FONCTIONNEMENT DES SYSTEMES2

TECHNIQUES NUMÉRIQUES
TRAVAUX PRATIQUES AVANCÉS

T-PELN-207 (30h, 60%)

- Notes provisoires
- Protocoles

BACHELIER EN INFORMATIQUE ET SYSTÈMES
Finalité Réseaux et Télécommunications
Cycle1 Bloc1
Michelle VANDEVILLE
michelle.vandeville@heh.be
Maître-assisante
Ing E/e

RECOMMANDATIONS.

Analyser la **fiche ECTS** de cette activité d'apprentissage sur le site de la HEH.

Ces notes non corrigées constituent une aide à l'étude.
Il est vivement recommandé d'assister à **tous les cours**. La prise de notes est **indispensable**.

Cette présence vous apportera:

- des **explications complémentaires**
- des **éventuelles corrections** (évolutions)
- des **exercices complémentaires corrigés**
- les **résumés** de chaque chapitre

Je reste toujours à la disposition des étudiants, il suffit de me contacter par mail (quand je ne suis pas à la HEH) à l'adresse suivante:

michelle.vandeville@heh.be
vandeville.m@gmail.com

Bon travail! Et bonne réussite!

PLAN DE L'ACTIVITÉ D'APPRENTISSAGE

HEH be
Sciences
et technologies

PARTIE I

COURS DE

Travaux pratiques

PROBLÈMES DE LOGIQUE COMBINATOIRE

I. ARITHMÉTIQUE BINAIRE: OPÉRATIONS ET CIRCUITS

1. Additionneurs

A) semi-additionneur (half adder)

C'est un circuit d'addition qui ajoute deux éléments binaires de même rang.

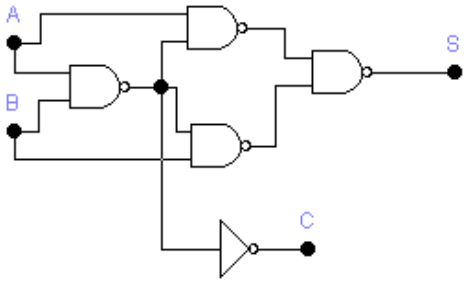
S = A + B (S → Somme et + → Plus)

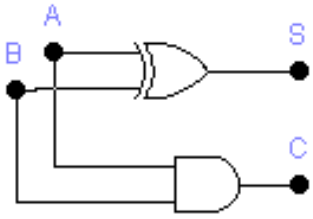
Table de vérité C est le report (ou carry) au rang i+1 du rang i

A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

De cette table, on peut tirer les équations logiques:

$$S = A \oplus B$$
$$C = A.B$$



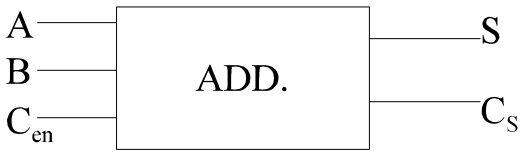


B)additionneur (full adder)

C'est un circuit d'addition qui ajoute deux éléments binaires de même rang, mais qui tient compte du report éventuel du rang précédent.

Table de vérité C_{en}= entrée du bit de report = report du rang précédent
C_s = sortie du bit de report

A	B	C_{en}	S	C_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



De cette table, on peut tirer les équations logiques:

$$S = \overline{A}\overline{B}C_{en} + \overline{A}B\overline{C_{en}} + A\overline{B}\overline{C_{en}} + ABC_{en}$$

$$S = \overline{A}\left(\overline{B}C_{en} + B\overline{C_{en}}\right) + A\left(\overline{B}\overline{C_{en}} + BC_{en}\right)$$

$$S = \overline{A}\left(B \oplus C_{en}\right) + A\left(\overline{B \oplus C_{en}}\right)$$

$$S = A \oplus B \oplus C_{en}$$

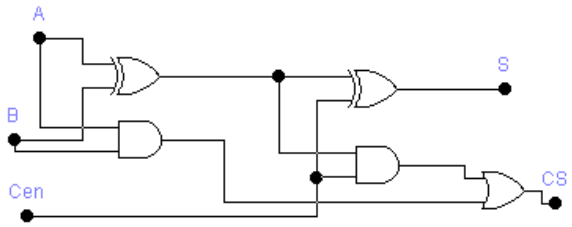
C_S =

C_S =

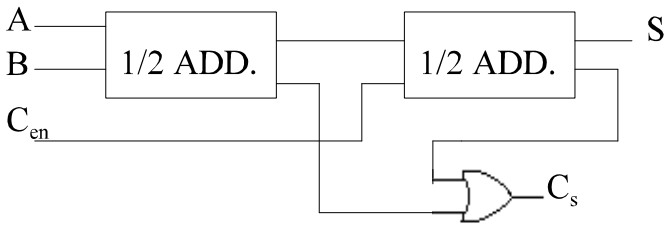
C_S =

$$C_S = A.B + (A \oplus B).C_{en}$$

De ces équations, on peut déduire les schémas suivants:



On peut donc réaliser l'additionneur complet avec 2 semi-additionneurs et un circuit ou.



2. Soustracteurs

A) semi-soustracteur (half subtractor)

C'est un circuit de soustraction qui soustrait un élément binaire d'un autre de même rang..

$$\mathbf{D = A - B}$$ (D → Différence et - → Moins)

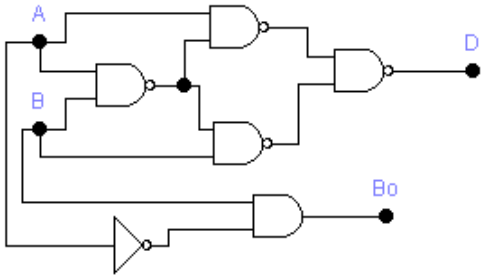
Table de vérité B0 est l'emprunt (Borrow) au rang supérieur.

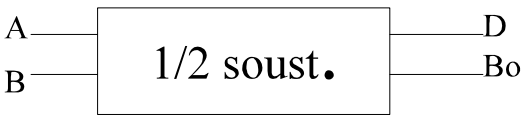
A	B	D	B0
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

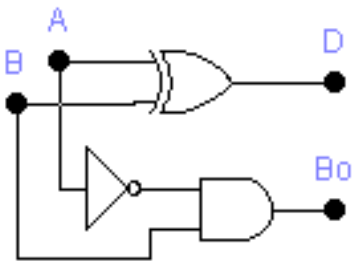
De cette table, on peut tirer les équations logiques:

$$D = A \oplus B$$

$$Bo = \overline{A}.B$$







Remarque: La différence entre le semi-additionneur et le semi-soustracteur est tout simplement que l'on entre $\overline{A}$ pour le semi-soustracteur dans le ET qui donne Bo et A pour le semi-additionneur dans le ET qui donne C.

B)Soustracteur (full subtractor)

C'est un circuit de soustraction qui soustrait un élement binaire d'un autre de même rang, mais qui tient compte du retrait éventuel d'un emprunt du rang précédent.

Table de vérité Bo_{en}= retrait d'un emprunt du rang précédent
Bos = emprunt

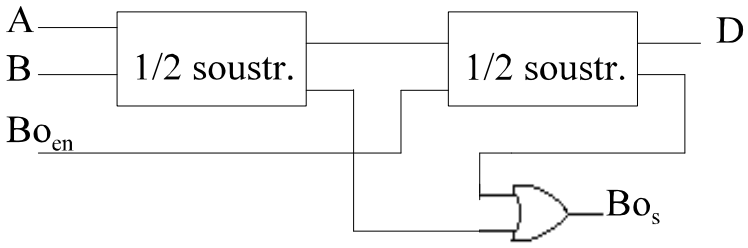
A	B	Bo_{en}	D	Bo_s
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

De cette table, on peut tirer les équations suivantes:

$$D = A \oplus B \oplus Bo_{en}$$

$$Bo_S = \overline{A}.B + \overline{(A \oplus B)}..Bo_{en}$$

On peut réaliser le soustracteur complet avec 2 semi-soustracteur et un circuit OU



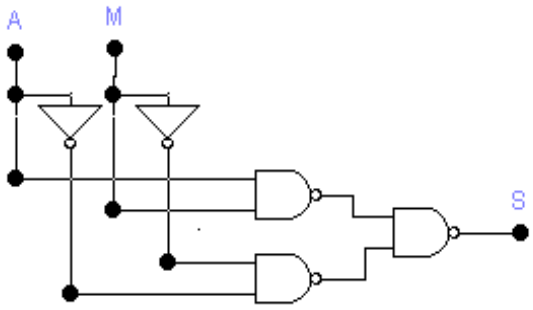
3. Additionneur-soustracteur

On peut réaliser assez facilement un additionneur-soustracteur pour 2 éléments binaires en faisant un aiguillage par la commande de MODE qui fera entrer A (ADD) ou $\overline{A}$ (SOUSTR.) Puisque les fonctions logiques sont identiques.

Par exemple:

M=1 → ADD → A → $S = M.A + \overline{M}.\overline{A}$

M=0 → SOUSTR. → $\overline{A}$ $S = \overline{\overline{M.A}.\overline{\overline{M}.\overline{A}}}$

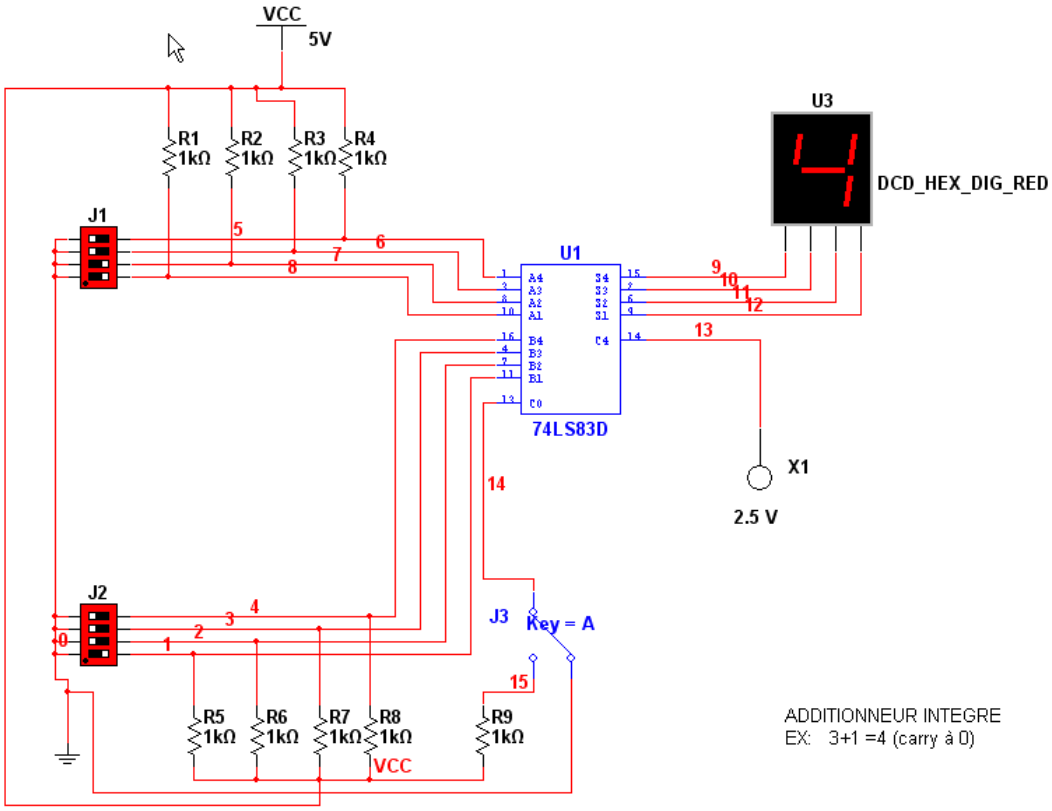


4. Additionneur parallèle intégré

II existe plusieurs additionneurs parallèles dans des boîtiers CI. L'un des plus courants est un boîtier d'additionneurs parallèles de 4 bits comprenant 4 additionneurs complets (AC) et leurs connexions et les circuits de l'anticipation du report nécessaires à un fonctionnement rapide. Les 7483A, 74LS83A, 74283 et 74LS283 sont tous des boîtiers d'additionneurs parallèles 4 bits de la famille TTL. Les 283 sont identiques aux 83, à l'exception du fait que Vcc et la masse sont respectivement sur les broches 16 et 8; c'est désormais la norme sur toutes les nouvelles puces d'avoir les broches d'alimentation et de la masse aux coins du boîtier. Le 74HC283 est la version CMOS rapide du même additionneur parallèle 4 bits.

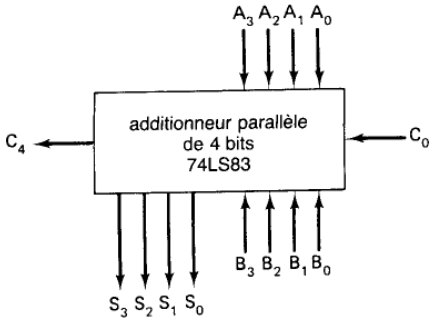
On peut voir à la figure suivante le symbole logique de l'additionneur parallèle de 4 bits 74LS83. Les entrées de ce C I sont deux groupes de 4 bornes $A_3A_2A_1A_0$ et $B_3B_2B_1B_0$ servant à appliquer deux nombres de 4 bits, et le report C_0, appliqué au rang de poids faible. Les sorties sont les 4 bits de somme $S_3S_2S_1S_0$ et le report C_4 quittant l'étage de poids fort. Les bits de somme sont souvent désignés $\Sigma_3\Sigma_2\Sigma_1\Sigma_0$ où Σ est la lettre grecque *sigma*.

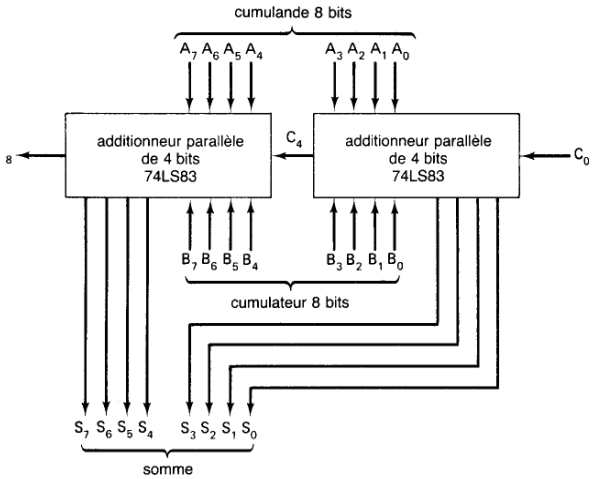
Simulation:



Montage en cascade d'additionneurs parallèles

II est possible de raccorder deux ou plusieurs additionneurs parallèles en cascade afin d'additionner des





nombres ayant un plus grand nombre de bits. La figure ci-dessous nous montre deux additionneurs 74LS83 sont reliés afin de pouvoir additionner deux nombres de 8 bits. L'additionneur de droite additionne les 4 bits de poids le plus faible des nombres. La sortie C_4 de cet additionneur est reliée à l'entrée report du second additionneur: ce dernier additionne les 4 bits de poids le plus fort des nombres. Les huit sorties de la somme indiquent le résultat de l'addition de deux nombres de 8 bits. C_8 est le report fourni par l'étage du rang de poids le plus fort du second additionneur. Cette sortie peut devenir un bit de dépassement ou servir de report pour un autre étage d'additionneur, si on traite des nombres binaires encore plus longs.

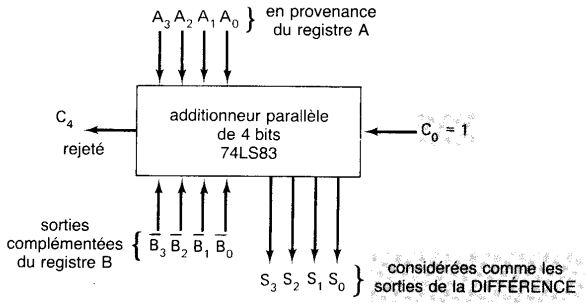
5. NOTATION COMPLÉMENT À 2

Aujourd'hui, la majorité des ordinateurs expriment leurs nombres négatifs selon la notation en complément à 2 et utilisent cette notation quand ils effectuent des soustractions. L'addition et la soustraction des nombres signés se résument à une simple addition si on exprime les nombres négatifs selon la notation en complément à 2.

Soustraction

Quand on soustrait en utilisant la notation complément à 2, le diminuteur est complémenté à 2 puis additionné au diminuande (le nombre dont on soustrait le diminuteur). Supposons que le diminuande est déjà mémorisé dans l'accumulateur (registre A). Le diminuteur se voit alors transféré dans le registre B (dans un ordinateur ce nombre provient de la mémoire) puis est complémenté à 2 avant d'être additionné au nombre contenu dans le registre A. Les sorties somme de l'additionneur représentent la différence entre le diminuande et le diminuteur.

Le circuit additionneur parallèle déjà examiné peut être modifié pour soustraire si on prévoit une façon de prendre le complément à 2 du nombre contenu dans le registre B. On sait que le complément à 2 d'un nombre binaire s'obtient en complémentant chaque bit et en ajoutant 1 à son bit de poids le plus faible.



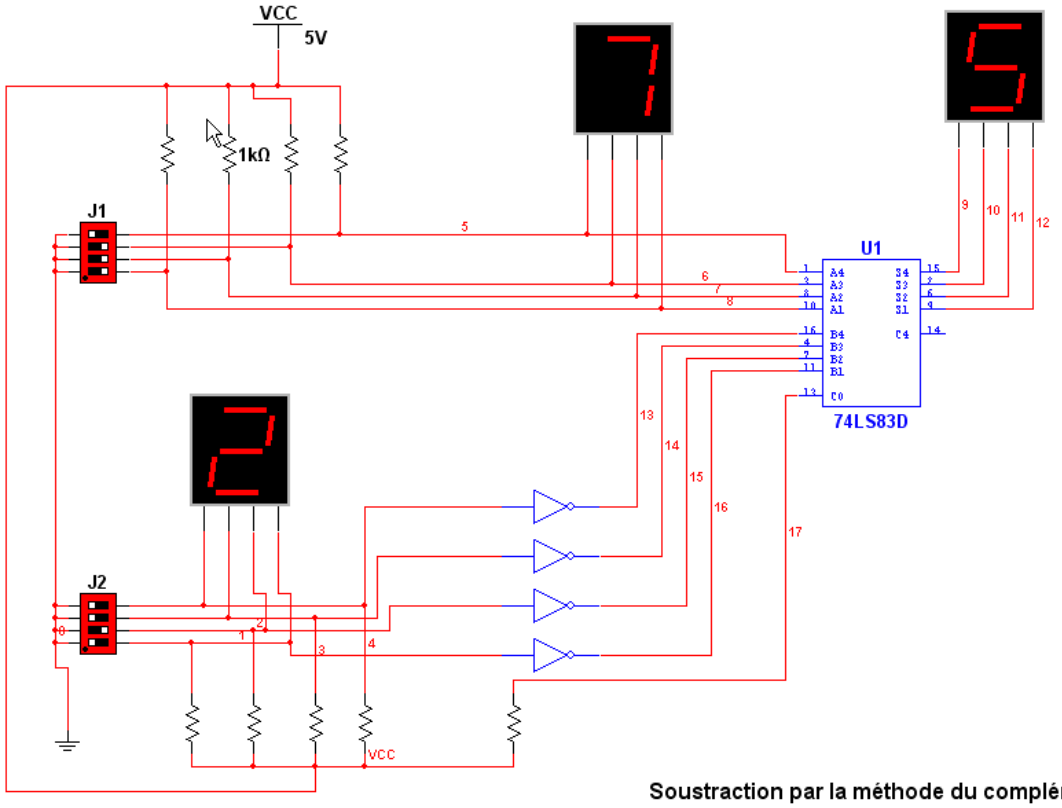
Cette figure montre comment un additionneur peut servir pour la soustraction. On utilise les sorties complémentées du registre B plutôt que les sorties normales; c'est-à-dire qu'on applique aux entrées de l'additionneur $\overline{B_0}, \overline{B_1}, \overline{B_2}, \overline{B_3}$ (se rappeler que B3 est le bit de signe). De cette manière, on complémente chaque bit du nombre B. En outre, on transforme Co en un 1 logique, de sorte qu'on

additionne ainsi un 1 au bit de poids le plus faible de l'additionneur; cette façon de procéder a le même effet que d'additionner 1 au bit de poids le plus faible du registre B afin d'obtenir son complément à 2.

Les sorties S3-S0 présentent un nombre correspondant au résultat de l'opération de soustraction. Évidemment S3 est le bit de signe du résultat et indique qu'il est soit positif, soit négatif. Le report C4 est à nouveau rejeté.

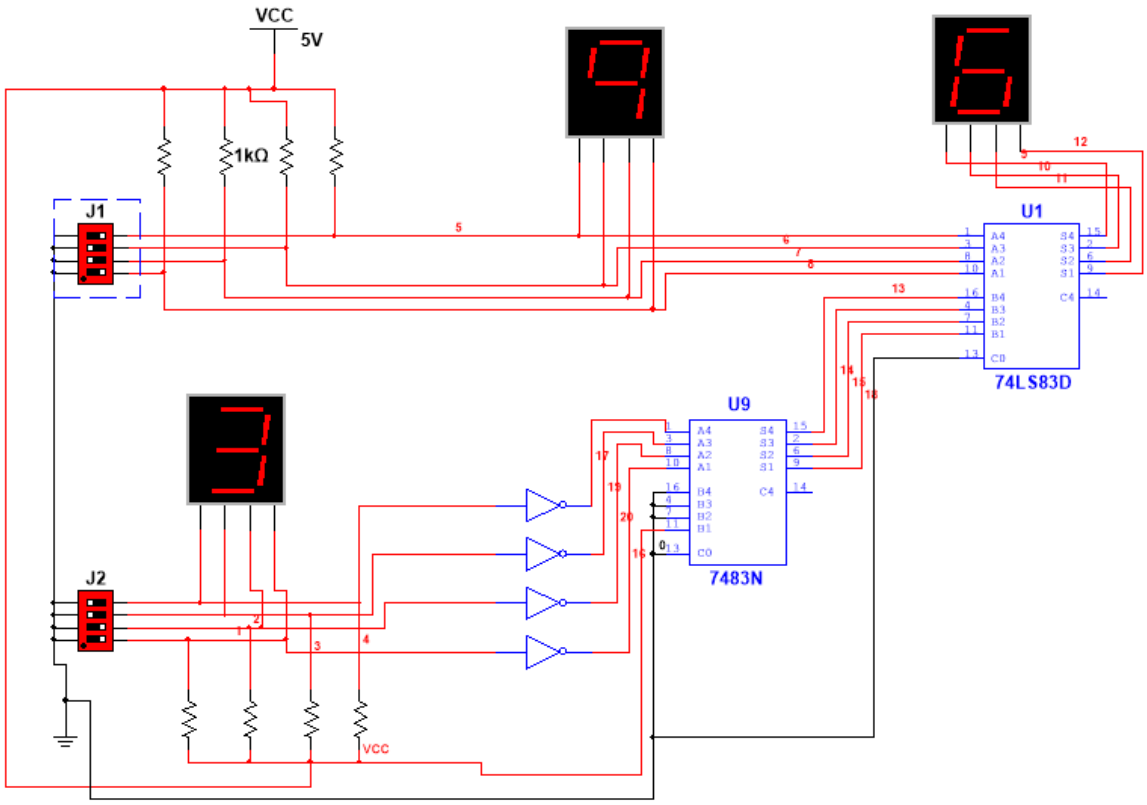
Simulations

A) Soustraction par la méthode du complément à 1 et addition



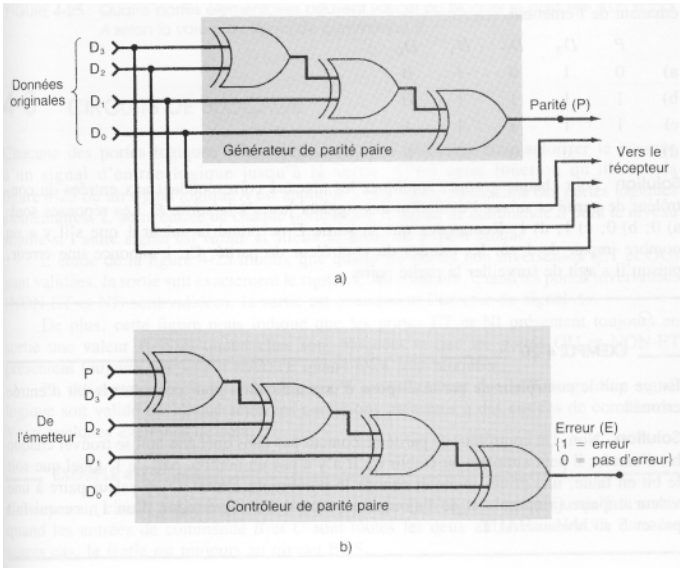
Soustraction par la méthode du complément à 1.

