

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE
SCIENTIFIQUE
UNIVERSITÉ 8 MAI 1945 GUELMA
FACULTÉ DES SCIENCES ET DE LA TECHNOLOGIE
DÉPARTEMENT D'ÉLECTRONIQUE ET TÉLÉCOMMUNICATIONS



Polycopie de

Travaux Pratiques

2^{ème} Année Télécommunications/Électronique

Logique Combinatoire et Séquentielle

Présenté par : Dr Bouchemel Ammar

Année universitaire : 2018 - 2019

Préface

Ce manuscrit de Travaux Pratique d'Electronique Numérique (Logique combinatoire et séquentielle) est dédié aux étudiants de deuxième année Licence Télécommunications/Electroniques. Ce polycopie contient six chapitres :

Le chapitre 1 présente un rappel sur la logique combinatoire et séquentielle (Algèbre de Boole, Portes Logique, tableau de Karnaugh, logigrammes, Systèmes Logiques,...etc.) ;

Le chapitre 2 donne brièvement un aperçu sur la Technologie des Circuits intégrés logique TTL et CMOS ;

Le chapitre 3 est réservé à l'étude et la réalisation de fonctions logiques combinatoires usuelles ;

Le chapitre 4 évoque la réalisation des circuits combinatoires arithmétique ;

Le chapitre 5 traite les techniques de réalisation des circuits combinatoires logiques ;

Le chapitre 6 est consacré à l'étude et la réalisation de circuits compteurs ;

Ce polycopie étant essentiellement un ouvrage d'enseignement, nous y avons inclus le brochage des circuits intégrés nécessaires à la réalisation et la compréhension de travaux pratiques.

Nous espérons qu'il sera utile pour les étudiants de la deuxième année Licence Télécommunications/Electronique.

Bouchemel Ammar

1.1 Objectif

Ce chapitre comprend les notions de base de la logique. à savoir, la logique combinatoire et la logique séquentielle. A la fin de ce chapitre le lecteur doit être capable de :

- Connaître les principaux théorèmes et outils de l'Algèbre de Boole.
- Transcrire un cahier de charge en équations logiques.

1.2 Notions Liées à la Logique Combinatoire

1.2.1 Qu'est ce que l'Algèbre de Boole ?

L'algèbre de Boole est née en 1854 de la volonté de **George Boole**¹ de ramener les problèmes de logique classique (ou scolastique), alors traités par des raisonnements, parfois des diagrammes, à des calculs algébriques assimilant 0 à faux et 1 à vrai, de telle façon que les raisonnements soient remplacés par l'évaluation d'une formule logique, ou, au pire, la résolution d'un système d'équations logiques. Ce but amena G. Boole à un formalisme distinct de l'algèbre traditionnelle, avec notamment l'association à la conjonction d'un produit idempotent (i.e. tel que $x.x = x$).

¹**George Boole** (1815 - 1864), mathématicien logicien anglais. C'est en 1854, dans Recherches sur les lois de la pensée qu'il décrit un système algébrique qui portera plus tard le nom d'Algèbre de Boole.

1.2.2 Définitions

- **États** : les états logiques sont représentés par '0' et '1'. L'ensemble $E = \{ 0, 1 \}$ est donc appelé ensemble des états logiques. L'état logique peut pratiquement définir l'état d'un capteur ou d'un actionneur. Une convention peut être de dire que si K est la variable qui définit l'état d'un moteur, $K = 1$ signifie que le moteur tourne et $K = 0$ signifie que le moteur est à l'arrêt.
- **Variable logique** : on appelle variable logique toute variable qui ne peut prendre que deux états de l'ensemble des états logiques E , ('0' ou '1'). Par exemple, une lampe peut être assimilée à une variable logique (L) qui possède 2 états :
 - La lampe est éteinte : 0
 - La lampe brille : 1
- **Fonction logique** : il s'agit d'un groupe de variables logiques reliées par des opérateurs logiques. En général, il s'agit de l'expression d'une variable s dite de **sortie** en fonction des variables **d'entrée** sur lesquels des opérateurs logiques sont appliqués à l'image d'une fonction mathématique.
- **Table de vérité** : est un tableau permettant de représenter les états d'une fonction logique en fonction des variables d'entrée. Le nombre de lignes d'une table de vérité en logique combinatoire est égal à 2^n , n étant le nombre de variables d'entrées.
- **Logigramme** : Il s'agit de la représentation d'une fonction logique à partir des symboles logiques élémentaires utilisés. Le sens de lecture des logigrammes est de la gauche vers la droite ou du bas vers le haut.
- **Schémas électriques** : Il s'agit de la représentation électrique de la fonction logique. Le sens de lecture de tels schémas est de la gauche vers la droite ou du haut vers le bas. Dans un schéma, les variables sont représentées par des contacts normalement ouverts et les variables complémentées par des contacts normalement fermés.

1.2.3 Portes Logiques élémentaires


Définition : une porte logique est un circuit ayant au moins une entrée et exactement une sortie. Les valeurs des entrées et de la sortie sont les valeurs logiques vrai (*pour* '1') et faux (*pour* '0').

Les porte logique élémentaires sont, OU(OR), ET(AND), NON(NOT).

1.2.3.1 Porte ET(AND)

Une porte ET peut avoir un nombre arbitraire d'entrées. Sa sortie vaut 1 si et seulement si toutes les entrées valent 1. Donc, la sortie vaut 0 si et seulement si au moins une des entrées vaut 0, (Cf. Tableau 1.1).


Table 1.1 : Porte ET (2 entrées)

Symbole	Table de vérité	Sortie															
	<table border="1" style="border-collapse: collapse; text-align: center;"> <thead> <tr> <th>x</th> <th>y</th> <th>z</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	x	y	z	0	0	0	0	1	0	1	0	0	1	1	1	$z = x \cdot y$
x	y	z															
0	0	0															
0	1	0															
1	0	0															
1	1	1															

1.2.3.2 Porte OU(OR)

Comme la porte ET, la porte OU peut avoir un nombre arbitraire d'entrées. La sortie vaut 1 si et seulement si au moins une des entrées vaut 1. Autrement dit, la sortie vaut 0 si et seulement si toutes les entrées valent 0, (Cf. Tableau 1.2).

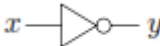
Table 1.2 : Porte OU (2 entrées)

Symbole	Table de vérité	Sortie															
	<table border="1" style="border-collapse: collapse; text-align: center;"> <thead> <tr> <th>x</th> <th>y</th> <th>z</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	x	y	z	0	0	0	0	1	1	1	0	1	1	1	1	$z = x + y$
x	y	z															
0	0	0															
0	1	1															
1	0	1															
1	1	1															

1.2.3.3 Porte NON(NOT)

La porte NON ou (*inverseur*) a exactement une entrée et une sortie. Sa sortie vaut 1 si et seulement si l'entrée vaut 0. Sinon la sortie vaut 0. Autrement dit, la valeur de la sortie est exactement l'inverse de la valeur de l'entrée (Cf. Tableau 1.3.2.3).

Table 1.3 : Porte NON

Symbole	Table de vérité	Sortie						
	<table border="1" style="border-collapse: collapse; text-align: center;"> <thead> <tr> <th style="padding: 2px 5px;">x</th> <th style="padding: 2px 5px;">y</th> </tr> </thead> <tbody> <tr> <td style="padding: 2px 5px;">0</td> <td style="padding: 2px 5px;">1</td> </tr> <tr> <td style="padding: 2px 5px;">1</td> <td style="padding: 2px 5px;">0</td> </tr> </tbody> </table>	x	y	0	1	1	0	$y = \bar{x}$
x	y							
0	1							
1	0							


1.2.4 Portes Logiques Complexes

Souvent, il est pratique de combiner plusieurs fonctions élémentaires dans une seule porte plus complexe, par exemple afin de conserver l'espace de dessin dans un diagramme de circuits. Dans cette section, nous présentons quelques portes complexes fréquemment utilisées.

1.2.4.1 Porte NON-ET

La porte NON-ET est une porte ET avec un inverseur sur la sortie (Cf. Tableau 1.4).


Table 1.4 : Porte NON-ET (2 entrées)

Symbole	Table de vérité	Sortie															
	<table border="1" style="border-collapse: collapse; text-align: center;"> <thead> <tr> <th style="padding: 2px 5px;">x</th> <th style="padding: 2px 5px;">y</th> <th style="padding: 2px 5px;">z</th> </tr> </thead> <tbody> <tr> <td style="padding: 2px 5px;">0</td> <td style="padding: 2px 5px;">0</td> <td style="padding: 2px 5px;">1</td> </tr> <tr> <td style="padding: 2px 5px;">0</td> <td style="padding: 2px 5px;">1</td> <td style="padding: 2px 5px;">1</td> </tr> <tr> <td style="padding: 2px 5px;">1</td> <td style="padding: 2px 5px;">0</td> <td style="padding: 2px 5px;">1</td> </tr> <tr> <td style="padding: 2px 5px;">1</td> <td style="padding: 2px 5px;">1</td> <td style="padding: 2px 5px;">0</td> </tr> </tbody> </table>	x	y	z	0	0	1	0	1	1	1	0	1	1	1	0	$z = \overline{x \cdot y}$
x	y	z															
0	0	1															
0	1	1															
1	0	1															
1	1	0															

1.2.4.2 Porte NON-OU

La porte NON-OU est une porte OU avec un inverseur sur la sortie (Cf. Tableau 1.5).


Table 1.5 : Porte NON-ET (2 entrées)

Symbole	Table de vérité	Sortie															
	<table border="1" style="border-collapse: collapse; text-align: center;"> <thead> <tr> <th>x</th> <th>y</th> <th>z</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	x	y	z	0	0	1	0	1	0	1	0	0	1	1	0	$z = \overline{x + y}$
x	y	z															
0	0	1															
0	1	0															
1	0	0															
1	1	0															

1.2.4.3 Porte OU-Exclusif

Elle peut avoir un nombre arbitraire d'entrées. Sa sortie vaut 1 si et seulement si exactement une des entrées vaut 1. Sinon la sortie vaut 0 (Cf. Tableau 1.6).

Table 1.6 : Porte OU-exclusif (2 entrées)

Symbole	Table de vérité	Sortie															
	<table border="1" style="border-collapse: collapse; text-align: center;"> <thead> <tr> <th>x</th> <th>y</th> <th>z</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	x	y	z	0	0	0	0	1	1	1	0	1	1	1	0	$z = \bar{x}.y + \bar{y}.x = x \oplus y$
x	y	z															
0	0	0															
0	1	1															
1	0	1															
1	1	0															

1.2.5 Règles de l'Algèbre de Boole

1.2.5.1 Théorèmes

L'algèbre de Boole respecte les théorèmes suivants (Cf. Tableau 1.7).

Table 1.7 : *Théorèmes*

Idempotence	$x + x = x$	$x.x = x$
Absorption	$x + (x.y) = x$	$x.(x + y) = x$
Morgan	$\overline{x + y} = \bar{x}.\bar{y}$	$\overline{x.y} = \bar{x} + \bar{y}$
Éléments neutres	$x + 0 = x$	$x.1 = x$

1.2.5.2 Axiomes

Les axiomes de l'algèbre de Boole sont illustrées sur le Tableau 1.8.

