Transistor, Portes Logiques et Circuits Combinatoires

Stéphane Devismes Fabienne Carrier

Université Grenoble Alpes

20 avril 2020

Plan

Plan

Définition

Le transistor est le composant électronique actif fondamental en électronique utilisé principalement comme

- **interrupteur commandé** et
- pour l'**amplification**, mais aussi
- pour **stabiliser une tension**, **moduler un signal**
- ainsi que de nombreuses autres utilisations.

Type de transistors

Il y a deux principaux types de transistors :

- Le transistor à jonctions (technologie dite bipolaire).

  Dans cette famille, on trouve notamment les circuits **RTL** (*Resistor Transistor Logic*), **TTL** (*Transistor Transistor Logic*), **DTL** (*Diode Transistor Logic*) ou encore **ECL** (*Emitter Coupled Logic*).

- Le transistor à effet de champ (technologie dite unipolaire).

  Dans cette famille, on trouve notamment les circuits **MOS** (*Metal Oxyde Semiconductor*) qui sont en général moins rapides, sauf les plus récents comme **HMOS** ou **XMOS3**.

Type de transistors

Il y a deux principaux types de transistors :

- Le transistor à jonctions (technologie dite bipolaire).

  Dans cette famille, on trouve notamment les circuits **RTL** (*Resistor Transistor Logic*), **TTL** (*Transistor Transistor Logic*), **DTL** (*Diode Transistor Logic*) ou encore **ECL** (*Emitter Coupled Logic*).

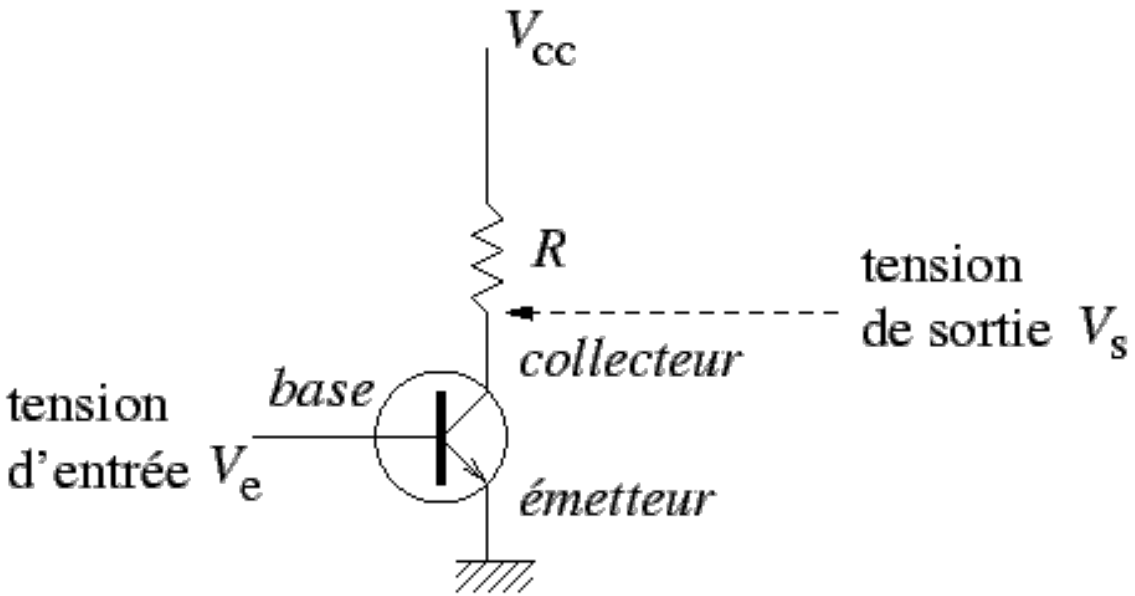
- Le transistor à effet de champ (technologie dite unipolaire).

  Dans cette famille, on trouve notamment les circuits **MOS** (*Metal Oxyde Semiconductor*) qui sont en général moins rapides, sauf les plus récents comme **HMOS** ou **XMOS3**.

Remarque : Aujourd'hui, tous les processeurs sont en technologie **MOSFET**, ou des améliorations.

Fonctionnement d'un transistor

Un transistor fonctionne comme un **interrupteur** (ici, un transistor *à jonction NPN*).



- Si la tension d'entrée V_e est inférieure à une valeur critique, le transistor se bloque et n'est pas conducteur ; l'interrupteur est ouvert et la tension de sortie V_s est proche de V_{cc}, donc à un niveau haut.
- Si la tension d'entrée V_e est supérieure à la valeur critique, le transistor bascule, ce qui le rend conducteur ; l'interrupteur est fermé et la tension de sortie est au niveau bas, proche de la masse. La résistance R est conçue pour limiter le courant à travers le transistor, dans ce dernier cas.

Plan

Définition

La forme la plus élémentaire de **circuit** est la porte logique.

Son comportement est dit **binaire** car il est caractérisé par deux états : l'état 0, qui représente la valeur logique faux et l'état 1, qui représente la valeur logique vrai.

Les constructeurs utilisent dans certains cas une logique dite **positive** : l'état 1 correspond à une tension comprise entre 2 et 5 volts (niveau haut) et l'état 0 à une tension comprise entre 0 et 1 volt (niveau bas).

Dans d'autres cas, ils utilisent une logique dite **négative** où l'état 1 correspond au niveau bas, tandis que l'état 0 correspond au niveau haut.

Définition

La forme la plus élémentaire de **circuit** est la porte logique.

Son comportement est dit **binaire** car il est caractérisé par deux états : l'état 0, qui représente la valeur logique faux et l'état 1, qui représente la valeur logique vrai.

