Chapitre 5: Les mémoires

1. Introduction:

Une mémoire est un circuit permettant, de conserver et de restituer des informations. Les informations peuvent être écrites ou lues. Il y a écriture lorsqu'on enregistre des informations en mémoire, lecture lorsqu'on récupère des informations précédemment enregistrées.

2. Organisation d'une mémoire :

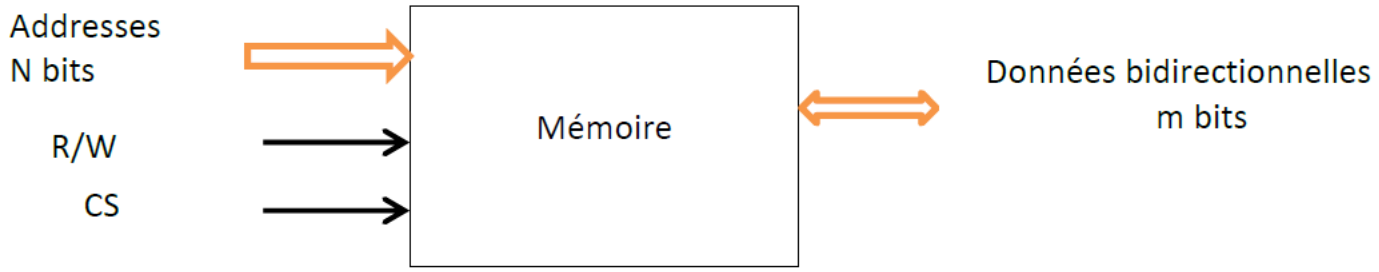
Une mémoire peut être représentée comme une armoire de rangement composée de différents tiroirs. Chaque tiroir représente alors une case mémoire qui peut contenir une donnée. Le nombre de cases mémoires est très élevé, on les identifie par des numéros appelés **adresses**.

Exemple :

Adresse	Case mémoire
000	00101100
001	----------
010	----------
011	----------
100	----------
101	----------
110	----------
111	----------

3. Communication de la mémoire avec son environnement :

La communication entre la mémoire et son environnement est établie par le biais de la ligne de contrôle, de lignes d'adresse et de lignes de données.



- Ligne de contrôle (R/W) détermine la direction de transfert de l'information binaire avec la mémoire (stocker ⇒ écriture ; extraire ⇒ lecture)
- Ligne de contrôle (CS) : permet de sélectionner la mémoire
- Ligne d'adresse : spécifie le mot choisi dans la mémoire.
- Les lignes de données bidirectionnelle (d'entrée/sortie)

4. Caractéristiques d'une mémoire :

a) Format :

Le format précise la répartition de bits stockés en nombre de mots et taille de mots.

Exemple :

Une mémoire de 8192 bits s'organise en 8192 mots d'un bit ou 1024 mots de 8 bits.

b) Capacité :

La capacité d'une mémoire est le nombre de bits qu'elle contient.

Unités de mesure :

- O : Octet= 8bits
- 1 K = 1 kilo = 2^{10}= 1024
- 1M= 1 Méga=2^{20}
- 1G = 1 Giga =2^{30}
- 1 T = 1 Tira =2^{40}

Exemple : 6 MO= $6x2^{20}$ Octets = $6x2^{20}$x 8bits

c) Adresse :

L'accès à un mot particulier se fait par une adresse préalablement définie.

d) Temps d'accès :

Temps qui s'écoule entre l'instant où a été lancée une opération de lecture/écriture et l'instant où la première information est disponible sur le bus de données.

5. Différents types de mémoire :

5.1 Les mémoires vives RAM (Random Access Memory)

C'est une mémoire vive qui sert au stockage temporaire de données. En générale, elle est volatile c.-à-d. elle perd ses informations en cas de coupure d'électricité.

