

OFPPT

ROYAUME DU MAROC

مكتب التكوين المهني وإنعاش الشغل

Office de la Formation Professionnelle et de la Promotion du Travail

DIRECTION RECHERCHE ET INGENIERIE DE FORMATION

**RESUME THEORIQUE
&
GUIDE DE TRAVAUX PRATIQUES**

**MODULE N° 11 MAINTENANCE D'UN
SYSTEME SEQUENTIEL**

SECTEUR : ELECTROTECHNIQUE

SPECIALITE : MMOAMPA

NIVEAU : TECHNICIEN SPECIALISE

ANNEE 2008

Document élaboré par :

| Nom et prénom | EFP | DR |
|----------------------|------------|-----------|
| Mme ELKORNO NAIMA | CDC - GE | |

Révision linguistique

-
-
-

Validation

-
-
-

SOMMAIRE

| | |
|---|----|
| RESUME THEORIQUE | 7 |
| I. Les éléments de mémoires : | 8 |
| I.1 Concept de mémorisation : | 8 |
| I.2 Synchronisation des circuits : | 9 |
| I.3 Les bascules : | 10 |
| I.3.1 Bascules R S : | 10 |
| I.3.2 Bascule R S H (Bascule synchrone) : | 11 |
| I.3.3 Bascule J K synchrone : | 13 |
| I.3.4 Bascule D synchrone : | 14 |
| I.3.5 Bascule maître-esclave: | 15 |
| I.3.6 Bascule T : | 16 |
| I.3.7 Initialisation des bascules : | 16 |
| I.4 Monostable – Astable | 17 |
| II. Les compteurs : | 18 |
| II.1 Identification de la fonction : | 18 |
| II.2 Caractéristiques des compteurs : | 18 |
| II.2.1 Compteur Modulo 2 : | 18 |
| II.2.2 Compteur Modulo 2^N : | 19 |
| II.2.3 Compteur dont le modulo est différent de 2^N : | 19 |
| II.2.4 Compteur binaire asynchrone : | 19 |
| II.2.5 Compteur binaire synchrone : | 19 |
| II.3 Les compteurs asynchrones: | 19 |
| II.3.1 Compteur modulo 8 asynchrone : | 19 |
| II.3.2 Compteur modulo 10 asynchrone (compteur DCB) : | 20 |
| II.3.3 Décompteur modulo 8 asynchrone : | 22 |
| II.4 Les compteurs synchrones: | 22 |
| II.4.1 Compteur modulo 8 synchrone: | 22 |
| II.4.2 Compteur DCB (modulo 10) synchrone: | 24 |
| II.4.3 Décompteur modulo 8 synchrone: | 24 |
| II.5 Les compteurs intégrés: | 25 |
| III. Les registres: | 29 |
| III.1 Types de registres : | 29 |
| III.2 Registres à décalage : | 31 |
| III.3 Registres intégrés : | 33 |
| IV. Les Mémoires | 37 |
| IV.1 Introduction aux mémoires | 37 |
| IV.2 Fonctionnement des Mémoires | 38 |
| IV.2.1 Principes de fonctionnement | 38 |
| IV.2.2 Définitions | 38 |
| IV.2.3 Connexions Mémoires | 39 |
| IV.3 Mémoire Morte : | 40 |
| IV.3.1 Types de Mémoire Morte | 41 |
| IV.3.2 Exemples de mémoires mortes : | 42 |
| IV.4 Mémoires vives | 44 |
| IV.4.1 Mémoire VIVE Statique (SRAM) | 44 |

| | | |
|--|--|----|
| IV.4.2 | Mémoire VIVE Dynamique (DRAM) | 45 |
| IV.5 | Extension de la longueur du mot et de la capacité: | 46 |
| IV.5.1 | Extension de la longueur du mot. | 46 |
| IV.5.2 | Extension de la capacité..... | 48 |
| GUIDE DE TRAVAUX PRATIQUES | | 50 |
| Exercices : | | 51 |
| TP.1 – Essai des bascules élémentaires : | | 59 |
| TP.2 – Essai des registres à décalage:..... | | 63 |
| TP.3 – Essai des compteurs asynchrones:..... | | 65 |
| TP.4 – Essai des compteurs synchrones:..... | | 68 |
| TP.5 – Essai des compteurs intégrés : | | 71 |
| ÉVALUATION DE FIN DE MODULE : | | 74 |
| Liste bibliographique | | 76 |

MODULE 11 :

MAINTENANCE D'UN SYSTEME SEQUENTIEL

Code :

Durée : 70 h

OBJECTIF OPERATIONNEL

COMPORTEMENT ATTENDU

*Pour démontrer sa compétence le stagiaire doit
appliquer des notions de logique séquentielle
Selon les conditions, les critères et les précisions qui suivent.*

CONDITIONS D'EVALUATION

- A partir :
 - de directives ;
 - d'un schéma.
- A l'aide :
 - de manuels techniques;
 - de fiches techniques ;
 - de composants logiques ;
 - d'instruments de mesure.

CRITERES GENERAUX DE PERFORMANCE

- *Respect des règles de santé et de sécurité au travail.*
- *Pertinence de l'utilisation des outils et des instruments.*
- *Pertinence de la terminologie utilisée.*
- *Qualité des travaux.*

OBJECTIF OPERATIONNEL

**PRECISIONS SUR LE
COMPOTEMENT ATTENDU**

**CRITERES PARTICULIERS DE
PERFORMANCE**

- | | |
|---|--|
| <p>A) Analyser le fonctionnement d'un circuit numérique à bases de bascules.</p> <p>B) Distinguer les différents types de registres</p> <p>C) Utiliser et vérifier le fonctionnement des différents types de compteurs</p> <p>D) Analyser et dépanner une carte électronique contenant des circuits séquentiels.</p> <p>E) Modifier la taille de la zone mémoire d'une unité centrale.</p> <p>F) Programmer correctement une mémoire EPROM et analyser son fonctionnement dans une carte électronique</p> | <ul style="list-style-type: none">- Identification juste des symboles.- Distinction exacte des caractéristiques des bascules.- Utilisation correcte des bascules. - Distinction exacte des caractéristiques des registres.- Utilisation correcte des registres. - Distinction exacte des caractéristiques des compteurs.- Utilisation correcte des compteurs. - Lecture et interprétation correctes d'un schéma à base de circuit séquentiel.- Localisation et remplacement opérationnels de la partie défectueuse.- Fonctionnement correct du circuit. - Choix approprié de la mémoire.- Branchement fonctionnel des mémoires. - Programmation correcte d'un mémoire EPROM.- Lecture et interprétation exactes d'un schéma, à bases de mémoires. |
|---|--|

Présentation du Module :

Ce module de logique séquentielle constitue la suite du module sur la logique combinatoire. Son objectif est de faire acquérir aux stagiaires des connaissances relatives aux éléments de mémoire <les bascules>, aux compteurs, aux registres et aux mémoires. Dans un même temps le stagiaire aura l'occasion de faire l'étude de montages de circuits de base en logique séquentielle. Il vise donc à rendre le stagiaire apte à appliquer des notions de logique séquentielle.

MODULE N° 11: MAINTENANCE D'UN SYSTEME SEQUENTIEL

RESUME THEORIQUE

I. Les éléments de mémoires :

I.1 Concept de mémorisation :

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 (voir figure 1.1): ils sont dits séquentiels 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'une seule donnée).

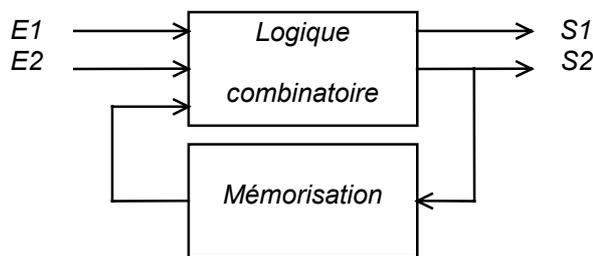


Figure 1.1 – Schéma général d'un circuit séquentiel

Exemple :

On prend l'exemple d'un poste marche-arrêt qui commande le fonctionnement d'un moteur. Une action momentanée sur le bouton «< marche >> met le moteur en fonction aussi longtemps que le bouton-poussoir «< arrêt >> n'est pas actionné. Dans le tableau de la figure 1.2, on remarque que les variables d'entrée des étapes 1 et 3 ont la même valeur, mais que l'état de la sortie est différent ; un dispositif de mémoire a maintenu le moteur en marche. Il devient donc impossible de construire une table de karnaugh comme en logique combinatoire et de réaliser le circuit à l'aide de simples portes logiques.

| étape | Bouton-poussoir «< marche >> | Bouton-poussoir «< arrêt >> | Moteur |
|-------|------------------------------|-----------------------------|--------|
| 1 | 0 | 0 | 0 |
| 2 | 1 | 0 | 1 |
| 3 | 0 | 0 | 1 |
| 4 | 0 | 1 | 0 |
| 5 | 0 | 0 | 0 |

Figure 1.2 – Poste marche-arrêt

On peut dire que le concept de mémorisation est l'élément fondamental de la logique séquentielle.

1.2 Synchronisation des circuits :

Il existe des circuits synchrones et asynchrones. La différence entre ces deux catégories se situe au regard du synchronisme des actions. Un circuit asynchrone peut changer d'état à tout moment selon les variables d'entrée. Par contre, un circuit synchrone incorpore un signal d'horloge qui sert à enclencher les actions

a) Circuit séquentiel asynchrone :

Dans un circuit séquentiel asynchrone, le changement de l'état logique de la sortie peut s'effectuer à n'importe quel moment selon le changement des variables d'entrée. La détermination de l'état logique du système s'effectue instantanément, d'où la difficulté de concevoir, et surtout de déboguer des circuits asynchrones.

b) Circuit séquentiel synchrone :

Dans un circuit séquentiel synchrone, un signal d'horloge commande le changement de l'état logique de la sortie. Le système est alors susceptible de changer d'état uniquement à des moments précis déterminés par l'horloge. Cela facilite la conception et le débogage du circuit par une analyse pas à pas dictée par chaque coup de l'horloge. Il faut noter que les variables doivent être constantes et non fugitives au moment précis du coup de l'horloge, sinon elles ne seront pas prises en considération. (L'expression <<variables constantes>> signifie que les variables sont dans un état stable).

c) Signal d'horloge :

L'horloge génère habituellement un train d'ondes carrées d'une durée fixe, comme le présente la figure 1.3. cette durée s'appelle période et se mesure en secondes.

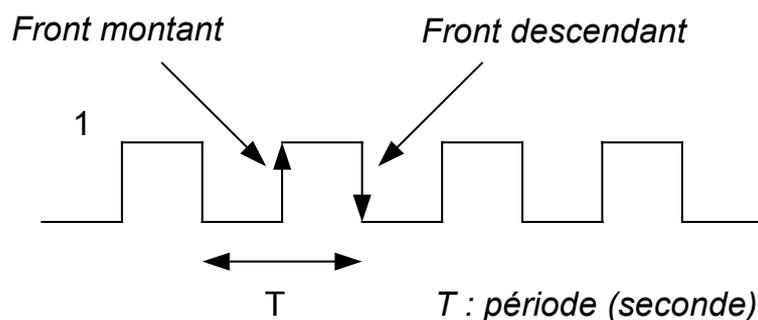


Figure 1.3 – Signal d'horloge

Il importe de distinguer les niveaux et les transitions d'une onde carrée. On connaît déjà le niveau BAS (état logique 0) et le niveau HAUT (état logique 1) associés aux états stables d'une onde. Les transitions correspondent aux changements d'un état stable vers un autre. On désigne le signal ascendant par l'expression front montant (transition positive de 0 à 1) et le signal descendant, par front descendant (transition négative de 1 à 0).

I.3 Les bascules :

La bascule est un circuit bistable pouvant prendre deux états logiques "0" ou "1". L'état de la bascule peut être modifié en agissant sur une ou plusieurs entrées. Le nouvel état de la bascule dépend de l'état précédent, c'est l'élément de base des circuits séquentiels. La bascule peut conserver son état pendant une durée quelconque, elle peut donc être utilisée comme mémoire.

I.3.1 Basculés RS :

Une bascule RS peut être réalisée par l'association d'opérateurs NON-OU (NOR) ou NON-ET (NAND).

La bascule RS présente : (voir figure 1.4)

- S : entrée de mise à 1 (SET) de Q ;
- R : entrée de mise à 0 (RESET) de Q ;
- Q et \bar{Q} : sorties complémentaires ;

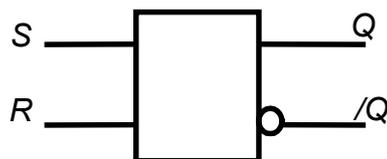


Figure 1.4 - Symbole d'une bascule RS

a) Bascule RS à opérateurs NON-OU : (Voir Figure 1.5)

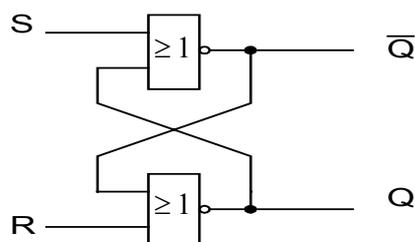


Figure 1.5 – Bascule RS à opérateurs NON-OU

Table de fonctionnement : (Voir figure 1.6)

| Entrées | | Sorties | |
|---------|---|-----------|-----------|
| R | S | Q | \bar{Q} |
| 0 | 0 | Inchangé | |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | Ambiguïté | |

Figure 1.6 - Table de vérité de la bascule RS à opérateurs NON-OU

b) Bascule RS à opérateurs NON-ET : (Voir figure 1.7)

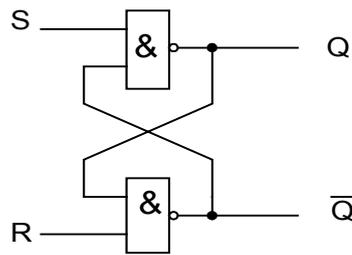


Figure 1.7 - Bascule RS à opérateurs NON-ET

Table de fonctionnement : (Voir figure 1.8)

| Entrées | | Sorties | |
|---------|---|-----------|-----------|
| R | S | Q | \bar{Q} |
| 0 | 0 | Ambiguïté | |
| 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 1 | 1 | Inchangé | |

Figure 1.8 - Table de vérité de la bascule RS à opérateurs NON-ET

1.3.2 Bascule R S H (Bascule synchrone) :

Dans la bascule RS, la sortie change d'état, au temps de propagation près, au moment où la combinaison des états des entrées est changée, son mode de fonctionnement est asynchrone.

Dans une bascule synchrone RSH le changement d'état de la sortie qui correspond à une nouvelle combinaison d'états d'entrées ne peut s'effectuer que sur le front actif, montant ou descendant, d'un signal d'horloge (voir figure 1.9).

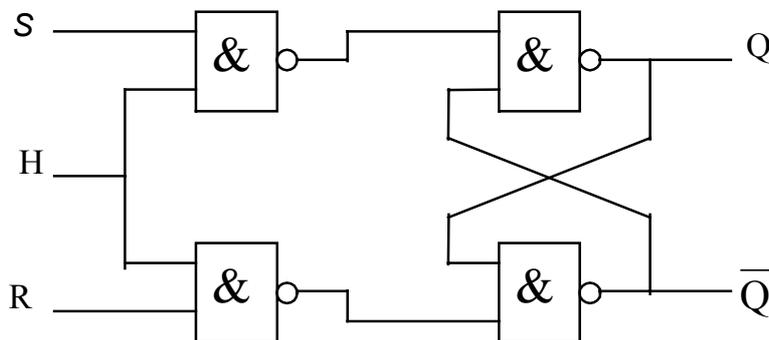
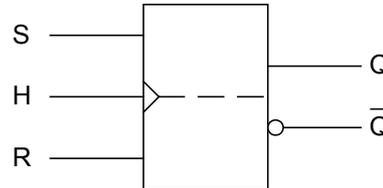


figure 1.9 - Réalisation de la bascule RSH avec des opérateurs NON-ET

La bascule RSH comprend : (voir figure 1.10).

- Trois entrées :
 - S : mise à 1 ;
 - R : mise à 0 ;
 - H : entrée d'horloge, active sur le front montant ou descendant du signal ;
- Deux sorties : Q et \bar{Q} dont les états sont complémentaires ;

Bascule synchrone active sur le front montant du signal d'horologe



Bascule synchrone active sur le front descendant du signal d'horologe

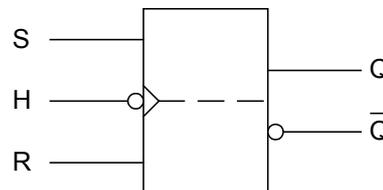


Figure 1.10 – Symboles d'une bascule RSH

Table de fonctionnement : (Voir figure 1.11)

| Entrées | | | Sorties | | Mode de fonctionnement de la bascule |
|---------|---|---|------------------|--------------------------|---|
| H | S | R | Q _{n+1} | \bar{Q} _{n+1} | |
| | 0 | 0 | Q _n | \bar{Q} _n | Mémorisation de l'état précédent (inchangé) |
| | 1 | 0 | 1 | 0 | Mise à 1 |
| | 0 | 1 | 0 | 1 | Mise à 0 |
| | 1 | 1 | Ambiguïté | | Les états de sorties sont indéterminés ne pas utiliser. |

Figure 1.11 - Table de vérité de la bascule RSH

Exemple de bascule RSH déclenchée par front montant \uparrow : voir figure 1.12

| R | S | Q | \bar{Q} |
|---|---|----------|-----------|
| 0 | 0 | x | \bar{x} |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | Interdit | |

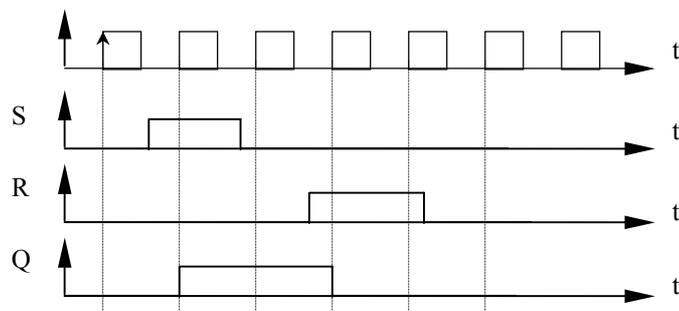


figure 1.12 - Table de vérité et chronogramme de bascule RSH déclenchée par front montant \uparrow

1.3.3 Bascule J K synchrone :

La bascule J K synchrone (simple étage) est obtenue à partir d'une bascule R S H dont les sorties sont rebouclées sur les entrées. Ceci permet d'éliminer l'état indéterminé (voir figure 1.13).

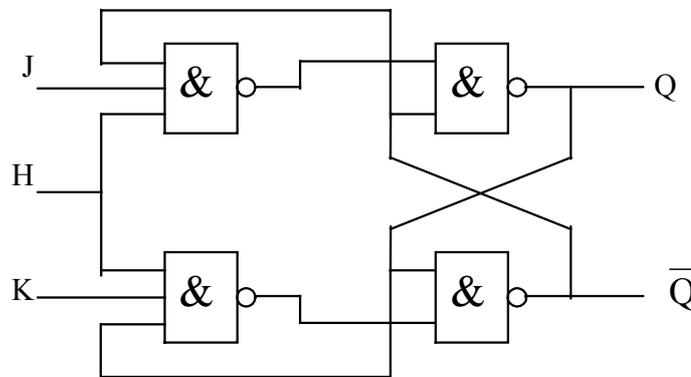


figure 1.13 – bascule JK réalisée avec les portes Nand

La bascule JK présente : (voir figure 1.14).

- Deux entrées J et K ;
- Une entrée d'horloge H ;
- Deux sorties Q et \bar{Q} dont les états sont complémentaires ;

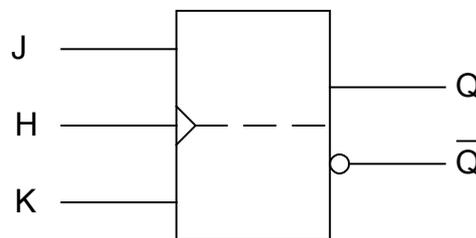


Figure 1.14 – Symbole d'une bascule JK

Fonctionnement d'une bascule JK :

Les entrées J et K de ce type de bascule ont le même rôle que les entrées S et R de la bascule RSH à la différence que la condition $J = K = 1$ n'est pas une condition ambiguë sur l'état de Q et \bar{Q} .

L'état $J = K = 1$ provoque le changement d'état de la sortie Q ou un basculement successif à chaque top d'horloge. Il est utilisé dans de nombreux systèmes numériques.

Exemple de bascule JK déclenchée par front montant \uparrow : voir figure 1.15

| | | | |
|---|---|-----------|-----------|
| K | J | Q | \bar{Q} |
| 0 | 0 | x | \bar{x} |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | \bar{x} | x |

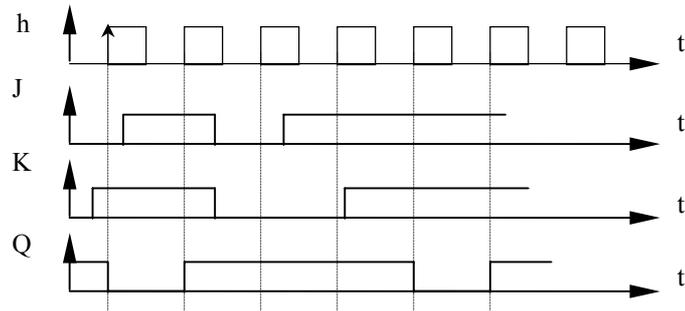


figure 1.15 - Table de vérité et chronogramme de bascule JK déclenchée par front montant \uparrow

1.3.4 Bascule D synchrone :

Une bascule D est réalisée à partir d'une bascule R S ou J K dont les entrées sont reliées par un inverseur. Ceci impose donc que les entrées prennent des états complémentaires.

Réalisation: (voir figure 1.16)

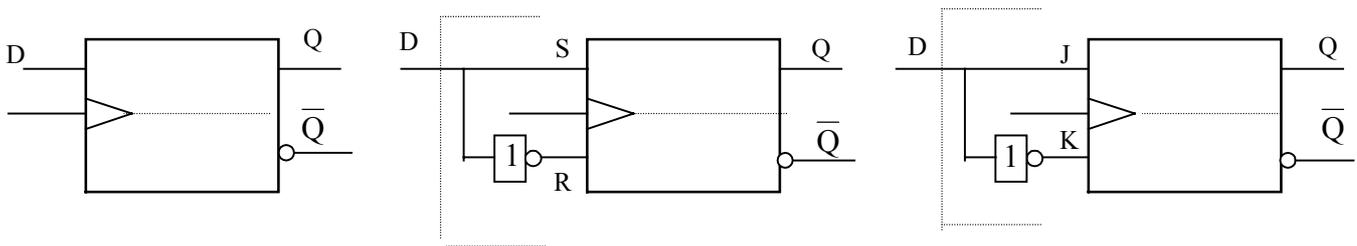


figure 1.16 – Réalisation de la bascule D

Exemple de bascule D déclenchée par front montant \uparrow : voir figure 1.17

| | |
|---|---|
| D | Q |
| 0 | 0 |
| 1 | 1 |

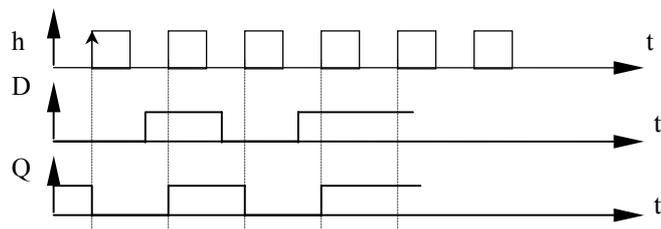


figure 1.17 - Table de vérité et chronogramme de bascule D déclenchée par front montant \uparrow

La table de vérité met bien en évidence que la sortie Q recopie l'état de l'entrée D sur le front actif du signal d'horloge, ici le front montant. Ce type de bascule, à déclenchement sur front actif du signal d'horloge, est très utilisé : Compteurs, mémoire tampon,...

1.3.5 Bascule maître-esclave:

Problème: Les bascules synchrones nécessitent des états stables sur leurs entrées au moment de la transition du signal d'horloge, cela n'est pas toujours possible lorsque plusieurs bascules sont câblées entre elles (exemple: en comptage) et l'on a des aléas de fonctionnement (voir figure 1.18).

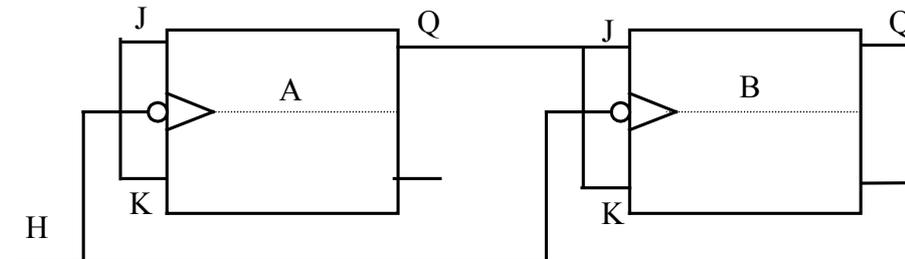
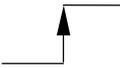
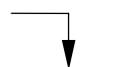


Figure 1.18 – Association de bascules synchrones

Solution: Il existe des bascules à 2 étages qui évoluent en 2 temps.

