

CHAPIV: CIRCUITS LOGIQUES COMBINATOIRES

Les circuits combinatoires

Objectifs

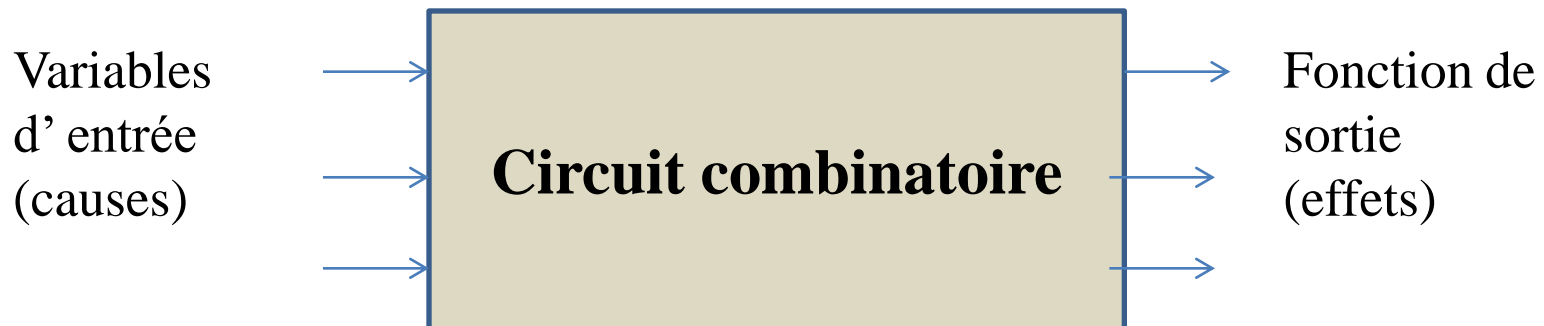
- Apprendre la structure de quelques **circuits combinatoires souvent utilisés** (décodeur, codeur, multiplexeur, démultiplexeur,.....).
- Apprendre **comment utiliser** des circuits combinatoires pour concevoir d'autres circuits **plus complexes**.

Exemple de Circuits combinatoires

1. Demi Additionneur
2. Additionneur complet
3. Comparateur
4. Multiplexeur
5. Demultiplexeur
6. Encodeur
7. Décodeur

CIRCUITS LOGIQUES COMBINATOIRES

Un circuit est combinatoire quand ses sorties ne dépendent que de ses entrées et non pas aussi de ses états antérieurs: à chaque combinaison des variables d'entrée correspond toujours une seule combinaison des fonctions de sortie, et toujours la même; il n'y a pas de rebouclage des sorties vers les entrées.



Dans les systèmes numériques, on retrouve toujours des données et des informations codées sous forme binaire qui sont sans cesse soumises à des opérations, comme:

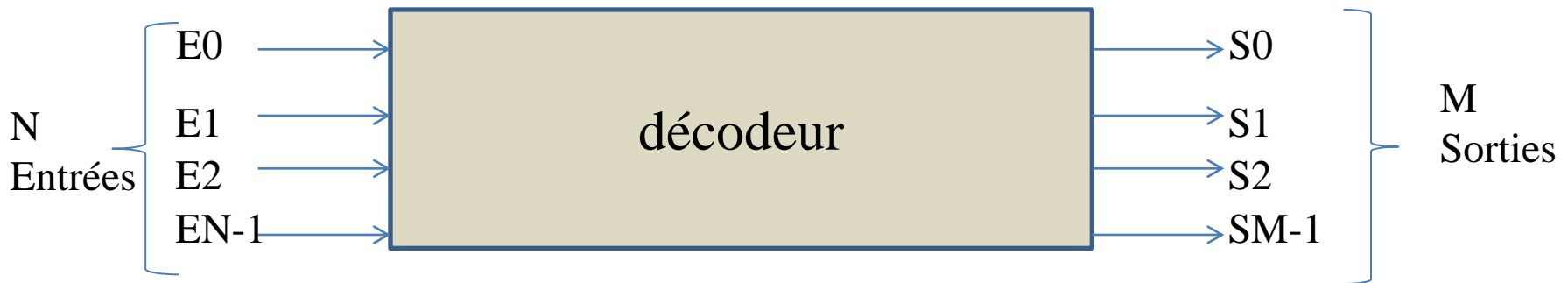
- 1) **Le décodage et le codage**: transposition des données d'un code à un autre;
- 2) **Le multiplexage**: choix d'un groupe de données parmi plusieurs;
- 3) **Le démultiplexage**: aiguillage des données vers une destination parmi plusieurs;
- 4) **L'acheminement par bus**: transmission de données entre plusieurs dispositifs par l'intermédiaire d'un bus commun.

Toutes ces opérations se matérialisent facilement grâce aux nombreux circuits intégrés de la classe MSI (Intégration à Moyenne Echelle).

Il s'agit ici d'étudier plusieurs types de dispositifs MSI parmi les plus courants.

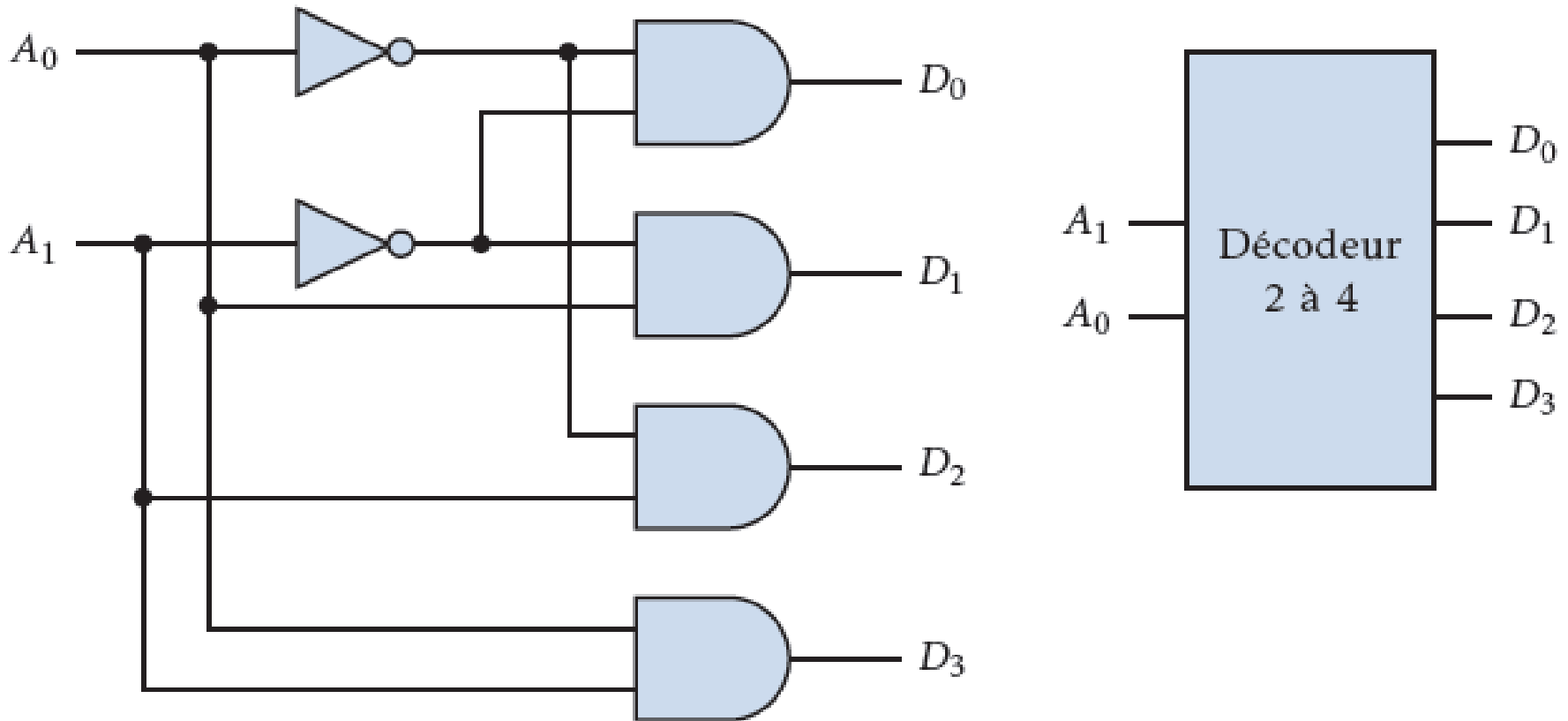
I- DECODEURS

Le décodeur est un circuit logique qui établit la correspondance entre un code d'entrée binaire de N -bits et M lignes de sortie, avec $M \leq 2^N$

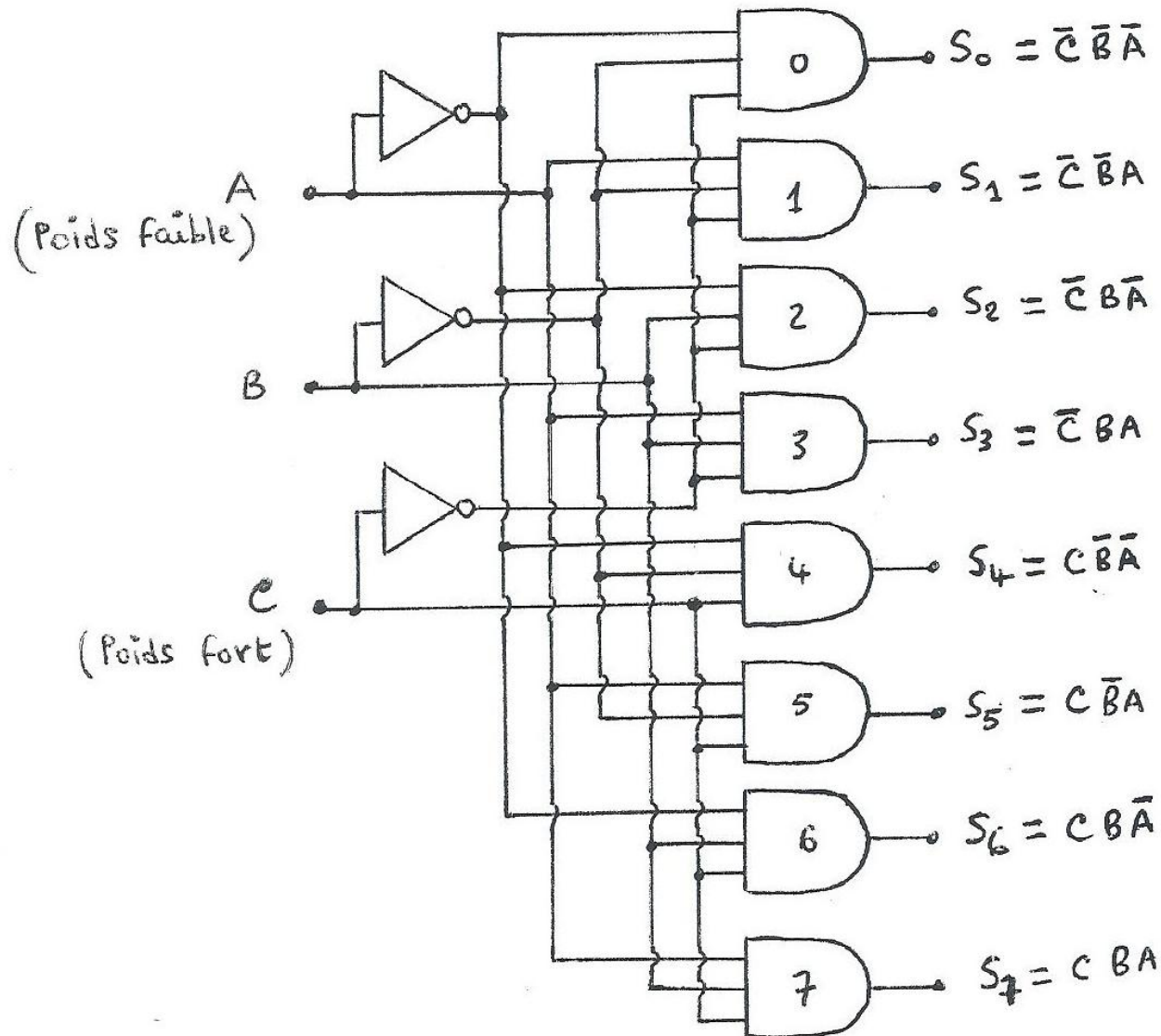


Pour chacune des combinaisons possibles des entrées, une seule ligne de sortie est validée. Etant donné que chacune des N entrées peut être soit 0 soit 1, il y a 2^N combinaisons ou codes d'entrées possibles.

* Exemple: de décodeur ayant deux voies d'entrée et $2^2 = 4$ voies de sortie:



* Exemple de décodeur ayant trois voies d'entrée et $2^3 = 8$ voies de sortie:



Pour un code d'entrée donné, la seule sortie qui est vraie (niveau HAUT ici) est celle correspondant à l'équivalent décimal du code d'entrée binaire (par exemple, la sortie S_6 passe à 1 quand $CBA=110_2=6_{10}$).

Types de désignation de ce décodeur:

- Décodeur entrées 3 voies, sortie 8 voies;
- Décodeur ou convertisseur binaire-octal: établit la correspondance entre un code d'entrée binaire de 3 bits et une sortie parmi 8 (octal);
- Décodeur 1 parmi 8: 1 sortie parmi les 8 est activée à la fois.

a) Entrées VALIDATION

Certains décodeurs sont dotés d'une ou de plusieurs entrées VALIDATION qui servent à commander leur fonctionnement.

*Exemple: décodeur 74LS138 TTL de Fairchild.

-Schéma logique-

* Exemple: décodeur 74LS 138 TTL de Fairchild.

