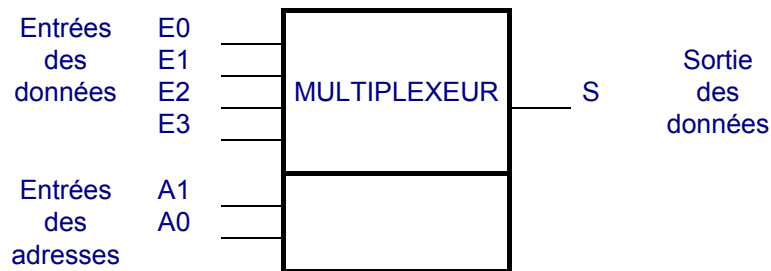


**MULTIPLEXEURS – ADDITIONNEURS- COMPAREURS
LEÇON 10 -**

Multiplexeurs

Réalisation d'un multiplexeur

Un multiplexeur ou sélecteur de données est un commutateur qui va pouvoir, à l'aide de n bits d'adresse, sélectionner une de ses 2^n entrées et la mettre en communication avec sa sortie. Le schéma ci-dessous donne une image d'un multiplexeur 4 voies (E3 à E0) vers une (S) sélectables à l'aide des bits d'adresse A1 et A0.



Pour concevoir cette fonction, il serait possible d'établir la table de vérité du circuit en tenant le raisonnement suivant:

- (1) - Lorsque $A1, A0 = 00$ si $E0 = 0 \rightarrow S = 0$, si $E0 = 1 \rightarrow S = 1$ et ceci quelles que soient les entrées E1, E2, E3
- (2) - Lorsque $A1, A0 = 01$ si $E1 = 0 \rightarrow S = 0$, si $E1 = 1 \rightarrow S = 1$ et ceci quelles que soient les entrées E0, E2, E3
- Etc...

La sortie S est donc tributaire de 6 entrées, l'établissement d'un tableau de Karnaugh nécessiterait 2^6 cases alors que la table de vérité ne comportera que 8 combinaisons.

Il peut être procédé à une mise en équation plus simple ainsi la description (1) ci-dessus peut s'énoncer:

- Lorsque $A1, A0 = 00$ $S = E0$ on peut en déduire dans ces conditions $S = \overline{A1} \cdot \overline{A0} \cdot E0$

De même pour la condition (2)

- Lorsque $A1, A0 = 01$ $S = E1$ on peut en déduire dans ces conditions $S = \overline{A1} \cdot A0 \cdot E1$

La sélection de l'entrée E2 pourra s'écrire:

- Lorsque $A1, A0 = 10$ $S = E2$ on peut en déduire dans ces conditions $S = A1 \cdot \overline{A0} \cdot E2$

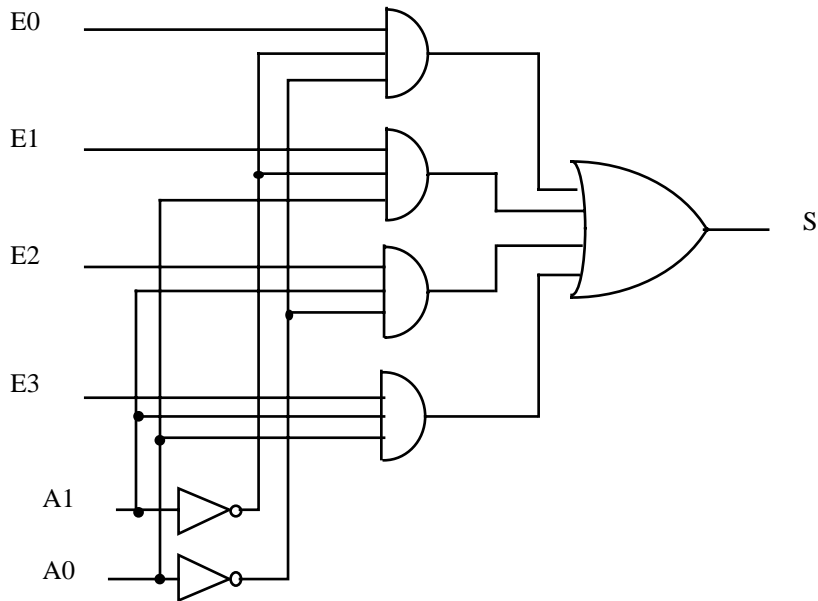
Et E3:

- Lorsque $A1, A0 = 11$ $S = E3$ on peut en déduire dans ces conditions $S = A1 \cdot A0 \cdot E3$

L'ensemble de ces conditions réunies dans un OU logique permet d'écrire l'équation générale de S:

$$S = (\overline{A1} \cdot \overline{A0} \cdot E0) + (\overline{A1} \cdot A0 \cdot E1) + (A1 \cdot \overline{A0} \cdot E2) + (A1 \cdot A0 \cdot E3)$$

Ce qui aboutit au logigramme ci dessous:



On remarquera sur ce logigramme que les portes ET fonctionnent comme des commutateurs dont la validation est assurée par les entrées d'adresse A1 et A0. Il est possible afin de mieux rationaliser l'étude d'un tel circuit de le dissocier en deux fonctions, l'une assurée par les portes ET et OU qui constituent la commutation et l'autre par un décodage des adresses venant sélectionner les portes ET.

