
Chapitre 4

La mémoire centrale

Pré requis	Représentation de l'information Circuits logiques de bases
Objectifs du chapitre	Comprendre le principe de fonctionnement de la mémoire centrale. Connaître l'organisation interne de la mémoire centrale. Enumérer les caractéristiques de la mémoire centrale. Recenser les types de mémoire qui constituent la mémoire centrale. Se familiariser avec la manière d'assembler les blocs mémoires pour constituer la mémoire centrale.
Mots clés du chapitre	Mémoire centrale, Mot mémoire, Accès, Lecture, Ecriture, Bloc mémoire, Module (Carte) mémoire.
Éléments de contenu	<ol style="list-style-type: none">1. Introduction2. Le fonctionnement de la mémoire centrale<ol style="list-style-type: none">2.1. Terminologie et définition2.2. Schéma de principe de la mémoire centrale2.3. Organisation interne de la mémoire centrale2.4. Caractéristiques de la mémoire centrale2.5. Typologies de la mémoire centrale2.6. Assemblage de blocs mémoires pour constituer la mémoire centrale3. La mémoire cache4. En pratique :<ol style="list-style-type: none">4.1. Les barrettes mémoires5. En résumé :6. Série d'Exercices

1. Introduction

Jusqu'à présent on a parlé de la mémoire centrale comme étant une sorte de « boîte noire » dans laquelle le processeur peut placer des suites binaires pour les retrouver ultérieurement.

Tout au long de l'histoire de l'informatique, la mémoire centrale a été réalisée selon plusieurs technologies. On trouve les tubes à vide, les tores magnétiques et les mémoires à semi conducteurs. Dans toutes ces technologies, on cherchait à mémoriser l'information élémentaire ou le bit, en lui attribuant une composante matérielle capable d'avoir deux états stables. Commenant par un tube à vide, passant vers un tore magnétique et finissant par les bascules, on parle toujours d'une cellule ou un point mémoire.

Nous avons déjà rencontré les registres de mémorisation, mais ceux-ci ne sont pas adaptés aux grandes capacités de stockage. On a aussi défini la mémoire comme étant la capacité de maintenir des valeurs binaires en sortie des circuits logiques et ce même si on élimine les valeurs appliquées à leurs entrées.

2. Le fonctionnement de la mémoire centrale

2.1. Terminologie et définition

Mémoire centrale

Appelée aussi mémoire principale, elle contient les instructions et les données, donc des programmes que l'on désire exécuter.

Accès en lecture et écriture

L'accès en entrée vers la mémoire centrale s'appelle écriture (on raisonne par rapport à la mémoire centrale). De même l'accès en sortie s'appelle lecture.

Point mémoire

C'est un circuit à deux états stables, capable de stocker l'un des deux chiffres binaires 1, 0 (digits).

Mot mémoire

Pour accéder à la mémoire centrale, on doit lire ou écrire un ensemble de bits de taille fixée à l'avance. Ce lot de bits à accéder soit lecture ou en écriture, s'appelle mot mémoire. La mémoire centrale n'est donc qu'une suite de mots. Chaque mot est identifié par un numéro unique appelé adresse.

2.2. Schéma de principe de la mémoire centrale

Extérieurement, et en ne tenant compte que des signaux logiques, un bloc mémoire peut être représenté comme sur la figure 1. Pour pouvoir identifier individuellement chaque mot on utilise k lignes d'adresse. La taille maximale d'un bloc mémoire est donc 2^k mots, le premier mot se situant à l'adresse 0 et le dernier à l'adresse $2^k - 1$. Une ligne de commande (R/W) indique si la mémoire est accédée en écriture (l'information doit être mémorisée) ou en lecture (l'information doit être restituée). Les accès en entrée et en sortie peuvent être confondus en un seul canal bidirectionnel. La ligne de validation ou de sélection du bloc (CS) n'est autre que la commande d'autorisation de la mémoire.