B) Soustraction par la méthode du complément à 2 et addition



6. GÉNÉRATEUR ET CONTRÔLEUR DE PARITÉ

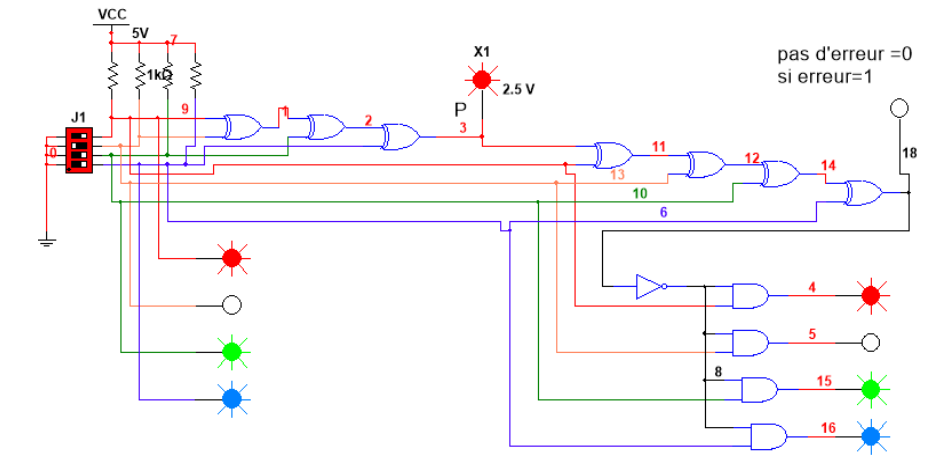
Au chapitre précédent, nous avons vu qu'un émetteur peut joindre un bit de parité aux données qu'il transmet avant de les acheminer au récepteur. Nous avons également appris que le récepteur grâce à cette astuce pouvait détecter les erreurs monobits survenant durant la transmission. La figure suivante nous fait voir un exemple de circuit logique pouvant servir à la production et au contrôle de la parité. Cet exemple particulier porte sur des mots de quatre bits et concerne la parité paire. On peut adapter ces solutions telles quelles à la parité impaire et au nombre de bits que l'on veut.



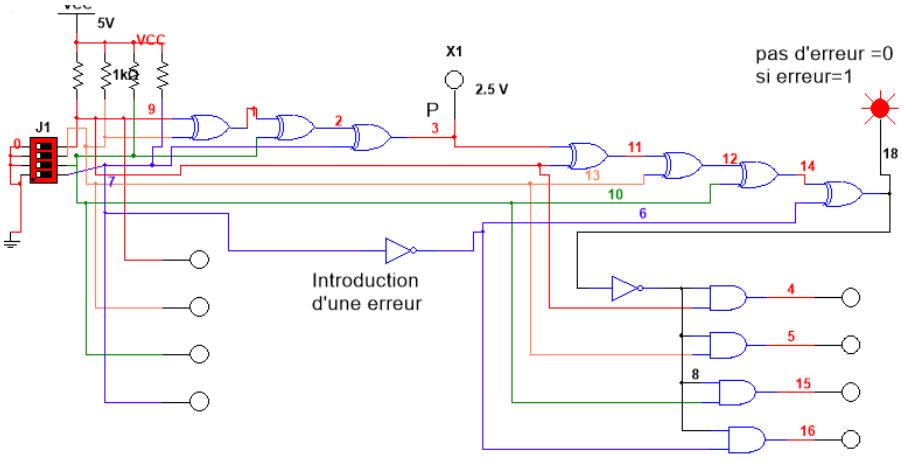
À la figure a), le groupe de données à transmettre est appliqué aux entrées d'un circuit générateur de parité, qui produit en sortie un bit de parité paire, P. Ce bit de parité est communiqué au récepteur en même temps que les données binaires, soit un total de cinq bits. À la figure b), on peut voir ces cinq bits (donnée + parité) appliqués aux entrées du circuit de contrôle de la parité du récepteur, circuit qui fournit en sortie un signal d'erreur, E, qui annonce la présence ou l'absence d'une erreur monobit.

Il n'est pas étonnant de retrouver des portes OU exclusif dans ces circuits, étant donné qu'une porte OU exclusif opère de telle manière qu'elle produit un 1 si un nombre impair de ses entrées sont à 1 et un 0 si un nombre pair de ses entrées sont à 1.
Le contrôle de la parité est utilisé, par exemple, pour augmenter la fiabilité d'un système de transmission ou de stockage de données. La figure suivante montre l'utilisation du circuit précédent en générateur de parité du côté de l'émission et contrôleur de parité du côté de la réception.

Remarquons cependant que la parité ne permet de détecter qu'un nombre impair de bits en erreur dans un mot. Par ailleurs il ne permet pas corriger les erreurs détectées. Pour ce faire il faut utiliser des codes correcteurs d'erreur qui nécessitent plusieurs bits supplémentaires.



Générateur et récepteur sans erreur de transmission



Générateur et récepteur avec erreur de transmission

7. COMPARATEURS DE GRANDEURS

Il s'agit d'un circuit logique combinatoire qui compare deux grandeurs binaires et produit des sorties qui désignent lequel des mots est le plus grand. La figure suivante illustre le symbole logique et la table de vérité du comparateur de grandeurs à 4 bits 74LS85, qui existe également dans les versions 7485 et 74HC85.

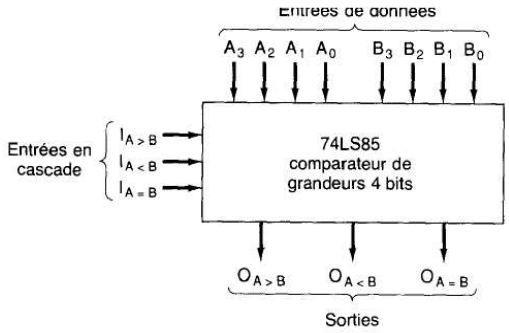


TABLE DE VÉRITÉ

COMPARAISON DES ENTRÉES				ENTRÉES EN CASCADE			SORTIES		
A_3, B_3	A_2, B_2	A_1, B_1	A_0, B_0	$I_{A>B}$	$I_{A<B}$	$I_{A=B}$	$O_{A>B}$	$O_{A<B}$	$O_{A=B}$
$A_3 > B_3$	X	X	X	X	X	X	H	B	B
$A_3 < B_3$	X	X	X	X	X	X	B	H	B
$A_3 = B_3$	$A_2 > B_2$	X	X	X	X	X	H	B	B
$A_3 = B_3$	$A_2 < B_2$	X	X	X	X	X	B	H	B
$A_3 = B_3$	$A_2 = B_2$	$A_1 > B_1$	X	X	X	X	H	B	B
$A_3 = B_3$	$A_2 = B_2$	$A_1 < B_1$	X	X	X	X	B	H	B
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 > B_0$	X	X	X	H	B	B
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 < B_0$	X	X	X	B	H	B
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	H	B	B	H	B	B
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	B	H	B	B	H	B
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	X	X	H	B	B	H
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	B	B	B	H	H	B
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	H	H	B	B	B	B

H = niveau HAUT
B = niveau BAS
X = indifférent

Entrées des données

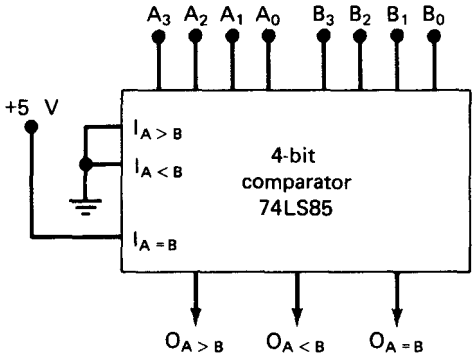
Le 74LS85 compare deux nombres binaires de 4 bits non signés. Le premier, placé sur $A_3A_2A_1A_0$, s'appelle le mot A; l'autre, placé sur $B_3B_2B_1B_0$, s'appelle le mot B. L'expression « mot » sert en technologie numérique à désigner un ensemble de bits véhiculant un certain genre d'information. Dans le cas présent, les mots A et B sont des grandeurs numériques.

Sorties

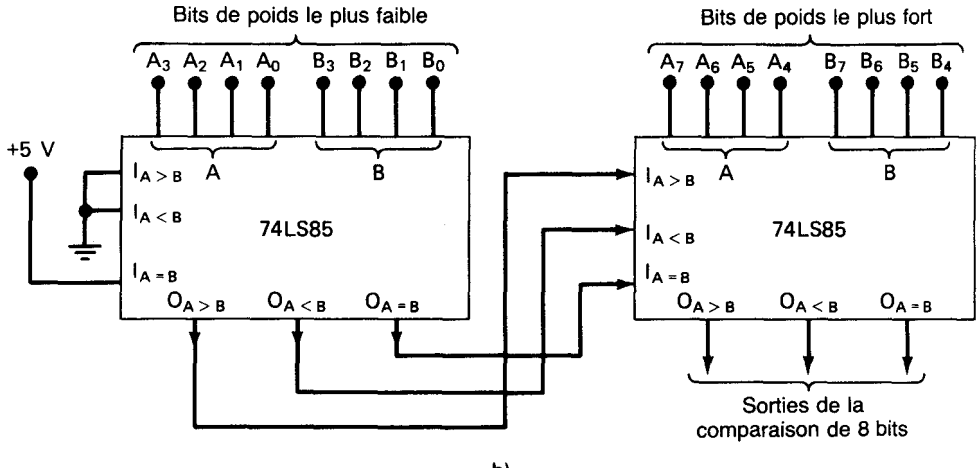
Le 74LS85 possède trois sorties vraies au niveau HAUT; la sortie $O_{A>B}$ qui passe au niveau HAUT quand le mot A est plus grand que le mot B; la sortie OA<B qui passe au niveau HAUT quand le mot A est plus petit que le mot B; enfin, la sortie $O_{A=B}$ qui devient à son niveau vrai (HAUT) quand les deux mots ont la même grandeur.

Entrées en cascade

Le montage en cascade des entrées constitue une façon d'étendre la comparaison à plus de 4 bits. Dans cette solution, on monte en cascade deux ou plusieurs comparateurs de 4 bits. On constate que les entrées en cascade ont les mêmes indices que les sorties correspondantes. Quand une comparaison de 4 bits est effectuée, comme à la figure suivante les entrées en cascade doivent être raccordées selon l'ordre indiqué afin d'obtenir du comparateur des sorties correctes.



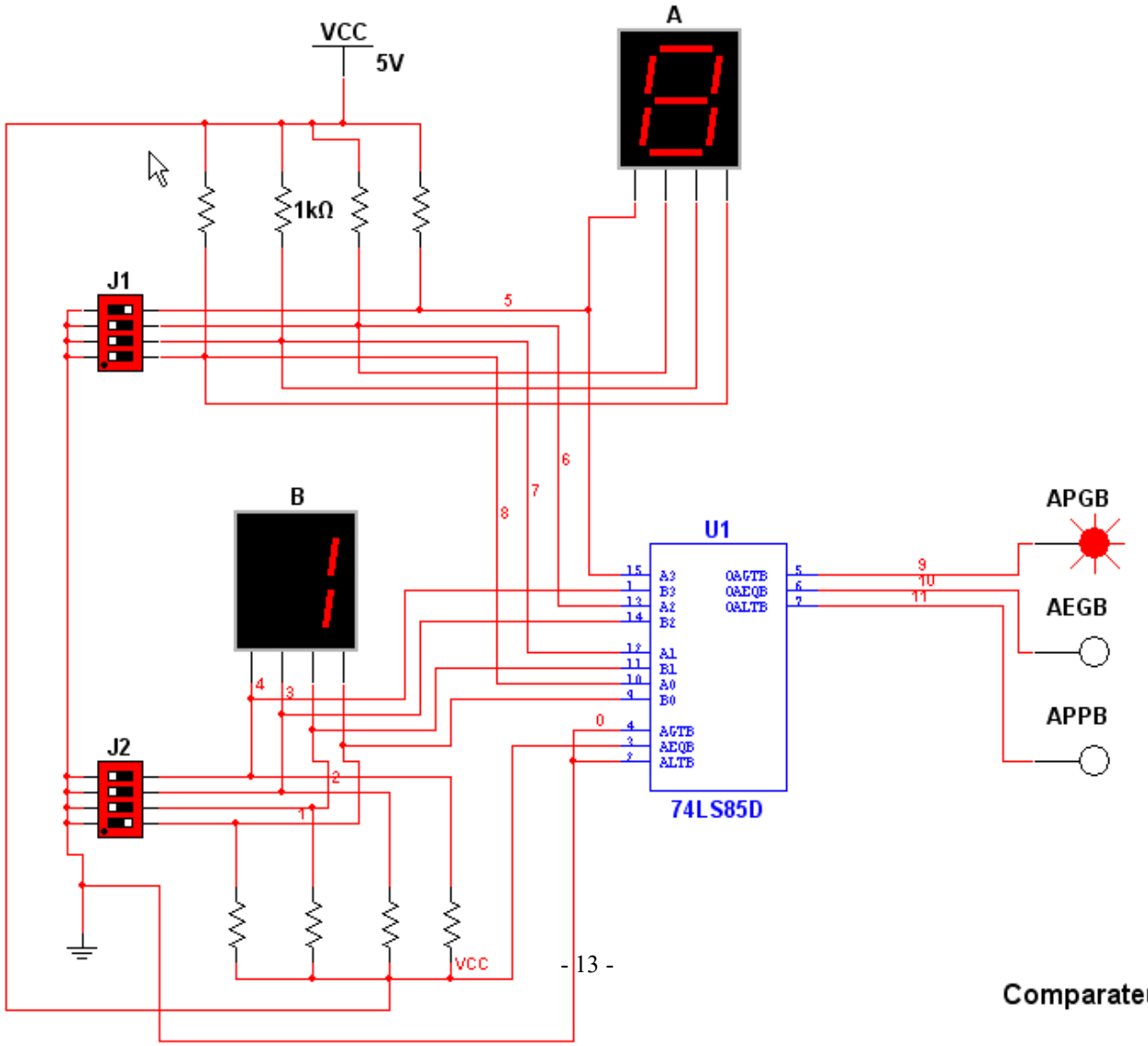
a)



Pour monter en cascade deux comparateurs, il faut que les sorties de celui de poids le plus faible soient connectées aux entrées ayant les mêmes indices dans le comparateur de rang supérieur. C'est ce que montre la figure b), où le comparateur de gauche compare les 4 bits de poids inférieur de deux mots de 8 bits: $A_7A_6A_5A_4A_3A_2A_1A_0$ et $B_7B_6B_5B_4B_3B_2B_1B_0$. Les sorties sont amenées aux entrées en cascade du comparateur e droite, qui confronte déjà les 4 bits de poids le plus fort. Les sorties de ce dernier étage de comparaison contiennent le résultat final de la comparaison des mots de 8 bits.

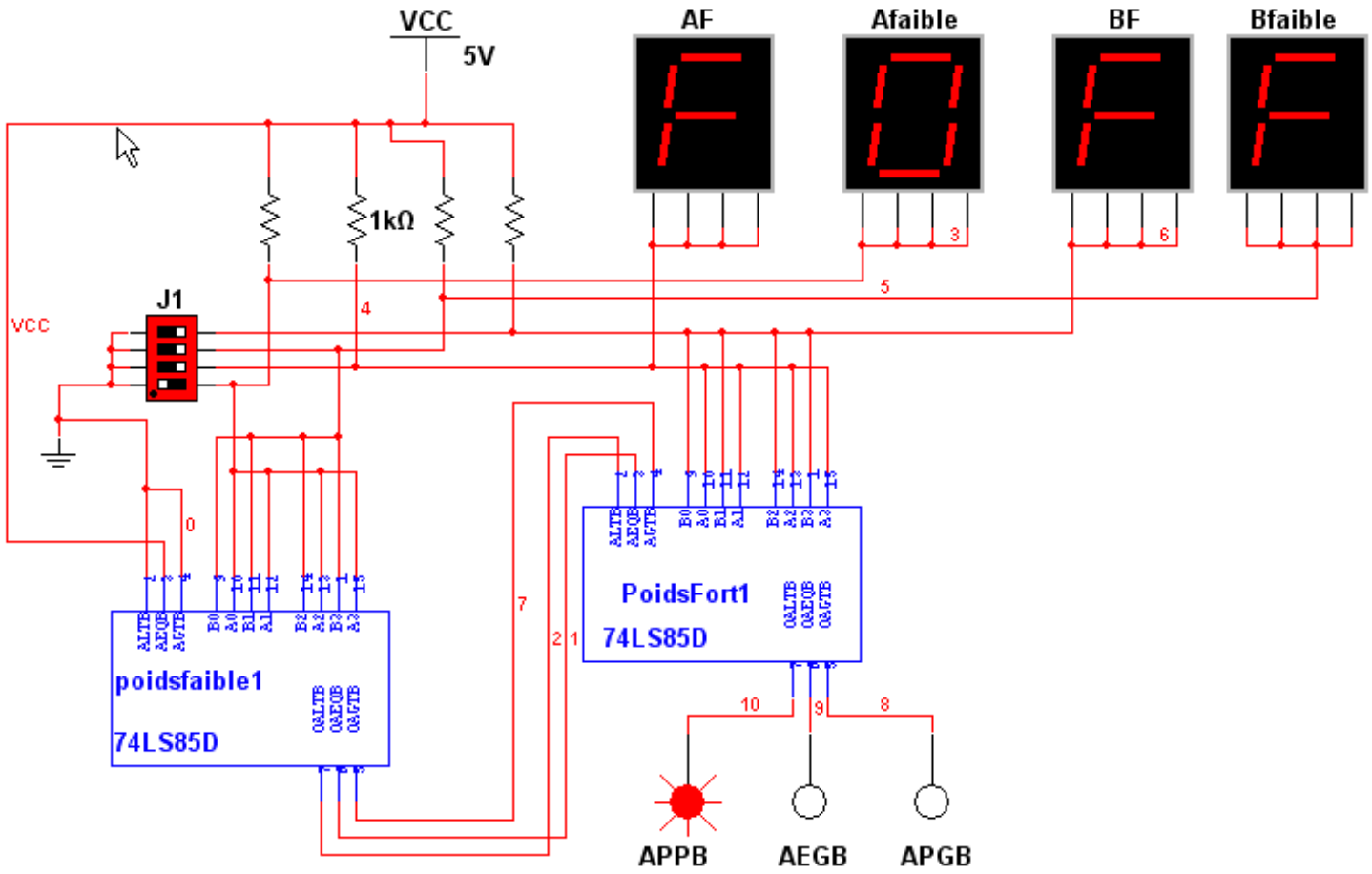
Simulations

A- Comparateur de 2 quartets (> = <) Ex: 8>1



Comparateur d'un quartet.

B-Comparateur de 2 octets (> = <) Ex: F0<FF



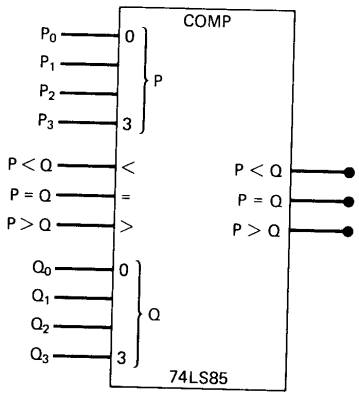
Comparateur d'octets.

Applications

Les comparateurs de grandeurs sont employés assez intensivement dans les circuits de décodage des adresses des ordinateurs; ce sont eux qui permettent de sélectionner le périphérique d'entrée/ sortie ou de localiser la zone mémoire contenant les données que l'on veut retrouver. Ces éléments comparent le code d'adresse envoyé par le processeur central (UCT) à un code d'adresse matériel; si les deux coïncident, la sortie $O_{A=B}$ du comparateur active le dispositif ayant l'adresse correspondante. Les comparateurs de grandeurs sont également très utiles dans les applications de régulation où un nombre binaire figurant le comportement d'une variable physique régulée (comme la vitesse, la position) est comparé à une valeur de consigne. Les sorties du comparateur servent de déclencheur à l'envoi de signaux pour la conduite des mécanismes qui ramènent la variable physique vers son point de consigne.

Symbole

La figure 9-41 nous montre le symbole IEEE/ANSI du comparateur 74LS85. Notez l'emploi des lettres P et Q pour représenter les variables d'entrée. Il s'agit de la désignation sanctionnée par la norme IEEE/ANSI.



II. CIRCUITS LOGIQUES MSI (intégration à moyenne échelle)

INTRODUCTION

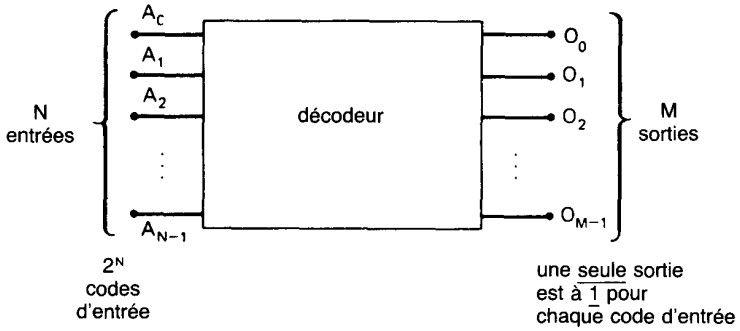
Dans les systèmes numériques, on retrouve toujours des données et des informations codées sous forme binaire qui sont sans cesse soumises à des opérations, comme:
1) le décodage et le codage - transposition des données d'un code à un autre,
2) le multiplexage - choix d'un groupe de données parmi plusieurs,
3) le démultiplexage - aiguillage des données vers une destination parmi plusieurs, et
4) l'acheminement par bus - transmission de données entre plusieurs dispositifs par l'intermédiaire d'un bus commun. Toutes ces opérations, ainsi que d'autres, sont désormais faciles à matérialiser grâce aux nombreux circuits intégrés de la classe MSI (intégration à moyenne échelle).
Dans cette partie de chapitre, nous étudierons plusieurs types de dispositifs MSI parmi les plus courants.

Pour chaque type de dispositif,

- nous commencerons par une courte discussion de son principe de fonctionnement,
- nous présenterons des CI spécifiques,
- nous montrerons quel rôle ces dispositifs jouent dans diverses applications quand on les utilise soit seuls, soit combinés à d'autres circuits intégrés.

1. LES DÉCODEURS

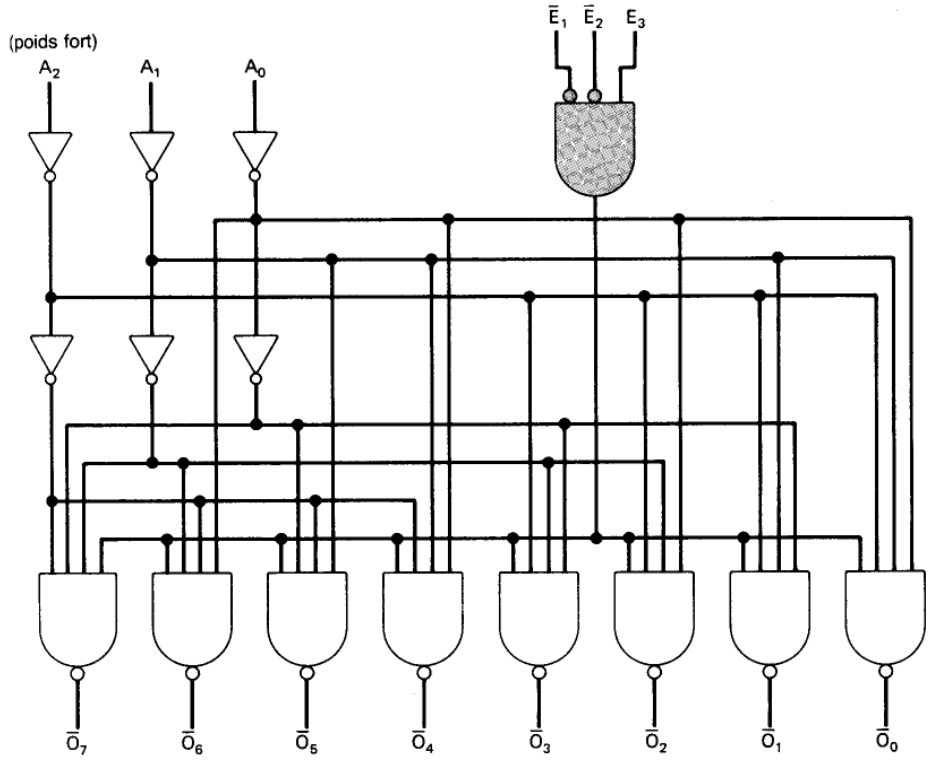
Le décodeur est un circuit logique qui établit la correspondance entre un code d'entrée binaire de N-bits et M lignes de sortie; **pour chacune des combinaisons possibles des entrées, une seule ligne de sortie est validée**.
N est un entier quelconque et M est un entier inférieur ou égal à 2^N



Cette figure nous fait voir le schéma général d'un décodeur ayant N entrées et M sorties. Étant donné que chacune des N entrées peut être soit 0 soit 1, il y a 2^N combinaisons ou codes d'entrée possibles. Pour chacune des combinaisons d'entrée possibles, une seule des M sorties passera au niveau HAUT; toutes les autres sorties demeureront au niveau BAS. De nombreux décodeurs sont conçus pour avoir des sorties vraies au niveau BAS, c'est-à-dire que seule la sortie choisie est au niveau BAS tandis que les autres demeurent au niveau HAUT. Quand cette convention s'applique, il y a toujours sur les lignes de sortie du schéma des petits ronds.

Certains décodeurs n'utilisent pas toute la gamme des 2^N codes d'entrée possibles, mais seulement un sous-ensemble de celle-ci. Par exemple, un décodeur DCB-décimal a comme entrée un code binaire de 4 bits et dix lignes de sortie, une pour chacune des dix représentations du code DCB, 0000 à 1001. Souvent, les décodeurs de ce genre sont conçus de façon à ce que les codes inutilisés n'activent aucune des sorties lorsqu'ils se retrouvent appliqués sur l'entrée.

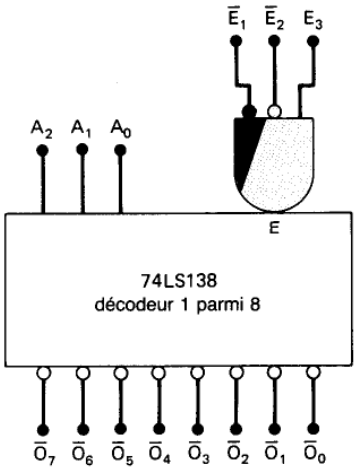
Exemple 1: le 74138



a)

$\overline{E}_1$	$\overline{E}_2$	E_3	sorties
0	0	1	réagit au code d'entrée $A_2A_1A_0$
1	X	X	invalidées – toutes au niveau HAUT
X	1	X	invalidées – toutes au niveau HAUT
X	X	0	invalidées – toutes au niveau HAUT

b)



c)

FIGURE a) Schéma logique d'un décodeur 74LS138; b) sa table de vérité; c) son symbole logique. (Gracieuseté de Fairchild, filiale Schlumberger).

