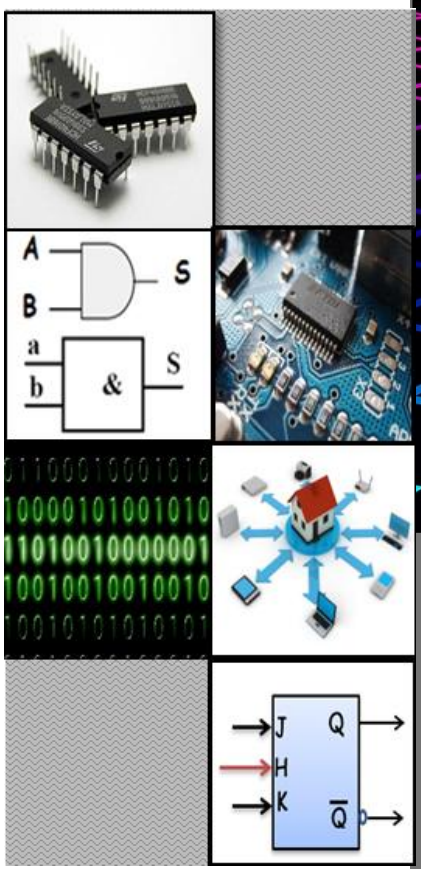
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE LA FORMATION ET DE L'ENSEIGNEMENT PROFESSIONNELS
INSTITUT DE FORMATION ET D'ENSEIGNEMENT PROFESSIONNELS
« SANHADRI ABDELHAFID - SIDI BEL-ABBES.

HNIQUE ET PROFESSIONNEL DU STAGIAIRE

Logique combinatoire et logique séquentielle

NIVEAU 5 DOMOTIQUE





Conçu par : Mr HALAILI Mohamed

Décembre / 2020

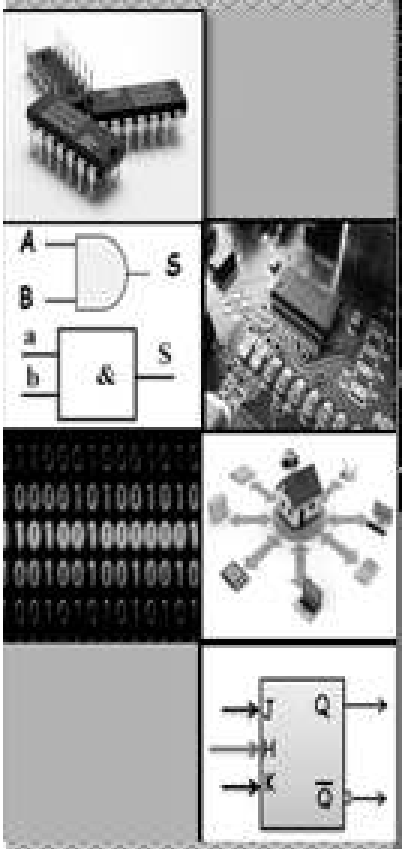
WWW.DZPEN.COM | قلم الجزائر

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE LA FORMATION ET DE L'ENSEIGNEMENT PROFESSIONNELS
INSTITUT DE FORMATION ET D'ENSEIGNEMENT PROFESSIONNELS
« SANHADRI ABDELHAFID - SIDI BEL-ABBES.

MANUEL TECHNIQUE ET PROFESSIONNEL DU STAGIAIRE

Logique combinatoire et logique séquentielle

NIVEAU 5 DOMOTIQUE





Conçu par : Mr HALAILI Mohamed

Décembre / 2020

WWW.DZPEN.COM | قلم الجزائر

Equipe de production

Conception et rédaction

M^{R} HALAILI Mohamed

Professeur spécialisé de la formation et de l'enseignement professionnel chargéd'ingénierie pédagogique (P.S.F.E.P.CIP) en électronique, institut de formation et d'enseignement professionnels« SANHADRI Abdelhafid - sidi bel-abbés.

Remerciements

Nous remercions toutes les personnes qui ont contribué, de près ou de loin à la conception et à la réalisation de ce manuel technico pédagogique pour stagiaire (M.T.P.S) jusqu'à son édition finale.

Nous remercions également les organismes qui nous ont gracieusement autorisés à utiliser à des fins pédagogiques les illustrations et les textes tirés de leurs documentations.

Les droits de production d'adaptation ou de traduction de ce manuel professionnel sont réservés à l'IFEP.

WWW.DZPEN.COM | قلم الجزائر

TABLE DES MATIERES

Préambule..... 7

CHAPITRE 1 LES SYSTEMES DE NUMERATION ET DE CODAGE 13

1.1 Systèmes de numération..... 14

1.1.1 Base d'un système de numération 14

1.1.2 Rang d'un chiffre dans un nombre quelconque..... 14

1.1.3 Forme polynomiale 15

1.1.4 Présentation des trois principaux systèmes de numération..... 15

1.1.5 Exercice 01..... 16

1.2 Changement de bases(conversions)..... 16

1.2.1 Conversion d'un nombre N de base b quelconque en décimal..... 16

1.2.2 Conversion d'un nombre entier décimal en un nombre de base b 16

1.2.3 Conversion binaire-octal et octal-binaire..... 17

1.2.4 Conversion binaire-hexadécimal et hexadécimal –binaire..... 18

1.2.5 Conversion d'un nombre fractionnaire..... 18

1.2.6 Exercice 02..... 20

1.3 Opérations arithmétiques en binaire..... 20

1.3.1 Addition en binaire..... 20

1.3.2 Soustraction en binaire 20

1.3.3 La multiplication 21

1.3.4 Complémentation..... 21

1.3.5 Soustraction par complément à 2..... 21

1.3.6 Exercice 03..... 22

1.4 Systèmes de codage 23

1.4.1 Code Gray ou binaire réfléchi 23

1.4.2 Code BCD..... 24

1.4.3 Code majoré de trois..... 25

1.4.4 Code Aiken..... 26

1.4.5 Exercice 04..... 27

1.5 Résumé 27

1.6 Exercice de synthèse1..... 28

CHAPITRE2 ALGEBRE DE BOOLE 29

2.1 Variables et fonctions logiques..... 30

2.1.1 Variable logique 30

2.1.2 Fonction logique 31

2.1.3 Exercice 05..... 31

2.2 Opérateurs Logiques (Portes logiques)..... 32

2.2.1 Porte NON (NOT) 32

2.2.2 Porte ET (AND) 32

2.2.3 Porte OU (OR) 32

2.2.4 Porte NON-ET (NAND) 33

2.2.5 Porte NON-OU (NOR) 33

2.2.6 Porte OU EXCLUSIF (XOR) 33

2.2.7 Porte NON-OU EXCLUSIF (XNOR) 33

WWW.DZPEN.COM | قلم الجزائر

2.2.8 Synthèse des portes logiques..... 34
2.2.9 Exercice 06..... 35
2.3 Table de vérité..... 35
2.3.1 Extraction de la fonction logique à partir de la T.D.V..... 35
2.3.2 Déduction de la TDV à partir de la fonction logique 37
2.3.3 Exercice 7..... 38
2.4 Schéma d'un circuit logique (Logigramme)..... 38
2.4.1 Logigramme à l'aide des portes NAND uniquement 39
2.4.2 Exercice 8..... 40
2.5 Circuits intègres des portes logique..... 40
2.5.1 Principe des circuits intégrés..... 40
2.5.2 Caractéristiques des circuits intègres..... 42
2.5.3 Exercice 9..... 43
2.6 Simplification des fonctions logiques..... 44
2.6.1 Lois fondamentales de l'algèbre de Boole..... 44
2.6.2 Simplification en utilisant la méthode algébrique..... 45
2.6.3 Tableau de KARNAUGH..... 45
2.6.4 Simplification en utilisant le tableau de KARNAUGH..... 48
2.6.5 Fonctions incomplètement définies 51
2.6.6 Exercice 10..... 52
2.7 Résumé..... 52
2.8 Exercice de synthèse 2..... 53
CHAPITRE 3 LESCIRCUITS COMBINATOIRES 54
3.1 Circuits de transcodage..... 55
3.1.1 Codeur..... 55
3.1.2 Décodeur..... 57
3.1.3 Convertisseur..... 58
3.1.4 Exercice 11..... 59
3.2 Circuits de multiplexage..... 60
3.2.1 Multiplexeur..... 60
3.2.2 Démultiplexeur..... 62
3.2.3 Exercice 12..... 63
3.3 Circuits de calcul..... 64
3.3.1 Demi-additionneur..... 64
3.3.2 Additionneur complet..... 64
3.3.3 Comparateur 65
3.3.4 Exercice 13..... 66
3.3.5 Exercice 14..... 66
3.4 Résumé..... 67
3.5 Exercice de synthèse 3 68
CHAPITRE 4 LES CIRCUITS SEQUENTIELS 69
4.1 Circuits combinatoires et circuits séquentiels..... 70
4.1.1 Circuits combinatoires..... 70
4.1.2 Circuits séquentiels..... 70
4.1.3 Fonctionnement des circuits séquentiels 71
4.1.4 Exercice 14..... 71

WWW.DZPEN.COM | قلم الجزائر

4.2 Les bascules..... 72
4.2.1 Bascule RS Asynchrone..... 72
4.2.2 Bascule RS Synchrone..... 74
4.2.3 Bascule D..... 75
4.2.4 Bascule JK Asynchrone..... 77
4.2.5 Bascule T..... 78
4.2.6 Déclenchement d'une bascule..... 79
4.2.7 Exercice 16..... 81
4.3 Les compteurs..... 82
4.3.1 Compteurs Asynchrones..... 82
4.3.2 Compteurs Synchrones 85
4.3.3 Compteur Synchrone réversible (Compteur/Décompteur) 88
4.3.4 Exercice 17..... 90
4.4 Les registres..... 91
4.4.1 Différents types de registres 91
4.4.2 Fonctionnement d'un registre à décalage 93
4.4.3 Registre à décalage à l'aide de bascule JK 94
4.4.4 Exercice 18..... 95
4.5 Résumé..... 97
4.6 Exercice de synthèse 98
CHAPITRE 5 LES PRINCIPAUX ELEMENTS D'UN AUTOMATE 101
5.1 Technologie des automates programmables industriels 102
5.1.1 Type compact (centralisé) 102
5.1.2 Type modulaire 103
5.1.3 Avantages et inconvénients des automates programmables industriels..... 103
5.1.4 Domaines d'emploi des automates programmables industriels..... 104
5.1.5 Critères de choix des automates programmables industriels..... 105
5.1.6 Exercice 19..... 105
5.2 Architecture d'un automate programmable industriel..... 106
5.2.1 Aspect extérieur d'un automate programmable industriel 106
5.2.2 Structure interne d'un automate programmable industriel..... 108
5.2.3 Exercice 20..... 113
5.3 Langage de programmation..... 114
5.3.1 Liste d'instructions 114
5.3.2 Langage littéral structuré 114
5.3.3 Langage à contacts 115
5.3.4 Blocs Fonctionnels 115
5.3.5 Langage GRAFCET 116
5.3.6 Exercice 21..... 117
5.4 Résumé..... 118
5.5 Exercice de synthèse 5..... 119
Résumé général : 122
Activité de synthèse : 124
Bibliographie : 127
Annexe : Corrigé des exercices: .....I.....XXXIII

WWW.DZPEN.COM | قلم الجزائر

Préambule

La domotique apporte des solutions pratiques pour améliorer le quotidien des usagers avec une interaction sur le plan du confort (gestion optimisée du chauffage et/ou de l'éclairage), de la sécurité (alarme et surveillance) et sur le pilotage des appareils électroménagers connectés (commande des appareils à distance ou déclenchement automatique des appareils), elle est définie aussi comme l'ensemble des techniques et technologies électroniques, électronique numérique, informatiques et des télécommunications qui permettent de gérer localement ou à distance l'automatisation, la programmation et le contrôle de la maison (confort, sécurité, énergie).

Ce manuel est rédigé à l'intention des stagiaires Technicien supérieur en « Domotique » afin de lui apporter une aide précieuse dans l'analyse et la conception des circuits combinatoires et séquentiels (Circuits numériques). Il est constitué de cours et d'exercices d'applications qui expliquent d'une manière simple et facile le principe de la logique combinatoire et de la logique séquentiel (Electronique numerique).

- Le premier chapitre traite en détail des différents systèmes de numération, ainsi que les méthodes de conversion entre ces systèmes. Nous étudierons également les opérations arithmétiques sur les nombres binaires et nous terminerons ce chapitre par l'étude de plusieurs codes numériques.

- Le deuxième chapitreest consacré à l'algèbre de Boole qui va nous permettre de traduire les signaux électriques en expressions mathématiques en utilisant des variables logiques (Vrai /faux). On va aborder en détail les variables et fonctions logiques, les opérateurs logiques et leurs simplifications afin de réaliser le circuit logique le plus simplifier

- Dans Le troisième chapitre, nous allons aborder les circuits combinatoires. Ces circuits font comme tous les autres circuits : ils prennent des données sur leurs entrées, et fournissent un résultat en sortie. Pour donner quelques exemples, on peut citer les circuits de transcodage, de multiplexage et les Circuits de calcul
  Dans ce qui va suivre, nous allons étudier et concevoir ce genre de circuits. Il existe des méthodes et procédures qui vont nous permettre de créer n'importe quel circuit combinatoire.Ce type de circuits ne permet en fait de traiter qu'une classe restreinte de problèmes ne nécessitant pas de besoin de mémorisation

- Le quatrième chapitre est consacré aux circuits séquentielsqui ont quant à eux la capacité de mémoriser des informations et par conséquent de traiter des séquences de données. Dans ce chapitre on va faire l'étude des circuits séquentiels élémentaires, comme par exemple les bascules, les compteurs et les registres.

WWW.DZPEN.COM | قلم الجزائر

- Dans le cinquième chapitreon va aborder ces principaux éléments desautomates programmables vue l'importance qu'occupe ces derniers dans les technologies d'automatisation des installations industrielles,

WWW.DZPEN.COM | قلم الجزائر

PRESENTATION DE LA PROFESSION

A. Dénomination de la profession : Technicien Supérieur en Domotique

B. Définition de la profession :

Le titulaire du BTS en Technicien Supérieur en Domotique son métier consiste à concevoir, installer, programmer et mettre en service des solutions techniques dans l'habitat, et il assure l'entretien et le dépannage des appareils et composants domotique.

C. Tâches essentielles :

Le titulaire du BTS en Technicien Supérieur en Domotique est chargé de ;

- Installer, et configurer les circuits d'éclairage.
- Installer, et gérer le confort des systèmes de régulation de chauffage/climatisation.
- Installer, et gérer les systèmes de sécurité, incendie, intrusion, inondation, fuite de gaz, de fumée.
- Installer, et configurer un système Vidéosurveillance.
- Installer et configurer un système de porte et volets roulants automatique.
- Intégrer les différents systèmes domotique.
- Assurer la maintenance préventive et curative du système domotique.
- Evaluer les coûts de travaux de maintenance

WWW.DZPEN.COM | قلم الجزائر

PRESENTATION DU MODULE

Intitulé du Module : **logique combinatoire et logique séquentielle.**
Code du module : MC 11
Durée : 68 heures.

Objectif modulaire

Comportement attendu :
Le stagiaire doit être capable de :

- Appliquer les notions d'algèbre de Boole,
- Appliquer les notions de logique combinatoire et de logique séquentielle,
- Comprendre le principe de fonctionnement des circuits séquentiels,
- Interpréter et analyser les différentes représentations graphiques d'une séquence : algorithme, logigramme.
- Distinguer les principaux éléments d'un automate programmable

Conditions d'évaluation :

A partir de :

- Documentation appropriée

A l'aide de :

- Matériel didactique approprié.
- Data show.

Critères généraux de performance :

- Interprétation exacte des chronogrammes et logigrammes.

WWW.DZPEN.COM | قلم الجزائر

Objectifs intermédiaires.	Critères particuliers de performance.	Eléments de contenu.
Appliquer les notions de base des systèmes logiques.	✓ Application correcte des concepts de base des systèmes logiques.	➢ Système de numération et de codage : ✓ Conversion des différents systèmes de numération. ✓ Opérations arithmétiques. ➢ Algèbre de Boole : ✓ Définition d'une variable binaire. ✓ Définition d'une fonction booléenne. ✓ Opérations logiques : • Inversion (porte NON). • Somme logique (porte OU) • Produit logique (porte ET) • Opération NON OU (porte NI) • Opération NON ET (porte NAND) • Opération Ou exclusif (porte Ou exclusif). ➢ Règles générales de l'algèbre de Boole ✓ Distributivité de la somme et du produit logique. ✓ Théorème de Morgan
Analyser un circuit combinatoire.	✓ Utilisation appropriée des fonctions logiques. ✓ Lecture et interprétation juste d'un logigramme d'un circuit combinatoire. ✓ Utilisation correcte des opérateurs logiques de base.	➢ Circuit combinatoire : ✓ Définition et caractéristiques. ✓ Rôle et mode de fonctionnement. ✓ Type : ✓ Les circuits de transcodage : o Codeur – décodeur. o Transcodeur. o Afficheur à sept segments. ✓ Les circuits de comparaison : o Comparateur. ✓ Les circuits arithmétiques : o Additionneur. o Soustracteur. ✓ Les circuits de multiplexage : o Multiplexeur. o Démultiplexeur.

WWW.DZPEN.COM | قلم الجزائر

Objectifs intermédiaires.	Critères particuliers de performance.	Eléments de contenu.
Analyser les circuits séquentiels de base.	Analyse correcte des circuits séquentiels de base.	➢ Circuits séquentiels : ✓ Définition et caractéristiques. ✓ Rôle et mode de fonctionnement. ✓ Exemples : o Registres. o Convertisseur série-parallèle. ➢ Distinction entre les systèmes séquentiels synchrone et asynchrone : ➢ Système séquentiel synchrone ➢ Système séquentiel asynchrone ➢ Exemple de circuits séquentiels synchrone et asynchrone ➢ Réalisation des différents types de bascules : ✓ Les bascules : RS, JK, D, T ✓ Les bascules en circuits intégrés. ➢ Réalisation d'un compteur binaire et à décade : ✓ Compteurs binaires ✓ Définition des concepts : o Mode de départ, de marche et d'arrêt. o Condition de départ dans un compteur à décade (manuel, automatique, remise à zéro des compteurs). ✓ Réalisation d'un registre binaire et à décalage : ✓ Registres
Distinguer les principaux éléments d'un automate	✓ Identification exacte des principaux éléments d'un automate programmable.	✓ Principaux modules o Module de traitement o Modules des entrées et des sorties o Module d'alimentation o Périphériques

WWW.DZPEN.COM | قلم الجزائر

CHAPITRE 1

LES SYSTEMES DE NUMERATION ET DE CODAGE

Les ordinateurs qui fonctionnent en utilisant les propriétés de l'électricité, ne connaissent que les 1 ou les 0. C'est la raison pour laquelle, dès que l'on aborde le monde des systèmes numériques, il est nécessaire pour leur bonne compréhension de connaître certaines bases de numération (binaire, décimale et hexadécimale) et d'être capable d'effectuer des conversions entre-elles. De plus, ces systèmes pour pouvoir communiquer entre-autre (vers un autre système ou vers l'homme) utilisent certains codes (Code Gray, Code BCD, code Aiken…etc.) quand doit connaître.

La **numération** décrit la façon dont les nombres sont représentés. Un système de numération est composé de :

- D'un **alphabet** : les signes ou symboles disponibles pour la représentation des nombres
- Des **règles d'écriture** : elles définissent comment un nombre est construit à partir des symboles de l'alphabet.

1.1 Systèmes de numération

1.1.1 Base d'un système de numération

La base d'un système de numération est le nombre de chiffres qu'utilise ce système.

- Dans un système **décimal**, on utilise un maximum de dix symboles pour représenter un nombre quelconque N, soit : 0, 1, 2, 3, 4, 5, 6, 7, 8,9.
- Dans un système **binaire,** on utilise un maximum de **deux symboles** pour représenter un nombre quelconque N, soit : 0,1.
- Dans un système **octal,** on utilise un maximum de **huit symboles** pour représenter un nombre quelconque N, soit : 0, 1, 2, 3, 4, 5, 6,7.
- Dans un système **hexadécimal,** on utilise un maximum de **seize symboles** pour représenter un nombre quelconque N,
  soit :0,1,2,3,4,5,6,7,8,9,A ,B,C,D,E,F.

❖ **Exemple:**

Donner les symboles que vous utiliseriez dans les systèmes de numération des bases : **5,7**.

- Le système dans la base =5, utilise 5 symboles (0, 1, 2, 3,4).
- Le système dans la base =7, utilise 7 symboles (0, 1, 2, 3, 4, 5,6).

1.1.2 Rang d'un chiffre dans un nombre quelconque

Dans tous les systèmes de numération, le chiffre de poids le plus fort **(Most Significant Bit en binaire : MSB)** d'un nombre est dans la colonne extrême gauche, le chiffre de poids le plus faible **(Least Significant Bit en binaire : LSB)** est dans la colonne extrême droite. Le rang d'un chiffre dans un nombre est égal au numéro de sa colonne, la première colonne (**numèro0)** étant celle du poids le plus faible.

❖ **Exemple:** Soit à définir le rang des chiffres du nombre $(256987)_{10}$

Le poids le plus fort (MSB) = 2
Le poids le plus faible (LSB)=7

WWW.DZPEN.COM | قلم الجزائر

- Rang du chiffre 7 :0
- Rang du chiffre 8 : 1
- Rang du chiffre 9 : 2
- Rang du chiffre 6 : 3
- Rang du chiffre 5 : 4
- Rang du chiffre 2 : 5

1.1.3 Forme polynomiale

Tout nombre **N** peut être décomposé en fonction de puissances entières de la base**n**

$$N = \sum_{i=0}^{n} a_i \times b^i \text{ où } a_i \in \{0,1.....b-1\}$$

i rang (décimal) du chiffre a_i

n exposant du chiffre de poids fort

❖ **Exemples :**

- Dans le système décimal : $a_i \in \{0,1,2,3,4,5,6,7,8,9\}$

  Soit : $(54219)_{10} = 9x10^0+1x10^1+2x10^2+4x10^3+5x10^4$
- Dans le système à base 4 : $a_i \in \{0,1,2,3\}$

  Soit : $(30212)_4 = 2x4^0+1x4^1+2x4^2+0x4^3+3x4^4$
- Dans le système binaire (à base 2) : $a_i \in \{0,1\}$

  Soit : $(1011)_2 = 1x2^0+1x2^1+0x2^2+1x2^3$

1.1.4 Présentation des trois principaux systèmes de numération

Tableau 01 Les trois principaux systèmes de numération

Décimal	Binaire naturel	Octal	Hexadécimale
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10
17	10001	21	11

WWW.DZPEN.COM | قلم الجزائر

1.1.5 Exercice 1

Quelles sont parmi ces suites de nombre celles qui peuvent être la représentation d'un nombre en base 16, 8 ou 2 ?

Tableau 02 : Représentation des nombres dans différents systèmes de

	Binaire (Base 2)	Octal (Base 8)	Hexadécimal (Base 16)
1001010210	Faux	Vrai	Vrai
1001011			
1200111201			
1431901			
2A21F			
9GF28			
1789012			

1.2 Changement de bases (conversions)

1.2.1 Conversion d'un nombre N de base b quelconque en décimal

La valeur en décimal d'un nombre n de base b quelconque s'obtient en effectuant les opérations de l'expression de sa forme polynomiale.

- **Exemples :**Soit à déterminer la valeur décimale des nombres de l'exemple précédent :
  - $(30212)_4 = 2x4^0 + 1x4^1 + 2x4^2 + 0x4^3 + 3x4^4 = (806)_{10}$
  - $(1011)_2 = 1x2^0 + 1x2^1 + 0x2^2 + 1x2^3 = (11)_{10}$

1.2.2 Conversion d'un nombre entier décimal en un nombre de base b quelconque

- **Règle** :L'opération consiste à procéder à des divisions successives du nombre à convertir par la base du nouveau système tout en conservant les restes de ces divisions. On écrit ensuite tous les restes à partir de la fin de gauche à droite, en les convertissant en lettres s'il y a lieu**.**

WWW.DZPEN.COM | قلم الجزائر

❖ **Exemple(1) :** Convertir le nombre N= $(231)_{10}$ en binaire.

230 :2= 115 reste 0
115 :2= 57reste 1
57 :2= 28 reste 1
28 :2= 14 reste 0N= $(11100110)_2$
14 :2= 7 reste 0
7 :2= 3 reste 1
3 :2=1 reste 1
1 :2=0 reste 1

❖ **Exemple(2) :** Convertir le nombre N= $(189520)_{10}$ en hexadécimal.

189520 :16=11845 reste 0
11845 :16= 740 reste 5
740 :16= 46 reste 4N= $(2E450)_{16}$
46 :16= 2 reste 14=E
2 :16= 0 reste 2

1.2.3 Conversion binaire-octal et octal-binaire

❖ **Règle :**

Conversion binaire-octal : A partir de la virgule, grouper les bits par groupes de trois en allant vers la gauche pour la partie entière et vers la droite pour la partie fractionnaire. Convertir ensuite chaque bloc séparément en octal selon le code binaire naturel.

Conversionoctal binaire : Chaque chiffre octal est converti on binaire sur 3 bits

❖ **Exemple(1) :**Soit à convertir en octal le nombre binaire N= $(001110011101)_2$

N= $(001110011101)_2$

N= $(16\ 35\ )_8$= $(1635)_8$

On lit : un six trois cinq,six quatre un

❖ **Exemple 2 :**Soità convertir en binaire le nombre en octal N= $(7510)_8$

N=$(7510)_8$

N= $(111101001000)_2$= $(111101001000)_2$

WWW.DZPEN.COM | قلم الجزائر

1.2.4 Conversion binaire-hexadécimal et hexadécimal-binaire

❖ **Règle :**

Conversion binaire-hexadécimal : A partir de la virgule, grouper les bits par groupes de quatre en allant vers la gauche pour la partie entière et vers la droite pour la partie fractionnaire. Convertir ensuite chaque bloc séparément en hexadécimal

Conversionhexadécimal binaire : Chaque chiffre hexadécimal est converti on binaire sur 4 bits

❖ **Exemple(1): Soit à convertir en hexadécimal le nombre binaire**

$N=(1110011101)_2$

$N= (001110011101)_2$

$N= (39D)_{16} = (39D)_{16}$

❖ **Exemple(2) : Soit à convertir en binaire le nombre en hexadécimal**

$N=(7A1F,B46)_{16}$

$N= (7A1F, B46)_{16}$

$N= (0111101000011111,101101000110)_2$

$N= (0111101000011111,101101000110)_2$

1.2.5 Conversion d'un nombre fractionnaire

A. Forme polynomiale d'un nombre fractionnaire :

Tout nombre fractionnaire **N** peut être décomposé en fonction de puissances entières de la base

$$N = \sum_{i=0}^{n} a_i \times b^{-i} \text{ où } a_i \in \{0,1.....b-1\}$$

i rang (décimal) du chiffre a_i

n exposant du chiffre décimal le plus à droite

❖ **Exemples :**Soit à déterminer la forme polynomiale des nombres :

➢ $(12,30212)_4 = 1x4^{+1} +2x4^{0} +3x4^{-1} +0x4^{-2}+2x4^{-3}+1x4^{-4}+2x4^{-5}$

➢ $(10,1011)_2 = 1x2^{+1} +0x2^{0} +1x2^{-1} +0x2^{-2} +1x2^{-3}+1x2^{-4}$

WWW.DZPEN.COM | قلم الجزائر

B. Conversion d'un nombre fractionnaire de base b en décimal :

❖ **Exemples :**Soit à déterminer l'équivalent décimal des nombres fractionnaires suivants

➢ $(10,1011)_2 = 1x2^{+1} + 0x2^{0} + 1x2^{-1} + 0x2^{-2} + 1x2^{-3} + 1x2^{-4}$
$= 1x2+0x1+1x0,5+1x0,125+1x0,0625 = (2,6875)_{10}$

➢ $(12,16)_8 = 1x8^{+1} + 2x8^{0} + 1x8^{-1} + 6x8^{-2} = 8+2+(1/8)+(6/64)$
$= 10+0.125+0,09375 = (10,21875)_{10}$

C. Conversion d'un nombre décimal fractionnaire en un nombre de base b:

❖ **Règle :**L'opération consiste à procéder à des multiplications successives du nombre à convertir puis des différents résultats par la base du nouveau système tout en conservant les nombres entiers de ces multiplications.

❖ **Exemple(1) :Soit à convertir en binaire le nombre $(8,35)_{10}$**

➢ **Partie entière : $(8)_{10} = (?)_2$**

8 :2= 4 reste 0
4 :2= 2 reste 0
2 :2= 1 reste 0
1 :2= 0 reste 1

$(8)_{10} = (1000)_2$

➢ **Partie fractionnaire : $(0.35)_{10} = (?)_2$**

0,35 x2 =0, 70 0
0,70x2 = 1, 40 1
0,40x2 = 0, 800 $(0.35)_{10} = (0,01011)_2$
0,80x2 = 1, 601
0,60x2 = 1, 20 1

Donc $(8,35)_{10} = (1000,01011)_2$

➢ **Remarque :** En augmentant le nombre de multiplications, on améliore l'approximation.

WWW.DZPEN.COM | قلم الجزائر

1.2.6 Exercice 2

Complétez le tableau suivant, puis vérifier les résultats en faisant l'opération inverse

Tableau 03 : Conversion des nombres dans différents systèmes de numération

Base	Nombre	Equivalent	Dans la base
10	45		2
10	57		8
8	75		16
2	1101,101		10
10	42,0625		8
16	ABC, A		8
4	1320		16

1.3 Opérations arithmétiques en binaire

1.3.1 Addition en binaire

❖ **Règles de l'addition :**

0 + 0 = 0
1 + 1 = 0 ➡ report 1
1 + 0 = 1
1 + 1 + 1 = 1 ➡ report 1

❖ **Exemple(1) :**Effectuer en binaire l'opération suivante $(00010011)_2$ + $(00111011)_2$

```
   1  11 1
      0  0  0  1  0  0  1  1
+     0  0  1  1  1  0  1  1
-------------------------------
=     0 1  0  0 1  1  1  0
```

1.3.2 Soustraction en binaire

❖ **Règles de la soustraction :**

0 - 0 = 0
1 - 1 = 0
1 - 0 = 1
0 - 1= 1 ➡ retenue 1

❖ **Exemple(1) :**Effectuer en binaire l'opération suivante $(01011011)_2$ + $(00111001)_2$

```
      0  1  10  1  10 1  1
-     0 10  1  1  1  0  0  1
-------------------------------
=     0  010  0 01 0
```


WWW.DZPEN.COM | قلم الجزائر

1.3.3 La multiplication

Regardons d'abord comment on fait une multiplication en base 2 sur un aspect mathématique pur sans limitation sur le nombre de bits. Remarquons au passage qu'ici on considère que tous les nombres sont positifs car le concept de représentation des nombres négatifs par le complément à 2 n'est valable que si on travaille avec un nombre de bit fixe.

❖ **Exemple(1) :**Effectuer en binaire l'opération suivante $(1011)_2 + (1101)_2$

```
  1011   (11)
 *1101   (13)
  1011
 1011•
1011
10001111  (143)
```

1.3.4 Complémentation

A. Complément à 1 d'un nombre binaire :

Pour obtenir le complément à 1 d'un nombre binaire, il suffit de complémenter chaque bit : devient 0 et 0 devient 1.

Exemple : Le complément à 1 du nombre 011011101 est 100100010. La somme de ces deux nombres est égale à 111111111.

B. Complément à 2 d'un nombre binaire

Pour obtenir le complément à 2 d'un nombre binaire, il suffit d'ajouter 1 à son complément à 1.

Une autre méthode consiste à conserver tous les bits à partir de la droite jusqu'au premier 1 compris et de changer les autres bits de 0 en 1 ou de 1 en 0.

❖ **Exemple(1) :**Le complément à 2 du nombre 011011101 est :

```
      1 0 0 1 0 0 0 1 0
+               1
  -----------------------
=   1 0 0 1 0 0 0 1 1
```

1.3.5 Soustraction par complément à 2

❖ **Règle :**

Etape 1 : Additionner le 1er chiffreau complément à 2 du 2ieme chiffre

Etape 2 : Eliminer le 1 situé le plus à droite du résultat obtenu (débordement). La différence cherchée est ainsi obtenue

WWW.DZPEN.COM | قلم الجزائر

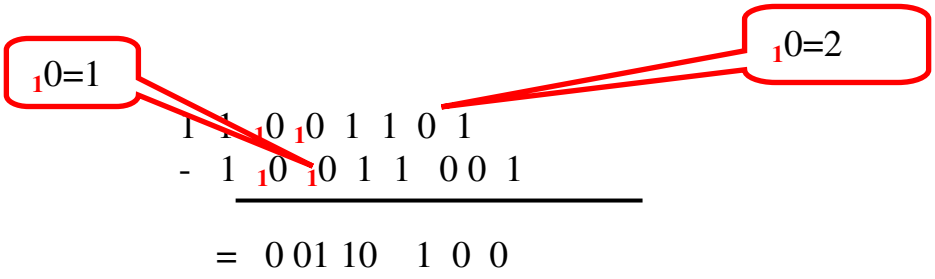
❖ **Exemple(1) :** Réaliser la soustraction suivante $(1100110001)_2$ – $(1001100110)_2$:

A. Utiliser la méthode de la soustraction directe

B. Utiliser la méthode du complément à 2

❖ **Solution :**

A. Soustraction directe : $(11001101)_2$ –$(10011001)_2$



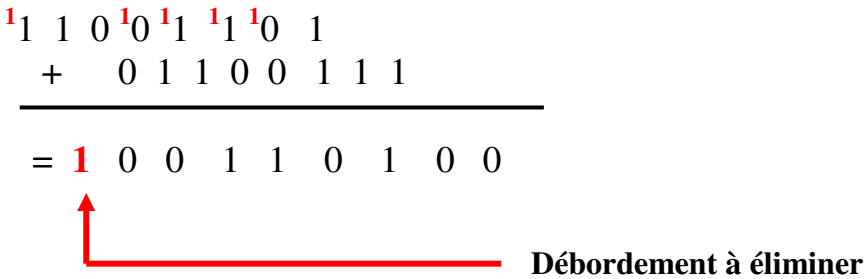
Donc $(11001101)_2$ –$(10011001)_2$= $(00110100)_2$

B. Soustraction (Complément à 2) :$(11001101)_2$ –$(10011001)_2$

$(11001101)_2$ –$(10011001)_2$ = $(11001101)_2$ + [-$(10011001)_2$]

= $(11001101)_2$ +Ca2 $(10011001)_2$

= $(11001101)_2$ + $(01100111)_2$



Donc$(11001101)_2$– $(10011001)_2$= $(00110100)_2$

1.3.6 Exercice 3

A. Effectuez les opérations suivantes en binaire sur 8 bits

- $(9)_{10} + (8)_{10}$
- $(45)_{10}- (41)_{10}$
- $(56)_{10} - (4)_{10}$

B. Effectuez les soustractions suivantes en binaire sur 8 bits en utilisant la méthode directe
puis celle qui utilise le complément a 2

- $(42)_{10} – (29)_{10}$
- $(25)_{10} – (9)_{10}$
- $(12)_{10} – (52)_{10}$

WWW.DZPEN.COM | قلم الجزائر

1.4 Systèmes de codage

Les systèmes utilisés précédemment constituent des systèmes naturels ou codes naturels ou encore codes pondérés. Ils sont caractérisés par le fait que le poids du chiffre de rang **i** est **B** fois celui du rang **i-1**, **B** étant la base du système.

Il existe d'autres codes qui possèdent des avantages particuliers pour des utilisations particulières, on peut citer : le code Gray, code BCD (BinaryCoded Décimal), code majoré de trois,code Aiken .

L'utilisation d'un code particulier peut rendre le traitement d'un message plus au moins économique du point de vue temps de traitement ou encombrement en mémoire ou encore en nombre de composant nécessaire pour effectuer le traitement.

1.4.1 Code Gray ou binaire réfléchi

Ce code est utilisé essentiellement dans la conversion d'une grandeur analogique en une grandeur numérique ; car dans ces conversions on a besoin d'un code dans lequel les grandeurs successives ne diffèrent que d'un seul caractère.

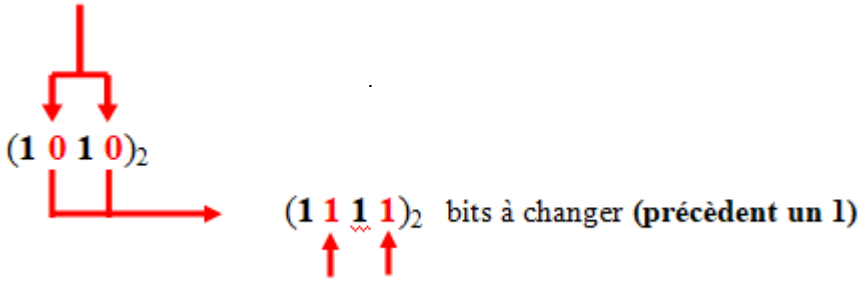
Le code binaire réfléchi est utilisé pour simplifier des équations dans les tableaux de KARNAUGH. Le principe consiste à changer l'état d'un seul bit entre deux nombres consécutifs.

Par exemple pour passer de 7 à 8 décimal, soit de 0111 à 1000 binaire naturel les quatre bits changent.

A. Conversion du Binaire Naturel vers le Binaire Réfléchi

❖ **Règle :**Pour convertir un nombre du binaire naturel au binaire réfléchi, il suffit de changer **le bit qui précède directement un bit 1**.

❖ **Exemple(1) :**Convertir le nombre binaire naturel $(1010)_2$ en binaire réfléchi.



WWW.DZPEN.COM | قلم الجزائر

❖ **Exemple(2)**Conversion des 15 premiers chiffres du binaire naturel en Gray

Tableau 04 : Exemple de conversion Binaire naturel - Gray

Décimal	Binaire naturel	Gray
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100

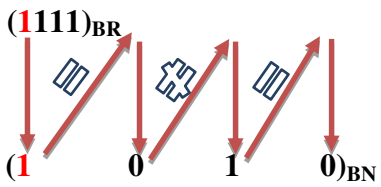
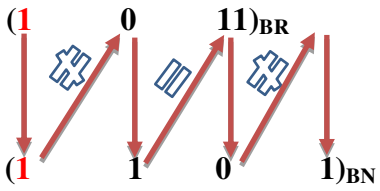
Décimal	Binaire naturel	Gray
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

B. Conversion du Binaire Réfléchi vers le Binaire Naturel

❖ **Règle** : Conversion du Binaire Réfléchi vers le Binaire Naturel :
Le premier bit à gauche reste inchangé
Il s'agit de comparer le bit **bn+1** du **binaire naturel** et le bit **bn**du **binaire réfléchi** le résultat est **bn**du binaire naturel qui vaut **0** si **bn+1=bn**ou **1** sinon.

❖ **Exemple:**Convertir les nombres binairesréfléchi $(1111)_{BR}$ et $(1111)_{BR}$ en binaire naturel

Tableau 05 : Principe de conversion du binaire réfléchi (Gray) en binaire naturel

$(1111)_{BR}=(?)_{BN}$	$(1011)_{BR}=(?)_{BN}$
 $(1111)_{BR}=(1010)_{BN}$	 $(1011)_{BR}=(1101)_{BN}$

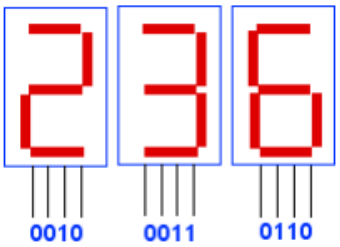
WWW.DZPEN.COM | قلم الجزائر

1.4.2 Code BCD (BinaryCoded Décimal) ou (DCB) Décimal codé en binaire :

Ce code est utilisé par les calculateurs, il est destiné à l'affichage de valeurs décimales, chaque digit doit être codé en binaire sur 4 bits. Il est uniquement destiné à la saisie et à l'affichage de données

❖ **Exemple 01 :** 236 = 0010 0011 0110 (soit un mot de 12 bits)

Figure 06 : Nombre décimal code en DCB



Cours sur la numérotation, science de l'ingénieur en S, Académie Caen

❖ **Exemple 02 :**

Tableau 07 : Exemple de conversion Binaire naturel - Gray

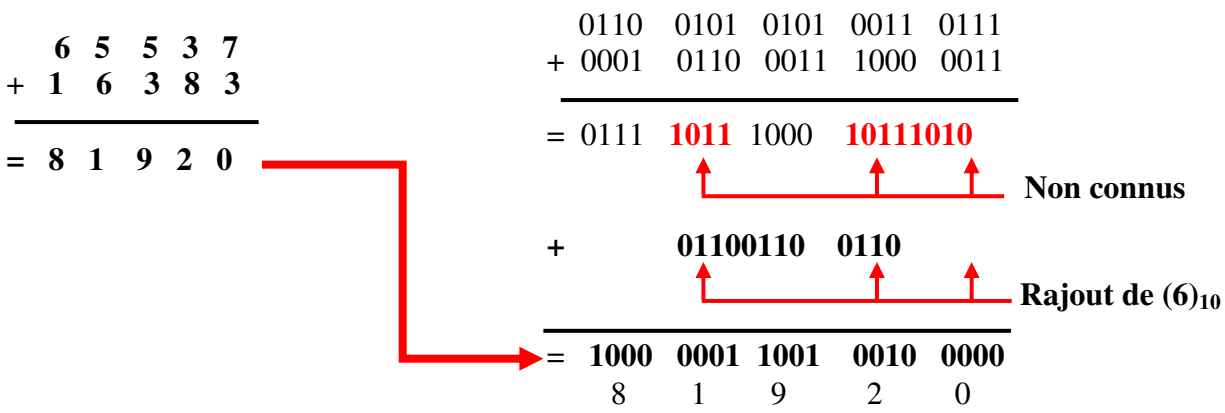
Décimal	Binaire	BCD
258	1 00000010	0010 0101 1000
32	10 0000	0011 0010
69	100 0101	0110 1001
4096	1 0000 0000 0000	0100 0000 1001 0110

❖ **Addition dans le code BCD** :

Règle : Pour additionner deux nombres décimaux codés en binaire, il suffit de

1- Procéder à l'addition en binaire naturel ;
2- Ajouter le nombre $(6)_{10}$ c'est à dire $(0110)_2$, à tout quartet ne correspondant pas à une valeur connue par le code BCD (les nombres dépassant $(9)_{10}$ ou $(1001)_2$).

❖ **Exemple :** Faire l'addition suivante en DCB , $(65537)_{10} + (16383)_{10}$



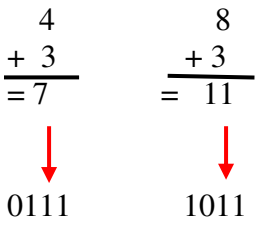
WWW.DZPEN.COM | قلم الجزائر

1.4.3 Code majoré de trois (excess 3)

On a parfois recours àce code en raison de la facilité avec laquelle on peut faire certains calculs arithmétiques.La valeur d'un mot en code majoré de trois est en fait égale au code DCB auquel on a ajouté 3.

❖ **Règle :**Le code majoré de trois consiste à prendre chaque chiffre décimal a part, à lui additionner 3, puis à convertir le résultat obtenu en binaire**.**

❖ **Exemple :**Convertir le nombre $(48)_{10}$ en code majoré de 3



$(48)_{10}= (0111\ 1011)_{XS3}$

1.4.4 Code binaire de Aiken

Pondéré par 2421, le code Aiken est un code auto complémentaire, c'est-à-dire les représentationsde 2 chiffres dont la somme est 9 sont complémentaires l'une de l'autre.

❖ **Règle :**Le code Aiken peut être constitué par les règles suivantes :
- ✓ De 0 à 4 on code en binaire pur ;
- ✓ De 5 à 9 on ajoute 6 et on code en binaire pur.

Exemple : 5→5+6 =11
6→6+6= 12

❖ **Exemple**

Tableau 08 : Exemple de conversion Décimal - Aiken

Décimal	Aiken			
	2	4	2	1
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	1	0	1	1
6	1	1	0	0
7	1	1	0	1
8	1	1	1	0
9	1	1	1	1

WWW.DZPEN.COM | قلم الجزائر

1.4.5 **Exercice 4**Complétez le tableau suivant

Tableau 09: Tableau de conversion entre systèmes de numération et systèmes de codage

Code/Base	Nombre	Equivalent	Dans le Base/Code
2	111011101		Gray
10	7		XS3
2	110010		DCB
8	17		Gray
Gray	110010001010		2
Gray	1011		XS3

1.5 Résumé

On vient de terminer l'étude sur les systèmes de numérotation et de codage. Il apparaît intéressant de récapituler les points suivants.

Outre le système de numérotation décimale, il existe trois autres systèmes de numérotions couramment utilisés dans les dispositifs industriels, ce sont le système binaire, le système octal, et le système hexadécimal. Ces trois systèmes sont des systèmes pondérés où le poids de pondération de chaque chiffre dépend de son rang dans le nombre.

On se qui concerne les opérations (addition, soustraction, multiplication) leurstechniques de base restent exactement les mêmes qu'en notation décimale ; elles sont juste simplifiées de façon drastique parce qu'il n'y a que les deux chiffres 0 et 1

Parmi les techniques de codage les plus utilisées, on retrouve les codes *DCB*, le code *Gray* et le code *Aiken.*

- Le code DCB est un code pondéré. Chaque chiffre du système décimal est représenté dans ce code par une séquence de quatre bits. L'avantage de ce code est de fournir une méthode plus rapide et plus simple de représenter les nombres décimaux en nombres binaires.
- Le code Gray, aussi appelé "code binaire réfléchi", est un code non pondéré où les chiffres ne diffèrent des précédents que d'un seul bit. Ce code est très utilisé dans les convertisseurs analogiques numériques.
- Le code Aiken étant fréquemment utilisé dans les automatismes industriels, il est par conséquent nécessaire de le convertir aux interfaces en code 8421 et inversement Le code a été développé par Howard Hathaway Aiken et est encore utilisé aujourd'hui dans les horloges numériques , calculatrices de poche et autres appareils similaires.

Après avoir appris les systèmes de numérotation et de codage, une étude vous permet d'aborder la logique booléenne, les notions de base de l'algèbre booléenne et les opérateurs de base dans cette algèbre.

WWW.DZPEN.COM | قلم الجزائر

1.6 Exercice de synthèse1

- **Spécialité** : Domotique
- **Module** : Logique combinatoire et logique séquentielle
- **Durée :** 1h
- **But :** Familiariser les stagiaires avec les systèmes de numération et les codes utilisés par les appareils et équipements numériques et informatiques pour traiter les informations.
- **Matériel requis** : Documents et support de cours
- **Mise en situation :**Réaliser les conversions et opérations suivantes
- **Marche à suivre :**

A. Complétez le tableau10 suivant

Tableau 10 : Conversion entre systèmes de numération

Base10	Base 2	Base 8	Base16
125			
			F0CD
	1011,11		
		125,4	

B. Complétez le tableau 11 suivant

Tableau 11 : Conversion entre systèmes de numération et de codage

Base	Nombre	Equivalent	Dans la base
10	121,875		2
16	A01		DCB
DCB	10010110		Gray
8	25,45		16

WWW.DZPEN.COM | قلم الجزائر

C. Soit les deux tableaux suivants

- Complétez le tableau 12 suivant

Tableau 12 : Complémentations

Nombre	Complément à 1	Complément à 2
A=10000100		
B=01110001		

- Calculer en Binaire sur 8 bits

- **Tableau 13** : Operations arithmétique

A+B	
A-B	
B-A	
-A-B	

CHAPITRE 2

WWW.DZPEN.COM | قلم الجزائر

Les machines numériques sont constituées d'un ensemble de circuits électroniques.Chaque circuit fournit une fonction logique bien déterminée (addition, comparaison,….).
La fonction F (A, B) peut être : la somme de A et B, ou lerésultat de la comparaison de A et B ou une autre fonction

***Figure*14** : Circuit électronique



Pour concevoir et réaliser ce circuit on doit avoir un modèle mathématique de la fonction réalisée par ce circuit et qui doit prendre en considération le système binaire.
Le modèle mathématique utilisé est ***l'Algèbre de BOLLE***de***George BOOLE***qui est un mathématicien anglais (1815-1864).Il a fait des travaux dont les quels les fonctions (expressions) sont constituées par des variables qui peuvent prendre les valeurs '**OUI**' ou '**NON**'.
Ces travaux ont été utilisés pour faire l'étude des systèmes qui possèdent deux états

- Le système peut être uniquement dans deux états E1 et E2 tel que E1 est l'opposé de E2.
- Le système ne peut pas être dans l'état E1 et E2 en même temps

Ces travaux sont bien adaptés au Système binaire (0 et 1).

