

LOGIQUE COMBINATOIRE

I- Les circuits intégrés combinatoires

1- Additionneur:

- ✎ Binaire.
- ✎ BCD.

2- Comparateur

3- Multiplexeur/démultiplexeur

II- Unité Arithmétique et Logique (U.A.L)

1- Fonction

2- Architecture

3- Applications

CONTENU
DU PROGRAMME



OBJECTIFS
DU PROGRAMME

- ☞ OS A₁₁ - Mettre en œuvre un circuit intégré combinatoire.
- ☞ OS A₁₂ - Réaliser des applications à base d'U.A.L.

LES CIRCUITS INTÉGRÉS COMBINATOIRES

A. MISE EN SITUATION

La rareté des espaces disponibles en milieu urbain, conjuguée à une augmentation croissante du nombre de véhicules en circulation ont fait des systèmes de gestion de parkings, des outils familiers du paysage urbain. Ces parkings peuvent être publics ou privés et par conséquent payants, à accès par abonnement ou gratuits.

I- Présentation:

Système d'étude: contrôleur d'accès de parking

Le parking objet de cette mise en situation et du type privé dont l'accès est autorisé ou refusé en fonction des droits de l'utilisateur et du nombre de places disponibles. Ainsi, lorsqu'une personne arrive en voiture et présente son badge à l'entrée du parking, si ses droits ne l'y autorisent pas ou si le nombre maximal de places affectées à sa catégorie est atteint, l'accès lui sera refusé et il sera redirigé vers la sortie par le biais de balises.

Ce type de fonction est utile quand plusieurs sociétés se partagent le même parking ou lorsqu'une même société gère ses places de parking en les affectant à des services distincts et identifiés.

L'agent d'exploitation peut visualiser en temps réel le nombre de places disponibles et dispatcher manuellement le nombre de places restantes pour chaque groupe de personnes.

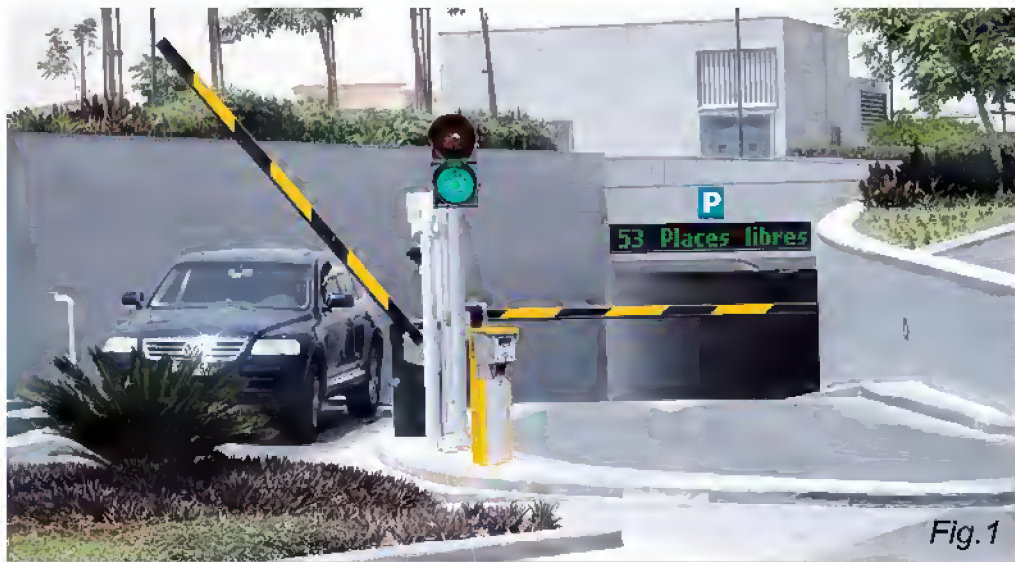


Fig.1

NB : Certains parkings peuvent aller plus loin dans la qualité des services offerts aux clients en définissant par exemple les zones autorisées et en y affectant des plages horaires d'accès. Ils peuvent ainsi gérer les flux de personnel en toute sécurité.

II- Fonctionnement

Ce système permet, à partir des informations contenues dans le badge, de consulter la base de données des abonnés afin de vérifier les droits affectés à cet identifiant: date de début et de fin de validité, zones autorisées, dates et heures d'accès, etc.

En fonction de ces droits et du nombre de places disponibles, le système autorise ou non l'accès au parking et déclenche ou non l'ouverture de la barrière.

III- Contraintes imposées par le cahier des charges

- ✎ Le contrôle de l'accès au parking est à effectuer par une signalisation bicolore conforme à la réglementation du code de la route: feu vert accès autorisé et feu rouge accès interdit.
- ✎ Il doit y avoir la possibilité de **modifier** le nombre de places utilisables (N_p) qui est au maximum de 99 places.
- ✎ L'agent d'exploitation peut **visualiser** à travers un affichage lumineux le nombre de places occupées. Il peut aussi **initialiser** le nombre de véhicules présents dans le parking selon la réservation de chaque groupe.
- ✎ Le système **compare** le nombre de véhicules garés (N_v) au nombre de places du parking (N_p), si ces deux derniers sont égaux, le feu passera au rouge et la barrière ne s'ouvrira plus pour interdire l'entrée d'éventuelles voitures.
- ✎ Il doit y avoir la possibilité d'**afficher** le nombre de places libres ($N_{pl}=N_p-N_v$) ou l'information «Parking complet» visible de l'extérieur.

En plus, d'autres options telles que la mise en marche continue du feu rouge et la fermeture de la barrière même si le nombre maximum de véhicules n'est pas atteint, permettant ainsi d'interdire temporairement l'accès pour cause de travaux, d'accident ou autres.

IV- Schéma fonctionnel

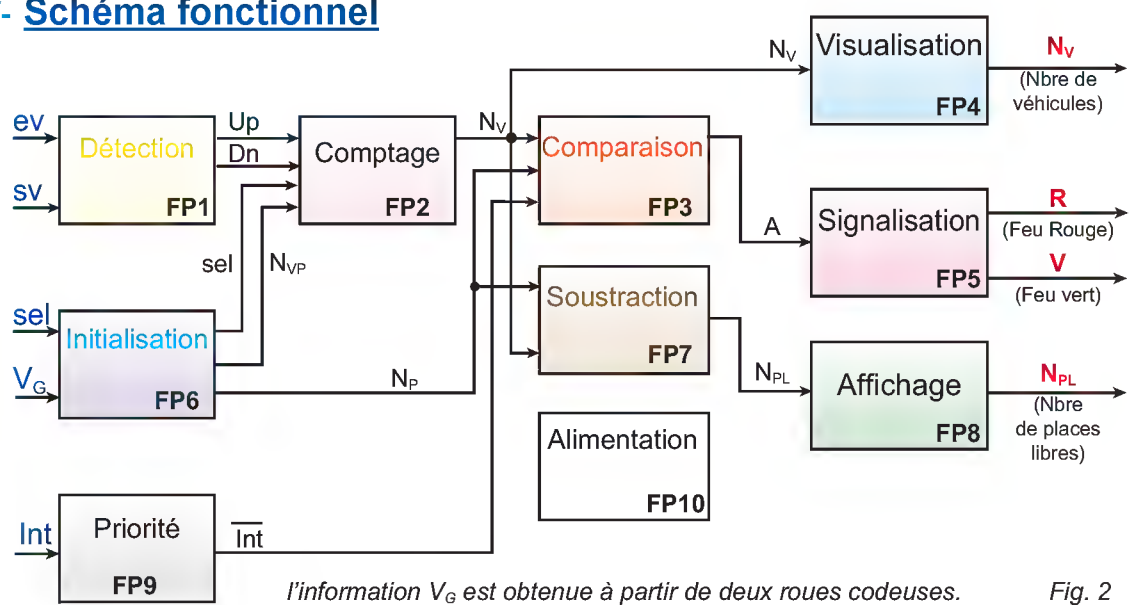


Fig. 2

AUTOMATIQUE

REMARQUE:

Le schéma structurel du système est donné à la fin du manuel d'activités.

LÉGENDE

- ✎ **ev**: entrée d'un véhicule (ev=1);
- ✎ **sv**: sortie d'un véhicule (sv=1);

- ✎ **Up**: impulsion au niveau logique bas lorsqu'un véhicule entre dans le parking;
- ✎ **Dn**: impulsion au niveau logique bas lorsqu'un véhicule sort du parking;

- ✎ **N_{VP}**: Mot de 8 bits codé en BCD représentant le nombre de véhicules présents dans le parking (utilisé pour la mise en fonctionnement ou la correction d'anomalie);
- ✎ **N_V**: Mot de 8 bits codé en BCD représentant le nombre de véhicules garés;
- ✎ **N_P**: Mot de 8 bits codé en BCD représentant le nombre de places du parking;
- ✎ **N_{PL}**: Mot de 8 bits codé en BCD représentant le nombre de places libres;

- ✎ **\overline{INT}** : Information logique permettant l'interdiction d'accès au parking en cas de travaux, etc...;
- ✎ **V_G**: Nombre de voitures garées (y compris les places réservées);
- ✎ **sel**: bouton poussoir permettant la validation de la mise à jour du nombre de voitures garées (**N_{VP}**) par réglage des roues codeuses.
 - Si **sel** n'est pas appuyé, les roues codeuses indiquent le nombre de places du parking (**N_P**) et l'afficheur indique le nombre de véhicules entrés.
 - Si **sel** est appuyé après une mise à jour du nombre de places du parking suite à une réservation, l'afficheur indiquera le nombre de places utilisées.

- ✎ **A**: permet de commander l'allumage d'un feu vert en cas d'autorisation d'accès au parking et d'un feu rouge dans le cas contraire;
- ✎ **R**: feu rouge signalant l'interdiction d'accès au parking;
- ✎ **V**: feu vert signalant l'autorisation d'accès au parking;

- ✎ **Alimentation**: le rôle de cette fonction est de fournir l'énergie électrique nécessaire au fonctionnement du système.

Problématique:

Quelles fonctions techniques et, par conséquent, quels circuits doit-on mettre en œuvre pour répondre aux exigences du cahier des charges de ce parking?

B. LES CIRCUITS INTÉGRÉS COMBINATOIRES

1- L'additionneur

1- Introduction

Pour gérer les informations à afficher à l'utilisateur (exemple: nombre de places disponibles), l'unité de gestion de ce parking est appelée à faire entre autres des opérations d'arithmétique telles que l'addition et la soustraction.

L'addition est l'opération arithmétique la plus importante dans les systèmes numériques.

Cette opération peut être matérialisée soit par des cellules logiques de base ou par un circuit combinatoire spécialisé, appelé additionneur.

Cette section a pour fin la mise en œuvre des additionneurs binaires intégrés parallèles et B.C.D.

2- Principe

L'addition de deux nombres binaires est analogue à l'addition de deux nombres décimaux. Le principe est résumé dans la table de vérité suivante:

a	b	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

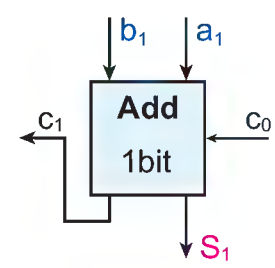
Avec **S** : la somme
C : la retenue éventuelle

3- L'additionneur binaire

a. Additionneur élémentaire

La cellule de base est un additionneur élémentaire de deux nombres binaires à 1 seul bit, répondant à la notation suivante:

- ✎ a_1 : le bit du nombre binaire A
- ✎ b_1 : le bit du nombre binaire B
- ✎ S_1 : étant la somme
- ✎ c_0 : la retenue en entrée
- ✎ c_1 : la retenue à la sortie



b. Structure interne d'un additionneur à n bits

Un additionneur à n bits est l'association de n additionneurs élémentaires à 1 bit.

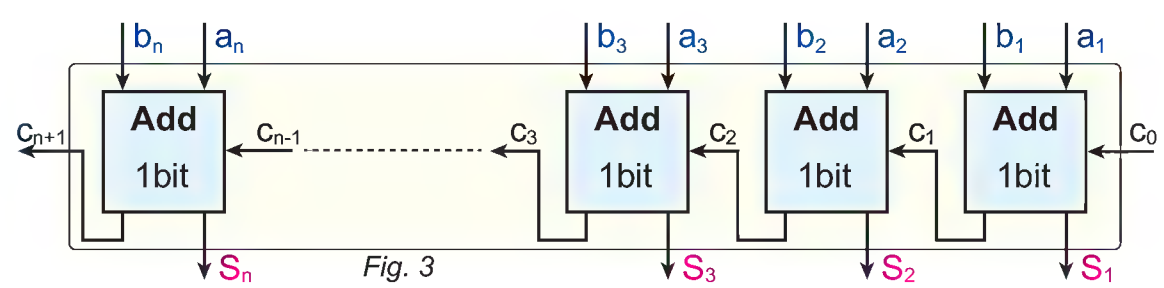


Fig. 3

LOGIQUE COMBINATOIRE

c. Additionneurs binaires intégrés

références usuelles

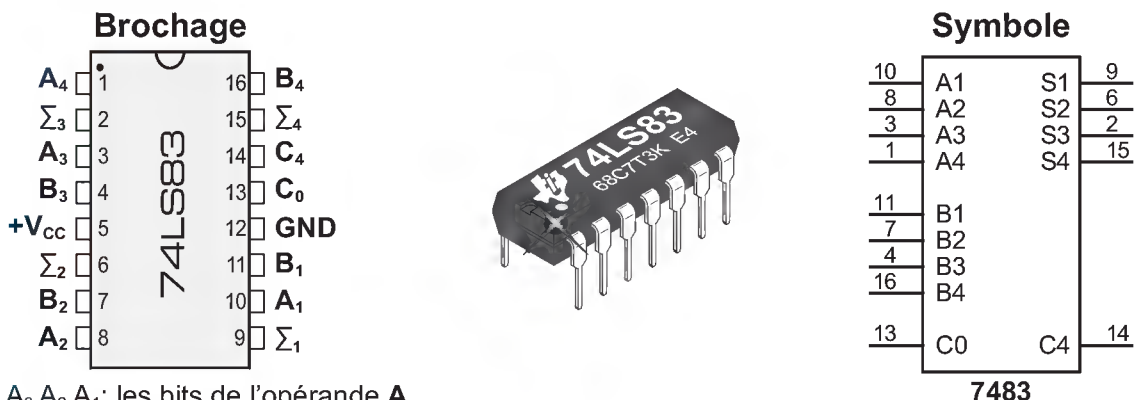
Le tableau ci-dessous illustre quelques exemples d'additionneurs en circuits intégrés.