(Poids fort)

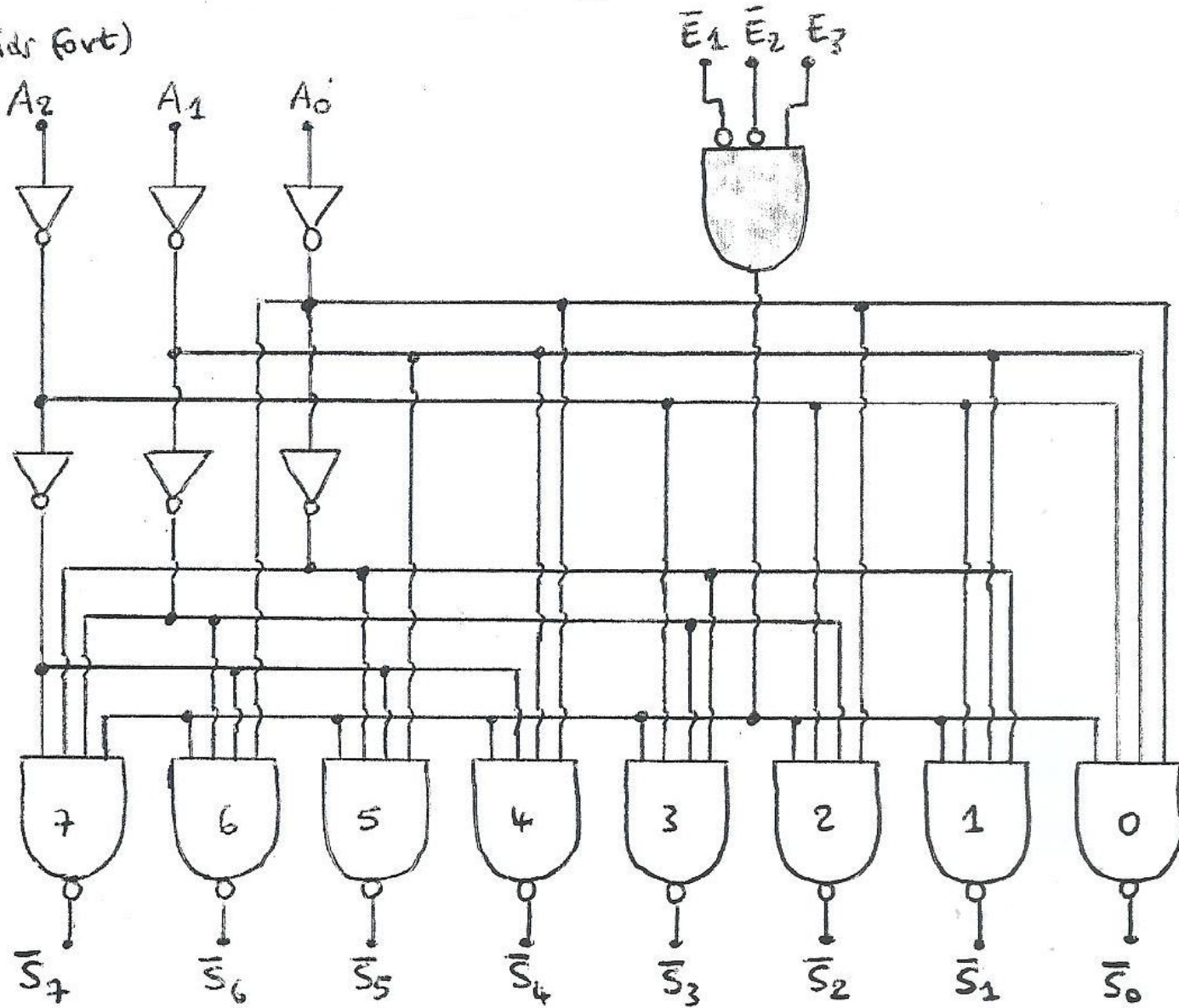
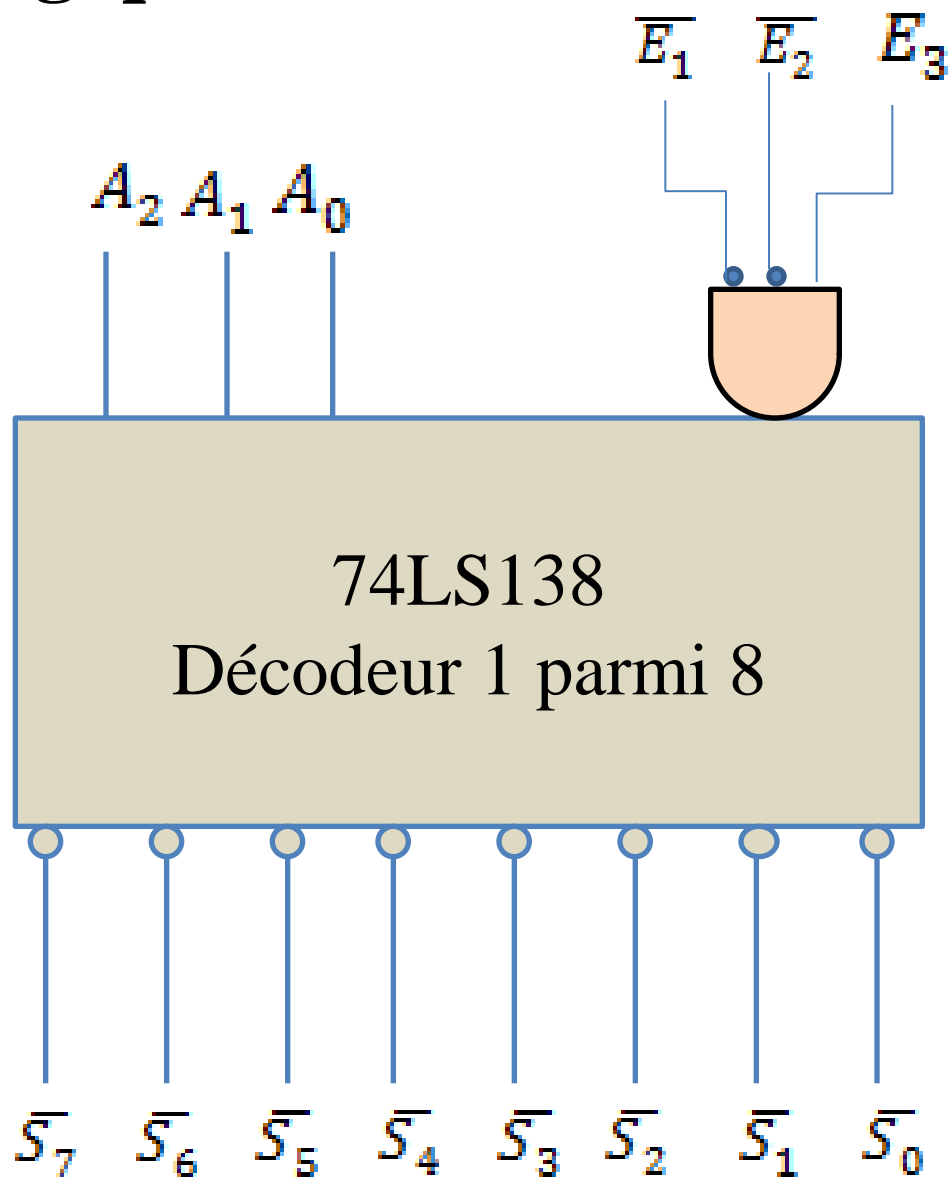


Table de vérité

\overline{E}_1	\overline{E}_2	E_3	Sorties (\overline{S}_i)
0	0	1	Réagit au code d'entrée $A_2 A_1 A_0$
1	X	X	Invalidées-toutes au niveau HAUT
X	1	X	Invalidées- toutes au niveau HAUT
X	X	0	Invalidées- toutes au niveau HAUT

X = condition indifférente

Symbole logique:



-Fonctionnement

- Les sorties de ce décodeur sont des portes NON-ET; par conséquent, elles sont vraies au niveau BAS. C'est pourquoi ces sorties sont désignées avec une barre de complémentation (\bar{S}_0 , \bar{S}_1 , ..., \bar{S}_7).
- Le code d'entrée est appliqué aux bornes A_2 , A_1 et A_0 , où A_2 est le bit de poids fort. C'est un décodeur 1 parmi 8.
- Les entrées \bar{E}_1 , \bar{E}_2 et E_3 sont des entrées de validation distinctes combinées dans une porte ET.

Les portes NON-ET de sortie ne sont validées pour indiquer le code d'entrée $A_2 A_1 A_0$ correspondant que lorsque la porte ET a une sortie au niveau HAUT. Ceci se produit uniquement lorsque $\overline{E_1} = \overline{E_2} = 0$ et $E_3 = 1$.

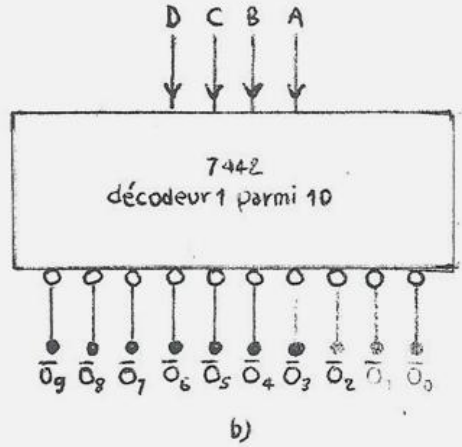
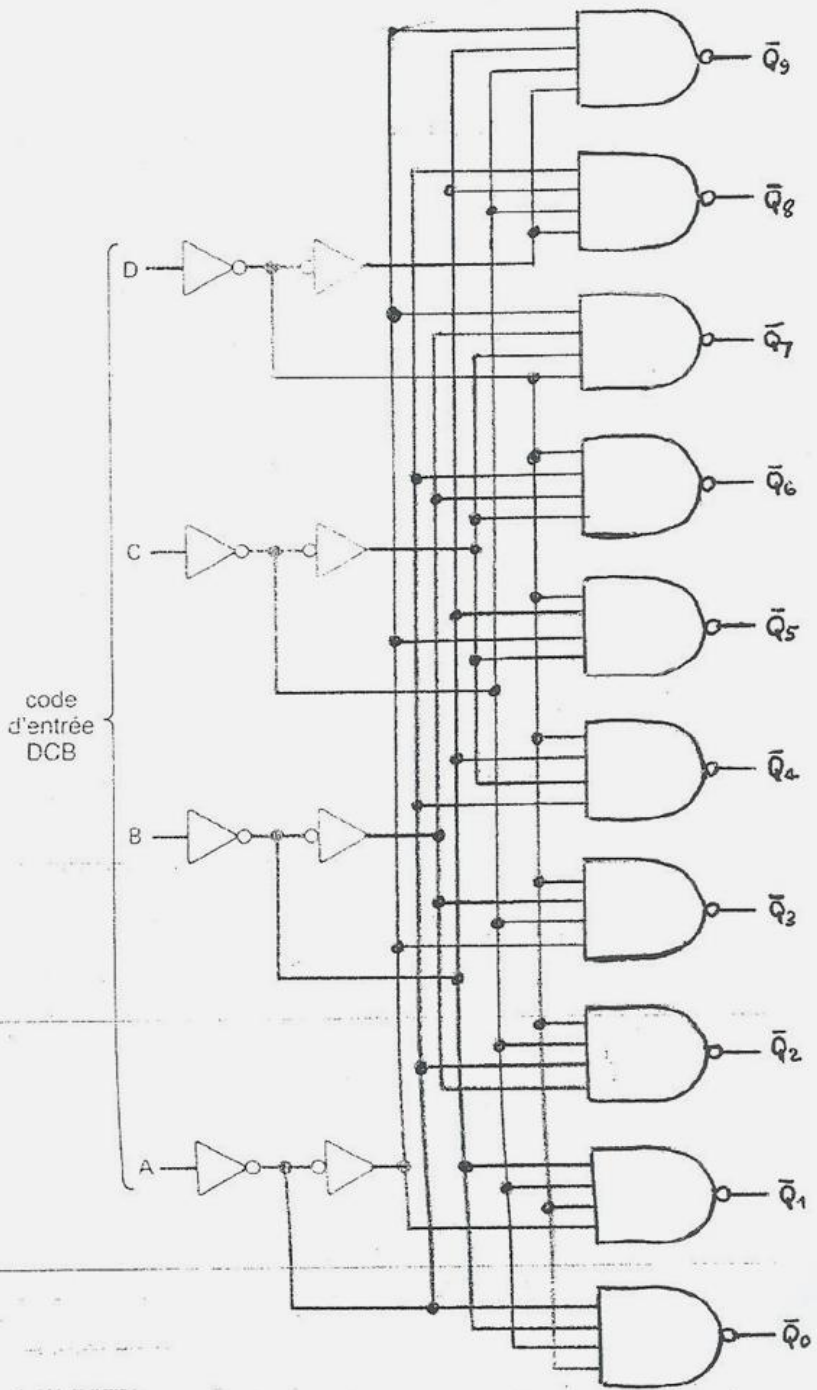
Dans tous les autres cas, la sortie de la porte ET est au niveau BAS, ce qui impose aux sorties de toutes les portes NON-ET leur état inactif HAUT quelque soit le code d'entrée.

NB: Sur le schéma du symbole logique, la porte ET de validation dessinée à l'extérieur fait partie des circuits internes de la puce 74LS138.

b) Décodeurs DCB-décimal

La figure ci-dessous montre le schéma logique d'un décodeur DCB-décimal 7442, que l'on retrouve aussi en version 74LS42 et 74HC42. Une sortie ne passe à 0 qu'au moment où son entrée correspondante DCB est appliquée. Par exemple, la sortie $\overline{Q_5}$ ne passe au niveau BAS que lorsque les valeurs sur les entrées sont DCBA=0101; de même, $\overline{Q_9}$ ne passe au niveau BAS que lorsque DCBA=1001.

Dans le cas des combinaisons DCB non valides, aucune des entrées n'est validée. Ce décodeur est couramment appelé décodeur entrée 4 voies, sortie 10 voies ou décodeur 1 parmi 10. On remarque dans ce décodeur l'absence d'une entrée de validation.