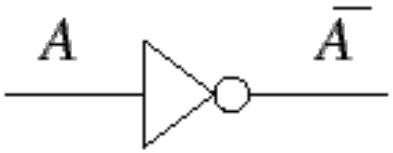
Les constructeurs utilisent dans certains cas une logique dite **positive** : l'état 1 correspond à une tension comprise entre 2 et 5 volts (niveau haut) et l'état 0 à une tension comprise entre 0 et 1 volt (niveau bas).

Dans d'autres cas, ils utilisent une logique dite **négative** où l'état 1 correspond au niveau bas, tandis que l'état 0 correspond au niveau haut.

Remarque : La transmission n'est pas instantanée : le délai de traversée d'une porte correspond au temps de propagation des signaux de l'entrée vers la sortie (de l'ordre de quelques dixième nanosecondes pour la porte la plus simple).

Porte NON

La porte logique la plus simple est celle qui réalise une inversion logique. Elle est symbolisée par :



Elle est construite avec un seul transistor :

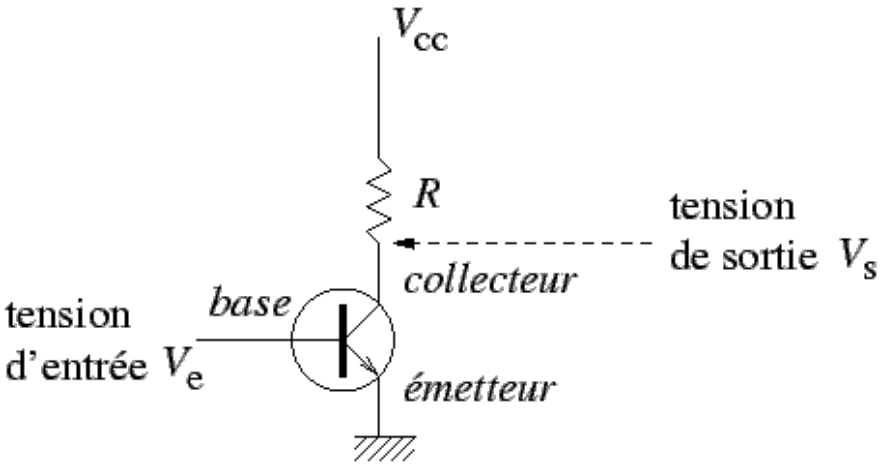


Table de vérité

V_e	V_s
0	1
1	0

Porte NON-ET

La porte logique du NON-ET :

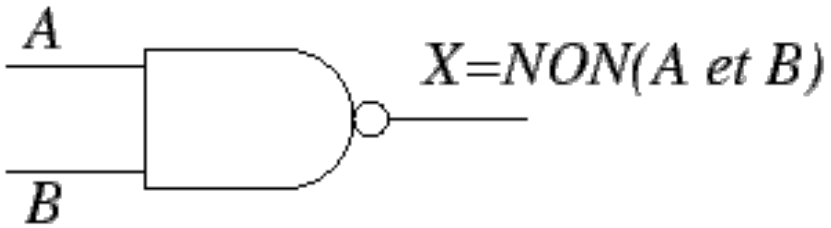
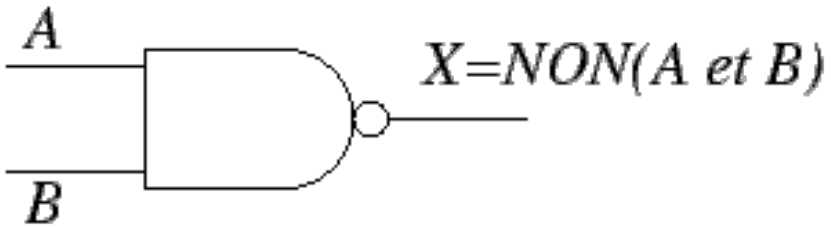


Table de vérité

V_1	V_2	V_s
0	0	1
0	1	1
1	0	1
1	1	0

Porte NON-ET

La porte logique du NON-ET :



Elle est construite en reliant deux transistors en série :

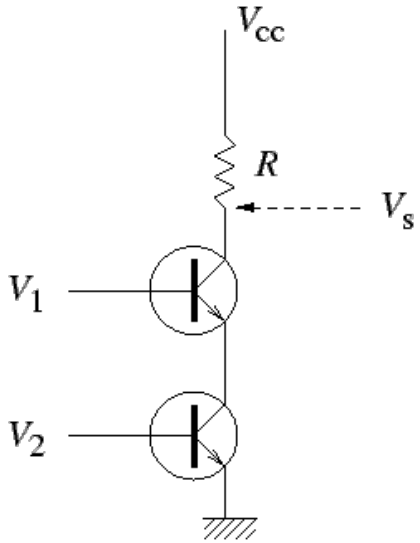


Table de vérité

V_1	V_2	V_s
0	0	1
0	1	1
1	0	1
1	1	0

On vérifie facilement que la tension de sortie V_s est au niveau bas si et seulement si V_1 et V_2 sont au niveau haut.