En technologie T.T.L		En technologie C.M.O.S
Additionneur à 2 bits	Additionneur à 4 bits	Additionneur à 4 bits
7482	7483, 74LS83, 74283 74HC283, 74LS283	4008

REMARQUE:

Les circuits **74283** sont identiques aux circuits **7483** sauf que **Vcc** et la **masse** sont respectivement sur les broches 16 et 8. Ce brochage est conforme à la norme appliquée actuellement qui consiste à placer les broches d'alimentation et de la masse aux coins des boîtiers.

brochage et symbolisation des circuits de la série **74XX83**



$A_4 A_3 A_2 A_1$: les bits de l'opérande **A**
 $B_4 B_3 B_2 B_1$: les bits de l'opérande **B**.
 $S_4 S_3 S_2 S_1$ ou $(\Sigma_4 \Sigma_3 \Sigma_2 \Sigma_1)$: les bits de la somme
 C_0 : la retenue à l'entrée.
 C_4 : la retenue à la sortie

Fig. 4

d. Mise en œuvre des additionneurs binaires intégrés

L'additionneur **7482**

Soit à additionner **3+2**. Pour réaliser cette opération, l'unité de traitement procède comme suit:

- Convertir 3 en base 2 $(3)_{10} = (0 \ 1 \ 1)_2$
- Convertir 2 en base 2 $(2)_{10} = (0 \ 1 \ 0)_2$
- Additionner les deux nombres bit à bit $(5)_{10} = (1 \ 0 \ 1)_2$
- Convertir le résultat trouvé en décimal $(1 \ 0 \ 1)_2 = (5)_{10}$

Schéma de montage:

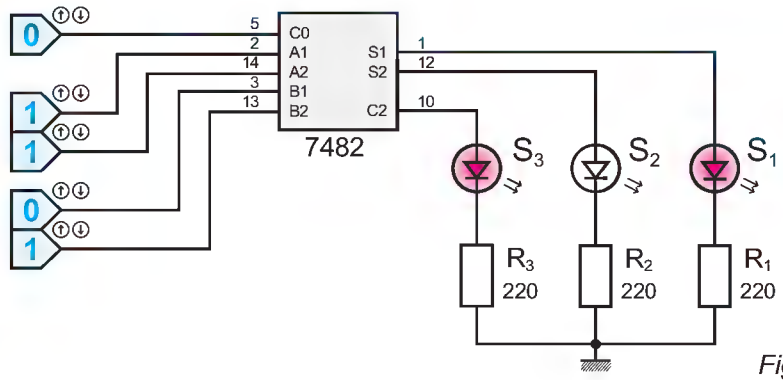


Fig. 5

➤ L'additionneur **7483**

Soit à additionner par exemple **14+7**, l'unité de traitement procède comme suit:

➤ Par le biais d'un dispositif adéquat

♦ Convertir **14** en base 2

$$(14)_{10} = \begin{matrix} & 1 & 1 & 1 \\ & 0 & 1 & 1 & 1 & 0 \end{matrix} _2$$

♦ Convertir **7** en base 2

$$(7)_{10} = \begin{matrix} & 0 & 0 & 1 & 1 & 1 \end{matrix} _2$$

Au moyen de l'additionneur

♦ Additionner les deux nombres bit à bit

$$= \underline{\underline{\begin{matrix} & 1 & 0 & 1 & 0 & 1 \end{matrix} _2}} = (21)_{10}$$

➤ Un dispositif interne convertit le résultat trouvé en décimal soit:

$$(10101)_2 = (21)_{10}$$

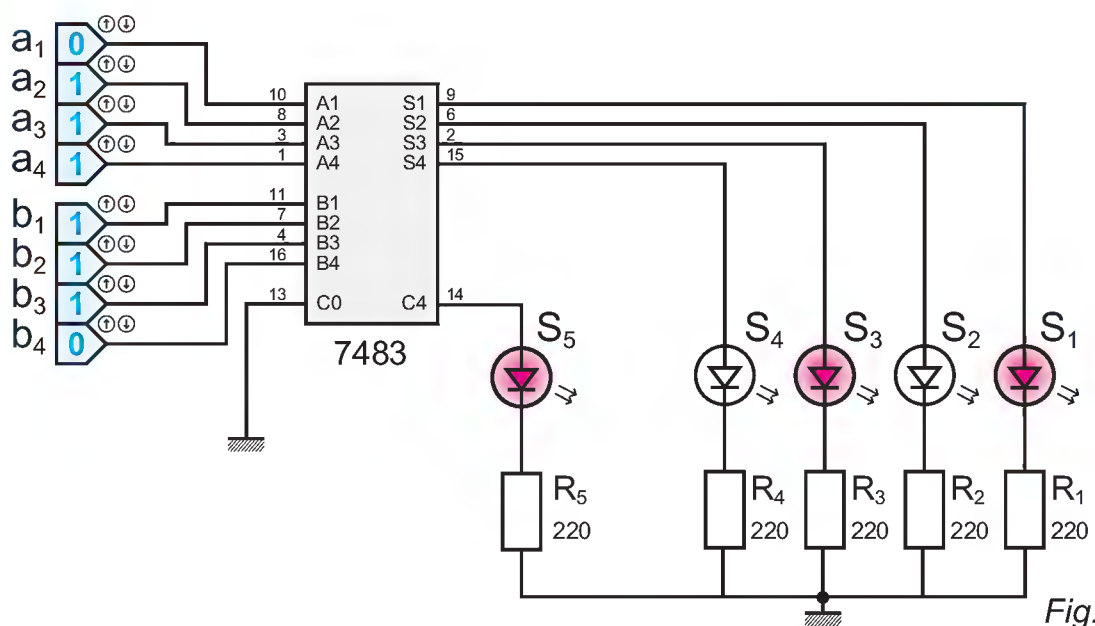


Fig. 6

AUTOMATIQUE

🔗 Mise en cascade d'additionneurs intégrés

Pour additionner deux nombres de plus de 4 bits, il faut associer en cascade plusieurs additionneurs de 2 ou de 4 bits.

Exemple: additionneur de deux nombres à **6 bits** à base du circuit **7483**

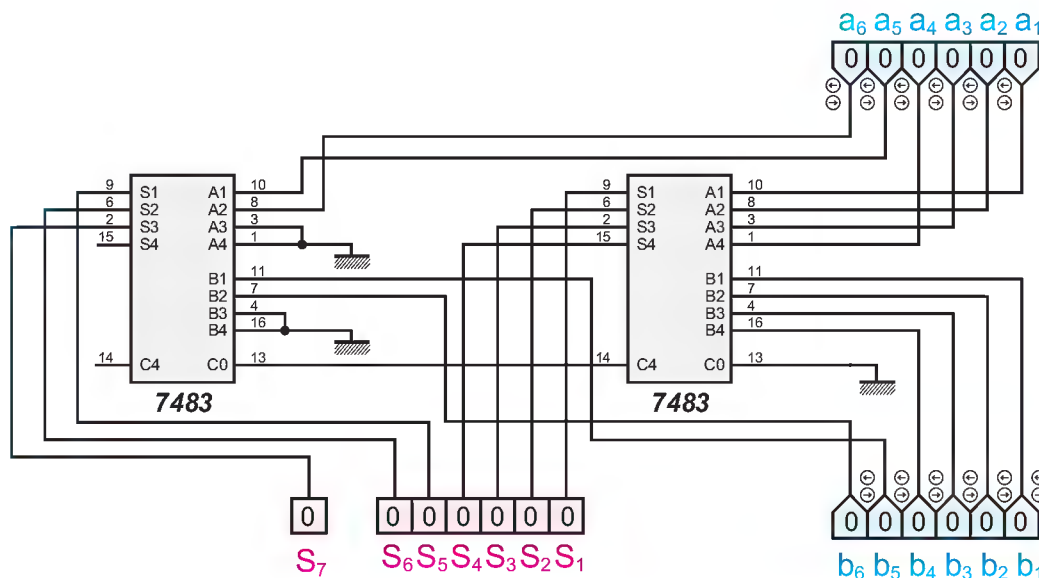


Fig. 7

4- L'additionneur BCD

a. Définition

Un additionneur **B.C.D** est un circuit électronique permettant d'additionner deux nombres codés en B.C.D

Rappelons que dans le code **B.C.D** chaque chiffre décimal (**digit**) est représenté par son équivalent binaire codé sur quatre bits (**quartet**).

b. Principe

En B.C.D l'opération d'addition est réalisée comme suit:

Exemple 1: Effectuons l'opération **73 + 16**

En décimal **73 + 16 = 89**

$$\begin{array}{r}
 \text{En BCD } (73)_{10} = (0111 \ 0011)_{\text{BCD}} \\
 + (16)_{10} = (0001 \ 0110)_{\text{BCD}} \\
 \hline
 = (1000 \ 1001)_{\text{BCD}}
 \end{array}$$

Le résultat est: **(1000 1001)_{BCD} = (89)₁₀**

Exemple 2:

Effectuons l'opération $7 + 8$

En décimal $7 + 8 = 15$

$$\begin{array}{r} \text{En BCD } (7)_{10} = (0111)_{\text{BCD}} \\ + (8)_{10} = (1000)_{\text{BCD}} \\ \hline = (1111)_{\text{BCD}} \end{array}$$

Le résultat est (1111) , cette valeur dépasse l'intervalle de définition d'un digit B.C.D (de 0 à 9). Pour remédier à ce problème on ajoute l'équivalent binaire de $(6)_{10} = (0110)_2$ au résultat trouvé.

retenu	1 1
résultat erroné	(0000 1111)
ajout de 6	(0000 0110)
résultat exact	$\begin{array}{r} \hline (0001 \ 0101) \\ \hline \end{array}$
	$\underbrace{\hspace{2em}}_1 \quad \underbrace{\hspace{2em}}_5$

Exemple 3: Effectuons maintenant l'opération $9 + 9$

En décimal $9 + 9 = 18$

$$\begin{array}{r} \text{En BCD } (9)_{10} = (1001)_{\text{BCD}} \\ + (9)_{10} = (1001)_{\text{BCD}} \\ \hline = 1(0010) \end{array}$$

Le résultat est $(0001\ 0010)_{\text{BCD}} \neq (18)_{10}$, pour remédier à ce problème on ajoute l'équivalent binaire de $(6)_{10} = (0110)_2$ au résultat trouvé.

retenue	1 1
résultat erroné	(0001 0010)
ajout de 6	(0000 0110)
résultat exact	$\begin{array}{r} \hline (0001 \ 1000) \\ \hline \end{array}$
	$\underbrace{\hspace{2em}}_1 \quad \underbrace{\hspace{2em}}_8$

CONCLUSION:

On doit ajouter $(6)_{10} = (0110)_2$ au résultat trouvé si:

- ✎ la somme des digits par colonne dépasse 9;
- ✎ la somme produit un report sur le quartet immédiatement à gauche.

LOGIQUE COMBINATOIRE

c. Réalisation industrielle

➤ Additionneur **B.C.D** à base d'additionneurs parallèles binaires

Un additionneur **B.C.D** doit être en mesure de répondre aux exigences suivantes:

- Additionner deux grandeurs codées en **B.C.D** selon les règles ordinaires de l'addition binaire.
- Vérifier si la somme est supérieure à **9**, dans ce cas ajouter le terme de correction (**0110**) c'est à dire **6** en décimal à cette somme et générer un report ramené sur le rang décimal immédiatement à gauche.

Exemple: Soient **A** et **B** deux nombres en BCD appliqués à un additionneur parallèle à 4 bits.

$$\begin{array}{r} a_3 a_2 a_1 a_0 \\ + b_3 b_2 b_1 b_0 \\ \hline S_4 S_3 S_2 S_1 S_0 \end{array}$$

S_4 est en réalité C_4 , le report produit par le rang du bit de poids le plus fort.

Soit E_c une sortie logique qui prendra **1** seulement quand la somme est supérieure à (01001). L'étude des différents cas nous permet d'affirmer que E_c est à **1** quand l'une ou l'autre des conditions suivantes est réalisée:

- Quand $C_4 = 1$ (somme supérieure à 15)
- Quand $S_3 = 1$ et que S_2 ou S_1 ou les deux sont à 1 (somme entre 10 et 15).

Formulée sous forme d'équation booléenne, cette constatation est traduite comme suit : $E_c = C_4 + S_3 \cdot (S_2 + S_1)$

Donc, pour réaliser un additionneur BCD nous avons besoin de:

➤ 2 additionneurs binaires parallèles.

➤ Un circuit de correction.

La figure suivante montre la structure d'un additionneur **B.C.D** à base d'additionneurs parallèles binaires.

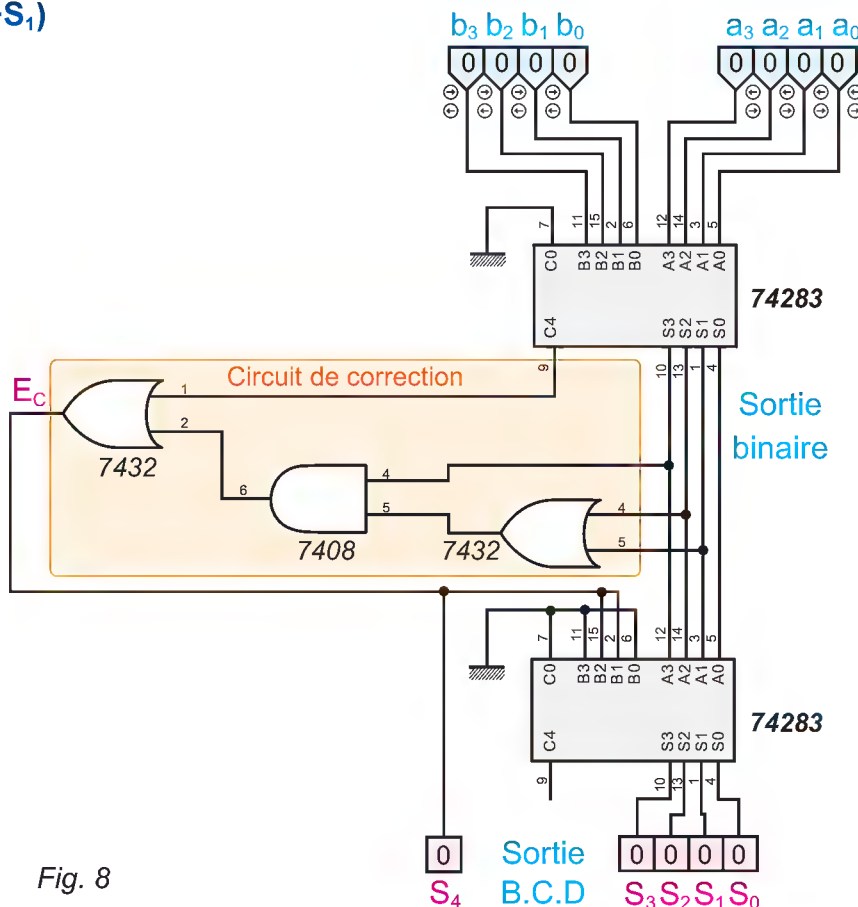


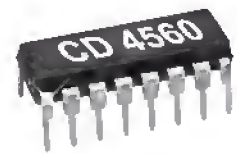
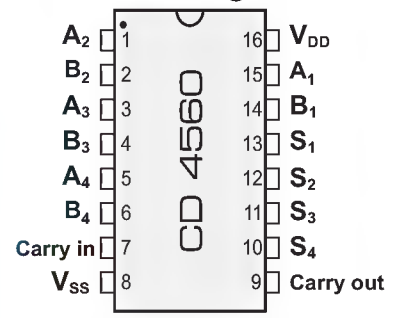
Fig. 8

➤ Additionneur B.C.D intégré

Ce type de circuit intègre dans le même boîtier le circuit d'addition et celui de la correction.