Entrées VALIDATION

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

Par exemple, reportez-vous au codeur illustré à la figure précédente et imaginez qu'il y ait une ligne commune VALIDATION raccordée à une quatrième entrée de chaque porte.

> **Quand cette ligne est gardée au niveau HAUT,** le décodeur fonctionne normalement et le code d'entrée A, B, C détermine quelle sortie passe au niveau HAUT.
>
> **Quand VALIDATION est gardée au niveau BAS**, toutes les sorties sont forcées dans l'état BAS quels que soient les niveaux appliqués aux entrées A, B, C.
>
> **Donc ce décodeur est VALIDÉ seulement si le signal VALIDATION est au niveau HAUT**.

La figure suivante a) nous montre le schéma logique du décodeur 74LS138 tel que le donne le manuel des fiches techniques TTL de Fairchild.

Fonctionnement de ce décodeur.

On remarquera tout d'abord, que **ses sorties** sont celles de portes NON-ET, par conséquent elles sont vraies au niveau BAS. C'est d'ailleurs ce qu'indique la façon dont sont désignées les sorties, $\overline{O_7}, \overline{O_6}, \overline{O_5}, \ldots$; la barre de complémentation indique des **sorties vraies au niveau BAS.**

Le **code d'entrée est appliqué aux bornes A_2, A_1, A_0**, où A_2 est le bit de poids fort.

Comme il a trois entrées et huit sorties, c'est un décodeur entrée trois voies, sortie huit voies ou, ce qui est équivalent, **un décodeur 1 parmi 8.**

Les entrées $\overline{E_1}, \overline{E_2}$ et E_3 sont des **entrées de validation distinctes** qui sont combinées dans une porte ET.

Pour valider les portes NON-ET de sortie afin qu'elles indiquent le code d'entrée **A_2, A_1, A_0** correspondant, il faut que la sortie de cette porte ET soit au niveau HAUT.

C'est ce qui se produit seulement quand $\overline{E_1} = \overline{E_2} = 0$ et $E_3 = 1$

Autrement dit, ces trois entrées doivent être dans leur état vrai pour rendre actives les sorties du décodeur. Si au moins une de ces entrées validation est dans son état non vrai, la sortie de ET est au niveau BAS, ce qui impose aux sorties de toutes les NON-ET leur état inactif HAUT, et cela quel que soit le code d'entrée.

D'ailleurs, la figure b) résume tout ceci dans une table de vérité.

Rappelons que “X” veut toujours dire “condition indifférente”.

Le symbole logique du 74LS138 vous est montré à la figure c).

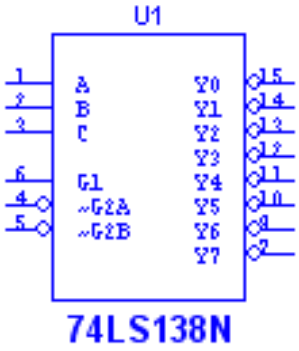
Notez comment on représente les sorties vraies au niveau BAS et les entrées de validation. Même si la porte ET de validation est dessinée à l'extérieur, en réalité, elle fait partie des circuits internes de la puce.

Voilà ce que nous donne Multisim

74xx138 (3-to-8 Dec)

This device decodes one of eight lines dependent on the conditions at the three binary select inputs and the three enable inputs
3-to-8 decoder/demultiplexer truth table:

			SELECT										
$\overline{GL}$	G1	$\overline{G2}$	C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X	X	1	X	X	X	1	1	1	1	1	1	1	1
X	0	X	X	X	X	1	1	1	1	1	1	1	1
0	1	0	0	0	0	0	1	1	1	1	1	1	1
0	1	0	0	0	1	1	0	1	1	1	1	1	1
0	1	0	0	1	0	1	1	0	1	1	1	1	1
0	1	0	0	1	1	1	1	1	0	1	1	1	1
0	1	0	1	0	0	1	1	1	1	0	1	1	1
0	1	0	1	0	1	1	1	1	1	1	0	1	1
0	1	0	1	1	0	1	1	1	1	1	1	0	1
0	1	0	1	1	1	1	1	1	1	1	1	1	0
1	1	0	X	X	X	Output corresponding to stored address 0; all others 1							



Exercices:

1) Indiquez les états des sorties d'un 74LS 138 pour chacune des conditions d'entrée que voici:

a) $\overline{E_1} = 0$, $\overline{E_2} = 1$, $E_3 = 1$, $A_2 = 1$, $A_1 = 1$ et $A_0 = 0$

b) $\overline{E_1} = 0$, $\overline{E_2} = 0$, $E_3 = 1$, $A_2 = 0$, $A_1 = 1$ et $A_0 = 1$

Solution

a) Lorsque $\overline{E_2} = 1$, le décodeur est désactivé et toutes ses sorties se trouvent dans leur état HAUT (inactif). C'est ce qu'on peut déduire de la table de vérité ou bien en relevant les niveaux logiques tout au long du circuit logique.

b) Toutes les sorties de validation sont activées, de sorte que la section de décodage est validée. Elle décode donc le code d'entrée $011_2 = 3_{10}$ afin d'activer la sortie $\overline{O_3}$. Donc, $\overline{O_3}$ se trouve au niveau BAS et toutes les autres sorties sont au niveau HAUT.

2) La figure suivante montre comment il est possible d'agencer quatre 74LS 138 et un INVERSEUR pour obtenir **un décodeur 1 parmi 32**. Ces décodeurs sont notés Z0-Z3 afin de s'y référer facilement et les huit sorties de chacun de ceux-ci sont combinées pour donner un total de **32** sorties. Un code d'entrée de 5 bits $\mathbf{A_4\ A_3\ A_2\ A_1\ A_0}$ n'ouvre qu'une seule des 32 sorties pour chacune des 32 représentations d'entrée possibles.

a) Dites quelle sortie est active si $\mathbf{A_4\ A_3\ A_2\ A_1\ A_0} = 01101$.

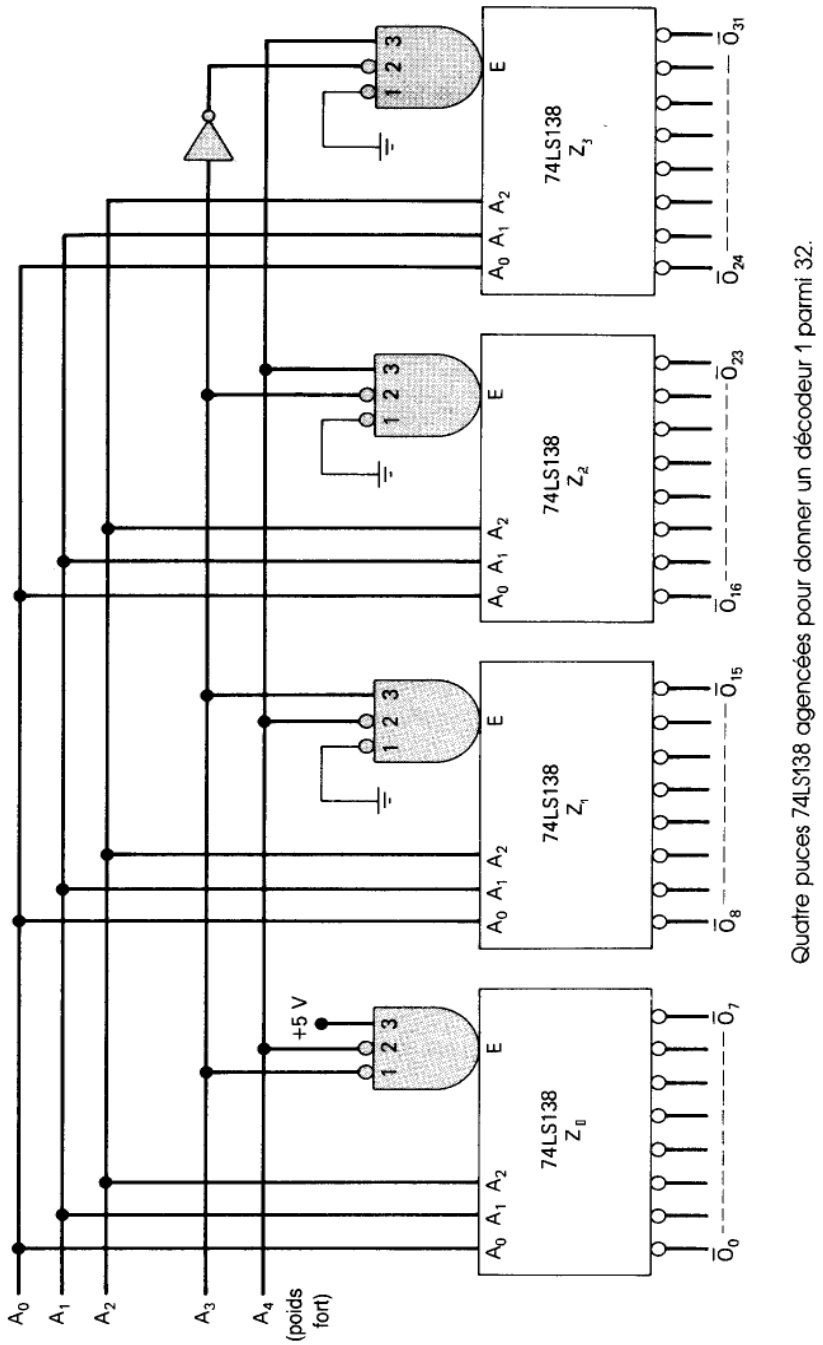
b) Indiquez la plage des codes d'entrée qui rend actives toutes les sorties de la puce Z3.

Solution

a) Le code d'entrée à 5 bits est composé de 2 parties distinctes. Les bits A_4 et A_3 déterminent lequel des quatre boîtiers Z1-Z4 est validé, alors que la section $A_2\ A_1\ A_0$ fixe la sortie de la puce validée qui passe à son niveau vrai. Si $A_4A_3 = 01$, seul Z2 a ses entrées de validation au niveau vrai. Donc Z2 réagit au code $A_2\ A_1\ A_0 = 101$ et fait passer sa sortie $\overline{O_5}$ à sa valeur vraie, sortie rebaptisée $\overline{O_{13}}$.

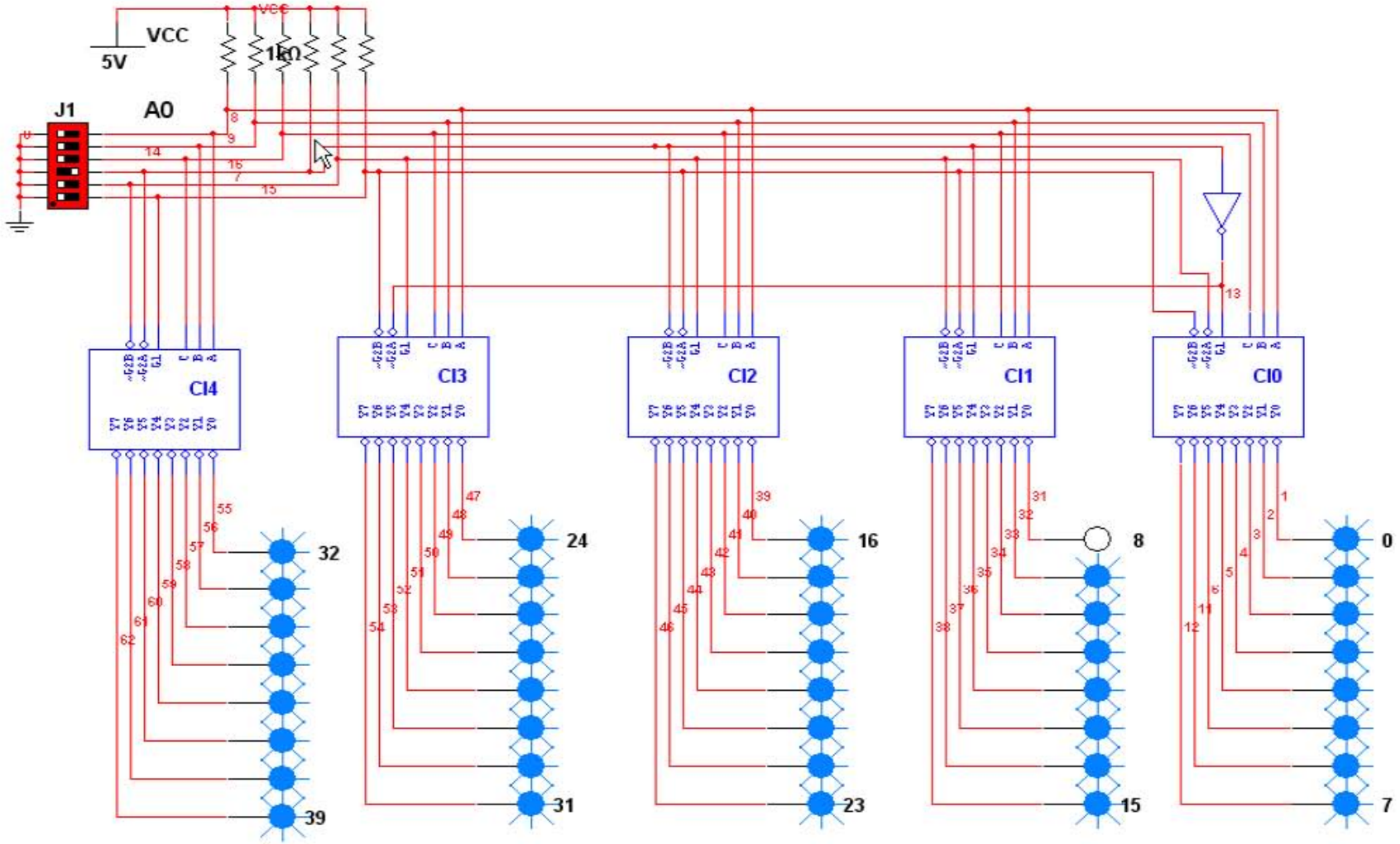
Donc le code d'entrée 01101, l'équivalent binaire du chiffre décimal 13, fait passer la sortie $\overline{O_{13}}$ au niveau BAS tandis que toutes les autres sorties restent au niveau HAUT.

b) Pour valider Z4, il faut que A_4 et A_3 soient tous les deux à 1. Donc tous les codes d'entrée compris entre $11000(24_{10})$ et $11111(31_{10})$ rendent active une sortie de Z4. Cette plage de codes correspond aux sorties $\overline{O_{24}}$ à $\overline{O_{31}}$.



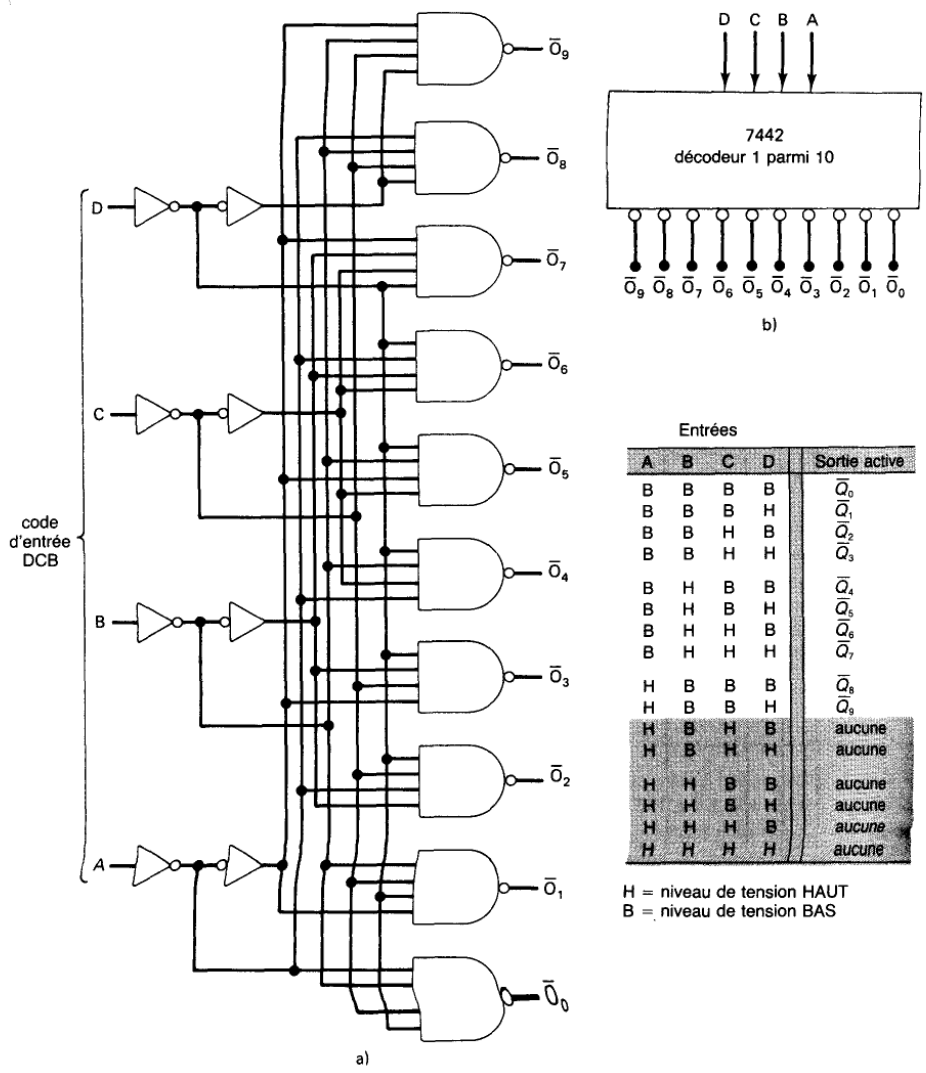
Quatre puces 74LS138 agencées pour donner un décodeur 1 parmi 32.

Simulation.



Décodeur 1 parmi 40.

Exemple2:Décodeurs DCB-décimal



Entrées

A	B	C	D	Sortie active
B	B	B	B	$\overline{Q}_0$
B	B	B	H	$\overline{Q}_1$
B	B	H	B	$\overline{Q}_2$
B	B	H	H	$\overline{Q}_3$
B	H	B	B	$\overline{Q}_4$
B	H	B	H	$\overline{Q}_5$
B	H	H	B	$\overline{Q}_6$
B	H	H	H	$\overline{Q}_7$
H	B	B	B	$\overline{Q}_8$
H	B	B	H	$\overline{Q}_9$
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

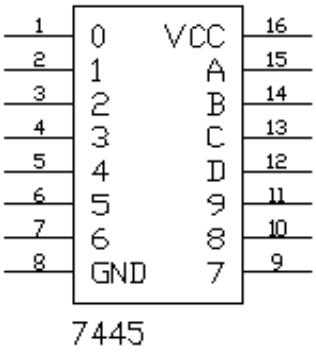
H = niveau de tension HAUT
B = niveau de tension BAS

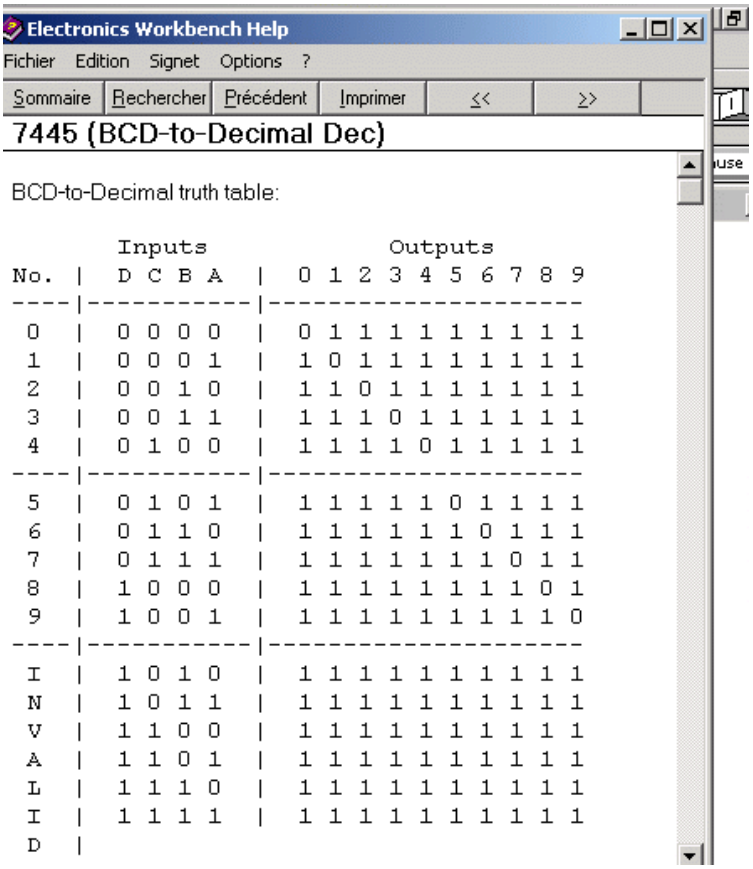
FIGURE a) Schéma logique du décodeur DCB-décimal 7442; b) son symbole logique; c) sa table de vérité. (Gracieuseté de Fairchild, une filiale Schlumberger).

La figure précédente a) 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 $\overline{O_5}$ ne devient au niveau BAS que lorsque les valeurs sur les entrées sont DCBA = 0101. Dans le cas des combinaisons DCB non valides, aucune des entrées n'est validée. Ce décodeur reçoit fréquemment le nom de **décodeur entrée 4 voies, sortie 10 voies** ou décodeur **1 parmi 10**. Le symbole logique et la table de vérité du 7442 sont reproduits également sur cette figure. Remarquez l'absence dans ce décodeur d'une entrée de validation.

Exemple3: Décodeur/pilote DCB-décimal

Le TTL 7445 est un décodeur/pilote DCB-décimal. Le terme « pilote » est ajouté à la désignation de cet élément pour indiquer des sorties en **collecteur ouvert** qui permettent à une sortie TTL de fonctionner avec un courant plus intense et dans des limites de tension supérieures aux valeurs normales. Les sorties du 7445 peuvent absorber jusqu'à 80 mA dans l'état BAS et être portées jusqu'à 30 V dans l'état HAUT. Ceci lui permet d'attaquer directement des charges comme des LEDs, des relais ou des moteurs à courant continu.





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. Quand les entrées d'un décodeur sont alimentées par un compteur qui dénombre sans arrêt, les sorties du décodeur sont activées les unes à la suite des autres et peuvent alors servir à une tâche de synchronisation ou de séquencement 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. C'est une application que nous étudierons plus en détail plus tard, quand nous aborderons le sujet des mémoires.

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.

2. PILOTES / DÉCODEURS 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 (figure a). Chaque segment est constitué d'un matériau qui émet de la lumière quand il est traversé par un courant. Les matériaux les plus utilisés sont les diodes électroluminescentes (LED) et les filaments incandescents. La figure b) nous montre comment sont disposés les segments afin de pouvoir afficher les différents chiffres. Par exemple, pour afficher le chiffre « 6 » il faut que les segments c, d, e, f et g soient allumés et que les segments a et b soient éteints.

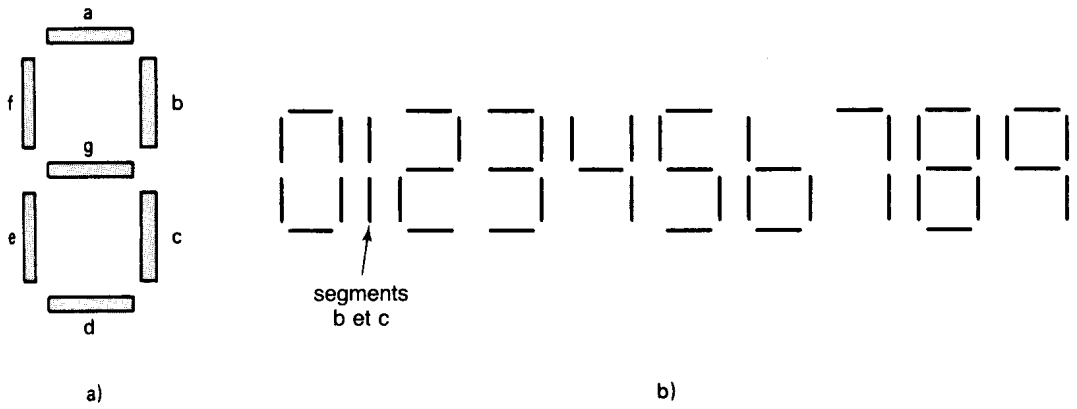


FIGURE a) Disposition des 7 segments; b) les segments qui doivent être allumés pour former chacun des chiffres.

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. La logique de ce décodeur est plus compliquée que ce que l'on a vu précédemment, parce que chaque sortie peut être mise 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'est-à-dire quand les codes d'entrée sont 0000, 0010, 0110 ou 1000.

Le pilote/décodeur DCB-7 segments, illustré à la figure suivante a) (TTL 7446 ou 7447), pilote un afficheur DEL à 7 segments. Chaque segment est constitué d'une ou deux diodes électroluminescentes. Les anodes de ces diodes sont toutes réunies à Vcc (+ 5 V). 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, soit des transistors pilotes à collecteur ouvert pouvant absorber un courant passablement intense.

C'est la raison pour laquelle les afficheurs DEL peuvent nécessiter entre 10 mA et 40 mA par segment, selon leur type et leur taille.