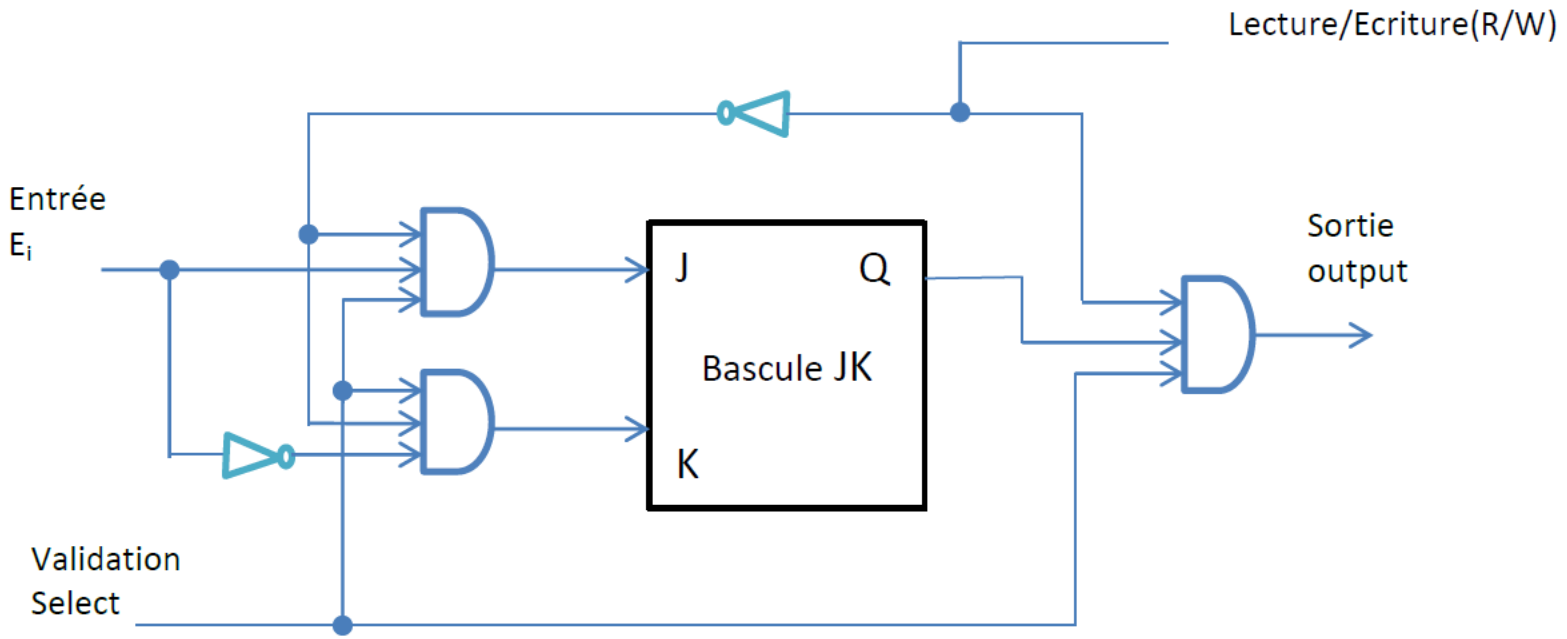
Il existe deux grandes familles de mémoires RAM :

- Les RAM statiques
- Les RAM dynamiques

a) Les RAM statiques :

Le bit mémoire d'une RAM statique (SRAM) est composé d'une bascule et un circuit logique pour la lecture ou l'écriture.

Le circuit logique d'une cellule de stockage binaire à base d'une bascule JK est donné comme suit :



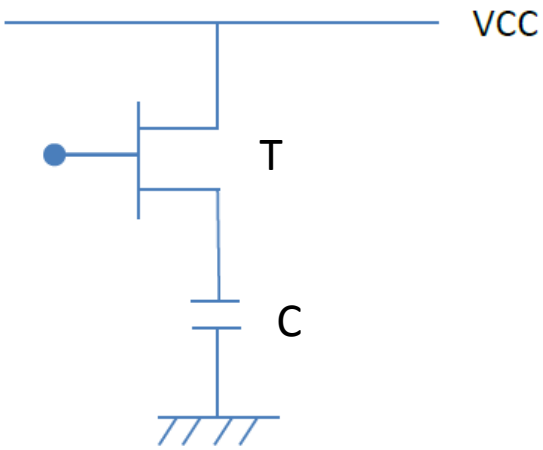
cellule de stockage binaire à base d'une bascule JK

Fonctionnement :

- Si select =1⇒ cellule sélectionnée et si R/W=1 ⇒ le bit mémorisée dans la bascule serait disponible à la sortie output.
- Si select=1 et R/W=0 alors le bit Ei disponible à l'entrée E serait stocké dans la bascule.
- Si select=0 alors les opérations de R/W ne sont pas permises dans cette cellule binaire.

b) Les RAM dynamiques DRAM :

Dans les RAMs dynamiques l'élément de mémorisation est constitué par un condensateur C et un transistor T. Le transistor joue le rôle d'un interrupteur commandé. L'information est mémorisée sous forme d'une charge électrique stockée dans le condensateur.