1er temps:  Verrouillage du 2^{ème} étage
Prise en compte des entrées par le 1^{er} étage

2ème temps  Verrouillage du 1^{er} étage
Prise en compte des données par le 2^{ème} étage

Exemple : bascule J K Maître-Esclave : (voir figure 1.19)

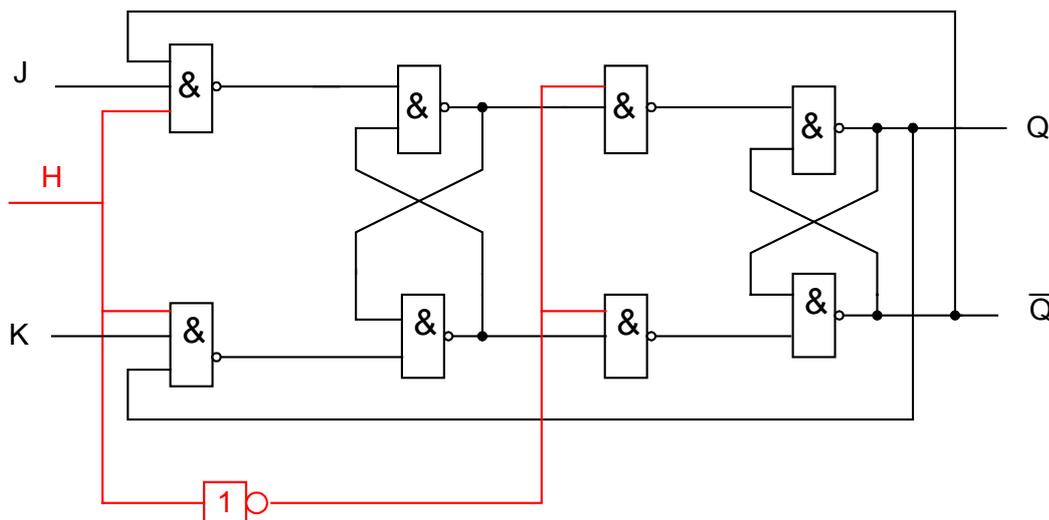


Figure 1.19 - bascule J K Maître-Esclave :

1.3.6 Bascule T :

La bascule T présente : (voir figure 1.20)

- Une entrée d'horloge H ;
- Deux sorties Q et \bar{Q} dont les états sont complémentaires.

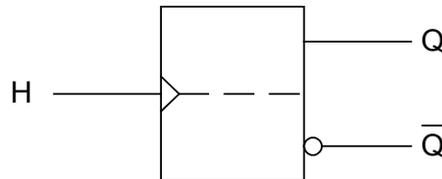


Figure 1.20 – Symbole d'une bascule T

Table de fonctionnement :

| Entrée H | Sortie | | Modes de fonctionnement de la bascule |
|-------------|-----------|-----------|---------------------------------------|
| | Q_{n+1} | Q_{n+1} | |
| | Q_n | Q_n | Changement d'état |
| | Q_n | Q_n | |

Quand l'entrée H passe à l'état dynamique 1, les sorties changent d'état.
Quand l'entrée H passe à l'état 0, les sorties restent dans leur état.

1.3.7 Initialisation des bascules :

Les bascules RSH, JK et D ont un fonctionnement synchrone par rapport à un signal d'horloge. Leurs entrées de commande R, S, J, K et D sont des entrées synchrones. Pour le fonctionnement d'un système, il est souvent nécessaire que les bascules soient initialisées, c'est à dire que leur sortie Q soit à 1 ou à 0 et ce indépendamment du signal d'horloge. D'où, deux entrées supplémentaires asynchrones, présentes sur pratiquement tous les circuits intégrés :

- Preset : mise à 1 de la sortie Q
- Clear : mise à 0 de la sortie Q.

Ces deux entrées asynchrones sont désignées entrées d'initialisation ou de forçage.
Exemple pour la bascule JK :(Voir figure 1.21)

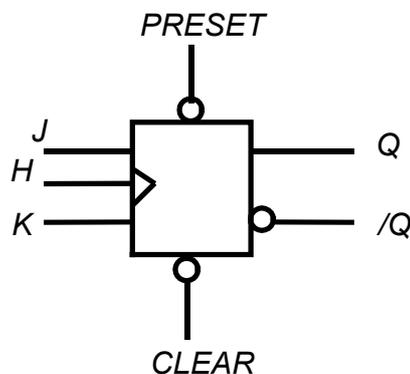


Figure 1.21 - la bascule JK avec les entées de forçage

Remarque : La négation logique sur les deux entrées asynchrones PRESET et CLEAR indique qu'elles sont actives sur le niveau bas du signal qui leur est appliqué.

Table de vérité (voir figure 1.22).

| Preset | Clear | H | Q |
|--------|-------|---|--|
| 0 | 0 | X | A ne pas utiliser |
| 0 | 1 | X | 1 |
| 1 | 0 | X | 0 |
| 1 | 1 | ↑ | Fonctionnement synchrone de la bascule |

Figure 1.22 – Table de vérité de la bascule JK avec les entrées de forçage

I.4 Monostable – Astable

a) Monostable :

C'est une microstructure séquentielle, qui en sortie possède deux états complémentaires l'un de l'autre.

- L'un des deux états étant stable.
- L'autre ne pouvant être occupé que momentanément [état pseudo-stable].

Une impulsion à l'entrée (c-à-d changement d'état de 0 à 1) amène la sortie à l'état 1, la sortie reste dans cet état pendant une durée définie par les caractéristiques particulières de l'opérateur, indépendamment du temps pendant lequel l'entrée reste à l'état 1 puis revient à l'état 0.

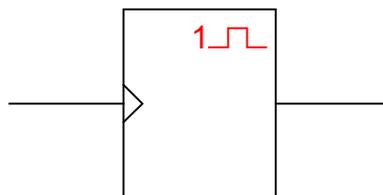


Figure 1.23 - Symbole d'un monostable

b) Astable ou oscillateur :

Microstructure séquentielle qui, en sortie, possède deux états pseudo-stables (complémentaire l'un de l'autre) ; le passage d'un état à l'autre s'effectuant périodiquement avec un facteur de forme quelconque.

Dans certaines utilisations, les bascules astables peuvent prendre le nom de multivibrateurs.

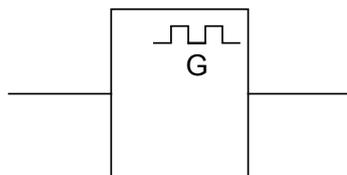


Figure 1.24 - Symbole d'un Astable

II. Les compteurs :

II.1 Identification de la fonction :

La fonction comptage existe dans de nombreux systèmes dans lesquels le résultat d'un calcul effectué :

- sur une série d'objets,
- ou sur la répétition d'un événement

déclenche :

- une décision prise par l'utilisateur,
- Ou une action gérée automatiquement par le système.

Exemples :

- Un poste de sciage s'arrête après avoir débité 20 pièces d'un même lot.
- Dans un système à programmation chronologique, le comptage du temps écoulé permet de déclencher automatiquement certaines opérations : arrosage d'un jardin, mise en marche d'un appareil électroménager, allumage d'un feu de signalisation...

Un système de comptage comprend toujours un compteur.

Un compteur est un système logique dont le mot binaire en sortie se modifie chaque fois qu'une information est appliquée à son entrée.

Suivant qu'une nouvelle impulsion incrémente ($n+1$) ou décrémente ($n-1$) la valeur binaire du mot de sortie, le circuit fonctionne en compteur ou en décompteur.

II.2 Caractéristiques des compteurs :

La caractéristique principale d'un compteur est sa capacité de comptage. Cette capacité de comptage détermine le nombre de bits du mot binaire de sortie.

II.2.1 Compteur Modulo 2 :

Le mot binaire de sortie du compteur le plus simple n'a qu'un bit qui ne peut donc prendre que l'état logique 0 ou 1 lorsque les impulsions à compter se succèdent sur l'entrée du compteur. Le mot de sortie est à l'état logique 1 toutes les deux impulsions, le compteur divise par 2 le nombre des impulsions d'entrée. Cette caractéristique de division est désignée par le modulo du compteur qui dans ce cas est 2. La structure d'un compteur modulo 2 est une bascule :

- du type D,
- ou du type JK,

II.2.2 Compteur Modulo 2^N :

Un compteur modulo 2 étant constitué d'une bascule, il est possible :

- d'associer 2 bascules pour réaliser un compteur modulo 4 soit 2^2
- d'associer 3 bascules pour réaliser un compteur modulo 8 soit 2^3

II.2.3 Compteur dont le modulo est différent de 2^N :

Il est toujours possible de réaliser un compteur dont le modulo est différent de 2^N . Il suffit de forcer sa réinitialisation à zéro avec le mot de sortie qui suit immédiatement la dernière impulsion comptée.

De plus, les compteurs binaires sont classés en 2 catégories :

- Les compteurs asynchrones,
- Les compteurs synchrones.

II.2.4 Compteur binaire asynchrone :

Dans ce type de structure, l'impulsion de progression du compteur est appliquée sur l'entrée d'horloge du premier étage, les entrées d'horloge des autres bascules reçoivent le signal de sortie de l'étage précédent.

II.2.5 Compteur binaire synchrone :

Dans la structure synchrone, l'horloge est la même pour tous les étages : le basculement de toutes les bascules se fait en même temps.

II.3 Les compteurs asynchrones:

Le terme « asynchrone » se définit comme l'absence de synchronisme des signaux qui déclenchent la commande d'un circuit. Comme les entrées d'horloge remplissent habituellement cette fonction, les compteurs asynchrones ne possèdent pas de ligne commune qui relie toutes les entrées d'horloge des bascules. Par conséquent, on branche le signal d'horloge uniquement à la première bascule. Le signal de sortie de cette première bascule sert d'horloge à la seconde bascule et ainsi de suite.

Les compteurs asynchrones sont les plus simples à concevoir. Toute fois, le délai de propagation qu'ils engendrent provoque des imprécisions importantes. On évite donc leur utilisation pour la mesure précise du temps.

II.3.1 Compteur modulo 8 asynchrone :

On réalise ce compteur en branchant en cascade trois bascules JK. La sortie de la première bascule devient l'entrée d'horloge de la deuxième bascule, la sortie de la deuxième bascule devient l'horloge de la troisième bascule (voir figure 2.1).

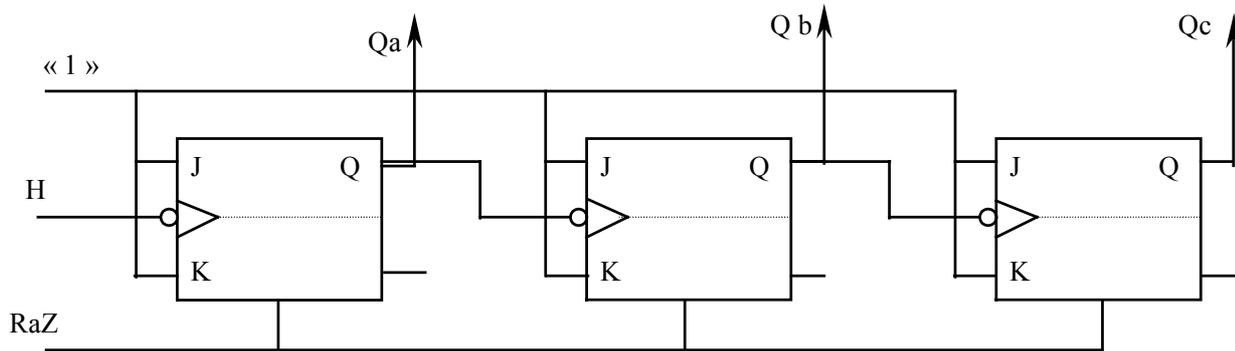


Figure 2.1 - Compteur modulo 8 asynchrone

Table de vérité et chronogramme du Compteur modulo 8 asynchrone (voir figure 2.2).

| N | Qc | Qb | Qa |
|---|----|----|----|
| 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 |
| 2 | 0 | 1 | 0 |
| 3 | 0 | 1 | 1 |
| 4 | 1 | 0 | 0 |
| 5 | 1 | 0 | 1 |
| 6 | 1 | 1 | 0 |
| 7 | 1 | 1 | 1 |

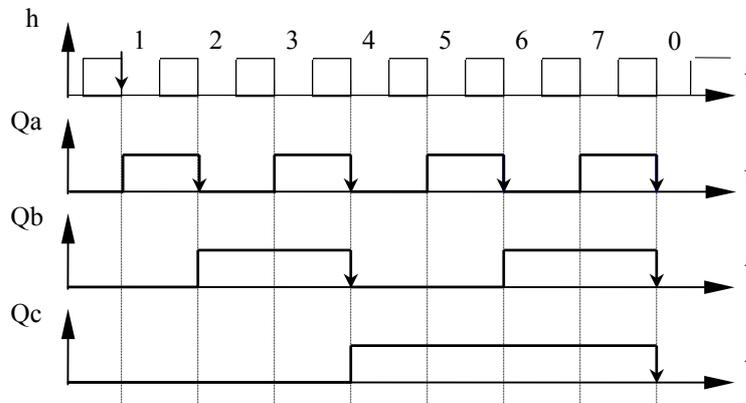


Figure 2.2 - Table de vérité et chronogramme du compteur modulo 8 asynchrone

II.3.2 Compteur modulo 10 asynchrone (compteur DCB) :

Un compteur modulo 10 possède dix états, qui correspondent aux équivalents binaires des nombres de 0 à 9. Sa réalisation nécessite l'utilisation de quatre bascules. De même il peut diviser la fréquence d'entrée par dix. Un compteur DCB dénombre les états de $(0000)_2$ à $(1001)_2$. Il est très répandu, car il établit le lien avec les dispositifs d'affichage numériques qui permettent de représenter les nombres décimaux de 0 à 9 par l'intermédiaire du code DCB. Il suffit de raccorder la sortie des bascules à un décodeur DCB qui pilote un afficheur à sept segments. (voir figure 2.3 et 2.4)

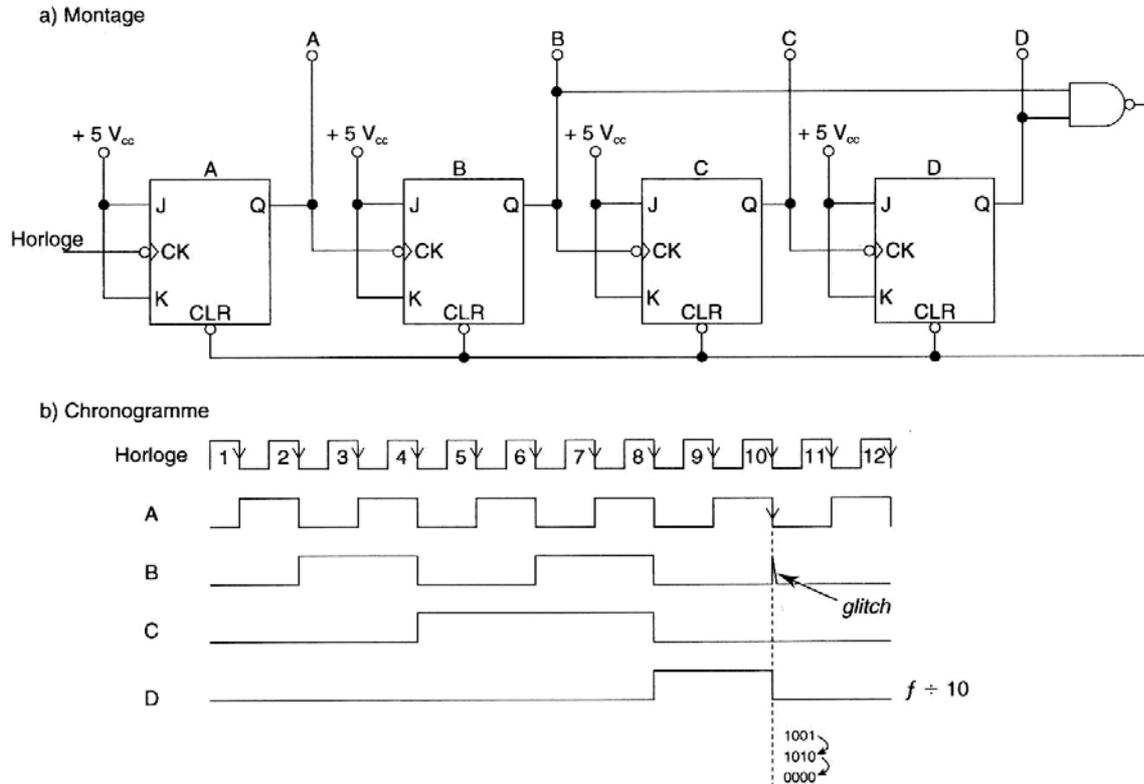


Figure 2.3 - Compteur modulo-10 asynchrone

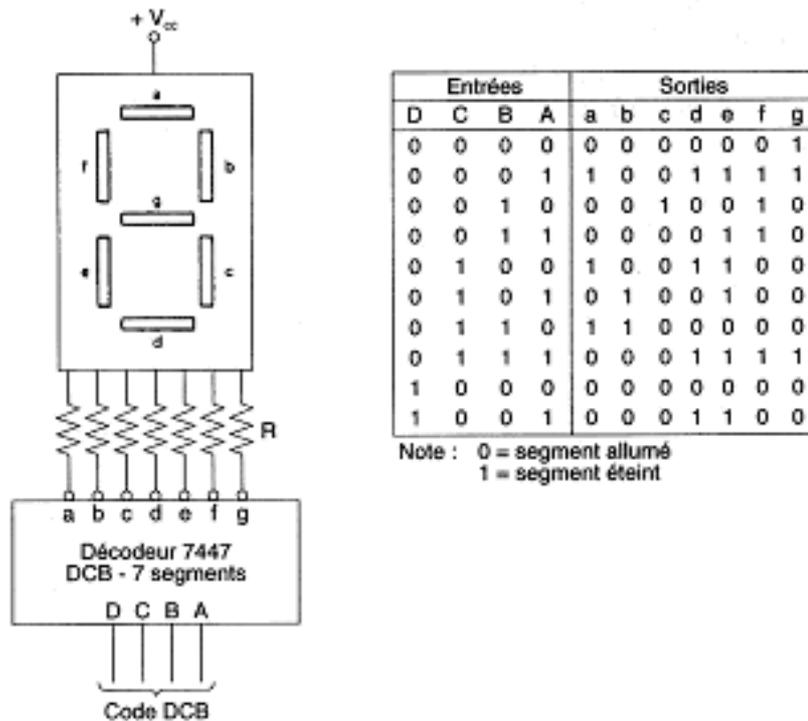


Figure 2.4 – Décodeur 7447 et afficheur à anode commune

II.3.3 Décompteur modulo 8 asynchrone :

Pour obtenir un décompteur, il faut regarder les sorties $\overline{Q_i}$ ou brancher les sorties $\overline{Q_i}$ de chaque bascule sur l'horloge de la bascule suivante et regarder l'évolution des sorties Q_i , (voir figure 2.5)

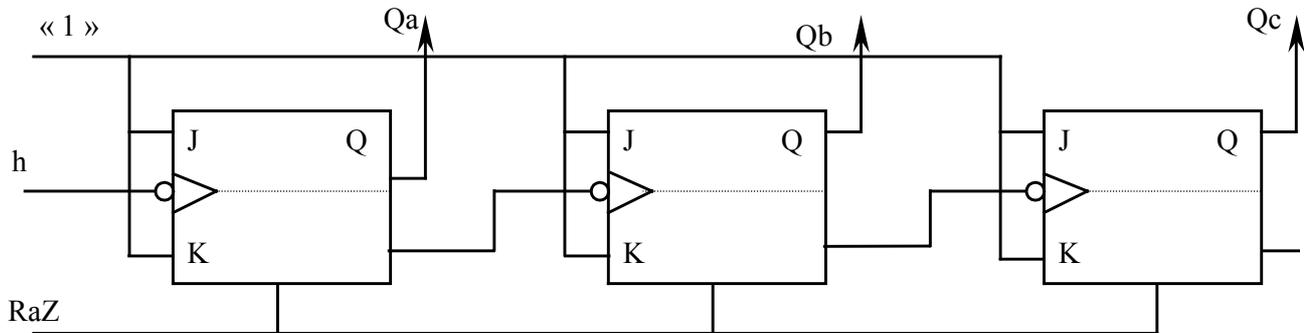


Figure 2.5 - décompteur modulo 8 asynchrone

Table de vérité et chronogramme du décompteur modulo 8 asynchrone (voir figure 2.6)

| N | Qc | Qb | Qa |
|---|----|----|----|
| 7 | 1 | 1 | 1 |
| 6 | 1 | 1 | 0 |
| 5 | 1 | 0 | 1 |
| 4 | 1 | 0 | 0 |
| 3 | 0 | 1 | 1 |
| 2 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 |

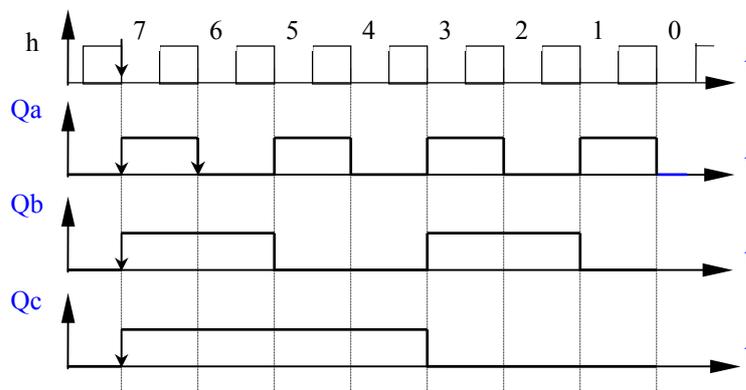


Figure 2.6 - Table de vérité et chronogramme du décompteur modulo 8 asynchrone

II.4 Les compteurs synchrones:

II.4.1 Compteur modulo 8 synchrone:

L'impulsion d'horloge est appliquée simultanément à chaque bascule. Celles-ci évoluent en fonction des informations présentant sur leurs entrées J, K au moment où apparaît l'impulsion. Il faut donc prépositionner J et K à l'instant t pour obtenir le basculement désiré à l'instant t+1 (voir figure 2.7).

| | | |
|---|---|-----------|
| J | K | Q |
| 0 | 0 | Q |
| 0 | 1 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | \bar{Q} |

| | |
|----------------|------------------|
| Q _n | Q _{n+1} |
| 0 | → 1 |
| 1 | → 0 |
| 1 | → 1 |
| 0 | → 0 |

| | |
|---|---|
| J | K |
| 1 | x |
| x | 1 |
| x | 0 |
| 0 | x |

Figure 2.7 – la table de vérité de la bascule JK entre l'état n et n+1

Si Q_C, Q_B, Q_A, sont les sorties de trois bascules on a la table de vérité suivante:

| N | Q _c | Q _b | Q _a | J _c | K _c | J _b | K _b | J _a | K _a |
|---|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| 0 | 0 | 0 | 0 | 0 | x | 0 | x | 1 | x |
| 1 | 0 | 0 | 1 | 0 | x | 1 | x | x | 1 |
| 2 | 0 | 1 | 0 | 0 | x | x | 0 | 1 | x |
| 3 | 0 | 1 | 1 | 1 | x | x | 1 | x | 1 |
| 4 | 1 | 0 | 0 | x | 0 | 0 | x | 1 | x |
| 5 | 1 | 0 | 1 | x | 0 | 1 | x | x | 1 |
| 6 | 1 | 1 | 0 | x | 0 | x | 0 | 1 | x |
| 7 | 1 | 1 | 1 | x | 1 | x | 1 | x | 1 |

Figure 2.8 – la table de vérité du Compteur modulo 8 synchrone

Equations des entrées J et K :

On peut obtenir ces équations en utilisant le tableau de KARNAUGH

$$J_A = K_A = 1$$

$$J_B = K_B = Q_a$$

$$J_C = K_C = Q_a \cdot Q_b$$

Schéma: (voir figure 2.9)

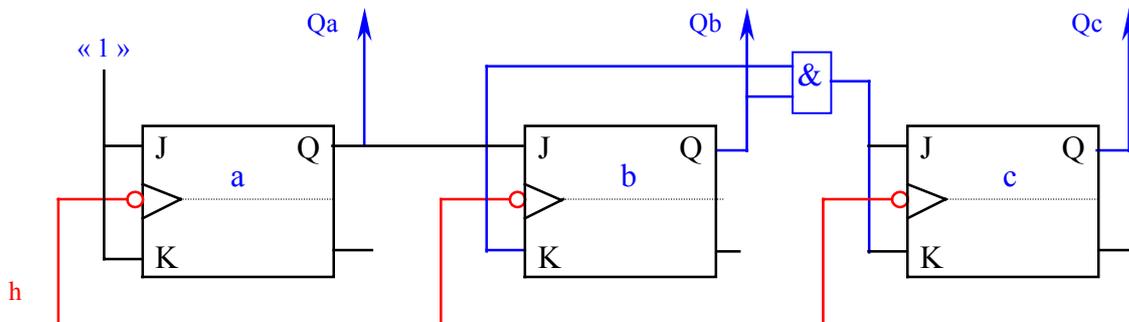


Figure 2.9 - Compteur modulo 8 synchrone

II.4.2 Compteur DCB (modulo 10) synchrone:

Par le même raisonnement on peut obtenir le compteur modulo 10 synchrone (figure 2.10)

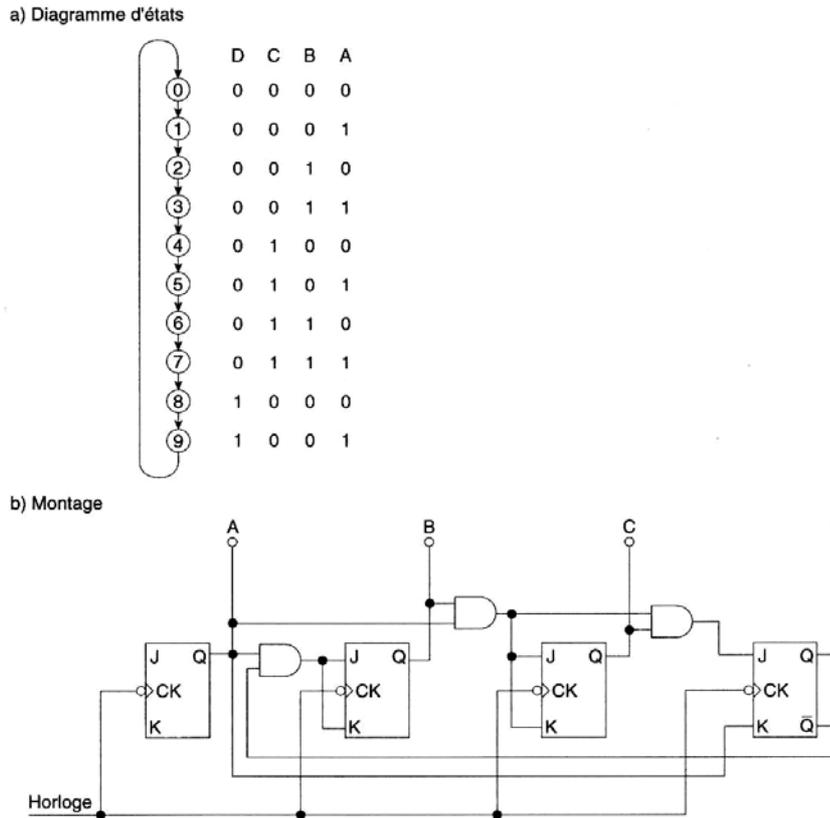


Figure 2.10 - Compteur modulo 10 synchrone

II.4.3 Décompteur modulo 8 synchrone:

Par le même raisonnement on peut obtenir (voir figure 2.11)

| N | Qc | Qb | Qa | Jc | Kc | Jb | Kb | Ja | Ka |
|---|----|----|----|----|----|----|----|----|----|
| 7 | 1 | 1 | 1 | x | 0 | x | 0 | x | 1 |
| 6 | 1 | 1 | 0 | x | 0 | x | 1 | 1 | x |
| 5 | 1 | 0 | 1 | x | 0 | 0 | x | x | 1 |
| 4 | 1 | 0 | 0 | x | 1 | 1 | x | 1 | x |
| 3 | 0 | 1 | 1 | 0 | x | x | 0 | x | 1 |
| 2 | 0 | 1 | 0 | 0 | x | x | 1 | 1 | x |
| 1 | 0 | 0 | 1 | 0 | x | 0 | x | x | 1 |
| 0 | 0 | 0 | 0 | 1 | x | 1 | x | 1 | x |

Figure 2.11 – la table de vérité du décompteur modulo 8 synchrone

Equations des entrées J et K :

On peut obtenir ces équations en utilisant le tableau de KARNAUGH

$$J_A = K_A = 1.$$

$$J_B = K_B = \overline{Q_a}$$

$$J_C = K_C = \overline{Q_a} \cdot \overline{Q_b}$$

Schéma: (voir figure 2.12)

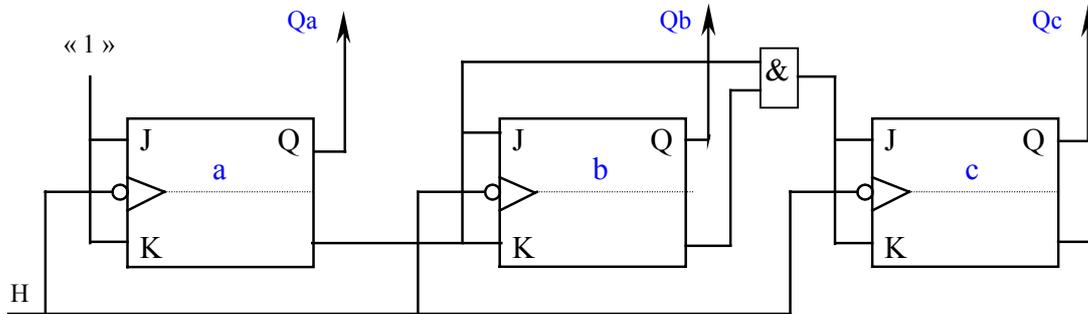


Figure 2.12 - Décompteur modulo 8 synchrone

II.5 Les compteurs intégrés:

On prend comme exemple le circuit intégré 7490. Ce compteur peut servir de diviseur à décade (diviseur de fréquence par dix) symétrique, de compteur MODULO, et surtout de compteur DCB pour les dispositifs d'affichage.

Principe de fonctionnement :

Il renferme quatre bascules et un ensemble de porte logique pour créer la séquence DCB. Il possède de plus des entrées doubles de remise à zéro (R0) et der remise à neuf (R9) asynchrones.

D'autre part, comme le montre la figure 2.13 La bascule A n'a pas de connexion interne avec les trois autres bascules. Il faut alors raccorder les deux sections de bascules selon la fonction à réaliser

Brochage

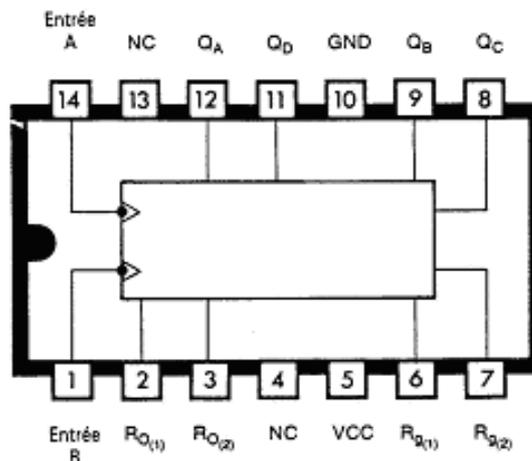


Figure 2.13 – Compteur 7490

Schéma logique
interne

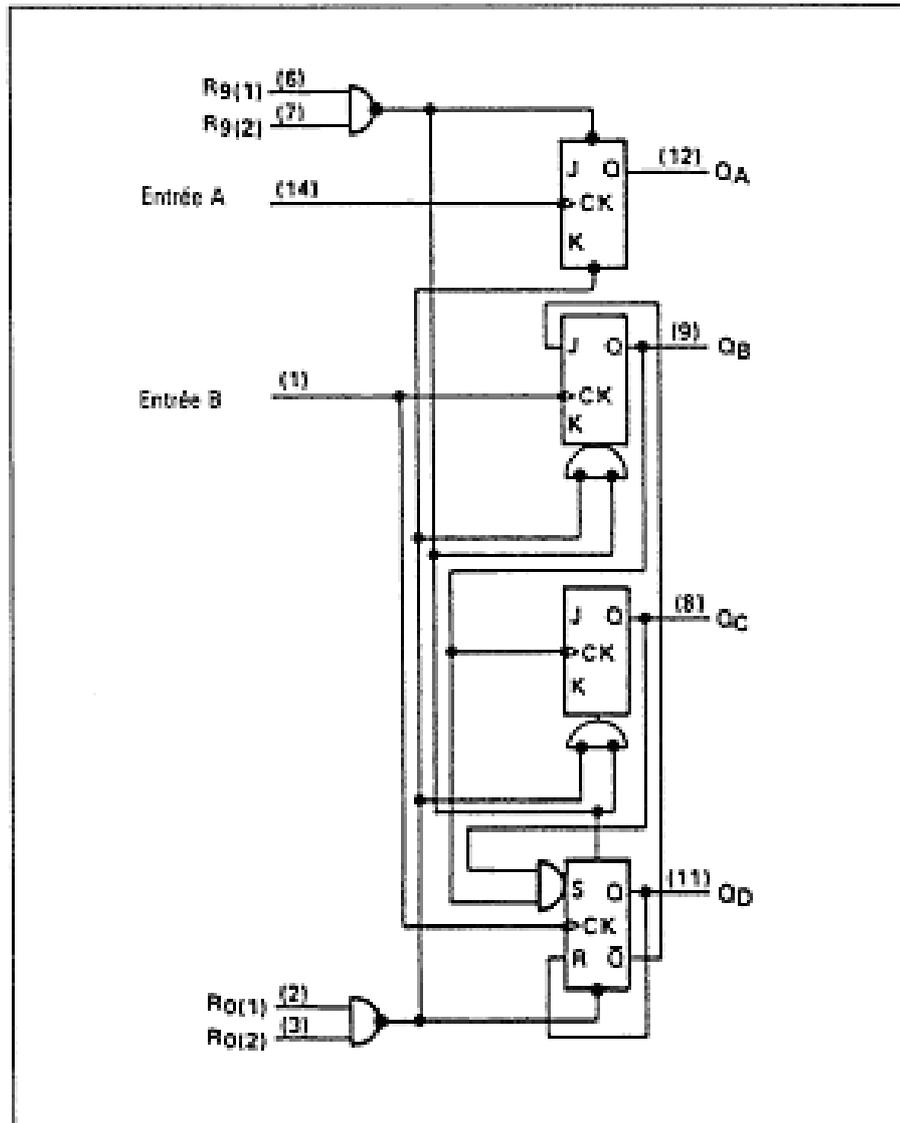


Figure 2.13 – Compteur 7490 (suite)

a) **Diviseur par dix symétrique :**

Les diviseurs à décade par dix n'ont pas une onde de sortie symétrique. Le C I 7490 permet de corriger cette situation.

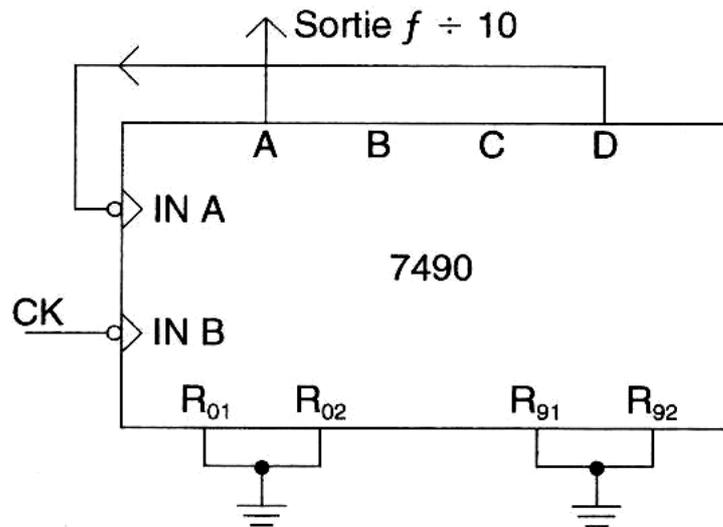
On sait que qu'un diviseur par deux est symétrique. Il devient possible de produire cette symétrie avec le C I 7490 en inversant simplement l'ordre de succession des bascules (voir figure 2.14)

a) Table de vérité

'90A, 'L90, 'LS90
BI-QUINARY (5-2)
(See Note B)

| COUNT | OUTPUT | | | |
|-------|----------------|----------------|----------------|----------------|
| | Q _A | Q _D | Q _C | Q _B |
| 0 | L | L | L | L |
| 1 | L | L | L | H |
| 2 | L | L | H | L |
| 3 | L | L | H | H |
| 4 | L | H | L | L |
| 5 | H | L | L | L |
| 6 | H | L | L | H |
| 7 | H | L | H | L |
| 8 | H | L | H | H |
| 9 | H | H | L | L |

b) Branchement



b) Chronogramme

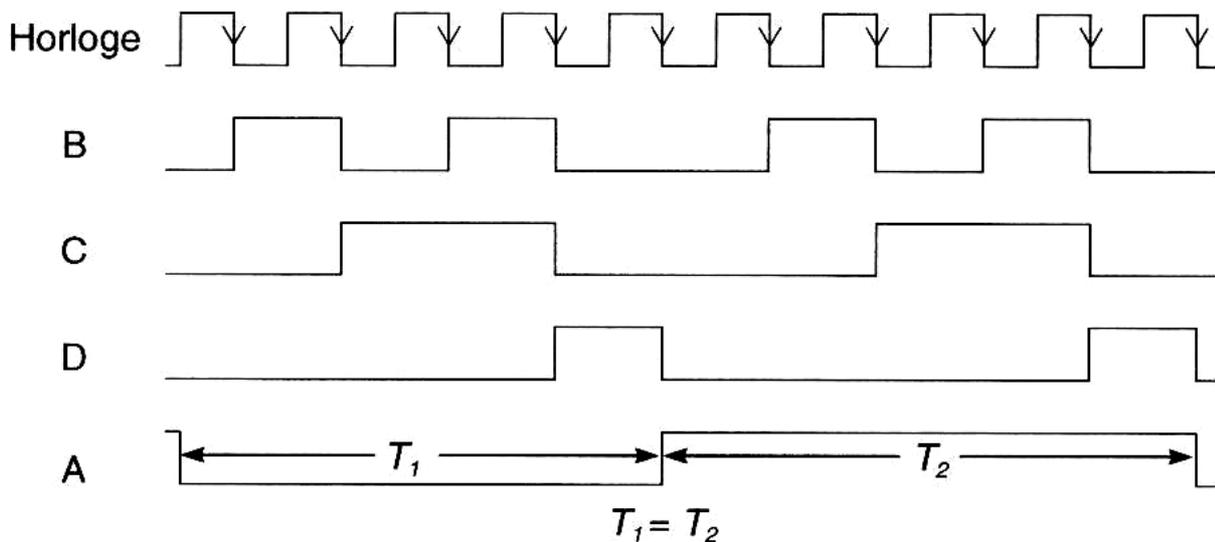


Figure 2.14 – Diviseur par dix symétrique

En branchant la sortie D à l'entrée de la bascule A, on inverse l'ordre du diviseur, qui effectue d'abord la division par cinq et ensuite la division par deux. Le contenu des données binaires devient $(ADCB)_2$ permettant ainsi au signal de la sortie A d'être symétrique.

b) Compteur MODULO-N :

On réalise, des compteurs MODULO asynchrone de manière identique à celle vue précédemment. Par contre, on fait la remise à zéro sans avoir recours à d'autres portes logiques externes en tirant profit de la porte NON-ET à l'intérieur du CI 7490.

Selon l'exemple du compteur MODULO-6 de la figure 2.15 on décode l'état intermédiaire $(0110)_2$ pour connecter ce signal à l'entrée de remise à zéro asynchrone (R0) du compteur.

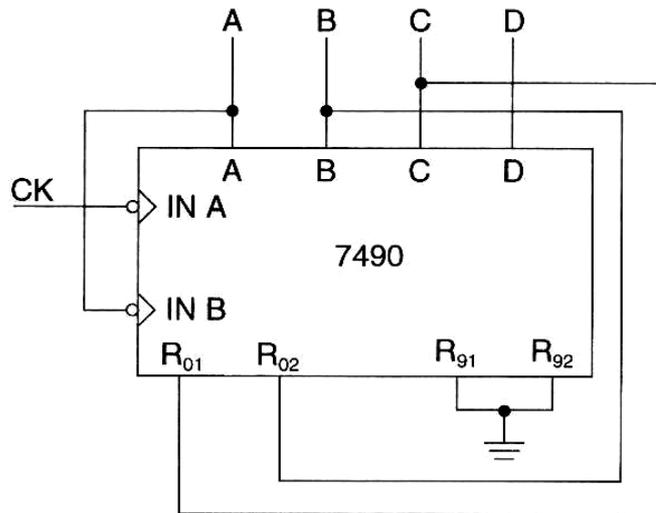


Figure 2.15 – compteur modulo-6

c) compteur DCB (modulo10) :

Pour réaliser un compteur DCB, il suffit simplement de coupler le diviseur par deux avec le diviseur par cinq (voir figure 2.16)

a) Table de vérité

'90A, 'L90, 'LS90
BCD COUNT SEQUENCE
(See Note A)

| COUNT | OUTPUT | | | |
|-------|----------------|----------------|----------------|----------------|
| | Q _D | Q _C | Q _B | Q _A |
| 0 | L | L | L | L |
| 1 | L | L | L | H |
| 2 | L | L | H | L |
| 3 | L | L | H | H |
| 4 | L | H | L | L |
| 5 | L | H | L | H |
| 6 | L | H | H | L |
| 7 | L | H | H | H |
| 8 | H | L | L | L |
| 9 | H | L | L | H |

b) Branchement

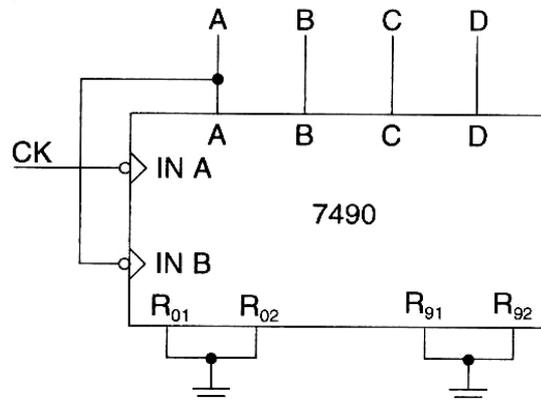


Figure 2.16 – compteur DCB (modulo10)

d) Compteur en cascade :

Si l'on analyse la séquence des nombres binaires d'un compteur DCB à deux chiffres (MODULO-100), on s'aperçoit que le chiffre des dizaines doit s'incrémenter au moment où les unités passent de 9 à 0. Si l'on utilise la sortie D des unités comme signal d'horloge pour le compteur des dizaines, le chiffre des dizaines s'incrémentera au bon moment en accord avec le signal d'horloge suivant. La figure 2.17 présente le branchement d'un compteur à deux chiffres.

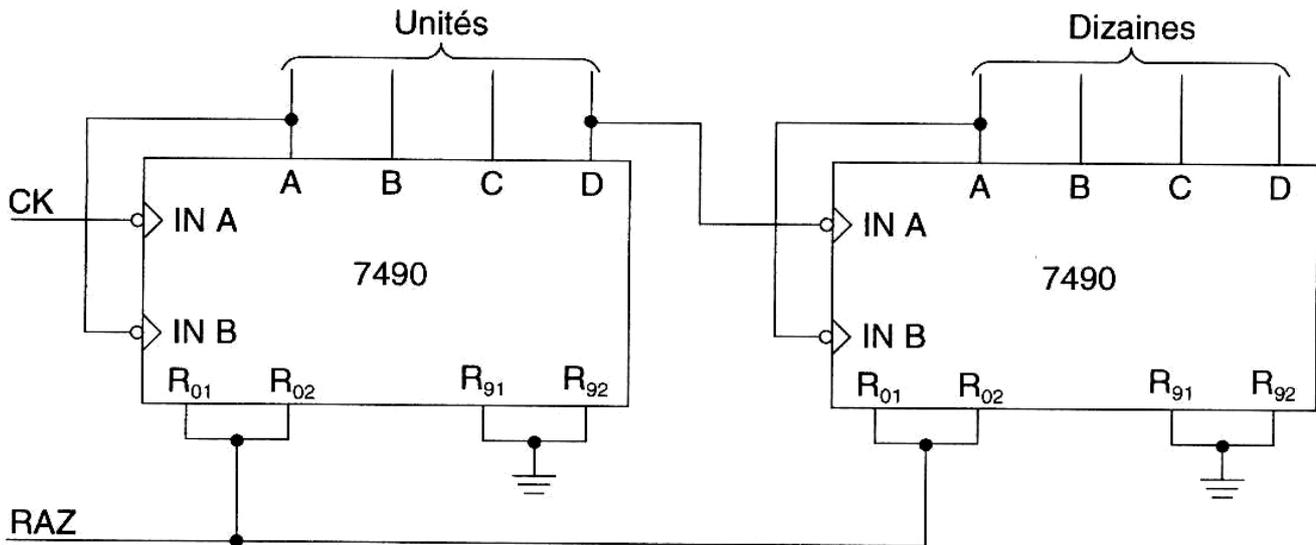


Figure 2.17 – compteur modulo-100

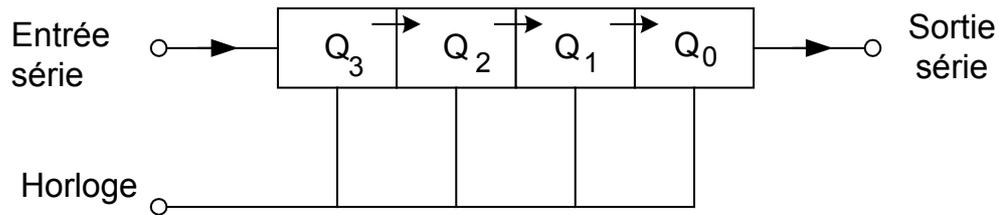
III. Les registres:

Un registre est un circuit permettant d'enregistrer provisoirement un « mot » binaire en vue de son transfert ultérieur dans un autre circuit (pour traitement, stockage, affichage, ...).

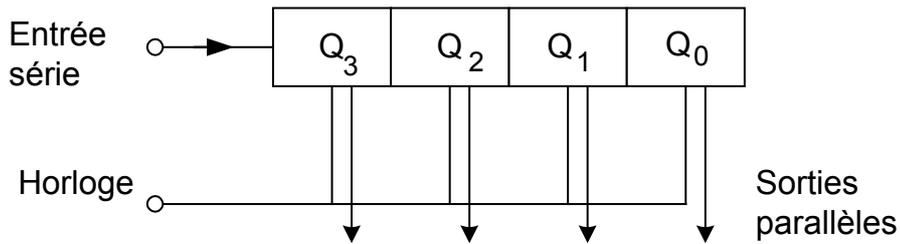
Un registre mémorise un mot binaire de n bits, il est constitué de n mémoires élémentaires qui sont les bascules (voir figure 3.1).

