

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

BADJI MOKHTAR- ANNABA UNIVERSITY

UNIVERSITE BADJI MOKHTAR ANNABA



جامعة باجي مختار- عنابة

Faculté de technologie

Département Electronique

Brochure de travaux pratiques

TP logique combinatoire et séquentielle

Licence : Automatique et Electronique

Novembre 2022

Présentation

Ce module de logique combinatoire et séquentielle (LCS) fait appel aux préalables suivants: notions de base sur les circuits intégrés, les portes logiques, la réalisation de fonctions logiques combinatoires, et la réalisation des circuits combinatoires arithmétiques, logiques et compteurs.

Description du module

Le but de ce module est d'enseigner la manipulation et la réalisation de différentes portes logiques et le câblage de différents circuits combinatoires et séquentiels en tant que discipline de l'électronique.

Ce support a été préparé conformément au programme de la matière logique combinatoire et séquentielle (LCS) figurant au cursus des étudiants Licence de plusieurs options du département d'électronique. Il constitue également une introduction à la commande des systèmes linéaires et par conséquent, il donne une plateforme intéressante à la logique combinatoire et séquentielle.

TABLES DES MATIERES

Programme

TP1: Technologie des circuits intégrés TTL et CMOS

1. Objectifs.....	1
2- Introduction aux différentes portes logiques.....	1
3- Opérations de base de l’algèbre de Boole.....	2
3.1- Opération NON.....	2
3.2-Porte ET (AND).....	3
3.3-Porte OU (OR).....	3
3.4- Opération NON-OU.....	4
3.5- Opération NON-ET.....	5
3.6- Opération OU Exclusif	6
3.7- Opération NON-OU Exclusif	8
4- Manipulation.....	9

TP2: Simplification des équations logiques par la pratique

1. Objectif.....	10
2. Les lois de MORGAN.....	10
3. Illustration des lois de MORGAN.....	10
4. Manipulation.....	11
5. Exercices 01.....	12
6. Exercice 02.....	13
7. Exercice 03.....	13

TP3: Etude et réalisation de fonctions logiques combinatoires usuelles

1. Objectifs.....	14
2. Multiplexeur.....	14
2.1. Introduction au circuit d'aiguillage (Multiplexeur).....	14

2.2. Manipulation.....	15
2.3 Le démultiplexeur (DEMUX).....	16
2.4 Manipulation.....	16
3. Circuit de décodage.....	16
3.1. Manipulation.....	17
3.2 Décodeur de BCD vers les sept segments.....	17
3.3 Manipulation.....	18
4. Circuit de codage.....	18

TP4: Etude et réalisation d'un circuit combinatoire arithmétique

1. Objectifs.....	19
2. L'Additionneur.....	19
2.1 Demi-additionneur.....	19
2.2 Additionneur complet.....	20
3. Le Soustracteur.....	21
3.1 Demi-soustracteur.....	21
3.2 Soustracteur complet.....	22
4. Manipulation.....	23
4.1 Demi-additionneur.....	23
4.2 Additionneur complet.....	24
4.3 Demi-soustracteur.....	25
4.4 Soustracteur complet.....	26

TP5: Etude et réalisation d'un circuit combinatoire logique

1. Objectifs.....	29
2. Introduction	29
3. Complément à deux.....	29
3.1 Exemple.....	29
3.2 Manipulation.....	30
4. Convertisseur Gray-Binaire naturel.....	30

4.1 Manipulation.....	31
TP6: Etude et réalisation de circuits combinatoire logique	
1. Objectifs.....	32
2. Compateur.....	32
2.1 Première partie.....	32
2.2 Deuxième partie.....	32
3. Manipulation.....	33
TP7: Etude et réalisation de circuits compteurs	
1. Objectifs.....	34
2. Bascule SR	34
3. Manipulation de la bascule SR	35
4. Bascule D.....	36
5. Manipulation de la bascule D	36
6. Bascule T.....	37
7. Manipulation de la bascule T	37
8. Bascule JK.....	38
9. Manipulation de la bascule JK	39
10. Les Compteurs.....	40
11. Manipulation d'un compteur asynchrone	41
TP8: Etude et réalisation des registres	
1. Objectifs.....	43
2. Les registres.....	43
3. Manipulation.....	43
DIGITAL LOGIC LAB	45
Annexe.....	46
Bibliographie.....	55

LISTE DES FIGURES

Figure 1 : porte NOT.....	2
Figure 2: Schéma du circuit intégré.....	2
Figure 3 : porte AND.....	3
Figure 4: Schéma du circuit intégré.....	3
Figure 5 : porte OR.....	4
Figure 6: Schéma du circuit intégré.....	4
Figure 7 : porte NOR.....	4
Figure 8: Schéma du circuit intégré 7402.....	5
Figure 9 : porte NAND.....	5
Figure 10: Schéma du circuit intégré 7400.....	6
Figure 11 : porte OU EXCUSIF.....	6
Figure 12: Schéma du circuit intégré 7486.....	7
Figure 13 : schéma interne d'un ou-exclusif.....	7
Figure 14 : représentation de la porte ou-exclusif a l'aide des portes NAND.....	7
Figure 15 : porte NON- OU EXCLUSIF.....	8
Figure 16: Schéma du circuit intégré 74266.....	8
Figure 17 : réalisation d'une porte NOT à l'aide d'une porte NOR.....	11
Figure 18 : réalisation d'une porte NOT à l'aide d'une porte NAND.....	12
Figure 19: Schéma interne du Multiplexeur.....	15
Figure 20 : afficheur à 7 segments.....	17
Figure 21 : schéma bloc d'un demi-additionneur.....	19
Figure 22 : architecture interne d'un demi- additionneur.....	20
Figure 23: schéma bloc d'un additionneur complet.....	20
Figure 24: réalisation d'un additionneur complet à l'aide de deux demi-additionneur.....	21
Figure 25: architecture interne d'un additionneur complet.....	21
Figure 26: schéma bloc d'un demi-soustracteur.....	21
Figure 27 : réalisation d'un demi-soustracteur à l'aide des portes logique.....	22
Figure 28: schéma bloc d'un demi-soustracteur.....	22
Figure 29: réalisation d'un soustracteur complet à l'aide des portes logique.....	23

Figure 30: réalisation d'un demi-additionneur à l'aide du module KL-33004.....	24
Figure 31: réalisation d'un additionneur complet à l'aide du module KL-33004.....	25
Figure 32: réalisation d'un soustracteur complet à l'aide du module KL-33004.....	26
Figure 33: réalisation d'un additionneur/soustracteur à l'aide du module KL-33004.....	27
Figure 34: schéma typique d'un additionneur/soustracteur à l'aide du module KL-33004.....	28
Figure 35: réalisation d'un comparateur de deux nombre à un bit chacun.....	32
Figure 36: réalisation d'une bascule SR à l'aide d portes NAND.....	35
Figure 37: réalisation d'une bascule SR à l'aide du module KL-33008.....	35
Figure 38: bascule D.....	36
Figure 39: réalisation d'une bascule D à l'aide du module KL-33008.....	36
Figure 40: réalisation d'une bascule T a partir de la bascule SR.....	37
Figure 41: réalisation d'une bascule T à l'aide du module KL-33008.....	38
Figure 42: bascule JK.....	38
Figure 43: schéma interne d'une bascule JK.....	39
Figure 44: réalisation d'une bascule JK à l'aide du module KL-33008.....	39
Figure 45: réalisation d'un compteur à base d bascule JK (diviseur par deux).....	40
Figure 46: réalisation d'un compteur à base d bascule JK (diviseur par quatre).....	41
Figure 47: réalisation d'un compteur asynchrone à l'aide du module K-33009 bloc a.....	42

LISTE DES TABLEAUX

Table 1 : Table de vérité NOT.....	2
Table 2 : table de vérité AND.....	3
Table 3 : Table de vérité OR.....	4
Table 4 : Table de vérité NOR.....	5
Table 5 : Table de vérité NAND.....	5
Table 6 : Table de vérité XOR.....	6
Table 7 : La table de vérité NON- OU.....	8
Table 8 : Table de vérité MUX 2 to 1.....	14
Table 9 : table de vérité BCD vers les 7 segments.....	17
Table 10 : Table de vérité d'un demi-additionneur.....	19
Table 11 : Table de vérité d'un additionneur complet.....	20
Table 12 : Table de vérité demi-soustracteur.....	22
Table 13 : Table de vérité d'un soustracteur complet.....	22
Table 14 : Table de vérité d'un complément à 2.....	29
Table 15 : Table de vérité code GRAY.....	30
Table 16 : Table de vérité d'un comparateur de deux nombres à 4 bits.....	33
Table 17 : Table de vérité d'une bascule SR.....	34

Semestre: 4
Unité d'enseignement: UEM 2.2
Matière 3: TP Logique combinatoire et séquentielle
VHS: 22h30 (TP: 1h30)
Crédits: 2
Coefficient: 1

Objectifs de l'enseignement:

Consolider les connaissances acquises pendant le cours de la matière "Logique Combinatoire et Séquentielle" par des travaux pratiques pour mieux comprendre et assimiler le contenu de cette matière.

Connaissances préalables recommandées

Logique Combinatoire et Séquentielle.

Contenu de la matière :

L'enseignant choisit parmi cette liste de TP entre 4 et 6 TP à réaliser et traitant les deux types de circuits logiques (combinatoire et séquentiel).

TP1 : Technologie des circuits intégrés TTL et CMOS.

Appréhender et tester les différentes portes logiques

TP2 : Simplification des équations logiques par la pratique

Découvrir les règles de simplification des équations dans l'algèbre de Boole par la pratique

TP3 : Etude et réalisation de fonctions logiques combinatoires usuelles

Exemple : les circuits d'aiguillage (MUX, DMUX), les circuits de codage et de décodage, ...

TP4 : Etude et réalisation d'un circuit combinatoire arithmétique

Réalisation d'un circuit additionneur et /ou soustracteur de 2 nombres binaires à 4 bits.

TP5 : Etude et réalisation d'un circuit combinatoire logique

Réalisation d'une fonction logique à l'aide de portes logiques. Exemple un afficheur à 7 segments et/ou un générateur du complément à 2 d'un nombre à 4 bits et/ou générateur du code de Gray à 4 bits, ...

TP6 : Etude et réalisation d'un circuit combinatoire logique

Etude complète (Table de vérité, Simplification, Logigramme, Montage pratique et Essais) d'un circuit combinatoire à partir d'un cahier de charge.

TP7 : Etude et réalisation de circuits compteurs

Circuits compteurs asynchrones incomplets à l'aide de bascules, Circuits compteurs synchrones à cycle irrégulier à l'aide de bascules

TP8 : Etude et réalisation de registres

Mode d'évaluation :

Contrôle continu : 100 %

Références bibliographiques:

1. J. Letocha, Introduction aux circuits logiques, Edition Mc-Graw Hill.
2. J.C. Lafont, Cours et problèmes d'électronique numérique, 124 exercices avec solutions, Edition Ellipses.

TP1: Technologie des circuits intégrés TTL et CMOS

1- Objectifs:

Le but de ce premier TP est:

- De ce familiarisé avec le simulateur utilisé.
- La vérification de quelques portes logiques en utilisant les circuits intégrés et le module KL-33002.

2- Introduction aux différentes portes logiques:

Les opérateurs NON (NOT), ET (AND) et OU (OR) jouent un rôle privilégié dans la mesure où ils sont « génériques », c'est à dire que toute fonction combinatoire peut être exprimée à l'aide de ces opérateurs élémentaires.

Pour chaque opérateur, nous indiquerons le ou les symboles couramment rencontrés, puis nous en donnerons une description sous forme de *table de vérité*, d'expression algébrique, et de *diagramme*

L'algèbre de Boole est un ensemble de variables à deux états de valeurs de vérité 1 (vrai), 0 (faux), qu'on appelle *Variable d'entrée* (les variables d'entrée sont celles sur lesquelles on peut agir directement, ce sont des variables logiques indépendantes), muni d'un nombre limité d'opérateur pour fournir la *Variable de sortie* (variable contenant l'état de la fonction après l'évaluation des opérateurs logiques sur les variables d'entrée).

Dans l'algèbre de Boole on s'intéresse à définir les trois opérations de bases : NON (négation), ET (intersection) et OU (union).

Et :

- de dresser leurs tables de vérités et leurs portes logiques.
- de donner une définition sous forme algébrique.
- d'appliquer les relations de bases de l'algèbre de Boole.
- d'utiliser le théorème de Morgan et de simplifier algébriquement une fonction logique.

3- Opération de base de l'algèbre de Boole:

3.1- Opération NON:

Soit A une variable booléenne (prend la valeur 0 ou 1) la table de vérité ci-dessous pour l'opération NON (inverseur) est donnée par la variable (inversé par rapport à A).

Table 1 : Table de vérité NOT

A	S
0	1
1	0

$S = \bar{A}$, S est vraie si A est fausse

Toutes les fonctions sont implantées à l'aide de portes de la série TTL les plus répondu sont les circuits intégrés TTL série 74 (transistor, transistor logique). La tension d'alimentation est de +5V par rapport à la masse, la tension de sortie possèdent deux niveaux :

- Niveau bas (L) entre 0 et 0.8V.
- Niveau haut (H) entre 3 et 5V.