Étude de la fonction **Décodage des adresses**.

Table de vérité

Adresse		Sélection			
A1	A0	s3	s2	s1	s0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

Equations déduites:

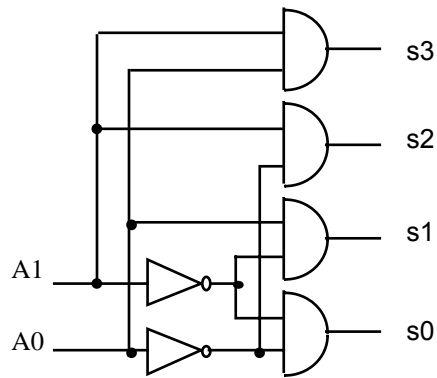
$$s_0 = \overline{A_1} \cdot \overline{A_0}$$

$$s_1 = \overline{A_1} \cdot A_0$$

$$s_2 = A_1 \cdot \overline{A_0}$$

$$s_3 = A_1 \cdot A_0$$

Schéma du décodeur



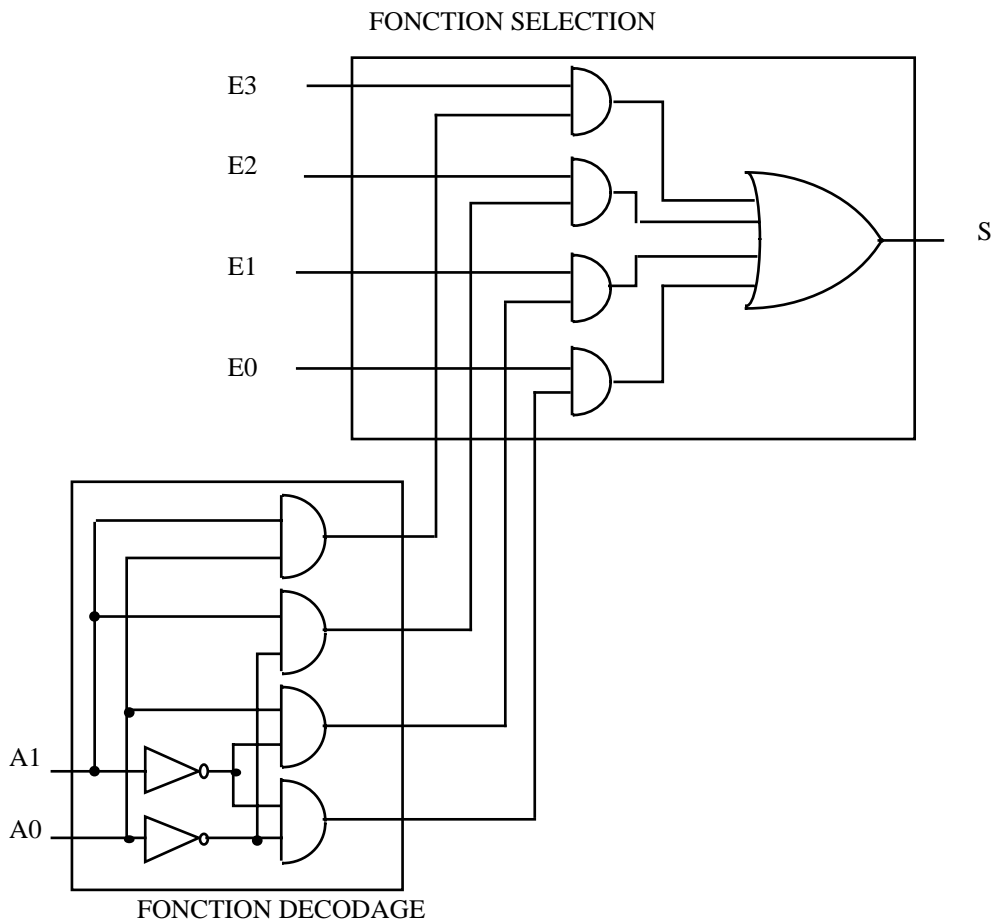
FONCTION DECODAGE

Le décodeur ainsi obtenu qui va assurer la sélection des entrées est un décodeur binaire Décimal. Celui-ci va placer un état actif sur la sortie dont la Numérotation décimale correspond au nombre binaire de l'adresse.

Exemple:

l'adresse 11 fait apparaître un 1 sur S3

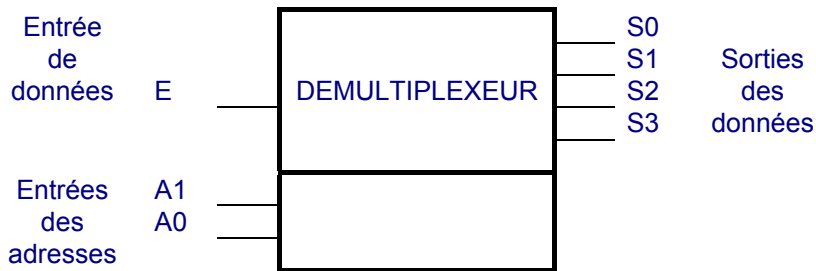
Le schéma ci-dessous fait apparaître le multiplexeur réalisé à l'aide de deux sous-ensembles, d'une part le décodeur d'adresses et d'autre part le sélecteur d'entrée dont les portes ET qui assurent la sélection sont validées par les sorties du décodeur.



