



Université 8 mai 1945 Guelma

Faculté des Sciences et de la Technologie

Département de Génie Electrotechnique et Automatique



Polycopié de TP

Logique Combinatoire et Séquentielle

Dédié aux étudiants de la 2^{ème} Année ST, spécialités : Electronique, Electrotechnique, Télécommunication et Automatique.

Préparé par : Dr. Boubidi Assia

Année Universitaire 2024-2025

SOMMAIRE

Introduction	1
Conseils pratiques	2
Technologie des circuits intégrés TTL – CMOS	3
TP N 1 : Les portes logiques	7
TP N 2 : Analyse et synthèse des fonctions logiques	10
TP N 3 : L'additionneur et le comparateur	12
TP N 4 : Le codeur et le décodeur	16
TP N 5 : Le multiplexeur et le démultiplexeur	21
TP N 6 : Les bascules	25
TP N 7 : Les compteurs asynchrones	29
TP N 8 : Les compteurs synchrones	34
TP N 9 : Les registres	38
ANNEXE : Brochage des circuits intégrés logiques TTL	43

INTRODUCTION

Ce polycopié de travaux pratiques sur les circuits logiques combinatoires et séquentiels, s'adresse aux étudiants de niveau L2 de génie électrique.

Ce travail de laboratoire est composé de deux parties.

➤ **La première partie comprend l'étude des circuits logiques combinatoires.**

Dans un premier TP, les opérateurs logiques communément appelés portes logiques ont été abordés. Ce TP porte non seulement sur la vérification du fonctionnement de ces portes mais aussi sur la vérification de quelques lois fondamentales de l'algèbre de Boole.

Le deuxième TP est consacré à l'étude des fonctions logiques (complètement définies et incomplètement définies) et leurs simplifications ainsi que la réalisation de ces fonctions en utilisant les portes logiques universelles.

Les TPs 3, 4 et 5 sont réservés à l'étude des circuits logiques combinatoires fondamentaux. Il s'agit de réaliser et de vérifier le fonctionnement de quelques circuits combinatoires usuels tels que les additionneurs, les comparateurs, les multiplexeurs, les démultiplexeurs et les décodeurs.

➤ **La deuxième partie comprend l'étude des circuits logiques séquentiels.**

Le TP 6 entame des différents types de bascules qui représentent l'élément de base de tout circuit séquentiel ainsi que les différents modes de synchronisation (synchronisation sur niveau et synchronisation sur front).

Les TPs 7 et 8 ont abordé une application majeure des circuits séquentiels, largement utilisée dans les circuits numériques : les compteurs. Ces derniers ont été étudiés sous leurs deux formes ; synchrone et asynchrone, d'abord à l'aide de bascules discrètes puis à l'aide de circuits intégrés.

Le TP 9 traite un autre circuit aussi important que le précédent, il s'agit du registre. Les registres sont abordés avec les différents modes d'écriture (chargement) et de lecture ainsi qu'un exemple d'application pour chaque type.

CONSEILS PRATIQUES

1. Préparation du TP

La préparation du TP est importante. Vous devez, avant de vous présenter au TP, avoir préparé votre travail, c'est à dire avoir rédigé la partie théorique du TP.

2. Travail durant la séance de TP

a. Le matériel

Vous disposez d'une table de travail. Chaque table est équipée d'au moins :

- Une alimentation continue stabilisée de 5V.
- Un multimètre numérique.
- Un générateur de signaux.
- Une platine d'essai.
- Des câbles de connexion.

b. Les composants

Selon le TP, ils sont principalement des circuits intégrés logiques, des résistances de protection, des diodes électroluminescentes, et des afficheurs 7 segments.

c. La réalisation

Le travail s'effectuera en binômes ou trinômes. Le comportement pendant les TPs, incluant l'autonomie, la capacité à résoudre les problèmes et l'efficacité, sera pris en compte dans l'évaluation.

Pour éviter les erreurs de câblage, il est très important de :

- Dessiner le montage en indiquant les numéros des broches,
- Vérifier que tous les circuits utilisés sont alimentés (une erreur courante est d'avoir un ou des circuits non alimentés).

d. Le compte-rendu

Un compte-rendu est exigé à la fin de chaque séance de TP et doit comporter pour chaque manipulation :

- Une partie théorique.
- Un ou des schémas (logigrammes) avec des symboles normalisés, un ou des chronogrammes.
- Une analyse des résultats obtenus.
- Remarques, suggestions et conclusion.

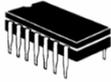
3. Contrôle (Soutenance de TP)

Un contrôle, d'une durée maximum d'une heure, aura lieu à la fin de toutes les séances de TP. Ce contrôle, qui sera individuel, comptera pour 1/3 de la note finale des TPs.

4. Note finale du TP

La note finale attribuée est la moyenne entre la note de préparation TP, participation (efficacité durant le TP), le compte rendu et le contrôle.

IMPORTANT : Dans l'intérêt de tous, chaque groupe doit s'assurer à la fin de chaque séance, de laisser une table de travail propre et rangée. Cela inclut la mise hors tension des appareils, le rangement des câbles, et des composants.



TECHNOLOGIE DES CIRCUITS INTEGRES TTL - CMOS

1. Généralités

1.1. Définition de circuit intégré logique

La matérialisation des fonctions logiques a été d'abord réalisée avec des composants discrets puis elle s'est transformée en intégrant plusieurs composants discrets sur un seul circuit.

Un circuit intégré (C.I.) désigne un bloc constitué de silicium de quelques millimètres carrés en forme parallélépipède rectangle aplati, à l'intérieur duquel se trouve inscrit en nombre variable des composants électroniques (transistors, diodes, résistances et, plus rarement des condensateurs).

Donc les C.I. sont constitués d'un boîtier qui contient la puce, laquelle est reliée à l'extérieur par un certain nombre de pattes (ou broches). Ce nombre varie généralement entre 14 et 28.

1.2. Notion de familles de circuits logiques

Les C.I. peuvent être classés en deux familles : **TTL** ou **CMOS**

A) La famille TTL (Transistors Transistor Logic) : Elle est principalement conçue de transistor bipolaire.

B) La famille CMOS (Complementary Metal Oxyde Semiconductor) : Elle dérive principalement des transistors à effet de champ MOS.

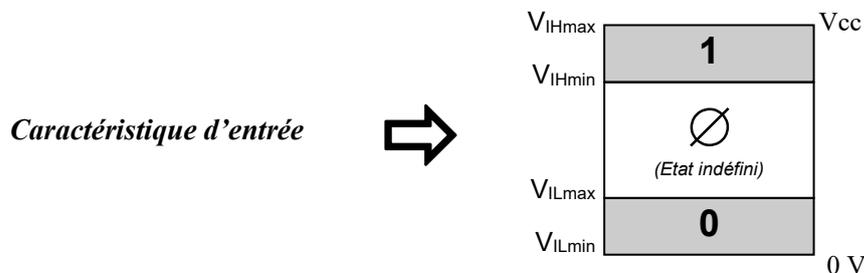
1.3. Notion de niveaux logiques

Pour une famille donnée, les niveaux logiques « 0 » et « 1 » ne correspondent pas à une tension précise, mais à une certaine « plage » de tension.

➤ On appellera pour les valeurs de tension en **entrée (Input) :**

V_{IHmin} : Tension minimale en entrée qui assure le niveau logique haut.

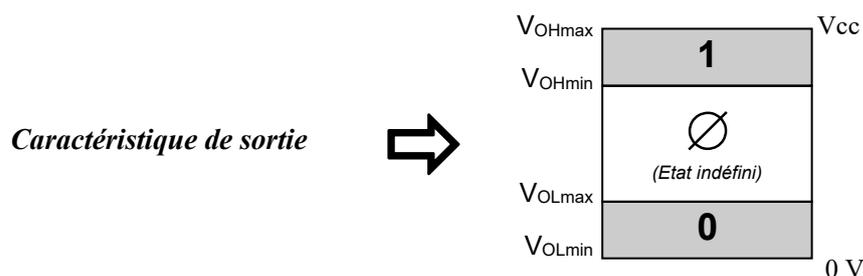
V_{ILmax} : Tension maximale en entrée qui assure le niveau logique bas.



➤ On appellera pour les valeurs de tension en **sortie (Output) :**

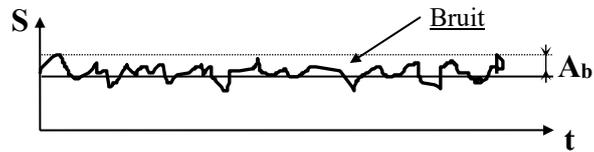
V_{OHmin} : Tension minimale de sortie à l'état logique haut.

V_{OLmax} : Tension maximale de sortie à l'état logique bas.



1.4. Immunité aux bruits

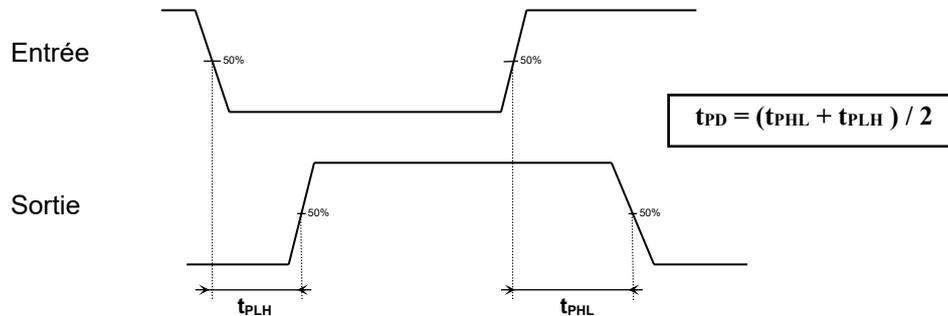
- La différence $V_{OHmin} - V_{IHmin}$ est appelée : marge de bruit à l'état haut.
- La différence $V_{ILmax} - V_{OLmax}$ est appelée : marge de bruit à l'état bas.



Si A_b est inférieur à la marge de bruit au niveau bas, il n'y aura pas d'influence sur le fonctionnement (*Fonctionnement correct*).

1.5. Temps moyen de propagation

Lorsqu'on applique à l'entrée d'un circuit un niveau logique, il y a un certain retard pour que la sortie réagisse. Cette durée est le temps moyen de propagation t_{PD} .



t_{PHL} : Temps de propagation du niveau haut au niveau bas.

t_{PLH} : Temps de propagation du niveau bas au niveau haut.

Remarque : Ce temps détermine la fréquence maximale F_{MAX} à laquelle les circuits intégrés sont capables de réagir.

2. La Famille logique TTL

2.1. Tension d'alimentation

Elle est fixe et égale à +5V avec une tolérance de $\pm 5\%$ (*sauf séries Low Voltage*).

2.2. Niveaux logiques

- ✓ $V_{IHmin} = 2\text{ v}$
- ✓ $V_{ILmax} = 0,8\text{ v}$
- ✓ $V_{OHmin} = 2,4\text{ v}$
- ✓ $V_{OLmax} = 0,4\text{ v}$

2.3. Les différentes technologies

La famille TTL se subdivise en plusieurs technologies (séries), dont chacune possède ses propres caractéristiques de fonctionnement.

TTL standard	74 xx
TTL Low power (à faible consommation)	74 L xx
TTL High speed (rapide)	74 H xx
TTL Schottky (2 fois plus rapide que la 74Hxx)	74 S xx
TTL Fast	74 F xx
TTL Low power Schottky	74 LS xx
TTL Advanced Schottky	74 AS xx
TTL Advanced Low power Schottky	74 ALS xx

3. La Famille logique CMOS

3.1. Tension d'alimentation

Elle peut varier de +3V à + 18V (Voir documentation constructeur).

3.2 niveaux logiques

- ✓ $V_{IH\min} = 70\% V_{CC}$
- ✓ $V_{OH\min} = V_{CC}$
- ✓ $V_{IL\max} = 30\% \text{ de } V_{CC}$
- ✓ $V_{OL\max} = 0V$

3.3 Les différentes technologies

a) Circuits intégrés CMOS série 4000 : **4000 et 4000 B**

b) Circuits intégrés CMOS série 74 :

- CMOS classique **74 C xx (technologie identique à la série 4000 B)**
- CMOS rapide (High-speed CMOS) **74 HC xx et 74 HCT xx**
- CMOS avancée (Advanced CMOS) **74 AC 11 xx et 74 ACT 11 xx**

4. Principales caractéristiques des technologies les plus utilisées

	HCMOS	TTL LS	TTL ALS	CMOS
Vcc/Vdd	3 à 6V	5 V ± 5%	5 V ± 10 %	3 à 15 V
plage de °C	-40 à 85	0 à 70	0 à 70	-40 à 85
Vil à Vcc = 5V	1 V	0.8 V	0.8 V	1.5 V
ViH à Vcc = 5V	3.5 V	2 V	2 V	3.5 V

5. Choix d'une famille

Pour une application donnée le choix de la famille dépend d'un ensemble de caractéristiques dont les plus importants sont : la tension d'alimentation, la consommation et le temps de propagation.

	T.T.L.			C.M.O.S.		
	74	74 LS	74 ALS	4000	74 HC	74 HCT
Tension d'alimentation	5 V	5 V	5 V	3 à 18 V	2 à 6 V	5 V
Dissipation par porte	10 mW	2 mW	1 mW	2.5 mW	2.5 mW	2.5 mW
Temps de propagation par porte	10 ns	9.5 ns	4 ns	40 ns	10 ns	10 ns

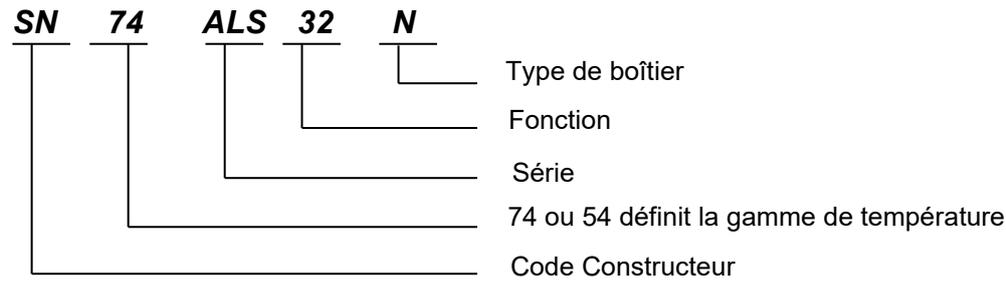
6. Comparaison des familles TTL - CMOS

	TTL	CMOS
+	<ul style="list-style-type: none"> ✓ très large gamme de fonctions ✓ bonne immunité aux bruits ✓ temps de propagation faible 	<ul style="list-style-type: none"> ✓ tension d'alimentation variable ✓ excellente immunité aux bruits ✓ consommation statique quasi-nulle * ✓ densité d'intégration élevée
-	<ul style="list-style-type: none"> ✓ consommation statique importante ✓ densité d'intégration réduite 	<ul style="list-style-type: none"> ✓ mauvaise immunité aux bruits ✓ temps de propagation important

* **Remarque :** En statique, la consommation de la famille CMOS est quasi-nulle. En haute fréquence (> 1 MHz), elle rattrape la consommation de la famille TTL.