Table 1.8 : *Axiomes*

Commutativité	$x + y = y + x$	$x.y = y.x$
Associativité	$(x + y) + z = x + (y + z)$	$(x.y).z = x.(y.z)$
Distributivité	$x.(y + z) = x.y + x.z$	$x + (y.z) = (x + y).(x + z)$
Éléments absorbants	$x + 1 = 1$	$x.0 = 0$
Complémentarité	$\bar{x} + x = 1$	$\bar{x}.x = 0$

1.2.5.3 Identités Remarquables

Sont utilisées pour simplifier les fonction logique. Elles découlent des théorèmes de l'algèbre de Boole (Cf. Tableau 1.9).

Table 1.9 : *Identités Remarquables*

Identités
$x + \bar{x}.y = x + y$
$(x + y).(x + \bar{y}) = x$
$x.(x + y) = x$

1.2.6 Système Logique Combinatoire

Un système logique est dit **combinatoire**, si la sortie ne dépend que des variables logiques d'entrées $s = F(e_i)$. Pour un tel système le nombre de cas possible est de 2^n , où n correspond au nombre de variables d'entrées.

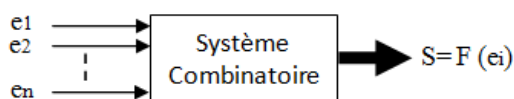


Figure 1.1: *Système Combinatoire.*

1.2.7 Formes Canoniques d'une Fonction Logique

La fonction logique F peut s'écrire sous un grand nombre de formes différentes. Deux d'entre elles, dites **formes canoniques**, sont particulièrement utiles :

- Elle peut s'écrire sous forme une somme de produit : groupe des variables logiques liées par des ET (1^{er} théorème de SHANNON).
- Elle peut s'écrire sous forme un produit de sommes : groupe des variables logiques liées par des OU (2^{ème} théorème de SHANNON).

1.2.7.1 Synthèse d'une Fonction Logique à partir de la Table de Vérité

La résolution d'un problème à partir de la table de vérité est le cas le plus usuel. l'exemple qui suit montre la synthèse d'une fonction logique à partir de la table de vérité (Cf. Tableau 1.10).

Table 1.10 : *Table de vérité*

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

On peut identifier les valeurs '1' de la sortie S comme correspondant aux situations des variables d'entrées : (a=1 ET b=0 ET c=0) OU (a=1 ET b=1 ET c=0) OU (a=0 ET b=0 ET c=1).

Dans ce cas la fonction S s'écrit,

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

1.2.8 Simplification des Fonctions Logiques

L'expression d'une fonction logique sous sa forme la plus simple n'est pas quelque chose d'évident. Deux principales méthodes sont utilisées pour la simplification des fonction logiques : la méthode algébrique et la méthode des tableaux de Karnaugh.

1.2.8.1 Méthode Algébrique

C'est une méthode efficace, mais il est facile de rater une simplification, notamment quand la fonction est compliquée. Elle ne fait appel qu'aux théorèmes de l'algèbre de Boole. En pratique, elle devient lourde pour un nombre de variables supérieur à 3.

Exemple : On souhaite simplifier la fonction suivante,

$$S = \bar{a}.\bar{b}.\bar{c}.d + \bar{a}.\bar{b}.c.d + \bar{a}.b.\bar{c}.d + \bar{a}.b.c.d$$

Solution :

$$S = \bar{a}.\bar{b}.d.(\bar{c} + c) + \bar{a}.b.d.(\bar{c} + c) \text{ avec, } \bar{c} + c = 1$$

$$S = \bar{a}.d.(\bar{b} + b)$$

$$S = \bar{a}.d$$

1.2.8.2 Méthode des Tableaux de Karnaugh

C'est une méthode plus efficace pour la simplification des fonctions logiques. Elle est adaptée même pour les fonctions logiques qui ont un nombre de variables supérieur à trois.

Tableau de Karnaugh : C'est une représentation matricielle dans 2^n cases. Chaque case indique l'état de la fonction à simplifier pour une certaine combinaison des variables d'entrées. Pour simplifier une fonction logique S montrée sur le tableau 1.11, il faut introduire dans cet tableau, les '1' de la table de vérité. Le code de Gray est utilisé pour indiquer le changement d'état des variables.

Exemple : Voici un exemple de tableau de Karnaugh à trois variables.

Table 1.11 : Tableau de Karnaugh à 3 variables

$c \backslash ab$	00	01	11	10
0	0	1	0	0
1	0	1	0	0

Solution :

Pour simplifier la fonction il faut suivre les étapes suivants,

- 1 - Rechercher les blocs de dimension 2^n , avec $n = 1, 2, \dots$ qui comprennent des cases adjacentes de '1'.
- 2 - Ecrire l'expression de chaque bloc trouvé en éliminant la variable qui change d'état.

Dans cet exemple la variable (a) reste à '0' et la variable (b) reste à '1', la variable (c) change d'état lorsqu'on passe de la première ligne à la deuxième (Cf. Figure 1.2).

c \ ab	00	01	11	10
0	0	1	0	0
1	0	1	0	0

Figure 1.2: Regroupement possible.

La fonction S simplifiée est donnée par,

$$S = \bar{a}.b$$

1.3 Notions Liées à la Logique Séquentielle

1.3.1 Système Logique Séquentiel

Un système logique est dit **séquentiel**, si la sortie dépend des variables d'entrées et de son état précédent (Cf. Figure 1.3).

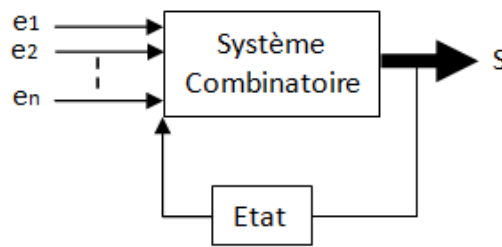


Figure 1.3: Système séquentiel.

Pour modéliser le fonctionnement d'un système séquentiel, plusieurs outils peuvent être utilisés, nous citons : Chronogramme (diagramme temporel), GRAFCET, tableau des états, graphe d'état,...

Dans ce qui suit nous allons nous intéresser au Chronogramme. Dans tel outil, on tient compte de l'aspect temporel en faisant varier à des instants précis les variables d'entrées pour observer l'état des sorties.

La Figure 1.4 montre un Chronogramme de la fonction, $S = a + b.c$.

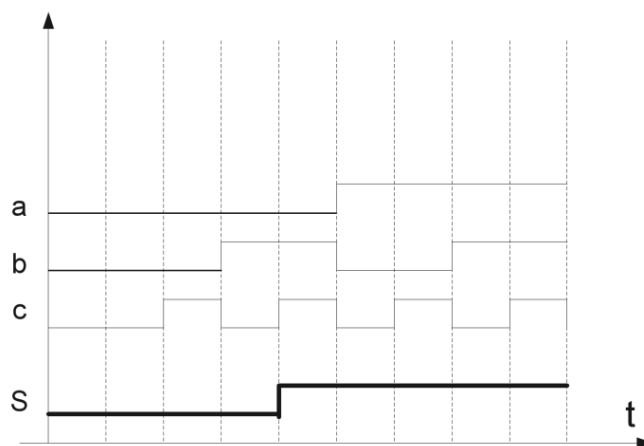


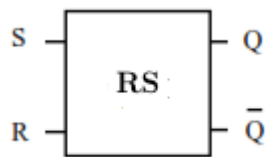
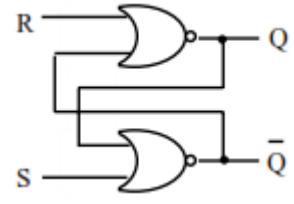
Figure 1.4: Chronogramme de la fonction S

1.3.2 Éléments de Base des Circuits Séquentiels (Bascules)

La bascule est un système bistable permettant de mémoriser des états logiques élémentaires "0" ou "1". L'état de la bascule peut être modifié en agissant sur les entrées. Le nouvel état de la bascule dépend de l'état précédent. On distingue quatre types : **RS**, **D**, **JK** et **T**.

1.3.2.1 Bascule RS

La bascule **RS** peut être réalisée à partir de deux portes **NOR**, comme le montre le tableau ci-dessous.

Schéma bloc	Schéma interne	Table de vérité															
		<table border="1"> <thead> <tr> <th>S</th> <th>R</th> <th>Q(t)</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Q (t-1) mémorisation</td> </tr> <tr> <td>0</td> <td>1</td> <td>0 écriture d'un 0 (Reset)</td> </tr> <tr> <td>1</td> <td>0</td> <td>1 écriture d'un 1 (set)</td> </tr> <tr> <td>1</td> <td>1</td> <td>indéterminé</td> </tr> </tbody> </table>	S	R	Q(t)	0	0	Q (t-1) mémorisation	0	1	0 écriture d'un 0 (Reset)	1	0	1 écriture d'un 1 (set)	1	1	indéterminé
S	R	Q(t)															
0	0	Q (t-1) mémorisation															
0	1	0 écriture d'un 0 (Reset)															
1	0	1 écriture d'un 1 (set)															
1	1	indéterminé															

Chronogrammes de la Bascule RS

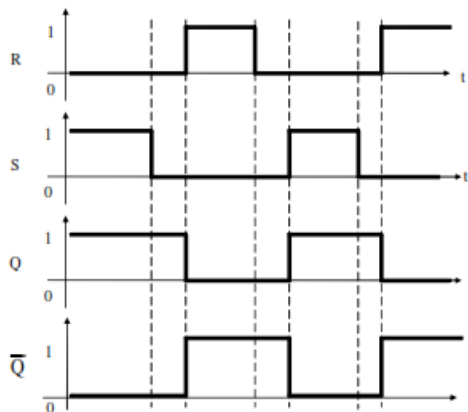
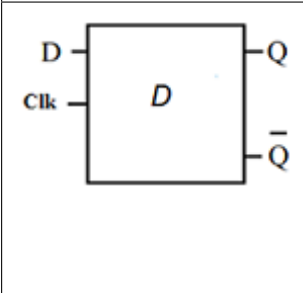
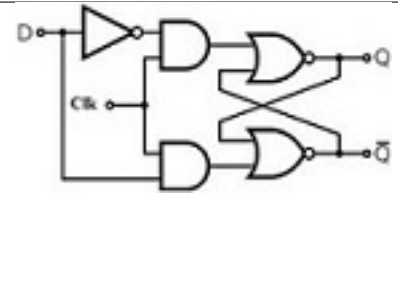


Figure 1.5: Chronogrammes de la bascule **RS**.

1.3.2.2 Bascule D(LATCH)

Le tableau ci-dessous montre les schémas et la table de vérité de la bascule **D**.

Schéma bloc	Schéma interne	Table de vérité												
		<table border="1"> <thead> <tr> <th>D</th> <th>Clk</th> <th>Q(t)</th> </tr> </thead> <tbody> <tr> <td>X</td> <td>↓</td> <td>Q (t-1) mémorisation</td> </tr> <tr> <td>1</td> <td>↑</td> <td>1</td> </tr> <tr> <td>0</td> <td>↑</td> <td>0</td> </tr> </tbody> </table>	D	Clk	Q(t)	X	↓	Q (t-1) mémorisation	1	↑	1	0	↑	0
D	Clk	Q(t)												
X	↓	Q (t-1) mémorisation												
1	↑	1												
0	↑	0												

Principe de Fonctionnement

La sortie **Q** prend l'information présentée en **D** au moment de l'apparition d'un front montant sur **Clk** et seulement à cet instant. le reste du temps **Q** garde d'information en mémoire jusqu'au prochain front montant sur **Clk**.