Son symbole graphique est :

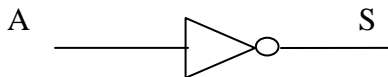


Figure 1 : porte NOT

Cette fonction est implantée par le circuit intégré 7404

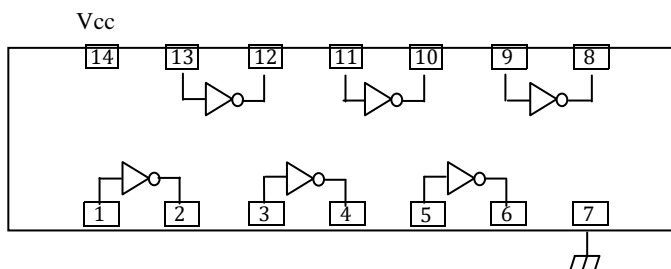


Figure 2: Schéma du circuit intégré 7404

3.2-Porte ET (AND):

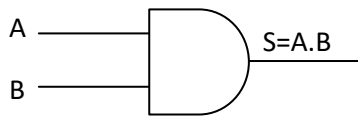
Soit A et B deux variables booléennes. La table de vérité $S = A \cdot B$ est la suivante:

Table 2 : table de vérité AND

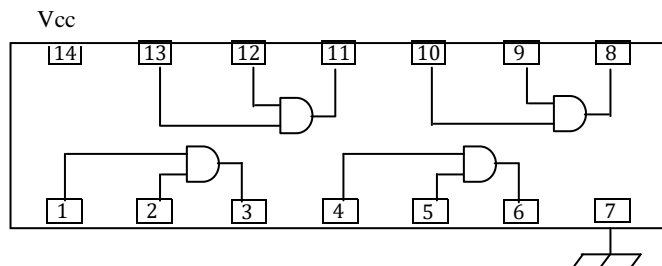
Entrées		Sortie
A	B	$S = A \cdot B$
0	0	0
0	1	0
1	0	0
1	1	1

$S = A \cdot B$, S est vraie si A est vraie et B est vraie.

Son symbole graphique est :

**Figure 3** : porte AND

Cette fonction est implantée par le circuit intégré 7408

**Figure 4**: Schéma du circuit intégré 7408**3.3-Porte OU (OR):**

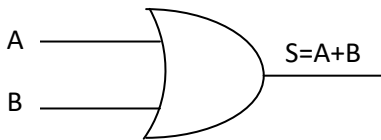
Soit A et B deux variables booléennes. La table de vérité $S = A + B$ est la suivante:

Table 3 : Table de vérité OR

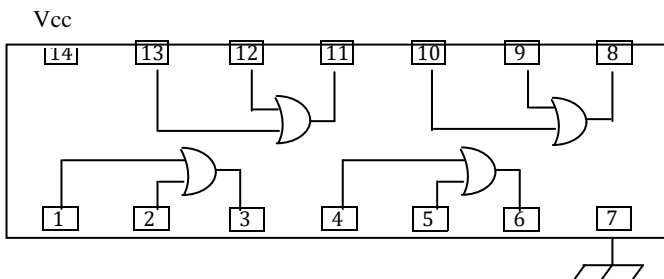
Entrées		Sortie
A	B	$S=A+B$
0	0	0
0	1	1
1	0	1
1	1	1

$S=A+B$, S est vraie si A est vraie ou B est vraie, ou les deux

Son symbole graphique est :

**Figure 5 :** porte OR

Cette fonction est implantée par le circuit intégré 7432

**Figure 6:** Schéma du circuit intégré 7432

3.4- Opération NON-OU:

Est donné par l'équation $S = \overline{A + B}$, sa porte logique est la suivante:

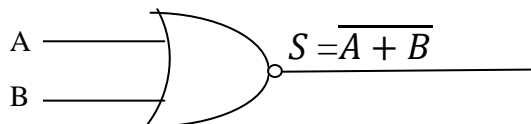
**Figure 7 :** porte NOR

Table 4 : Table de vérité NOR

Entrées		Sorties	
A	B	A + B	$S = \overline{A + B}$
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

$S = \overline{A + B}$, S est vraie si A est fausse et B est fausse.

Cette fonction est implantée par le circuit intégré 7402

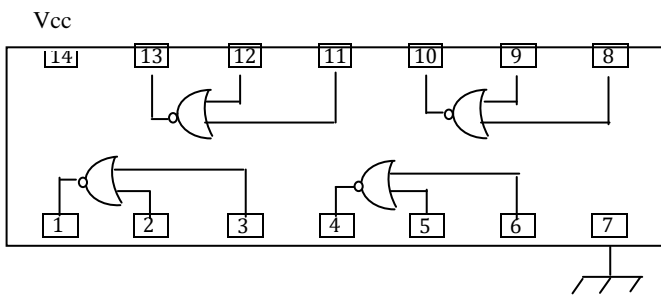


Figure 8: Schéma du circuit intégré 7402

3.5- Opération NON-ET :

Etant donné par l'équation $S = \overline{A \cdot B}$, sa porte logique est la suivante:

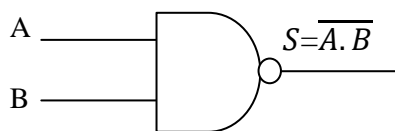


Figure 9 : porte NAND

Table 5 : Table de vérité NAND

Entrées		Sorties	
A	B	A.B	$S = \overline{A \cdot B}$
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

$S = \overline{A \cdot B}$, S est vraie si A est fausse ou B est fausse, ou les deux.

Cette fonction est implantée par le circuit intégré 7400

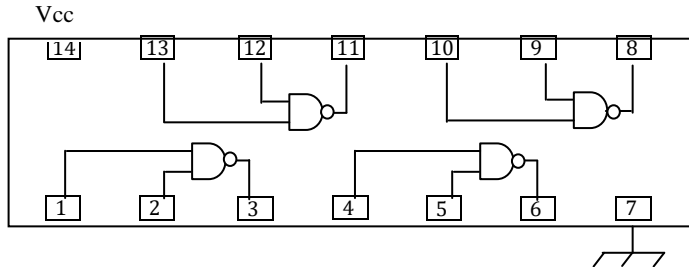


Figure 10: Schéma du circuit intégré 7400

3.6- Opération OU Exclusif : Le « ou exclusif », ou somme modulo 2

Opérateur aux multiples applications, le OU EXCLUSIF (*XOR*) est sans doute l'opérateur à deux opérands le plus riche et le moins trivial. Il trouve ses applications dans les fonctions :

- arithmétiques, additionneurs, comparateurs et compteurs ;
- de contrôle et de correction d'erreurs ;
- où l'on souhaite pouvoir programmer la convention logique ;
- de cryptage de l'information.

Après avoir donné la définition de cet opérateur, nous donnerons quelques exemples de ces applications.

Étant donné par l'équation $S = A \oplus B = \overline{A}B + A\overline{B}$, sa porte logique est la suivante:

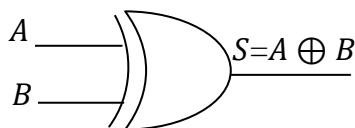


Figure 11 : porte OU EXCUSIF

Table 6 : Table de vérité XOR

Entrées		Sortie
A	B	$S=A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

$S=A \oplus B$, S est vraie si A est vraie ou B est vraie, mais pas les deux.

Cette fonction est implantée par le circuit intégré 7486

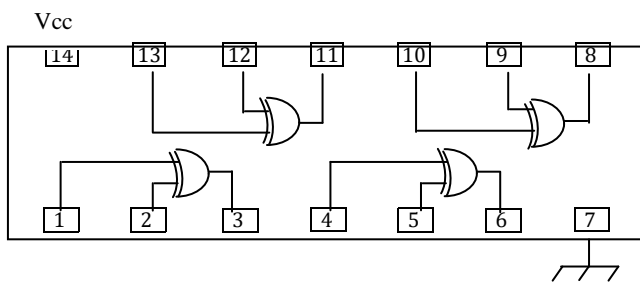
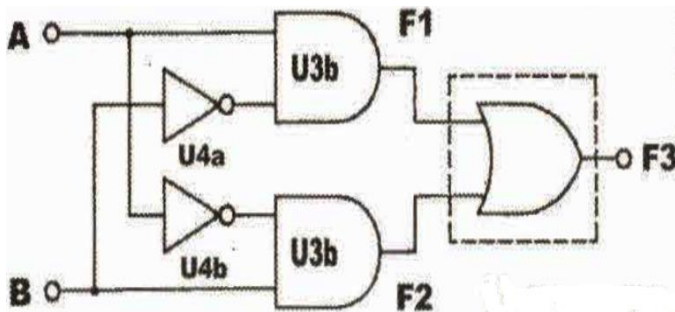


Figure 12: Schéma du circuit intégré 7486



- Figure 13 : schéma interne d'un ou-exclusif

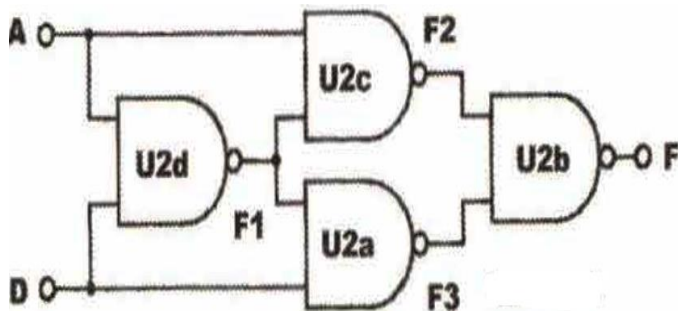


Figure 14 : représentation de la porte ou-exclusif a l'aide des portes NAND

3.7- Opération NON-OU Exclusif :

Comme opérateur à deux opérands, ce nouvel opérateur indique *l'identité* entre les deux opérands, d'où le nom parfois employé pour le désigner de « *coïncidence* ». Comme opérateur généralisé à un nombre quelconque d'opérands, le complément du ou exclusif indique par un '1' qu'un nombre *pair* de ses opérands vaut '1'.

C'est la fonction inverse de la fonction OU exclusif $S = \overline{A \oplus B} = \overline{A} \overline{B} + AB$, sa porte logique est la suivante:

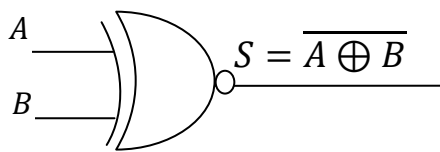


Figure 15 : porte NON- OU EXCLUSIF

Table 7 : La table de vérité NON- OU EXCLUSIF

Entrées		Sortie
A	B	$S = \overline{A \oplus B}$
0	0	1
0	1	0
1	0	0
1	1	1

Cette fonction est implantée par le circuit intégré 74266

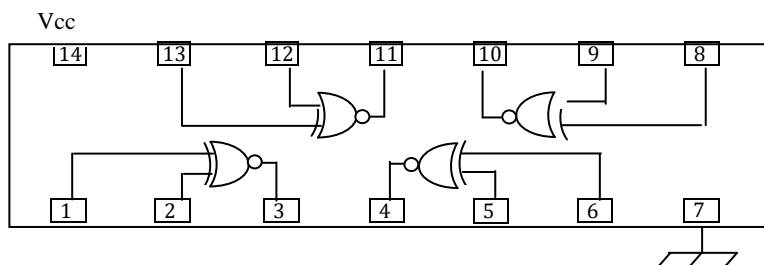


Figure 16: Schéma du circuit intégré 74266

Généralisation

À l'exception des portes XOR et XNOR, ces notions peuvent être généralisées pour des portes à plusieurs entrées.

4-Manipulation

- tester les portes logiques suivante NOT, OR, AND, NOR, NAND et XOR des circuits logiques CI 7404, CI 7408 ; CI 7432, CI 7402, CI 7400, et CI 7486 correspondant, donner leur table de vérité et de vérifier son bon fonctionnement.

TP2: Simplification des équations logiques par la pratique

1- Objectifs:

Le but de ce deuxième TP est:

- De ce familiarisé avec le simulateur utilisé.
- La vérification de quelques portes logiques en utilisant les circuits intégrés et le module KL-33002.
- la vérification des équations simplifiées obtenus sur simulateur

2- Les lois de MORGAN

Les deux lois de De Morgan permettent le passage d'une fonction logique à son complément, elles sont utilisées systématiquement par les logiciels d'aide à la synthèse de circuits logiques, pour déterminer la convention logique qui conduit à l'équation la plus simple qui rende compte d'un problème donné. Les voici :

$$\overline{A + B} = \bar{A} \cdot \bar{B}$$

et :

$$\overline{A \cdot B} = \bar{A} + \bar{B}$$

Bien évidemment, dans les expressions précédentes, A et B peuvent être elles-même des expressions. Sous ces formules apparemment simples se cachent parfois des calculs importants.

3- Une illustration des lois de Morgan

A titre d'illustration des lois de De Morgan, et pour préciser ce que l'on entend par un opérateur générique, montrons qu'une expression quelconque peut être construite en n'utilisant que des opérateurs de type NAND :

$$A + (B \cdot C) = \overline{\overline{A + (B \cdot C)}} = \overline{\bar{A} \cdot \overline{B \cdot C}}$$

La dernière expression ne fait appel qu'à des opérateurs de type NAND, et à des inverseurs qui sont des NAND à une seule entrée.

4-Manipulation : On utilisant le module KL-33002, réaliser le câblage sur simulateur des portes :

- d'une porte NOT n'utilisant qu'une porte NOR.
- d'une porte NOT n'utilisant qu'une porte NAND.
- d'une porte AND n'utilisant que des portes NOR.
- d'une porte AND n'utilisant que des portes NAND.
- d'une porte OR n'utilisant que des portes NAND.
- d'une porte OR n'utilisant que des portes NOR.
- d'une porte XOR n'utilisant que des portes NAND.

Note: utiliser les SWITCH'S du simulateur comme entrées, et les LED'S comme sorties.