7. Code des circuits logiques TTL et CMOS

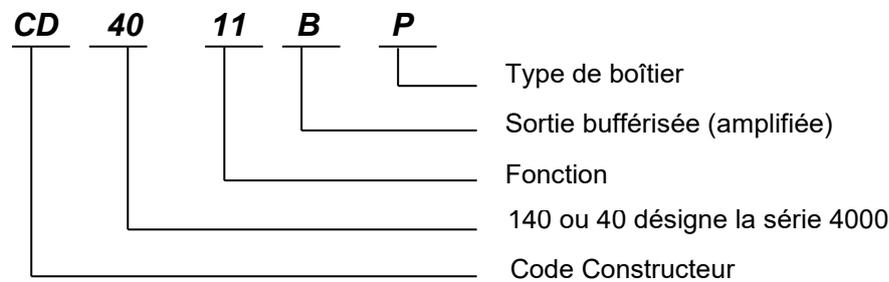
TTL



Gamme de température : 74 : série grand public (de 0° a 70°)
54 : série militaire (de -55° a 125°)

Code Constructeur : SN TEXAS
MC MOTOROLA
USN SPRAGUE
ITT I.T.T.....

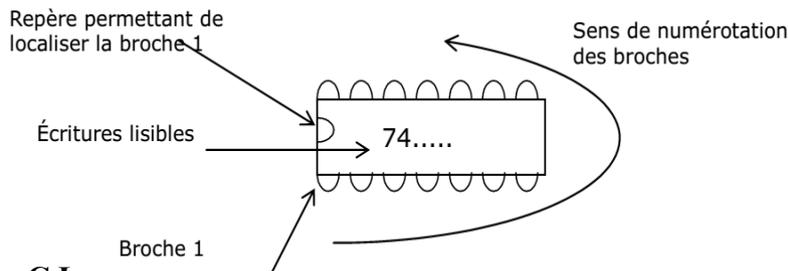
CMOS



Code Constructeur : CD R.C.A
SCL SOLID STATE SCIENTIFIC
MC MOTOROLA

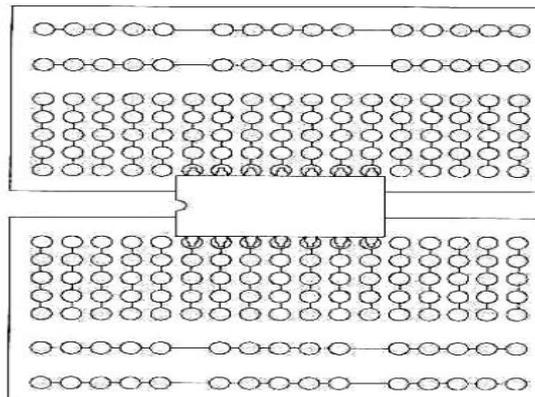
8. Repérage des circuits intégrés logiques TTL et CMOS

Le repérage des C.I. s'obtient à l'aide d'une encoche de repère (figure ci-dessous).



9. Placement des C.I.

Lorsqu'on utilise des C.I., on doit les placer de façon qu'il n'y ait aucun contact entre les pattes comme le montre la figure suivante.



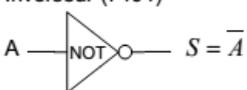
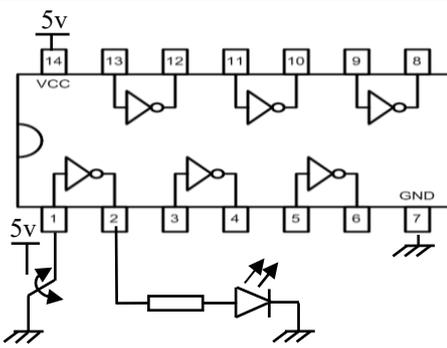
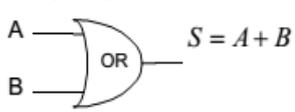
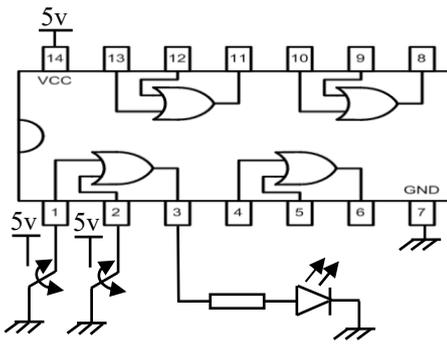
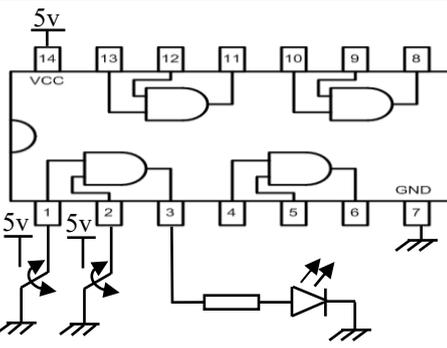
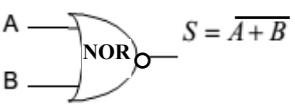
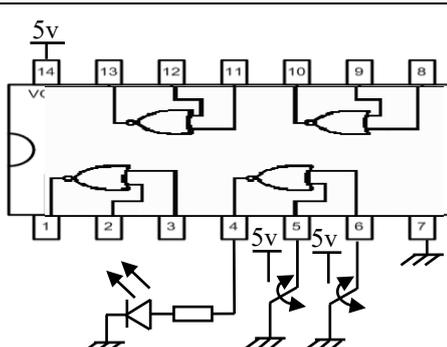
TP 1 : LES PORTES LOGIQUES

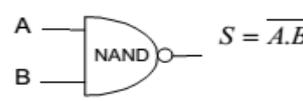
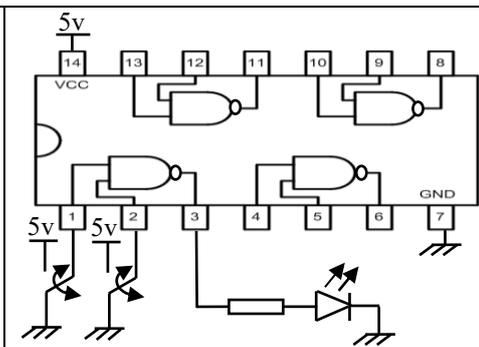
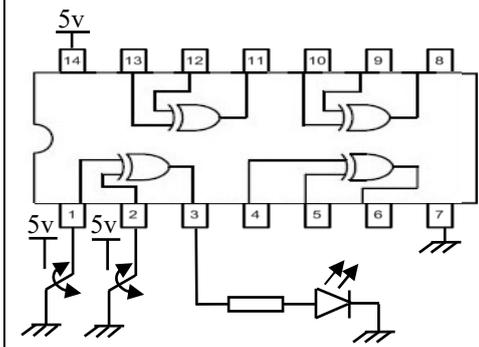
Objectif du TP

- ✓ Découvrir les circuits intégrés logiques.
- ✓ Vérifier pratiquement quelques théorèmes de l'algèbre de BOOLE.
- ✓ Réaliser des schémas de câblage.

1. Portes logiques

- 1) Rappeler les tables de vérité des portes ci-dessous.
- 2) **Réaliser** les câblages puis **vérifier** ses tables de vérité.

Porte logique, Symbole Américain et Table de vérité	Câblage des circuits															
<p>Inverseur (7404)</p> <div style="display: flex; align-items: center; justify-content: center;"> <div style="text-align: center;">  </div> <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>A</th> <th>S</th> </tr> </thead> <tbody> <tr> <td>0</td> <td></td> </tr> <tr> <td>1</td> <td></td> </tr> </tbody> </table> </div>	A	S	0		1											
A	S															
0																
1																
<p>OU (7432)</p> <div style="display: flex; align-items: center; justify-content: center;"> <div style="text-align: center;">  </div> <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>A</th> <th>B</th> <th>S</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td></td> </tr> <tr> <td>0</td> <td>1</td> <td></td> </tr> <tr> <td>1</td> <td>0</td> <td></td> </tr> <tr> <td>1</td> <td>1</td> <td></td> </tr> </tbody> </table> </div>	A	B	S	0	0		0	1		1	0		1	1		
A	B	S														
0	0															
0	1															
1	0															
1	1															
<p>ET (7408)</p> <div style="display: flex; align-items: center; justify-content: center;"> <div style="text-align: center;">  </div> <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>A</th> <th>B</th> <th>S</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td></td> </tr> <tr> <td>0</td> <td>1</td> <td></td> </tr> <tr> <td>1</td> <td>0</td> <td></td> </tr> <tr> <td>1</td> <td>1</td> <td></td> </tr> </tbody> </table> </div>	A	B	S	0	0		0	1		1	0		1	1		
A	B	S														
0	0															
0	1															
1	0															
1	1															
<p>NOR (7402)</p> <div style="display: flex; align-items: center; justify-content: center;"> <div style="text-align: center;">  </div> <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>A</th> <th>B</th> <th>S</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td></td> </tr> <tr> <td>0</td> <td>1</td> <td></td> </tr> <tr> <td>1</td> <td>0</td> <td></td> </tr> <tr> <td>1</td> <td>1</td> <td></td> </tr> </tbody> </table> </div>	A	B	S	0	0		0	1		1	0		1	1		
A	B	S														
0	0															
0	1															
1	0															
1	1															

<p>NON ET (7400)</p>  <table border="1" style="margin-left: auto; margin-right: auto; border-collapse: collapse; text-align: center;"> <thead> <tr> <th>A</th> <th>B</th> <th>S</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td></td></tr> <tr><td>0</td><td>1</td><td></td></tr> <tr><td>1</td><td>0</td><td></td></tr> <tr><td>1</td><td>1</td><td></td></tr> </tbody> </table>	A	B	S	0	0		0	1		1	0		1	1		
A	B	S														
0	0															
0	1															
1	0															
1	1															
<p>7486</p>  <table border="1" style="margin-left: auto; margin-right: auto; border-collapse: collapse; text-align: center;"> <thead> <tr> <th>A</th> <th>B</th> <th>S</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td></td></tr> <tr><td>0</td><td>1</td><td></td></tr> <tr><td>1</td><td>0</td><td></td></tr> <tr><td>1</td><td>1</td><td></td></tr> </tbody> </table>	A	B	S	0	0		0	1		1	0		1	1		
A	B	S														
0	0															
0	1															
1	0															
1	1															

2. Lois de De Morgan

Vérifier pratiquement les lois de De Morgan.

	Câblage du 1 ^{er} membre & TV			Câblage du 2 ^{ème} membre & TV		
	A	B	S1	A	B	S2
1 ^{ère} lois : = Rq : porte déjà testée						
2 ^{ème} lois : = Rq : porte déjà testée						

3. Utilisation des portes NOR et NAND

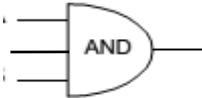
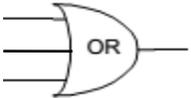
Les portes NOR et NAND sont des portes universelles, c-à-d tout circuit peut être réalisé en utilisant uniquement des portes NOR ou NAND. On dit que ces deux portes constituent des *opérateurs complets*.

En utilisant les propriétés de l'algèbre de BOOLE, exprimer les portes NOT, OR et AND uniquement en fonction des portes NOR puis NAND.

	Réalisation en utilisant NOR		Réalisation en utilisant NAND	
NOT	
OR	A réaliser
AND	

4. Portes logiques à trois entrées

- 1) Comment peut-on utiliser une porte AND à trois entrées en porte à deux entrées ? Proposer deux solutions possibles.
- 2) Comment peut-on utiliser une porte OR à trois entrées en porte à deux entrées ? Proposer deux solutions possibles.

	1 ^{ère} solution		2 ^{ème} solution	
<p>7411</p> 	$A \cdot B = \dots\dots\dots$		$A \cdot B = \dots\dots\dots$	A réaliser
	$A + B = \dots\dots\dots$		$A + B = \dots\dots\dots$	

TP N 2 : ANALYSE ET SYNTHÈSE DES FONCTIONS LOGIQUES

Objectif du TP

- ✓ Synthétiser des fonctions logiques combinatoires après simplification,
- ✓ Réaliser le câblage d'un logigramme,
- ✓ Analyser des circuits logiques combinatoires.

Partie 1 : Synthèse de fonctions logiques

1) Fonction complètement définie : Vote au directoire

Le comité directeur d'une entreprise est constitué de quatre membres :

- Le directeur D,
- Ses trois adjoints A, B, C.

Lors des réunions, les décisions sont prises à la majorité. Chaque personne dispose d'un interrupteur pour voter sur lequel elle appuie en cas d'accord avec le projet soumis au vote. En cas d'égalité du nombre de voix, celle du directeur compte double.

Dans le but de réaliser un dispositif logique permettant l'affichage du résultat du vote R, suivre les étapes suivantes.

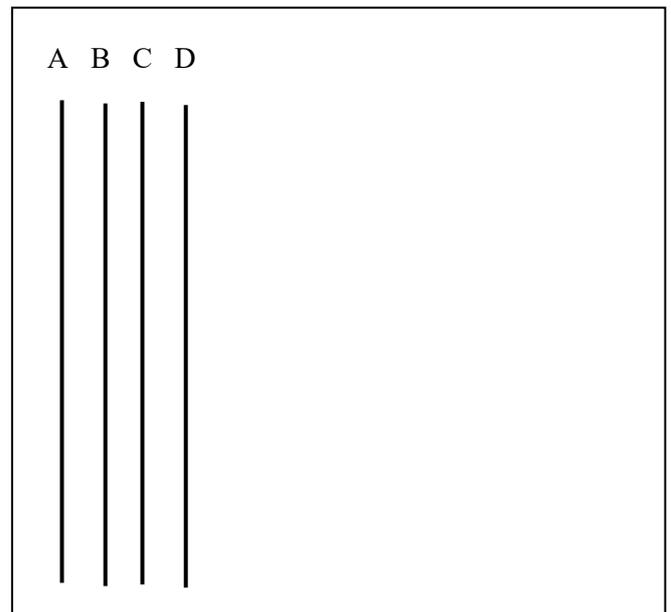
1) Compléter la table de vérité.

D	C	B	A	R
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	

2) Donner l'expression logique simplifiée de R.

R =

3) Réaliser le logigramme de la sortie R.



2) Fonction incomplètement définie

Dans une automobile, les exigences de l'alarme liée à la ceinture de sécurité sont :

- On désire que l'alarme soit actionnée pendant quelques secondes lorsqu'il y a un poids sur le siège du conducteur, lorsque la clé est dans le contact en position de marche (ON) et la ceinture n'est pas attachée.
- On désire également que l'alarme soit actionnée même sans poids sur le siège mais avec la clé de contact en place (position ON), pour éviter que le conducteur quitte la voiture en laissant la clé dans le contact.

Les variables du système sont :

- P : poids sur le siège (oui/non),
- C : état de la ceinture (attachée/détachée),
- Cl : état de la clé seulement en position ON ou OFF ('1' ou '0'),
- A : l'alarme (actionnée/non actionnée).

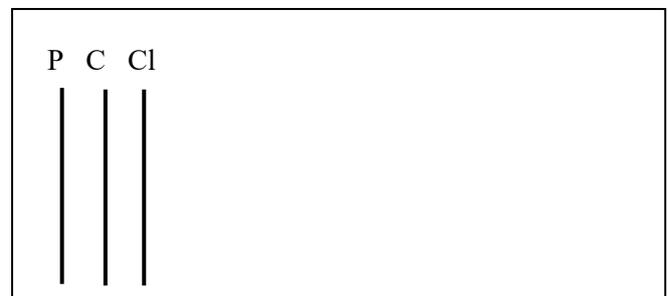
1) Compléter la table de vérité.