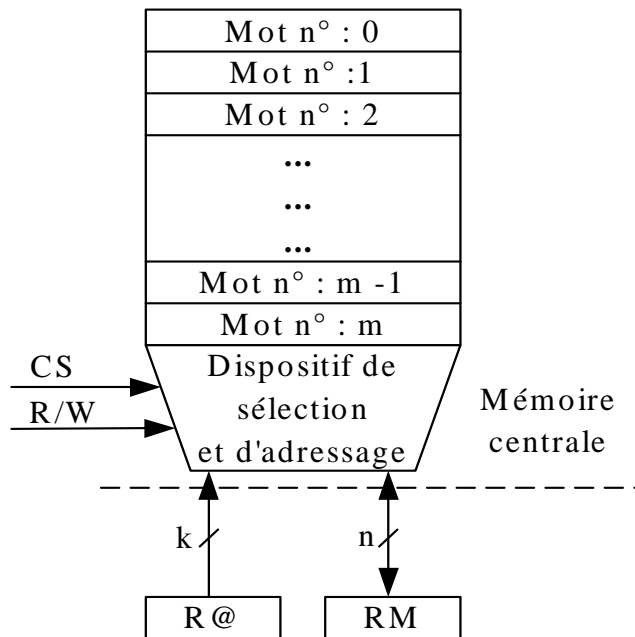


Figure 1.1. Schéma de principe de la mémoire centrale.

Dispositif d'adressage et de sélection

Ce circuit permet de localiser un mot mémoire étant donnée son adresse dans le registre d'adresses, puis d'effectuer l'opération d'accès (en lecture ou en écriture) depuis ou vers le registre mot.

Registre d'adresses

Le registre d'adresse qui appartient à l'unité centrale de traitement, sert comme renseignement pour le dispositif de sélection et d'adressage, pour localiser le mot à lire ou à écrire.

Registre mot

Il doit contenir la valeur du mot à écrire dans la mémoire centrale (MC) et ce avant une opération d'écriture. Il contiendra aussi la valeur du mot après une opération de lecture.

Lecture d'un mot mémoire

Pour lire un mot mémoire, l'unité centrale de traitement doit mémoriser l'adresse de ce mot dans le registre d'adresses, ensuite elle donne l'ordre d'accès en lecture au dispositif de sélection et d'adressage (CS à 1 et R/W à 1). Celui-ci va faire sortir la valeur du mot considéré vers le registre mot.

Ecriture d'un mot mémoire

L'unité centrale de traitement va mémoriser l'adresse du mot à écrire dans le registre d'adresses. En même temps, elle va passer la valeur à écrire dans le registre mot. Elle donne ensuite un ordre d'écriture au dispositif de sélection et d'adressage. Celui-ci va transférer la valeur du registre mot vers le mot mémoire concernée.

Remarques

1. La taille du registre mot doit être identique à la taille du mot mémoire.
2. La taille du registre d'adresses détermine le nombre maximum de mots dans la mémoire centrale.

Application :

Exprimer la taille minimale du registre d'adresses en fonction de :

- La taille de la mémoire centrale : TMC (bits).
- La taille d'un mot mémoire : TMM (bits).
- Le nombre de mot mémoire : N.

2.3. Organisation interne de la mémoire centrale

Le dispositif de sélection et d'adressage est conçu à base d'un décodeur. Une adresse en entrée du décodeur n'active à sa sortie qu'une seule ligne à la fois. Chaque ligne de sortie du décodeur nous permettra de sélectionner un mot mémoire. La figure ci-dessous nous montre, comment réaliser une mémoire de quatre mots, chacun de quatre bits. Ce raisonnement peut être applicable à n'importe quel bloc mémoire indépendamment de sa taille.