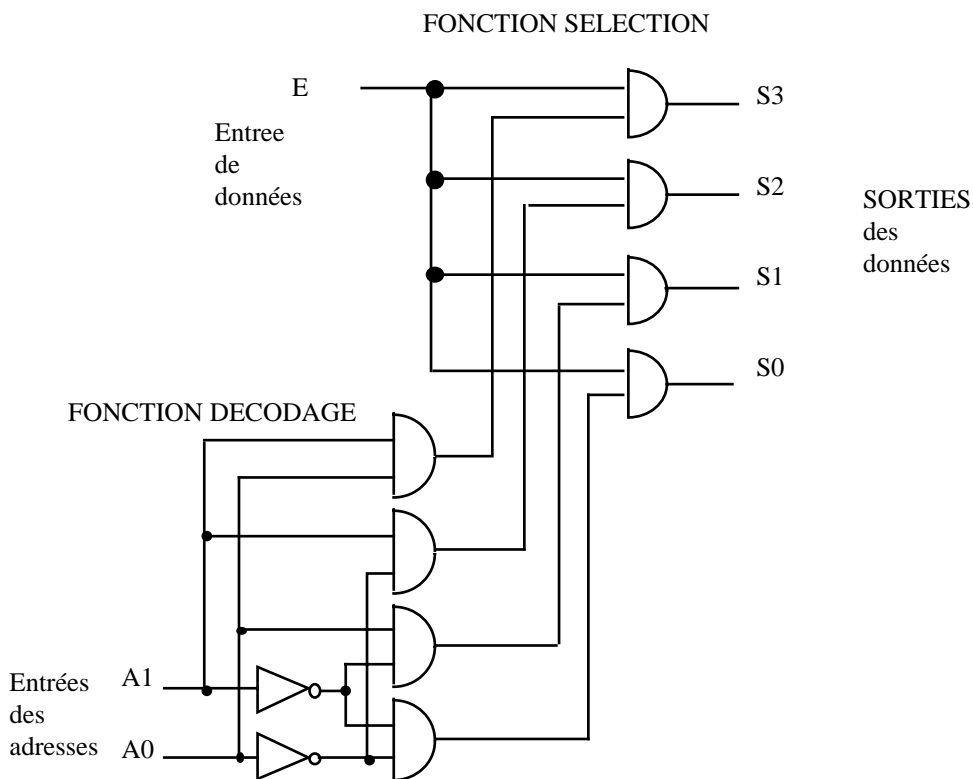
Réalisation d'un démultiplexeur

Un démultiplexeur ou répartiteur de données est un commutateur qui va pouvoir, à l'aide de n bits d'adresse, aiguiller la donnée présente sur son entrée vers l'une de ses 2^n sorties.

Le schéma ci-dessous donne une image d'un démultiplexeur une entrée (E) vers 4 sorties (S3 à S0) sélectables à l'aide des bits d'adresse A1 et A0.



Nous procéderons comme précédemment en séparant les fonctions



On remarquera que si l'entrée E est placée à 1 l'ensemble du circuit se conduit comme un décodeur. Le démultiplexeur est de ce fait souvent assimilé à un décodeur binaire décimal.

Réalisation d'un additionneur

Posons-nous le problème de la réalisation d'un circuit capable d'additionner deux nombres binaires de deux bits résultat sur trois bits.

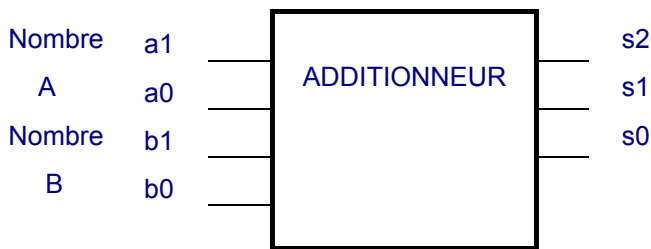
$$A + B = S$$

soit en décomposant les nombres A, B, S

$$\begin{array}{r} \\ + \\ \hline s2 \end{array}$$

Il serait possible de procéder comme pour la réalisation d'un transcodeur. Nous aurions deux entrées pour le nombre A, deux entrées pour le nombre B et trois sorties pour S.

Nous pourrions le schématiser de la façon suivante:



Et sa table de vérité suivante:

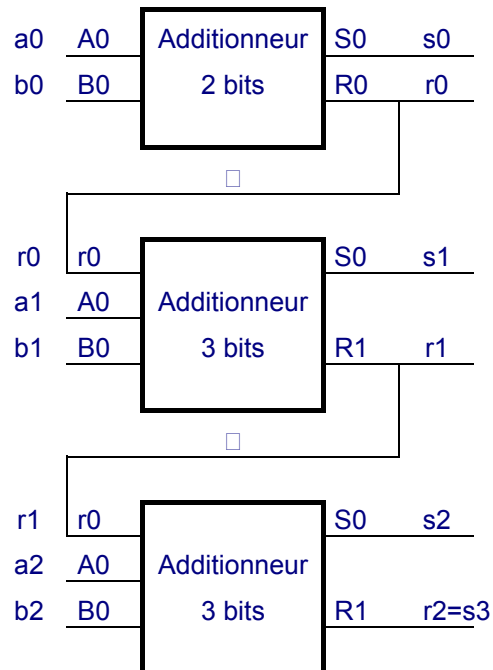
A		B		S		
a1	a0	b1	b0	s2	s1	s0
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	0	0	1	0
0	0	1	1	0	1	1
0	1	0	0	0	0	1
0	1	0	1	0	1	0
0	1	1	0	0	1	1
0	1	1	1	1	0	0
1	0	0	0	0	1	0
1	0	0	1	0	1	1
1	0	1	0	1	0	0
1	0	1	1	1	0	1
1	1	0	0	0	1	1
1	1	0	1	1	0	0
1	1	1	0	1	0	1
1	1	1	1	1	1	0

Nous laisserons au lecteur le soin de réaliser les tableaux de Karnaugh et de s'apercevoir qu'il y aura de grandes difficultés à mettre en évidence une structure répétitive qui permettrait d'extrapoler un additionneur deux fois trois bits ou deux fois quatre bits.

Pourtant, la répétitivité est indiscutable aussi nous pourrions nous poser le problème différemment en pensant à la façon dont nous opérons lorsque nous réalisons cette opération manuellement

$$\begin{array}{r} a1 \ a0 \\ + b1 \ b0 \\ \hline s2 \ s1 \ s0 \end{array}$$

Nous additionnons d'abord a_0 à b_0 ce qui nous donne le terme s_0 et éventuellement un report r_0 , puis nous additionnons a_1 à b_1 et au report r_0 , ce qui nous donne s_1 et un report r_1 (qui est peut être nul) qui devient le terme résultat s_2 . Une tranche suivante, composée de deux termes a_2 et b_2 serait traitée rigoureusement de la même façon que la précédente.



Nous voyons que nous avons besoin de réaliser deux circuits, le premier que nous nommerons un **demi additionneur**, capable d'additionner deux bits et le second qui additionnera deux bits en tenant compte du report de l'opération précédente que l'on appelle un **additionneur complet**.

Réalisation d'un demi additionneur:

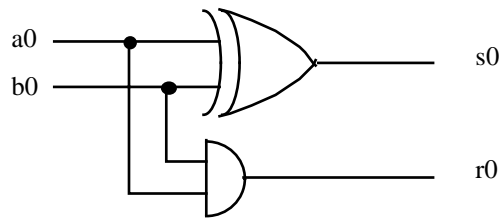
Table de vérité:

a0	b0	r0	s0
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Les équations se déduisent directement de cette table:

$$\begin{aligned} r_0 &= a_0 \cdot b_0 \\ s_0 &= a_0 \oplus b_0 \end{aligned}$$

On en déduit le schéma ci-dessous:



Réalisation d'un additionneur complet :

Table de vérité:

r0	a1	b1	r1	s1
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

tableau de r1

tableau de s1

a1 b1→	00	01	11	10
r0↓	0	0	1	0
1	0	1	1	1

a1 b1→	00	01	11	10
r0↓	0	1	0	1
1	1	0	1	0

les équations obtenues :

$$s1 = r0 \oplus a1 \oplus b1$$

$$r1 = r0b1 + a1b1 + r0a1$$

Il est facile de montrer également que ce circuit peut être réalisé à l'aide de deux demi additionneurs. En effet le tableau de karnaugh de r1 peut être obtenu en faisant un OU logique entre les deux tableaux ci dessous:

Équation à déterminer

tableau de l'équation a1.r0

a1 b1→	00	01	11	10
r0↓	0	0	1	0
1	0	1	0	0

a1 b1→	00	01	11	10
r0↓	0	0	0	0
1	0	0	1	1

et le tableau de gauche par un ET entre les deux tableaux suivants:

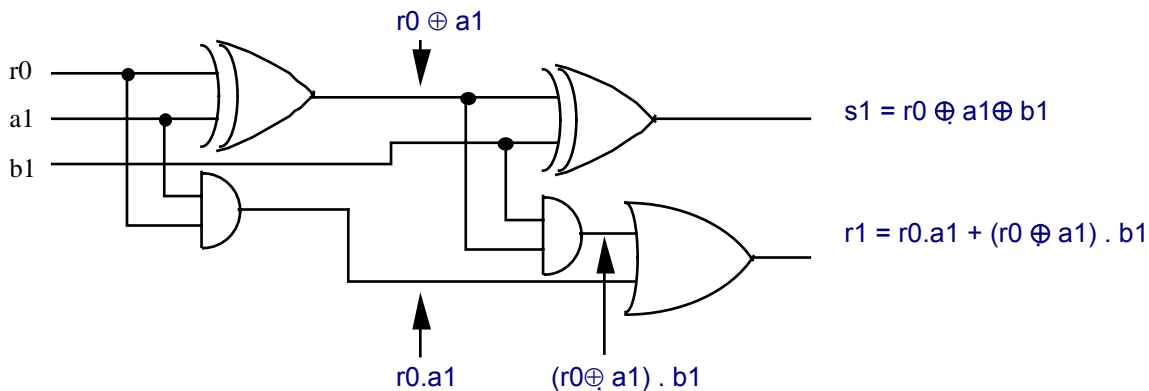
		$(r0 \oplus a1)$			
$a1 \ b1 \rightarrow$	$r0 \downarrow$	00	01	11	10
0	0	0	0	1	1
1	0	1	1	0	0