Chronogrammes de la Bascule D

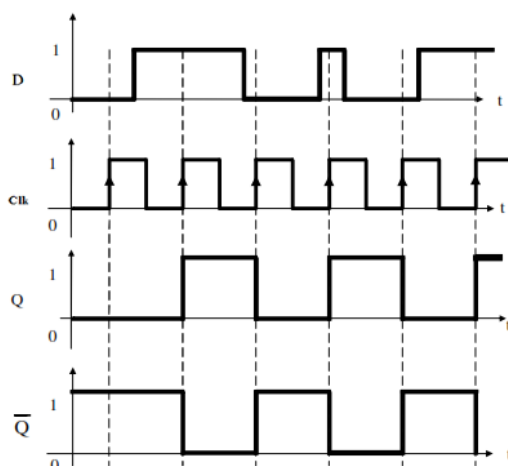
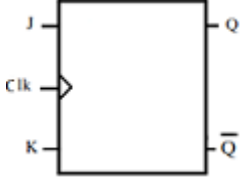
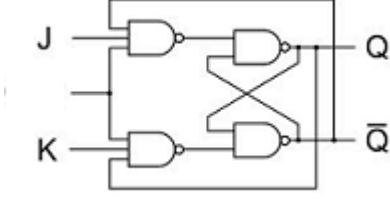


Figure 1.6: Chronogrammes de la bascule **D**

1.3.2.3 Bascule JK

C'est un bascule synchrone, son principe de fonctionnement est montré sur le tableau ci-dessous.

Schéma bloc	Schéma interne	Table de vérité																								
		<table border="1"> <thead> <tr> <th>J</th> <th>K</th> <th>Clk</th> <th>Q(t)</th> </tr> </thead> <tbody> <tr> <td>X</td> <td>X</td> <td>↓</td> <td>Q (t-1) mémorisation</td> </tr> <tr> <td>0</td> <td>0</td> <td>↑</td> <td>Q (t-1) mémorisation</td> </tr> <tr> <td>1</td> <td>0</td> <td>↑</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>↑</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>↑</td> <td>$\overline{Q(t-1)}$</td> </tr> </tbody> </table>	J	K	Clk	Q(t)	X	X	↓	Q (t-1) mémorisation	0	0	↑	Q (t-1) mémorisation	1	0	↑	1	0	1	↑	0	1	1	↑	$\overline{Q(t-1)}$
J	K	Clk	Q(t)																							
X	X	↓	Q (t-1) mémorisation																							
0	0	↑	Q (t-1) mémorisation																							
1	0	↑	1																							
0	1	↑	0																							
1	1	↑	$\overline{Q(t-1)}$																							

Principe de Fonctionnement

- Si l'entrée **Clk** n'est pas sur un front montant, les sorties ne changent pas d'état.
- Si on applique un front montant sur l'entrée **Clk** alors que $\mathbf{J} \neq \mathbf{K}$, la sortie **Q** prend la valeur de l'entrée **J**.
- Si on applique un front montant sur l'entrée **Clk** alors que $\mathbf{J} = \mathbf{K} = 0$, les sorties ne changent pas d'état (mémorisation).
- Si on applique un front montant sur l'entrée **Clk** alors que $\mathbf{J} = \mathbf{K} = 1$, les sorties changent d'état (basculement).

Chronogrammes de la Bascule JK

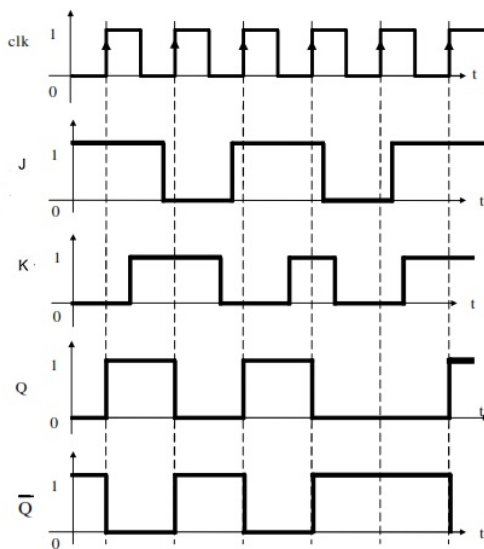


Figure 1.7: Chronogrammes de la bascule JK

1.3.3 Les compteurs

Un compteur est un ensemble de bascules reliées en cascade. Il réalise également la division de fréquence. En effet, le passage cyclique de N impulsions par comptage est équivalent à diviser la fréquence du signal d'entrée de la première bascule par N . On classe les compteurs en deux familles

- Compteur synchrone : toutes les bascules possèdent une horloge commune de commande
- Compteur asynchrone : Pas d'horloge commune, d'une manière générale, le signal d'horloge d'une bascule de X n'est autre que le signal de sortie de la bascule $X-1$

1.3.3.1 Compteurs asynchrones

Un compteur asynchrone est un système logique composé de bascules dans lesquels les impulsions que l'on applique à l'entrée doivent traverser la première bascule avant de pouvoir commander la seconde et ainsi de suite jusqu'à la dernière bascule. Avec n bascules on obtient 2^n combinaisons alors un compteur modulo 2^n . La figure suivante représente un logigramme d'un compteur modulo 16

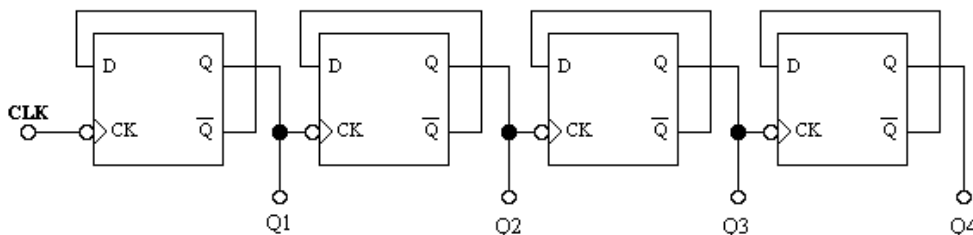


Figure 1.8: Compteur modulo 16

La figure suivante représente un chronogramme d'un compteur modulo 16

Comme vous pouvez le constater, Q_1 Q_2 Q_3 Q_4 sont successivement égales à 0000, 0001, 0010, ..., 1111, c'est-à-dire 0, 1, 2, ..., 15. Bien sûr ce cycle continue indéfiniment, aussi longtemps que le signal d'horloge est présent. C'est la façon la plus simple de réaliser un compteur, et bien sûr ce type de compteur est disponible sous forme de divers circuits intégrés prêts à l'emploi. La table de séquences du compteur est donnée par la table suivante

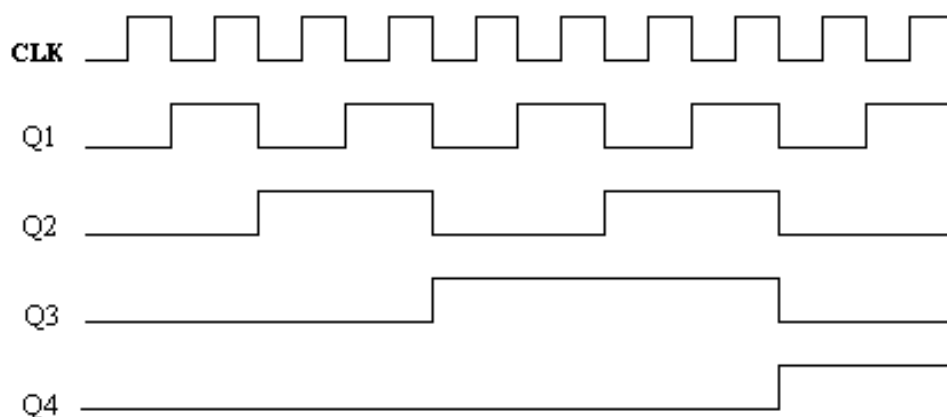


Figure 1.9: chronogramme dun compteur modulo 16

N°IMPUL	D	C	B	A
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

1.3.3.2 Compteur synchrone

Dans les compteurs synchrones, toutes les bascules sont synchronisées par le même signal d'horloge, soit sur front montant, soit sur front descendant du signal d'horloge, ce qui évite les retards de propagation et les sorties des bascules restent stables entre deux impulsions successives. On peut réaliser un compteur synchrone à partir de bascules JK ,D. Exemple : Compteur synchrone Modulo 16. La figure suivante représente logigramme d'un compteur synchrone modulo 16

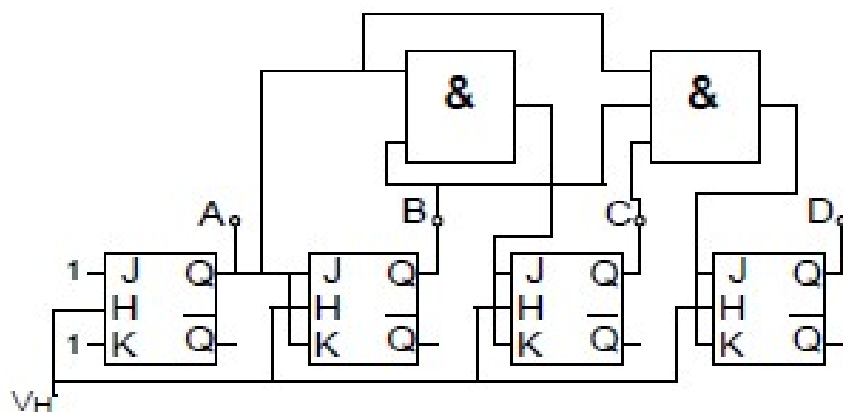


Figure 1.10: *Compteur synchrone modulo 16*

La figure suivante représente chronogramme dun compteur synchrone modulo 16

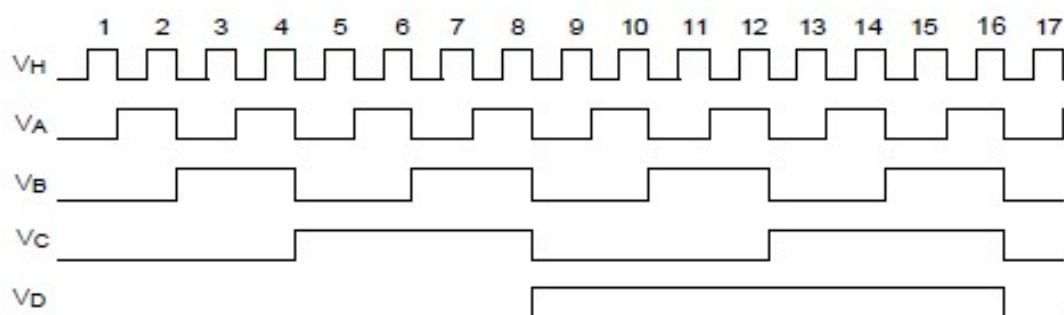


Figure 1.11: *synchrone compteur synchrone modulo 16*

Chapitre 2

TP1 : Technologie des Circuits Intégrés TTL et CMOS

2.1 Objectifs

- Découverte des circuits intégrés Logiques.
- Réaliser un schéma de câblage.
- Réaliser un câblage claire et conforme au schéma de câblage.
- Réaliser un compte rendu de TP.