Exemple: le 4560

Brochage



Symbole

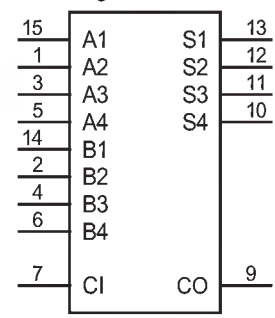


Fig. 9

d. Mise en œuvre du circuit 4560

Le schéma suivant montre les niveaux logiques des entrées et des sorties du C.I 4560 lors de l'addition de 8 avec 7.

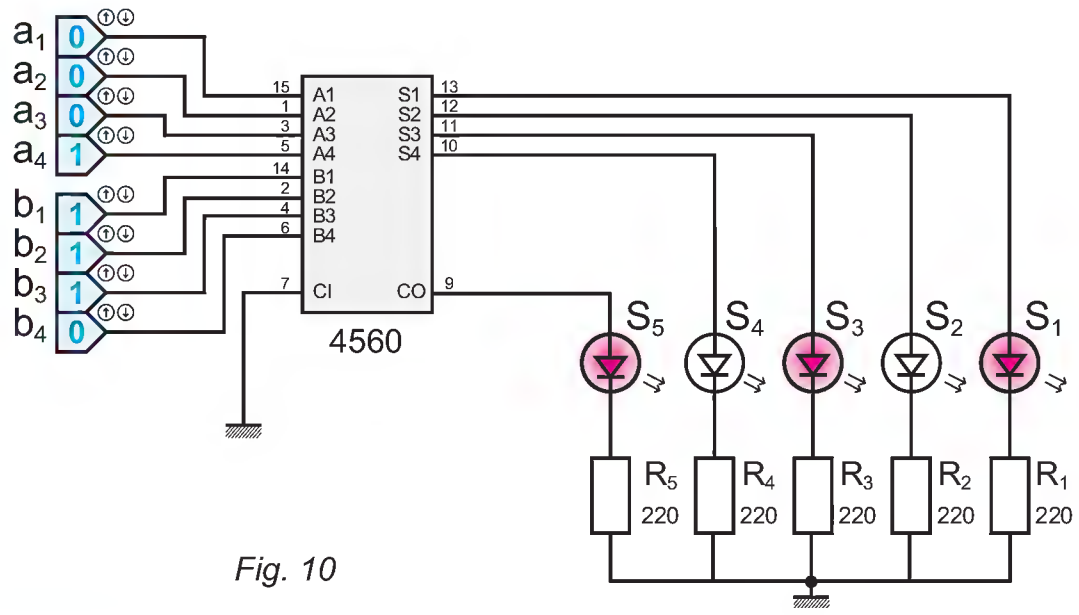


Fig. 10

e. Mise en cascade des circuits 4560

EXEMPLE:

Soit à additionner en B.C.D les nombres A et B, avec A=49 et B =35.

Schéma de câblage

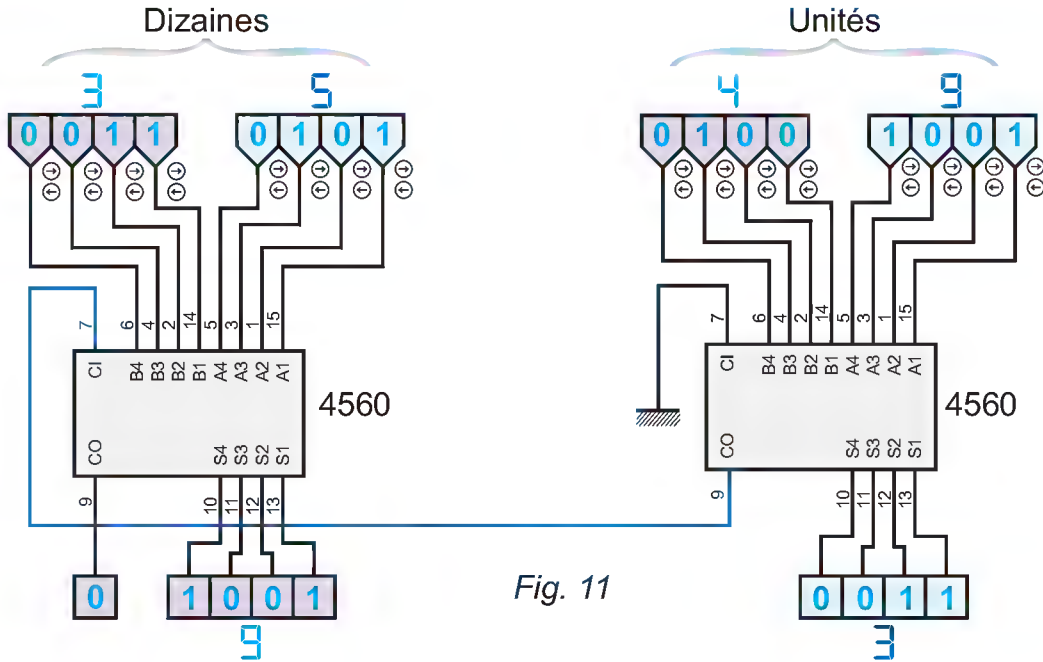


Fig. 11

5- L'addition en complément à 2

a. Rappel: le complément à 2 d'un nombre binaire s'obtient de deux façons:

- ☞ Complémenter bit par bit les éléments du nombre binaire (le complément à 1) et ajouter (1) au poids le plus faible.

EXEMPLE:

0	1	0	1	1	0	1	+ (45) ₁₀
Bit de signe		Grandeur exacte					
1	0	1	0	0	1	1	- (45) ₁₀
Bit de signe		Complément à 2					

- ☞ Pratiquement il est plus simple de procéder comme suit : en allant de droite à gauche, garder tous les chiffres depuis la droite jusqu'au premier 1 (y compris) puis inverser tous les suivants.

b. Intérêt: le complément à 2 d'un nombre binaire signé transforme un nombre positif en un nombre négatif est vice versa.

c. Principe de l'addition en complément à 2

- ☞ Cas de deux nombres positifs:

+9 →	0	1	0	0	1	Vraie grandeur
+5 →	0	0	1	0	1	Vraie grandeur
	0	1	1	1	0	Somme VG = + (14) ₁₀
		↑ Bit de signe				

✎ Cas d'un nombre positif et un nombre négatif plus petit

$+ 9 \rightarrow$	0	1	0	0	1	Vraie grandeur
$- 5 \rightarrow$	1	1	0	1	1	Complément à 2
Dépassement \rightarrow 1	0	0	1	0	0	Somme VG = + (4) ₁₀
	↑					Bit de signe

d. Avantage de la notation en complément à 2

Comme nous l'avons vu plus haut, avec cette notation on parvient à soustraire en effectuant en réalité une addition. Pratiquement avec les mêmes circuits on peut effectuer des additions et des soustractions.

N.B: La soustraction en **BCD** est effectuée comme l'addition. Dans certains cas il faut ajuster le résultat par **la soustraction d'un terme correctif**.

II- Le comparateur

1- Introduction

L'affichage du nombre de places libres évoqué plus haut dans ce cours ne peut avoir lieu et ne peut être fiable qu'après comparaison du nombre de places occupées à la capacité du parking.

Comme pour les additionneurs, cette opération peut être réalisée avec des cellules logiques de base, néanmoins dans cette section on va s'intéresser aux circuits spécialisés appelés comparateurs et pouvant remplir la dite fonction.

2- Principe

Il s'agit de comparer deux nombres binaires A et B pour indiquer en sortie si (**A > B ; A < B** ou **A = B**).

Avec **A = a_n a_{n-1} a₂ a₁ a₀** et **B = b_n b_{n-1} b₂ b₁ b₀**

La comparaison commence par les bits de poids le plus fort (**M.S.B**):

- ✎ Si **a_n > b_n** on peut conclure que **A > B**
- ✎ Si **a_n < b_n** on peut conclure que **A < B**
- ✎ Si **a_n = b_n** il faut poursuivre la comparaison de la même façon avec les autres bits de poids inférieurs de bit en bit jusqu'au bit de poids le plus faible (**L.S.B**) si nécessaire.

3- Structure

La comparaison se fait bit à bit, d'où la nécessité d'un comparateur élémentaire de deux nombres à un bit dont la structure est représentée par la figure 12:

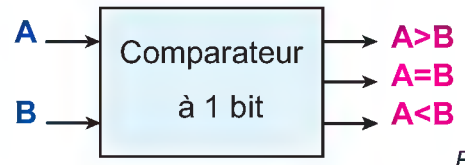


Fig. 12

LOGIQUE COMBINATOIRE

Pour comparer deux nombres codés sur plusieurs bits, il est nécessaire d'associer entre eux autant de comparateurs élémentaires qu'il y a de bits.

EXEMPLE:

Structure d'un comparateur de deux nombres **A** et **B** à 2 bits avec:

$$A = a_1 a_0 \quad \text{et} \quad B = b_1 b_0$$

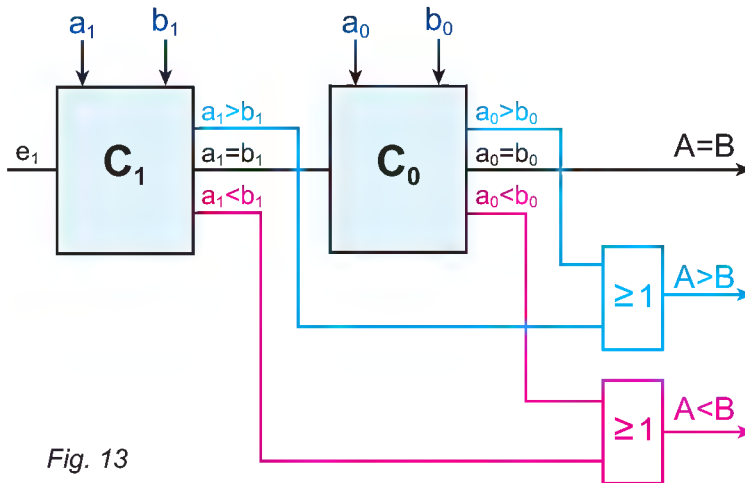


Fig. 13

4- Exemples de comparateurs en circuits intégrés

Le tableau ci-dessous illustre quelques exemples de comparateurs en circuits intégrés.

En technologie T.T.L		En technologie C.M.O.S
Comparateurs à 4 bits	Comparateurs à 8 bits	Comparateurs à 4 bits
7485, 74LS85	74LS682, 74LS688	4063, 4585

5- Brochage et symbolisation de la série 74XX85



Fig. 14

Ce circuit compare deux mots binaires **A** et **B** de **4 bits** chacun:

- ✎ **A3 A2 A1 A0**: les bits du mot binaire **A**.
- ✎ **B3 B2 B1 B0**: les bits du mot binaire **B**.
- ✎ **QA<B ; QA>B ; QA=B**: sorties (résultat de la comparaison).
- ✎ **A<B ; A>B ; A=B**: entrées de mise en cascade pour comparer des nombres de plus de 4 bits.

6- Mise en œuvre d'un comparateur de type 74xx85

La comparaison de deux nombres à 4 bits $A=a_3a_2a_1a_0$ et $B=b_3b_2b_1b_0$ nécessite un circuit 7485 dont les broches de mise en cascade (broche **2, 3 et 4**) doivent être connectées respectivement aux niveaux logiques «0», «1» et «0».

JUSTIFICATION:

D'après la fiche technique du circuit **7485** si $(a_3=b_3)$ et $(a_2=b_2)$ et $(a_1=b_1)$ et $(a_0=b_0)$ la sortie dépend de l'état logique des entrées de mise en cascade.

Autrement dit la sortie **QA=B** (broche 6) ne prend 1 que si l'entrée de mise en cascade **A=B** (broche 3) est égale à 1.

- ✎ Comparateur de **deux nombres binaires de 4 bits** à base de **C.I 74HC85**
 Schéma de câblage

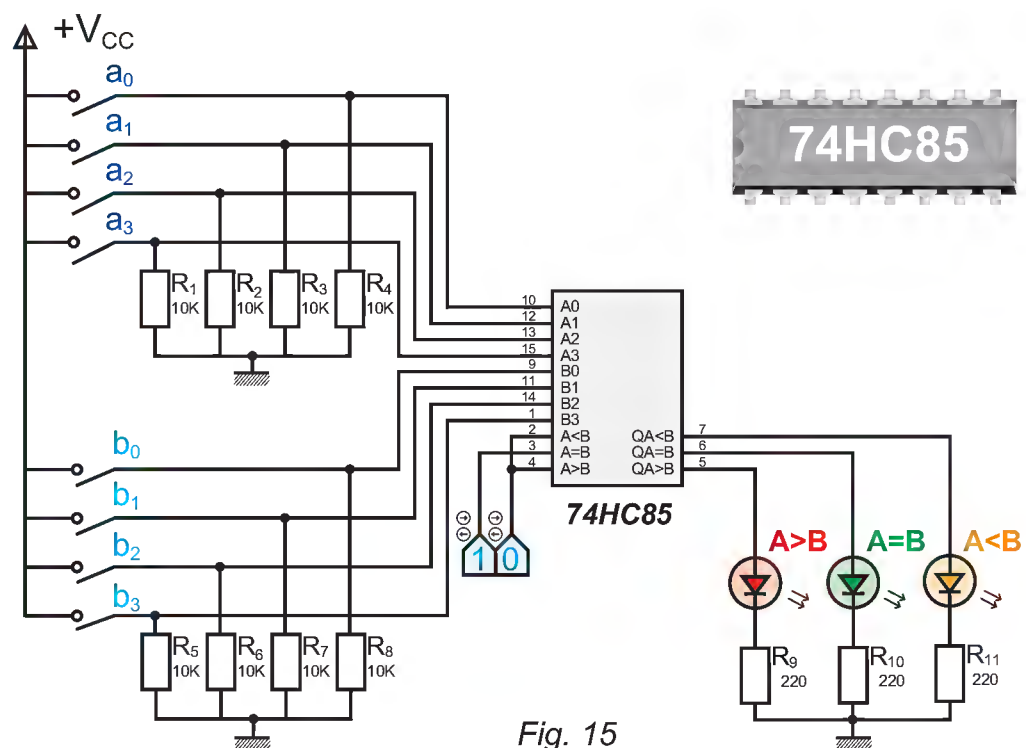


Fig. 15

7- Mise en cascade des comparateurs intégrés

Le montage en cascade de ces circuits constitue une façon d'étendre la comparaison à plus de 4 bits.

a. Principe

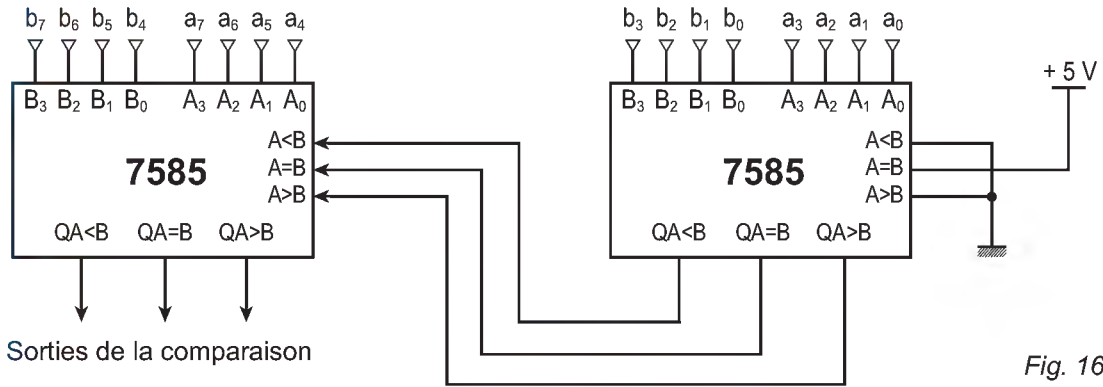


Fig. 16

b. Réalisation d'un comparateur de deux nombres à 8 bits

avec $A = a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0$ et $B = b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0$

- ✎ Nombre de circuits nécessaires de la série 7485: **2 circuits**.
- ✎ Les sorties du circuit gérant les 4 bits **du poids faible** (broche «5», «6» et «7») doivent être reliées respectivement aux entrées du comparateur gérant les bits **du poids fort** (broche «4», «3» et «2»).
- ✎ Les entrées de mise en cascade du comparateur gérant les bits **du poids faible** (broche «4», «3» et «2») sont reliées respectivement aux niveaux logiques «0», «1» et «0».