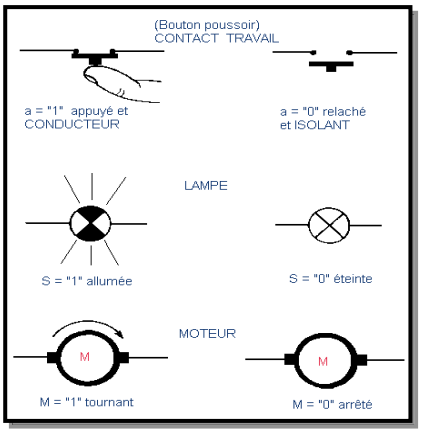
2.1 Variables et fonctions logiques

2.1.1 Variable logique (booléenne)

Une variable logique est une grandeur qui ne peut prendre que deux valeurs 0 ou 1,de manière général, elle peut être associé à un événement, si ce dernier est vrai, alors la variable prend la valeur 1, si non 0.

- **Exemple :** état d'un interrupteur, d'un bouton poussoir, la présence d'une tension,...Soit a la variable associée à l'état d'un bouton poussoir, alors ***A=0 (faux ou bas)*** signifiequ'il n'est pas actionné, ***A= 1 (vrai ou haut)*** signifie qu'il est actionné.

Figure 15 : Quelques illustrations de l'état logique



https://www.electronique-et-informatique.fr/

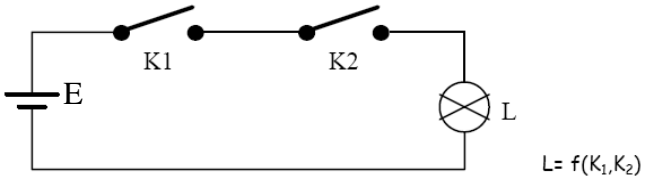
WWW.DZPEN.COM | قلم الجزائر

2.1.2 Fonction logique (booléenne)

C'est une fonction d'une ou plusieurs variables booléennes, ne pouvant prendre elle-même qu'une des deux valeurs 0 ou 1. Pour la définir, il faut préciser sa valeur pour toutes les combinaisons possibles.

- **Exemple :** La lampe **L** s'allume lorsque les interrupteurs **K1** et **K2** sont fermés tous les deux. Elle est éteinte dans tous les autres cas. Voir **figure16**

***Figure* 16** : Commutateurs en série



Bases des systèmes Numériques (Par A. Oumnad)

Le fonctionnement est illustré par la table de vérité suivante :
Le doublet des deux variables (**K1, K2**) peut prendre $\mathbf{2^2}$ **valeurs** distinctes. Dans le cas général de **N** variables, il y aura $\mathbf{2^N}$ configurations possibles
Les variables **K1** et **K2** sont dites les entrées du système, **L** est dite la sortie du système.

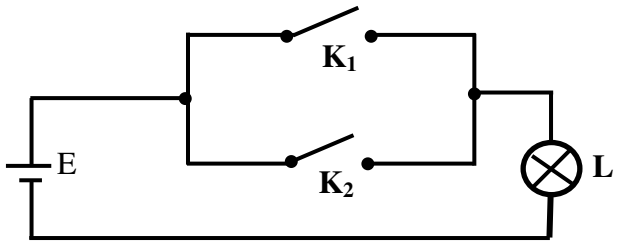
Tableau 17 Table de vérité

K_1	K_2	L
0	0	**0**
0	1	**0**
1	0	**0**
1	1	**1**

De la table de vérité en peut déduire $\mathbf{L= K_1 . K_2}$

2.1.3 Exercice 05 Donner la table de vérité du circuit électronique suivant

***Figure* 18** : Commutateurs en parallèle



WWW.DZPEN.COM | قلم الجزائر

2.2 Opérateurs Logiques (Portes logiques)

Les fonctions logiques sont conçues à partir d'un groupe d'opérateurs élémentairesappelés **« portes ».** Chaque opérateur est représenté par un symbole et safonction est définie par une table de vérité.

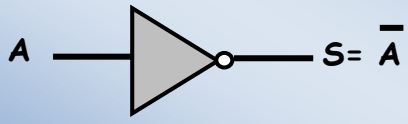
2.2.1 Opérateur (Porte)NON (NOT)

❖ **Table de vérité :**

A	S
0	1
1	0

❖ **Equation :** $S = \overline{A}$

❖ **Symbole :**



$\overline{A}$ est le complément à 1 de A

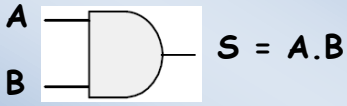
2.2.2 Opérateur (Porte) ET (AND)

❖ **Table de vérité :**

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

❖ **Equation: S= A et B = A.B**

❖ **Symbole**



A.B est **vrai** si et seulement si **A est vraie et B est vraie**

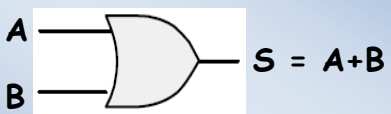
2.2.3 Opérateur (Porte) OU (OR)

❖ **Table de vérité :**

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

❖ **Equation: S= A ou B = A+B**

❖ **Symbole :**



A+B est vraie si au moins une des deux variables A ou B est vraie

WWW.DZPEN.COM | قلم الجزائر

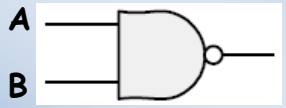
2.2.4 Opérateur (Porte) NON-ET (NAND)

❖ **Table de vérité :**

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

❖ **Equation:** $S = \overline{A \text{ et } B} = \overline{A.B}$

❖ **Symbole :**



C'est le **complément** de l'opérateur **ET.** C'est l'opérateur le plus couramment utilisé dans la pratique.

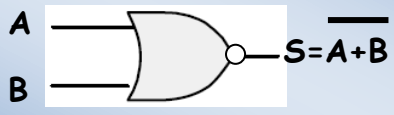
2.2.5 Opérateur (Porte) NON-OU (NOR)

❖ **Table de vérité :**

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

❖ **Equation:** $S = \overline{A \text{ ou } B} = \overline{A+B}$

❖ **Symbole :**



C'est le **complément** de l'opérateur **OU**

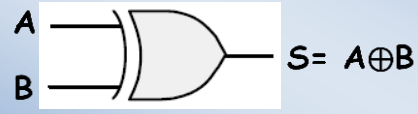
2.2.6 Opérateur (Porte) OU EXCLUSIF (XOR)

❖ **Table de vérité :**

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

❖ **Equation:** $S=\overline{A}.B + A.\overline{B} = A \oplus B$

❖ **Symbole :**



A⊕Best **vrai**e si une des deux variables A ou B est **vraie** mais pas les deux à la fois.

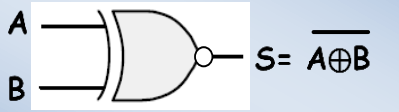
2.2.7 Opérateur (Porte) NON-OU EXCLUSIF (XNOR)

❖ **Table de vérité :**

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

❖ **Equation:** $S= \overline{A}.\overline{B} + A.B = \overline{A \oplus B}$

❖ **Symbole :**

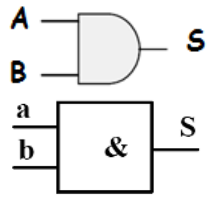
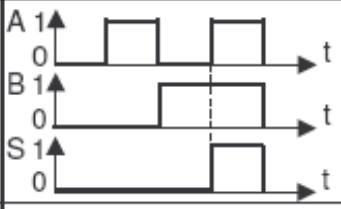
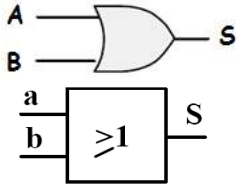
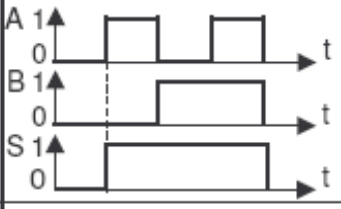
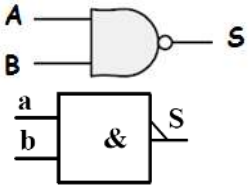
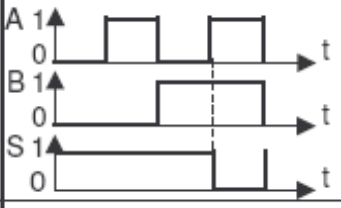
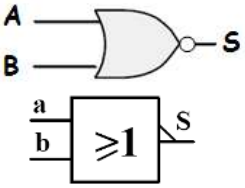
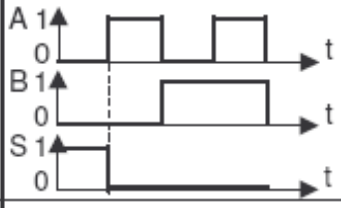
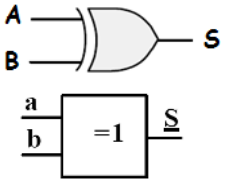
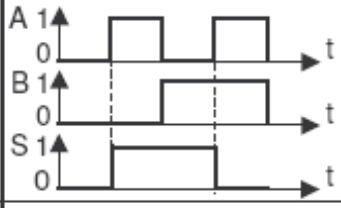
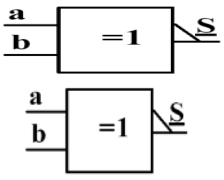
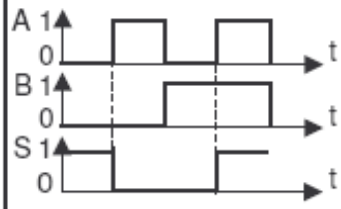
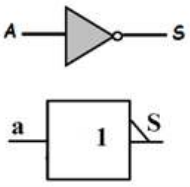
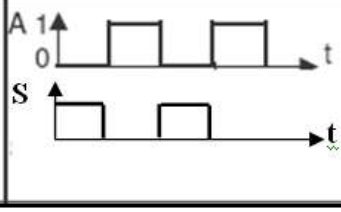


$\overline{A \oplus B}$ est **vraie** si **les deux variables à la fois** A etB sont**vraie**

WWW.DZPEN.COM | قلم الجزائر

2.2.8 Synthèse des portes logiques

Tableau *19* : Commutateurs en parallèle

Fonction	Symbole	Equation	Table de vérité	Chronogramme
ET AND		$S = A \bullet B$	A B S 0 0 0 0 1 0 1 0 0 1 1 1	
OU OR		$S = A + B$	A B S 0 0 0 0 1 1 1 0 1 1 1 1	
ET-NON NAND		$S = \overline{A \bullet B}$	A B S 0 0 1 0 1 1 1 0 1 1 1 0	
OU-NON NOR		$S = \overline{A + B}$	A B S 0 0 1 0 1 0 1 0 0 1 1 0	
OU exclusif		$S = A \oplus B$ ou $S = A \bullet \overline{B} + \overline{A} \bullet B$	A B S 0 0 0 0 1 1 1 0 1 1 1 0	
NON OU exclusif		$S = \overline{A \oplus B}$ ou $S = \overline{A} \bullet \overline{B} + A \bullet B$	A B S 0 0 1 0 1 0 1 0 0 1 1 1	
NON (NOT)		$S = \overline{A}$	A S 0 1 1 0	

WWW.DZPEN.COM | قلم الجزائر

Remarque :

- **1erSymbole :**Symbole **A.N.S.I.** (Américan National Standards Institute) ou « **AMERICAIN** ». C'est ceux que l'on utilisera dans nos schémas électriques
- **2^{iem} Symbole :**Symbole**I.E.E.E.** (Institute of Electrical and Electronics Ingenieers) ou « **EUREPEEN** »

2.2.9 Exercice 06Tracer le circuit logique des fonctions suivantes :

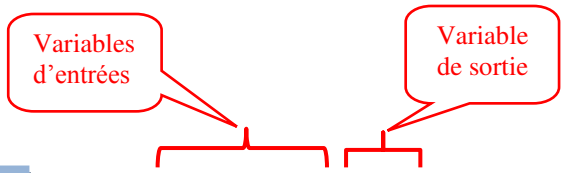
S1 = (A + B). (C + D)

S2 = $(\bar{A}$ + B). $\overline{C.D}$

2.3 Table de vérité

La table de vérité (***TV***) est un tableau qui permet de connaître systématiquement les états que peut prendre une fonction logique pour TOUTES les combinaisons des variables d'entrés. En d'autres termes, c'est un tableau qui interprète logiquement un cahier de charge en tenant compte de toutes les éventualités voir figure 20.

Tableau 20 : Exemple de TV à 3 variables



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

Dans une TV pour **n** variables, nous aurons **2^{n}**combinaisons différentes, c'est-à-dire :

- Pour **1** variable, la TV aura $2^1 = 2$ lignes
- Pour **2** variables, la TV aura $2^2 = 4$ lignes
- Pour **3** variables, la TV aura $2^3 = 8$ lignes
- Pour **4** variables, la TV aura $2^4 = 16$ lignes

2.3.1 Extraction de la fonction logique à partir de la T.D.V

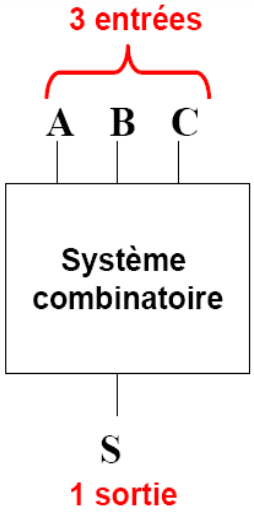
Si une fonction logique possède **N** variables logiques on aura doc **2^{N}**combinaisons.Les **2^{N}** combinaisons sont représentés dans une table qui s'appelle table de vérité.

Une table de vérité recense l'ensemble des états d'une sortie pour toutes les combinaisons possibles des variables d'entrée.

WWW.DZPEN.COM | قلم الجزائر

➢ **Exemple 01** Soit un système combinatoire à 3 entrées **A**, **B**, **C** et une sortie **S**

Figure 21 : Système combinatoire



❖ **Table de vérité**

A	B	C	S	
0	0	0	0	→ A+B+C : max terme
0	0	1	0	→ A+B+$\overline{C}$: max terme
0	1	0	0	→ A+$\overline{B}$+C : max terme
0	1	1	1	→ A.$\overline{B}$.C : min terme
1	0	0	0	→ $\overline{A}$+B+C : max terme
1	0	1	1	→ A.$\overline{B}$.$\overline{C}$: min terme
1	1	0	1	→ A.B.$\overline{C}$: min terme
1	1	1	1	→ A.B.C : min terme

D'après la table de vérité précédente en déduit

➢ S = somme min termes

S (A, B, C) = (A.B.C) + ($\overline{A}$.B.C) + (A.$\overline{B}$.C) + (A.B.$\overline{C}$)

C'est la **Première forme canonique** (somme des produits).Cette forme est la forme la plus utilisée.

➢ S = produit des max termes

S (A, B, C) = (A+B+C). (A+B+$\overline{C}$).(A+$\overline{B}$+C). ($\overline{A}$+B+C)

C'est la**Deuxième forme canonique** (produit des sommes).

❖ **Remarque :**

Première forme canonique = expression des 1 de la fonction

Deuxième forme canonique = expression des 0 de la fonction

Les deux formes canoniques sont équivalentes

On choisit celle qui donne le résultat le plus simplepeu de 0 => deuxième forme / peu de 1 => premièreforme

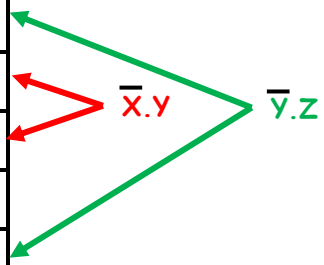
WWW.DZPEN.COM | قلم الجزائر

2.3.2 Déduction de la TDV à partir de la fonction logique

❖ **Exemple01** Déduire la TDV de la fonction suivante : F (X, Y, Z) = $\overline{X}$. Y + $\overline{Y}$.Z

➢ **Table de vérité**

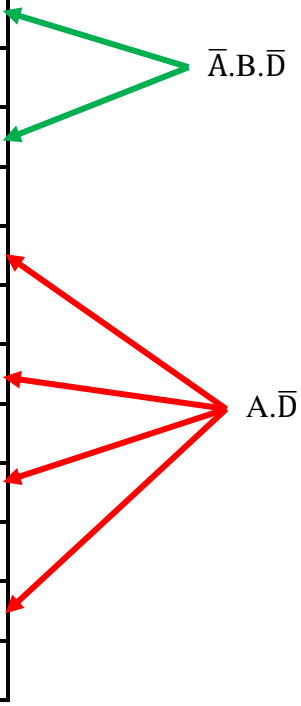
X	Y	Z	F
0	0	0	**0**
0	0	1	**1**
0	1	0	**1**
0	1	1	**1**
1	0	0	**0**
1	0	1	**1**
1	1	0	**0**
1	1	1	**0**



❖ **Exemple02** Déduire la Table de vérité(TDV) de la fonction suivante :S(A, B, C, D) = A.$\overline{D}$ + $\overline{A}$.B.$\overline{D}$

➢ **Table de vérité**

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



WWW.DZPEN.COM | قلم الجزائر

2.3.3 Exercice 7

A. Donner la table de vérité de la fonction suivantes :$F_1 = X.Y.Z + X.(Y + \overline{Z})$

B. Déduire de la table de vérité la deuxième forme canonique de F_1

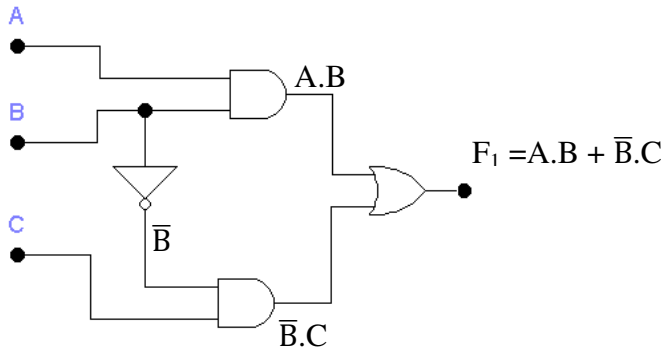
2.4 Schéma d'un circuit logique (Logigramme)

C'est la traduction de la fonction logique en un schéma électronique. Le principe consiste à remplacer chaque opérateur logique par la porte logique qui lui correspond.

2.4.1 Exemple01 Tracer le logigramme de la fonction suivante :

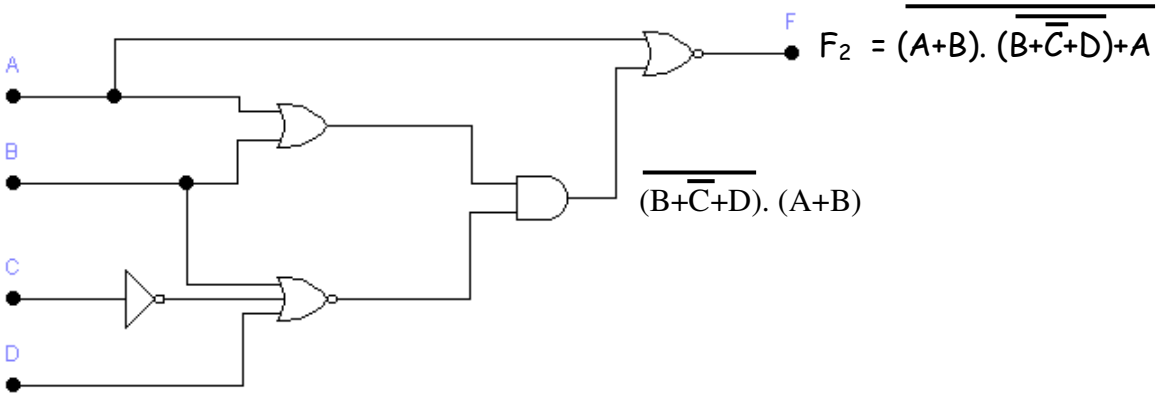
$F_1(A, B, C) = A.B + \overline{B}.C$

Figure22: Logigramme de la fonction F_1



2.4.2 Exemple02 $F_2(A, B, C, D) = \overline{(A+B).\overline{(B+C+\overline{D})}+A}$

Figure23: Logigramme de la fonction F_2



WWW.DZPEN.COM | قلم الجزائر

2.4.3 Logigramme à l'aide des portes NAND uniquement

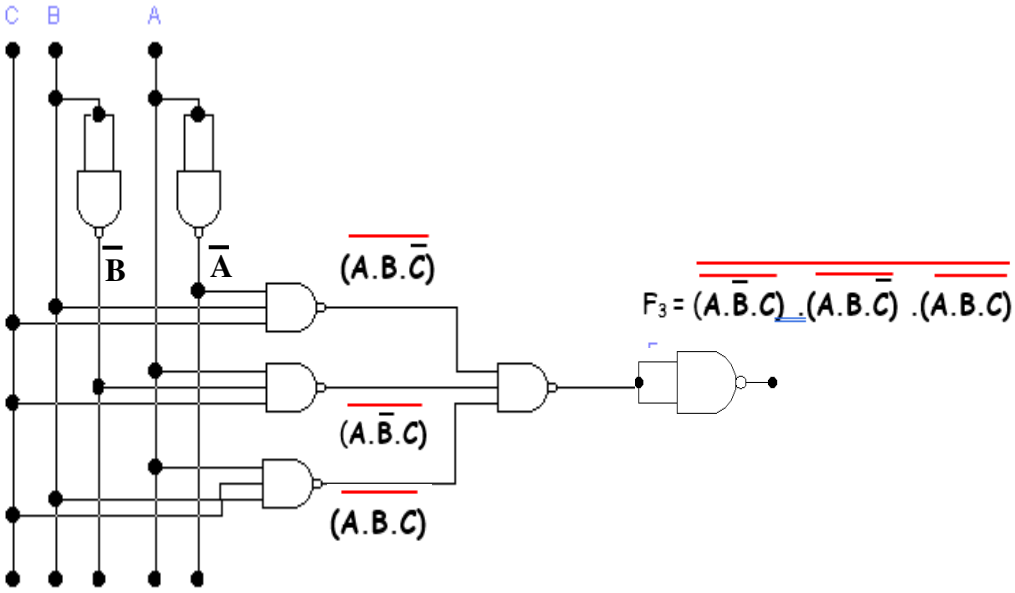
Soit la TDV suivante :

A	B	C	F_3	
0	0	0	**0**	
0	0	1	**0**	
0	1	0	**0**	
0	1	1	**1**	→ $\bar{A}.B.C$: min terme
1	0	0	**0**	
1	0	1	**1**	→ $A.\bar{B}.C$: min terme
1	1	0	**0**	
1	1	1	**1**	→ $A.B.C$: min terme

$F_3=\bar{A}.B.C + A.\bar{B}.C + A.B.C$(Première forme canonique : somme des produits)

$F_3=A.B.C + \bar{A}.B.C + A.\bar{B}.C = \overline{(A.B.C)}\ .\overline{(A.\bar{B}.C)}\ .\overline{(A.\bar{B}.C)}$

Figure24 : Logigramme à l'aide des portes NAND



WWW.DZPEN.COM | قلم الجزائر

2.4.4 Exercice 8

Soit la TDV suivante :

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

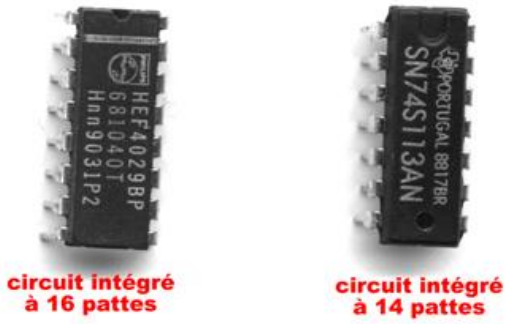
a) Trouver la fonction S ?
b) Réalisez le circuit logique de S

2.5 Circuits intègres des portes logiques

2.5.1 Principe des circuits intégrés

En électronique, les portes logiques sont fabriquées et renfermées dans des **circuits intégrés**. Voir Figure 25

Figure25: Circuits intégrés



www.gecif.net
Auteur : **Jean-Christophe MICHEL**
Professeur de Sciences Industrielles de l'Ingénieur
option Informatique et Numérique

On peut remarquer sur le haut des circuits intégré un petit creux appelé « ergo ». L'ergo permet d'orienter correctement le circuit intégré afin de repérer les différentes bornes.
Un circuit intégré renferme plusieurs portes logiques, dont les entrées et les sorties sont accessibles sur les différentes bornes(pattes) du circuit intégré. Pour identifier chaque borne sans ambiguïté, elles sont numérotées de manière normalisée en respectant le principe suivant :

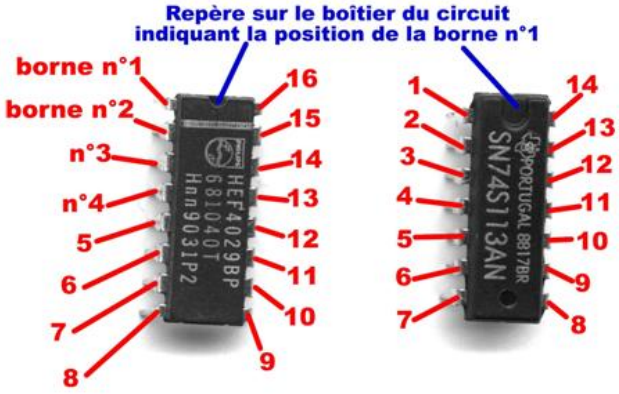
- En regardant le circuit intégré avec l'ergo vers le haut, la borne n°1 est la borne située en haut à gauche.
- Les autres bornes sont numérotées en tournant dans le sens inverse des aiguilles d'une montre.

Ce principe reste vrai quel que soit le nombre de bornes (de « pattes ») du circuit intégré. La Figure 26 montre le numéro de chacune des bornes pour un circuit intégré à 16 bornes (16 « pattes ») et pour un circuit intégré à 14 bornes (14 « pattes »).

WWW.DZPEN.COM | قلم الجزائر

Remarquez le repère (appelé « l'ergo ») situé en haut de chacun des boîtiers, et observez les circuits intégrés dont vous disposez dans le but de repérer cet ergo.

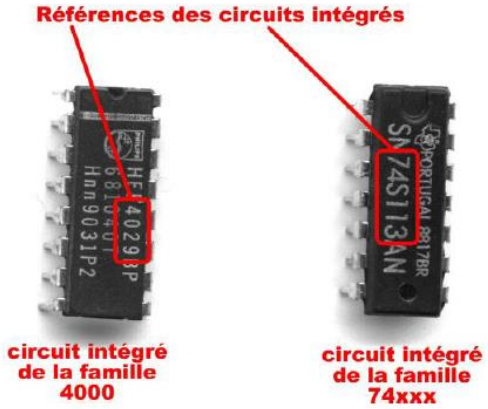
Figure26: *Bornes d'un circuit intégré*



www.gecif.net
Auteur : **Jean-Christophe MICHEL**
Professeur de Sciences Industrielles de l'Ingénieur
option Informatique et Numérique

Un circuit intégré logique renferme des portes logiques. Mais comment savoir quel type de porte (des OU ? des ET-NON ? des Ou-Exclusif ? etc.). Pour cela, chaque circuit intégré possède une référence imprimée sur le dessus de son boîtier. Cette référence est composée de 4 à 7 caractères (chiffres et/ou lettres).
Par exemple, sur la photo ci-contre la référence du circuit de gauche est 4029, et la référence du circuit de droite est 74S113.

FIG 27 : *Référence du circuit intégré*



www.gecif.net
Auteur : **Jean-Christophe MICHEL**
Professeur de Sciences Industrielles de l'Ingénieur
option Informatique et Numérique

WWW.DZPEN.COM | قلم الجزائر

2.5.2 Caractéristiques des circuits intègres série 74-TTL et CMOS rapides :

2.5.2.1 Familles logiques

Dans cette série dix grandes familles sont disponibles :

A. Sept en technologie TTL (Transistor-Transistor-Logic) ou logique à transistors bipolaires.

Famille TTL	Code de désignation
TTL Standard	74XX
TTL low power (faible consommation)	74LXX
TTL schottky	74SXX
TTL fast	74FXX
TTL low power schottky	74LSXX
TTL advancedschottky	74ASXX
TTL advanced low power schotky	74ALXX

B. Deux en technologie CMOS (ComplementaryMetalOxideSemiconducteur)

Famille CMOS	Code de désignation
CMOS rapides (high speed CMOS)	74HCXX et 74HCTXX.
CMOS classiques (technologie identique à la serie 4000)	74CXX

Ces opérateurs logiques sont réalisés par une intégration plus ou moins grande de leurs éléments :

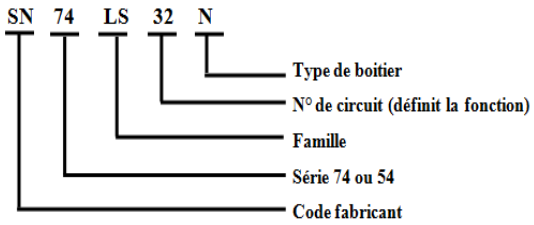
- **SSI** (Small ScaleIntegration : Intégration à petite échelle) permet de supporter plus de 100 composants électroniques sur une même puce .Elles sont réservées aux opérateurs élémentaires et aux bascules.
- **MSI** (MeanScaleIntegration : Intégration à moyenne échelle) permet de supporter plus de 500 composants électroniques sur une même puce . Elles sont réservées pour les compteurs et les registres.
- **LSI**(Large ScaleIntegration : Intégration à grande échelle) c'est-à-dire en des circuits comportant de 1 000 à 10 000 composants. et VLSI (Very Large ScaleIntegration : Intégration à très grande échelle)permet de supporter plus de 100 000 composants électroniques sur une même puce. Elles sont appliquées aux circuits complexes tels que microprocesseurs.

Pour une application donnée le choix de la famille dépend des caractéristiques :

- De tension d'alimentation,
- De sortie,
- De consommation,
- De temps de propagation,
- D'immunité aux bruits.

WWW.DZPEN.COM | قلم الجزائر

2.5.2.2 Code de désignation



- **Exemple :**

  SN : Fabriquant TEXAS
  74 : Série grand public (de 0°C à 70°C)
  54 : Série militaire (de -55°C à 125°C)
  LS : Famille Low power schottky
  32 : Quadruple porte OU à 2 entrées
  N : Boitier DIP 14 (Dual InlinePackage : Double rangées de pattes)

2.5.2.3 Tension d'alimentation

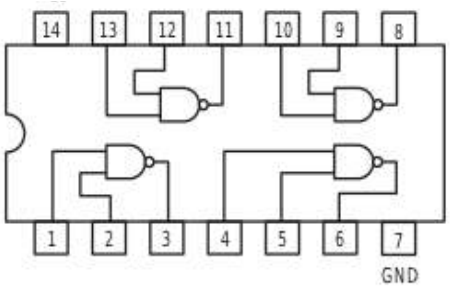
74, 74 L, 74 S, 74 F, 74 LS: 5V+/- 5%
74 ALS, 74 AS, 74 HCT : 5V+/- 10%
74 HC : 2V à 6V
74 C et série 4000 : 3V à 15V

2.5.2.4 Brochage d'un circuit intègre

Exemple de brochage du CI 74 LS 00

(Quad 2-In NAND : 4 Portes NAND à 2 Entrées)

Figure28: Brochage du CI 74 LS 00

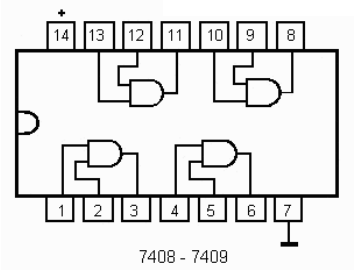


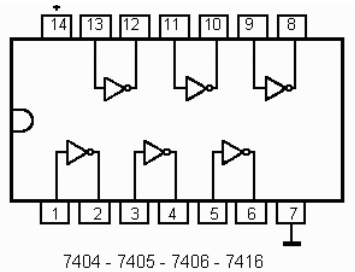
https://www.electronique-et-informatique.fr/

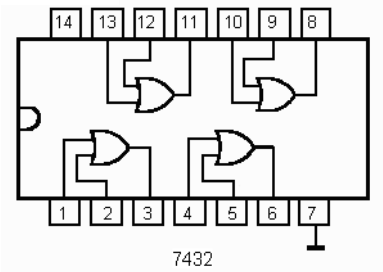
2.5.3 Exercice 9

Réaliser la fonction F= ($\overline{A}$+B).C,utiliser pour cela les circuits intégrersuivants

Figure29:Brochages des circuits intégrer AND-Not-OR







https://www.electronique-et-informatique.fr/

WWW.DZPEN.COM | قلم الجزائر

2.6 Simplification des fonctions logiques

Après la recherche de l'expression algébrique de la fonction, l'étape suivante consiste à minimiser le nombre de termes (de portes logiques) afin d'obtenir une réalisation matérielle plus simple donc plus facile à construire et à dépanner, en plus moins coûteuse.

Trois méthodes de simplification sont utilisées, on abordera que les deux premières:

- La méthode algébrique
- La méthode graphique (diagramme de KARNAUGH).
- La méthode de Mac clauskie

2.6.1 Lois fondamentales de l'algèbre de Boole

Tableau*30*: Lois fondamentales de l'algèbre de Boole

Propriétés	Opérateur OU	Opérateur ET
D'involution	$\overline{\overline{A}} = A$	
Commutativité	A+B = B+A	A. B = B. A
Associativité	(A+B)+C =A+ (B +C)=A+B+C	(A.B).C=A. (B.C)=A.B.C
Distributivité	A+(B.C)= (A+B). (A+C)	A. (B+C)=(A.B) +(A.C)
Complémentarité	$A+\overline{A}=1$	$A.\overline{A}=0$
Idempotence	A + A = A	A. A = A
Elément neutre	A + 0 = A A + 1 = 1	A. 0 = 0 A. 1 = A
Absorption	A + (A. B) = A $A + (\overline{A} . B) = A + B$	A. (A + B) = A $A. (\overline{A} + B) = A . B$
Théorème de MORGAN	$\overline{A + B}=\overline{A} .\overline{B}$	$\overline{A.B}=\bar{A} + \bar{B}$

WWW.DZPEN.COM | قلم الجزائر

2.6.2 Simplification en utilisant la méthode algébrique

Il s'agit d'appliquer les théorèmes et les propriétés de l'algèbre de Boole pour obtenir une expression plus simple de la fonction.

❖ **Exemple 01Simplifier** algébriquement la fonction suivante :

$S=A.B.C + A.\overline{B}.\overline{(\overline{A}.\overline{C})}$

$S=A.B.C + A.\overline{B}.(\overline{\overline{A}.\overline{C}})$

$S=A.B.C + A.\overline{B}.(\overline{\overline{A}}+\overline{\overline{C}})$ $[\overline{\overline{A}}=A]$

$S=A.B.C + A.\overline{B}.(A+C)$

$S=A.B.C + A.\overline{B}.A+A.\overline{B}.C$ $[A.A=A]$

$S=A.B.C + A.\overline{B} + A.\overline{B}.C$

$S=A.B + \overline{A}.C.(B+B)$ $[B+B=1]$

$S=A.B + \overline{A}.C$

$S=A.(B+C)$

❖ **Exemple 02**Simplifier algébriquement la fonction suivante : $F=\overline{\overline{A}.B+\overline{A}+B}$

$F=\overline{\overline{A}.B+\overline{A}+B} = \overline{\overline{A}.B}.\overline{(\overline{A}+B)}$

$F=(\overline{\overline{A}.B}).(\overline{A}+B)$

$F= (A+\overline{B}).(\overline{A}+B)$

$F= A.\overline{A}+A.B+\overline{B}.\overline{A}+\overline{B}.B$

$F= A.B+\overline{A}.\overline{B}$

2.6.3 Tableau de KARNAUGH

Le diagramme de **Karnaugh** est un outil graphique, méthodique. Il permet d'obtenir une solution optimale à la simplification logique.

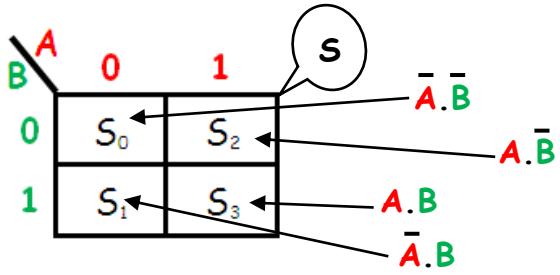
- Le diagramme de Karnaugh consiste à présenter les états d'une fonction logique, non pas sous la forme d'une table de vérité, mais en utilisant un tableau à double entrée.
- Chaque case du tableau correspond à une combinaison des variables d'entrées, donc à une ligne de la table de vérité.
- Le tableau de Karnaugh aura autant de cases que la table de vérité possède de lignes.
- Les lignes et les colonnes du tableau sont numérotées selon le code **binaire réfléchi** (**code de Gray**) : à chaque passage d'une case à l'autre, une **seule** variable change d'état (**cases adjacentes**).
- Le diagramme de Karnaugh est un tableau de $\mathbf{2^N}$ **cases**, **N** étant le nombre de variables.

WWW.DZPEN.COM | قلم الجزائر

2.6.3.1 Diagramme de KARNAUGH à 2 variables

A. Table de vérité B. Tableau de KARNAUGH

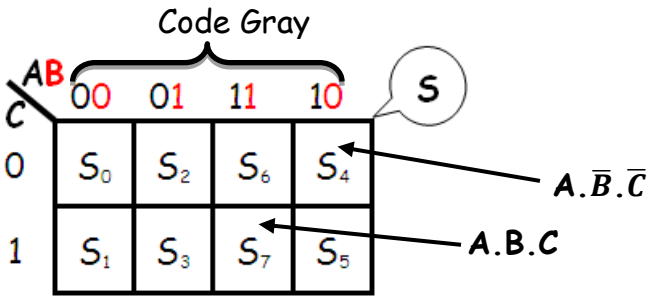
	A	B	S
0	0	0	S_0
1	0	1	S_1
2	1	0	S_2
3	1	1	S_3



2.6.3.2 Diagramme de KARNAUGH à 3 variables

A. Table de vérité B. Tableau de KARNAUGH

	A	B	C	S
0	0	0	0	S_0
1	0	0	1	S_1
2	0	1	0	S_2
3	0	1	1	S_3
4	1	0	0	S_4
5	1	0	1	S_5
6	1	1	0	S_6
7	1	1	1	S_7

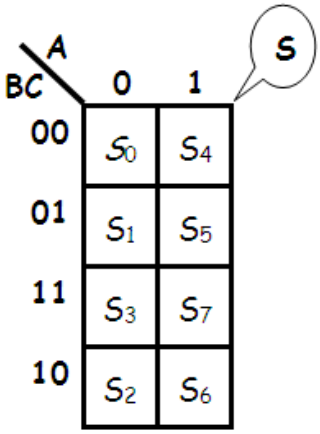


Remarque 01 Dans tableau de KARNAUGH deux cases sont dites **ADJACENTES** si **UNE** seule variable change d'état d'une case à l'autre

Exemple : Les cases S_0 et S_2sont des cases ADJACENTES (S_0=A.B.C et S_2=A.$\bar{B}$.C une seul variable change d'état c'est B)

WWW.DZPEN.COM | قلم الجزائر

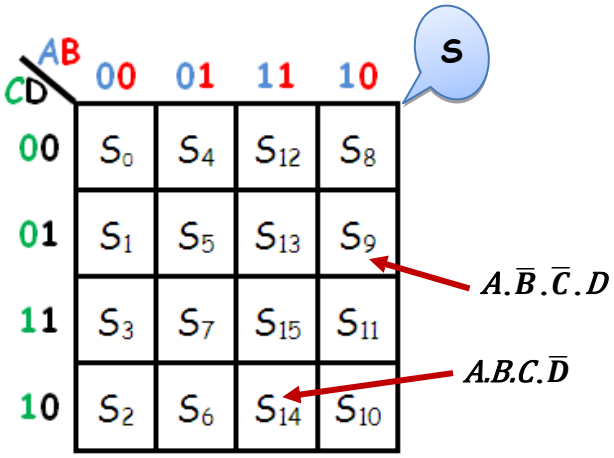
Remarque 01 Autre représentation du tableau de KARNAUGH à 3 variables



2.6.3.3 Diagramme de KARNAUGH à 4 variables

A. Table de vérité B. Tableau de KARNAUGH

	A	B	C	D	S
0	0	0	0	0	S_0
1	0	0	0	1	S_1
2	0	0	1	0	S_2
3	0	0	1	1	S_3
4	0	1	0	0	S_4
5	0	1	0	1	S_5
6	0	1	1	0	S_6
7	0	1	1	1	S_7
8	1	0	0	0	S_8
9	1	0	0	1	S_9
10	1	0	1	0	S_{10}
11	1	0	1	1	S_{11}
12	1	1	0	0	S_{12}
13	1	1	0	1	S_{13}
14	1	1	1	0	S_{14}
15	1	1	1	1	S_{15}



WWW.DZPEN.COM | قلم الجزائر

2.6.4 Simplification en utilisant le tableau de KARNAUGH

On a pu s'apercevoir que la méthode de simplification algébrique devenait vite très longue et fastidieuse dès que le nombre de variables devenait important.

La méthode du tableau de **KARNAUGH** va nous permettre d'effectuer des simplifications beaucoup plus rapidement sans avoir à écrire de longues équations

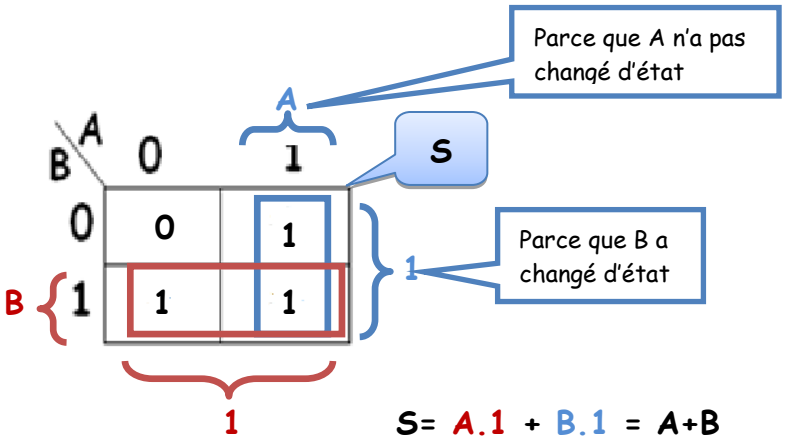
Règles desimplification :

- ✓ Constituer des groupes de **1, 2, 4, 8,16**…… cases **ADJACENTES** (2^n cases) (on choisit des cases contenant des **1** ou des **0** pas les deux à la fois)
- ✓ Ces groupes de taille maximale, doivent être carré ou rectangulaire ;
- ✓ Ils doivent contenir un nombre de cases égal à une puissance de deux ;
- ✓ Les bords de la table doivent êtres adjacents ;
- ✓ On ne retient que les variables dont l'état logique d'entré n'est pas modifié à l'intérieur du groupement ;
- ✓ Les mêmes termes peuvent participer à plusieurs groupements,
- ✓ On doit avoir le moins de groupes possible, chaque groupe doit contenir le maximum de terme (1 ou 0)
- ✓ Les variables d'un même groupement sont liées par une fonction ET(.), les groupements sont liés par des fonctions OU (+).
- ✓ Après avoir fait la somme des fonctions de chaque groupement on obtiendra l'équation finale de **S**

2.6.4.1 Simplification d'une fonction à 2 variables

A. Table de véritéB.Tableau de KARNAUGH

	A	B	S
0	0	0	0
1	0	1	1
2	1	0	1
3	1	1	1



WWW.DZPEN.COM | قلم الجزائر

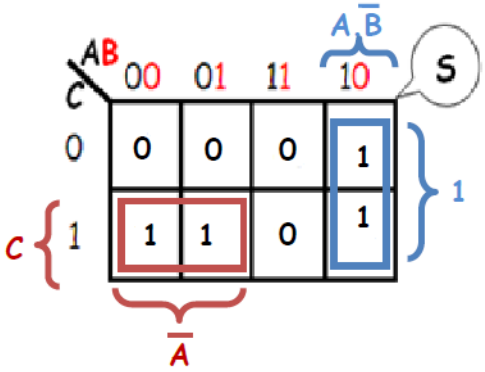
2.6.4.2 Simplification d'une fonction à 3 variables

A. Exemple 01

❖ Table de vérité

	A	B	C	S
0	0	0	0	0
1	0	0	1	1
2	0	1	0	0
3	0	1	1	1
4	1	0	0	1
5	1	0	1	1
6	1	1	0	0
7	1	1	1	0

❖ Tableau de KARNAUGH



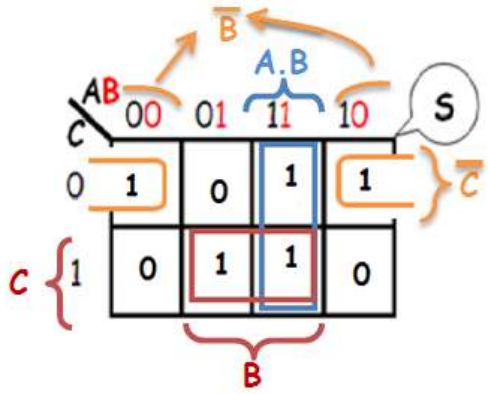
$S = \overline{A}.C + A.\overline{B}$

B. Exemple 02

❖ Table de vérité

	A	B	C	S
0	0	0	0	1
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	1
5	1	0	1	0
6	1	1	0	1
7	1	1	1	1

❖ Tableau de KARNAUGH



$S = B.C + A.B + \overline{B}.\overline{C}$

WWW.DZPEN.COM | قلم الجزائر

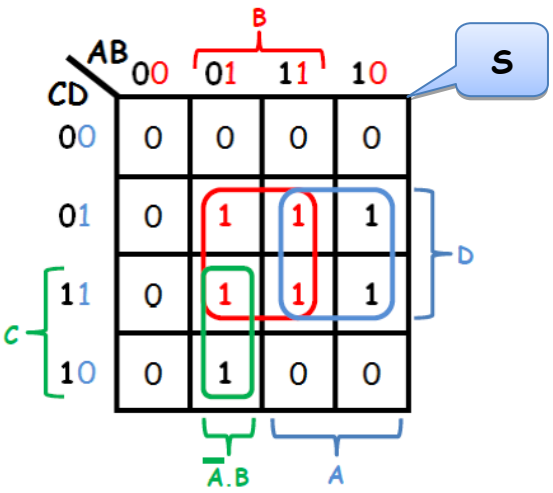
2.6.4.3 Simplification d'une fonction à 4 variables

A. Exemple 01 Simplifier la fonction de la TDV (Table de vérité) suivante en utilisant le tableau de KARNAUGH (TDK).

❖ Table de vérité

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

❖ Tableau de KARNAUGH



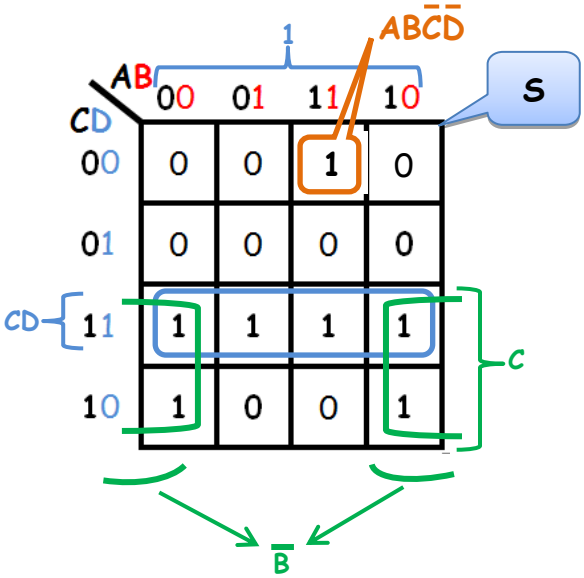
$S = A.D + B.D + \bar{A}.B.C$

B. Exemple 02 Simplifier la fonction de la TDV (Table de vérité) suivante en utilisant le tableau de KARNAUGH (TDK).

❖ **Table de vérité**

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

❖ Tableau de KARNAUGH



$S = C.D + \bar{B}.C + A.B.\bar{C}.\bar{D}$

WWW.DZPEN.COM | قلم الجزائر

2.6.5 Fonctions incomplètement définies

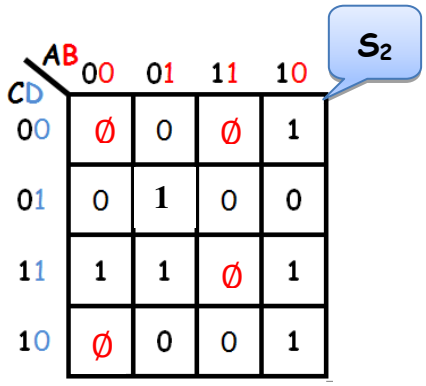
Parfois, la fonction à réaliser n'est pas complètement définie, c'est à dire que pour certaines combinaisons des variables, la valeur logique que prend la fonction n'est pas connue. Ceci pour plusieurs raisons:

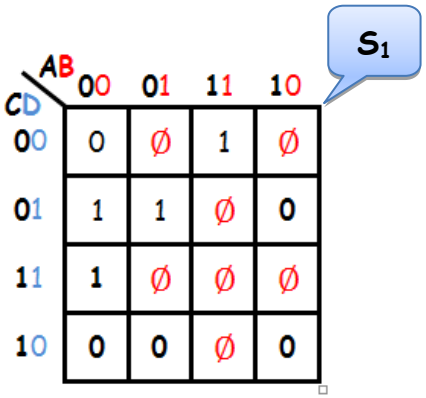
- ✓ Soit qu'on est indifférent à la valeur qu'elle peut prendre pour certaines combinaisons des variables.
- ✓ Soit que ces combinaisons correspondent, dans la pratique, à des configurations impossibles.