III.1 Types de registres :

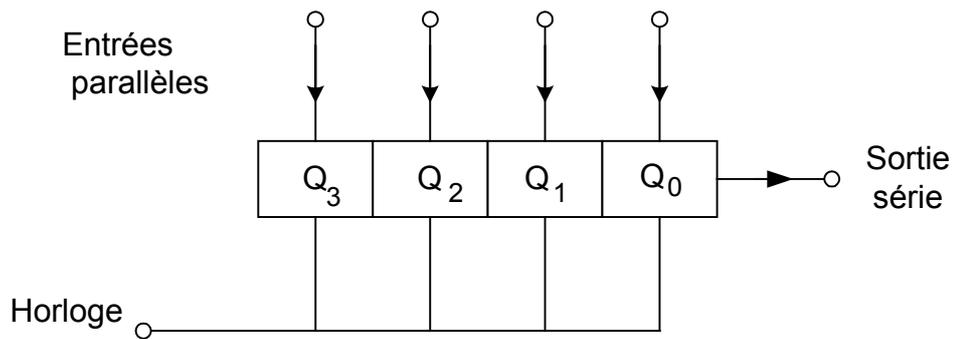
(Pour 4 bits) : (voir figure 3.1)



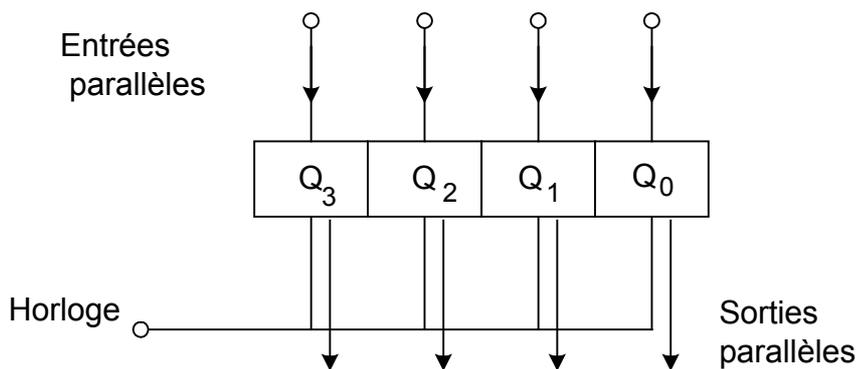
Ecriture série – lecture série (SISO)



Ecriture série – lecture parallèle (SIPO)



Ecriture parallèle – lecture série (PISO)



Ecriture parallèle – lecture parallèle (PIPO)

Figure 3.1 – Les types de registres

III.2 Registres à décalage :

Exemple de registre à décalage réalisé avec les bascules : SISO

Ce registre à décalage consiste en un assemblage de bascules commandées par une horloge commune. L'information est décalée en série d'une extrémité à l'autre du registre, de sorte que le transfert s'effectue sous la forme d'écriture série-lecture série. De ce fait, le registre à décalage comprend généralement une seule entrée de données et une seule sortie en plus de l'entrée d'horloge (voir figure 3.2)

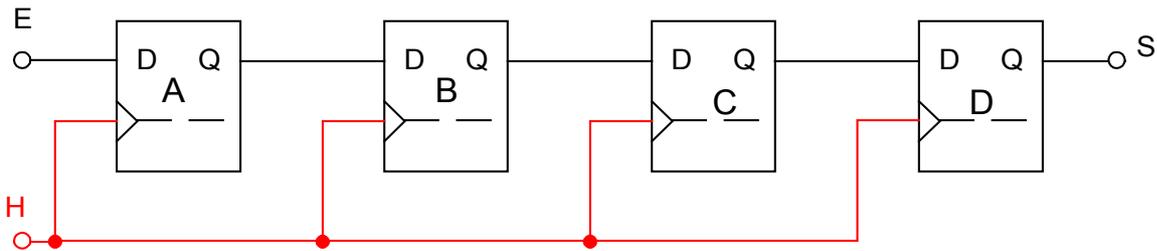


Figure 3.2 - Registre à décalage SISO à quatre bits réalisé avec quatre bascules D

Au commandement de l'horloge, la donnée introduite dans la première bascule se propage dans les bascules suivantes. Ainsi la bascule C acquiert l'état précédent de la bascule B, alors que la bascule B acquiert l'état de la bascule A, qui à son tour, acquiert l'état de l'entée au moment où l'impulsion d'horloge se produit. Le bit disponible à la sortie doit être traité sinon il est perdu. La figure 3.3 schématise le fonctionnement d'un registre à décalage à quatre bascules D quand on introduit les données $(0110)_2$ à l'entrée d'un registre contenant initialement l'information $(1011)_2$.

| Signal d'horloge | Données à l'entrée | Contenu du registre | Données à la sortie | | | | | | | |
|------------------|--------------------|---|---------------------|---|---|---|---|--|---------|---|
| 1 | 0 1 1 0 | <table border="1"><tr><td> </td><td>1</td><td>0</td><td>1</td><td>1</td><td> </td></tr></table> | | 1 | 0 | 1 | 1 | | | État initial |
| | 1 | 0 | 1 | 1 | | | | | | |
| 2 | 0 1 1 | <table border="1"><tr><td> </td><td>0</td><td>1</td><td>0</td><td>1</td><td> </td></tr></table> | | 0 | 1 | 0 | 1 | | 1 | |
| | 0 | 1 | 0 | 1 | | | | | | |
| 3 | 0 1 | <table border="1"><tr><td> </td><td>1</td><td>0</td><td>1</td><td>0</td><td> </td></tr></table> | | 1 | 0 | 1 | 0 | | 1 1 | |
| | 1 | 0 | 1 | 0 | | | | | | |
| 4 | 0 | <table border="1"><tr><td> </td><td>1</td><td>1</td><td>0</td><td>1</td><td> </td></tr></table> | | 1 | 1 | 0 | 1 | | 0 1 1 | |
| | 1 | 1 | 0 | 1 | | | | | | |
| 5 | | <table border="1"><tr><td> </td><td>0</td><td>1</td><td>1</td><td>0</td><td> </td></tr></table> | | 0 | 1 | 1 | 0 | | 1 0 1 1 | Transfert après la 4 ^e impulsion |
| | 0 | 1 | 1 | 0 | | | | | | |

Figure 3.3 – Décalage des données dans un Registre à décalage à quatre bits

A la lumière de cet exemple, on peut conclure la lenteur d'un registre à décalage série. Toutefois, comme toute l'information circule sur un seul fil, ce système réduit au minimum le nombre de lignes nécessaires à la transmission. Il devient donc très avantageux dans le cas des parcours à longue distance. Au contraire la transmission en parallèle nécessite autant de lignes que de bits à transmettre. On privilégie donc la rapidité des systèmes en parallèle surtout dans les applications à courte distance, comme le transfert d'informations d'un ordinateur à une imprimante.

- **Recirculation :**

Dans un registre à décalage, le bit présent à la sortie n'est plus mémorisé et cela devient parfois un inconvénient. Dans certaines situations on veut lire le contenu du registre tout en le préservant. Il devient possible de le faire à l'aide d'une contre-réaction en renvoyant la sortie d'un registre à décalage vers l'entrée. Cette option demande cependant l'ajout d'une entrée « MODE » pour commander soit la recirculation des données (mode = 1), soit l'enregistrement de nouvelles données en série (mode = 0). La figure 3.4 montre ces deux situations.

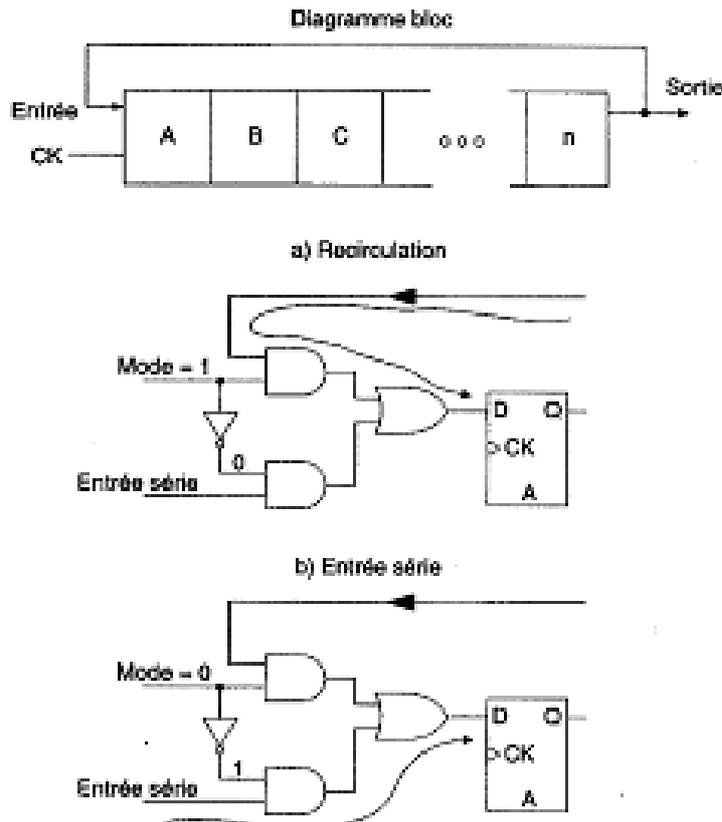


Figure 3.4 – Recirculation des données

- **Décalage à gauche- à droite :**

Le registre à décalage qu'on vient de voir réalise la fonction standard de décalage des données vers la droite. Toutefois, certaines applications, comme les opérations mathématiques, exigent un décalage à gauche ou à droite à l'intérieur du registre selon la commande d'un signal externe. A l'instar de la recirculation, il est possible de réaliser cette fonction par l'entremise d'une combinaison de portes logiques. Cette fois l'entrée de commande réagira de la manière suivante :

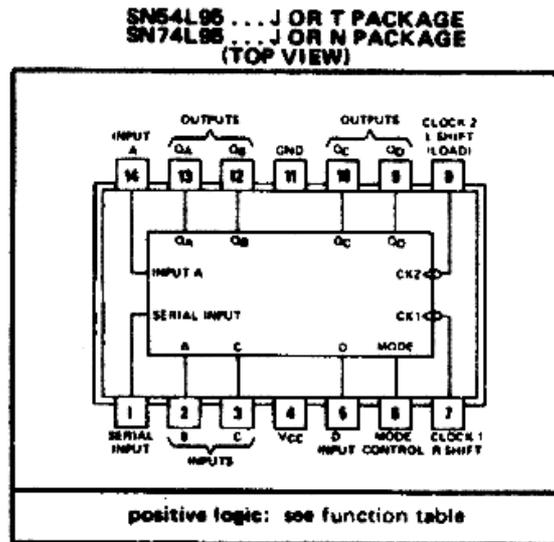
Mode = 0 → décalage à droite
 Mode = 1 → décalage à gauche

III.3 Registres intégrés :

Il existe, dans la famille TTL, deux registres à décalage de 4 bits qualifiés d'universels. Il s'agit des C I 7495 et 74194. Ils consistent en quatre bascules S-R dont les entrées sont sollicitées par des portes qui réalisent une fonction logique dépendante du raccordement externe. Il devient essentiel d'analyser la table de vérité de chacun de ces composants pour connaître leurs conditions de mise en marche en série ou en parallèle.

a) Circuit intégré 7495 :

La figure 3.5 montre d'abord le symbole et l'emplacement des bornes du composant. Vous y retrouverez aussi le diagramme fonctionnel en plus de la table de vérité qui dicte son comportement. Remarquez la présence de deux entées horloge. Elles sont reliées avec la borne « mode control » permettant de sélectionner un fonctionnement en série ou en parallèle.



FUNCTION TABLE

| MODE CONTROL | CLOCKS | | INPUTS | | | | OUTPUTS | | | | |
|--------------|--------|-------|--------|-----|-----|-----|---------|-----|------|------|------|
| | 2 (L) | 1 (R) | SERIAL | A | B | C | D | QA | QB | QC | QD |
| H | H | X | X | X | X | X | X | QA0 | QB0 | QC0 | QD0 |
| H | ↓ | X | X | a | b | c | d | a | b | c | d |
| H | ↓ | X | X | QB† | QC† | QD† | d | QBn | QCn | QDn | d |
| L | L | H | X | X | X | X | X | QA0 | QB0 | QC0 | QD0 |
| L | X | ↓ | H | X | X | X | X | H | QA n | QB n | QC n |
| L | X | ↓ | L | X | X | X | X | L | QA n | QB n | QC n |
| ↑ | L | L | X | X | X | X | X | QA0 | QB0 | QC0 | QD0 |
| ↓ | L | L | X | X | X | X | X | QA0 | QB0 | QC0 | QD0 |
| ↓ | L | H | X | X | X | X | X | QA0 | QB0 | QC0 | QD0 |
| ↑ | H | L | X | X | X | X | X | QA0 | QB0 | QC0 | QD0 |
| ↑ | H | H | X | X | X | X | X | QA0 | QB0 | QC0 | QD0 |

functional block diagram

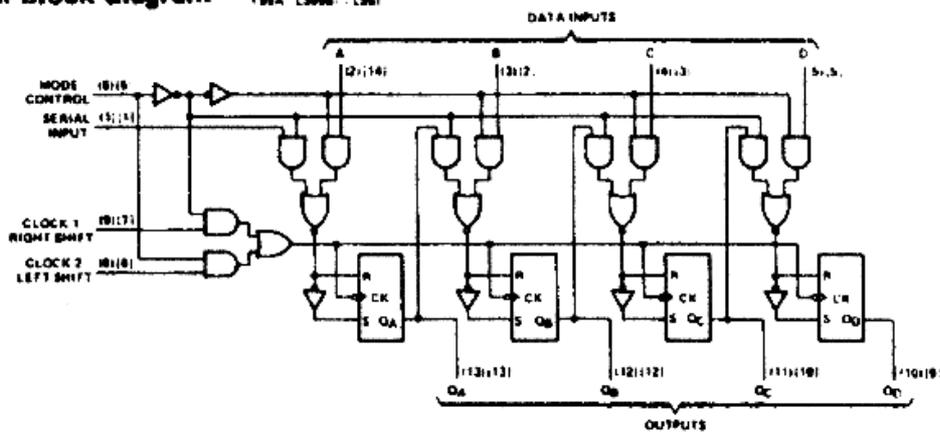


Figure 3.5 – C I 7495

Comme le transfert en parallèle est plus rapide que le transfert en série, le C I 7495 offre la possibilité d'utiliser une fréquence d'horloge plus élevée pour le mode parallèle (clock2) et une fréquence plus faible pour le mode série (clock1). Le signal d'horloge peut aussi être branché en commun avec les entrées pour osciller à la même fréquence indépendamment du mode de fonctionnement. Voici l'essentiel de l'analyse de la table de vérité.

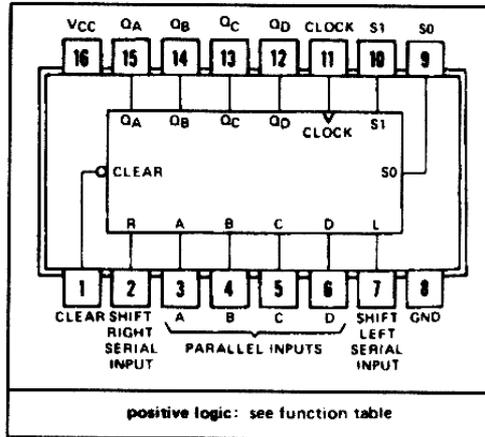
- Si l'entrée « mode control » = 0 :
 - L'entée série devient active, permettant à un bit d'être chargé ;
 - L'horloge 1 commande les bascules ;
 - Le registre réalise l'opération standard de décalage à droite ;
 - La sortie D fait office de sortie série.

- Si l'entrée « mode control » = 1 :
 - Les entées parallèles deviennent actives, permettant à un mot de 4 bits d'être chargé ;
 - L'horloge 2 commande les bascules ;
 - Le registre mémorise le mot et le rend disponible sur les sorties parallèles ;
 - Le décalage à gauche s'effectue seulement si l'on connecte Q_B à A, Q_C à B et Q_D à C ; l'entée série devient D.

b) Circuit intégré 74194 :

Le registre à décalage 74194 (voir figure 3.6) compte quelques améliorations. Il peut facilement fonctionner dans tous les modes de transfert. En outre, il possède deux entrées permettant la sélection du mode de fonctionnement (S0 et S1). Il possède aussi des bornes d'entrée-sortie distinctes pour le décalage à gauche (L) et à droite (R). Il renferme également une entrée asynchrone pour la remise à zéro (CLEAR) automatique, une lacune dans le cas du C.I.7495. On utilise fréquemment le C.I. 74194 pour effectuer la conversion des données de série à parallèle ou de parallèle à série.

SN54194, SN54LS194A, SN64S194 ... J OR W PACKAGE
SN74194, SN74LS194A, SN74S194 ... J OR N PACKAGE
(TOP VIEW)



positive logic: see function table

FUNCTION TABLE

| CLEAR | MODE | | CLOCK | INPUTS | | | | OUTPUTS | | | | | |
|-------|------|----|-------|--------|-------|----------|---|---------|---|-----------------|-----------------|-----------------|-----------------|
| | S1 | S0 | | SERIAL | | PARALLEL | | | | QA | QB | QC | QD |
| | | | | LEFT | RIGHT | A | B | C | D | | | | |
| L | X | X | X | X | X | X | X | X | X | L | L | L | L |
| H | X | X | L | X | X | X | X | X | X | QA0 | QB0 | QC0 | QD0 |
| H | H | H | ↑ | X | X | a | b | c | d | a | b | c | d |
| H | L | H | ↑ | X | H | X | X | X | X | H | QA _n | QB _n | QC _n |
| H | L | H | ↓ | X | L | X | X | X | X | L | QA _n | QB _n | QC _n |
| H | H | L | ↑ | H | X | X | X | X | X | QB _n | QC _n | QD _n | H |
| H | H | L | ↓ | L | X | X | X | X | X | QB _n | QC _n | QD _n | L |
| H | L | L | X | X | X | X | X | X | X | QA0 | QB0 | QC0 | QD0 |

functional block diagrams

194

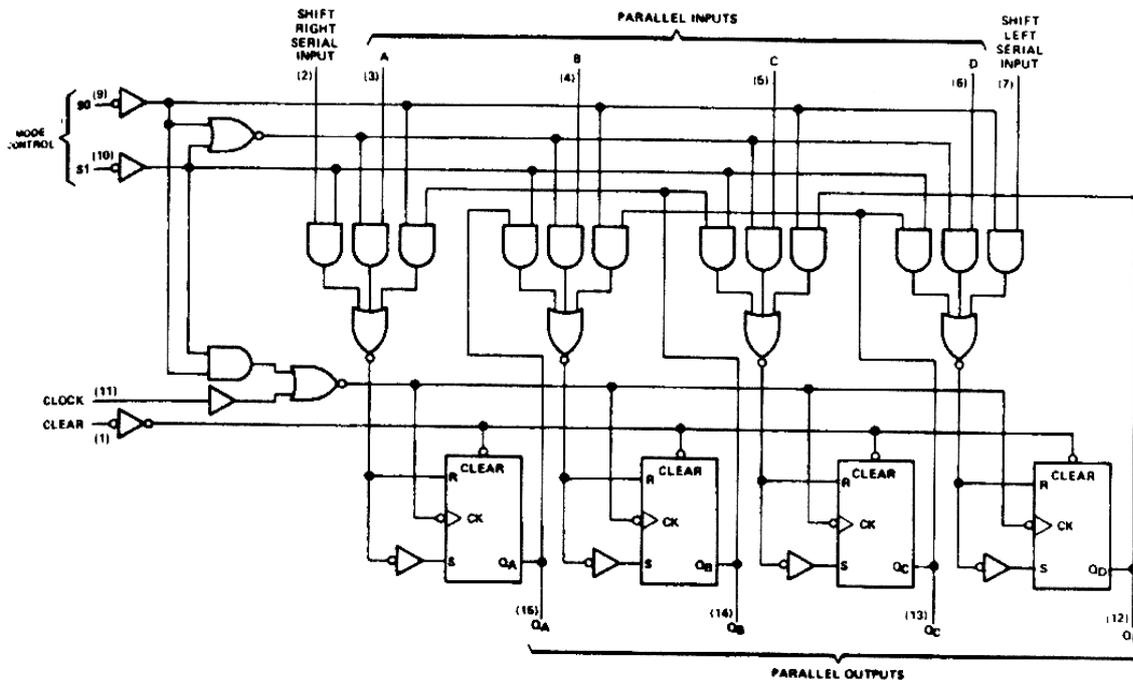


Figure 3.6 – C I 74194

On peut énumérer aussi d'autres circuits intégrés de la famille TTL comme :

- 7491 pour écriture série-lecture série ;
- 74164 pour écriture série-lecture parallèle;
- 74165 et 74166 pour écriture parallèle –lecture série;
- 74178 et 74195 pour écriture parallèle –lecture parallèle;

IV. Les Mémoires

IV.1 Introduction aux mémoires

Dans presque tous les systèmes numériques, les circuits disposent d'un système permettant la mémorisation de certaines informations. Ces différents systèmes sont appelés «mémoire». Ces derniers possèdent des aptitudes pour mémoriser de grandes quantités d'informations indispensables au bon fonctionnement des circuits numériques.

La figure 4.1 représente un système informatique utilisant deux dispositifs de «mémoire».

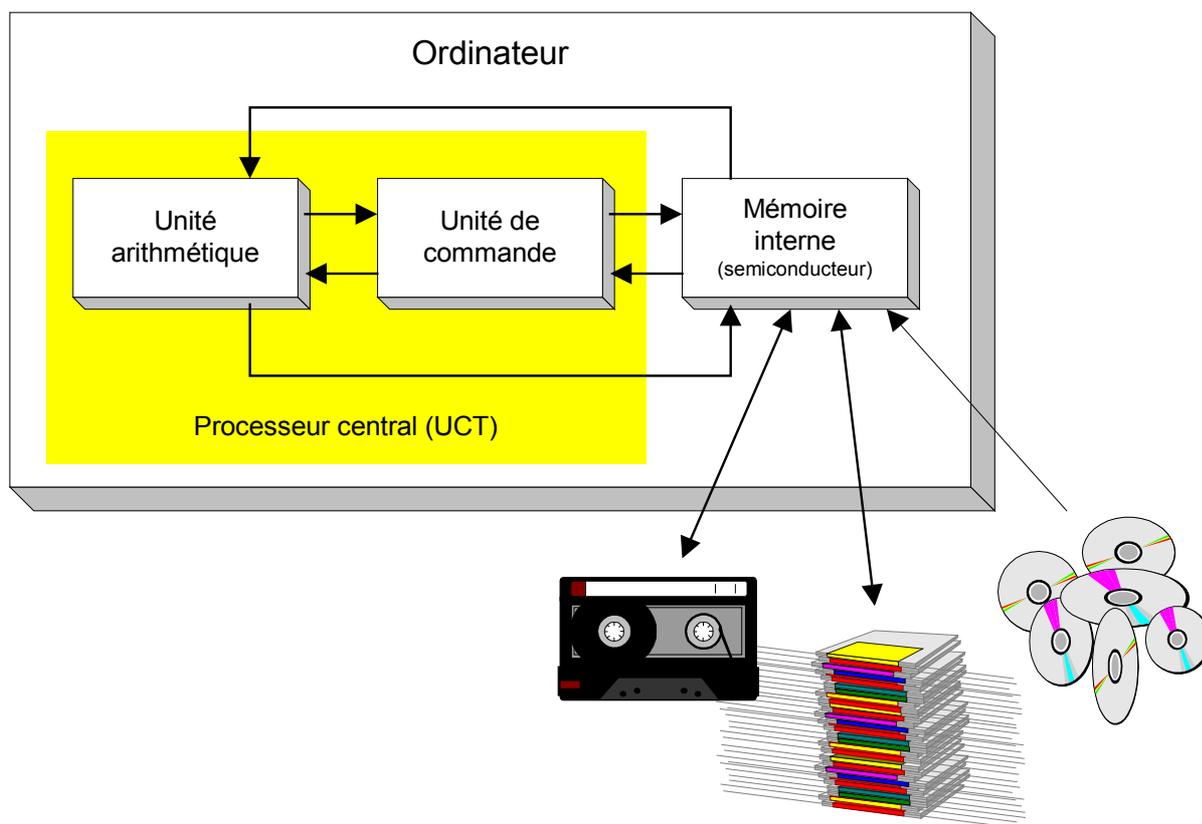


figure 4.1 système informatique

IV.2 Fonctionnement des Mémoires

IV.2.1 Principes de fonctionnement

Tous les types de mémoires, quels qu'ils soient et cela malgré leur fonctionnement interne différent fonctionnent selon de grands principes généraux. Chaque système mémoire est doté de plusieurs types de lignes d'entrée et de sortie qui ont les rôles suivants.

- 1- Choisir l'adresse en mémoire qui donne accès à l'emplacement touché par une opération de lecture ou d'écriture.
- 2- Sélectionner entre une opération de lecture ou d'écriture.
- 3- Acheminer l'information mémorisée durant une opération d'écriture.
- 4- Maintenir l'information tirée de la mémoire durant une opération de lecture.
- 5- Valider ou invalider la mémoire afin qu'elle réagisse ou non aux entrées d'adresse et à la commande lecture / écriture.

IV.2.2 Définitions

La mémoire (32 x 4) (voir figure 4.2) est un dispositif qui mémorise 32 mots de 4 bits. Etant donné que la longueur du mot est de 4 bits, il y a 4 lignes d'entrée de données $I_0 - I_3$ et 4 lignes de sortie de données $O_0 - O_3$. Pendant une opération d'écriture, la donnée que l'on place en mémoire est appliquée sur les lignes d'entrée de données. Durant une opération de lecture, le mot prélevé en mémoire apparaît sur les lignes de sortie de données.

a) Adressage (voir figure 4.2)

Il y a dans cette mémoire 32 emplacements de stockage allant de 00000_2 à 11111_2 (0 à 31 en décimal). Il est donc nécessaire d'utiliser 5 lignes d'adresse (A_0 à A_4) afin d'être en mesure de sélectionner l'adresse (ou l'endroit) désirée pour aller écrire ou lire l'information (la donnée). En général N lignes d'adresse sont nécessaires pour une mémoire de capacité 2^N mots.

b) Lecture / Écriture (L/E ou R/W) (voir figure 4.2)

Cette ligne (R/W Read or Write) permet de sélectionner l'opération désirée, c'est-à-dire une lecture ou une écriture. Lorsqu'on utilise une seule entrée L/\bar{E} il est courant que $L/\bar{E} = 1$ désigne une opération de lecture et que $L/\bar{E} = 0$ désigne une opération d'écriture.

c) Validation de la mémoire (voir figure 4.2)

Les systèmes mémoires disposent d'un mécanisme d'invalidation, quand la mémoire est invalidée, cette dernière ne réagit plus aux autres différentes actions d'entrées.

C'est le rôle de l'entrée *VALIDATION MEMOIRE* sur la figure 4.2. Quand cette entrée est à son niveau vrai haut, la mémoire est validée et fonctionne correctement. Au contraire, quand cette entrée est basse, la mémoire ne répond plus aux codes d'adresse et à la valeur sur L/\bar{E} .