Schéma de câblage

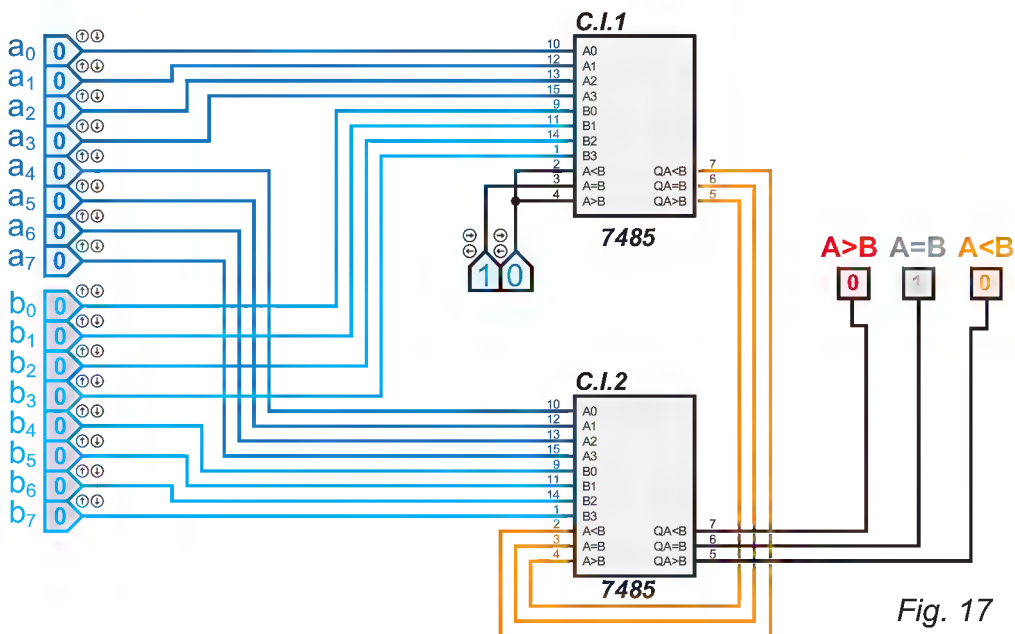


Fig. 17

III- Multiplexeur/démultiplexeur

1- Introduction:

Le cahier des charges du parking préconise, entre autre, l'affichage de diverses informations pour guider l'automobiliste durant l'exploration des lieux.

Vue la diversité de ces informations, ces dernières nécessitent un support d'affichage par information et donc un câblage dédié pour chacun d'eux et par conséquent des frais de câblage, de maintenance et d'entretien plus élevés.

Pour réduire ces frais et pour gérer judicieusement ces équipements on fait appel au **MULTIPLEXAGE**.

Cette fonction est généralement assurée par des circuits spécialisés appelés multiplexeurs/démultiplexeurs.

2- Le multiplexeur

Le multiplexeur désigné généralement par (**MUX**) est un circuit logique ayant plusieurs entrées de données et une seule sortie communiquant ces données.

L'aiguillage de l'entrée de données désirée vers la seule sortie est conditionné par la combinaison des entrées de sélection (appelées parfois entrées D'ADRESSES).

Le multiplexage consiste donc à envoyer sur une même ligne de transmission des informations provenant de sources différentes.

a. Principe



Fig. 18

Dans cet exemple, le multiplexeur a **4** entrées logiques E_0 , E_1 , E_2 et E_3 , et une sortie logique **S**. En fonction de la sélection, une des 4 entrées se retrouvera à la sortie du multiplexeur:

- ✎ si la sélection est placée en position **0**, la sortie prend l'état logique de l'entrée E_0 ;
- ✎ si la sélection est placée en position **1**, la sortie prend l'état logique de l'entrée E_1 ;
- ✎ si la sélection est placée en position **2**, la sortie prend l'état logique de l'entrée E_2 ;
- ✎ si la sélection est placée en position **3**, la sortie prend l'état logique de l'entrée E_3 .

b. Etude des multiplexeurs

Exemple: multiplexeur intégré à 4 voies (74153)

Le circuit intégré **74153** contient deux multiplexeurs à **4 voies** à entrées de sélection **A** et **B** communes. Chaque multiplexeur dispose d'une entrée de validation **E** ou **G** (**STROBE**). Celle-ci, portée à l'état **1**, force la sortie du multiplexeur correspondant à l'état **0** indépendamment de l'état des autres entrées.

Le circuit **74153** compte:

- ✎ **4** entrées de données pour le multiplexeur 1 (1C0, 1C1, 1C2, 1C3);
- ✎ **4** entrées de données pour le multiplexeur 2 (2C0, 2C1, 2C2, 2C3);
- ✎ **2** entrées de sélection **A** et **B**;

LOGIQUE COMBINATOIRE

- 1 entrée de validation $1\bar{G}$ pour le multiplexeur 1;
- 1 entrée de validation $2\bar{G}$ pour le multiplexeur 2;
- 2 sorties: 1Y pour le multiplexeur 1 et 2Y pour le multiplexeur 2.



FONCTIONNEMENT:

- Si le multiplexeur n'est pas validé (entrée **EN** ou **G** à **1**), la sortie **Y** est à **0** quel que soit l'état des entrées de données et celui des entrées de sélection.
- Si le circuit est validé (entrée de validation **EN** ou **G** à **0**), on retrouve sur la sortie du multiplexeur l'état logique de l'entrée de donnée qui est sélectionnée avec les entrées de sélection **A** et **B** selon la table de fonctionnement précédente.

c. Multiplexeurs usuels

Le tableau ci-dessous illustre quelques exemples de multiplexeurs en circuits intégrés.

- En technologie **TTL**

Type	Fonction
74150	Multiplexeur 16 voies vers 1
74151	Multiplexeur 8 voies vers 1 (2 sorties complémentaires)
74152	Multiplexeur 8 voies vers 1
74153	Double multiplexeur 4 voies vers 1
74157	Quadruple multiplexeur 2 voies vers 1

- En technologie **CMOS**

Type	Fonction
4019	Quadruple multiplexeur 2 voies vers 1
4512	Multiplexeur 8 voies vers 1
4519	Quadruple multiplexeur 2 voies vers 1
4539	Double multiplexeur 4 voies vers 1

d. Utilisation des multiplexeurs

Les applications d'un multiplexeur en électronique sont principalement:

- ✎ la génération de fonctions logiques (cette application sera traitée plus en détails dans la section suivante);
- ✎ la conversion parallèle/série d'informations: dans de nombreux systèmes numériques, le traitement des données binaires se fait en parallèle (c'est-à-dire simultanément). Cependant, quand on doit transférer ces données sur de longues distances (cas de la téléphonie par exemple), il n'est pas souhaitable de le faire en parallèle parce que cela exige un grand nombre de lignes de transmission. C'est la raison pour laquelle les informations ou les données binaires qui sont exprimées sous forme parallèle sont souvent converties sous forme série avant d'être transmises à l'endroit de destination;
- ✎ l'affichage multiplexé sur des afficheurs 7 segments: cette technique permet généralement la limitation de la consommation de ce type d'afficheurs et la réduction du nombre de fils de connexions;
- ✎ D'autres applications ont vu le jour avec l'apparition des multiplexeurs tels que (la concentration de données et leur transmission, le décodage des claviers matriciels et d'autres...).

e. Le multiplexeur en tant que générateur de fonctions logiques

✎ Introduction: Il est possible d'exploiter les multiplexeurs afin de matérialiser directement des fonctions logiques à partir d'une table de vérité donnée sans pour autant devoir passer par le processus de simplification.

✎ Principe:

Pour mettre en œuvre un multiplexeur dans de telle configuration on procède comme suit :

- Les variables d'entrée de la dite fonction doivent être reliées aux entrées de sélection.
- Les entrées de données du multiplexeur doivent être raccordées en permanence à un niveau logique (0) ou (1) en fonction des données de la table de vérité.

EXEMPLE

Soit la table de vérité suivante:

c	b	a	S
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

LOGIQUE COMBINATOIRE

CONSTATATIONS

➤ la table de vérité comporte 3 variables d'entrée. Le multiplexeur à utiliser doit comporter alors au moins 3 entrées de sélection donc à 8 entrées tel que le **74151**.

➤ D'après la table de vérité, **S** prend (0) quand **cba = 000, 011, 100, 101 et 110**.

Dans les autres cas **S** prendra (1).

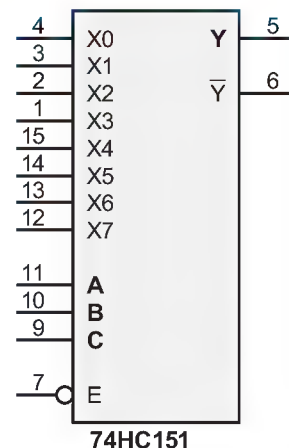


Fig. 20

SOLUTION

➤ Les variables d'entrée a, b et c doivent être raccordées respectivement aux entrées de sélection A, B et C.

➤ Relier en permanence les entrées de données (X_0 à X_7) à un niveau logique bas ou à un niveau logique haut en respectant à la fois les données de la table de vérité et celles de la table de fonctionnement du multiplexeur utilisé.

La confrontation des différentes combinaisons de la table de vérité à celles de la table de fonctionnement du multiplexeur exige la mise en permanence au niveau logique bas (0) des entrées de données suivantes (X_0, X_3, X_4, X_5 et X_6) par contre les autres entrées (X_1, X_2 et X_7) doivent être reliées en permanence au niveau logique haut (1).

D'où le montage suivant:

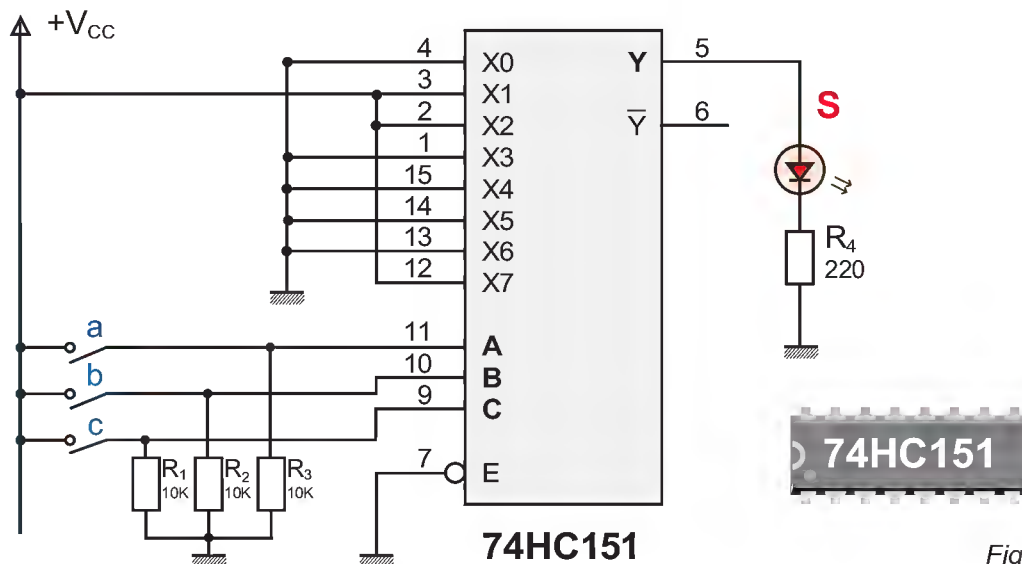


Fig. 21

3- Le démultiplexeur

Le démultiplexeur (**DMUX**) est un circuit logique ayant une entrée de données et plusieurs sorties.

L'aiguillage de la donnée vers l'une des sorties est conditionné par la combinaison des entrées de sélection (appelées parfois entrées D'ADRESSES).

Le démultiplexage consiste donc à répartir une information série provenant d'une seule et unique entrée vers des sorties différentes.

a. Principe

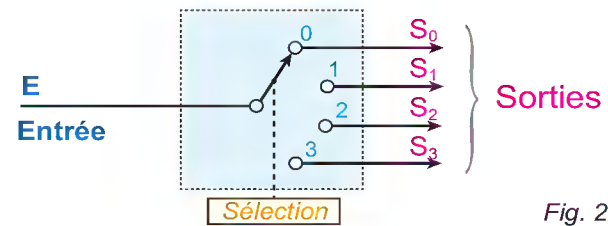


Fig. 22

Dans cet exemple, le démultiplexeur a une entrée logique **E**, et **4** sorties logiques **S₀**, **S₁**, **S₂** et **S₃**. En fonction de la sélection, les informations présentes sur l'entrée du démultiplexeur se retrouvent sur l'une des sorties :

- ✎ si la sélection est placée en position **0**, l'état logique de l'entrée **E** se retrouve sur la sortie **S₀**;
- ✎ si la sélection est placée en position **1**, l'état logique de l'entrée **E** se retrouve sur la sortie **S₁**;
- ✎ si la sélection est placée en position **2**, l'état logique de l'entrée **E** se retrouve sur la sortie **S₂**;
- ✎ si la sélection est placée en position **3**, l'état logique de l'entrée **E** se retrouve sur la sortie **S₃**.

b. Etude des démultiplexeurs

Exemple: démultiplexeur intégré (**74LS139**)

Brochage

74LS139

Table de fonctionnement

Entrées			Sortie			
E	B	A	Y0	Y1	Y2	Y3
1	x	x	1	1	1	1
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0

$E \equiv EN \equiv \bar{E}$

Symbole

74LS139

Fig. 23

Le circuit **74LS139** comporte **2** démultiplexeurs **1** voie vers **4**.