Les cases du tableau de KHARNAUGH correspondantes à ces combinaisons sont appelées des cases Ø .

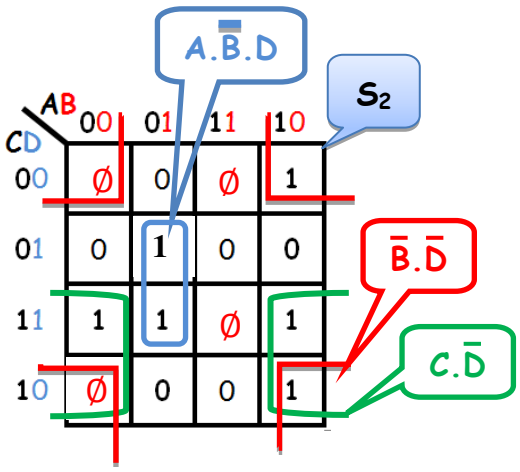
Pour trouver l'expression la plus simple de la fonction logique, on pourra utiliser ces casesØ, selon le cas soit comme des cases 1, soit comme des cases 0. Il n'y a pas de règle permettant un choix judicieux, mais en général on attribuera à chaque case Ø la valeur (0 ou 1) qui permet d'obtenir le plus petit nombre de contractions de plus grande taille possible qui englobe toutes les cases.

❖ **Exemple** Simplifier les fonctions suivantes :

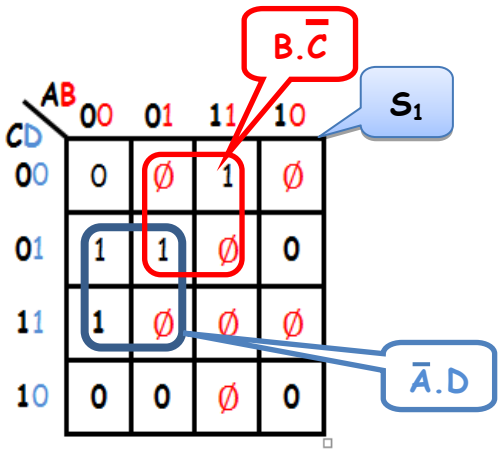




❖ **Solution**



$S_1 = \bar{A}.B.D + \bar{B}.\bar{D} + C.\bar{D}$



$S_1 = \bar{A}.D + \bar{B}.C$

WWW.DZPEN.COM | قلم الجزائر

2.6.6 Exercice 10

Une serrure de sécurité s'ouvre en fonction de quatre clés A, B, C D. Le fonctionnement de laserrure est défini comme suite :

S (A, B, C, D)= 1 si au moins deux clés sont utilisées

S (A, B, C, D)= 0 sinon

Les clés A et D ne peuvent pas être utilisées en même temps

On remarque que si la clé A et D sont utilisées en même temps l'état du système n'est pas déterminé. Ces cas sont appelés cas impossibles ou interdites.

A. Trouver la table de vérité
B. Simplifier la fonction en utilisant tableau de ***KHARNAUGH***

2.7 Résumé

Dans ce chapitre le plus important à retenir :

- Une variable logique est une grandeur qui ne peut prendre que deux valeurs 0 ou 1, est une fonction logique est constituer d'une ou plusieurs variables logiques, ne pouvant prendre elle-même qu'une des deux valeurs 0 ou 1
- Les fonctions logiques sont conçues à partir d'un groupe d'opérateurs élémentaires appelés **« Portes logiques »** et qui sont **:** Porte NON,Porte OU, Porte ET,Porte NON-ET,Porte NON-OU,Porte OU-Exclusif,Porte NON-OU Exclusif
- La table de vérité (***TV***) est un tableau qui permet de connaître systématiquement les états que peut prendre une fonction logique pour toutes les combinaisons des variables d'entrés et que dans une table de vérité de **n** variables, nous avons $\mathbf{2^n}$combinaisons différentes
- Un circuit logique ou logigrammeest la traduction de la fonction logique en un schéma électronique. Le principe consiste à remplacer chaque opérateur logique par la porte logique qui lui correspond
- En électronique, les portes logiques sont fabriquées et renfermées dans des **circuits intégrés**

- Avant de traduire de la fonction logique en un circuit logique, il faut d'abord minimiser le nombre de portes logiques afin d'obtenir une réalisation matérielle plus simple donc plus facile à construire et à dépanner et moins coûteuse, pour cela il existe 3 méthodes de simplification :
  - ✓ La méthode algébrique (Elle devient difficile au de la de 3 variables)
  - ✓ La méthode graphique (diagramme de ***KARNAUGH***). Elle devient difficile au de la de 5 variables
  - ✓ La méthode de ***QUINE MAC CLUSKEY*** (Qui n'est pas aborder dans ce document)

WWW.DZPEN.COM | قلم الجزائر

2.8 Exercice de synthèse 2

- **Spécialité** : Domotique
- **Module** : Logique combinatoire et logique séquentielle
- **Durée :** 1h30
- **But :** Amener les stagiaires à :
  - ✓ Appliquer les concepts de la logique combinatoire en utilisant les notions de variables et d'opérations logiques
  - ✓ Prendre connaissances de la théorie formelle basée sur l'Algèbre de Boole pour résoudre des problèmes et concevoir des circuits logiques simplifiés.
- **Matériel requis** : Documents, support de cours, simulateur électronique
- **Mise en situation :** Réaliser les conversions et opérations suivantes
- **Marche à suivre :**

  - **Question1 :** Établir la table de vérité de la fonction suivante, puis l'écrire sous la 1ère forme canonique :F1= $(X + Y).(\bar{X} + Y + Z)$
  - **Question2 :** Réaliser la porte EXOR a l'aide de portes NAND uniquement
  - **Question3 :** Simplifier algébriquement la fonction suivante

    $$F2 = (X + Y + Z)(\bar{X} + Y + Z) + XY + YZ$$

  - **Question4 :** Soitla fonction booléenne suivante :

    $F(A, B, C, D) = A\bar{B}\bar{C}\bar{D} + \bar{A}B\bar{C}D + A\bar{B}CD + A\bar{B}\bar{C}D + \bar{A}BCD + \bar{A}BC\bar{D} + ABC\bar{D}$

    A. Simplifier F par la méthode des diagrammes de KARNAUGH, sachantque quatre combinaisons de variables sont impossibles :$AB\bar{C}D + ABCD + \bar{A}\bar{B}C\bar{D} + A\bar{B}C\bar{D}$

    B. Donner le schéma logique ou logigramme de la fonctionsimplifiée

WWW.DZPEN.COM | قلم الجزائر

CHAPITRE 3

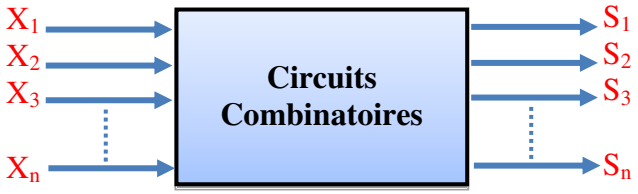
LES CIRCUITS COMBINATOIRES

WWW.DZPEN.COM | قلم الجزائر

WWW.DZPEN.COM | قلم الجزائر

La logique combinatoire concerne l'étude des fonctions dont la valeur de sortie ne dépend que de l'état logique des entrées se traduisant par une modification de la valeur des sorties et non pas non plus de ses états antérieurs : à chaque combinaison des variables d'entrée correspond toujours une seule combinaison des fonctions de sortie. Un circuit numérique réalisant une fonction combinatoire est un**circuit combinatoire**

Figure31: Circuits combinatoires



Outre les opérateurs élémentaires cités au chapitre 2, on distingue comme opérateurscombinatoires standard :

- ✓ Les opérateurs de transcodage,
- ✓ Les opérateurs d'aiguillage,
- ✓ Les opérateurs de comparaison,
- ✓ Les opérateurs arithmétiques.

3.1 Circuits de transcodage

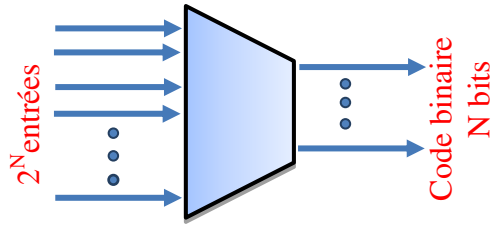
On désigne par circuit de **transcodage** un circuit qui traduit une information dans un codedonné (**codeur** ou **encodeur**) ou bien qui, au contraire, permet d'extraire une ou des informations àpartir d'un code donné (**décodeur**) ou bien encore réalise la conversion d'un code vers un autre(**convertisseur de code** ou **transcodeur**).

3.1.1 Codeur

Le codeur (ou encodeur) est un circuit logique qui possède 2^N voies entrées, dont une seule est activée et N voies de sorties,avec $\mathbf{2^{N-1}<N\leq 2^N}$,Il fournit en sortie le code binaire correspondant

Le principe de fonctionnement d'un codeur est le suivant : lorsqu'une entrée est activée, les sorties affichent la valeur correspondant au numéro de l'entrée dans le code binaire choisi. Une seule entrée du codeur doit normalement être activée à la fois. Dans le cas où le code en sortieest le code binaire pur,

Figure32: Schéma fonctionnel d'un codeur

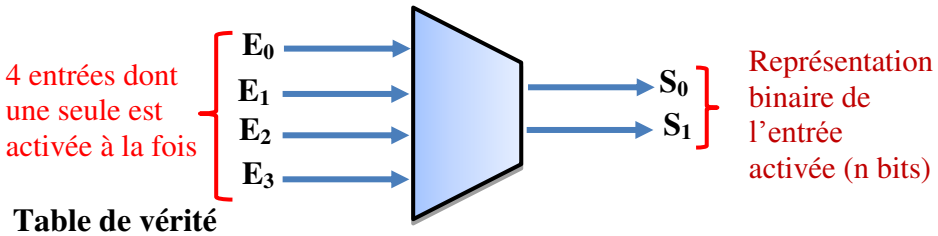


WWW.DZPEN.COM | قلم الجزائر

❖ **Exemple :** Codeur décimal vers binaire (4 entrées vers 2 sorties)

A. Schéma fonctionnel

Figure33: Schéma fonctionnel du codeur 4 voies d'entrées et 2 bits de sortie



B. Table de vérité

Entrées				Sorties	
Codage 1 parmi 2^N				Nombre binaire de N bits	
E_3	E_2	E_1	E_0	S_1	S_0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

A. Table de vérité réduite du codeur 4 vers 2

Entrée Activée (=1)	Sorties S_1	Sorties S_0
E_0	0	0
E_1	0	1
E_2	1	0
E_3	1	1

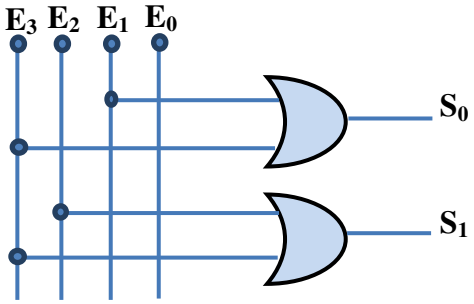
B. Equation des sorties

S_1=1 si (E_2=1) ou (E_3=1) ; $\mathbf{S_1=E_2+E_3}$

S_0=1 si (E_1=1) ou (E_3=1) ; $\mathbf{S_0=E_1+E_3}$

C. Logigramme

Figure34: Logigramme du codeur 4 vers 2

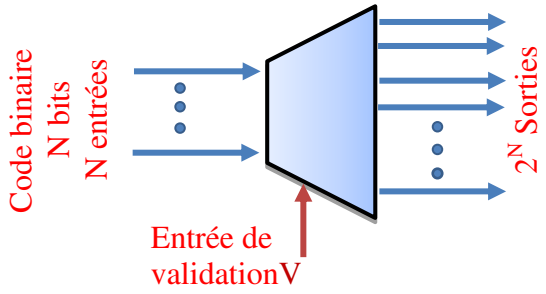


WWW.DZPEN.COM | قلم الجزائر

3.1.2 Décodeur C'est un circuit combinatoire qui est constitué de :

- ✓ **N** : entrées de données
- ✓ 2^N sorties
- ✓ Pour chaque combinaison en entrée une seule sortie estactive à la fois
- ✓ Une entrée de validation *V* (si l'entrée de validation *V* vaut 1, alors le décodage est autorisé ; dans le cas contraire (*V* = 0), les sorties sont inhibées et restent inactives)

Figure 35 :Schéma fonctionnel d'un décodeur

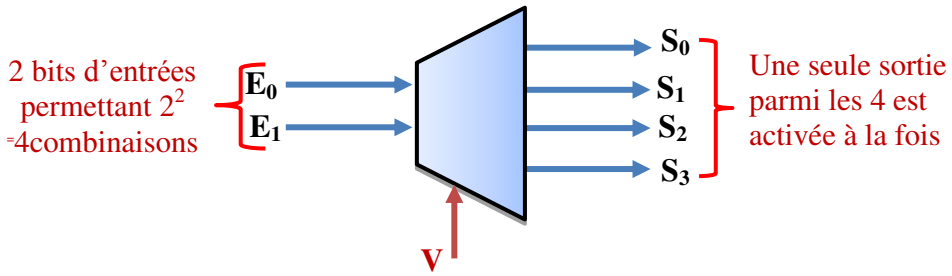


❖ **Exemple :**Décodeur 1 parmi 4

Pour pouvoir activer toutes les 4 voies on a besoin de 2 bits à l'entrée car c'est 2^2=4 ⟹2 entrées et 2^2 = 4 sorties (1 sortie parmi les 4 sera active) et une entrée de validation V

A. Schéma fonctionnel

Figure36: Schéma fonctionnel d'un décodeur 1 parmi 4



B. Table de vérité

Code binaire d'entrée			Codage 1 parmi 4 sorties			
V	E_1	E_0	S_3	S_2	S_1	S_0
0	X	X	0	0	0	0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0

WWW.DZPEN.COM | قلم الجزائر

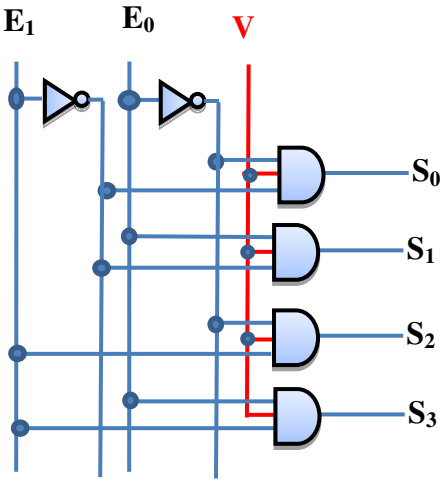
C. Equations de sorties

$S_0 = V(\bar{E}_0 . \bar{E}_1 .)$, $S_1 = V(E_0 . \bar{E}_1)$,

$S_2 = V(\bar{E}_0 . E_1)$, $S_3 = V(E_1 . E_0)$

D. Logigramme

Figure 37: Logigramme du décodeur 1 parmi 4



3.1.3 Convertisseurs

C'est un circuit combinatoire qui permet de transformer un codeX (sur n bit) en entrée en un code Y (sur m bit) en sortie.

Figure38: Schéma fonctionnel d'un convertisseur



WWW.DZPEN.COM | قلم الجزائر

❖ **Exemple 1 :**Transcodeur binaire Gray 2bits

Pour passer d'un code à un autre, on utilisera un convertisseur de code. A titre d'illustration nous allons étudier le transcodeur binaire Gray.

A. Schéma fonctionnel

Figure39: Schéma fonctionnel d'un convertisseur binaire Gray 2bits



B. Table de vérité

Entrée Code binaire		Sortie Code gray	
B_1	B_0	G_1	G_0
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

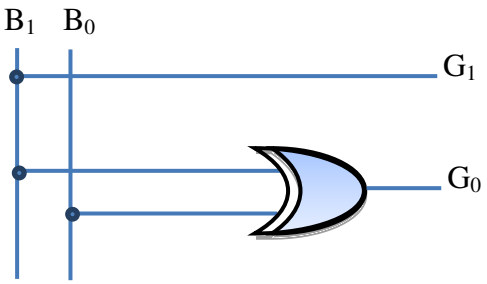
C. Equations de sorties

$$G_0 = \bar{B}_0 B_1 + B_0 \bar{B}_1 = B_0 \oplus B_1$$

$$G_0 = \bar{B}_0 B_1 + B_0 B_1 = B_1$$

D. Logigramme

Figure 40: Logigramme d'untranscodeur binaire Gray 2bits



3.1.4 Exercice 11 : Réaliser un convertisseur Binaire Naturel-Binaire Réfléchi

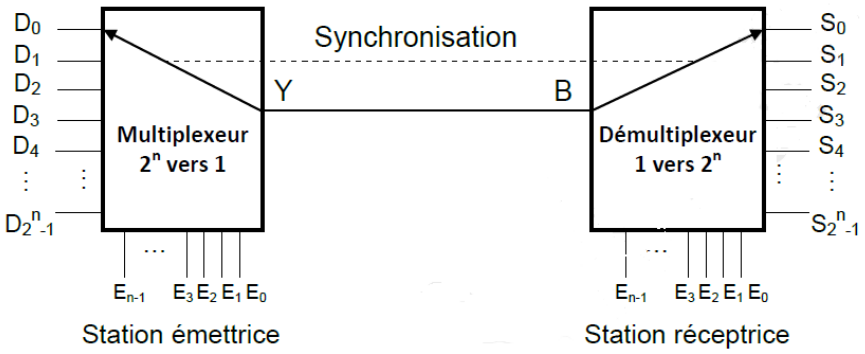
WWW.DZPEN.COM | قلم الجزائر

3.2 Circuits de multiplexage

La transmission des informations d'une station à une autre nécessite plusieurs lignes en parallèle, ce qui est difficile à réaliser et très couteux lorsque les stations sont géométriquement éloignées l'une de l'autre.

La solution est alors, transmettre en série sur une seule ligne, en utilisant à la station émettrice un convertisseur parallèle/série (Multiplexeur) et à la station réceptrice un convertisseur série/parallèle (Démultiplexeur).

Figure41: Station de transmission des informations



Support de cours : systèmes Logiques (1) Logique combinatoire
Institut Supérieur des Etudes Technologiques de Nabeul
Département de Génie Electrique)

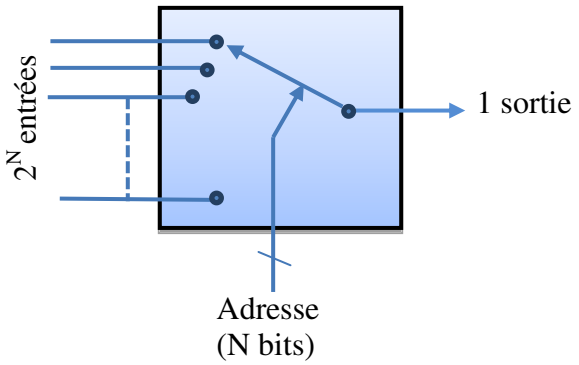
3.2.1 Multiplexeur (MUX)

Un multiplexeur est un circuit servant à concentrer sur une même voie de transmission différents types de liaisons (***informatique, télécopie, téléphonie, télétex***) en sélectionnant une entrée parmi N, c'est un circuit combinatoire qui permet desélectionner une information (1 bit) parmi 2^N valeurs en entrée.

Il possède :

- 2^N entrées d'information
- Une seule sortie
- N entrées de sélection (commandes ou adresses)

Figure42: Schéma fonctionnel d'un multiplexeur 2^N vers 1

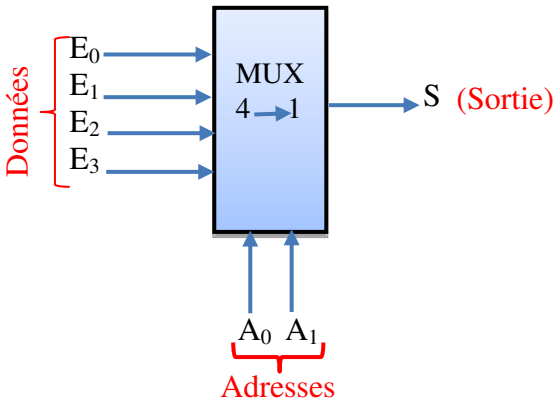


WWW.DZPEN.COM | قلم الجزائر

❖ **Exemple : Multiplexeur 4 vers 1**

A. Schéma fonctionnel

Figure43: Schéma fonctionnel d'un multiplexeur 4 vers 1



B. Table de vérité

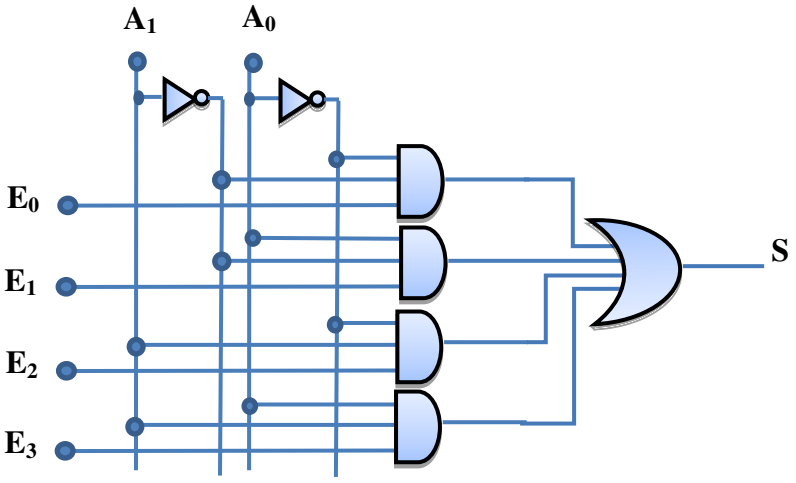
A_1	A_0	S
0	0	E_0
0	1	E_1
1	0	E_2
1	1	E_3

E. Equation de sortie

$S= \bar{A}_0\bar{A}_1(E_0) + A_0\bar{A}_1(E_1) + \bar{A}_0A_1(E_2) + A_0A_1(E_3)$

F. Logigramme

Figure44: Logigramme d'unmultiplexeur 4 vers 1



❖ **Exemples de circuits integres multiplexeurs** On trouve chez les constructeurs, les circuits multiplexeurs suivants :

- ✓ 2 vers 1 : 74157 (4 Mux 2 vers 1)
- ✓ 4 vers 1 : 74153 (2 Mux 4 vers 1)
- ✓ 8 vers 1 : 74151 (2 Sorties complémentaires), 74152 (1 Sortie complémentée)
- ✓ 16 vers 1 : 74150 (1 Sortie complémentée)

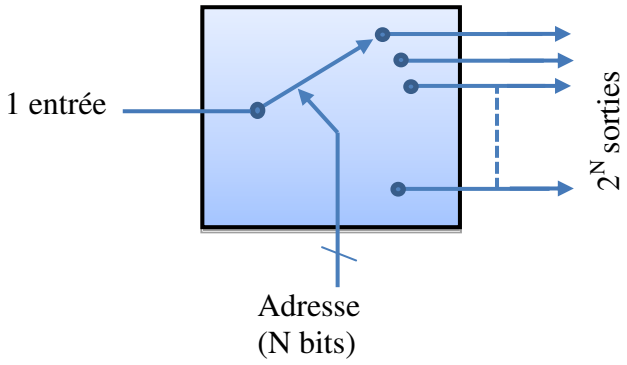
WWW.DZPEN.COM | قلم الجزائر

3.2.2 Démultiplexeur (DEMUX)

Il joue le rôle inverse d'un multiplexeur, il permet de fairepasser une information dans l'une des sorties selon les valeursdes entrées de commandes.Il possède :

- Une seule entrée
- 2^N sorties
- **N** entrées de sélection (commandes ou adresses)

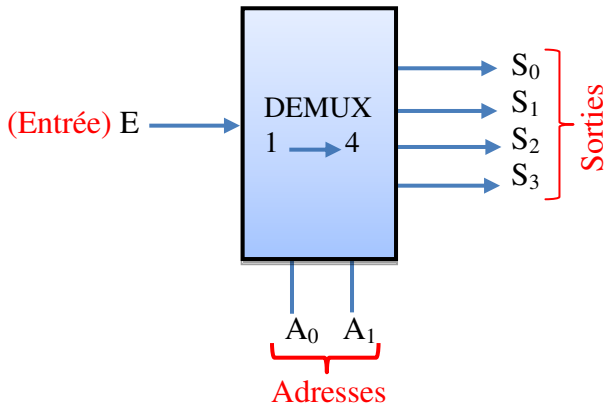
Figure45: Schéma fonctionnel d'un démultiplexeur 1 vers 2^N



❖ **Exemple : démultiplexeur 1 vers 4**

A. Schéma fonctionnel

Figure46: Schéma fonctionnel d'un démultiplexeur 1 vers 4



WWW.DZPEN.COM | قلم الجزائر

B. Table de vérité

A_1	A_0	S_3	S_2	S_1	S_0
0	0	0	0	0	E
0	1	0	0	E	0
1	0	0	E	0	0
1	1	E	0	0	0

C. Equations de sorties

$$S_0 = \overline{A_0}\ \overline{A_1}\ (E)$$

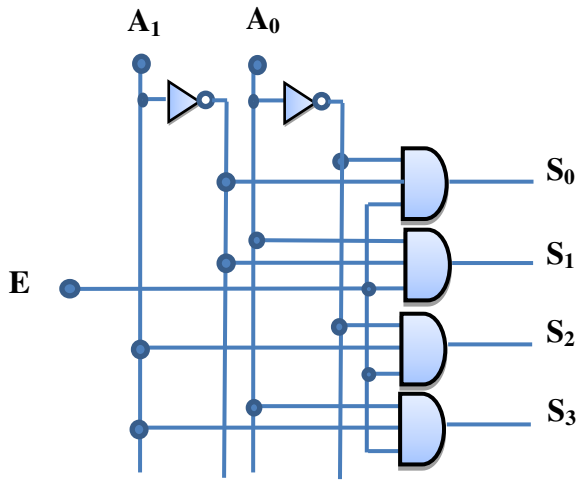
$$S_1 = A_0\ \overline{A_1}\ (E)$$

$$S_2 = \overline{A_0}\ A_1\ (E)$$

$$S_0 = A_0\ A_1\ (E)$$

D. Logigramme

Figure 47: Logigramme d'unmultiplexeur 4 vers 1



❖ **Exemples de circuits integres démultiplexeur :**On trouve chez les constructeurs, des circuits Démultiplexeurs :

- ✓ 1 vers 4 : 74139 (2 DEMUX 1 vers 4, Sorties complémentées)
- ✓ 1 vers 8 : 74137, 74138 (Sorties complémentées)
- ✓ 1 vers 16 : 74154,74159 (Sorties complémentées)

3.2.3 Exercice 12

A. Donner le synoptique et la TDV d'un MUX 2 vers 1
B. Donner le synoptique et la TDV d'un MUX 8 vers 1
C. Réaliser le MUX 8 vers 1 a l'aide de MUX 2 vers 1 (Synoptique)

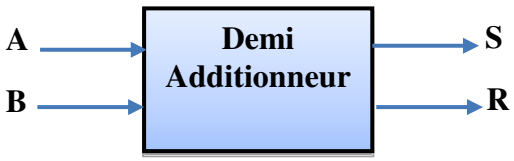
WWW.DZPEN.COM | قلم الجزائر

3.3 Circuits de calcul

3.3.1 Demi-Additionneur

Le demi additionneur est un circuit combinatoire qui permet de réaliser la somme arithmétique de deux nombres **A** et **B** sur un bit, a la sotie on va avoir la somme **S** et la retenu **R** (***Carry***)

Figure48: Schéma fonctionnel d'un demi-additionneur



A. Table de vérité

En binaire l'addition sur un seulbit se fait de la manière suivante

0 + 0 = 0 retenue 0
0 + 1 = 1 retenue 0
1 + 0 = 1 retenue 0
1 + 1 = 0 retenue 1

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

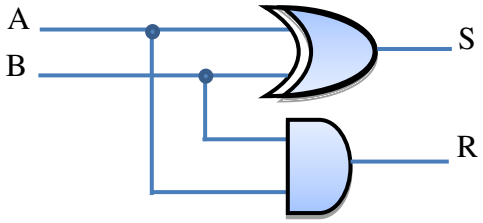
B. Equations de sorties

$S = \overline{A}B + A\overline{B} = A \oplus B$

$R = A\ B$

C. Logigramme

Figure 49: Logigramme d'undemi-additionneur 1 bit



3.3.2 Additionneur complet 1 bit

L'additionneur complet un bit possède 3 entrées :

- $\mathbf{a_i}$: le premier nombre sur un bit.
- **bi** : le deuxième nombre sur un bit.
- $\mathbf{r_{i-1}}$: le retenue entrante sur un bit.

Il possède deux sorties :

- **Si** : la somme
- **Ri** la retenue sortante

Figure50: Schéma fonctionnel d'un additionneur complet 1 bit



WWW.DZPEN.COM | قلم الجزائر

A. Table de vérité

a_i	b_i	r_{i-1}	S_i	R_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

B. Equations de sorties

$$S_i = \bar{a}_i \bar{b}_i r_{i-1} + \bar{a}_i b_i \bar{r}_{i-1} + a_i \bar{b}_i \bar{r}_{i-1} + a_i b_i r_{i-1}$$

$$S_i = \bar{a}_i (\bar{b}_i r_{i-1} + b_i \bar{r}_{i-1}) + a_i (\bar{b}_i \bar{r}_{i-1} + b_i r_{i-1})$$

$$S_i = \bar{a}_i (b_i \oplus r_{i-1}) + a_i \overline{(b_i \oplus r_{i-1})}$$

$$\mathbf{S_i = a_i \oplus b_i \oplus r_{i-1}}$$

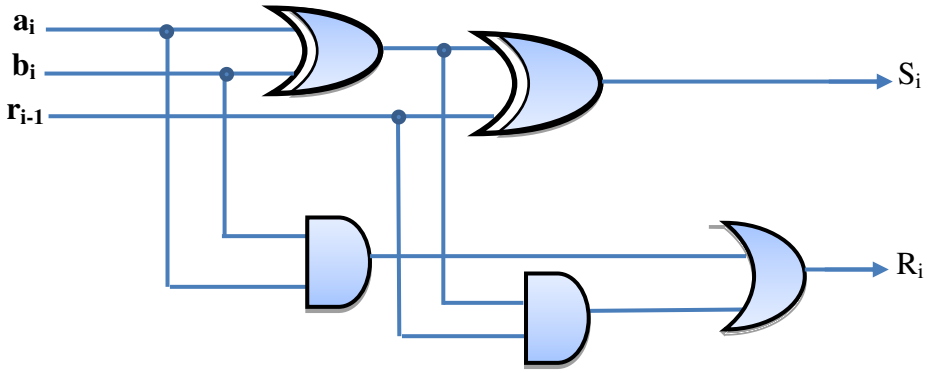
$$R_i = \bar{a}_i b_i r_{i-1} + a_i \bar{b}_i r_{i-1} + a_i b_i \bar{r}_{i-1} + a_i b_i r_{i-1}$$

$$R_i = r_{i-1} (\bar{a}_i b_i + a_i \bar{b}_i) + a_i b_i (\bar{r}_{i-1} + r_{i-1})$$

$$\mathbf{R_i = r_{i-1} (a_i \oplus b_i) + a_i b_i}$$

C. Logigramme

Figure 51: Logigramme d'un additionneur complet 1 bit



3.3.3 Comparateur 1 bit

C'est un circuit combinatoire qui permet de comparer entredeux nombres binaire A et B.

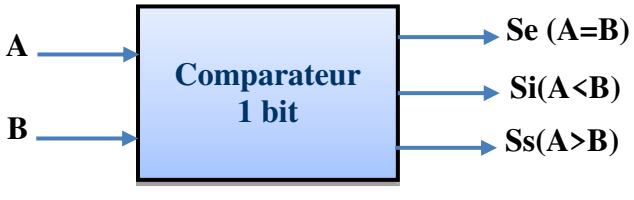
Il possède 2 entrées :

- **A** : sur un bit
- **B** : sur un bit

Il possède 3 sorties

- **Se** : égalité (A=B)
- **Si** : inférieur (A< B)
- **Ss** : supérieur (A > B)

Figure52: Schéma fonctionnel d'un comparateur 1 bit



WWW.DZPEN.COM | قلم الجزائر

A. Table de vérité

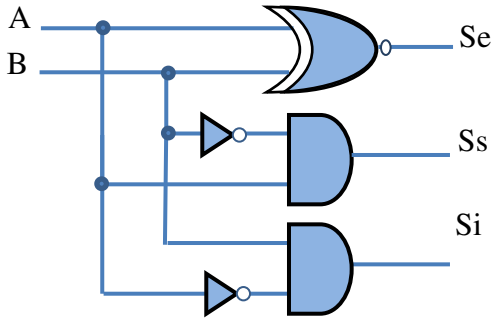
A	B	Ss	Se	Si
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0

B. Equations de sorties

Ss= A.$\overline{B}$
Se= $\overline{A}\overline{B}$ + A B =$\overline{A \oplus B}$
Si= $\overline{A}$.B

C. Logigramme

Figure 53: Logigramme d'un comparateur 1 bit



3.3.4 Exercice 13 Donner le schéma fonctionnel d'un additionneur complet 4 bits

3.3.5 Exercice 14 Donner le circuit logique d'un comparateur qui permet de faire la comparaison entre deux nombres A (A_1A_0) et B (B_1B_0) chacun sur deux bits

WWW.DZPEN.COM | قلم الجزائر

3.4 Résumé

On résume ce chapitre comme suite

- ✓ Dans un circuit combinatoire, chaque combinaison des variables d'entrée correspond toujours une seule combinaison des fonctions de sortie
- ✓ Un codeur est un circuit combinatoire qui possède $\mathbf{2^N}$ voies entrées, dont une seule est activée et **N** voies de sorties, avec $\mathbf{2^{N-1}<N\leq 2^N}$
- ✓ Un décodeur est un circuit combinatoire qui possède **N** entrées de données, $\mathbf{2^N}$ sorties, ou une seule sortie est active à la fois et une entrée de validation ***V*** (***V=0***le décodeur et bloqué, ***V=1***le décodage est autorisé)
- ✓ Un Convertisseurs est un circuit combinatoire qui permet de transformer un code **X** (sur **N** bit) en entrée en un code **Y** (sur **M** bit) en sortie
- ✓ Un multiplexeur est un circuit combinatoire qui permet de sélectionner une information (1 bit) parmi 2^N valeurs en entrée. Il possède 2^N entrées d'information, une seule sortie et N entrées d'adresses
- ✓ Un Démultiplexeur est l'inverse du multiplexer, il possède : Une seule entrée de donnée, $\mathbf{2^N}$ sorties et **N** entrées d'adresses
- ✓ Un additionneur complet un bit possède 3 entrées :$\mathbf{A_i}$: le premier nombre sur **N**bit,**Bi**le deuxième nombre sur **N** bit et $\mathbf{R_{i-1}}$ la retenue entrante sur un bit.Il possède aussi2 sorties :**Si** la somme et **Ri** la retenue sortante
- ✓ Un comparateur binaire est un circuit logique qui effectue la comparaison entre 2 nombres binaires généralement notés A et B.Il possède 3 sorties notées **A = B**,**A > B** et **A < B**

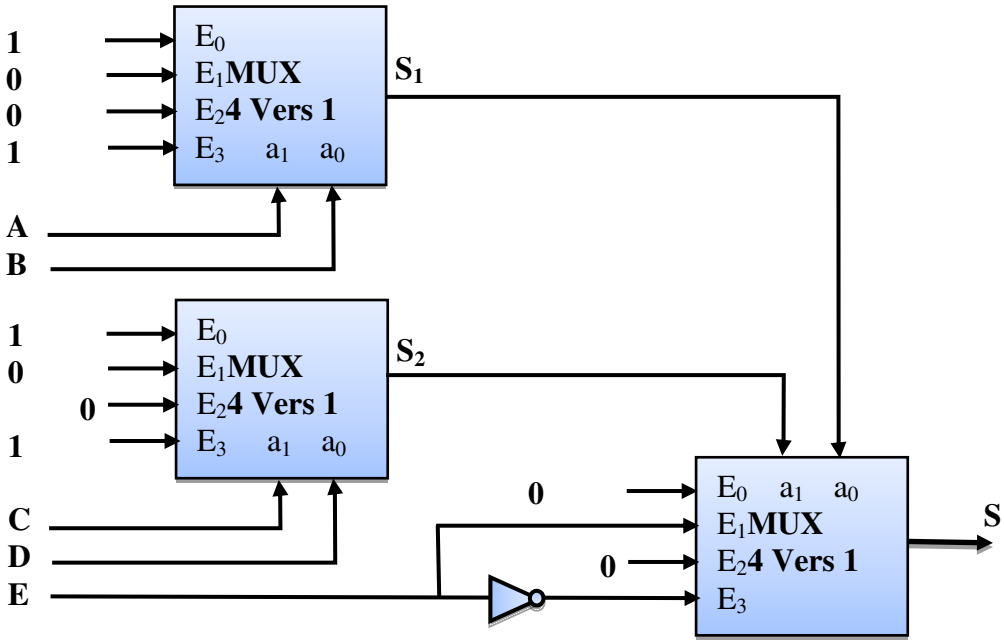
WWW.DZPEN.COM | قلم الجزائر

3.5 Exercice de synthèse 3

- **Spécialité** : Domotique
- **Module** : Logique combinatoire et logique séquentielle
- **Durée :** 1h30
- **But :**
  - ✓ Etudier les principaux circuits logiques combinatoires utilisés dans les systèmes numériques (tels que : les circuits arithmétiques, les codeurs et les transcodeurs),
  - ✓ Réaliser des fonctions logiques en utilisant les circuits combinatoires.
- **Matériel requis** : Documents, support de cours et logiciel d'électronique
- **Mise en situation :**Trouver l'équation logique réalisé par les multiplexeurs.
- **Marche à suivre :**

Donner l'équation logique de la fonction S du montage suivant :

Figure 54: Montage de la fonction S



WWW.DZPEN.COM | قلم الجزائر

CHAPITRE 4

LES CIRCUITS SEQUENTIELS

WWW.DZPEN.COM | قلم الجزائر

Les circuits logiques combinatoires ne suffisent pas à eux seuls à la manipulation de l'informationcomme cela se fait dans les systèmes numériques modernes. Le caractère figé des circuitscombinatoires que traduit la correspondance stricte entre les entrées et les sorties limiteconsidérablement le champ de leurs applications. C'est là que les circuits séquentiels prennent toute leurimportance.

Dans les systèmes numériques modernes (appareil de mesure, ordinateur…) La***mémorisation*** des informations est une opération indispensable dans tout système de traitement, on veut on outre ***Compter*** (des impulsions), ***décaler*** d'un ou plusieurs pas (une suite de niveaux **1** et **0**),***recopier***au rythme d'une horloge (des niveaux **1** et **0**), ***transmettre*** en série sur un fil des information, ***stockées*** (en parallèle)d'une façon permanente dans une mémoire, d'où la nécessite des circuits séquentiels.

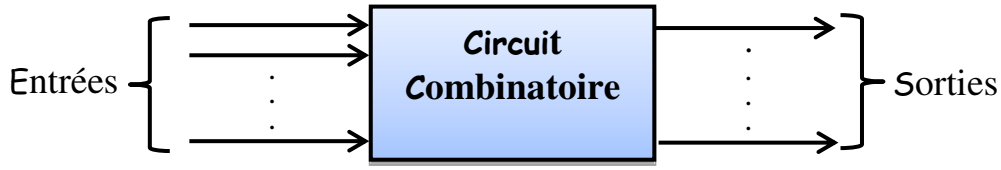
4.1 Circuits combinatoires et circuits séquentiels

4.1.1 Circuits combinatoires

Dans les opérateurs logiques examinés précédemment l'état de sortie ne dépend que de l'état présent des entrées, c'est à dire que pour chaque état d'une combinaison d'entrée correspond **une valeur toujours la même** à la sortie (Circuits combinatoires).

En logique combinatoire, à chaque combinaison d'entrée correspond une seule possibilité de combinaison sur les sorties.

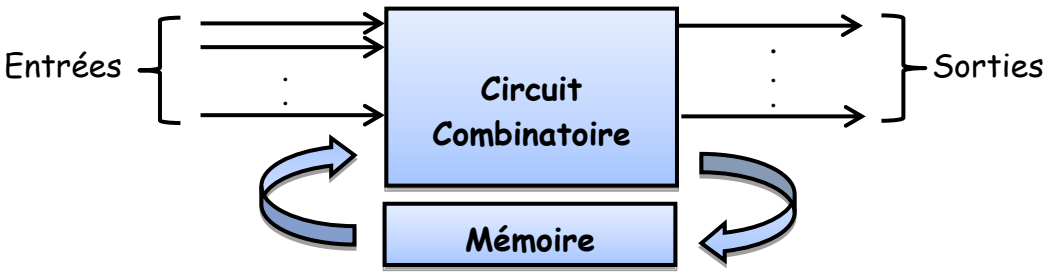
Figure55: Circuit combinatoire



4.1.2 Circuits séquentiels

Pour certains opérateurs, l'état de la sortie dépend non seulement de la combinaison appliquée à l'entrée (logique combinatoire) mais aussi de l'état ***précédent*** des sorties du circuit : ils sont ditsséquentiels et ont un effet « mémoire ». La logique séquentielle est donc une logique combinatoire avec une mémorisation des sorties. Cette mémorisation est réalisée par ce qu'on appelle une bascule ; c'est un organe de mémorisation unitaire (mémorisation d'un seul bit).

Figure56: Circuit séquentiel



WWW.DZPEN.COM | قلم الجزائر

4.1.3 Fonctionnement des circuits séquentiels

Nous aurons deux grands types de fonctionnement, soit la valeur des entrées est significative à tout instant (lecture de l'entrée à toute instant), on parle alors de circuits asynchrones, soit elle n'est prise en compte qu'à des instants précis synchronisés par une fréquence d'horloge, on parle alors de circuits synchrones.

Figure57: Circuit séquentiel **Asynchrone**

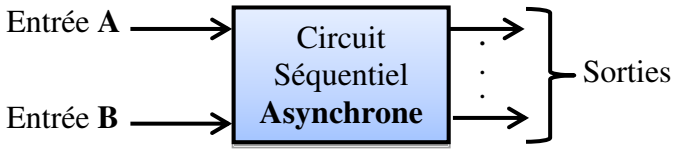
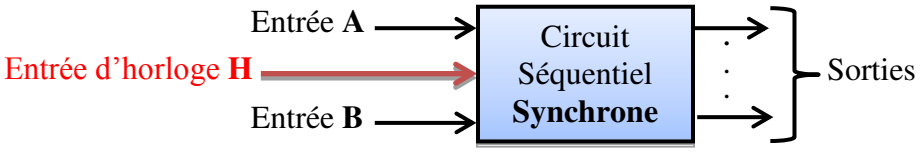
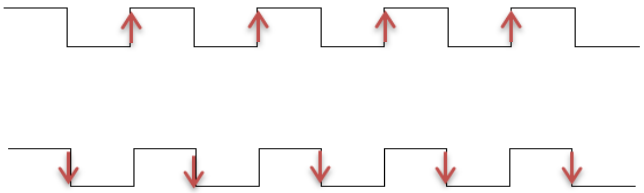


Figure58: Circuit séquentiel **Synchrone**



Le signal d'horloge est généralement un signal carré, il est habituellement distribué à tous les étages du système, de sorte que la plupart des sorties changent d'état seulement quand le signal d'horloge effectue une transition.
Ces transitions, appelées fronts, sont identifiées sur un chronogramme au moyen d'une flèche dans le sens de variation du signal :

Figure59: front montant et front descendant d'un signal d'horloge



Le front montant ou transition positive caractérise un signal passant de 0 à 1

Le front descendant ou transition négative caractérise un signal passant de 1 à 0.

4.1.4 Exercice 15 Répondre par vrai ou faux

- Un signal d'horloge est généralement un signal sinusoïdal
- Un front descendant caractérise un signal passant de 0 à 1
- l'état de la sortie d'un circuit séquentiel dépend non seulement de la combinaison appliquée à ces l'entrées mais aussi de l'état précédent de ces sorties
- La logique séquentielle est une logique combinatoire avec une mémorisation des sorties
- Un circuit est dit synchrone si la lecture de ces entrées se fait à tout instant
- Dans un circuit l'état de sortie ne dépend que de l'état présent des entrées
- Une bascule est un organe de mémorisation unitaire

WWW.DZPEN.COM | قلم الجزائر

4.2 Les bascules

Ce sont des circuits ayant deux états stables et dont la sortie ne dépend pas seulement de l'état des entrées à un instant donnée mais aussi de l'état antérieur, ils sont donc pas combinatoires mais séquentiels.

Les bascules sont des mémoires élémentaires qui ne peuvent mémoriser qu'un seul bit.

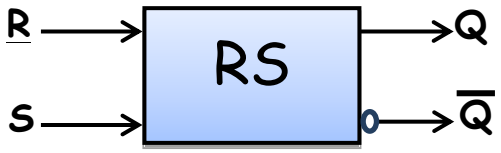
Exemple d'utilisation :

- Appel d'un ascenseur ; on appuie sur le bouton, l'appel est enregistré et le voyant s'allume.Si on relâche le bouton, le voyant reste allumé, il y a donc mémorisation.
- Le piéton qui appuie sur le bouton des feux tricolores pour traverser la route : l'informationest mémorisée même si le piéton relâche le bouton.
- La commande « coupure du son » du téléviseur, etc.

4.2.1 Bascule RS (Bistable RS Asynchrone)

A. Synoptique

Figure60: Synoptique d'une Bascule RS Asynchrone



B. Table de vérité

R	S	Q_{n-1}	Qn	
0	0	0	**0**	Mémorisation
0	0	1	**1**	$Qn=Q_{n-1}$
0	1	0	**1**	Mise à 1
0	1	1	**1**	Qn=1
1	0	0	**0**	Mise à 0
1	0	1	**0**	Qn=0
1	1	0	**?**	Etat métastable
1	1	1	**?**	Cas indéterminé

R	S	Qn	
0	0	**0**	Mémorisation($Qn=Q_{n-1}$)
0	1	**1**	Mise à 1
1	0	**0**	Mise à 0
1	1	**?**	Cas indéterminé

S : « SET » mise à 1

R : « RESET » remise à 0

R=S=0 : Conservation de l'état précèdent Q_n(mémorisation)

R=S=1 : Cas indéterminé

Remarque : la bascule RS est une bascule qui peut mémoriser qu'un seul bit.