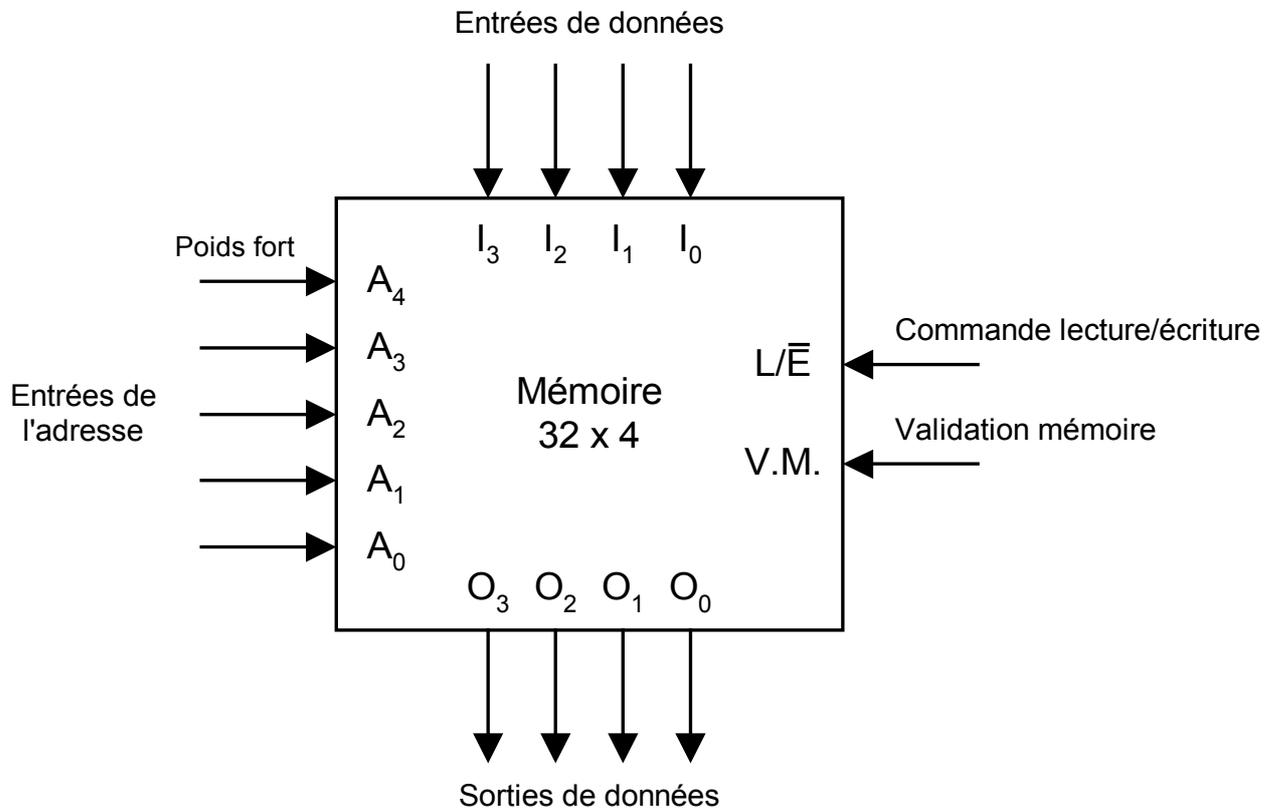


figure 4.2 schéma d'une mémoire 32 x 4

IV.2.3 Connexions Mémoires

La capacité mémoire dans les ordinateurs s'avère d'une grande importance au niveau des performances et sur les possibilités d'emmagasinage. La mémoire interne d'un ordinateur est constituée d'une mémoire **VIVE** et d'une mémoire **MORTE** qui sont directement reliées à l'Unité Centrale de Traitement par l'intermédiaire de BUS. La figure 4.3 représente le schéma interne d'un tel système. On peut voir un bus d'adresse, un bus de donnée et un bus de commande.

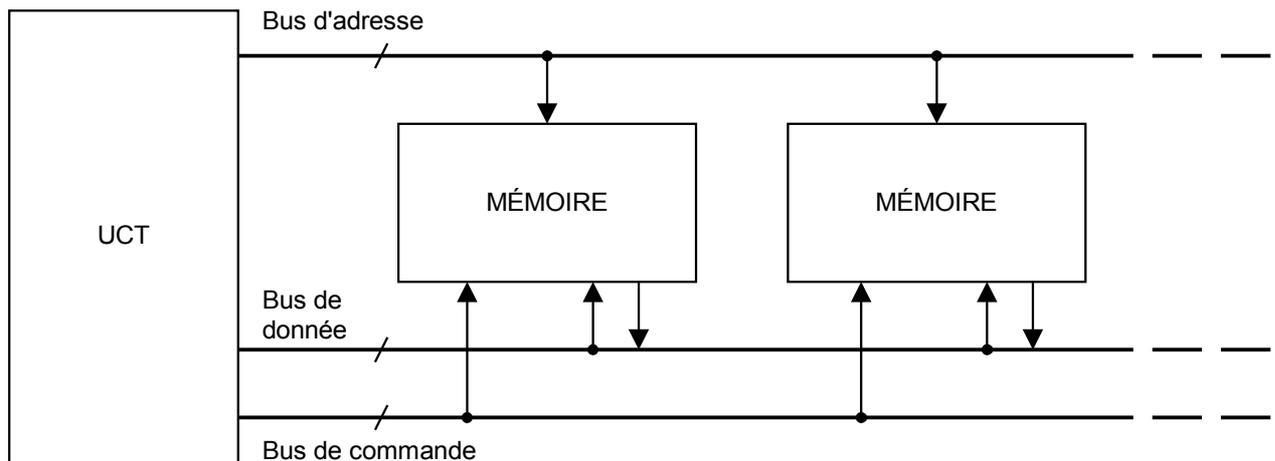


figure 4.3 Les bus réunissant les puces de la mémoire interne à l'UCT

IV.3 Mémoire Morte :

Dispositif de mémoire à semi-conducteurs qui gardent en permanence des données ou dont les données sont rarement modifiées. Les données dans une mémoire morte sont programmées généralement une seule fois. L'information s'y trouvant ne peut **seulement qu'être lue**. (voir figure 4.4)

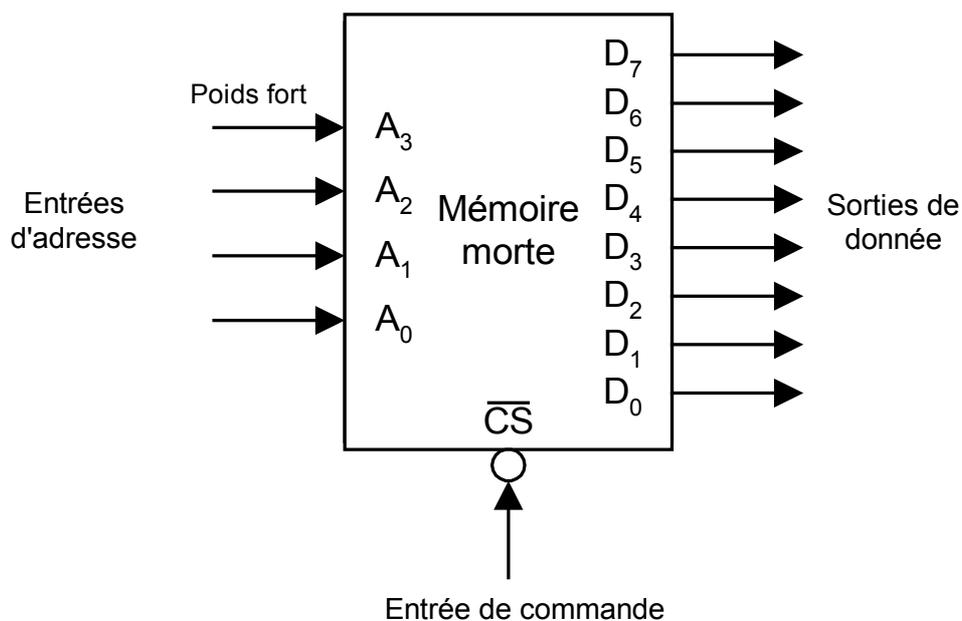


figure 4.4 Schéma fonctionnel typique d'une mémoire morte

IV.3.1 Types de Mémoire Morte

Il existe plusieurs types de mémoire morte permettant ainsi des méthodes de programmation différentes. Le terme ROM (Read Only Memory) permet de reconnaître la mémoire morte.

- 1) **MROM (Mask Read Only Memory)**: Utilisation d'un négatif (**masque**) d'une épreuve photographique pour établir les différentes connexions électriques de la mémoire. Un des grands inconvénients de ce type de mémoires est l'impossibilité de la reprogrammer s'il faut apporter des changements au programme mémorisé. On utilise ce type de mémoires surtout lorsqu'on a besoin d'un grand volume de mémoires mortes toutes programmées de la même façon.
- 2) **PROM (Programmable Read Only Memory)**: La structure interne de PROM est basée sur les liaisons fusibles. Lorsqu'il s'agit d'applications de volume plus modeste l'utilisateur peut programmer cette mémoire selon ses besoins. Il est important de noter que l'utilisateur ne peut programmer cette dernière **qu'une seule fois** c'est à dire une fois programmée elle ne diffère en rien d'une MROM.
- 3) **EPROM (Erasable Programmable Read Only Memory)**: Tout comme PROM, une EPROM peut être programmée selon les besoins de son utilisateur. Cependant, il est possible de la reprogrammer **plusieurs fois**. Les cellules de stockage d'une EPROM sont des transistors à effet de champ utilisant une grille de silicium. Il est alors possible de ramener la grille à son état initial en l'exposant à une lumière ultraviolet pendant 10 à 20 minutes
- 4) **EEPROM (Electrical Erasable Programmable Read Only Memory)**: Tout comme PROM et EPROM, une EEPROM peut être programmée par l'utilisateur pour ainsi répondre à ses besoins. L'avantage principal d'utiliser une EEPROM provient de sa faculté d'être **effaçable électriquement** (le temps nécessaire pour cette opération étant environ de 10 msec) avec la possibilité de reprogrammation de mots individuels de la mémoire matrice. Par contre pour une EPROM l'opération d'effacement touche tout le contenu de la mémoire et nécessite de les retirer de leur fiche.

IV.3.2 Exemples de mémoires mortes :

1) Exemple de mémoire morte EPROM :

La mémoire morte EPROM 4k x 8 (2732) (voir figure 4.5) présente 12 entrées d'adresse, 8 bornes de sortie et deux entrées de contrôle. \overline{CE} est l'entrée de validation de la puce que l'on utilise aussi pour mettre la puce en mode de réduction de consommation (attente). L'autre entrée de contrôle est \overline{OE}/V_{pp} dont le rôle varie selon le mode opératoire de l'élément. V_{pp} est une tension de programmation spéciale que l'on applique quand on programme la puce. Cette mémoire a plusieurs modes opératoires qui sont activés selon la combinaison des tensions sur \overline{CE} et sur \overline{OE}/V_{pp} (voir figure 4.6).

Le processus de programmation peut être effectué manuellement, ce qui prend quelques heures pour une puce 2732. Par contre, il existe dans le commerce des programmeurs d'EPROM qui permettent de programmer et de vérifier la totalité d'un 2732 en moins de deux minutes, après que l'utilisateur a entré au clavier ses données dans la mémoire VIVE du programmeur. On peut remplir la mémoire VIVE de certains programmeurs en transférant directement les données se trouvant dans un ordinateur. Cette procédure, appelée chargement, permet à l'utilisateur de développer d'abord son programme à l'ordinateur, ensuite, quand tout est parfait, de transférer les données de la mémoire de l'ordinateur dans la mémoire du programmeur. L'EPROM peut ensuite être configurée dans le programmeur.

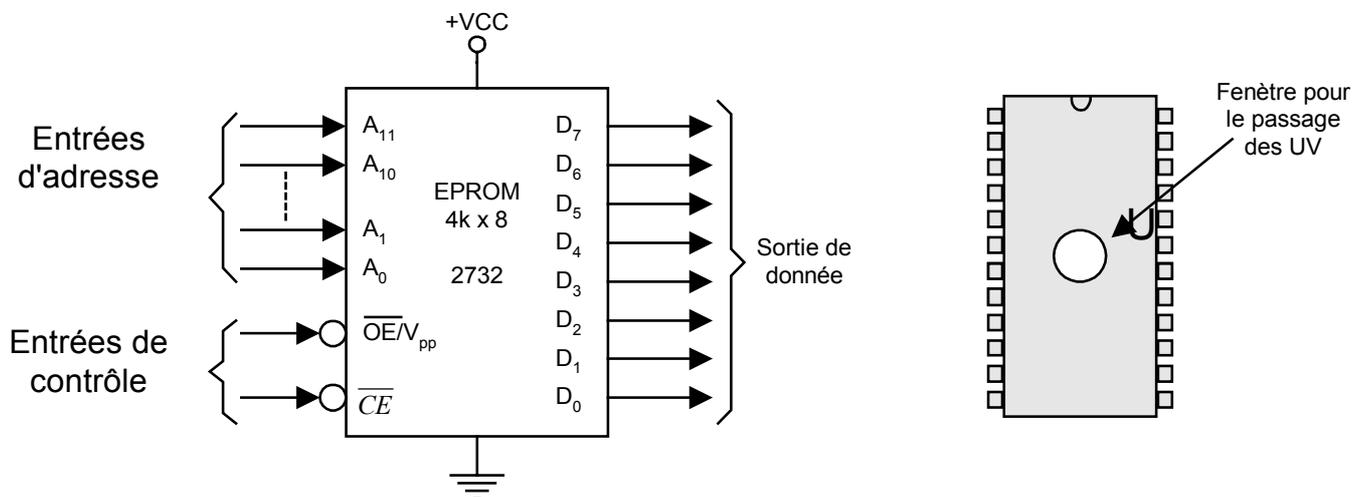


Figure 4.5 : Mémoire morte EPROM 4k x 8 (2732)

| Mode | Entrées | | Sorties |
|----------------------|-----------------|------------------------|--------------------------|
| | \overline{CE} | \overline{OE}/V_{PP} | |
| Lecture vérification | V_{IL} | V_{IL} | Donnée _{SORTIE} |
| Invalidation sortie | V_{IL} | V_{IH} | Haute-Z |
| Attente | V_{IH} | X | Haute-Z |
| Programmation | V_{IL} | V_{PP} | Donnée _{ENTRÉE} |

note: V_{IL} = bas TTL
 V_{IH} = haut TTL
 X = indifférent
 V_{pp} = nominal 21 V

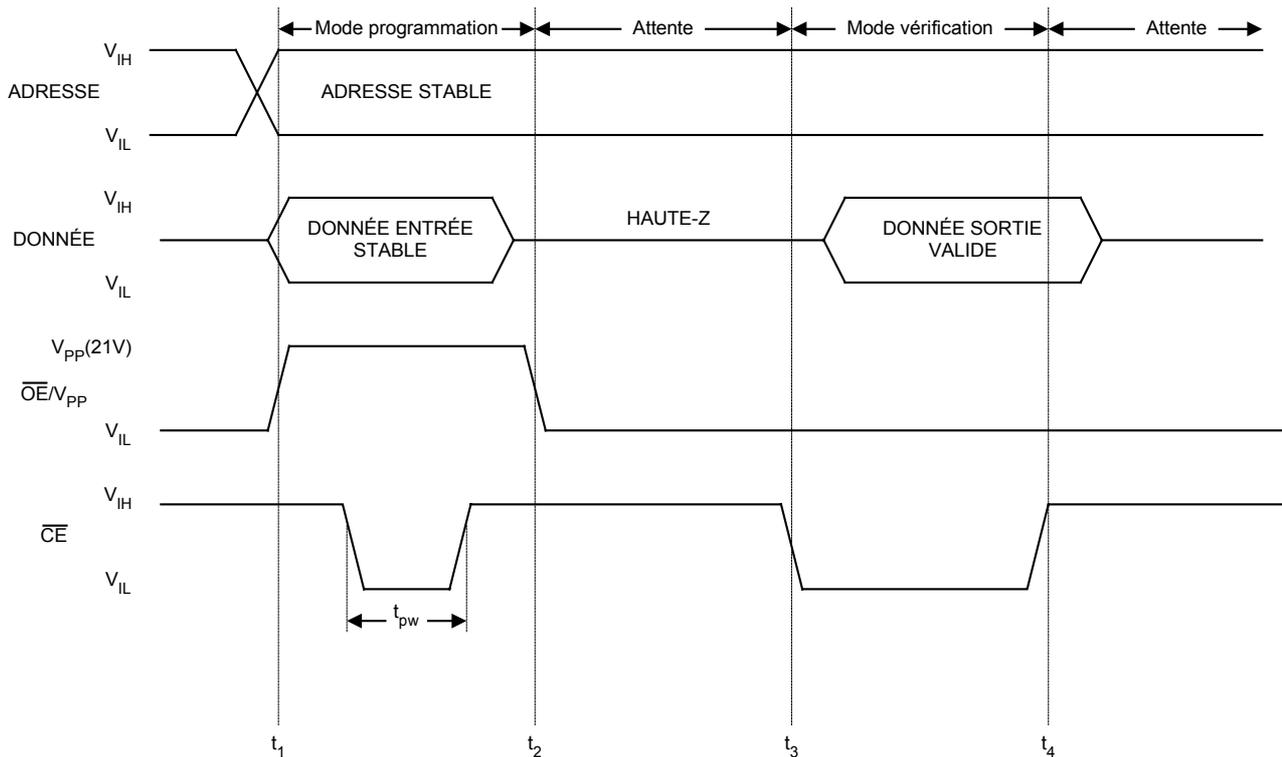


Figure 4.6 : Chronogramme des modes programmation et vérification de l'EPROM (2732)

2) Exemple de mémoire morte EEPROM :

La figure 4.7 présente l'EEPROM 2864 (8 k x 8) qui renferme dans son boîtier tous les circuits nécessaires pour générer les fortes tensions qui rendent possible l'effacement et la reprogrammation. Sur cette figure on peut voir le tableau des différents modes opératoires de cette mémoire.

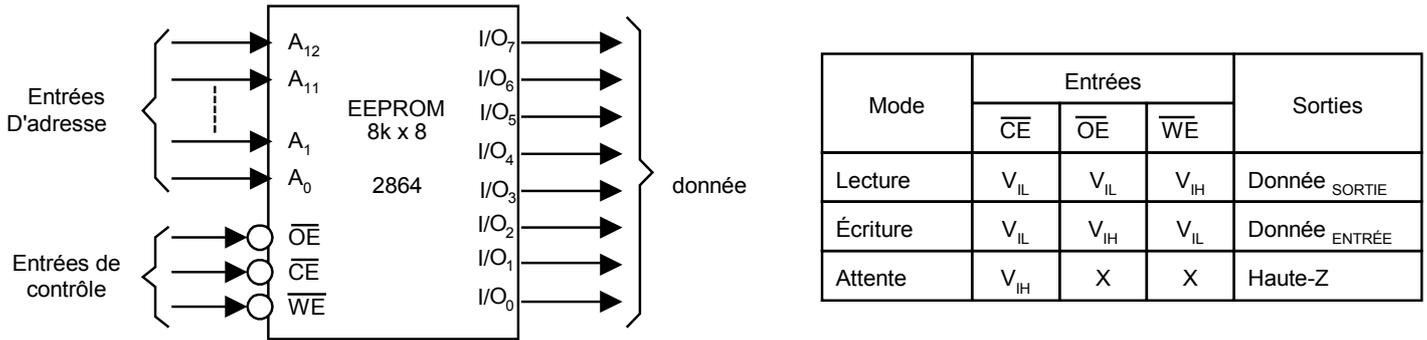


Figure 4.7: Mémoire EEPROM 8 k x 8 (2864)

IV.4 Mémoires vives

Un dispositif à mémoire vive permet la lecture et l'écriture des données en mémoire. Ces types de mémoire doivent disposer d'un temps de cycle très court afin de ne pas ralentir l'UCT. Rappelons également que le principal inconvénient est la volatilité, c'est-à-dire que les informations sont perdues si la tension est interrompue.

IV.4.1 Mémoire VIVE Statique (SRAM).

Les cellules de mémoire vive statique sont des bascules qui demeurent dans un état donné pour le stockage d'un bit, tant que le circuit demeure alimenté.

Exemple de mémoire vive SRAM :

La figure 4.8 représente le symbole logique de la SRAM CMOS 6264. Les quatre conducteurs de commande fixent le mode opératoire du dispositif.

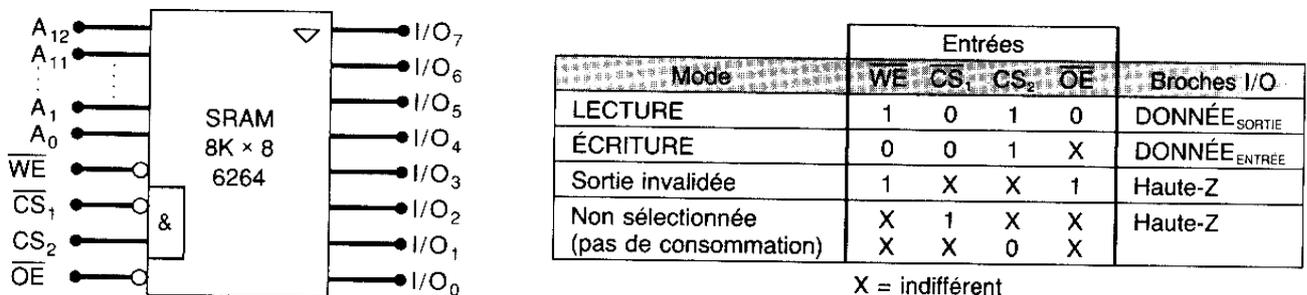


Figure 4.8: SRAM CMOS 6264.

IV.4.2 Mémoire VIVE Dynamique (DRAM)

Les cellules de mémoire vives dynamiques emmagasinent les données (0 ou 1) sous forme de charges dans des condensateurs, d'où la nécessité de rafraîchir périodiquement les mémoires dynamiques. Ces mémoires sont fabriquées en utilisant une technologie MOS et se distinguent par leur grande capacité, leur vitesse et leur faible consommation.

La plus part des mémoires vives dynamiques peuvent être considérées comme une matrice de registres à 1 bit (cellule). Dans la figure 4.9 il y a une matrice de 128 x 128 cellules. Chaque cellule est repérée par une rangée particulière et une colonne particulière de la matrice. 14 entrées d'adresse sont nécessaires pour choisir une des cellules : les bits d'adresse de rangs inférieurs A_0 - A_6 choisissent la rangée et les bits d'adresse des rangs supérieurs choisissent la colonne.

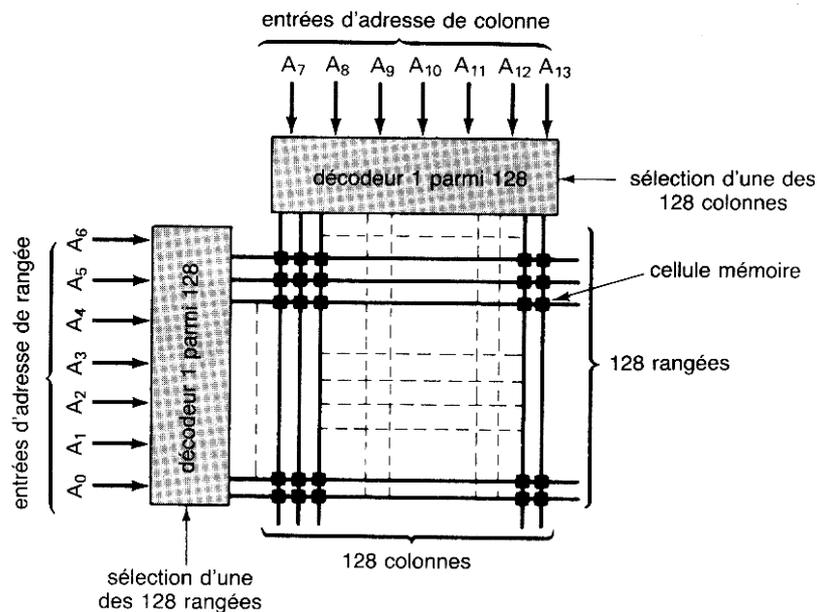


Figure 4.9: organisation de cellule dans une mémoire vive 16 k X 1

Multiplexage d'adresse :

Les fabricants de puces DRAM de grande capacité ont trouvé une façon de réduire le nombre de bornes. Ils ont implanté ce qu'on appelle multiplexage d'adresse.

La figure 4.10 représente une UCT pilotant de la mémoire VIVE dynamique de capacité 16 k mots par l'entremise d'un multiplexeur choisissant les différentes lignes d'adresse.