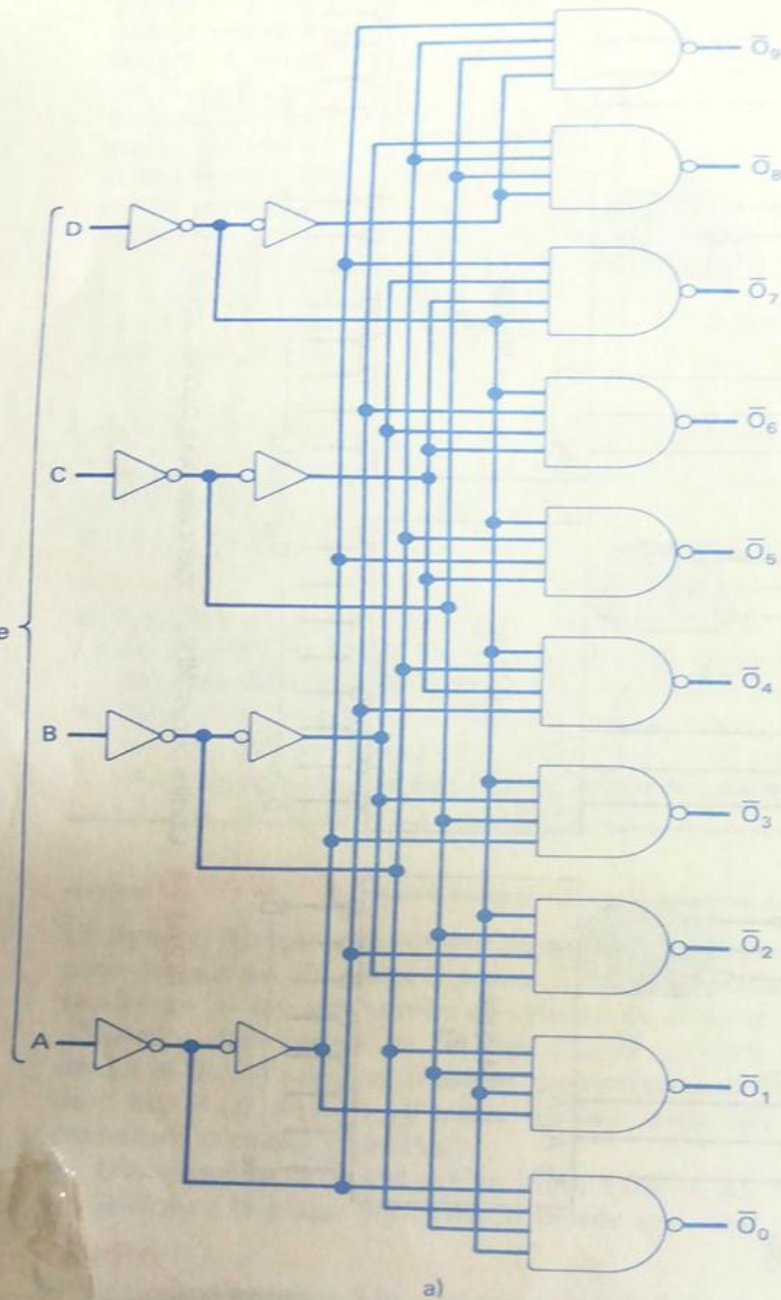
b)

Entrées				Sortie active
D	C	B	A	
0	0	0	0	\bar{Q}_9
0	0	0	1	\bar{Q}_8
0	0	1	0	\bar{Q}_7
0	0	1	1	\bar{Q}_6
0	1	0	0	\bar{Q}_5
0	1	0	1	\bar{Q}_4
0	1	1	0	\bar{Q}_3
0	1	1	1	\bar{Q}_2
1	0	0	0	\bar{Q}_1
1	0	0	1	\bar{Q}_0
1	0	1	0	aucune
1	0	1	1	aucune
1	1	0	0	aucune
1	1	0	1	aucune
1	1	1	0	aucune
1	1	1	1	aucune

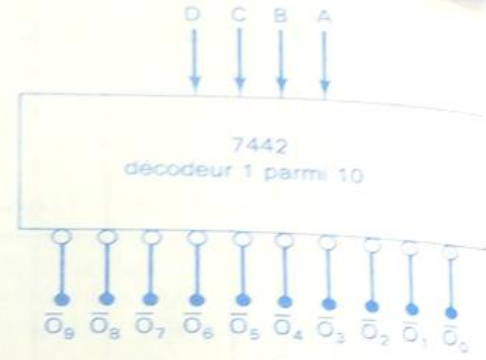
H = niveau de tension HAUT
 B = niveau de tension BAS

c)

code d'entrée DCB



a)



b)

Entrées				Sortie active
A	B	C	D	
B	B	B	B	\bar{O}_9
B	B	B	H	\bar{O}_8
B	B	H	B	\bar{O}_7
B	B	H	H	\bar{O}_6
B	H	B	B	\bar{O}_5
B	H	B	H	\bar{O}_4
B	H	H	B	\bar{O}_3
B	H	H	H	\bar{O}_2
H	B	B	B	\bar{O}_1
H	B	B	H	\bar{O}_0
H	B	H	B	aucune
H	B	H	H	aucune
H	H	B	B	aucune
H	H	B	H	aucune
H	H	H	B	aucune
H	H	H	H	aucune

H = niveau de tension HAUT
B = niveau de tension BAS

NB: Il existe aussi le décodeur/pilote DCB-décimal, tel que le TTL 7445 dont les sorties peuvent absorber jusqu'à 80 mA dans l'état BAS et être portées jusqu'à 30V dans l'état HAUT. Ceci lui permet d'attaquer directement des charges comme des voyants LED, des relais ou des moteurs à courant continu.

C) Applications du décodeur

Les décodeurs servent chaque fois qu'une sortie ou un groupe de sorties ne doit être validé qu'à la réalisation d'une certaine combinaison des niveaux d'entrée. (Ces niveaux d'entrée sont fréquemment fournis par un compteur ou un registre). Les sorties du décodeur peuvent servir à une tâche de synchronisation ou de séquençement pour mettre en marche ou à l'arrêt différents dispositifs.

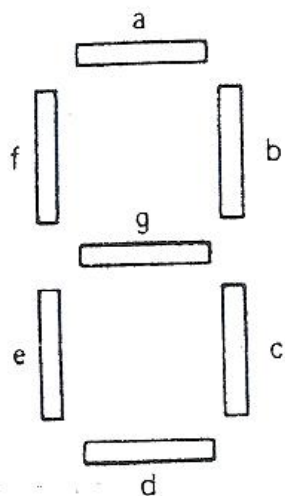
. Les décodeurs sont largement utilisés dans les mémoires des ordinateurs; ce sont eux qui reçoivent le code d'adresse du processeur central et activent l'emplacement mémoire désigné par l'adresse.

Un autre domaine d'applications considérable des décodeurs est celui de la conversion de données binaires en une forme se prêtant à un affichage numérique.

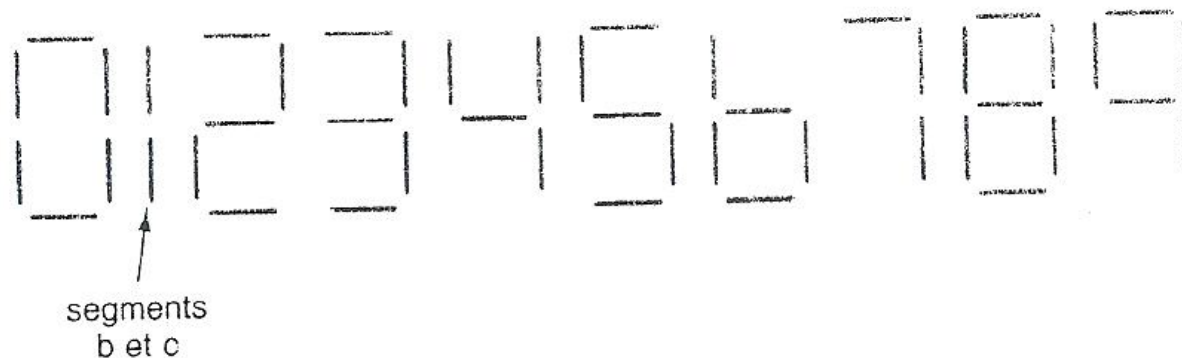
II- PILOTES/DECODEURS DCB-7 SEGMENTS

Dans de nombreux affichages numériques, les dix chiffres 0 à 9, et parfois les caractères hexadécimaux A à F, sont configurés au moyen de 7 segments (voir figure ci-dessous). Chaque segment est constitué d'un matériau qui émet de la lumière quand il est traversé par un courant (LED, filaments incandescents). Par exemple, pour afficher le chiffre « 3 », il faut que les segments a, b, c, d et g soient allumés et que les segments e et f soient éteints.

FIGURE: a) Disposition des 7 segments; b) les segments qui doivent être allumés pour former chacun des chiffres.



a)



b)

FIGURE

a) Disposition des 7 segments; b) les segments qui doivent être allumés pour former chacun des chiffres.

Un pilote/ décodeur DCB-7 segments accepte en entrée les 4 bits DCB et rend actives les sorties qui vont permettre de faire passer un courant dans les segments qui forment le chiffre décimal correspondant.

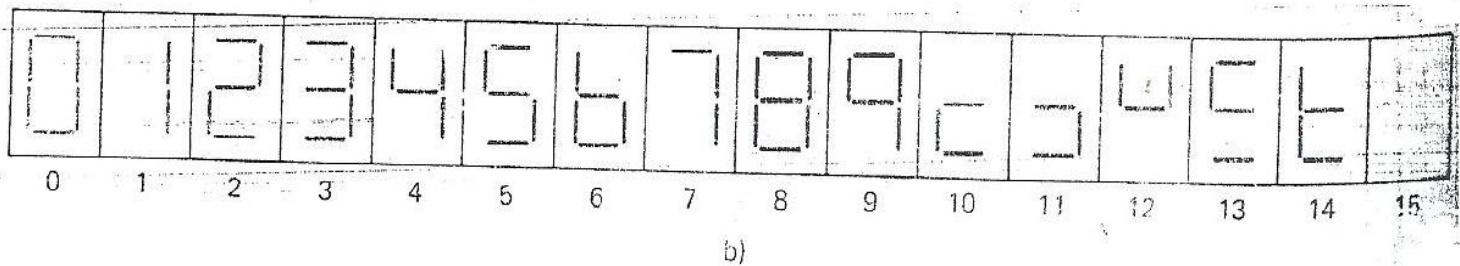
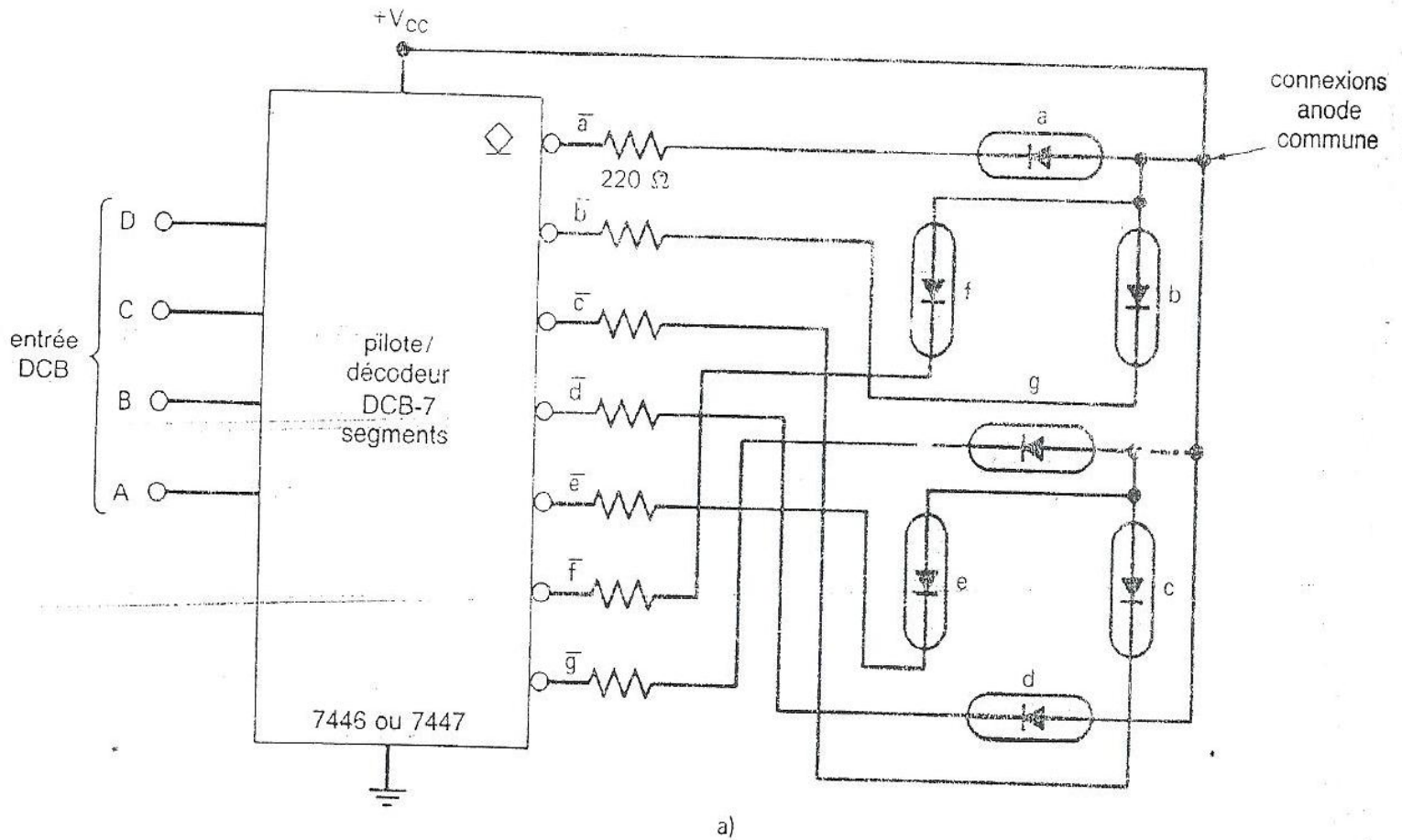
Ici, chaque sortie peut être au niveau VRAI dans plus d'une combinaison de bits d'entrée. Par exemple, le segment **e** est allumé lorsque sont formés les chiffres 0, 2, 6, et 8, c.à.d. quand les codes d'entrée sont 0000, 0010, 0110 ou 1000.

Le pilote/décodeur DCB-7 segments, illustré à la figure ci-après (TTL 7446 ou 7447), pilote un afficheur LED à 7 segments. Les anodes de ces diodes sont toutes réunies à V_{cc} (+5V). Leurs cathodes sont connectées au travers de résistances limitatrices de courant aux sorties appropriées du pilote/décodeur. Celui-ci a des sorties qui sont vraies au niveau BAS.

Pour illustrer le fonctionnement de ce circuit, soit l'entrée DCB: $D=0$, $C=1$, $B=0$, $A=1$, correspondant à la représentation DCB de 5. En réponse à cette entrée, les sorties \bar{a} , \bar{f} , \bar{g} , \bar{c} , et \bar{d} du pilote/décodeur sont amenées au niveau BAS (raccordées à la masse), ce qui a pour effet de faire passer un courant à travers les segments LED a , f , g , c et d et d'afficher le chiffre 5. Les sorties \bar{b} et \bar{e} demeurent au niveau HAUT (ouvert) et les segments correspondants b et e restent éteints.

