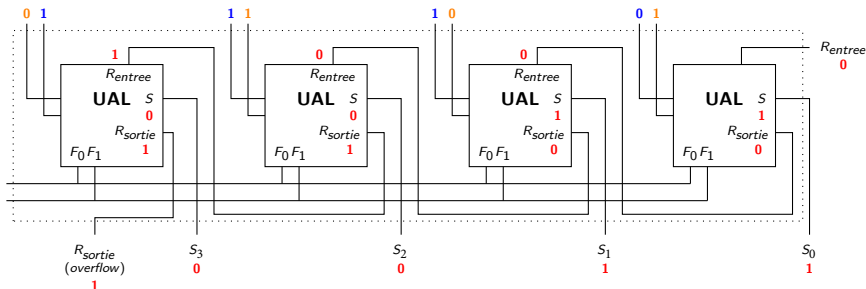
Exemple

- UAL 4 bits
 - On souhaite faire l'addition entre A et B (données) telle que :
 - ▶ A et B sont codés sur 4 bits
 - ▶ $A = 14$ (en base 10) = 1110 (en base 2)
 - ▶ $B = 5$ (en base 10) = 0101 (en base 2)
- ⇒ $A_0 = 0, A_1 = 1, \dots$ et $B_0 = 1, B_1 = 0, \dots$



Exemple

- UAL 4 bits
 - On souhaite faire l'addition entre A et B (données) telle que :
 - ▶ A et B sont codés sur 4 bits
 - ▶ $A = 14$ (en base 10) = 1110 (en base 2)
 - ▶ $B = 5$ (en base 10) = 0101 (en base 2)
- ⇒ $A_0 = 0, A_1 = 1, \dots$ et $B_0 = 1, B_1 = 0, \dots$



$$A + B = 14 + 5 = 19 \text{ (en base 10)} = 1\ 0011 \text{ (en base 2)}$$

UAL - Résumé des fonctions

| F_0 | F_1 | $EN A$ | $EN B$ | $INV A$ | R_{entree} | Fonction |
|-------|-------|--------|--------|---------|--------------|-------------------|
| 0 | 0 | 1 | 1 | 0 | 0 | $A \text{ ET } B$ |
| 0 | 1 | 1 | 1 | 0 | 0 | $A \text{ OU } B$ |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 | B |
| 0 | 1 | 1 | 0 | 0 | 0 | A |
| 0 | 1 | 1 | 0 | 1 | 0 | \overline{A} |
| 1 | 0 | 1 | 1 | 0 | 0 | \overline{B} |
| 1 | 1 | 1 | 1 | 0 | 0 | $A + B$ |
| 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 | 0 | -1 |
| 1 | 1 | 0 | 1 | 0 | 1 | $B + 1$ |
| 1 | 1 | 0 | 1 | 1 | 0 | $B - 1$ |
| 1 | 1 | 1 | 0 | 0 | 1 | $A + 1$ |
| 1 | 1 | 1 | 0 | 1 | 1 | $-A$ |
| 1 | 1 | 1 | 1 | 0 | 1 | $A + B + 1$ |
| 1 | 1 | 1 | 1 | 1 | 1 | $B - A$ |