		$b1$			
$a1 \ b1 \rightarrow$	$r0 \downarrow$	00	01	11	10
0	0	0	1	1	0
1	0	0	1	1	0

d'où l'équation:

$$r1 = (r0 \oplus a1) \cdot b1 + r0 \cdot a1$$

et le schéma:



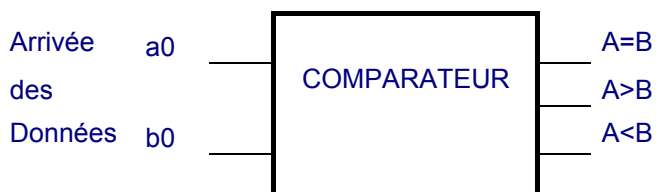
Réalisation d'un comparateur binaire

Nous avons vu au cours des précédentes études diverses techniques qui nous ont permis de réaliser nos systèmes combinatoires sans nous lancer dans des séries de tableaux de Karnaugh fastidieux.

- mise en évidence de structures répétitives rendant possible l'extrapolation (transcodeurs BN/BR et BR/BN)
- séparation des fonctions permettant l'étude séparée des diverses fonctions (multiplexeur, démultiplexeur)
- présentation du problème permettant la mise en "cascade" de circuits élémentaires (additionneur)

Nous allons à travers l'étude de comparateurs de grandeurs binaires montrer un cheminement qui permettra de rendre cascadables des cellules qui au départ ne le sont pas.

Réalisons une cellule capable de comparer deux bits. Cette cellule sera munie de deux entrées $a0, b0$ et de trois sorties $a=b$ qui passe à 1 lorsque $a0=b0$, $a>b$ qui passe à 1 lorsque a est supérieur à b et $a<b$.



Et devra répondre à la table de vérité suivante:

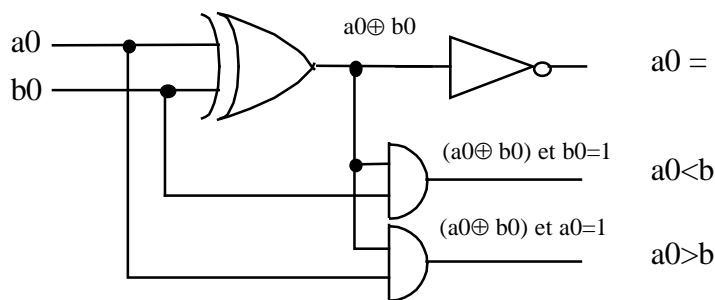
ENTREES		SORTIES		
a0	b0	A=B	A>B	A<B
0	0	1	0	0
0	1	0	0	1
1	0	0	1	0
1	1	1	0	0

Le problème est simple à résoudre à l'aide de tableaux de karnaugh ou même par une mise en équation directe cependant nous opèrerons par raisonnement:

Nous savons que l'opérateur dilemme donne 1 sur sa sortie lorsque ses entrées sont à des états logiques différents, c'est un détecteur de différence. Un dilemme complémenté nous permettra donc de réaliser la première fonction $A=B$.

Maintenant en ce qui concerne les deux autres sorties nous nous tiendrons le raisonnement suivant: Si la sortie de l'opérateur ou exclusif fournit un 1 c'est que a_0 est différent de b_0 donc l'un est à 0 l'autre à 1. Si, simultanément, a_0 est à 1, alors on peut dire que $a_0 > b_0$ de même si b_0 est à 1 $a_0 < b_0$.

Nous déduirons de ce raisonnement la cellule élémentaire ci-dessous:



Il faut maintenant définir la façon de mettre en cascade ces cellules en tenant compte de la hiérarchie. L'une recevra les bits a_0 et b_0 et l'autre les bits a_1 et b_1 il est évident que si l'une nous dit $a_0 < b_0$ et l'autre $a_1 > b_1$, c'est la cellule qui compare les bits ayant le poids le plus fort qui doit emporter la décision. Les portes qui vont nous permettre de respecter la hiérarchie devront être incluses dans la cellule élémentaire et toutes les cellules devant rester identiques, ces portes devront apparaître dans les deux cellules.

L'organigramme de la page suivante reproduit le raisonnement que nous nous tenons lorsque nous comparons deux nombres binaires de deux bits.