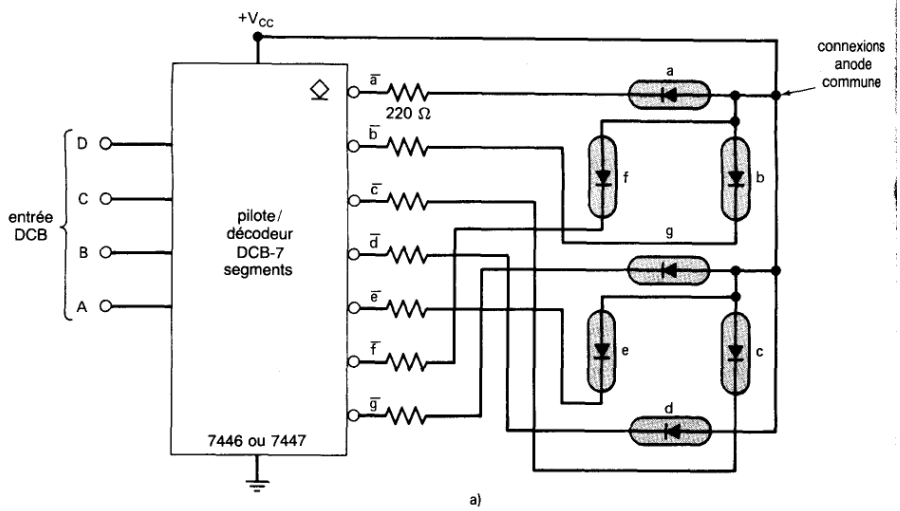
Pour illustrer le fonctionnement de ce circuit, soit l'entrée DCB D = 0, C = 1, B = 0, A _ 1, correspondant à la représentation DCB de 5. En réponse à cette entrée, les sorties $\overline{a}$, $\overline{f}$, $\overline{g}$, $\overline{c}$ et $\overline{d}$ du pilote/décodeur sont amenées au niveau BAS (raccordées à la masse), ce qui a pour effet de faire passer un courant à travers les

segments DEL a, f, g, c et d et d'afficher le nombre 5. Les sorties b et e demeurent au niveau HAUT (ouvert) et les segments correspondants $\overline{b}$ et $\overline{e}$ restent éteints.

Les pilotes/ décodeurs 7446 et 7447 ont été conçus de façon à allumer certains segments même si le code d'entrée est supérieur à 1001 (9). D'ailleurs à la figure b), on peut voir les segments qui s'allument en réponse à tous les codes d'entrée de 0000 à 1111 (15). Notez que le code d'entrée 1111 éteint tous les segments.

L'afficheur DEL de cette figure est dit à anode commune parce que toutes les anodes des segments sont réunies à Vcc. Il existe un autre type d'afficheur DEL à 7 segments dit à cathode commune dans lequel toutes les cathodes des segments sont réunies à la masse. Ce dernier type d'afficheur est attaqué par un pilote/décodeur DCB-7 segments dont les sorties sont vraies au niveau HAUT; ainsi quand des sorties sont rendues actives, les anodes des segments correspondants passent à la tension HAUTE. Le pilote/ décodeur 7448 fonctionne de cette manière.

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.



Photocopies du data handbook.

BCD TO SEVEN-SEGMENT DECODER/DRIVERS — 54/74 SERIES "46 & 47"

54/7446A
54/7447A

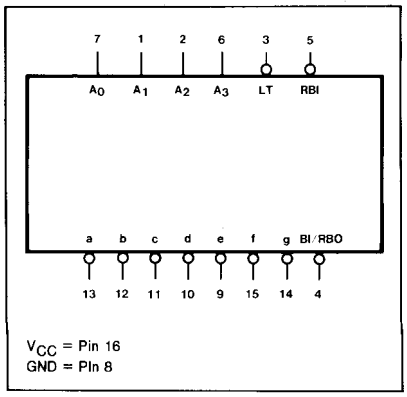
DESCRIPTION

The "46" and "47" are BCD to 7-Segment Decoder/Drivers with open collector outputs. They accept 4-bit BCD data and provide seven active LOW decoded outputs to drive 7-segment incandescent displays directly. Their 40mA output sink current capability makes them useful for driving multiplexed common anode LED displays. Both devices feature overriding lamp test (all segments "on") and ripple blanking for leading and trailing zeroes.

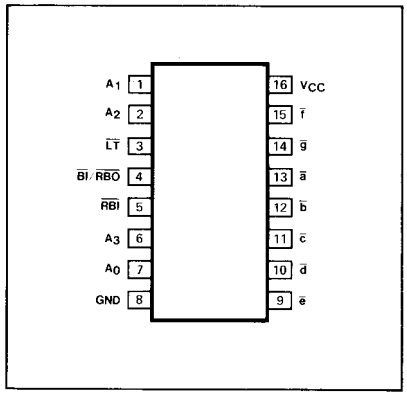
FEATURES

- **40mA output current sink per segment**
- **30V segment voltage on 54/7446A**
- **15V segment voltage on 54/7447A**
- **Overriding lamp test**
- **Ripple Blanking for leading and trailing zeroes**

LOGIC SYMBOL



PIN CONFIGURATION



ORDERING CODE (See Section 9 for further Package and Ordering Information)

PACKAGES	COMMERCIAL RANGES V_{CC}=5V ± 5%; T_A=0°C to +70°C	MILITARY RANGES V_{CC}=5V ± 10%; T_A=−55°C to +125°C
Plastic DIP	N7446AN N7447AN	
Ceramic DIP	N7446AF N7447AF	S5446AF S5447AF
Flatpak		

INPUT AND OUTPUT LOADING AND FAN-OUT TABLE[a]

PINS	DESCRIPTION		54/74	54S/74S	54LS/74LS
A_0-A_3	BCD Address inputs	I_{IH} (μA) I_{IL} (mA)	40 −1.6		
$\overline{LT}$	Lamp Test (active LOW) input	I_{IH} (μA) I_{IL} (mA)	40 −1.6		
$\overline{RBI}$	Ripple Blanking (active LOW) input	I_{IH} (μA) I_{IL} (mA)	40 −1.6		
$\overline{BI}/\overline{RBO}$	Blanking (active LOW) input	I_{IH} (μA) I_{IL} (mA)	N.A. −4.0		
$\overline{BI}/\overline{RBO}$	Ripple Blanking (active LOW) output	I_{OH} (μA) I_{OL} (mA)	−200 8.0		
$\overline{a}$-$\overline{g}$	Seven-segment (active LOW) outputs	I_{OH} (μA) I_{OL} (mA)	+250 40		

NOTE
a. The slashed numbers indicate different parametric values for Military/Commercial temperature ranges respectively.

BCD TO SEVEN-SEGMENT DECODER/DRIVERS — 54/74 SERIES "46 & 47"

FUNCTIONAL DESCRIPTION

The "46A" and "47A" 7-segment decoders accept a 4-bit BCD code input and produce the appropriate outputs for selection of segments in a 7-segment matrix display used for representing the decimal numbers "0-9." The seven outputs ($\overline{a}$, $\overline{b}$, $\overline{c}$, $\overline{d}$, $\overline{e}$, $\overline{f}$, $\overline{g}$) of the decoder select the corresponding segments in the matrix shown in Figure A. The numeric designations chosen to represent the decimal numbers are shown in Figure B, together with the resulting displays for input code configurations in excess of binary "9."

The "46A" and "47A" have provisions for automatic blanking of the leading and/or trailing edge zeroes in a multidigit decimal number, resulting in an easily readable decimal display conforming to normal writing practice. In an 8-digit mixed integer fraction decimal representation, using the automatic blanking capability, 0070.0500 would be displayed as 70.05. Leading edge zero suppression is obtained by connecting the Ripple Blanking Output ($\overline{BI}/\overline{RBO}$) of a decoder to the Ripple Blanking Input ($\overline{RBI}$) of the next lower stage device. The most significant decoder stage should have the $\overline{RBI}$ input grounded; and, since suppression of the least significant integer zero in a number is not usually desired, the $\overline{RBI}$ input of this decoder stage should be left open. A similar procedure for the fractional part of a display will provide automatic suppression of trailing edge zeroes.

The decoder has an active LOW input Lamp Test which overrides all other input combinations and enables a check to be made on possible display malfunctions. The $\overline{BI}/\overline{RBO}$ terminal of the decoder can be OR-tied with a modulating signal via an isolating buffer to achieve pulse duration intensity modulation. A suitable signal can be generated for this purpose by forming a variable frequency multivibrator with a cross coupled pair of open collector gates.

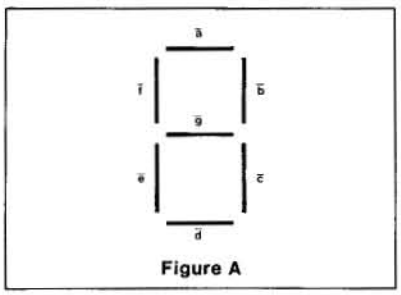


Figure A

NOTE

b. $\overline{BI}/\overline{RBO}$ is wire-AND logic serving as blanking input (BI) and/or ripple-blanking output ($\overline{RBO}$). The blanking out (BI) must be open or held at a HIGH level when output functions 0 through 15 are desired, and ripple-blanking input ($\overline{RBI}$) must be open or at a HIGH level if blanking of a decimal 0 is not desired.

TRUTH TABLE

DECIMAL OR FUNCTION	INPUTS							OUTPUTS						
	$\overline{LT}$	$\overline{RBI}$	A_3	A_2	A_1	A_0	$\overline{BI}/\overline{RBO}$(b)	$\overline{a}$	$\overline{b}$	$\overline{c}$	$\overline{d}$	$\overline{e}$	$\overline{f}$	$\overline{g}$
0	H	H	L	L	L	L	H	L	L	L	L	L	L	H
1	H	X	L	L	L	H	H	H	L	L	H	H	H	H
2	H	X	L	L	H	L	H	L	L	H	L	L	H	L
3	H	X	L	L	H	H	H	L	L	L	L	H	H	L
4	H	X	L	H	L	L	H	H	L	L	H	H	L	L
5	H	X	L	H	L	H	H	L	H	L	L	H	L	L
6	H	X	L	H	H	L	H	H	H	L	L	L	L	L
7	H	X	L	H	H	H	H	L	L	L	H	H	H	H
8	H	X	H	L	L	L	H	L	L	L	L	L	L	L
9	H	X	H	L	L	H	H	L	L	L	H	H	L	L
10	H	X	H	L	H	L	H	H	H	H	L	L	H	L
11	H	X	H	L	H	H	H	H	H	L	L	H	H	L
12	H	X	H	H	L	L	H	H	L	H	H	H	L	L
13	H	X	H	H	L	H	H	L	H	H	L	H	L	L
14	H	X	H	H	H	L	H	H	H	H	L	L	L	L
15	H	X	H	H	H	H	H	H	H	H	H	H	H	H
$\overline{BI}$(b)	X	X	X	X	X	X	L	H	H	H	H	H	H	H
$\overline{RBI}$(b)	H	L	L	L	L	L	L	H	H	H	H	H	H	H
$\overline{LT}$	L	X	X	X	X	X	H	L	L	L	L	L	L	L

H = HIGH voltage level
L = LOW voltage level
X = Don't care

NUMERICAL DESIGNATIONS—RESULTANT DISPLAYS

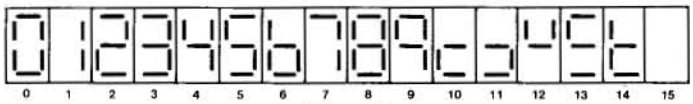
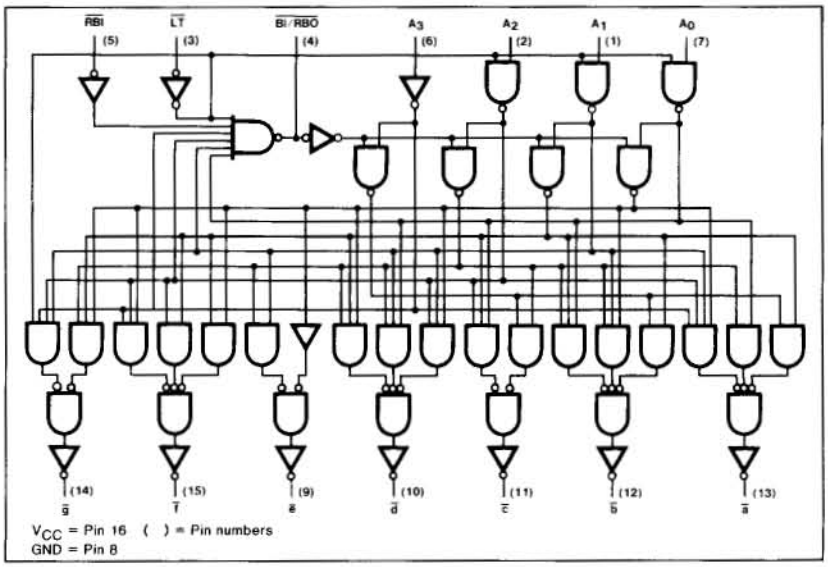


Figure B

LOGIC DIAGRAM



V_{CC} = Pin 16 ( ) = Pin numbers
GND = Pin 8

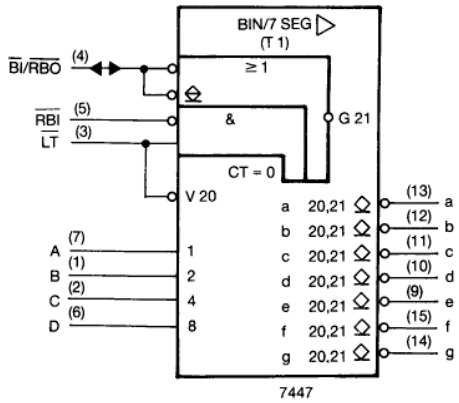
Définition de certaines entrées/sorties

LT: Lamp test: borne de test (allumage de tous les segments)
BI/RBO: Blanking Input / Ripple Blanking Ouput
RBI : Ripple Blanking Input
Ces 2 derniers permettent d'éliminer la visualisation des zéros non significatifs. On raccorde BI/RBO à RBI de l'état suivant (de rang plus élevé)
Tous les segments seront allumés si LT =0 et BI/RBO = 1
Si BI/RBO = 0 alors tous les segments seront éteints

Remarques complémentaires concernant la broche BI/RBO

BI/RBO est un ET logique câblé et est utilisé en entrée pour la commande d'extinction (BI) ou en sortie correspondante (RBO).
1) L'entrée BI doit être ouverte ou au niveau Haut pour les sorties de 0 à 15. RBI doit être ouvert ou au niveau Haut pour l'effacement des zéros décimaux.
2) Lorsque l'entrée BI est maintenue au niveau bas (L), tous les segments sont éteints, quels que soient les niveaux des autres entrées.
3) Lorsque RBI et les entrées A, B, C et D sont au niveau bas (L) et LT au niveau Haut, tous les segments sont éteints et RBO passe au niveau bas (L).
4) Lorsque BI/RBO est ouvert ou au niveau Haut et que LT passe au niveau bas (L), tous les segments sont allumés.

Symboles IEE/ANSI



Remarques
1) on numérote les entrées et les sorties à l'intérieur du rectangle.
2) un losange souligné signale une sortie à collecteur ouvert.
3) le triangle signifie que l'on a affaire à un tampon-pilote doté de caractéristiques de tension et de courant dépassant les valeurs normales

3. CODEURS

Si le décodage est un processus à partir duquel une représentation de N bits produit un signal HAUT (ou BAS) sur une et seulement une des lignes de sortie d'un décodeur, alors le processus inverse est le codage et il utilise un circuit logique appelé “codeur”. **Un codeur 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**. La figure suivante nous montre le schéma général d'un codeur ayant M entrées et N sorties. Dans ce cas-ci, les entrées sont vraies au niveau HAUT, ce qui veut dire que normalement elles sont au niveau BAS.

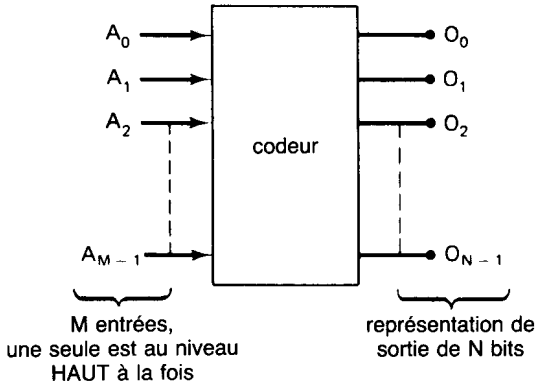
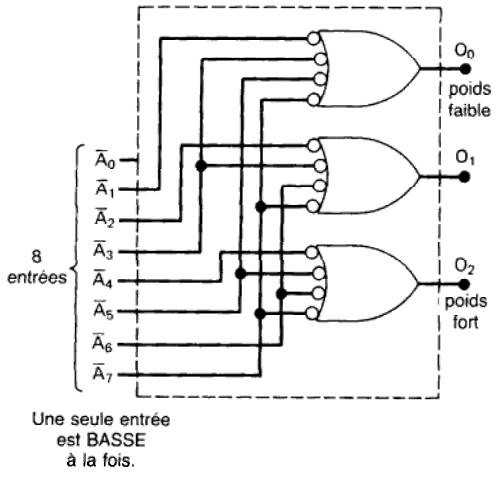


Schéma général d'un codeur.

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. À l'inverse, un codeur octal-binaire a 8 voies d'entrée et produit une représentation de sortie binaire de 3 bits.

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

En analysant successivement les réponses des circuits logiques, on en conclue qu'un niveau BAS sur une seule entrée donne lieu en sortie à un code binaire qui correspond à cette entrée.

Par exemple, un BAS sur $\overline{A_3}$ (pendant que toutes les autres entrées sont à 1) produit $O_2 = 0,\ O_1 = 1$ et $O_0 = 1$ ce qui est le code binaire de 3. Observez 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.

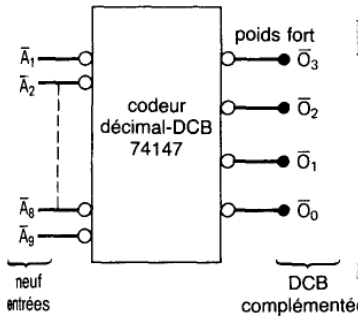
Codeurs de priorité

Comme on l'a dit précédemment, 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 et cette version modifiée 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**. Dans un tel codeur, quand les deux entrées $\overline{A_5}$ et $\overline{A_3}$ sont actives en même temps, la réponse donnée en sortie est 101. De même, ce codeur produit en sortie le code 110 si les entrées $\overline{A_0}$, $\overline{A_2}$ et $\overline{A_6}$ sont rendues actives en même temps. Les 74148, 74LS148 et 74HC148 sont des codeurs de priorité octal-binaire.

Codeur de priorité décimal-DCB 74147

La figure suivante nous montre le symbole logique du codeur de priorité décimal-DCB 74147 (74LS 147, 74HC 147). Celui-ci a 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.

Codeur de priorité décimal-DCB 74147.



$\overline{A_1}$	$\overline{A_2}$	$\overline{A_3}$	$\overline{A_4}$	$\overline{A_5}$	$\overline{A_6}$	$\overline{A_7}$	$\overline{A_8}$	$\overline{A_9}$	$\overline{O_3}$	$\overline{O_2}$	$\overline{O_1}$	$\overline{O_0}$
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	X	0	1	0	1	1	1
X	X	X	X	X	X	0	1	1	1	0	0	0
X	X	X	X	X	0	1	1	1	1	0	0	1
X	X	X	X	0	1	1	1	1	1	0	1	0
X	X	X	0	1	1	1	1	1	1	0	1	1
X	X	0	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

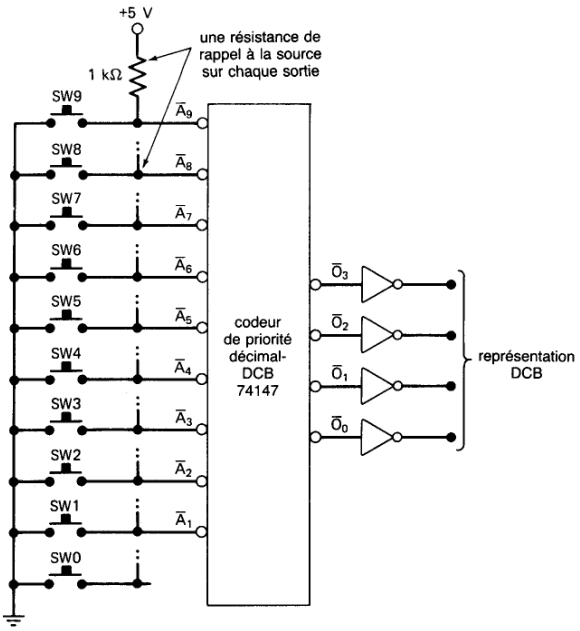
X = 0 ou 1

Examinons la table de vérité afin de déduire le fonctionnement de ce CI. La première ligne de cette table

montre que toutes les entrées sont inactives au niveau HAUT. Pour cette condition, les sorties sont 1111, soit l'inverse de 0000, le code DCB de 0. La deuxième ligne de cette table montre qu'un niveau BAS sur $\overline{A_9}$, et cela quelles que soient les valeurs sur les autres entrées, donne lieu au code de sortie 0110, soit l'inverse de 1001, le code DCB de 9. La troisième ligne démontre qu'un niveau BAS sur $\overline{A_8}$, à la condition que le niveau sur $\overline{A_9}$ soit HAUT, produit le code de sortie 0111, soit l'inverse de 1000, le code DCB pour 8. Le même raisonnement pour toutes les lignes de cette table nous apprend 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.

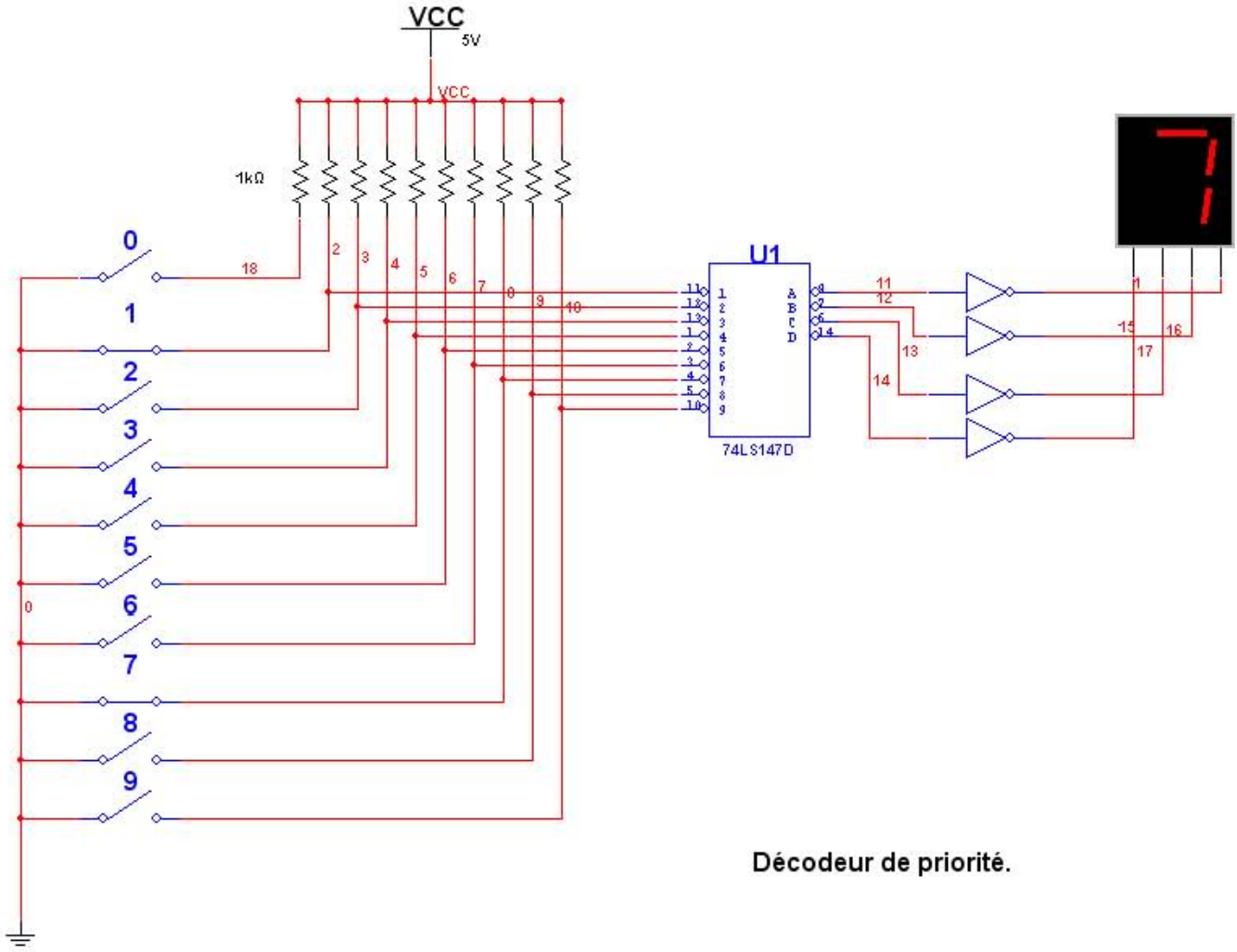
Exemple d'utilisation d'un 74147: codeur d'interrupteurs. On place 10 interrupteurs aux 10 entrées. Les 10 interrupteurs peuvent être ceux d'un clavier de calculatrice représentant les chiffres 0 à 9. Les interrupteurs sont du type ouvert au repos, de sorte que les entrées du codeur sont généralement toutes à 1; quand c'est le cas la sortie DCB est 000 (remarquez les inverseurs). Quand une touche numérique est enfoncée, le circuit donne en sortie le code DCB correspondant à ce chiffre. Comme le 74147 est un codeur de priorité, l'enfoncement simultané de plusieurs touches place en sortie le code DCB de la touche numériquement la plus élevée.

Codeur d'interrupteurs décimal-DCB.