Avantage:

permet une plus grande densité d'intégration.

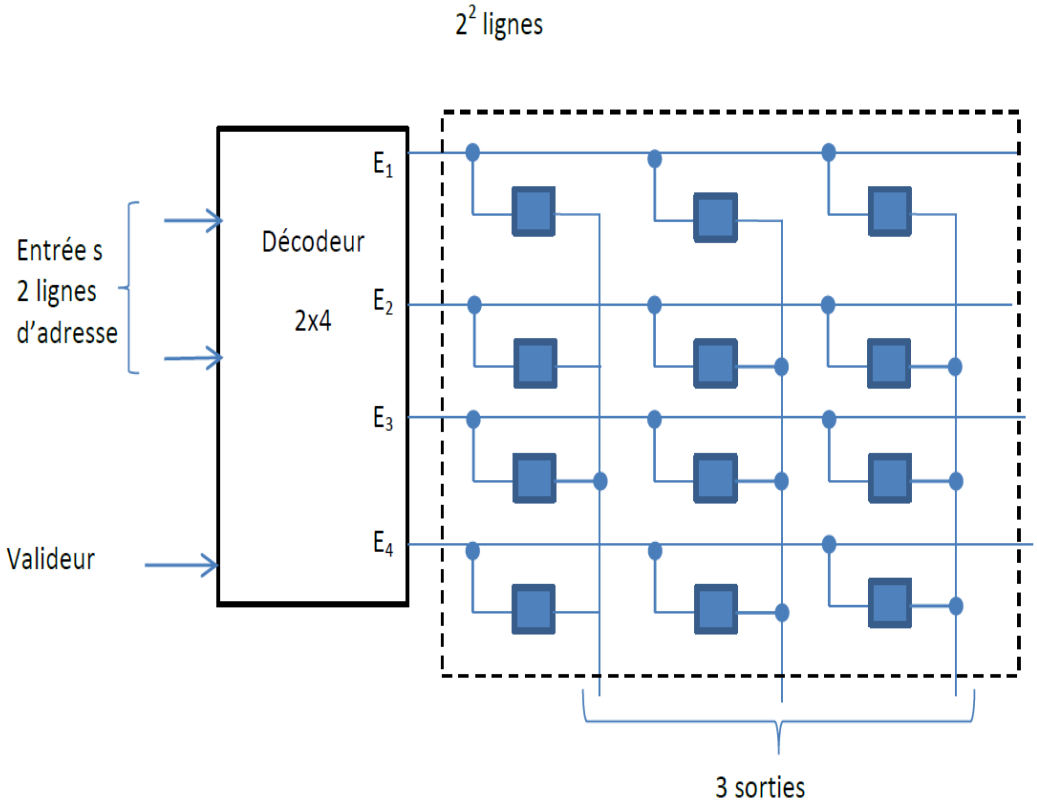
Inconvénient:

doit être rafraichie régulièrement pour entretenir la mémorisation à cause des courants de fuites au niveau des condensateurs qui ont tendance à se décharger.

5.1.1 Adressage mémoire :

À chaque case mémoire d'une RAM est associée une adresse. En adressant une case mémoire, la donnée qui s'y trouve peut être lue, comme on peut écrire dans cette case mémoire.

Exemple d'une RAM statique 4x3 (4mots, 3bits /mot). Donc elle est composée de 12 cellules binaires.



- Si valideur = 0 toutes les sorties du décodeur sont à zéro, la mémoire donc n'est pas sollicitée.
- Si valideur =1 : une sortie Ei du décodeur serait égale à 1 donc le mot **i** de la mémoire sera sélectionné par une opération de lecture ou écriture grâce aux entrées des lignes d'adresses .