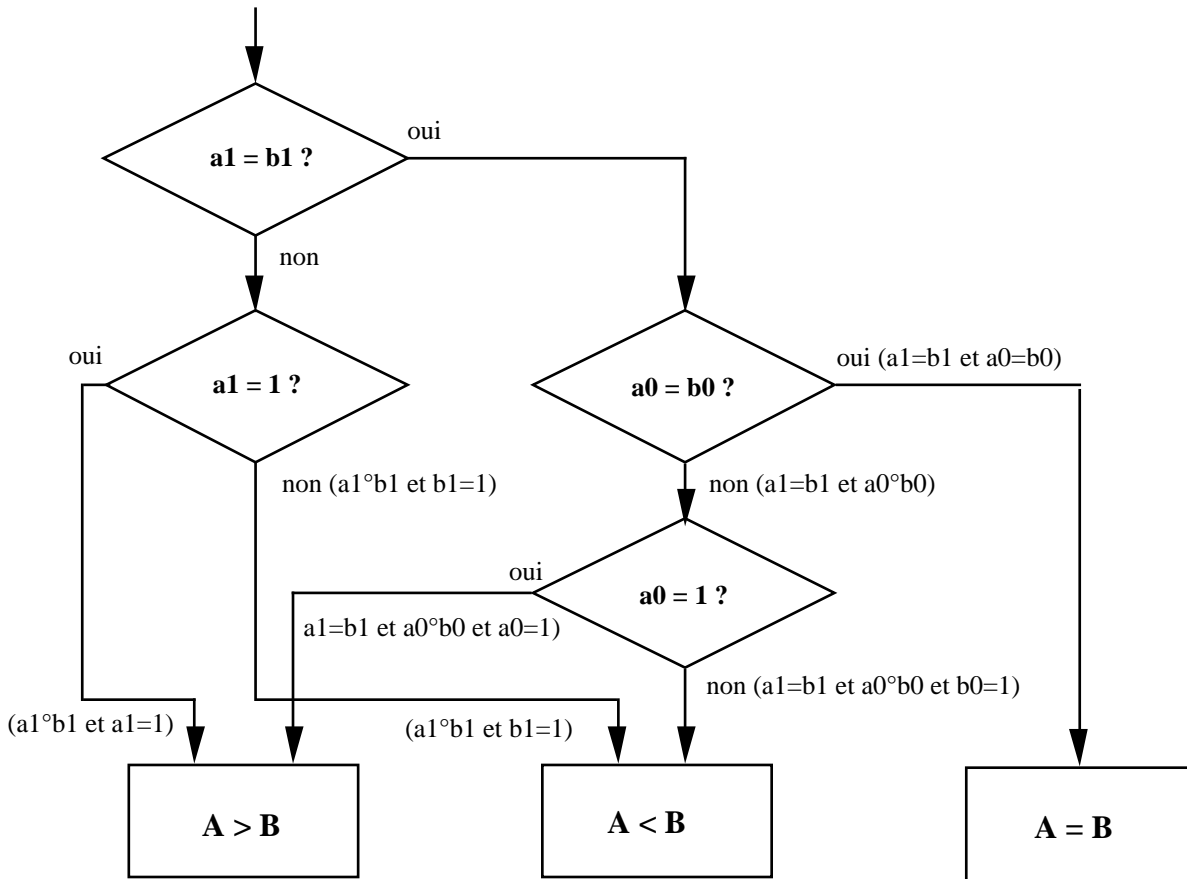
Pour établir les équations il suffit de suivre les cheminements, des conditions en série seront satisfaites par des ET logiques, deux cheminements qui aboutissent à un résultat seront matérialisés par un OU.

Ainsi l'équation de $A = B$ sera **($a_1 = b_1$) et ($a_0 = b_0$)** soit un et entre les deux sorties $a_0 = b_0$ de la première cellule et $a_1 = b_1$ de la seconde.

La conclusion $A < B$ est obtenue par deux cheminements. L'équation sera: **($a_1 = b_1$ et $a_0 \neq b_0$ et $b_0 = 1$) ou ($a_1 \neq b_1$ et $b_1 = 1$)** on remarquera que l'expression ($a_0 \neq b_0$ et $b_0 = 1$) correspond à la sortie $a_0 < b_0$ et que ($a_1 \neq b_1$ et $b_1 = 1$) correspond à la sortie $a_1 < b_1$

La conclusion $A > B$ est de même type que la précédente nous obtenons: **($a_1 = b_1$ et $a_0 \neq b_0$ et $a_0 = 1$) ou ($a_1 \neq b_1$ et $a_1 = 1$)**

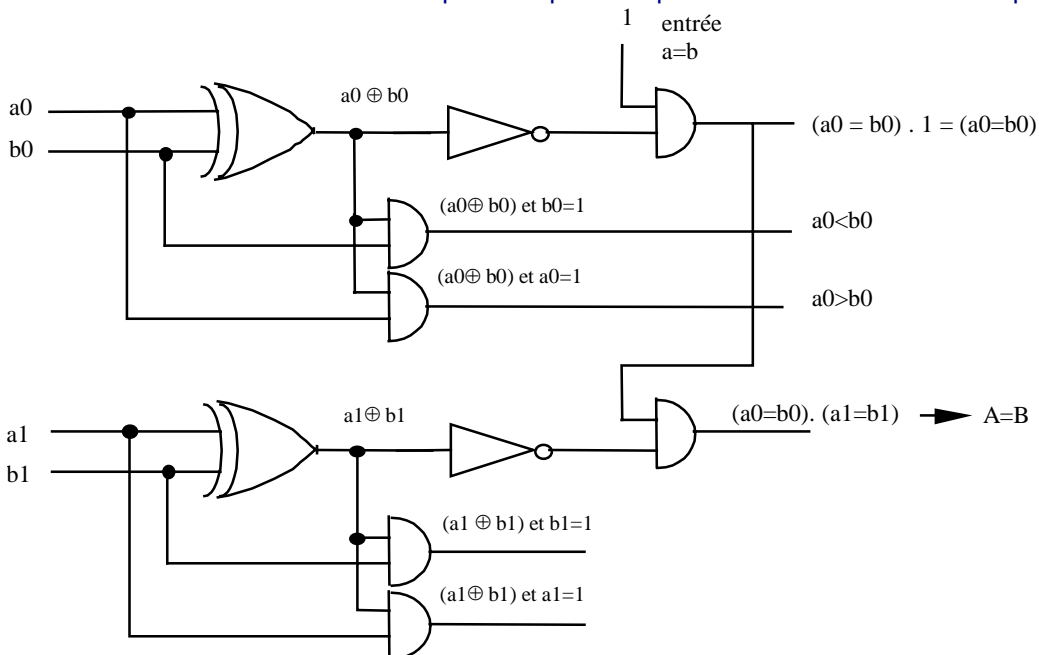
Il est maintenant nécessaire de procéder aux modifications des cellules afin de permettre leur mise en cascade.(suite du texte sous l'organigramme de la page suivante)



