

Systemes Numériques

COURS L2
Année 2013-2014

Isabelle Sirot

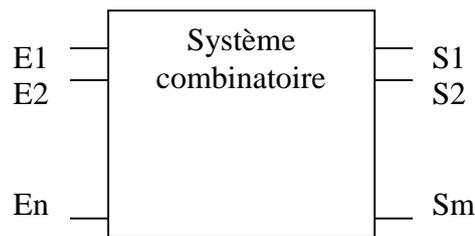
Table des matières

Chapitre 1 : Opérations séquentielles	3
1 Généralités	3
2 Les Bascules	4
2.1 Bascules asynchrones	4
2.2 Bascules synchrones	6
2.2.1 Synchronisation sur niveau	6
2.2.2 Synchronisation sur front positif ou négatif	7
2.2.3 Les entrées asynchrones	10
Chapitre 2 : Registres et Compteurs	12
1 Les registres	12
1.1 Les fonctions de décalage et de rotation	12
1.2 Les différents types de registres	14
1.3 Registres universels	16
2 Les compteurs	16
2.1 Compteurs asynchrones	18
2.2 Compteurs synchrones	20
2.3 Compteurs intégrés	22
Chapitre 3 : Les mémoires	24
1 Généralités	24
1.1 Caractéristiques générales	24
1.2 Hiérarchie des mémoires	25
2 Mémoire centrale : mémoire à semi-conducteurs	26
2.1 Fonctionnement d'une mémoire	26
2.2 Les mémoires vives	28
2.3 Les mémoires mortes	30
2.4 Constitution des blocs mémoires	34
2.5 Décodage d'adresse	35

Chapitre 1 : Opérations séquentielles

1 Généralités

Dans les circuits combinatoires les sorties sont déterminées uniquement en fonction des variables d'entrée.



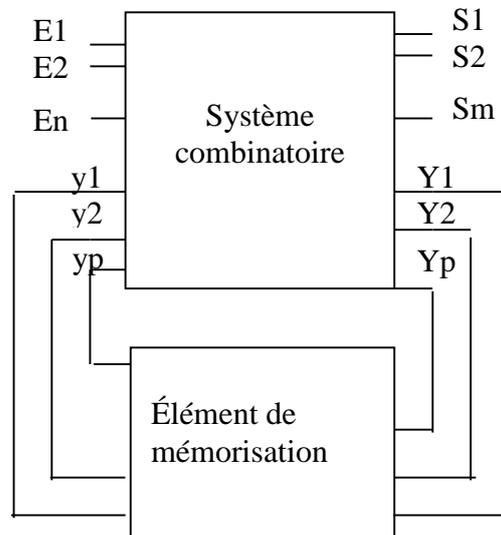
Système combinatoire à n entrées ($E1, E2, \dots, En$) et m sorties ($S1, S2, \dots, Sm$).

Un système combinatoire possédant n entrées est un système à 2^n état

Dans un système séquentiel l'état ne dépend pas uniquement de la combinaison des entrées à un instant donné mais de la **séquence**, au cours du temps, des événements qui ont été appliqués sur les entrées. Les sorties sont en fonction du passé du circuit, il possède donc une fonction mémoire.

On peut considérer un circuit séquentiel comme un circuit combinatoire ayant pour entrées des variables internes, elle même fonction de l'état du système. Les systèmes séquentiels apparaissent donc comme des **systèmes bouclés**.

Soit un système à n entrées ($E1, E2, \dots, En$), p variables internes et m sorties ($S1, S2, \dots, Sm$), il peut être représenté par figure suivante.



Y_1, Y_2, \dots, Y_p représente les états suivants qui sont calculées en fonction des entrées E_1, E_2, \dots, E_n et des états présents y_1, y_2, \dots, y_p . La plupart du temps l'élément de mémorisation est un retour direct.

On distingue deux types de système et circuit séquentiels :

- Les circuits séquentiels synchrones, les sorties ne peuvent évoluer que si un signal d'horloge est actif.
- Les circuits séquentiels asynchrones, les sorties évoluent dès qu'il y a un changement sur l'une des entrées.

Les systèmes asynchrones sont généralement moins fiables car ils sont plus sensibles aux parasites.

2 Les Bascules

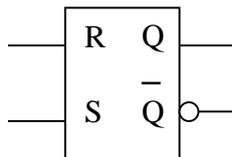
Les bascules sont les circuits logiques de base de la logique séquentielle. On retrouve les deux types de fonctionnement décrit précédemment.

2.1 Bascules asynchrones

La bascule RS est la bascule asynchrone la plus simple.

Elle comporte

- Une entrée R (Reset) de mise à zéro
- Une entrée S (Set) de mise à un
- Une sortie Q et son inverse \bar{Q}



La table de vérité de cette bascule est la suivante :

R	S	Q_{t+1}
0	0	Q_t (mémoire)
0	1	1
1	0	0
1	1	Non défini

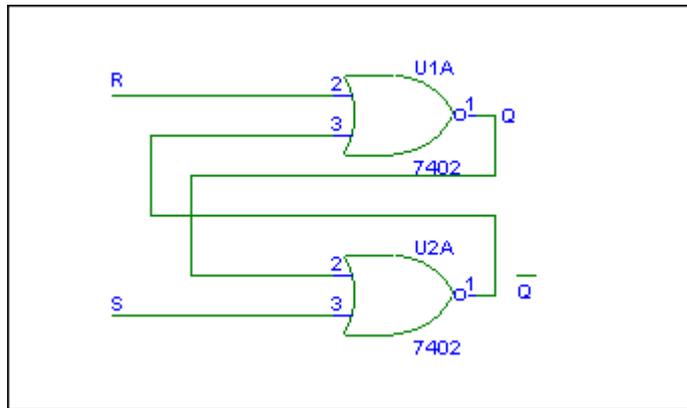
Quand $R=S=0$, la sortie Q_{t+1} (sortie au temps $t+1$) ne change pas elle est égale à Q_t (sortie à l'instant t), la bascule fonctionne en mémoire.

Quand $S=1$ et $R=0$ la sortie est mise à 1 (Set).

Quand $R=1$ et $S=0$ la sortie est mise à zéro.

La combinaison des entrées $S=R=1$ est une combinaison interdite car les sorties Q et \bar{Q} ont la même valeur logique.

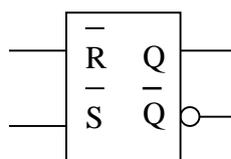
Cette porte peut être constituée de deux portes NOR.



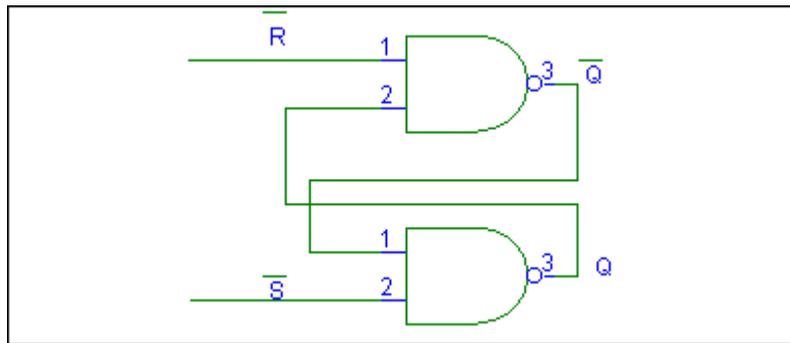
On définit également la bascule \bar{R} \bar{S} qui a la table de vérité suivante :

\bar{R}	\bar{S}	Q_{t+1}
0	0	Non défini
0	1	0
1	0	1
1	1	Q_t (mémoire)

Le symbole de cette porte est :



Cette porte peut être constituée de portes NAND.



On remarque que les sorties sont bouclées sur les entrées.

2.2 Bascules synchrones

La synchronisation des circuits séquentiels peut se faire de trois façons différentes.

- Synchronisation sur niveau : les changements d'état s'effectuent en fonction du niveau (0 ou 1) de l'horloge.
- Synchronisation sur front : Les bascules changent d'état uniquement sur un front montant ou descendant de l'horloge.
- Synchronisation par impulsion : les bascules changent d'état après deux fronts successifs de l'horloge (front montant puis descendant ou vice versa).

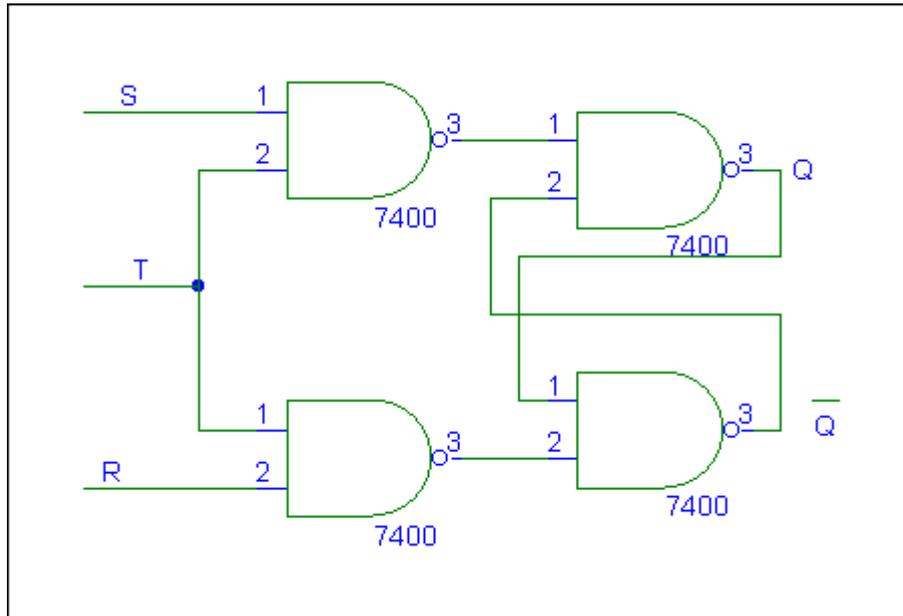
2.2.1 Synchronisation sur niveau

a) Bascule RST

La bascule $\bar{R} \bar{S}$ est construite à partir de portes NAND. On introduit alors un signal de synchronisation d'horloge noté T. Le changement d'état est autorisé quand le signal d'horloge T vaut 1. Les portes NAND bloquent le signal quand T=0.

Quand T=1, la bascule fonctionne comme une RS

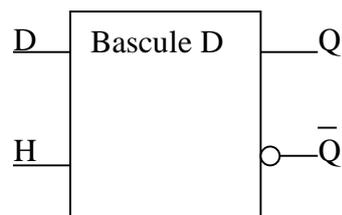
Quand T=0, la bascule garde l'état précédent : la bascule fonctionne en mémoire.



b) Bascule D Latch

Cette porte se représente sous forme de circuit intégré avec deux entrées D (donnée) H (horloge) et sa sortie Q sous forme directe et complémentée \bar{Q} .

La bascule recopie l'entrée D en sortie Q quand l'horloge est active c'est à dire sur niveau haut en logique positive (sur le niveau bas en logique négative). Quand l'horloge est inactive, la bascule garde l'état précédent. $Q_{t+1} = Q_t$.



2.2.2 Synchronisation sur front positif ou négatif

Ces bascules changent d'état uniquement sur un front du signal d'horloge sinon elle fonctionne en mémoire.

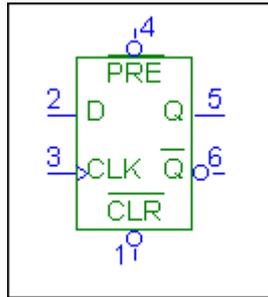
Ce front peut être montant \uparrow ou descendant \downarrow

Ce mode de fonctionnement protège d'éventuels parasites sur les entrées car les entrées ne sont prises en compte que pendant la durée d'un front (qui est considéré comme nul dans la suite de ce cours).

a) Bascule D (edge triggered)

L'entrée D n'est prise en compte qu'au front montant (\rightarrow) ou descendant (\leftarrow) de l'horloge.

