

J. BEZY - ESIR1 - ARC1

ARC1 - Cours n°6

CIRCUITS COMBINATOIRES

J. BEZY - ESIR1 - ARC1

A. CIRCUITS LOGIQUES

1. Correspondance logique / électrique
2. Divers types de broches (Alimentation, entrées, sorties)
3. Sorties normales et sorties 3 états
4. Assemblage de circuits logiques (règles d'assemblage, cas des 3 états)

B. CIRCUITS COMBINATOIRES

1. Comparateurs
2. Aiguillage d'information (multiplexeurs/démultiplexeurs)
3. Transformation de codes (codeurs/décodeurs)
4. Circuits arithmétiques (additionneurs binaires/DCB)

5. Contrôle et calcul de parité (code détecteur erreur)

A. Circuits logiques

CircuitsLogiquesCircuitsTroisEtats

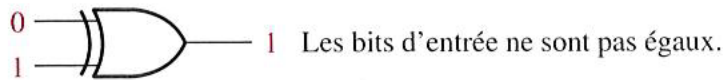
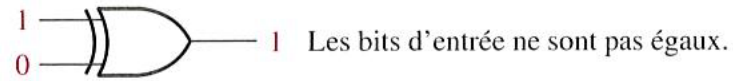
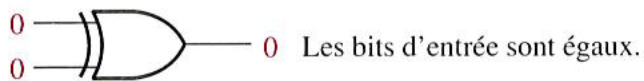
J. BEZY - ESIR1 - ARC1

1. Compareurs

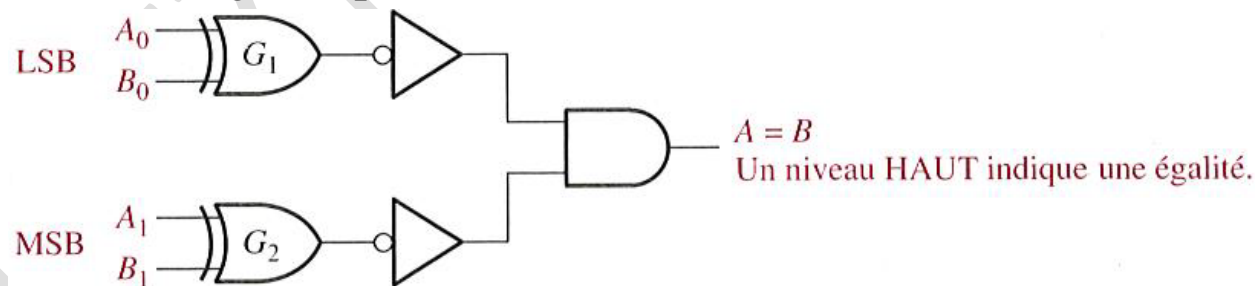
Fonction : comparer les grandeurs de deux quantités binaires \Rightarrow connaître la relation existant entre les deux quantités (ex : savoir si deux nombres sont égaux)

1.1 Egalité

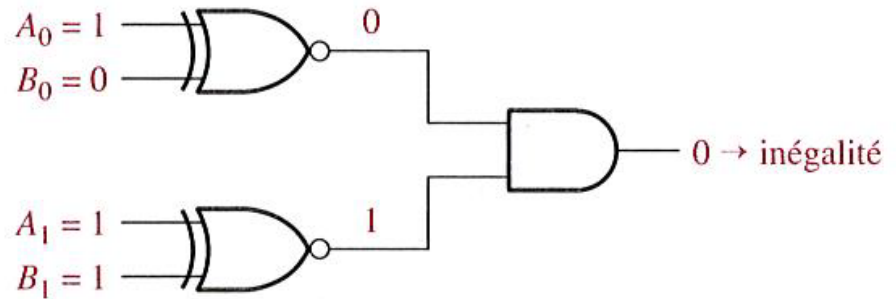
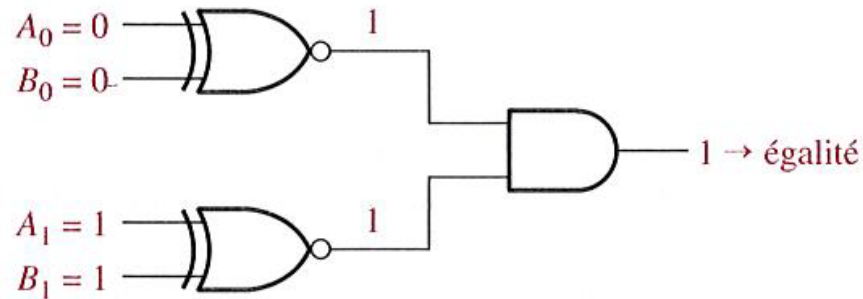
- Comparer deux nombres de 1 bit : Porte OU EXCLUSIF (sortie vaut 1 si les deux bits sont différents et 0 si égaux)