Porte NON-OU

La porte logique du NON-OU :

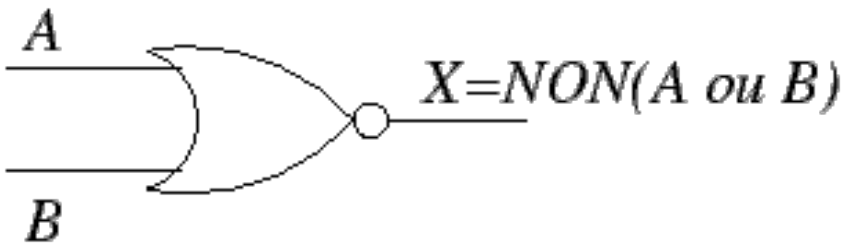
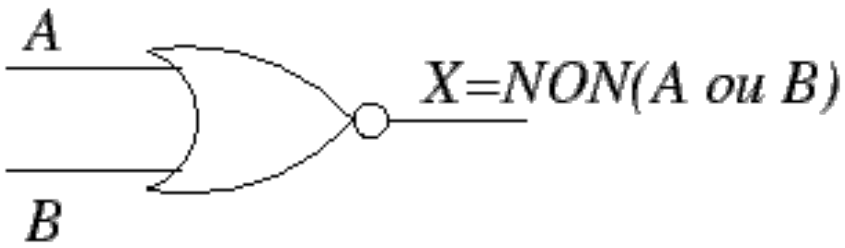


Table de vérité

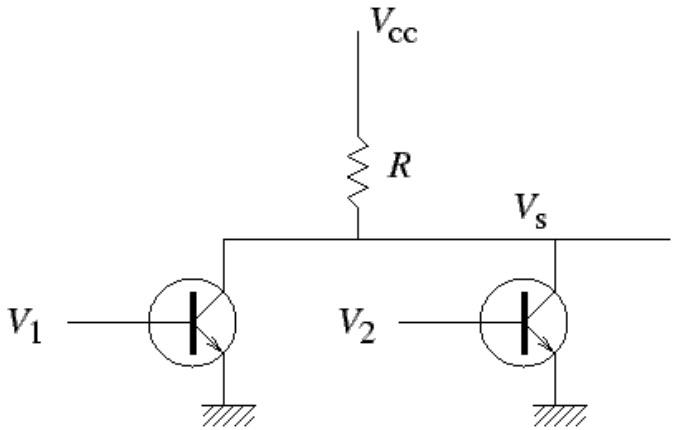
V_1	V_2	V_s
0	0	1
0	1	0
1	0	0
1	1	0

Porte NON-OU

La porte logique du NON-OU :



Elle est construite en reliant deux transistors en parallèle :



On vérifie facilement que la tension de sortie V_s est au niveau bas si et seulement si l'une des deux tensions V_1 ou V_2 est au niveau haut.

Table de vérité

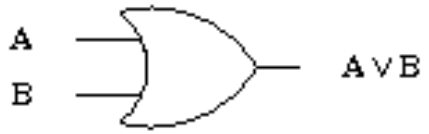
V_1	V_2	V_s
0	0	1
0	1	0
1	0	0
1	1	0

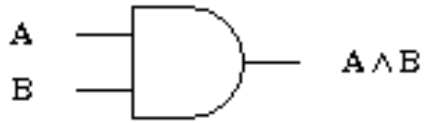
Fonctions booléennes

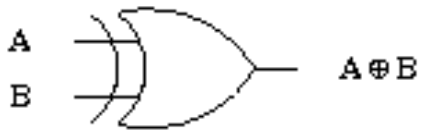
Avec les trois portes **NON, NON-ET et NON-OU** ont obtient une logique complète : toute fonction booléenne peut être décrite avec ces opérateurs !

Exemple : Pour obtenir un **OU** il suffit de relier un **NON-OU** et un **NON**.

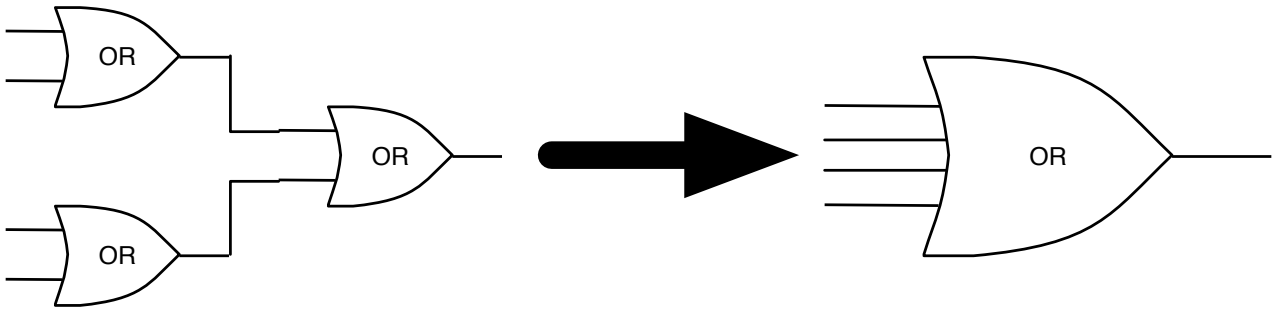
Autres symboles







Simplifications



- Possible car commutativité et associativité
- Pour n'importe quel nombre d'entrées ≥ 2
- Possible aussi pour **et, non et, non ou**

Principes des circuits intégrés

Un circuit intégré est une plaquette de silicium sur laquelle sont intégrées les portes du circuit. La plaquette est encapsulée dans un boîtier avec sur les côtés des broches permettant les connexions électriques.

Ces circuits sont classés suivant la densité d'intégration, c'est-à-dire le nombre de portes ou transistors par circuits (ou par mm^2) :

SSI	*Small Scale Integration*	1 à 10 portes par circuit
MSI	*Medium Scale Integration*	10 à 100
LSI	*Large Scale Integration*	100 à 100 000
VLSI	*Very Large Scale Integration*	plus de 100 000

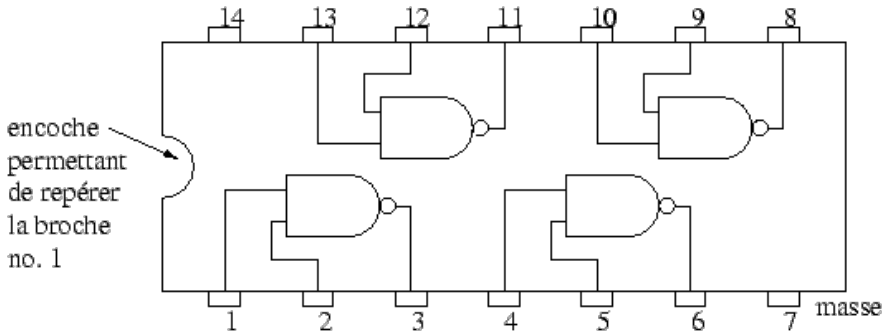


FIGURE – Un circuit SSI dans un boîtier à 14 broches : circuit TTL 7400 de Texas Instruments.