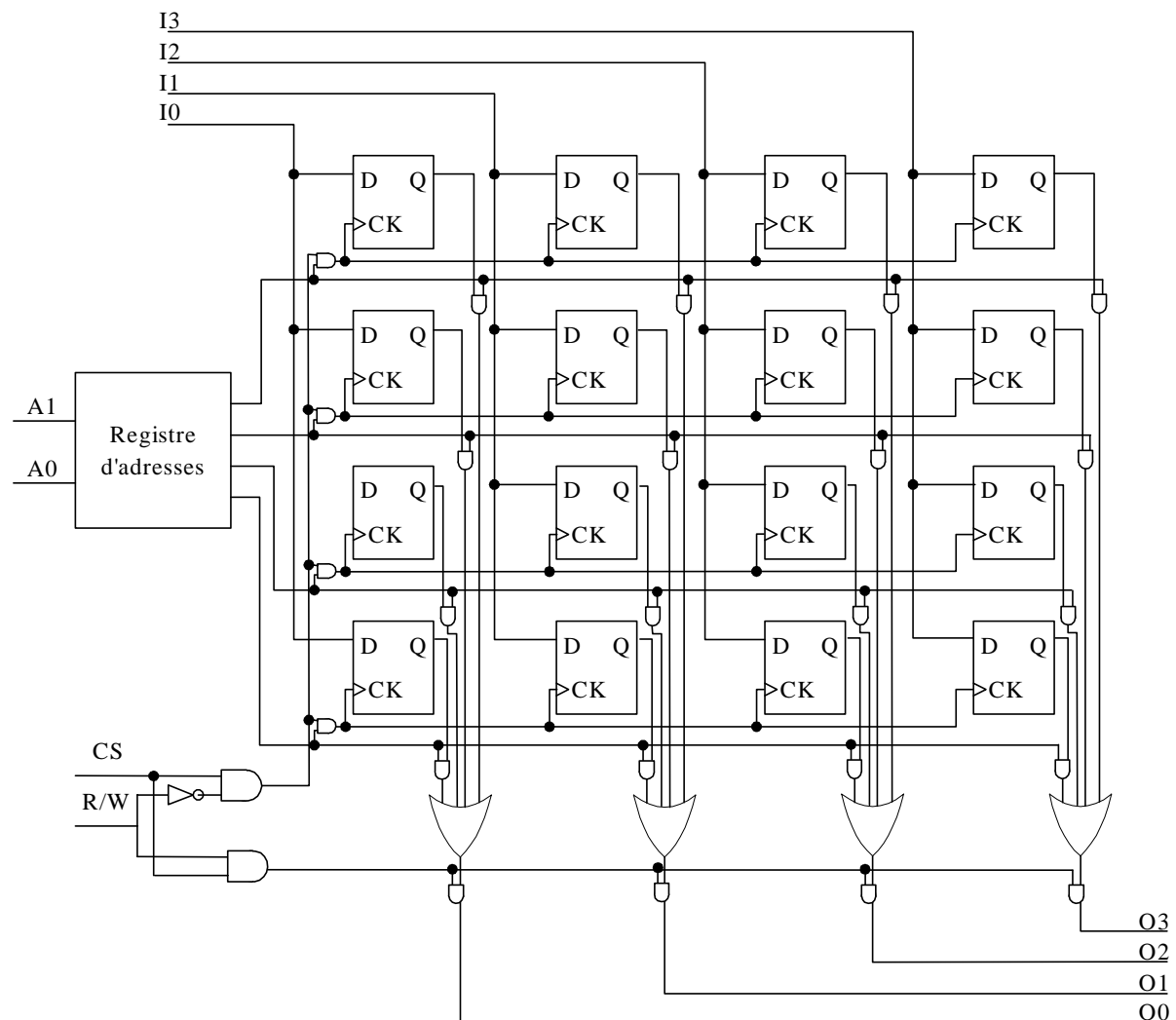


Figure 1.2. Organisation d'une mémoire de 4 mots de 4 bits chacun.

On sait qu'une lecture et une écriture ne peuvent pas être effectuées simultanément. Il est donc possible d'utiliser les mêmes lignes pour lecture et pour l'écriture. Ce qui permettra de diminuer le nombre de broches dans les circuits de mémoire. Le fait de lier une entrée et une sortie sur la

même ligne, cause un problème d'interférence. Pour le résoudre, on fait appel à des portes « 3 états ». Cet interrupteur électronique appelé aussi « buffer », comprend une entrée, une sortie et une commande. Quand cette dernière est à 0, ce circuit se comporte comme un interrupteur ouvert. Dans le cas contraire, il se comporte comme un interrupteur fermé.