Exemple : Réalisation d'une porte NOT avec une porte NOR, et de la réalisation d'une porte NOT avec une porte NAND : en utilisant le bloc KL-33002

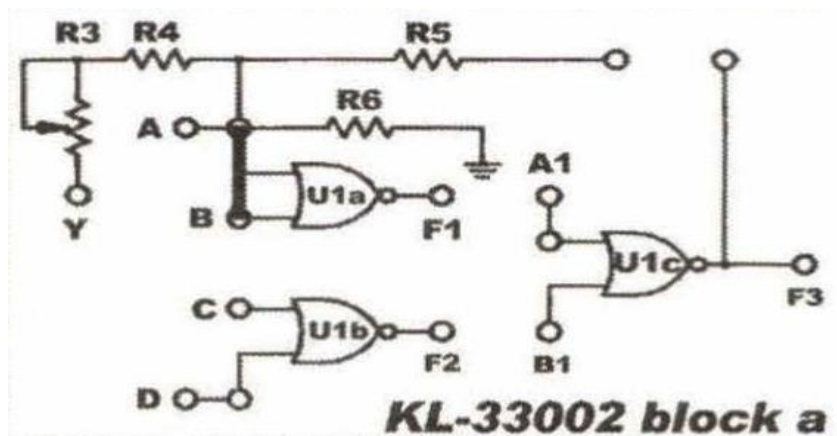


Figure 17 : réalisation d'une porte NOT à l'aide d'une porte NOR

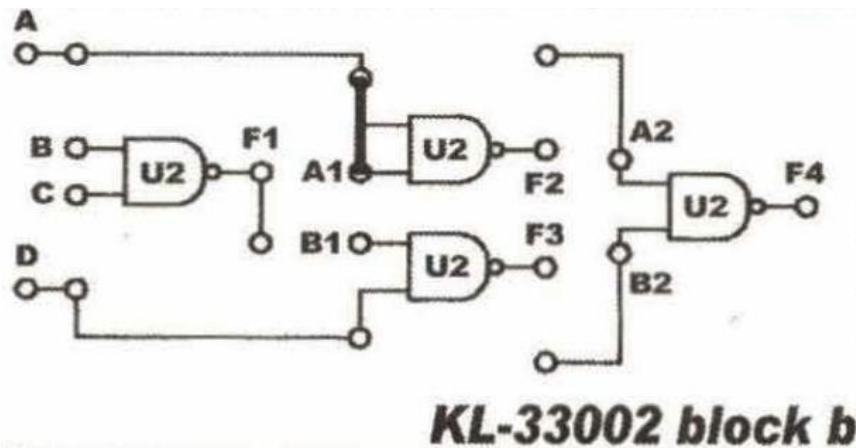


Figure 18 : réalisation d'une porte NOT à l'aide d'une porte NAND

5. Exercice 01:

Soit la table de vérité d'une fonction à deux entrées et une sortie suivante :

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

- Donner l'équation S en utilisant que des porte NAND.
- Réaliser le câblage sur simulateur et vérifier son bon fonctionnement.

$$\text{Soit } S1 = \bar{S}$$

- Donner l'équation de S1 n'utilisant que des portes NOR.
- Réaliser le câblage sur simulateur et vérifier son bon fonctionnement.
- Conclure.

6. Exercice 02 : Soit la table de vérité d'une fonction à trois entrées et une sortie.

A	b	c	F(a, b,c)
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

- Donner le logigramme simplifié en n'utilisant que des portes NAND
- Réaliser le câblage sur simulateur et vérifier son bon fonctionnement.

7. Exercice 03 : Etant donnée la fonction suivante :

$$F(a,b,c,d) = \Sigma (0, 1, 2, 4, 5, 6, 8, 9, 10, 14).$$

- Réaliser de l'inverse de cette fonction sous sa forme simplifiée sur simulateur à partir des résultats de la synthèse.
- Réaliser le câblage sur simulateur en n'utilisant que des portes NAND
- Réaliser le câblage sur simulateur en n'utilisant que des portes NOR

TP3: Etude et réalisation de fonctions logiques combinatoires usuelles

1. Objectifs:

L'étude de quelques sous systèmes combinatoires typiques :
Multiplexeur, de codage et de décodage.

2. Multiplexeur:

2.1. Introduction au circuit d'aiguillage (Multiplexeur):

Le lecteur attentif n'aura pas manqué de remarquer que beaucoup de choses, en logique combinatoire, peuvent s'exprimer par des alternatives SI... ..ALORS... ..AUTREMENT.

Mais quel est donc l'opérateur élémentaire qui, en logique câblée, permet de matérialiser directement ce type de propositions ? Le *sélecteur*, ou *multiplexeur*. Nous donnons ci-dessous la description de sa version la plus simple, quand il n'y a que deux choix possibles dans l'alternative, mais il est bien sûr possible de le généraliser pour représenter des choix multiples.

Le sélecteur est construit comme un opérateur où l'on sépare les variables d'entrée en deux groupes :

- Les entrées de *données*, qui sont en général issues d'autres fonctions ;
- L'entrée de sélection, qui est une *commande*.

Donc le multiplexeur est un circuit qui permet de sélectionner une ligne d'entrée par une adresse et de faire apparaître à la sortie l'état de cette ligne. On général un multiplexeur possède 2^n entrées, n entrées de commande et une seule sortie.

Exemple:

Si $n = 1$, on a une entrée de commande, 2 entrées et une seule sortie. Sa table de vérité est la suivante:

Table 8 : Table de vérité MUX 2 to 1

A	B	S
0	0	e_0
0	1	e_1
1	0	e_2
1	1	e_3

$S = \bar{a}\bar{b}e_0 + \bar{a}be_1 + a\bar{b}e_2 + ab e_3$, d'ou le logigramme suivant:

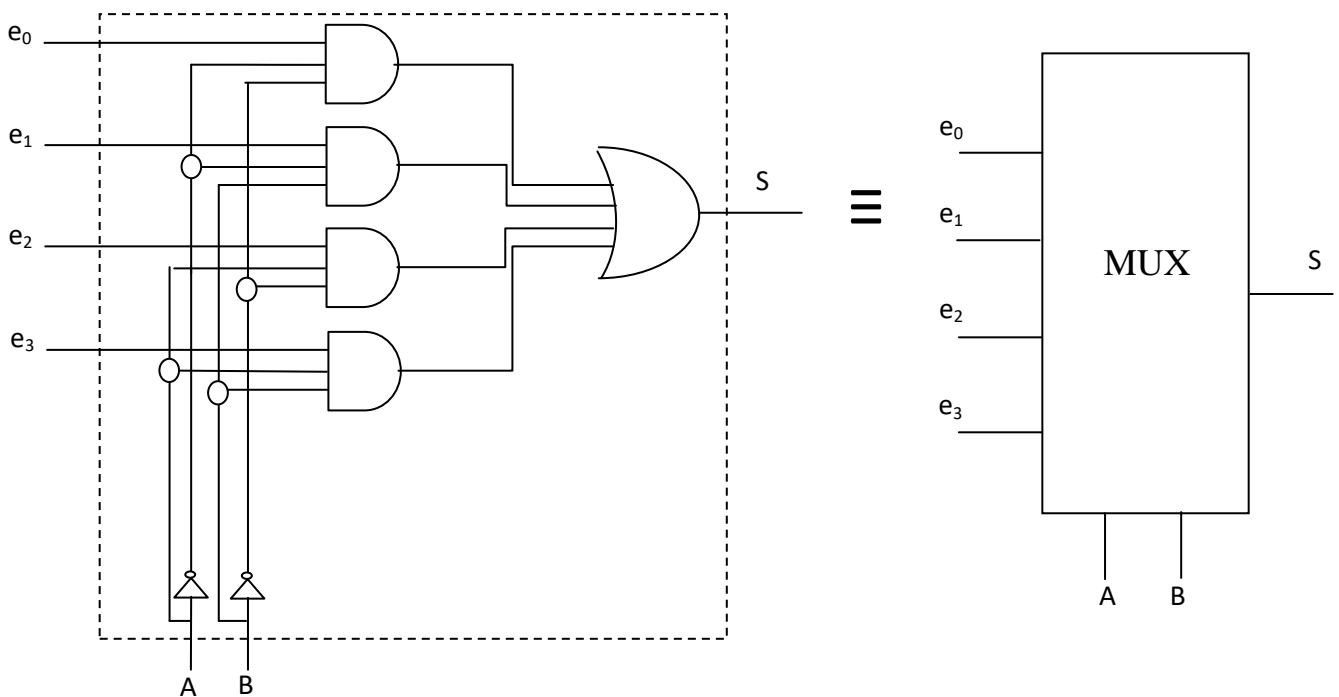


Figure 19: Schéma interne du Multiplexeur

2.2. Manipulation:

a) dresser la table de vérité d'un multiplexeur MUX 2 to 1

- réaliser le logigramme à l'aide des portes NAND.

- vérifier son bon fonctionnement sur simulateur.

Utiliser le module KL-33006 block e, A et B les entrées du MUX, C et le bit de contrôle et F3 la sortie.

b) réaliser un MUX 8 to 1 sur simulateur et vérifier son bon fonctionnement.

Utiliser le module KL-33006 block f

c) Générer la fonction suivante à l'aide d'un MUX 8 to 1.

Quels sont les bits d'adressage (contrôle) qu'il faut choisir pour minimiser les entrées de ce multiplexeur. $F = \sum m(0, 2, 4, 6, 8, 10, 12)$.

Utiliser le module KL-33006 block f, A, B et C les bits de contrôle, D0, D1, ..., D7 les entrées du MUX, Y est la sortie.

2.3 Le démultiplexeur (DEMUX)

Le démultiplexeur est un système combinatoire ayant pour fonction de transmettre une entrée vers une des 2^n sorties. La sélection est faite à l'aide de n lignes d'adresse et les sorties sont mutuellement exclusives. La notation usuelle du DEMUX est: DEMUX 1 à 2^n .

2.4. Manipulation:

a) dresser la table de vérité d'un démultiplexeur DEMUX 1 to 2

- réaliser le logigramme à l'aide des portes NAND.

- vérifier son bon fonctionnement sur simulateur.

Utiliser le module KL-33006 block e, A l'entrée du DEMUX, C et le bit de contrôle, F1 et F2 les sorties.

b) réaliser un DEMUX 1 to 8 sur simulateur et vérifier son bon fonctionnement.

Utiliser le module KL-33006 block b

Utiliser le module KL-33006 block b, A, B et C les bits de contrôle, Y0, Y1, ..., Y7 les sorties du DEMUX, D est l'entrée.

3. Circuit de décodage:

Le décodeur est un système combinatoire ayant pour fonction d'activer une des 2^n sorties.

La sélection est faite à l'aide de n lignes d'adresse et les sorties sont mutuellement exclusives. La notation usuelle du décodeur est: décodeur 1 parmi 2^n . Le décodeur se comporte exactement comme un DEMUX avec son entrée toujours à 1. Par exemple, un décodeur 1 parmi 8 aura 3 lignes d'adresse.

3.1 Manipulation

En utilisant le module KL33005 bloc c.

- Réaliser le décodeur (2^4) suivant : A et B comme entrées, F1, F2, F3 et F4 les sorties.
- dresser la table de vérité
- conclure ?

3.2. Décodeur de BCD vers les sept segments:

Le code BCD a surtout été utilisé aux débuts des machines à calculer. Il est parfois encore utilisé dans des systèmes qui doivent souvent afficher de l'information numérique à l'utilisateur ou pour des systèmes très simples qui manipulent de l'argent. Le code BCD encode le nombre à représenter de façon très directe. Chaque chiffre du nombre est encodé sur 4 bits. Les possibilités binaires de 10 à 15 ne sont pas utilisées.

Les sept segments lumineux, disposés de la façon suivante, permettent l'écriture de tous les chiffres.

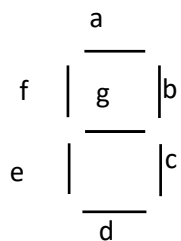


Figure 20 : afficheur à 7 segments

La table de vérité dressée est la suivante:

Table 9 : table de vérité BCD vers les 7 segments

	D	C	B	A	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	0	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	0	0	1	1

Pour faire sortir les équations de ce décodeur, il faut établir la table de Karnaugh de chaque segment.

3.3. Manipulation:

Visualiser le décodeur BCD vers les sept segments en utilisant l'afficheur à 7 segments du simulateur KL-33000 et le module KL 33005 bloc b, connecter les entrées A, B, C, D aux switches, les sorties sont connectées aux entrées de l'afficheur à 7 segments.

4. Circuit de codage:

En utilisant le module KL 33005 bloc a

- Réaliser l'encodeur (4*2) avec A, B, C, D comme entrées et F8 et F9 comme sorties.
- Dresser la table de vérité des sorties F8 et F9 en fonction des entrées (A relié à A1, B à B1, et C à C1)

Refaire la table de vérité et comparer le résultat avec cette fois ci (A1 relié à F1, B1 à F2, et C1 à F3)

TP4: Etude et réalisation d'un circuit combinatoire arithmétique

1. Objectifs:

Notre objectif est de concevoir un circuit qui permet de résoudre des opérations arithmétiques et logiques.

L'étude, la vérification et la réalisation du demi-additionneur (H.A)

L'étude, la vérification et la réalisation du l'additionneur complet (F.A)

L'étude, la vérification et la réalisation du demi-soustracteur (H.S)

L'étude, la vérification et la réalisation du soustracteur complet (F.S)

2. L'Additionneur:

2.1 Demi-additionneur:

Concevoir un circuit capable d'additionner deux bits (variables d'entrées) et qui permet de générer leur somme S et leur report C (variables de sorties).

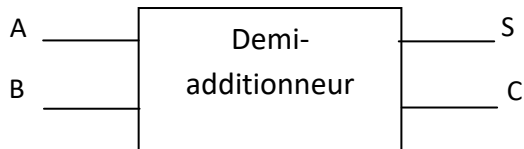


Figure 21 : schéma bloc d'un demi-additionneur

Table 10 : Table de vérité d'un demi-additionneur

Entrées		Sorties	
B	A	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

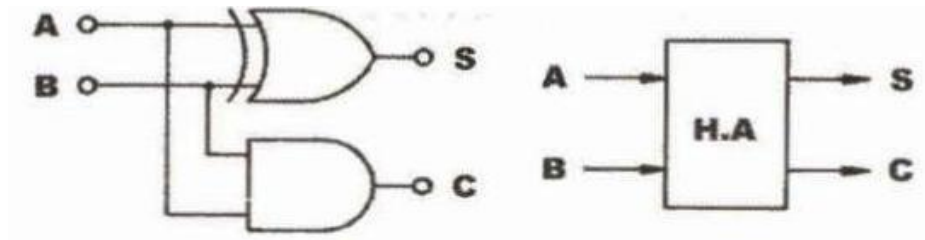


Figure 22 : architecture interne d'un demi- additionneur

2.2 Additionneur complet:

Pour faire l'addition de deux nombres il faut savoir faire la somme de trois chiffres : les chiffres de rang n des deux opérands et le report C_n issu de l'addition des chiffres de rang inférieur ; outre la somme il faut également générer le report C_{n+1} pour l'étage suivant. On appelle *additionneur complet* un tel opérateur

Afin de tenir compte du report de l'étage précédent; il faut donc concevoir un additionneur complet à trois variables d'entrées et qui génère deux variables de sorties.