FIGURE

a) Pilote/décodeur DCB-7 segments attaquant un afficheur DEL à 7 segments à anode commune; b) caractères configurés par les segments pour toutes les représentations d'entrée possibles.



* Exemple:

Chaque segment d'un afficheur LED 7 segments typique donne une luminosité normale quand il est traversé par 10mA sous 2,7V. Calculer la valeur de la résistance limitatrice de courant nécessaire pour que chaque segment soit traversé approximativement par 10mA.

Solution:

D'après la figure ci-dessus, la chute de tension dans la résistance en série vaut:

$$V_{CC} - V_{LED} = 5 - 2,7 = 2,3V$$

Pour avoir un courant de 10 mA, il faut une résistance de:

$$R_s = \frac{2,30V}{10mA} = 230 \Omega$$

Soit en valeur normalisée, $R_s = 220 \Omega$.

III- AFFICHEURS A CRISTAUX LIQUIDES (ACL ou LCD)

Théoriquement, les afficheurs à cristaux liquides fonctionnent sous une basse tension (typiquement 3 à $15V_{\text{eff}}$) à basse fréquence (25 à 60Hz), et consomment très peu de courant. Souvent, ces afficheurs possèdent 7 segments afin de figurer les chiffres (voir figure ci-dessous).

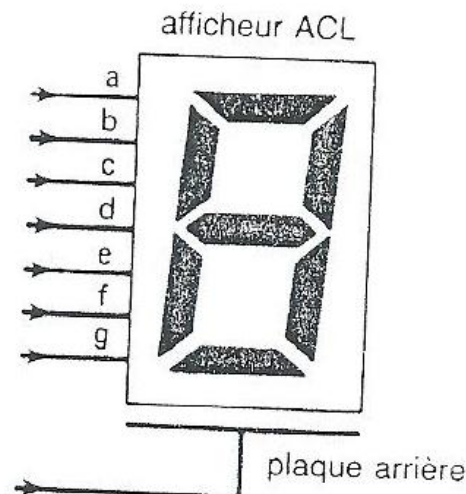


FIGURE Afficheur à cristaux liquides.

La tension alternative nécessaire pour allumer un segment est appliquée entre ce dernier et la plaque arrière, commune à tous les segments.

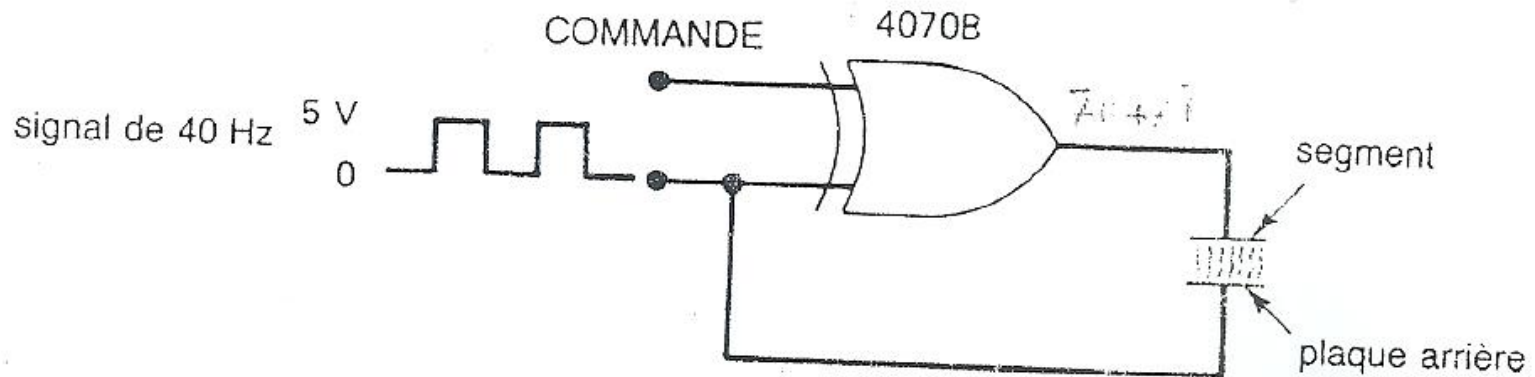
Le segment et la plaque arrière forment un condensateur qui tire très peu de courant tant que la fréquence reste faible. Généralement, elle ne doit pas être quand même inférieure à 25Hz, sans quoi les segments pourraient clignoter.

Les ACL consomment moins de courant que les afficheurs LED et sont très utilisés dans les dispositifs à pile comme les calculatrices et les montres.

Un ACL n'émet pas d'énergie lumineuse comme une LED; il faut donc l'éclairer de l'extérieur.

Attaque d'un ACL:

On applique une onde carrée de 40Hz à la plaque arrière de même qu'à l'entrée d'une porte OU-EXCLUSIF CMOS 4070. A l'autre entrée de la porte, on applique un signal de COMMANDE qui donne l'ordre d'ALLUMER ou d'ETEINDRE le segment.



FIGURE

Méthode d'excitation d'un segment ACL. Quand COMMANDE est BASSE, le segment est ÉTEINT. Quand COMMANDE est HAUTE, le segment est ALLUMÉ.

Quand l'entrée **COMMANDE** est **BASSE**, la sortie de la porte **OU-EXCLUSIF** est une réplique de l'onde carrée de 40Hz, de sorte que les signaux appliqués au segment et à la plaque arrière sont égaux. Il n'y a donc pas de différence de tension et le segment est **ETEINT**.

Quand l'entrée **COMMANDE** est **HAUTE**, la sortie de la porte est l'inverse de l'onde carrée de 40Hz et le signal appliqué au segment est en inversion de phase par rapport au signal appliqué à la plaque.

Par conséquent, la tension du segment sera alternativement à $+5V$ et à $-5V$ par rapport à la plaque. Cette tension alternative ALLUME le segment.

- Ce principe de base peut être appliqué à une rangée complète d'afficheurs ACL 7-segments, (voir figure ci-dessous). Sur cette figure, le pilote/ décodeur DCB- 7 segments CMOS 4511 fournit les signaux de COMMANDE aux 7 portes OU-EXCLUSIF qui excitent les 7 segments. Les sorties du 4511 sont vraies au niveau HAUT,

puisque'il faut un niveau HAUT pour ALLUMER un segment.

Généralement, les dispositifs CMOS sont utilisés pour piloter les ACL parce qu'ils dissipent moins de puissance que les TTL et conviennent mieux aux applications à pile qui utilisent des ACL. Ils n'affectent pas leur durée de vie.

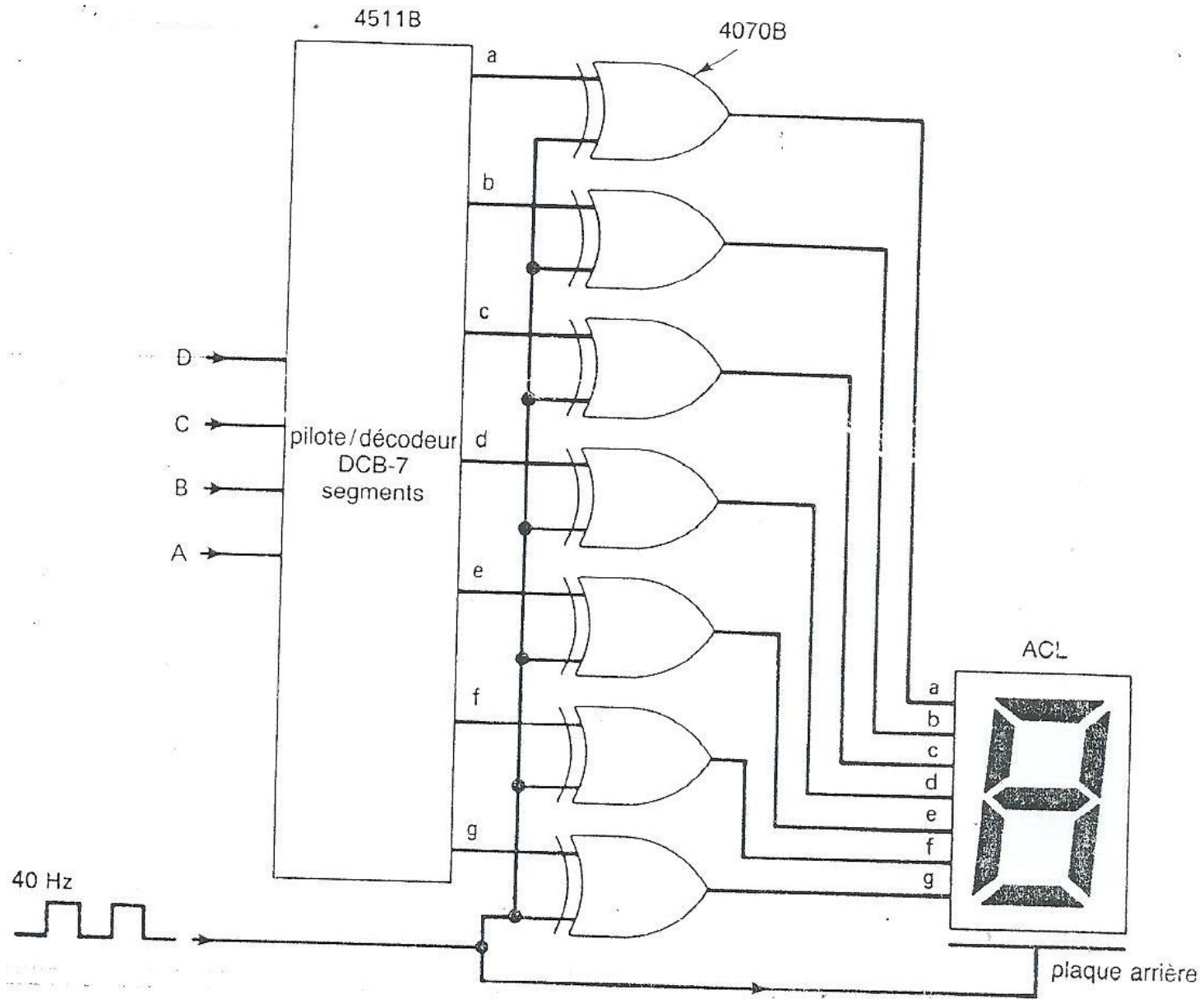
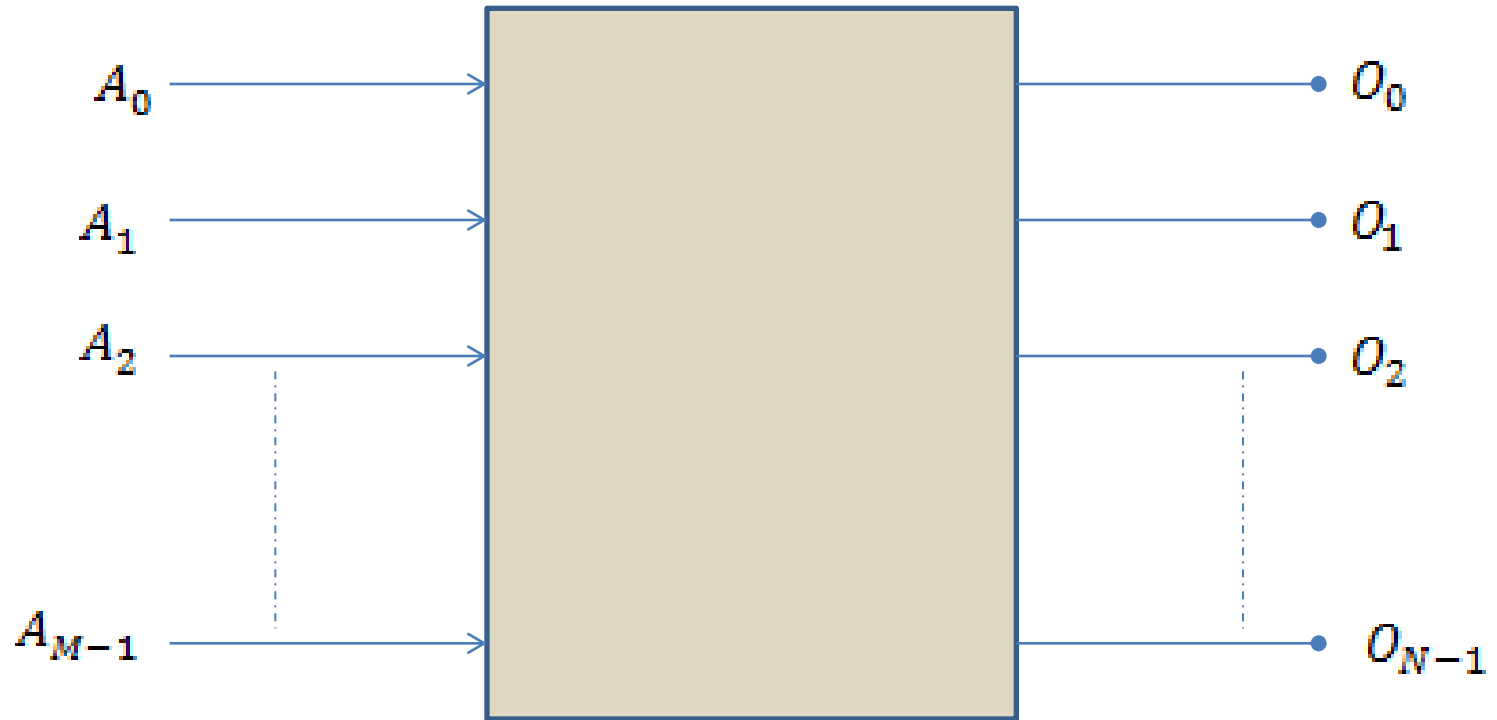


FIGURE: EXCITATION d'un ACL 7-segments

IV CODEURS

Un codeur réalise le processus inverse du décodeur: il a un certain nombre de voies d'entrée, dont une seule est active à la fois; à une certaine voie d'entrée correspond une représentation de sortie de N bits.