5.2 Mémoire morte ROM (Read Only Memory):

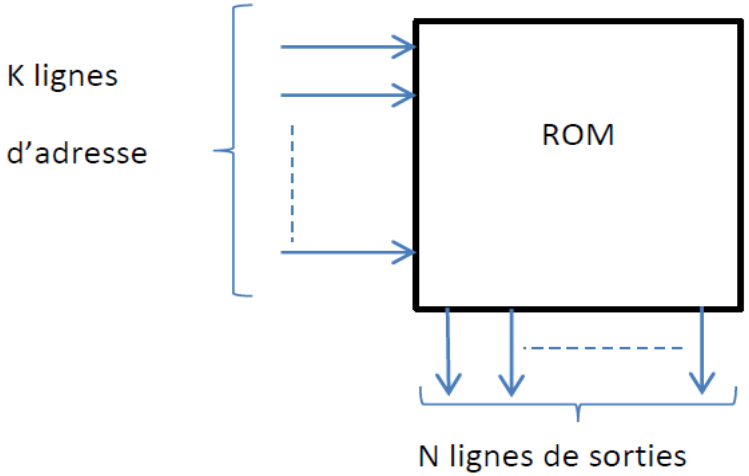
Pour certaines application, il est nécessaire de pouvoir conserver des informations (programmes, données… etc) de façon permanente même lorsque l'alimentation électrique est interrompu on utilise alors **des mémoires mortes ou mémoires à lecture seule**. Dans ce cas les informations contenues en mémoire ne peuvent être accédées qu'en lecture. Ceci implique que l'information binaire stockée dans une ROM ne peut être modifiée ni intentionnellement, ni accidentellement.

5.2.1 Principe de fonctionnement :

Une ROM de mxn bits est une matrice de cellules binaires organisée en m mots de n bits/mot.

Pour sélectionner un mot parmi m il faut k lignes d'adresse tel que $2^k = m$

Une ROM possède aussi n lignes de sortie.



5.2.2 Structure interne d'une ROM :

La procédure utilisée par une ROM pour emmagasiner une information diffère d'une technologie à une autre. Dans la plupart des ROM, l'absence ou la présence d'une diode ou d'un transistor différencie entre un '0' et un '1'.

Exemple :
Soit une ROM 4x4 (4 mots, 4 bits/ mot)

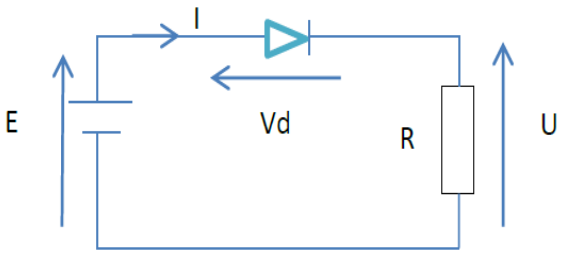
On veut que cette ROM contienne les combinaisons des bits donnés selon la table suivante:

Addresses		Mots			
X_1	X_2	Y_3	Y_2	Y_1	Y_0
0	0	1	0	1	0
0	1	1	1	0	0
1	0	0	0	1	0
1	1	1	1	1	1

Réalisation :

Rappel :

Circuit à diode :



Equations :

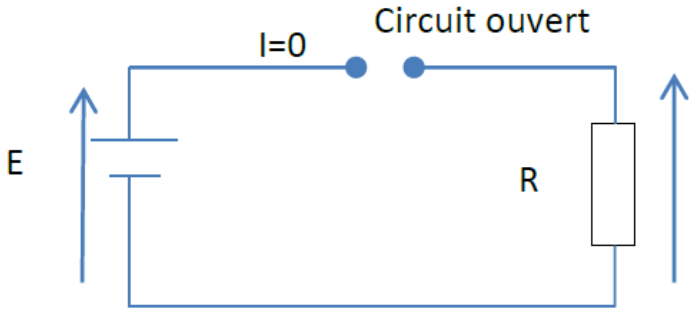
E=Vd+U

Loi d'hom : U=RxI

Lorsque la diode est conductrice on peut négliger la chute de tension dans la diode et on aura U=E=R×I.