Son symbole pour un fonctionnement sur front montant est le triangle devant l'horloge CLK

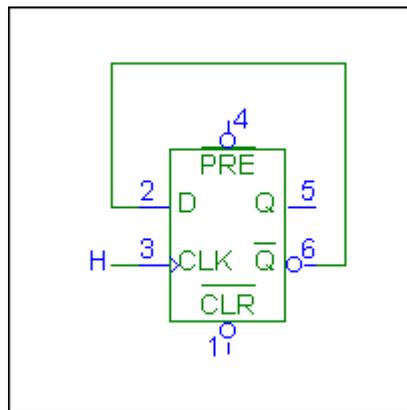


Si un rond est présent devant le triangle la synchronisation se fait sur front descendant

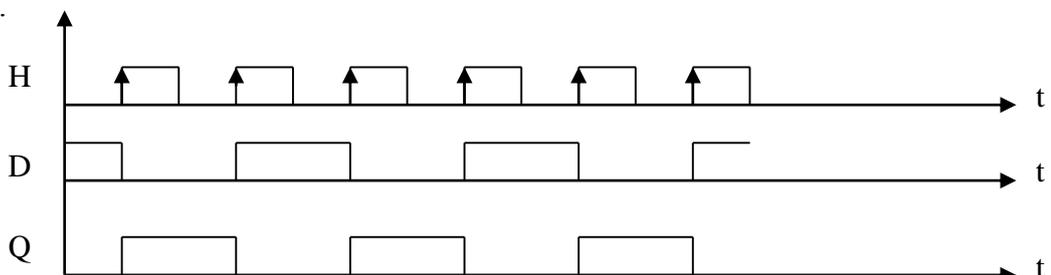
Les entrées \overline{PRE} et \overline{CLR} sont des entrées asynchrones prioritaires que nous traiterons dans un prochain paragraphe.

Quand il y a un front actif la sortie recopie l'entrée sinon la sortie garde son état précédent.

Considérons le circuit suivant : la sortie complémentée est rebouclée sur l'entrée D.



Considérons que D est au départ à 1. Traçons le chronogramme (schéma représentant l'évolution des entrées et des sorties en fonction du temps) de ces signaux pour 6 impulsions d'horloge.



Au départ $Q = 0$ donc $D = \bar{Q} = 1$. Au premier front d'horloge actif, la sortie Q recopie la valeur de D ($D=1$) juste avant le front d'horloge et \bar{Q} passe à 0 ainsi que D . Les temps de transition des signaux ne sont pas pris en compte.

Au second front actif de l'horloge, la sortie recopie de nouveau la valeur de l'entrée D juste avant le front d'horloge $D=0$ donc Q passe à 0 et \bar{Q} passe à 1.

Ce circuit permet de diviser par deux la fréquence de l'horloge.

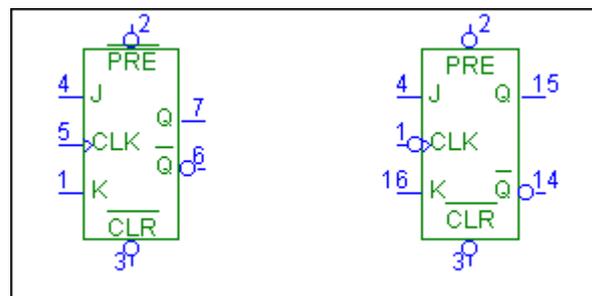
b) Bascule J K edge triggered

Cette bascule a deux entrées données J et K .

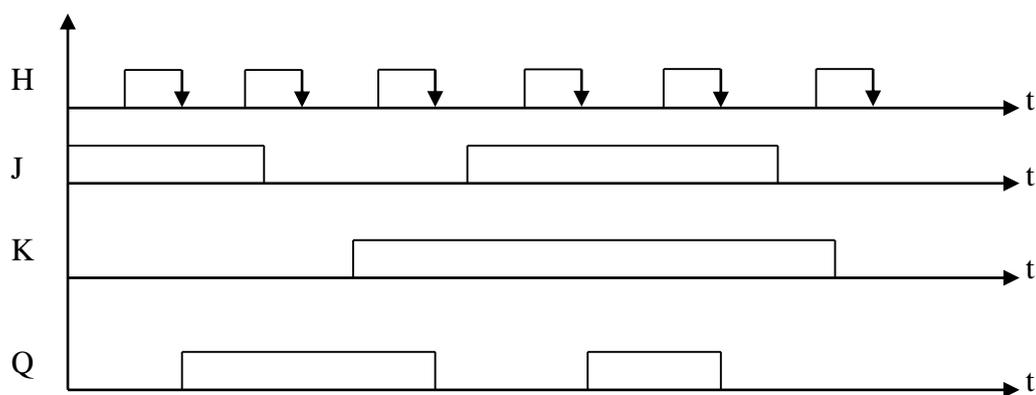
Sa table de fonctionnement est la suivante:

H	J	K	Q_{t+1}
front montant ou descendant	0	0	Q_t Mémoire
front montant ou descendant	0	1	0 Mise à zéro
front montant ou descendant	1	0	1 Mise à un
front montant ou descendant	1	1	\bar{Q}_t changement d'état

Les symboles de ces portes fonctionnant sur front montant et descendant sont :



Exemple de fonctionnement d'une bascule J K sur front descendant, au départ $Q = 0$:



Si l'on place les deux entrées de données J et K à 1, la sortie changera d'état à chaque front actif d'horloge. On réalise alors une bascule T (Toggle). Le signal en sortie aura une fréquence divisée par deux par rapport à celui de l'horloge. Cette propriété est utilisée dans les compteurs que nous étudierons au chapitre deux.

On peut réaliser une bascule D à partir d'une bascule J K, il suffit de poser $J=D$ et $K=\bar{D}$.

Remarque

Il existe également des bascules J K dites à verrouillage: elles prennent en compte les entrées J et K sur front montant (front descendant) et elles affichent les sorties sur front descendant (front montant).

2.2.3 Les entrées asynchrones

Les entrées que nous avons étudiées sont appelées entrées synchrones car les transferts des données sont synchronisés par le signal d'horloge.

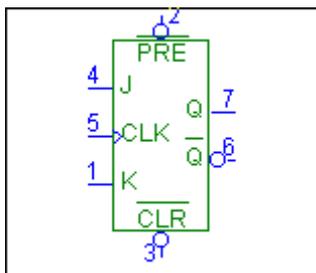
La plupart des bascules sont également munie d'entrées **asynchrones** ou entrées de forçage qui ne dépendent pas de l'horloge et qui sont **prioritaires**.

Il existe deux commandes :

Une commande de mise à 1 : SET ou PRESET ou RAU

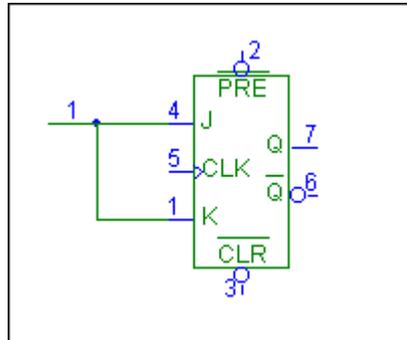
Une commande de mise à 0 : RESET ou CLEAR ou RAZ

Sur les bascules, elles sont situées en haut et en bas du boîtier comme le montre la figure suivante :

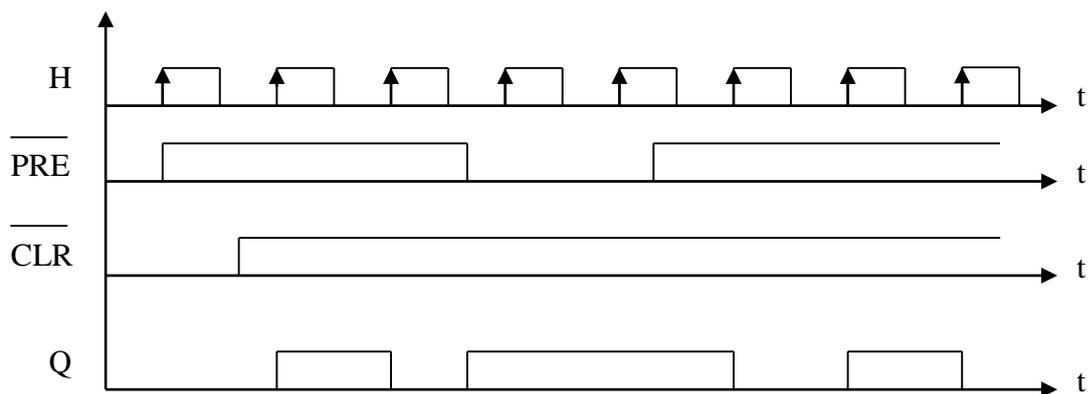


Elles sont généralement active à l'état bas. Un zéro par exemple sur l'entrée \overline{CLR} provoque une remise à zéro de la bascule quelles que soient les valeurs sur les entrées synchrones. Bien évidemment ces deux entrées asynchrones ne peuvent être actives en même temps, c'est à dire toutes les deux à zéro en même temps.

On considère le circuit suivant :



Le chronogramme de fonctionnement est le suivant :



Quand les entrées asynchrones sont inactives (égale à 1), la bascule J K fonctionne en diviseur de fréquence par deux ($J=K=1$ la sortie change d'état à chaque front actif de l'horloge).

Applications des bascules

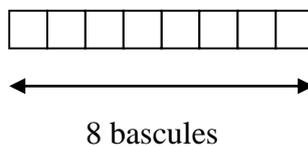
Les bascules sont très utilisées dans les systèmes séquentiels. Outre le fait qu'elles puissent servir de mémoire tampon de 1 bit, elles sont notamment utilisées pour constituer des circuits comme les registres et les compteurs que nous allons étudier dans le chapitre suivant.

Chapitre 2 : Registres et Compteurs

1 Les registres

Un registre est un circuit qui permet de mémoriser de façon temporaire un mot de N bits en attendant son traitement ultérieur. Un registre de longueur N est constitué de N bascules synchronisées par la même horloge. Le nombre de bascules qui le constitue détermine la taille de l'information qu'il peut stocker. Les registres sont des zones de mémorisation interne des microprocesseurs.

Un registre de huit bits est constitué de 8 bascules synchronisées par la même horloge



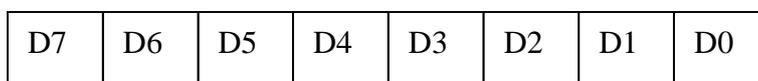
Un registre est un circuit permettant :

- de stocker une information binaire
- de transférer une information dans certaines conditions
- de faire des traitements simples sur les éléments binaires comme des décalages ou des rotations.

1.1 Les fonctions de décalage et de rotation

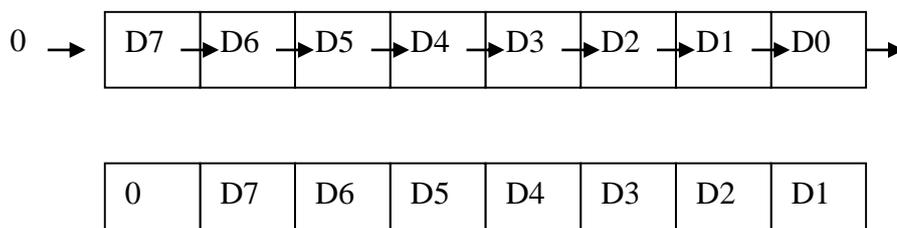
Les systèmes numériques utilisent des mots ou des nombres de n éléments binaires. On a souvent besoin d'avoir accès à un élément binaire ou un groupe d'éléments binaires. On utilise, pour cela, des fonctions de décalage des éléments binaires.

Soit un mot de huit éléments binaires (constitué de huit bascules).