2.2 Matériel Nécessaire

- Alimentation stabilisé 0-5V
- Plaquette d'essais
- Diode Photo-lumineuse (LED)
- Fils de connexion
- Résistances de protection
- Circuits intégrés TTL :7404, 7408, 7400, 7432, 7402, 7486
- Circuits intégrés CMOS :4049, 4081, 4011, 4071, 4001, 4030

2.3 Manipulation des Portes Logiques Standard

2.3.1 Portes Logiques

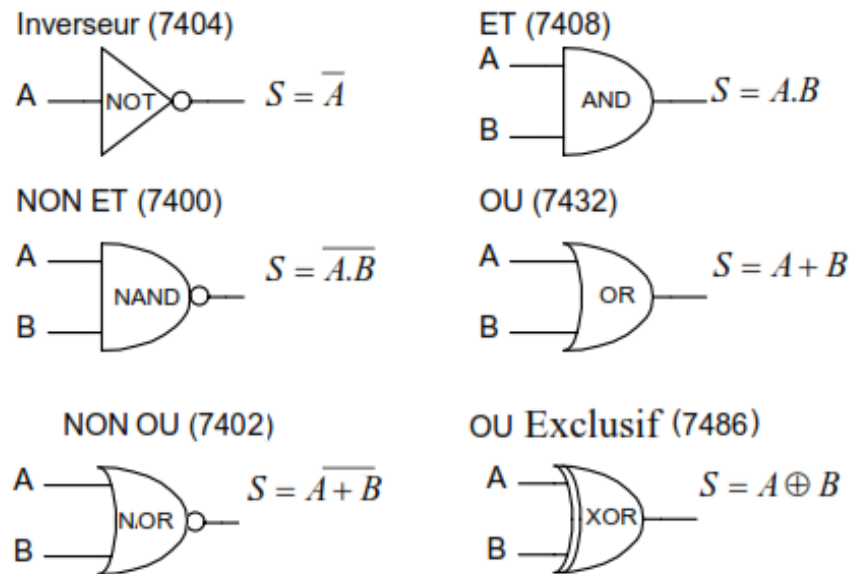


Figure 2.1: Portes logiques de base

2.3.2 Travail à Faire

- 1) Donner les tables de vérité des portes ci-dessus.
- 2) Faire fonctionner ces circuits et vérifier la table de vérité de chaque porte logique par la technologie TTL et CMOS.
- 3) Vérifier les lois de Morgan en câblant les fonctions : $S = \overline{\overline{A + B}}$ et $S = \overline{\bar{A}. \bar{B}}$
- 4) Exprimer les fonctions AND, OR, NAND en utilisant seulement des portes NOR. vérifier l'expression par câblage.
- 5) Exprimer les fonctions AND, OR, NAND en utilisant seulement des portes NAND. vérifier l'expression par câblage.
- 6) Comment peut-on utiliser le circuit 7410 (NAND à 3 entrées) comme une porte à deux entrées ? Donner deux solutions possibles.

2.4 Synthèse de Fonction Logique

Une lampe L est commandée par trois interrupteurs. Elle est allumée dans les deux cas suivants :

- Si l'un seulement des interrupteurs est fermé.
- Si les trois interrupteurs sont fermés.

2.4.1 Travail à Faire

- 1) Établir la table de vérité du système.
- 2) Déterminer l'équation logique de fonctionnement de la lampe L.
- 3) Câbler le circuit en utilisant que des portes XOR.

2.5 Annexe : Brochage des Circuits TTL

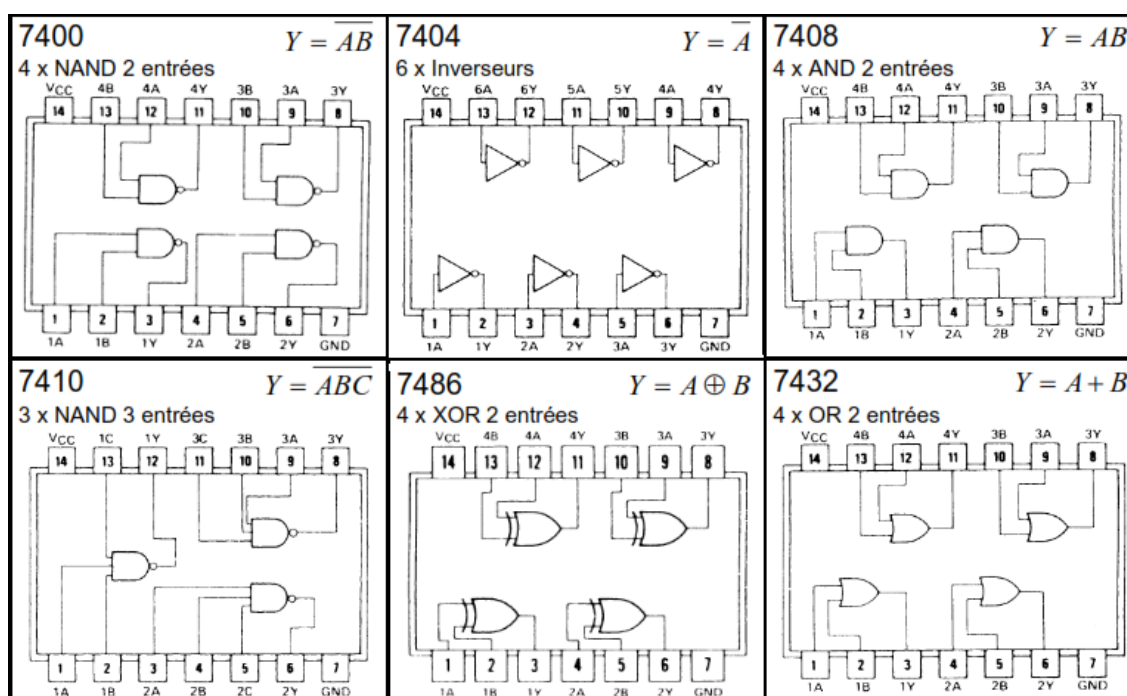


Figure 2.2: Circuits intégrés TTL

Chapitre 3

TP2 : Etude et Réalisation de Fonctions Logiques Combinatoires Usuelles

3.1 Objectifs

L'objectif de ce Tp est de vérifier expérimentalement le fonctionnement de quelques circuits combinatoires fondamentaux donnés en circuit intégré tels que le décodeur et BCD-7 segments.

3.2 Matériel Nécessaire

- Alimentation stabilisé 0-5V
- Plaquette d'essais
- Diode Photo-lumineuse (LED)
- Fils de connexion
- Résistances de protection
- Circuits intégrés : 7442, 7448

3.3 Décodeur (7442)

La figure ci-dessous montre la table de vérité de décodeur BCD-décimal. il possède 4 bits BCD en entrée et 10 sorties actives au niveau bas.

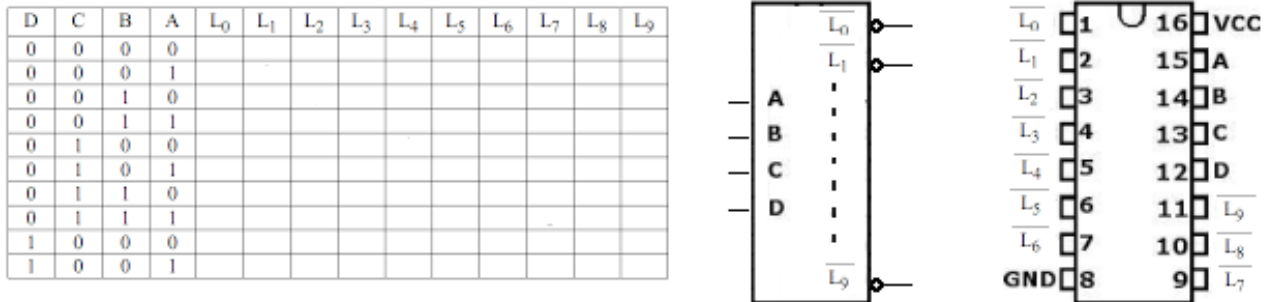


Figure 3.1: Brochage du circuit intégré 7442

3.3.1 Travail à Faire

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

- $\bar{L}_0 = \dots\dots\dots$
- $\bar{L}_1 = \dots\dots\dots$
- $\bar{L}_2 = \dots\dots\dots$
- $\bar{L}_3 = \dots\dots\dots$
- $\bar{L}_4 = \dots\dots\dots$
- $\bar{L}_5 = \dots\dots\dots$
- $\bar{L}_6 = \dots\dots\dots$
- $\bar{L}_7 = \dots\dots\dots$
- $\bar{L}_8 = \dots\dots\dots$
- $\bar{L}_9 = \dots\dots\dots$

3.4 Décodeur BCD-7 Segments (7448)

Dans les applications numériques, les dix chiffres de 0 à 9 sont donnés au moyen d'un afficheur 7 segments. Les segments sont des **LED**. l'afficheur à 7 segments comporte également un point appelé DP qui n'est autre que l'équivalent de notre virgule décimale. chaque segment est désigné par une lettre A, B, C, D, E, F, G et le point par DP.

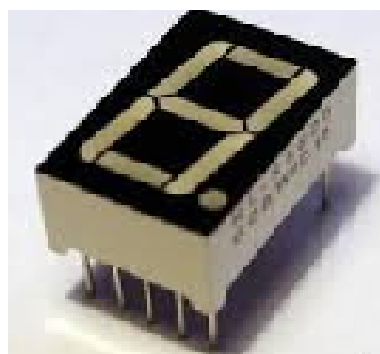
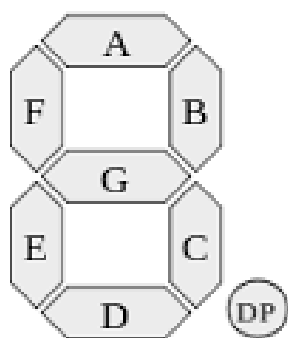


Figure 3.2: Afficheur à 7 Segments 7448

Deux types d'afficheurs :

- 1) **Afficheur à cathode commune (CC)** : les cathodes de toutes les LEDs sont connectées ensemble et reliées à 0 volt.
- 2) **Afficheur à anode commune (AC)** : les anodes de toutes les LEDs sont connectées ensemble et reliées à Vcc.

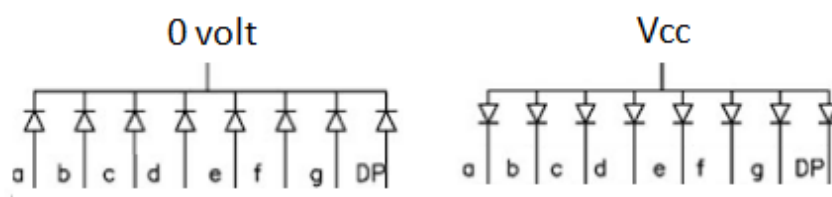


Figure 3.3: Câblage les LED des segments dans un afficheur à CC et AC

L'utilisation de ce type d'afficheurs nécessite un décodeur pour traduire le code BCD en code 7 segments. Il existe également plusieurs décodeurs BCD-7 tel que :

- 1- le décodeur 7447 à anode commune.
- 2- le décodeur 7448 à cathode commune.