WWW.DZPEN.COM | قلم الجزائر

C. Tableau de KARNAUGH

Q_{n-1} \ RS	00	01	11	10
0	0	1	Ø	0
1	1	1	Ø	0

Q_n

$$Q_n = \overline{R}.Q_{n-1} + S$$

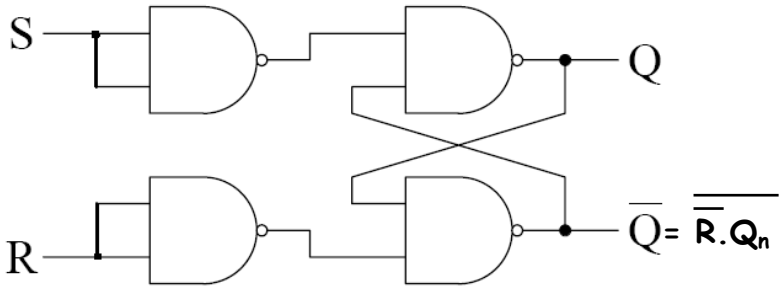
D. Réalisation de la bascule RS à l'aide des portes NAND

$Q_n = \overline{R}.Q_{n-1} + S = \overline{\overline{R}.Q_{n-1}+S} = \overline{R}.\overline{Q}_{n-1}.S$ (Q_{n-1} c'est Q_n à l'instant t-1)

donc

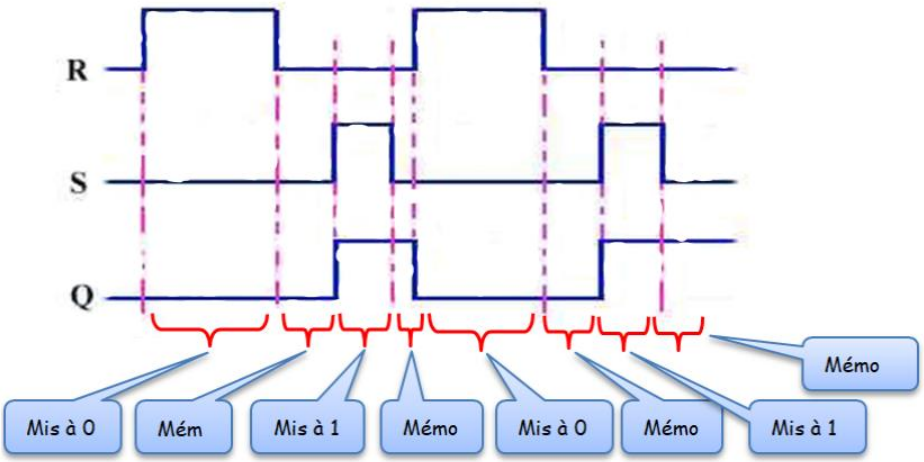
$$Q_n = \overline{\overline{R}.Q_n}.\overline{S}$$

Figure61: Schémas logique d'une bascule **R**S asynchrone



E. Exemple :Chronogramme d'une bascule RS

Figure 62 : Chronogramme d'une bascule **RS** asynchrone

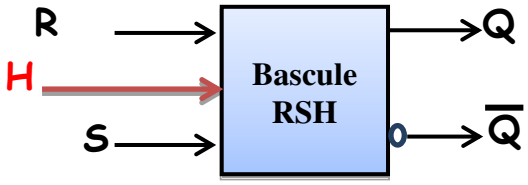


WWW.DZPEN.COM | قلم الجزائر

4.2.2 Bascule RS Synchrone 'RSH'(Bistable RS synchrone)

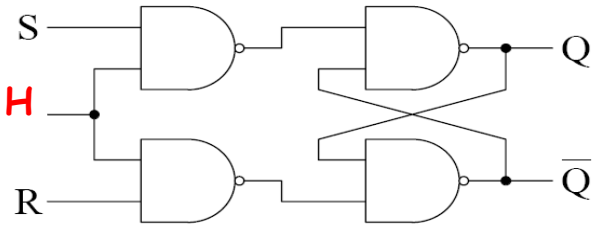
La bascule R.S.H. est une bascule pour laquelle les entrées S et R ne sont prises en compte qu'en coïncidence avec un signal de commande. Ce signal peut être fourni par une horloge, nous avons alors une bascule synchrone. Voir synoptique figure 63

Figure63: Synoptique d'une bascule **R**S synchrone



Lorsque le signal de commande, noté ici **H**, est à **1** la bascule fonctionne comme indiqué précédemment et les sorties suivent les variations des entrées **S** et **R**. Par contre, lorsque le signal de commande est à **0**, la bascule est **bloquée** : Q est indépendant des éventuels changements de **S** et **R**. L'état mémorisé correspond au dernier état avant le passage de la ligne de commande de **1** à **0.**

Figure64: Schémas logique d'une bascule RS synchrone



A. Tablede vérité

H	R	S	Qn
0	0	0	Qn_{-1}
0	0	1	Qn_{-1}
0	1	0	Qn_{-1}
0	1	1	Qn_{-1}
1	0	0	Qn_{-1}
1	0	1	**1**
1	1	0	**0**
1	1	1	**?**

Mémorisation
Mise à 1
Mise à 0
Cas indéterminé

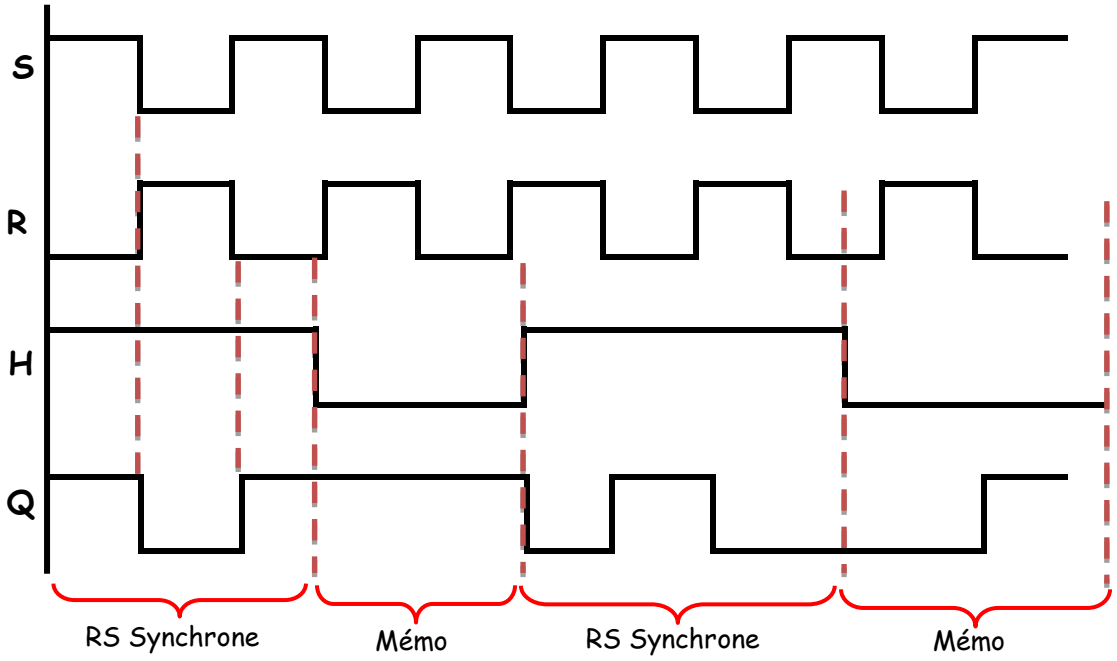
- Si **H=0** (Conservation de l'état précèdent) On peut dire que la bascule est bloquée (Mémorisation)
- Si **H=1,** le RS Synchrone se comporte comme une bascule RS asynchrone

WWW.DZPEN.COM | قلم الجزائر

B. Fonctionnement dynamique

En appliquant des impulsions à **R**, **S** et **H** on obtient le chronogramme suivant :

Figure65: Chronogramme d'une bascule RS synchrone



C. Les avantages des entrées synchronisées

- Préparer les commandes R et S sans perturber la mémoire (H=0) et de ne les valider qu'au moment souhaité (H=0).
- Protéger la mémoire des parasites survenant en R et S tant que l'on à H=0 (on donne à H=1 la durée minimale).

4.2.3 Bascule D ou LATCH (Verrou) (Bistable D)

La bascule de type **D** ou **Latch** est dérivée de la bascule **R,S,H** . Elle possède quant à elle, une seul entrée « **D** » pour positionner les sorties. En effet, on place un inverseur entre l'entrée **S** et **R** de la bascule **R,S,H.**
L'entrée **S** devient l'entrée **D** de la bascule de type **D** dont le schéma et le suivant.

Figure66: Synoptique d'une bascule **D**

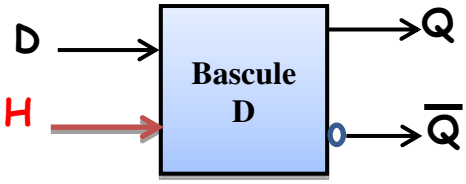
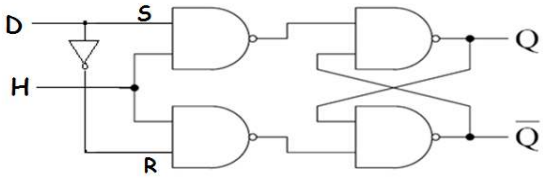


Figure67: Schémas logique d'une bascule D



WWW.DZPEN.COM | قلم الجزائر

A. Table de vérité

H	D	Q_{n-1}	Qn	
0	0	0	0	Mémorisation $Qn=Q_{n-1}$
0	0	1	1	
0	1	0	0	
0	1	1	1	
1	0	0	0	Recopie D Qn= D
1	0	1	0	
1	1	0	1	
1	1	1	1	

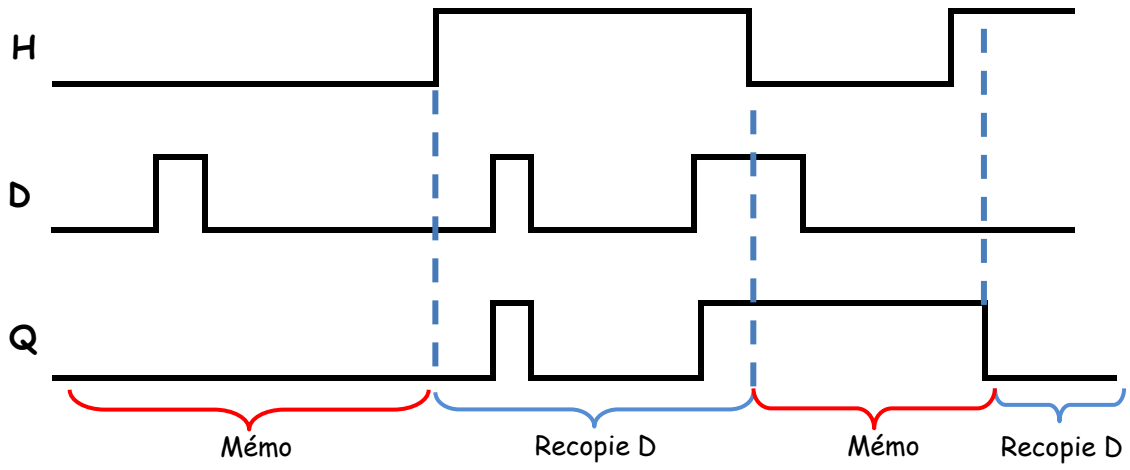
H	Qn
0	Q_{n-1}
1	D

Remarque : On remarque, d'après la table de vérité que le bistable D joue le rôle de mémoire élémentaire : le signale d'horloge représentant le signale de validation pour la donnée entrant dans le bistable

D. Fonctionnement dynamique d'une bascule D

En appliquant des impulsions à **D** et **H** on obtient le chronogramme suivant :

Figure68: Chronogramme d'une bascule D



WWW.DZPEN.COM | قلم الجزائر

4.2.4 Bascule JK Asynchrone

La bascule Bistable JK est similaire à la bascule RS, la seul différence est que la bascule JK permet d'éliminer l'indétermination (R=S=1) en ramenant les sorties Q et $\overline{Q}$ aux entrées, ce qui vas forcer la sortie à passer à son état complémentaire ($Q_n = \overline{Q}_{n-1}$). La seule modification et que les sorties du circuit logique sont reliées aux entrées.

Figure69: Synoptique d'une bascule **JK**

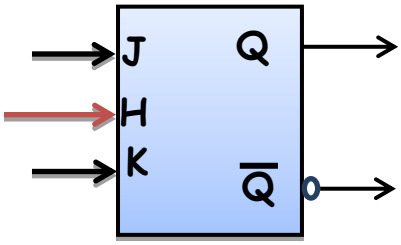
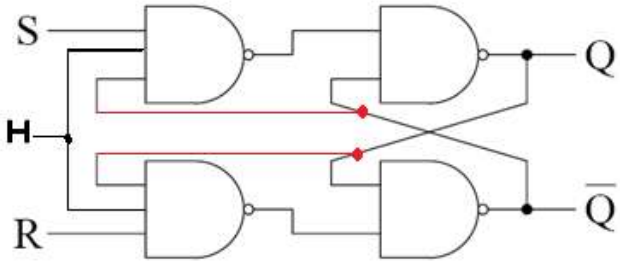


Figure70: Schémas logique d'une bascule **JK**



J : mise à 1
K : « RESET » remise à 0
J=K=0 : Conservation de l'état précèdent Q_n(mémorisation)
J=K=1 : Basculement de l'état précèdent(1 ⟶0 ou 0 ⟶1)

A. Table de vérité

J	K	Q_{n-1}	Qn	
0	0	0	**0**	Mémorisation
0	0	1	**1**	Qn=Q_{n-1}
0	1	0	**0**	Mise à 0
0	1	1	**0**	Qn=0
1	0	0	**1**	Mise à 1
1	0	1	**1**	Qn=1
1	1	0	**1**	Basculement
1	1	1	**0**	Qn=$\overline{Q}_{n-1}$

J	K	Qn	
0	0	Q_{n-1}	Mémorisation (Qn=Q_{n-1})
0	1	**0**	Mise à 1
1	0	**1**	Mise à 0
1	1	$\overline{Q}_{n-1}$	Basculement

H=0 Mémorisation
H=1 Voire Tdv Bascule JK

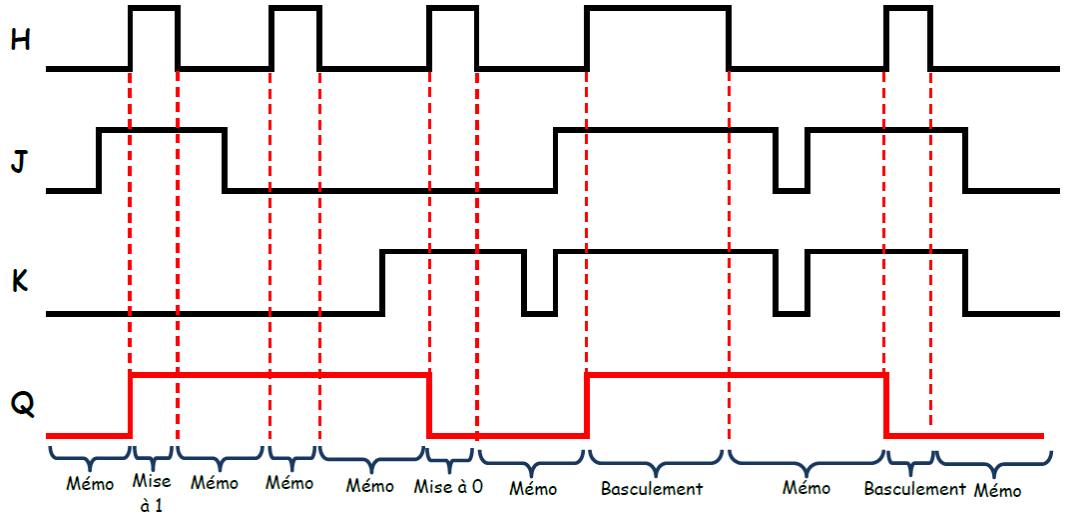
Remarque : La bascule JK est la plus complète, offrant tous les modes de fonctionnement que l'on peut demander à une bascule.

WWW.DZPEN.COM | قلم الجزائر

B. Fonctionnement dynamique d'une bascule JK

En appliquant des impulsions à **J**, **K** et **H** on obtient le chronogramme suivant :

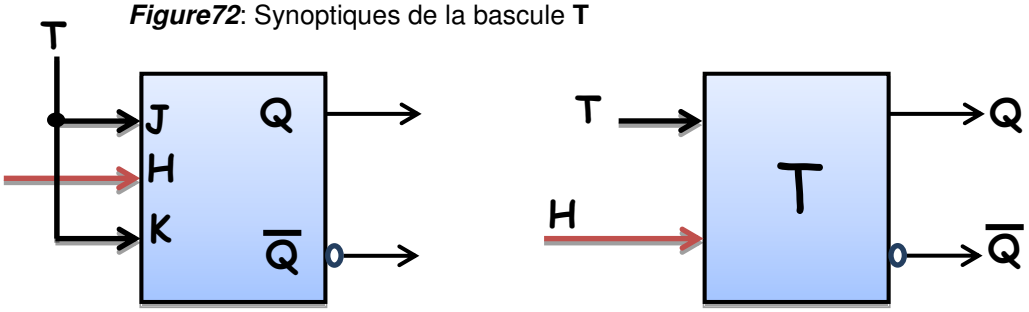
FIG 71 : Chronogramme d'une bascule **JK**



4.2.5 Bascule T

La bascule **T** (Transit ou Toggle) comprend deux entrées « **T** » et « **H** ». L'entrée **T** communique les mêmes niveaux au entrées **J** et **K**, donc **deux états** seulement sont possibles : le **Maintien** ou la **Complémentation**, la bascule **T** est connue spécialement pour son état de complémentation qui donne au circuit la fonction de **diviseur de fréquence (d'horloge) par deux**.

Figure72: Synoptiques de la bascule **T**



A. Table de vérité

T	H	Q_{n-1}	Qn	
1	0	0	0	Mémorisation $Qn=Q_{n-1}$
1	0	1	1	
1	1	0	1	Basculement $Qn=\overline{Q_{n-1}}$
1	1	1	0	

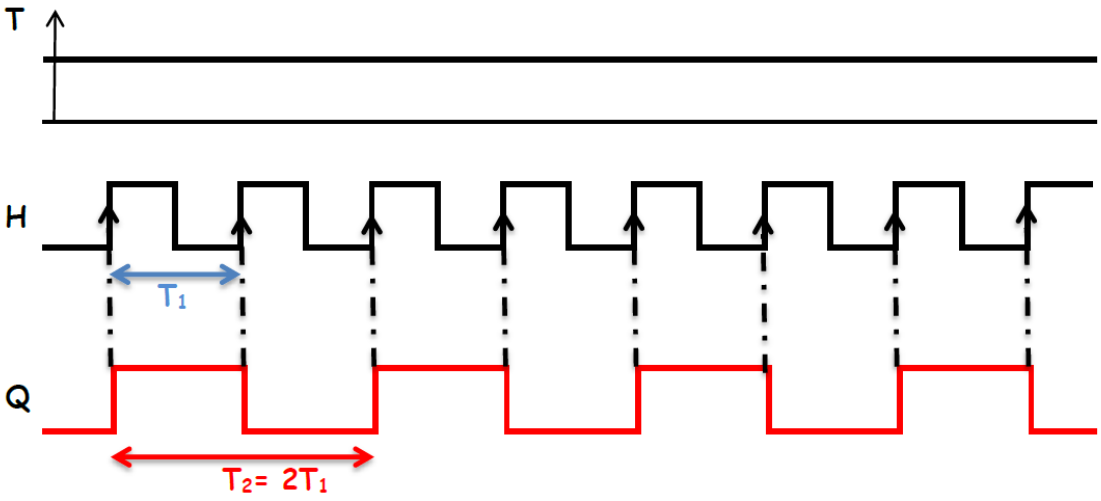
T	H	Qn
1	0	Q_{n-1}
1	1	$\overline{Q_{n-1}}$

Remarque : La bascule T change d'état à chaque impulsion d'horloge

WWW.DZPEN.COM | قلم الجزائر

B. Exemple de chronogramme d'une bascule T

FIG 73 : Chronogramme d'une bascule **T**



Remarque : On remarque que si on donne à T le niveau logique 1, la période T_2 du signale de sortie est égale à deux fois la période T_1 du signale d'horloge, donc la bascule T est un diviseur de fréquence par deux

4.2.6 Déclenchement d'une bascule

Une bascule asynchrone est déclenchée lorsque les signaux d'entrées changent. Dans le cas d'une bascule synchrone le déclenchement est plus compliqué, une bascule synchrone est pilotée par une horloge, de ce fait son déclenchement est provoquer par des impulsions. Donc on distingue deux types de bascules synchrones :

- Les bascules ***Latchs***
- Les bascules ***Flip_Flop***

A. Les bascules Latchs : Les bascules latchs réagissent sur un niveau d'horloge, cela revient à dire que ce type de bascule est déclencher quand l'horloge H=1 (***niveau haut***) ou quand H=0 (***niveau bas***)

FIG 74 : Synoptique d'une bascule D Laths déclenchable quand ***H=1***

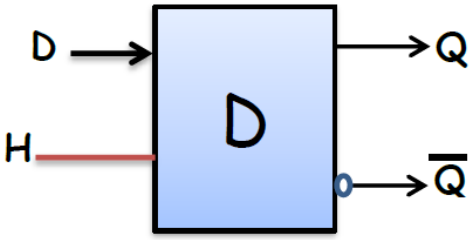
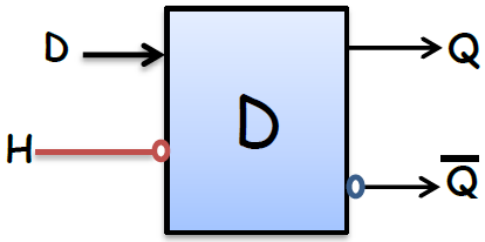


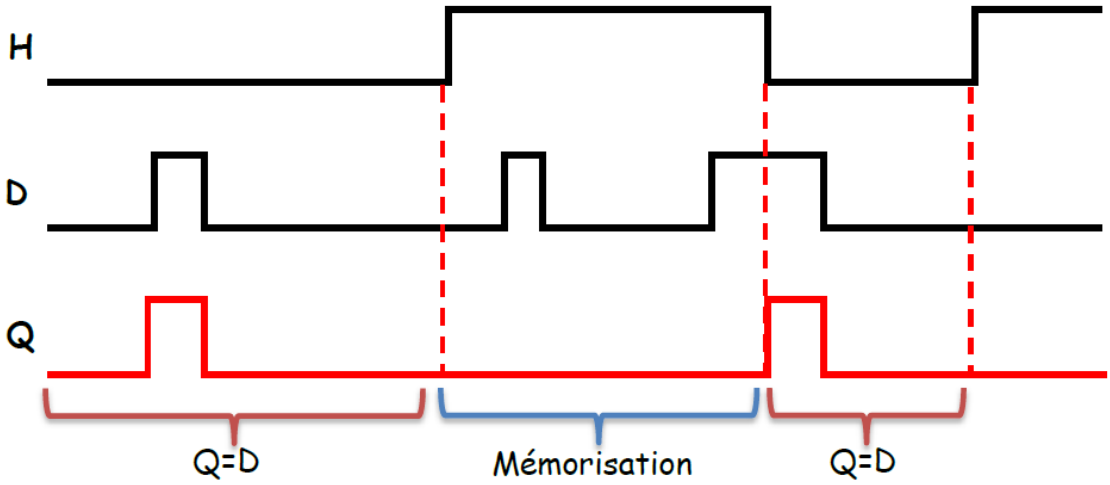
FIG 75 : Synoptique d'une bascule D Laths déclenchable quand ***H=0***



WWW.DZPEN.COM | قلم الجزائر

Exemple : Chronogramme d'une bascule D Latchsdéclenchable par un niveau bas

FIG 76 : Chronogramme d'une bascule **D**



B. Les bascules Flip_Flop :

Ce type de bascule change d'état non pas quand l'horloge est au niveau 1 ou 0 , mais pondant la ***transition*** du signale d'horloge de l'état 0 à 1 (***Front montant***) ou l'ors de la transition de l'état 1 à 0 (***Front descendent***).

Figure77: les fronts d'un signal d'horloge



a) Front montantdu signale d'horloge

b) Front descendentdu signale d'horloge

❖ Synoptiqued'une bascule D Flip-Flo

FIG 78 : Synoptique d'une bascule D Flip-Flop déclenchable Front ***montant***

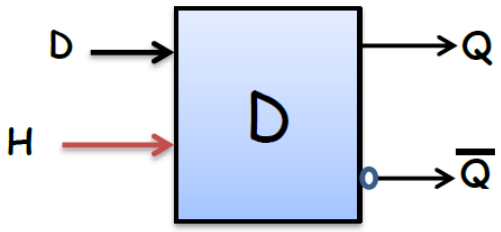
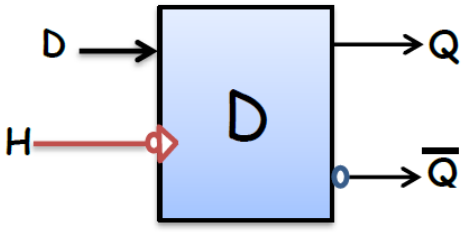


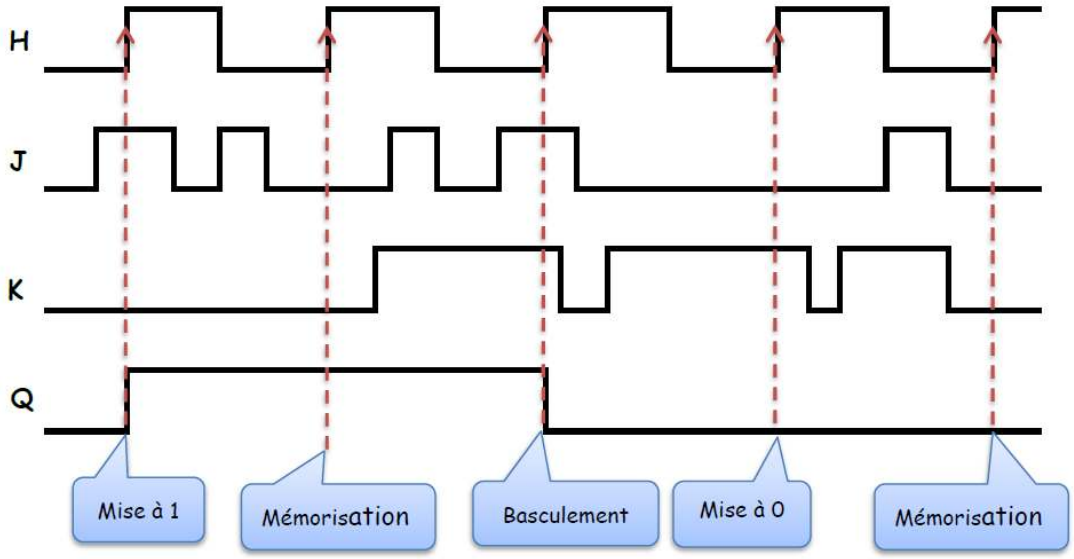
FIG 79 : Synoptique d'une bascule D Flip-Flop déclenchable Front ***descendent***



WWW.DZPEN.COM | قلم الجزائر

❖ **Exemple :** Chronogramme d'une bascule **JK** Flip-Flop déclenchableFront montant

FIG 80 : Chronogramme d'une **bascule D Flip-Flop**



4.2.7 Exercice 16 Donner les chronogrammes de Q1 et Q2 (On suppose qu'au début Q1 =0 et Q2=0)

Figure81: Bascule D et JK en cascade

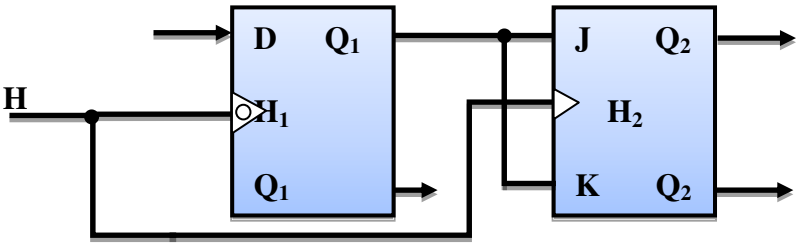


Figure82: Chronogrammesdes bascule D et JK en cascade



WWW.DZPEN.COM | قلم الجزائر

4.3 Les compteurs

Un compteur est une succession de bascule JK ou T dans le mot binaire se modifie à chaque impulsion d'horloge, c'est-à-dire qu'il compte le nombre d'impulsion appliquer à son entrer d'horloge.

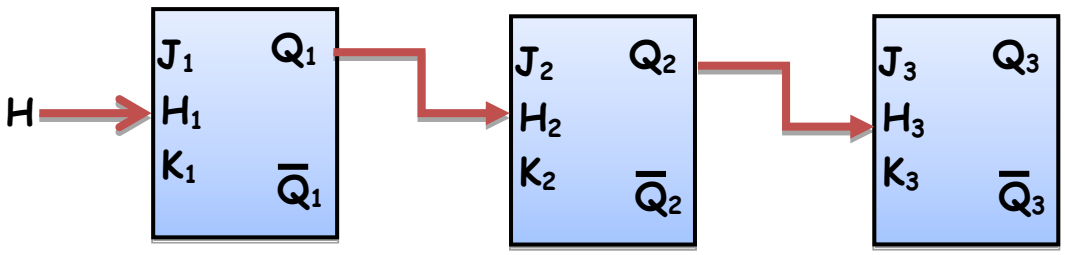
Il existe deux catégories de compteur :

- Les compteurs Asynchrone.
- Les compteurs Synchrone.

4.3.1 Les compteurs Asynchrones

Un compteur est dit asynchrone, si les impulsions à compter (***impulsionsd'horloge***) sont appliquer seulement sur l'entrée de la **première** bascule JK, puis l'**horloge** de la **deuxième** bascule JK est relié à la **sortie** ($\mathbf{Q_1}$ou $\mathbf{\overline{Q}_1}$) de la première bascule JK, ensuite l'horloge de la **troisième** bascule JK est relié à la sortie ($\mathbf{Q_2}$ou $\mathbf{\overline{Q}_2}$) de la **deuxième** bascule est ainsi de suite.

Figure83: Synoptique d'un compteur asynchrone



A. Réalisation d'un compteur Asynchrone :

Pour réaliser un compteur binaire asynchrone, on utilise les propriétés de la bascule JK, pour cela en relient les entrées J et K au niveau 1(5V), puis en injectent des impulsions à l'entrée d'horloge.

Remarque :

- ✓ Un compteur modulo **N** compte de **0** à **N-1** (**N** : capacité de comptage).
- ✓ $\mathbf{N \leq 2^X}$(**x** : nombres de bascules JK).

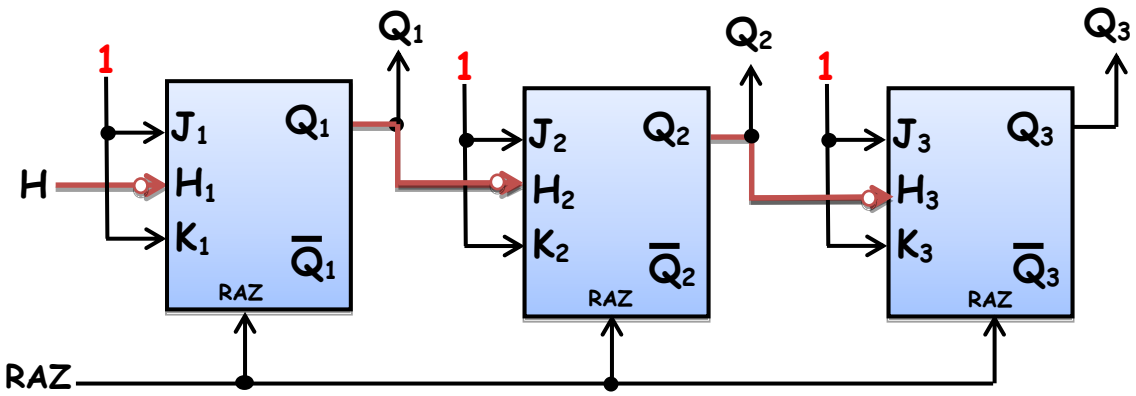
WWW.DZPEN.COM | قلم الجزائر

B. Exemple :

Réaliser un compteur asynchrone modulo 8 (Flip Flop déclenchable sur front descendent).

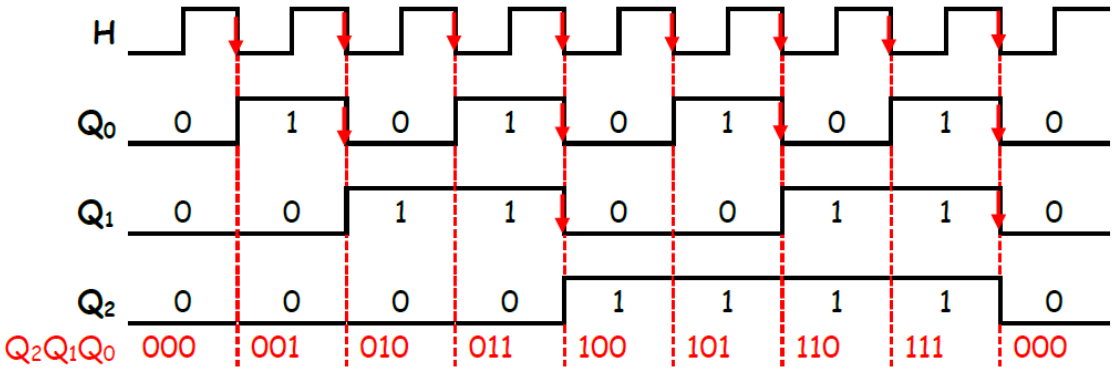
$N=8=2^3 \Longrightarrow$ **3** bascules JK (La bascule J_1 K_1 est celle du poids le plus faible)

Figure84: Synoptique d'un Compteur Asynchrone Modulo 8



❖ **Chronogrammes**

FIG 85 : Chronogramme du compteur Asynchrone Modulo 8

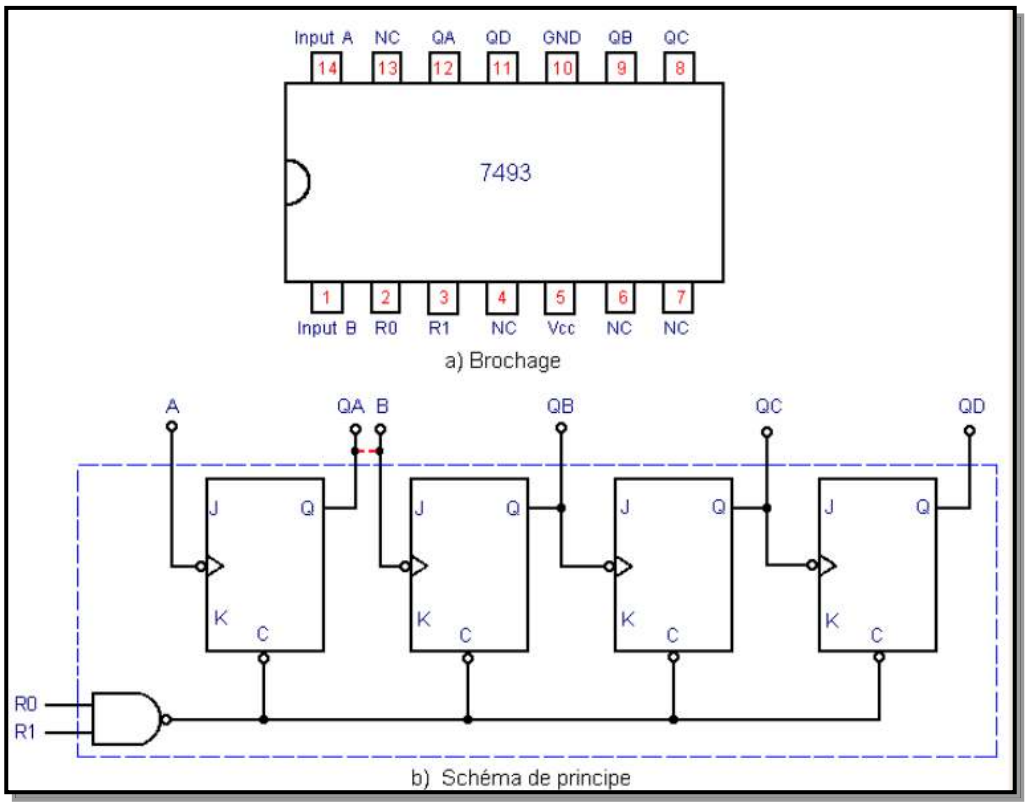


WWW.DZPEN.COM | قلم الجزائر

C. Exemple de brochage d'un compteur Asynchrone

La figure 63 représente le schéma de principe du compteur intégré 7493 réalisé entechnologie TTL(Transistor Transistorlogic) ainsi que son brochage.

Figure86: Compteur Asynchrone 7493



https://www.electronique-et-informatique.fr/Electronique-et-Informatique/Digit/Digit_9T.php

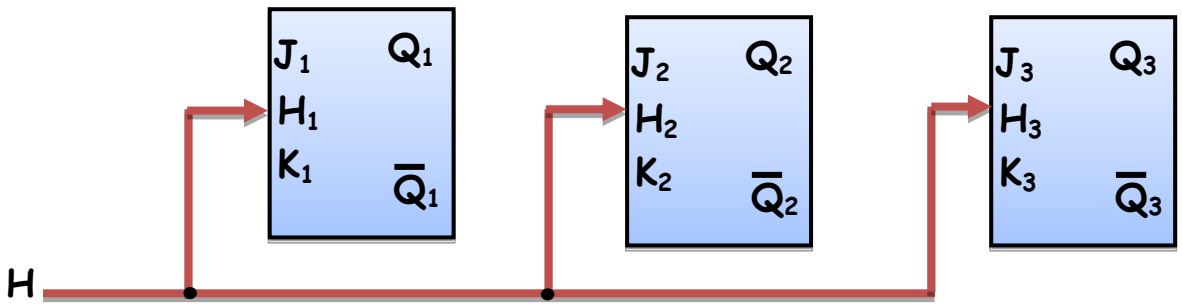
- Le symbole «**NC**.» signifie «non connecté».
- Les entrées **J** et **K**des bascules sont câblées intérieurement à «**1**».
- Une remise à zéro générale asynchrone du compteur est possible grâce aux entrées
  R0 et **R1.** Pour cela les deux entrées **R0** et **R1** doivent être simultanément à «**1**».
- Ce compteur peut fonctionner en **diviseur** par **8** en présentant l'horloge sur l'entrée**B** ou en diviseur par **16** en présentant l'horloge sur l'entrée **A** et en reliant la sortie**QA** à l'entrée **B**.

WWW.DZPEN.COM | قلم الجزائر

4.3.2 Les compteurs Synchrones :

Un compteur est dit synchrone, si les impulsions à comptées (***horloge***) sont appliquer simultanément (aux même temps) sur l'entrée ***CLOCK*** (horloge) de toutes les bascule,

Figure87: Compteurs Synchrones



A. Réalisation d'un compteur Synchrone :

Pour réaliser un compteur Synchrone modulo N, on procède comme suite :

- ✓ On définit le nombre de bascule nécessaire pour réaliser ce compteur suivant la formule $N \leq 2^x$ (**x** : nombres de bascules JK).
- ✓ Ecrire le tableau de séquence (Table de vérité) des bascules A,B,C… selon la séquence de comptage.
- ✓ Etablir à partir des Table de karnaugh les équations de $(J_A,K_A)(J_B,K_B)(J_C,K_C)$ … de chaque bascule en fonction des sorties Q_A , Q_B,Q_C ……

  $J_A= f(Q_A ,Q_B ,Q_C …)$, $K_A=f(Q_A ,Q_B ,Q_C …)$
  $J_B= f(Q_A ,Q_B ,Q_C …)$, $K_B=f(Q_A ,Q_B ,Q_C …)$
  $J_C= f(Q_A ,Q_B ,Q_C …)$, $K_C=f(Q_A ,Q_B ,Q_C …)$

  .

  .
- ✓ Réaliser le schéma logique du compteur.

WWW.DZPEN.COM | قلم الجزائر

B. Exemple : Réaliser un compteur synchrone modulo 10(Flip Flop déclenchable sur front Montant).

Un compteur modulo 10 compte de 0 à 10-1

$N=10 \leq 2^4 \Longrightarrow$ 4 bascules JK

❖ **Table de vérité**

Q_n	Q_{n+1}	J	K
0	0	0	0
		1	0
		0	Ø
1	1	0	0
		0	1
		Ø	0

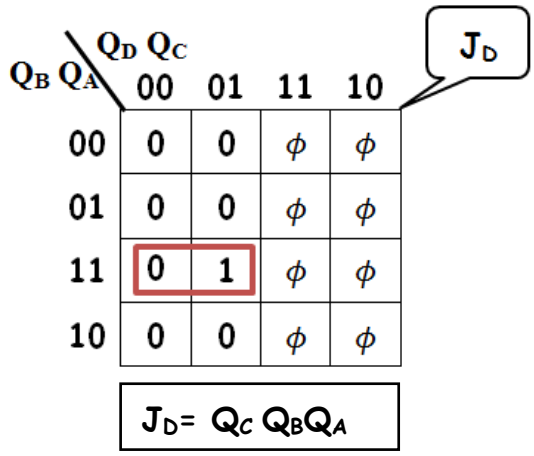
Q_n	Q_{n+1}	J	K
0	1	1	1
		1	0
		1	Ø
1	0	1	1
		0	1
		Ø	1

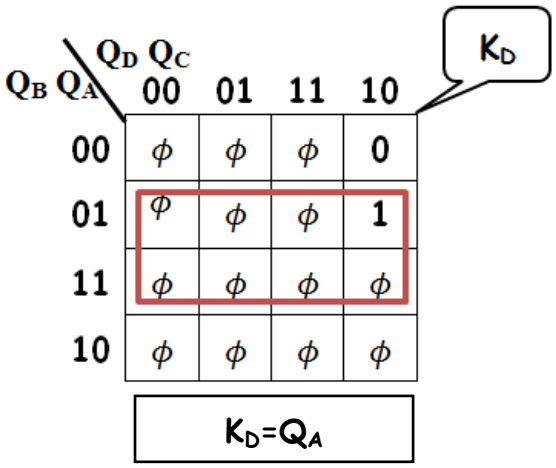
Qn	Qn+1	J	K
0 ⇒	0	0	Ø
1 ⇒	1	Ø	0
0 ⇒	1	1	Ø
1 ⇒	0	Ø	1

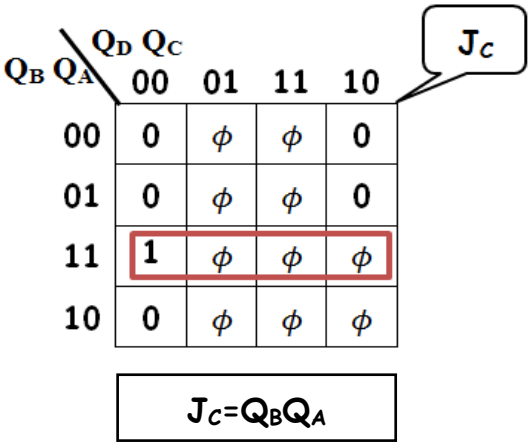
	Q_D	Q_C	Q_B	Q_A	J_D	K_D	J_C	K_C	J_B	K_B	J_A	K_A
0	0	0	0	0	0	ϕ	0	ϕ	0	ϕ	1	ϕ
1	0	0	0	1	0	ϕ	0	ϕ	1	ϕ	ϕ	1
2	0	0	1	0	0	ϕ	0	ϕ	ϕ	0	1	ϕ
3	0	0	1	1	0	ϕ	1	ϕ	ϕ	1	ϕ	1
4	0	1	0	0	0	ϕ	ϕ	0	0	ϕ	1	ϕ
5	0	1	0	1	0	ϕ	ϕ	0	1	ϕ	ϕ	1
6	0	1	1	0	0	ϕ	ϕ	0	ϕ	0	1	ϕ
7	0	1	1	1	1	ϕ	ϕ	1	ϕ	1	ϕ	1
8	1	0	0	0	ϕ	0	0	ϕ	0	ϕ	1	ϕ
9	1	0	0	1	ϕ	1	0	ϕ	0	ϕ	ϕ	1

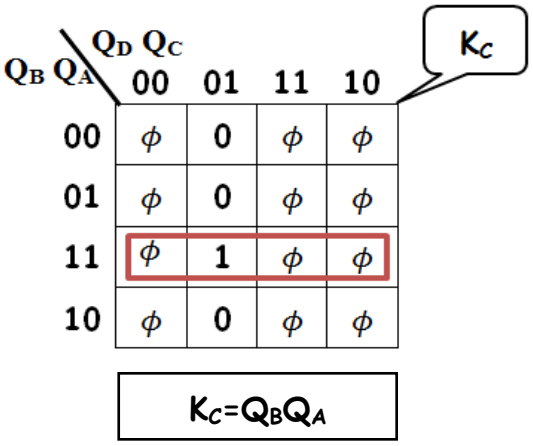
WWW.DZPEN.COM | قلم الجزائر

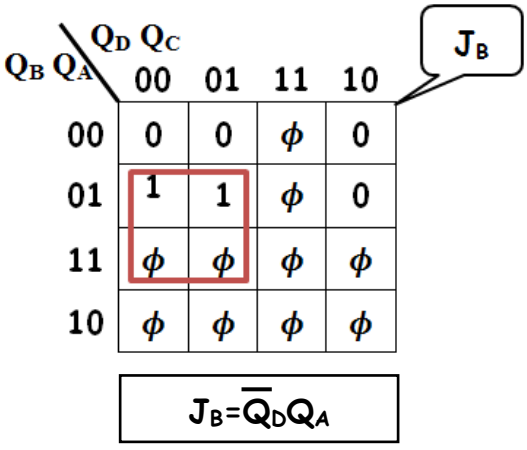
❖ **Tableau Karnaugh**

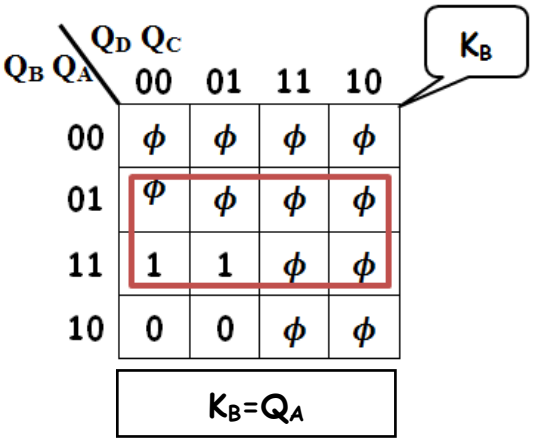












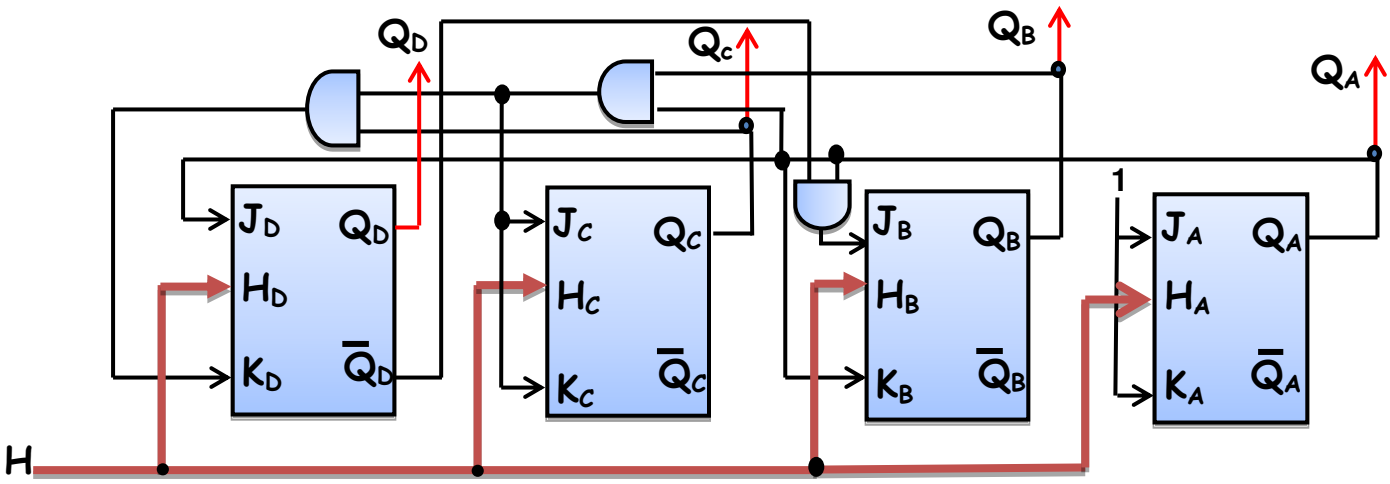
❖ **Equations :**

✓ $J_A=1$ ✓ $K_A=1$	✓ $J_B=\overline{Q}_D Q_A$ ✓ $K_B=Q_A$	✓ $K_C=Q_B Q_A$ ✓ $J_C=Q_B Q_A$	✓ $J_D=Q_A$ ✓ $K_D= Q_C Q_B Q_A$