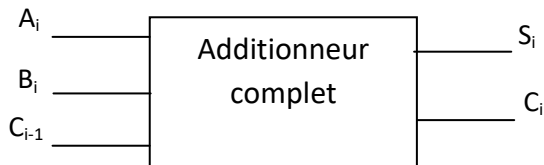


Figure 23: schéma bloc d'un additionneur complet

Table 11 : Table de vérité d'un additionneur complet

Entrées			Sorties	
C_{i-1}	B_i	A_i	S_i	C_i
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

La réalisation de l'addition de deux nombres peut se faire en cascade des opérateurs précédents, on parle alors de « propagation de retenue », ou en calculant « en parallèle » toutes les retenues, au prix d'une complexité non négligeable¹⁹, on parle alors de « retenue anticipée ».

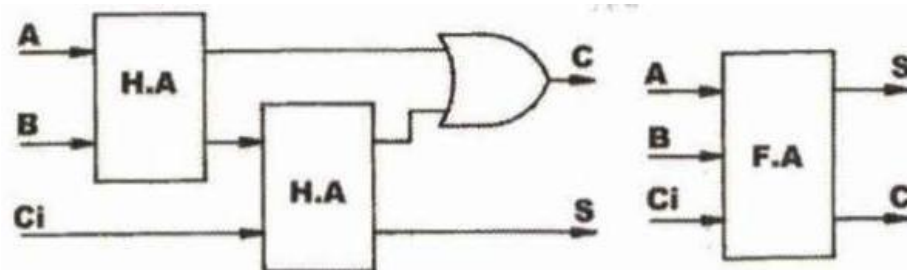


Figure 24: réalisation d'un additionneur complet à l'aide de deux demi-additionneur

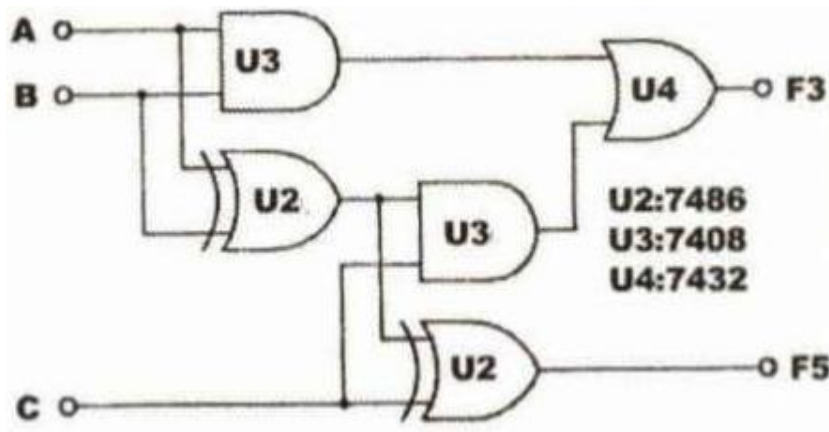


Figure 25: architecture interne d'un additionneur complet

3. Le Soustracteur:

3.1 Demi-soustracteur:

Concevoir un circuit qui permet de soustraire un bit d'un autre et qui génère leur différence D et leur emprunt E.

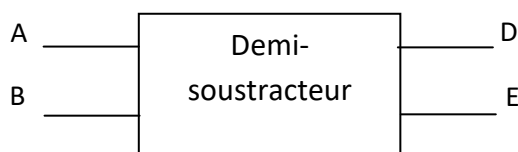
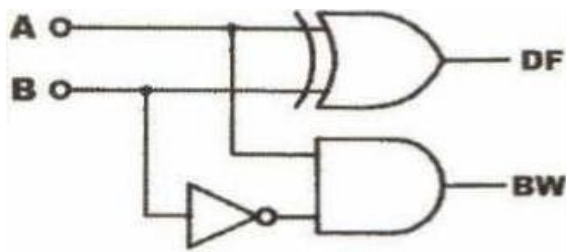


Figure 26: schéma bloc d'un demi-soustracteur

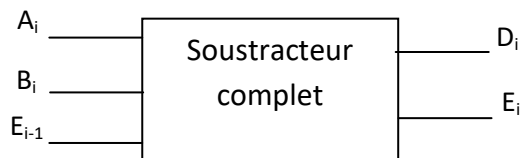
Table 12 : Table de vérité demi-soustracteur

Entrées		Sorties	
B	A	D	E
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

**Figure 27** : réalisation d'un demi-soustracteur à l'aide des portes logiques

3.2 Soustracteur complet:

Pour tenir compte de l'emprunt de l'étage précédent, il faut donc concevoir un soustracteur complet à trois variables d'entrées et deux variables de sorties.

**Figure 28**: schéma bloc d'un demi-soustracteur**Table 13** : Table de vérité d'un soustracteur complet

Entrées			Sorties	
E_{i-1}	B_i	A_i	D_i	E_i
0	0	0	0	0
0	0	1	1	1
0	1	0	1	0
0	1	1	0	0
1	0	0	1	1
1	0	1	0	1
1	1	0	0	0
1	1	1	1	1

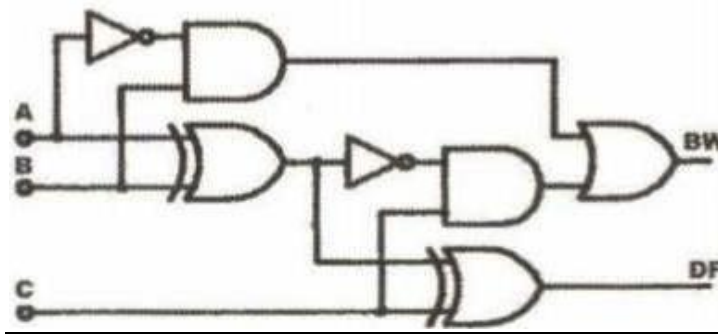


Figure 29: réalisation d'un soustracteur complet à l'aide des portes logiques

4. Manipulation:

4.1 Demi-additionneur:

Il additionne deux chiffres binaires sans tenir compte de la retenue de l'étage précédent. On aura donc:

$$S = A + B, \text{ avec retenue } C.$$

-Dresser la table de vérité.

-Donner la fonction réalisée par S?

-Donner la fonction réalisée par C?

-Donner leurs logigrammes, et réaliser le câblage des deux fonctions S et C sur simulateur, en utilisant le module KL-33004 block a, A et B les deux entrées (relier A et A1), F1 et F2 sont les deux sorties.

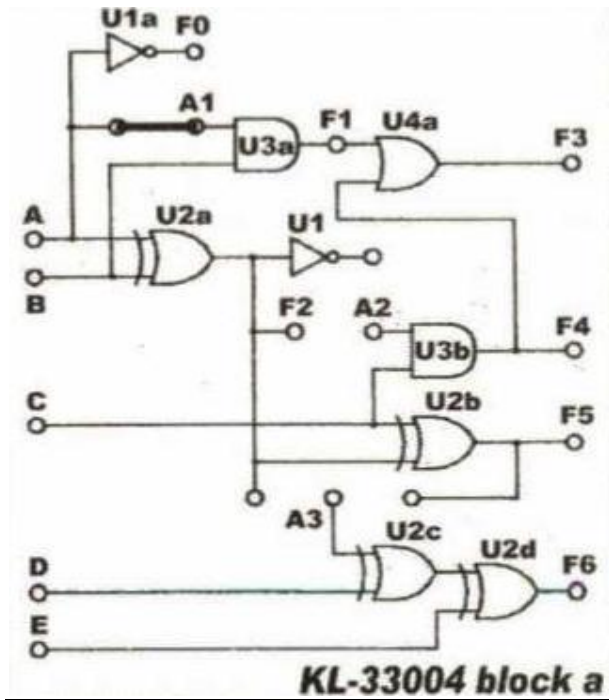


Figure 30: réalisation d'un demi-additionneur à l'aide du module KL-33004

4.2 Additionneur complet:

Le circuit additionneur complet doit tenir compte des deux chiffres A et B, ainsi de l'éventuelle retenue C_{i-1} provenant de l'étage précédent.

On aura donc:

$$S_i = A_i + B_i + C_{i-1}, \text{ avec retenue } C_i.$$

-Dresser la table de vérité.

-Donner la fonction réalisée par S_i ?

-Donner la fonction réalisée par C_i ?

-Donner leurs logigrammes, et réaliser le câblage des deux fonctions S_i et C_i sur simulateur, en utilisant le module KL-33004 block a avec A, B et C les entrées (relier de A1, F2, et A2), les entrées et F3 et F5 les sorties

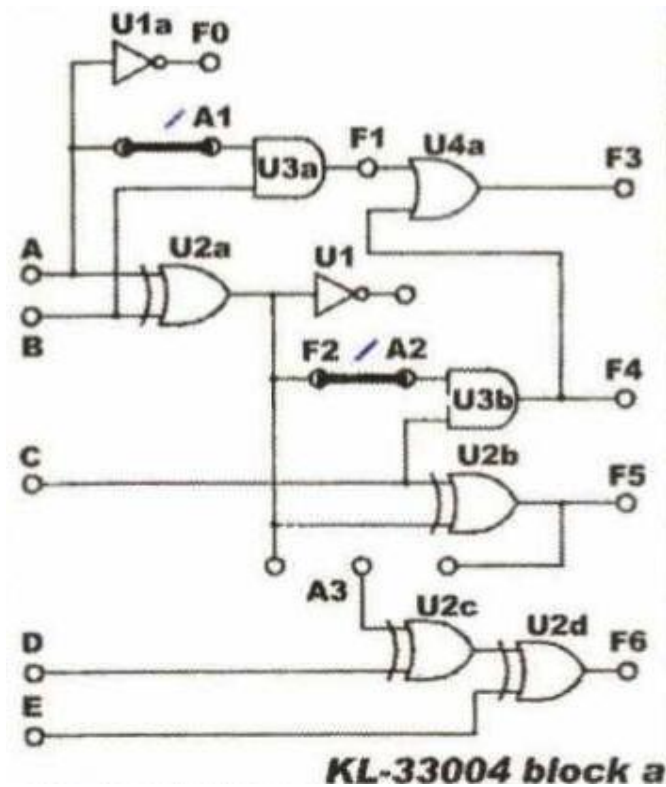


Figure 31: réalisation d'un additionneur complet à l'aide du module KL-33004

4.3 Demi-soustracteur:

C'est un circuit qui effectue la soustraction de deux chiffres binaires sans tenir compte de l'emprunt de l'étage précédent.

On aura donc:

$D = A - B$, avec L'emprunt E.

-Dresser la table de vérité.

-Donner la fonction réalisée par D?

-Donner la fonction réalisée par E?

-Donner leurs logigrammes, et réaliser le câblage des deux fonctions D et E sur simulateur, en utilisant le module KL-33004 block a. A et B les deux entrées (relier A1 et F0), F1 et F2 sont les deux sorties.

4.4 Soustracteur complet:

Ce dernier doit tenir compte de l'emprunt de l'étage précédent E_{i-1} .

On aura donc:

$$D_i = A_i + B_i + E_{i-1}, \text{ avec retenue } E_i.$$

-Dresser la table de vérité.

-Donner la fonction réalisée par D_i ?

-Donner la fonction réalisée par E_i ?

-Donner leurs logigrammes, et réaliser le câblage des deux fonctions D_i et E_i sur simulateur en utilisant le module KL-33004 block a. A, B et C sont les entrées (relier F0, A1 et U1 et A2), F3 et F5 sont les deux sorties.

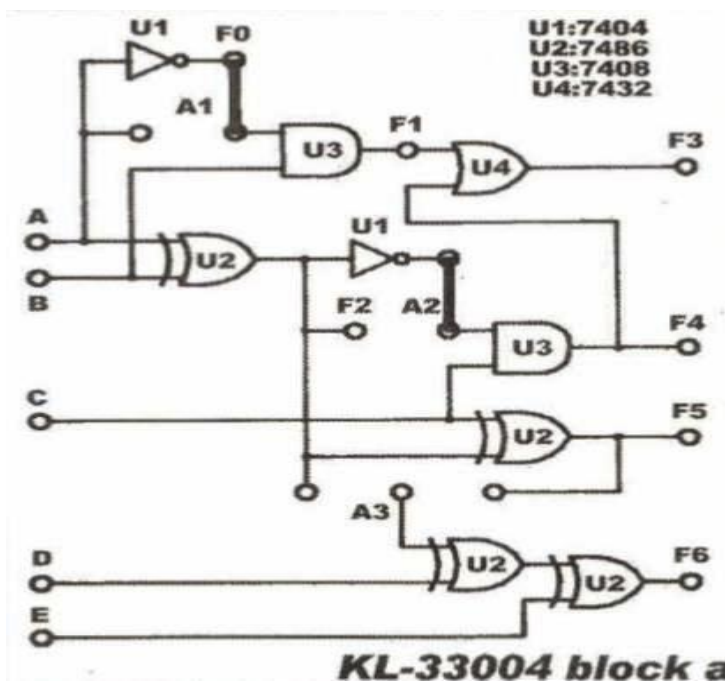


Figure 32: réalisation d'un soustracteur complet à l'aide du module KL-33004

4.5 Additionneur/soustracteur complet

En utilisant le bloc KL-33004, on prend comme entrées le deux nombres de 4 bits , le premier nombre est Y_0, Y_1, Y_2 et Y_3 et le deuxième nombre est $X_0, X_1, X_2,$ et X_3 , la sortie sera un nombre de 4 bits $F_8, F_9, F_{10},$ et F_{11} , la sortie F_1 est réservée au carry.