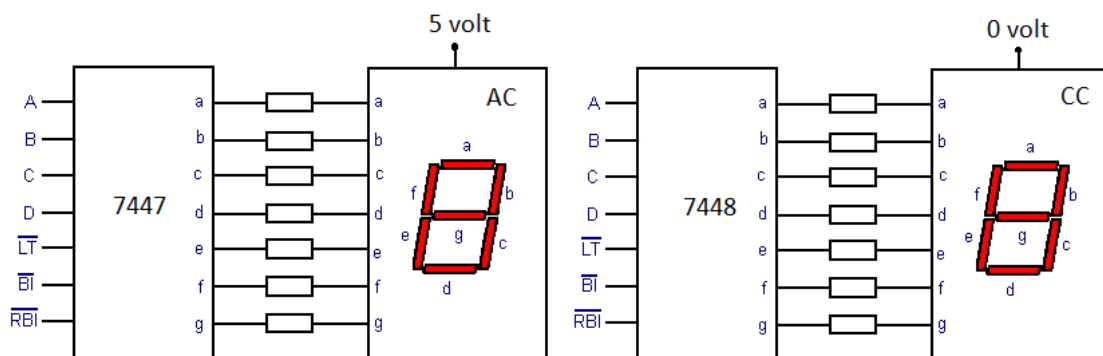


Figure 3.4: Brochage d'un afficheur 7 segments commandé par un décodeur

Dans ce qui suit nous allons utiliser le décodeur 7448 avec un afficheur 7 segments à cathode commune. Ce décodeur dispose de 7 sorties reliées directement avec un afficheur. En entrée, il dispose de quatre entrées BCD et 3 entrées supplémentaires qui sont :

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

3.4.1 Travail à Faire

1) Câbler le décodeur et l'afficheur de la figure ci-dessous.

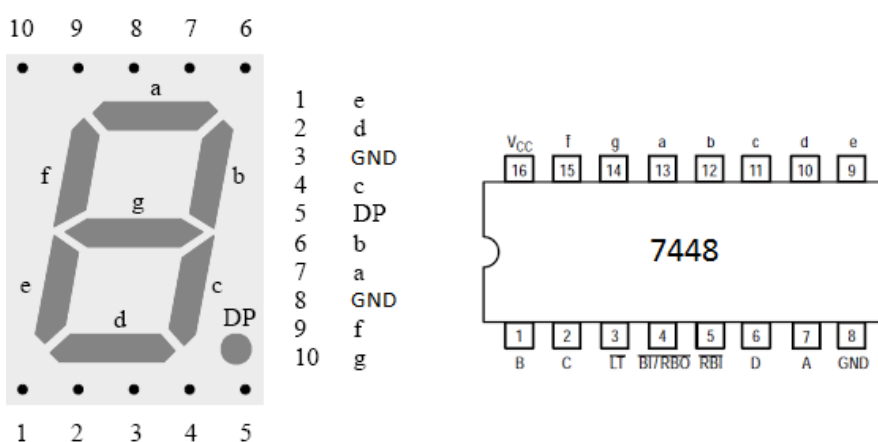


Figure 3.5: Afficheures 7 segments, décodeur à cathode commune

$\overline{\mathbf{BI/RBO}} = 1$ et $\overline{\mathbf{RBI}} = 0$ ou 1

2) Vérifier si tous les segments sont en bon état ($\overline{\mathbf{LT}} = 0$).

3) Vérifier l'affichage des 10 chiffres décimaux ($\overline{\mathbf{LT}} = 1$) et faire varier les entrées BCD.

Tracer la table de vérité.

Chapitre 4

TP3 : Etude et Réalisation d'un Circuit Combinatoire Arithmétique

4.1 Objectifs

L'objectif de ce Tp est que l'étudiant devra capable de cabler les semi additionneurs et additionneurs complets à l'aide des portes logiques. l'étudiant devra vérifier expérimentalement le fonctionnement du circuit intégré tels que l'additionneur 7483.

4.2 Matériel Nécessaire

1. Alimentation stabilisé 0-5V
2. Plaquette d'essais
3. Diode Photo-lumineuse (LED)
4. Fils de connexion
5. Résistances de protection
6. Circuits intégrés logiques 7486, 7408, 7432 et 7483

4.3 Additionneur à l'aide des Portes Logiques

4.3.1 Semi-Additineur

Le circuit ci-dessus montre le brochage d'un semi additionneur de deux nombres A et B

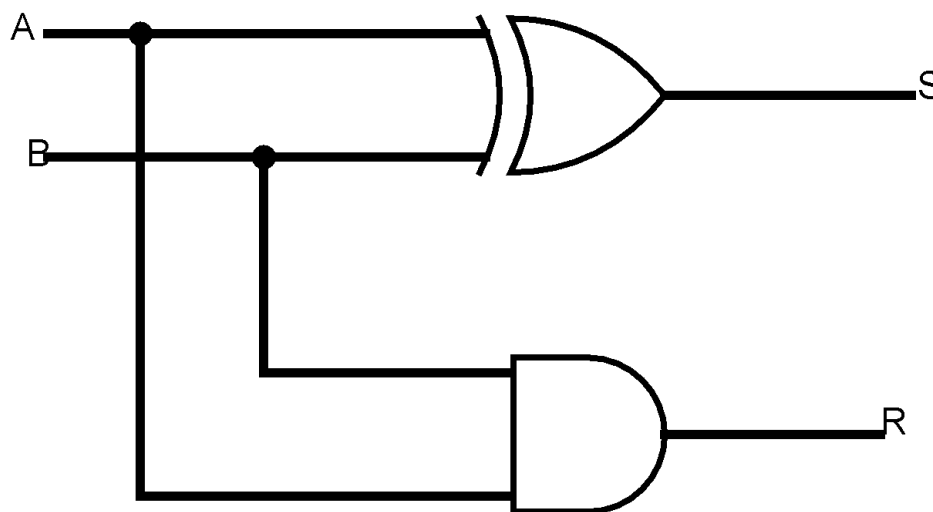


Figure 4.1: *Semi-Additionneur*

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

4.3.2 Additionneur Complet

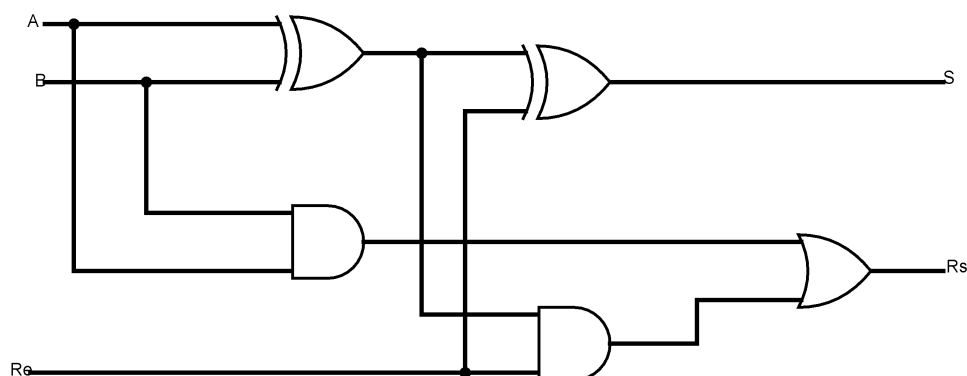


Figure 4.2: Additionneur Complet

A	B	Re	S	Rs
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

4.3.3 Travail Demandé :

1. Réaliser le montage correspond au logigramme d'un semi-additionneur.
2. Utiliser l'état de la diode ■ LED ■ pour remplir le tableau de vérité d'un semi-additionneur.
3. Réaliser le montage correspond au logigramme d'un additionneur complet.
4. Utiliser l'état des deux diodes ■ LED ■ pour remplir le tableau de vérité d'un additionneur complet.

4.4 Additionneur Complet (7483)

4.4.1 Portes Logiques

Le circuit ci-dessus montre le brochage d'un additionneur de deux nombres A et B de 4 bits chacun.

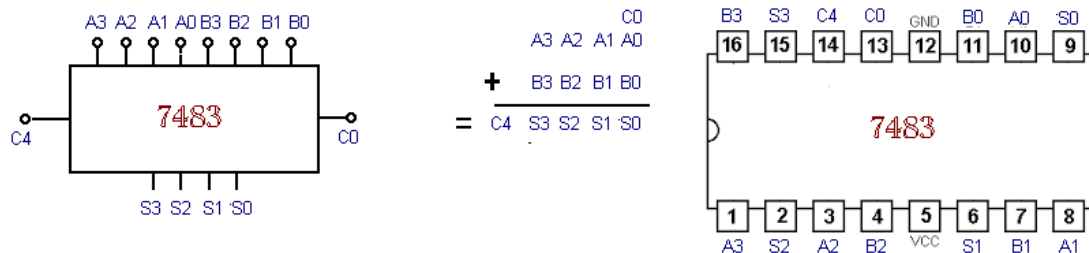


Figure 4.3: Circuits intégrés TTL 7483

4.4.2 Travail à Faire

1) A l'aide de l'additionneur complet donner la structure interne du 7483.

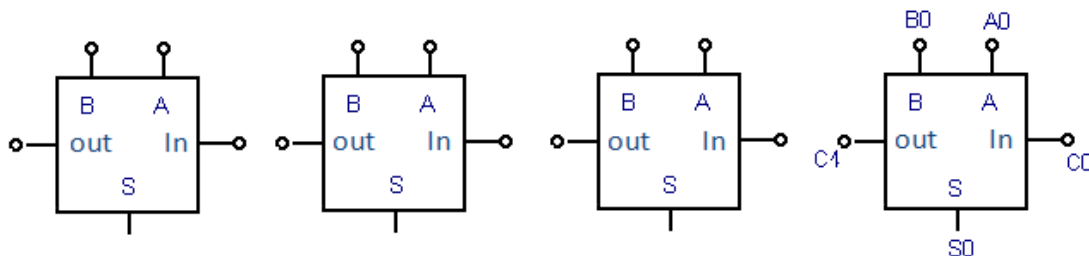
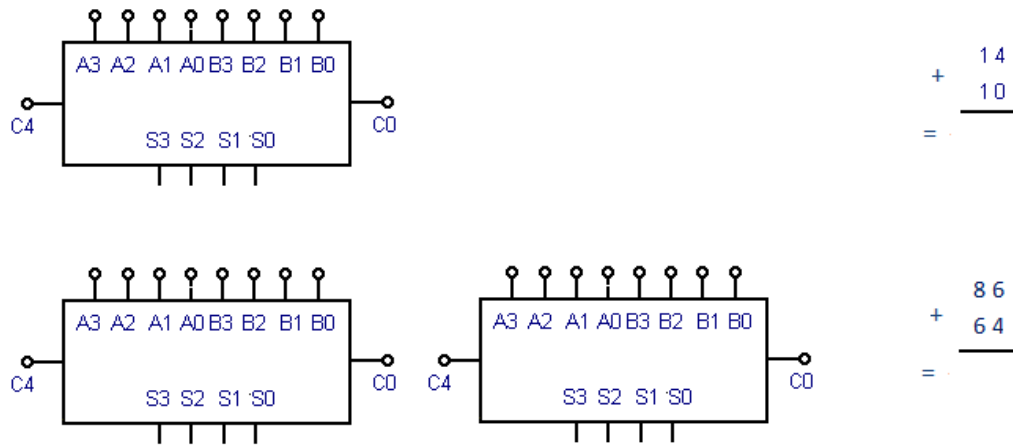


