

Les comparateurs numériques

 Domaine d'application :
Traitement programmé de l'information

 Type de document :
Cours

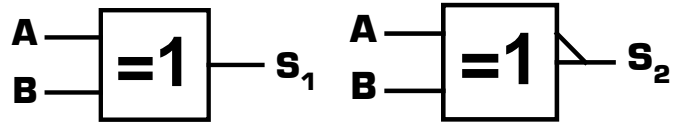
 Classe :
Terminale

Date :

I - Introduction

Cette famille de circuits logiques exploite la propriété de la fonction "OU exclusif" ou de son complément :

A	B	$S_1 = A \oplus B$	$S_2 = \overline{A \oplus B}$
0	0		
0	1		
1	0		
1	1		



La comparaison de 2 nombres binaires A [$A_0, A_1, A_2 \dots A_n$] et B [$B_0, B_1, B_2 \dots B_n$] s'effectue dans de nombreuses opérations. On peut simplement demander une détection d'égalité ou bien savoir si le nombre A est supérieur ou inférieur au nombre B.

II - Détection d'égalité

II - 1 - Principe

Les bits de même rang A_i et B_i des 2 mots à comparer sont analysés par une fonction "OU-exclusif-NON" pour donner en sortie l'indication d'égalité [$e = 1$] ou de non égalité [$e = 0$]. Les 2 mots A et B sont égaux si et seulement si tous leurs bits de même rang A_i et B_i sont égaux. En conséquence pour obtenir $A=B$, il suffit de mettre en condition "ET" les différents résultats. **Exemple :** Comparaison de 2 mots de 4 bits :

II - 2 - Variante utilisant des portes OU-Exclusif

Les bits de même rang A_i et B_i des 2 mots à comparer sont cette fois analysés par une fonction "OU-exclusif". Lorsqu'il y a égalité entre 2 bits de même rang, la sortie de la porte OU-Exclusif passe à 0. Pour détecter l'égalité entre les deux mots binaires A et B, il faut alors détecter que toutes les sorties des portes OU-Exclusif sont à 0, ce qui se fait avec une fonction OU-NON. **Exemple :** Comparaison de 2 mots de 4 bits :

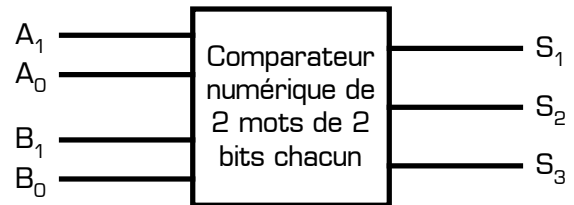
III - Comparateur donnant $A > B$, $A < B$, et $A = B$

III - 1 - Principe de la comparaison de deux nombres binaires

Soient 2 nombres binaires A et B de 2 bits chacun [$A_0 A_1$ et $B_0 B_1$] à comparer. Pour traduire les 3 possibilités, on délivre le résultat à l'aide de 3 sorties spécialisées. La table de vérité est donc la suivante (les sorties sont actives sur niveau haut).

Entrées				Sorties		
Mot A		Mot B		S_1	S_2	S_3
A_1	A_0	B_1	B_0	$A < B$	$A = B$	$A > B$
0	0	0	0			
0	0	0	1			
0	0	1	0			
0	0	1	1			
0	1	0	0			
0	1	0	1			
0	1	1	0			
0	1	1	1			
1	0	0	0			
1	0	0	1			
1	0	1	0			
1	0	1	1			
1	1	0	0			
1	1	0	1			
1	1	1	0			
1	1	1	1			

Symbole :



Equation des sorties :

$S_1 = \dots\dots\dots$

$S_2 = \dots\dots\dots$

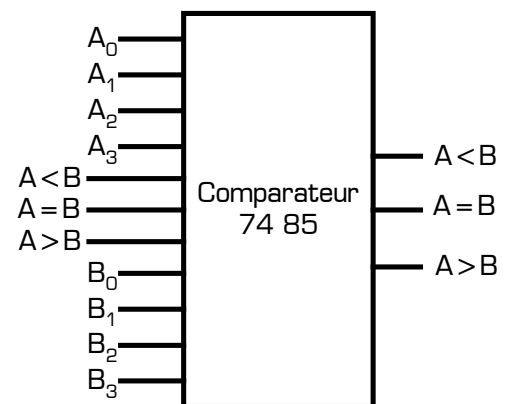
$S_3 = \dots\dots\dots$

III - 2 - Exemple de circuit intégré : le comparateur 4 bits 7485

Ce comparateur possède 3 entrées supplémentaires qui lui permet de tenir compte d'une comparaison effectuée sur des bits **de rang inférieur** et de traiter ainsi des mots de longueur quelconque en mettant en cascade plusieurs circuits. Ces 3 entrées sont appelées : entrée $A > B$, entrée $A = B$, entrée $A < B$:

Entrées				Sorties		
Mots A et B	$A > B$	$A = B$	$A < B$	$A > B$	$A = B$	$A < B$
$A > B$	1	0	0	1	0	0
$A > B$	0	1	0	1	0	0
$A > B$	0	0	1	1	0	0
$A = B$	1	0	0	1	0	0
$A = B$	0	1	0	0	1	0
$A = B$	0	0	1	0	0	1
$A < B$	1	0	0	0	0	1
$A < B$	0	1	0	0	0	1
$A < B$	0	0	1	0	0	1
$A > B$	1	1	1	1	0	0
$A = B$	1	1	1	0	1	0
$A < B$	1	1	1	0	0	1

Symbole du 7485



* Si l'on souhaite que la sortie $A = B$ passe à l'état 1 chaque fois que les deux nombres binaires sont égaux, il suffit de mettre l'entrée $A = B$ à l'état 1, l'état des entrées $A < B$ et $A > B$ n'ayant alors pas d'importance.

- * Si l'on souhaite que la sortie $A > B$ passe à l'état 1 également dans le cas où les deux nombres binaires sont égaux, il faut mettre l'entrée $A > B$ à 1 et mettre les entrées $A < B$ et $A = B$ à 0. Dans cette configuration de l'état des entrées $A > B$, $A < B$ et $A = B$, la sortie $A > B$ est à l'état 1 lorsque le nombre binaire A est supérieur au nombre binaire B ou quand ces deux nombres sont égaux. Elle indique donc si A est « supérieur ou égal » à B.
- * De même, en mettant l'entrée $A < B$ à l'état 1 et les entrées $A > B$ et $A = B$ à l'état 0, la sortie $A < B$ indique si le nombre binaire A est « inférieur ou égal » au nombre binaire B.

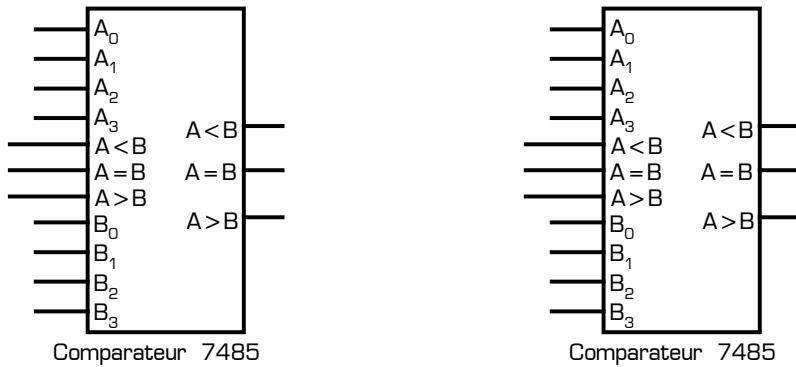
En mettant en cascade deux comparateurs 7485, on peut comparer deux nombres de 8 bits. Il suffit de relier la sortie $A = B$ du premier comparateur à l'entrée $A = B$ du second et de faire de même avec les sorties $A > B$ et $A < B$.

Voyons le principe de mise en cascade de deux circuits 7485 pour comparer deux mots binaires X et Y de 8 bits chacun [deux octets]. Les 8 bits de l'octet X sont X_0 à X_7 [X_0 étant le bit de poids faible et X_7 le bit de poids forts] et les 8 bits de l'octet Y sont Y_0 à Y_7 [Y_0 le bit de poids faible et Y_7 le bit de poids forts].

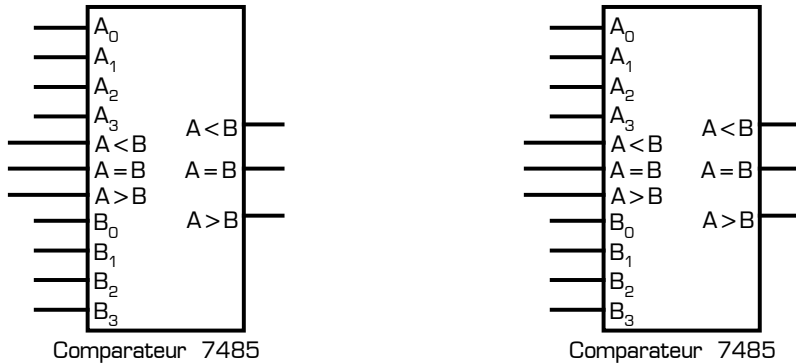
Le premier circuit compare les poids faibles de X avec les poids faibles de Y. Le résultat de cette comparaison est transmis aux entrées $A < B$, $A = B$ et $A > B$ du deuxième circuit. Celui-ci compare les poids forts de X avec les poids forts de Y et, éventuellement en fonction du résultat de la comparaison des bits de poids faibles de X et Y, indique sur ses sorties $A > B$, $A = B$ et $A < B$ le résultat de la comparaison des nombres X et Y.