Ce codeur d'interrupteurs peut servir toutes les fois que l'on doit introduire manuellement des donnés DCB dans un système numérique. Un exemple parfait de cela est la calculatrice électronique dans laquelle un opérateur introduit le nombre décimal en appuyant successivement sur des interrupteurs du clavier. Dans une calculatrice élémentaire, le code DCB de chaque chiffre décimal est introduit dans un registre mémoire de 4 bits. Ce qui veut dire que lorsqu'on appuie sur la première touche, le code DCB correspondant à ce chiffre est envoyé à un registre de 4 bits, quand on appuie sur le second interrupteur, le code DCB de ce nouveau chiffre est envoyé à un autre registre à bascules de 4 bits, et ainsi de suite.

Simulation. (Ex: si 1 et 7 sont actionnés en même temps,7 est prioritaire)



4. TRANSCODEURS

Dans ce cas, on code les sorties à partir d'un codage différent des entrées. Le transcodeur va donc élaborer les grandeurs de sortie codées à partir des entrées codées. On rencontre 2 types de transcodeurs:

- conversion de code.
- affichage par segments ou matrice de points.

Le second a déjà été étudié dans les décodeurs, nous nous attacherons donc plus précisément aux transcodeurs (de conversion de code)

Principe:

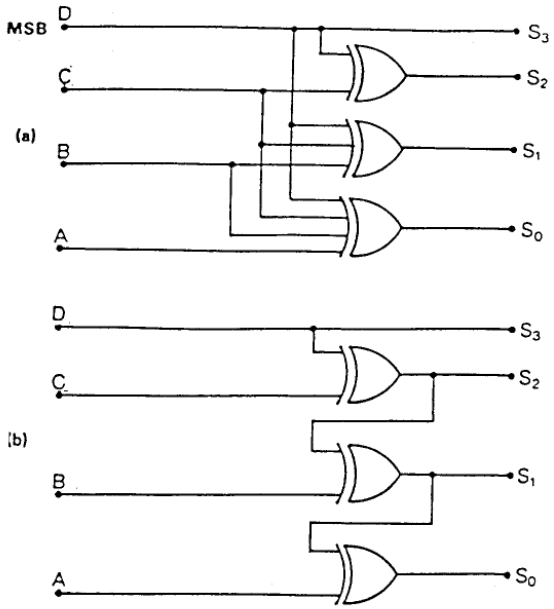
Pour passer d'un code binaire à l'autre, il suffit de considérer les entrées et les sorties dans une double table de vérité qui permet d'établir les équations logiques de transcodage et les méthodes traditionnelles.

Exemple:

décimal	GRAY				BINAIRE			
N	D	C	B	A	S_3	S_2	S_1	S_0
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	1	0	0	1	0
3	0	0	1	0	0	0	1	1
4	0	1	1	0	0	1	0	0
5	0	1	1	1	0	1	0	1
6	0	1	0	1	0	1	1	0
7	0	1	0	0	0	1	1	1
8	1	1	0	0	1	0	0	0
9	1	1	0	1	1	0	0	1
10	1	1	1	1	1	0	1	0
11	1	1	1	0	1	0	1	1
12	1	0	1	0	1	1	0	0
13	1	0	1	1	1	1	0	1
14	1	0	0	1	1	1	1	0
15	1	0	0	0	1	1	1	1

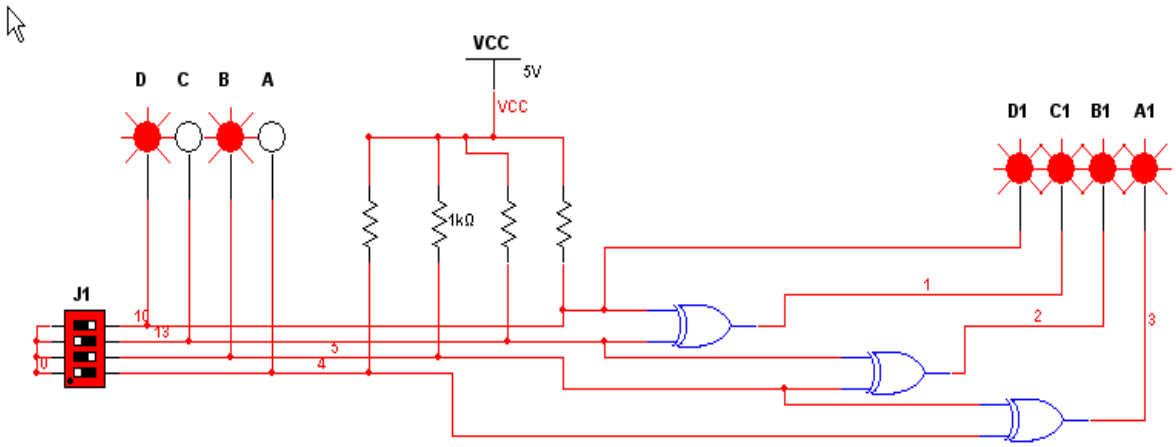
On obtient les équations suivantes:

$$S_3 = D$$
$$S_2 = C \oplus D$$
$$S_1 = B \oplus C \oplus D = B \oplus S_2$$
$$S_0 = A \oplus B \oplus C \oplus D = A \oplus S_1$$



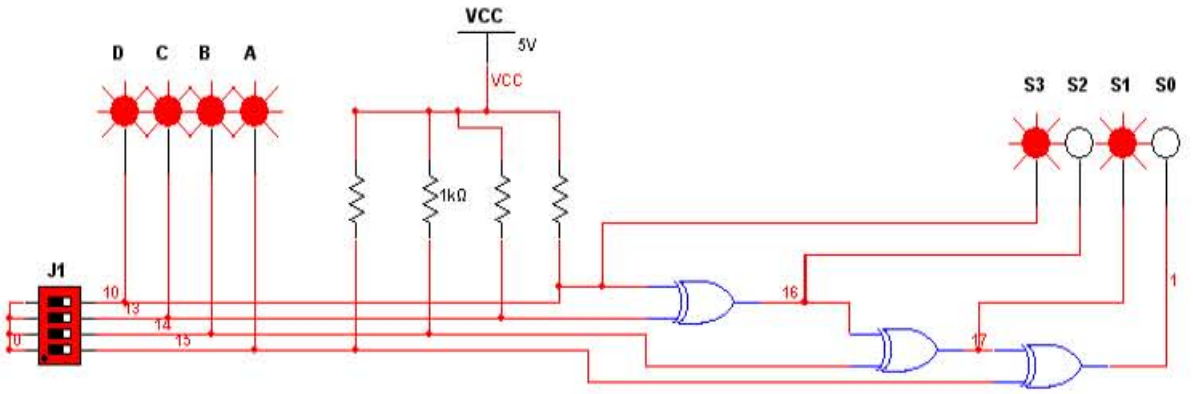
On obtient 2 possibilités de câblages selon que l'on adopte la première ou la deuxième forme d'équation. La seconde permet une économie de matériel, mais sera moins rapide (le 'repiquage' de grandeur logique accumule les retards)

Simulations:



Binaire	Gray
0000	0000
0001	0001
0010	0011
0011	0010
0100	0110
0101	0111
0110	0101
0111	0100
1000	1100
1001	1101
1010	1111
1011	1110
1100	1010
1101	1011
1110	1001
1111	1000

TRANSCODEUR BINAIRE-GRAY

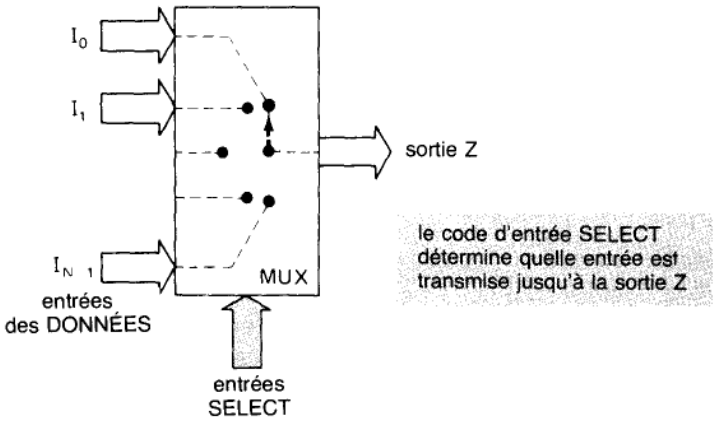


Gray	Binaire
0000	0000
0001	0001
0011	0010
0010	0011
0110	0100
0111	0101
0101	0110
0100	0111
1100	1000
1101	1001
1111	1010
1110	1011
1010	1100
1011	1101
1001	1110
1000	1111

TRANSCODEUR GRAY-BINAIRE

5. MULTIPLEXEURS

<u>Un multiplexeur ou sélecteur de données</u> est un circuit logique ayant plusieurs entrées de données, mais seulement une sortie qui communique les données. L'aiguillage de l'entrée de données qui nous intéresse sur la sortie est commandé par les entrées SELECT (appelées parfois entrées d'adresse). La figure suivante illustre le symbole d'un multiplexeur général (MUX). Dans ce symbole, les entrées et les sorties sont représentées par des flèches épaisses afin d'indiquer qu'il peut y avoir plus d'un conducteur.

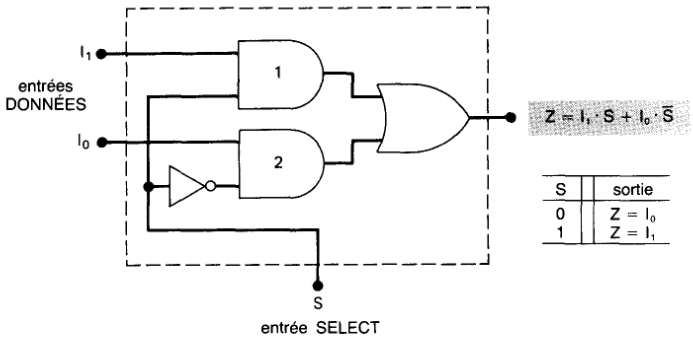


Symbole d'un multiplexeur numérique (MUX).

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. Il peut y avoir sur la sortie Z les données introduites sur I_0 quand on applique un certain code d'entrée SELECT; ou bien Z peut avoir les données de I_1 en réponse à un autre code d'entrée SELECT, et ainsi de suite. Autrement dit, un multiplexeur choisit une source de données d'entrée parmi N et transmet celles-ci à la seule voie de sortie existante. C'est ce qu'on appelle le multiplexage.

Exemple1 Multiplexeur élémentaire à deux entrées

La figure suivante schématise les circuits logiques d'un multiplexeur à deux entrées ,I_0 et I_1, et à une entrée de sélection S. Le niveau logique appliqué à S choisit la porte ET qui est validée, de façon à ce que son entrée de données traverse la porte OU jusqu'à la sortie Z



Multiplexeur à deux entrées.

D'après l'algèbre booléenne, l'expression de la sortie est:

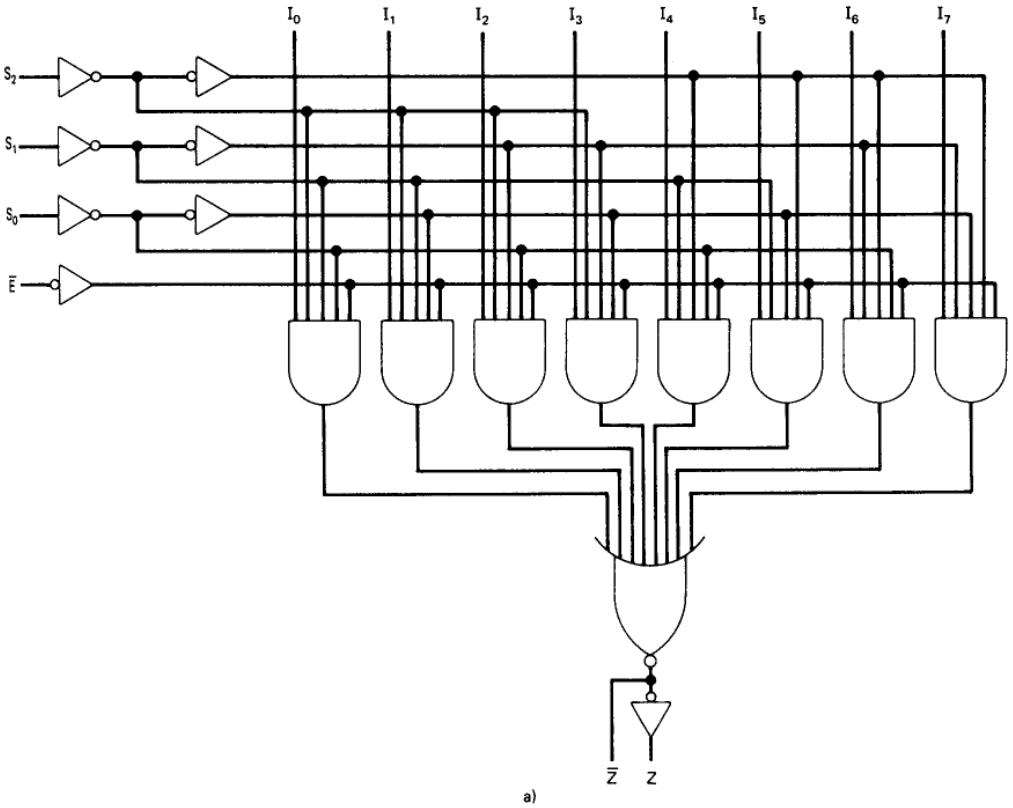
$$Z = I_0\overline{S} + I_1 S$$

$$\text{Si } S = 0 \longrightarrow Z = I_0$$
$$\text{Si } S = 1 \longrightarrow Z = I_1$$

Exemple2: Multiplexeur à huit entrées

Voyez à la figure suivante a) le schéma logique du multiplexeur à huit entrées 74151 (74LS 151, 74HC 151).

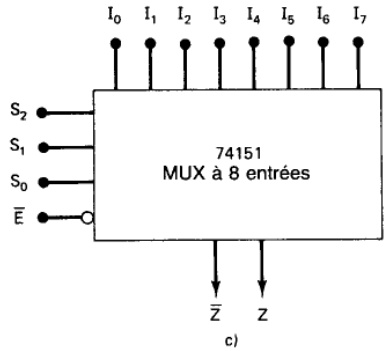
a) Schéma logique du multiplexeur 74151; b) sa table de vérité; c) son symbole logique. (Gracieuseté de Fairchild, filiale Schlumberger).



a)

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

b)

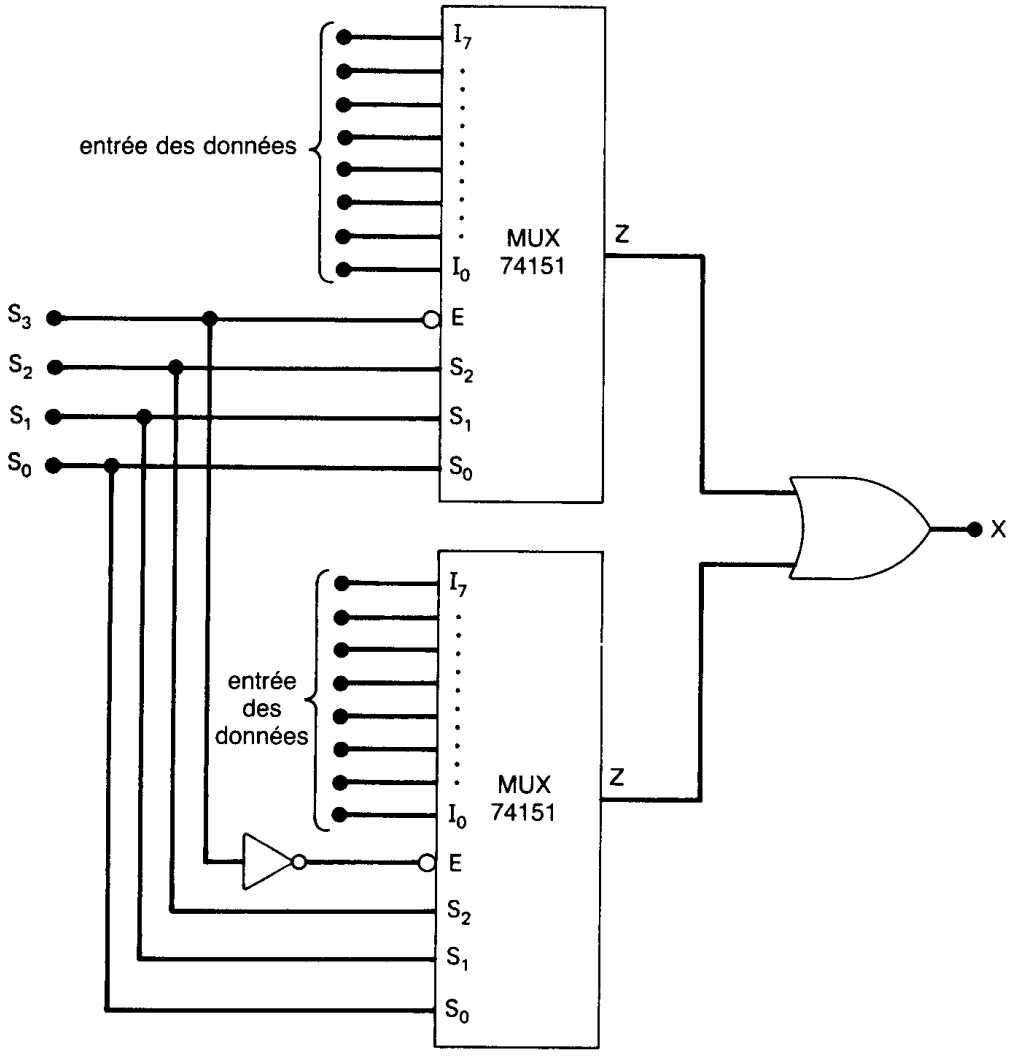


c)

Ce multiplexeur dispose d'une entrée validation, $\overline{E}$, et il fournit la sortie normale et la sortie complémentée. Quand $\overline{E} = 0$, les entrées de sélection $S_2S_1S_0$ choisissent une entrée de données (I_0 à I_7) dont les valeurs se retrouvent sur la sortie Z. Quand $\overline{E} = 1$, le multiplexeur est invalidé de sorte que Z = 0 quel que soit le code d'entrée de sélection. Ce fonctionnement est résumé dans le tableau de la figure b); le symbole logique du 74151 est illustré à la figure c).

Le circuit de la figure suivante a recours à deux 74151, à un INVERSEUR et à une porte OU.

Explication:



Exemple : deux 74151 combinés pour réaliser un multiplexeur à 16 entrées.

Au total, ce circuit a 16 entrées de données, 8 pour chaque multiplexeur. Les deux sorties du multiplexeur sont réunies dans une porte OU pour produire une seule sortie X. Ce circuit fonctionne comme un multiplexeur à 16 entrées. Les 4 entrées de sélection $S_3S_2S_1S_0$ choisissent l'une des 16 entrées pour la faire passer jusqu'à X.
L'entrée S_3 choisit le multiplexeur qui est validé. Quand $S_3 = 0$, le multiplexeur du haut est validé, et les

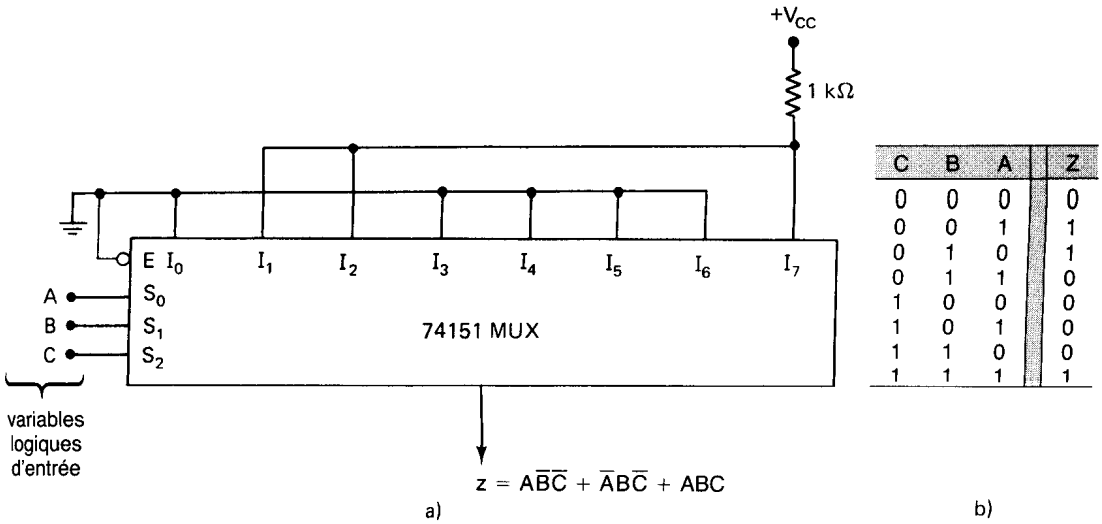
entrées $S_2S_1S_0$ définissent laquelle des entrées de ce multiplexeur passera jusqu'à la sortie et franchira la porte OU pour atteindre X. Quand $S_3 = 1$, c'est le multiplexeur du bas qui est validé et les entrées $S_2S_1S_0$ choisissent l'entrée de données de ce dernier qui atteint la sortie X.

Déterminez les conditions d'entrée qui placent sur chacune des sorties Z le niveau logique de son entrée I_0 correspondante. Même question pour I_1 .

Réponse:

Application du 74151

La figure précédente nous montre comment on utilise un multiplexeur à 8 entrées pour matérialiser un circuit logique qui fonctionne conformément aux indications de la table de vérité. Les variables d'entrée A, B, C sont



C	B	A	Z
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

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

raccordées respectivement à S_0, S_1, S_2, et ce sont les niveaux appliqués à ces entrées qui déterminent l'entrée de données qui est transférée à la sortie Z. Conformément à cette table de vérité, Z doit être à 0 quand CBA = 000. Ainsi l'entrée du multiplexeur I_0 doit être raccordée à 0. De la même façon, Z est supposée être à 0 quand CBA = 011, 100, 101 et 110, de sorte que les entrées I_3, I_4, I_5 et I_6 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 multiplexeur I_l, I_2 et I_7 à un niveau HAUT.

Il est facile de voir que toute table de vérité à 3 variables peut être matérialisée au moyen d'un multiplexeur à 8 entrées. Cette méthode de matérialisation est souvent beaucoup plus efficace que l'utilisation de portes logiques distinctes. Par exemple, si nous écrivons la somme de produits correspondant à la table de vérité de la figure précédente, nous trouvons:

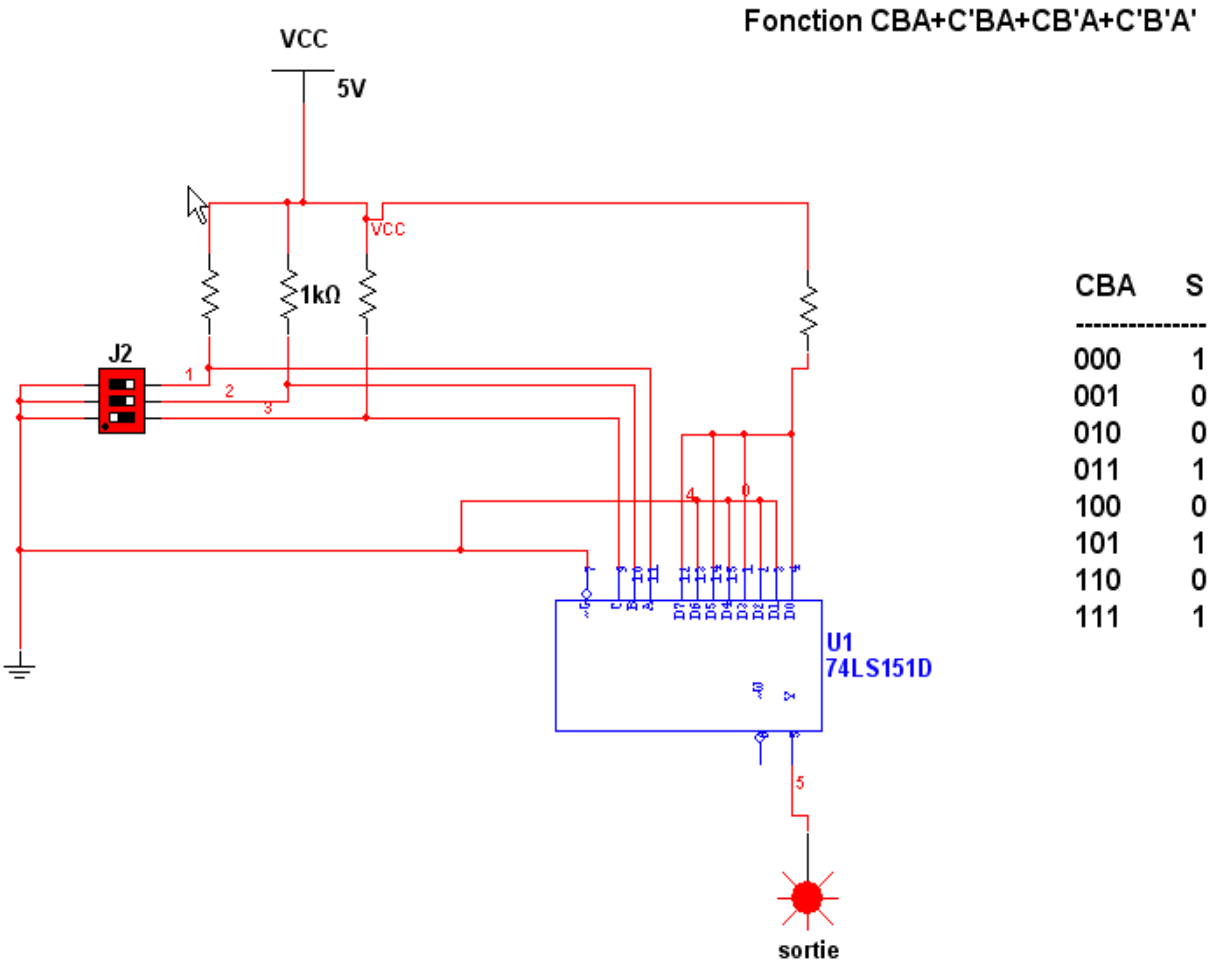
$$Z = A\overline{B}\overline{C} + \overline{A}B\overline{C} + ABC$$

Il est impossible de simplifier cette expression au moyen de l'algèbre ou d'un diagramme K, ce qui signifie que pour la matérialiser il faut 3 INVERSEURS et 4 portes NON-ET, soit au total 2 circuits intégrés.