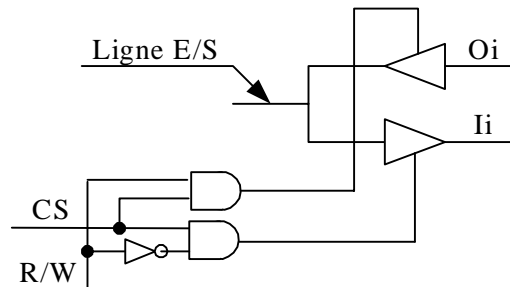


Figure 1.3. Réalisation de lignes bidirectionnelles.

2.4. Caractéristiques de la mémoire centrale

Capacité :

Dite aussi taille de la mémoire, elle correspond au nombre d'informations qu'elle peut contenir. Elle est généralement exprimée en bits en octets ou en nombre de mots mémoire.

Temps d'accès

C'est le temps qui sépare le début de l'opération d'accès et sa terminaison. Dans la pratique et pour plusieurs types de mémoire centrale, le temps que demande une opération de lecture peut être différent de celui d'une opération d'écriture. Dans ce cas on compte le temps le plus long.

Cycle mémoire

C'est le temps minimal s'écoulant entre deux accès successifs à la mémoire.

Théoriquement, le cycle mémoire est égal au temps d'accès. Mais pratiquement le premier est plus long que le deuxième, car le bon fonctionnement de la mémoire nécessite quelques opérations de maintien et de stabilisation des signaux dans le circuit.

Débit

C'est le nombre d'informations (exprimé en bits) lues ou écrites par seconde. Il est exprimé en fonction du cycle mémoire (CM) et de la taille du mot mémoire (TMM) comme suit :

$$D = \frac{TMM}{CM}$$

Volatilité

Elle caractérise la permanence des informations dans la mémoire centrale. Une mémoire volatile perd son contenu lorsqu'on coupe le courant. Celle-ci a donc besoin d'un apport constant d'énergie électrique pour conserver ses informations. La mémoire à base de transistors est volatile. On peut réaliser des mémoires non volatiles avec les transistors moyennant un générateur de courant électrique (batterie). Tel est le cas pour la mémoire CMOS qui contient les paramètres de configuration de l'ordinateur. Il y a une autre catégorie de mémoire, à base de résistances électriques, qui constitue une mémoire permanente. C'est le cas de la mémoire centrale qui contient le bios.

2.5. Typologies de la mémoire centrale

RAM (Random Access Memory)

Appelée aussi, mémoire à accès aléatoire, ou encore à accès direct, c'est une mémoire où, tous les mots sont accessibles directement moyennant leur adresse.

RWM, ROM (Read Write Memory, Read Only Memory):

Les points mémoires, décrits jusqu'ici ont la particularité de pouvoir être lus et écrits. Il constituent une mémoire RWM, appelé aussi mémoire vive par opposition à la mémoire morte qui ne permet que des opérations de lecture et qui est appelé aussi ROM. Dans le cas d'une mémoire ROM les opérations d'écriture sont soit impossibles, soit possibles sous des conditions particulières non existantes au niveau d'un ordinateur (Avec un dispositif spécial appelé programmeur).

SRAM, DRAM (Static RAM, Dynamic RAM)

La première est une mémoire vive statique (stable) c'est-à-dire, qui ne nécessite aucune stabilisation ni maintient des signaux électriques. Ce type de mémoire est conçu à base de transistors uniquement.

La deuxième est une mémoire vive dynamique, qui doit être rafraîchie périodiquement (par exemple tous les millisecondes), ce type de mémoire est conçu à base de transistors et de condensateurs.

PROM, EPROM (Programmable ROM, Erasable Programmable ROM)

La mémoire de type PROM est une mémoire morte programmable une seule fois (avec une machine spéciale) et ce d'une manière irréversible.

Les mémoires EPROM, sont des mémoires mortes, qui offrent la possibilité d'être effaçables et programmables un certain nombre de fois.