Plan

Définition

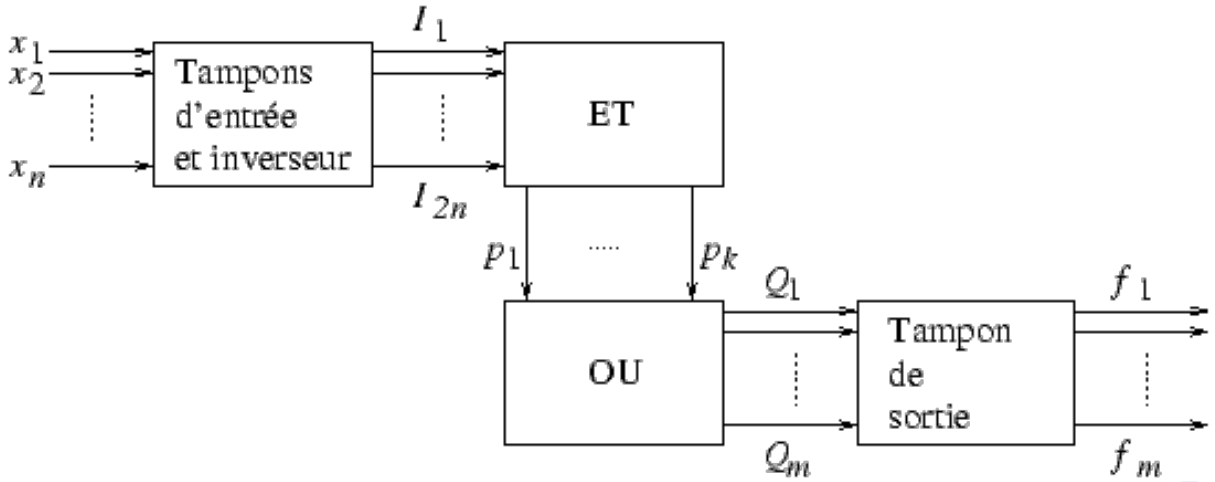
Un circuit combinatoire est un circuit comportant **des entrées** et **des sorties** booléennes et dont les sorties sont une expression logique des valeurs d'entrées.

Circuit FPGA

Ce sont des réseaux logiques programmables (**Field Programmable Logic Array** ou **Field Programmable Gate Array**) qui permettent de réaliser des fonctions lorsqu'elles sont sous la forme d'une somme de produits.

Considérons par exemple 20 broches dont 12 correspondent à des entrées et 6 à des sorties. Les 12 entrées sont inversées, ce qui fournit 24 variables internes. Ces 24 variables sont toutes connectées à 50 portes ET, ce qui donne 1 200 fusibles, au départ intacts.

Programmer le circuit consiste alors à détruire certains fusibles pour obtenir les produits de la fonction à programmer. Les 6 sorties proviennent de 6 portes OU qui reçoivent chacune les sorties des 50 portes ET. On obtient ainsi 300 fusibles dont certains sont détruits pour obtenir la somme des termes de la fonction.



Multiplexeur

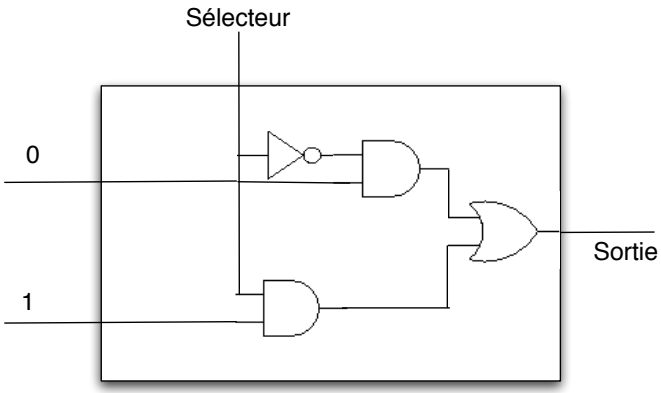
Un multiplexeur comporte 2^n entrées, 1 sortie et n lignes de sélection. La configuration des n lignes de sélection fournit une valeur parmi les 2^n entrées et connecte cette entrée à la sortie.

Exemple : $n = 1$

Multiplexeur

Un multiplexeur comporte 2^n entrées, 1 sortie et n lignes de sélection. La configuration des n lignes de sélection fournit une valeur parmi les 2^n entrées et connecte cette entrée à la sortie.