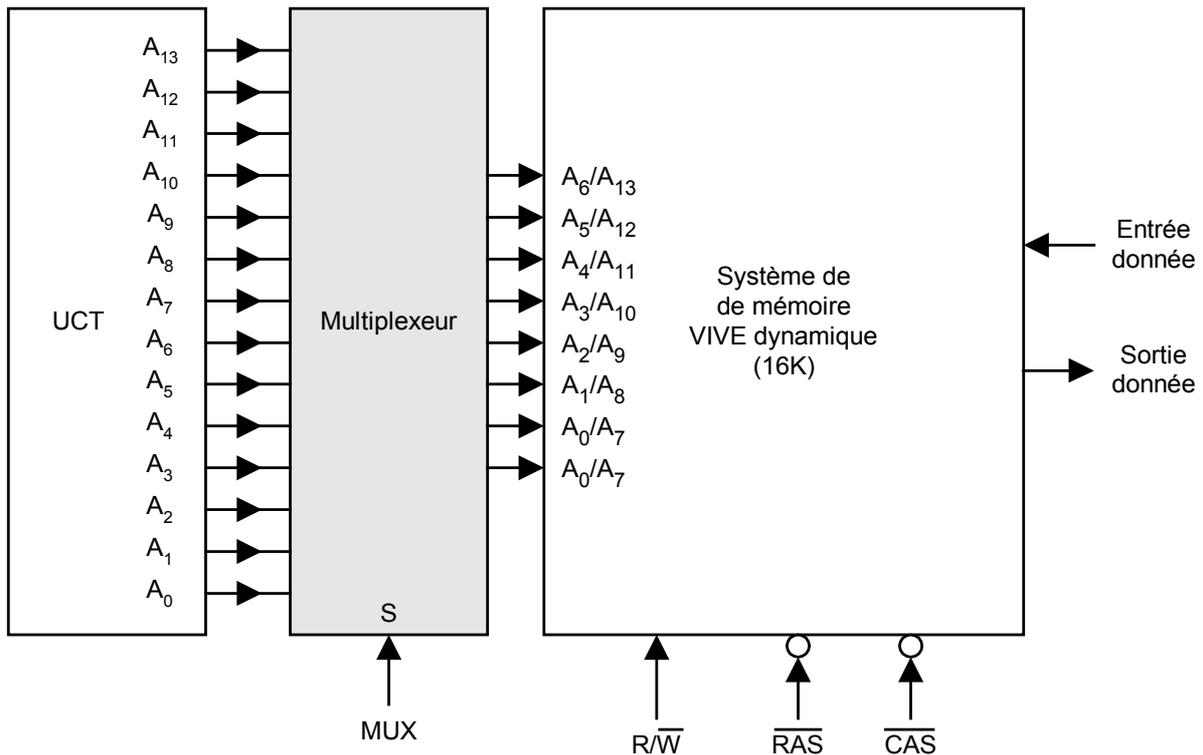


Figure 4.10 : adresses de l'UCT pilotant une mémoire vive dynamique par l'entremise d'un multiplexeur.

IV.5 Extension de la longueur du mot et de la capacité:

IV.5.1 Extension de la longueur du mot.

Supposons que nous ayons besoin d'une mémoire qui peut emmagasiner 16 mots de 8 bits et que nous ayons à notre disposition des puces de mémoire VIVE 16 x 4 ayant des lignes E/S communes. Il est possible d'assembler deux de ces puces 160 x 4 pour obtenir la mémoire recherchée. On réussit cet assemblage en configurant les systèmes comme sur la figure 4.11

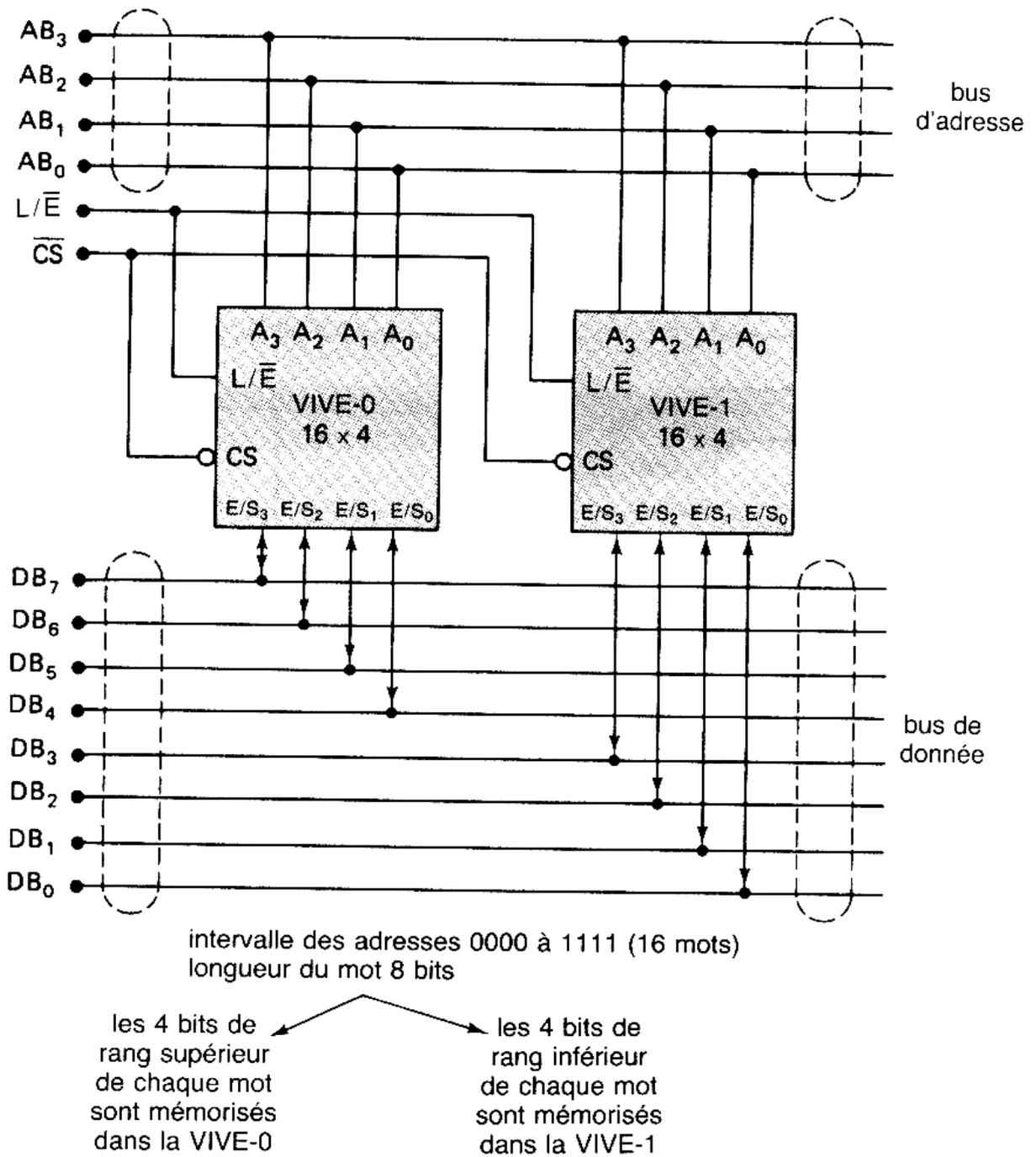


Figure 4.11 : Assemblage de deux mémoires VIVE 16 x 4 pour constituer mémoire 16 x 8

IV.5.2 Extension de la capacité.

Supposons que l'on ait besoin d'une mémoire de 32 mots de 4 bits et que l'on ne dispose que de puces de 16 x 4. Il est possible en assemblant deux puces 16 x 4 (voir figure 4.12) de construire la mémoire recherchée.

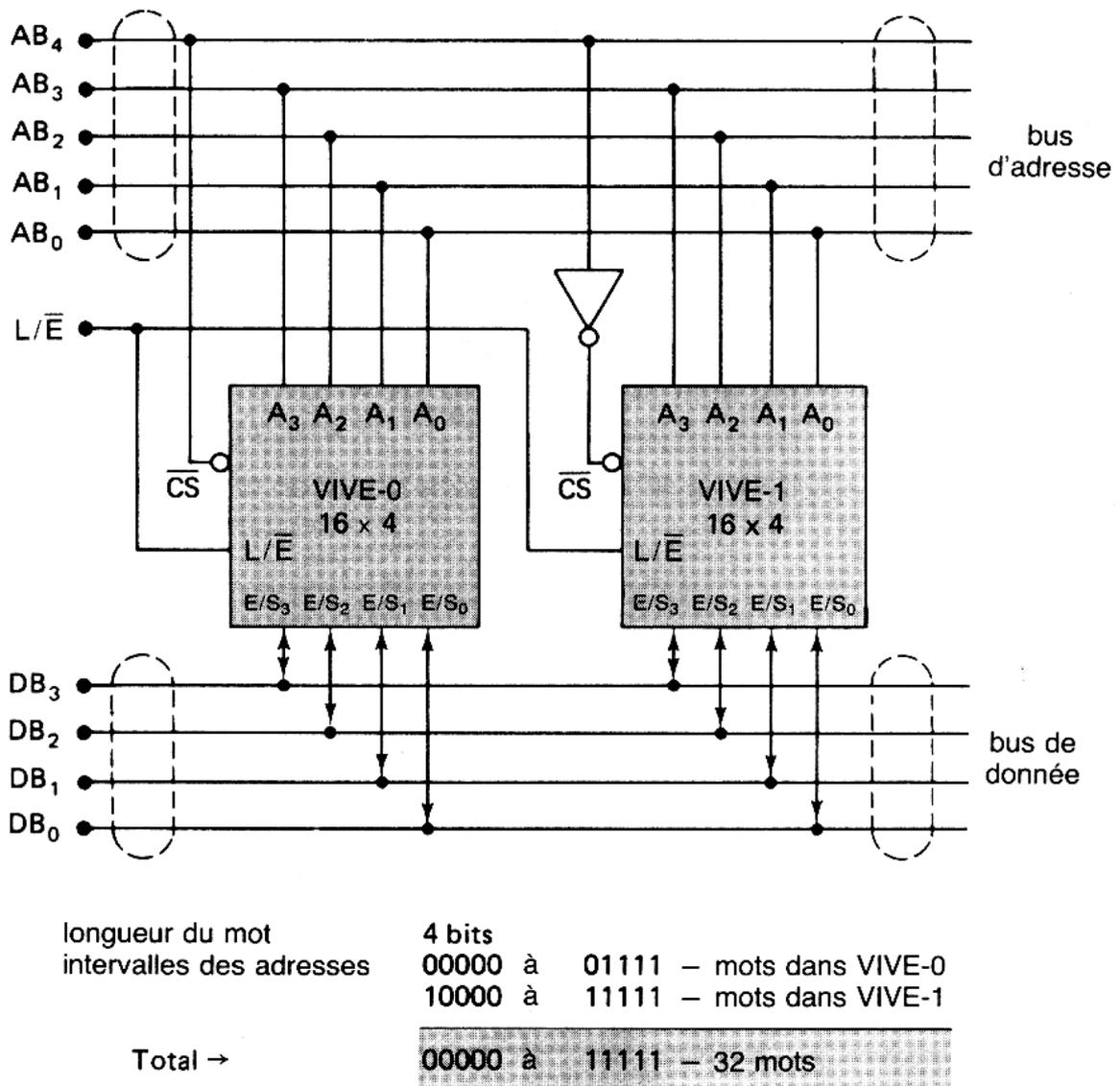


Figure 4.12 : Assemblage de deux mémoires VIVE 16 x 4 pour constituer mémoire 32 x 4

La configuration de la mémoire de la figure 4.13 est similaire à celle de la mémoire 32 x 4 de la figure 4.12. Toutefois, on utilise un circuit de décodage pour produire les signaux d'entrée \overline{CS} .

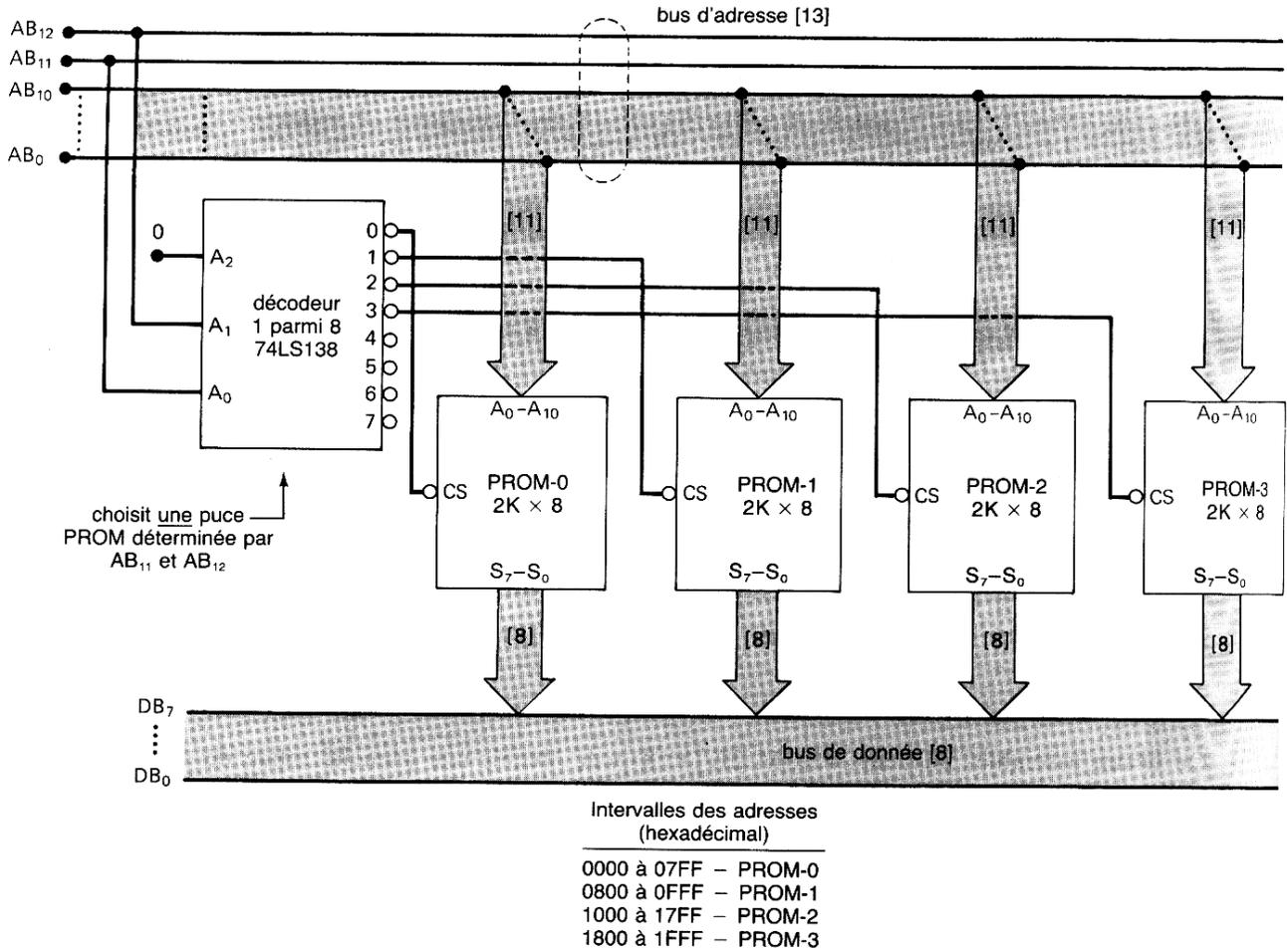


Figure 4.13 : PROM 8k x 8 formée au moyen de 4 puces 2k x 8

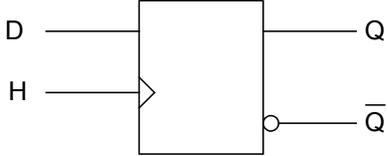
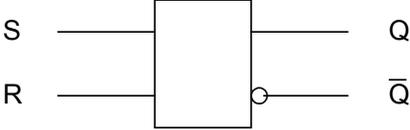
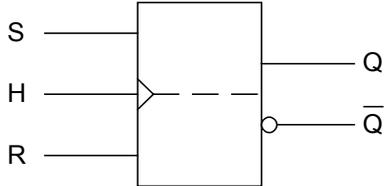
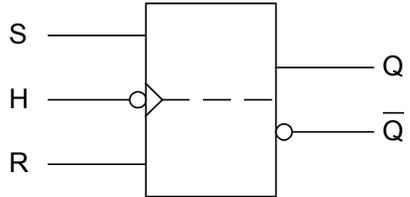
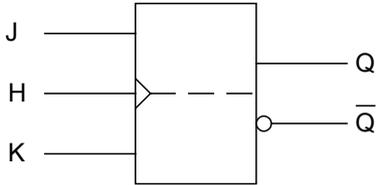
**MODULE N° 11: MAINTENANCE D'UN SYSTEME
SEQUENTIEL**

GUIDE DE TRAVAUX PRATIQUES

Exercices :

Exercices 1 :

Compléter le tableau suivant

| Symbole | Désignation |
|--|-------------|
|  <p>D ———> [] ———> Q H ———> [] ———> Q̄</p> | |
|  <p>S ———> [] ———> Q R ———> [] ———> Q̄</p> | |
|  <p>S ———> [] ———> Q H ———> [] ———> Q̄ R ———> [] ———> Q̄</p> | |
|  <p>S ———> [] ———> Q H ———> [] ———> Q̄ R ———> [] ———> Q̄</p> | |
|  <p>J ———> [] ———> Q H ———> [] ———> Q̄ K ———> [] ———> Q̄</p> | |

Exercice2 :

Reconnaître les composants logiques du tableau suivant d'après leur table de fonctionnement

| Table de fonctionnement | | | | | Composant logique correspondant | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--|---|-----------|-----------------|-----------------|---------------------------------|--|---------|---------|---|---|-----------|-----------------|---|-----------------|---|---|---|----------|-------------|---|---|---|---|---|---|---|-----------|---|---|---|---|---|-------------|-------|--|--|
| <table border="1"> <thead> <tr> <th colspan="2">Entrées</th> <th colspan="2">Sorties</th> </tr> <tr> <th>S</th> <th>R</th> <th>Q</th> <th>\bar{Q}</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td colspan="2">Inchangé</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td colspan="2">Ambiguïté</td> </tr> </tbody> </table> | | | | | Entrées | | Sorties | | S | R | Q | \bar{Q} | 0 | 0 | Inchangé | | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | Ambiguïté | | | | | | | | | |
| Entrées | | Sorties | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| S | R | Q | \bar{Q} | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | Inchangé | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 0 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | 1 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | Ambiguïté | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <table border="1"> <thead> <tr> <th colspan="3">Entrées</th> <th colspan="2">Sorties</th> </tr> <tr> <th>H</th> <th>S</th> <th>R</th> <th>Q</th> <th>\bar{Q}</th> </tr> </thead> <tbody> <tr> <td></td> <td>0</td> <td>0</td> <td colspan="2">Inchangé</td> </tr> <tr> <td></td> <td>0</td> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td></td> <td>1</td> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td></td> <td>1</td> <td>1</td> <td colspan="2">Ambiguïté</td> </tr> </tbody> </table> | | | | | Entrées | | | Sorties | | H | S | R | Q | \bar{Q} |  | 0 | 0 | Inchangé | |  | 0 | 1 | 0 | 1 |  | 1 | 0 | 1 | 0 |  | 1 | 1 | Ambiguïté | | | |
| Entrées | | | Sorties | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| H | S | R | Q | \bar{Q} | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  | 0 | 0 | Inchangé | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  | 0 | 1 | 0 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  | 1 | 0 | 1 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  | 1 | 1 | Ambiguïté | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <table border="1"> <thead> <tr> <th colspan="3">Entrées</th> <th colspan="2">Sorties</th> </tr> <tr> <th>H</th> <th>J</th> <th>K</th> <th>Q_{n+1}</th> <th>\bar{Q}_{n+1}</th> </tr> </thead> <tbody> <tr> <td></td> <td>0</td> <td>0</td> <td>Q_n</td> <td>\bar{Q}_n</td> </tr> <tr> <td></td> <td>0</td> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td></td> <td>1</td> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td></td> <td>1</td> <td>1</td> <td>\bar{Q}_n</td> <td>Q_n</td> </tr> </tbody> </table> | | | | | Entrées | | | Sorties | | H | J | K | Q_{n+1} | \bar{Q}_{n+1} |  | 0 | 0 | Q_n | \bar{Q}_n |  | 0 | 1 | 0 | 1 |  | 1 | 0 | 1 | 0 |  | 1 | 1 | \bar{Q}_n | Q_n | | |
| Entrées | | | Sorties | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| H | J | K | Q_{n+1} | \bar{Q}_{n+1} | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  | 0 | 0 | Q_n | \bar{Q}_n | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  | 0 | 1 | 0 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  | 1 | 0 | 1 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  | 1 | 1 | \bar{Q}_n | Q_n | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <table border="1"> <thead> <tr> <th colspan="2">Entrées</th> <th colspan="2">Entrées</th> </tr> <tr> <th>H</th> <th>D</th> <th>Q_{n+1}</th> <th>\bar{Q}_{n+1}</th> </tr> </thead> <tbody> <tr> <td></td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td></td> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table> | | | | | Entrées | | Entrées | | H | D | Q_{n+1} | \bar{Q}_{n+1} |  | 0 | 0 | 1 |  | 1 | 1 | 0 | | | | | | | | | | | | | | | | |
| Entrées | | Entrées | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| H | D | Q_{n+1} | \bar{Q}_{n+1} | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  | 0 | 0 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  | 1 | 1 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Exercice 3:

- 1) La capacité d'une mémoire est de $8K \times 16$.
 - a) Quelle est la longueur en bits de chacun des mots ?
 - b) Quel est le nombre de cellules (ou bits) que dispose cette mémoire ?
 - c) Combien peut-on y emmagasiner d'informations (en mots) ?
- 2) Déterminez la différence entre l'opération de lecture et celle d'écriture.
- 3) Quelles sont les différentes étapes pour aller lire l'information située à l'adresse 0101? Et quelle est cette information ?

Mémoire 16X4:

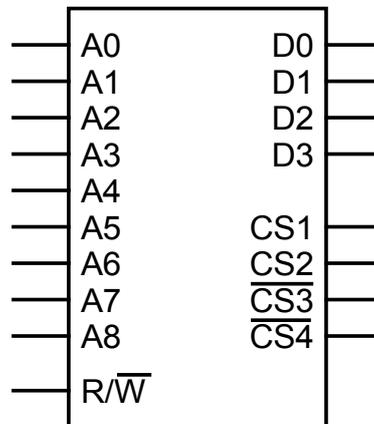
| Adresses | Informations |
|-------------|--------------|
| 0000 | 0000 |
| 0001 | 1000 |
| 0010 | 1111 |
| 0011 | 1111 |
| 0100 | 1111 |
| 0101 | 0001 |
| 0110 | 1101 |
| 0111 | 1100 |
| 1000 | 1001 |
| 1001 | 0011 |
| 1010 | 0110 |
| 1011 | 0110 |
| 1100 | 1100 |
| 1101 | 1001 |
| 1110 | 1110 |
| 1111 | 1000 |

- 4) Vrai ou Faux.
 - a) Une mémoire volatile perd les données qu'elle contient si l'alimentation est interrompue.
 - b) Toutes les mémoires mortes sont reprogrammables par l'utilisateur.
 - c) Le PROM existe en version bipolaire et MOS.
 - d) Il est impossible d'effacer seulement qu'une partie d'une mémoire EPROM.
 - e) Une MROM peut être programmée par l'utilisateur.
- 5) Donnez le nom des trois BUS qui permettent la communication entre l'UCT et le dispositif de mémoire.
- 6) Quelles sont les différentes étapes utilisées par l'UCT pour la lecture d'une donnée en mémoire ?

- 7) Quelles sont les différences entre une cellule de mémoire VIVE statique et celle d'une mémoire dynamique ?
- 8) Quel est l'avantage du multiplexage d'adresse?
- 9) On désire assembler plusieurs PROM 2K x 8 pour obtenir une capacité totale de 8K x 8. Indiquez combien il faut de puces PROM et combien de conducteurs dans le bus d'adresse.

Exercice 4:

- 1) Sur la figure suivante s'agit-il de mémoire RAM ou ROM? Justifiez votre réponse.