Figure 4.4: structure interne de 7483

2) A l'aide de l'additionneur complet 7483, réaliser les opérations d'addition suivantes :



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

$$\begin{array}{r}
 86 \\
 + 64 \\
 \hline
 \end{array}$$

Figure 4.5: Circuits intégrés TTL

Chapitre 5

TP4 : Etude et Réalisation d'un Circuit Combinatoire Logique

5.1 Objectifs

1. Réaliser un transcodeur permettant le passage d'un nombre en code GRAY vers un nombre binaire.
2. S'entraîner sur la simplification des fonctions logiques
3. Réaliser le câblage des logigrammes
4. Réaliser un compte rendu de TP

5.2 Matériel Nécessaire

- Alimentation stabilisé 0-5V
- Plaquette d'essais
- Diode Photo-lumineuse (LED)
- Fils de connexion
- Résistances de protection
- Circuits intégrés : 7486

5.3 Code GRAY

Considérer le code de GRAY définie par la table de vérité ci-dessous :

Décimal	Binaire	Binire	Binaire	Binaire	Gray	Gray	Gray	Gray
Décimal	A ₄	A ₃	A ₂	A ₁	G ₄	G ₃	G ₂	G ₁
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	1
3	0	0	1	1	0	0	1	0
4	0	1	0	0	0	1	1	0
5	0	1	0	1	0	1	1	1
6	0	1	1	0	0	1	0	1
7	0	1	1	1	0	1	0	0
8	1	0	0	0	1	1	0	0
9	1	0	0	1	1	1	0	1
10	1	0	1	0	1	1	1	1
11	1	0	1	1	1	1	1	0
12	1	1	0	0	1	0	1	0
13	1	1	0	1	1	0	1	1
14	1	1	1	0	1	0	0	1
15	1	1	1	1	1	0	0	0

5.3.1 Travail Demandé

- Générer les expressions logiques de code de Gray G_4, G_3, G_2 et G_1 correspondantes :
 - sous forme de sommes de produits.
 - sous forme de produits de sommes.
- Simplifier les expressions en utilisant les règles de l'algèbre de Boole.
- Construire le diagramme de Karnaugh et déterminer les expressions logiques associées.
- Donner les circuits logiques (logigrammes) correspondants.

5.4 Réalisation

- Câbler les circuits logiques obtenus sur une plaque d'essai.

2. Vérifier les valeurs de code de Gray pour les seize états d'entrée A_4, A_3, A_2, A_1 afin de dresser la table de vérité en faisant apparaître le résultat (code de Gray) sur quatre LED.
3. Que remarquer-vous ?

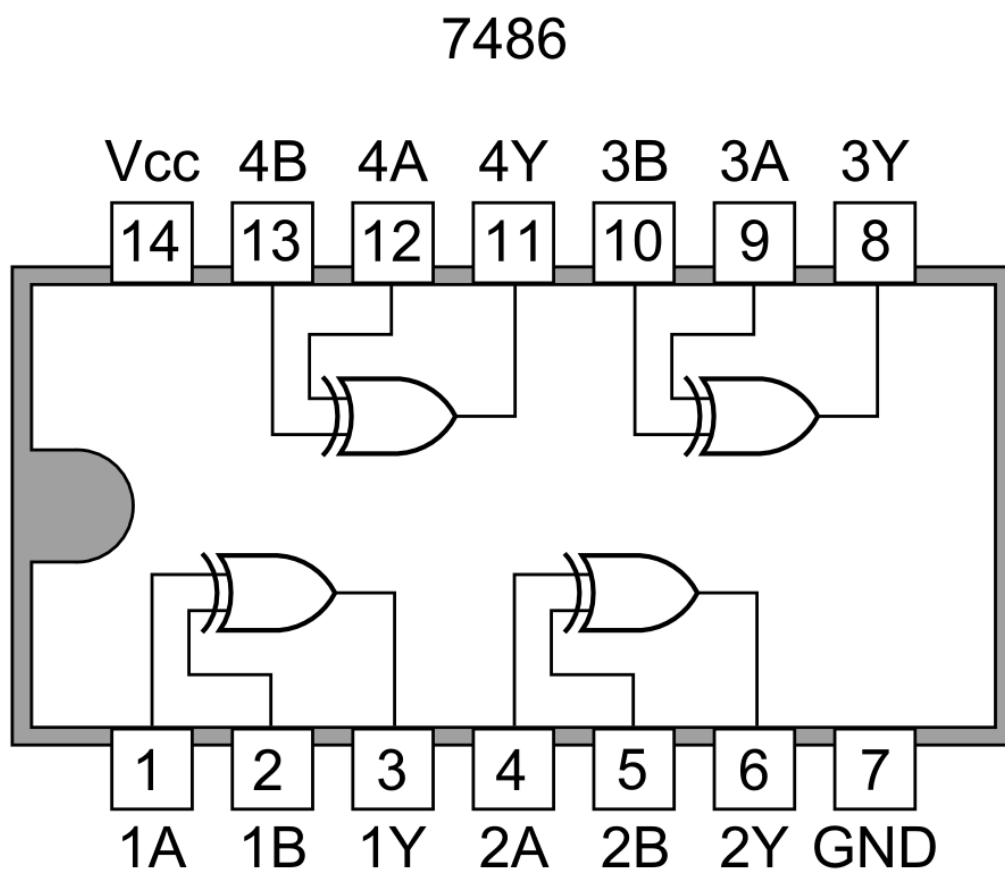


Figure 5.1: Circuit Intégré 7486

Chapitre 6

TP5.1 : Etude et Réalisation de Circuits Compteurs

6.1 Objectifs

Vérifier expérimentalement le fonctionnement de quelques circuits séquentiels fondamentaux tels que : Bascules RS, D et JK.

6.2 Matériels Utilisé

- Alimentation stabilisé 0-5V
- Plaquette d'essais
- Diode Photo-lumineuse (LED)
- Fils de connexion
- Résistances de protection
- Générateur TTL
- Oscilloscope
- Circuits intégrés logiques : 7400, 7404, 7474, 74LS73

6.3 Définitions

Les bascules sont les éléments de base de la logique séquentielle, elles peuvent avoir deux états stables (0, 1) se sont des mémoires élémentaires qui peuvent écrire le 1 logique, le 0 logique. Une bascule peut avoir deux types de fonctionnement :

- * Fonctionnement Asynchrone : La sortie de la bascule change d'état uniquement en fonction des d'entrées.
- * Fonctionnement Synchrone : Le changement de la sortie de la bascule est conditionné par une autorisation d'un signal d'horloge.

6.4 Bascule Asynchrone avec des Portes Logiques

6.4.1 Bascule RS Asynchrone avec des Portes NAND

La bascule RS dispose de deux entrées S (Set) et R (Reset) et deux sorties Q et \bar{Q} . La bascule a deux états stables $Q = 1, \bar{Q} = 0$ et $Q = 0, \bar{Q} = 1$. Son état dépend du signal appliqué sur les entrées S et R et de son état antérieur.

6.4.2 Travail Demandé

- a) Réaliser le montage de la bascule RS à l'aide d'opérateurs logiques NON-ET à deux entrées.

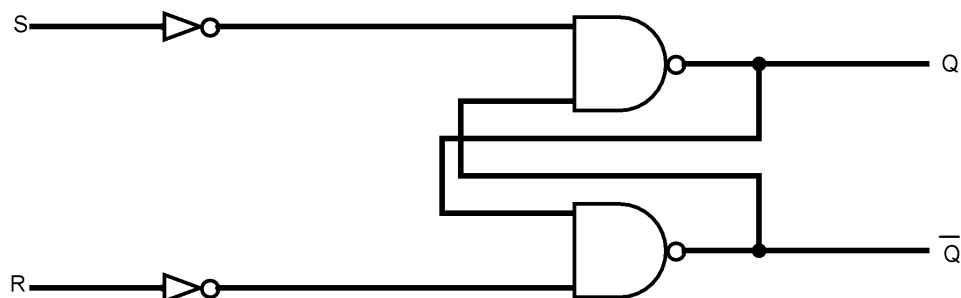


Figure 6.1: Bascule RS

- b) Compléter le chronogramme de la bascule en faisant apparaître le résultat (sortie Q et \bar{Q}) sur une LED. Que remarquer vous pour les valeurs de Q et \bar{Q} dans le cas $S = 1$ et $R = 1$

.....

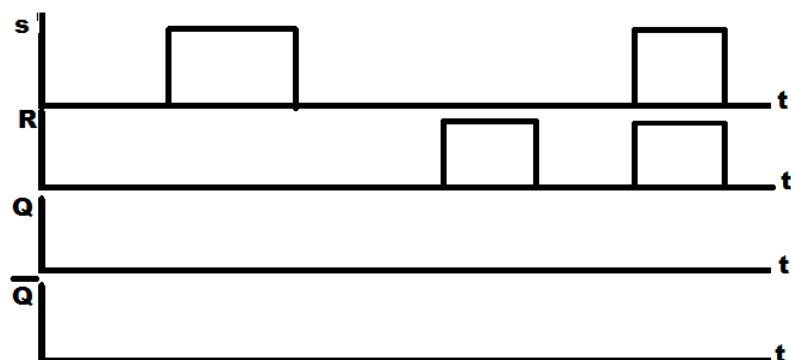


Figure 6.2: Chronogramme de la Bascule RS

- c) En déduire la table de vérité de la bascule

Remarque :

- Si la sortie ne change pas l'état par rapport au précédent, on écrit : $Q_n = Q_{n-1}, \bar{Q}_n = \bar{Q}_{n-1}$.
- Si $Q_n = \bar{Q}_n$ on écrit : Etat interdit ou indéterminé (il est interdit de l'utiliser dans les applications)

R	S	Q_n	\bar{Q}_n
0	0		
0	1		
1	0		
1	1		

6.5 Bascule Synchrones sous Forme de Circuit Intégré

6.5.1 Bascule D Synchrones : le 7474

On a vu que le type de fonctionnement $Q_n = \bar{Q}_n$ devait être évité pour une bascule RS. Un moyen de s'assurer qu'il ne peut pas se produire ce type de situation est de n'utiliser qu'une seule entrée. C'est le cas de la bascule D. La figure 5.2 vous montre le brochage du circuit intégré 7474 qui contient 2 bascules D indépendantes.