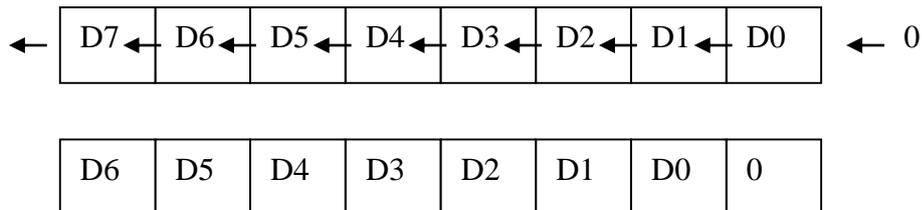
On peut effectuer différents types de décalage sur ce mot.

Décalage à droite : tous les éléments binaires sont décalés d'un rang vers la droite ; il apparaît un 0 (ou un 1) sur l'élément binaire de poids fort (bit le plus à gauche). L'élément binaire de poids faible (le bit le plus à droite) est perdu.



Si la valeur 0 est entrée sur l'élément binaire de poids fort, on obtient une division par deux du nombre initial.

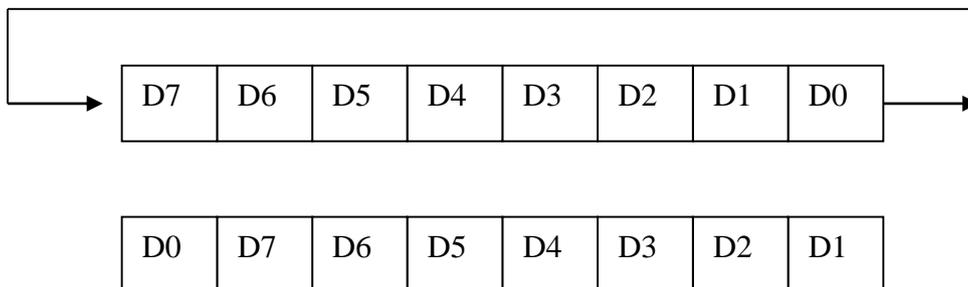
Décalage à gauche : tous les éléments binaires sont décalés d'un rang vers la gauche ; il apparaît un 0 (ou un 1) sur l'élément binaire de poids faible (bit le plus à droite). L'élément binaire de poids fort (le bit le plus à droite) est perdu.



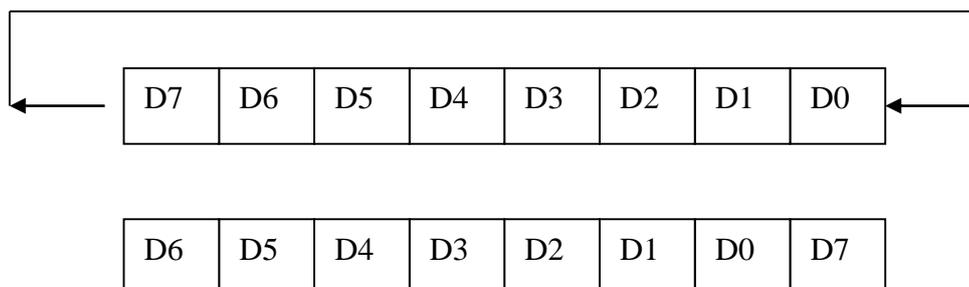
Si la valeur 0 est entrée sur l'élément binaire de poids faible, on obtient une multiplication par deux du nombre initial.

Une rotation est un décalage circulaire.

Une rotation à droite : tous les éléments binaires sont décalés vers la droite et le bit de poids fort prend la valeur du bit de poids faible.



Une rotation à gauche : tous les éléments binaires sont décalés vers la gauche et le bit de poids faible prend la valeur du bit de poids fort.



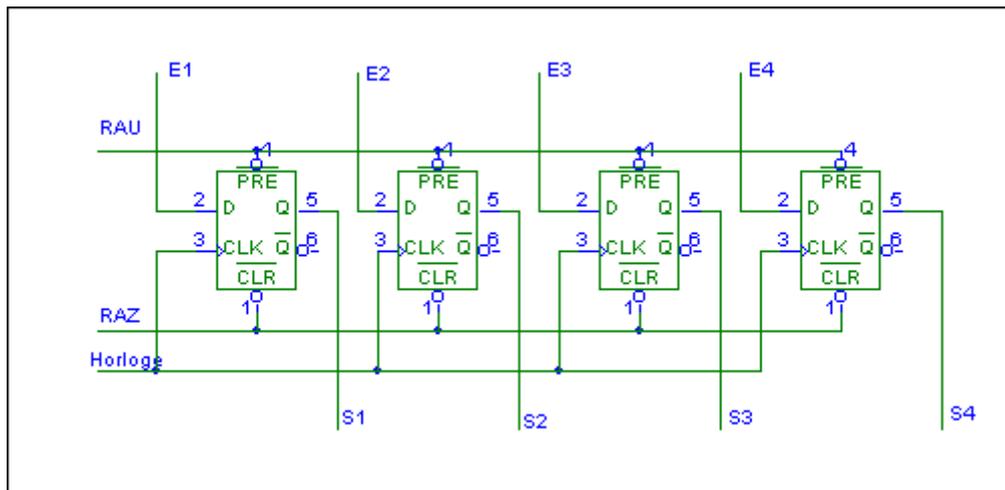
1.2 Les différents types de registres

On distingue quatre types de registres selon la façon dont sont connectées les entrées et les sorties.

Les registres à entrées parallèles, sorties parallèles

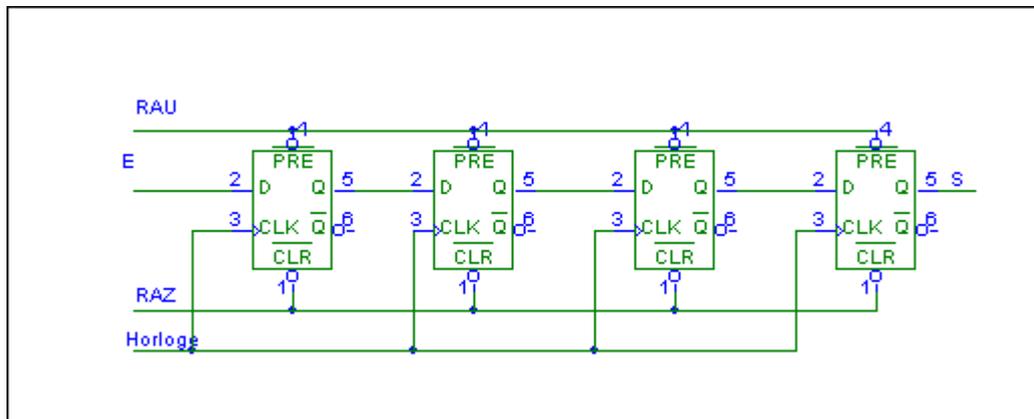
Toutes les entrées (E1, E2, E3, E4) sont introduites en même temps dans le registre. Toutes les sorties (S1, S2, S3, S4) sont disponibles au même instant. Les signaux RAZ et RAU sont des entrées asynchrones permettant la remise à zéro ou la remise à un de toutes les bascules en même temps. Ce type de registre est aussi appelé registre tampon : il est souvent utilisé pour la mémorisation de données dont la présence est de durée brève ou pour le transfert d'information.

On considère un registre de quatre bits. Les bascules utilisées dans les exemples suivants sont des bascules D mais un registre peut également être construit à partir de bascules JK.



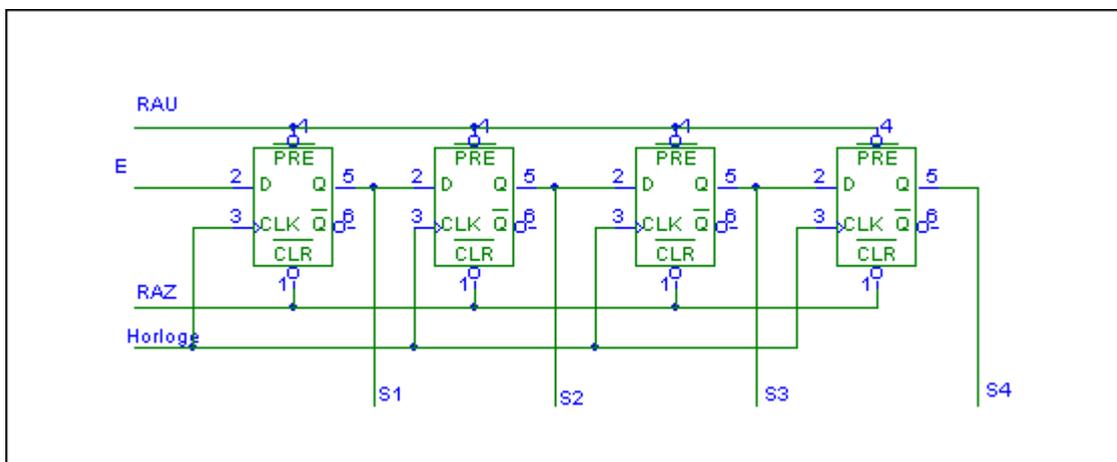
Les registres à entrée série, sortie série

Ce registre possède une entrée E et une sortie S. Les données binaires d'entrée sont introduites les unes après les autres. Elles sont également disponibles les unes après les autres au rythme de l'horloge en sortie. Ce type de registre est utilisé pour effectuer des décalages.



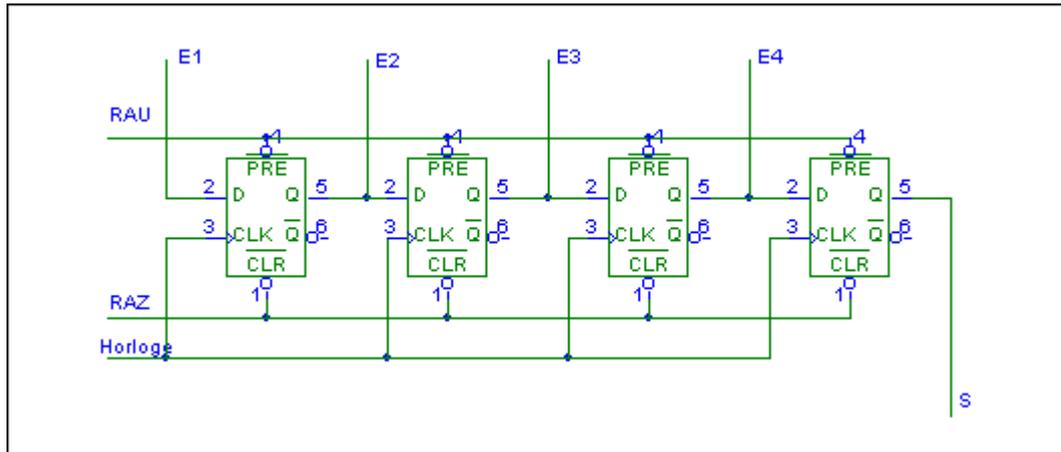
Les registres à entrée série, sorties parallèles

Ce registre possède une entrée E et plusieurs sorties (S1, S2, S3, S4). Les données binaires en entrées sont introduites les unes après les autres. Les sorties sont toutes disponibles en même temps. Ces registres peuvent être utilisés pour faire une transformation série parallèle des informations.



Les registres à entrées parallèles, sortie série

Toutes les entrées (E1, E2, E3, E4) sont introduites en même temps dans le registre. Les informations en sortie sur S sont disponibles les unes après les autres au rythme de l'horloge. Ces registres peuvent être utilisés pour faire une transformation parallèle série des informations.

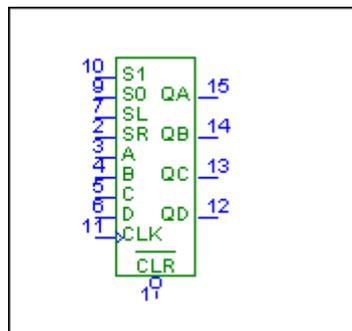


1.3 Registres universels

Ce circuit intégré regroupe les quatre types de registre précédents. Il permet les modes de fonctionnement suivant:

- Chargement et lecture parallèles
- Chargement série et décalage à droite ou à gauche lecture série ou parallèle
- Chargement parallèle et décalage à droite ou à gauche lecture série ou parallèle.