P	C	Cl	A
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

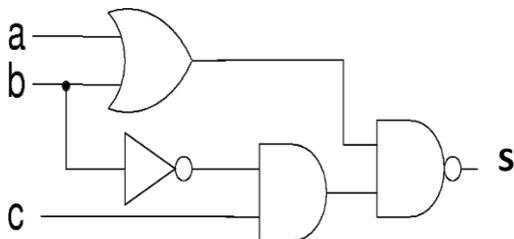
2) Donner l'expression logique simplifiée de A.

A =

3) Réaliser le logigramme de la sortie A.



Partie 2 : Analyse d'un circuit logique



1. Etablir l'expression de S en fonction des variables a, b, c.

S =

2. Simplifier algébriquement cette expression.

.....

3. Réaliser le logigramme.

4. Tester et compléter la table de vérité :

a	b	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	

TP N 3 : L'ADDITIONNEUR ET LE COMPAREUR

Objectif du TP

- ✓ Vérifier expérimentalement le fonctionnement de l'additionneur (application : addition et soustraction)
- ✓ Vérifier expérimentalement le fonctionnement du comparateur.

1. L'additionneur

1.1. Qu'est ce qu'un additionneur ?

C'est un circuit réalisant l'addition de deux nombres binaires.

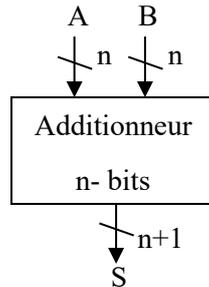


Figure 3.1 : Symbole logique d'un additionneur n bits.

1.2. Additionneurs en circuits intégrés

Parmi les additionneurs parallèles disponibles en circuits d'intégration à moyenne échelle MSI, nous citons :

- ✓ Les circuits TTL :
 - Additionneur à 2 bits : 7482,
 - Additionneur à 4 bits : 74LS83, 74LS283
- ✓ Les circuits CMOS :
 - Additionneur à 4 bits : 4008

1.3. Exemple d'additionneurs 4 bits : le 7483

Le 7483 est un additionneur complet de deux nombres A et B de 4 bits chacun. Il possède :

- ✓ 1 mot d'entrée : $A = A_3A_2A_1A_0$
- ✓ 1 mot d'entrée : $B = B_3B_2B_1B_0$
- ✓ 1 entrée de mise en cascade : C_0
- ✓ 5 sorties d'addition : $C_4S_3S_2S_1S_0$

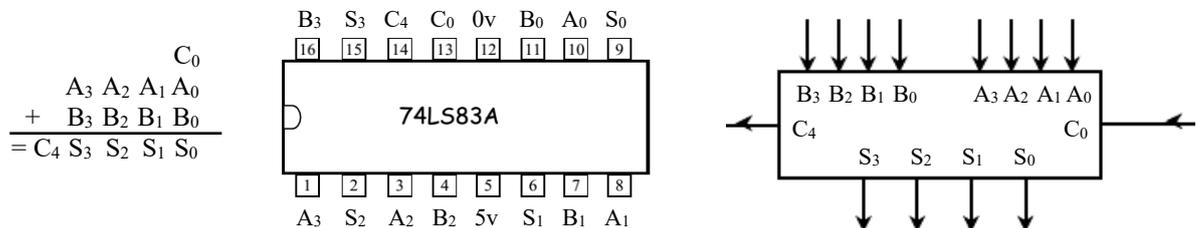
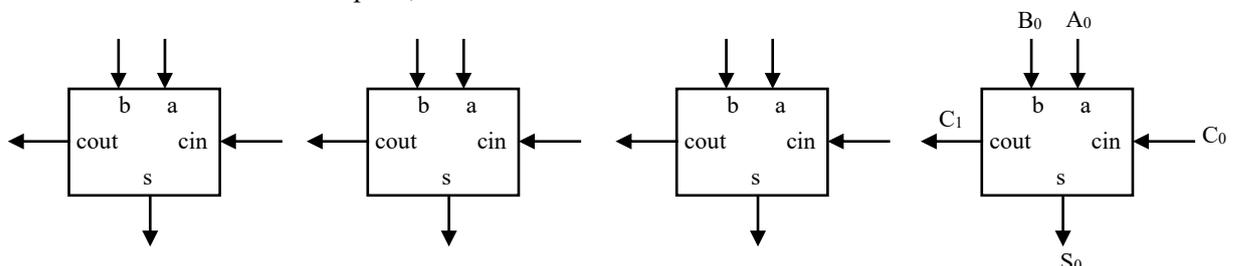


Figure 3.2: Brochage et symbole logique du 7483

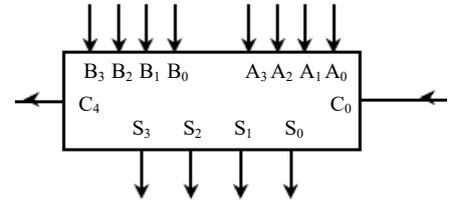
1) A l'aide d'additionneurs complets, donner la structure interne du 7483.



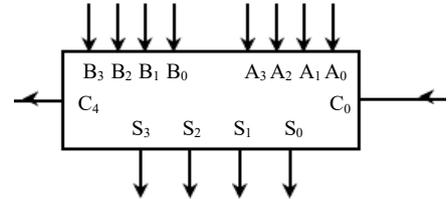
Application 1 : l'addition binaire

1) Compléter le câblage des CI 7483 pour réaliser les opérations d'addition suivantes :

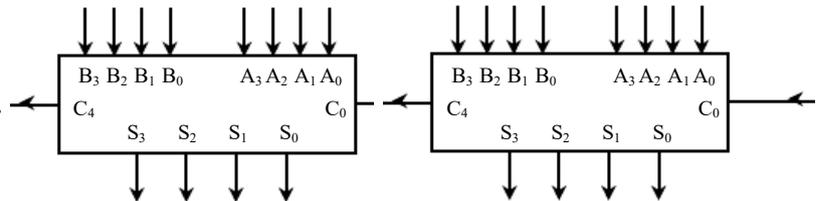
$$\begin{array}{r} 5 \\ + 7 \\ \hline = \end{array}$$



$$\begin{array}{r} 14 \\ + 10 \\ \hline = \end{array}$$



$$\begin{array}{r} 67 \\ + 73 \\ \hline = \end{array}$$



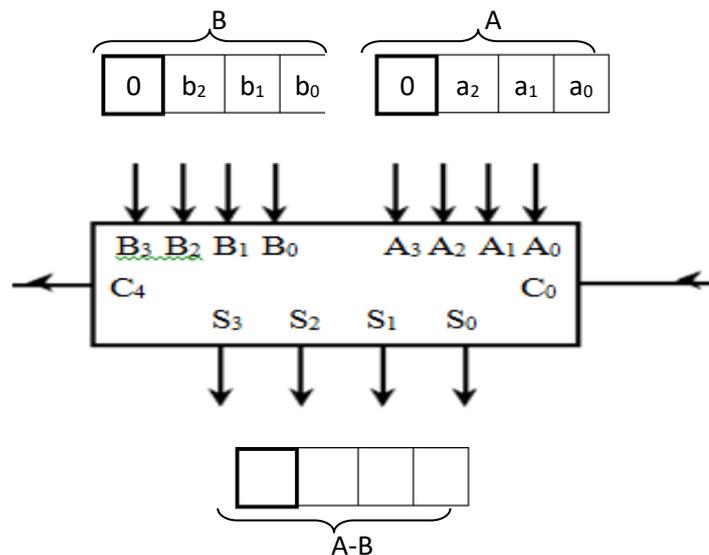
Application 2 : la soustraction binaire

L'additionneur binaire en plus de l'addition permet entre autres de réaliser une soustraction binaire en complément à 2.

La soustraction binaire est équivalente à une addition : $A - B = A + C2(B)$

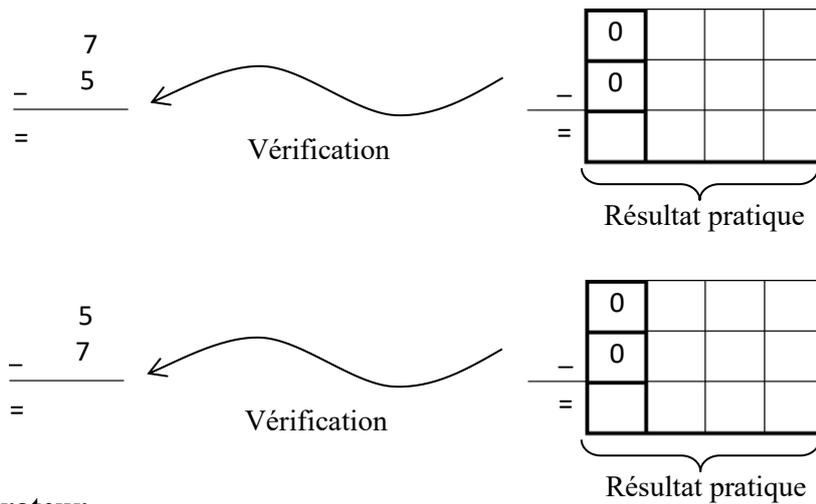
Le complément à 2 d'un nombre binaire signé s'obtient : $C2(B) = \dots\dots\dots$

1) Compléter le logigramme du soustracteur 4 bits (A-B) en utilisant la notation en complément à 2.



2) Quelles sont les valeurs limites de A et B que peut traiter ce soustracteur. $A_{max} = \dots, B_{max} = \dots$

3) Câbler et réaliser les opérations de soustraction suivantes :



1. Le comparateur

2.1. Qu'est ce qu'un comparateur ?

C'est un circuit réalisant la comparaison entre deux nombres binaires.

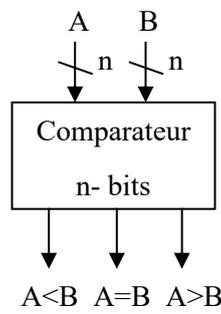


Figure 3.3 : Symbole logique d'un comparateur n bits.

2.2. Compareurs en circuits intégrés

Parmi les comparateurs disponibles en circuits d'intégration à moyenne échelle MSI, nous citons :

✓ Les circuits TTL :

- Comparateur à 4 bits : 7485, 74LS85
- Comparateur à 8 bits : 74LS682, 74LS688

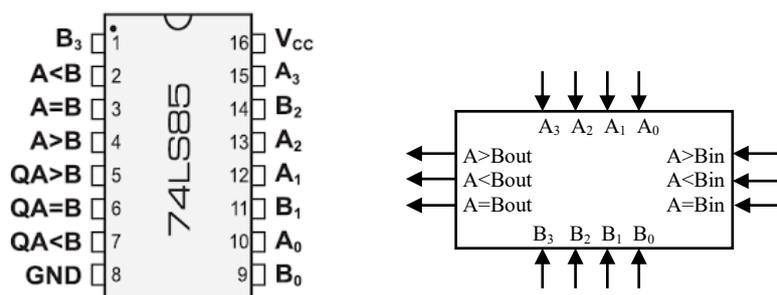
✓ Les circuits CMOS :

- Comparateur 4 bits : 4063, 4585

2.3. Exemple de comparateurs 4 bits : le 7485

Le 7485 est un comparateur de deux nombres A et B à 4 bits chacun. Il possède :

- ✓ 1 mot d'entrée : $A = A_3A_2A_1A_0$
- ✓ 1 mot d'entrée : $B = B_3B_2B_1B_0$
- ✓ 3 entrées de mise en cascade : $A > B_{in}$, $A = B_{in}$ et $A < B_{in}$
- ✓ 3 sorties de comparaison : $A > B_{out}$, $A = B_{out}$ et $A < B_{out}$



	A ₃ B ₃	A ₂ B ₂	A ₁ B ₁	A ₀ B ₀	A>Bin	A<Bin	A=Bin
A>Bout	>	X	X	X	X	X	X
	=	>	X	X	X	X	X
A<Bout							
A=Bout							

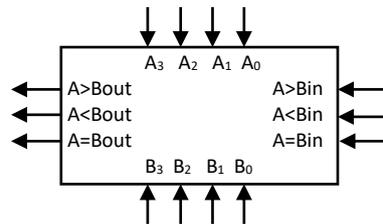
Figure 3.4 : Brochage, Symbole logique et Tableau de vérité du 7485.

Travail demandé

- 1) Compléter la table de vérité du 7485.
- 2) Compléter le câblage des CI 7485 pour comparer deux nombres A et B dans chaque cas.

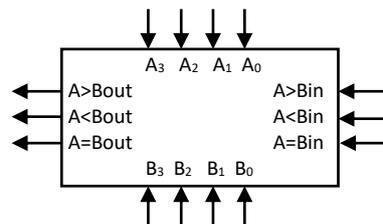
A = 5 =

B = 7 =



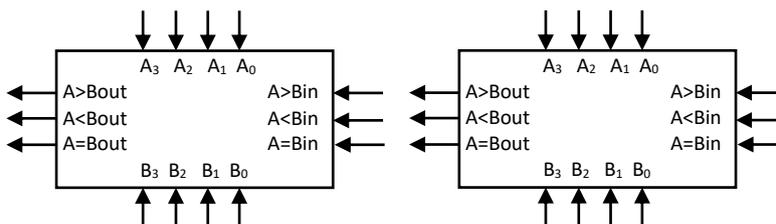
A = 14 =

B = 10 =



A = 67 =

B = 73 =



TP N 4 : LE CODEUR ET LE DECODEUR

Objectif du TP

Vérifier expérimentalement le fonctionnement des codeurs (de priorité), des décodeurs (décimal, BCD/7 segments) et des afficheurs.

1. Le codeur

1.1. Qu'est-ce qu'un codeur (encodeur)

C'est un circuit qui traduit les valeurs d'une entrée dans le code binaire (ou un autre code). Parmi les m entrées, une seule est activée à la fois. Les n sorties représentent le n° de l'entrée activée dans le code binaire.

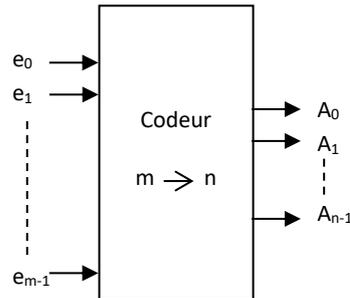
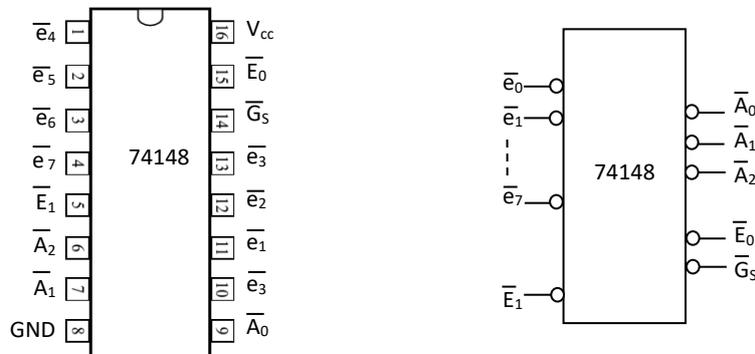


Figure 4.1 : Symbole logique d'un codeur m vers n ($2^{n-1} < m \leq 2^n$).