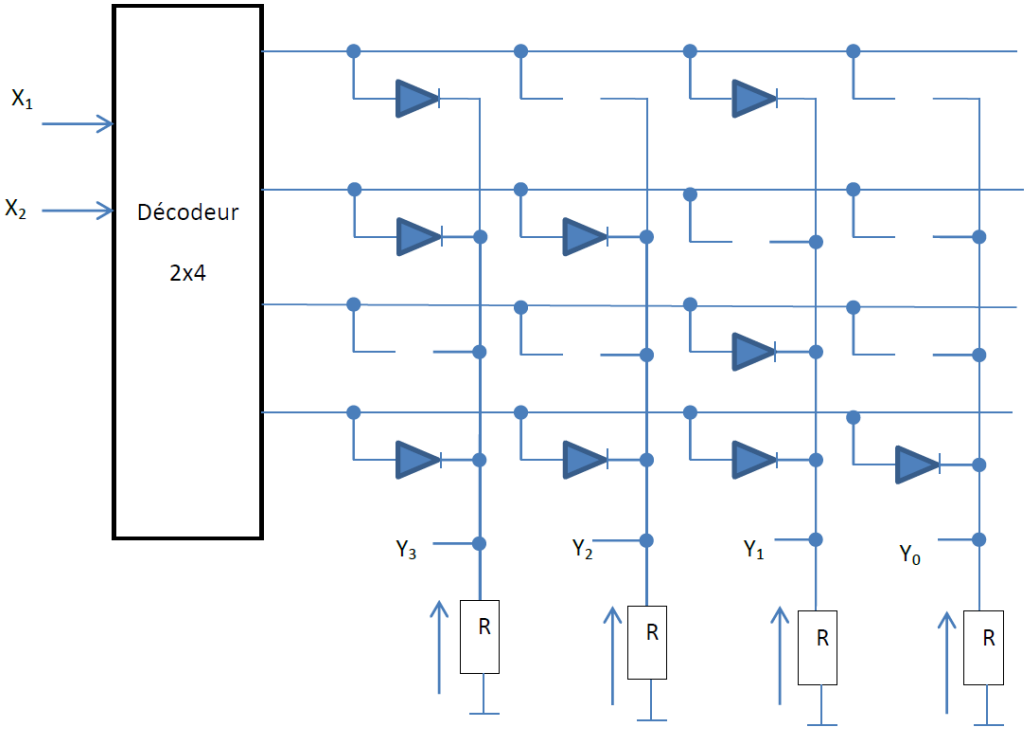
Si E= 5v= '1' binaire alors U='1'

Si la diode est brisée (circuit ouvert), elle devient non conductrice le schéma devient comme suit :



U=RxI=Rx0=0V= '0' binaire

En utilisant ce principe on peut réaliser dans une ROM les '1' et les '0' de la table de vérité précédente comme suit.



Addresses | Mots

X_1	X_2	Y_3	Y_2	Y_1	Y_0
0	0	1	0	1	0
0	1	1	1	0	0
1	0	0	0	1	0
1	1	1	1	1	1

Structure interne d'une ROM 4X4 bits

Présence d'une diode signifie la présence d'un '1' et son absence un '0'

5.2.2 Applications des ROM :

Réalisation des fonctions logiques usuelles :

Exemple :

Réalisation de 8 fonctions logiques programmables f0 ⟶ f7 à deux variables a et b à partir d'une ROM.

$$f_0(a,b) = a.b$$
$$f_1(a,b) = a.\overline{b}$$
$$f_2(a,b) = \overline{a.b}$$
$$f_3(a,b) = a + b$$
$$f_4(a,b) = a + \overline{b}$$
$$f_5(a,b) = \overline{a + b}$$
$$f_6(a,b) = a \oplus b$$
$$f_7(a,b) = \overline{a \oplus b}$$

Pour ces 8 fonctions par rapport à 4 combinaisons de a et b, nous obtenons 8x4=32 valeurs possibles