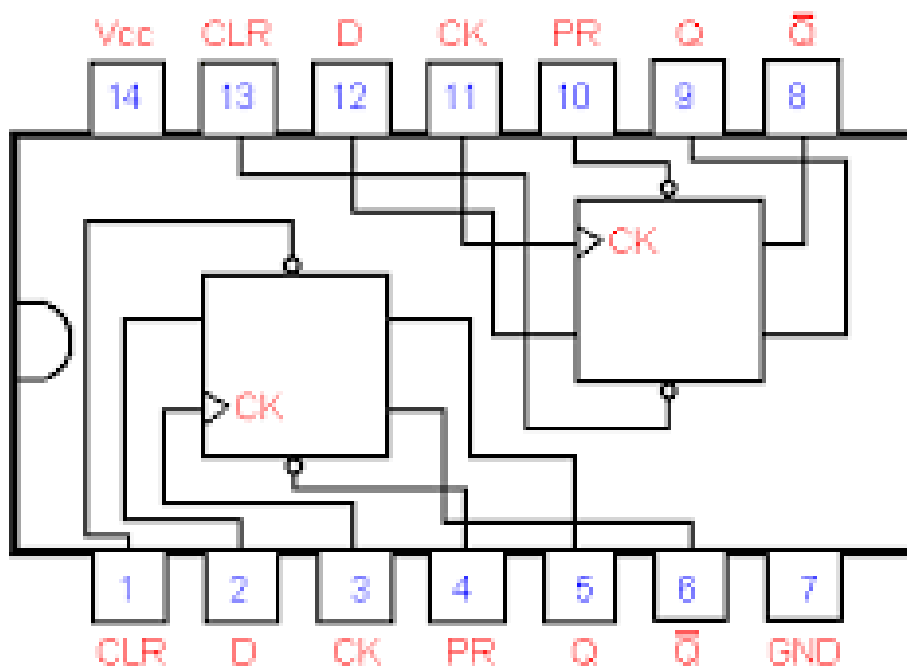


Figure 6.3: Brochage du Circuit Intégré 7474

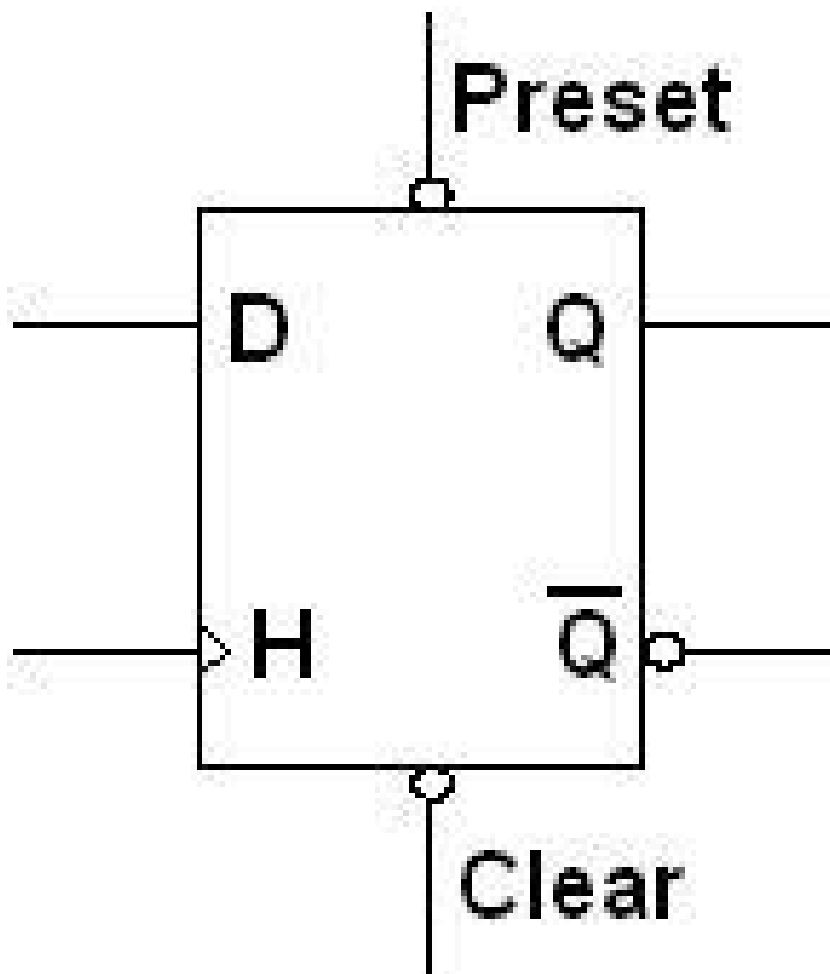


Figure 6.4: Symbole Logique du Circuit Intégré 7474

PR	CLR	CK	D	Q_n	\bar{Q}_n
0	1	X	X	1	0
1	0	X	X	0	1
0	0	X	X	interdit	interdit
1	1	\uparrow	1	1	0
1	1	\uparrow	0	0	1
1	1	0	X	Q_{n-1}	\bar{Q}_{n-1}
1	1	1	X	Q_{n-1}	\bar{Q}_{n-1}

6.5.2 Bascule JK Synchrones : le 74LS73

C'est une bascule à deux entrées qui ne présente pas d'état interdit ou indéterminé. Le circuit intégré 74LS73 contient deux bascules JK fonctionne au front descendant avec entrée de remise à zéro. Le brochage de ce circuit est donné à la figure 6.5.

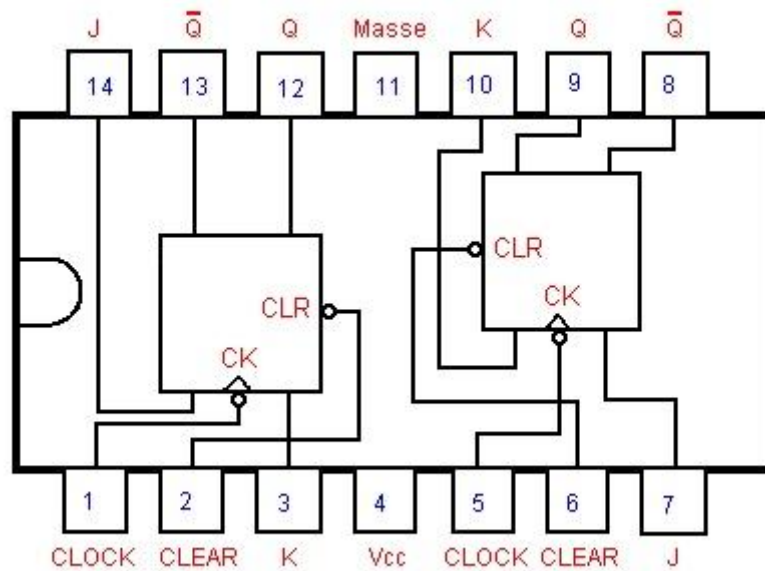


Figure 6.5: Brochage du Circuit Intégré 74LS73

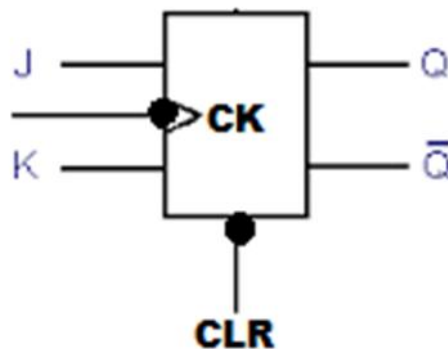


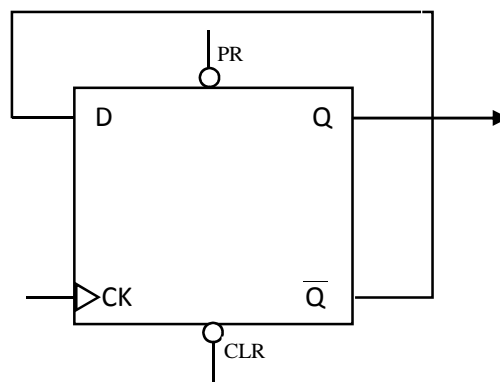
Figure 6.6: Symbole Logique du Circuit intégré 74LS73

CLEAR	CLOCK	J	K	Q_n	Q_n
0	X	X	X	0	1
1	↓	0	0	Q_{n-1}	Q_{n-1}
1	↓	1	0	1	0
1	↓	0	1	0	1
1	↓	1	1	Basculement	Basculement
1	1	X	X	Q_{n-1}	Q_{n-1}
1	0	X	X	Q_{n-1}	Q_{n-1}

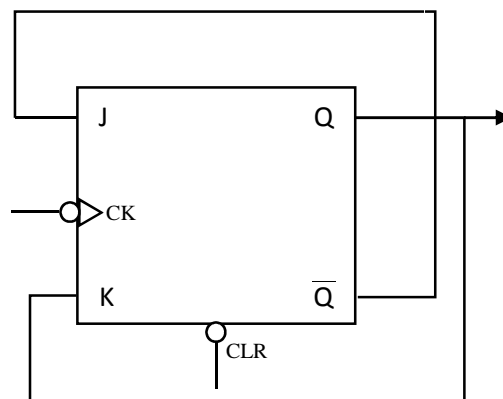
Table de vérité de chaque bascule JK du circuit intégré

6.6 Travail Demandé

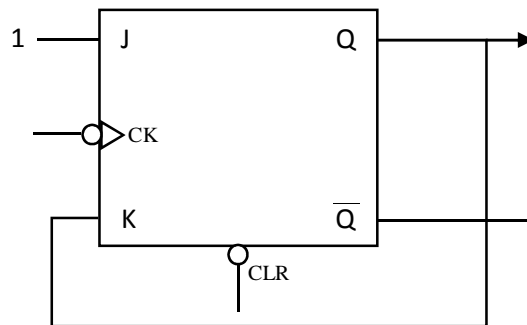
1. Câbler les quatre montages (a, b, c et d) comme le montre les schémas suivants (utiliser le brochage du 7474 et du 74LS73) : PR=1 et CLR=1.



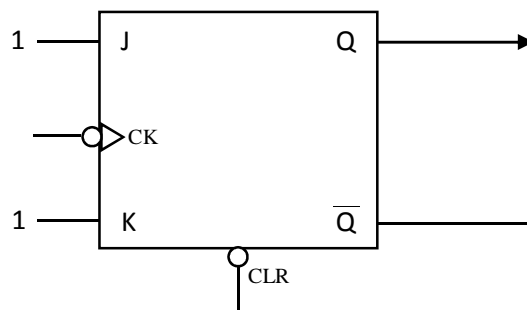
Montage (a) utilisé pour le test de la bascule D



Montage (b) utilisé pour le test de la bascule JK



Montage (c) utilisé pour le test de la bascule JK



Montage (d) utilisé pour le test de la bascule JK

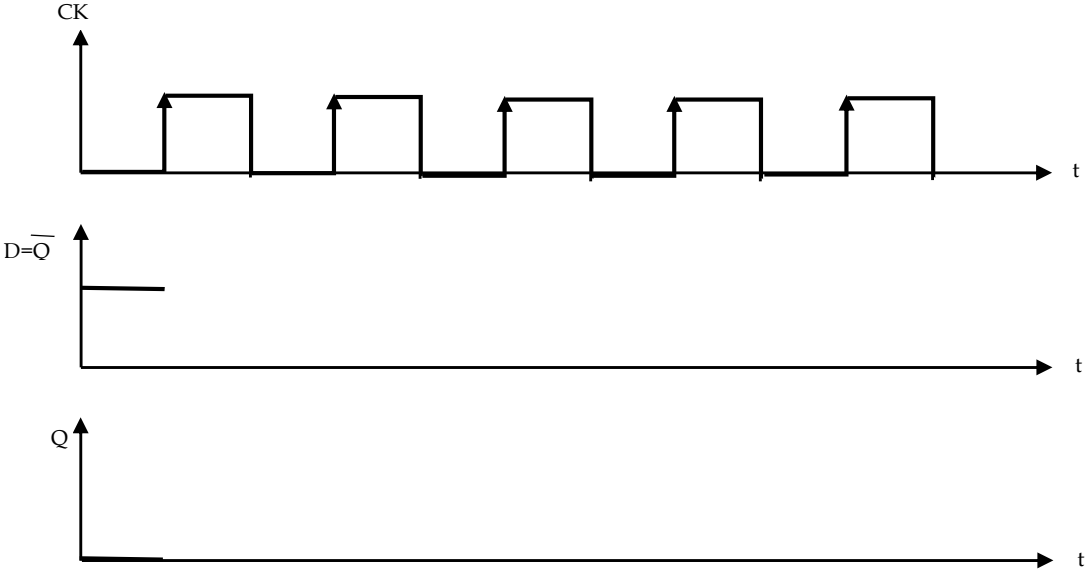
2. Utiliser deux LED. Une LED pour le signal d’horloge et l’autre pour la sortie Q. A l’aide d’un GBF (utiliser sortie TTL), générer un signal d’horloge d’amplitude égale à 5v et de fréquence égale à 1 ou 2Hz. Que remarquer vous ?