WWW.DZPEN.COM | قلم الجزائر

❖ **Circuit logique**

Figure88: Circuit logique d'un compteur synchrone modulo 10



4.3.3 Compteur Synchrone réversible (Compteur décompteur)

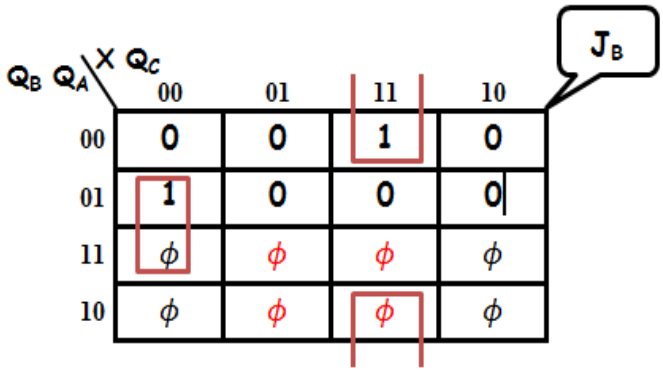
Exemple : Réalisation d'un compteur/décompteur modulo 6

A. Table de vérité :

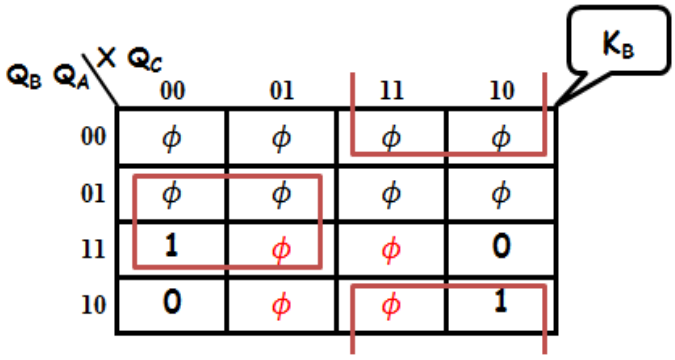
	Imp.	X	Q_C	Q_B	Q_A	J_C	K_C	J_B	K_B	J_A	K_A
COMPTAGE	1	0	0	0	0	0	ϕ	0	ϕ	1	ϕ
	2	0	0	0	1	0	ϕ	1	ϕ	ϕ	1
	3	0	0	1	0	0	ϕ	ϕ	0	1	ϕ
	4	0	0	1	1	1	ϕ	ϕ	1	ϕ	1
	5	0	1	0	0	ϕ	0	0	ϕ	1	ϕ
	6	0	1	0	1	ϕ	1	0	ϕ	ϕ	1
DECOMPTA	1	1	1	0	1	ϕ	0	0	ϕ	ϕ	1
	2	1	1	0	0	ϕ	1	1	ϕ	1	ϕ
	3	1	0	1	1	0	ϕ	ϕ	0	ϕ	1
	4	1	0	1	0	0	ϕ	ϕ	1	1	ϕ
	5	1	0	0	1	0	ϕ	0	ϕ	ϕ	1
	6	1	0	0	0	1	ϕ	0	ϕ	1	ϕ

WWW.DZPEN.COM | قلم الجزائر

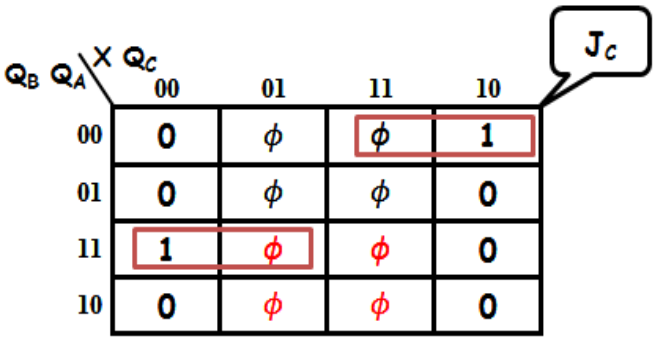
B. Tableau de Karnaugh :



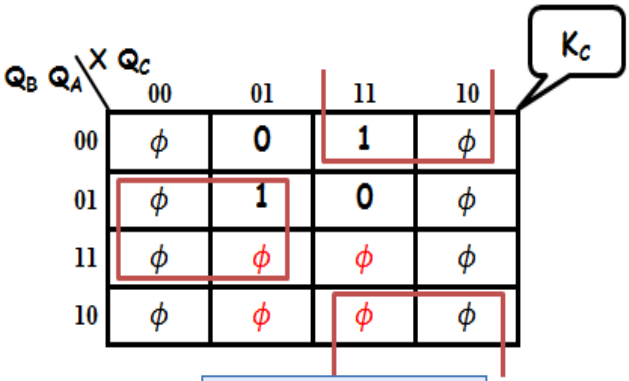
$J_B = X.Q_C + \overline{X}.\overline{Q_C}.Q_A$



$K_B = X \oplus Q_A$



$J_C = X.\overline{Q_A}.\overline{Q_B} + \overline{X}.Q_A.Q_B$



$K_C = X \oplus Q_A$

C. Equations :

$J_A = 1$
$K_A = 1$

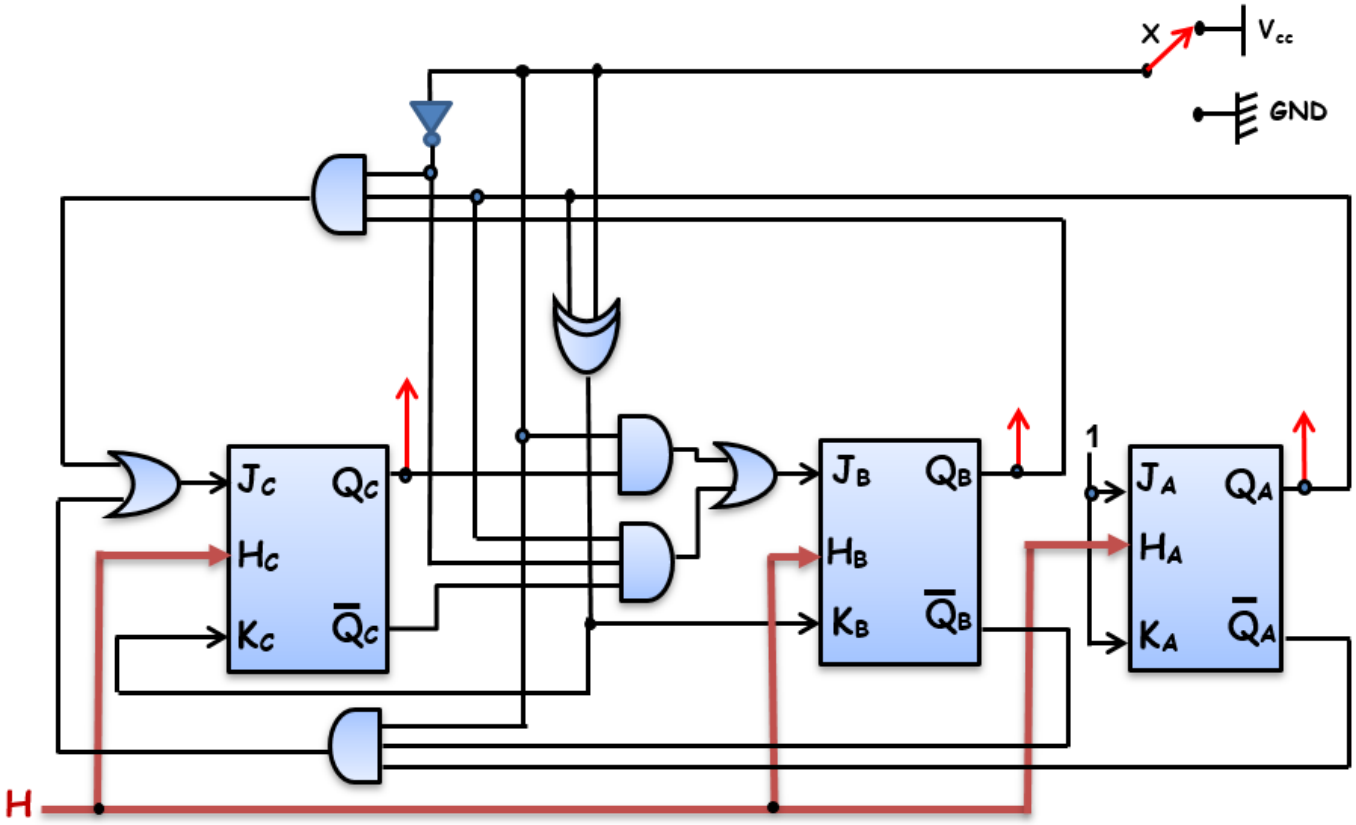
$J_B = X.Q_C + \overline{X}.\overline{Q_C}.Q_A$
$K_B = X \oplus Q_A$

$J_C = X.\overline{Q_A}.\overline{Q_B} + \overline{X}.Q_A.Q_B$
$K_C = X \oplus Q_A$

WWW.DZPEN.COM | قلم الجزائر

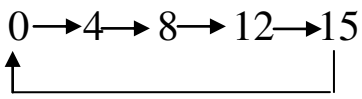
D. Circuit logique

Figure89: Circuit logique d'un compteurdécompteur modulo 6



4.3.4 Exercice 17Réalisez un compteur synchrone qui affiche la séquence suivante :

:



WWW.DZPEN.COM | قلم الجزائر

4.4 Les registres

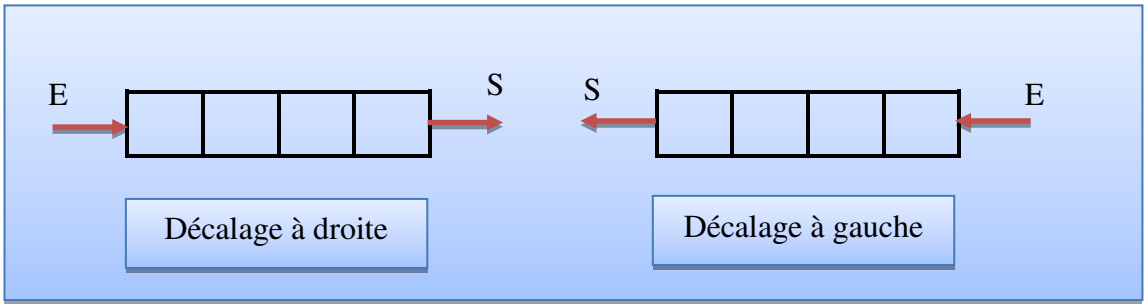
Un registre est **une mémoire** qui peut mémoriser un mot (information) de **N** bits, avec possibilité de décalé à droite ou à gauche le contenu de ce registre, on parle alors de **registre à décalage**, dans ce type de registre le contenue de la bascule se décale d'un rang à chaque impulsion d'horloge.

4.4.1 Différents types de registres

Les registres à décalages différents par le nombre d'information (bits) qu'elles peuvent stockées et par le mode d'introduction et d'extraction de ces informations :

A. Registre à décalage 4 bits entrée série sortie série :SISO *(Serial IN- Serial OUT).*

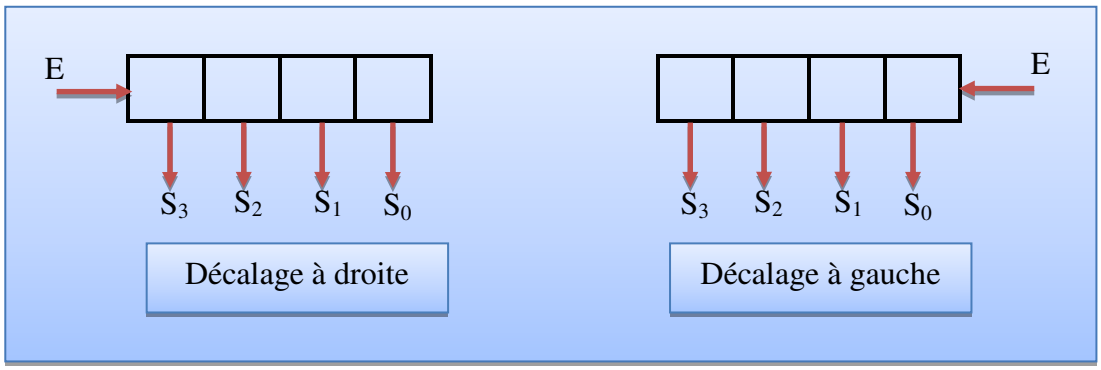
Figure90: Registre à décalage 4 bits entrée série sortie série



Exemple d'utilisation : ces registres permettent de décaler les informations

B. Registre à décalage 4 bits entrée série sortie parallèle :SIPO *(Serial IN-Parallel OUT).*

Figure91: Registre à décalage 4 bits entrée série sortie

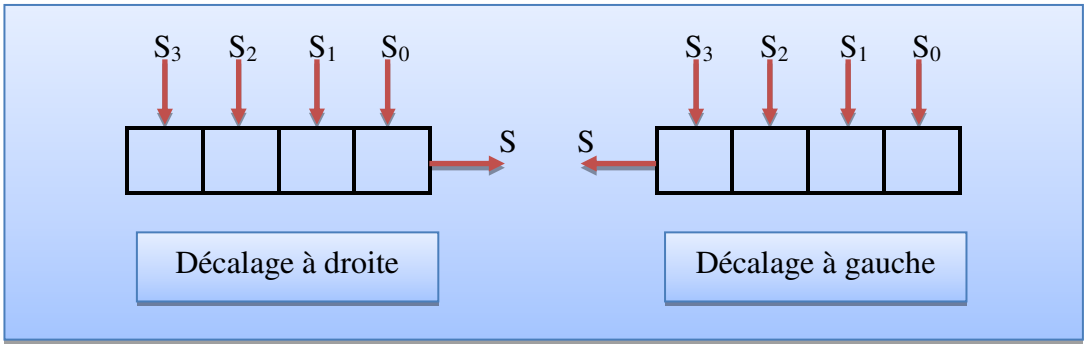


Exemple d'utilisation : conversion série parallèle

WWW.DZPEN.COM | قلم الجزائر

C. Registre à décalage 4 bitsentrée parallèle sortie série : PISO*(Parallel IN-Serial OUT).*

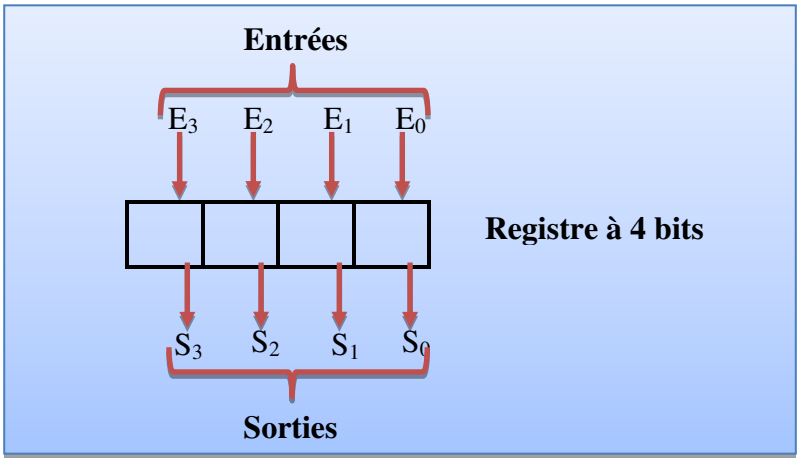
Figure92: Registre à décalage 4 bits entrée parallèle sortie



Exemple d'utilisation: transmission en série des données sur une ligne (conversion parallèle série)

D. Registre à décalage 4 bitsentrée parallèle sortie parallèle :PIPO *(Parallel IN-Parallel OUT)*

Figure93: Registre à décalage 4 bits entrée parallèle sortie

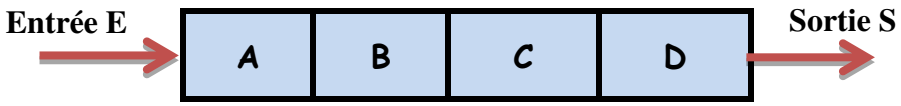


Exemple d'utilisation :mémorisation des états des sorties des compteurs avant affichage

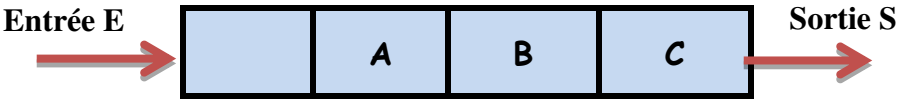
WWW.DZPEN.COM | قلم الجزائر

4.4.2 Fonctionnement d'un registre à décalage

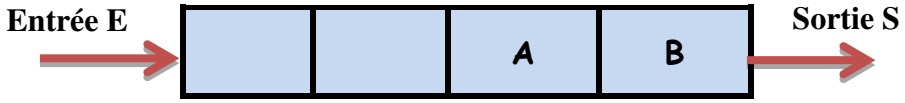
A. Principe de fonctionnement d'un registre à décalage 4 bits entrée/sortie série



➢ **1er Impulsion**

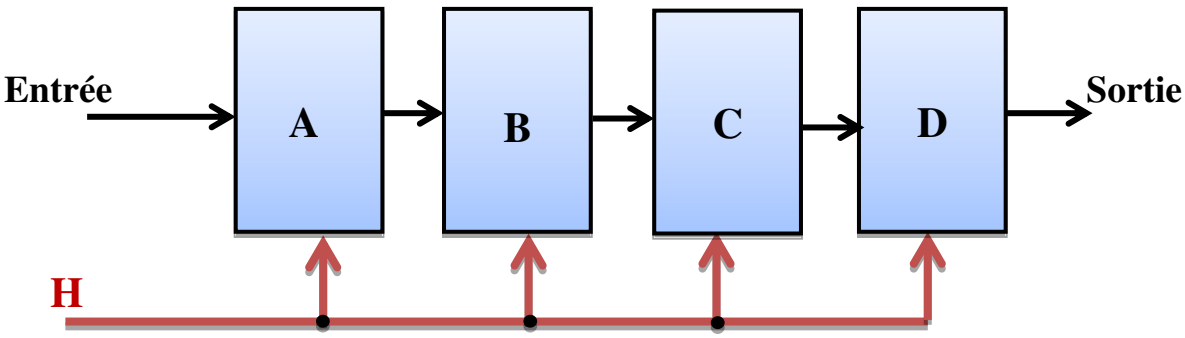


➢ **2eme Impulsion**



B. Registre à décalage 4 bits entrée/sortie série à base de bascule JK ou D

Figure94: Synoptique d'un Registre à décalage 4 bitsentrée/sortie série

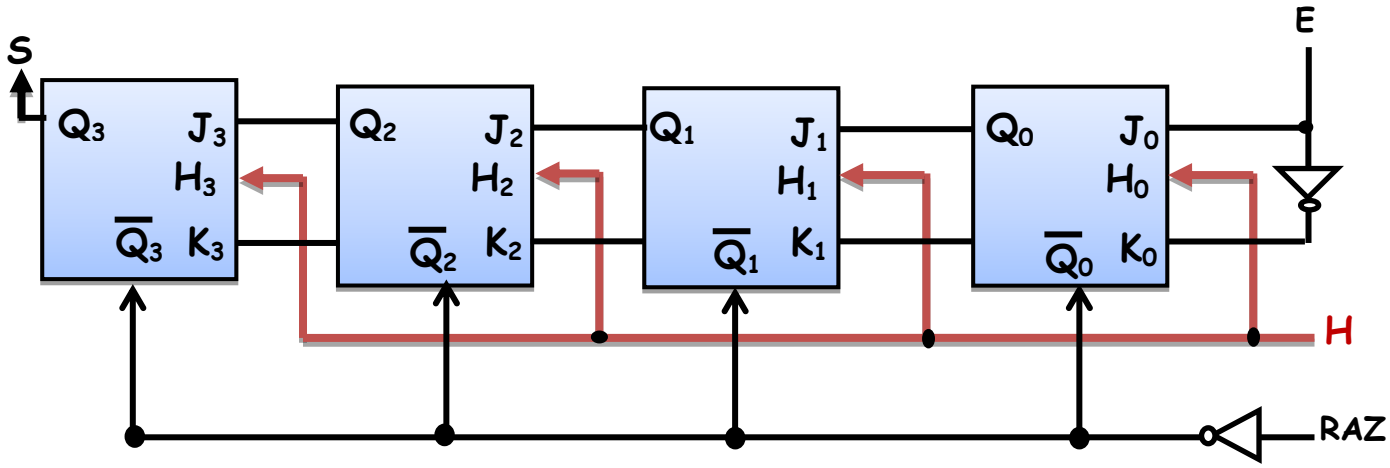


La 1ère impulsion d'horloge introduit la 1er information (1er bit) dans la **bascule A**, la 2eme impulsion décale l'information de la **bascule A** vers la **bascule B** et introduit la 2eme information (2eme bit) dans la **bascule A**et ainsi de suite jusqu'à la 4ieme impulsion.

WWW.DZPEN.COM | قلم الجزائر

4.4.3 Registre à décalage à l'aide de bascule JK

Figure95: Registre à décalage 4 bits entrée/sortie série à base debascule JK



Ce registre contient 4 bascules JK, il peut stocker une information de 4 bits, les sorties Q et $\overline{Q}$ sont relier à J et K de la bascule suivante.

Exemple : On veut introduire l'information **1011** dans le registre (Q_3=1,Q_2=0,Q_1=1,Q_0=1)

E=11erImp d'horloge	E=02emeImp d'horloge
J_1=1, K_1=0 ⇒ Q_0=1	J_1=0, K_1=1 ⇒ Q_0=0
J_2=0, K_2=1 ⇒ Q_1=0	J_2=1, K_2=0 ⇒ Q_1=1
J_3=0, K_3=1 ⇒ Q_2=0	J_3=0, K_3=1 ⇒ Q_2=0
J_4=0, K_4=1 ⇒ Q_3=0	J_4=0, K_4=1 ⇒ Q_3=0
Le Registre affiche **(0001)**	Le Registre affiche **(0010)**

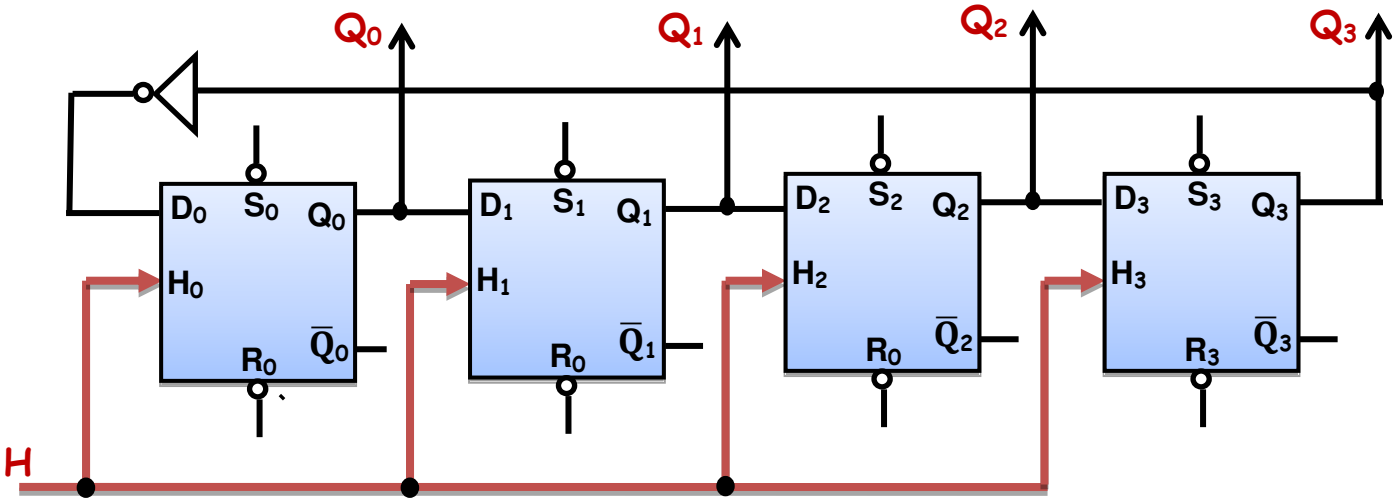
WWW.DZPEN.COM | قلم الجزائر

E=1 3eme Imp d'horloge	E=14emeImp d'horloge
$J_1=1$, $K_1=0$ ⇒ $Q_0=1$	$J_1=1$, $K_1=0$ ⇒ $Q_0=1$
$J_2=0$, $K_2=1$ ⇒ $Q_1=0$	$J_2=1$, $K_2=0$ ⇒ $Q_1=1$
$J_3=1$, $K_3=0$ ⇒ $Q_2=1$	$J_3=0$, $K_3=1$ ⇒ $Q_2=0$
$J_4=0$, $K_4=1$ ⇒ $Q_3=0$	$J_4=1$, $K_4=0$ ⇒ $Q_3=1$
Le Registre **affiche(0101)**	Le Registre affiche **(1011)**

	J_3	K_3	J_2	K_2	J_1	K_1	J_0	K_0	Q_3	Q_2	Q_1	Q_0
E=1 1erImp d'horloge	0	1	0	1	0	1	1	0	0	0	0	1
E=0 2emeImp d'horloge	0	1	0	1	1	0	0	1	0	0	1	0
E=1 3emeImp d'horloge	0	1	1	0	0	1	1	0	0	1	0	1
E=1 4emeImp d'horloge	1	0	0	1	1	0	1	0	1	0	1	1

4.4.4 Exercice 18 Soit le circuitde figure 96 suivant :

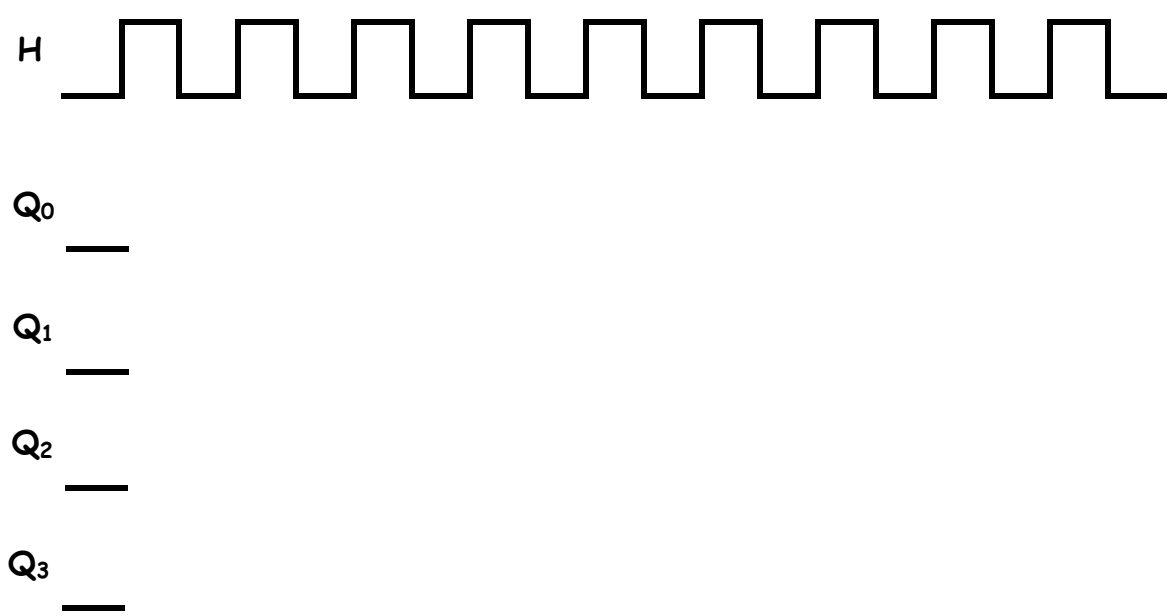
Figure96: Registre à décalage 4 bits à base debascule D



WWW.DZPEN.COM | قلم الجزائر

A. Compléter le chronogramme de la figure 97 suivant :

Figure97: Chronogramme du registre à décalage 4 bits à base debascule D



B. Quelle est la fonction réalisée?

WWW.DZPEN.COM | قلم الجزائر

4.5 Résumé

- Dans un circuit ***combinatoire*** l'état de sortie ne dépend que de l'état présent des entrées, par contre dans un circuit ***séquentiel***, l'état de la sortie dépend non seulement de la combinaison appliquée à l'entrée mais aussi de l'état ***précédent*** des sorties du circuit.
- Dans un circuit asynchrone la lecture de l'entrée se fait à tout instant, alors que dans un circuit synchrone elle n'est prise en compte qu'à des instants précis synchronisés par une fréquence d'horloge
- Le signal d'horloge est généralement un signal carré, il est habituellement distribué à tous les étages du système
- Une bascule est une mémoire élémentaire qui ne peut mémoriser qu'un seul bit.
- Dans une bascule "***RS***", l'entrée ***S*** permet de mettre la sortie ***Qn*** à l'état logique 1(***Qn=1***) alors que l'entrée **R** la remet à l'état logique 0 (***Qn=0***), si ***R=S=0*** la bascule mémorise l'état précèdent (***Qn= Qn-1***), c'est une bascule qui possède un état indéterminé qui est ***R=S=1*** et qui la rend imprévisible.
- La bascule **D** (Verrou) joue le rôle de mémoire élémentaire,le signale d'horloge représentant le signale de validation ***V=0*** la bascule mémorise l'état précèdent, ***V=1*** la bascule recopie l'entrée ***D*** à la sortie.
- Dans une bascule "***JK***", l'entrée ***J***permet de mettre la sortie ***Qn*** à l'état logique 1(***Qn=1***) alors que l'entrée ***K*** la remet à l'état logique 0(***Qn=0***),***J=K=0*** la bascule mémorise l'état précèdent (***Qn=Qn-1***), **J=K=1** la bascule inverse l'état précèdent de sa sortie (**Qn=$\overline{Q}_{n-1}$**)
- La bascule **T** comprend deux entrées « ***T*** » et « ***H*** »le signale d'horloge ***H***représentant le signale de validation ***H=0*** la bascule mémorise l'état précèdent, si ***H=1*** et ***T=1*** la bascule recopie à sa sortie l'inverse de la sortie précédente (**Qn=$\overline{Q}_{n-1}$**)
- la bascule ***T***est connue spécialement pour son état de complémentation qui donne au circuit la fonction de ***diviseur de fréquence (d'horloge) par deux***.
- Un compteur est une succession de bascule ***JK*** ou ***T*** dans le mot binaire se modifie à chaque impulsion d'horloge
- Dans un compteur synchrone, toute les bascules possède le même signal d'horloge, alors que dans un compteur synchrone l'entrée d'horloge de chaque bascule vient de l'une des sorties de la bascule précédente
- Un registre à décalage est un assemblage de bascules commandées par une horloge commune.
- Dans un registre à décalage, une donnée introduite à l'entrée de la première bascule se propage dans les bascules suivantes à chaque signal de l'horloge.

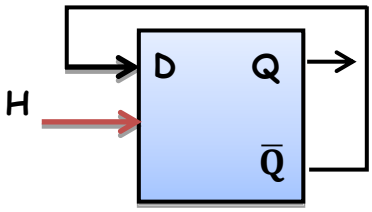
WWW.DZPEN.COM | قلم الجزائر

4.6 Exercices de synthèse 4

- **Spécialité** : Domotique
- **Module** : Logique combinatoire et logique séquentielle
- **Durée : 1h30**
- **But :** Amener les stagiaires
  - ✓ Ácomprendre le fonctionnement des circuits séquentiels(bascules, compteurs et registres)
  - ✓ Á réaliser des circuits séquentiels (bascules, compteurs et registres)
- **Matériel requis :** Documents et support de cours et simulateur des circuits électroniques
- **Mise en situation :** Expliquer le fonctionnement puis déduire les logigrammes des circuits suivants
- **Marche à suivre :**

1. Soit la figure98 suivante

Figure98: Bascule D



A. On applique à l'entrée d'horloge de la bascule D le signal d'horloge suivant :

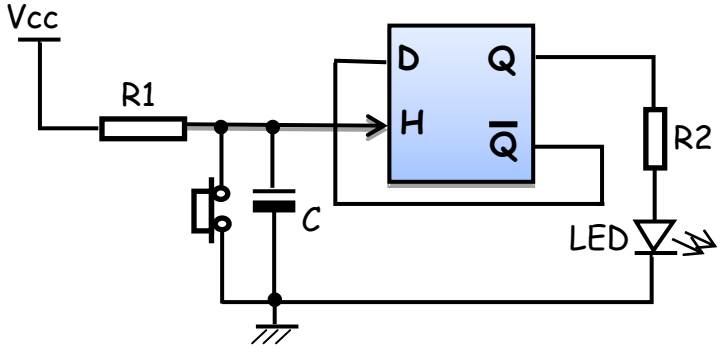
Figure99: Signal d'horloge



B. Tracer le chronogramme obtenu de H et Q.

C. Expliquer le fonctionnement du montage de la figure100 suivant :

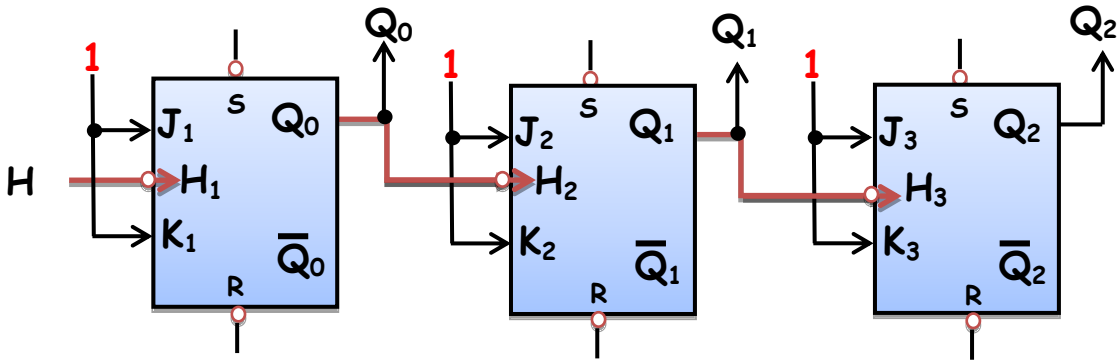
Figure100: Circuit à base de Bascule D



WWW.DZPEN.COM | قلم الجزائر

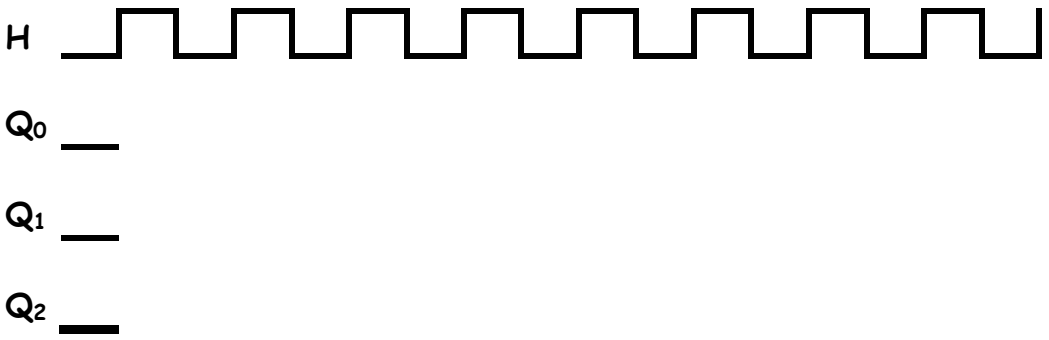
2. Soit la figure101 suivante :

Figure101: Compteur Asynchrone



A. Compléter le chronogramme suivant :

Figure102: Chronogramme du compteur asynchrone de la figure101

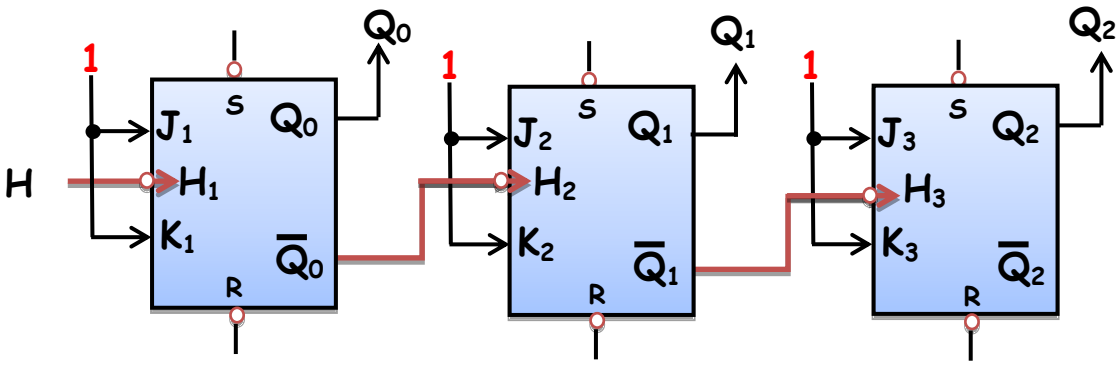


B. Quelle est la séquence obtenue ?

C. Quel est le modulo ?

3. Soit la figure103 suivante :

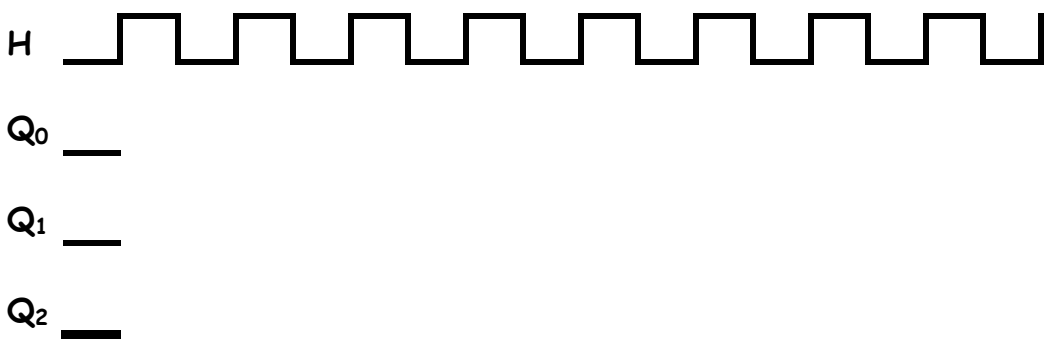
Figure103: Compteur Asynchrone



WWW.DZPEN.COM | قلم الجزائر

A. Compléter le chronogramme suivant :

Figure104 : Chronogramme du compteur asynchrone de la figure103



B. Quelle est la séquence obtenue ?
C. Quel est le modulo ?
D. Conclure pour le compteur et le décompteur
E. A partir des 2 schémas précédents concevoir un compteur/décompteur

WWW.DZPEN.COM | قلم الجزائر

CHAPITRE 5

LES PRINCIPAUX ELEMENTS D'UN AUTOMATE PROGRAMMABLE

WWW.DZPEN.COM | قلم الجزائر

Les Automates Programmables Industriels (***API***), ou en anglais, Programmable Logic Controller (***PLC***), sont apparus aux Etats-Unis vers 1969 (*Modicon* :*entreprise de conception d'automatismes*) où ils ont remplacés la logique a relais câblée et ils répondaient aux désirs des industries de l'automobile de développer des chaînes de fabrication automatisées qui pourraient suivre l'évolution des techniques et des modèles fabriqués.

Un automate programmable industriel est une machine électronique, programmable par un personnel non informaticien et destiné à commander au moyen de signaux d'entrées et de sorties, et d'un programme informatique, en temps réel, des procédés de processus industriels par un traitement séquentiel.

Un automate programmable industriel (API) est une forme particulière de contrôleur à microprocesseur.

5.1 Technologie des automates programmables industriels :

Les automates programmables industriels peuvent être de type compact ou modulaire.

5.1.1 Type compact (centralisé)

Il intègre le processeur, l'alimentation, les entrées et les sorties dans un seul boitier (***rack***). Selon les modèles et les fabricants, il pourra réaliser certaines fonctions supplémentaires (comptage, E/S analogiques ...) et recevoir des extensions en nombre limité. Ces automates, de fonctionnement simple, sont généralement destinés à la commande de petits automatismes

***Figure*105**:Automates programmables industriels (API)

SIEMENS LOGO	CROUZET MILLENIUM	SCHNEIER ZELIO	SCHNEIDER TWIDO	MOELLER PS4
				

WWW.DZPEN.COM | قلم الجزائر

5.1.2 Type modulaire

L'automate programmable se présente comme un ensemble de blocs fonctionnels. Généralement, chaque bloc est physiquement réalisé par un module spécifique (coffret, rack, baie ou cartes). Ces différents modules s'articulant autour d'un canal de communication (bus interne). L'automate programmable est du type modulaire contenant un rack, un module d'alimentation, un processeur, des modules d'entrées/sorties (d'E/S), des modules de communication et de comptage. Cette organisation modulaire permet une grande souplesse de configuration pour les besoins de l'utilisateur, ainsi qu'un diagnostic et une maintenance facile.

***Figure*106**:Automates programmables industriels (API) modulaires

SIEMENS S7-300	SCHNEIDER TSX 37	MOELLER	SCHNEIDER TSX 57
			

Cours « Automates Programmables Industriels »Présenté par l'enseignant : Dr. Sofiane DoudouUniversité Mohammed SeddikBenyahia-Jijel

Remarque : Les automates compacts permettent de commander des sorties en tout ou rienet gèrent parfois des fonctions de comptage et de traitement analogique mais Lesautomates modulaires permettent de réaliser de nombreuses autres fonctions grâce à desmodules intelligents que l'on dispose sur un ou plusieurs racks. Ces modules ont l'avantagede ne pas surcharger le travail de la CPU (Central processing unit :unité centrale de traitement) car ils disposent bien souvent de leur propreprocesseur**.**

5.1.3 Avantages et inconvénients des automates programmables industriels

Les automates programmables industriels, avec leur solution programmée, présentent de nombreux avantages par rapport à la technologie de logique câblée. Parmi ces intérêts, on cite :

- La simplicité car avec le même API, il devient possible de traiter une variété d'applications qui, autrement, auraient chaque fois requis des matériels différents.

WWW.DZPEN.COM | قلم الجزائر

- La flexibilité : car le changement du mode de fonctionnement de la machine commandée s'obtient par simple modification du programme enregistré en mémoire.
- La réduction des coûts de câblage et de maintenance.
- La réduction de beaucoup d'espace requis pour l'installation.
- Les éléments qui les composent sont particulièrement robustes leur permettant de fonctionner dans des environnements particulièrement hostiles.
- Ils permettent d'assurer un temps d'exécution minimal, respectant un déterminisme temporel et logique, garantissant un temps réel effectif (le système réagit forcément dans le délai fixé).

En contrepartie, ils présentent les inconvénients suivants :

- Le prix est cher : le prix est notamment dépendant du nombre d'entrées/sorties nécessaires, de la mémoire dont on veut disposer pour réaliser le programme.
- La connaissance des langages de programmation.

5.1.4 Domaines d'emploi des automates programmables industriels

On utilise les API dans tous les secteurs industriels pour la commande des machines (convoyage, emballage ...) ou des chaînes de production (automobile, agroalimentaire, ...) ou il peut également assurer des fonctions de régulation de processus (métallurgie, chimie ...). Il est de plus en plus utilisé dans le domaine du bâtiment (tertiaire et industriel) pour le contrôle du chauffage, de l'éclairage, de la sécurité ou des alarmes.

- **Nature des informations traitées par l'automate :** Les informations peuvent être de type :
  - Tout ou rien (T.O.R.) : l'information ne peut prendre que deux états (vrai/faux, 0 ou 1 …). C'est le type d'information délivrée par un détecteur, un bouton poussoir …
  - Analogique : l'information est continue et peut prendre une valeur comprise dans une plage bien déterminée. C'est le type d'information délivrée par un capteur (pression, température …)
  - Numérique : l'information est contenue dans des mots codés sous forme binaire ou bien hexadécimale. C'est le type d'information délivrée par un ordinateur ou un module intelligent.

WWW.DZPEN.COM | قلم الجزائر

5.1.5 Critères de choix des automates programmables industriels

Le choix d'un automate programmable est en premier lieu le choix d'une société ou d'un groupe et les expériences vécues sont déjà un point de départ. Les grandes sociétés privilégieront deux fabricants pour faire jouer la concurrence et pouvoir "se retourner" en cas de "perte de vitesse" de l'une d'entre elles. Le personnel de maintenance doit toutefois être formé sur ces matériels et une trop grande diversité des matériels peut avoir de graves répercussions. Un automate utilisant des langages de programmation de type GRAFCET est également préférable pour assurer les mises au point et dépannages dans les meilleures conditions. La possession d'un logiciel de programmation est aussi source d'économies (achat du logiciel et formation du personnel). Des outils permettant une simulation des programmes sont également souhaitables. Il faut ensuite quantifier les besoins :

- Nombre d'entrées / sorties : le nombre de cartes peut avoir une incidence sur le nombre de racks dès que le nombre d'entrées / sorties nécessaires devient élevé.
- Type de processeur : la taille mémoire, la vitesse de traitement et les fonctions spéciales offertes par le processeur permettront le choix dans la gamme souvent très étendue.
- Fonctions ou modules spéciaux : certaines cartes (commande d'axe, pesage ...) permettront de "soulager" le processeur et devront offrir les caractéristiques souhaitées (résolution, ...). Fonctions de communication : l'automate doit pouvoir communiquer avec les autres systèmes de commande (API, supervision ...) et offrir des possibilités de communication avec des standards normalisés (Profibus ...).