Schéma général d'un codeur:



M entrées, une seule est
au niveau HAUT à la fois

Représentation de sortie
de N bits

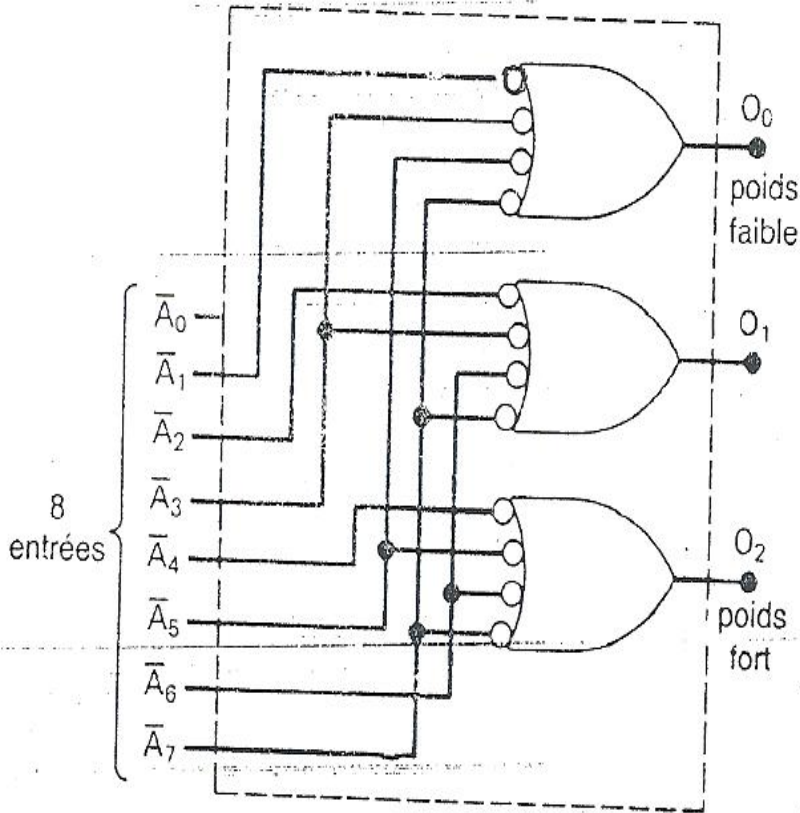
Ici, les entrées sont vraies au niveau HAUT, ce qui veut dire que normalement elles sont au niveau BAS.

Nous avons vu qu'un décodeur binaire -octal fait correspondre à un code d'entrée binaire de 3 bits une seule des huit voies de sortie possibles. A l'inverse, un codeur octal- binaire a 8 voies d'entrée et produit une représentation de sortie binaire de 3 bits.

La figure ci-dessous donne le circuit et sa table de vérité (entrées vraies au niveau BAS):

FIGURE

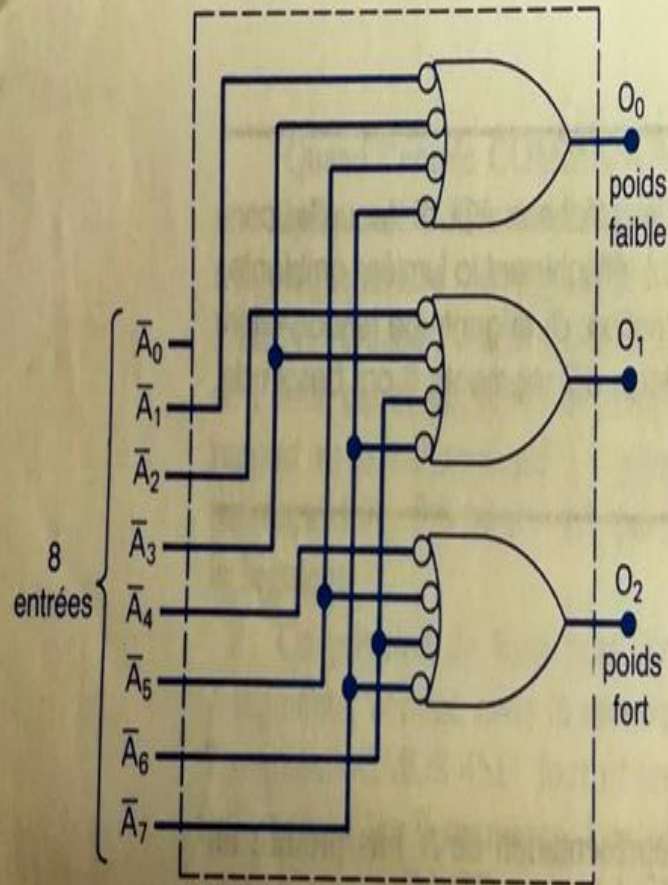
Schéma logique d'un codeur octal-binaire (entrée 8 voies, sortie 3 voies).
Ce circuit ne fonctionne correctement que lorsqu'une seule entrée est active à la fois.



Entrées								Sorties		
\bar{A}_0	\bar{A}_1	\bar{A}_2	\bar{A}_3	\bar{A}_4	\bar{A}_5	\bar{A}_6	\bar{A}_7	O_2	O_1	O_0
X	1	1	1	1	1	1	1	0	0	0
X	0	1	1	1	1	1	1	0	0	1
X	1	0	1	1	1	1	1	0	1	0
X	1	1	0	1	1	1	1	0	1	1
X	1	1	1	0	1	1	1	1	0	0
X	1	1	1	1	0	1	1	1	0	1
X	1	1	1	1	1	0	1	1	1	0
X	1	1	1	1	1	1	0	1	1	1

Une seule entrée
est BASSE
à la fois.

Schéma logique d'un codeur octal-binaire (entrée 8 voies, sortie 3 voies).
Ce circuit ne fonctionne correctement que lorsqu'une seule entrée est active à la fois.



Entrées								Sorties		
\bar{A}_0	\bar{A}_1	\bar{A}_2	\bar{A}_3	\bar{A}_4	\bar{A}_5	\bar{A}_6	\bar{A}_7	O_2	O_1	O_0
X	1	1	1	1	1	1	1	0	0	0
X	0	1	1	1	1	1	1	0	0	1
X	1	0	1	1	1	1	1	0	1	0
X	1	1	0	1	1	1	1	0	1	1
X	1	1	1	0	1	1	1	1	0	0
X	1	1	1	1	0	1	1	1	0	1
X	1	1	1	1	1	0	1	1	1	0
X	1	1	1	1	1	1	0	1	1	1

Une seule entrée
est BASSE
à la fois.

L'analyse successive des réponses des circuits logiques permet de conclure qu'un niveau BAS sur une seule entrée donne lieu en sortie à un code binaire correspondant à cette entrée. Par exemple, un niveau BAS sur $\overline{A_5}$, pendant que toutes les autres entrées sont à 1, produit $O_2 = 1$, $O_1 = 0$ et $O_0 = 1$, ce qui est le code binaire de 5.

On observe par ailleurs que $\overline{A_0}$ n'est pas connectée à une porte logique, puisque les sorties du décodeur sont normalement à 000 quand aucune des entrées de $\overline{A_1}$ à $\overline{A_7}$ n'est au niveau BAS.

a. Codeurs de priorité:

Le codeur de la figure précédente produit des résultats erronés si au moins deux entrées sont rendues actives simultanément. Un codeur de priorité est une version modifiée du codeur élémentaire qui possède les circuits logiques nécessaires pour que le code de sortie choisi, quand deux entrées sont actives, soit celui qui correspond au nombre le plus haut. Par exemple, pour un tel codeur, si les entrées $\overline{A_0}$, $\overline{A_2}$ et $\overline{A_6}$ sont rendues actives en même temps, il produit en sortie le code 110 correspondant au code binaire 6.

Exemples de codeurs de priorité octal- binaire:

74148, 74LS148, 74HC148.

b. Codeur de priorité décimal-DCB 74147

(74LS147,74HC147)

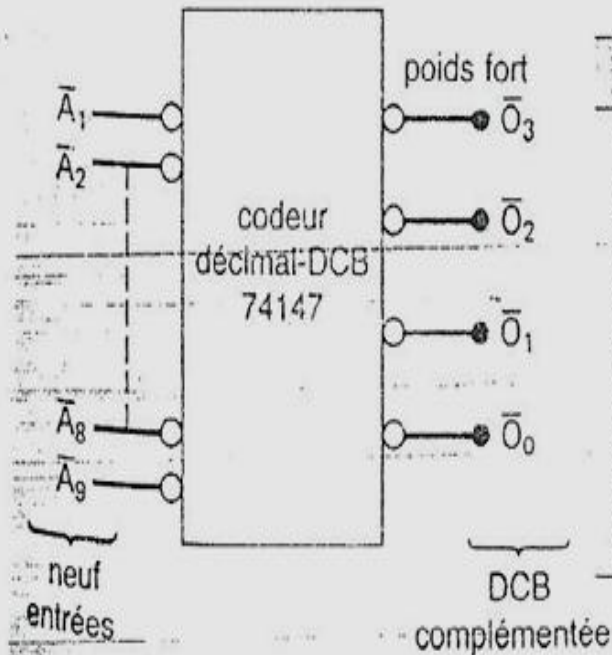
La figure ci-dessous montre le symbole logique du codeur de priorité décimal- DCB74147 ainsi que sa table de vérité. Celui-ci à 9 entrées vraies au niveau BAS représentant les 9 chiffres décimaux 1 à 9 et il produit la représentation **DCB complémentée** correspondant à l'entrée la plus haute mise au niveau vrai.

L'analyse de la table de vérité montre qu'une entrée au niveau BAS, si toutes les entrées de rang supérieur sont au niveau HAUT, donne l'inverse du code DCB correspondant à cette entrée.

Les sorties du 74147 sont normalement à 1 quand aucune des entrées n'est à son niveau vrai. Ceci correspond à la condition d'entrée du chiffre décimal 0. Il n'y a en réalité aucune entrée $\overline{A_0}$, puisque le codeur suppose que l'état d'entrée du chiffre décimal 0 est celui créé quand toutes les autres entrées sont à 1.

Pour obtenir le code DCB naturel à partir des sorties DCB complémentées du 74147, il faut ajouter un INVERSEUR à chacune des sorties.

FIGURE . Codeur de priorité décimal-DCB 74147.



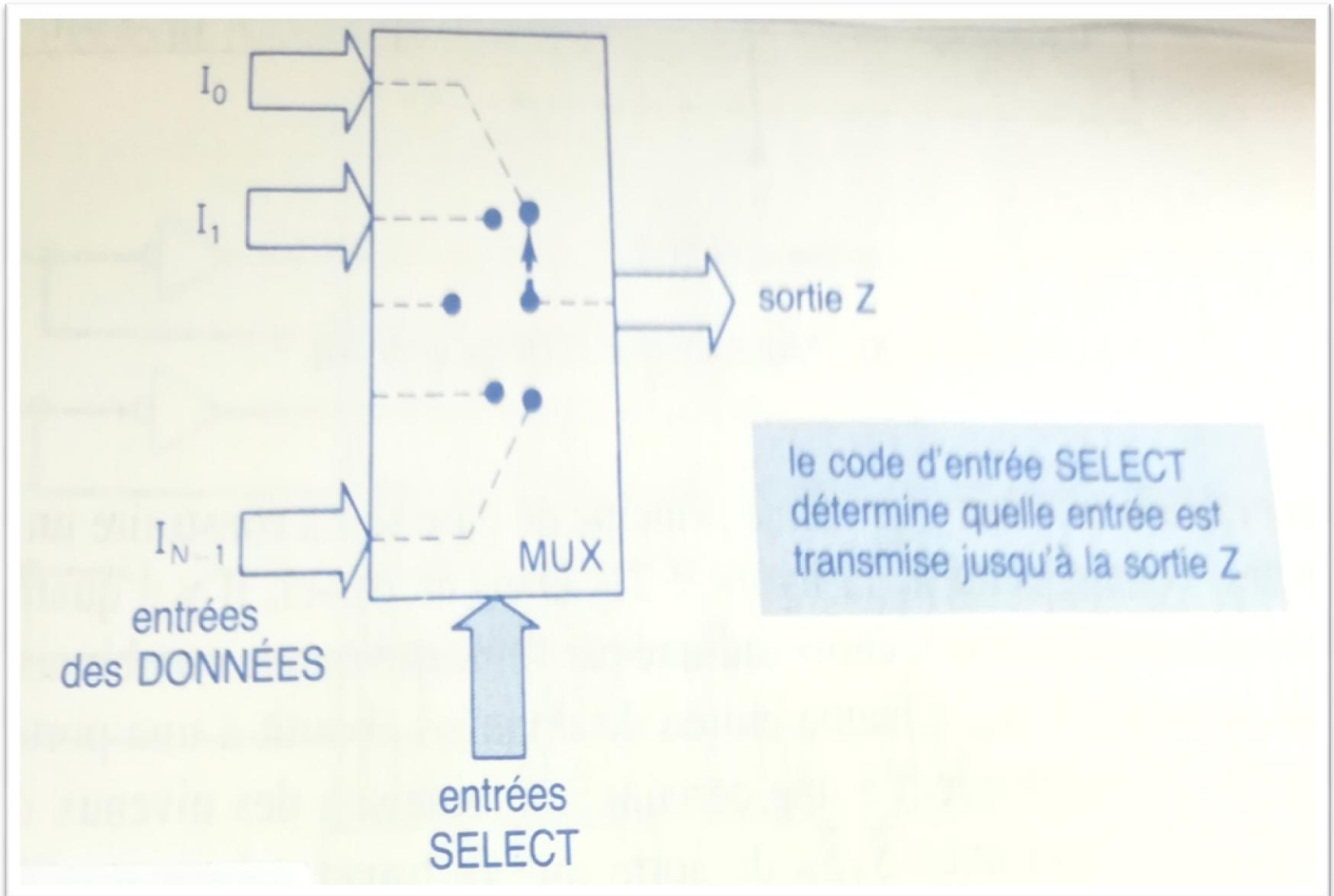
\bar{A}_1	\bar{A}_2	\bar{A}_3	\bar{A}_4	\bar{A}_5	\bar{A}_6	\bar{A}_7	\bar{A}_8	\bar{A}_9	\bar{O}_3	\bar{O}_2	\bar{O}_1	\bar{O}_0
1	1	1	1	1	1	1	1	1	1	1	1	1
X	X	X	X	X	X	X	X	0	0	1	1	0
X	X	X	X	X	X	0	1	1	1	0	0	0
X	X	X	X	0	1	1	1	1	1	0	0	1
X	X	X	0	1	1	1	1	1	1	0	1	0
X	0	1	1	1	1	1	1	1	1	1	0	0
X	0	1	1	1	1	1	1	1	1	1	0	1
0	1	1	1	1	1	1	1	1	1	1	1	0