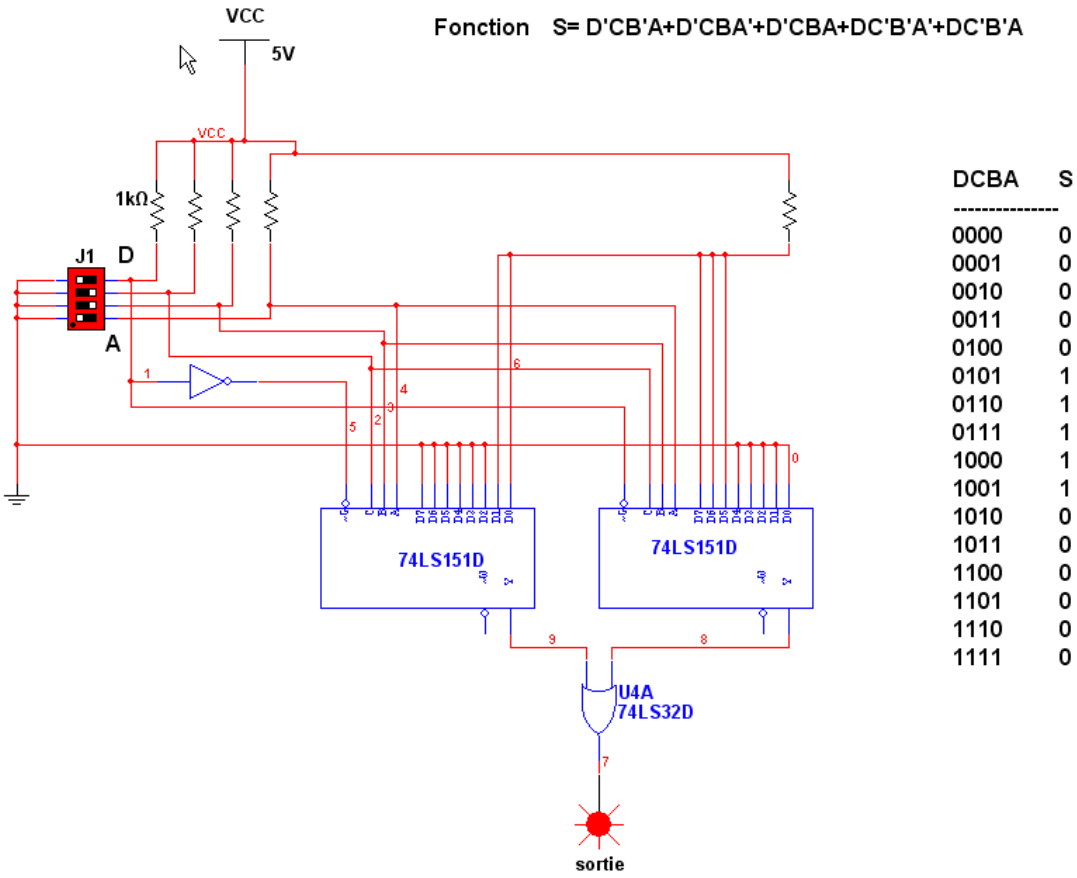
Il existe une méthode encore plus efficace, soit réaliser de telles fonctions logiques en se servant de multiplexeurs. Cette méthode permet à un concepteur logicien de prendre un multiplexeur à trois entrées de sélection (par ex. le 74151) et de construire une fonction logique à quatre variables.

Simulations

a) Fonction à 3 variables



b) fonction à 4 variables



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.

6. DÉMULTIPLEXEURS

On vient de voir 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**. La figure suivante nous montre le schéma général d'un démultiplexeur (DEMUX). Les flèches plus larges utilisées pour représenter les entrées et les sorties signifient plusieurs conducteurs. Le code d'entrée de sélection détermine à quelle sortie l'entrée DONNÉES est transmise. Autrement dit, le démultiplexeur reçoit des données d'une entrée et choisit de les diriger vers une des N voies de sortie possibles; il fonctionne comme un commutateur.

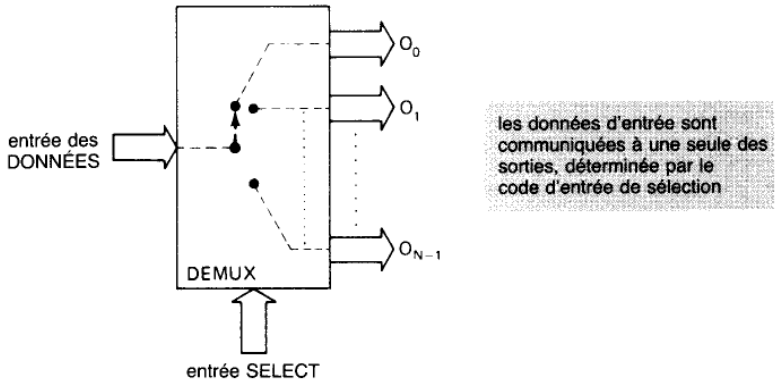
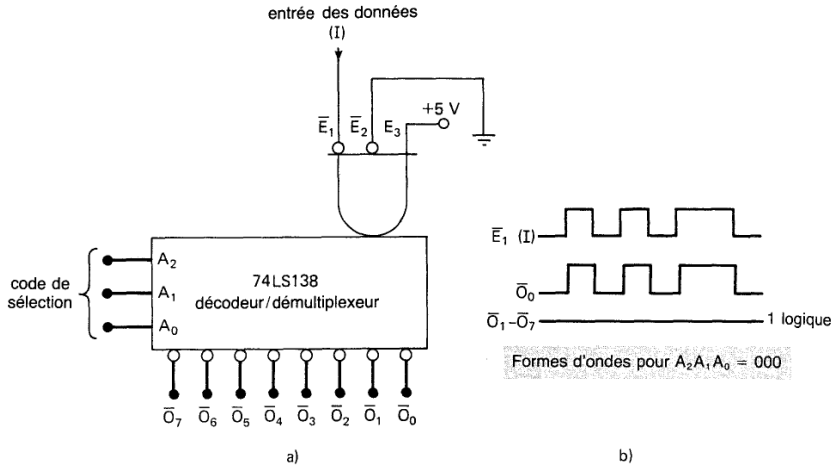


FIGURE Schéma général d'un démultiplexeur.

Exemple1: Démultiplexeur entrée une voie, sortie huit voies

Nous avons déjà étudié le 74LS138 dans son rôle de décodeur un parmi huit. La figure suivante nous montre maintenant comment l'utiliser à titre de démultiplexeur. L'entrée de validation E, 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 A_2 A_1 A_0 accueillent le code de sélection. Pour illustrer son fonctionnement, supposons que les entrées de sélection sont 000. On sait que ce code ouvre seulement la sortie appelée $\overline{O_0}$ et garde toutes les autres sorties au niveau HAUT. $\overline{O_0}$ passe à 0 seulement si $\overline{E_1}$ passe à 0, et cette sortie passe à 1 si $\overline{E_1}$ passe à 1. Autrement dit, $\overline{O_0}$ suit les niveaux appliqués à la borne $\overline{E_1}$ (c'est-à-dire l'entrée de données I). Pendant ce temps toutes les autres sorties demeurent à 1. De la même façon, un code de sélection différent appliqué aux entrées A_2 A_1 A_0 ouvre la sortie correspondante qui alors suit les valeurs appliquées sur l'entrée de données I.

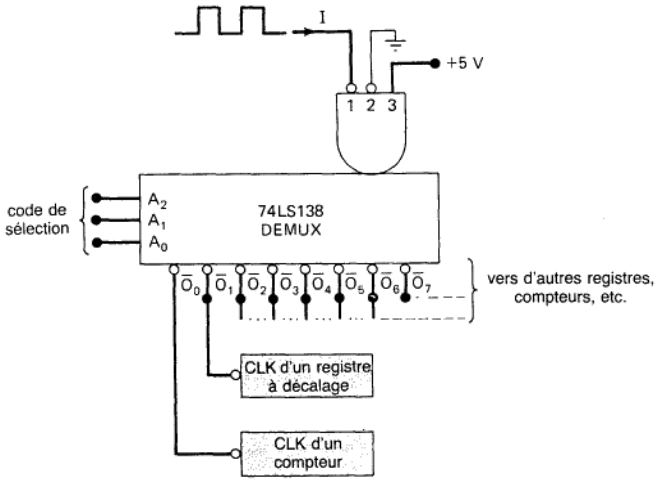


a) Le décodeur 74LS138 peut agir comme multiplexeur si on détermine que $\overline{E}_1$ sera l'entrée de données; b) formes d'ondes types quand le code de sélection est $A_2A_1A_0$ = 000, la sortie $\overline{O}_0$ est identique à l'entrée de données.

La figure b) montre les formes d'ondes types pour le cas sélectionné par $A_2\ A_1\ A_0 = 000$, soit la sortie $\overline{O_0}$. Dans cet exemple, le signal de données appliqué sur $\overline{E_1}$ est acheminé sur $\overline{O_0}$ et toutes les autres sorties restent dans leur état inactif, soit l'état HAUT.

Démultiplexeur d'horloge

Le principe du démultiplexage trouve de nombreuses applications un peu partout. La figure suivante nous fait voir le démultiplexeur 74LS138 utilisé à titre de démultiplexeur d'horloge. Commandé par les lignes SELECT, ce dispositif aiguille le signal d'horloge vers la destination voulue. Par exemple, quand $S_2\ S_1\ S_0 = 000$, le signal d'horloge appliqué à I est passé à la sortie $\overline{O_0}$. $S_2\ S_1\ S_0 = 101$, l'horloge est passée à la sortie $\overline{O_5}$.



Un démultiplexeur d'horloge achemine le signal d'horloge à la sortie désignée par les bornes du code de sélection.

HEH be
Sciences
et technologies

PARTIE II

PROTOCOLES DE

TRAVAUX PRATIQUES

NOMS: DATE:
Prénoms:
GROUPE:

Techniques numériques- Travaux pratiques avancés

MANIPULATION n° 0: INTRODUCTION

Les séances de laboratoires se subdivisent en 2 parties:

- **partie de théorie de laboratoire**
- **manipulations**

Cette partie se déroule de la même manière que le cours de techniques nuùmérique - travaux pratiques, à savoir, on travaillera en parallèle sur la plaquette didactique qu'en simulation.

Vous travaillerez sur les mêmes plaques didactiques, avec les composants de vos boites du quad 1(tous les composants). Aussi nous appliquerons les mêmes directives**.**

5- Evaluation du laboratoire

La présence au laboratoire est obligatoire, les absences ne peuvent être justifiées que par un certificat médical.

séances			points	
1	th p 1-9			
2	Labo1		20	
3				
4	lars2 et 3		40	
5	labo4		20	
6	th p 10-fin			
7	labo5		20	
8	labo6		20	
8	labo7		20	
9				
10	**labo8**	labo spécial: Evaluation des manip 1-7: plaquette + multisim		50
		moyenne des labos		20
		moyenne des interros		30
	TOTAL			100

Toutes ces cotes ne pourront être représentées, c'est la conséquence d'un travail journalier.

NOMS: DATE:
Prénoms:
GROUPE:

Techniques numériques- Travaux pratiques avancés

MANIPULATION n° 1 : EXERCICES

(Temps prévu: 2 séance de 3 heures)

---

1. But de la manipulation.

Utilisation des méthodes de simplification de fonctions vue au cours de travaux pratiques de base.

2. Rappel théorique.

Tout le cours du quad 1

3. Matériel utilisé:

- 1 Plaquette didactique.
- 1 multimètre.

4. Manipulation

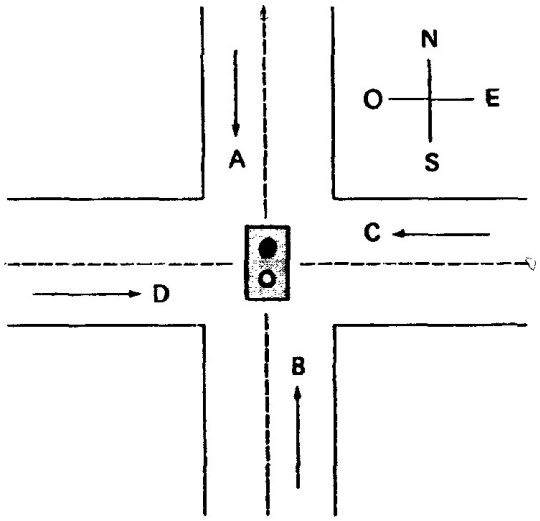
Exercice 1

La figure suivante nous montre l'intersection entre une route principale et une route secondaire. Des capteurs de voitures ont été placés le long des voies C et D (route principale) et des voies A et B (route secondaire). Les sorties de ces capteurs sont à 0 quand il n'y a pas de voiture et à 1 quand il y en a. Le feu de circulation se trouvant à cette intersection est commandé par les règles de décision suivantes:

1. Le feu E-O est vert quand il y a des voitures dans les 2 voies C et D.
2. Le feu E-O est vert quand il y a des voitures dans C ou D et quand il y en a dans A ou dans B mais pas dans les deux.
3. Le feu N-S est vert quand il y a des voitures dans les voies A et B et qu'il y en a dans C ou dans D mais pas dans les deux.
4. Le feu N-S est aussi vert quand il y a des voitures dans A ou B et qu'il n'y a pas de voiture dans C et D.
5. Le feu E-O est vert quand il n'y a pas de voiture du tout.

En utilisant les tensions de sortie des capteurs A,B,C et D comme entrées, concevez les équations d'un circuit logique qui commande le feu de circulation. Ce circuit a deux sorties, soit E-O et N-S, qui prennent la valeur haute quand le feu doit être vert.

NOMS: DATE:
Prénoms:
GROUPE:



D	C	B	A	N-S	E-O
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		

Equations N-S

NOMS: DATE:
Prénoms:
GROUPE:

Equations E-O

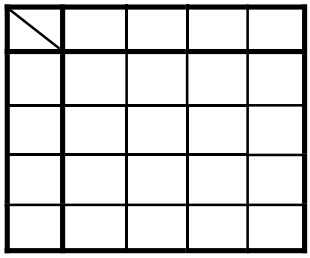


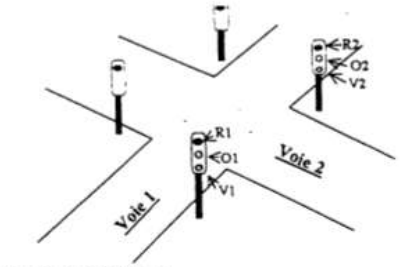
Schéma en Nand2 câbler et simuler

NOMS: DATE:
Prénoms:
GROUPE:

Exercice2

Commande de feux tricolores (source Lycée Jules Ferry Versailles)

Nous nous proposons de réaliser, à l'aide de portes NAND à 2entrées, le **décodeur** d'un montage électronique permettant le fonctionnement des feux tricolores d'un carrefour routier comportant 2 voies (voie 1 et 2. voir le dessin du carrefour ci-contre).



Dessin du carrefour

Le principe du montage électronique complet est présenté dans le schéma synoptique ci-dessous :

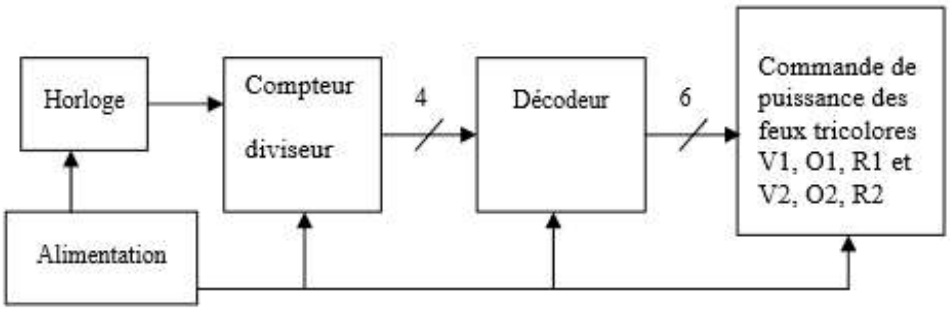


Schéma synoptique

Explication du principe:

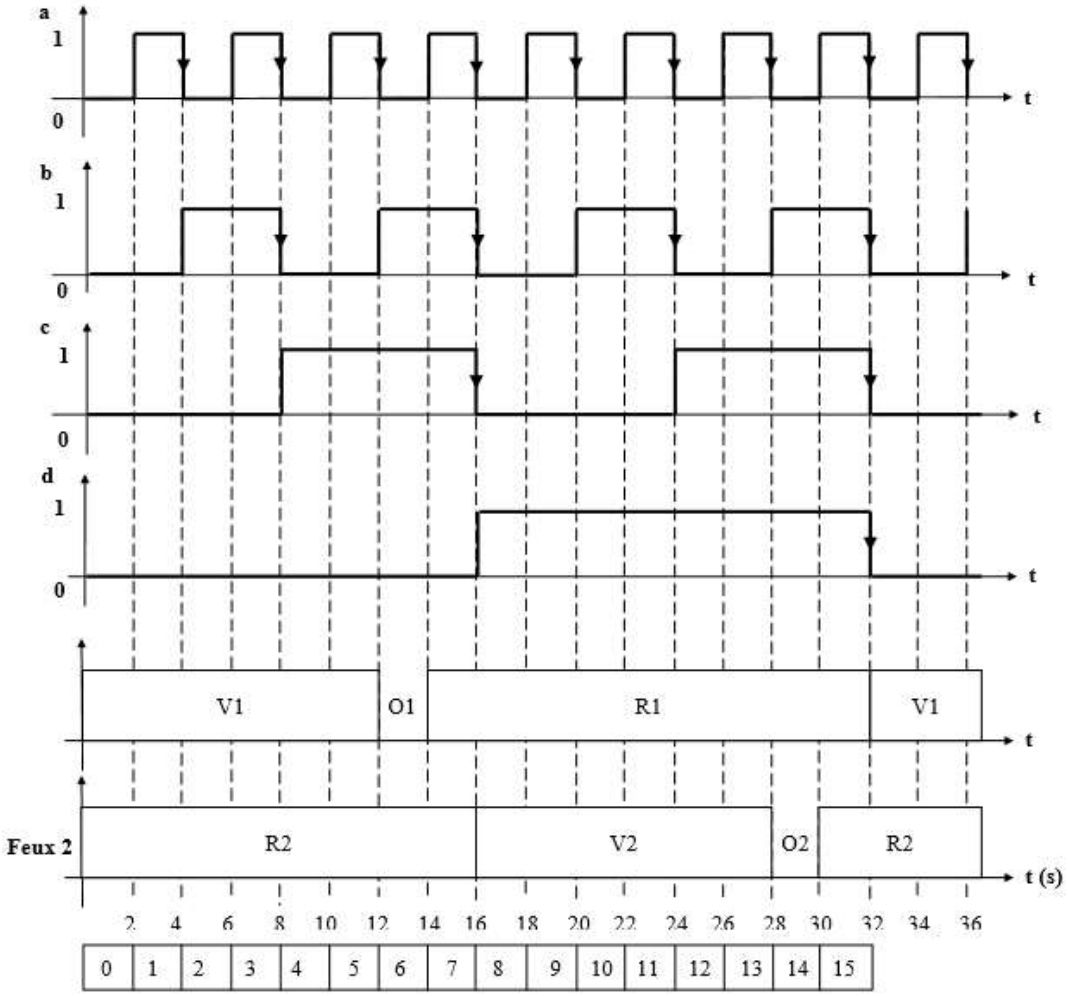
- L'horloge délivre une impulsion toutes les 2 secondes.
- Cette impulsion est appliquée à l'entrée d'horloge d'un compteur diviseur par 16.
- Les 4 sorties (a, b, c, d) du compteur délivrent des signaux logiques conformes aux chronogrammes qui suivent, et sont appliqués aux entrées du décodeur (voir chronogrammes).

Travail demandé:

1. A partir des chronogrammes, remplir les tableaux de KARNAUGH de chaque sortie du décodeur en fonction des sorties du compteur.
2 En déduire les équations de chaque sortie.
3. Transformez les équations pour n'utiliser que les portes demandées dans la présentation. (Remarque : on pourra utiliser le fait qu'entre V1, 01 et R1 il n'y a toujours qu'une seule lampe d'allumée. Idem pour V2, 02 et R2).
4. Simuler. On remplacera le compteur par 16 par des 4 interrupteurs.

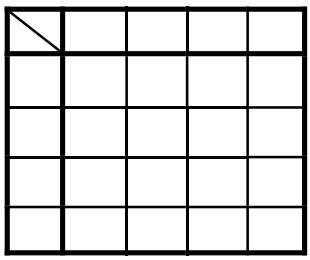
NOMS: DATE:
Prénoms:
GROUPE:

Chronogrammes:



Equations de chaque Sortie:

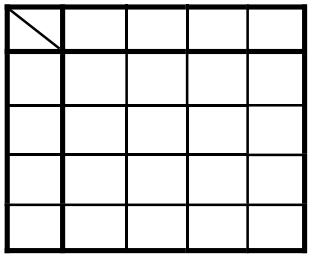
V1=



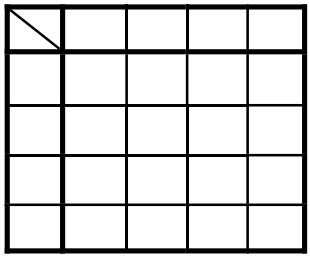
O1=

NOMS: DATE:
Prénoms:
GROUPE:

R1=

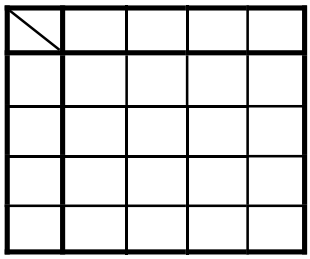


V2=



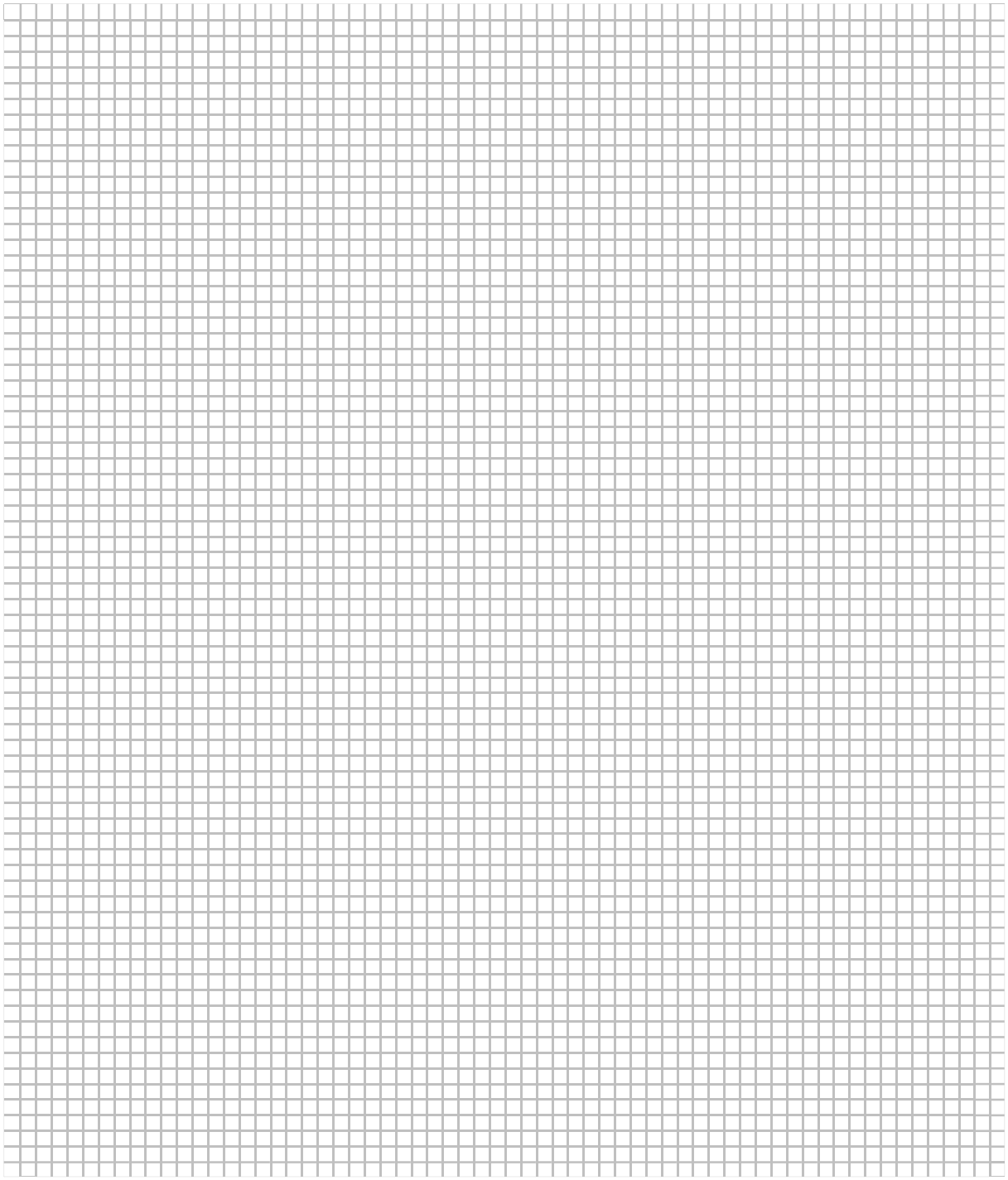
O2=

R2=



NOMS: DATE:
Prénoms:
GROUPE:

Schéma en Nand2 simuler

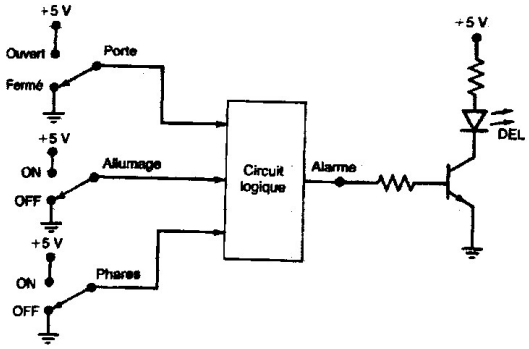


NOMS: DATE:
Prénoms:
GROUPE:

Exercice3

La figure ci-dessous illustre le schéma d'un circuit d'alarme d'une automobile qui détecte diverses situations non souhaitables . Les trois interrupteurs servent à désigner l'état de la porte du conducteur, de l'allumage et des phares, respectivement. Concevez le circuit logique ayant ces 3 interrupteurs comme entrées, qui déclenche l'alarme quand l'une des situations que voici se produit:

* les phares sont allumés et l'allumage est coupé.
* la porte est ouverte et le contact d'allumage est mis.