- Comparer deux nombres de 2 bits : deux portes OU EXCLUSIF, une pour bits de poids faibles et une pour bits de poids fort. Montage à 5 portes :

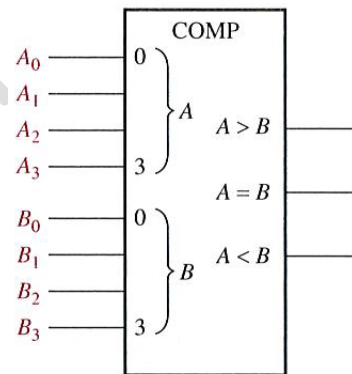


- Exemple (les portes OU EXCLUSIF suivies des inverseurs sont remplacées par des NON OU EXCLUSIF) – Comparaison des nombres 10 à 10, et 10 à 11.



1.2 Inégalité

- Les CI de comparaison ont souvent 3 sorties :
 - ✓ une sortie Egalité ($A=B$)
 - ✓ une sortie Supériorité ($A>B$)
 - ✓ une sortie Infériorité ($A<B$)



Symbole logique d'un comparateur 4 bits

- Méthode de comparaison :

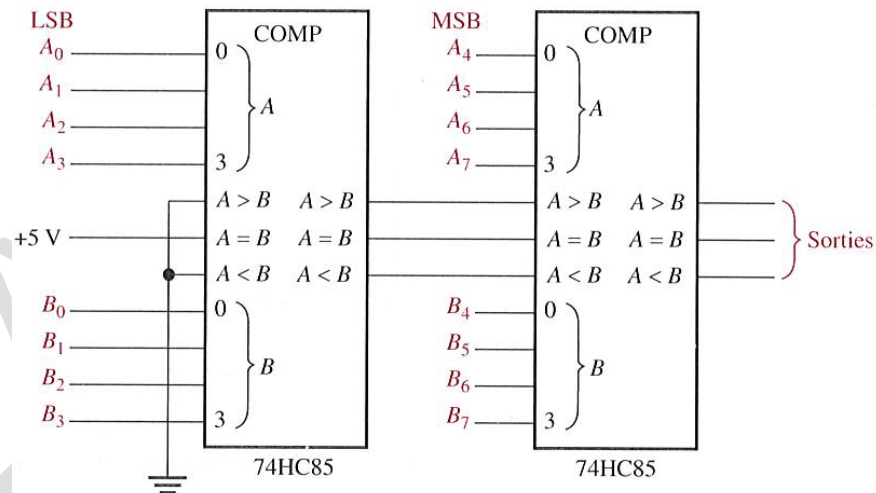
- ✓ Comparer les bits de poids forts
- ✓ Si inégalité, la relation entre les deux nombres est établie (relations aux autres rangs ignorées)
- ✓ Contient aussi 3 entrées permettant la mise en cascade de plusieurs comparateurs (nombres de plus de 4 bits)
- ✓ Si égalité au rang de poids fort, comparer bits rangs plus faibles

Nombres à comparer				Entrées de cascade			Sorties		
a3,b3	a2,b2	a1,b1	a0,b0	A>B	A<B	A=B	A>B	A<B	A=B
a3>b3	x	x	x	x	x	x	1	0	0
a3<b3	x	x	x	x	x	x	0	1	0
a3=b3	a2>b2	x	x	x	x	x	1	0	0
a3=b3	a2<b2	x	x	x	x	x	0	1	0
a3=b3	a2=b2	a1>b1	x	x	x	x	1	0	0
a3=b3	a2=b2	a1<b1	x	x	x	x	0	1	0
a3=b3	a2=b2	a1=b1	a0>b0	x	x	x	1	0	0
a3=b3	a2=b2	a1=b1	a0<b0	x	x	x	0	1	0
a3=b3	a2=b2	a1=b1	a0=b0	1	0	0	1	0	0