2.6. Assemblage de blocs mémoires pour constituer la mémoire centrale

Les techniques d'intégration ne permettent pas d'obtenir des boîtiers ayant des capacités ou des formats suffisants pour toutes les applications. Il est alors nécessaire d'associer plusieurs boîtiers pour augmenter la longueur des mots ou le nombre de mots. L'association de plusieurs blocs peut permettre d'améliorer les performances temporelles de la mémoire en faisant fonctionner plusieurs blocs en parallèle.

2.6.1. Augmentation de la longueur des mots

La figure suivante montre qu'il est aisé d'associer deux boîtiers de 2^k mots de n bits pour obtenir un bloc de 2^k mots de $2n$ bits. L'adressage doit être appliqué simultanément aux deux circuits, l'un fournissant les n bits de bas poids et l'autre les n bits de haut poids.

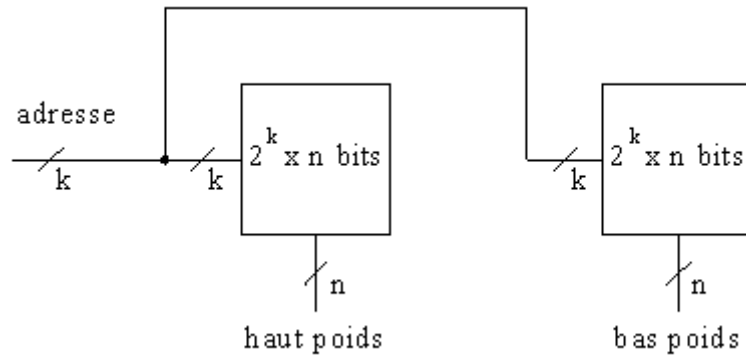


Figure 1.4. Augmentation de la longueur des mots.

2.6.2. Augmentation du nombre de mots

La figure suivante montre la réalisation d'un bloc de 4×2^k mots de n bits à l'aide de 4 boîtiers de $2^k \times n$ bits. Il faut $k+2$ lignes d'adresses. Les k bits de bas poids de l'adresse sont appliqués simultanément sur les 4 boîtiers. Les deux bits de haut poids attaquent un décodeur à quatre sorties. Chacune de ces quatre lignes permet de sélectionner un boîtier (entrée de validation du boîtier : CS). Un seul boîtier est alors connecté aux lignes de sortie.

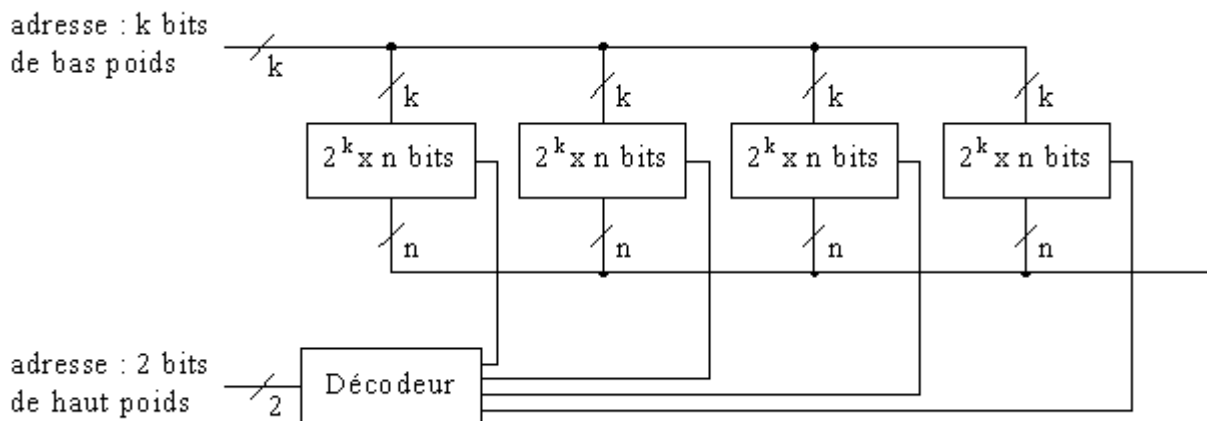


Figure 1.5. Augmentation du nombre de mots.