1.2 Codeur en circuit intégré : le 74148

Le 74148 est un codeur de priorité doté de entrées (e_0, e_1, \dots, e_7) et 3 sorties (A_2, A_1, A_0) qui donnent en binaire l'indice de l'entrée active ou celui le plus élevé si plusieurs entrées sont activées simultanément. Les entrées et les sorties sont actives au niveau bas. De plus, il possède une entrée de validation (\bar{E}_1) et deux sorties de validation (\bar{E}_0 et \bar{G}_S) pour la mise en cascade.

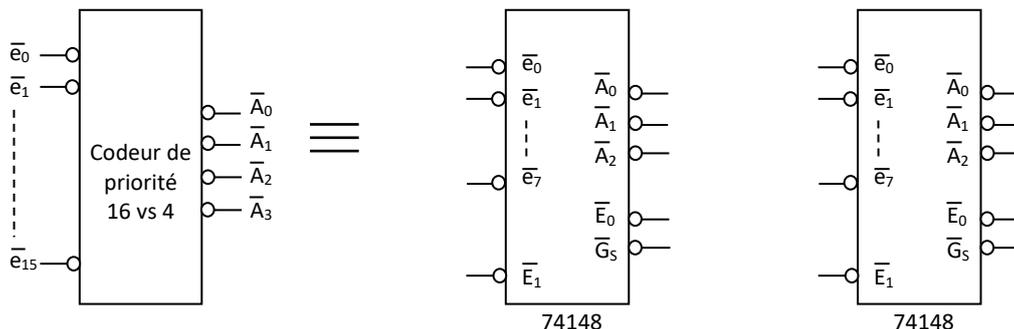


Entrées									Sorties				
\bar{E}_1	\bar{e}_0	\bar{e}_1	\bar{e}_2	\bar{e}_3	\bar{e}_4	\bar{e}_5	\bar{e}_6	\bar{e}_7	\bar{A}_2	\bar{A}_1	\bar{A}_0	\bar{G}_S	\bar{E}_0
1	X	X	X	X	X	X	X	X					
0	1	1	1	1	1	1	1	1					
0	0	1	1	1	1	1	1	1					
0	X	0	1	1	1	1	1	1					
0	X	X	0	1	1	1	1	1					
0	X	X	X	0	1	1	1	1					
0	X	X	X	X	0	1	1	1					
0	X	X	X	X	X	0	1	1					
0	X	X	X	X	X	X	0	1					
0	X	X	X	X	X	X	X	0					

Figure 4.2: Brochage, Symbole logique et Tableau de vérité du 74148.

Travail demandé

- 1) Tester pratiquement le fonctionnement du 74148 et compléter sa table de vérité.
- 2) Compléter le câblage suivant pour avoir un codeur de priorité 16 vers 4 (à ne pas câbler).



2. Le décodeur

2.1. Qu'est-ce qu'un décodeur

C'est un circuit qui délivre en sortie l'équivalent décimal de la combinaison binaire présente en entrée. Parmi les m sorties une seule est activée à la fois. Un décodeur réalise la fonction inverse du codeur.

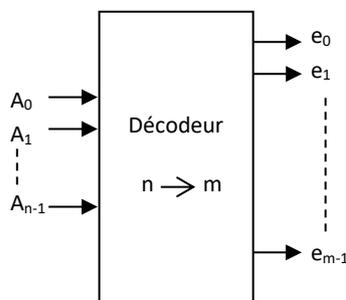
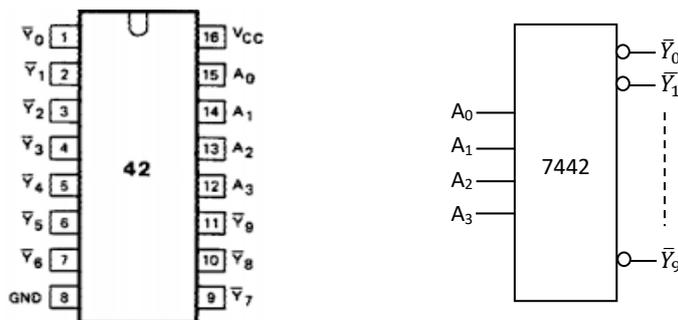


Figure 4.3 : Symbole logique d'un décodeur n vers m ($2^{n-1} < m \leq 2^n$).

2.2. Exemple de décodeur en circuit intégré : le 7442

Le 7442 est un décodeur BCD-décimal doté de 4 bits BCD en entrée et 10 sorties actives au niveau bas.



A_3	A_2	A_1	A_0	\bar{Y}_0	\bar{Y}_1	\bar{Y}_2	\bar{Y}_3	\bar{Y}_4	\bar{Y}_5	\bar{Y}_6	\bar{Y}_7	\bar{Y}_8	\bar{Y}_9
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										

Figure 4.4: Brochage, Symbole logique et Tableau de vérité du 7442

Travail demandé

- 1) Compléter expérimentalement la table de vérité
- 2) Déduire les expressions de chaque sortie :

$$\bar{Y}_0 = \dots\dots\dots \quad \bar{Y}_1 = \dots\dots\dots \quad \dots \quad \bar{Y}_9 = \dots\dots\dots$$

2.3. Décodeur BCD/7 segments : le 7447/7448

Dans de nombreuses applications numériques, les dix chiffres 0 à 9 sont données au moyen d'un afficheur 7 segments. Les segments sont des diodes électroluminescentes (LED).

L'afficheur à 7 segments comporte également un point appelé **DP (Decimal Point)** en anglais) qui n'est autre que l'équivalent de notre virgule décimale.

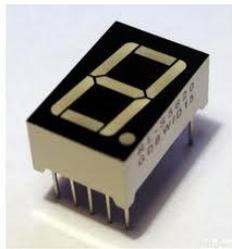


Figure 4.5 : Afficheur 7 segments dans son boîtier

Chaque segment est désigné par une lettre **a, b, c, d, e, f, g** et le point par **D.P.** En commandant convenablement l'allumage et l'extinction des 7 segments, on visualise les nombres désirés.

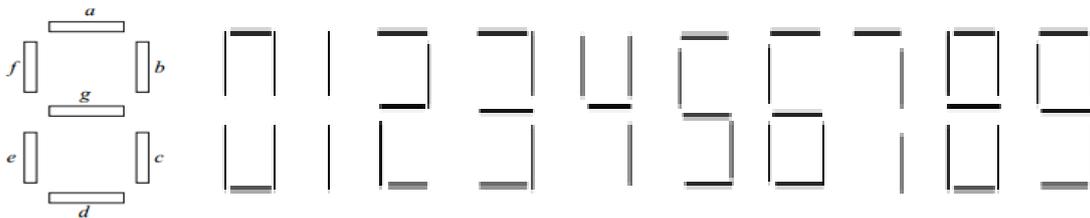


Figure 4.6 : Représentation des 10 chiffres décimaux

Chacun des 7 segments est éclairé par une DEL. Une huitième DEL permet l'éclairement du point décimal. Les afficheurs sont deux types :

- **A cathode commune (CC)** : les cathodes de toutes les LED sont connectées ensemble et reliées à 0 volt. Pour allumer une LED, il faut commander l'anode par Vcc ;
- **A anode commune (AC)** : les anodes de toutes les LED sont connectées ensemble et reliées à Vcc. Pour allumer une LED, il faut commander la cathode par un zéro.

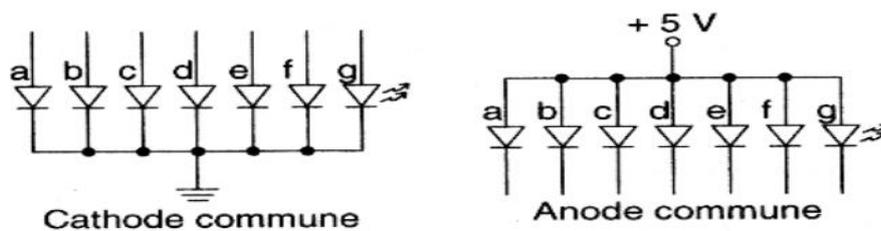


Figure 4.7 : Les types d'afficheurs

Pour utiliser un afficheur 7 segments, il est nécessaire de disposer d'un décodeur qui traduit le code BCD en code d'allumage des segments de l'afficheur. Ce décodeur est appelé décodeur BCD/7 segments.

Il existe plusieurs références de décodeurs BCD/affichage 7 segments. On peut citer :

- Le 7447 : décodeur BCD/7segment pour afficheur AC
- Le 7448 : décodeur BCD/7segment pour afficheur CC

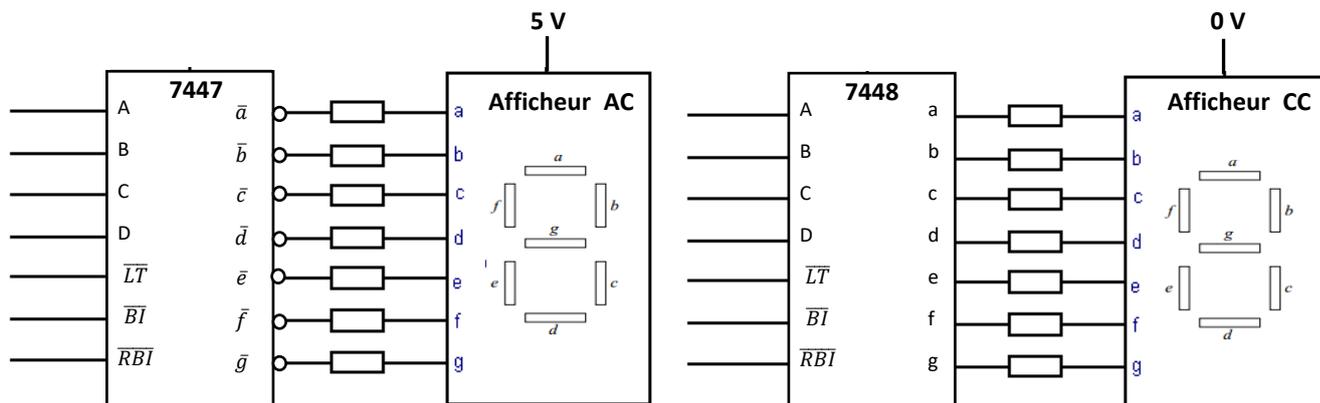


Figure 4.8 : Afficheur 7 segments commandé par un décodeur BCD/7segments.

Dans ce TP, nous allons utiliser le décodeur **7448** commandant un afficheur 7 segments **CC**. Ce décodeur dispose de 7 sorties reliées directement avec un afficheur à anode commune. En entrée, il dispose de 4 entrées BCD qui sont (A,B,C,D) et de 3 entrées supplémentaires qui sont :

- \overline{LT} ou « lamp test » qui permet de vérifier le fonctionnement de l'afficheur en allumant tous les segments si \overline{BI} est à l'état 1.
- $\overline{BI}/\overline{RBO}$ ou « blanking input » qui permet l'effacement des segments de l'afficheur quelque soit l'état des autres entrées.
- \overline{RBI} ou « ripple blanking input » qui permet l'effacement des 0 à gauche si A, B, C, D sont à 0.

Les résistances sont nécessaires pour limiter le courant dans les segments.

Travail demandé

- 1) Câbler le décodeur et l'afficheur comme le montre la figure 4.9. On donne le brochage du 7448 et de l'afficheur à cathode commune.

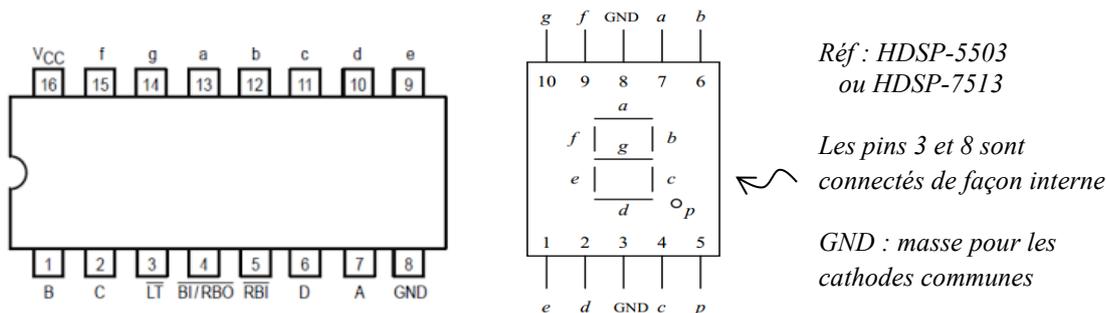


Figure 4.9 : Brochage du 7448 et de l'afficheur à cathode commune.

$\overline{BI}/\overline{RBO} = 1$ et $\overline{RBI} = 1$

- 2) Vérifier si tous les segments de l'afficheur sont en bon état en posant $\overline{LT} = 0$.
- 3) Vérifier l'affichage des 10 chiffres décimaux en posant $\overline{LT} = 1$. Puis faire varier les entrées du code BCD et compléter la table de vérité suivante.

D	C	B	A	Affichage	a	b	c	d	e	f	g
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								

TP N 5 : LE MULTIPLEXEUR ET LE DEMULTIPLEXEUR

Objectif du TP

Vérifier expérimentalement le fonctionnement des multiplexeurs et de démultiplexeur en CI.

1. Le multiplexeur

1.1. Qu'est-ce qu'un multiplexeur

Un multiplexeur (MUX) est un circuit réalisant un aiguillage (recopie) de l'une des données (par la commande des entrées d'adresse) vers une sortie unique. Il y a sélection d'une donnée parmi 2^n (n entrées d'adresse appelées aussi entrées de sélection).

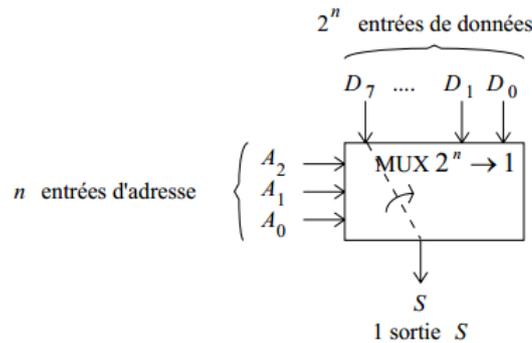


Figure 5.1 : Symbole logique d'un multiplexeur 2^n vers 1.

1.2. Les multiplexeurs en circuits intégrés

Parmi les multiplexeurs disponibles en circuits d'intégration à moyenne échelle MSI, nous citons:

✓ Les multiplexeurs TTL :

- MUX 2 vers 1 : 74157
- MUX 4 vers 1 : 74153
- MUX 8 vers 1 : 74151, 74151
- MUX 16 vers 1 : 74150

✓ Les multiplexeurs CMOS :

- MUX 2 vers 1 : 4019, 4519
- MUX 4 vers 1 : 4539
- MUX 8 vers 1 : 4512

1.3. Exemple de multiplexeur en CI : le 74151

Le 74151 est un multiplexeur 8 vers 1 muni d'une entrée de validation \bar{E} active au niveau bas.