Exemple : $n = 1$



Décodeur

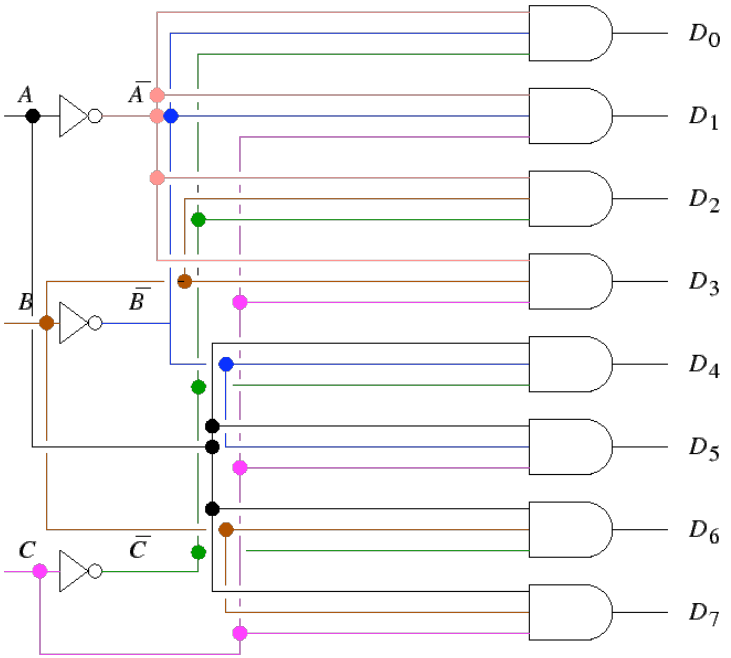
Un décodeur comprend n entrées et 2^n sorties, la sortie activée correspondant à la configuration binaire du mot formé par les n entrées. Un tel circuit sert à sélectionner des adresses de la mémoire.

Exemple : Un décodeur 3 vers 8 ($n = 3$).

Décodeur

Un décodeur comprend n entrées et 2^n sorties, la sortie activée correspondant à la configuration binaire du mot formé par les n entrées. Un tel circuit sert à sélectionner des adresses de la mémoire.

Exemple : Un décodeur 3 vers 8 ($n = 3$).



Comparateur

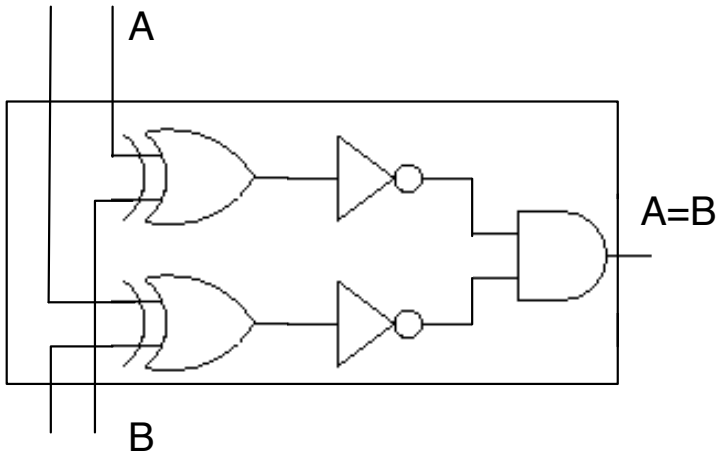
Un comparateur à $2 \times n$ entrées et 1 sortie, les $2 \times n$ entrées formant deux mots de n bits : A et B. La sortie vaut 1 si le mot $A = B$, 0 sinon.

Exemple : $n = 2$

Comparateur

Un comparateur à $2 \times n$ entrées et 1 sortie, les $2 \times n$ entrées formant deux mots de n bits : A et B. La sortie vaut 1 si le mot $A = B$, 0 sinon.

Exemple : $n = 2$



Décaleur

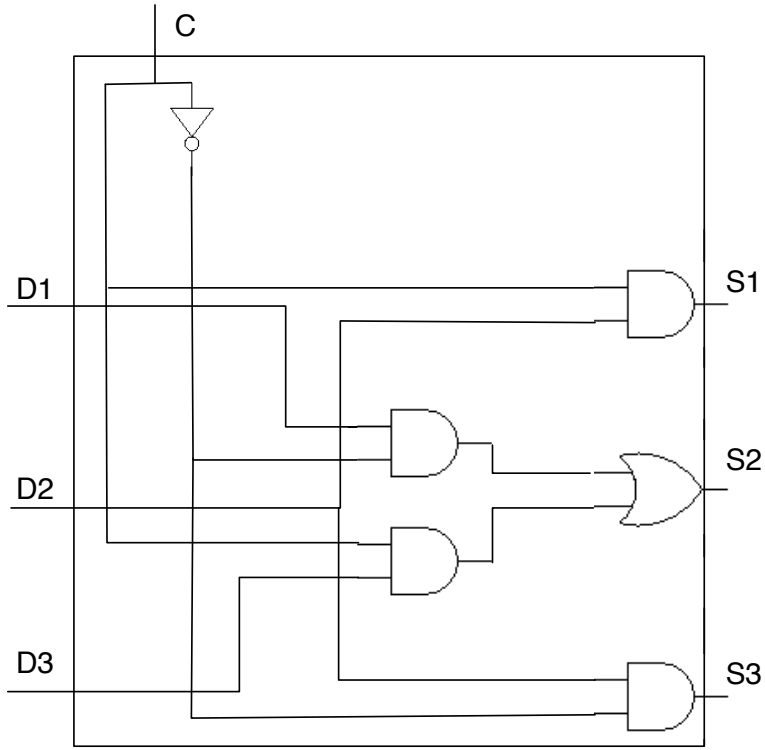
Un décaleur est formé de $(n+1)$ entrées $D_1, \ldots, D_n$, C et de n sorties $S_1, \ldots, S_n$ et opère un décalage de 1 bit sur les entrées $D_1, \ldots, D_n$. Si $C = 1$, il s'agit d'un décalage à droite et si $C = 0$, d'un décalage à gauche.

Exemple : $n = 3$

Décaleur

Un décaleur est formé de $(n+1)$ entrées $D_1, \ldots, D_n$, C et de n sorties $S_1, \ldots, S_n$ et opère un décalage de 1 bit sur les entrées $D_1, \ldots, D_n$. Si $C = 1$, il s'agit d'un décalage à droite et si $C = 0$, d'un décalage à gauche.