a3=b3	a2=b2	a1=b1	a0=b0	0	1	0	0	1	0
a3=b3	a2=b2	a1=b1	a0=b0	0	0	1	0	0	1

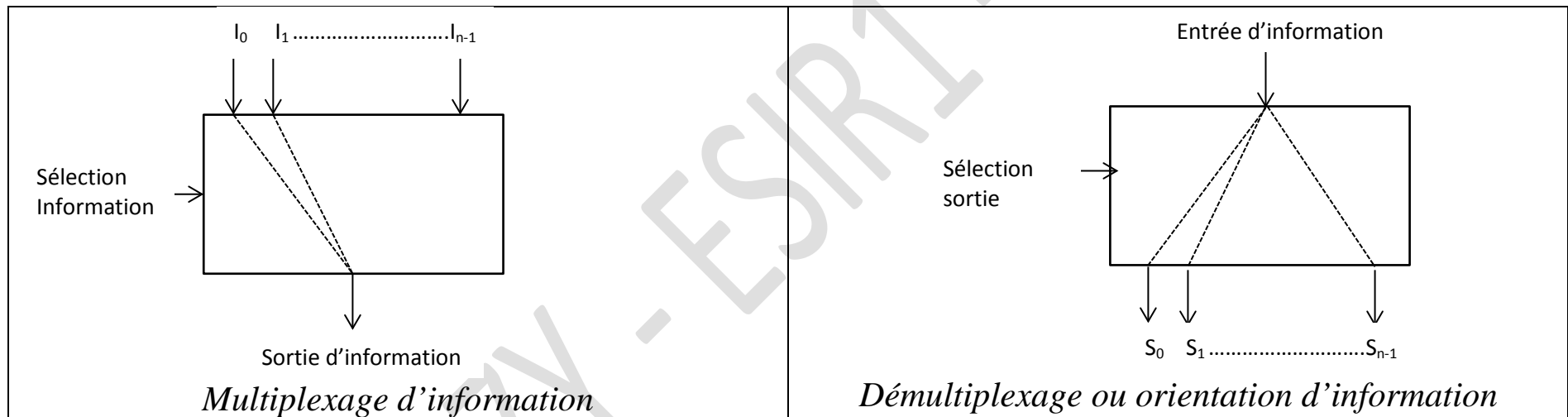
- CI de Comparateurs : ex : 74HC85

✓ Exemple : comparaison de deux nombres de 8 bits :



2. Aiguillage d'information (multiplexeur / démultiplexeur)

- Schémas d'aiguillage d'information :



2.1 Multiplexeur

- Fonction : acheminer les informations numériques de plusieurs sources sur une seule ligne (destination commune)

- Plusieurs lignes de données d'entrée / une seule ligne de sortie
- Entrées de sélection de données
- Principe et schéma :

Adresse de sélection	Sortie
CBA	
0 0 0	E ₀
0 0 1	E ₁
0 1 0	E ₂
0 1 1	E ₃
1 0 0	E ₄
1 0 1	E ₅
1 1 0	E ₆
1 1 1	E ₇

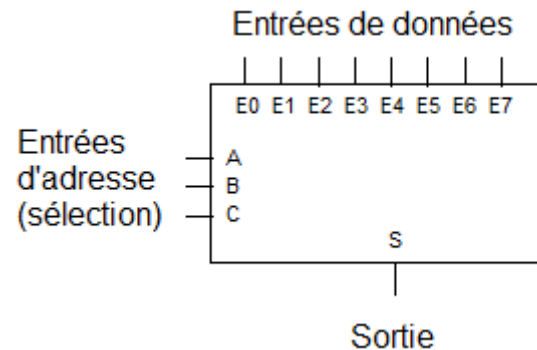


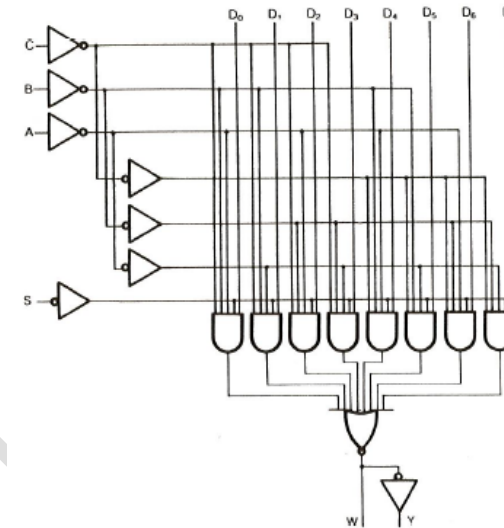
Schéma symbolique d'un multiplexeur à 8 entrées