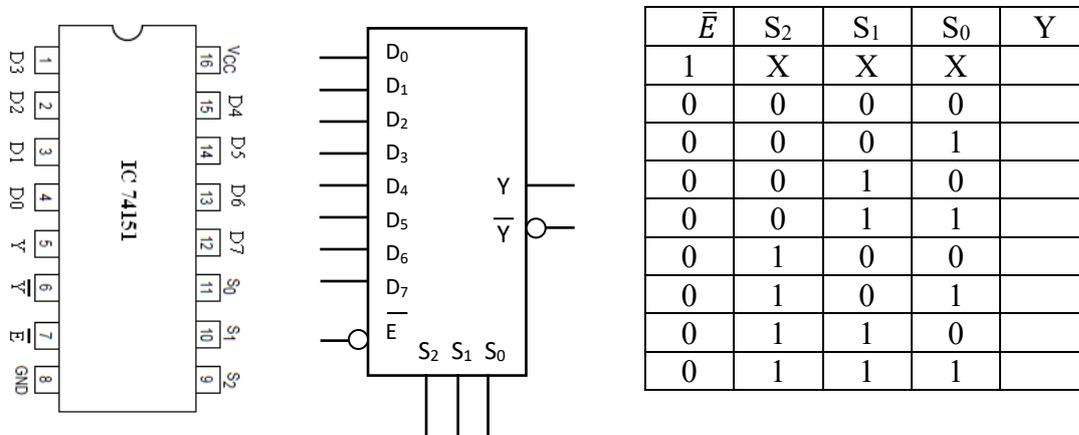


Figure 5.2: Brochage, Symbole logique et Tableau de vérité du 74151.

Travail demandé

- 1) Compléter pratiquement la table de vérité du 74151.
- 2) Déduire l'expression de Y.

Y =

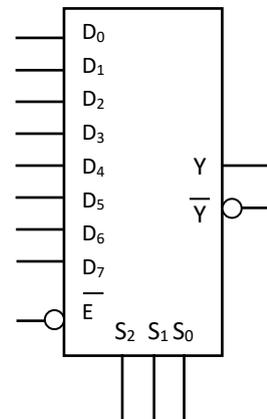
- 3) Soit la fonction $f(a, b, c) = \sum(1,3,4,5)$. Dresser la table de vérité de f.

a	b	c	f
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

- 4) Les multiplexeurs bien qu'inventés pour aiguiller les informations, peuvent réaliser des fonctions logiques. Ils permettent de remplacer des logigrammes complexes et sans même qu'il est nécessaire de simplifier préalablement ces fonctions.

Nous disposons d'un seul multiplexeur 74 LS 151. Compléter le schéma ci-dessous, en proposant un câblage de ce circuit permettant de réaliser la fonction logique décrite précédemment.

.....



2. Le démultiplexeur

2.1. Qu'est-ce qu'un démultiplexeur

Un démultiplexeur (DEMUX) effectue la fonction inverse d'un MUX. Il aiguille une donnée sur une parmi 2^n sorties (n entrées d'adresse).

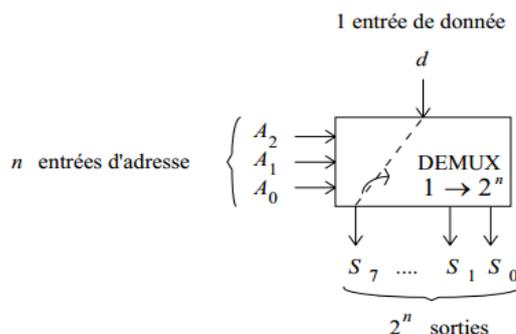


Figure 5.3 : Symbole logique d'un démultiplexeur 1 vers 2^n .

2.2 Le démultiplexeur en circuits intégrés

Parmi les démultiplexeurs disponibles en CI à moyenne échelle MSI, nous citons :

✓ Les démultiplexeurs TTL :

- 2 DEMUX 1 vers 4 : 74139
- DEMUX 1 vers 8 : 74138
- DEMUX 1 vers 16 : 74154

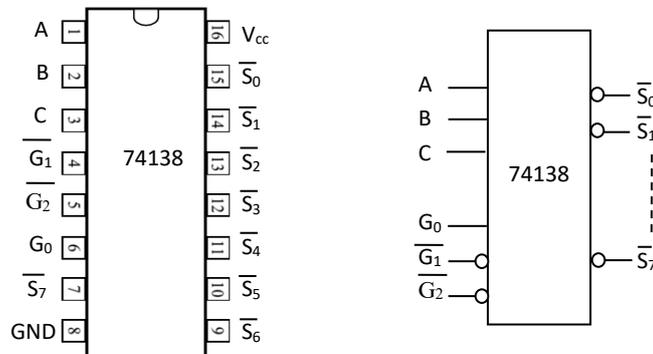
✓ Les démultiplexeurs CMOS :

- 2 DEMUX 1 vers 4 : 4555
- DEMUX 1 vers 16 : 4514

2.3. Exemple de démultiplexeur en CI : le 74138

Le circuit 74 LS 138 est appelé décodeur/démultiplexeur 1 parmi 8. Il comporte :

- ✓ 8 sorties de données S_0 à S_7
- ✓ 3 entrées de sélection A, B et C
- ✓ 3 entrées de validation G_0 , \overline{G}_1 et \overline{G}_2



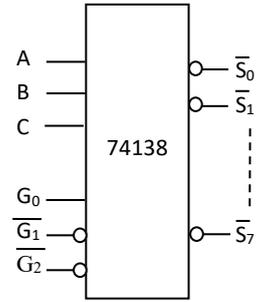
Entrées						Sorties							
G_0	\overline{G}_1	\overline{G}_2	C	B	A	\overline{S}_0	\overline{S}_1	\overline{S}_2	\overline{S}_3	\overline{S}_4	\overline{S}_5	\overline{S}_6	\overline{S}_7
0	X	X	X	X	X								
X	1	X	X	X	X								
X	X	1	X	X	X								
1	0	0	0	0	0								
1	0	0	0	0	1								
1	0	0	0	1	0								
1	0	0	0	1	1								
1	0	0	1	0	0								
1	0	0	1	0	1								
1	0	0	1	1	0								
1	0	0	1	1	1								

Figure 5.4: Brochage, Symbole logique et Tableau de vérité du 74138

Travail demandé

- 1) Compléter expérimentalement la table de vérité
- 2) Comment peut-on utiliser ce CI comme démultiplexeur 1 vers 8 ?

D : entrée Donnée



3) Utiliser le 74138 et les portes logiques adéquates pour réaliser la fonction logique décrite plus haut.

.....

.....

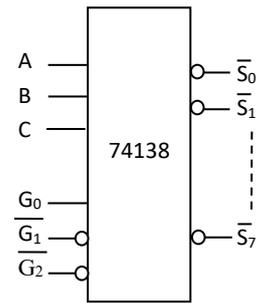
.....

.....

.....

.....

.....



TP N 6 : LES BASCULES

Objectif du TP

- ✓ Se familiariser avec les différents modes de fonctionnement des bascules,
- ✓ Réaliser la bascule T à l'aide des bascules JK et D.

1. Qu'est-ce qu'une bascule ?

La bascule (ou flip-flop en anglais) est le circuit séquentiel le plus simple. Son rôle consiste à enregistrer une information fugitive et à conserver cet état lorsque l'information disparaît.

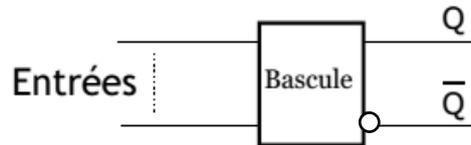


Figure 6.1 : Symbole logique d'une bascule.

La sortie Q est la sortie « normale » de la bascule, tandis que la sortie \overline{Q} est la sortie complémentaire. Lorsqu'on fait référence à l'état d'une bascule, on considère l'état de la sortie Q.

Les bascules peuvent être **asynchrones** ou **synchrones**.

a) Bascules asynchrones

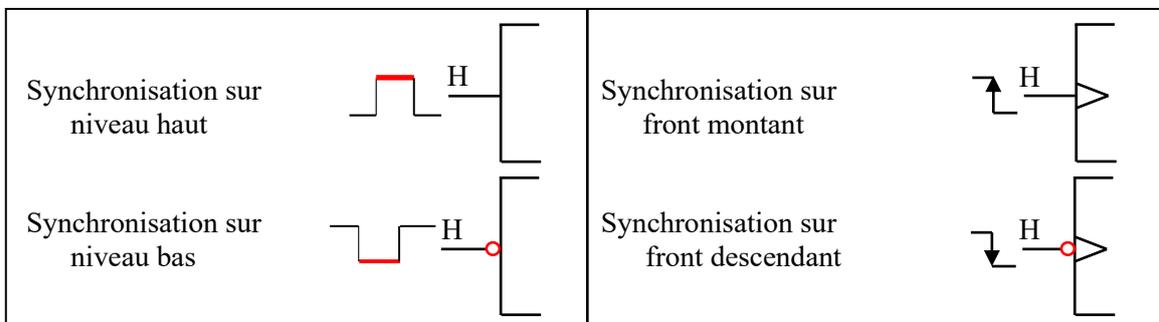
Une bascule est dite *asynchrone* si l'état des sorties change à des instants quelconques. Ainsi une modification des variables d'entrées entraîne la bascule à un nouvel état.

b) Bascules synchrones

Une bascule est dite *synchrone* si les modifications des variables de sorties ne peuvent s'effectuer qu'à des instants bien déterminés. Ces instants peuvent être fixés par un signal de synchronisation appelé horloge.

Les bascules synchrones fonctionnent selon l'un des deux modes de synchronisation suivants :

- Synchronisation sur un niveau d'horloge « **latch flip-flop** » : Il suffit d'appliquer le niveau convenable (haut ou bas), appelé niveau actif pour que la sortie de bascule puisse changer d'état.
- Synchronisation sur un front d'horloge : En anglais « **edge triggered flip-flop** ». Les modifications en sortie ne peuvent s'effectuer qu'aux moments des fronts de l'horloge (montants ou descendants).



2. Exemples de bascules en CI

2.1. La bascule D « latch » : 7475 (2 double bascules D)

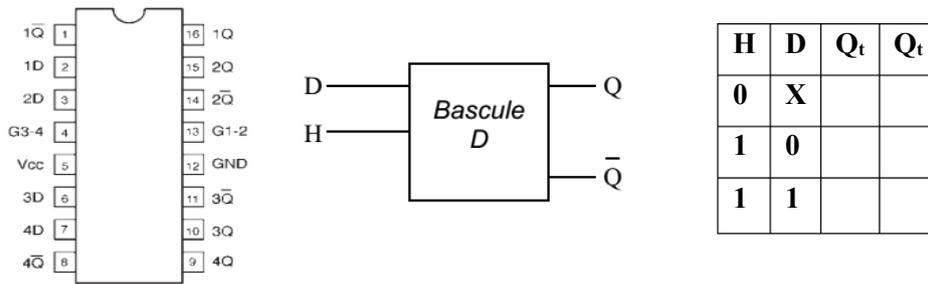
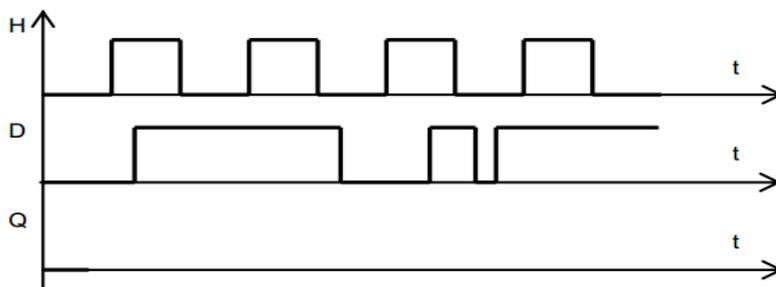


Figure 6.2 : Brochage, Symbole logique et Table de vérité du 7475

1) Choisir une bascule et compléter les chronogrammes suivants.



2) Déduire la table de vérité.

3) Sur quel niveau d'horloge cette bascule est synchrone ?

2.2. La bascule D « edge triggered flip-flop » : 7474

Le 7474 contient 2 bascules D. Chaque bascule est déclenchée sur front montant avec deux entrées asynchrones actives au niveau bas : une remise à zéro $\overline{\text{CLR}}$ et une remise à un $\overline{\text{PR}}$.

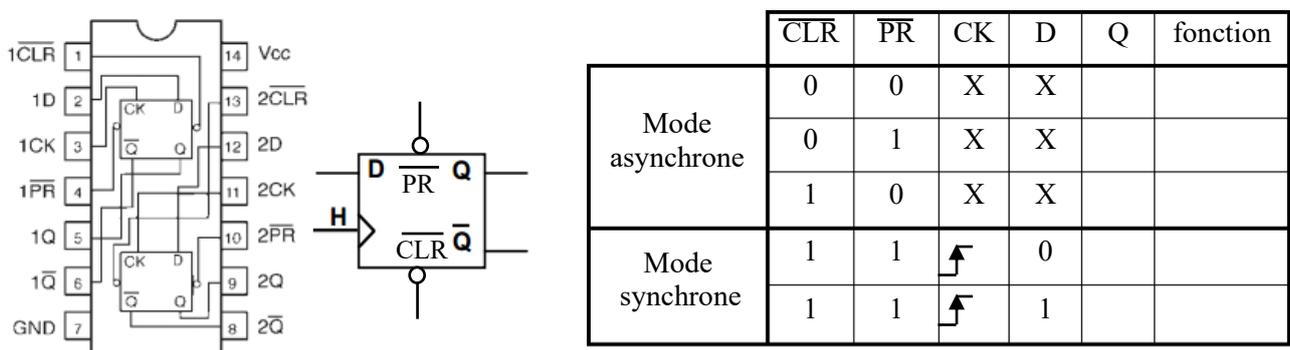
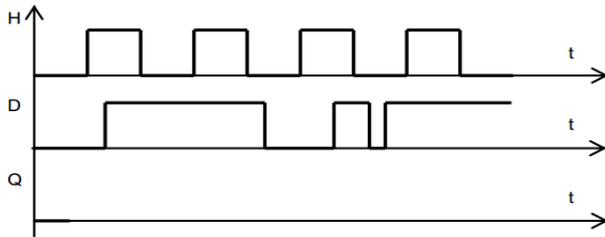


Figure 6.3 : Brochage, Symbole logique et Table de vérité du 7474

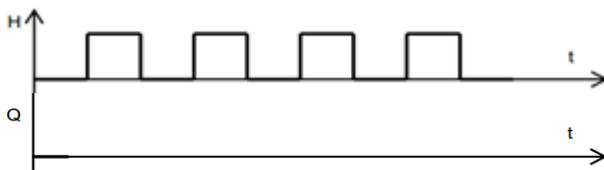
1) Choisir une bascule D du CI 7474 et compléter les chronogrammes suivants.

Mode synchrone : $\overline{CLR}=1, \overline{PR}=1$

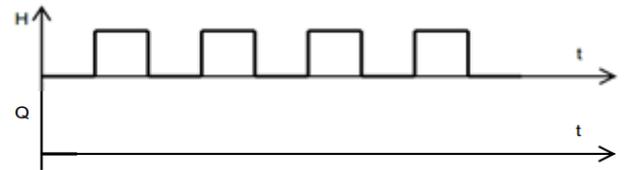


Mode asynchrone :

➤ $\overline{CLR}=0, \overline{PR}=1, D$ variable

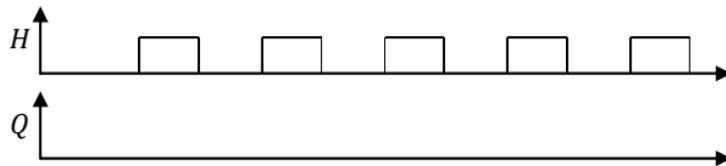
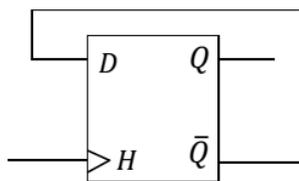


➤ $\overline{CLR}=1, \overline{PR}=0, D$ variable



2) Dédurre la table de vérité.