5.1.6 Exercice 19

A. Qu'elle est la différenceentre un automate programmable compact et modulaire ?

B. Classé ses automates selon leur type (compact ou modulaire) MoellerPS4, Siemens S7-300 ,Crouzet Millenium , SchneiderTSX 57, SCHNEIER Zelio

WWW.DZPEN.COM | قلم الجزائر

5.2 Architecture d'un automate programmable industriel(API)

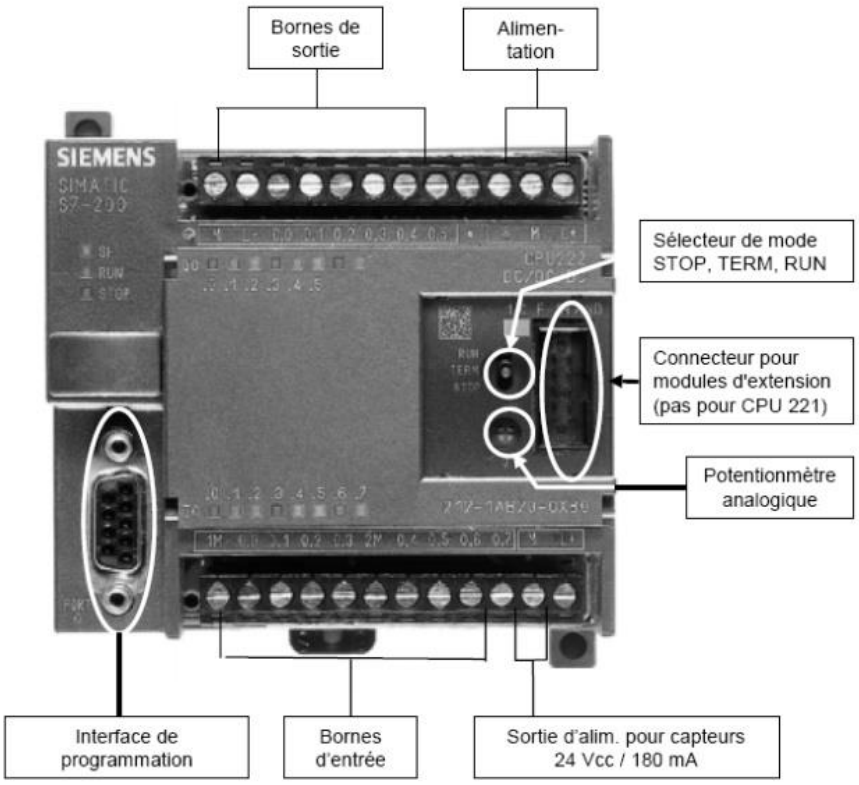
Les caractéristiques principales d'un automate programmable industriel (**API**) sont :coffret, rack, baie ou cartes

- ✓ Compact ou modulaire
- ✓ Tension d'alimentation
- ✓ Taille mémoire
- ✓ Sauvegarde (EPROM, EEPROM, pile, …)(***Voir page 109 et page 110***)
- ✓ Nombre d'entrées / sorties
- ✓ Modules complémentaires (analogique, communication,..)
- ✓ Langage de programmation

5.2.1 Aspect extérieur d'un automate programmable industriel (API)

On va prendre comme un exemple d'automate compact le S7-200 CPU222 et un autre modulairele TSX3721

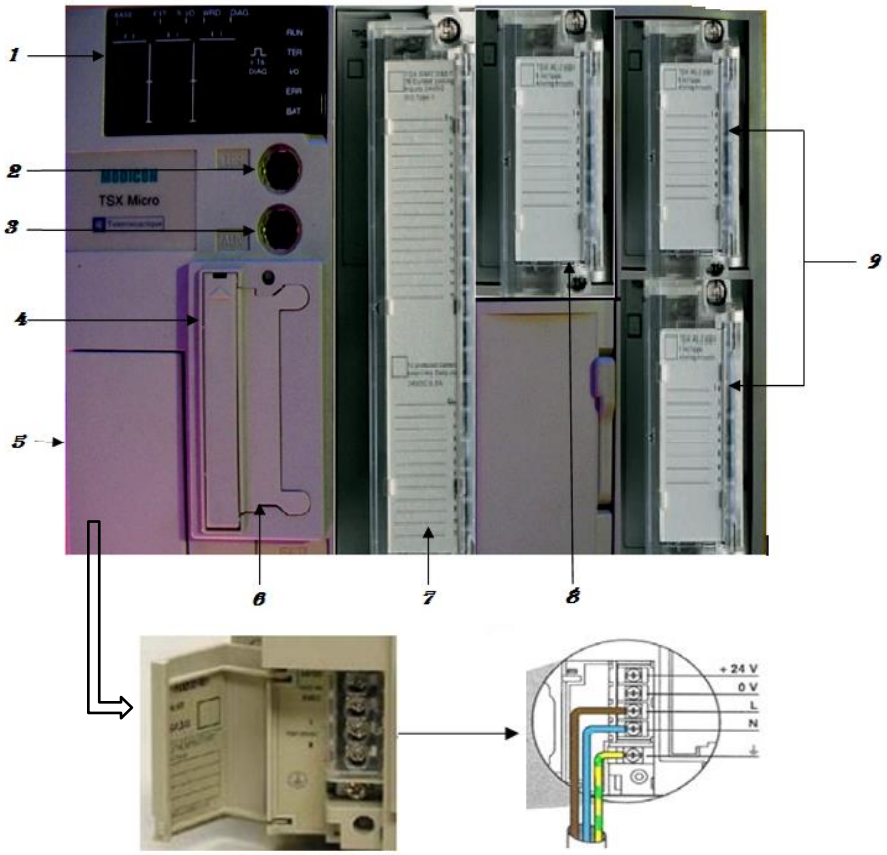
***Figure*109 :**Descriptif physique de l'automate Compact le S7-200 CPU222



https://www.technologuepro.com/cours-automate-programmable-industriel/Les-automates-programmables-industriels-API.htm

WWW.DZPEN.COM | قلم الجزائر

***Figure*110 :**Descriptif physique de l'automate modulaire le TSX3721



Mémoire de Master. Thème. « Etude et réalisation de systèmes automatises didactiques mise en œuvre de l'automate Schneider TSX37 » .Promotion 2013/2014 .Université Constantine 1

1- Bloc de visualisation (affiche des informations sur l'état del'automate).
2- Prise TERMINAL (TER) permet de connecté le terminal deprogrammation.
3- Prise de dialogue opérateur (AUX) permet de connecté un pupitre de dialogue.
4- Emplacement pour carte d'extensionmémoire.
5- Trappe d'accès aux bornesd'alimentation.
6- Emplacement pour coupleur decommunication.
7- Emplacement pour module d'entrée dans le module de base de type TOR sous la référence TSX DEZ32D2.
8- Emplacement pour module d'entrée dans le module de base de type analogique sous la référence TSX AEZ 4144multirage(TC.RTD).
9- Emplacement pour module de sortie dans le module de base de type TOR sous la référence TSX DSZ08R5.

WWW.DZPEN.COM | قلم الجزائر

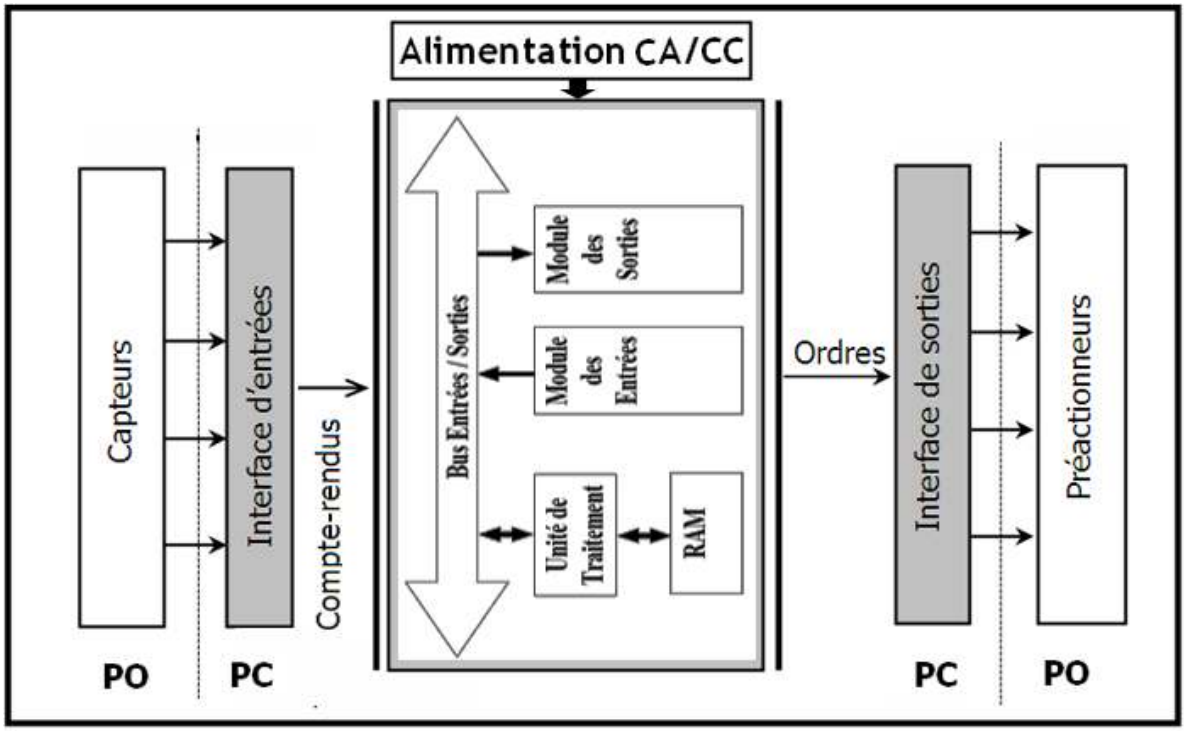
Un automate programmable industriel modulairese présente sous la forme d'un ou plusieurs profiléssupports **(racks)** dans lesquels viennent s'enficher les différents modules fonctionnels :

- L'alimentation 110/220 VCA ou 24 VCC
- L'unité centrale de traitement à base de microprocesseur,
- Des cartes d'entrées/sorties logiques (TOR),
- Des cartes d'entrées/sorties analogiques (ANA),
- Des cartes de comptage rapide,
- Des cartes de communication (CP),
- Des cartes spécifiques pour : réseaux, asservissement, régulation ou commande

5.2.2 Structure interne d'un automate programmable industriel (API)

La structure interne d'un automate programmable industriel (API) est assez voisine de celle d'un système informatique simple. Cette structure comporte quatre parties principales : Une unité de traitement (un processeur CPU) ;Une mémoire ; des Interfaces et des modules d'entrées-sorties ;Une alimentation. Un bus interne (liaisons parallèles) est utilisé pour échanger les informations entre les différents éléments de l'automate (entrées, sorties, mémoires).

***Figure*111**:Structure interne d'un automate programmable industriel (API)



https://www.technologuepro.com/cours-automate-programmable-industriel/Les-automates-programmables-industriels-API.htm?fbclid=IwAR03Cm3ozTdKBopZ1JJ1eus0KT8CLrKf3-Eo5LCfLvVzMsU2G9tB-0i5WaI

WWW.DZPEN.COM | قلم الجزائر

5.2.2.1 Unité de traitement (Processeur)

Un processeur est l'unité fonctionnelle capable d'interpréter et d'exécuter les instructions du programme. Dans un API le processeur gère l'ensemble des échanges informationnels en assurant :

- La lecture des informations d'entrée.
- L'exécution des instructions du programme mis en mémoire.
- La commande ou l'écriture des sorties.

Pour réaliser ces différentes fonctions, le processeur se compose :

- D'une ***Unité Logique*** (***UL***) qui traite les opérations logiques ET, OU et Négation.
- D'une ***Unité Arithmétique et Logique*** (***UAL***) qui traite les opérations de temporisation, decomptage et de calcul.
- D'un ***Accumulateur*** qui est un registre de travail dans lequel se range une donnée ou unrésultat.
- D'un ***Registre d'Instruction*** qui contient, durant le temps de traitement, l'instruction aexécuté.
- D'un ***Décodeur d'Instruction*** qui décode l'instruction a exécuté en y associant lesmicroprogrammes de traitement.
- D'un ***Compteur Programme*** ou Compteur Ordinal qui contient l'adresse de la prochaineinstruction a exécuter et gère ainsi la chronologie de l'exécution des instructions duprogramme.

5.2.2.2 Mémoire Elle est conçue pour :

- Recevoir les informations issues des capteurs d'entrées
- Les informations générées par le processeur et destinées à la commande dessorties.
- Recevoir et conserver le programme du processus.

Il existe dans les automates trois types de mémoires qui remplissent des fonctions différentes

A. Mémoire de programme : Cette mémoire est utilisée pour stocker le programme. Elle est engénéral de type ***EEPROM*** (***ElectricallyErasableProgrammable Read Only Memory*** : Mémoire à lecture seulprogrammableseffaçable électriquement).

WWW.DZPEN.COM | قلم الجزائر

B. **Mémoire système :**Cette mémoire, présente dans le cas d'automates à microprocesseurs, estutilisée pour stocker le système d'exploitation et elle est programmée en usine par leconstructeur. Elle peut donc sans problème être réalisée en technologie ***PROM*** (c'est-à-direprogrammable une seule fois, sans possibilité d'effacement) voire ***ROM*** (mémoire morteaccessible uniquement en lecture).

C. **Mémoire de données :**Elle est utilisable en lecture-écriture des données pendant lefonctionnement. C'est une mémoire de type ***RAM***(*Random Access Memory : Mémoire à accès aléatoire)*mémoire vive dans laquelle on peut lire, écrire et effacer) utilisant une technologie spéciale (***CMOS*** :ComplementaryMetalOxideSemiconductor : semi-conducteur complémentaire à l'oxyde de métal) à très faible consommationélectrique du moins, à l'état de repos et elle nécessite une batterie de sauvegarde.

5.2.2.3 Le module d'entrées

Un module d'entrées doit permettre à l'Unité Centrale de l'automate, d'effectuer une "lecture" de l'état logique des capteurs qui lui sont associes (module 4, 8, 16 ou 32 entrées). A chaque entrée correspond une voie qui traite le signal électrique pour élaborer une information binaire, le bit d'entrée qui est mémorise. L'ensemble des bits d'entrées forme le "mot" d'entrées. Périodiquement, le Processeur de l'automate programmable vient questionner (adresser) le module : le contenu du mot d'entrées du module est alors recopié dans la ***mémoire données*** de l'automate programmable.

***Figure* 112 :**Module d'entrée numérique



https://www.directindustry.fr/prod/brainchild-electronic-co-ltd/product-25130-492371.html

WWW.DZPEN.COM | قلم الجزائر

Un module d'entrées est principalement défini par sa modularité (nombre de voies) et les caractéristiques électriques acceptées (tension, nature du courant...).

A. **Les cartes d'entrées logiques :**Les cartes d'entrées logiques (cartes d'entrées tout ou rien) permettent de raccorder à l'automate les différents capteurs logiques tels que :
   - Boutons poussoirs.
   - Fins de course.
   - Capteurs de proximité inductifs ou capacitifs.
   - Capteurs photoélectriques.etc.…

   Elles assurent l'adaptation, l'isolement, le filtrage et la mise en forme des signauxélectriques. Une diode électroluminescente située sur la carte donne l'état de chaqueentrée.

B. **Les cartes d'entrées analogiques :**Les cartes d'entrées analogiques permettent de gérer des grandeurs analogiques en faisant varier un code numérique au sein du module.

   Les entrées analogiques disposent d'un seul convertisseur analogique/numérique, elles sontscrutées les unes a la suite des autres par un multiplexeur a relais.

5.2.2.4 Le module de sorties

Un module de sorties permet à l'automate programmable d'agir sur les actionneurs. Périodiquement, le processeur adresse le module et provoque l'écriture des bits d'un mot mémoire sur les voies de sorties du module. L'élément de commutation du module est soit électronique (transistors, triac) soit électromécanique (contacts de relais internes au module).

A. **Les cartes de sorties logiques** :Les cartes de sorties logiques (tout ou rien) permettent de raccorder à l'automate lesdifférents pré- actionneurs tels que :
   - Les contacteurs.
   - Les voyants.
   - Les distributeurs.
   - Les afficheurs...

WWW.DZPEN.COM | قلم الجزائر

Les tensions de sorties usuelles sont de 5 volts en continu ou de 24, 48, 110, 220 volts enalternatif. Ces cartes possèdent soit des relais, soit des triacs, soit des transistors. L'état dechaque sortie est visualisé par une diode électroluminescente.

B. **Les cartes de sortie analogiques :**Les cartes de sortie analogiques permettent de gérer des grandeurs analogiques en faisantvarier un code numérique au sein du module. Ces modules assurent la conversionnumérique/analogique, Les sorties analogiques peuvent posséder un convertisseur par voie.Le nombre de voies sur ces cartes est de 2 ou 4.

5.2.2.5 Le module d'alimentation

Compose de blocs qui permettent de fournir à l'automate l'énergie nécessaire à sonfonctionnement. A partir d'une alimentation en 220 volts alternatifs, ces blocs délivrent dessources de tension dont l'automate a besoin : 24V, 12V ou 5V en continu.

En règle générale, un voyant positionne sur la façade indique la mise sous tension del'automate.

5.2.2.6 Le module de communication

Comprend les consoles, les boitiers de tests et les unités de dialogue en ligne

A. **Les consoles :**Il existe deux types de consoles. L'une permet le paramétrage et les relèves d'informations(modification des valeurs, et visualisation), l'autre permet en plus la programmation, leréglage et l'exploitation. Cette dernière dans la phase de programmation effectue :
   - L'écriture.
   - La modification.
   - L'effacement.
   - Le transfert d'un programme dans la mémoire de l'automate ou dans une mémoireREPROM.

La console peut également afficher le résultat de l'autotest comprenant l'état des modulesd'entrées et de sorties, l'état de la mémoire, de la batterie, etc. Les consoles sont équipéessouvent d'un écran à cristaux liquides. Certaines consoles ne peuvent être utilisées queconnectées à un automate, d'autres peuvent fonctionner de manière autonome grâce à lamémoire interne et à leur alimentation.

WWW.DZPEN.COM | قلم الجزائر

B. Les boîtiers de tests : Destinées aux personnels d'entretien, ils permettent de visualiser le programme ou lesvaleurs des paramètres. Par exemple :

- Affichage de la ligne de programme a contrôlé.
- Visualisation de l'instruction (code opératoire et adresse de l'opérande).
- Visualisation de l'état des entrées.
- Visualisation de l'état des sorties.

C. Les unités de dialogue en ligne : Elles sont destinées aux personnels spécialistes du procède et non de l'automateprogrammable, elle leur permet d'agir sur certains paramètres :

- Modification des constantes, compteurs temporisations.
- Forçage des entrées/sorties.
- Exécution de parties de programme.
- Chargement de programmes en mémoire à partir de cassettes.

5.2.3 Exercice20 :Répondez par vrai ou faux

A. A l'aide de la technologie de logique câblée il devient possible de traiter une variété d'applications qui, autrement, auraient chaque fois requis des matériels différents.
B. Les boîtiers de tests permettent de visualiser le programme ou les valeurs des paramètres
C. Le module d'alimentation délivre des sources de tension dont l'automate a besoin : 24V, 12V ou 5V en alternatifs.
D. Les cartes de sortie analogiques assurent la conversion analogique / numérique
E. La mémoire de données est une mémoire vive (RAM)
F. La mémoire système est une mémoire à lecture seul programmables effaçable électriquement.
G. un Compteur Programme ou Compteur Ordinal qui contient l'adresse de l'instruction qui est en court d'exécution

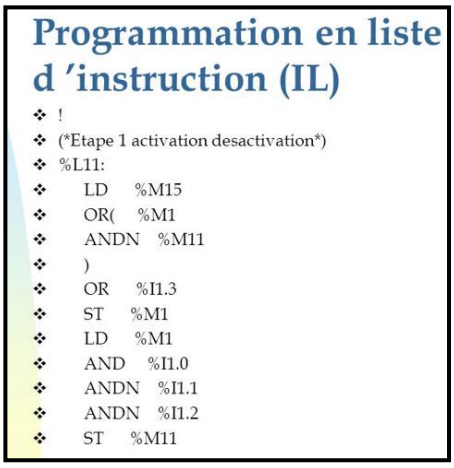
WWW.DZPEN.COM | قلم الجزائر

5.3 Langage de programmation

Il existe 4 langages de programmation des automates qui sont normalisés au plan mondial par la normeCEI 61131-3.Chaque automate se programmant via une console de programmation propriétaire ou par un ordinateuréquipé du logiciel constructeur spécifique.

5.3.1 Liste d'instructions*(IL : Instruction list) :* Langage textuel de même nature que l'assembleur (programmation des microcontrôleurs). Très peu utilisé par les automaticiens.

***Figure*113 :**Programmation en liste d'instruction (IL)



https://slideplayer.fr/slide/185210/

***5.3.2* Langage littéral structuré** *(ST : StructuredText) :* Langage informatique de même nature que lePascal, il utilise les fonctions comme *if ... then ... else ...* (si ... alors ... sinon ...)Peu utilisé par les automaticiens.

***Figure* 114 :**Programmation en Langage littéral structuré(ST)

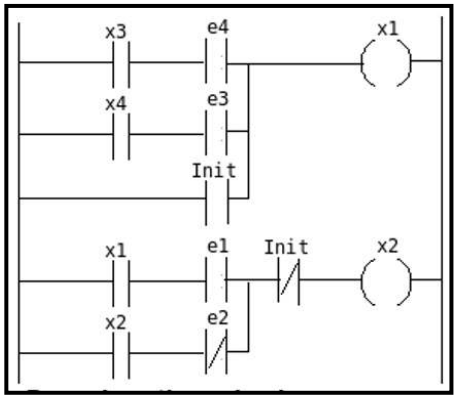


https://www.automation-sense.com/blog/automatisme/tutoriel-language-structured-text-ou-texte-structure-st.html

WWW.DZPEN.COM | قلم الجزائر

5.3.3 Langage à contacts *(LD : Ladderdiagram) :* Langage graphique développé pour lesélectriciens. Il utilise les symboles tels que :contacts, relais et blocs fonctionnels et s'organiseen réseaux (labels).C'est le plus utilisé.

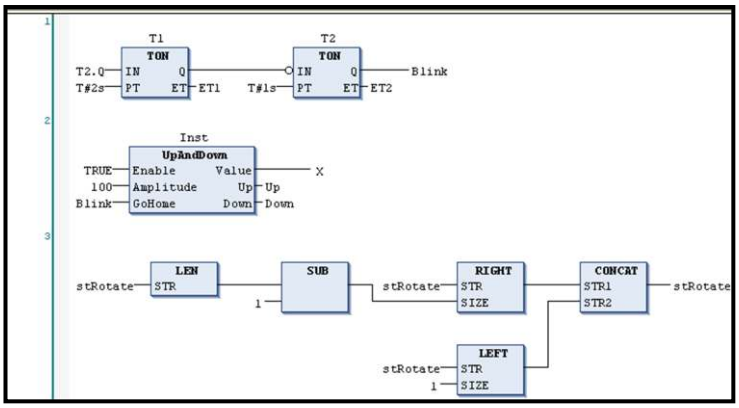
***Figure*115:**Programmation en langage ladder (LD)



Site :
Wikiversitéhttps://fr.wikiversity.org/wiki/Automatisme_:_norme_1131-3_et_GRAFCET/Retour_sur_le_graphe_d%27%C3%A9tat_%E2%80%93_Langage_LD_de_la_norme_1131-3

5.3.4 Blocs Fonctionnels *(FBD : Function Bloc Diagram) :* Langage graphique ou des fonctions sont représentéespar des rectangles avec les entrées à gauche et lessorties à droites. Les blocs sont programmés(bibliothèque) ou programmables.Utilisé par les automaticiens.

***Figure*116:**Programmation en langage Blocs Fonctionnels (FBD)

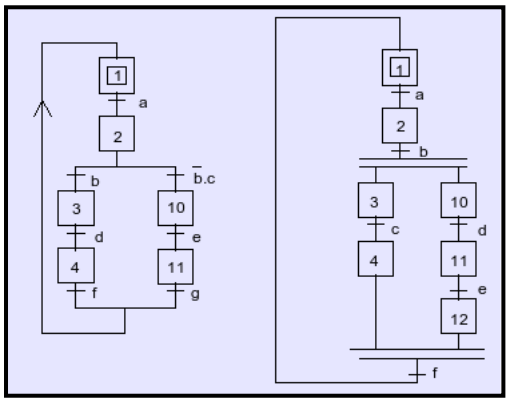


Site : Schneider Electric
https://product-help.schneider-electric.com/Machine%20Expert/V1.1/fr/SoMProg/SoMProg/FBD_LD_IL_Editor/FBD_LD_IL_Editor-4.htm

WWW.DZPEN.COM | قلم الجزائر

5.3.5 Langage GRAFCETou SFC(SequentialFunction Chart) est un langage graphique utilisé pour décrire les opérations séquentielles. Le procédé est représenté comme une suite connue d'étapes (états stables), reliées entre elles par des transitions, une condition booléenne est attachée à chaque transition

***Figure* 117:**Programmation en langage ladder (LD)



Site : Wikiversité

https://fr.wikiversity.org/wiki/Automatisme_:_norme_1131-3_et_GRAFCET/Exercices/Le_langage_SFC_de_la_norme_1131-3

Parmi ces langages, deux sont **textuels**, le texte structure **(ST)** et les listes d'instruction **(IL),** et serontsaisis sur l'appareil de programmation à partir d'un clavier, une ligne à la fois.

Les autres, le langage a contacts**(LD)**, les graphes de fonction séquentielle**(SFC)** et les diagrammes de schémasfonctionnels**(FBD)**, sont **graphique**s et un programme peut donc être développé à partir d'élémentsgraphiques sur l'écran de I 'appareil de programmation.

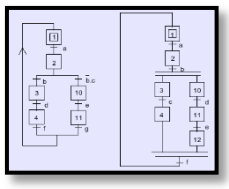
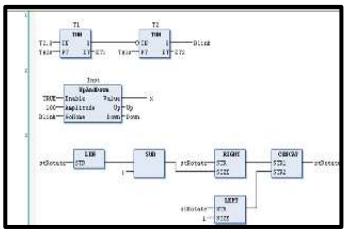
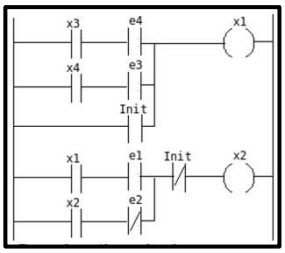
Pour programmer l'automate, l'automaticien peut utiliser une console de programmation (portable) ou un PC doté d'un logiciel de programmation (exemple : Step 7 pour les API Siemens ou PL7 pour Schneider).

Pour envoyer le programme réalisé dans la mémoire de l'automate on utilise une liaison série entre l'automate et l'ordinateur ou un câble spécifique lors de l'utilisation d'une console.

WWW.DZPEN.COM | قلم الجزائر

5.3.6 Exercice21 :Remplir le tableau de la figure118

Tableau 118: Tableau récapitulatif des langages de programmation des

Exemple de langage de programmation	Le Nom du langage	Langage Graphique ou Textuel
		
		
		
		
		

WWW.DZPEN.COM | قلم الجزائر

5.4 Résumé Dans ce chapitre le plus important à retenir :

- Un automate programmable industriel est une machine électronique, programmable destiné à commander au moyen de signaux d'entrées et de sorties, et d'un programme informatique, en temps réel, des procédés de processus industriels par un traitement séquentiel.
- Il existe deux technologies des automates programmables industriels :
  1. Automates programmables industriels compact qui intègre le processeur, l'alimentation, les entrées et les sorties dans un seul boitier (***rack***) qui sont généralement destinés à la commande de petits automatismes
  2. Dans un automate programmableindustriel modulaire le processeur, l'alimentation et les interfaces d'entrées / sorties résident dans des unitésséparées (modules) et sont fixées sur un ou plusieurs racks.
- Les automates programmables industriels, avec leurs solutions programmées, présentent de nombreux avantages par rapport à la technologie de logique câblée
- On utilise les API dans tous les secteurs industriels pour la commande des machines (convoyage, emballage ...) ou des chaînes de production (automobile, agroalimentaire, ...) Il est de plus en plus utilisé dans le domaine du bâtiment pour le contrôle du chauffage, de l'éclairage, de la sécurité ou des alarmes.
- Un automate programmable est constitué essentiellement de 5modules :
- Unité centrale (processeur, mémoires)
  1. module d'entrées (cartes d'entrées logiques, cartes d'entrées analogiques)
  2. module de sorties (cartes de sorties logiques, cartes de sortie analogiques)
  3. module d'alimentation
  4. module de communication (consoles, boîtiers de tests, unités de dialogue en ligne)
- Il existe 4 langages de programmation des automatesprogrammables industriels, 2 langages textuels (langage liste d'instructions IL,langage littéral structuréST) et 3 langages graphique(Langage à contacts LD, langageblocs fonctionnelsFBD,langage GRAFCET) .

WWW.DZPEN.COM | قلم الجزائر

5.5 Exercice de synthèse5

- **Spécialité** : Domotique
- **Module** : Logique combinatoire et logique séquentielle
- **Durée :** 1h30
- **But :** Familiariser les stagiaires avec les principaux éléments d'un automate programmable
- **Matériel requis** : Documents et support de cours
- **Mise en situation :** identifier les éléments qui constituent l'architecture externe et interne d'un automate programmable
- **Marche à suivre :**

Aspect extérieur

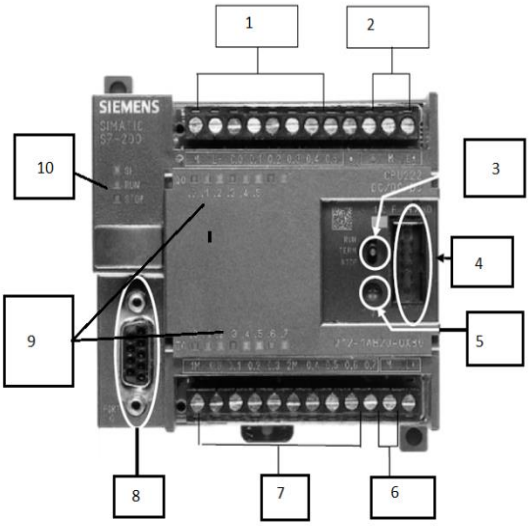
1- Compact et modulaire, Répondez par vrai ou faux:

A. Le S7-200 CPU222 est un API compact.
B. Le TSX 3721 est un API compact.
C. L'appellation " Modulaire" est due au mot " Mode".
D. Dans un API modulaire les parties processeur, alimentation et interface entrée/sortie résident dans des unités séparées.
E. L'information délivrée par un capteur est une information analogique.

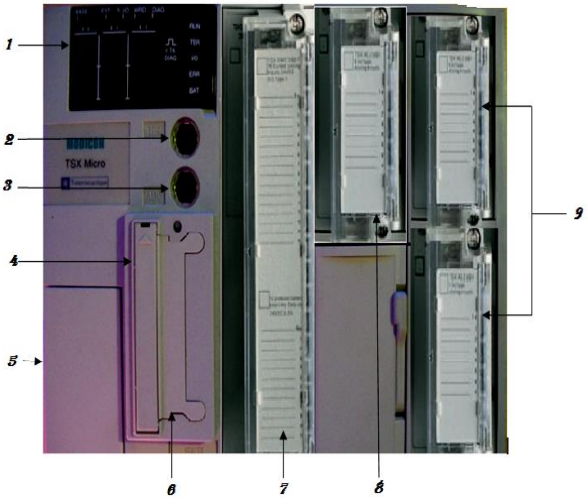
WWW.DZPEN.COM | قلم الجزائر

2- Identifiez l'emplacement des éléments d'un API:

Identifier les différentselements sur un API enreportant les repèrescorrespondantsdans le tableau ci-dessous.	
Alimentation de l'automate.	
Bornes des entrées de l'automate.	
Bornes de sorties des automates.	
Sélecteur de mode STOP, RUN	
Connecteur pour modules d'extinction	
Interface de programmation.	
Bornierspermettantd'alimenter les capteurs pour les entrées.	
Visualisation des états des entrées et des sorties del'automate .	
Visualisation de l'état de l'automate .	



Emplacement pour carte d'extensionmémoire.	
Trappe d'accès aux bornesd'alimentation	
Emplacement pour coupleur decommunication	
Bloc de visualisation (affiche des informations sur l'état del'automate)	
Prise TERMINAL (TER) permet de connecté le terminal de programmation	
Prise de dialogue opérateur (AUX) permet de connecté un pupitre de dialogue	



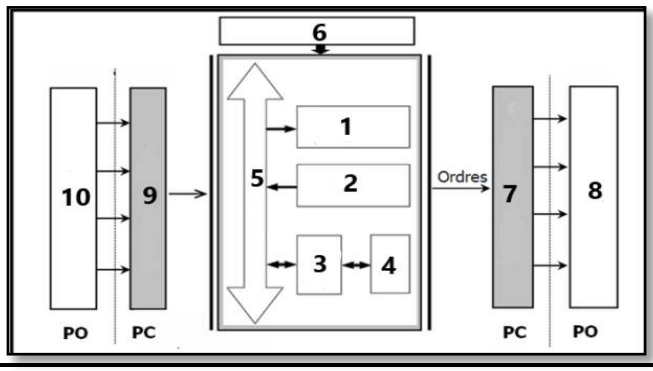
Emplacement pour module de sortie dans le module de base de type TOR sous la référence TSX DSZ08R5.	
Emplacement pour module d'entréedans le module de base de type TOR sous la référence TSX DEZ 32D2.	
Emplacement pour module d'entréedansle module de base de type analogique sous la référence TSX AEZ 4144multirage(TC.RTD).	

WWW.DZPEN.COM | قلم الجزائر

Aspect intérieur

1. Identifiez l'architecture d'un API.

Capteurs	
Unité de traitement	
Interface d'entrées	
RAM	
Bus d'entrées/ sorties	
Modules de sorties	
Alimentation CA/CC	
Préactionneurs	
Interface de sorties	
Modules d'entrées	



2. Identifier le rôle des différentes parties de l'API.

Le Microprocesseur		Reçoit les informations issues des capteursd'entrées et les informationsgénérées par le processeur et destinées à la commande des sorties
Module d'entrées		Gèrel'ensemble des échangesinformationnelsenassurant, la lecture des informationsd'entrée, l'exécution des instructions du programmemisenmémoire et la commandeoul'écriture des sorties.
Mémoire		Permet à l'Unité Centrale de l'automate, d'effectuerune "lecture" de l'étatlogique des capteurs qui luisontassocies.

3. Mémoires

	Sinificationenanglais	Sinification enfrancais
ROM		
RAM		
PROM		
EPROM		
EEPROM		
CMOS		

4. Mettre le type de mémoire correspondant: ROM, RAM,…

Mémoire du systèmed'exploitation	
Mémoire du programme	
Mémoire des donnéeslors du fonctionnement	

WWW.DZPEN.COM | قلم الجزائر

RÉSUMÉ GÉNÉRAL

WWW.DZPEN.COM | قلم الجزائر

On vient de terminer l'étude sur la logique combinatoire et la logique séquentielle. Il apparaît intéressant de récapituler les points suivants.

Dans le 1er chapitre on a constaté qu'on plus dessystèmes de numérotation décimale, il existe trois autres systèmes de numérotions couramment utilisés dans les dispositifs industriels, ce sont le système binaire, le système octal, et le système hexadécimal. Ces trois systèmes sont des systèmes pondérés où le poids de pondération de chaque chiffre dépend de son rang dans le nombre, nous avons passé aussi en revue les codes numériques les plus fréquemment utilisés et avons appris à les manipuler.

Concernant le chapitre 2 nous avons fait connaissance avec les notions de base de l'algèbre de Boole : variables et fonctions Booléennes, opérations booléennes de base, table de vérité et propriétés des opérations logiques de base.Cette activité nous a permis de faire connaissance avec les bases de l'algèbre logique utilisée dans les circuits numériques, ces connaissances nous ont permis d'effectuer la synthèse des circuits logiques qui sont les composants de bases des systèmes numériques .
Dans ce chapitre on a aussi appris à faire la synthèse des fonctions logiques à partir de leurs tables de vérité et leurs expressions analytiques, à simplifier les expressions booléennes en utilisant la méthode algébrique ou la méthode de Karnaugh.

Dans cette partie nous avons abordébrièvement la famille TTL et la famille CMOS. Nous avons vu leurs compositions, leurs caractéristiques de base, leurs avantages et leurs inconvénients. Ce qui va nous permettre de choisir la technologie la plus adaptée aux systèmes qu'on veut construire.

Le chapitre 3, nous a permis de connaître la structure et le fonctionnement des circuits de base pour les traitements dans les systèmes numériques que sont les décodeurs, multiplexeurs, additionneurs, comparateurs. Elle a permis aussi de connaître la méthodologie de conception des circuits logiques de ces dispositifs.

Le chapitre 4, nous a permis de connaître la structure et le fonctionnement des circuits séquentiels de base pour les traitements dans les systèmes numériques que sont les bascules, registres et compteurs. Elle a permis aussi de connaître la méthodologie de conception des circuits logiques de ces dispositifs.
Ces dernières années, les avancées technologiques ont conduit au développement des automates programmables industriels (API) et à une révolution importante dans l'automatique.
Ce dernier chapitre constitue une introduction aux automates programmables industriels (API), ainsi qu'à leur fonction générale, leurs formes matérielles et leur architecture interne. Les API sont employés dans de nombreuses tâches d'automatisation, dans différents domaines, comme les processus de fabrication industriels.

WWW.DZPEN.COM | قلم الجزائر

ACTIVITÉ DE SYNTHÈSE

WWW.DZPEN.COM | قلم الجزائر

- **Spécialité** : Domotique
- **Module** : Logique combinatoire et logique séquentielle
- **Durée :** 3h
- **But :**
  - ✓ Appliquer les concepts de la logique combinatoire en utilisant les notions de variables et d'opérations logiques.
  - ✓ Représenter la logique combinatoire par des tables de vérité, l'algèbre de Boole et les tables de Karnaugh
  - ✓ Concevoir et utiliser des circuits combinatoires simples tels que (additionneurs, multiplexeur, décodeurs) et des circuits séquentiels tels que (les bascules, les compteurs et les registres à décalage).
  - ✓ Avoir des notions générales sur les automates programmables
- **Matériel requis** : Documents , supports de cours , simulateur électronique .
- **Mise en situation :**Réaliser les conversions et opérations suivantes
- **Marche à suivre :**

1. Algèbre de Boole

Distributeur de boissons chaudes

- ✓ Un distributeur de boissons chaudes permet de distribuer du café ou du thé, avec ou sans lait, ou du lait seul.
- ✓ Trois boutons permettent de commander le distributeur : « café », « thé », « lait ». Pour obtenir l'une de ces boissons seule, il suffit d'appuyer sur le bouton correspondant. Pour obtenir une boisson avec lait, il faut appuyer en même temps sur le bouton correspondant à la boisson choisie et sur le bouton « lait ».
- ✓ De plus, le distributeur ne fonctionne que si un jeton a préalablement été introduit dans la fente de l'appareil. Une fausse manœuvre après introduction du jeton (par exemple, appui simultané sur « café » et « thé ») provoque la restitution du jeton. Le lait étant gratuit, le jeton est également restitué si du lait seul est choisi.
- ✓ Calculer et simplifier les fonctions de restitution du jeton, J, de distribution du café, C, du thé T, et du lait, L. On notera que la fonction de restitution du jeton peut indifféremment être active ou non lorsque aucun jeton n'est introduit dans l'appareil

2. Circuits combinatoires

Demi-Soustracteur.

Réaliser un demi-soustracteur :

1. Ecrire la table de vérité.
2. Donner les équations de sortie.
3. Etablir le schéma logique.

WWW.DZPEN.COM | قلم الجزائر

Soustracteur complet

On veut réaliser un circuit qui effectue la soustraction Ai - Bi en tenant compte d'une éventuelle retenue R i-1. Ce circuit doit donc générer la différence Di et l'éventuelle retenue Ri à transmettre à la colonne de gauche.

1. Remplir la table de vérité de Di et Ri.
2. Remplir les tableaux de Karnaugh et en déduire les équations simplifiées de Di et Ri.
3. Dessiner le schéma de ces deux fonctions réunies en un seul bloc fonctionnel : le soustracteur complet.
4. Réaliser un soustracteur binaire complet (ou étage de soustracteur) selon deux modes :
   a. Avec deux demi-soustracteurs ;
   b. Avec un demi-additionneur et un demi-soustracteur.
5. Dessiner le schéma d'un soustracteur de 2 nombres de 4 bits en utilisant 4 blocs fonctionnels identiques.

Additionneur Soustracteur

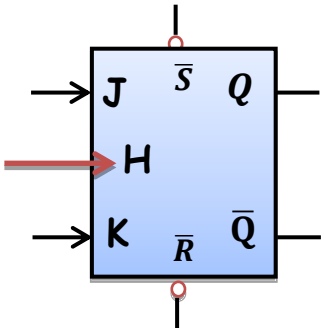
1. Réaliser un circuit qui inverse ou non l'état d'une entrée E selon qu'un bit de commande C est à 1 ou à 0: si C= 0 on veut S= E, si C= 1 on veut S = $\overline{E}$
2. En utilisant cette fonction et un additionneur sur 4 bits, réaliser un circuit qui effectue l'addition de deux nombres de 4 bits (A + B) si un bit de commande C est à 0 et la soustraction (A – B) si C = 1.

3. Circuits séquentiels

On dispose de bascules JK synchronisées sur front montant de la figure119. Chaque bascule possède des entrées asynchrones prioritaires actives à l'état bas : ***set (S)*** **et** ***reset (R)***.

1. Réalisez un compteur asynchrone modulo 16.
2. Modifiez le montage pour en faire un compteur asynchrone modulo 12.
3. En partant de zéro, tracez son chronogramme sur un cycle complet.
4. Ajoutez un interrupteur automatique de remise à zéro à l'allumage.
5. Ajoutez un interrupteur manuel de remise à zéro.
6. Que suffit-il de faire pour remplacer les bascules JK par des bascules D

Figure119 : Bascule JK



WWW.DZPEN.COM | قلم الجزائر

BIBLIOGRAPHIE

WWW.DZPEN.COM | قلم الجزائر

- **BELAYACHI, Naima.** Cours et Exercices,Architecture des Ordinateurs 1.Département des Sciences Exactes, Filière d'Informatique,Ecole Normale Supérieure d'Oran (ENSO).
- **BENAMARA Mahmoud .GAALOUL**: 2015/2016. Département de Génie Electrique,Institut Supérieur des Etudes, Technologiques de Nabeul.
- **BENAMARA Mahmoud .GAALOUL Kamel.**Support de cours, Systèmes logiques 2, Logique séquentielle. Année universitaire: 2015/2016. Département de Génie Electrique,Institut Supérieur des Etudes, Technologiques de Nabeul.
- **CARIOU**, **Éric**.Cours, Algèbre de Boole, P29. Université de Pau et des Pays de l'Adour, Département Informatique.
- **CHRISTOPHE, Adessi.** Cours Electronique Numérique 2016-2017, Audioprothèse 2eme Année. Université Claude Bernard Lyon1.
- **C,Belleudy . D, Gaff.** Cours Premier semestre.Electronique Numérique.Université de Nice-Sophia, Antipolis, DEUG Première année.
- **DIOU, Camille**. Maître de Conférences,Cours, P187.D'électronique numérique. Laboratoire Interfaces Capteurs et Microélectronique. Université Paul Verlaine–Metz.
- **DOUILLARD, Catherine. THEPAUT, André.** Polycopié 1/2. Logique combinatoire et circuits MOS.**Année scolaire 2008-2009.**Telecom Bretagne, France
- **GUILLAUME, Blin.** Cours, Algèbre de Boole et circuits logiques, Architecture des ordinateurs. Université de Marne La Vallée, France
- **JEZEQUEL,Michel.** Cours 2, Circuits combinatoires. Département d'électronique .Telecom Bretagne, France.
- **KARA, Abdelaziz.**Cours, Les Circuits Combinatoires.1er année Tronc Commun Mathématique Informatique, 2018/19. Faculté des Sciences, Université Sétif 1.
- **KONATE**, **Karim.** Electroniquenumérique.Informatique appliquée: csi 2201. Université virtuelle africaine
- **LE BRUN,Philippe.**Cours,Electronique numérique.Lycée Louis ARMAND,Strasbourg, France.

WWW.DZPEN.COM | قلم الجزائر

- **MEZAACHE, Salah Eddine.** Architecture des ordinateurs et programmation,2009/2010. P42.Département d'électronique, centre universitaire de bordj bouarréridj .
- **PIERRE, Mayé .**Aide-mémoire. Composants électroniques. 3e édition, L'usine Nouvelle, DUNOD.
- **TRABELSI, Hichem.** Cours,Circuits logiques combinatoires, Systèmes de numération et codes. Université virtuelle de Tunis.

- **TRABELSI,Hichem.** Cours, Circuits logiques séquentiels**,** Bascules bistables.Université Virtuelle de Tunis.
- **WEBER,Jacques . MEAUDRE, Maurice.**Circuits numériques et synthèse logique, un outil : VHDL. L'IUT de CACHAN, Université de Paris.

SITES WEB

- http://www.composelec.com/index.php « Composant électronique / Composant actif / Composant passif »
- https://www.electronique-et-informatique.fr « Bureau d'étude Electronique Informatique », Cree le 12 /06/2019 ,misà jour le 02/01/2020 .
- https://mrproof.blogspot.com«Bibliothèque de la programmation ».
- https://f2school.com/electronique-numerique/ « Bibliothèque en ligne F2School »
- https://openclassrooms.com « Se former en alternance avec OpenClassrooms »
- https://www.technologuepro.com/ « Technologue pro - Ressources pédagogiques pour l'enseignement technologique »
- https://f2school.com/electronique-numerique/ « Bibliothèque en ligne F2School »Bibliothèque en ligne

WWW.DZPEN.COM | قلم الجزائر

ANNEXE 1

CORRIGES DES EXERCICES

WWW.DZPEN.COM | قلم الجزائر

Exercice 1

Quelles sont parmi ces suites de chiffres celles qui peuvent être la représentation d'un nombre en base 16, 8 ou 2 ?