Le circuit suivant (74194) est un exemple de registre universel



Les entrées A, B, C, D sont les entrées parallèles. Les entrées SL et SR sont respectivement les entrées séries gauche et droite. Les entrées S1 et S0 permettent de choisir le mode de fonctionnement de ce registre (blocage, décalage à droite, décalage à gauche, chargement parallèle). L'entrée \overline{CLR} (active sur niveau bas) permet une remise à zéro asynchrone des sorties. L'entrée CLK est l'entrée horloge de synchronisation. Les sorties sont QA, QB, QC, QD.

2 Les compteurs

Il est possible de connecter des bascules pour effectuer des opérations de comptage. Le nombre de bascules utilisées et la façon de les interconnecter déterminent le nombre d'états du compteur.

L'état du compteur est défini par le nombre binaire formé par l'ensemble des sorties des bascules.

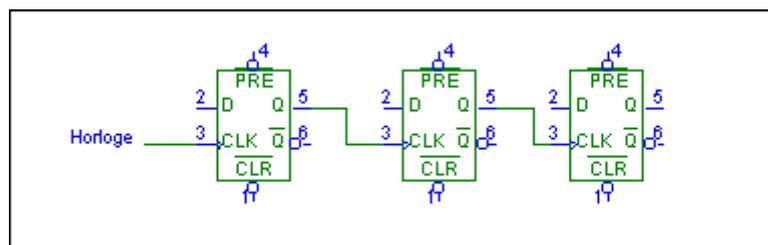
Les compteurs sont des éléments essentiels de la logique séquentielle car ils sont utilisés pour les systèmes arithmétiques mais ils permettent aussi d'établir une relation d'ordre de succession d'événements. Ils sont utilisés notamment pour le comptage d'événement, pour diviser la fréquence, dans les automates programmables,...

Les compteurs sont classés en deux catégories selon leur mode de fonctionnement. On distingue

- Les compteurs asynchrones ou compteurs séries

La première bascule est synchronisée par une horloge externe mais le déclenchement des autres bascules est déterminé par une combinaison logique des sorties des bascules précédentes. La propagation de l'ordre de changement d'état se fait en cascade. Les sorties des bascules ne changent pas d'état exactement en même temps car elles ne sont pas reliées au même signal d'horloge.

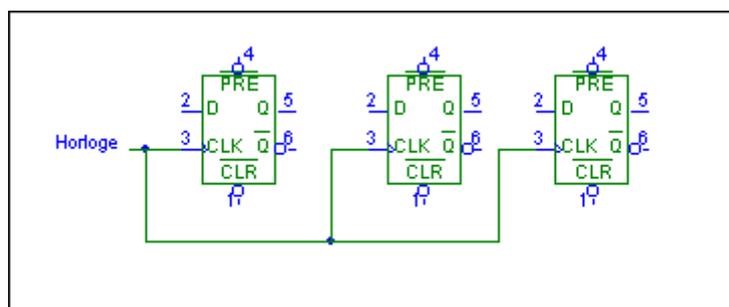
Exemple de connexion d'horloge pour obtenir un compteur asynchrone avec propagation de l'ordre de changement d'état en cascade.



- Les compteurs synchrones ou parallèles

Le signal d'horloge externe est connecté à toutes les bascules et permet de les déclencher toutes simultanément.

Exemple de connexion de l'horloge pour obtenir un compteur synchrone : la synchronisation des bascules est faite par le même signal d'horloge

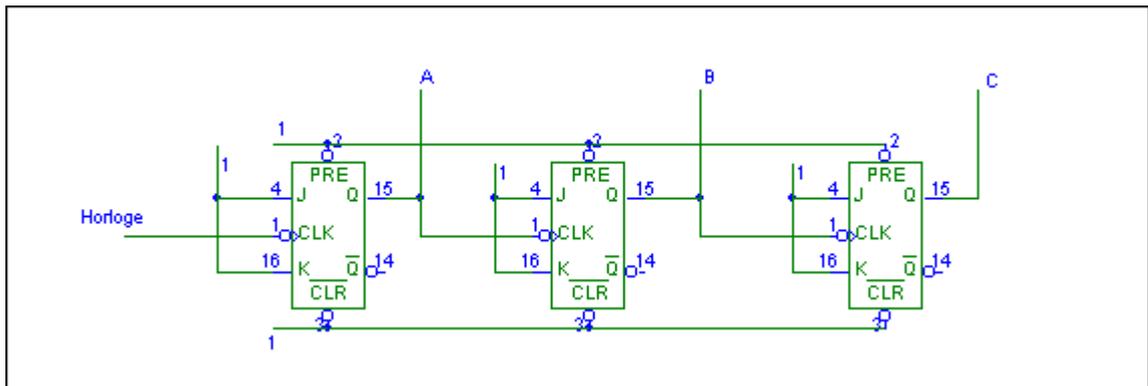


Pour chacune de ces catégories, les compteurs sont classés selon leurs séquences, le nombre d'état ou le nombre de bascules qu'ils comportent.

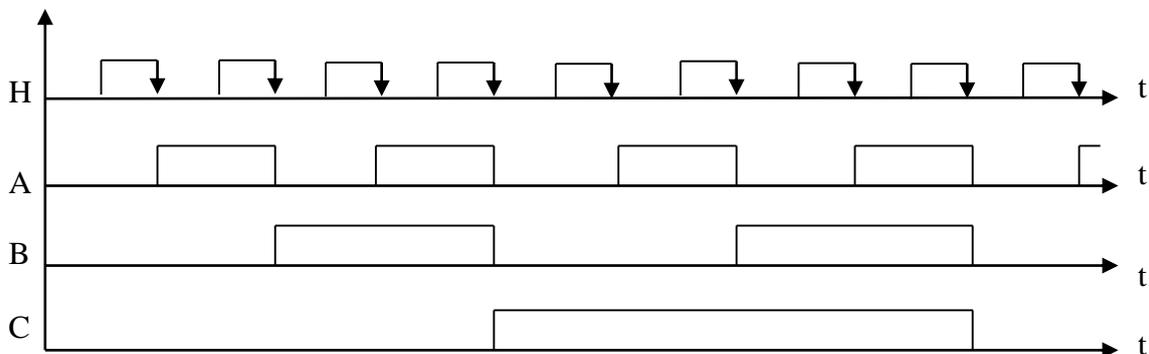
2.1 Compteurs asynchrones

Compteur binaire

Exemple d'un compteur à 3 bascules J K.



Les entrées asynchrones sont inactives car elles sont placées au niveau haut. Les entrées J et K sont à 1 donc les bascules changent d'état à chaque front actif de leur horloge (front descendant). La bascule A change d'état à chaque front descendant de l'horloge externe, la bascule B à chaque front descendant de A et la bascule C à chaque front descendant de B.



On affecte les poids 0,1,2 aux sorties A, B, C des bascules. On considère qu'au départ toutes les sorties sont à zéro.

On obtient le tableau suivant (il y a un changement d'état après chaque front descendant de l'horloge).

front actif de l'horloge	C	B	A	État
état initial	0	0	0	0
1	0	0	1	1
2	0	1	0	2

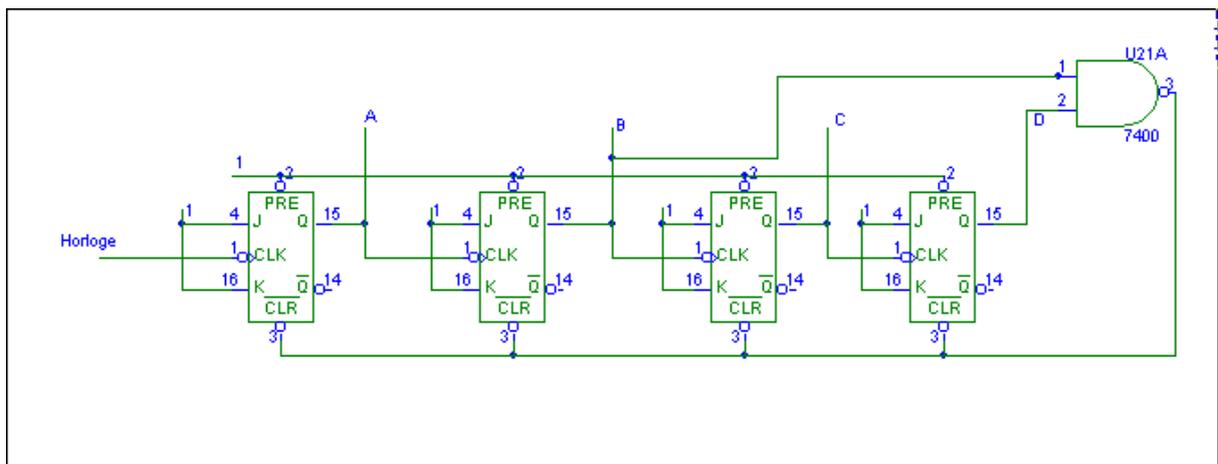
3	0	1	1	3
4	1	0	0	4
5	1	0	1	5
6	1	1	0	6
7	1	1	1	7
8	0	0	0	0

Compteur modulo N

On appelle compteur modulo N, un compteur qui compte de 0 à N-1. Le compteur le plus utilisé est le compteur modulo 10 ou compteur à décade ou compteur DCB (ou BCD). Il produit une séquence de 0 à 9.

Pour construire un tel compteur, on va prendre un compteur binaire et tronquer la séquence. On choisit donc la puissance de deux tout de suite supérieure $16 = 2^4$, il faut 4 bascules pour réaliser le compteur et on va utiliser l'entrée de remise à zéro asynchrone pour tronquer la séquence à 9. Le compteur comptera donc de 0000 à 1001. Le principe est identique pour réaliser n'importe quel compteur modulo N.

Schéma d'un compteur décimal à base de bascule J K :



Quand le compteur passe à l'état 1010_2 (10_{10}), c'est à dire quand $B = D = 1$, on provoque une remise à zéro asynchrone du compteur par l'intermédiaire d'un NAND car l'entrée \overline{CLR} est active au niveau bas.

Ce compteur reste dans l'état transitoire non souhaité 1010_2 (10_{10}) pendant le temps de traversée de la porte NAND (quelques nano secondes), il est ensuite remis à zéro. Pour fonctionner correctement les temps de propagation de toutes les portes doivent être identiques.

Inconvénients et avantages des compteurs asynchrones :

Les compteurs asynchrones sont assez lents car les temps de propagation de chaque bascule s'ajoutent.

La propagation des signaux de déclenchement des bascules provoque des états transitoires qui sont indésirables quand ils sont présents durant un temps non négligeable.

Par contre, la conception de ces compteurs est très simple et les liaisons entre les bascules sont peu nombreuses.

2.2 Compteurs synchrones

Les compteurs synchrones permettent d'éliminer les problèmes temporels et les états transitoires et de rendre possible l'exécution d'un cycle quelconque. Toutes les bascules sont connectées au même signal d'horloge.