- Dresser la table de vérité.

Prenez maintenant les même entrées, mais cette fois ci F4, F5, F6 et F7 comme sorties, connectez F1, F2, et F3, au LED. Conclusion ?

Répétez les mêmes étapes précédentes avec cette fois-ci $Y5 = 1$.

Dresser la table de vérité. Conclusion ?

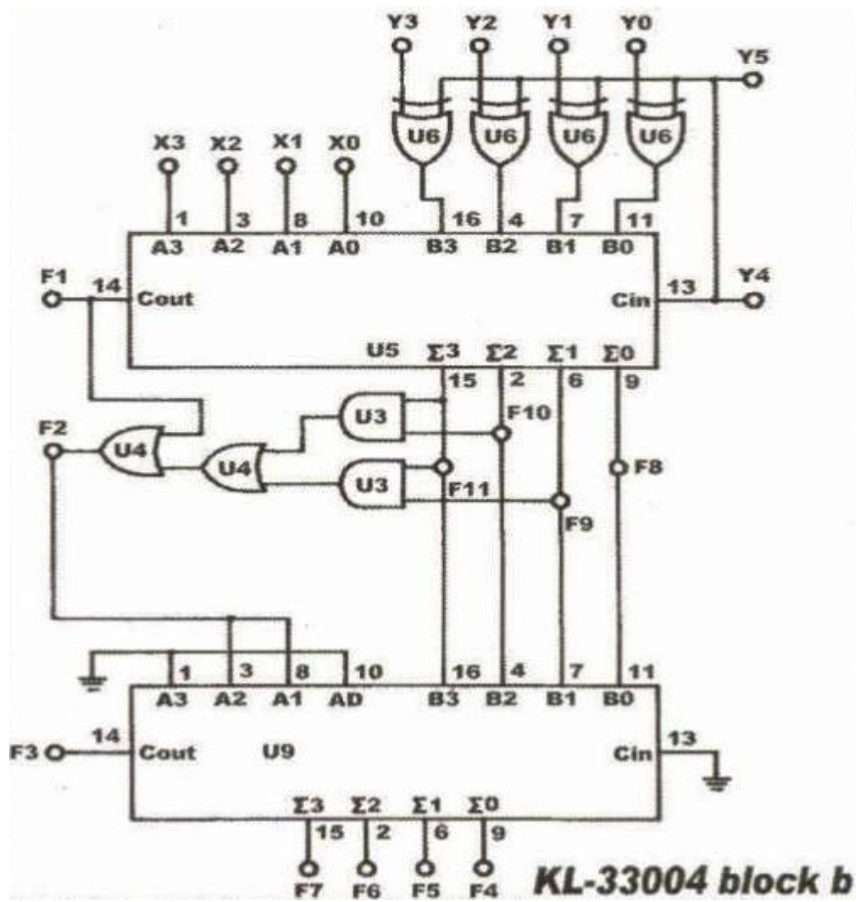


Figure 33: réalisation d'un additionneur/soustracteur à l'aide du module KL-33004

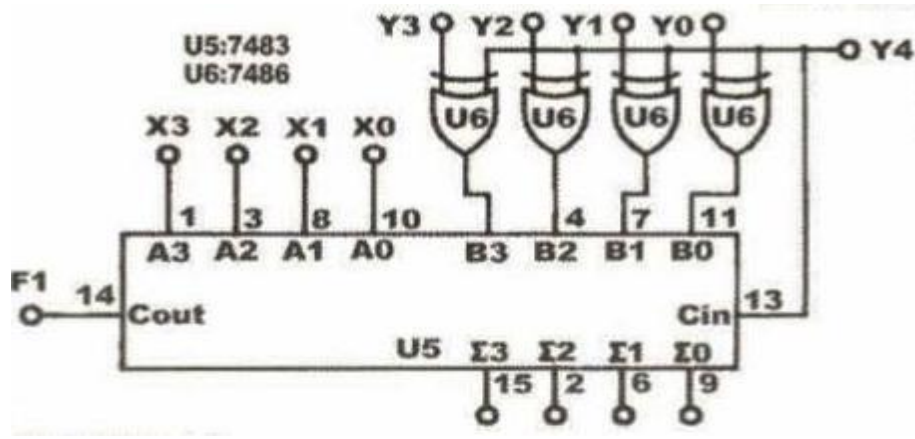


Figure 34: schéma typique d'un additionneur/soustracteur à l'aide du module KL-33004

TP5: Etude et réalisation d'un circuit combinatoire logique

1. Objectifs:

L'étude de quelques sous systèmes combinatoires typiques :

- Complément à deux
- Convertisseur Gray - Binaire.

2. Introduction:

La réalisation d'une fonction logique peut se faire à l'aide des portes logiques. Nous allons entamer dans ce qui suit deux exemples dont lesquels on réalise les fonctions à l'aide des portes logiques.

3. Complément à deux

Dans un code à complément à 2, on obtient le code du négatif d'un nombre en effectuant le complément du mot de code de ce nombre et en lui ajoutant 1 sans garder de retenue. Avec un mot de n bits, il est possible de représenter les valeurs de -2^{n-1} à $2^{n-1}-1$ et il n'existe qu'une représentation possibles pour la valeur 0.

Le code à complément à 2 est très important puisqu'il permet d'effectuer des opérations arithmétiques avec signes sans avoir à se préoccuper de celui-ci.

3.1 Exemple :

Implanter le complément à deux d'un nombre à trois bits.

Table 14 : Table de vérité d'un complément à 2

Entrées			Sorties		
A ₃	A ₂	A ₁	Y ₃	Y ₂	Y ₁
0	0	0	0	0	0
0	0	1	1	1	1
0	1	0	1	1	0
0	1	1	1	0	1
1	0	0	1	0	0
1	0	1	0	1	1
1	1	0	0	1	0
1	1	1	0	0	1

Pour faire sortir les équations de cette table de vérité, il faut établir la table de Karnaugh de chaque sortie.

3.2. Manipulation:

La réalisation d'un complément à deux d'un nombre écrit sur trois bits.

- Dresser la table de vérité.
- trouver les fonctions logiques.
- donner leur logigramme et réaliser le câblage des fonctions sur simulateur.

Utiliser le module KL-33004 block a.

4. Convertisseur Gray-Binaire naturel:

Le code de Gray a été établi pour éviter les problèmes de transition lors du changement d'un mot au mot suivant. Dans l'ordre séquentiel du code, il n'y a qu'un bit qui change entre deux mots consécutifs qui sont, par conséquent, adjacents. De plus, ce code est cyclique.

Conversion du Code Gray de trois bits en code binaire. La table de vérité est la suivante:

Table 15 : Table de vérité code GRAY

-Entrées code GRAY			Sorties code binaire		
G ₃	G ₂	G ₁	B ₃	B ₂	B ₁
0	0	0	0	0	0
0	0	1	0	0	1
0	1	1	0	1	0
0	1	0	0	1	1
1	1	0	1	0	0
1	1	1	1	0	1
1	0	1	1	1	0
1	0	0	1	1	1

Pour faire sortir les équations de cette table de vérité, il faut établir la table de Karnaugh de chaque sortie.

4.1 Manipulation:

Réaliser le transcodeur permettant de transformer des nombres exprimés dans le code Gray écrits sur trois bits en leurs équivalents représentés en binaire à l'aide uniquement de portes OU exclusif.

Utiliser le module KL-33004 block a.

TP6: Etude et réalisation de circuits combinatoire logique

1. Objectifs :

L'étude de table de vérité, simplification montage d'un circuit combinatoire à partir d'un cahier de charge d'un comparateur

2. Comparateur :

2.1 Première partie :

- Dresser la table de vérité d'un comparateur de deux nombres à un bit chacun
- Donner le logigramme réalisé par ces fonctions
- Vérifier son bon fonctionnement sur simulateur, en utilisant le module KL 33002 bloc c, construire le comparateur de deux nombres à un bit chacun en procédant comme suit : A, B sont les entrées à comparer (relier A1, F7 et B1, F6). F1, F2, et F5 sont les sorties.

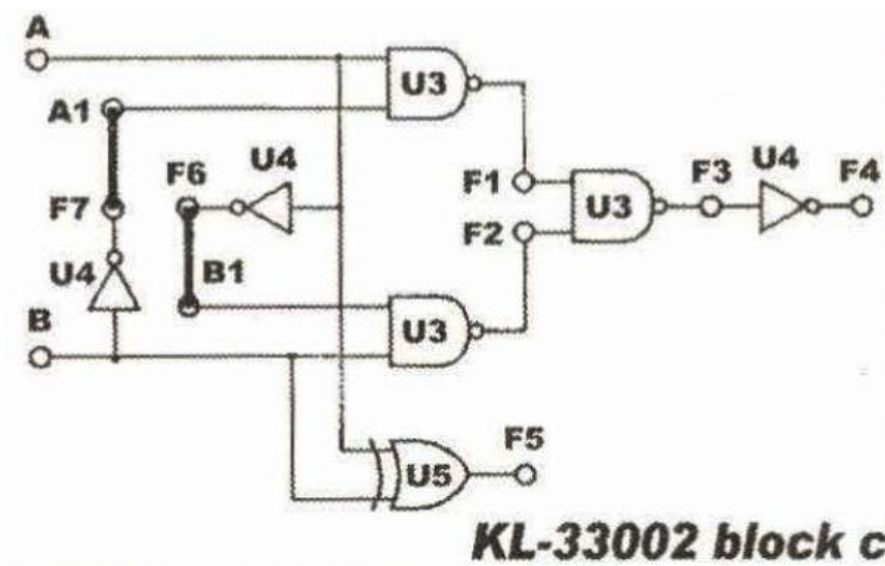


Figure 35: réalisation d'un comparateur de deux nombres à un bit chacun à l'aide du module KL-33004

2.2 Deuxième partie :

Réaliser un circuit qui permet de comparer deux nombres A et B écrits sur quatre bits chacun.

Table 16 : Table de vérité d'un comparateur de deux nombres à 4 bits

a ₄ b ₄ a ₃ b ₃ a ₂ b ₂ a ₁ b ₁	S (A>B)	I (A<B)	E (A=B)
a ₄ > b ₄	1	-	-
a ₄ < b ₄	-	1	-
a ₄ = b ₄ a ₃ > b ₃	1	-	-
a ₃ < b ₃	-	1	-
a ₃ = b ₃ a ₂ > b ₂	1	-	-
a ₂ < b ₂	-	1	-
a ₂ = b ₂ a ₁ > b ₁	1	-	-
a ₁ < b ₁	-	1	-
a ₁ = b ₁	-	-	1

$$S = a_4 \bar{b}_4 + a_4 \oplus b_4 \cdot a_3 \bar{b}_3 + a_4 \oplus b_4 \cdot a_3 \oplus b_3 \cdot a_2 \bar{b}_2 + a_4 \oplus b_4 \cdot a_3 \oplus b_3 \cdot a_2 \oplus b_2 \cdot a_1 \bar{b}_1$$

$$I = \bar{a}_4 b_4 + a_4 \oplus b_4 \cdot \bar{a}_3 b_3 + a_4 \oplus b_4 \cdot a_3 \oplus b_3 \cdot \bar{a}_2 b_2 + a_4 \oplus b_4 \cdot a_3 \oplus b_3 \cdot a_2 \oplus b_2 \cdot \bar{a}_1 b_1$$

$$E = a_4 \oplus b_4 \cdot a_3 \oplus b_3 \cdot a_2 \oplus b_2 \cdot a_1 \oplus b_1$$

3. Manipulation:

- Dresser la table de vérité d'un comparateur de deux nombres à quatre bits chacun
- Donner le logigramme réalisé par ces fonctions
- Vérifier son bon fonctionnement sur simulateur

Utiliser le module KL-33002 block d.

TP7: Etude et réalisation de circuits compteurs

1. Objectifs :

L'étude, la vérification et la réalisation de bascule SR à l'aide de portes logiques

L'étude, la vérification et la réalisation de bascule D

L'étude, la vérification et la réalisation de bascule T

L'étude, la vérification et la réalisation de bascule JK

Construire des compteurs à base de bascules JK.

2. Bascule SR :

Une mémoire est un réseau combinatoire ayant pour fonction d'emmagasiner de l'information. La mémoire est une matrice de cellules capable d'emmagasiner une information unitaire (bit) qui a une adresse spécifique unique.

Dans la logique séquentielle on opte à utiliser les composants à mémoire comme les bascules. C'est une cellule de mémoire qui conserve l'état commande.

La bascule S/R est un élément de mémoire dont la sortie peut être établie à un état actif ou inactif en forçant des conditions temporaires sur les entrées S et R. L'entrée S sert à mettre la sortie à 1 et l'entrée R met la sortie à 0. Si aucune des entrées n'est active, la sortie conserve la valeur courante. De plus, il est illogique que les deux entrées soient actives en même temps. Ce cas est à éviter en tout temps lors que ce composant est utilisé comme bascule.