Dans ce logigramme le symbole \circ doit être remplacé par \neq ainsi $a0 \circ b0$ devient $a0 \neq b0$

Mise en cascade des sorties $a=b$

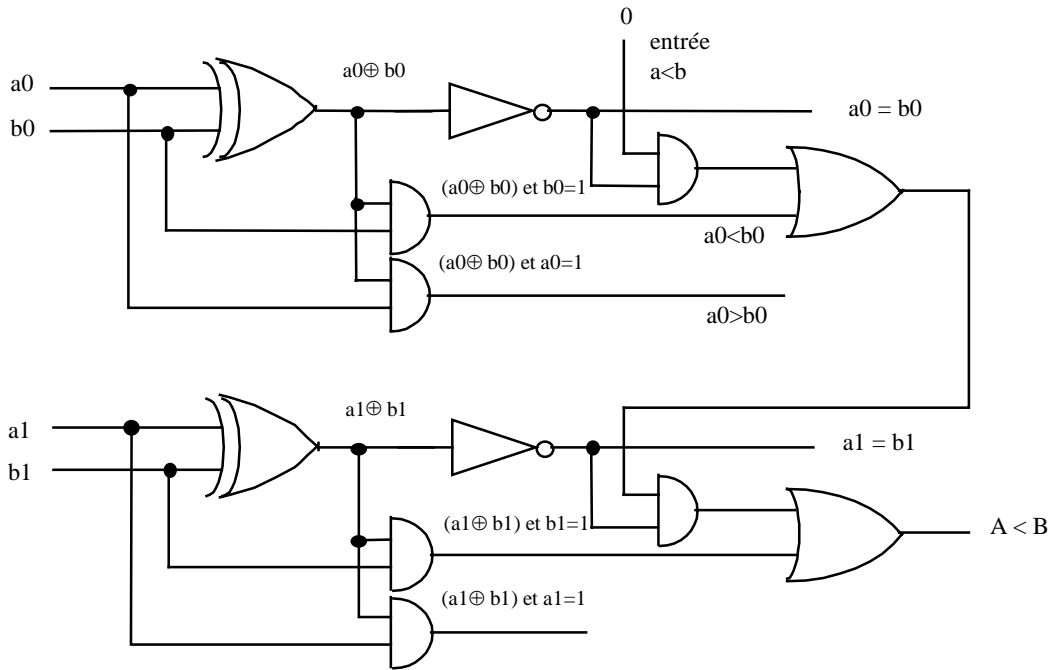
La condition nécessaire à la conclusion $A=B$ est $(a0=b0)=1$ et $(a1=b1)=1$, il est donc nécessaire de faire entrer ces deux lignes dans un ET, cependant, les cellules devant rester identiques la même porte sera placée dans l'autre cellule et une même entrée sera ménagée. Il conviendra de polariser correctement cette entrée sans utilisation de telle sorte quelle ne perturbe pas le fonctionnement du comparateur.