P	Al	Ph	S
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Equation de S

Schéma en Nand2 de S câbler ou simuler

NOMS: DATE:
Prénoms:
GROUPE:

Exercice 4

Quatre grandes cuves dans une usine de fabrication de produits chimiques contiennent différents liquides chauffés. Des capteurs de niveau servent à déceler le dépassement d'un niveau préétabli dans les cuves A et B. Des capteurs thermométriques surveillent la température des cuves C et D pour qu'elle ne descende pas sous une valeur de consigne. Supposez que les capteurs de niveau sont 0 à quand le niveau est correct et à 1 quand il est trop haut. En outre, supposez que les capteurs thermométriques sont à 0 quand la température est acceptable et à 1 quand elle est trop basse. Concevez un circuit logique qui sonne l'alerte quand se produisent en même temps un niveau trop haut dans A ou B et une température trop basse dans la cuve C ou la cuve D.

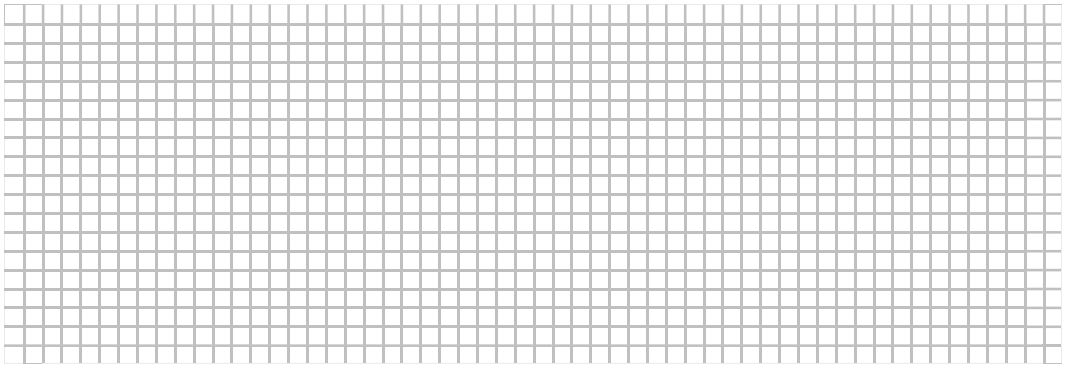
D	C	B	A	S
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

Equations S

S=

NOMS: DATE:
Prénoms:
GROUPE:

Schéma en Nand2



CONCLUSIONS

NOMS: DATE:
Prénoms:
GROUPE:

Techniques numériques- Travaux pratiques avancés

MANIPULATION n°2: Recherche des équations logiques et vérification des montages particuliers suivants:

- **semi- Additionneur**
- **additionneur**

(Temps prévu: 1 séance de 3 heures)

---

1. Introduction.

Dans les Data sheets, ou dans la documentation donnée en début d'année, rechercher les n° des CI correspondant au OR2, AND2, XOR2 , donner le brochage et les tables de vérité

OR2

A	B	S
0	0	
0	1	
1	0	
1	1	

AND2

A	B	S
0	0	
0	1	
1	0	
1	1	

XOR2

A	B	S
0	0	
0	1	
1	0	
1	1	

NOMS: DATE:
Prénoms:
GROUPE:

---

2. Rappel théorique.

Connaître les notes du cours de labo de la page 1 à la page 2.

3. Matériel utilisé:

- 1 Plaquette didactique.
- 1 multimètre.

4. Manipulation:

1. Semi-additionneur. (half adder)

C'est un circuit d'addition qui ajoute deux éléments binaires de même rang.

S = A + B

Compléter la table de vérité C est le report (ou carry) au rang i+1 du rang i

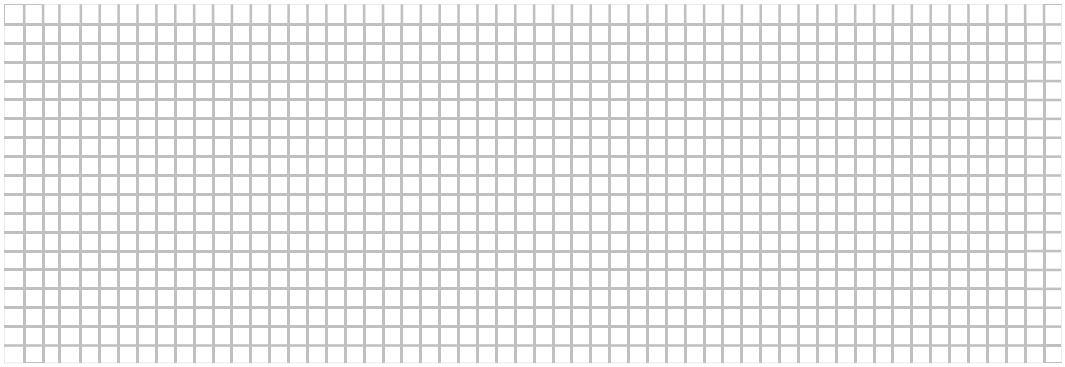
A	B	S	C
0	0		
0	1		
1	0		
1	1		

Tirer les équations logiques

S =

C =

Donner, câbler et simuler les schémas en ET OU PAS XOR



NOMS: DATE:
Prénoms:
GROUPE:

2.Additionneur. (full adder)

C'est un circuit d'addition qui ajoute deux éléments binaires de même rang, mais qui tient compte du report éventuel du rang précédent.

Table de vérité C_{en}= entrée du bit de report = report du rang précédent

C_s = sortie du bit de report

Compléter la table de vérité

A	B	C_{en}	S	C_s
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

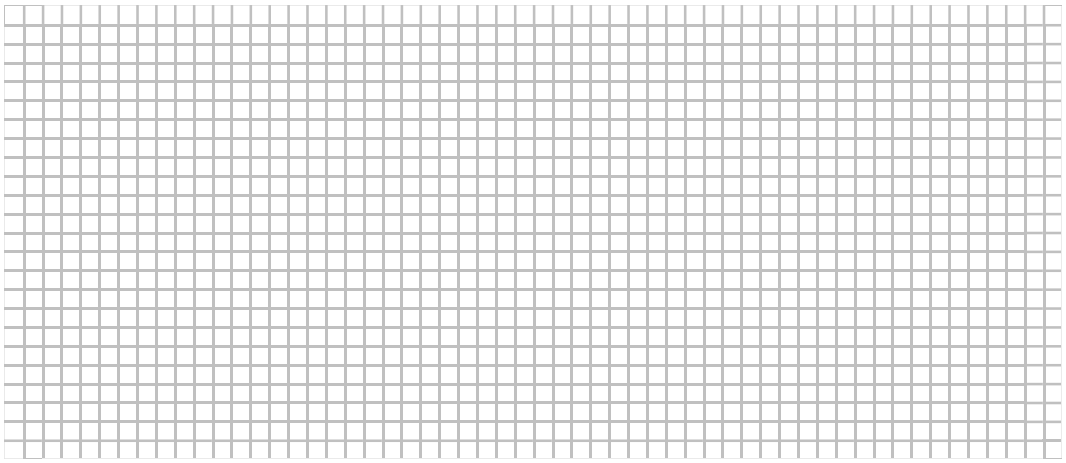
Tirer les équations logiques

S =

Cs =

NOMS: DATE:
Prénoms:
GROUPE:

Donner, câbler et simuler le schéma en ET OU PAS XOR...:



5. CONCLUSIONS

NOMS: DATE:
Prénoms:
GROUPE:

Techniques numériques- Travaux pratiques avancés

MANIPULATION n° 3 : Recherche des équations logiques et vérification des montages particuliers suivants:

- **semi- soustracteur**
- **soustracteur**

(Temps prévu: 1 séance de 3 heures)

---

1. Introduction. (Cf Manipulation n°8)

Dans les Data sheets, ou dans la documentation donnée en début d'année, rechercher les n° des CI correspondant au OR2, AND2, XOR2 , donner le brochage et les tables de vérité

OR2

A	B	S
0	0	
0	1	
1	0	
1	1	

AND2

A	B	S
0	0	
0	1	
1	0	
1	1	

XOR2

A	B	S
0	0	
0	1	
1	0	
1	1	

NOMS: DATE:
Prénoms:
GROUPE:

---

2. Rappel théorique.

Connaître les notes du cours de labo de la page 3 à la page 4.

3. Matériel utilisé:

- 1 Plaquette didactique.
- 1 multimètre.

4. Manipulation:

1. Semi-soustracteur. (half subtractor)

C'est un circuit de soustraction qui soustrait un élément binaire d'un autre de même rang..

D = A – B (D → Différence et - → Moins)

Compléter la table de vérité B0 est l'emprunt (Borrow) au rang supérieur.

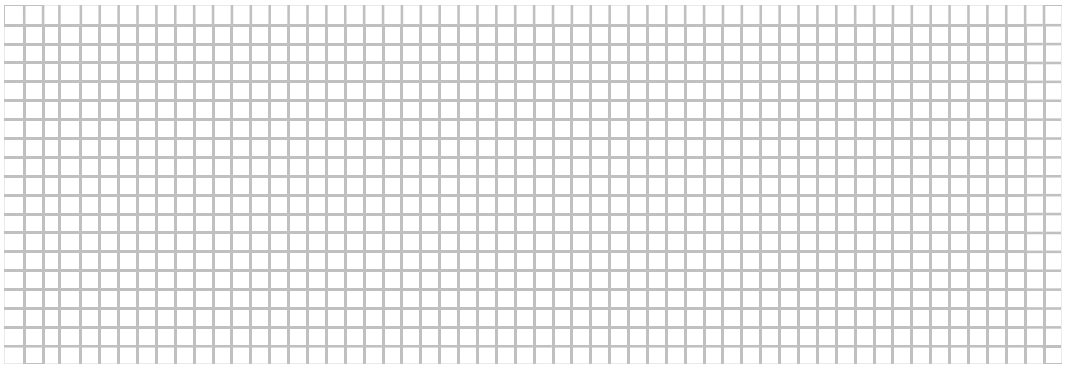
A	B	D	B0
0	0		
1	0		
1	1		
0	1		

Tirer les équations logiques

D =

Bo=

Donner, câbler et simuler le schéma en ET OU PAS XOR...:



NOMS: DATE:
Prénoms:
GROUPE:

2.Soustracteur. (full subtractor)

C'est un circuit de soustraction qui soustrait un élement binaire d'un autre de même rang, mais qui tient compte du retrait éventuel d'un emprunt du rang précédent.

Table de vérité Bo_{en}= retrait d'un emprunt du rang précédent
Bos = emprunt

A	B	Bo_{en}	D	Bo_s
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

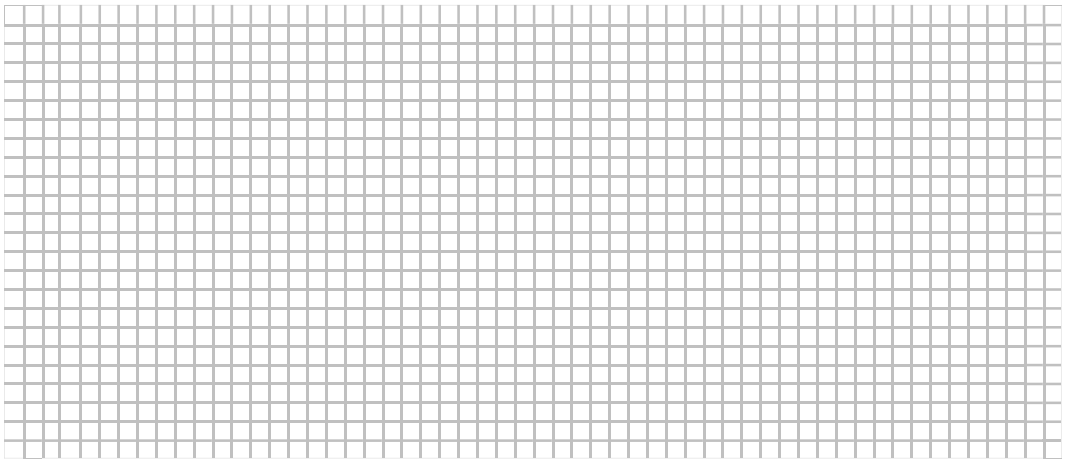
Tirer les équations logiques

D =

Bos =

NOMS: DATE:
Prénoms:
GROUPE:

Donner, câbler et simuler les schémas en ET OU PAS XOR...:



5. CONCLUSIONS.

NOMS: DATE:
Prénoms:
GROUPE:

Techniques numériques- Travaux pratiques avancés

MANIPULATION n° 4 : Additionneur parallèle intégré

(Temps prévu: 1 séance de 3 heures)

---

1. But de la manipulation.

Etude du circuit additionneur parallèle intégré: le 74LS83.

2. Rappel théorique.

Connaître les notes du cours de labo de la page 5 à la page 9.

3. Matériel utilisé:

- 1 Plaquette didactique.
- 1 multimètre.

4. Manipulation:

1. Rechercher dans le catalogue le brochage du 74LS83.

2. Additionner 2 quartets.

Alimenter le CI

Câbler les entrées A et B aux 8 dispwiches de la plaquette

Câbler les 4 sorties à l'afficheur 7 segments (PF= D) ainsi qu'à 4 diodes Led.

Câbler Cout = C_4 à une diode Led.

Câbler Cin= C_0 selon les directives imposées.

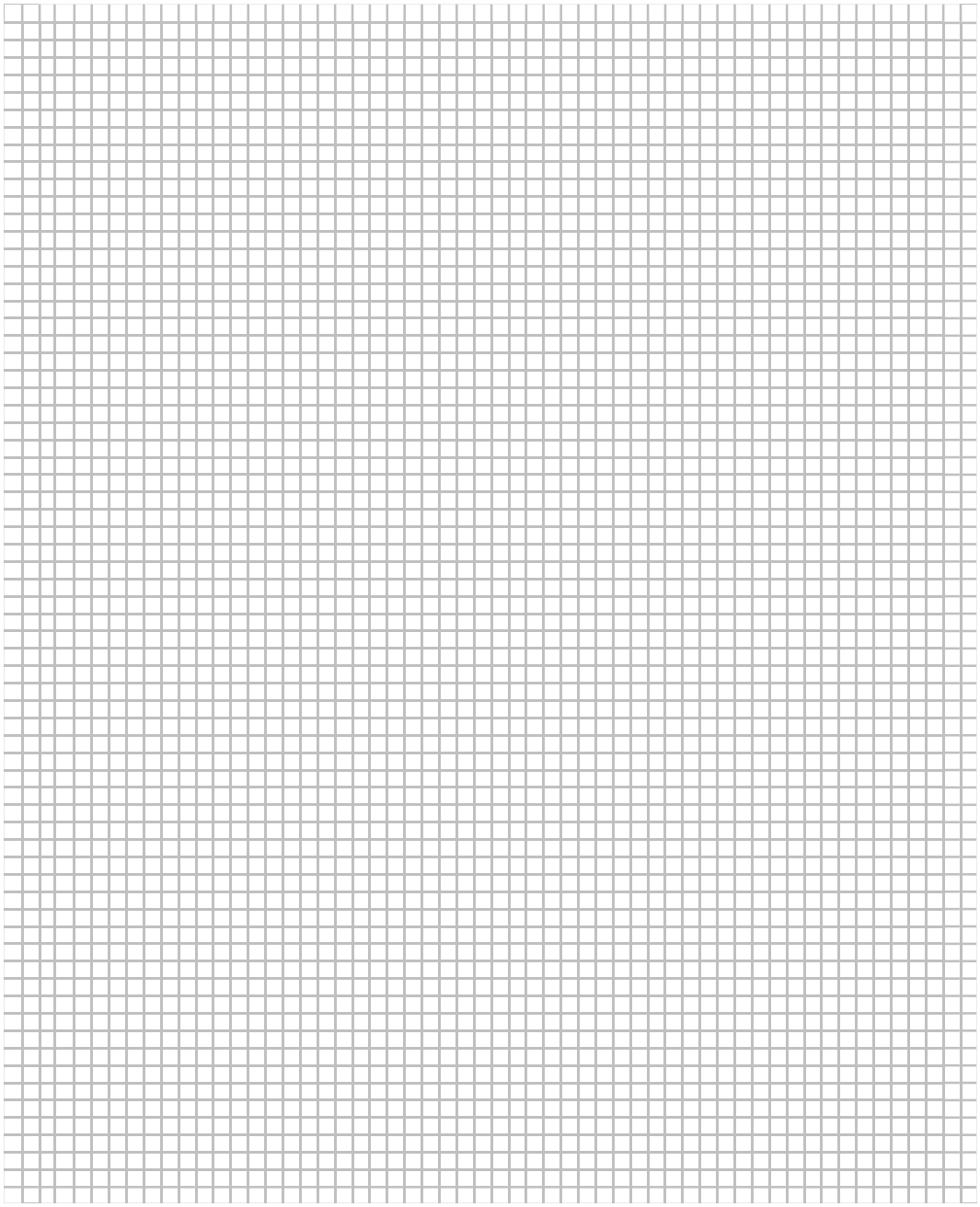
Réaliser le schéma de principe pour le premier exemple.

NOMS: DATE:
Prénoms:
GROUPE:

Exemples:

Cin	A		B		Cout	s		Justifier
	binaire	décimal	binaire	décimal		Leds	Afficheur	
0	1010		1010					
1	1010		1010					
0	1010		0101					

Schéma de principe.



Câbler et simuler.

NOMS: DATE:
Prénoms:
GROUPE:

---

3. Soustraire 2 quartets.

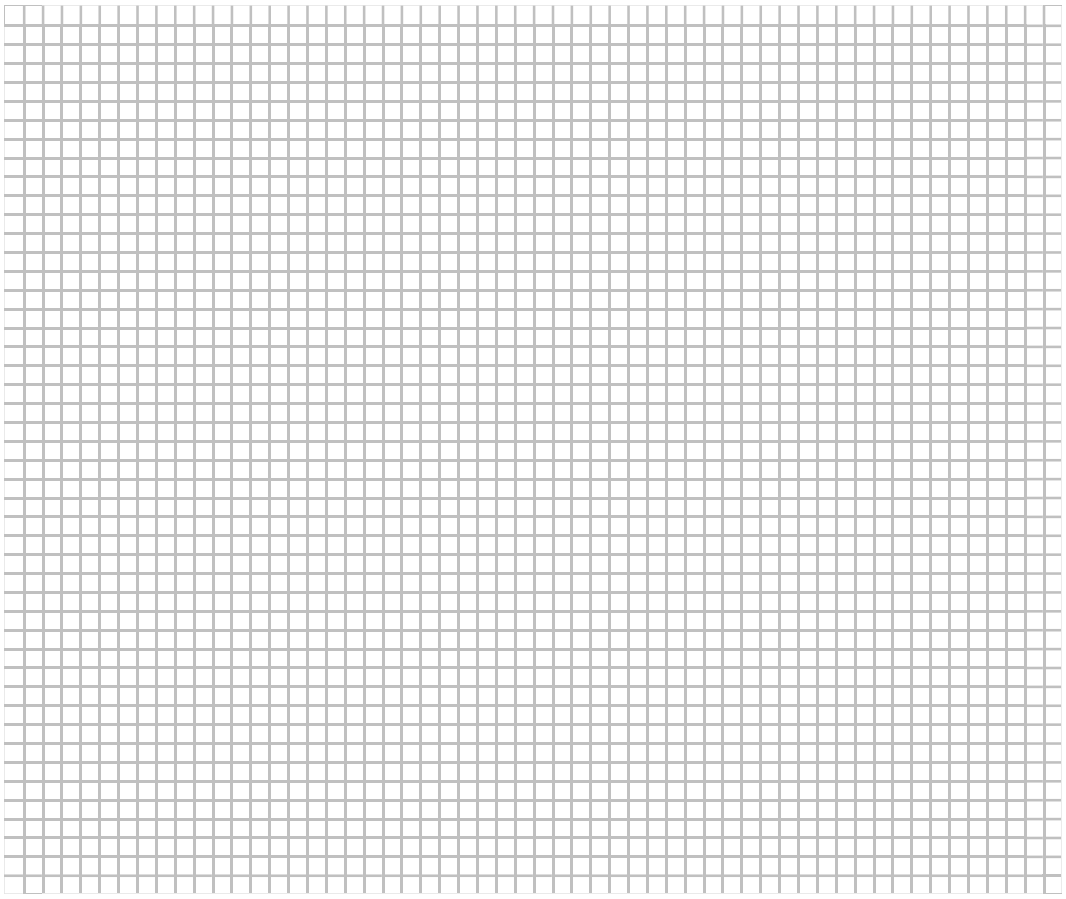
Donner 2 méthodes permettant de soustraire les nombres suivants:

A= 1101 B= 0110

I- Donner le schéma de principe selon la méthode du complément à 1.

a) Expliquer la méthode à partir de l'exemple ci-dessus.

b) Réaliser le câblage et simuler.

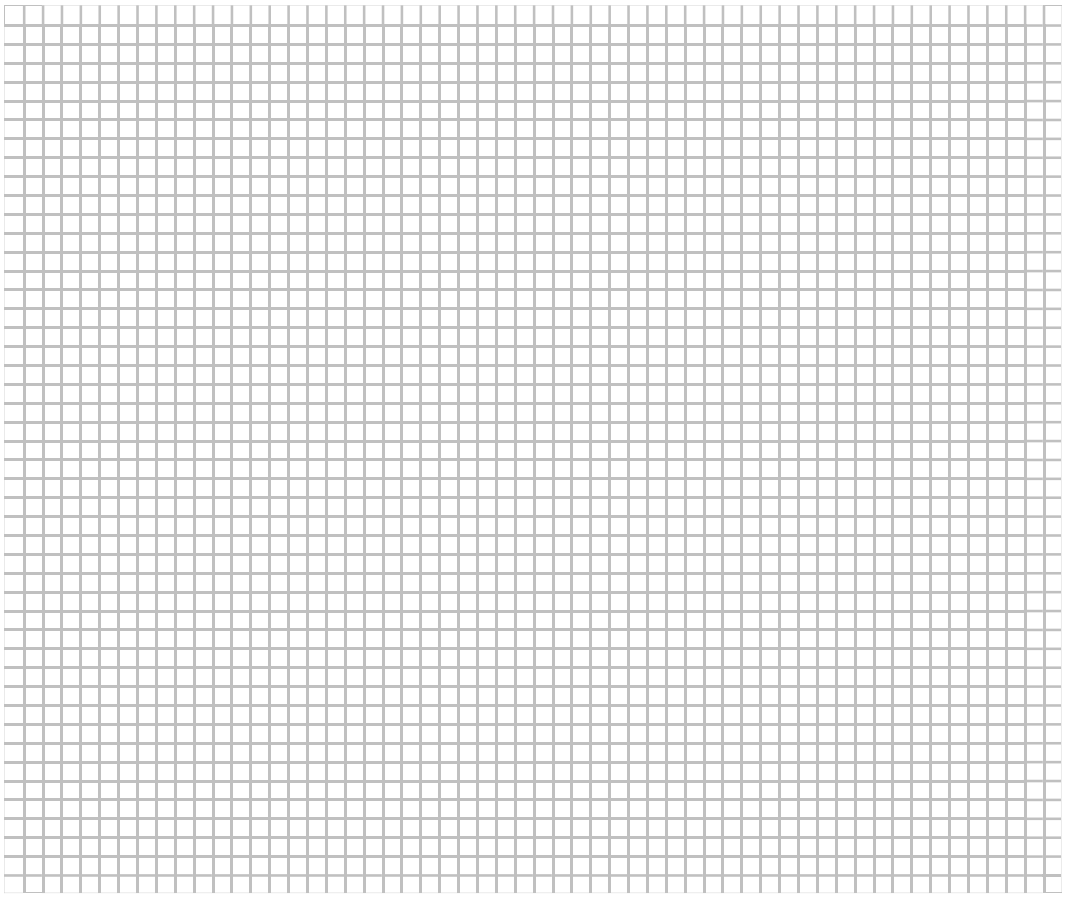


NOMS: DATE:
Prénoms:
GROUPE:

II- Donner le schéma de principe selon la méthode du complément à 2.

a) Expliquer la méthode à partir de l'exemple ci-dessus.

b) Réaliser le câblage et simuler.



5. Conclusions.

NOMS: DATE:
Prénoms:
GROUPE:

Techniques numériques- Travaux pratiques avancés

MANIPULATION n° 5 : comparateur intégré

(Temps prévu: 1 séance de 3 heures)

---

1. But de la manipulation.

Etude du circuit additionneur parallèle intégré: le 74LS85.