Chaque démultiplexeur est composé de :

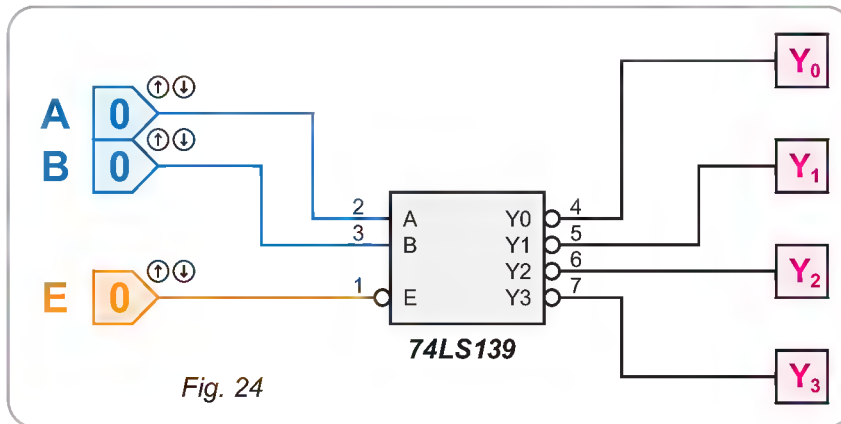
- ✎ deux entrées de sélection **A** et **B**;
- ✎ une entrée de donnée **E**;
- ✎ quatre sorties **Y₀**, **Y₁**, **Y₂** et **Y₃**.

FONCTIONNEMENT

L'entrée de validation **E** et les sorties **Y₀**, **Y₁**, **Y₂** et **Y₃** sont actives à l'état **0**.

- Si l'entrée **E** est à l'état **1**, le circuit 74LS139 n'est pas validé et toutes ses sorties passent à l'état **1**.
- Si l'entrée **E** est à l'état **0**, le circuit est validé et la sortie sélectionnée par les entrées **A** et **B** du démultiplexeur passe à l'état **0**.

Nous pouvons dire que la donnée «**0**» présente sur l'entrée **E** est transférée vers la sortie sélectionnée.



- Pour **AB = 00** l'entrée de donnée **E** est dirigée vers la sortie **Y₀** ($Y_0 = 0$).
- Pour **AB = 01** l'entrée de donnée **E** est dirigée vers la sortie **Y₁** ($Y_1 = 0$).
- Pour **AB = 10** l'entrée de donnée **E** est dirigée vers la sortie **Y₂** ($Y_2 = 0$).
- Pour **AB = 11** l'entrée de donnée **E** est dirigée vers la sortie **Y₃** ($Y_3 = 0$).

On remarque que le nombre binaire formé par l'état des entrées de sélection **B** et **A** donne l'**indice décimal** de la sortie concernée.

Par exemple, lorsque **AB = 10** (soit **2 en décimal**), la sortie concernée est **Y₂**.

c. Démultiplexeurs usuels

Le tableau ci-dessous illustre quelques exemples de démultiplexeurs en circuits intégrés.

- En technologie **TTL**

Type	Fonction
74139	Double démultiplexeur 2 voies vers 4
74238	Démultiplexeur 3 voies vers 8
74154	Démultiplexeur 4 voies vers 16

- En technologie **CMOS**

Type	Fonction
4555	Double démultiplexeur 2 voies vers 4
4514	Démultiplexeur 1 vers 16
4515	Démultiplexeur 4 voies vers 16

d. Extension d'un démultiplexeur

Il est possible de réaliser l'extension d'un démultiplexeur par une structure pyramidale.

Exemple: Réalisation d'un démultiplexeur **4 voies vers 16**

Les quatre sorties d'un démultiplexeur **2 voies vers 4** activent **quatre** démultiplexeurs **2 voies vers 4**. Nous obtenons ainsi un démultiplexeur **4 voies vers 16**.

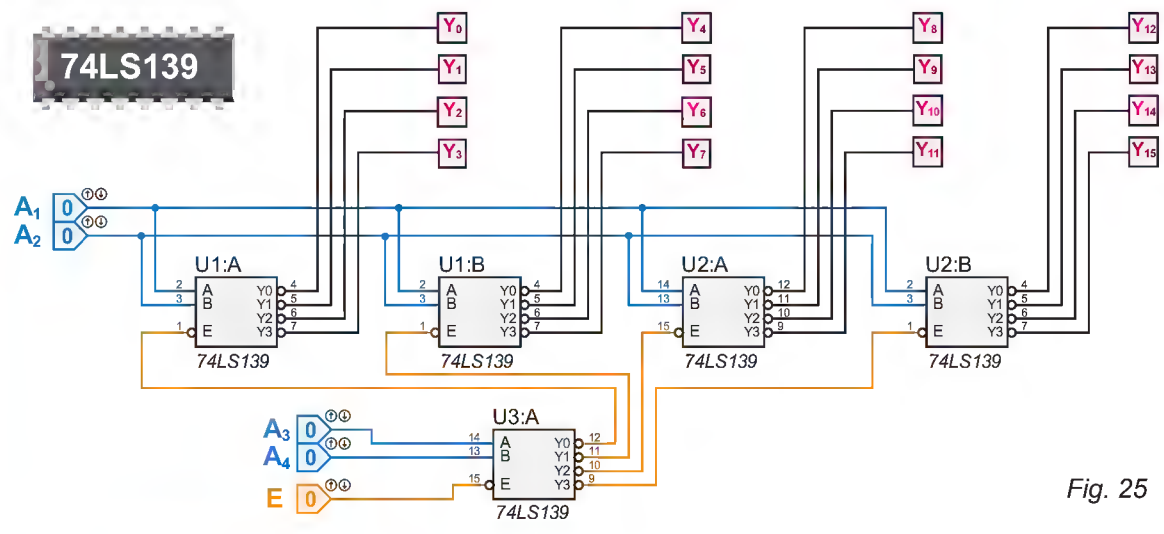


Fig. 25

Le tableau suivant résume le fonctionnement du précédent montage:

Entrées				Sortie active
A ₄	A ₃	A ₂	A ₁	
0	0	0	0	Y ₀
0	0	0	1	Y ₁
0	0	1	0	Y ₂
0	0	1	1	Y ₃
0	1	0	0	Y ₄
0	1	0	1	Y ₅
0	1	1	0	Y ₆
0	1	1	1	Y ₇
1	0	0	0	Y ₈
1	0	0	1	Y ₉
1	0	1	0	Y ₁₀
1	0	1	1	Y ₁₁
1	1	0	0	Y ₁₂
1	1	0	1	Y ₁₃
1	1	1	0	Y ₁₄
1	1	1	1	Y ₁₅

Remarque: le nombre binaire formé par l'état des entrées de sélection (A₁ à A₄) donne l'indice décimal de la sortie active.

4- L'affichage multiplexé sur des afficheurs 7 segments

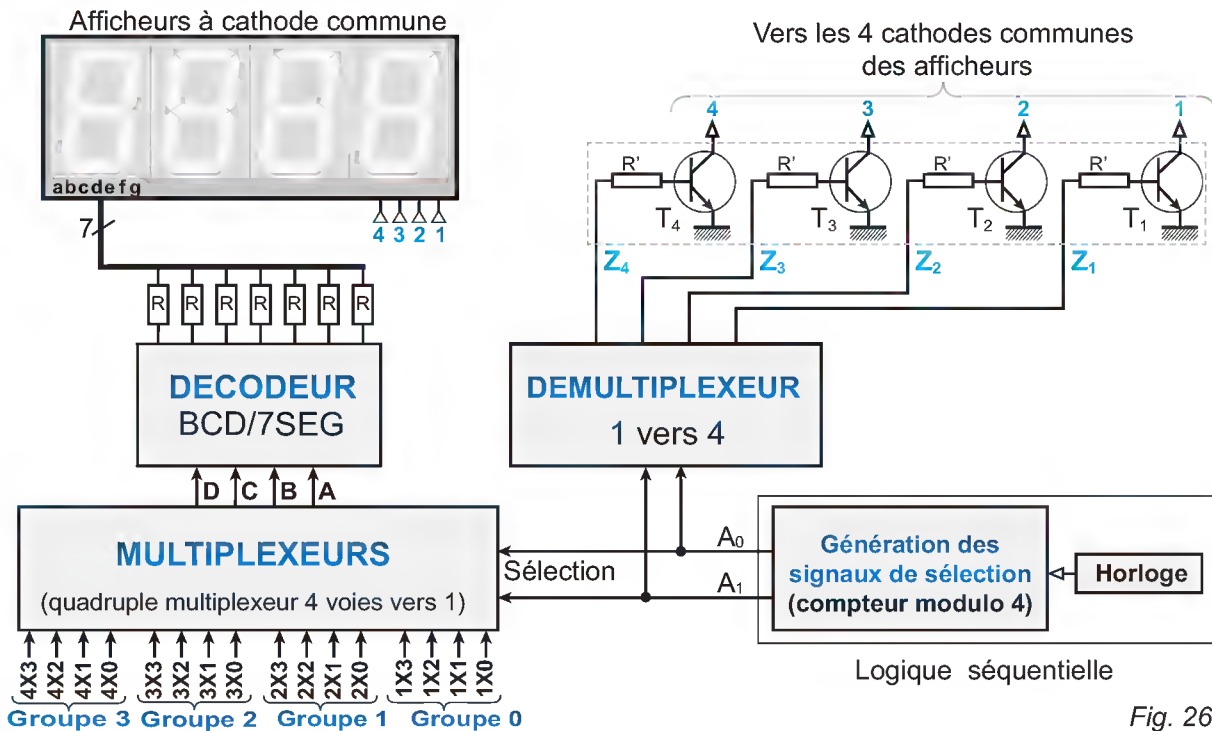
Pour lire le résultat d'un calcul sur des afficheurs sept segments, il nous faut autant d'afficheurs et autant de décodeurs que de chiffres significatifs, sans compter le grand nombre de résistors à utiliser, ce qui augmente considérablement le nombre de liaisons et le coût du montage.

A partir de 3 ou 4 chiffres, il est préférable d'adopter un affichage multiplexé consistant à n'utiliser qu'un seul décodeur et à commuter successivement et très rapidement, les afficheurs en question.

Cette solution est adoptée pour deux raisons:

- ✎ les diodes électroluminescentes ont un temps de réponse très court (elles s'allument et s'éteignent très rapidement);
- ✎ la persistance rétinienne de l'oeil de l'être humain est d'environ 1/25^{ème} de seconde. L'observateur ne s'aperçoit d'aucun clignotement.

Par exemple, l'affichage d'une radio réveil numérique, d'une machine à laver et même des nombres N_v et N_{pl} du système «gestion d'un parking» sont des exemples d'application de l'affichage multiplexé.



SCHEMA DE PRINCIPE

Ce dispositif utilise un démultiplexeur 1 vers 4 et un quadruple multiplexeur 4 voies vers 1. Le démultiplexeur sélectionne séquentiellement chacun des afficheurs (en mettant à la masse la cathode commune de l'afficheur) et le multiplexeur distribue sur les 4 afficheurs en même temps le digit de l'afficheur sélectionné.

On remarque qu'avec cette technique un seul décodeur « BCD/7 segments » suffit pour afficher un nombre en BCD de 4 chiffres (les données de sortie d'un compteur modulo 10 000 -0 à 9999- par exemple).

C. RÉSUMÉ

- ✂ Parmi les circuits combinatoires on distingue les circuits arithmétiques tels que les additionneurs et les soustracteurs et les circuits logiques tels que les comparateurs, les multiplexeurs les codeurs, etc.
- ✂ Dans la gamme des additionneurs on cite les additionneurs parallèles et ceux B.C.D.
- ✂ La notation en complément à 2 permet de ramener une soustraction à une simple addition.
- ✂ La mise en cascade de comparateurs logiques permet d'étendre la plage de comparaison (nombre de bits).
- ✂ L'utilisation des multiplexeurs est très répandue en électronique numérique:
Exemples: alimentation multiplexée des afficheurs sept segments, matérialisation des fonctions logiques, génération de formes d'ondes.
- ✂ Un multiplexeur joue le rôle d'un commutateur à plusieurs positions qui aiguille vers la sortie les informations de n'importe quelle entrée.
- ✂ Les entrées d'adresses permettent de connaître à chaque instant l'état d'une entrée de données.
- ✂ Par une utilisation cyclique des entrées d'adresses, les données des entrées, qui sont en parallèle à l'entrée du multiplexeur, se retrouvent en série à la sortie.
- ✂ Un démultiplexeur joue le rôle d'un commutateur à plusieurs positions qui oriente les informations d'entrée vers la sortie choisie.
- ✂ Les entrées d'adresses permettent de connaître à chaque instant l'état d'une sortie.
- ✂ Par une utilisation cyclique des entrées d'adresses, les données des entrées, qui sont en série à l'entrée du démultiplexeur, se retrouvent en parallèle à la sortie

D. ÉVALUATION

I- Contrôle des connaissances

- 1- Quelle est la différence entre un additionneur binaire parallèle et un additionneur B.C.D?
- 2- Quelle est la différence entre le C.I 7483 et le C.I 74283 ?
- 3- Quel est le rôle de C_0 dans les circuits d'addition ?
- 4- Quel est le rôle de V_{DD} et V_{SS} dans les circuits d'addition?
- 5- Dans quels cas doit-on ajouter 6 au résultat d'addition en B.C.D?
- 6- Représenter 135 et 265 en B.C.D, puis additionnez-les.
Vérifier en convertissant le résultat en décimal.
- 7- Quel est le rôle des broches «2», «3» et «4» du circuit 7485 ?
- 8- Quelle est la différence entre le C.I 74LS85 et le C.I 74LS682 ?
- 9- Préciser l'état logique des sorties du circuit 7485 si les entrées de mise en cascade $A > B$; $A = B$ et $A < B$ sont respectivement à l'état «1», «0» et «1».
- 10- Pour transmettre une entrée de donnée parmi quatre vers la sortie, un multiplexeur nécessite:
 - a. une entrée d'adresse;
 - b. deux entrées d'adresse;
 - c. trois entrées d'adresse.
- 11- Un multiplexeur est un circuit combinatoire permettant de:
 - a. transmettre un signal d'entrée parmi plusieurs vers une sortie;
 - b. distribuer un signal d'entrée vers une parmi plusieurs sorties.
- 12- Si la broche N°1 du C.I 74153 est à l'état haut, la broche N°7 sera à un niveau:
 - a. 0 quel que soit l'état des entrées de sélection;
 - b. 1 quel que soit l'état des entrées de sélection.
- 13- Si les broches 1,2,3 et 4 du C.I 74LS139 sont reliées à la masse, la broche N°4 sera portée à un niveau:
 - a. égale à 0;
 - b. égale à 1.