La table de vérité de ce circuit est la suivante :

Table 17 : Table de vérité d'une bascule SR

\bar{S}	\bar{R}	Q
0	0	Φ interdit
0	1	1 écriture
1	0	0 effacement
1	1	1 0 conserve l'état

$$Q_+ = S + Q \cdot \bar{R}$$

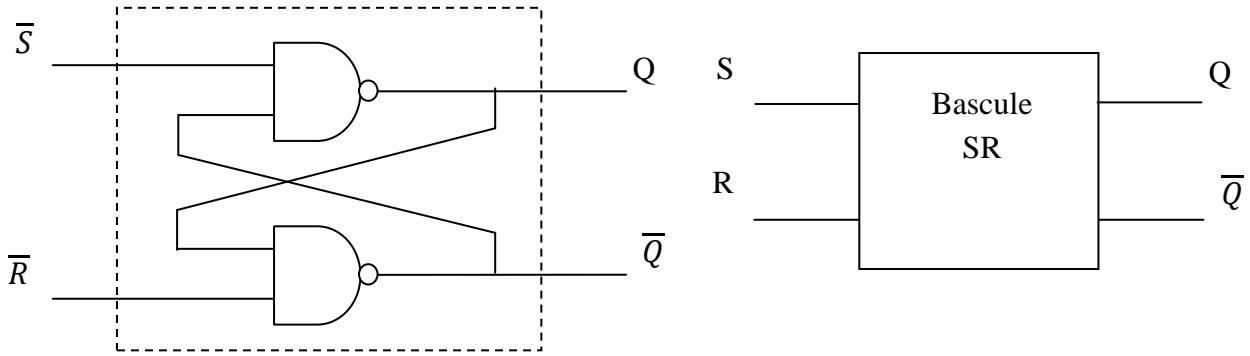


Figure 36: réalisation d'une bascule SR à l'aide d portes NAND

3. Manipulation de la bascule SR:

En utilisant le module KL-33008 block d, réaliser le montage de la figure 2 sur simulateur. En utilisant A3 et A4 comme des entrées, F6 et F7 comme des sorties.

Dresser la table de vérité de cette bascule

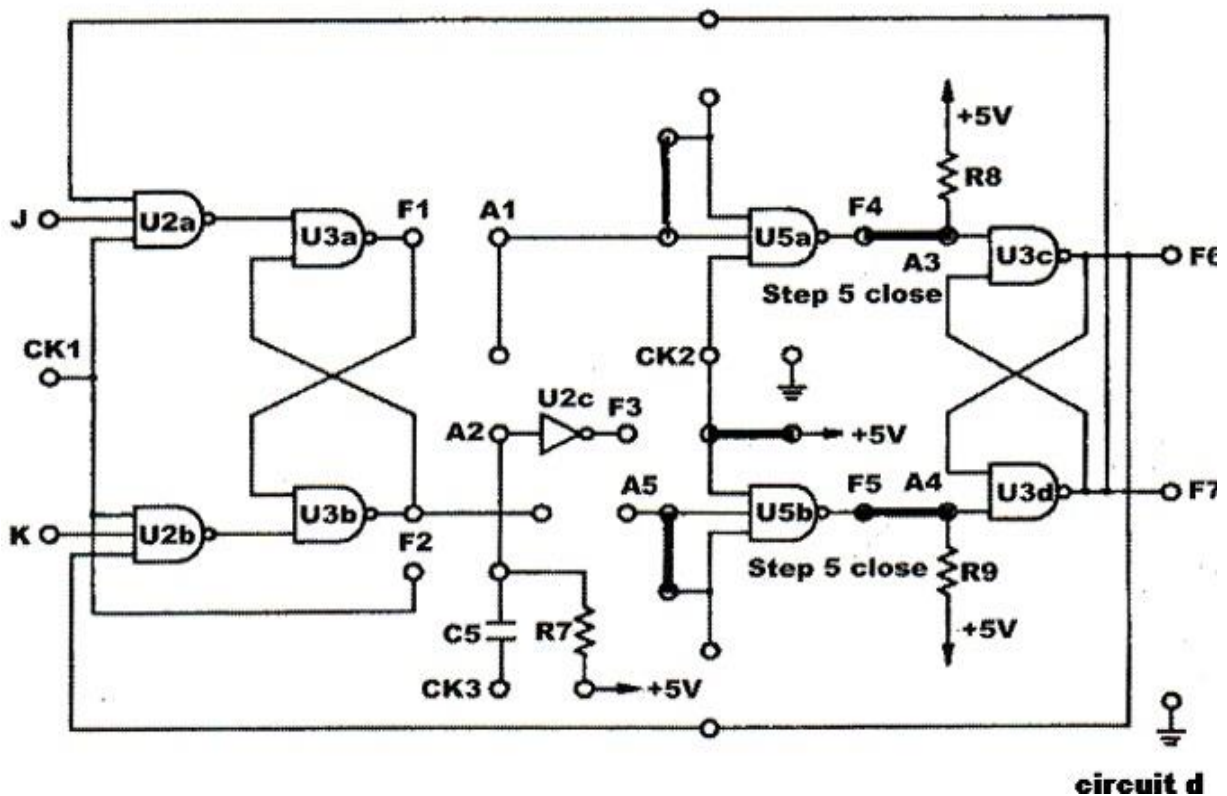


Figure 37: réalisation d'une bascule SR à l'aide du module KL-33008

4. Bascule D:

La bascule D est la plus intuitive puisque son excitation est égale à son résultat. L'horloge est active sur une transition et l'entrée D est copiée à la sortie à ce moment.

Elle n'a que deux sortie soit 1 soit 0, sa table de vérité est donnée par:

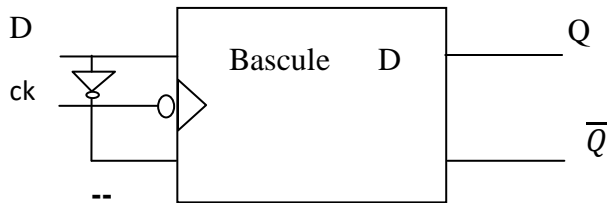


Figure 38: bascule D

5. Manipulation de la bascule D:

En utilisant le module KL-33008 block d, réaliser le montage de la figure 4 sur simulateur. N'utilisant A1 comme entrée (relier A1, A2, et F, A5), et F6 comme sortie

Dresser la table de vérité de cette bascule

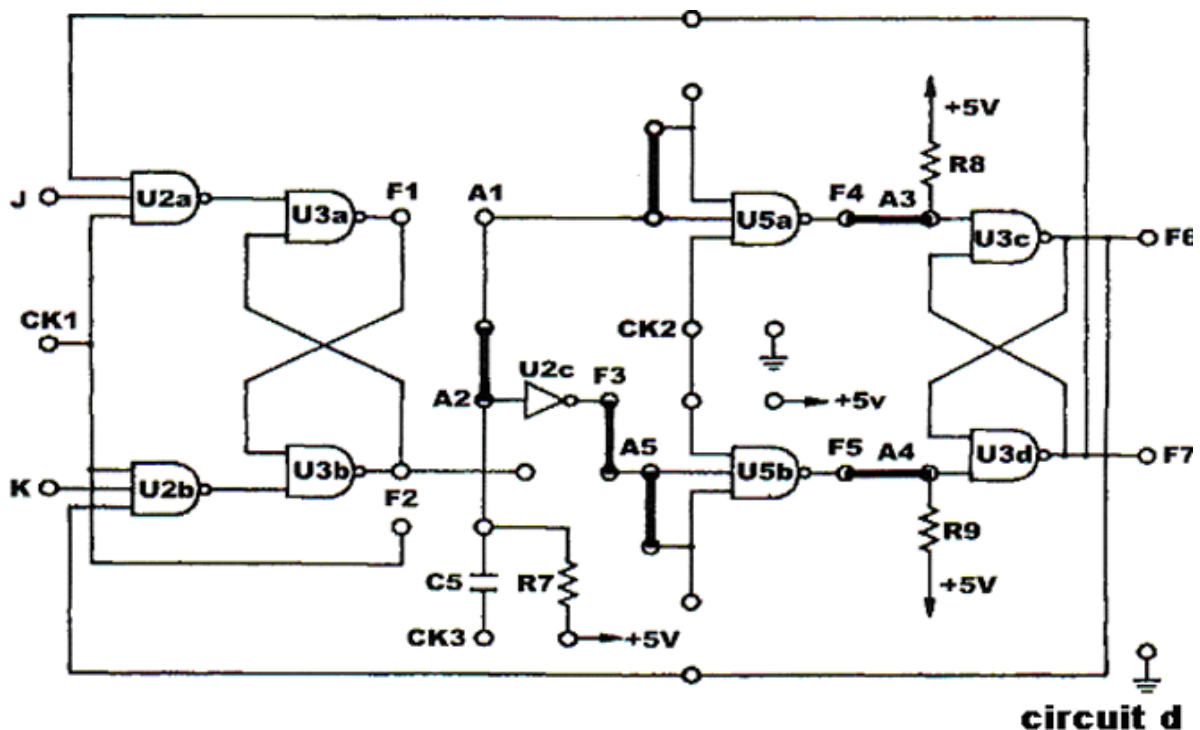


Figure 39: réalisation d'une bascule D à l'aide du module KL-33008

6. Bascule T:

La bascule T (*toggle*) est un élément de mémoire n'ayant qu'une entrée T. Lors d'une transition de l'horloge, la sortie sera inversée si T est actif et conservera son état sinon.

C'est une bascule qui injecte les sorties à l'entrée.

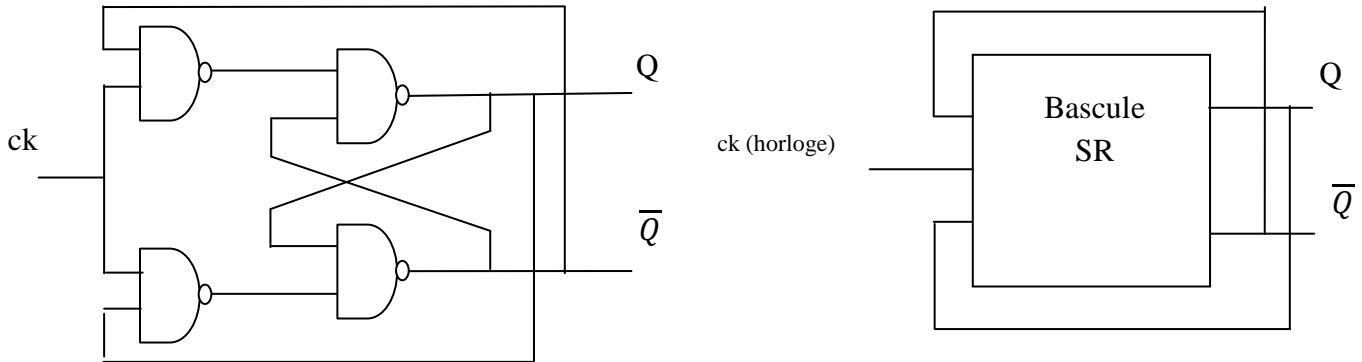


Figure 40: réalisation d'une bascule T a partir de la bascule SR

7. Manipulation de la bascule T:

En utilisant le module KL-33008 block d, réaliser le montage de la figure 6 sur simulateur. En utilisant A1 et A5 comme des entrées (relier F4, A3 et F5, A4), les sorties sont F6 et F7.

Dresser la table de vérité de cette bascule

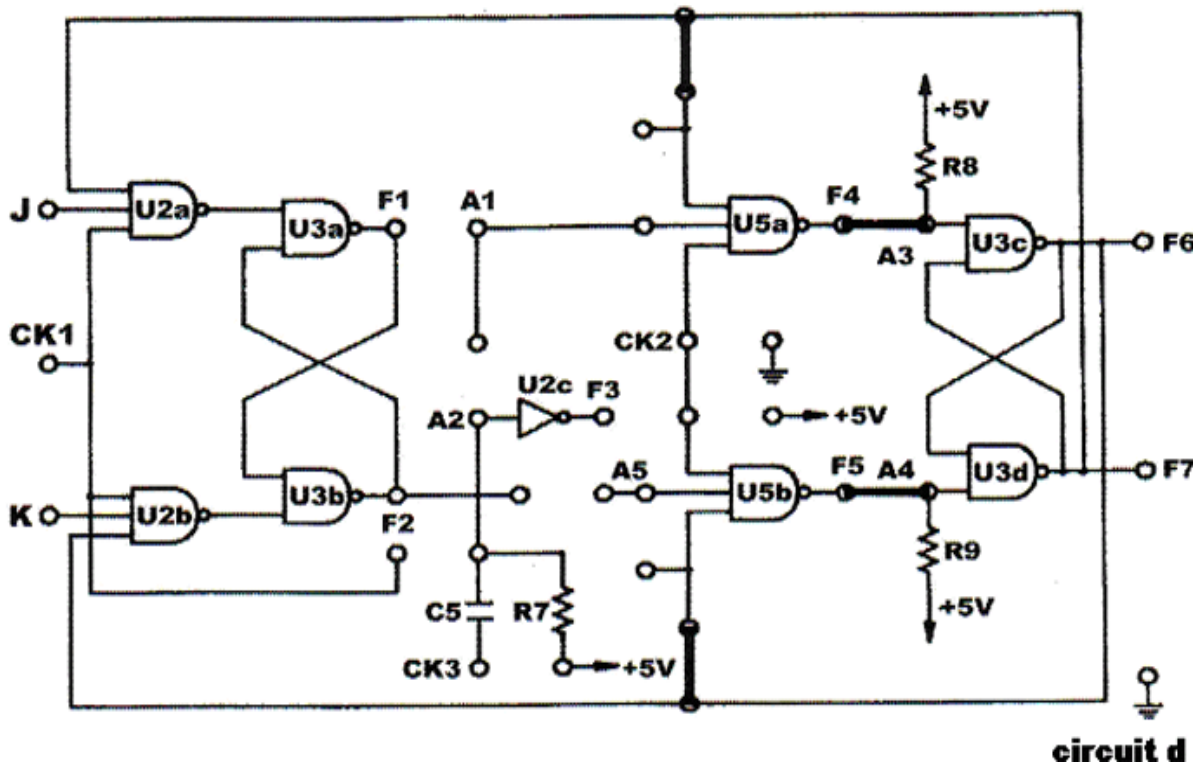


Figure 41: réalisation d'une bascule T à l'aide du module KL-33008

8. Bascule JK:

La bascule JK est la plus complexe mais aussi la plus versatile. Son comportement est commandé par deux signaux qui permettent de recréer le comportement de toutes les bascules vues jusqu'ici. Lors d'un design, l'utilisation des bascules JK demande un peu plus de travail pour le concepteur mais résulte habituellement en un nombre minimal de composants. Contrairement aux autres éléments de mémoire jusqu'ici, l'horloge est active sur une transition de 0 vers 1.