Voyons maintenant la mise en cascade des sorties $a < b$

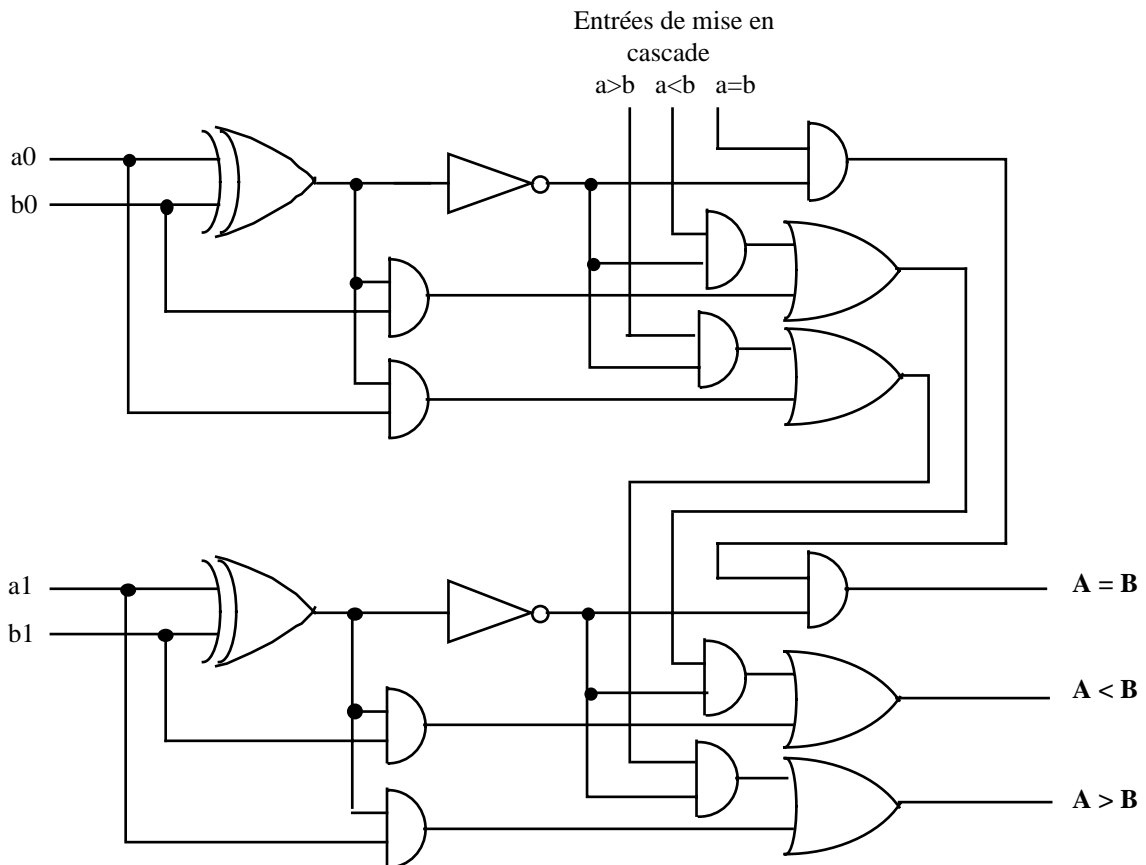
L'équation est $(a1 = b1 \text{ et } a0 \neq b0 \text{ et } b0 = 1)$ ou $(a1 \neq b1 \text{ et } b1 = 1)$

or nous avons déjà remarqué que l'expression $(a0 \neq b0 \text{ et } b0 = 1)$ correspond à la sortie $a0 < b0$ et que $(a1 \neq b1 \text{ et } b1 = 1)$ correspond à la sortie $a1 < b1$. Ce qui nous amène au schéma suivant:



Dans un souci de clarté, nous n'avons fait figurer que la mise en cascade des sorties $a < b$ ce qui a conduit à ajouter à chaque cellule une porte ET et une porte OU. Il est facile de vérifier que si nous plaçons un 0 sur l'entrée de mise en cascade de la cellule supérieure, la porte ET de la cellule inférieure reçoit bien d'une part la condition $a_0 \neq b_0$ et $b_0 = 1$ sur une entrée et $a_1 = b_1$ sur l'autre, le ET entre ces deux conditions est injecté dans une porte OU qui reçoit sur son autre entrée $a_1 \neq b_1$ et $b_1 = 1$ ce qui respecte la condition globale énoncée plus haut.

En procédant de la même façon pour $a > b$ nous obtenons le schéma suivant:



Exercices**multiplexeur**

a/ Modifier le schéma du multiplexeur de la page 74 en lui adjoignant une entrée de validation. Cette entrée mise à 1 autorise le fonctionnement normal du multiplexeur, à 0 la sortie du multiplexeur est bloquée à 0.

b/ Faire la même modification sur le démultiplexeur de la page 75

c/ A l'aide de 5 démultiplexeurs modifiés comme précédemment (utiliser la représentation symbolique) réaliser un démultiplexeur 1 voie vers 16

d/ Un multiplexeur peut être utilisé pour matérialiser un tableau de Karnaugh en appliquant les variables sur les entrées d'adresse et en câblant correctement les entrées. à l'aide d'un multiplexeur 8 vers 1 matérialiser les tableaux ci dessous.

Préciser le câblage en utilisant une représentation symbolique.

XY →				
Z ↓	00	01	11	10
0	1	0	1	0
1	0	1	1	0

XY →				
ZT ↓	00	01	11	10
00	1	1	0	1
01	0	1	1	0
11	1	0	0	0
10	0	0	1	1

décodeur

a/ Réaliser à l'aide d'opérateurs OU exclusifs un circuit capable d'adjoindre à des nombres de 4 bits fournis en code binaire naturel les bits de parité de Hamming

b/ Supposons un OU exclusif à deux entrées dont les broches sont appelées I, C pour les entrées et O pour la sortie. Appelons C l'entrée de commande, donner l'équation de O en fonction de I pour C=1 et C=0. Que peut-on déduire du comportement de l'opérateur?

c/ Pour corriger un nombre de 7 bits incluant les 3 bits de parité de Hamming, on décompose le correcteur trois fonction:

- Tests ($\gamma \beta \alpha$)
- Décodage
- Inversion

réaliser ces trois fonctions et faire un schéma global à l'aide de représentations symboliques.