La sortie S présente l'état de l'entrée E_i sélectionnée par l'adresse CBA₂=i₁₀

$$S = \bar{C}\bar{B}\bar{A} E_0 + \bar{C}\bar{B}A E_1 + \bar{C}B\bar{A} E_2 + \bar{C}BA E_3 + C\bar{B}\bar{A} E_4 + C\bar{B}A E_5 + CB\bar{A} E_6 + CBA E_7$$

*Existe sous forme de Circuit
Intégré (CI 74151 par exemple)*

*Figure extraite JM Bernard et J. Hugon, CENT,
ENST.*



2.2 Démultiplexeur

- L'information E d'entrée est aiguillée vers une sortie Y_i choisie parmi un groupe de sorties, grâce à une adresse CBA
- $Y_i = E$ quand $CBA_2 = i_{10}$

Adresse de sélection	Sortie							
CBA	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
0 0 0	E	X	X	X	X	X	X	X
0 0 1	X	E	X	X	X	X	X	X
0 1 0	X	X	E	X	X	X	X	X
0 1 1	X	X	X	E	X	X	X	X
1 0 0	X	X	X	X	E	X	X	X
1 0 1	X	X	X	X	X	E	X	X
1 1 0	X	X	X	X	X	X	E	X
1 1 1	X	X	X	X	X	X	X	E

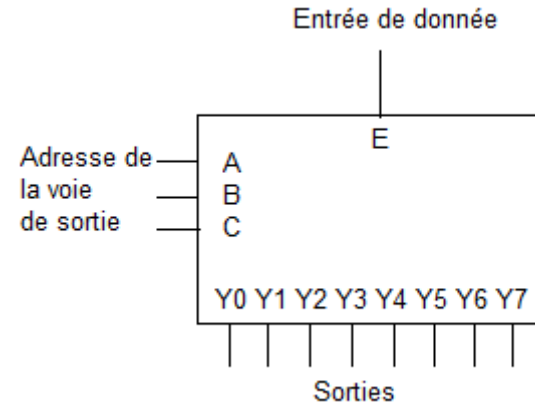


Schéma symbolique d'un démultiplexeur à 8 sorties

3. Transformation de code

3.1. Notions

- Plusieurs façons de représenter l'information codée / Passer d'un code à l'autre
- Circuits Intégrés pour codes courants
- Exemples :

- a) **Circuits qui décodent une combinaison codée en binaire pur, en DCB, en code Gray.. en une combinaison 1 parmi N**
- Une seule sortie du circuit active à la fois
 - Appelés « **Décodeurs** »
- b) **Circuits opération inverse : 1 code 1 parmi N (une seule entrée active parmi N) transformée en code binaire pur**
- Valeur code binaire=rang entrée active
 - Appelés « **codeurs** » ou « **encodeurs** »
 - Si plusieurs entrées actives : affectation d'une priorité (encodeur de priorité)
- c) **Circuits transforment un code DCB en code n parmi 7 pour représenter visuellement les chiffres de 0 à 9 (« décodeur 7 segments »)**

3.2. Fonction de décodage binaire

- Transformer un codage binaire pur en codage 1 parmi N.
- En entrée : un code sur n bits ($N=2^n$ codes possibles) et une entrée de validation
- En sortie : une seule sortie active parmi les N sorties.

- Entrée(s) supplémentaire(s) de validation souvent présente(s)