2. Rappel théorique.

Connaître les notes du cours de labo de la page 11 à la page 15.

3. Matériel utilisé:

- 1 Plaquette didactique.
- 1 multimètre.

4. Manipulation:

1. Brochage 74LS85

Rechercher dans le catalogue le brochage du 74LS85.

2. Comparer 2 quartets. (Câblage et simulation)

Alimenter le CI

Câbler les entrées A et B aux 8 dispwiches de la plaquette

Câbler les 3 sorties du comparateur à 3 diodes Led.

Réaliser le schéma de principe pour un exemple où le mot A >B

Trouver 2 exemples où A > B

NOMS: DATE:
Prénoms:
GROUPE:

Trouver 2 exemples où A < B

Trouver 2 exemples où A = B

Schéma de principe.

3. Comparer 2 octets.(câblage et simulations)

Donner le schéma de principe qui permette de comparer 2 octets

Raccorder tous les A0-A1-A2-A3 ensemble et au dipswich

Raccorder tous les A4-A5-A6-A7 ensemble et au dipswich

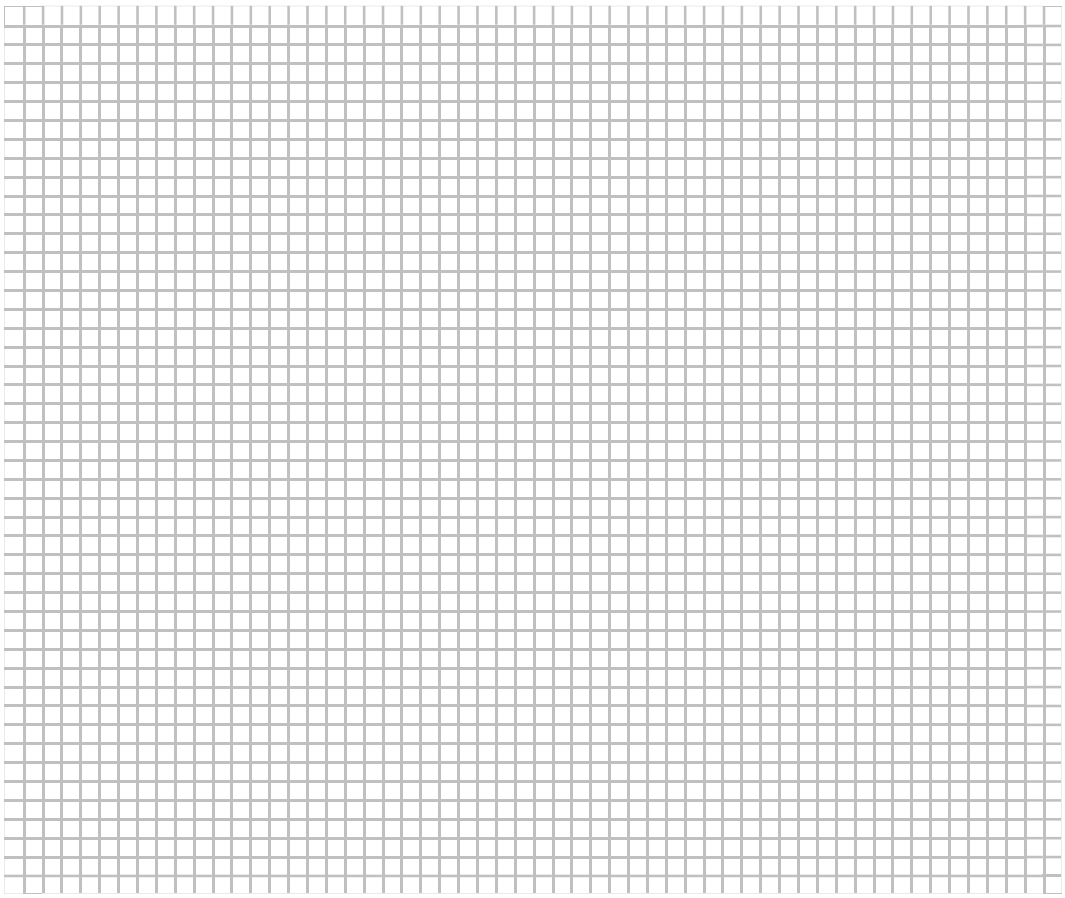
Raccorder tous les B0-B1-B2-B3 ensemble et au dipswich

Raccorder tous les B4-B5-B6-B7 ensemble et au dipswich

NOMS: DATE:
Prénoms:
GROUPE:

		Constatations
A = FF	B = F0	
A = FF	B = 0F	
A = F0	B = FF	
A= 0F	B = FF	
A= FF	B = FF	

Schéma de principe



5. Conclusion

NOMS: DATE:
Prénoms:
GROUPE:

Techniques numériques- Travaux pratiques avancés

MANIPULATION n° 6 : Recherche des équations logiques et vérification des montages particuliers suivants:

• Générateur de parité (4 entrées).

• Détecteur de parité (4 entrées).

(Temps prévu: 1 séance de 3 heures)

---

1. But.

Créer un générateur de bit de parité

Injecter ce bit dans un schéma détecteur de parité.

2. Rappel théorique..

Connaître les notes du cours de labo de la page 9 à la page 10.

3. Matériel utilisé:

- 1 Plaquette didactique.
- 1 multimètre.

4. Manipulation:

1. Générateur de parité.

Principe:

Le contrôle de parité consiste à ajouter un bit de parité tel que le nombre de 1 dans l'information totale soit pair.

Manipulation:

1. Compléter la table de vérité suivante:

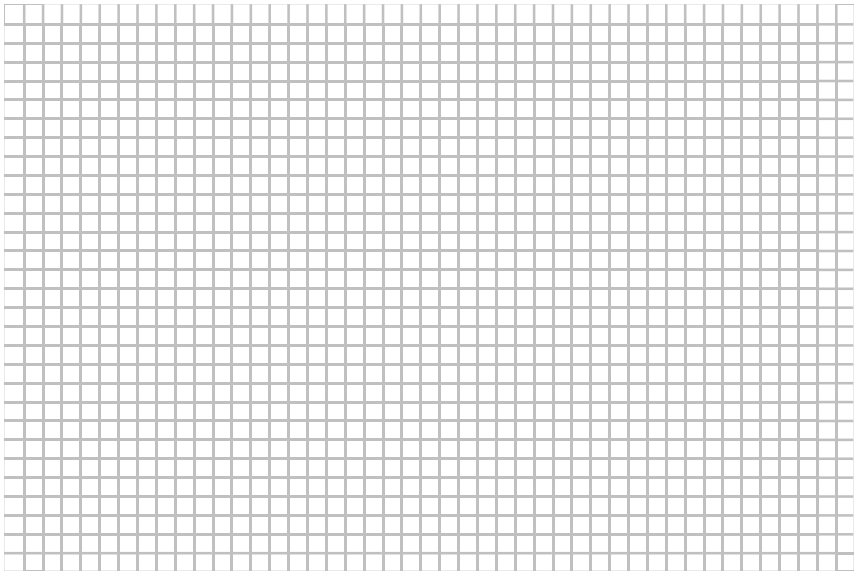
D	C	B	A	P	I
0	0	0	0	**0**	**1**
0	0	0	1	**1**	**0**
0	0	1	0	**1**	**0**
0	0	1	1	**0**	**1**
0	1	0	0	**1**	**0**
0	1	0	1	**0**	**1**
0	1	1	0	**0**	**1**

NOMS: DATE:
Prénoms:
GROUPE:

0	1	1	1	**1**	**0**

2. Etablir l'équation logique dans le cas de 4 entrées.

3. Rechercher le schéma de principe , le câbler et le simuler.



4. Vérifier la table de vérité.

NOMS: DATE:
Prénoms:
GROUPE:

---

2. Détecteur de parité.

Principe:

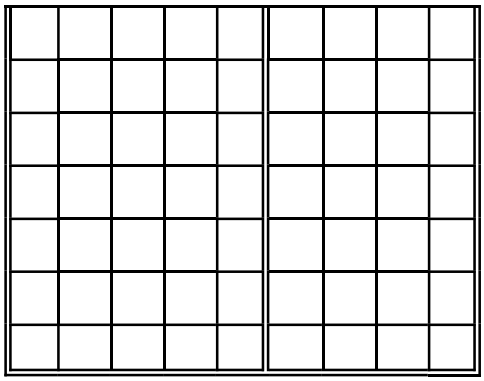
C'est un circuit qui contrôle si le nombre de bits à 1 est pair ou impair. Si l'on effectue un contrôle de parité paire et que le nombre de bits à 1 est impair, le circuit indiquera qu'il y a une erreur ou bloquera les données en sortie.(Sorties à '0' logique)

Manipulation:

1. Compléter la table de vérité suivante:

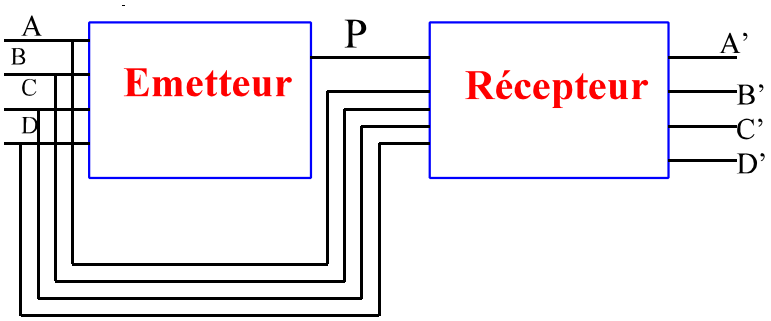
D	C	B	A	P	D'	C'	B'	A'
0	0	0	0	0	**0**	**0**	**0**	**0**
0	0	0	0	1	**0**	**0**	**0**	**0**
0	0	0	1	0	**0**	**0**	**0**	**0**
0	0	0	1	1	**0**	**0**	**0**	**1**
0	0	1	0	0	**0**	**0**	**0**	**0**
0	0	1	0	1	**0**	**0**	**1**	**0**
0	0	1	1	0	**0**	**0**	**1**	**1**
0	0	1	1	1	**0**	**0**	**0**	**0**
0	1	0	0	0	**0**	**0**	**0**	**0**
0	1	0	0	1	**0**	**1**	**0**	**0**
0	1	0	1	0	**0**	**1**	**0**	**1**
0	1	0	1	1	**0**	**0**	**0**	**0**

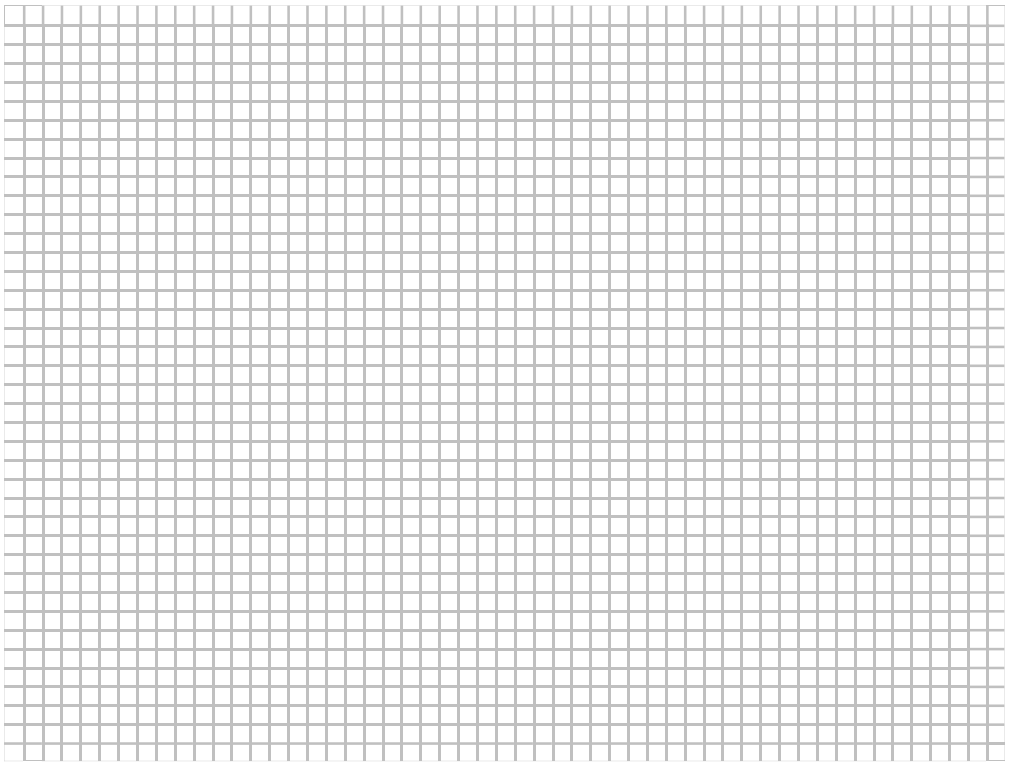
NOMS: DATE:
Prénoms:
GROUPE:



2. Etablir l'équation logique dans le cas de 4 entrées. (+ bit de parité).

3. Rechercher le schéma de principe, le câbler et le simuler.





4. Vérifier la table de vérité.

5. CONCLUSIONS

NOMS: DATE:
Prénoms:
GROUPE:

Techniques numériques- Travaux pratiques avancés

MANIPULATION n° 7 :

- Décodeur 1 parmi 8

- Transcodeur binaire - Gray

- Transcodeur Gray - binaire

(Temps prévu: 1 séance de 3 heures)

---

1. But.

- Câbler un décodeur 1 parmi 8. (Utiliser un 74138)
- Etablir les équations et câbler un transcodeur binaire- Gray, rechercher le schéma et le tester.
- Etablir les équations et câbler un transcodeur Gray- binaire; rechercher le schéma et le tester.

2. Rappel théorique..

Connaître les notes du cours de labo de la page 16 à la page 35.

3. Matériel utilisé:

- 1 Plaquette didactique.
- 1 multimètre.

4. Manipulation:

1. Décodeur 1 parmi 8.

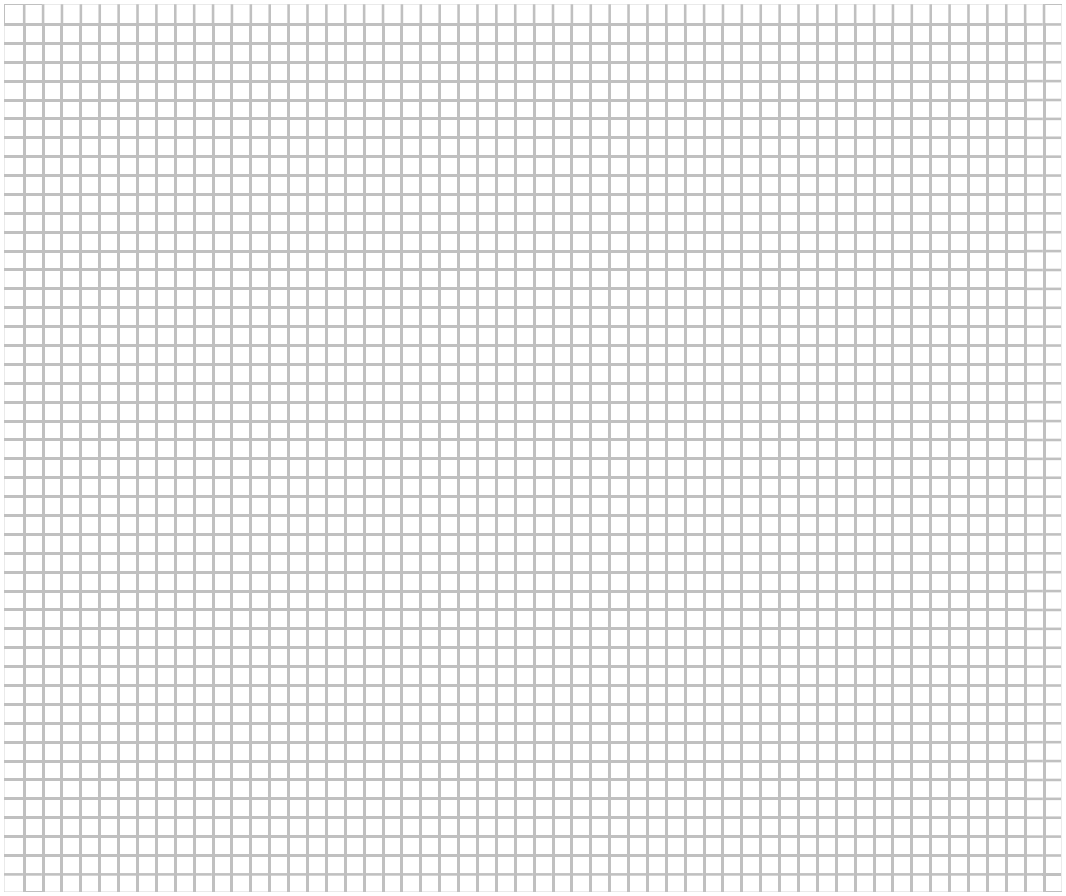
Rechercher le brochage du circuit 74LS138

NOMS: DATE:
Prénoms:
GROUPE:

Manipulation: (câblage et simulation)

Alimenter le CI
Câbler les entrées à 6 dispwiches
Câbler les sorties aux diodes led.

Schéma de câblage



Indiquez les états des sorties d'un 74LS 138 pour chacune des conditions d'entrée que voici et justifiez à l'aide de la table de vérité

a) $\overline{E_1} = 0,\ \overline{E_2} = 1,\ E_3 = 1,\ A_2 = 1,\ A_1 = 1$ et $A_0 = 0$

b) $\overline{E_1} = 0,\ \overline{E_2} = 0,\ E_3 = 1,\ A_2 = 0,\ A_1 = 1$ et $A_0 = 1$

NOMS: DATE:
Prénoms:
GROUPE:

Conclusion:

A quoi sert le CI?

Comment utilise-t'on le circuit? (Conditions de fonctionnement)

2. Transcodeurs

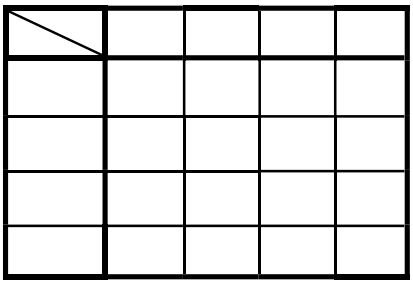
Créez un transcodeur Gray-binaire.

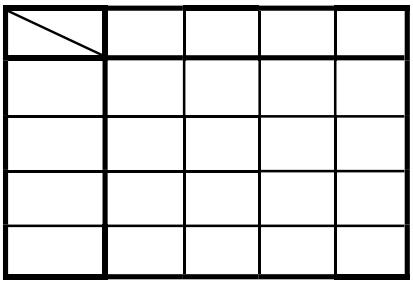
Table de vérité

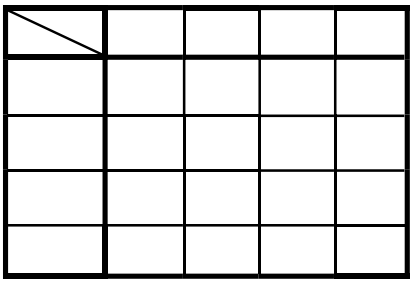
décimal	GRAY				BINAIRE			
N	D	C	B	A	S_3	S_2	S_1	S_0
0								
1								
2								
3								
4								
5								
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								

NOMS: DATE:
Prénoms:
GROUPE:

Recherche des équations.



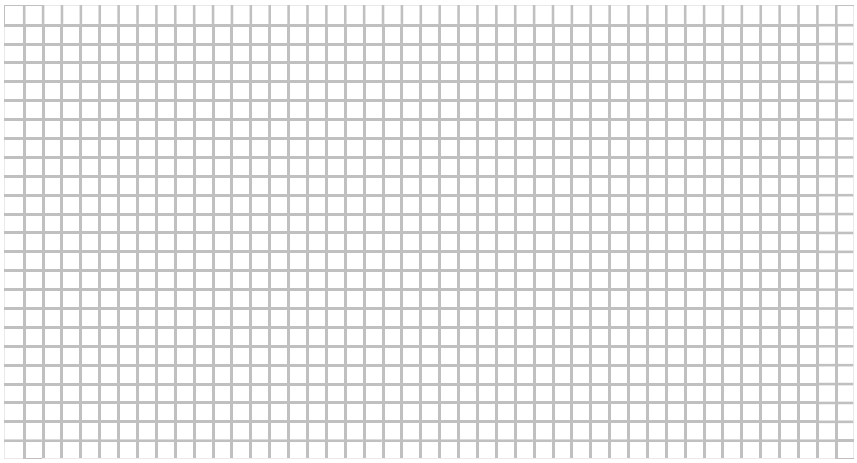




NOMS: DATE:
Prénoms:
GROUPE:

Schéma de câblage à câbler et simuler.

Réalisez le schéma et vérifiez-le.



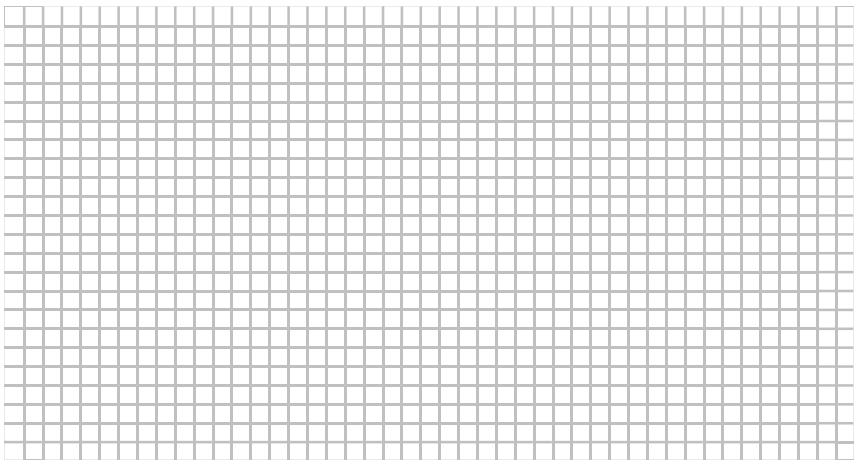
Créez un transcodeur binaire -Gray.

A l'aide de la TDV ci-dessus, retrouver les **équations logiques de ce transcodeur**

Recherche des équations.

NOMS: DATE:
Prénoms:
GROUPE:

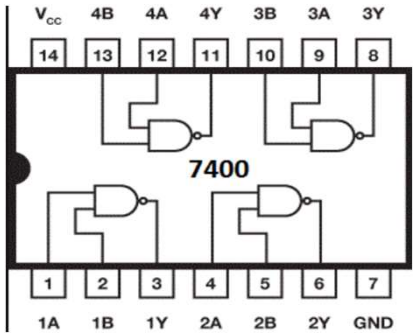
Recherchez le schéma, câblez-le et simulez-le



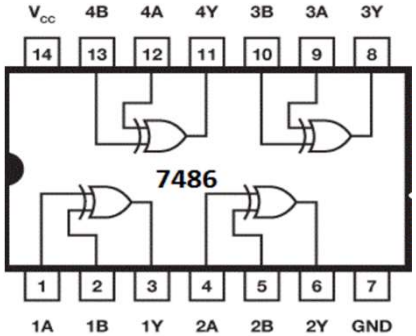
5- Conclusions:

BROCHAGE DES CI CONTENUS DANS LA BOITE.

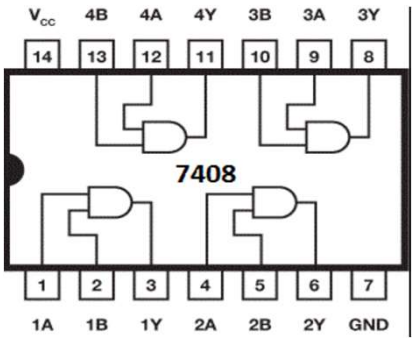
7400



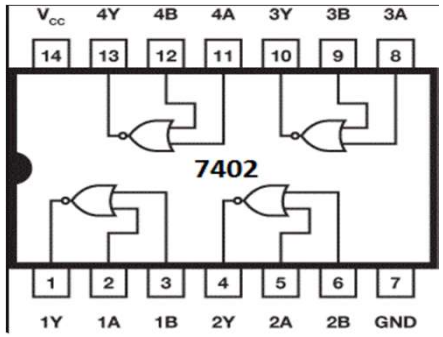
7486



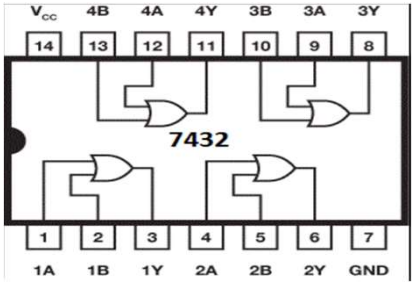
7408



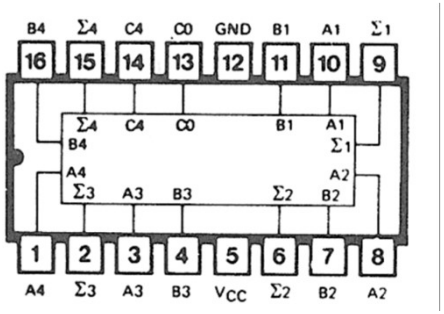
7402



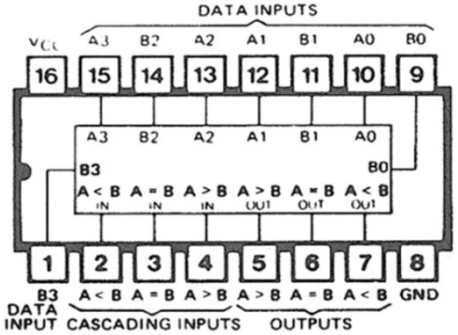
7432



7483



7485



74138