X = 0 ou 1

V. MULTIPLEXEUR (SELECTEUR DE DONNEES)

Un multiplexeur ou sélecteur de données est un circuit logique ayant plusieurs entrées de données, mais seulement une sortie qui communique les données. L'aiguillage de l'entrée de données sur la sortie est commandé par les entrées SELECT (appelées parfois entrées d'ADRESSE). La figure ci-dessous donne le symbole d'un multiplexeur général (MUX):

Symbole d'un multiplexeur numérique (MUX)

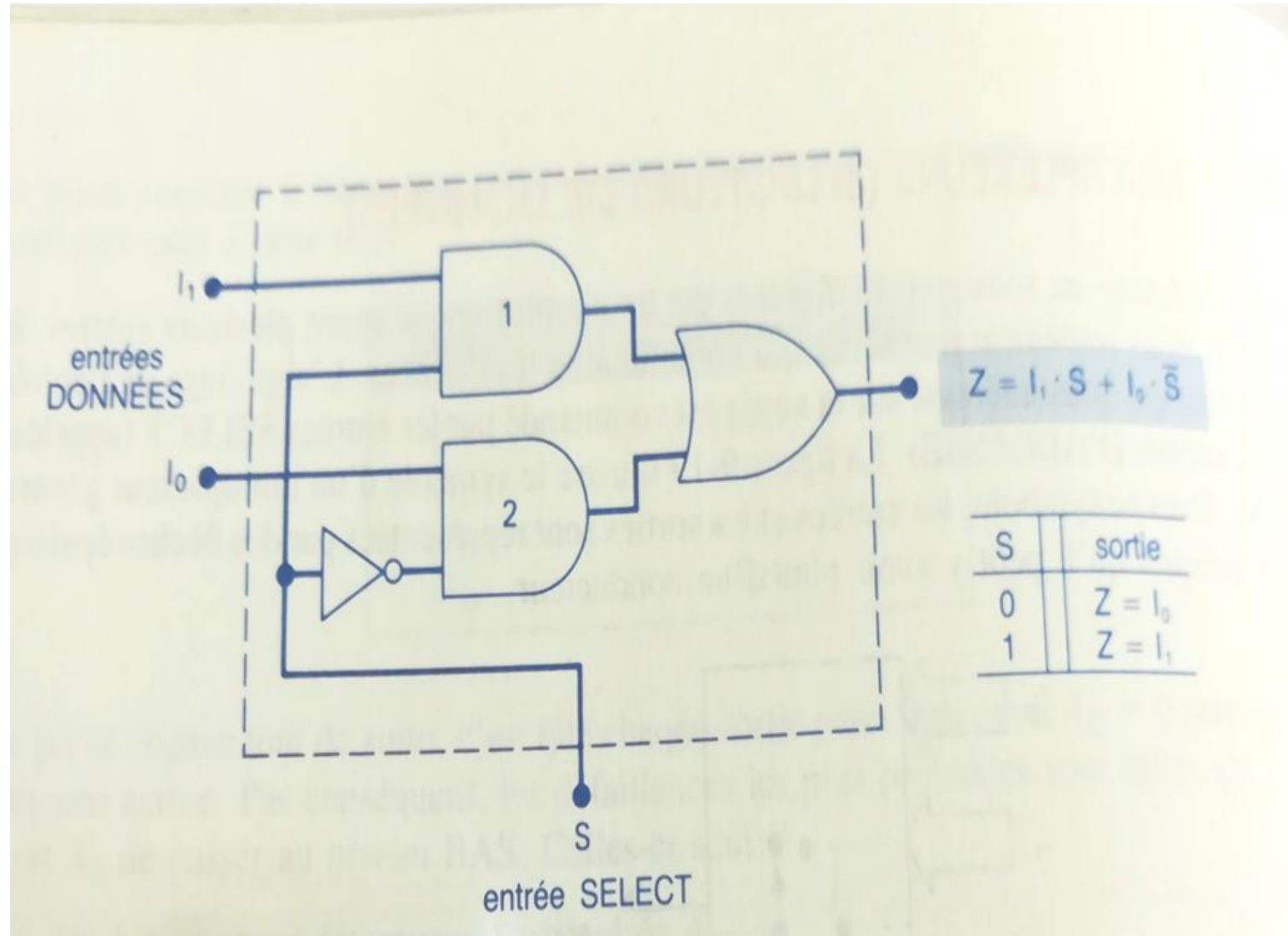


Un multiplexeur se comporte comme un commutateur dans lequel un code numérique appliqué aux entrées SELECT commande les entrées de données qui sont raccordées à la sortie.

Un multiplexeur choisit donc une source de données d'entrées parmi N et transmet celles-ci à la seule voie de sortie existante: On parle de

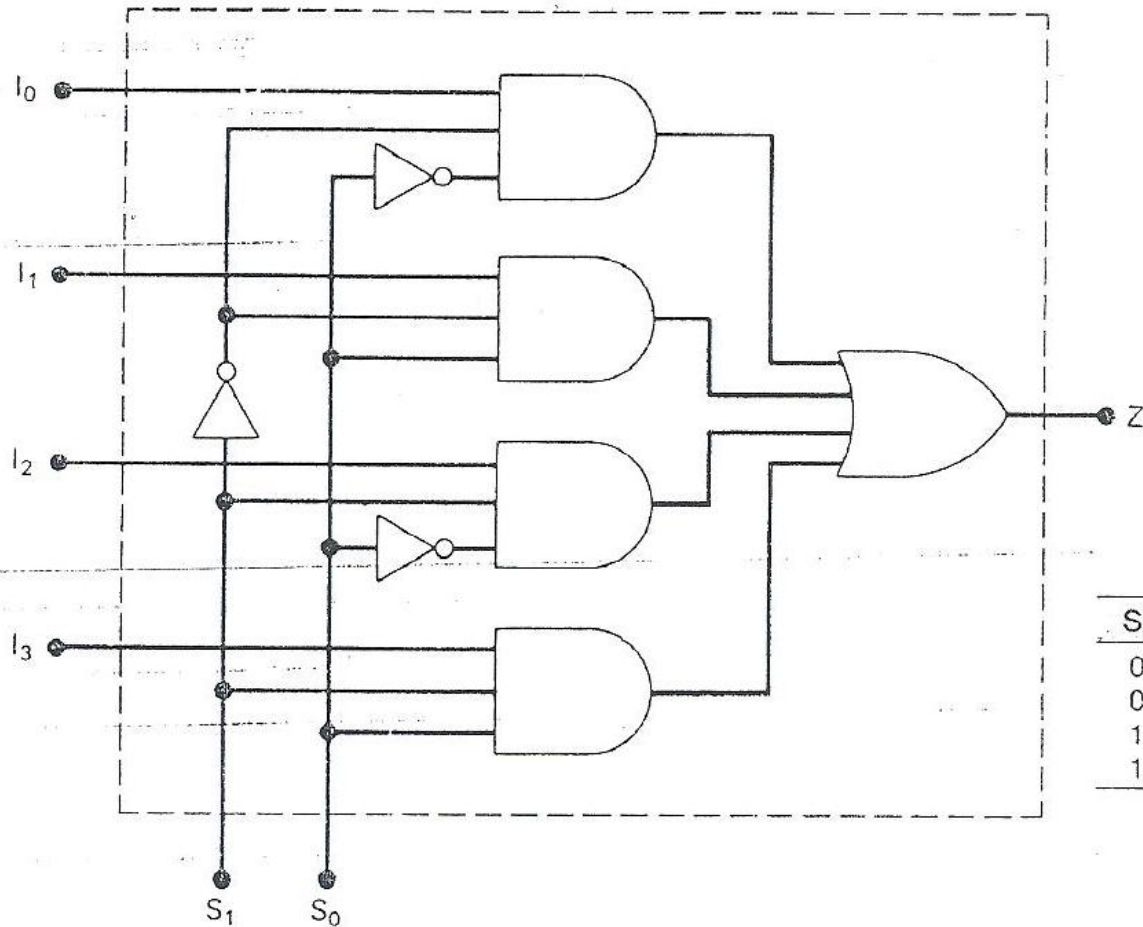
multiplexage.

a) Multiplexeur à deux entrées



b) Multiplexeur à quatre entrées

FIGURE Multiplexeur à quatre entrées.



On peut se procurer dans le commerce des multiplexeurs à deux, quatre, huit et seize entrées fabriqués selon la technologie TTL ou CMOS.

Ces CI de base peuvent être combinés afin de multiplexer un nombre d'entrées encore plus grand.

C) Multiplexeur à huit entrées

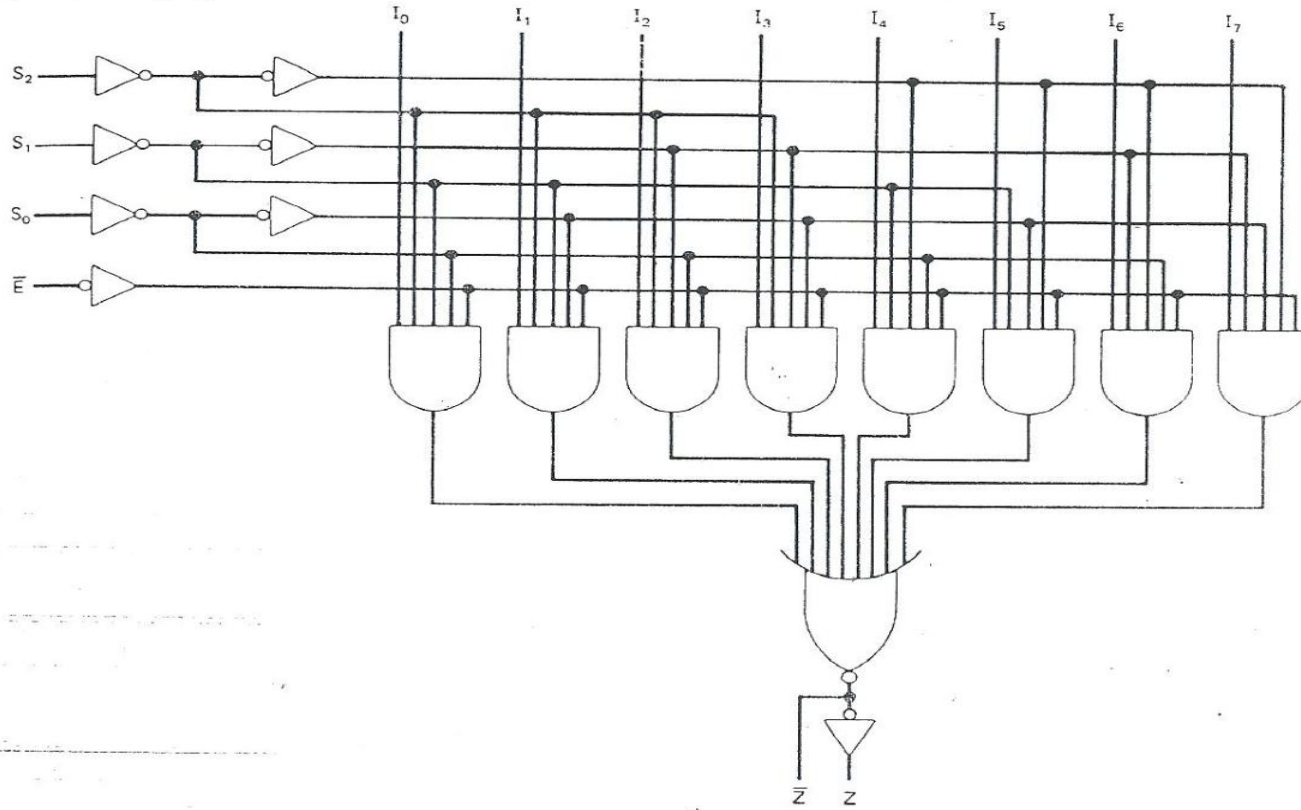
Exemple du MUX 74151 (74LS151, 74HC151): Ce multiplexeur dispose d'une entrée VALIDATION \bar{E} , et il fournit la sortie normale et la sortie complémentée.

Quand $\bar{E} = 0$, les entrées de sélection choisissent une entrée de données (I_0 à I_7) dont les valeurs se retrouvent sur la sortie Z .

Quand $\bar{E} = 1$, le multiplexeur est invalidé de sorte que $Z = 0$ quel que soit le code d'entrée de sélection.

La figure suivante donne le schéma logique du MUX 74151, sa table de vérité ainsi que son symbole logique.