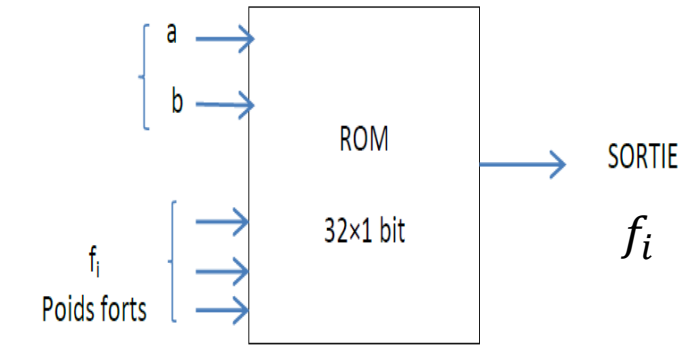
Table de vérité

I1	I2	I3	a	b	fi
0	0	0	0	0	0
0	0	0	0	1	0
0	0	0	1	0	0
0	0	0	1	1	1
0	0	1	0	0	0
0	0	1	0	1	0
0	0	1	1	0	1
.	.	.	.	.	.
.	.	.	.	.	.
1	1	1	1	1	1

32 combinaisons

⇒K=5 lignes d'adresse ⟶ 2 lignes d'adresse a et b

3 lignes d'adresse fonctions f_i et une sortie



i=0→7⇒000→111 (binaire)

5.2.3 Les variantes de la ROM :

- ✓ ***PROM : (Programmable ROM)***

C'est une mémoire morte programmable une seule fois par l'utilisateur.

- ✓ ***EPROM : (Erasable Programmable ROM)***

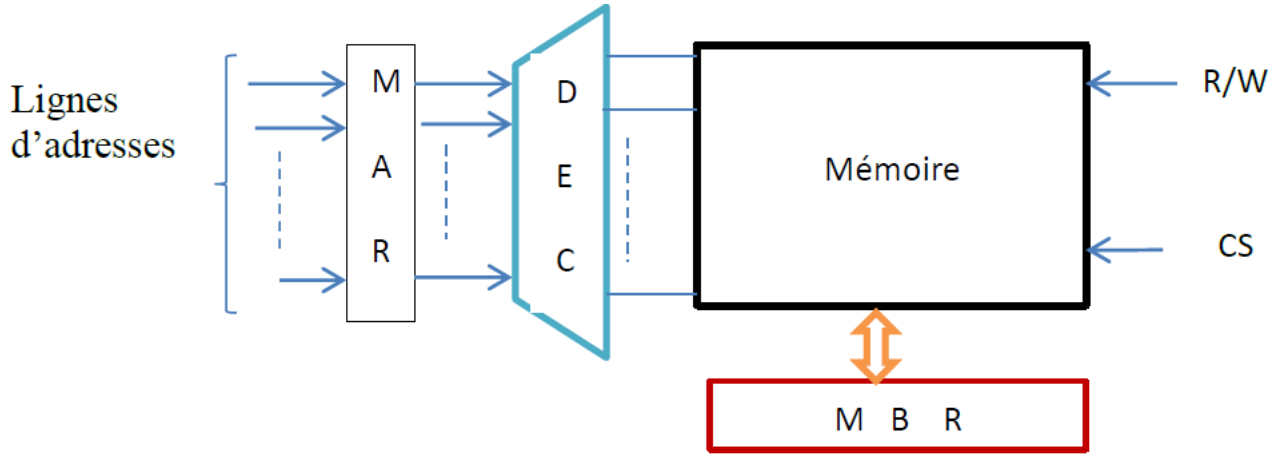
On l'appelle aussi REPROM (Reprogrammable ROM). Le principe de cette mémoire est le même que celui de la PROM mais elles fournissent, en plus, la possibilité de pouvoir être effacées un certain nombre de fois. L'effacement se fait par Ultra-Violet.

- ✓ ***EEPROM : Pour Electrically Erasable Programmable ROM :***

Programmable et effaçable électriquement.

5.3 Opérations sur les mémoires :

Les opérations du microprocesseur sur les mémoires sont simplifiées par l'utilisation de deux registres particuliers comme mentionné dans le schéma suivant :



MAR : Registre d'adresse mémoire : il contient une adresse qui nous permet d'accéder directement à un mot précis de la mémoire par le biais d'un DEC (décodeur).
MBR : Registre de données binaires : c'est un registre tampon par le quel passent toutes les données allant de la mémoire vers le microprocesseur et inversement.
CS : permet la sélection de la mémoire ou son verrouillage.

Séquences d'opérations pour la lecture et l'écriture :

- CS=1 : Mémoire sélectionnée.