2.2.1 Exemple d'un compteur binaire de 3 éléments binaires

On cherche à construire un compteur à l'aide de bascules JK à partir de la table de vérité suivante:

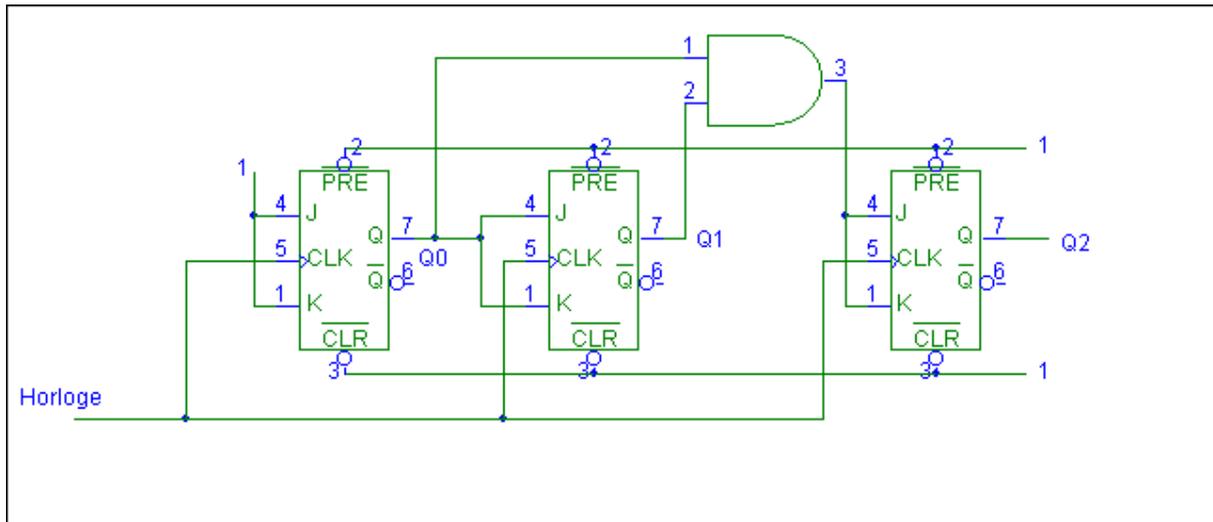
front actif de l'horloge	Q2	Q1	Q0	État
état initial	0	0	0	0
1	0	0	1	1
2	0	1	0	2
3	0	1	1	3
4	1	0	0	4
5	1	0	1	5
6	1	1	0	6
7	1	1	1	7
0	0	0	0	0

La bascule Q0 change d'état à chaque front d'horloge : on posera $J_0=K_0=1$

La bascule Q1 change d'état après chaque fois que $Q_0=1$: on posera $J_1=K_1=Q_0$

La bascule Q2 change d'état après chaque fois que $Q_1=Q_0=1$: on posera $J_2=K_2=Q_0.Q_1$

Le schéma du circuit est donc le suivant :



2.2.2 Compteur à décade ou DCB synchrone

De la même façon, nous pouvons construire à partir de la table de vérité un compteur modulo 10.

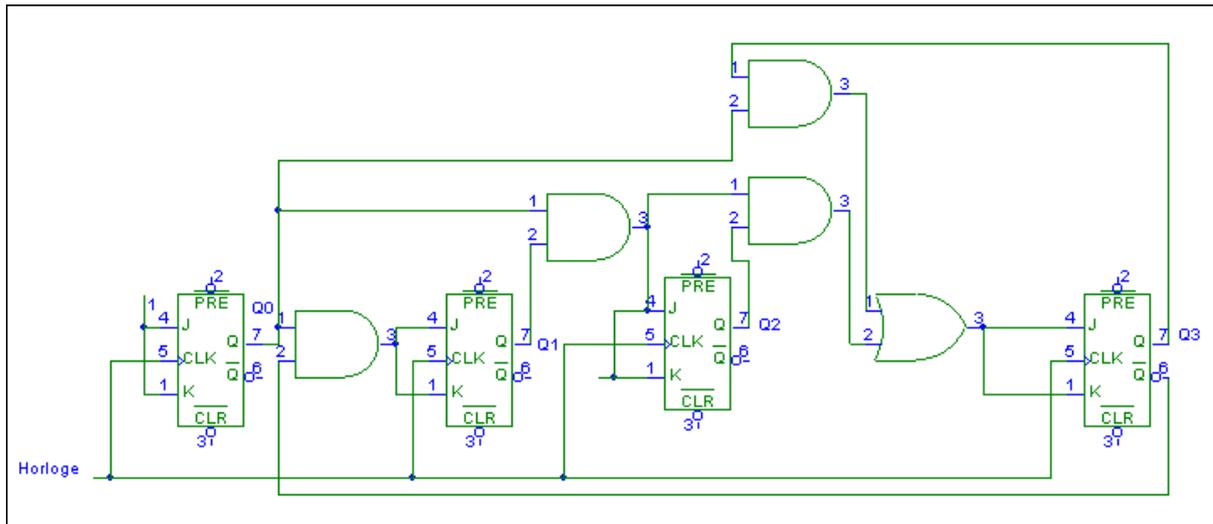
front actif de l'horloge	Q3	Q2	Q1	Q0	État
état initial	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	2
3	0	0	1	1	3
4	0	1	0	0	4
5	0	1	0	1	5
6	0	1	1	0	6
7	0	1	1	1	7
8	1	0	0	0	8
9	1	0	0	1	9

La bascule Q0 change d'état à chaque front actif de l'horloge.

La bascule Q1 change d'état quand $Q3=0$ et $Q0=1$.

La bascule Q2 change d'état quand $Q1 = Q0 = 1$.

La bascule Q3 change d'état quand $Q0=Q1=Q0=1$ ou quand $Q0=Q3=1$.

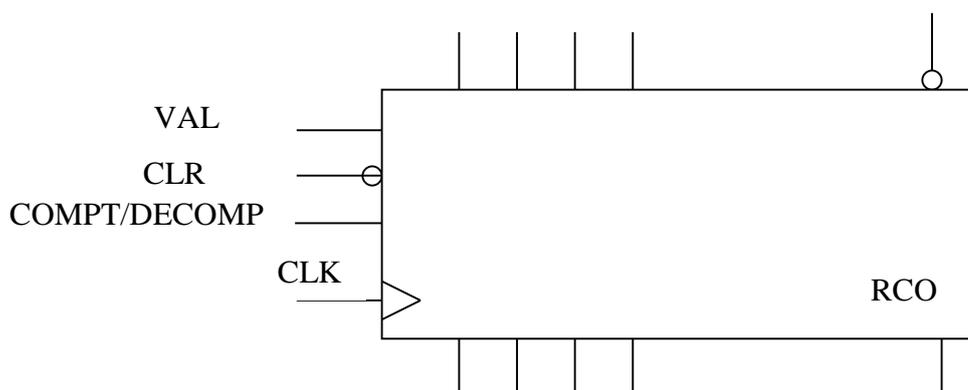


De la même façon, on peut réaliser n'importe quel compteur ou décompteur.

2.3 Compteurs intégrés

Il existe des circuits intégrés qui réalisent la fonction de compteur. Le circuit peut alors être vu comme une boîte noire qui possède en général les fonctions suivantes :

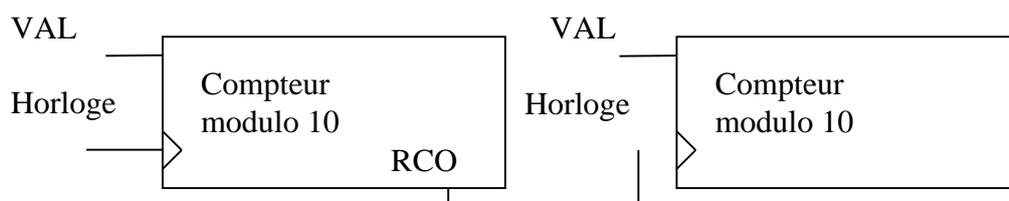
- Une entrée de remise à zéro asynchrone ou synchrone \overline{CLR} généralement active au niveau bas
- Une entrée chargement en général synchrone à une valeur particulière sur les entrées de chargement (A, B, C, D) commandée par une broche LOAD (généralement active sur niveau bas).
- Une commande de comptage ou de décomptage
- Une sortie de détection quand toutes les sorties valent 1 pour pouvoir propager une éventuelle retenue RCO (Ripple Carry Output)
- Une commande de validation ou d'interdiction de comptage VAL



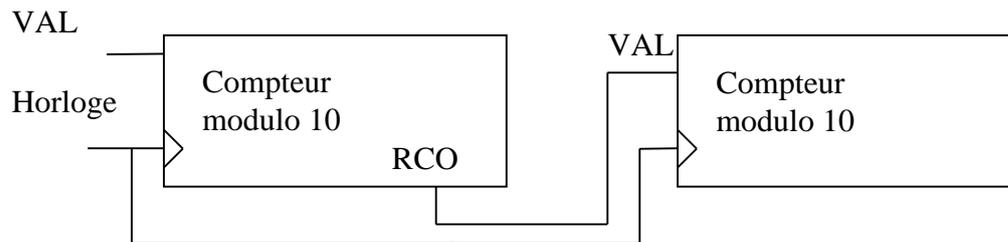
Grâce aux entrées de chargement, on peut obtenir des séquences tronquées.

On peut mettre des compteurs en cascade pour obtenir un modulo plus élevé. La sortie du dernier étage pilote une entrée du second compteur.

La mise en cascade peut être asynchrone en utilisant l'entrée horloge du second compteur.



La mise en cascade peut être synchrone en utilisant l'entrée de validation du second compteur.



La mise en cascade de ces deux compteurs permet de réaliser un compteur modulo 100.
On peut par exemple réaliser un chronomètre en mettant en cascade un compteur modulo 10 et un compteur modulo 6.

Chapitre 3 : Les mémoires

1 Généralités

On appelle mémoire tout dispositif capable d'enregistrer, de stocker et de restituer de l'information. Les opérations associées sont donc l'écriture, la mémorisation et la lecture. La mémorisation s'appuie sur des phénomènes physiques de nature électromagnétique, électronique, optique...

La mise en mémoire d'informations a été résolue au moyen de nombreuses méthodes, les principales actuellement sont : les mémoires magnétiques, les mémoires intégrées et les mémoires optiques.

Les principes de base de l'enregistrement d'un signal sur un support magnétique ont été exposés par Oberlin Smith en 1880. Ce principe a été repris par un physicien danois Valdemar Poulsen pour mettre au point un système d'enregistrement sur fil en 1898. Poulsen réalisera alors le premier répondeur-enregistreur téléphonique. Du magnétophone, on passera à la cassette magnétique, aux bandes magnétiques en informatique (1949), aux disques magnétiques (créé en 1956 par IBM), aux disquettes magnétiques (IBM 1970) et aux disques durs (1973 IBM). La découverte du laser a permis l'avènement des mémoires optiques sur disque, le CD audio a été commercialisé en 1982 par Philips et Sony. Le CD-ROM compact fait son apparition en 1985, puis le DVD et le DVD lecteur enregistreur.

Les mémoires totalement électroniques ont permis l'explosion de l'informatique. L'apparition des circuits intégrés en 1960 a permis le développement des mémoires intégrées. En 1966, IBM produit en technologie bipolaire une première mémoire RAM de 16 bits. La première ROM est due à Intel en 1971. La première RAM statique a été produite en 1971 (Intel), la première RAM dynamique est commercialisée en 1972 (Hewlett-Packard).

1.1 Caractéristiques générales

Les mémoires stockent des données de plusieurs éléments binaires. L'élément binaire est le plus petit élément de stockage, il est souvent appelé point mémoire. Généralement les données sont traitées par unité de huit bits (octet) mais une unité complète d'information s'appelle un mot il est constitué de un ou plusieurs octets (16, 32, 64 bits).

Adresse mémoire : c'est l'emplacement d'une unité de donnée.

Capacité ou taille de la mémoire: elle correspond à la quantité d'informations qu'elle peut contenir. On peut l'exprimer en bits, en octets ou en mots.

1 Kilo octet : noté Ko = 2^{10} octets = 1024 octets

1 Méga octets : noté Mo = 2^{20} octets

1 Giga octets : noté Go = 2^{30} octets

1 Téra octets : noté To = 2^{40} octets

Une mémoire de 512 Kmots de 32 bits est constitué de :

$$512 \times 1024 = 2^{19} \text{ mots de 32 bits}$$

$$512 \times 1024 \times 4 = 2^{21} \text{ octets}$$

$$512 \times 1024 \times 32 = 2^{24} \text{ bits}$$

Temps d'accès : c'est le temps qui s'écoule entre le lancement d'une opération d'accès (lecture ou écriture) et son accomplissement. Il se mesure en nano secondes pour les mémoires électroniques et en milli secondes pour les mémoires magnétiques. La fréquence est l'inverse du temps d'accès.