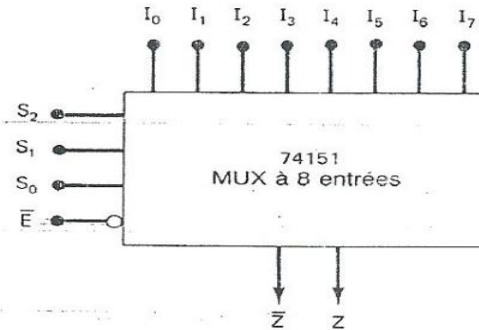
FIGURE a) Schéma logique du multiplexeur 74151; b) sa table de vérité; c) son symbole logique. (Gracieuseté de Fairchild, filiale Schlumberger).



a)

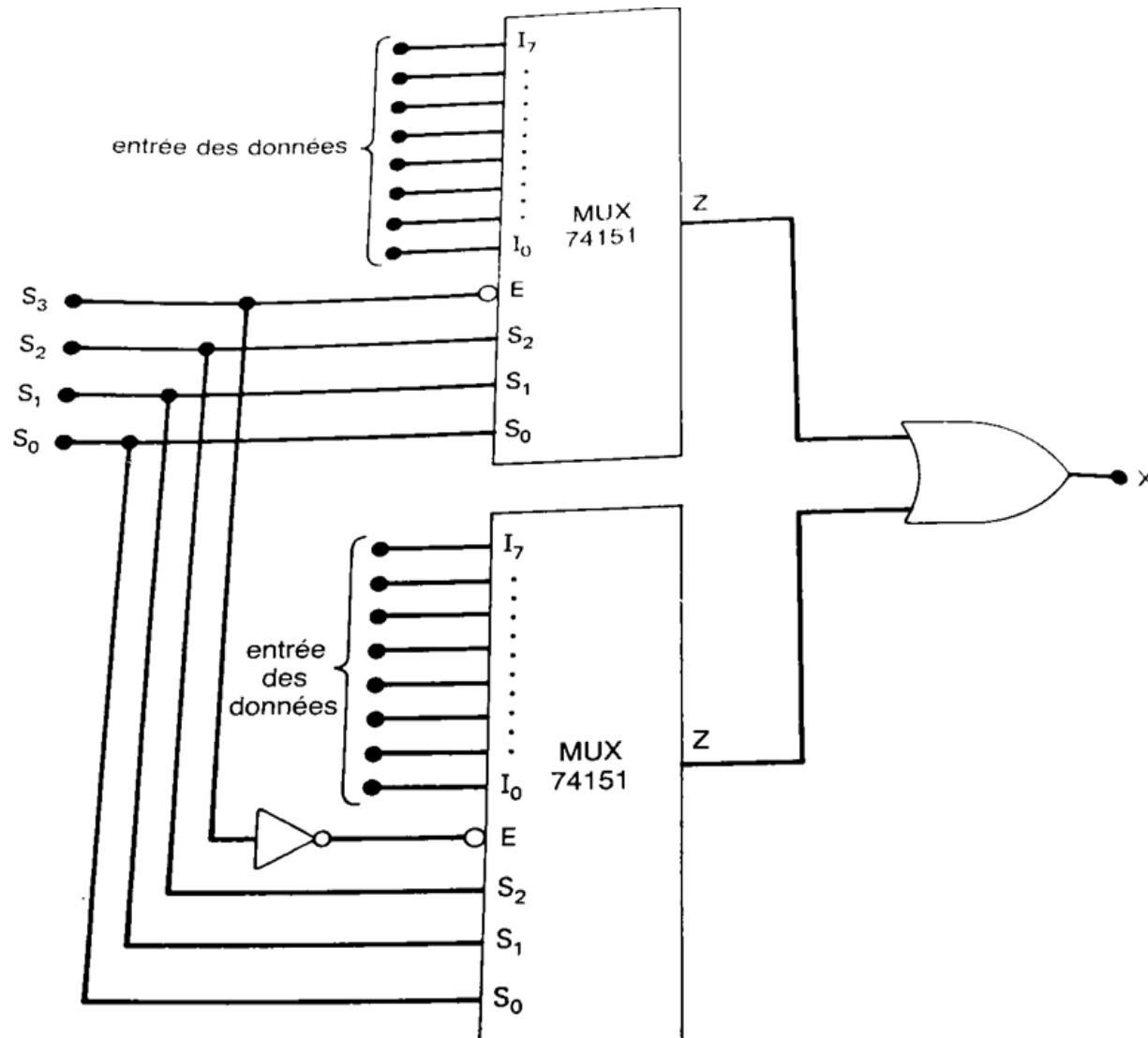
entrées				sorties	
\bar{E}	S_2	S_1	S_0	\bar{Z}	Z
H	X	X	X	H	B
B	B	B	B	\bar{I}_0	I_0
B	B	B	H	\bar{I}_1	I_1
B	B	H	B	\bar{I}_2	I_2
B	B	H	H	\bar{I}_3	I_3
B	H	B	B	\bar{I}_4	I_4
B	H	B	H	\bar{I}_5	I_5
B	H	H	B	\bar{I}_6	I_6
B	H	H	H	\bar{I}_7	I_7

b)



c)

Exemple de deux MUX 74151 (à 8 entrées)
combinés pour réaliser un MUX à 16 entrées:



VI) APPLICATIONS DES MULTIPLEXEURS

Les applications des multiplexeurs dans le domaine des techniques numériques sont nombreuses et variées.

On en retrouve dans les circuits de sélection de données, d'aiguillage de données, d'ordonnancement des opérations, de conversion parallèle-série, de génération de formes d'ondes et de production de fonction logique.

Exemple de la génération d'une fonction logique:

On peut utiliser des multiplexeurs pour matérialiser directement des fonctions logiques à partir d'une table de vérité sans devoir passer le processus de simplification. Quand un multiplexeur est utilisé à cette fin, les entrées de sélection reçoivent les variables logiques et chaque entrée de données est raccordée en permanence à un 0 ou à 1 de façon à respecter la table de vérité.

La figure ci-dessous donne l'exemple d'un MUX à 8 entrées utilisé pour matérialiser un circuit logique qui fonctionne conformément aux indications de la table de vérité.

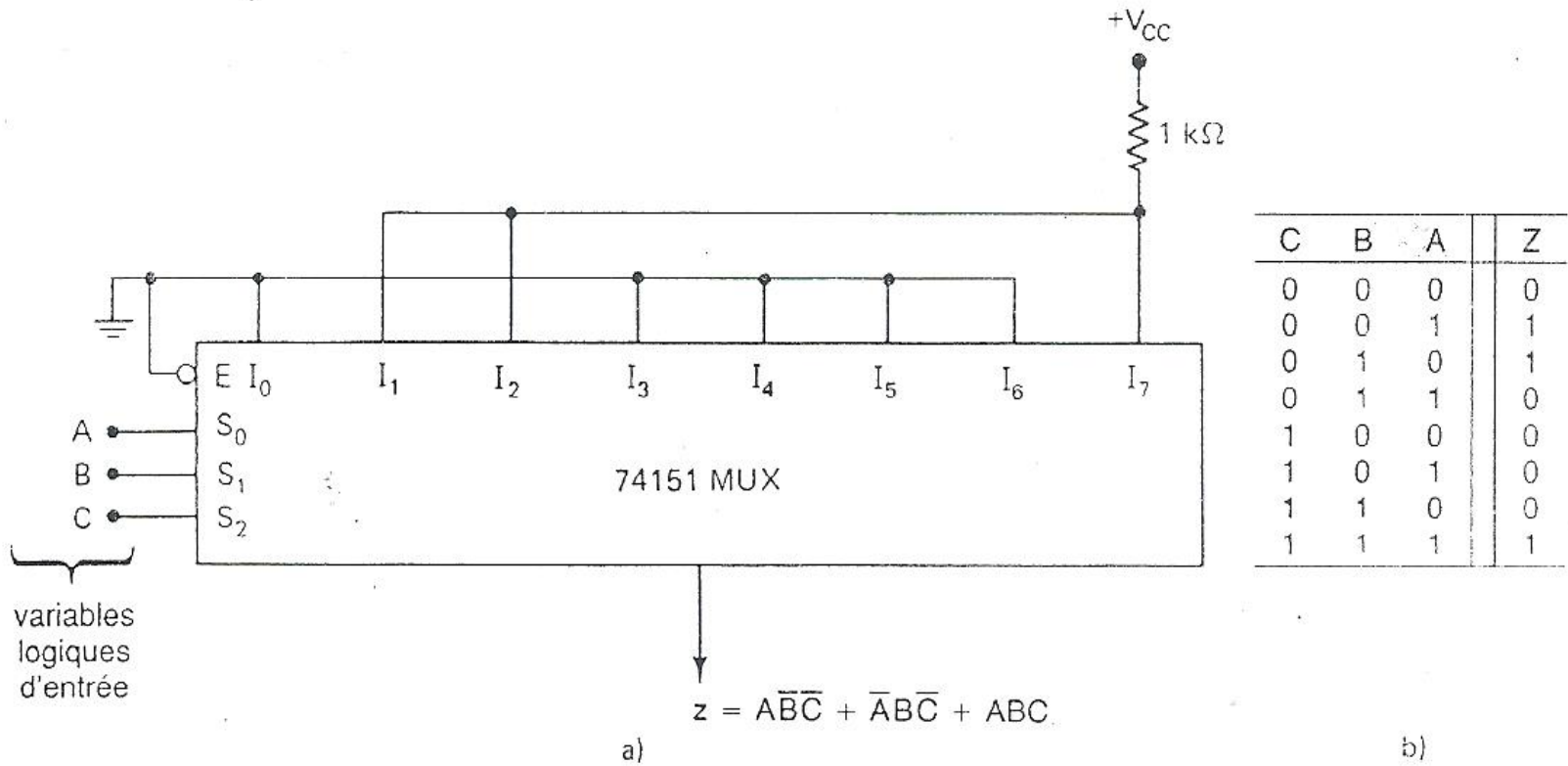


FIGURE Utilisation d'un multiplexeur pour matérialiser la fonction logique décrite par la table de vérité.

Les variables d'entrée A, B, C sont raccordées respectivement à S_0 , S_1 et S_2 , et ce sont les niveaux appliqués à ces entrées qui déterminent l'entrée de données qui est transférée à la sortie Z.

L'examen de la table de vérité montre que Z doit être à 0 quand CBA=000. Ainsi, on raccorde l'entrée I₀ du MUX à 0. De la même façon, Z est supposée être à 0 quand CBA=011, 100, 101 et 110, de sorte que les entrées I₃, I₄, I₅, et I₆ sont raccordées en permanence à la masse (niveau BAS). L'autre groupe de conditions CBA doit donner Z=1, de sorte qu'il faut raccorder en permanence les entrées du MUX I₁, I₂, et I₇ à un niveau HAUT.

On remarque que toute table de vérité à 3 variables peut être matérialisée au moyen d'un multiplexeur à 8 entrées. C'est une méthode beaucoup plus efficace souvent que l'utilisation de portes logiques distinctes. Par exemple, la somme des produits correspondant à la table de vérité de la figure précédente donne $Z = A\bar{B}\bar{C} + \bar{A}B\bar{C} + ABC$

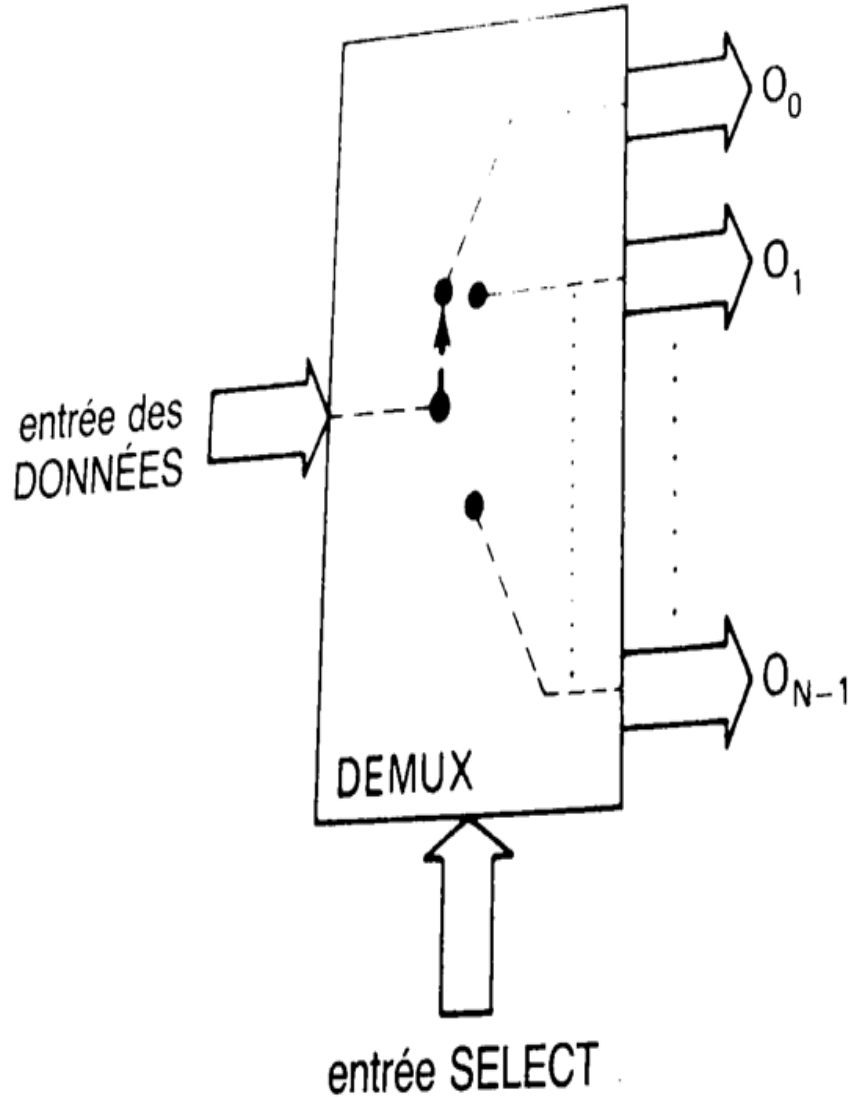
Et comme il est impossible de simplifier cette expression au moyen de l'algèbre ou d'un diagramme K, sa matérialisation nécessitera donc 3 INVERSEURS et 4 portes NON-ET, soit au total 2 circuits intégrés.

Notons aussi qu'un concepteur logiciel peut utiliser un MUX à 3 entrées de sélection (par exemple le 74151) pour construire une fonction logique à 4 variables.

VII- DEMULTIPLIXEURS (DISTRIBUTEURS DE DONNEES)

On a vu qu'un multiplexeur est sollicité par plusieurs entrées, mais ne transmet qu'une de ces dernières à la sortie. Un démultiplexeur effectue l'opération inverse: il n'a qu'une entrée et dirige celle-ci vers une sortie parmi plusieurs sorties:

Schéma général d'un démultiplexeur:



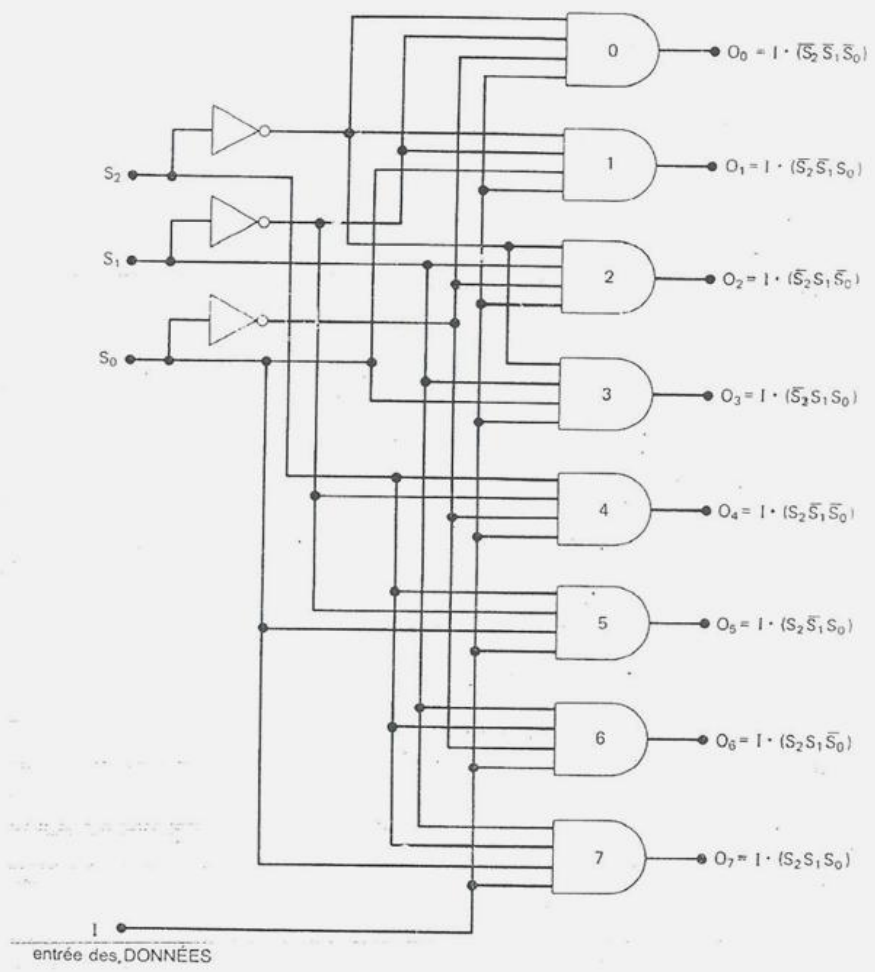
les données d'entrée sont
communiquées à une seule des
sorties, déterminée par le
code d'entrée de sélection.