3. Compléter expérimentalement la table suivante (visualiser à l’aide de l’oscilloscope le signal d’horloge CK fournie par le GBF et le signal généré par la sortie Q (en reliant la sortie Q au canal CH2 de l’oscilloscope).

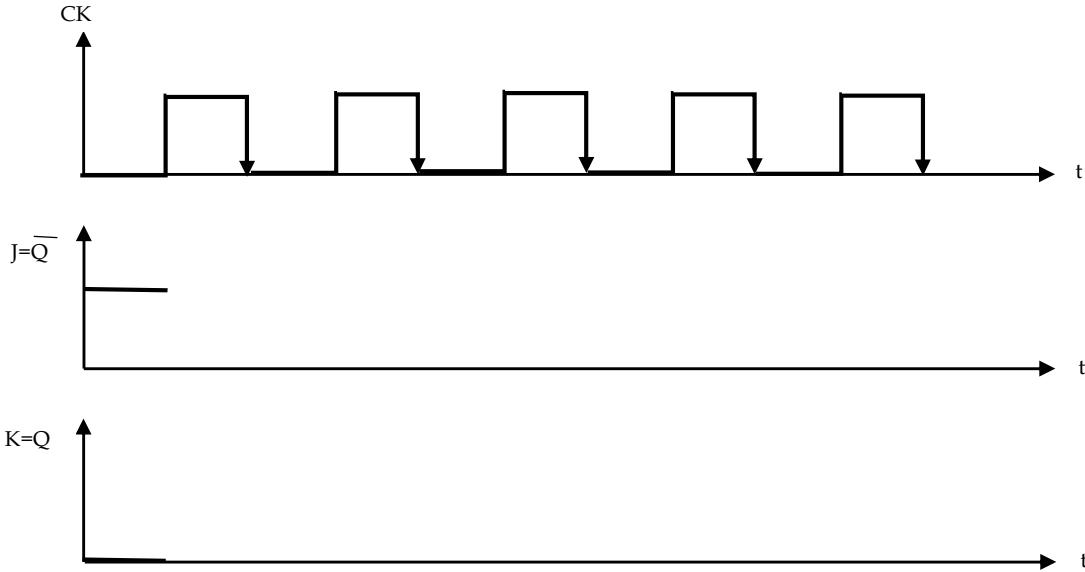
Fck(Hz)	100	200	600	1000
(a)fQ(Hz)				
(b)fQ(Hz)				
(c)fQ(Hz)				
(d)fQ(Hz)				

4. Quelle est la fonction des montages présentés ci-dessus ?.....

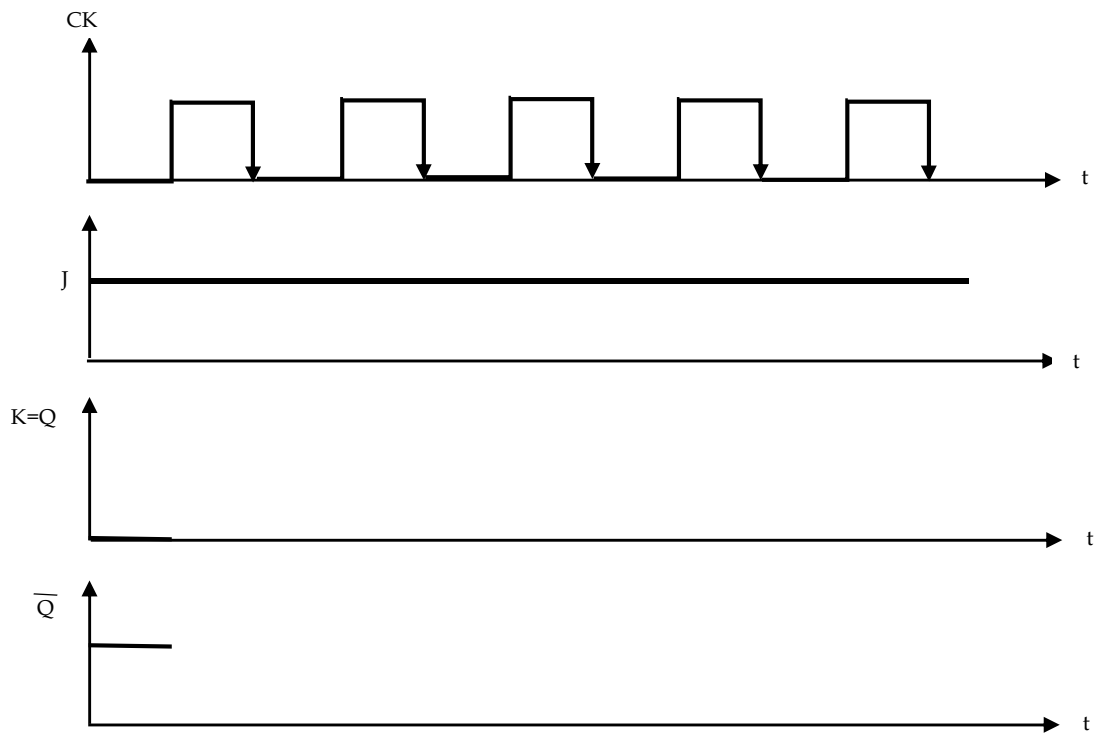
5. En déduire les chronogrammes pour chacun des montages présentés ci-dessus.



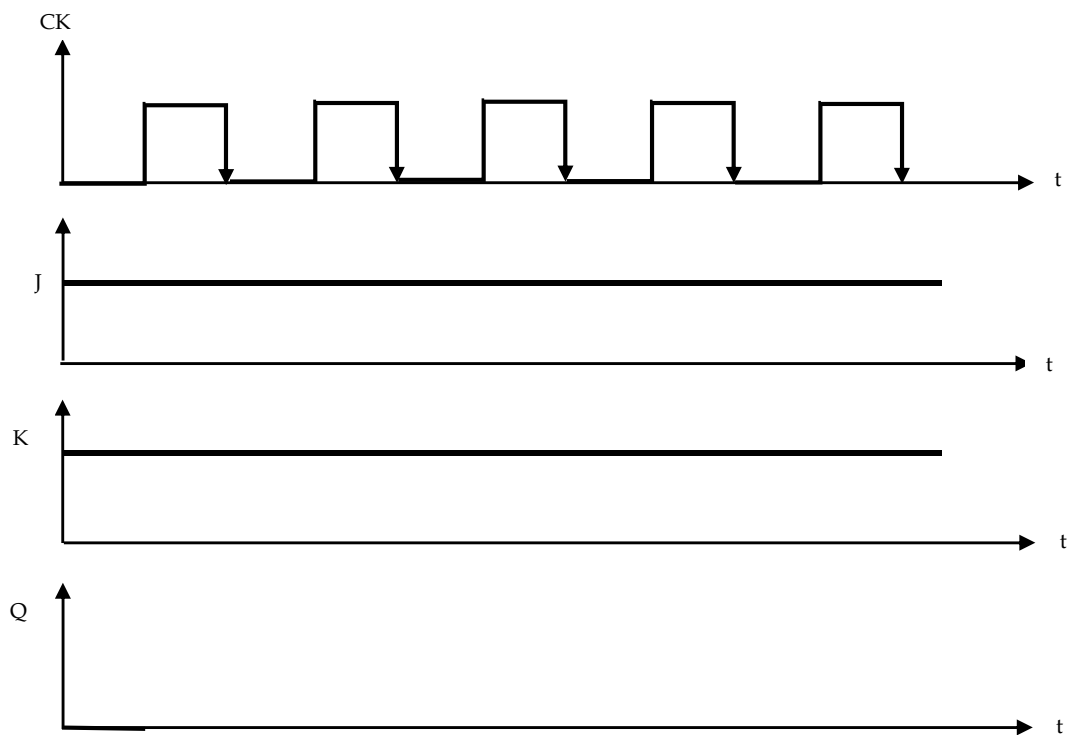
(a)



(b)



(c)



(d)

6. Faire une conclusion ?

Chapitre 7

TP5.2 : Etude et Réalisation de Circuits Compteurs

7.1 Objectifs

1. Construire expérimentalement des compteurs asynchrone incomplet à base de bascules
2. Construire expérimentalement des compteurs synchrone à cycle irrégulier à base de bascules

7.2 Matériels Utilisé

- Alimentation stabilisé 0-5V
- Plaquette d'essais
- Diode Photo-lumineuse (LED)
- Fils de connexion
- Résistances de protection
- Générateur TTL
- Les bascules JK, D, porte NAND
- Porte AND, OR, Inverseur

7.3 Compteurs asynchrone incomplet

7.3.1 Travail Demandé

- Utiliser les bascules JK pour donner les schémas d'un compteur asynchrone modulo 10
- Réaliser le montage que vous avez trouver

- Compléter la table de vérité ci-dessous :

Etat	Q_3	Q_2	Q_1	Q_0
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				

- Prendre ensuite les sorties complémentées \overline{Q} à la place des sorties Q .

Etat	$\overline{Q_3}$	$\overline{Q_2}$	$\overline{Q_1}$	$\overline{Q_0}$

7.4 Compteurs synchrone á cycle irrégulier

7.4.1 Travail Demandé

On va réaliser un compteur que répète la séquence 0, 3, 6, 9, 12, 0... à l'aide des bascules D

- * Etablir la table de vérité de ce compteur
- * Donner la relation entre les états $Q_3Q_2Q_1Q_0$, $D_3D_2D_1D_0$ et l'état suivante de $Q_3Q_2Q_1Q_0$
- * Réaliser les tableaux de Karnaugh de D_3 , D_2 , D_1 , D_0 en fonction de Q_3 , Q_2 , Q_1 , Q_0 .
- * Donner logigramme de compteur
- * Câbler le circuit et vérifier le fonctionnement en plaçant des LED en sortie

7.5 Faire une conclusion ?

Bibliographie

- [1] Letocha; Introduction aux circuits logiques, 2e edition Mc-Graw Hill. 3 novembre 1997
- [2] J.C. Lafont; Cours et problèmes d'électronique numérique, 124 exercices avec solutions; Edition Ellipses, 1992.
- [3] R. Delsol; Electronique numérique, Tomes 1 et 2; Edition Berti
- [4] P. Cabanis; Electronique digitale; Edition Dunod, 1997.
- [5] M. Gindre; Logique combinatoire; Edition Ediscience, 2000.
- [6] M. Gindre, Electronique numérique : logique combinatoire et technologie : cours et exercices, Mc Graw Hill, 1987