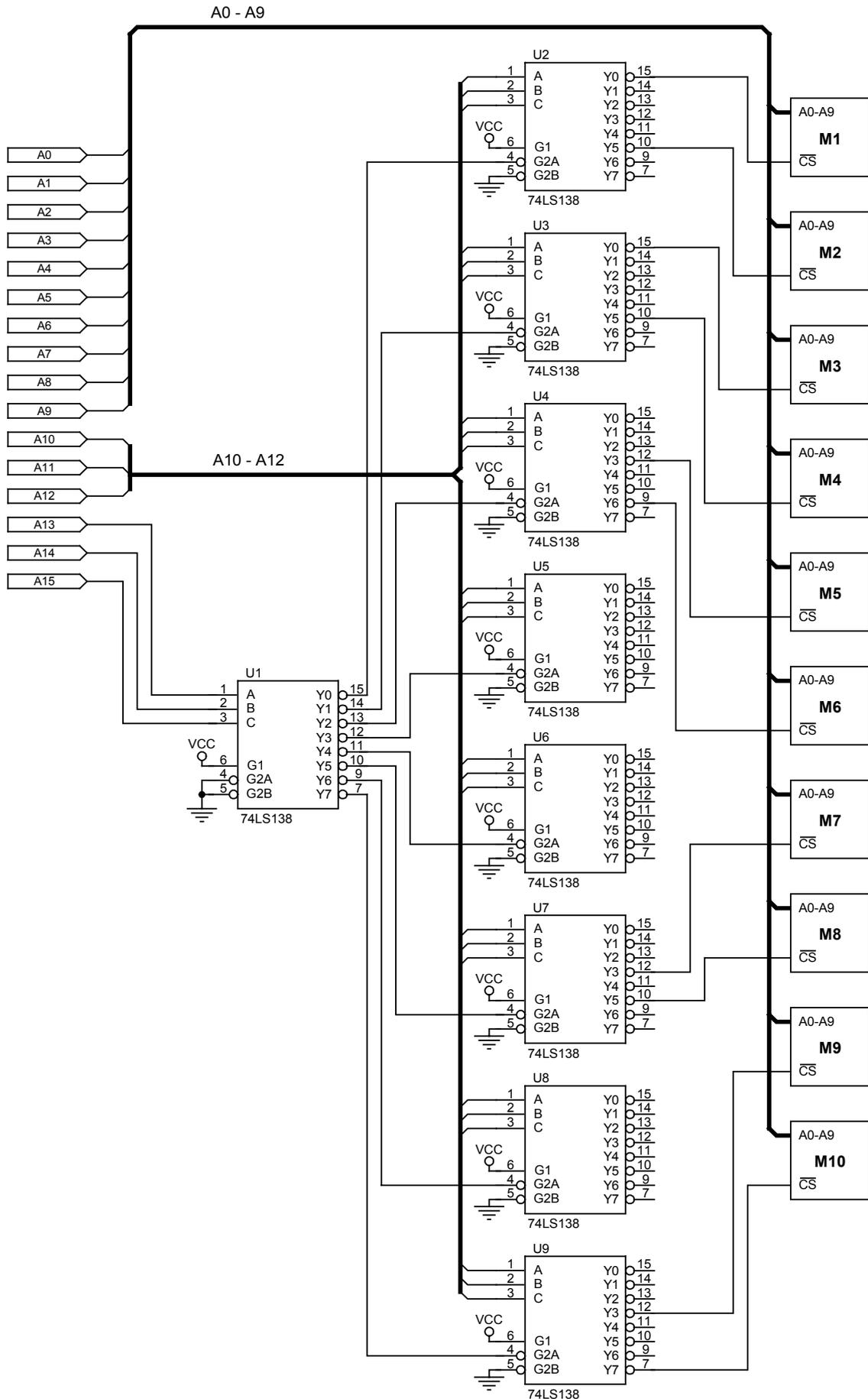


- 2) Donnez le nombre de mots et le nombre de bits par mot de cette mémoire.
- 3) Quelles sont les entrées CS actives sur des 0?
- 4) A quel niveau devez-vous placer l'entrée Read/Write pour être en mode d'écriture?
- 5) À l'aide de cette mémoire, réalisez une carte de 1k octets.

Exercice 5:

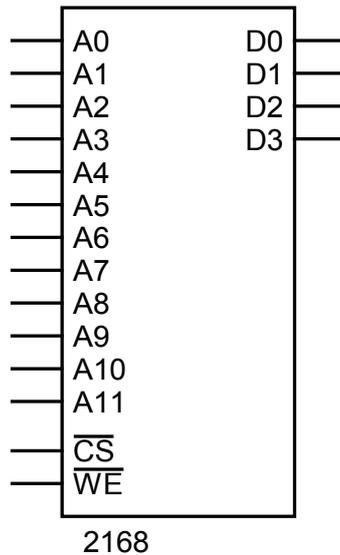
Sur la figure suivante :

- 1) Quelle est la capacité de chacune des mémoires ?
- 2) Quelle est la capacité totale de la carte mémoire ?
- 3) Quelle est la plage d'adresse de la mémoire M7 ?



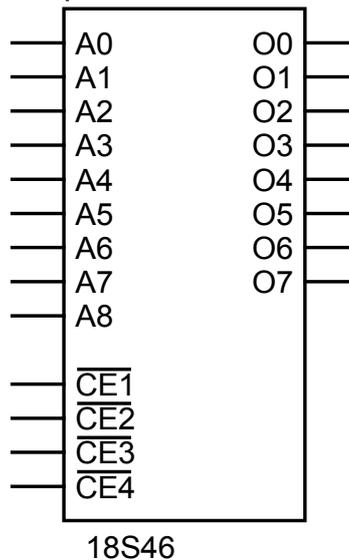
Exercice 6:

- 1) La mémoire 2168, représentée à la figure suivante est de type RAM ou ROM? Justifiez votre réponse.
- 2) Quelle est la capacité de cette mémoire (exprimée en nombre de mots par nombre de bits par mot)?



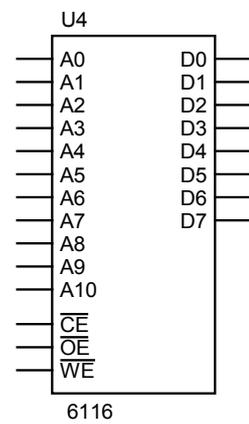
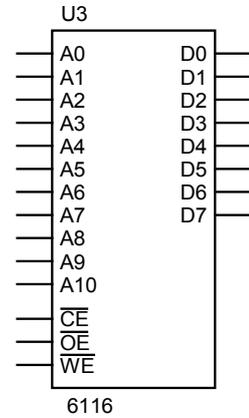
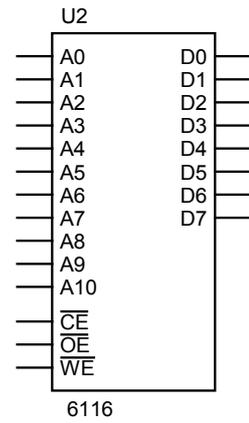
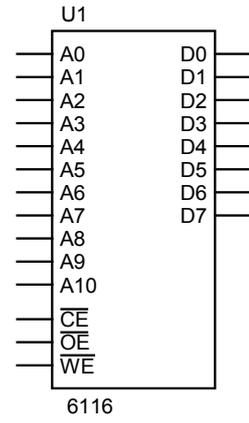
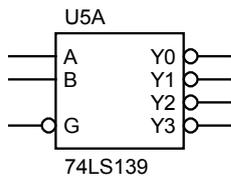
Exercice 7:

- 1) La mémoire 18S46, représentée à la figure suivante est de type RAM ou ROM? Justifiez votre réponse.



Exercice 8 :

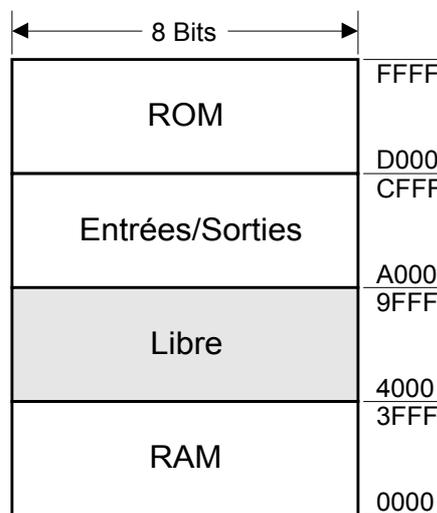
- 1) Utilisez les mémoires 6116 à la page suivante pour réaliser une carte mémoire de 8 k x 8.



Exercice 9:

D'après la cartographie de l'ordinateur, représentée à la figure suivante établissez, en kilo octets, l'espace occupé par:

- 1) La mémoire RAM = _____ k octets
- 2) La mémoire ROM = _____ k octets
- 3) Les entrées/sorties = _____ k octets
- 4) La région libre = _____ k octets



TP.1 – Essai des bascules élémentaires :

1) Objectif visé :

Vérifier le comportement des bascules et approfondir l'analyse de leur fonctionnement.

2) Durée du TP :

3 Heures.

3) Matériel requis :

- Circuits intégrés de la famille 74 : 7400,7402,7405 et 74LS76;
- Boutons-poussoirs N.O. (2);
- Diodes électroluminescentes (3);
- Résistances de 1/2 W (5) : 200 Ω (3) et 1 k Ω (2);
- Fiches techniques des circuits intégrés.

4) Description du TP :

Dans cet exercice, vous procéderez à la vérification de la table de vérité de quelques bascules. Vous aurez à solliciter les entrées synchrones et asynchrones et à déterminer leur effet sur l'état de la bascule.

5) Déroulement du TP :

1. Faites d'abord la lecture complète des diverses étapes de cet exercice.
2. À l'aide des fiches techniques des composants, procédez d'abord à la numérotation des bornes des circuits selon les composants mis à votre disposition. Réalisez ensuite le montage du circuit de la figure suivante :

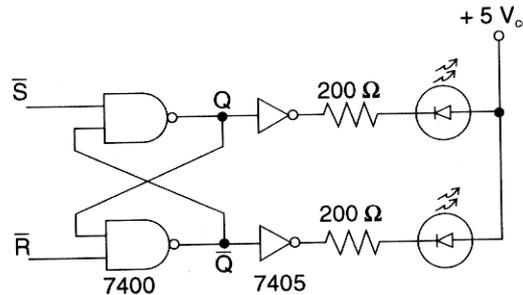


Table de vérité

| S | R | Q | Q̄ |
|---|---|---|----|
| 0 | 0 | | |
| 0 | 1 | | |
| 1 | 0 | | |
| 1 | 1 | | |

Bascule R S en porte NON - ET

- À l'aide des interrupteurs logiques, simulez les entrées synchrones \bar{S} et \bar{R} correspondant à chacune des étapes de la table de vérité afin de vérifier l'état de la bascule. Inscrivez le résultat des sorties.
- Lorsque $\bar{S} = \bar{R} = 1$, que pouvez-vous conclure sur l'état de la bascule par rapport à l'état précédent?
- Modifiez votre montage afin d'obtenir le circuit de la figure suivante. Répétez l'étape 3 pour déterminer vos résultats.

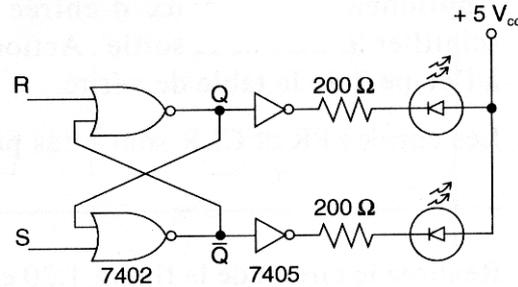


Table de vérité

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

Bascule R S en porte NON - OU

- Décrivez en quelques mots la différence entre les deux montages précédents selon leur table de vérité.
- Réalisez le montage de la figure suivante en appliquant un signal d'horloge à commande manuelle.

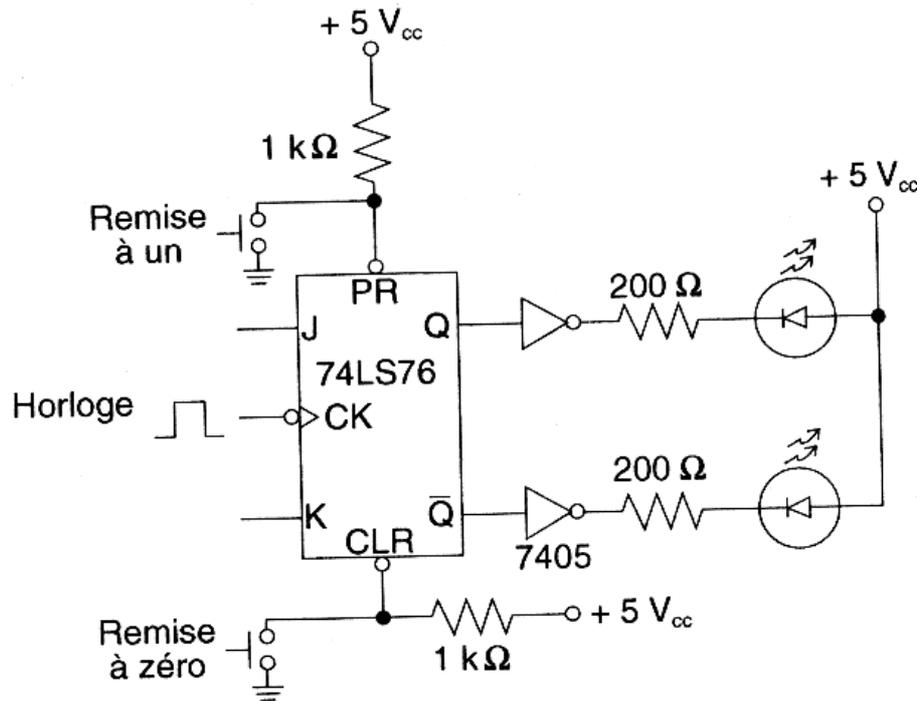


Table de vérité

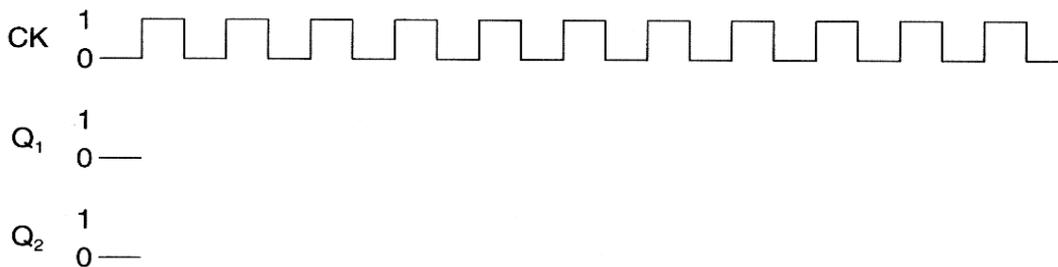
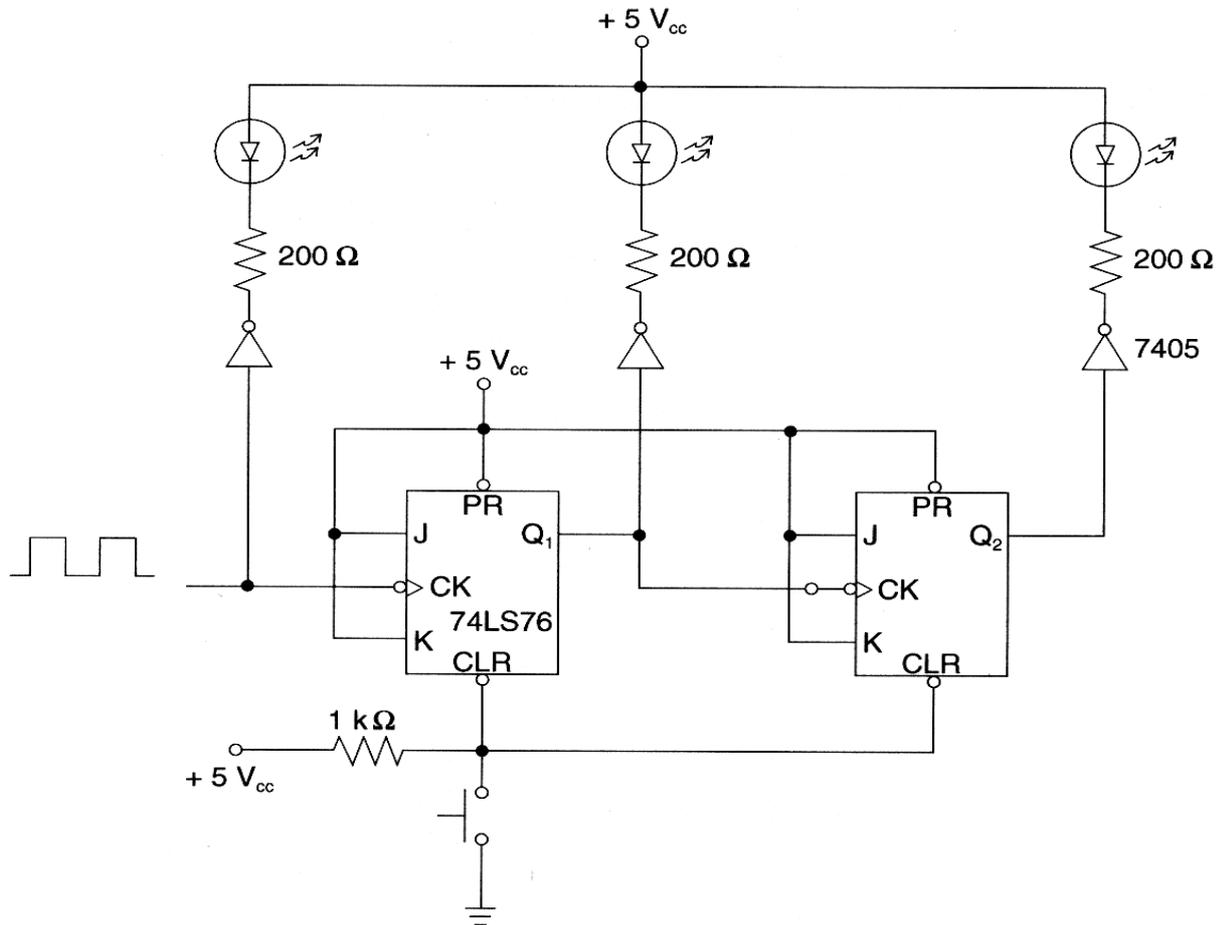
| Étape | PR | CLR | J | K | CK | Q | \bar{Q} |
|-------|----|-----|---|---|----|---|-----------|
| 1 | 1 | 1 | X | X | 0 | | |
| 2 | 1 | 1 | 0 | 0 | ↓ | | |
| 3 | 1 | 1 | 0 | 1 | ↓ | | |
| 4 | 1 | 1 | 1 | 0 | ↓ | | |
| 5 | 1 | 1 | 1 | 1 | ↓ | | |
| 6 | 1 | 0 | X | X | X | | |
| 7 | 0 | 1 | X | X | X | | |

Circuit d'une bascule J k

8. Vérifiez le fonctionnement asynchrone de la bascule (étape 1 de la table de vérité). Sans actionner l'horloge, faites varier les entrées Jet K à votre gré. Notez vos résultats dans la table de vérité. Est-ce que les entrées Jet K affectent le fonctionnement asynchrone de la bascule?
9. Indiquez le résultat des étapes 2 à 5 de la table de vérité
10. Appliquez des signaux d'entrée pour faire scintiller la DEL de la sortie \bar{Q} . Actionnez le bouton-poussoir de remise à zéro et portez vos résultats à l'étape 6 de la table de vérité.
11. Appliquez des signaux d'entrée pour faire scintiller la DEL de la sortie Q. Actionnez le bouton-poussoir de remise à un et portez vos résultats à l'étape 7 de la table de vérité.

12. Les entrées PR et CLR sont-elles prioritaires sur celle de l'horloge?

13. Réalisez le circuit de la figure suivante en appliquant un signal d'horloge de 1 Hz à la première bascule, observez le comportement des DEL et complétez le chronogramme de Q1 et de Q2.



14. À quoi sert le bouton-poussoir?

15. Faites vérifier vos résultats.

TP.2 – Essai des registres à décalage:**1) Objectif visé :**

Vérifier le fonctionnement et les caractéristiques des circuits de registres à décalage.

2) Durée du TP :

3 Heures.

3) Matériel requis :

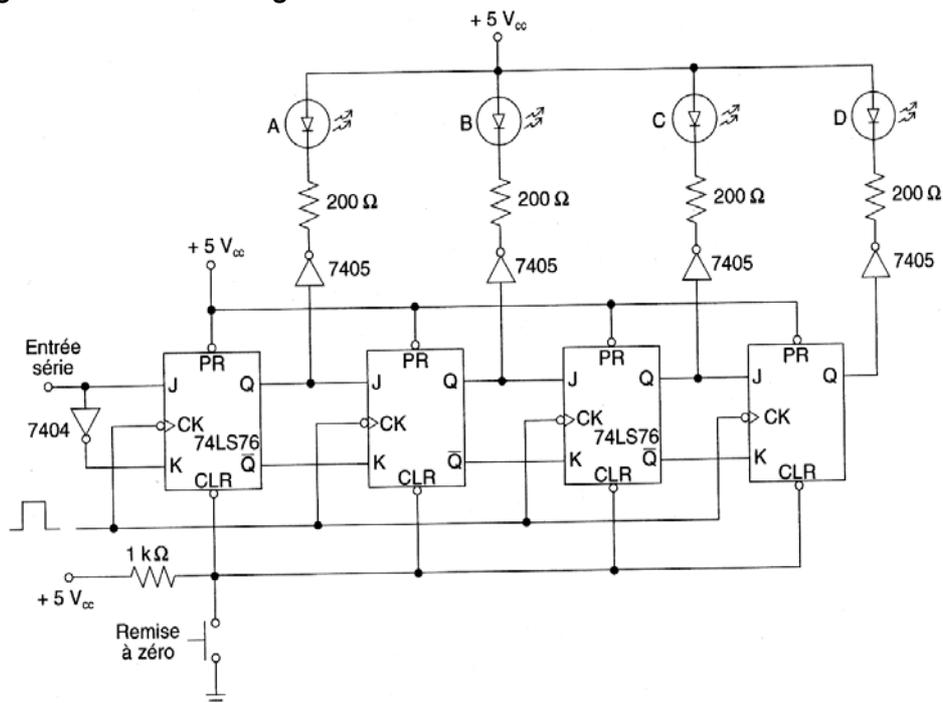
- Circuits intégrés de la famille 74 : 7404, 7405, 74LS76 (2) et 7495;
- Bouton-poussoir N.O.;
- Diodes électroluminescentes (4);
- Résistances de 1/2 W (5) : 200 Ω (4) et 1 k Ω ;
- Fiches techniques des circuits intégrés.

4) Description du TP :

Dans cet exercice, vous réaliserez un registre à décalage à l'aide de composants discrets. Vous pourrez aussi vérifier Le fonctionnement des registres à décalage universels comme le 7495.

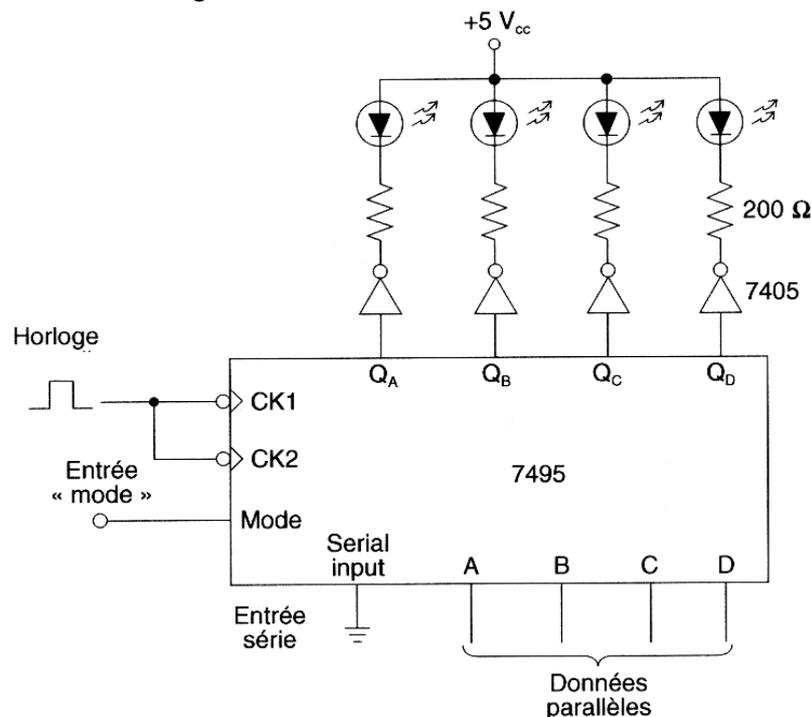
5) Déroulement du TP :

1. Faites d'abord la lecture complète des diverses étapes de cet exercice.
2. À l'aide des fiches techniques des composants, procédez d'abord à la numérotation des bornes des circuits selon les composants mis à votre disposition. Réalisez ensuite le montage du circuit de la figure suivante.



registres à décalage à composants discrets

3. Remettez d'abord le contenu du registre à zéro. Placez l'entrée série à l'état logique 1 et transférez les données à l'aide de quatre impulsions d'horloge. Notez le contenu du registre. DCBA =
4. Placez l'entrée série à l'état logique 0 et transférez les données à l'aide de quatre impulsions d'horloge. Notez le contenu du registre DCBA =
5. Manipulez l'entrée série et l'horloge de manière à charger la valeur DCBA = 1010 en tenant compte du fait que le bit A possède le poids le moins significatif.
6. Faites vérifier vos résultats.
7. Réalisez le circuit de la figure suivante :



Registre universel 7495.