Démultiplexeur entrée une voie, sortie huit voies:

Son schéma logique et sa table de vérité sont donnés à la figure suivante. La ligne d'entrée de données unique I est connectée aux huit portes ET, mais une seule de ces portes est validée par les entrées SELECT.

Par exemple, quand $S_2S_1S_0=011$, seule la porte ET N°3 est validée, et l'entrée de données I se retrouve sur la sortie 03.

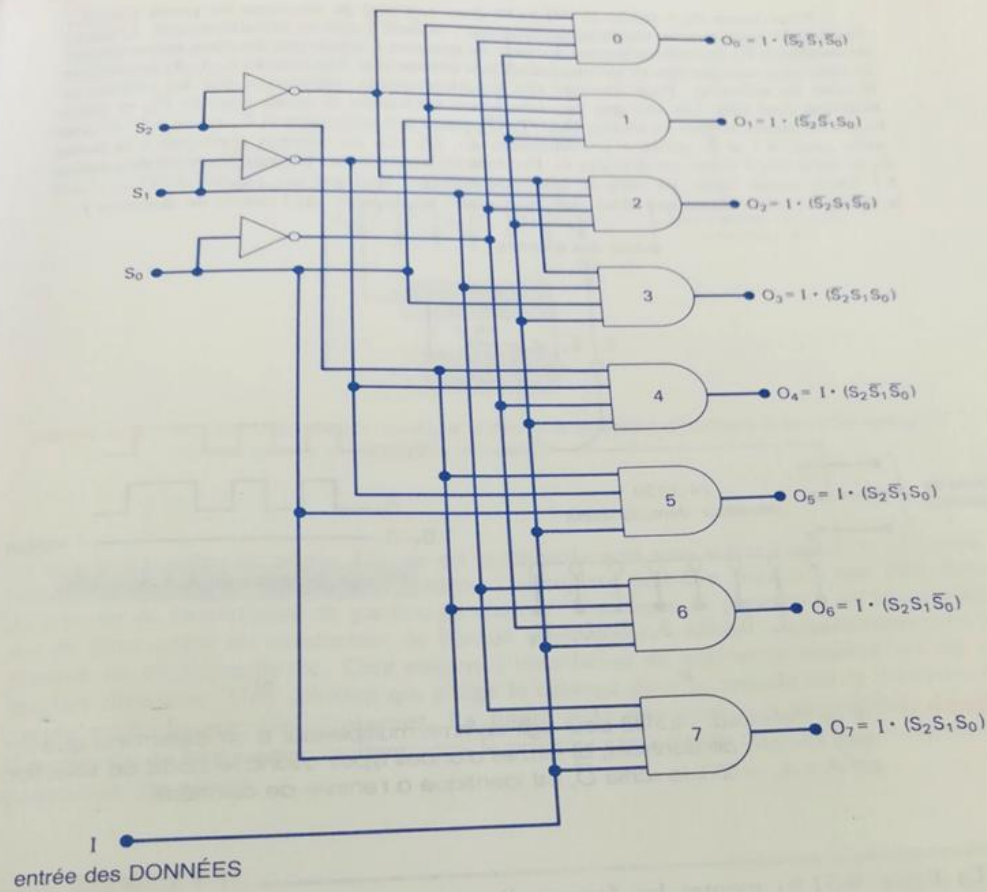
Le circuit de ce démultiplexeur est très semblable à celui du décodeur entrée trois voies, sortie huit voies étudié auparavant, si on excepte le fait qu'une quatrième entrée (I) a été ajoutée à chaque porte. Or, de nombreux décodeurs intégrés ont une entrée VALIDATION, ce qui correspond à une entrée supplémentaire pour les portes du décodeur. Ce genre de boîtier de décodeur peut donc servir de démultiplexeur, les entrées du code binaire (par exemple A,B,C) jouant le rôle des entrées SELECT et l'entrée VALIDATION jouant le rôle de l'entrée de DONNEES I. C'est donc une des raisons pour laquelle les fabricants de CI donnent souvent le nom de décodeur/démultiplexeur à ce dispositif puisqu'il peut servir à l'une ou l'autre de ces fonctions.



code SÉLECTION			sorties							
S ₂	S ₁	S ₀	O ₇	O ₆	O ₅	O ₄	O ₃	O ₂	O ₁	O ₀
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

Nota: ↑ est l'entrée de données

FIGURE . . . Démultiplexeur entrée une voie, sortie huit voies.



entrée des DONNÉES

code SÉLECTION			sorties							
S ₂	S ₁	S ₀	O ₇	O ₆	O ₅	O ₄	O ₃	O ₂	O ₁	O ₀
0	0	0	0	0	0	0	0	0	1	0
0	0	1	0	0	0	0	0	1	0	0
0	1	0	0	0	0	0	1	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	1	0	0	0	0	0
1	0	1	0	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

Note: 1 est l'entrée de données

Démultiplexeur entrée une voie, sortie huit voies.

Exemple du décodeur 74LS138 utilisé ici dans son rôle de démultiplexeur:

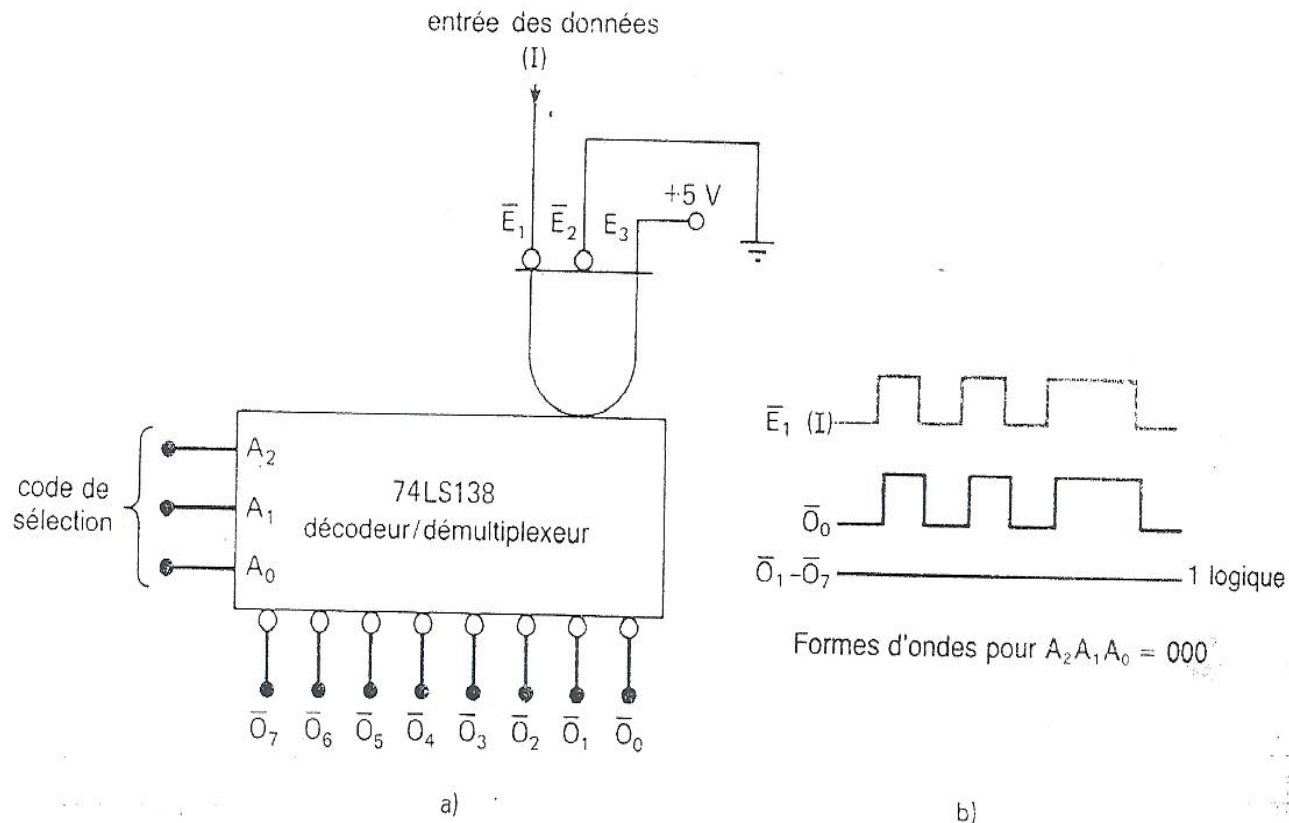
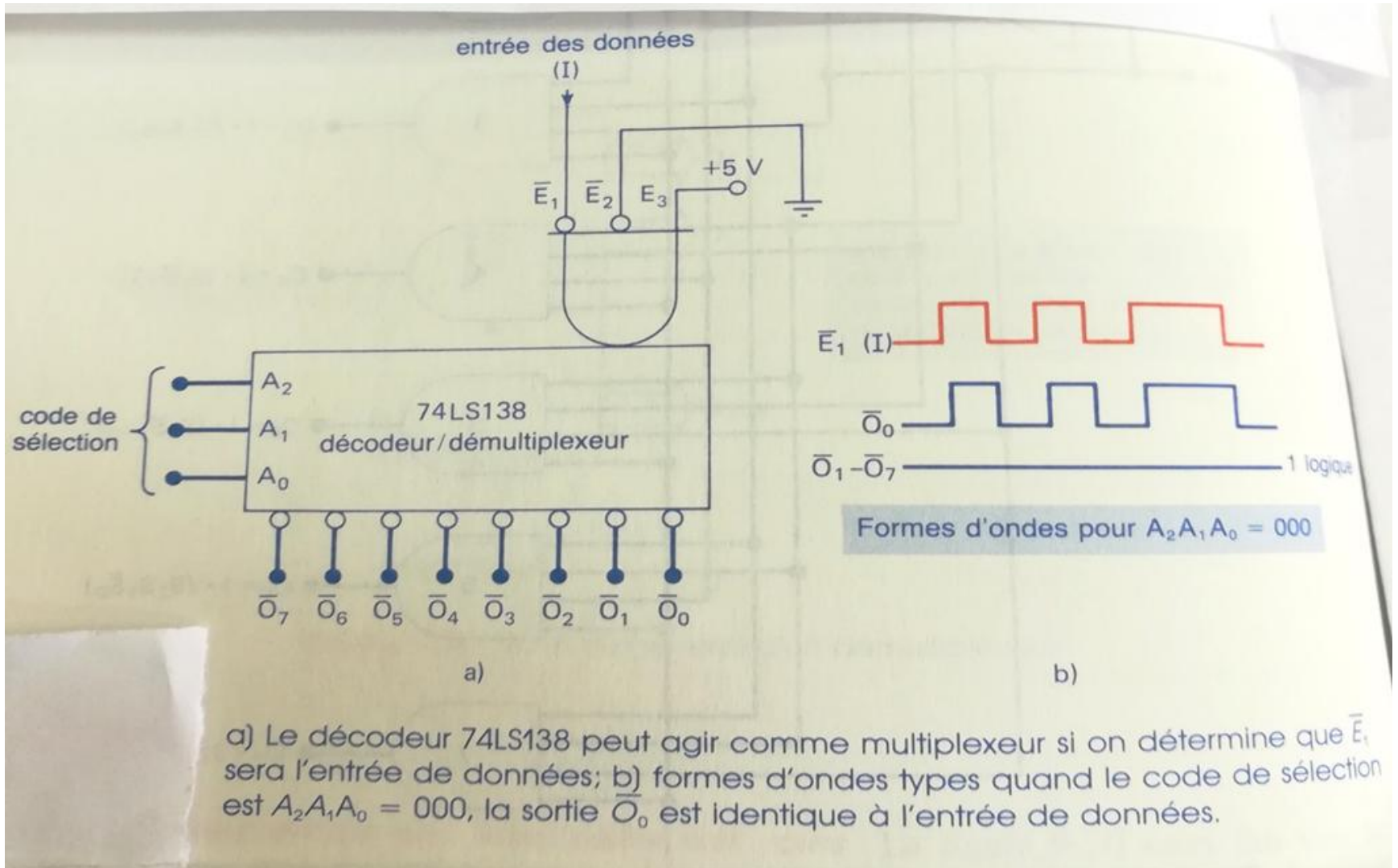


FIGURE 10-10 a) Le décodeur 74LS138 peut agir comme multiplexeur si on détermine que \bar{E}_1 sera l'entrée de données; b) formes d'ondes types quand le code de sélection est $A_2 A_1 A_0 = 000$, la sortie \bar{O}_0 est identique à l'entrée de données.

Exemple du décodeur 74LS138 utilisé ici dans son rôle de démultiplexeur:



L'entrée de validation \overline{E}_1 est utilisée comme l'entrée de données I, tandis que les deux autres entrées de validation sont gardées en permanence à leur niveau vrai. Les entrées A2 A1 A0 accueillent le code de sélection.

Par exemple, pour le code A2 A1 A0 =000, seule la sortie \overline{o}_0 est ouverte et les autres sorties sont gardées au niveau HAUT. \overline{o}_0 suit les niveaux appliqués à la borne \overline{E}_1 qui est l'entrée de données I (voir les formes d'ondes sur la figure).