3. La mémoire cache

La mémoire cache est une mémoire supplémentaire qu'on rajoute dans les ordinateurs pour qu'elle accélère l'opération de lecture de données et des instructions à partir de la mémoire centrale.

En effet, lors d'une action de lecture d'une donnée à partir de la mémoire centrale, le processeur passe une bonne partie de son temps à l'attente de la réponse de la mémoire centrale. On parle de temps de silence du processeur.

L'idée est de mettre en place une mémoire qui communique avec le processeur en un temps plus réduit. Et cette possibilité existe mais avec un coût relativement élevé. Dans le cas le plus écheant on construit des mémoires avec la technologie SRAM.

L'emplacement de la mémoire cache peut se faire selon les trois possibilités suivantes :

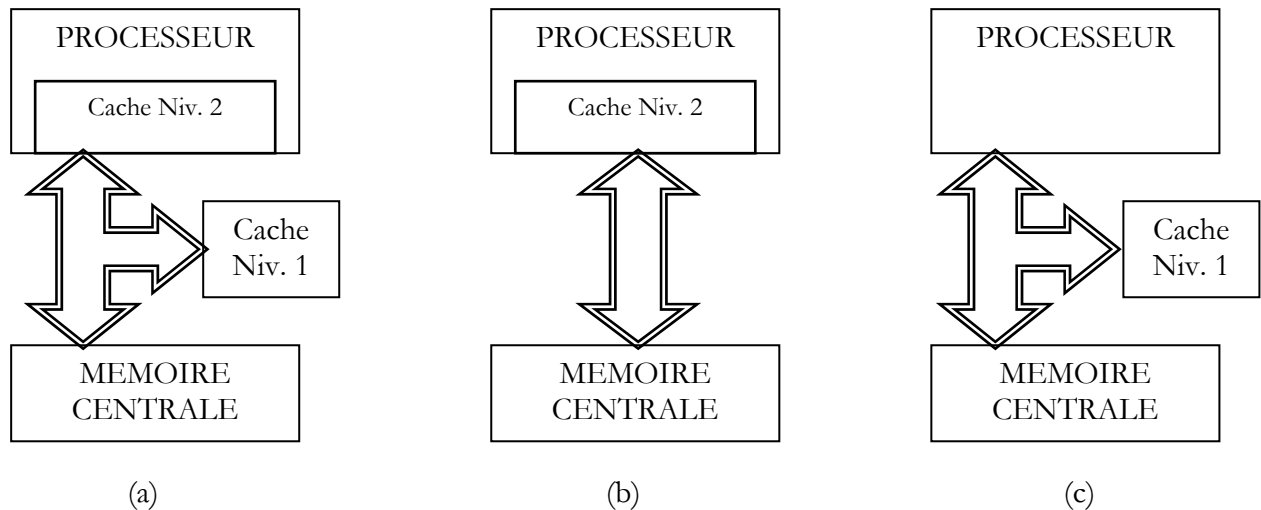


Figure 1.6. : (a) : deux niveaux de cache un interne et un autre externe
 (b) : un seul niveau de cache interne
 (c) : un seul niveau de cache externe

Si d'un autre coté on prend en considération que certaines opérations de lecture se font sur plus qu'un cycle imaginons alors les temps de silence qui êtres engendrés.

Le principe de fonctionnement d'un système construit autour d'une mémoire cache se base dans le cas le plus échéant sur le placement d'une copie de la donnée ou de l'instruction qui a été référencée en dernier lieu dans la mémoire cache. Cette manière de gestion à été décidé vu qu'en générale un programme contient très fréquemment des exécutions itératives et aussi vu qu'une donnée, si elle a été utilisée à un instant donné elle le sera très prochainement.

Que penseriez vous alors d'une UCT capable de trouver toutes les références dont elle a besoin dans une mémoire plus rapide que la mémoire centrale.