Bande passante : C'est le produit de la largeur du bus de données et de la fréquence mémoire. Pour une mémoire qui fonctionne à 100MHz qui a 64 bits de largeur de bus de données, la bande passante est de 800Mo/s. Ce critère permet d'évaluer les différents débits à synchroniser entre les modules de l'ordinateur.

Volatilité : Elle caractérise la permanence des informations dans une mémoire. Une mémoire volatile a besoin d'une énergie extérieure pour conserver l'information. Elle perd son contenu si l'on coupe le courant d'alimentation. Une mémoire centrale à semi-conducteurs peut être volatile, une mémoire magnétique ne l'est pas.

Les différents types d'accès aux mémoires:

Accès séquentiel : C'est le plus lent. Pour accéder à une information, on est obligé de parcourir toutes celles qui la précèdent.

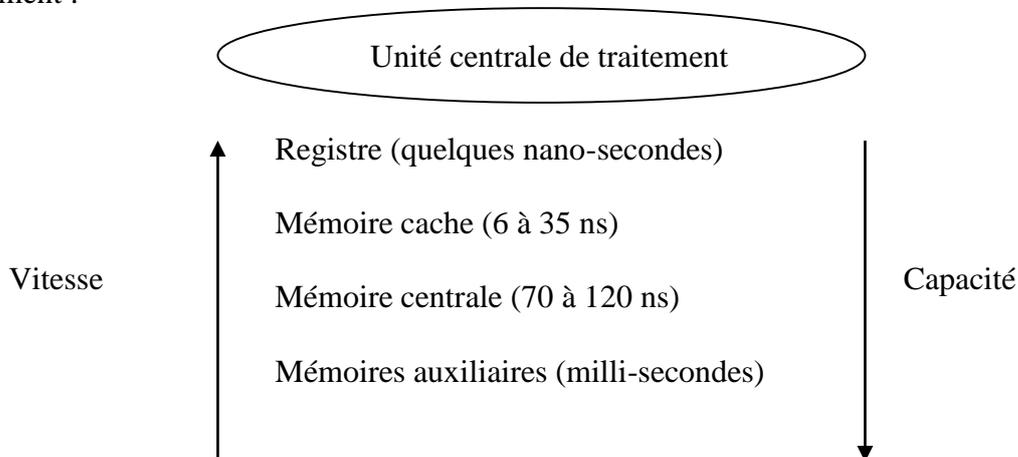
Accès direct : on accède directement à l'information par l'adresse mémoire.

Accès semi-séquentiel : c'est une combinaison des deux précédentes. Par exemple pour un disque magnétique, l'accès au cylindre est direct, l'accès au secteur est séquentiel.

Accès par contenu : Les informations sont identifiées par clé et la recherche s'effectue sur cette clé simultanément sur toutes les positions de la mémoire (mémoire cache).

1.2 Hiérarchie des mémoires

Les différentes mémoires sont classées en fonction de leur temps d'accès, de la capacité et du coût par élément binaire. On peut classer les mémoires en fonction de l'unité centrale de traitement :



Plus on se rapproche de l'unité centrale de traitement, plus le temps d'accès aux informations en mémoire est court mais la capacité de stockage diminue et le coût par élément binaire devient plus élevé.

Comme nous l'avons vu au chapitre précédent, dans l'unité centrale de traitement, les registres servent de stockage des opérandes et des résultats intermédiaires.

La mémoire cache est une mémoire rapide de faible capacité utilisée comme mémoire tampon entre l'unité centrale de traitement et la mémoire centrale. Cette mémoire permet au CPU de faire moins d'accès à la mémoire centrale (Voir Cours d'Architecture des Ordinateurs de Nicolas Flasque)

La mémoire centrale est l'élément principal de rangement des informations utilisées par l'unité centrale de traitement. Pour exécuter un programme, il faut charger les instructions et les données en mémoire centrale. Cette mémoire est à semi-conducteurs.

Les mémoires auxiliaires ou mémoire de masse sont des mémoires périphériques de grande capacité et de coût relativement faible utilisées pour le stockage permanent des données. Elles utilisent des supports magnétiques, magnéto-optiques ou optiques.

Il peut exister dans les gros systèmes une mémoire intermédiaire entre la mémoire centrale et les mémoires auxiliaires appelé mémoire d'appui.

Dans la suite ce chapitre, nous limiterons notre étude aux mémoires électroniques.

2 Mémoire centrale : mémoire à semi-conducteurs

On distingue deux classes de mémoires à semi-conducteurs :

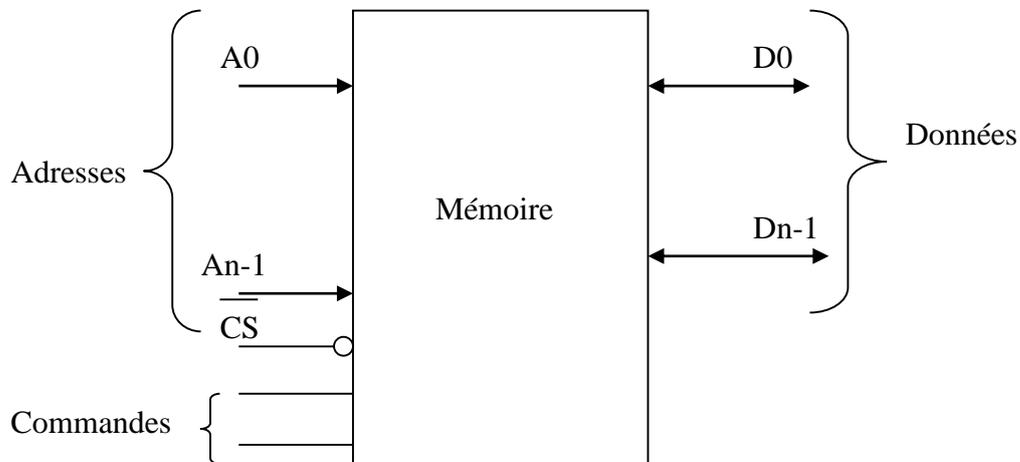
Les **mémoires vives** (RAM Random Access Memory) qui sont des mémoires volatiles. Elles ne peuvent conserver pas leurs données quand elles ne sont plus alimentées. Elles sont accessibles en lecture et écriture.

Les **mémoires mortes** (ROM Read Only Memory) qui conservent les informations même en l'absence d'alimentation électrique. Généralement quand elles sont alimentées, elles ne peuvent être que lues. L'écriture appelée aussi programmation, se fait à l'aide d'un dispositif spécial appelé programmeur.

La plupart des ordinateurs possèdent ces deux types de mémoire. En général la ROM est utilisée pour stocker un programme qui est exécuté automatiquement par l'ordinateur comme le programme d'amorçage ou de démarrage. La RAM est utilisée par exemple pour stocker les programmes, les données.

2.1 Fonctionnement d'une mémoire

Il existe plusieurs façons de réaliser une mémoire d'une capacité donnée, la figure suivante présente les principales broches d'une mémoire simple.



Les broches A0 à An-1 correspondent aux adresses de la mémoire. Pour identifier les cases mémoires, on leur attribue un numéro appelé adresse. Cette adresse s'obtient par la combinaison linéaire d'un ensemble de fils constituant le bus d'adresse. Ce bus est unidirectionnel, les informations sont envoyées du microprocesseur vers la mémoire.

Les données, stockées à l'intérieur des cases mémoires, sont accessibles par ensemble de fils appelé bus de données. Ces données peuvent être lues ou écrites, dans le cas d'une mémoire vive, aussi ce bus est bidirectionnel.

Des commandes doivent être échangées entre le microprocesseur et la mémoire pour déterminer le type d'opération (lecture ou écriture), pour sélectionner la mémoire ou pour synchroniser le microprocesseur et la mémoire. Le signal de synchronisation READY est envoyé par la mémoire au microprocesseur.

Les mémoires comme la plupart des circuits électroniques sont munies d'un signal de sélection de boîtier (CS chip select ou CE chip enable) en général actif sur l'état bas. Cette entrée permet de déconnecter le bus de donnée quand le composant n'est pas utilisé pour éviter les conflits lorsque plusieurs mémoires sont connectées au même bus de données.

Lecture d'une mémoire

Pour lire des données en mémoire, il faut que le processeur place l'adresse de la donnée sur le bus d'adresse et que les informations qui doivent être lues soient placées par la mémoire sur le bus de données. Une commande de lecture doit être envoyée par le microprocesseur (R Read ou OE Output Enable), elle est généralement active sur niveau bas. La lecture ne peut avoir lieu que si le circuit est sélectionné (Voir le schéma général d'une ROM).

Écriture d'une mémoire

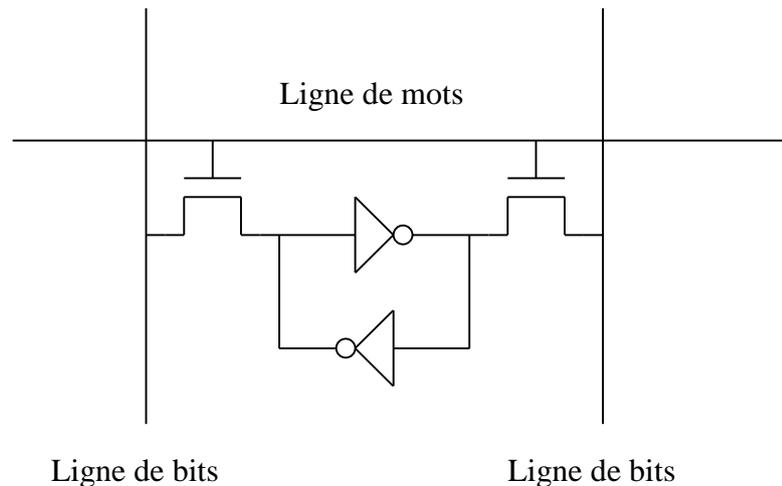
Pour écrire une donnée, le processeur dépose l'adresse de la donnée sur le bus d'adresse, il dépose la donnée sur le bus de données, et envoie une commande d'écriture. Une commande d'écriture est envoyée par le processeur à la mémoire (W Write ou WE Write Enable), elle est généralement active sur niveau bas.

2.2 Les mémoires vives

Elles sont également appelées RAM Random Access Memory ou mémoire à accès aléatoire. On peut accéder à n'importe quelle case mémoire en un temps donné. Ils existent deux catégories de mémoires vives : les mémoires vives statiques et les mémoires vives dynamiques.

Les mémoires vives statiques ou SRAM (Static RAM)

Elles emploient des bascules comme éléments de stockage et peuvent conserver les données tant qu'elles sont alimentées en tension. Elles sont rapides (6 à 15 nanosecondes) et consommatrice d'énergie. Le schéma de principe d'une cellule de mémoire RAM peut être le suivant:



Le point mémoire est constitué de deux inverseurs montés tête bêche et de deux transistors se comportant comme des interrupteurs. Les transistors sont passants quand la ligne de mot est active, sinon ils sont bloqués.

Pour écrire une valeur, on rend passante la ligne de mot et on propage la valeur adéquate sur la ligne de bits et la ligne de bits inversés.

Pour lire une valeur, on rend passante la ligne de mot, les deux transistors fonctionnent en interrupteur fermé et permettent de connecter les sorties des inverseurs à la ligne de bit et la ligne de bits renversés.

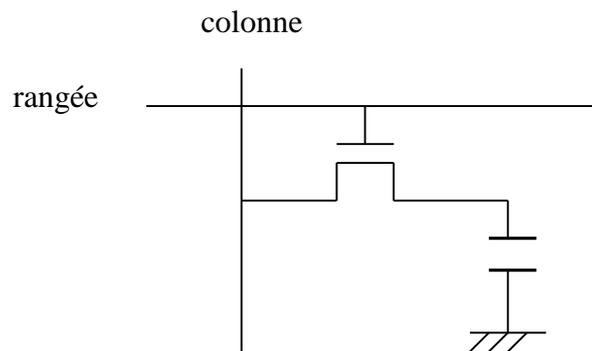
Une fois qu'une valeur binaire est mémorisée, elle est conservée tant que le circuit est alimenté car l'entrée de chaque inverseur est opposée à l'autre et les transistors sont bloqués (interrupteur ouvert).