3) Réaliser le câblage suivant et compléter le chronogramme. Quelle est la fonction réalisée ?.....



2.3. La bascule JK « edge triggered flip-flop » : 74112

Le 74112 contient 2 bascules JK. Chaque bascule est déclenchée sur front descendant avec deux entrées asynchrones actives au niveau bas : une remise à zéro \overline{CLR} et une remise à un \overline{PR} .

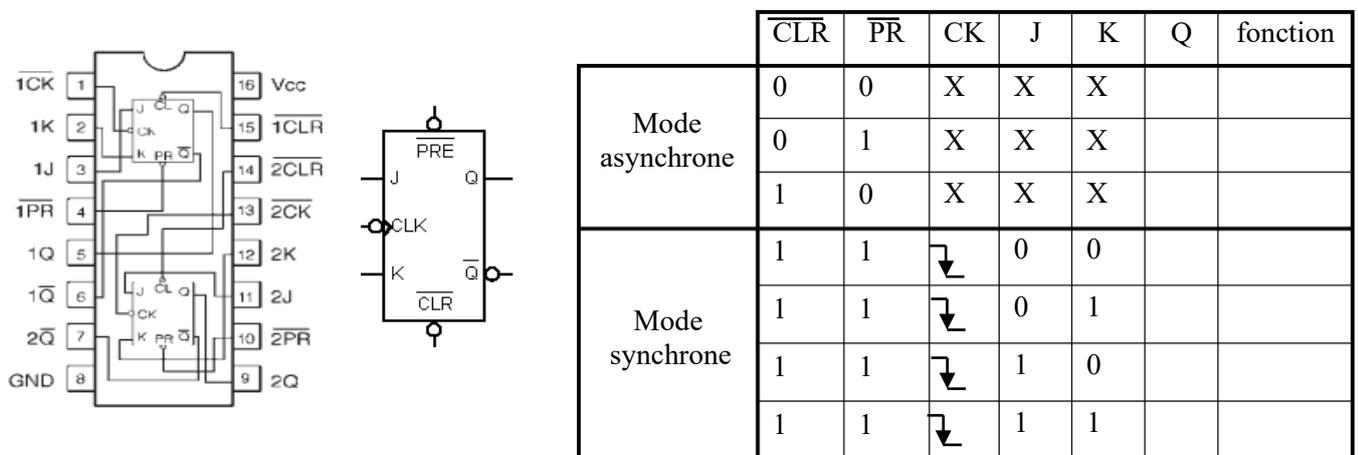
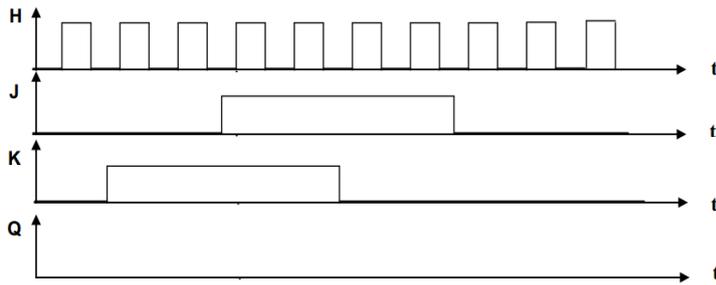


Figure 6.4 : Brochage, Symbole logique et Table de vérité du 7474.

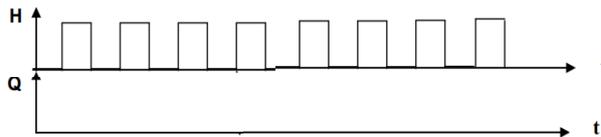
1) Choisir une bascule et compléter les chronogrammes suivants.

Mode synchrone : $\overline{CLR}=1, \overline{PR}=1$

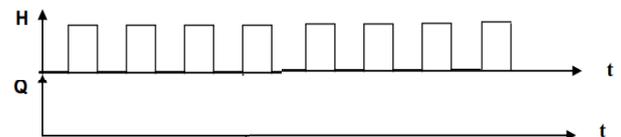


Mode asynchrone :

➤ $\overline{CLR}=0, \overline{PR}=1$, Jet K variables

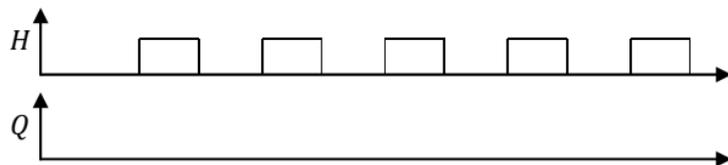
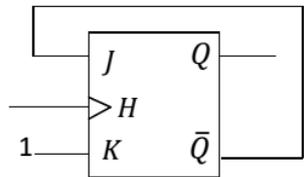
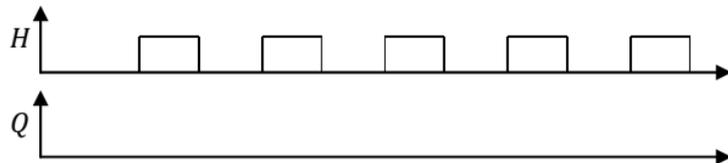
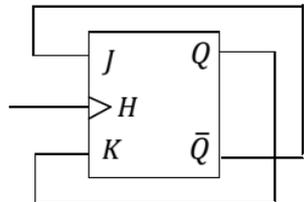
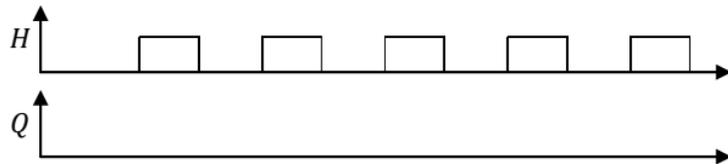
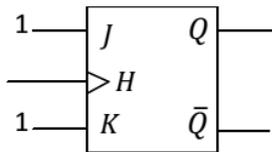


➤ $\overline{CLR}=1, \overline{PR}=0$, Jet K variables



2) D duire la table de v rit .

3) R aliser les c blages suivants et visualiser H et Q sur un oscilloscope. Quelle est la fonction r alis e ?.....



TP N 7 : LES COMPTEURS ASYNCHRONES

Objectif du TP

Vérifier expérimentalement le fonctionnement des compteurs asynchrones cycle complet et incomplet en utilisant au premier lieu les bascules (JK 74112 et D 7474) puis les compteurs asynchrones en CI (exemple le 7493).

1. Qu'est ce qu'un compteur asynchrone

Un compteur asynchrone est constitué de bascules JK (ou de bascules D) montées en diviseur de fréquence par deux ; c-à-d ces bascules fonctionnent en mode « basculement ». Il est dit *asynchrone* car ses bascules ne changent pas d'état simultanément car elles n'utilisent *pas le même signal d'horloge*. Un compteur asynchrone à partir du signal d'horloge placé sur la bascule. L'entrée d'horloge de la 2ème bascule reçoit de la bascule etc..

- compléter le câblage des deux montages suivants.

Front descendant :



Front montant :

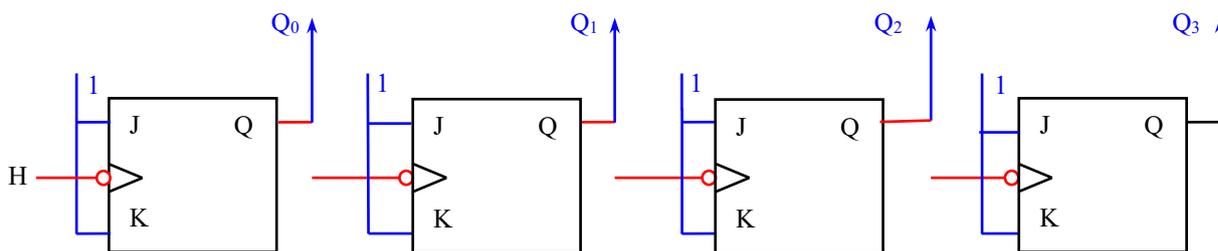


2. Manipulation 1 : Compteurs asynchrones à base de bascules JK

2.1 Compteur asynchrone à cycle complet (exemple : modulo 16)

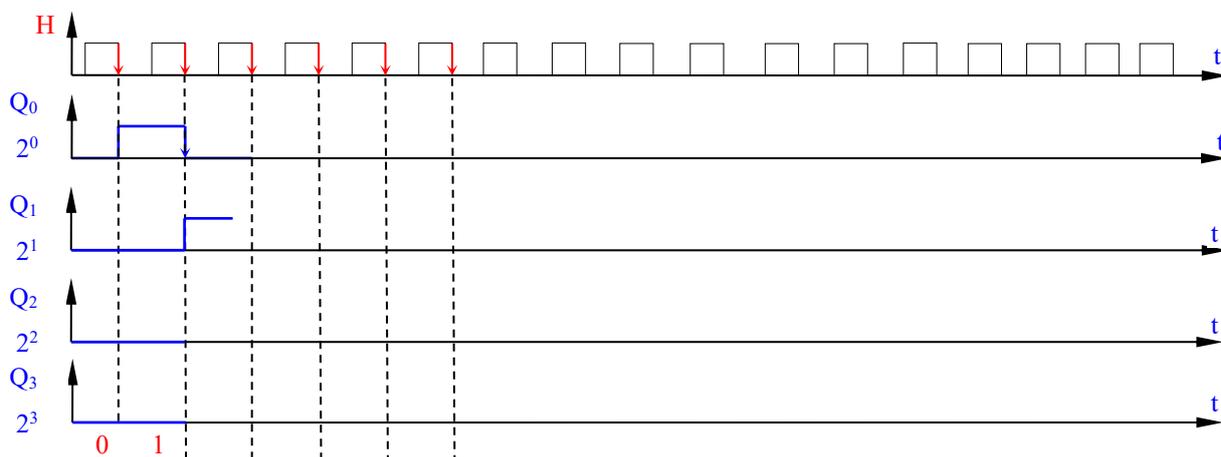
Le nombre de bascules n est déterminé par : $2^n \geq N$, avec N du compteur. $N=16=2^4 \Rightarrow n=4$

- 1) Utiliser des bascules JK (74112) et compléter le câblage suivant pour réaliser un compteur asynchrone modulo 16.



Le signal horloge est de fréquence $f_H = 0.2 \text{ Hz}$ provient du GBF. $\overline{\text{CLR}} = \overline{\text{PRE}} = 1$.

- 2) Compléter les chronogrammes suivants.

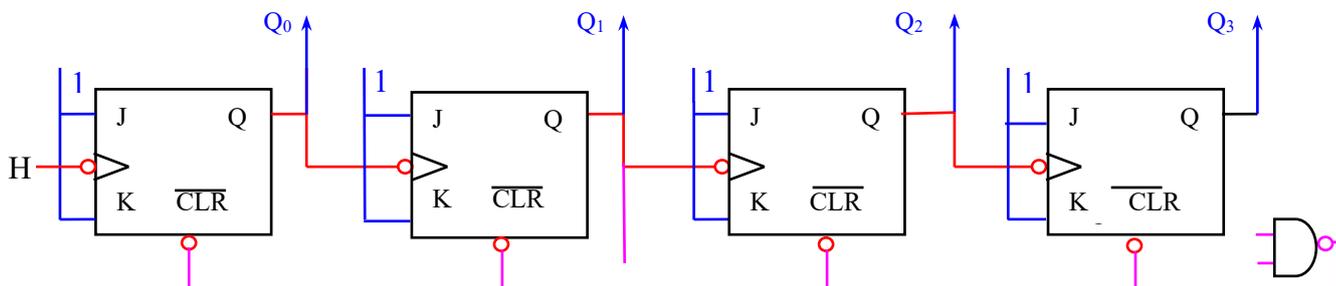


- 3) Au lieu de prendre les Q_i comme sorties prener les $\overline{Q_i}$ et observer la succession des 4 sorties.
 Quel est le role du montage ? C'est un modulo

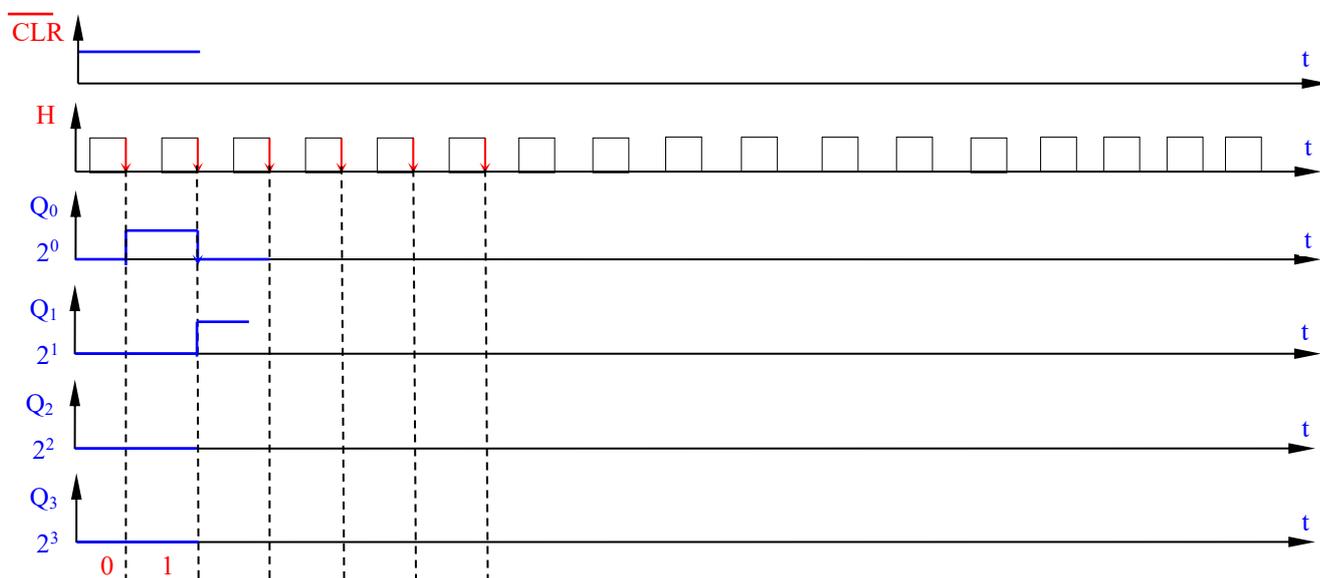
2.2 Compteur asynchrone à cycle incomplet (exemple : compteur modulo 10)

Le cycle de ce compteur est : 0, 1,9. Dès l'apparition de l'état 10, on doit initialiser toutes les bascules : $CLR=Q_3\overline{Q_2}Q_1\overline{Q_0}$. On remarque que $Q_1 = Q_3 = 1$ n'apparaît pour la 1ère fois que pour la 10^{ème} combinaison. L'équation se réduit à : $CLR=Q_3Q_1$. Donc $\overline{CLR} = \overline{Q_3Q_1}$ (Remise à zéro active au niveau bas).

- 1) Garder le montage du compteur précédent et arranger le de telle sorte à réaliser un compteur modulo 10.