Lecture :

- Transfert des bits d'adresse du mot choisi dans le MAR
- Activation de la ligne de contrôle de lecture **R/W=1**.

Opération réalisée : Le mot sélectionné est disponible dans le MBR.

Ecriture :

- Transfert des bits adresse du mot choisi dans le MAR
- Transfert des bits de données dans le MBR
- Activation de la ligne de contrôle **R/W=0**.

Opération réalisée : le mot contenu dans le MBR est alors rangé dans la case mémoire spécifiée par le MAR.

6. Extension de la mémoire :

L'objectif est constamment de chercher à augmenter le nombre de mots de la mémoire ou la taille des mots de cette mémoire , ou bien les deux à la fois.

Pour réaliser des mémoires de grande capacité on peut utiliser des mémoires de petites capacités et les connecter entre elles. En conséquence:

- Si on veut augmenter le nombre de mots on connecte les mémoires en parallèle tout en sélectionnant le nombre de ligne du bus d'adresse nécessaire.
- Si on veut augmenter la longueur du mot mémoire on connecte les mémoires en série en assurant que cette longueur ne dépasse pas la taille du bus de données.

6.1 Augmentation de l'espace d'adressage

Si on veut augmenter l'espace d'adressage d'une RAM (c.-à-d. augmenter le nombre de mot) dans ce cas il faut connecter plusieurs RAM , ayant la même longueur des mots , en parallèle.

Pour la réalisation d'une RAM (2^m × D) à l'aide de RAM (2^n× D) (n<m) il faut d'abord définir le nombre de RAM (2^n × D) nécessaires. Pour ce faire il suffit de diviser la capacité de la RAM requise par la capacité des RAM proposées :
(2^m × D) / (2^n × D) = 2^{m-n}

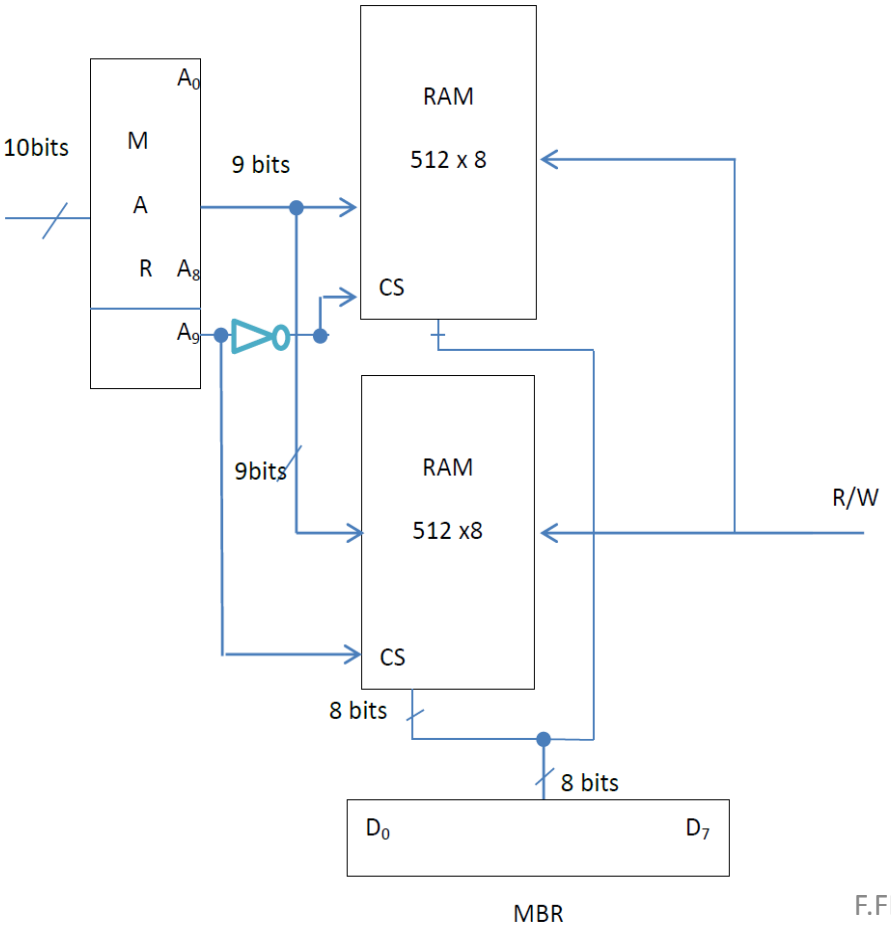
Registres utilisés:

- Un registre d'adresse (MAR) de **m** bits
- Un registre de données de **D** bits.