On distingue les SRAM asynchrones qui fonctionnent indépendamment du signal d'horloge et les SRAM synchrones qui sont synchronisés par le signal d'horloge (qui cadence à la fois la SRAM et le microprocesseur). Les mémoires SRAM synchrones peuvent comporter normalement une fonction de transfert en rafales qui permet à la mémoire d'écrire ou de lire plusieurs emplacements différents simultanément en n'utilisant qu'une seule adresse.

Une des applications importantes des mémoires SRAM est de servir de mémoire cache dans les ordinateurs.

Les mémoires vives dynamiques ou DRAM (Dynamic RAM)

Le point mémoire est formé par un simple condensateur comme l'indique le schéma suivant.



Lorsque la ligne rangée est active, le transistor conduit, il se comporte comme un interrupteur fermé, le condensateur est alors connecté à la ligne colonne. La valeur stockée dans le condensateur peut être lue à ses bornes ou écrite en plaçant une nouvelle tension.

Si la ligne rangée est au niveau bas, le transistor se comporte comme un interrupteur ouvert, la valeur est conservée par le condensateur.

La cellule est très simple, petite et de faible coût. On peut ainsi construire des grands tableaux de mémoire, la densité d'intégration est beaucoup plus importante que pour les SRAM. L'inconvénient de ce type de mémoire est que le condensateur ne peut conserver la charge à ses bornes que très peu de temps (à cause des courants de fuite). Peu à peu, le condensateur se décharge. On doit procéder régulièrement à une relecture et réécriture des informations. La mémoire doit être **rafraîchie** à intervalles réguliers. Cette caractéristique complexifie la réalisation de telles mémoires.

Le temps d'accès (60 à 70 nanosecondes) pour ce type de mémoire est plus important que pour les mémoires SRAM, les mémoires DRAM sont principalement utilisées dans les mémoires centrales des ordinateurs.

Il existe différents types de DRAM, la description des différents modes de fonctionnement de ces mémoires dépasse le cadre de ce cours, nous nous contenterons ici de seulement de citer quelques types de mémoires:

Les mémoires DRAM EDO : Extended Data Output ou sortie de mémoire étendue

Les mémoires DRAM BEDO : Burst Extended Data Output ou sorties de données étendues en rafales

Les mémoires SDRAM ou DRAM synchrone qui utilisent aussi les transferts en rafale.

Les mémoires DDR SDRAM double le transfert des SDRAM

Les mémoires RDRAM, Rambus DRAM, dix fois plus rapide que les DRAM classique et les mémoires DRDRAM direct Rambus DRAM qui pourraient atteindre des débits de 1,6 Go/s.

Les mémoires VRAM, WRAM ou SGRAM servent pour la gestion de l'affichage.

Ces mémoires sont en forte évolution au niveau des performances et des coûts.

2.3 Les mémoires mortes

Les mémoires mortes où ROM (Read Only Memory) ne peuvent être que lues, l'écriture dans ce type de mémoire s'appelle programmation.

Il existe plusieurs types de mémoires mortes qui diffèrent par :

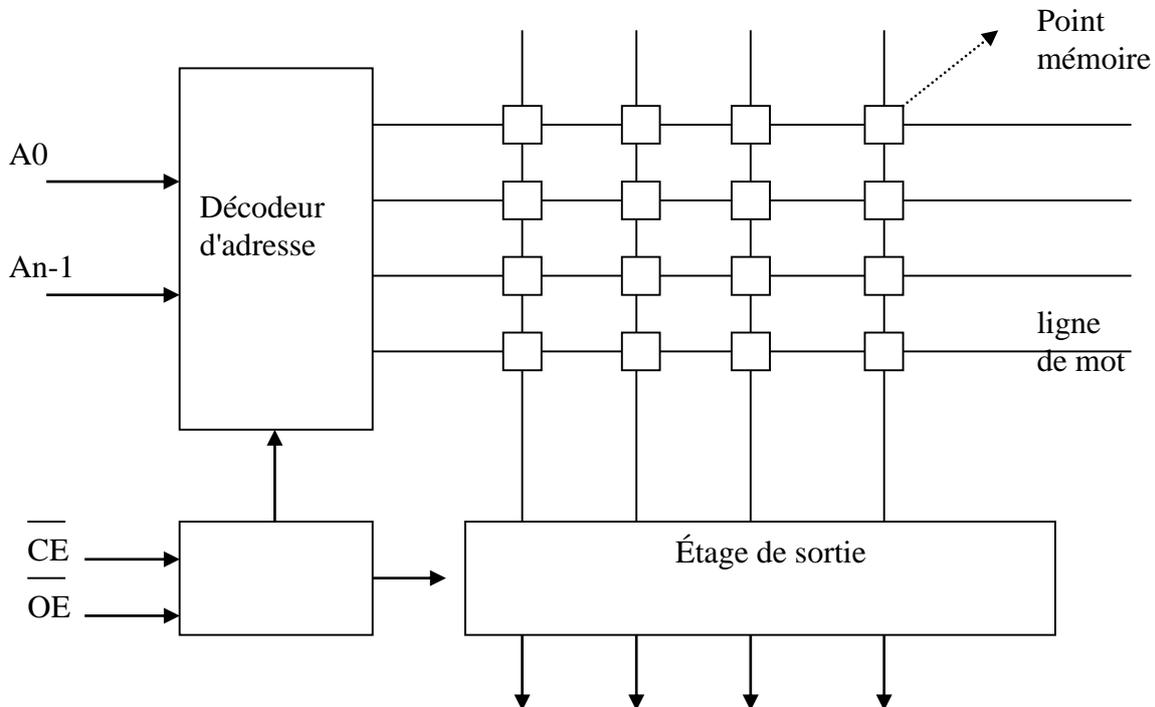
- leur structure interne
- le nombre de reprogrammations possibles
- le mode d'effacement.

On distingue :

- Les mémoires **ROM** qui sont programmables par le constructeur à la fabrication.
- Les mémoires **PROM** (Programmable ROM), elles sont programmables une seule fois par l'utilisateur.
- Les mémoires **EPROM** (Erasable PROM), elles sont programmables plusieurs fois et effaçables par exposition aux rayons ultraviolets à travers une fenêtre placée au-dessus du boîtier. L'effacement est global.
- Les mémoires **EEPROM** (Electrically EPROM), elles sont programmables plusieurs fois et effaçables électriquement, l'effacement peut se faire case par case.
- Les mémoires **Flash EPROM** (EPROM rapide en programmation), elles sont également programmables plusieurs fois et effaçable électriquement. La différence avec les EPROM est dans la programmation et le mode d'effacement qui peut se faire globalement ou par bloc de cases.

Technologie des mémoires mortes

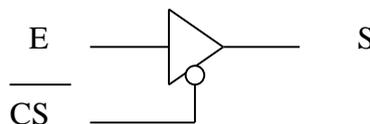
L'organisation générale d'une mémoire morte peut être la suivante :



Sur ce schéma, il est à noter que la sélection est linéaire, le décodeur choisit une de ses sorties en fonction de l'adresse qu'il reçoit. La sortie active alors une ligne complète de cellule qui correspond à un mot. Ce mode de sélection est utilisé pour les mémoires de faible capacité. Il existe d'autres modes de sélection, notamment la sélection à deux dimensions où les cellules sont adressées par lignes et par colonnes (il y a un décodeur pour sélectionner les lignes et un décodeur pour sélectionner les colonnes), un élément binaire est sélectionné (Voir cours L1 pour le fonctionnement d'un décodeur).

CE est une entrée de sélection du boîtier, OE est une entrée permettant la lecture des données.

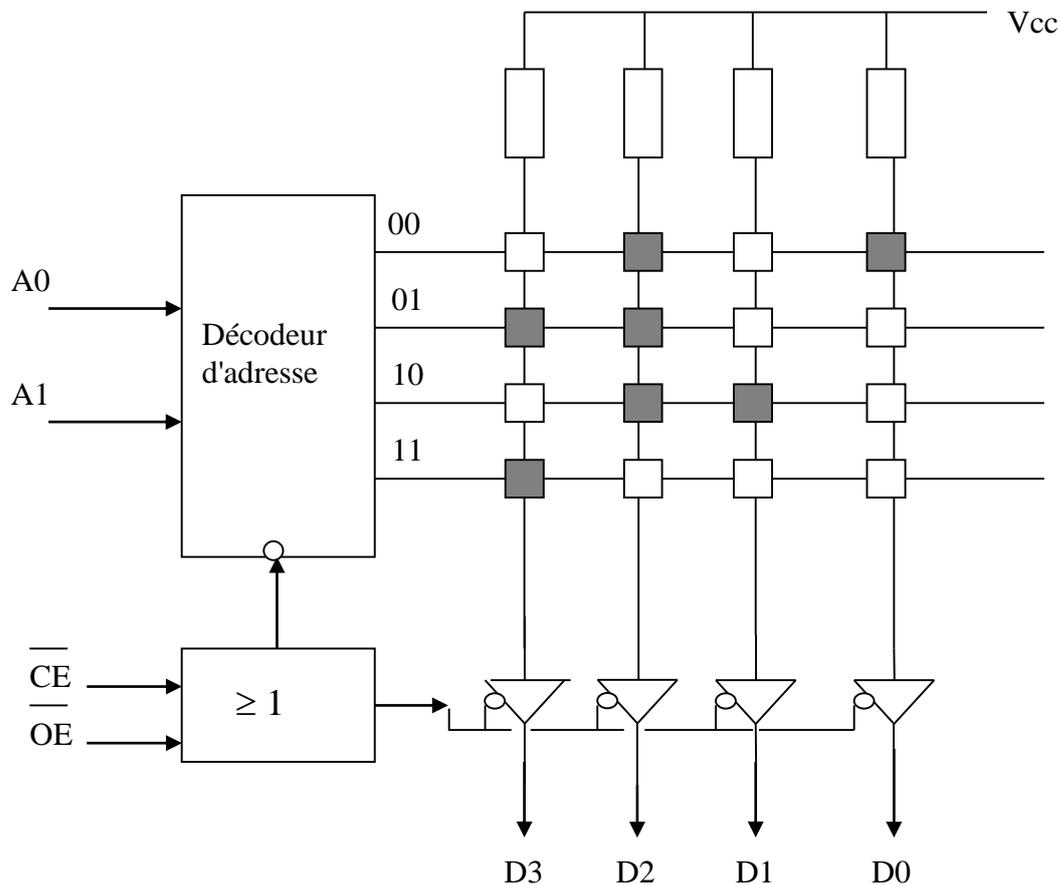
L'étage de sortie est constitué d'une porte 3 états (niveau haut, niveau bas, niveau haute impédance)



Si l'entrée de sélection (CS) est active (niveau bas) : $S = E$, la sortie est amplifiée par rapport au signal d'entrée. Si l'entrée de sélection (CS) est inactive (niveau haut), la sortie est mise dans l'état haute impédance : le circuit est déconnecté électriquement de la sortie. Cet état permet d'éviter d'éventuels conflits entre différents circuits quand plusieurs circuits sont raccordés entre eux.

Le point mémoire est à 1 ou à 0 selon la valeur que l'on souhaite stocker dans la mémoire.

Exemple de fonctionnement :



Un point mémoire grisé représente un 1 logique, un point mémoire en blanc représente un 0 logique. Une des entrées CE et OE doit être à zéro pour lire la mémoire sinon les sorties sont dans l'état haute impédance donc déconnectées électriquement du reste du circuit et le décodeur d'adresse n'est pas sélectionné.

Le tableau suivant résume les données stockées (D3, D2, D1, D0) en mémoire en fonction des entrées adresses A1, A0.