II- Exercices résolus

EXERCICE N°1

On se propose de réaliser un additionneur à base du circuit intégré 74283 capable d'additionner 137 et 106.

- 1- Déterminer le nombre de circuits nécessaires de référence 74283 assurant cette addition.
- 2- Tracer le schéma de montage et indiquer les niveaux logiques des entrées et des sorties.

EXERCICE N°2

A l'aide du circuit **4008** et de portes logiques, réaliser un soustracteur capable d'effectuer la différence (A-B) entre deux nombres binaires positifs exprimés chacun sur 4 bits avec **A** toujours supérieur à **B** (hypothèse simplificatrice).

Rappel: $A - B = A + [-B] = A + [\bar{B} + 1]$.

EXERCICE N°3

On se propose de réaliser un comparateur binaire permettant d'enclencher une alarme après le remplissage d'un carton par 10 pièces. L'opération de comptage est effectuée par un compteur binaire (ne faisant pas partie de l'étude).

- 1- Indiquer le type et le nombre de circuits à utiliser.
- 2- Proposer un schéma de câblage de l'ensemble.

EXERCICE N°4

Afin de diminuer le nombre de circuits intégrés utilisés, on peut mettre en œuvre des multiplexeurs pour réaliser des fonctions logiques.

Proposer un schéma réalisant la fonction **OU Exclusif** à l'aide d'un multiplexeur 4 vers 1 de référence 74153.

EXERCICE N°5

On veut réaliser un circuit logique à trois entrées (**a, b, c**) et dont la sortie **S** ne prend la valeur «1» que si une seule entrée parmi les trois est à «1» .

- a. Dresser la table de vérité.
- b. Ecrire l'expression de **S** en fonction de **a, b** et **c**.
- c. Proposer une solution en utilisant le circuit **74151**.

III- Exercices à résoudre

EXERCICE N°1

- 1- Rappeler la table vérité de l'additionneur 1 bit ainsi que le circuit correspondant.
- 2- Réaliser un additionneur 4 bits avec retenues d'entrée et de sortie.

EXERCICE N°2

- 1- En utilisant le datasheet relatif au circuit 74283, relever ses principales caractéristiques
- 2- Préciser le nombre de circuits nécessaires pour additionner 360 et 12.

EXERCICE N°3

Montrer comment assembler des additionneurs parallèles à 4 bits (74AC283) pour faire la somme de deux nombres de 7 bits chacun.

EXERCICE N°4

Un comparateur **n** bits est un circuit servant à comparer 2 mots $A_{n-1}A_1 \dots A_0$ et $B_{n-1}B_1 \dots B_0$ de **n** bits chacun. La sortie vaut **1** si les mots sont identiques et **0** sinon.

- 1- Proposer un schéma à base de portes logiques pour un comparateur 1 bit.
- 2- En déduire le circuit du comparateur 4 bits.

EXERCICE N°5

A l'aide d'un comparateur 4 bits intégré 7485 et de portes logiques, on veut réaliser la comparaison de deux nombres de 5 bits. Donner le schéma correspondant à ce système.

EXERCICE N°6

Montrer comment deux circuits **74157** et un circuit **74151** peuvent être réunis pour former un **MUX 16**→**1** sans devoir utiliser d'autres portes logiques.

EXERCICE N°7

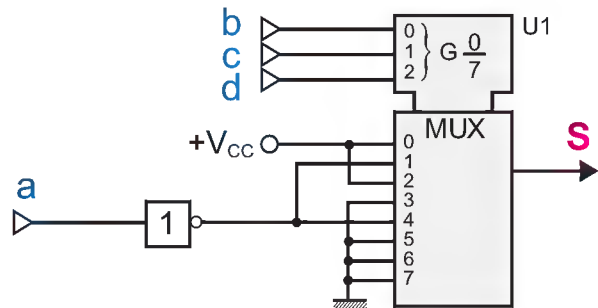
Réaliser un OU Exclusif à l'aide d'un MUX 4 vers 1.

EXERCICE N° 8

Montrer comment peut-on utiliser un 74151 (MUX 8→1) pour générer la fonction logique $S = a.b + b.c + a.c$

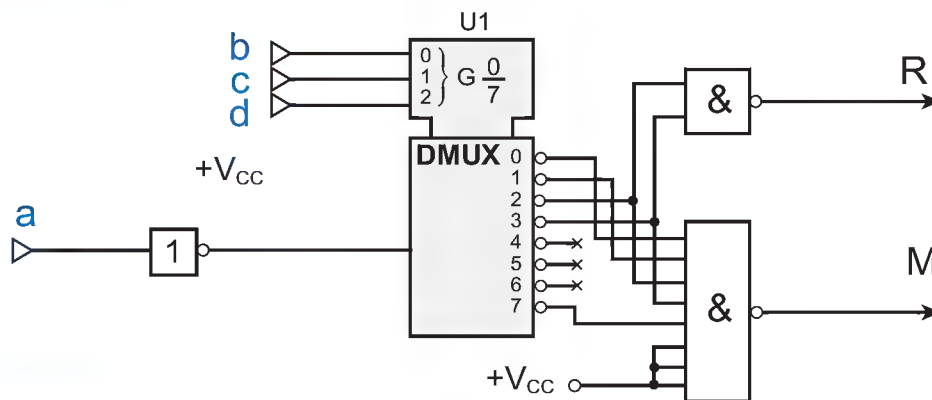
EXERCICE N°9

Donner l'équation simplifiée de la configuration suivante:



EXERCICE N°10

Donner les équations simplifiées de la configuration suivante:



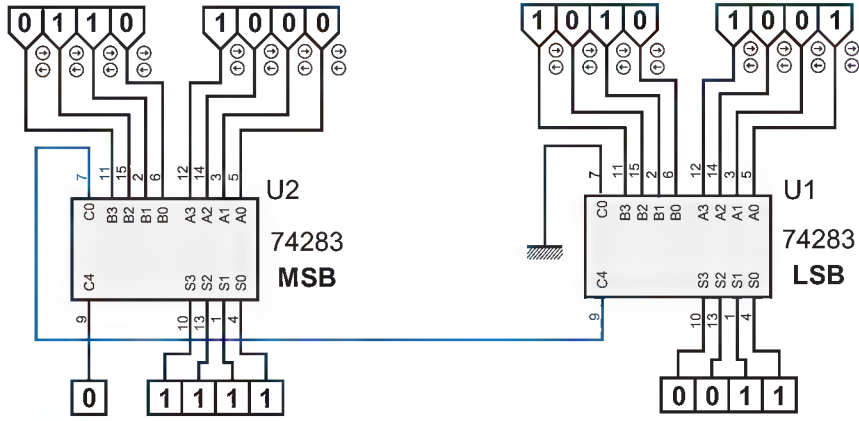
EXERCICE N°11

On veut réaliser un montage permettant d'effectuer la comparaison $A < B$, $A = B$, $A > B$ de 2 nombres de **2 bits** $A (a_1 a_0)$ et $B (b_1 b_0)$. Etudier le circuit et donner un schéma à base de multiplexeurs.

IV- Correction des exercices

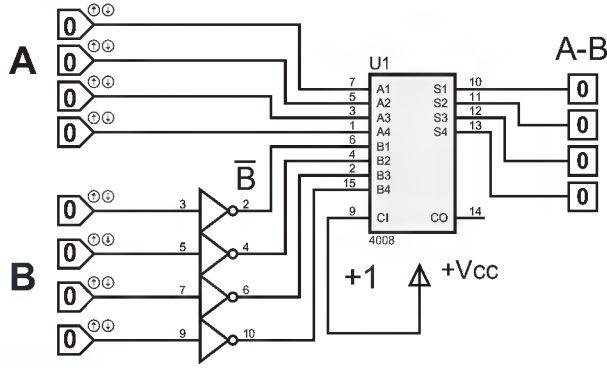
EXERCICE N°1

- 1- Nombre de circuits 74283:
Il nous faut deux circuits
- 2- Schéma de montage



EXERCICE N°2

Schéma de montage d'un soustracteur réalisant l'opération A-B, avec A>B.



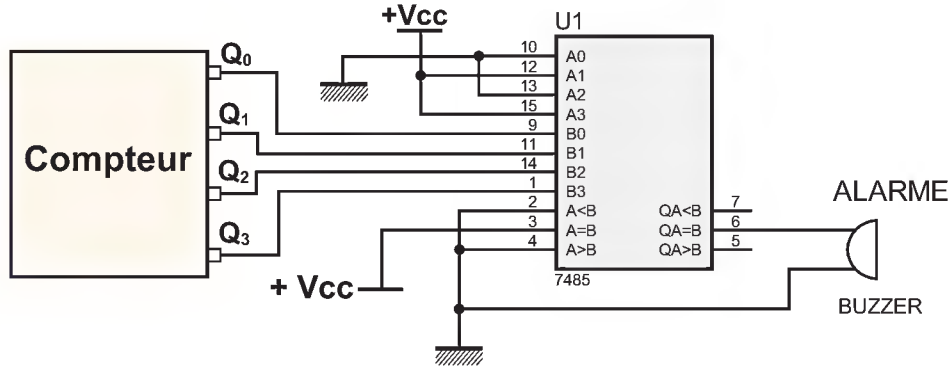
A > B

$$A - B = A + (-B)$$

$$= A + (\bar{B} + 1)$$

EXERCICE N°3

- 1- Nombre de circuit: Un circuit 74153
- 2- Schéma de montage



LOGIQUE COMBINATOIRE

EXERCICE N°4

$$S = a \oplus b = a \cdot \bar{b} + \bar{a} \cdot b$$

D'après la table de fonctionnement du circuit 74153:

$$S = \bar{A} \cdot \bar{B} \cdot 1X0 + A \cdot \bar{B} \cdot 1X1 + \bar{A} \cdot B \cdot 1X2 + A \cdot B \cdot 1X3$$

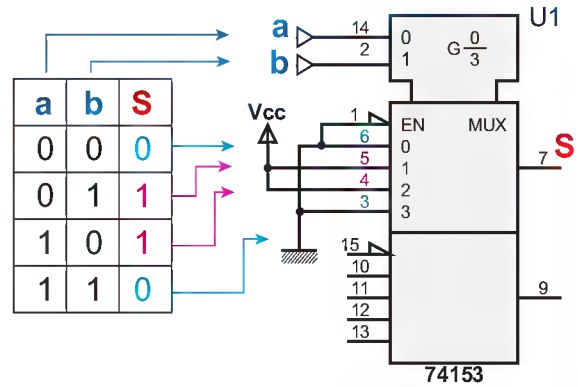
Législation avec léquation du OU exclusif

$$S = a \oplus b = a \cdot \bar{b} + \bar{a} \cdot b$$

$$\begin{cases} S = \bar{A} \cdot \bar{B} \cdot 1X0 + A \cdot \bar{B} \cdot 1X1 + \bar{A} \cdot B \cdot 1X2 + A \cdot B \cdot 1X3 \\ S = a \oplus b = a \cdot \bar{b} + \bar{a} \cdot b \end{cases}$$

donne

$$\begin{cases} 1X0 = 0 \\ 1X1 = 1 \\ 1X2 = 1 \\ 1X3 = 0 \end{cases} \quad \begin{cases} A = a \\ B = b \end{cases}$$



EXERCICE N°5

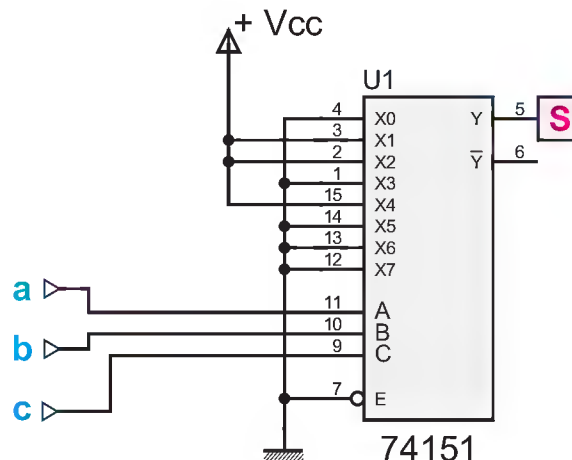
a. Table de vérité

c	b	a	S
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

b- Expression de la sortie

$$S = a \cdot \bar{b} \cdot \bar{c} + \bar{a} \cdot b \cdot \bar{c} + \bar{a} \cdot \bar{b} \cdot c$$

c- Schéma de câblage



L'UNITE ARITHMETIQUE ET LOGIQUE

A. MISE EN SITUATION

En électronique numérique existe des circuits intégrés simples pouvant réaliser:

- ✂ des fonctions logiques : **ET**(7408), **OU** (7432), **NON** (7404) ... ;
- ✂ des fonctions arithmétiques: **additionneur binaire** (74283), **B.C.D** (4560)...;
- ✂ **la comparaison** (7485:comparateur de deux nombres à 4 bits);
- ✂ **le multiplexage** (74157 ...);
- ✂ **le démultiplexage** (74153...).

Pour répondre aux exigences de certaines applications, on est parfois amené à associer ces dits circuits pour répondre à certaines fonctions.

Or par souci d'intégration et dans le but de réduire certaines dépenses on doit mettre en œuvre d'autres solutions techniques.

Problématique:

- ✂ Est-il possible d'intégrer toutes ou partie de ces fonctions dans un seul circuit ?
- ✂ Qu'appelle-t-on ce type de circuit ?
- ✂ Comment le mettre en œuvre ?

L'UNITÉ ARITHMÉTIQUE ET LOGIQUE

I- Introduction

Pour diminuer les coûts de fabrication, de développement et de maintenance, les fabricants des circuits intégrés ont inventé un circuit logique appelé: **unité arithmétique et logique (U.A.L)** regroupant plusieurs circuits dans un seul boîtier (Fig.1).