Exemple : m=10, n=9 et D=8

On veut réaliser une RAM 1K x 8 à partir de RAM 512 x 8

Le nombre de RAM nécessaires est : $2^{10}/\ 2^{9}$= 2RAM
On aura un MAR de 10bits (*A9…A0*) et le MBR contient 8 bits (*D*7….*D*0).



Plage des adresses

	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
Plage d'adresse RAM1, 512 mots	0	0	0	0	0	0	0	0	0	0
	.	.	.	.	.	.	.	.	.	.
	.	.	.	.	.	.	.	.	.	.
	0	1	1	1	1	1	1	1	1	1
Plage d'adresse RAM2, 512 mots	1	0	0	0	0	0	0	0	0	0
	.	.	.	.	.	.	.	.	.	.
	.	.	.	.	.	.	.	.	.	.
	1	1	1	1	1	1	1	1	1	1

6.2 Augmentation de la longueur du mot :

Si on veut augmenter la longueur d'un mot mémoire il suffit de connecter plusieurs boitiers en série qui ont le même nombre de mots.

Pour la réalisation d'une RAM (2^n× D) à l'aide de RAM (2^n× Q) il faut d'abord définir le nombre de RAM (2^n × Q) nécessaires. Pour ce faire Il suffit de diviser D par Q.

On utilisera dans ce cas un seul registre d'adresse (MAR) de n bits, et un même Chip Select pour les $\frac{D}{Q}$ RAMs.

Le registre de données doit être de taille au moins égale D.

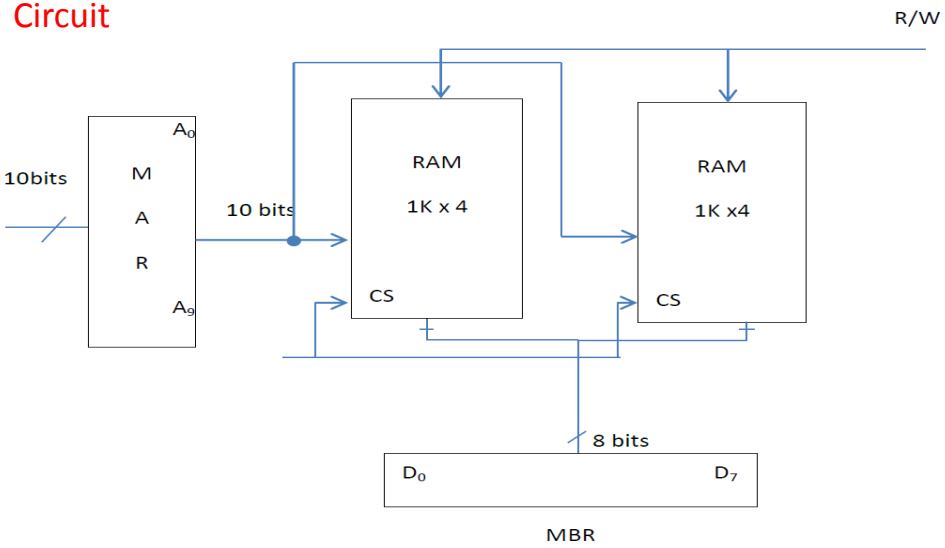
Exemple n = 10, D = 8 et Q = 4

On veut réaliser une RAM 1K x 8 à partir de RAM 1K x 4

Le nombre de RAM nécessaires est : 8 / 4 = 2 RAM

On aura un MAR de 10 bits et le MBR contient 2 x 4 = 8 bits.

Circuit



Ce circuit est équivalent à une RAM **1K** x 8.
Chaque mot est composé de 2 sous-mots ayant la même adresse.

Remarque :

En pratique, quel que soit le nombre de RAM connectées en série, il suffit de les connecter toutes au même Chip Select(Cs), au même registre d'adresse (MAR) et leur appliquer le même signal de lecture/écriture (R/W).