	Binaire (Base 2)	Octal (Base 8)	Hexadécimal (Base 16)
1001010210	Faux	Vrai	Vrai
1001011	Vrai	Vrai	Vrai
1200111201	Faux	Vrai	Vrai
1431901	Faux	Faux	Vrai
2A21F	Faux	Faux	Vrai
9GF28	Faux	Faux	Faux
1789012	Faux	Faux	Vrai

Exercice 2

Complétez le tableau suivant, puis vérifier les résultats en faisant l'opération inverse

Base	Nombre	Equivalent	Dans la base
10	45	101101	2
10	57	71	8
8	75	3D	16
2	1101,101	13,5	10
10	42,0625	52,04	8
16	ABC, A	5274,50	8
4	1320	78	16

Exercice 3

A. Les opérations en binaire sur 8 bits

- $(9)_{10}+(8)_{10}$

$$\begin{array}{rl} & 0\ 0\ 0\ {}_{1}0\ 1\ \ 00\ 1 \\ +\ & 0\ 0\ 0\ \ 01\ \ 0\ 0\ 0 \\ \hline = & \ \ 00\ \ 010\ \ 0\ \ 0\ 1 \end{array} \qquad \begin{array}{l} (9)_{10} \\ +(8)_{10} \\ \hline = (17)_{10} \end{array}$$

- $(45)_{10} - (41)_{10}$

$$\begin{array}{rl} & 0\,010\ 1\ 10\ 1 \\ -\ & 0\ 0\ \ 1\ 01\ 0\ 0\ 1 \\ \hline = & \ \ 0\ 0\ \ 0\ \ 0\ \ 0\ \ 1\ \ 0\ 0 \end{array} \qquad \begin{array}{l} (45)_{10} \\ _(41)_{10} \\ \hline = (04)_{10} \end{array}$$

WWW.DZPEN.COM | قلم الجزائر

➢ $(56)_{10}$ - $(4)_{10}$

$$\begin{array}{r} 0\,0\,1\;1\,1_1\,0\;\;0\,0 \\ -\;\;0\;\;0\;\;0\;\;0_1\,0\,1\;\;0\;\;0 \\ \hline =\;\;0\,0\;\;1\;\;1\;\;0\;\;1\;\;0\;\;0 \end{array} \qquad \begin{array}{r} (56)_{10} \\ _\,(04)_{10} \\ \hline = (52)_{10} \end{array}$$

B. Les soustractions en binaire sur 8 bits

❖ La méthode directe puis celle qui utilise le complément a 2

➢ **$(42)_{10} - (29)_{10}$**

$$\begin{array}{r} 0\,0\,1_1\,0_1\,1\,1_0\;\;1_1\,0 \\ -\;\;0\;\;0\;\;1_0\,1_1\,1\,1_1\,0\;\;1 \\ \hline =\;\;0\,0\;\;0\;\;0\;\;1\;\;1\;\;0\;\;1 \end{array} \qquad \begin{array}{r} (42)_{10} \\ _(29)_{10} \\ \hline = (13)_{10} \end{array}$$

➢ **$(25)_{10} - (9)_{10}$**

$$\begin{array}{r} 0\,0\,0\;1\,1\,0\,0\,1 \\ -\;\;0\;\;0\;\;0\,0\,1\,0\,0\;\;1 \\ \hline =\;\;0\,0\;\;0\;\;1\;\;0\,0\;\;0\,0 \end{array} \qquad \begin{array}{r} (25)_{10} \\ _(09)_{10} \\ \hline = (16)_{10} \end{array}$$

➢ **$(12)_{10} - (52)_{10}$**

$$\begin{array}{r} 1_0\,1_0\,1_0\,1_0\,1\,1\;\;0\,0 \\ -1_0\;\;1_0\;\;1_1\,1\,0\,1\;\;0\;\;0 \\ \hline =\;\mathbf{1}\;\;1\;\;1\;\;0\;\;1\;\;1\;\;0\;\;0\;\;0 \end{array} \qquad \begin{array}{r} (12)_{10} \\ _(52)_{10} \\ \hline = (-40)_{10} \end{array}$$

➢ **Remarque** :

- Le résultat est sur 8bits, c'est-à-dire que le 9ieme bit n'est pas pris en charge
- Le résultat est négatif, pour connaitre le nombre on doit faire le Cà2 du résultat

  Cà2 $(11011000)_2 = (00101000)_2 = (40)_{10}$

  $\Rightarrow (11011000)_2 = (-40)_{10}$

WWW.DZPEN.COM | قلم الجزائر

❖ La méthode qui utilise le complément à 2

➢ **(42) 10 – (29) 10** = (42) 10+ (–29)10

(42) 10 =(00101010)2

(29)10 = (00011101)2

(–29)10 = Cà2(00011101)2 = (11100011)2

(42) 10+ (–29)10 =(00101010)2 + (11100011)2

$_1 0_1 0101\ _1 0\ 10$ $(42)_{10}$
\+ 1 1 10 0 0 11 _(29)10

= 1 0 0 0 1 1 0 1 = (13)10

➢ **(25) 10 – (9) 10** = (25) 10+ (–9)10

(25) 10 =(00011001)2

(9)10= (00001001)2

(-9)10= Cà2 (00001001)2 = (11110111)2

(25) 10+ (–9)10 = (00011001)2 + (11110111)2

$_1 0_1 0\ _1 0_1 1_1 1_1 0_1 0\ 1$ $(25)_{10}$
\+ 1 1 1 10 1 11 _(09)10

= 1 0 0 0 1 0 0 0 0 = (16)10

➢ **(12) 10 – (52) 10**

(12) 10 =(00001100)2

(52) 10= (00110100)2

(-52) 10= Cà2 (00110100)2 = (11001100)2

(12) 10 + (–52) 10 = (00001100)2 + (11001100)2

$_1 0_1 0\ 0_1 0_1 11\ \ 00$ $(12)_{10}$
\+ 1 $_1$1 0 011 0 0 _(52)10

= 1 1 1 0 1 1 0 0 0 = (- 40)10

WWW.DZPEN.COM | قلم الجزائر

Exercice 04 Complétez le tableau suivant

Code/Base	Nombre	Equivalent	Dans le Base/Code
2	111011101	100110011	Gray
10	7	1010	XS3
2	110010	0101 0000	DCB
8	17	001000	Gray
Gray	110010001010	100011110011	2
Gray	1011	1110	XS3

Exercice de synthèse1

A. Complétez le tableau suivant

Base10	Base 2	Base 8	Base16
125	**1111101**	**175**	**7D**
61645	**1111000011001101**	**170315**	**F0CD**
11,75	**1011,11**	**13,6**	**B,C**
85,5	**1010101,100**	**125,4**	**55,8**

B. Complétez le tableau suivant

Base	Nombre	Equivalent	Dans la base
10	121,875	1111001,111	2
16	A01	0010 0101 0110 0001	DCB
DCB	10010110	1010000	Gray
8	25,45	15,94	16

C. Complétez le tableau suivant

Nombre	Complément à 1	Complément à 2
A=10000100	**0111 1011**	**0111 1100**
B=01110001	**1000 1110**	**1000 1111**

D. Calculer en Binaire sur 8 bits

A+B	**1111 0101**
A-B	**0001 0011**
B-A	**1110 1101**
-A-B	**0000 1011**

WWW.DZPEN.COM | قلم الجزائر

Exercice 05 Table de vérité du circuit électronique de la figure 05

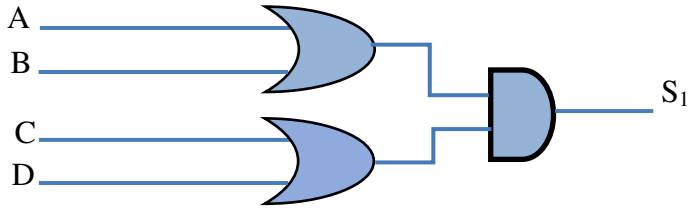
Table de vérité

K_1	K_2	L
0	0	**0**
0	1	**1**
1	0	**1**
1	1	**1**

Exercice 06 Circuits logiques des fonctions S_1 et S_2

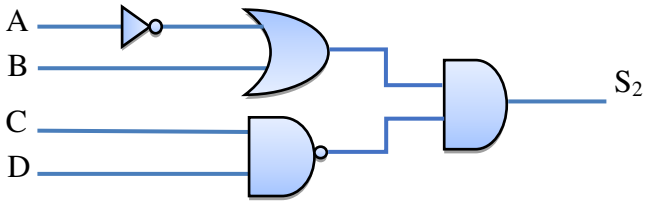
S1 = (A + B). (C + D)

Figure 120: Circuit logique de S_1



S2 = $(\bar{A} + B).\ \overline{C.D}$

Figure 121: Circuit logique de S_2



Exercice 7:

A. La table de vérité de la fonction :F1 = X.Y.Z +X .(Y + $\bar{Z}$) = X.Y.Z + X.Y + X.$\bar{Z}$

Table de vérité

X	Y	Z	F
0	0	0	**0**
0	0	1	**0**
0	1	0	**0**
0	1	1	**0**
1	0	0	**1**
1	0	1	**0**
1	1	0	**1**
1	1	1	**1**

X.$\bar{Z}$

X.Y

X.Y.Z

WWW.DZPEN.COM | قلم الجزائر

B. La deuxième forme canonique de F_1

Table de vérité

X	Y	Z	F	
0	0	0	**0**	$X+Y+Z$
0	0	1	**0**	$X+Y+\overline{Z}$
0	1	0	**0**	$X+\overline{Y}+Z$
0	1	1	**0**	$X+\overline{Y}+\overline{Z}$
1	0	0	**1**	
1	0	1	**0**	$\overline{X}+Y+\overline{Z}$
1	1	0	**1**	
1	1	1	**1**	

$$F=(X+Y+Z)\ +(X+Y+\overline{Z})+(X+\overline{Y}+Z)\ (X+Y+\overline{Z})\ +(\overline{X}+Y+\overline{Z})$$

Exercice 8

a) Table de vérité

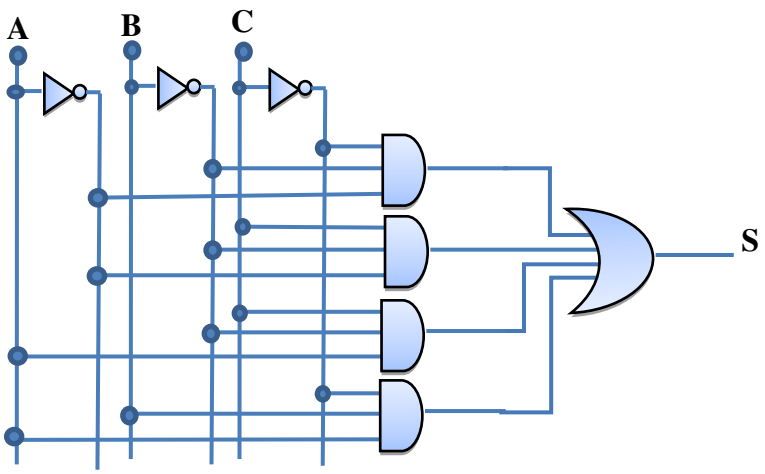
A	B	C	S	
0	0	0	**1**	$\overline{A}.\overline{B}.\overline{C}$
0	0	1	**1**	$\overline{A}.\overline{B}.C$
0	1	0	**0**	
0	1	1	**0**	
1	0	0	**0**	
1	0	1	**1**	$A.\overline{B}.C$
1	1	0	**1**	$A.B.\overline{C}$
1	1	1	**0**	

b) La fonction S

$$S=\overline{A}.\overline{B}.\overline{C}\ +\overline{A}.\overline{B}.C+A.\overline{B}.C+A.B.\overline{C}$$

c) Circuit logique de S

Figure 122: Circuit logique de S



WWW.DZPEN.COM | قلم الجزائر

Exercice 9

Réalisation de la fonction F= ($\overline{A}$+B).C en utilisant lescircuits intégrer

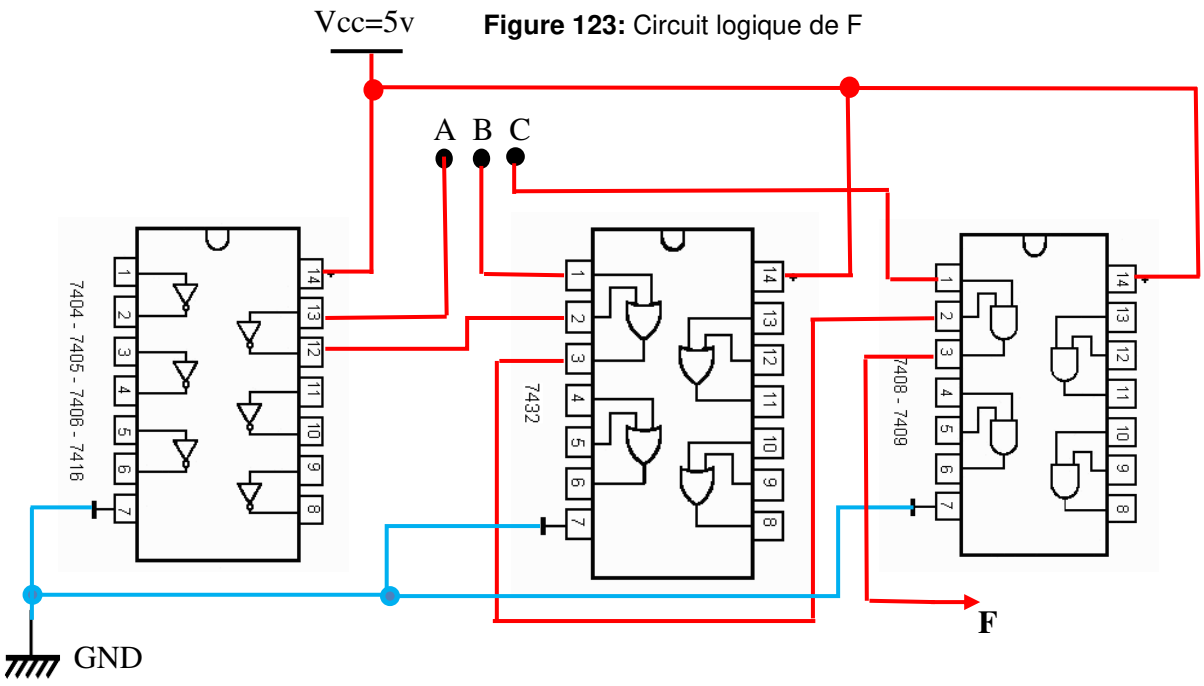


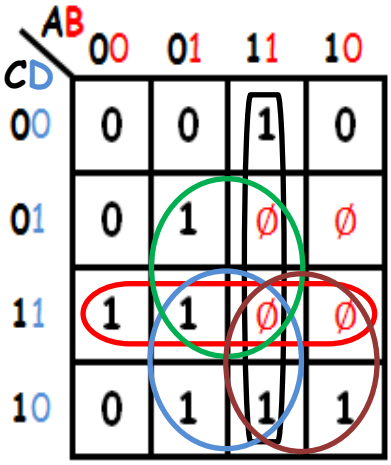
Figure 123: Circuit logique de F

Exercice 10

❖ Table de vérité

	A	B	C	D	S
0	0	0	0	0	**0**
1	0	0	0	1	**0**
2	0	0	1	0	**0**
3	0	0	1	1	**1**
4	0	1	0	0	**0**
5	0	1	0	1	**1**
6	0	1	1	0	**1**
7	0	1	1	1	**1**
8	1	0	0	0	**0**
9	1	0	0	1	ø
10	1	0	1	0	**1**
11	1	0	1	1	ø
12	1	1	0	0	**1**
13	1	1	0	1	ø
14	1	1	1	0	**1**
15	1	1	1	1	ø

B. Tableau de KARNAUGH

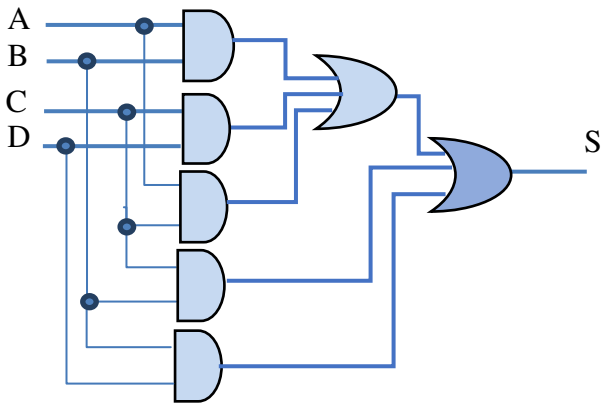


S= A.B + C.D + B.D + A.C + B.C

WWW.DZPEN.COM | قلم الجزائر

C. Circuit logique

Figure 124: Circuit logique de S



Exercice de synthèse 2

➢ **Reponse1** : Établir la table de vérité de la fonction suivante, puis l'écrire sous la 1ère forme canonique

$F_1 = (X + Y).(\bar{X} + Y + Z)$

X	Y	Z	F_1	
0	0	0	**0**	X+Y
0	0	1	**0**	
0	1	0	1	$\bar{X}.Y.\bar{Z}$
0	1	1	1	$\bar{X}.Y.Z$
1	0	0	**0**	$\bar{X}+Y+Z$
1	0	1	1	$X.\bar{Y}.Z$
1	1	0	1	$X.Y.\bar{Z}$
1	1	1	1	X.Y.Z

F_1sous la 1er forme canonique$F1 = \bar{X}.Y.\bar{Z} + \bar{X}.Y.Z + X.\bar{Y}.Z + X.Y.\bar{Z} + X.Y.Z$

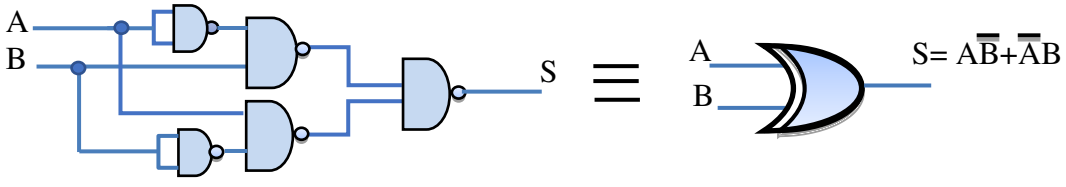
➢ **Réponse 2** : Réalisation de le porte EXOR a l'aide de portes NAND uniquement

✓ Equation de la porte EXOR $S = A.\bar{B}+\bar{A}.B$

$$S = \overline{A\bar{B}+\bar{A}B} = \overline{A\bar{B}} \,.\, \overline{\bar{A}B}$$

✓ Logigramme

Figure 125: Circuit logique de la porte EXOR a l'aide de portes NAND



WWW.DZPEN.COM | قلم الجزائر

➢ **Réponse 3** : Simplifier algébriquement la fonction suivante

$\mathbf{F_2}= (X + Y + Z).(\bar{X}+Y+Z)+XY+YZ$

$\mathbf{F_2}=X\bar{X}+XY+XZ+Y\bar{X}+YY+YZ+Z\bar{X}+ZY+ZZ+XY+YZ$ Sachant que $X\bar{X}=0$, $YY=Y$, $ZZ=Z$

$\mathbf{F_2}=XY+XZ+Y\bar{X}+Y+YZ+Z\bar{X}+ZY+Z+XY+YZ$ Sachant que $YZ+ZY+YZ=YZ$, $XY+XY=XY$

$\mathbf{F_2}=XY+XZ+Y\bar{X}+Y+YZ+Z\bar{X}+Z$

$\mathbf{F_2}=Y(X+\bar{X}+1+Z)+Z(\bar{X}+1)$ Sachant que $(X+\bar{X}+1+Z)=1$, $(X+1)=1$

$\mathbf{F_2 = Y + Z}$

➢ **Réponse 4** : Soit la fonction booléenne suivante :

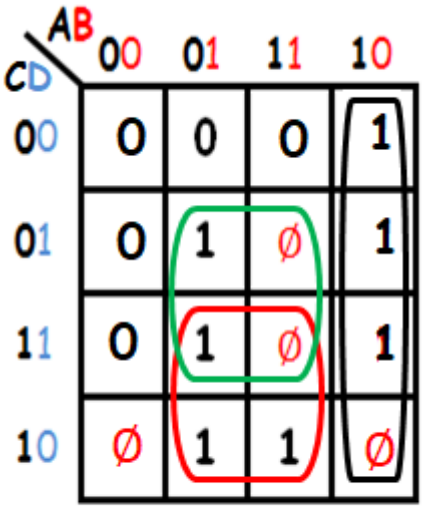
$F(A, B,C,D) = A\overline{B}\,\overline{C}\,\overline{D} + \overline{A}\,B\overline{C}\,D + A\overline{B}\,CD + A\overline{B}\,\overline{C}\,D + \overline{A}\,BC\,D + \overline{A}\,BC\overline{D} + ABC\overline{D}$

C. Simplification de F par la méthode des diagrammes de Karnaugh, sachantque quatre combinaisons de variables sont impossibles : $AB\overline{C}D$, $ABCD$, $\overline{A}\,B\overline{C}\overline{D}$, et $A\overline{B}\,C\overline{D}$

❖ Table de vérité

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

B. Tableau de KARNAUGH

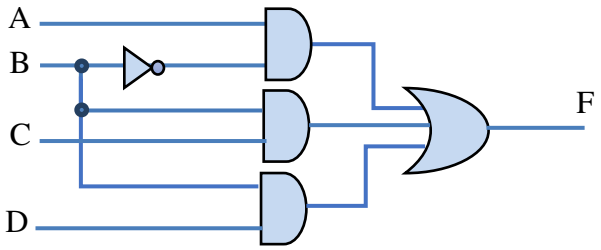


$\mathbf{F= A\overline{B} + BD + BC}$

WWW.DZPEN.COM | قلم الجزائر

D. Schéma logique ou logigramme de la fonction simplifiée

Figure 126: Circuit logique de F



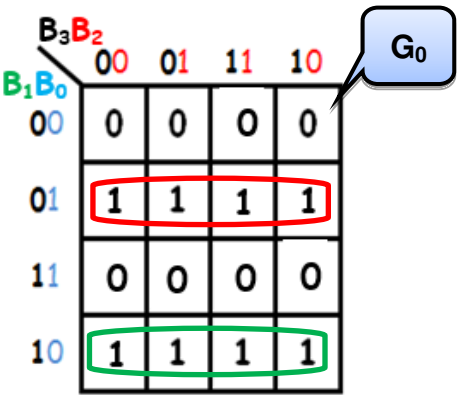
Exercice 11 : Réalisationd'un convertisseur Binaire Naturel-Binaire Réfléchi 4bits

A. Table de vérité

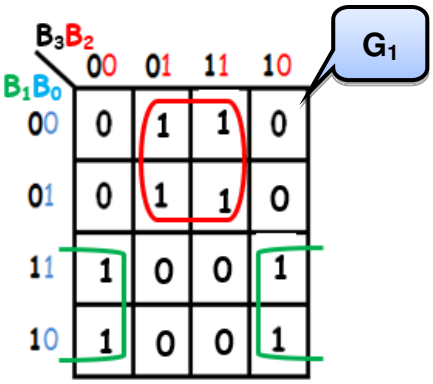
Entrée Binaire Naturel				Sortie Binaire Réfléchi			
B_3	B_2	B_1	B_0	G_3	G_2	G_1	G_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0

WWW.DZPEN.COM | قلم الجزائر

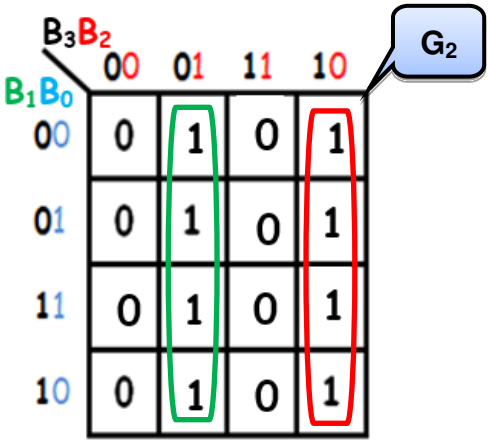
B. Tableau de KARNAUGH



$G_0=\overline{B_0}.B_1 + B_0.\overline{B_1}$
$G_0=B_0 \oplus B_1$



$G_1=\overline{B_1}.B_2 + B_1.\overline{B_2}$
$G_1=B_1 \oplus B_2$

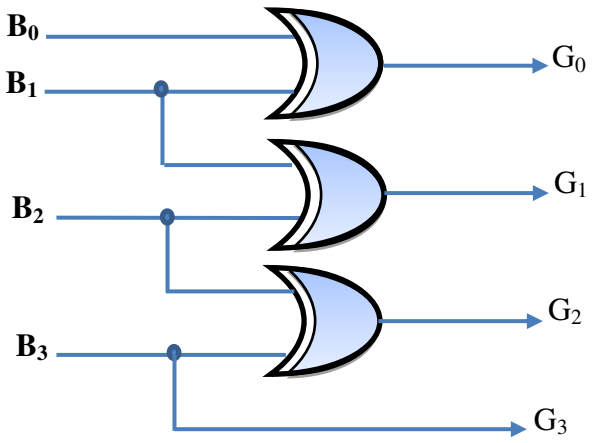


$G_2=\overline{B_3}.B_2 + B_3.\overline{B_2}$
$G_2=B_2 \oplus B_3$

$G_3=B_3$

C. Circuit logique

Figure 127: Circuit logique de convertisseur Binaire Naturel-Binaire Réfléchi 4bits



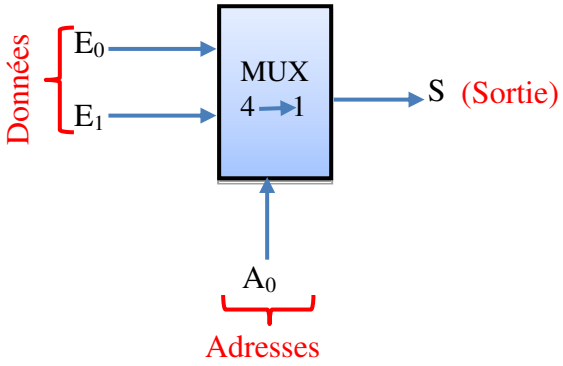
WWW.DZPEN.COM | قلم الجزائر

Exercice 12

A. MUX 2 vers 1

➢ **Synoptique**

Figure 128: Synoptique d'un MUX 2 vers 1



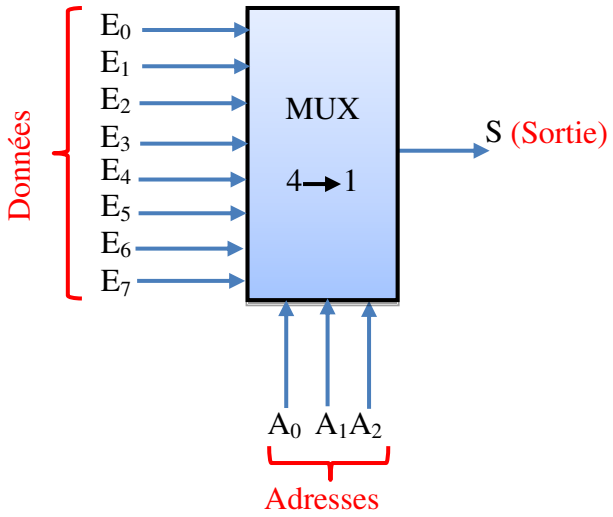
➢ **Table de vérité**

A_0	S
0	E_0
1	E_1

B. MUX 8 vers 1

➢ **Synoptique**

Figure 129: Synoptique d'un MUX 4 vers 1



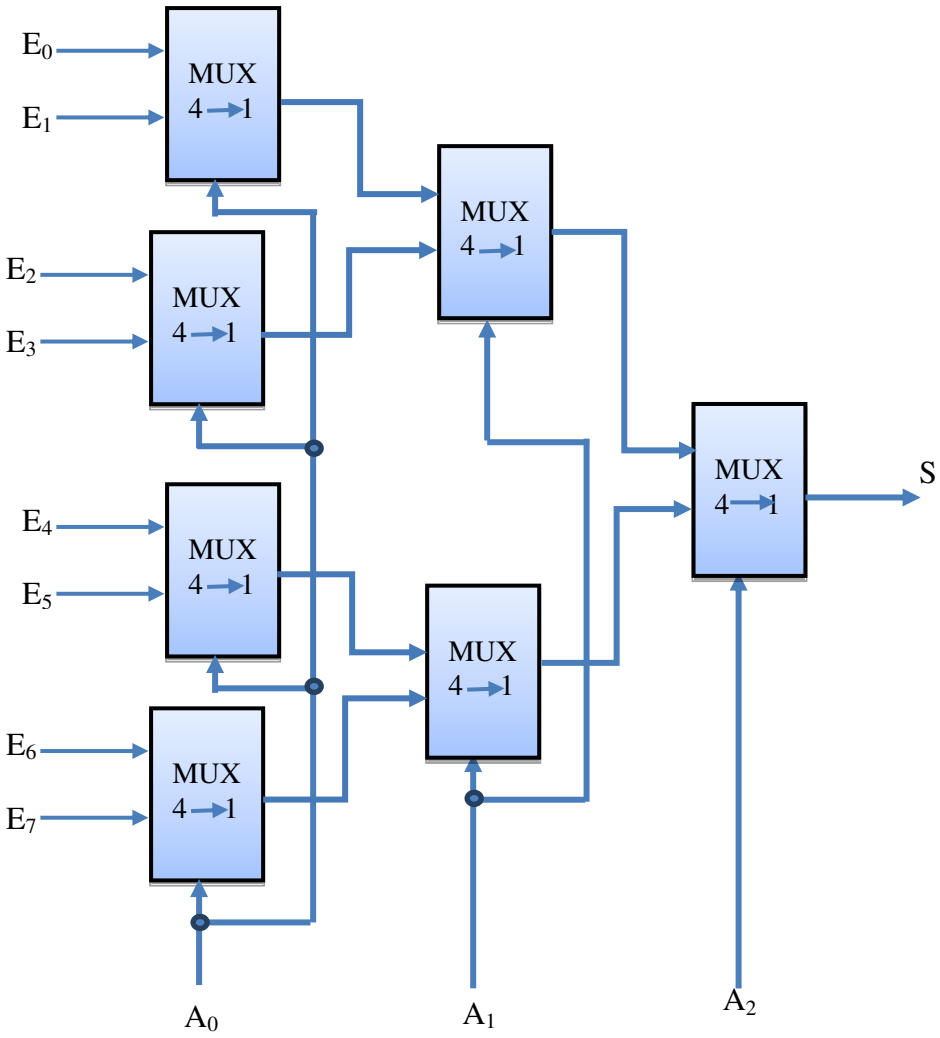
➢ **Table de vérité**

A_2	A_1	A_0	S
0	0	0	E_0
0	0	1	E_1
0	1	0	E_2
0	1	1	E_3
1	0	0	E_4
1	0	1	E_5
1	1	0	E_6
1	1	1	E_7

WWW.DZPEN.COM | قلم الجزائر

C. Synoptique d'un MUX 8 vers 1 à l'aide de MUX 2 vers 1

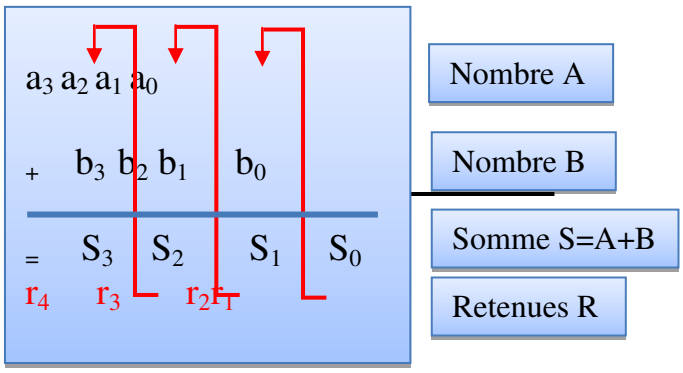
Figure 130: Synoptique d'un MUX 8 vers 1à l'aide de MUX 2 vers 1



Exercice 13 Schéma fonctionnel d'un additionneur complet 4 bits

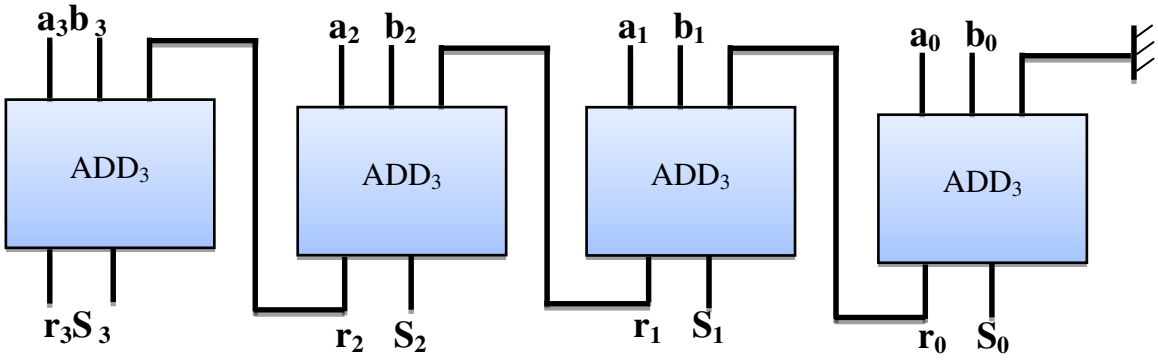
La figure ci-dessous montre la décomposition de l'addition de deux nombres binaires de quatre bits.

Figure 131 : Décomposition de l'addition de deux nombres binaires de 4 bits



WWW.DZPEN.COM | قلم الجزائر

Figure 132: Synoptique d'un additionneur complet 4 bits



Exercice 14

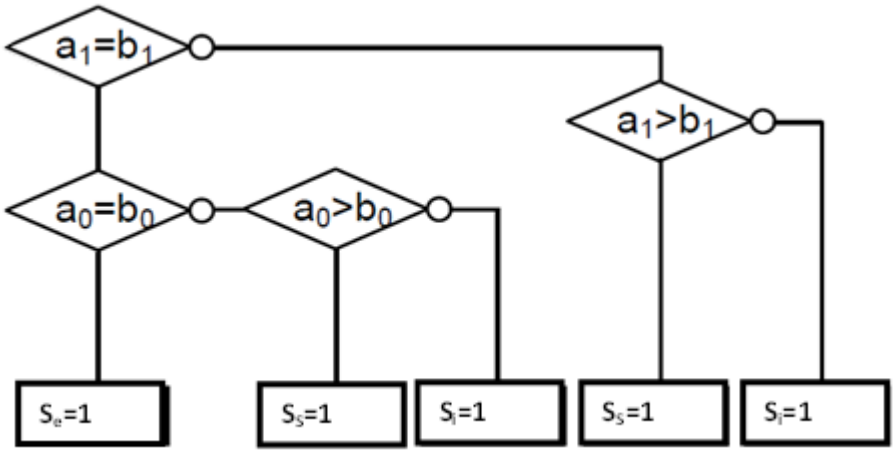
Comparateur 2 bitsqui permet de faire la comparaison entre deux nombres A (A_1A_0) et B (B_1B_0) chacun sur deux bits

A. Synoptique

Figure 133 : Schéma fonctionnel d'un comparateur 2 bits



B. Organigramme

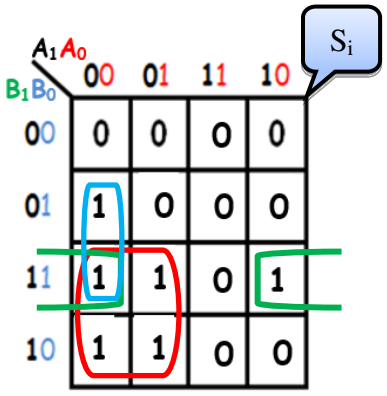


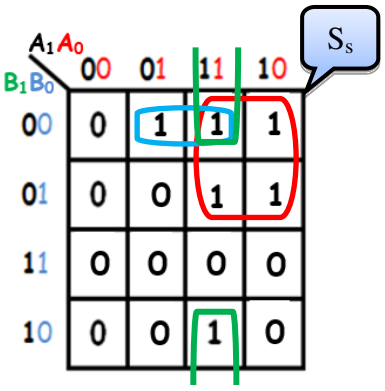
WWW.DZPEN.COM | قلم الجزائر

C. **Table de vérité**

B_1	B_0	A_1	A_0	S_e(A=B)	S_i (A<B)	S_s(A>B)
0	0	0	0	1	0	0
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	0	1
0	1	0	0	0	1	0
0	1	0	1	1	0	0
0	1	1	0	0	0	1
0	1	1	1	0	0	1
1	0	0	0	0	1	0
1	0	0	1	0	1	0
1	0	1	0	1	0	0
1	0	1	1	0	0	1
1	1	0	0	0	1	0
1	1	0	1	0	1	0
1	1	1	0	0	1	0
1	1	1	1	1	0	0

D. **Equations**





$S_i = \bar{A}_1 . B_1 + \bar{A}_0 . \bar{A}_1 . B_0 + \bar{A}_0 . B_0 . B_1$

$\mathbf{S_i = \bar{A}_1 . B_1 + (\overline{A_1 \oplus B_1}) . (\bar{A}_0 . B_0)}$

$S_S = A_1 . \bar{B}_1 + A_0 . \bar{B}_0 . \bar{B}_1 + A_0 . A_1 . \bar{B}_0$

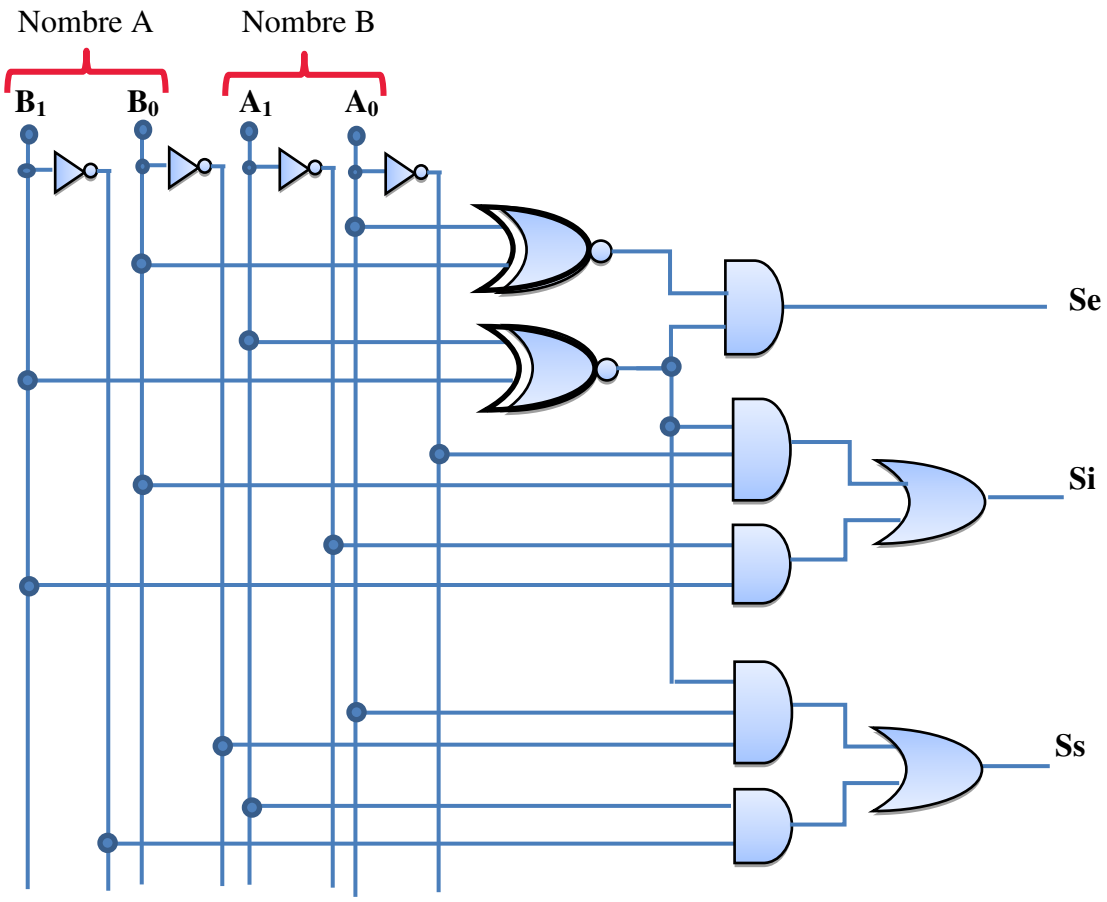
$\mathbf{S_S = A_1 . \bar{B}_1 + (\overline{A_1 \oplus B_1}) . (A_0 . \bar{B}_0)}$

$S_e = \bar{A}_0 . \bar{A}_1 . \bar{B}_0 . \bar{B}_1 + A_0 . \bar{A}_1 . B_0 . \bar{B}_1 + \bar{A}_0 . A_1 . \bar{B}_0 . B_1 + A_0 . A_1 . B_0 . B_1$

$\mathbf{S_e = (\overline{A_0 \oplus B_0}) . (\overline{A_1 \oplus B_1})}$

WWW.DZPEN.COM | قلم الجزائر

Figure 134 : Circuit logique d'un comparateur 2 bits



Exercice de synthèse 3 D'après la figure 54 de la page 54,l'équation logique de la fonction S est :

S1= $\overline{A}\overline{B}$ + AB = $\overline{A \oplus B}$

S_2= $\bar{C}\bar{D}$ + $\bar{C}D$ + CD = $\bar{C}$+ CD = $\bar{C}$+ CD

⟹ S= $\bar{s}_2 s_1 E + s_2 s_1 \bar{E}$ = $S_1(\bar{s}_2\, E + s_2 \bar{E})$= S_1(E⊕S_2)

S=$\overline{\boldsymbol{A} \oplus \boldsymbol{B}}$[E⊕($\bar{\boldsymbol{C}}$+D)

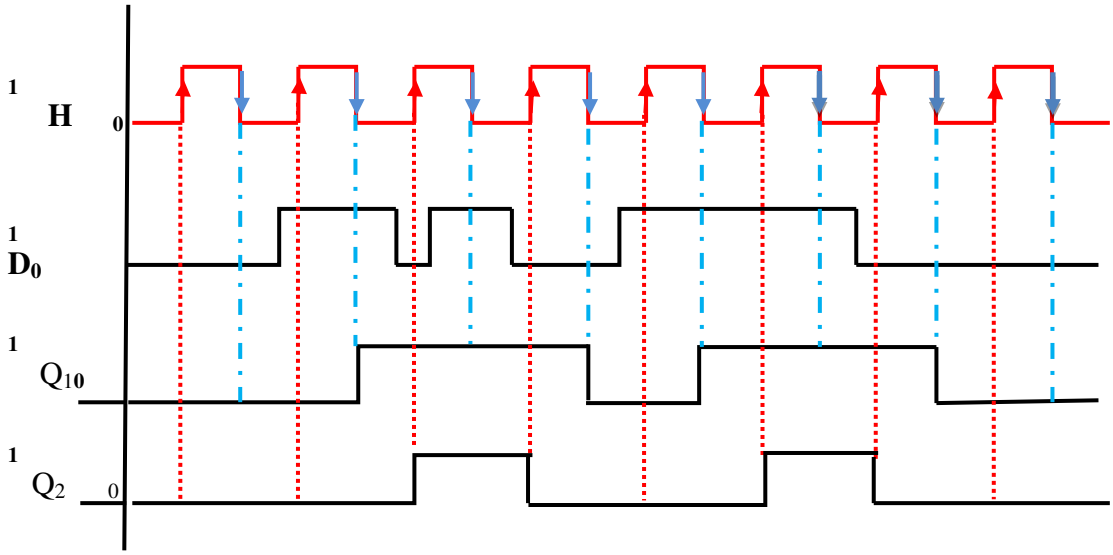
Exercice 15 Répondre par vrai ou faux

- Un signal d'horloge est généralement un signal sinusoïdal***(Faux)***
- Un front descendant caractérise un signal passant de 0 à 1 ***(Faux)***
- l'état de la sortie d'un circuit séquentieldépend non seulement de la combinaison appliquée à ces l'entréesmais aussi de l'état précédent de ces sorties***(Vrai)***
- La logique séquentielle est une logique combinatoire avec une mémorisation des sorties***(Vrai)***
- Un circuit est dit synchrone si la lecture de ces entrées se fait à tout instant***(Faux)***
- Dans un circuitl'état de sortie ne dépend que de l'état présent des entrées***(Vrai)***
- Une bascule est un organe de mémorisation unitaire***(Vrai)***

WWW.DZPEN.COM | قلم الجزائر

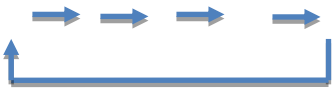
Exercice 16 Les chronogrammes de Q1 et Q2 de la figure 81 (On suppose qu'au début Q1 =0 et Q2=0)

Figure 135 : Chronogrammes du circuit logique de la figure 81



Exercice 17:Réalisation d'un compteur synchrone qui affiche la séquence suivante :

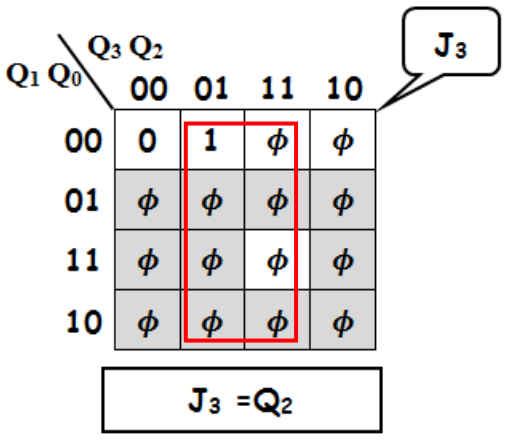
0 4 8 12 15

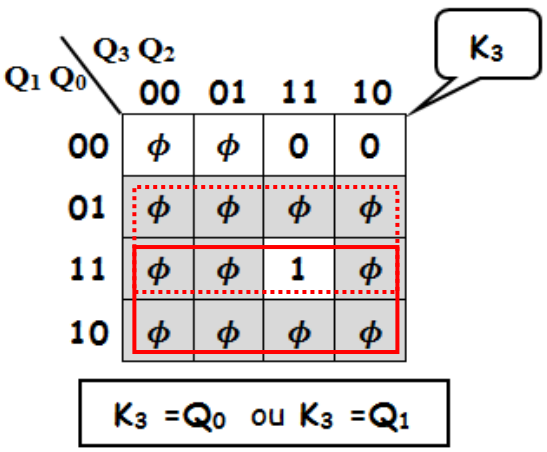


A. Table de Vérité

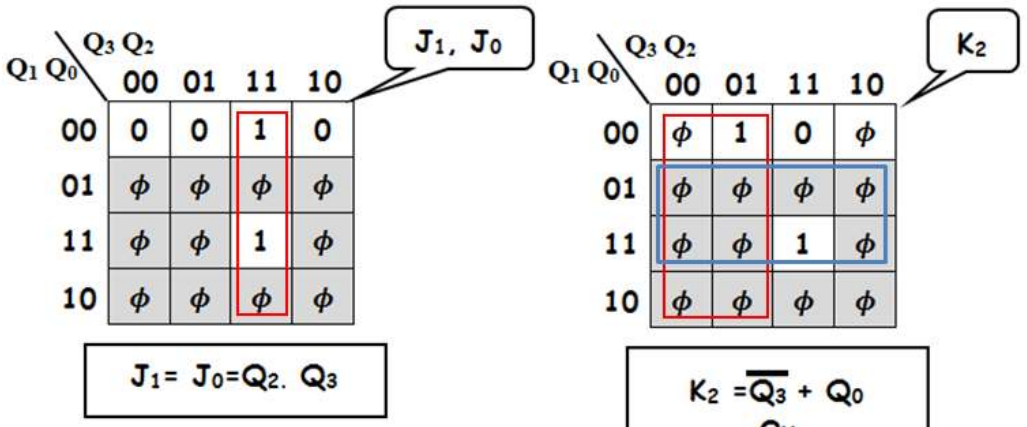
	Q_3	Q_2	Q_1	Q_0	J_3	K_3	J_2	K_2	J_1	K_1	J_0	K_0
0	0	0	0	0	0	Φ	1	Φ	0	Φ	0	Φ
4	0	1	0	0	1	Φ	Φ	1	0	Φ	0	Φ
8	1	0	0	0	Φ	0	1	Φ	0	Φ	0	Φ
12	1	1	0	0	Φ	0	Φ	0	1	Φ	1	Φ
15	1	1	1	1	Φ	1	Φ	1	Φ	1	Φ	1

B. Tableau de KARNAUGH





WWW.DZPEN.COM | قلم الجزائر

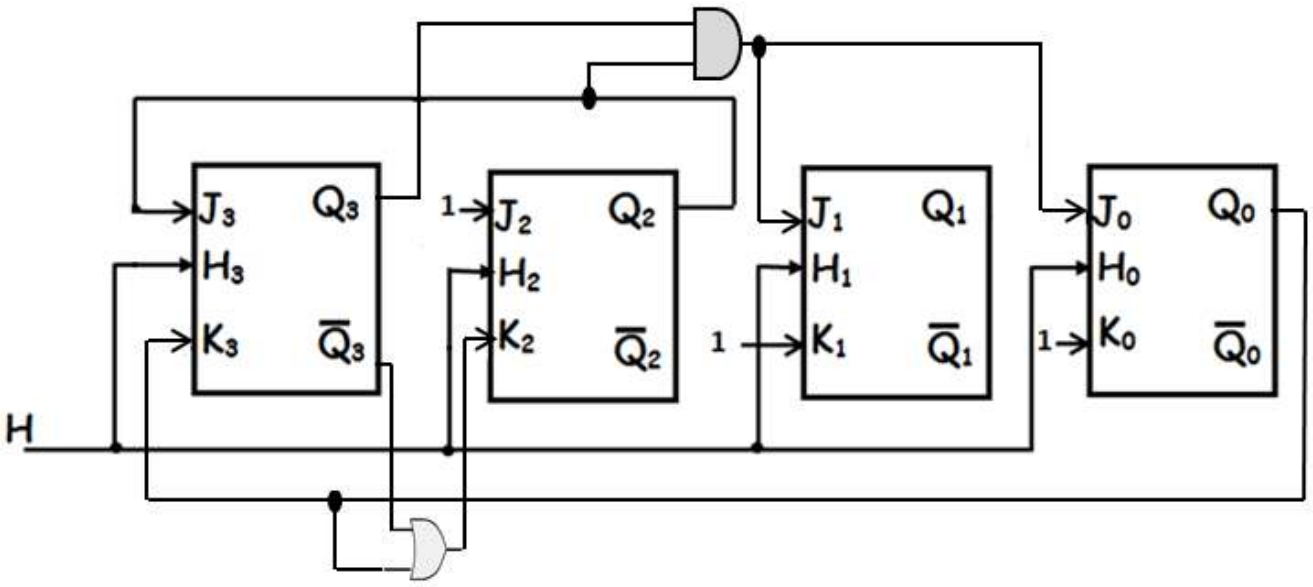


$\begin{cases} J_0=Q_2.Q_3 \\ K_0=1 \end{cases}$ $\begin{cases} J_1=Q_2.Q_3 \\ K_1=1 \end{cases}$

$\begin{cases} J_2=1\ J_3=Q_2 \\ K_2=\overline{Q_3}+Q_0 \\ \quad Ou \\ K_2=\overline{Q_3}+Q_1 \end{cases}$ $\begin{cases} K_3=Q_0 \text{ ou } K_3=Q_1 \end{cases}$

C. Logigramme

Figure136 : Circuit logiqued'un compteur synchrone



WWW.DZPEN.COM | قلم الجزائر

Exercice 18D'après le circuit de la figure 96 de la page 79, on a :

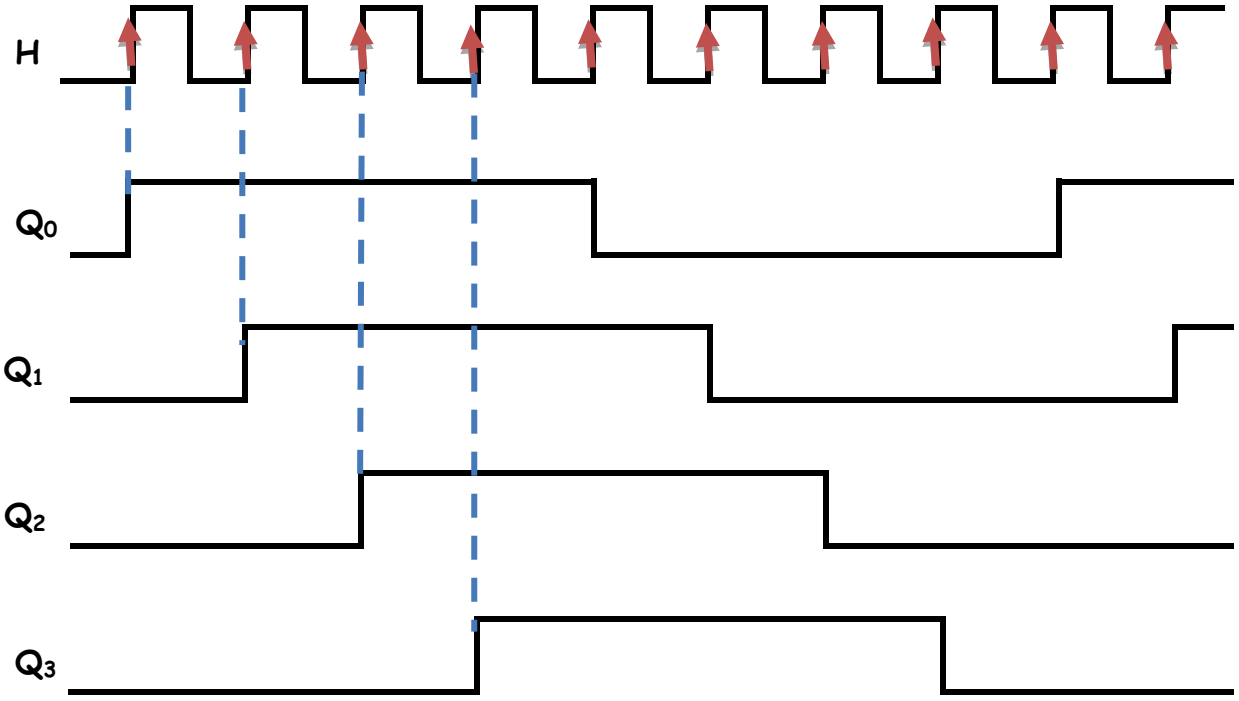
$D_0 = \overline{Q}_3$; $D_1 = Q_0$;$D_2 = Q_1$; $D_3 = Q_2$

L'état initial est :

$Q_0 = 0$; $Q_1 = 0$; $Q_2 = 0$ et $Q_3 = 0$ $\Longrightarrow D_0 = 1$; $D_1 = 0$; $D_2 = 0$ et $D_3 = 0$

A. Chronogramme :

Figure 137 : Chronogrammes du registre à décalagede la figure 96



B. La fonction réalisée est: registre à décalage circulaire à droite et le cycle réalisé est :

0 – 1 – 3 – 7 – F – E – C – 8 – 0.