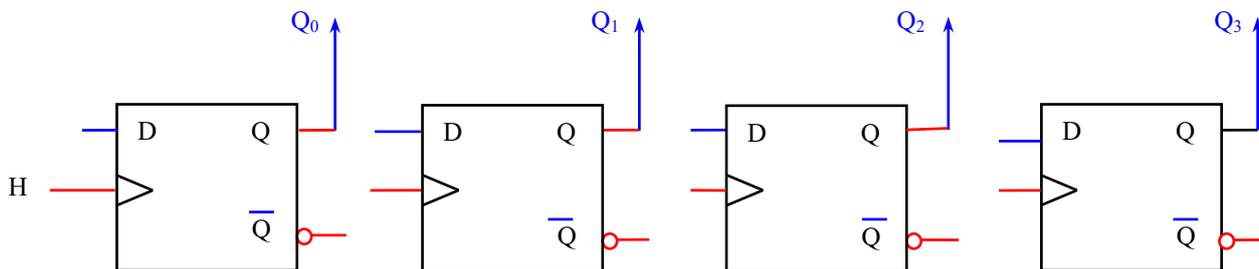
- 2) Compléter les chronogrammes suivants.



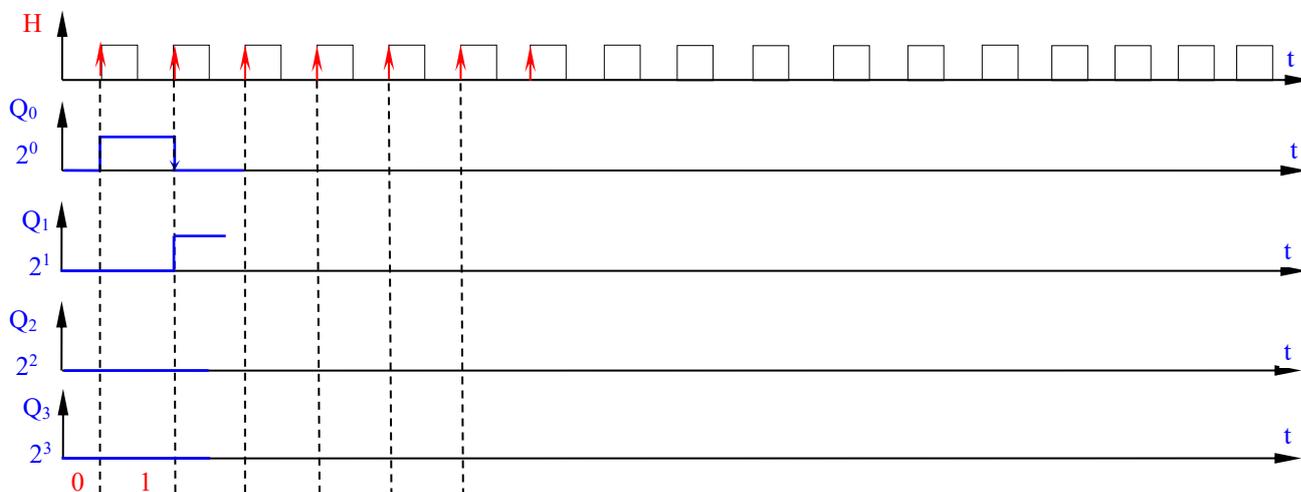
3. Manipulation 2 : Compteurs asynchrones à base de bascules D

3.1 Compteur asynchrone à cycle complet (exemple : compteur modulo 16)

1) Utiliser des bascules D (7474) et compléter le câblage suivant pour réaliser un compteur asynchrone modulo 16.



2) Compléter les chronogrammes suivants.

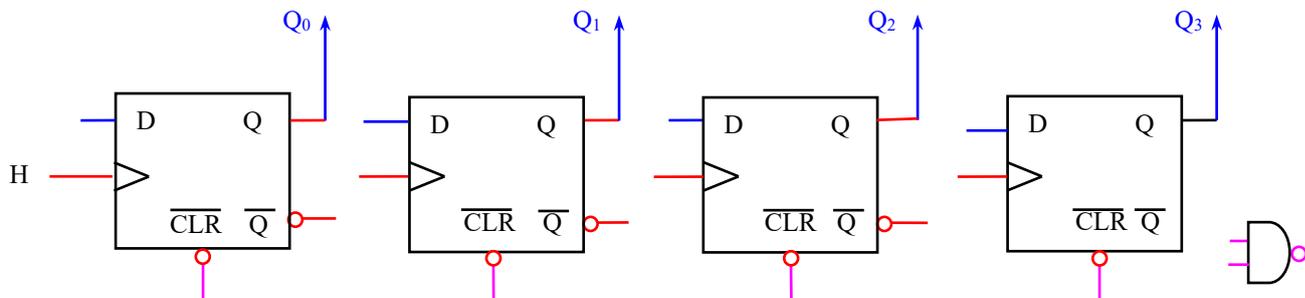


3) Au lieu de prendre les Q_i comme sorties prendre les \overline{Q}_i et observer la succession des 4 sorties.

Quel est le rôle du montage ? C'est un modulo (.....)

3.2 Compteur asynchrone à cycle incomplet : (exemple : compteur modulo 12)

1) Le cycle de ce compteur est : 0, 1, Compléter le câblage suivant pour réaliser un compteur modulo 12.



2) Compléter les chronogrammes.



4. Manipulation 3 : Compteur asynchrone en CI (7493)

Le 7493 est un compteur asynchrone à deux entrées horloge, deux entrées de mise à zéro et quatre sorties.

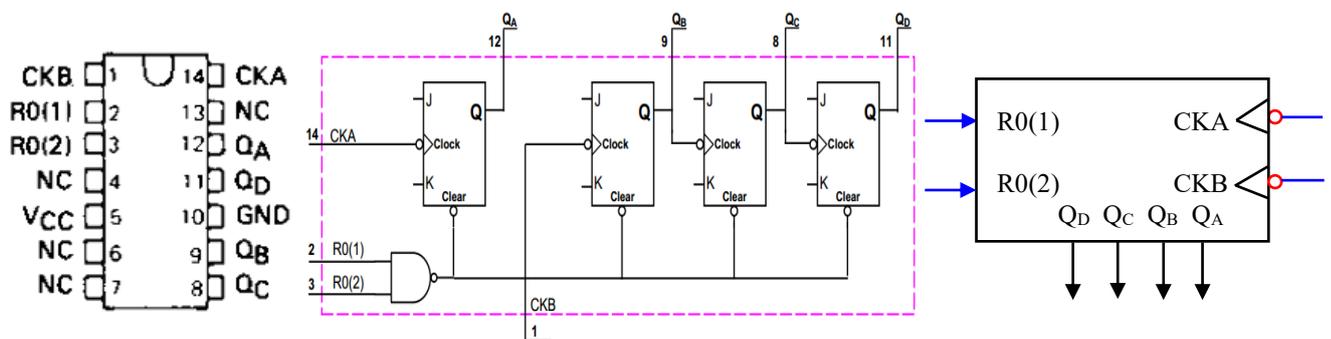
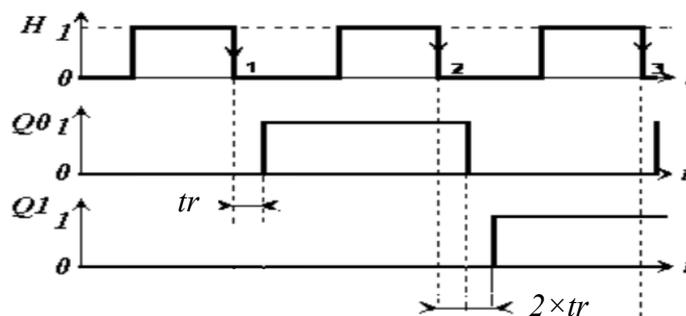


Figure 7.1 : brochage, structure interne et symbole logique du 7493.

N.B. Les entrées J et K sont à l'état logique 1 une fois que le circuit est alimenté.

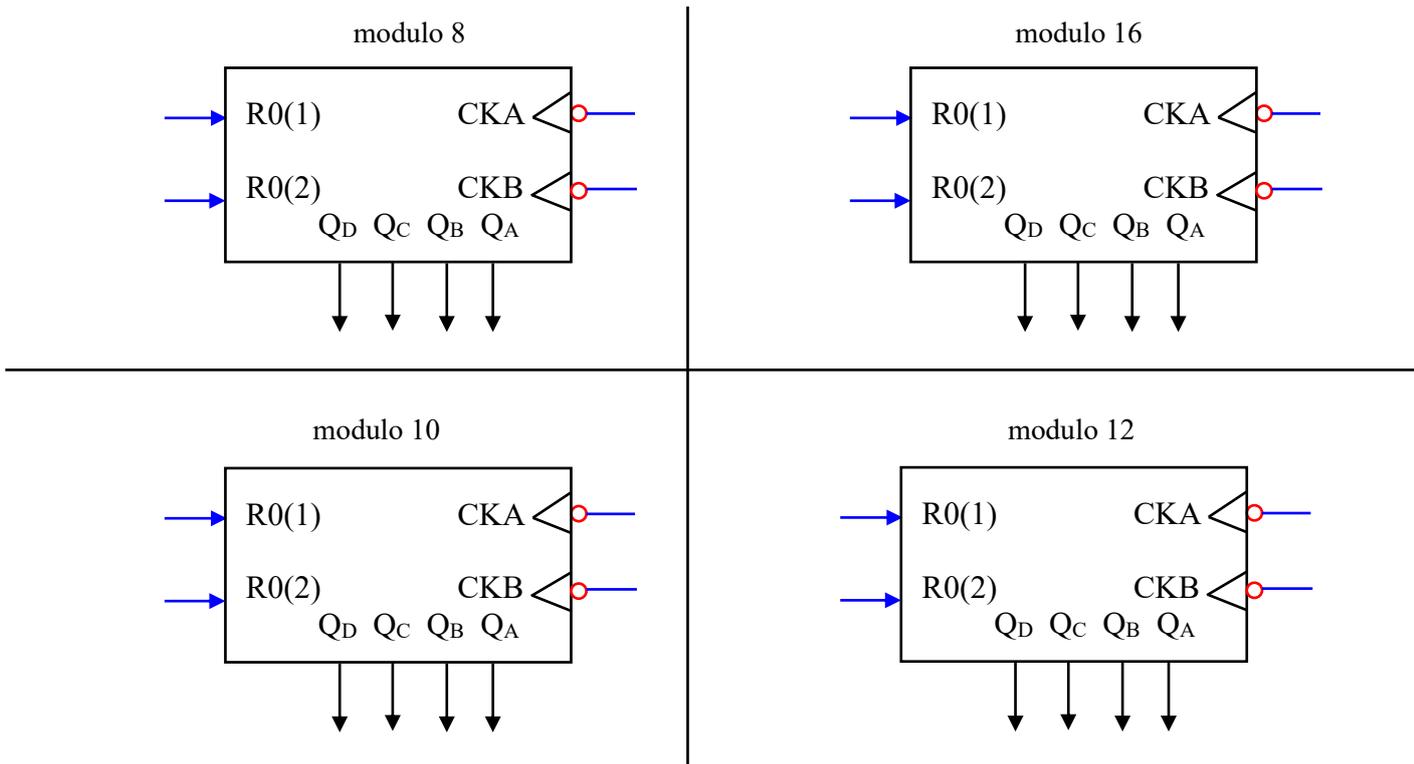
Utiliser un 7493 et dans chaque cas compléter les câblages nécessaires pour réaliser un compteur asynchrone.

Remarque :



On remarque que plus le nombre de bascules est grand, plus le nombre de retard introduit du fait du fonctionnement asynchrone est grand. Pour n étages \Rightarrow temps de retard = $n \times tr$. On constate que pour un compteur à 4 étages, aucun prélèvement de l'état du compteur ne sera possible si la période T de l'horloge n'est pas supérieure à $4tr$ (tr de l'ordre de dizaines de ns).

Par conséquent, les compteurs asynchrones ne conviennent pas pour les hautes fréquences.



TP N 8 : LES COMPTEURS SYNCHRONES

Objectif du TP

Vérifier expérimentalement le fonctionnement des compteurs synchrones en utilisant au premier lieu les bascules (D : 7474) puis les compteurs synchrones en CI (exemple : le 74193).

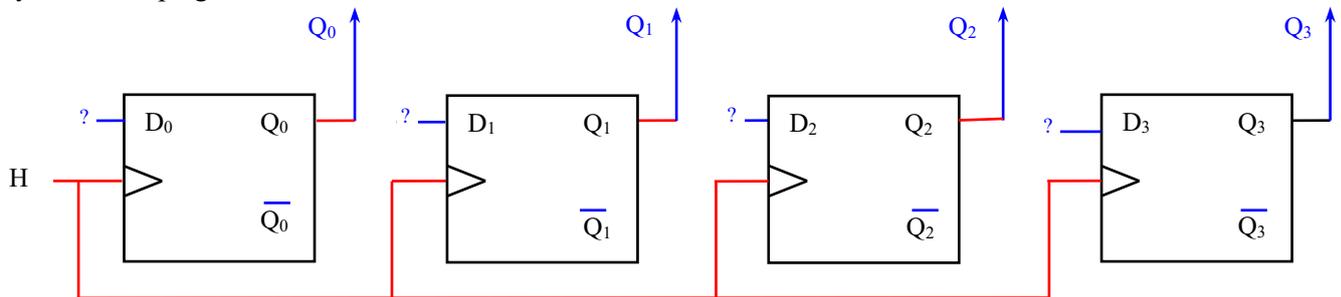
1. Qu'est ce qu'un compteur synchrone ?

L'inconvénient des compteurs *asynchrones* réside dans l'accumulation des retards de basculement d'une bascule à une autre. Pour surmonter ce problème, les compteurs *synchrones* ont été développés. Dans un compteur synchrone, toutes les bascules partagent le même signal d'horloge, ce qui permet de commuter simultanément. Cela améliore la vitesse de fonctionnement (vitesse plus rapide).

2. Compteur synchrone BCD à base de bascules D

2.1. Préparation théorique

Cycle de comptage : 0 – 1 – 2 – ...9, Nombre de bascules : 4



Pour déterminer les entrées des bascules suivons les étapes suivantes :

1) Table de transition de la bascule D

Q_t	Q_{t+1}	D
0	0	
0	1	
1	0	
1	1	

2) Table d'excitation des bascules

déc	t				t+1				Entrées des bascules			
	Q_3	Q_2	Q_1	Q_0	Q_3	Q_2	Q_1	Q_0	D_3	D_2	D_1	D_0
0	0	0	0	0	0	0	0	1				
1	0	0	0	1	0	0	1	0				
2	0	0	1	0	0	0	1	1				
3	0	0	1	1	0	1	0	0				
4	0	1	0	0	0	1	0	1				
5	0	1	0	1	0	1	1	0				
6	0	1	1	0	0	1	1	1				
7	0	1	1	1	1	0	0	0				
8	1	0	0	0	1	0	0	1				
9	1	0	0	1	0	0	0	0				

3) Expressions simplifiées des entrées des bascules

$Q_3Q_2 \backslash Q_1Q_0$	00	01	11	10
00				
01				
11				
10				

$D_0 = \dots\dots\dots$

$Q_3Q_2 \backslash Q_1Q_0$	00	01	11	10
00				
01				
11				
10				

$D_1 = \dots\dots\dots$

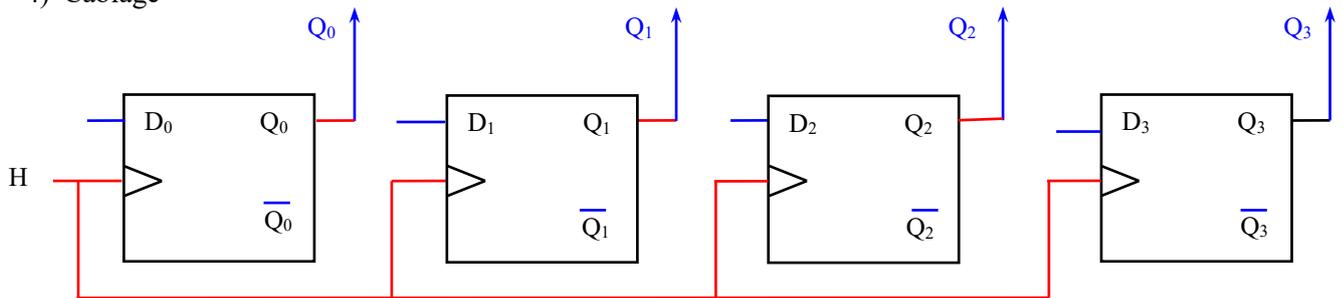
$Q_3Q_2 \backslash Q_1Q_0$	00	01	11	10
00				
01				
11				
10				

$D_2 = \dots\dots\dots$

$Q_3Q_2 \backslash Q_1Q_0$	00	01	11	10
00				
01				
11				
10				

$D_3 = \dots\dots\dots$

4) Câblage



2.2. Manipulation

- 1) Réaliser le montage en utilisant des bascules D (7474) et les portes adéquates. Vérifier le fonctionnement du compteur.
- 2) Les compteurs *synchrones* corrigent le problème de retard cumulé dans les compteurs *asynchrones*, au détriment d'un inconvénient. Lequel ?