Exemple : $n = 3$



Demi-additionneur

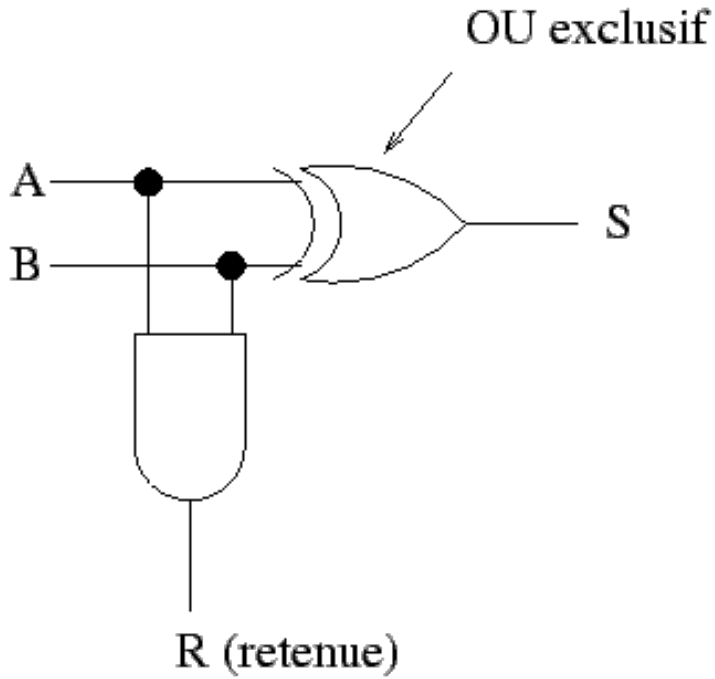
Un demi-additionneur additionne deux bits (A et B) et évalue la retenue éventuelle (R).

A	B	$S = A + B$	R
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Demi-additionneur

Un demi-additionneur additionne deux bits (A et B) et évalue la retenue éventuelle (R).

A	B	$S = A + B$	R
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



Additionneur

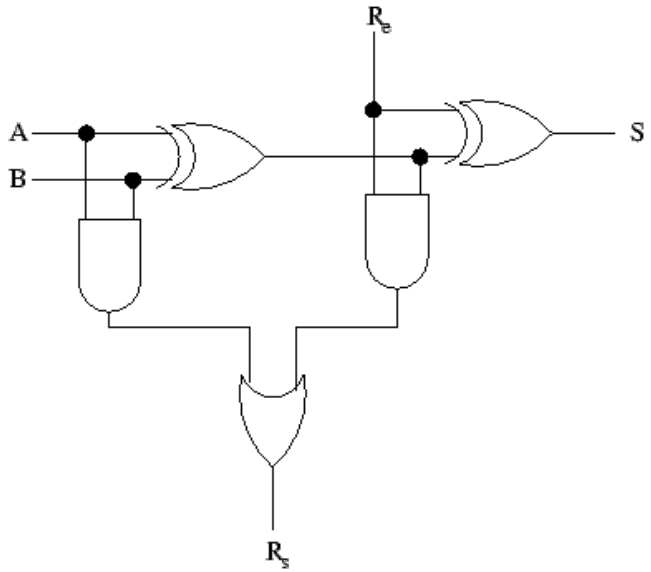
Un additionneur additionne deux bits (A et B) et une retenue (R_e) et évalue la retenue éventuelle (R_s).

A	B	R_e	$S = A + B$	R_s
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

Additionneur

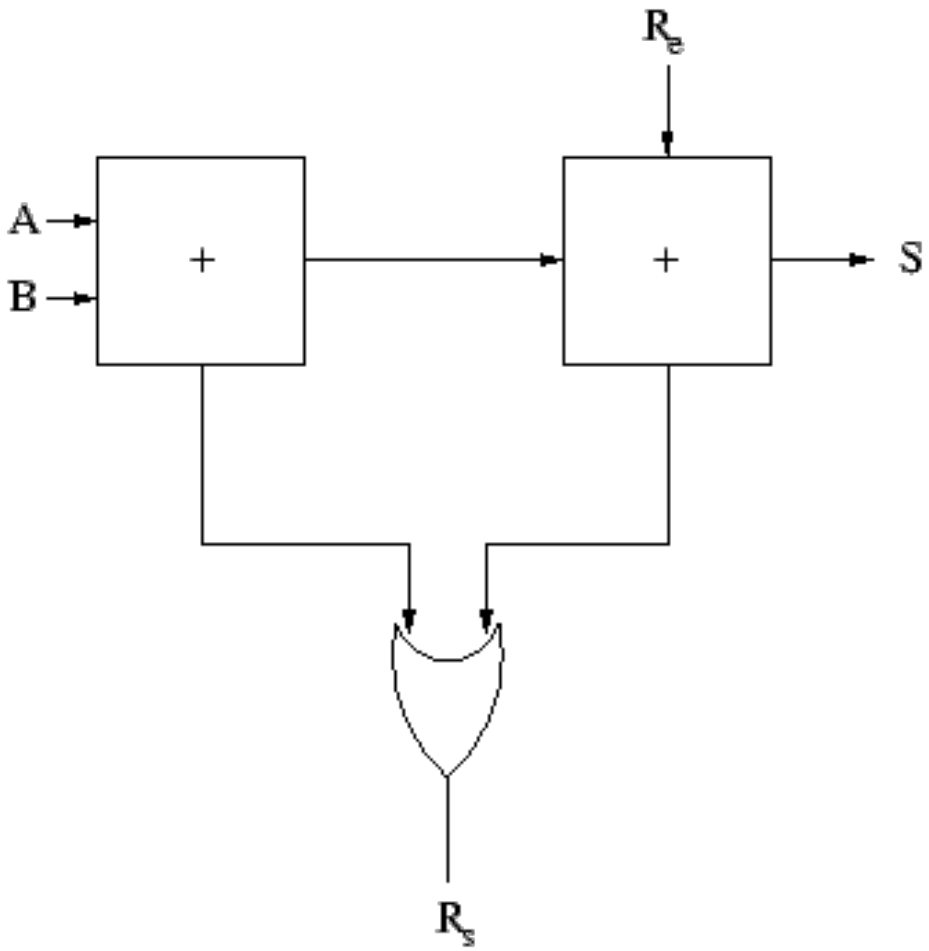
Un additionneur additionne deux bits (A et B) et une retenue (R_e) et évalue la retenue éventuelle (R_s).