Certains composants utilisent une transition de 1 vers 0 mais aucun composant n'est actif sur les deux transitions.

La bascule JK est utilisé pour éliminer les cas interdit dans la bascule SR, sa table de vérité est la suivante :

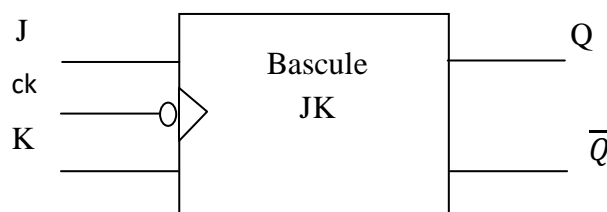


Figure 42: bascule JK

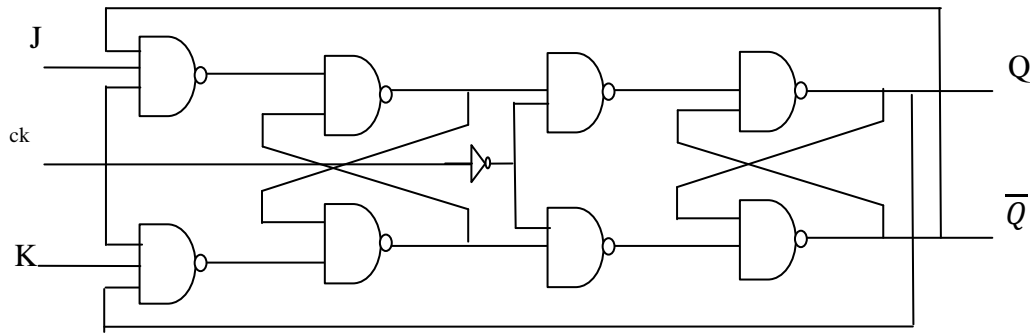


Figure 43: schéma interne d'une bascule JK

9. Manipulation de la bascule JK:

En utilisant le module KL-33008 block d, réaliser le montage de la figure 8 sur simulateur. SW0 (J) SW1 (K) sont les entrées de la bascule (relier F1, A1 et F2, A5), les sorties sont F1, F2, F6, et F7.

Dresser la table de vérité de cette bascule.

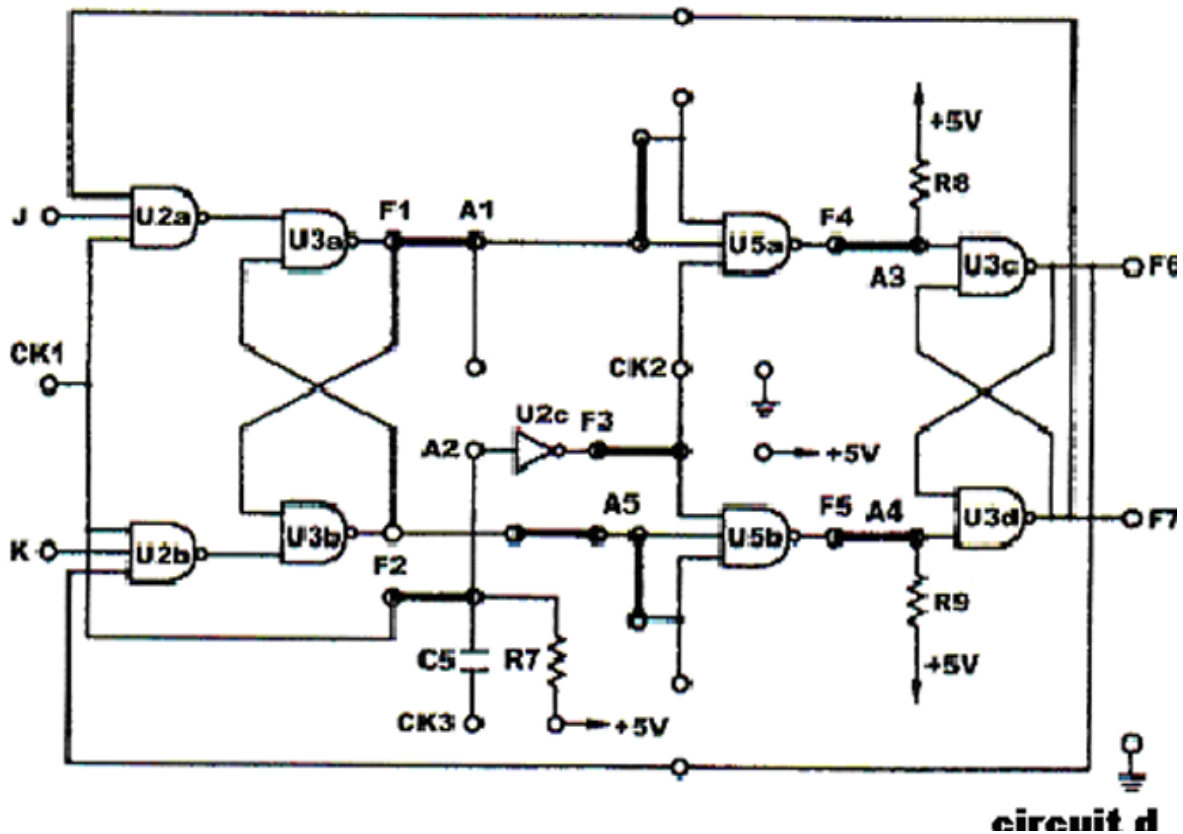


Figure 44: réalisation d'une bascule JK à l'aide du module KL-33008

10. Les Compteurs:

Les compteurs ne sont pas des systèmes combinatoires, ce sont des composants permettant d'emmagasiner et de traiter de l'information.

Pour une même combinaison d'entrées, ce type de composant peut produire des sorties différentes, selon l'information emmagasinée.

Le compteur dispose de deux signaux importants: le signal de compte est le résultat qui est représenté sur n bits. Sur une transition spécifique du signal de compte, habituellement la transition du "0" vers le "1", un incrément de 1 est ajouté au résultat. Même si la majorité des compteurs suit un ordre qui est binaire naturel, il est possible de trouver des compteurs avec d'autres séquences. Lorsque le compteur est rendu à la limite représentable de sa séquence,

Un compteur Modulo N avec $N = 2^n$ compte de 0 à N-1, il comporte n bascules placées en cascade. La bascule change d'état à chaque fois que $ck=1$, on a : $Q_+ = \bar{Q}$.

Il existe deux types de compteurs :

- Compteurs synchrones : Toutes les bascules sont commandées par le même signal d'horloge.
- Compteurs asynchrones : La première bascule reçoit le signal d'horloge. Toutes les bascules qui suivent celle-ci sont commandées par la bascule précédente.

Les compteurs à cycle complet est un compteur dont le modulo est multiple de 2.

Les compteurs à cycle incomplet un compteur dont le modulo ne sont pas multiple de 2.

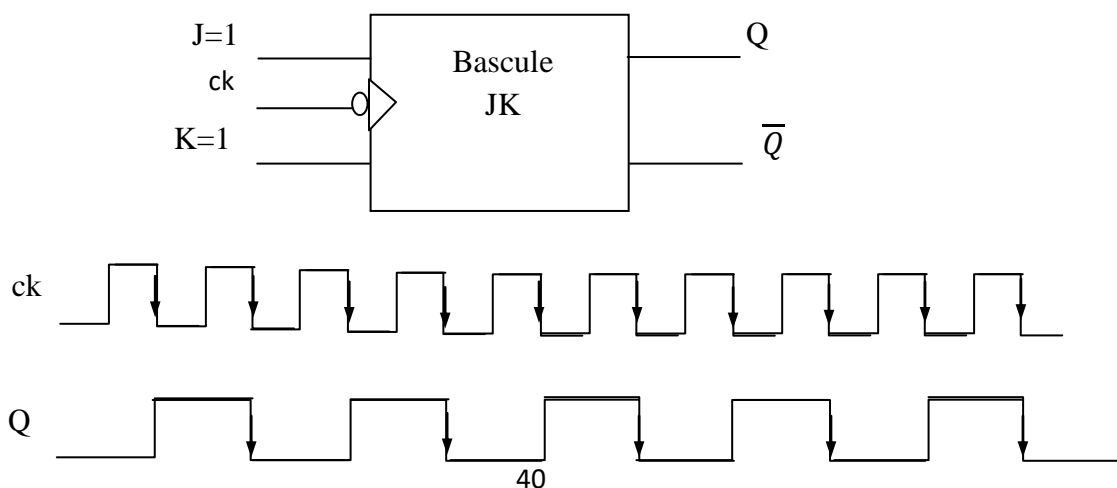
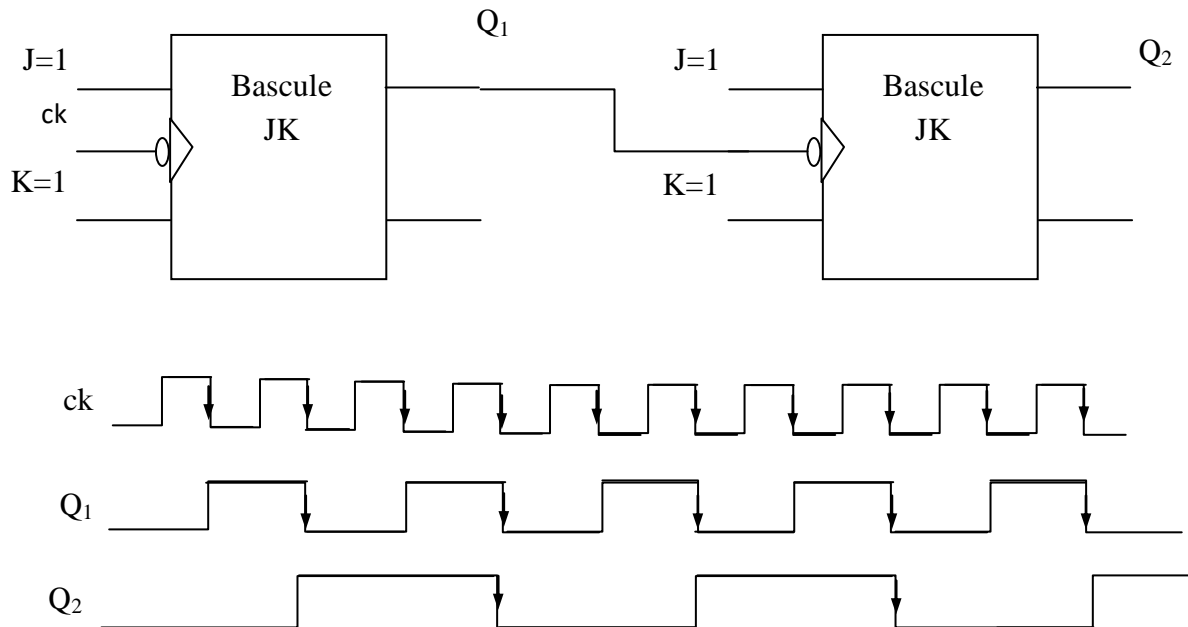


Figure 45: réalisation d'un compteur à base d bascule JK (diviseur par deux)

C'est un diviseur par deux, pour obtenir un diviseur par quatre on adopte le schéma suivant :

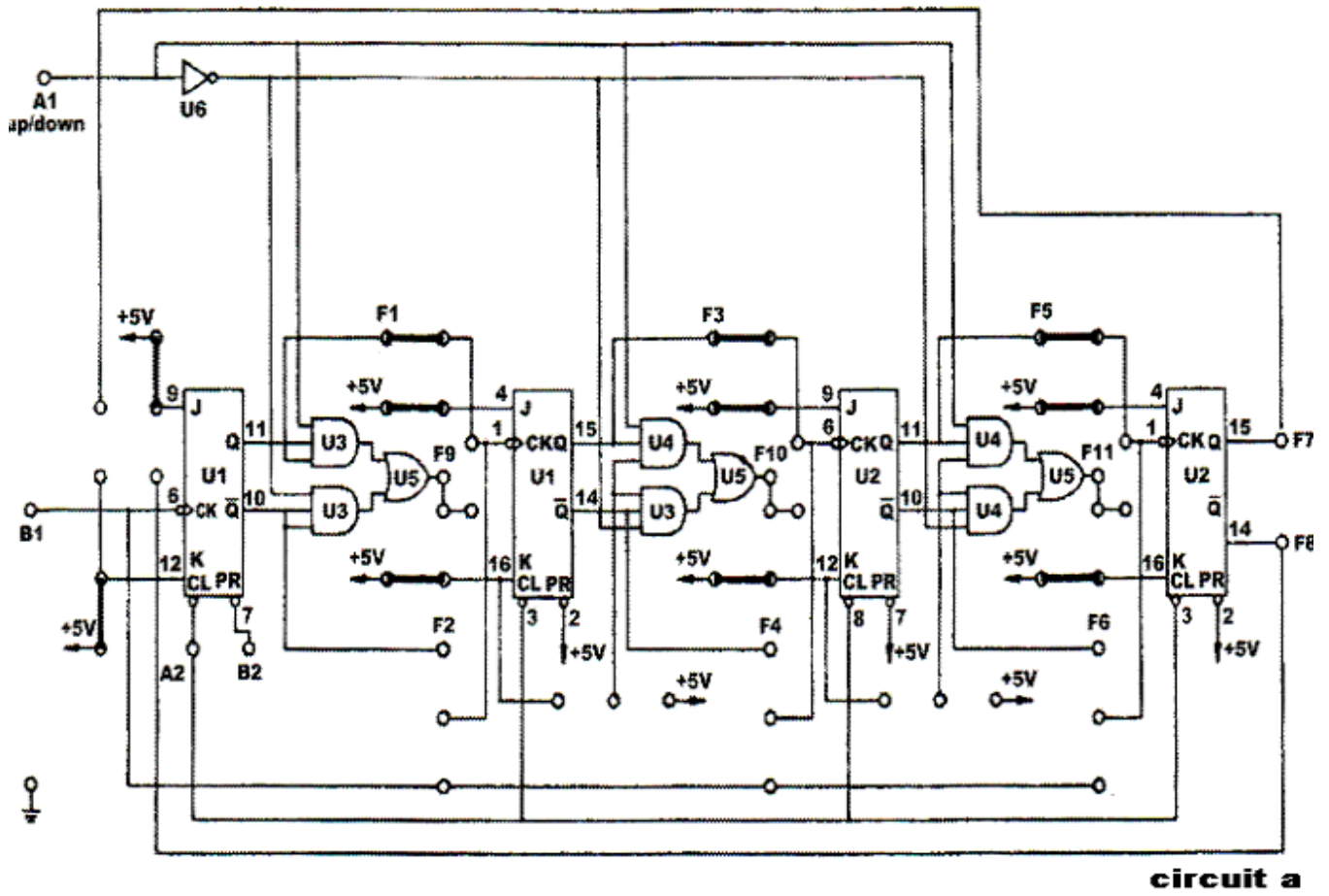
**Figure 46:** réalisation d'un compteur à base d bascule JK (diviseur par quatre)

11. Manipulation d'un compteur asynchrone:

En utilisant le module KL-33009 block a, réaliser le montage de la figure ci dessous.