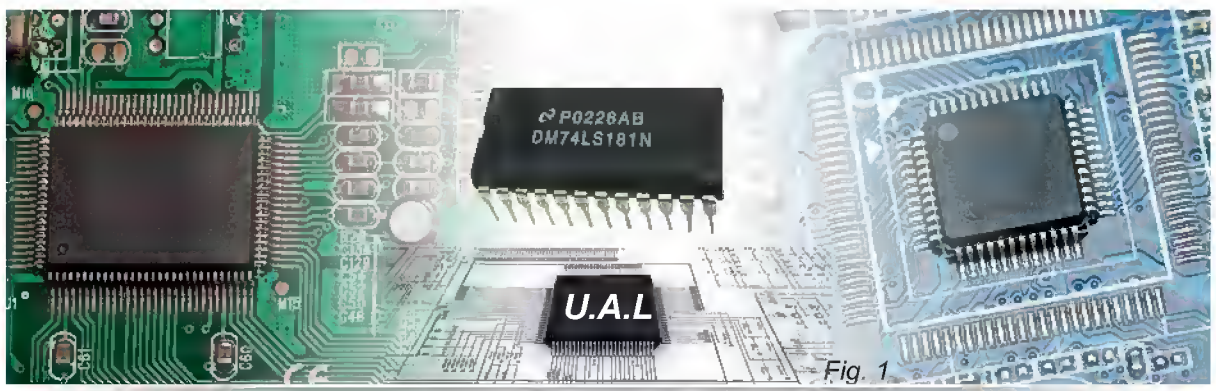


Fig. 1

AUTOMATIQUE

II- Définition

L'unité Arithmétique et Logique (**U.A.L** ou **A.L.U**: **A**rithmetic and **L**ogic **U**nit) est un circuit combinatoire capable d'effectuer une grande variété d'opérations logiques et arithmétiques.

L'U.A.L existe sous forme de circuit intégré indépendant, ou intégrée dans d'autres circuits numériques spécialisés tels que les microprocesseurs ou les microcontrôleurs.

III- Les fonctions de base d'une U.A.L

1- Opérations logiques

Une unité arithmétique et logique permet de réaliser les opérations logiques sur des données numériques:

- ✎ **ET** logique;
- ✎ **OU** logique;
- ✎ **NON** logique;
- ✎ **NAND**, **NOR**, **OU exclusif** ...

2- Opérations arithmétiques

L'U.A.L permet également de réaliser les opérations arithmétiques : (addition, soustraction, division et multiplication) sur des données numériques.

3- Opération de comparaison

Une unité arithmétique et logique permet également de comparer deux nombres en indiquant dans un registre d'état si le résultat est plus grand, plus petit ou égal.

4- Opérations de décalage et de rotation

Un circuit à décalage est constitué par un nombre bien déterminé de bascules reliées entre elles en cascade. A chaque front d'horloge, la donnée transite d'une bascule à la suivante.

IV- Constitution

L'U.A.L, dispose en gros de:

- ✎ deux entrées de données **A** et **B** sur lesquelles on présentera les données à traiter;
- ✎ une sortie **F** donnant le résultat de l'opération effectuée;
- ✎ une entrée de sélection permettant le choix de l'opération à effectuer;
- ✎ un indicateur donnant l'état du résultat après exécution de l'opération.(Drapeau qui indiquera s'il ya eu erreur: division par zéro, résultat négatif, dépassement de capacité ...)

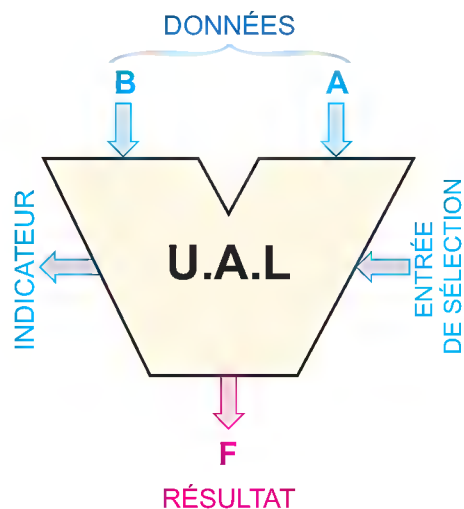
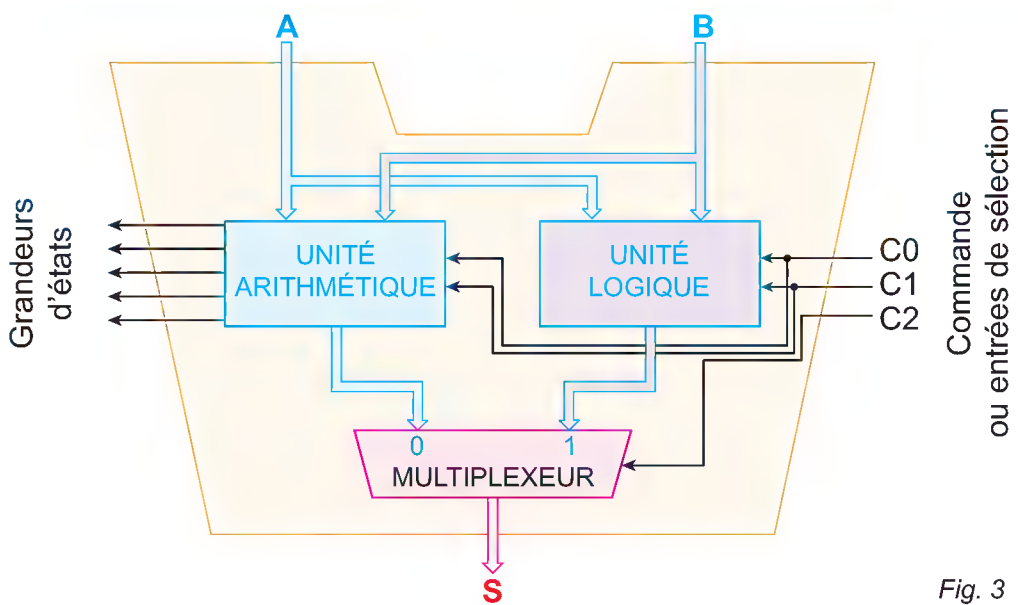


Fig. 2

V- Architecture simplifiée



Pour chacune des unités, le choix de la fonction se fait par les entrées de sélection **C0** et **C1**. Les deux unités fonctionnent en parallèle, et la sélection de la sortie d'une unité logique ou arithmétique se fait par la commande de multiplexage **C2**.

VI- Etude de L'U.A.L 74181

1- Présentation

Le **C.I 74181** est un circuit intégré pouvant réaliser des fonctions logiques et arithmétiques sur deux opérandes de 4 bits. Il comporte:

- ✦ Quatre entrées relatives à l'opérande **A**: $\bar{A}_3 \bar{A}_2 \bar{A}_1 \bar{A}_0$
- ✦ Quatre entrées relatives à l'opérande **B**: $\bar{B}_3 \bar{B}_2 \bar{B}_1 \bar{B}_0$
- ✦ Une entrée de commande **M** permettant de réaliser:
 - Des opérations logique si **M = 1**
 - Des opérations arithmétiques si **M = 0**
- ✦ Une entrée pour la retenue du poids le plus faible **Cn**
 - **Cn= 0**: opérations arithmétiques sans retenue
 - **Cn= 1**: opérations arithmétiques avec retenue
- ✦ Quatre sorties: $\bar{F}_3 \bar{F}_2 \bar{F}_1 \bar{F}_0$ pour l'affichage du résultat.
- ✦ Quatre entrées de sélection: $S_3 S_2 S_1 S_0$ pour choisir l'opération à réaliser.
- ✦ Une sortie pour la retenue de poids fort **Cn+4**
- ✦ Un comparateur qui met la sortie **A=B** à 1, chaque fois que l'entrée **A** est égale à l'entrée **B** indépendamment du résultat.
- ✦ Deux sorties \bar{P} et \bar{G} pour la liaison avec le circuit d'anticipation des retenues. (cette partie ne fera pas l'objet de notre étude).

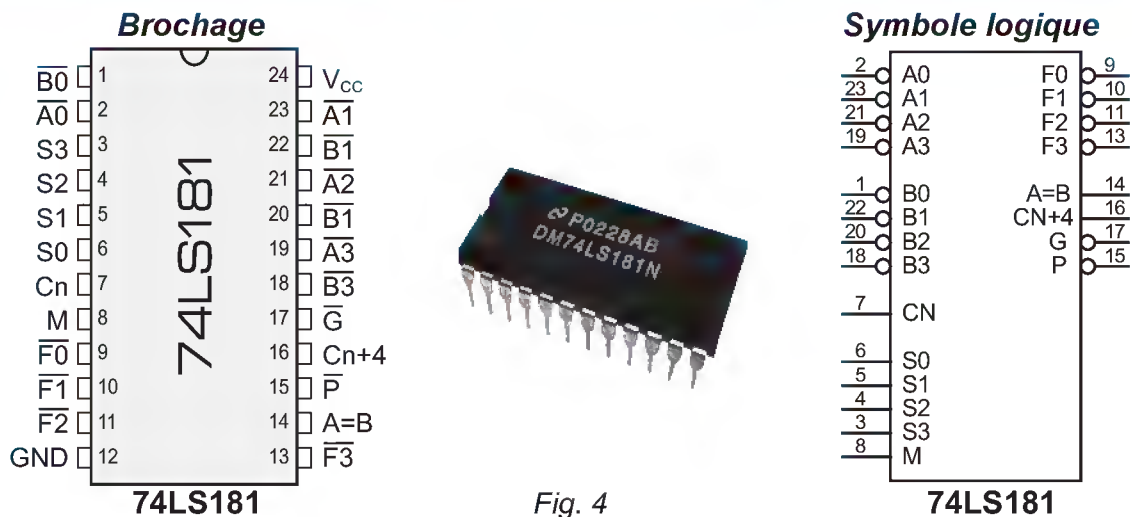


Fig. 4

2- Mise en œuvre du C.I 74181

La mise en œuvre de ce type de circuits repose en grande partie sur l'exploitation de la table de fonctionnement correspondante.

Selection S ₃ S ₂ S ₁ S ₀	Fonctions Logique (M = 1)	Fonctions arithmétiques (M = 0)	
		Cn= 0 (sans retenue)	Cn= 1 (avec retenue)
0 0 0 0	$F=\bar{A}$	F =A moins 1	F = A
0 0 0 1	$F=\bar{A}.B$	F =A.B moins 1	F =A.B
0 0 1 0	$F=\bar{A}+B$	F =A. \bar{B} moins 1	F=A. \bar{B}
0 0 1 1	$F=1$	F=moins 1(complément à 2)	F=0
0 1 0 0	$F=\bar{A}+B$	F= A plus (A+ \bar{B})	F = A plus(A+ \bar{B}) plus 1
0 1 0 1	$F=\bar{B}$	F = (A.B) plus (A+ \bar{B})	F = A.B plus(A+ \bar{B}) plus 1
0 1 1 0	$F=A\oplus\bar{B}$	F = A moins B moins 1	F = A moins B
0 1 1 1	$F=A+\bar{B}$	F=A+ \bar{B}	F=(A+ \bar{B}) plus 1
1 0 0 0	$F=\bar{A}.B$	F = A plus (A+B)	F = A plus (A+B) plus 1
1 0 0 1	$F=A\oplus B$	F = A plus B	F = A plus B plus 1
1 0 1 0	$F=B$	F=(A. \bar{B}) plus (A+B)	F=(A. \bar{B}) plus(A+B)plus1
1 0 1 1	$F=A+B$	F = A + B	F = (A+B) plus 1
1 1 0 0	$F=0$	F = A	F = A plus A plus 1
1 1 0 1	$F=A.\bar{B}$	F = (A.B) plus A	F = A.B plus A plus 1
1 1 1 0	$F=A.B$	F = (A. \bar{B}) plus A	F = A. \bar{B} plus A plus 1
1 1 1 1	$F=A$	F = A	F = A plus 1

3- Exemple: configurations et résultats pour quelques valeurs de A et B

Cn	M	S ₃ S ₂ S ₁ S ₀	Opération	A	B	F ₃ F ₂ F ₁ F ₀
X	1	1 0 0 1	A ⊕ B	1 1 0 1	0 1 0 1	1 0 0 0
0	0	0 1 1 0	A moins B moins 1	1 1 1 0	0 1 0 0	1 0 0 1
1	0	0 1 1 0	A moins B	0 1 1 1	0 1 0 0	0 0 1 1
1	0	1 1 0 0	A plus A plus 1	0 0 1 0	0 1 1 1	0 1 0 1

VII- ETUDE DE L'U.A.L 74381

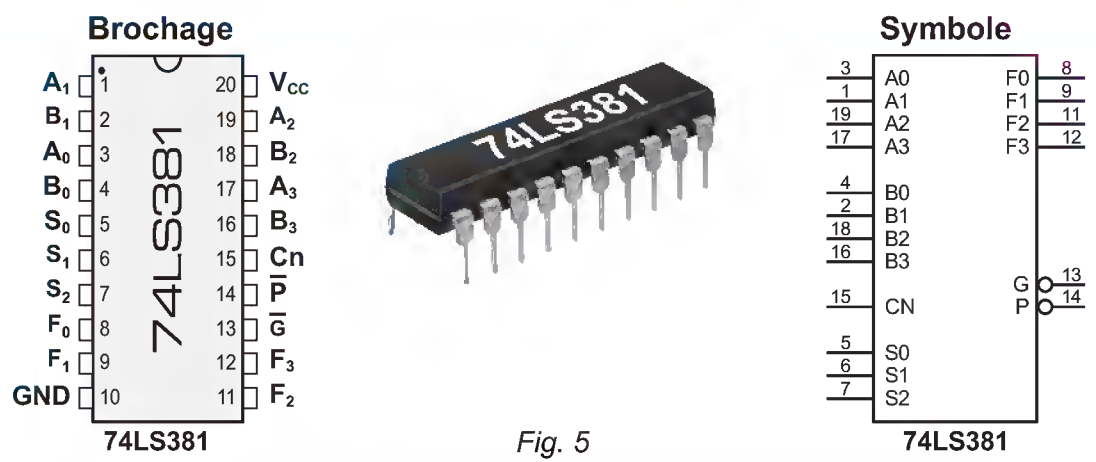


Fig. 5

1- Table de fonctionnement

Entrée de sélection			Opération réalisée
S ₂	S ₁	S ₀	F (F ₃ F ₂ F ₁ F ₀)
0	0	0	F = 0 0 0 0
0	0	1	F = B - A
0	1	0	F = A - B
0	1	1	F = A plus B
1	0	0	F = A XOR B
1	0	1	F = A OU B
1	1	0	F = A ET B
1	1	1	F = 1111

2- Exemple: configurations et résultats pour quelques valeurs de A et B

S ₂ S ₁ S ₀	Opération	A	B	F ₃ F ₂ F ₁ F ₀
0 0 1	B - A	1 1 1 1	1 1 0 1	0 0 1 0
1 1 0	A ET B	0 1 1 0	1 0 1 1	0 0 1 0
0 1 1	A plus B	0 1 1 1	0 1 0 1	1 1 0 0
1 1 1	F = 1111	1 1 1 0	0 0 0 0	1 1 1 1

AUTOMATIQUE

B. RÉSUMÉ

Définition: L'unité arithmétique et logique est une unité combinatoire permettant de réaliser plusieurs fonctions sur deux d'entrées à n bits. Elle permet d'exécuter plusieurs fonctions de type :

- ✂ Arithmétique (addition, soustraction, ...).
- ✂ Logiques (OU, ET, NOR...).
- ✂ Comparaison.
- ✂ Décalage et rotation.