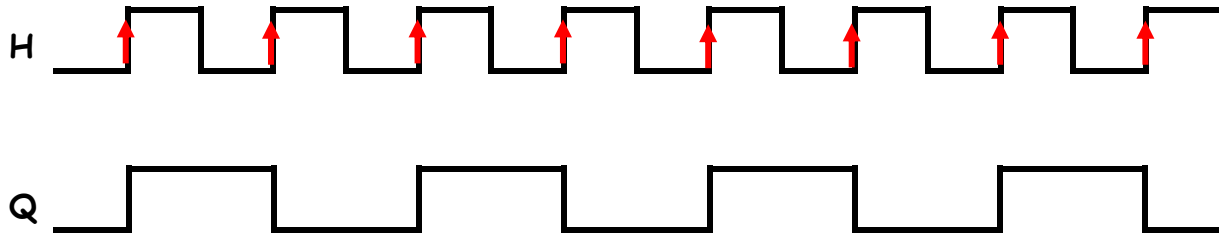
WWW.DZPEN.COM | قلم الجزائر

Exercices de synthèse 4

1. Soit la bascule D de la figure98 suivante

B. Chronogramme obtenu de H et Q. (Q bascule à chaque front montant de H.)

Figure138: Chronogrammes de la bascule D de la figure 98



C. Le fonctionnement du montage de figure 100:

- ✓ Au départ le bouton poussoir n'est pas appuyé et le condensateur C est déchargé (Vc = 0V). Q = 0 et D = 1. Le condensateur se charge et H (entrée d'horloge) passe de 0 à 1 (front montant) et Q = 1 et D = 0 (***la LED s'allume***).
- ✓ Lorsqu'on appui sur bouton poussoir le condensateur C est court-circuité (Vc = 0V) et H passe de 1 à 0 (front descendant), Q ne change pas (Q = 1 et D = 0).
- ✓ Lorsqu'on relâche le bouton poussoir, le condensateur se charge et CLK passe de 0 à 1 (front montant) et Q = 0 et D = 1 (***la LED s'éteint***).

Donc à chaque fois qu'on appui et on relâche le bouton poussoir, on a un basculement de Q (la LED passe d'un état à l'autre).

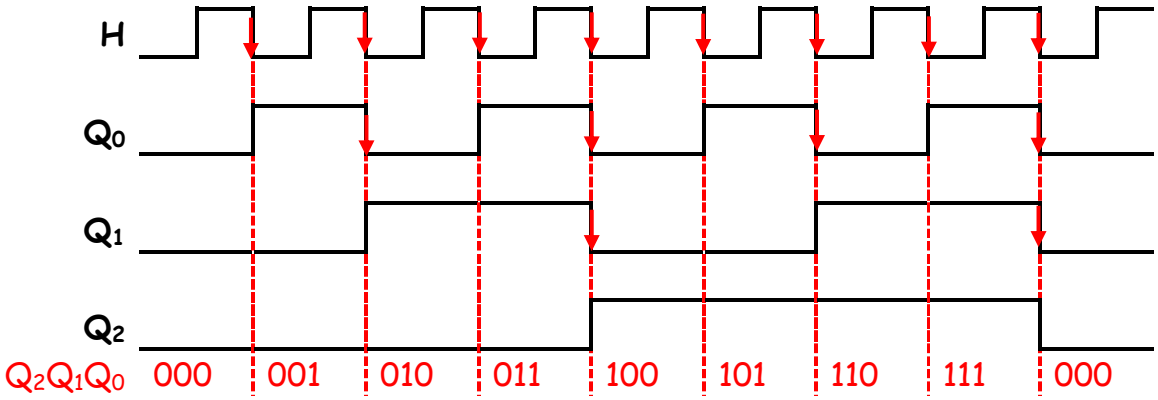
WWW.DZPEN.COM | قلم الجزائر

2. Soit le compteur asynchrone de la figure 101 :

A. Chronogramme :

- ✓ Q_0 a pour horloge H ; donc à chaque front descendant de H, Q_0 change d'état (bascule de 0 à 1 ou de 1 à 0).
- ✓ Q_1 a pour horloge Q_0; donc à chaque front descendant de Q_0, Q_1 change d'état.
- ✓ Q_2 a pour horloge Q_1; donc à chaque front descendant de Q_1, Q_2 change d'état.

Figure139: Chronogramme du compteur asynchrone de la figure101



B. On obtient la séquence suivante : 0, 1, 2, 3, 4, 5, 6, 7,0

C. On a un compteur modulo 8.

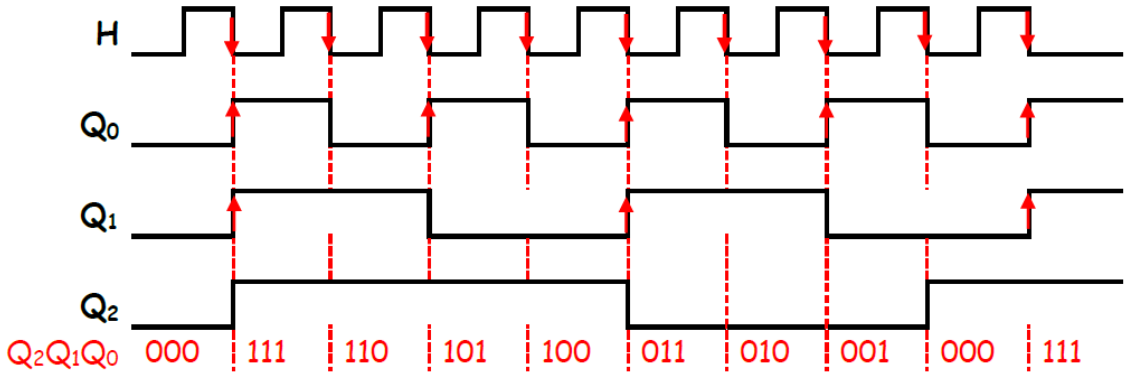
WWW.DZPEN.COM | قلم الجزائر

3. Soit le compteur asynchrone de la figure 103 :

A. Chronogramme :

- ✓ Q_0 a pour horloge H ; donc à chaque front descendant de H, Q_0 change d'état (bascule de 0 à 1 ou de 1 à 0).
- ✓ Q1 a pour horloge $\bar{Q}_0$; donc à chaque front descendant de $\bar{Q}_0$ (front montant de Q_0), Q_1 change d'état.
- ✓ Q_2 a pour horloge $\overline{Q_1}$; donc à chaque front descendant de $\overline{Q_1}$ (front montant de Q_1), Q_2 change d'état.

FIG140: Chronogramme du compteur asynchrone de la figure103



B. On obtient la séquence suivante : 0, 7, 6, 5, 4, 3, 2, 1, 0

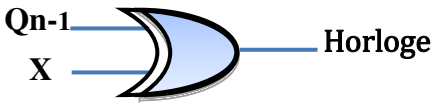
C. On a un décompteur modulo 8.

D. L'horloge de la 1ère bascule dans les 2 cas est H. Pour le compteur l'horloge de la bascule n est Qn-1 et pour le décompteur l'horloge de la bascule n est $\bar{Q}_{n-1}$

E. Pour concevoir un compteur/décompteur on doit choisir soit Qn-1, soit $\bar{Q}_{n-1}$ Pour faire le choix, on va utiliser une variable X, tel que :

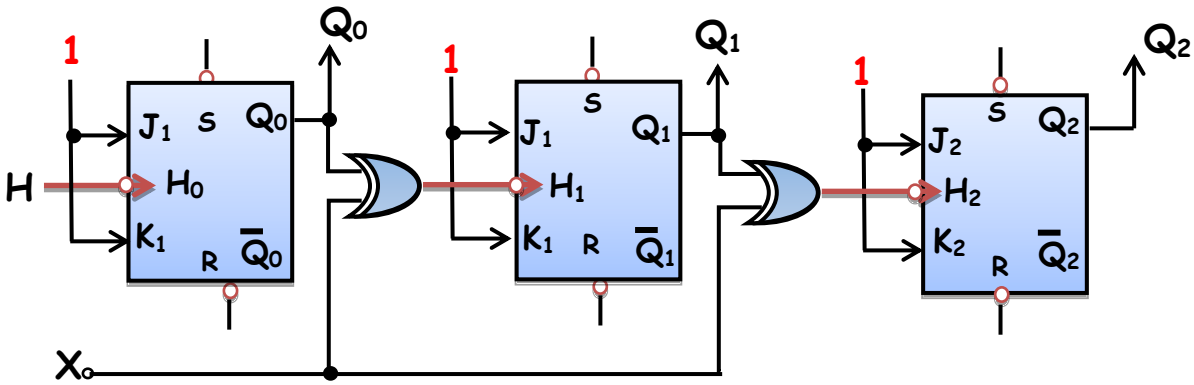
X	Horloge	Mode
0	Qn-1	Compteur
1	$\bar{Q}_{n-1}$	Décompteur

$$Horloge = Q_{n-1}\bar{X} + \bar{Q}_{n-1}X = Q_{n-1} \oplus X$$



WWW.DZPEN.COM | قلم الجزائر

Figure 141 : Circuit logiqued'un compteur/décompteur synchrone



Exercice 19

A. La différence entre un automate programmable compact et modulaire : le type compact intègre le processeur, l'alimentation, les entrées et les sorties dans un seul boitier (***rack***), a l'inverse du modulaire chacun est un module appart c'est à dire un rack contenant plusieurs module un module d'alimentation, un processeur, des modules d'E/S, des modules de communication et de comptage

B. Les automates classés selon leur type

Type compact	Type modulaire
CROZET Millenium SCHNEIER Zelio MoellerPS4	Siemens S7-300 SchneiderTSX 57

Exercice20 : Répondez par vrai ou faux

A. A l'aide de la technologie de logique câblée il devient possible de traiter une variété d'applications qui, autrement, auraient chaque fois requis des matériels différents. ***Faux***

B. Les boîtiers de tests permettent de visualiser le programme ou les valeurs des paramètres ***Vrai***

C. Le module d'alimentation délivre des sources de tension dont l'automate a besoin : 24V, 12V ou 5V en alternatifs.***Faux***

D. Les cartes de sortie analogiques assurent la conversion analogique / numérique ***Faux***

E. La mémoire de données est une mémoire vive (RAM)***Vrai***

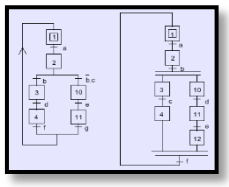
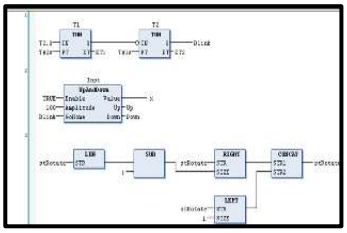
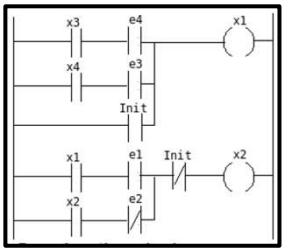
F. La mémoire système est une mémoire à lecture seul programmables effaçable électriquement.***Faux***

G. un Compteur Programme ou Compteur Ordinal qui contient l'adresse de l'instruction qui est en court d'exécution***Fau***

WWW.DZPEN.COM | قلم الجزائر

Exercice21 : Remplir le tableau de la figure118

Tableau 142: Tableau récapitulatif des langages de programmation des

Exemple de langage de programmation	Le Nom du langage	Langage Graphique ou Textuel
	Langage GRAFCET ou SFC	Graphique
	Blocs FonctionnelsFBD	Graphique
	Langage littéral structuréST	Textuel
	Liste d'instructions IL	Textuel
	Langage à contactsLD	Graphique

WWW.DZPEN.COM | قلم الجزائر

Exercice de synthèse 5

Aspect extérieur

1- Compact et modulaire, Répondez par vrai ou faux:

A. Le S7-200 CPU222 est un API compact; **Vrai**

B. Le TSX 3721 est un API compact; **Faux**

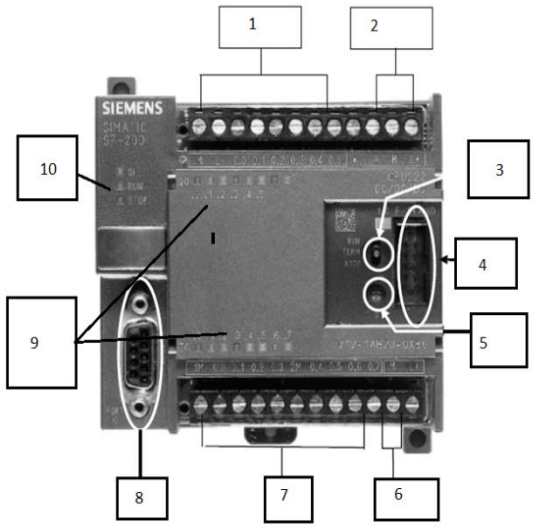
C. L'appellation " Modulaire" est due au mot " Mode"; **Faux**

D. Dans un API modulaire les parties processeur, alimentation et interface entrée/sortie résident dans des unités séparées. **Vrai**

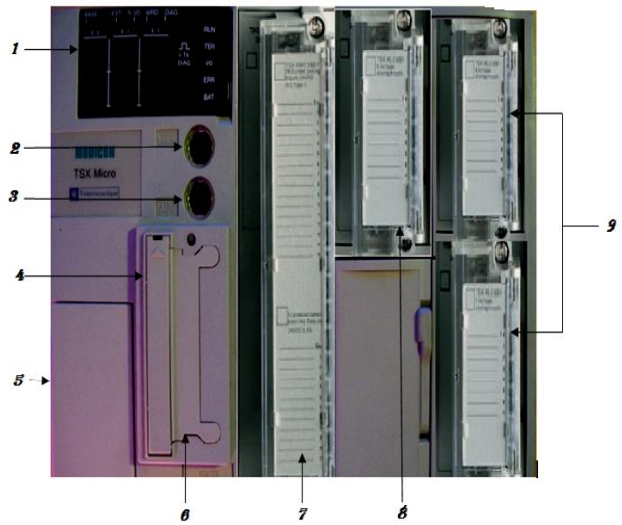
E. L'information délivrée par un capteur est une information analogique **Vrai**

2- Identifiez l'emplacement des éléments d'un API:

Identifier les différents elements sur un API enreportant les repères correspondants dans le tableau ci-dessous.	
Alimentation de l'automate.	**2**
Bornes des entrées de l'automate.	**7**
Bornes de sorties des automates.	**1**
Sélecteur de mode STOP, RUN	**3**
Connecteur pour modules d'extinction	**4**
Interface de programmation.	**8**
Borniers permettant d'alimenter les capteurs pour les entrées.	**6**
Visualisation des états des entrées et des sorties de l'automate .	**9**
Visualisation de l'état de l'automate .	**10**



Emplacement pour carte d'extension mémoire.	**4**
Trappe d'accès aux bornesd'alimentation	**5**
Emplacement pour coupleur decommunication	**6**
Bloc de visualisation (affiche des informations sur l'état del'automate)	**1**
Prise TERMINAL (TER) permet de connecté le terminal de programmation	**2**
Prise de dialogue opérateur (AUX) permet de connecté un pupitre de dialogue	**3**



Emplacement pour module de sortie dans le module de base de type TOR sous la référence TSX DSZ08R5.	**9**

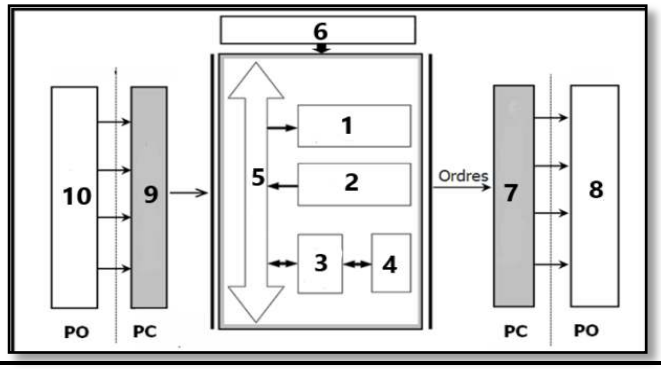
WWW.DZPEN.COM | قلم الجزائر

Emplacement pour module d'entrée dans le module de base de type TOR sous la référence TSX DEZ 32D2.	7
Emplacement pour module d'entrée dans le module de base de type analogique sous la référence TSX AEZ 4144multirage(TC.RTD).	8

Aspect intérieur

1. Identifiez l'architecture d'un API.

Capteurs	10
Unité de traitement	3
Interface d'entrées	9
RAM	4
Bus d'entrées / sorties	5
Modules de sorties	1
Alimentation CA/CC	6
Préactionneurs	8
Interface de sorties	7
Modules d'entrées	2



2. Identifier le rôle des différentes parties de l'API.

Le Microprocesseur	Reçoit les informations issues des capteurs d'entrées et les informations générées par le processeur et destinées à la commande des sorties
Module d'entrées	Gère l'ensemble des échanges informationnels en assurant, la lecture des informations d'entrée, l'exécution des instructions du programme mis en mémoire et la commande ou l'écriture des sorties.
Mémoire	Permet à l'Unité Centrale de l'automate, d'effectuer une "lecture" de l'état logique des capteurs qui lui sont associes.

3. Mémoires

ROM	Read Only Memory	Mémoire a lecture seule
RAM	Random Access Memory	Mémoire à accès aléatoire(mémoire vive)
PROM	Programmable Read Only Memory	Mémoireeffaçableprogrammablesa lecture seule
EPROM	Erasable Programmable Read Only Memory	Mémoireeffaçableprogrammablesa lecture seule
EEPROM	Electrically Erasable Programmable Read Only Memory	Mémoire a lecture seuleeffaçable programmables electriquement
CMOS	Complementary Metal Oxide Semiconductor	semi-conducteur complémentaire à l'oxyde de métal

4. Mettre le type de mémoire correspondant: ROM, RAM,…

Mémoire du système d'exploitation	PROM ou ROM
Mémoire du programme	EEPROM
Mémoire des données lors du fonctionnement	RAM

WWW.DZPEN.COM | قلم الجزائر

Correction Activité de synthèse

I. Algèbre de Boole

Distributeur de boissons chaudes

Soient u, v, w, z les variables logiques correspondant aux propositions suivantes :

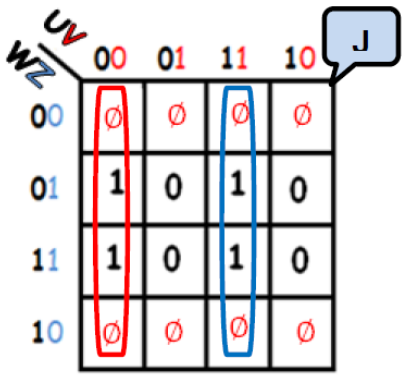
- Le bouton « café » est enfoncé : u = 1
- Le bouton « thé » est enfoncé : v = 1
- Le bouton « lait » est enfoncé : w = 1
- Un jeton a été introduit dans la fente de l'appareil : z = 1
- ∅: Indéfini (0 ou 1)

Simplification des fonctions de restitution du jeton, J, de distribution du café, C, du thé T, et du lait, L.

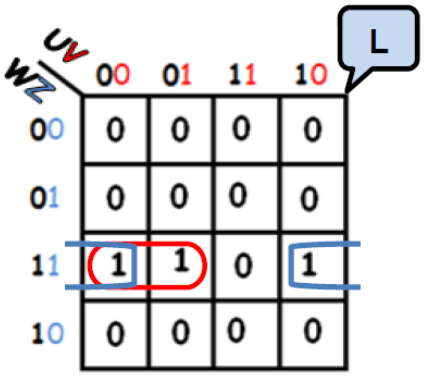
- Table de vérité de C, T, L et J :

u	v	w	z	C	T	L	J
0	0	0	0	0	0	0	∅
0	0	0	1	0	0	0	1
0	0	1	0	0	0	0	∅
0	0	1	1	0	0	1	1
0	1	0	0	0	0	0	∅
0	1	0	1	0	1	0	0
0	1	1	0	0	0	0	∅
0	1	1	1	0	1	1	0
1	0	0	0	0	0	0	∅
1	0	0	1	1	0	0	0
1	0	1	0	0	0	0	∅
1	0	1	1	1	0	1	0
1	1	0	0	0	0	0	∅
1	1	0	1	0	0	0	1
1	1	1	0	0	0	0	∅
1	1	1	1	0	0	0	1

- Tableau de KARNAUGH



$J = \bar{U}.\bar{V} + U.V = \overline{U \oplus V}$



$L = \bar{U}.W.Z + \bar{V}.W.Z = W.Z.(\bar{U} + \bar{V})$

WWW.DZPEN.COM | قلم الجزائر

$C=U.\bar{V}.\bar{W}.Z + U.\bar{V}.W.Z = U.\bar{V}.Z.(W+\bar{W})$

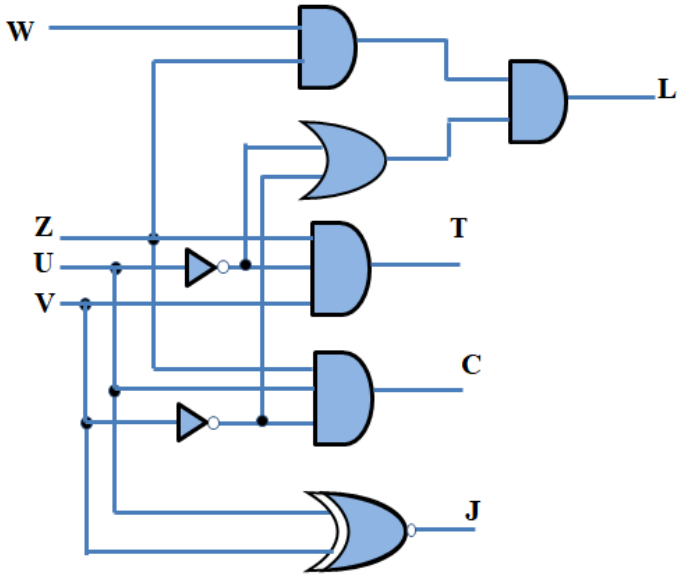
$C= U.\bar{V}.Z$

$T= \bar{U}.V.\bar{W}.Z+\bar{U}.V.W.Z = \bar{U}.V.Z.(W+\bar{W})$

$T= \bar{U}.V.Z$

- **Circuit logique**

Figure 143 : Circuit logiquedu distributeur deboissons chaudes



II. Circuits combinatoires

Demi-Soustracteur.

1. Table de vérité :

A_i	B_i	D_i	R_i
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

2. Equations :

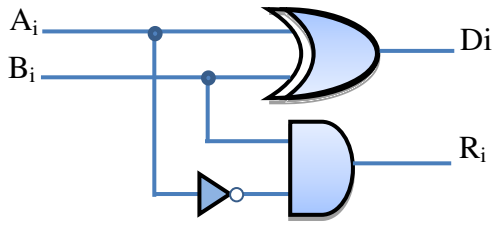
$D_i=\bar{A_1}+B_i+ A_i+\bar{B_1}= A_i \oplus B_i$

$R_i=\bar{A_1}+B_i$

WWW.DZPEN.COM | قلم الجزائر

3. Logigramme

Figure144 : Circuit logiquedu demi-soustracteur



Soustracteur complet :

1. Table de vérité :

R_{i-1}	A_i	B_i	D_i	R_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	1	0
1	1	1	0	1

2. Equations :

$D_i=\overline{R}_{i-1}\overline{A}_i.B_i+\overline{R}_{i-1}A_i.\overline{B}_i+R_{i-1}\overline{A}_i\overline{B} + R_{i-1}A_i\overline{B}_i$

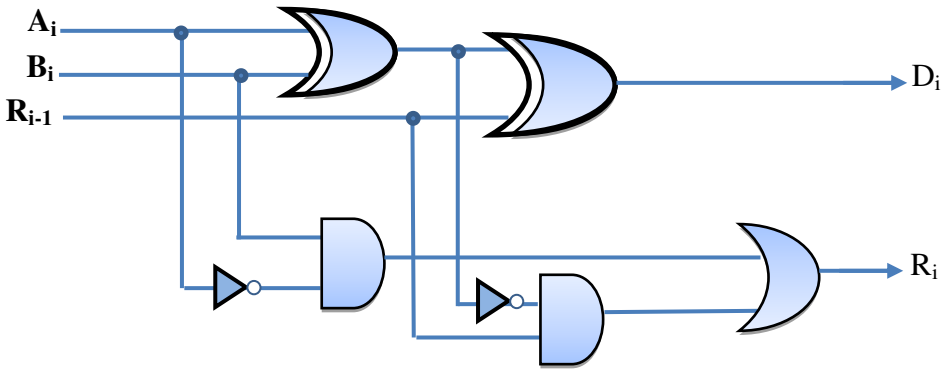
$\mathbf{D_i= (A_i \oplus B_i) \oplus R_{i-1}}$

$R_i= \overline{R}_{i-1}.\overline{A}_i.B_i+R_{i-1}.\overline{A}_i.\overline{B} + R_{i-1}.\overline{A}_i.B_i+R_{i-1}.A_i.B_i$

$\mathbf{R_i=\overline{A}_i.B_i + R_{i-1}\ (A_i \oplus B_i)}$

3. Logigramme:

Figure 145 : Circuit logiquedu soustracteur complet



WWW.DZPEN.COM | قلم الجزائر

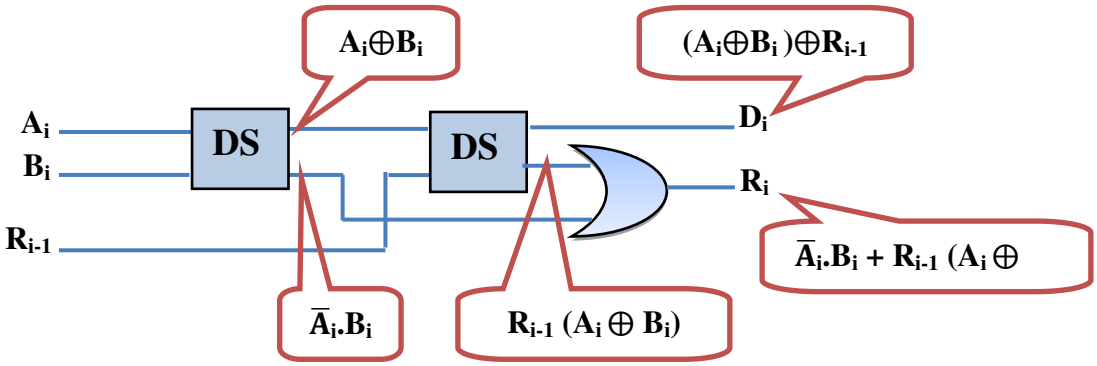
4.

A. Avec deux demi-soustracteurs :

Ce schéma correspond au fait que le soustracteur est réalisé en :

- Retranchant B_i de A_i (1er demi-soustracteur) (**DS**)
- Puis retranchant R_{i-1}de la différence obtenue

Figure 146 : Circuit logiquedu soustracteur complet à l'aide de 2 demi-soustracteurs



B. Avec un demi-additionneur et un demi-soustracteur.

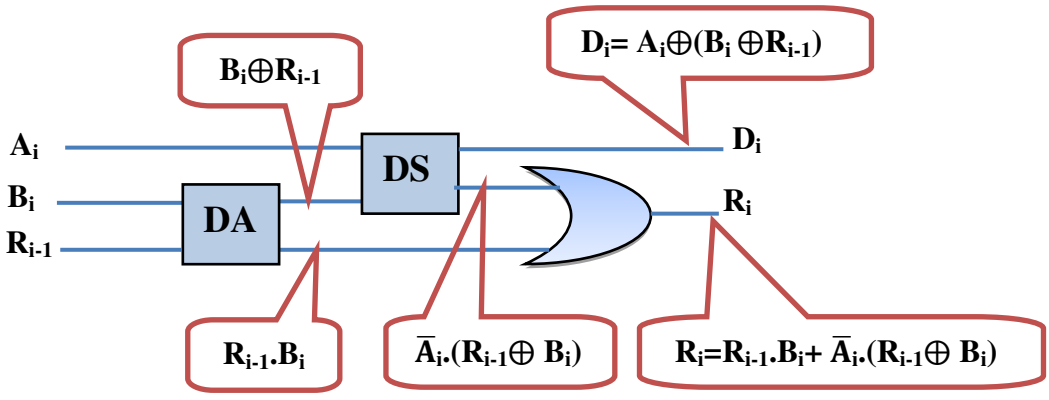
Une autre manière consiste à :

- Additionner Bi et Ri-1 avec un demi-additionneur (DA) (cette opération peut évidemment engendrer une retenue)
- Puis on retranche le résultat obtenu de Ai.

On peut écrire :

$$D_i = A_i \oplus (B_i \oplus R_{i-1})$$
$$R_i = R_{i-1}.B_i + \bar{A}_i.(R_{i-1} \oplus B_i)$$

Figure 147 : Circuit logiquedu soustracteur complet à l'aide d'un demi-additionneur et un demi-soustracteur



WWW.DZPEN.COM | قلم الجزائر

Additionneur Soustracteur :

A. Circuit inverseur:

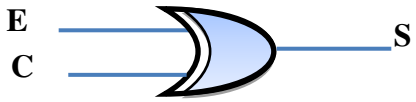
1. Table de vérité :

C	E	S
0	0	0
0	1	1
1	0	1
1	1	0

2. Equations :$S = E \oplus C$

3. Logigramme

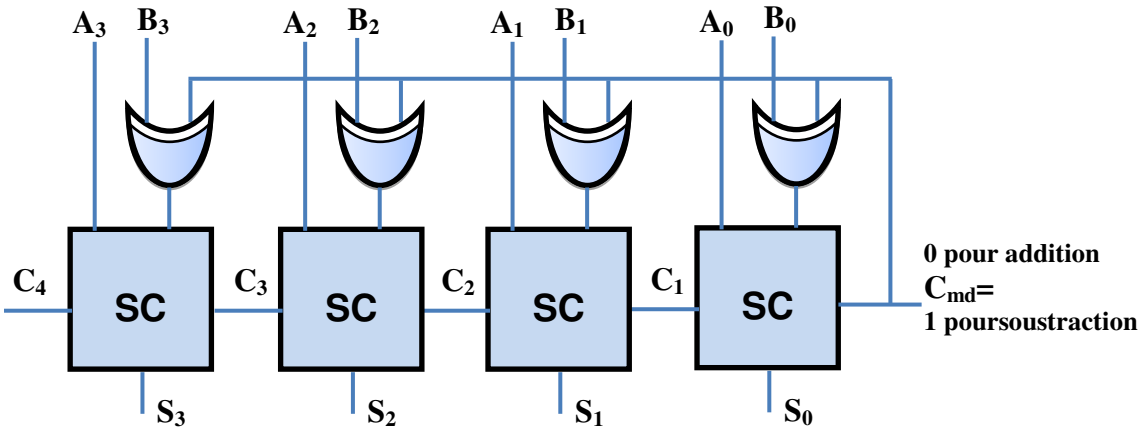
Figure 148 : Logigrammedu circuit inverseur



Pour calculer la différence **A - B** de deux nombre signés **A** et **B**, on utilise un circuit qui calcule d'abord l'opposé **-B** de **B** puis effectue la somme de **A** avec **-B** grâce à un additionneur. Le calcul de **-B** est réalisé en prenant la négation de **B** bit à bit puis en ajoutant **1** au résultat obtenu. Ce dernier **1** est en fait ajouté directement à la somme de **A** et **-B** en l'injectant comme retenue C_0 à l'additionneur. Le circuit ci-dessous effectue une somme ou une différence suivant la valeur de la commande Cmd. Si Cmd vaut 0, le circuit calcule la somme A + B. Si, au contraire, C_{md} vaut **1**, le circuit calcule la différence **A - B**. En effet, chacune des portes **XOR**effectue la négation ou non d'une entrée B_i suivant la valeur de C_{md}.

B. Circuit Additionneur Soustracteur :

Figure 149: Circuit logiquede l'additionneur/soustracteur

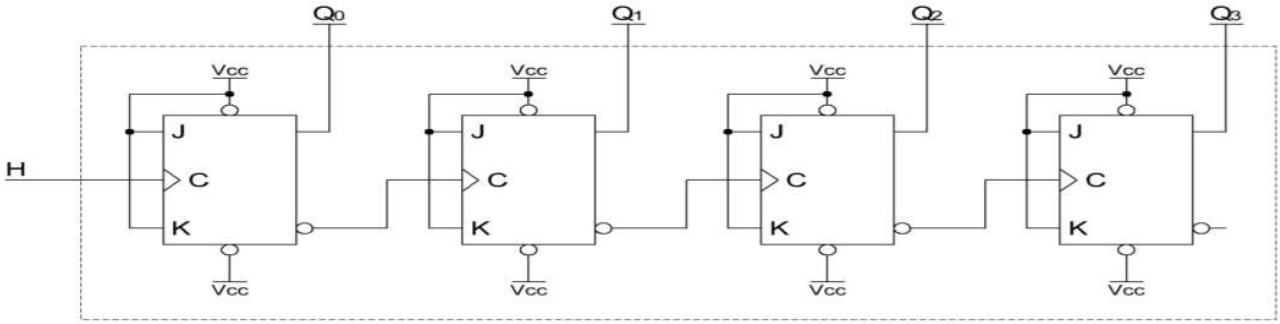


WWW.DZPEN.COM | قلم الجزائر

III. Circuits séquentiels

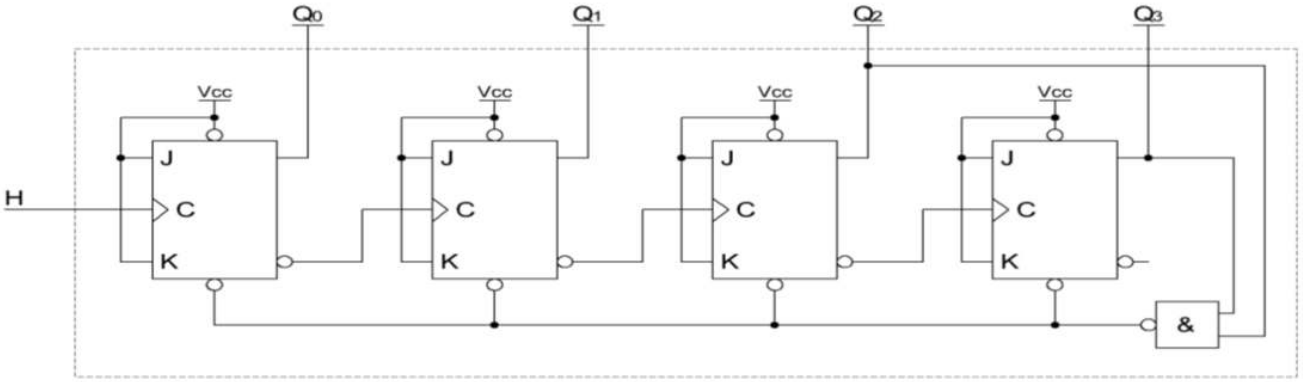
1. Compteur asynchrone modulo 16.

Figure 150: Circuit logiquedu Compteur asynchrone modulo 16



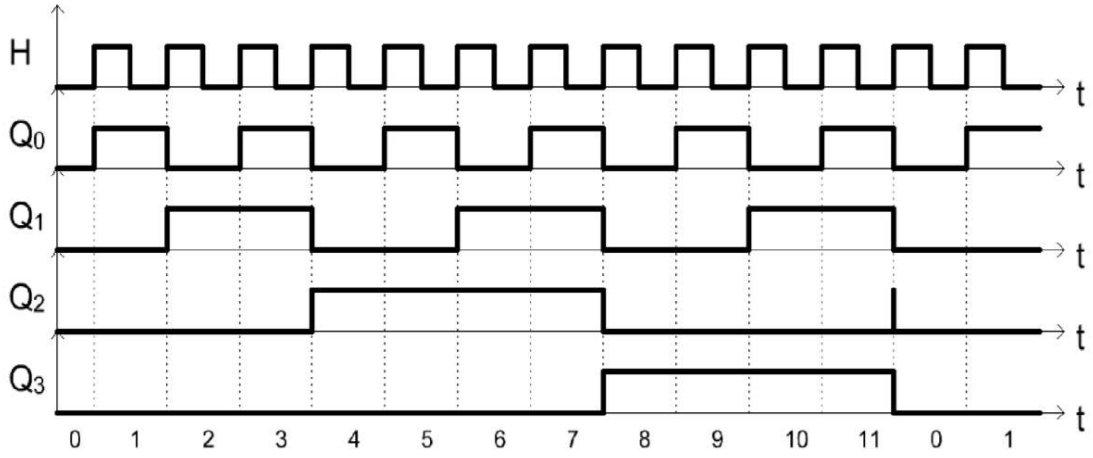
2. Compteur asynchrone modulo 12.

Figure 151: Circuit logiquedu Compteur asynchrone modulo 12



3. En partant de zéro, les chronogrammes sur un cycle complet sont les suivants :

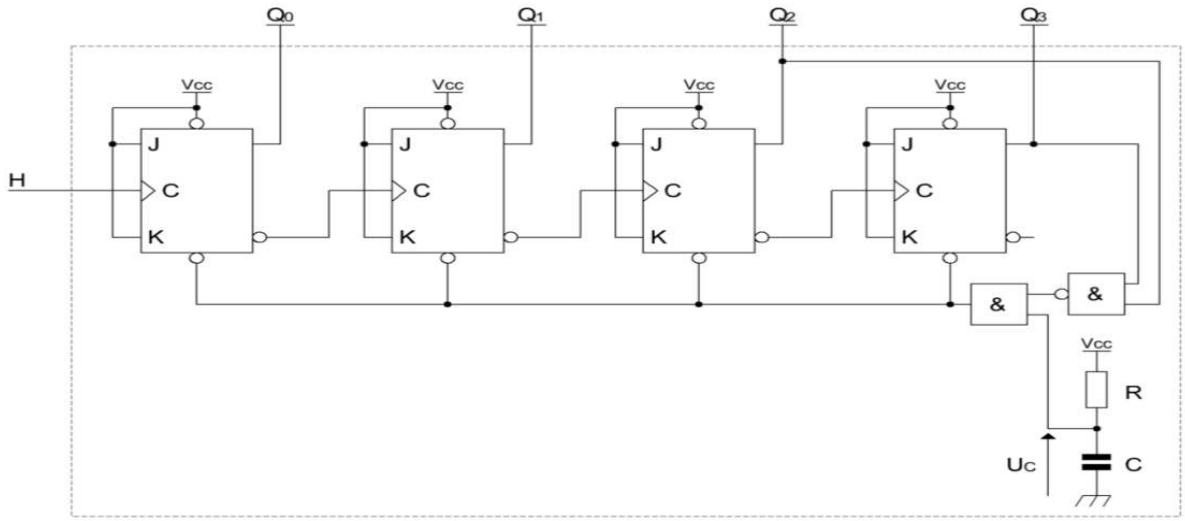
Figure 152: Chronogrammedu Compteur asynchrone modulo 12



WWW.DZPEN.COM | قلم الجزائر

4. Interrupteur automatique de remise à zéro à l'allumage

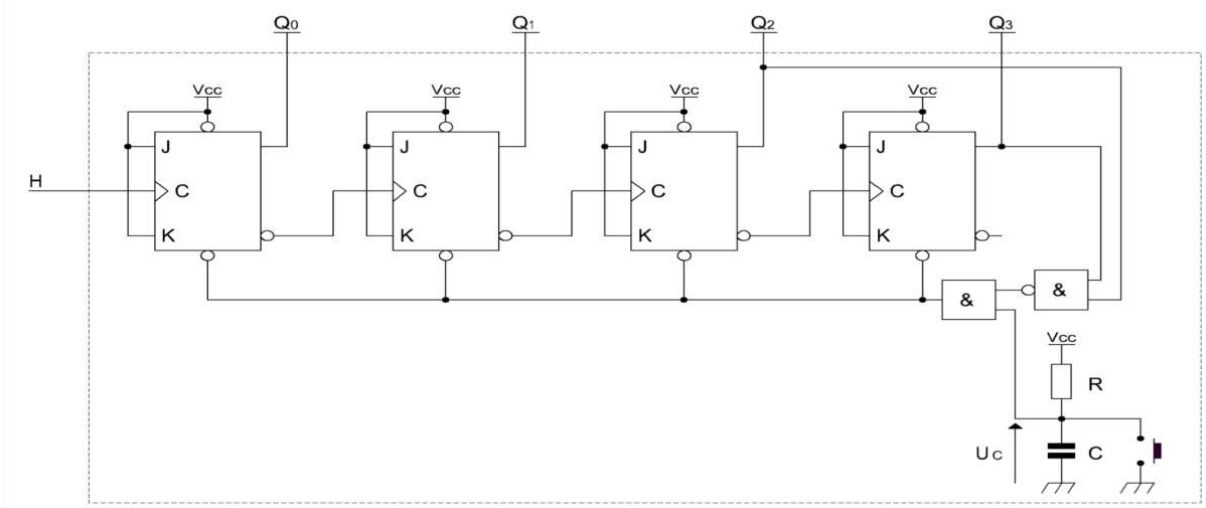
Figure 153: Circuit logiquedu Compteur asynchrone modulo 12



À l'allumage, le condensateur est déchargé et impose une tension de zéro volt à l'entrée de la porte ET. La porte considère cette tension nulle comme un niveau logique 0. Un *reset* est alors activé sur les bascules. Le condensateur se charge ensuite à travers la résistance jusqu'à la tension Vcc. Une fois que la tension a atteint un certain seuil (par ex : Vcc/2), la porte ET la considère comme un niveau logique 1. Le *reset* n'est plus imposé par la porte ET qui laisse passer le niveau de sortie de la porte NON-ET sur les entrées *reset* des bascules. Le compteur fonctionne alors dans son état normal.

5. Interrupteur manuel de remise à zéro.

Figure 154: Circuit logiquedu Compteur asynchrone modulo 12 Munit d'un Interrupteur manuel de remise à zéro à l'allumage



WWW.DZPEN.COM | قلم الجزائر

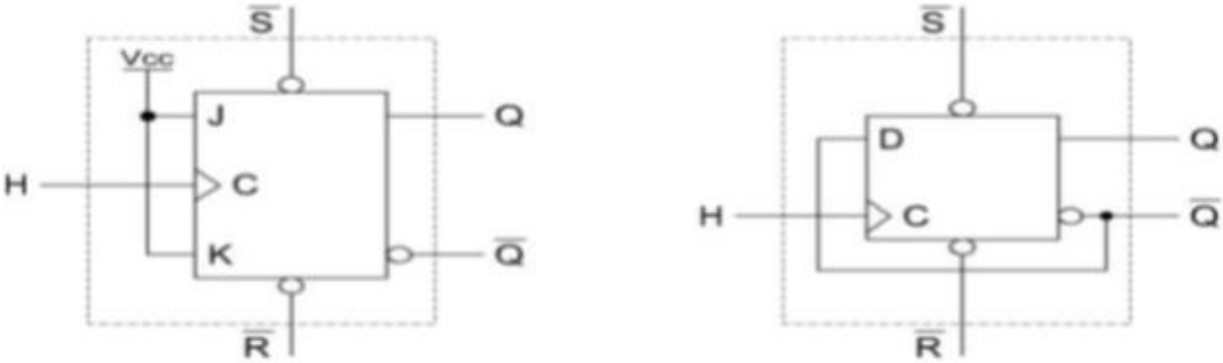
Décharger le condensateur permet de se replacer dans le même état qu'à l'allumage, et donc d'effectuer une remise à zéro. Pour décharger le condensateur, un simple bouton poussoir à ses bornes est suffisant. Un appui sur le bouton aura pour effet de court-circuiter le condensateur.

6. Remplacement des bascules JK par des bascules D

La cellule de base d'un compteur asynchrone est un diviseur de fréquence par deux. Il suffit donc de remplacer les bascules JK par des bascules D câblées en diviseur de fréquence par deux. C'est-à-dire de remplacer le câblage de la **figure 1** par le câblage de la **figure 2** ci-dessous:

Figure 155: Circuit de remplacement des bascules JK par des bascules

(a) (b)



WWW.DZPEN.COM | قلم الجزائر