8. Placez l'entrée mode à 1 pour valider le fonctionnement en parallèle. Placez les entrées de données parallèles à 0 et transférez l'information à l'aide d'une impulsion d'horloge. Notez le contenu du registre DCBA =
9. Maintenez l'entrée mode à 1. Placez les entrées de données parallèles à 1 et transférez l'information à l'aide d'une impulsion d'horloge. Notez le contenu du registre DCBA =
10. Placez maintenant l'entrée mode à 0 pour valider le fonctionnement en série. Donnez quatre impulsions d'horloge et notez le contenu du registre DCBA =

TP.3 – Essai des compteurs asynchrones:**1) Objectif visé :**

Vérifier le fonctionnement et les caractéristiques des circuits de comptage à 3 bits

2) Durée du TP :

3 Heures.

3) Matériel requis :

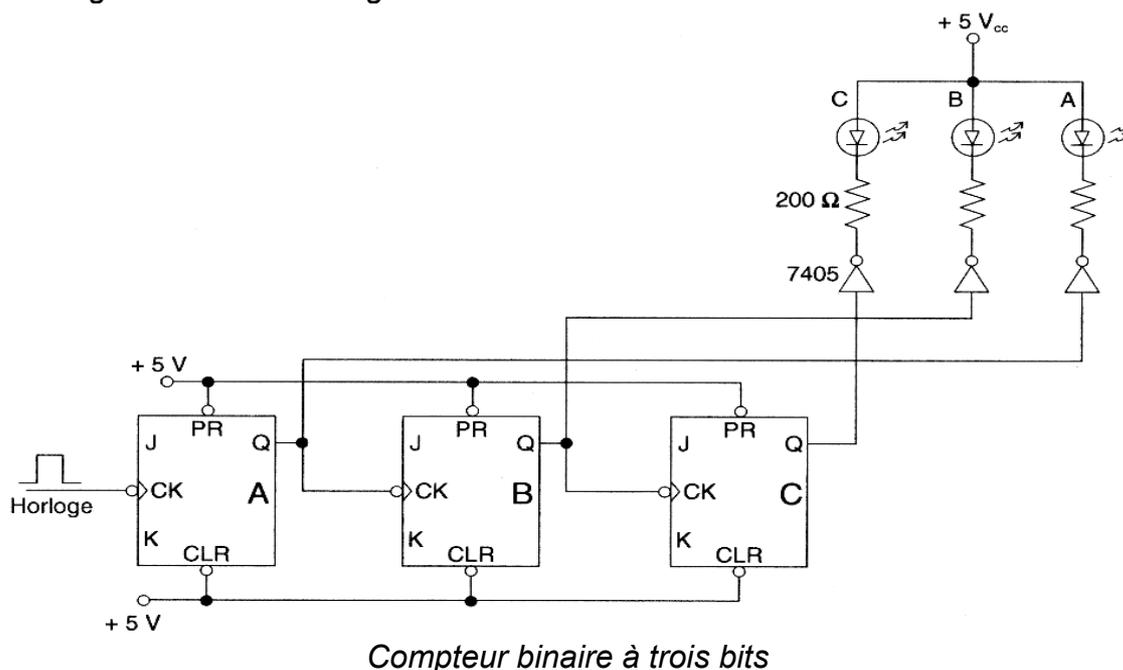
- Circuits intégrés de la famille 74 : ,7405 et 74LS76 (2);
- Diodes électroluminescentes (3);
- Résistances de 200Ω , 1/2 W (3);
- Fiches techniques des circuits intégrés.

4) Description du TP :

Dans cet exercice, vous réaliserez un compteur élémentaire à 3 bits à l'aide de composants discrets. Vous devrez modifier votre circuit pour permettre le comptage et le décomptage des événements

5) Déroulement du TP :

1. Faites d'abord la lecture complète des diverses étapes de cet exercice.
2. À l'aide des fiches techniques des composants, procédez d'abord à la numérotation des bornes des circuits selon les composants mis à votre disposition. Réalisez ensuite le montage du circuit de la figure suivante :



Compteur binaire à trois bits

Remarque :

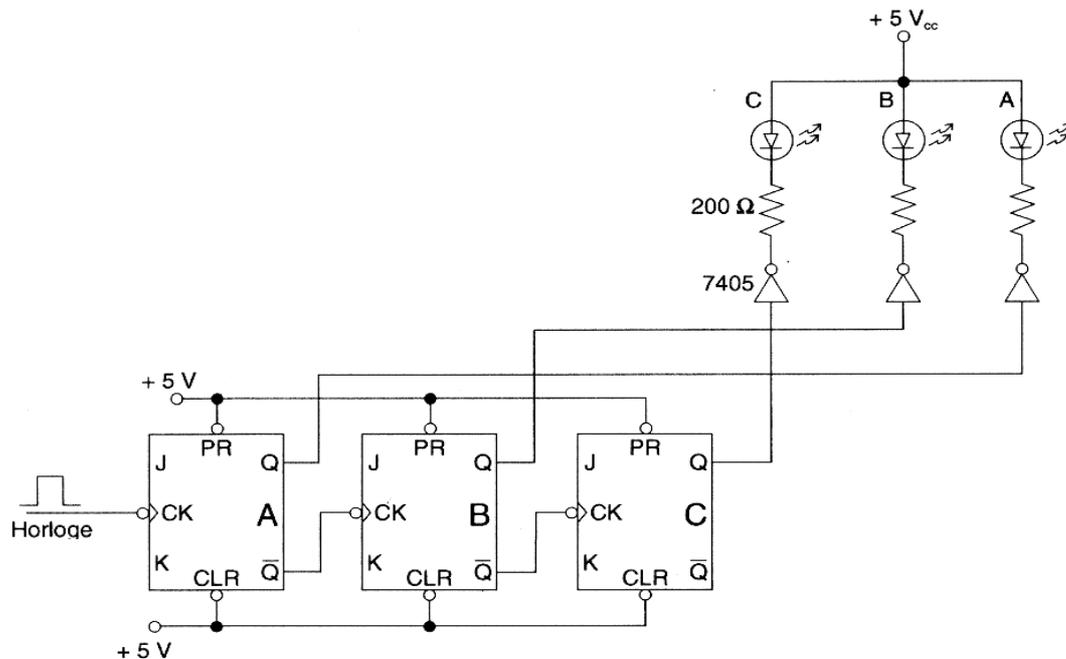
Les entrées $J = 1$ et $K = 1$ peuvent être laissées flottantes, mais il est préférable de toujours les raccorder au potentiel positif de la source. Il est toutefois essentiel de raccorder les entrées asynchrones Preset et Clear..

3. Remettez d'abord le contenu du compteur à zéro. Toutes les DEL doivent être éteintes.
4. Raccordez l'entrée horloge du compteur à une horloge de fréquence de 1 Hz. Notez les valeurs obtenues dans le tableau suivant. Les DEL sont montées dans un ordre ascendant (du bit du poids le plus fort au bit du poids le plus faible)

| | C | B | A |
|---|---|---|---|
| 0 | | | |
| 1 | | | |
| 2 | | | |
| 3 | | | |
| 4 | | | |
| 5 | | | |
| 6 | | | |
| 7 | | | |

tableau des résultats pour le compteur

5. Selon les résultats que vous avez notés dans le tableau quel genre de comptage ce montage accomplit-il ?
6. Modifiez votre montage pour le rendre conforme au circuit de la figure suivante :



Décompteur binaire à trois bits

7. Remettez d'abord le contenu du compteur à un. Toutefois les DEL doivent être allumées.
8. Raccordez l'entrée du compteur à une horloge de fréquence de 1Hz et notez les valeurs obtenues dans le tableau suivant :

| C | B | A | Valeur décimale |
|---|---|---|-----------------|
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

Tableau des résultats du décompteur

9. Selon les résultats que vous avez notés dans le tableau quel genre de comptage ce montage accomplit-il?

TP.4 – Essai des compteurs synchrones:

1) Objectif visé :

Vérifier le comportement des compteurs synchrones et approfondir l'analyse de leur fonctionnement

2) Durée du TP :

3 Heures.

3) Matériel requis :

- *Circuits intégrés de la famille 74 : 7408 et 74LS76;*
- *Fiches techniques des circuits intégrés ;*
- *Circuit d'affichage à sept segments ;*

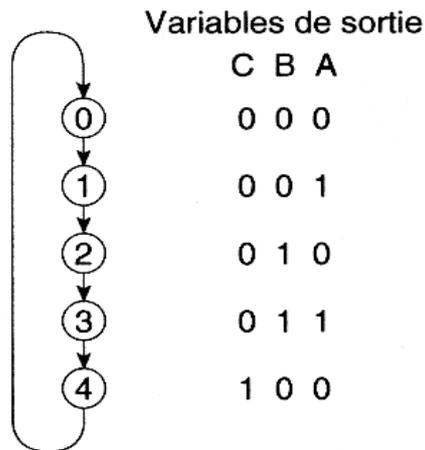
4) Description du TP :

Dans cet exercice, vous tracerez et monterez des circuits de comptage synchrones afin de procéder à la vérification de leur séquence d'opérations. Vous raccorderez vos circuits à un dispositif d'affichage à sept segments.

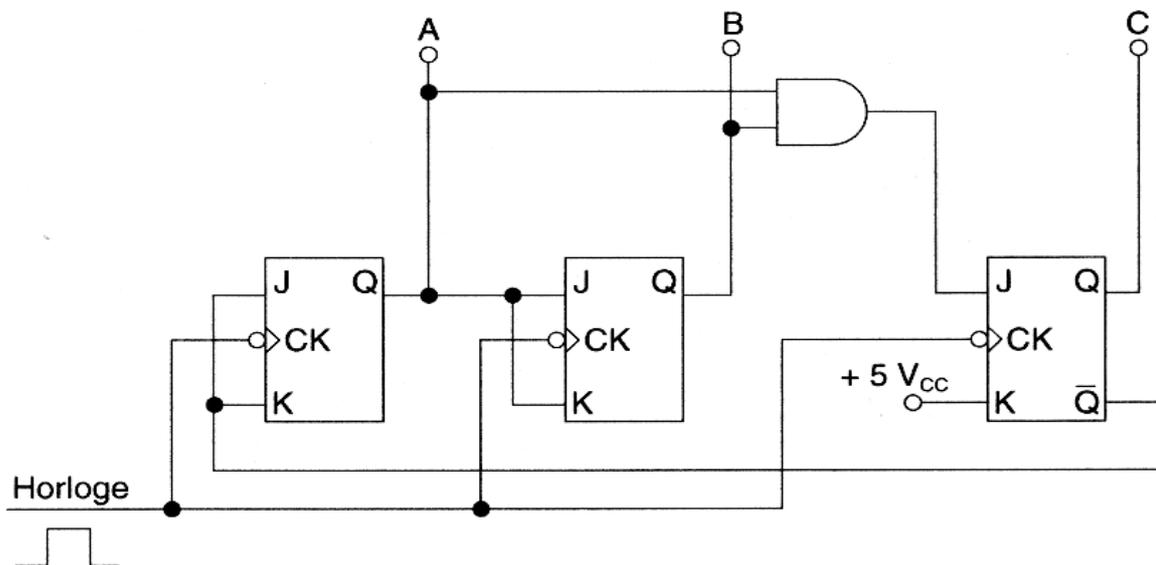
5) Déroulement du TP

- 1. Faites d'abord la lecture complète des diverses étapes de cet exercice.*
- 2. Réalisez le montage du compteur MODULO-5 de la figure suivante en n'oubliant pas de brancher les entrées asynchrones. Utilisez un signal d'horloge de 1 Hz. Reliez les sorties A, B et C à votre circuit d'affichage à sept segments. Prenez soin de forcer l'entrée D du décodeur à un niveau logique BAS pour respecter le compte de 0 à 4.*
- 3. Faites l'essai du compteur MODULO-5 synchrone. Énumérez la séquence de nombres obtenue.*
- 4. Débranchez l'entrée D du décodeur et laissez-la flottante. Énumérez la séquence de nombres obtenue. Est-ce encore un compteur MODULO-5?*

a) Diagramme d'états



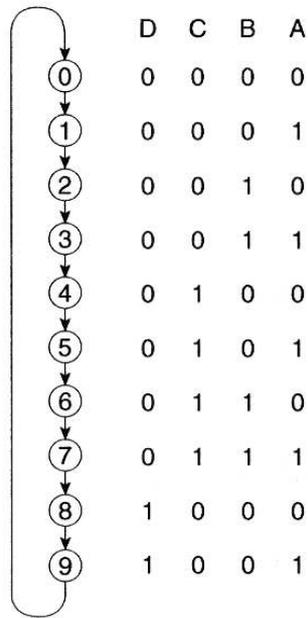
b) Montage



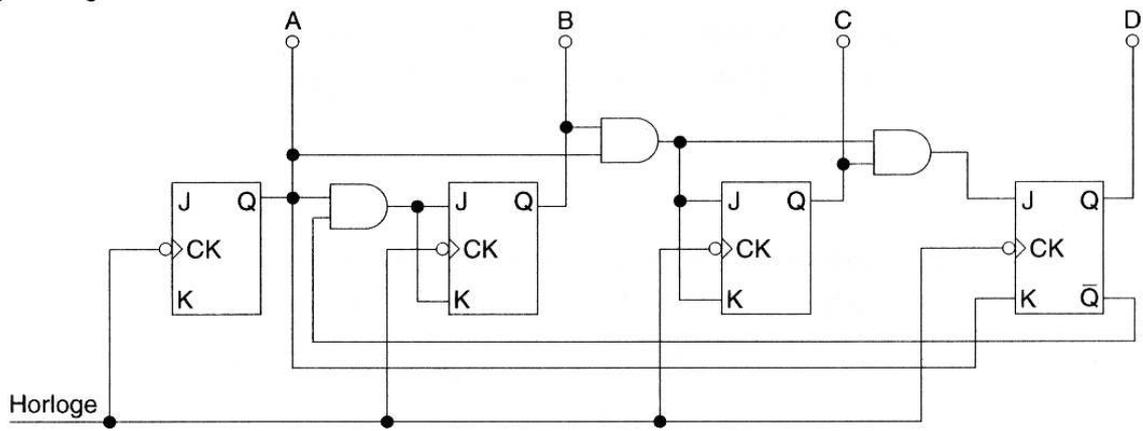
Compteur MODULO-5 synchrone

5. Modifier votre montage afin d'obtenir le compteur DCB. Brancher cette fois l'entrée D du décodeur au circuit de comptage. Vérifier le fonctionnement du circuit

a) Diagramme d'états



b) Montage



Compteur modulo-10 synchrone

TP.5 – Essai des compteurs intégrés :

1) Objectif visé :

Vérifier le comportement des compteurs intégrés et leur fonctionnement

2) Durée du TP :

3 Heures.

3) Matériel requis :

- Circuits intégrés de la famille 74 : 7490 ;
- Fiches techniques des circuits intégrés ;
- Circuit d'affichage à sept segments ;

4) Description du TP :

Dans cet exercice, vous monterez différents circuits de comptage à l'aide de compteurs intégrés 7490. Vous aurez l'occasion de vérifier leur polyvalence en réalisant des compteurs MODULO, des diviseurs de fréquence, un circuit de comptage à deux chiffres.

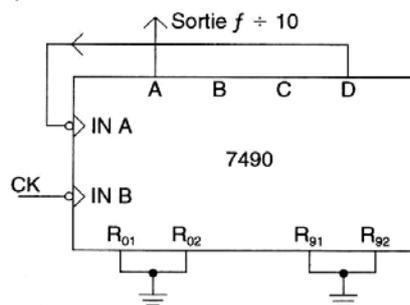
5) Déroulement du TP

1. A l'aide du compteur asynchrone 7490, réaliser le montage du circuit diviseur par dix symétriques de la figure suivante :

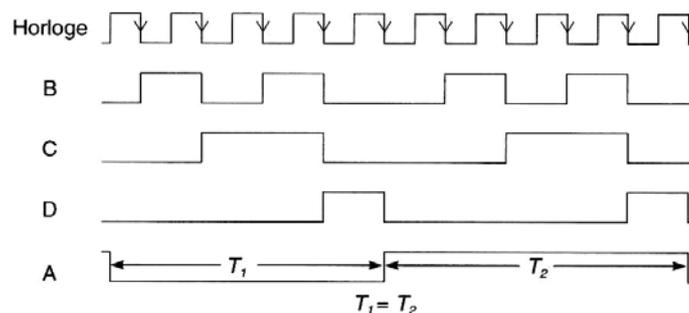
a) Table de vérité

| COUNT | OUTPUT | | | |
|-------|----------------|----------------|----------------|----------------|
| | Q _A | Q _B | Q _C | Q _D |
| 0 | L | L | L | L |
| 1 | L | L | L | H |
| 2 | L | L | H | L |
| 3 | L | L | H | H |
| 4 | L | H | L | L |
| 5 | H | L | L | L |
| 6 | H | L | L | H |
| 7 | H | L | H | L |
| 8 | H | L | H | H |
| 9 | H | H | L | L |

b) Branchement



b) Chronogramme



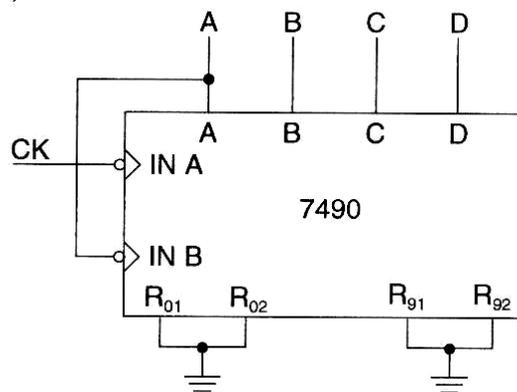
2. Par quel facteur le compteur divise-t-il la fréquence d'entrée ?
3. Modifier le montage pour obtenir un compteur DCB. Relier les sorties du compteur à votre circuit d'affichage à 7 segments et vérifier le fonctionnement à l'aide d'un signal d'horloge de 1Hz.

a) Table de vérité

'90A, 'L90, 'LS90
BCD COUNT SEQUENCE
(See Note A)

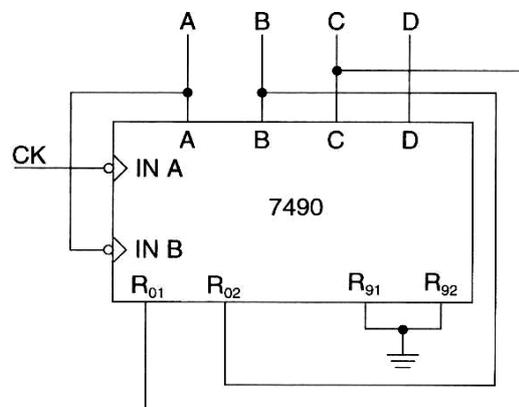
| COUNT | OUTPUT | | | |
|-------|----------------|----------------|----------------|----------------|
| | Q _D | Q _C | Q _B | Q _A |
| 0 | L | L | L | L |
| 1 | L | L | L | H |
| 2 | L | L | H | L |
| 3 | L | L | H | H |
| 4 | L | H | L | L |
| 5 | L | H | L | H |
| 6 | L | H | H | L |
| 7 | L | H | H | H |
| 8 | H | L | L | L |
| 9 | H | L | L | H |

b) Branchement



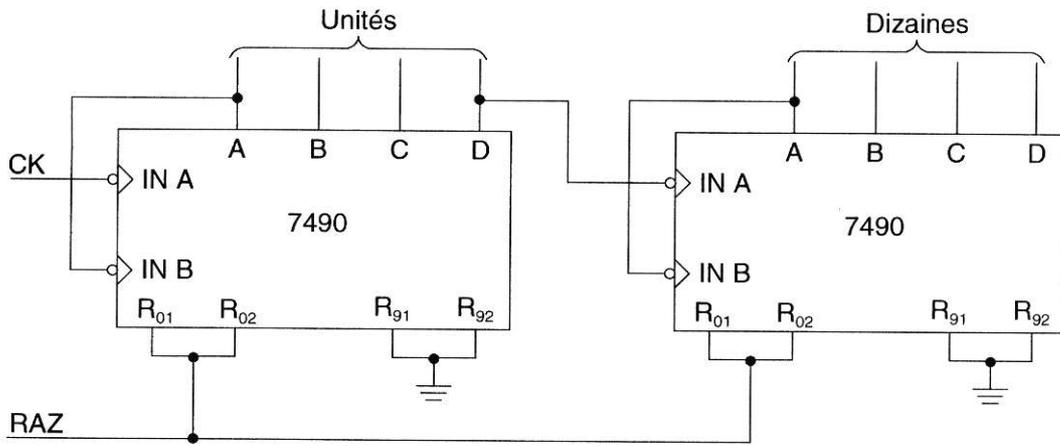
compteur DCB (modulo10)

4. Modifier le montage pour obtenir un compteur modulo-6 et vérifier le fonctionnement



compteur modulo-6

5. En utilisant deux compteurs 7490 réaliser un compteur modulo-100



compteur modulo-100

ÉVALUATION DE FIN DE MODULE :

1) Durée :3 h.

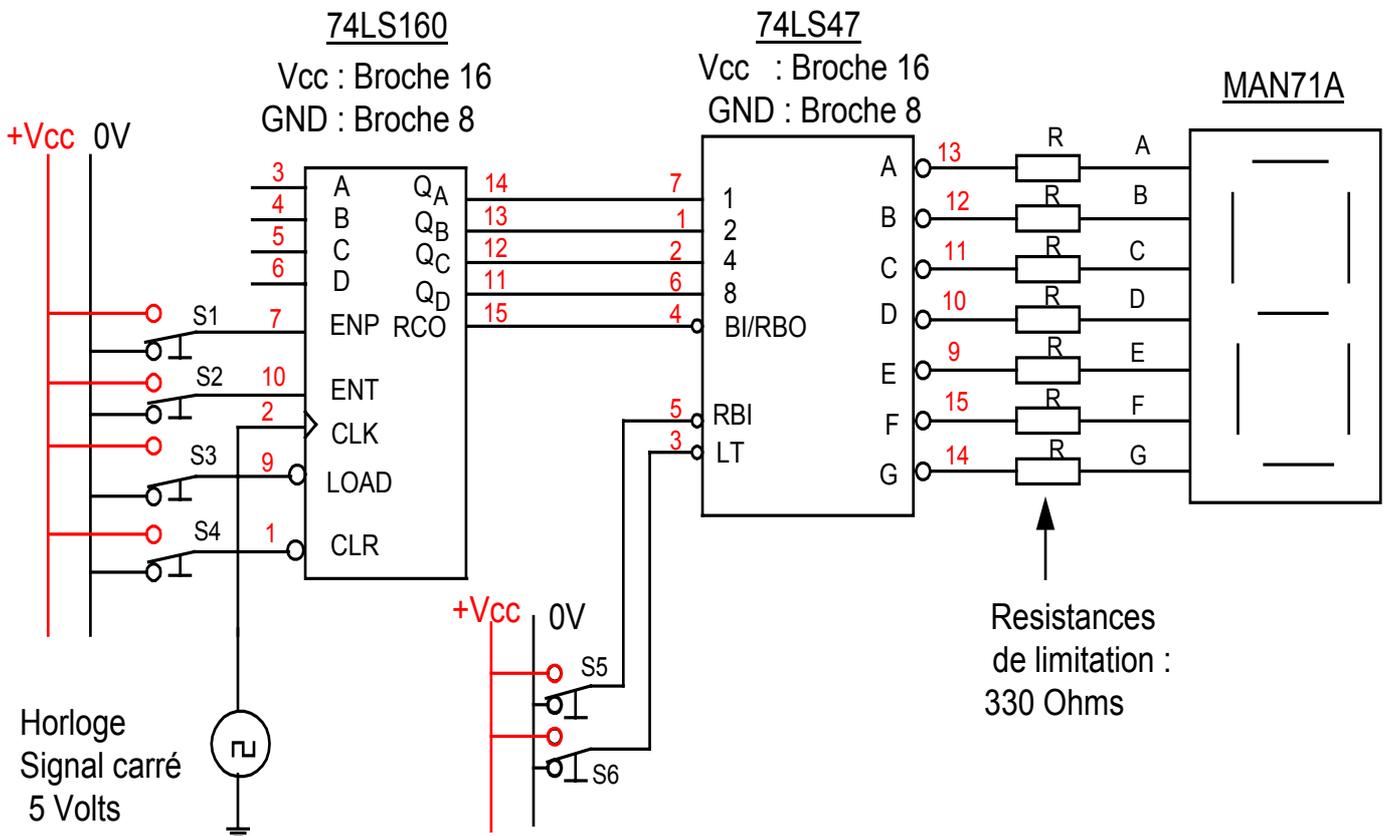
Compteur-décodeur et affichage numérique :

2) Compétence visée :

Monter un circuit dont la fonction est de compter puis d'afficher le résultat du compte en utilisant un décodeur intégré 74LS47 (interface un affichage 7 segments sur un compteur).

3) Démarches à suivre :

1. Monter le circuit de la figure suivante :



Compteur-décodeur et affichage numérique.

2. Vérifier le fonctionnement de votre circuit numérique en le faisant compter de 0 à 9 à l'aide d'une horloge à basse fréquence (utiliser le générateur de fonction)

N.B : N'oublier pas de choisir les niveaux logiques convenables pour ENP, ENT, LOAD, et CLR du compteur.

3. Quel est le rôle de la broche LT du décodeur 74LS47 ainsi que son niveau actif ?
4. Pendant que le compteur fonctionne et affiche les nombres de 0 à 9, que remarquez-vous de différent dans l'affichage de la séquence lorsque vous placez un niveau 0 sur l'entrée BI du décodeur.

Liste bibliographique

Liste des références bibliographiques

| Ouvrage | Auteur | Edition |
|---|---|------------------|
| <i>Equipements et installations électriques</i> | <i>G.Augereau A.Bianciotto P.Boyo</i> | <i>Delagrave</i> |
| <i>Schémas et études d'équipements</i> | <i>G.Augereau A.Bianciotto P.Boyo</i> | <i>Delagrave</i> |
| <i>Catalogue des principaux circuits intégrés</i> | <i>Raymond Dreyfuss Philippe Tixier</i> | <i>Weka</i> |
| <i>Module 21: (Électromécanique de systèmes Automatisés) Logique séquentielle</i> | <i>Alain Sirois</i> | <i>CEMEQ</i> |