Analysons ce principe de mise en cascade dans 3 cas bien précis :

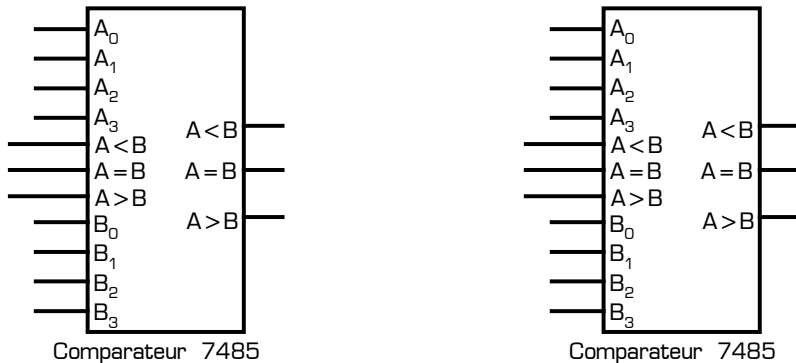
Premier cas : $X = \dots\dots\dots[10] \equiv \dots\dots\dots[2]$ et $Y = \dots\dots\dots[10] \equiv \dots\dots\dots[2]$



Second cas : $X = \dots\dots\dots[10] \equiv \dots\dots\dots[2]$ et $Y = \dots\dots\dots[10] \equiv \dots\dots\dots[2]$



Troisième cas : $X = \dots\dots\dots[10] \equiv \dots\dots\dots[2]$ et $Y = \dots\dots\dots[10] \equiv \dots\dots\dots[2]$



IV – Exercice : conception d'un comparateur numérique 3 bits

Complétez ci-dessous la table de vérité d'un comparateur numérique 3 bits comparant les mots binaires A et B sachant qu'une sortie vaut 1 si la condition qu'elle représente est vraie [exemple : $S_3 = 1$ si $A > B$]. Proposez ensuite une équation simplifiée pour chacune des 3 sorties S_1 , S_2 et S_3 du comparateur en utilisant la méthode de votre choix [algèbre de Boole, tableaux de Karnaugh, extraction directe depuis la table de vérité, ou simple analyse du problème]. Vérifiez à l'occasion sur ordinateur l'exactitude de vos propositions avec un simulateur en testant tous les cas.

Entrées								Sorties		
Mot binaire A				Mot binaire B				A < B	A = B	A > B
A ₂	A ₁	A ₀	Valeur décimale	B ₂	B ₁	B ₀	Valeur décimale	S ₁	S ₂	S ₃
0	0	0		0	0	0				
0	0	0		0	0	1				
0	0	0		0	1	0				
0	0	0		0	1	1				
0	0	0		1	0	0				
0	0	0		1	0	1				
0	0	0		1	1	0				
0	0	0		1	1	1				
0	0	1		0	0	0				
0	0	1		0	0	1				
0	0	1		0	1	0				
0	0	1		0	1	1				
0	0	1		1	0	0				
0	0	1		1	0	1				
0	0	1		1	1	0				
0	0	1		1	1	1				
0	1	0		0	0	0				
0	1	0		0	0	1				
0	1	0		0	1	0				
0	1	0		0	1	1				
0	1	0		1	0	0				
0	1	0		1	0	1				
0	1	0		1	1	0				
0	1	0		1	1	1				
0	1	1		0	0	0				
0	1	1		0	0	1				
0	1	1		0	1	0				
0	1	1		0	1	1				
0	1	1		1	0	0				
0	1	1		1	0	1				
0	1	1		1	1	0				
0	1	1		1	1	1				
1	0	0		0	0	0				
1	0	0		0	0	1				
1	0	0		0	1	0				
1	0	0		0	1	1				
1	0	0		1	0	0				
1	0	0		1	0	1				
1	0	0		1	1	0				
1	0	0		1	1	1				
1	0	1		0	0	0				
1	0	1		0	0	1				
1	0	1		0	1	0				
1	0	1		0	1	1				
1	0	1		1	0	0				
1	0	1		1	0	1				
1	0	1		1	1	0				
1	0	1		1	1	1				
1	1	0		0	0	0				
1	1	0		0	0	1				
1	1	0		0	1	0				
1	1	0		0	1	1				
1	1	0		1	0	0				
1	1	0		1	0	1				
1	1	0		1	1	0				
1	1	0		1	1	1				
1	1	1		0	0	0				
1	1	1		0	0	1				
1	1	1		0	1	0				
1	1	1		0	1	1				
1	1	1		1	0	0				
1	1	1		1	0	1				
1	1	1		1	1	0				
1	1	1		1	1	1				