Connecter A2 (Clear) à SW0, A1 (up/down) à +5V, B2 à +5V, B1 (CK) à un signal d'horloge de fréquence 1 Hz, F1, F3, F5, et F7 les sorties

Mettre initialement SW0 à "0" pour initialiser les sorties puis mettre SW0 à "1" pour commencer le comptage et déterminer le modulo de ce compteur.



circuit a

Figure 47: réalisation d'un compteur asynchrone à l'aide du module K-33009 bloc a

TP8: Etude et réalisation de registres

1. Objectifs :

L'étude du fonctionnement des registres à décalage

2. Les registres

Les registres sont ne sont pas des systèmes combinatoires, ce sont des composants permettant d'emmagasiner de l'information.

Pour une même combinaison d'entrées, ce type de composant peut produire des sorties différentes, selon l'information emmagasinée. Il existe deux types de registres: les bascules (*flipflop*) et les bistables (*latches*) qui diffèrent par leur signal d'activation. Les bistables ont un signal d'activation qui est basé sur un niveau logique tandis que le signal d'activation des bascules est sensible à une transition de niveau. Lorsque le signal d'activation d'une bistable est actif, tout ce qui est à l'entrée est transféré directement à la sortie.

Lorsque le signal d'activation devient inactif, la sortie conserve sa valeur jusqu'à ce que le signal redevienne actif. Avec une bascule, le comportement est légèrement différent. L'information à l'entrée est copiée à la sortie sur une transition spécifique du signal d'activation, habituellement la transition du "0" vers le "1".

3. Manipulation :

Le module KL 33008 bloc c représente le schéma d'un registre à décalage

- Entrées sorties parallèle :

Connecte B (clear) à SW0, A=D à SW1, CK à SWA A sortie, F1, F2, F3, F4 les sorties

Mettre SW0 à 0V pour RAZ B, ensuite à +5V

Ensuite suivre la séquence d'entrée A

à A=1, envoyer un signal d'horloge de SWA

à A=0, envoyer un signal d'horloge de SWA

à A=0, envoyer un signal d'horloge de SWA

à A=1, envoyer un signal d'horloge à SWA

Observer les sorties après quatre impulsions d'horloge, conclusion

- preset Left/Right Shift register

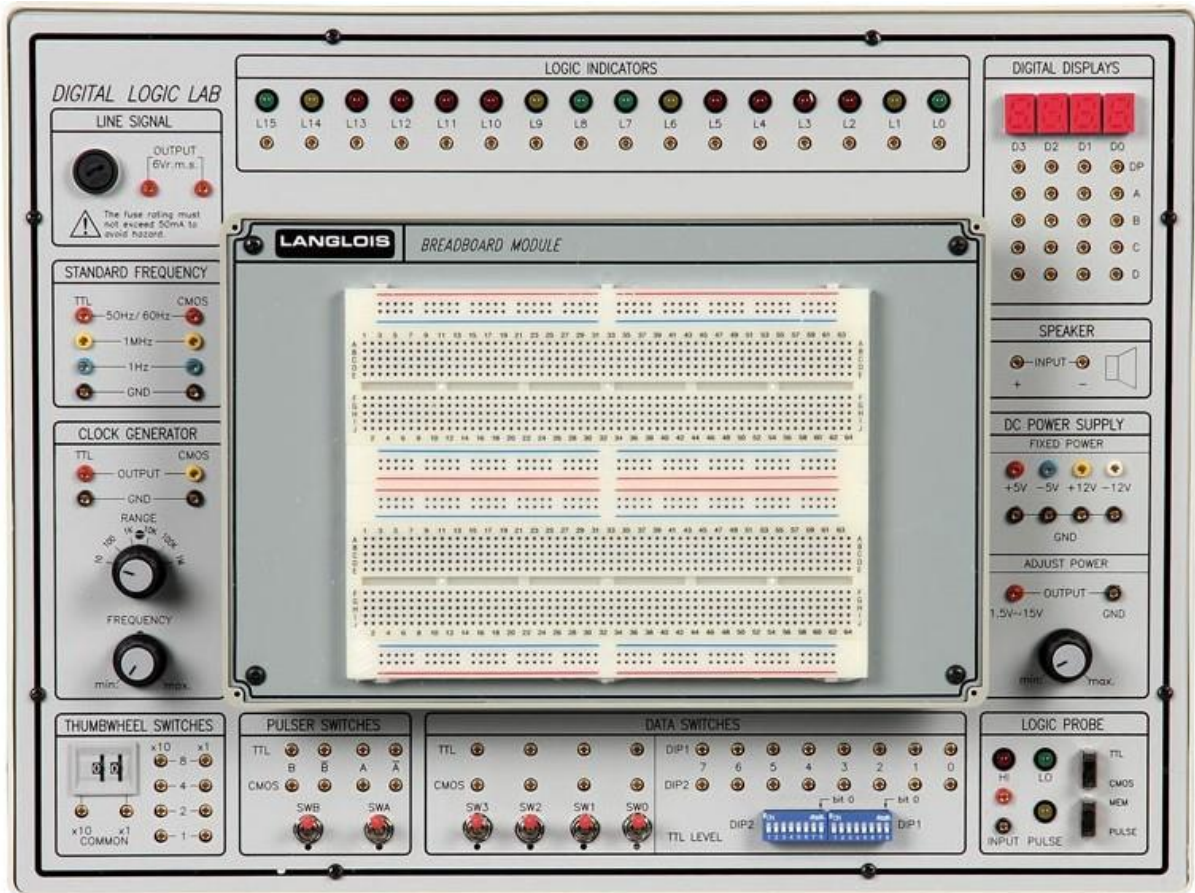
Le module KL 33008 bloc b représente un registre à décalage

Connecter A, B, C, D aux entrées (switches), les F1, F2, F3, F4 aux sorties (Leds)

Connecter D1 (Load) à SWA A sortie, C1 (CK) à SWB B sortie, B1 à DIP2.0 et A1 (MODE) à DIP2.1

Suivant la séquence d'entrée A1 (0 0 1 1 1 0 1 1), observer et noter les sorties

DIGITAL LOGIC LAB



Annexe : Travaux pratiques à préparer

TP1: Technologie des circuits intégrés TTL et CMOS

Préparation du TP:

Le but de ce premier TP est de tester les portes logiques suivantes : NOT, OR, AND, NOR, NAND et XOR intégrés dans les circuits logiques CI 7404, CI 7408 ; CI 7432, CI 7402, CI 7400, et CI 7486 correspondant, et de remplir les tables de vérités suivantes :

Table 1 :

A	S
0	
1	

S = ?, Son symbole graphique est ?

Table 2 :

Entrées		Sortie
A	B	S = A . B
0	0	
0	1	
1	0	
1	1	

S = ?, Son symbole graphique est ?

Table 3 :

Entrées		Sortie
A	B	S=A+B
0	0	
0	1	
1	0	
1	1	

S = ?, Son symbole graphique est ?

Table 4 :

Entrées		Sorties	
A	B	$A + B$	$S = \overline{A + B}$
0	0		
0	1		
1	0		
1	1		

$S = ?$, Son symbole graphique est ?

Table 5 :

Entrées		Sorties	
A	B	$A.B$	$S = \overline{A.B}$
0	0		
0	1		
1	0		
1	1		

$S = ?$, Son symbole graphique est ?

Table 6 :

Entrées		Sortie
A	B	$S = A \oplus B$
0	0	
0	1	
1	0	
1	1	

$S = ?$, Son symbole graphique est ?

Table 7 :

Entrées		Sortie
A	B	$S = \overline{A \oplus B}$
0	0	
0	1	
1	0	
1	1	

TP2: Simplification des équations logiques par la pratique

Préparation du TP:

Le but de ce deuxième TP est de réaliser une porte logique en utilisant d'autres portes logiques :

- d'une porte NOT n'utilisant qu'une porte NOR.
- d'une porte NOT n'utilisant qu'une porte NAND.
- d'une porte AND n'utilisant que des portes NOR.
- d'une porte AND n'utilisant que des portes NAND.
- d'une porte OR n'utilisant que des portes NAND.
- d'une porte OR n'utilisant que des portes NOR.
- d'une porte XOR n'utilisant que des portes NAND.

TP3: Etude et réalisation de fonctions logiques combinatoires usuelles**Préparation du TP:**

Le but de ce troisième TP est de vérifier :

- la table de vérité d'un multiplexeur MUX 2 to 1
- la table de vérité d'un multiplexeur MUX 8 to 1
- la table de vérité d'un démultiplexeur DEMUX 1 to 2
- la table de vérité d'un DEMUX 1 to 8
- la table de vérité d'un décodeur BCD vers les sept segments

TP4: Etude et réalisation d'un circuit combinatoire arithmétique

Préparation du TP:

Le but de ce quatrième TP est de vérifier :

- la table de vérité d'un demi-additionneur (H.A), donner $S = ?$, $C = ?$

Table 10 : Table de vérité d'un demi-additionneur

Entrées		Sorties	
B	A	S	C
0	0		
0	1		
1	0		
1	1		

- la table de vérité d'un l'additionneur complet (F.A), donner $S = ?$, $C = ?$

Table 11 : Table de vérité d'un additionneur complet

Entrées			Sorties	
C_{i-1}	B_i	A_i	S_i	C_i
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

- la table de vérité d'un demi-soustracteur (H.S), donner $D = ?$, $E = ?$

Table 12 : Table de vérité demi-soustracteur

Entrées		Sorties	
B	A	D	E
0	0		
0	1		
1	0		
1	1		

- la table de vérité d'un soustracteur complet (F.S), donner $D = ?$, $E = ?$

Table 13 : Table de vérité d'un soustracteur complet

Entrées			Sorties	
E_{i-1}	B_i	A_i	D_i	E_i
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

TP5: Etude et réalisation d'un circuit combinatoire logique**Préparation du TP:**

Le but de ce cinquième TP est de vérifier:

- la table de vérité d'un complément à deux, de trouver $Y_1 = ?$, $Y_2 = ?$, $Y_3 = ?$

Table 14 : Table de vérité d'un complément à 2

Entrées			Sorties		
A ₃	A ₂	A ₁	Y ₃	Y ₂	Y ₁
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

- la table de vérité d'un convertisseur Gray – Binaire, trouver B1= ?, B2= ?, B3= ?

Table 15 : Table de vérité code GRAY

-Entrées code GRAY			Sorties code binaire		
G ₃	G ₂	G ₁	B ₃	B ₂	B ₁
0	0	0			
0	0	1			
0	1	1			
0	1	0			
1	1	0			
1	1	1			
1	0	1			
1	0	0			

TP6: Etude et réalisation de circuits combinatoire logique**Préparation du TP:**

Le but de ce cinquième TP est de vérifier:

- la table de vérité d'un comparateur de deux nombres à 4 bits, de trouver $S = ?$, $I = ?$, $E = ?$

Table 16 : Table de vérité d'un comparateur de deux nombres à 4 bits

a_4 b_4 a_3 b_3 a_2 b_2 a_1 b_1	S ($A > B$)	I ($A < B$)	E ($A = B$)
$a_4 > b_4$			
$a_4 < b_4$			
$a_4 = b_4$ $a_3 > b_3$			
$a_3 < b_3$			
$a_3 = b_3$ $a_2 > b_2$			
$a_2 < b_2$			
$a_2 = b_2$ $a_1 > b_1$			
$a_1 < b_1$			
$a_1 = b_1$			

TP7: Etude et réalisation de circuits compteurs**Préparation du TP:**

Le but de ce cinquième TP est de vérifier:

- la table de vérité de la bascule SR, Q= ?

Table 17 : Table de vérité d'une bascule SR

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

- la table de vérité de la bascule D

- la table de vérité de la bascule T

- la table de vérité de la bascule JK

- un compteur Modulo N à base de bascules JK.

TP8: Etude et réalisation de registres**Préparation du TP:**

Le but de ce cinquième TP est de vérifier un registre à décalage

Bibliographie

1. 'Cours d'électronique numérique' Camille Diou, Laboratoire Interfaces Capteurs et Microélectronique, Université Paul Verlaine–Metz. Version du 24 février 2009.
2. 'Introduction aux circuits logiques' Jean Letocha / Collet Leon, Edition, 2 ; Publisher, McGraw-Hill, 1985.
3. 'Logique combinatoire et séquentielle' Djamel Gozim, Kamel Guesmi, Licence. Al gérie. 2019, hal-02927680.
4. 'logique combinatoire et technologie' M. Gindre, Electronique numérique, Mc Graw Hill, 1987.
5. 'Systèmes logiques I' Jean-François Harvey, Mohamad Sawan, Quatrième édition école polytechnique de Montréal septembre 1999