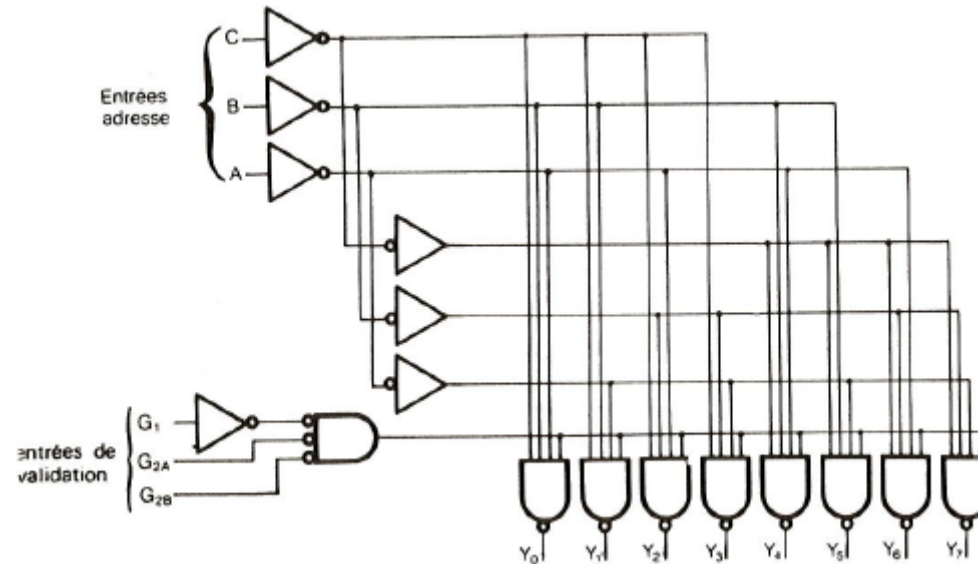
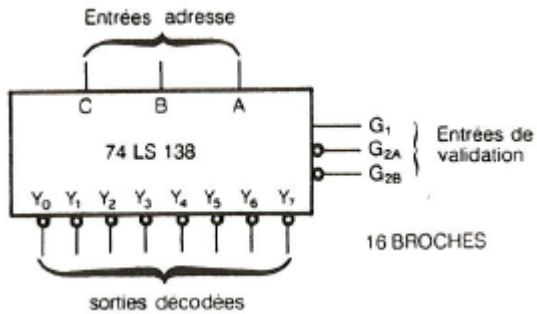
Exemple : table de vérité du décodeur 74138.

Entrées		Sorties							
validation	Adresse								
G ₁ G ₂ *	C B A	Y ₀	Y ₁	Y ₂	Y ₃	Y ₄	Y ₅	Y ₆	Y ₇
X 1	X X X	1	1	1	1	1	1	1	1
0 X	X X X	1	1	1	1	1	1	1	1
1 0	0 0 0	0	1	1	1	1	1	1	1
1 0	0 0 1	1	0	1	1	1	1	1	1
1 0	0 1 0	1	1	0	1	1	1	1	1
1 0	0 1 1	1	1	1	0	1	1	1	1
1 0	1 0 0	1	1	1	1	0	1	1	1
1 0	1 0 1	1	1	1	1	1	0	1	1
1 0	1 1 0	1	1	1	1	1	1	0	1
1 0	1 1 1	1	1	1	1	1	1	1	0

* $G_2 = G_{2A} + G_{2B}$

(Extrait de J.M. Bernard et J. Hugon, Ed CNET, ENST)

Schéma symbolique et schéma interne du décodeur 74138



(Extrait de J.M. Bernard et J. Hugon, Ed CNET, ENST)

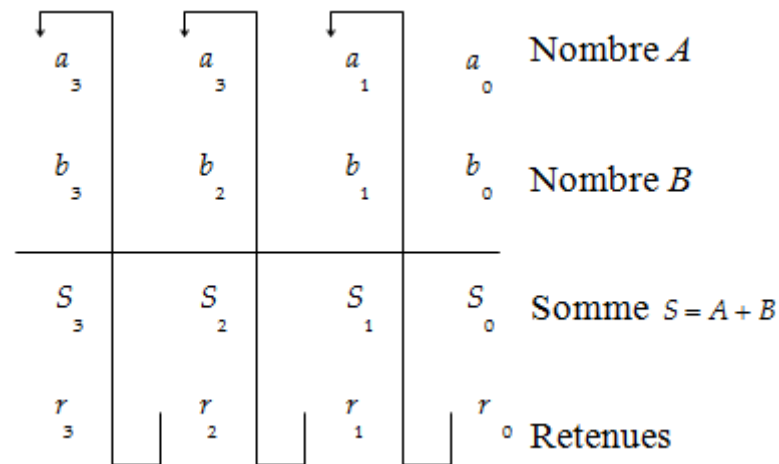
3.3. Fonction d'encodage binaire prioritaire - Voir TD3

3.4. Décodage binaire – 7 segments – Voir DT3

4. Circuits arithmétiques

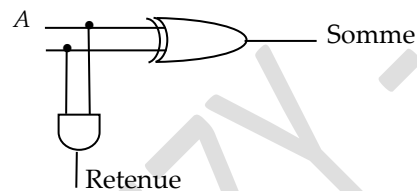
4.1. Additionneur binaire

- A chaque rang : addition de 3 bits



- Etape 1 : addition de 2 bits par un $\frac{1}{2}$ additionneur

A	B	S	R
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



D'après la table de vérité, on peut écrire l'expression de la somme S et de la retenue R :