Principe: Le choix de la fonction à réaliser se base sur le principe de multiplexage.

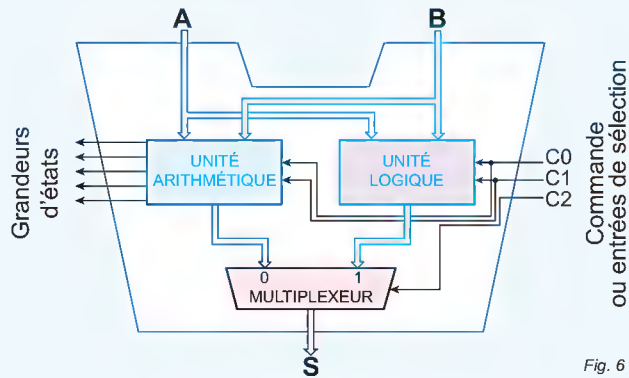


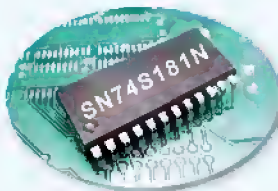
Fig. 6

Pour chacune des unités, le choix de la fonction se fait par les entrées de sélection C_0 et C_1 .

- ✂ $S_3 S_2 S_1 S_0$ pour le C.I 74181
- ✂ $S_1 S_0$ pour le C.I 74381

Les deux unités fonctionnent en parallèle, et la sélection de la sortie d'une unité (logique ou arithmétique) se fait par la commande de multiplexage C_2 .

- ✂ M : pour le C.I 74181
- ✂ S_2 : pour le C.I 74381



C. ÉVALUATION

I- Contrôle des connaissances

- ✎ Les entrées de sélection $S_2 S_1 S_0$ du C.I **74381** permettent de réaliser:
 - a. 6 fonctions différentes
 - b. 8 fonctions différentes
 - c. 16 fonctions différentes

- ✎ Le quel de ces trois circuits est une U.A.L ?
 - a. 74139;
 - b. 74181;
 - c. 74151.

- ✎ En logique:
 - a. $1 + 1 = 0$
 - b. $1 + 1 = 1$

- ✎ En arithmétique binaire:
 - a. $1 + 1 = 0$
 - b. $1 + 1 = 10$

II- Exercices résolus

EXERCICE N°1

Le circuit intégré 74381 est une UAL qui permet de réaliser des opérations logiques ou arithmétiques sur deux nombres A et B de 4 bits.

Sa table de fonctionnement est la suivante:

Entrée de sélection			Opération réalisée
S_2	S_1	S_0	$F (F_3 F_2 F_1 F_0)$
0	0	0	$F = 0000$
0	0	1	$F = B - A$
0	1	0	$F = A - B$
0	1	1	$F = A \text{ plus } B$
1	0	0	$F = A \text{ XOR } B$
1	0	1	$F = A \text{ OU } B$
1	1	0	$F = A \text{ ET } B$
1	1	1	$F = 1111$

En se référant à la table précédente:

- 1- Préciser les fonctions arithmétiques réalisées par ce circuit.
- 2- Préciser les fonctions logiques réalisées par ce circuit.

LOGIQUE COMBINATOIRE

3- Préciser la nature de l'opération réalisée et trouver les valeurs manquantes dans les configurations suivantes:

- ✎ $S_2S_1S_0 = 101$, $A = 1001$ et $B = 1101$
- ✎ $S_2S_1S_0 = 001$, $A = 1100$ et $F_3F_2F_1F_0 = 0011$
- ✎ $S_2S_1S_0 = 100$, $A = 1010$ et $B = 1110$
- ✎ $S_2S_1S_0 = 011$, $A = 0111$ et $F_3F_2F_1F_0 = 1110$

EXERCICE N°2

La carte électronique ci-dessous intégrée dans un système technique est destinée à faire des opérations logiques et arithmétiques.

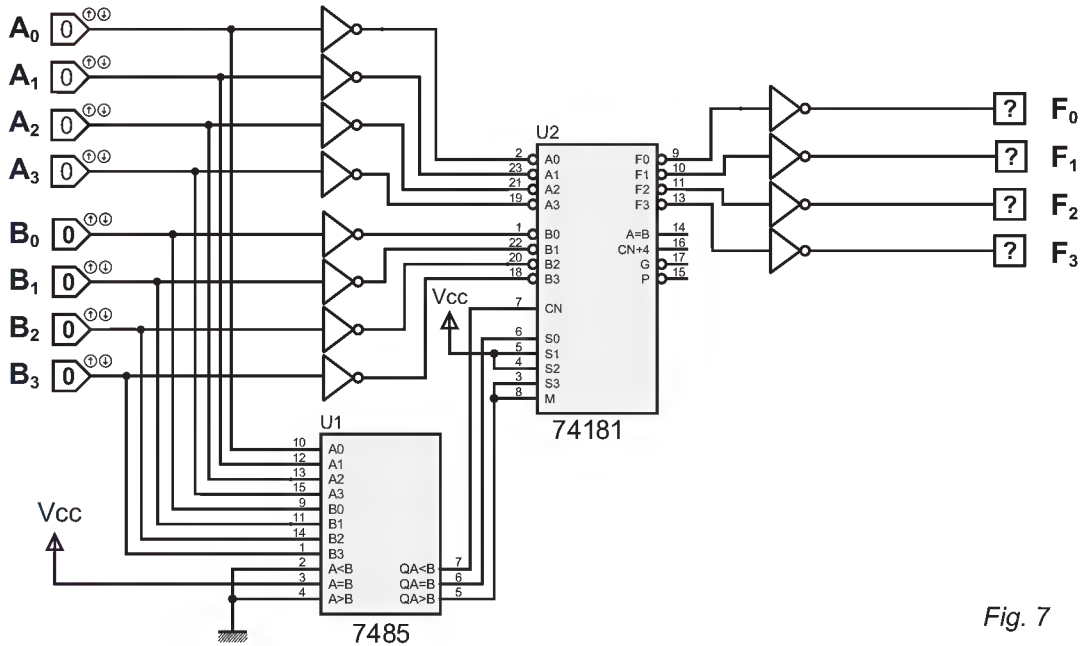


Fig. 7

Préciser la fonction réalisée par l'U.A.L et l'état logique de $F_3F_2F_1F_0$ pour les cas suivants:

- ✎ $A=1001$ et $B = 0110$
- ✎ $A=1001$ et $B = 1001$
- ✎ $A=0110$ et $B = 1001$

EXERCICE N°3

On désire réaliser la somme arithmétique de deux nombres de 8 bits chacun en utilisant le circuit 74181 $A=A_7A_6A_5A_4A_3A_2A_1A_0$ et $B=B_7B_6B_5B_4B_3B_2B_1B_0$.

- 1- Déterminer le nombre de circuits 74181 nécessaires pour réaliser cette addition.
- 2- Donner les configurations nécessaires des circuits utilisés.
- 3- Tracer le schéma de montage en précisant les niveaux logiques des différentes entrées et de la sortie si $A=(128)_{10}$ et $B=(64)_{10}$

III- Exercices à résoudre

EXERCICE N°1

Une carte électronique à base d'une U.A.L 74381 est définie par le schéma structurel suivant:

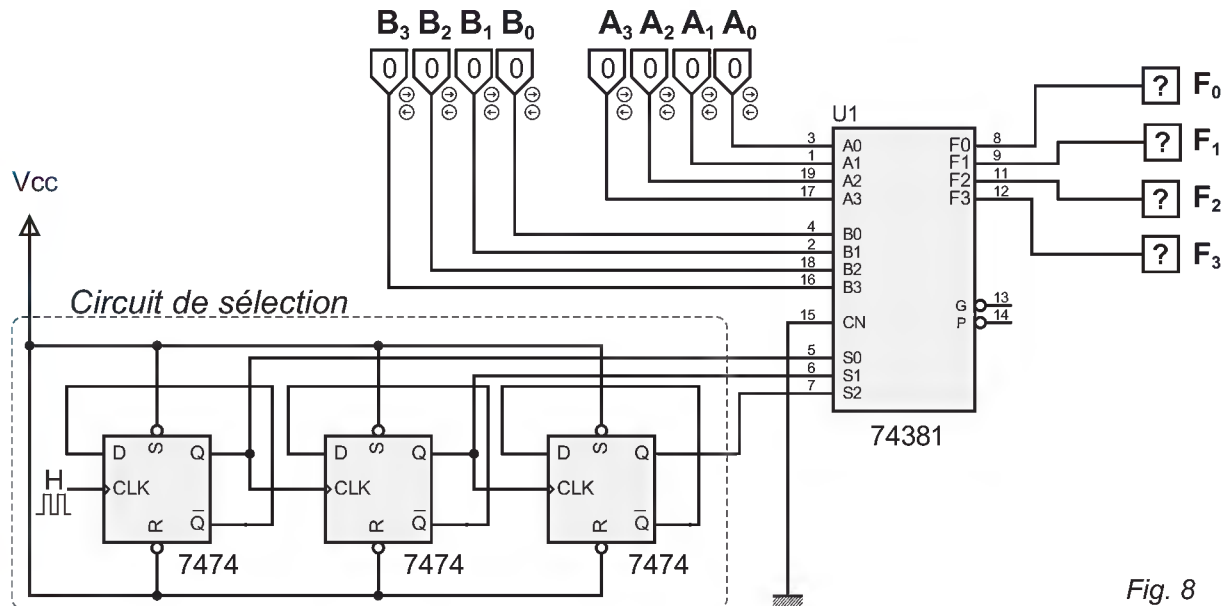


Fig. 8

En se référant au schéma structurel précédent et aux datasheets relatifs aux deux circuits (74LS381 et 7474):

- 1- Déterminer la nature et le type des bascules utilisées.
 - 2- Citer les caractéristiques du circuit de sélection ainsi que son modulo «M».
 - 3- Préciser la fonction réalisée par l'U.A.L et l'état logique de F_0 , F_1 , F_2 et F_3 dans les cas suivants :
 - Après trois impulsions du signal H.
 - Après sept impulsions du signal H.
 - Après dix impulsions du signal H.
- On donne **A= 1100** et **B= 1001**

EXERCICE N°2

En se référant au datasheet du circuit 74LS381 et au schéma structurel suivant :

- 1- Préciser la fonction réalisée par ce circuit.
- 2- Donner l'état logique de F_0 , F_1 , F_2 et F_3 .
- 3- Reprendre les questions 1 et 2 dans le cas où la broche 6 est reliée à la masse.

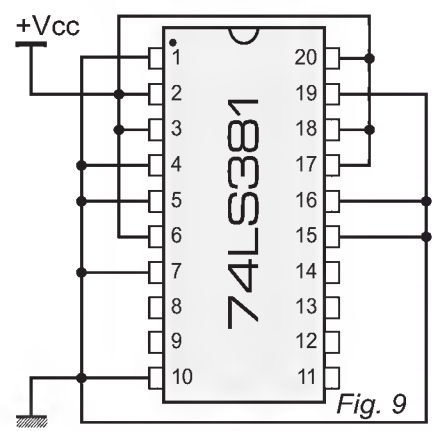


Fig. 9

AUTOMATIQUE

LOGIQUE COMBINATOIRE

IV- Correction des exercices

EXERCICE N°1

$S_2S_1S_0$	Opérande A	Opérande B	Opération réalisée	Sortie F
			Logique arithmétique	$F_3F_2F_1F_0$
101	1001	1101	A OU B.	1101
001	1100	1111	B - A	0011
100	1010	1110	A XOR B	0100
011	0111	0111	A PLUS B	1110

EXERCICE N°2

$A_3A_2A_1A_0$	$B_3B_2B_1B_0$	$S_3S_2S_1S_0$	M	Cn	Opération	$F_3F_2F_1S_0$
1001	0110	1110	1	0	A.B	0000
1001	1001	0111	0	0	$A+\bar{B}$	1111
0110	1001	0110	0	1	A moins B	1101

EXERCICE N°3

- 1- Nombre de circuits: deux 74181
- 2- Configuration

U.A.L 1						U.A.L 2					
S_3	S_2	S_1	S_0	M	Cn	S_3	S_2	S_1	S_0	M	Cn
1	0	0	1	0	0	1	0	0	1	0	CN+4

- 3- Schéma de montage

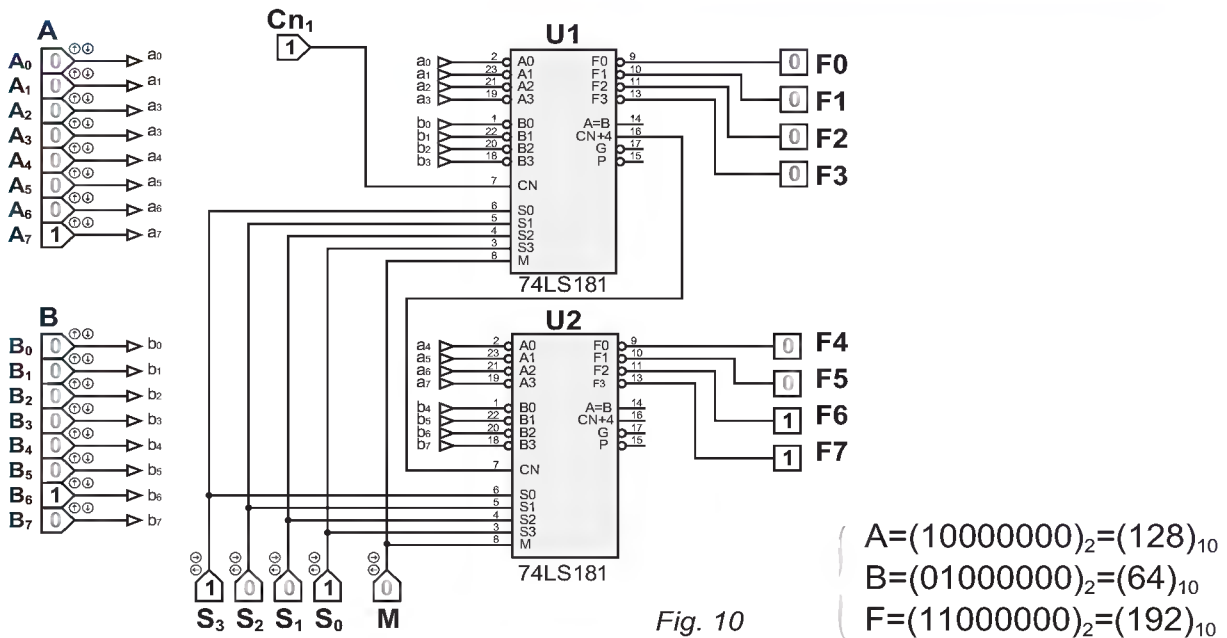


Fig. 10