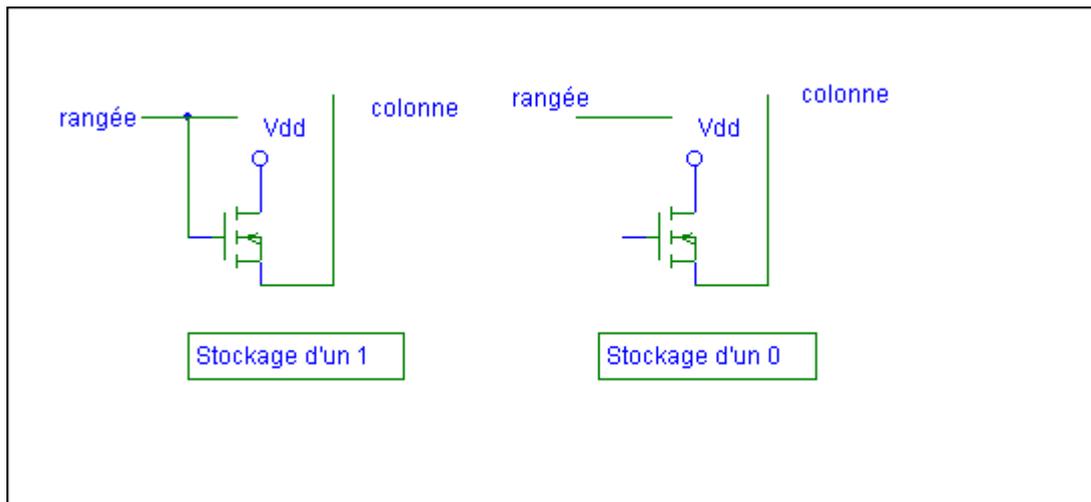
A1	A0	D3	D2	D1	D0
0	0	0	1	0	1
0	1	1	1	0	0
1	0	0	1	1	0
1	1	1	0	0	0

La technologie des points mémoires est différente selon le type de mémoire utilisée.

Dans les mémoires **ROM**, la plupart des circuits intégrés utilisent le principe de l'absence ou de la présence d'une connexion de transistors à une jonction de rangée/ colonne pour représenter un 1 ou un 0.

Exemple de fonctionnement:

Lorsque la ligne de rangée est connectée à la grille du transistor, le transistor est conducteur, la ligne de colonne associé est à 1. Quand la grille du transistor n'est pas connectée à la rangée le transistor n'est pas passant, les lignes de colonnes restent au niveau bas.



Dans les mémoires **PROM**, on utilise un procédé de fonte de fusibles pour stocker les éléments binaires. Le fusible est laissé intact ou fondu pour représenter un 1 ou un 0. Le procédé de fonte de fusible est irréversible. La programmation faite par l'utilisateur à l'aide d'un programmeur de PROM consiste à envoyer un courant suffisamment élevé est injecté pour faire fondre le fusible quand un 0 est nécessaire. Les mémoires PROM sont utilisées dans le cas de petites séries.

Dans les mémoires **EPROM et EEPROM**, on utilise un tableau d'éléments de NMOSFET muni d'une structure à grilles isolées. Les grilles isolées des transistors n'ont aucune connexion électrique et peuvent stocker des charges de manière permanente. La présence ou l'absence d'une charge à la grille détermine la valeur 1 ou 0 des éléments binaires. Pour les EPROM, l'effacement des données se fait par des rayons ultraviolets de forte intensité à travers la fenêtre de quartz placée sur le dessus de la puce. Les EEPROM s'effacent et se programment avec des impulsions électriques.

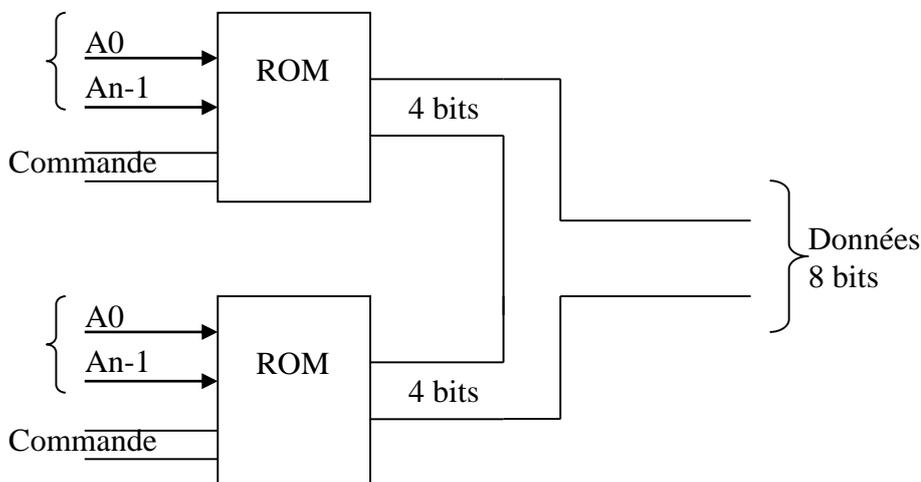
Les mémoires Flash sont des composants situés entre les ROM et les RAM. Ces mémoires peuvent être lues et écrites à volonté. Elle nécessite un seul transistor par point mémoire au lieu de deux pour les EEPROM classiques. Elle peut remplacer les disques durs des ordinateurs portables. Dans les appareils de photo numériques, les images captées sont enregistrées dans des mémoires flash.

2.4 Constitution des blocs mémoires

On peut réaliser des mémoires de grande capacité à partir de circuits mémoires élémentaires. L'extension de la mémoire s'effectue en ajoutant des puces de mémoires aux bus d'adresse, de donnée et de commande.

Extension de la longueur du mot

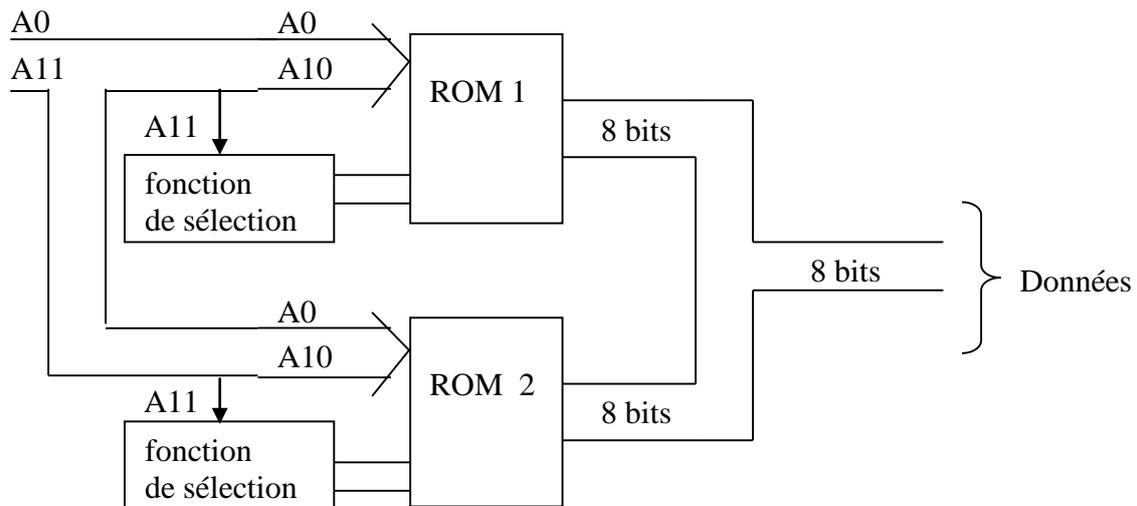
Pour augmenter la longueur des mots mémoires, il faut augmenter le nombre de bits du bus de données. Dans l'exemple suivant, on réalise à partir de deux mémoires de 4 bits de données, une mémoire de 8 bits de données. Les adresses (A_0 , A_{n-1}) et les éléments binaires de commande des deux mémoires sont identiques.



Pour les mémoires RAM, le bus de données est bidirectionnel puisque les données peuvent être lues et écrites. Les fils doivent donc pouvoir être connectés en entrée ou en sortie, on utilise des tampons à trois états, ces circuits à trois états sont en général présents dans les RAM.

Extension de la capacité mémoire

Pour augmenter la capacité de la mémoire, il faut augmenter le nombre d'emplacements mémoire, donc augmenter le nombre de bits d'adresses. Dans l'exemple suivant, nous disposons de deux mémoires de 2 kilooctets et nous cherchons à réaliser une mémoire de 4 kilooctets. Les sorties de données de chaque circuit sont connectées entre elles, les mémoires ne doivent évidemment pas être lues en même temps. Pour les RAM comme dans le paragraphe précédent, on utilise des tampons à trois états. Les éléments binaires du bus d'adresse de même poids sont connectés entre eux ($A_0 \dots A_{10}$). Les autres éléments binaires de l'adresse sont connectés à une entrée de commande (l'entrée de sélection du boîtier ou l'entrée lecture). Les deux boîtiers ne doivent pas être actifs en même temps, ainsi la mémoire ROM1 est par exemple active pour $A_{11}=1$ et ROM2 pour $A_{11}=0$.



2.5 Décodage d'adresse

On souhaite implanter des blocs élémentaires (périphériques ou des mémoires) dans un système à microprocesseur. On dispose d'un bus d'adresse qui définit l'espace adressable par le microprocesseur et on connaît les zones d'occupation des circuits périphériques.

Prenons l'exemple d'un microprocesseur de 128 K octets. On souhaite implanter trois blocs de mémoire morte aux emplacements suivants :

M1 est une mémoire de 4 K octets implantée dans la zone de \$0000 à \$0FFF

M2 est une mémoire de 8 K octets implantée dans la zone de \$1000 à \$2FFF

M3 est une mémoire de 16 K octets implantée dans la zone de \$3000 à \$6FFF

Le symbole \$ exprime des valeurs en base 16.

Le microprocesseur dispose d'un signal de commande \bar{R} pour la lecture des données. Chaque bloc mémoire dispose de deux signaux de commande \bar{R} et \bar{C}_s actifs au niveau bas.

Le microprocesseur 128K octets ($2^{16} \times 8$) possède un bus d'adresse de 16 bits de A0 à A15 et un bus de donnée de huit bits de D0 à D7.

A15	A14	A13	A12	A11	A10	A9	A8	A0	
0	0	0	0	0	0	0	0	0	M1 \$0000 \$0FFF
0	0	0	0	1	1	1	1	1	
0	0	0	1	0	0	0	0	0	M2 \$1000 \$2FFF
0	0	1	0	1	1	1	1	1	
0	0	1	1	0	0	0	0	0	M3 \$3000 \$6FFF
0	1	1	0	1	1	1	1	1	

On peut écrire les équations de sélection des mémoires en choisissant les éléments binaires du

$$\overline{C_{S2}} = \overline{A15} \overline{A14} (\overline{A13} A12 + A13 \overline{A12})$$

$$\overline{C_{S1}} = \overline{A15} \overline{A14} \overline{A13} \overline{A12}$$

bus d'adresse qui ne varient pas. Les signaux de commandes des mémoires sont les suivants :

$$\overline{C_{S3}} = \overline{A15} (\overline{A14} A13 A12 + A14 \overline{A13} \overline{A12} + A14 \overline{A13} A12 + A14 A13 \overline{A12})$$

Le signal \overline{R} des mémoires doit être connecté au signal \overline{R} du microprocesseur. Le bus de donnée du microprocesseur est connecté aux entrées données de chaque mémoire. Les éléments binaires (A0 à A11) sont connectés du microprocesseur en parallèle sur chaque boîtier mémoire. On peut également utiliser un circuit décodeur pour effectuer le décodage d'adresse (Voir Travaux Dirigés).

Bibliographie :

- Electronique Numérique, logique séquentielle, Marcel Gindre, Denis Roux, McGraw-Hill.
- Electronique Numérique, Jean Claude Lafont et Jean Paul Vabre, ellipses.
- Architecture des machines et des systèmes informatiques, A Cazes et J Delacroix, Sciences Sup, Dunod.
- Informatique Industrielle, Circuits logiques programmables, Alexandre Nketsa, ellipses.
- Systèmes numériques, Concepts et applications, Thomas L.Floyd, Reynald Goulet.
- Une brève histoire de l'électronique, Henry Lilen, Vuibert.