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

$$R = A \cdot B$$

- Etape 2 : addition de 3 bits par un additionneur complet

A	B	R_e	S	R_s
---	---	-------	---	-------

s : AB 00 01 11 10

R : AB 00 01 11 10

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

Re				
0	0	1	0	1
1	1	0	1	0

$$S = A \oplus B \oplus Re$$

Re				
0	0	0	1	0
1	0	1	1	1

$$Rs = AB + ARe + BRe$$

Ou encore

$$Re(A \oplus B) + AB$$

→ permet de re-utiliser le $A \oplus B$ de S

- Circuit intégré : CI 7483 : additionneur binaire de deux nombres de 4 bits

Exercice d'application n°1 : Additionneur/Soustracteur binaire de deux nombres positifs de 8 bits représentés en Complément à 2, avec les CI 7483.

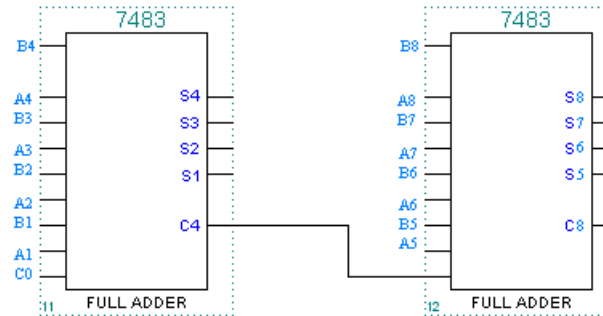
- Les circuits 7483 sont des additionneurs 4bits.
- Un 7483 a 4 sorties sommes : $\Sigma_1, \Sigma_2, \Sigma_3, \Sigma_4$ et une sortie retenue C_4 .
- Pour additionner deux nombres de 8 bits chacun, on utilise deux circuits 7483.
- Supposons que l'on dispose d'un signal de contrôle M qui permet de réaliser :
 - ✓ soit l'addition de A et B ($S=A+B$)
 - ✓ soit la soustraction ($S=B-A$)

- Ecriture de A et B en notation Cpt2 :

$$A : (\underline{A}_8 A_7 A_6 A_5 A_4 A_3 A_2 A_1)_{\text{Cpt2}} \\ B_1)_{\text{Cpt2}}$$

$$B : (B_8 B_7 B_6 B_5 B_4 B_3 B_2$$

- Connexion des deux additionneurs 7483



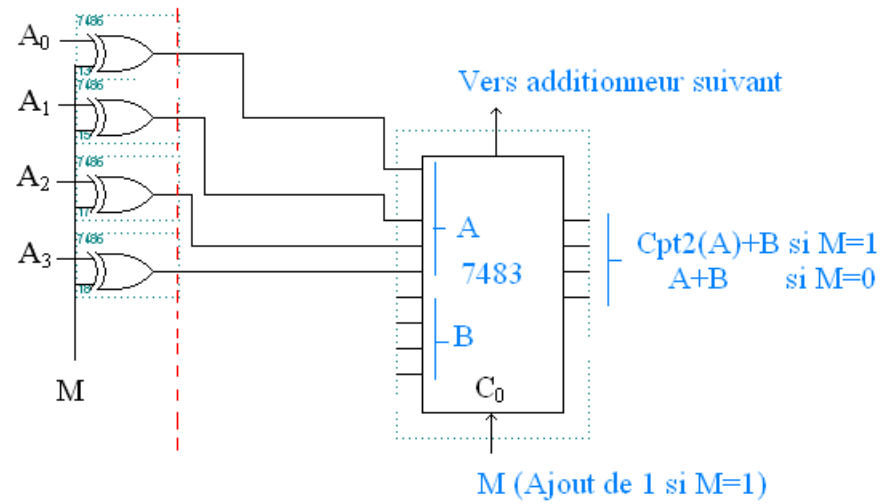
- Soustraction (B-A) obtenue en ajoutant à B le complément à 2 de A.

$$Cpt_2(A) = Cpt_1(A) + 1$$

- Pour faire B-A, on ajoute à B le Cpt1 de A et une retenue $C_0=1$

Additionneur/soustracteur 4 bits

Schéma complet
additionneur/soustracteur 8 bits



$$\begin{aligned}
 M \oplus A &= M \bar{A} + \bar{M} A \\
 &= C_{pt1}(A) \text{ (si } M=1) \\
 &= A \text{ (si } M=0)
 \end{aligned}$$