A	B	R_e	$S = A + B$	R_s
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

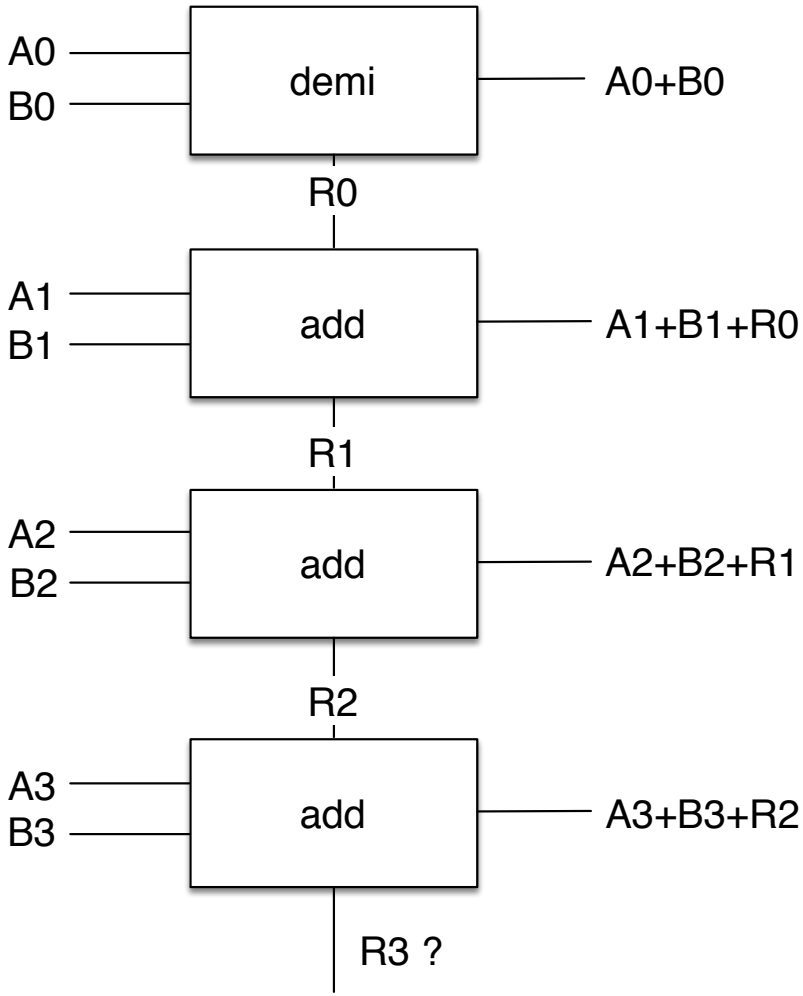


Du demi-additionneur à l'additionneur

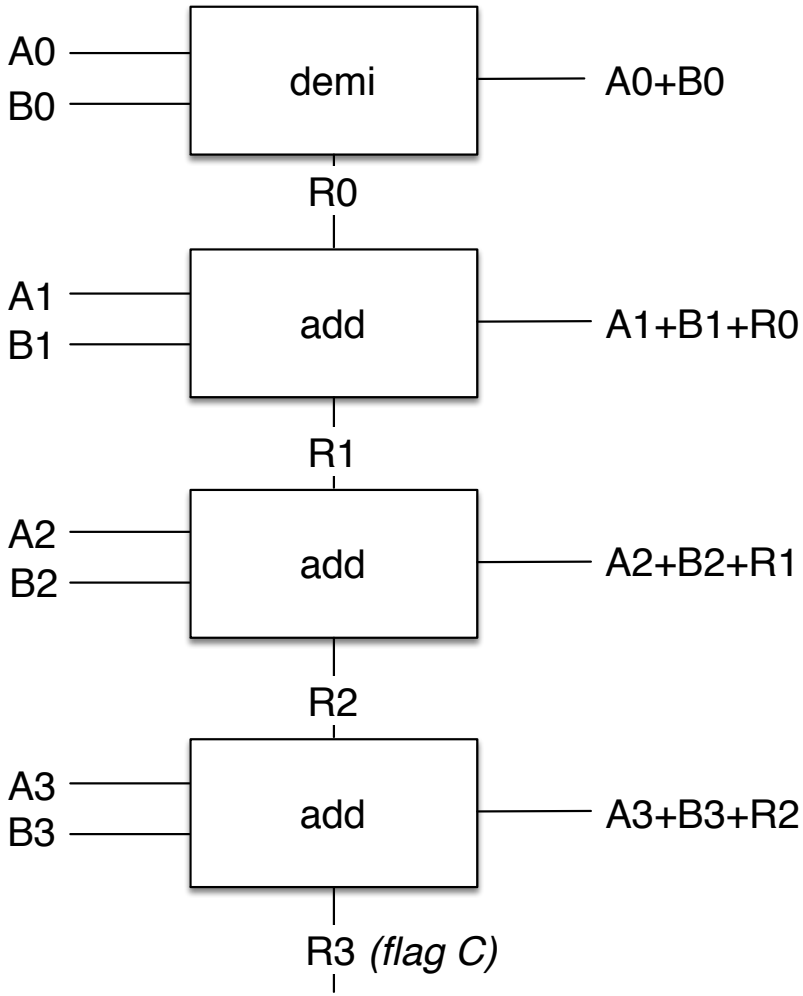
Du demi-additionneur à l'additionneur



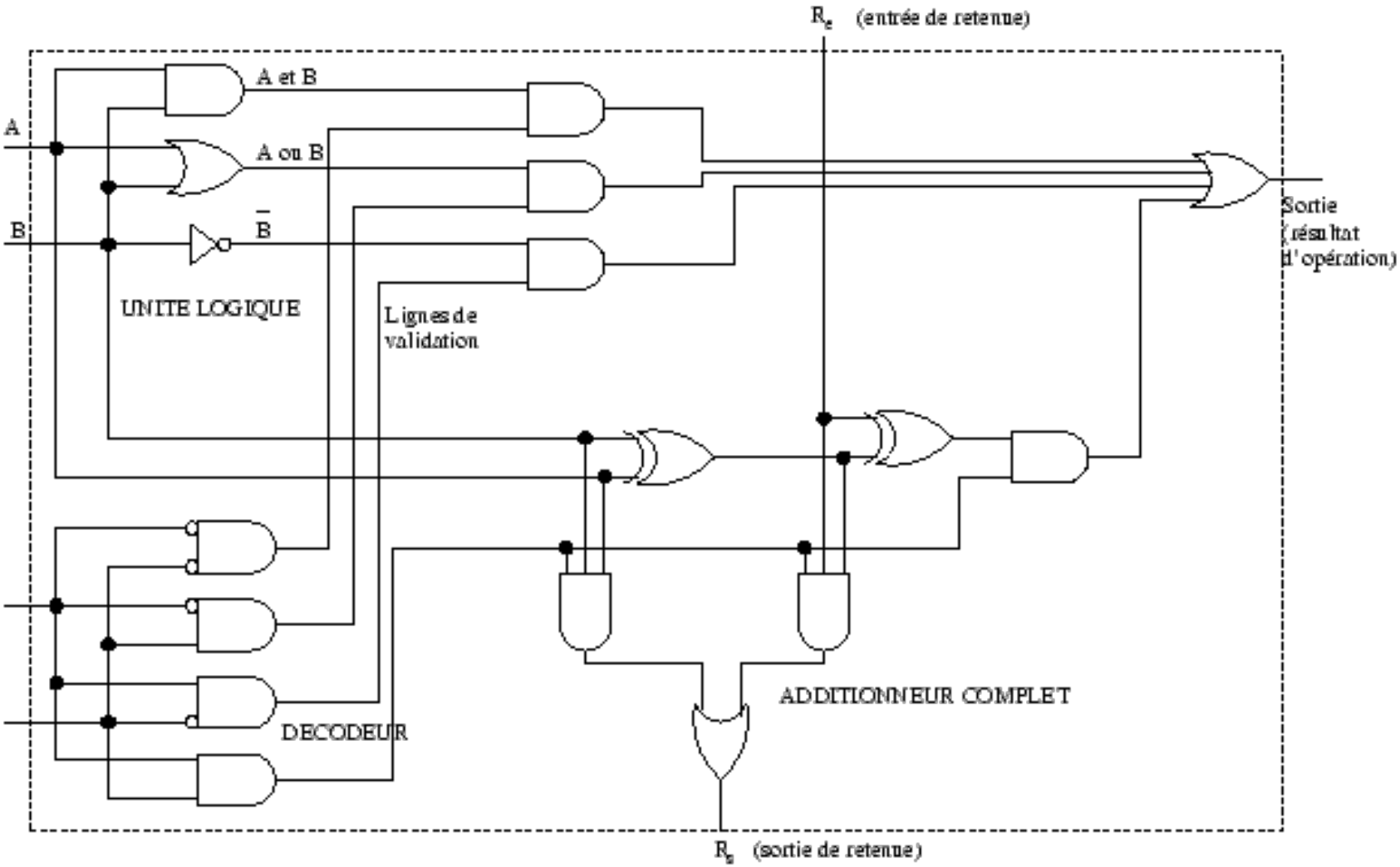
Addition de mots (sur 4 bits)



Addition de mots (sur 4 bits)



UAL


R_e (entrée de retenue)
A et B
A
A ou B
B
$\overline{B}$
UNITE LOGIQUE
Lignes de validation
Sortie (résultat d'opération)
ADDITIONNEUR COMPLET
DECODEUR
R_s (sortie de retenue)

Plan

Conclusion

Conclusion

- Avec **des transistors** on fait **des portes logiques**

Conclusion

- Avec **des transistors** on fait **des portes logiques**
- Avec **des portes logiques** on fait **des circuits combinatoires**

Conclusion

- Avec **des transistors** on fait **des portes logiques**
- Avec **des portes logiques** on fait **des circuits combinatoires**
- Avec **des circuits combinatoires** on fait **des UALs, des unités de commandes...**

Conclusion

- Avec **des transistors** on fait **des portes logiques**
- Avec **des portes logiques** on fait **des circuits combinatoires**
- Avec **des circuits combinatoires** on fait **des UALs, des unités de commandes...**
- Mais pour faire un processeur, il faut aussi faire des registres
- Il faut aussi pouvoir faire de la mémoire

Conclusion

- Avec **des transistors** on fait **des portes logiques**
- Avec **des portes logiques** on fait **des circuits combinatoires**
- Avec **des circuits combinatoires** on fait **des UALs, des unités de commandes. . .**
- Mais pour faire un processeur, il faut aussi faire des registres
- Il faut aussi pouvoir faire de la mémoire

Solution : circuit séquentiel, mais c'est une autre histoire . . .