3. Exemple de compteurs en CI : le 74193

Le 74XX193 est un compteur/décompteur synchrone binaire sur 4 bits, à prépositionnement parallèle et avec RAZ asynchrone.

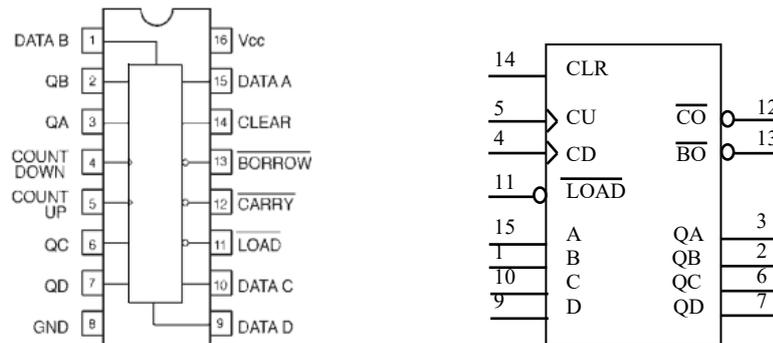


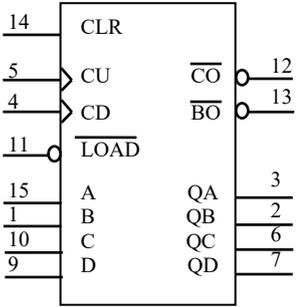
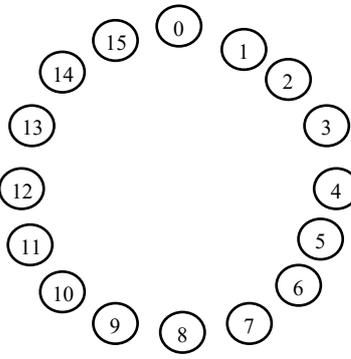
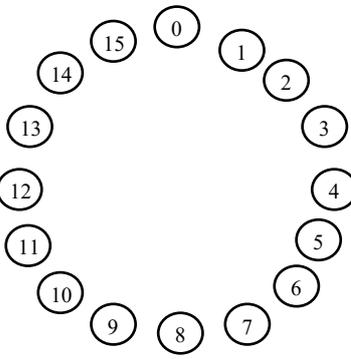
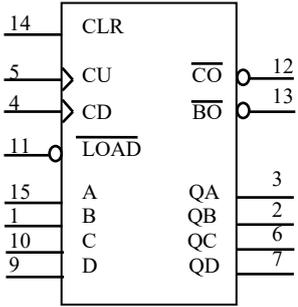
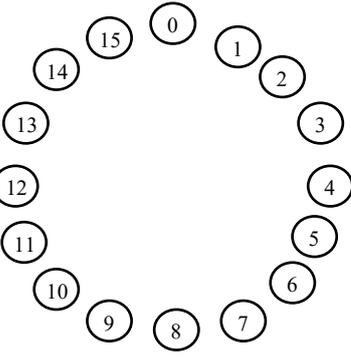
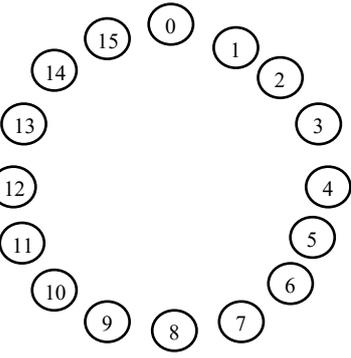
Figure 8.1: Brochage et Symbole logique du 74193

Examiner la table de fonctionnement de ce compteur.

Pattes	Description
A, B, C, D	Pattes d'entrées pour prépositionnement parallèle.
$\overline{\text{LOAD}}$	Patte de prépositionnement asynchrone, dès que cette patte passe à l'état bas, les sorties sont égales aux entrées, et cela jusqu'au passage à 1 de LOAD. Le cycle de comptage/décomptage ne commence que si $\text{LOAD} = 1$.
CU	Count Up : Horloge du compteur, active sur front montant
CD	Count Down : Horloge du décompteur, active sur front montant
CLR	Clear): Mise à zéro du compteur active au niveau bas
QA, QB, QC, QD	Sorties du compteur
$\overline{\text{RO}}$	Ripple Clock Output : Cette patte est mise à l'état bas si la valeur du compteur vaut 15. Elle sert d'horloge en cascade du compteur
$\overline{\text{BO}}$	Bower Clock Output : Cette patte est mise à l'état bas si la valeur du décompteur vaut 0. Elle sert d'horloge en cascade du décompteur.

Manipulation : Dans chacun des cas suivants, effectuer les câblages nécessaires autour du circuit 74193. Penser à positionner les entrées A, B, C et D à différentes valeurs. Valider chaque structure par un graphe des états.

Compteur synchrone modulo 16	Graphe des états	
	Débutant par	Débutant par
Compteur synchrone modulo 10	Graphe des états	
	Débutant par	Débutant par

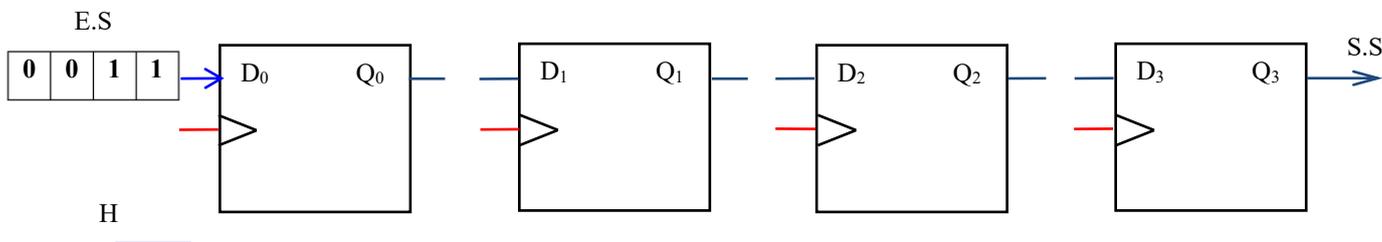
Décompteur synchrone modulo 16	Graphe des états	
	Débutant par	Débutant par
		
Décompteur synchrone modulo 12	Graphe des états	
	Débutant par	Débutant par
		

Le tableau montre que l'information appliquée à l'entrée est au fur et à mesure (au rythme des coups d'horloge) de à Elle est entièrement disponible sur les sorties Q_0 Q_1 Q_2 Q_3 aprèscoups d'horloge.

Application : Ce type de registre permet de transformer un codage temporel (succession des bits dans le temps) en un codage spatial (information stockée en mémoire).

2.2. Registre à décalage entrée série–sortie série

1) Pour réaliser un registre à décalage 4 bits de type : Entrée Série (E.S) – Sortie Série (S.S), garder le câblage précédent et récupérer la sortie série S.S de la dernière bascule Q_3 .



2) Compléter la table de fonctionnement suivante.

E.S	H	Q ₀	Q ₁	Q ₂	Q ₃	← Initialisation
		0	0	0	0	
DONNÉE { 1 1 0 0 0 0 0						
						SORTIE }

Le tableau montre que l'information appliquée à l'entrée E.S est disponible sur la sortie S.S bit après bit à partir ducoups d'horloge. Pour compléter le décalage, il faut coups d'horloge.

3. Application : Compteur et Générateur de signaux

➤ Le compteur de Johnson

Le registre à décalage entrée série-sortie série est toujours bouclé sur lui-même. Dans le cas du compteur de Johnson (voir figure 9.1), la sortie \overline{Q} de la dernière bascule est reliée à l'entrée série de la première bascule.

1) Réaliser le câblage suivant et visualiser les sorties au moyen des LEDs.

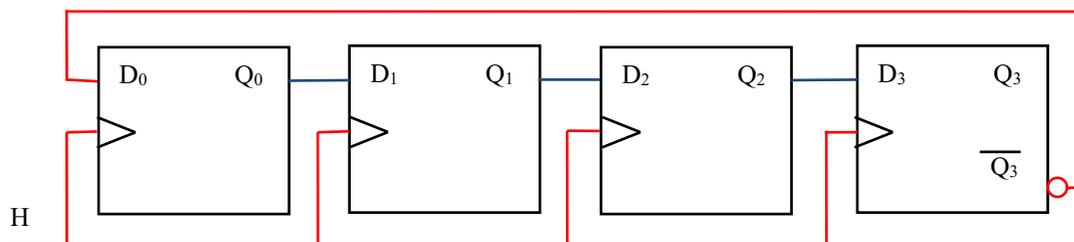
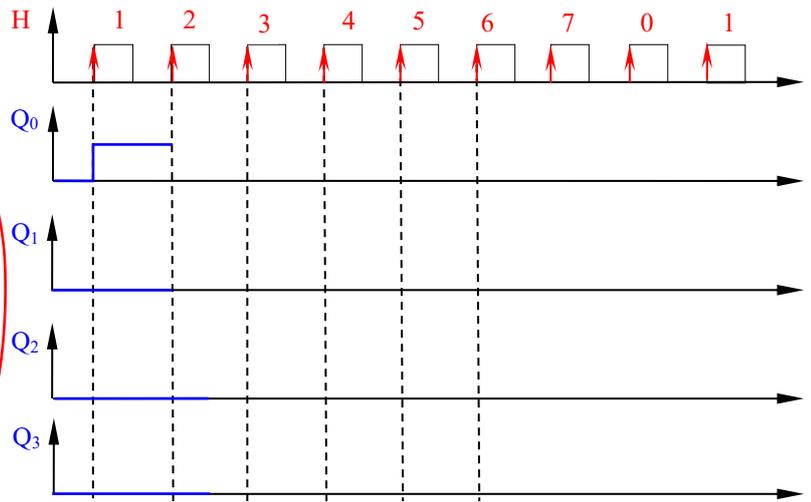


Figure 9.1: Montage de principe du compteur de Johnson.

2) Compléter la table de fonctionnement suivante.

H	Q ₀	Q ₁	Q ₂	Q ₃
	0	0	0	0
1				
2				
3				
4				
5				
6				
7				

3) Compléter les chronogrammes suivants.



Ce compteur de 4 bascules présente seulement états différents.

Le compteur de Johnson peut jouer le rôle de **générateur de signaux**.

4. Les registres à décalage en circuits intégrés

74164 : Registre à décalage (à droite) de 8 bits avec entrée en série et sortie en parallèle (SIPO),

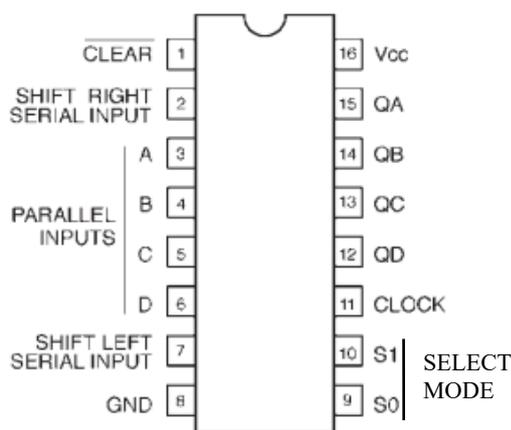
74165 : Registre à décalage (à droite) de 8 bits avec entrée série ou parallèle et sortie en série (SISO et PISO),

74195 : Registre à décalage (à droite) de 4 bits avec entrée en parallèle et sortie en parallèle (PIPO).

74194 : Registre à décalage bidirectionnel de 4 bits possédant les fonctions suivantes : chargement //, entrée série, sortie //, décalage à droite et à gauche.

5. Exemple de registres en CI : le 74194

On va s'intéresser au registre bidirectionnel **74194** qui regroupe les fonctions des autres registres. Le 74194 est appelé registre universel.



S1	S0	Mode
0	0	inhibition de l'horloge
0	1	décalage à droite
1	0	décalage à gauche
1	1	chargement parallèle

Figure 9.2: Brochage et table de sélection des modes 74194.

1) Câbler puis vérifier le fonctionnement du circuit 74194, en remplissant la table vérité suivante.

ENTREES										SORTIES			
Clr	MODE		Clk	SERIE		PARALLELE				Qa	Qb	Qc	Qd
	S1	S0		SL	SR	A	B	C	D				
0	X	X	X	X	X	X	X	X	X				
1	X	X	0	X	X	X	X	X	X				
1	1	1		X	X	a	b	c	d				
1	0	1		X	1	X	X	X	X				
1	0	1		X	1	X	X	X	X				
1	0	1		X	0	X	X	X	X				
1	0	1		X	0	X	X	X	X				
1	1	0		1	X	X	X	X	X				
1	1	0		1	X	X	X	X	X				
1	1	0		0	X	X	X	X	X				
1	1	0		0	X	X	X	X	X				
1	0	0	X	X	X	X	X	X	X				

2) *Chargement parallèle – sortie série (décalage à droite)*

Charger la valeur binaire « 0100 », puis effectuer 2 décalages à droite (SR étant mis à 0)

ENTREES										SORTIES			
Clr	MODE		Clk	SERIE		PARALLELE				Qa	Qb	Qc	Qd
	S1	S0		SL	SR	A	B	C	D				

Après combien de coups d'horloge l'information est entièrement récupérée ?

Quelle est la valeur affichée après un décalage ?

Quelle est l'opération arithmétique binaire ainsi réalisée ?

3) *Chargement parallèle – sortie série (décalage à gauche)*

Charger la valeur binaire « 0010 », puis effectuer 2 décalages à gauche (SL étant mis à 0)

ENTREES										SORTIES			
Clr	MODE		Clk	SERIE		PARALLELE				Qa	Qb	Qc	Qd
	S1	S0		SL	SR	A	B	C	D				

Après combien de coups d'horloge l'information est entièrement récupérée ?

Quelle est la valeur affichée après un décalage ?

Quelle est l'opération arithmétique binaire ainsi réalisée ?

ANNEXE : BROCHAGE DES CIRCUITS INTEGRES LOGIQUES TTL