Avec un tel système l'UCT sera sans aucun doute plus rapide. Généralement, l'UCT si elle fera référence à un mot mémoire d'adresse « A » elle fera probablement à l'instant suivant référence à un mot voisin. C'est pour cela que le microprogramme procède à une copie par bloc de donnée ou d'instructions dans la mémoire cache. Ainsi, on augmente la probabilité de retrouver le mot recherché dans la mémoire cache au lieu qu'on le trouve dans la mémoire centrale.

Nous parlons d'une performance maximale autour d'un système formé par une cache, lorsque, si un programme fait référence à un mot mémoire K fois, il fera au cours de l'exécution réelle : (K-1) fois accès au cache et une seule fois accès à la mémoire centrale.

Soient :

- C : le temps d'accès au cache.
- M : le temps d'accès à la mémoire
- H : le taux de présence d'une référence dans le cache qui exprime le nombre de référence au cache par rapport au nombre de références totales.

Au meilleur cas : $H = (K - 1) / K$

Au pire des cas : $H = 0 / K$

Ainsi, le temps moyen d'accès à la mémoire est : $T_{acc} = C + (1-H) \times M$ avec (1-H) est le taux d'insuccès.

4. En pratique :

4.1. Les barrettes mémoires

Il existe de nombreux types de mémoires vives. Celles-ci se présentent toutes sous la forme de barrettes de mémoire enfichables sur la carte mère.

Les premières mémoires se présentaient sous la forme de puces appelées *DIP (Dual Inline Package)*. Désormais les mémoires se trouvent généralement sous la forme de barrettes, c'est-à-dire des cartes enfichables dans des connecteurs prévus à cet effet. On distingue deux types de barrettes de RAM :

- les barrettes au format **SIMM** (*Single Inline Memory Module*) : il s'agit de circuits imprimés dont une des faces possède des puces de mémoire. Il existe deux types de barrettes SIMM, selon le nombre de connecteurs :
- Les barrettes SIMM à 30 connecteurs (dont les dimensions sont 89x13mm) sont des mémoires 8 bits qui équipaient les premières générations de PC (286, 386).

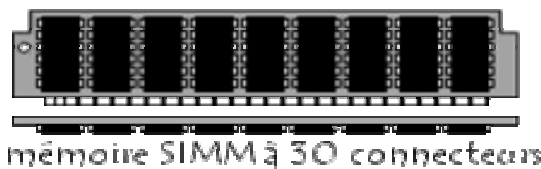


Figure 1.7. Premier exemple de carte mémoire SIMM.

- Les barrettes SIMM à 72 connecteurs (dont les dimensions sont 108x25mm) sont des mémoires capables de gérer 32 bits de données simultanés. Ces mémoires équipent des PC allant du 386DX aux premiers pentiums. Sur ces derniers le processeur travaille avec un bus de données d'une largeur de 64 bits, c'est la raison pour laquelle il faut absolument équiper ces ordinateurs de deux barrettes SIMM. Il n'est pas possible d'installer des barrettes 30 broches sur des emplacements à 72 connecteurs dans la mesure où un détrompeur (encoche au centre des connecteurs) en empêche l'enfichage.

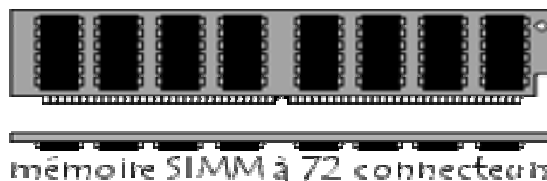


Figure 1.8. Deuxième exemple de carte mémoire SIMM.

- Les barrettes au format **DIMM** (*Dual Inline Memory Module*) sont des mémoires 64 bits, ce qui explique pourquoi il n'est pas nécessaire de les appairer. Les barrettes DIMM possèdent des puces de mémoire de part et d'autre du circuit imprimé et ont également 84 connecteurs de chaque côté, ce qui les dote d'un total de 168 broches. En plus de leurs dimensions plus grandes que les barrettes SIMM (130x25mm) ces barrettes possèdent un second détrompeur pour éviter la confusion.

A noter que les connecteurs DIMM ont été améliorés afin de permettre une insertion facile des barrettes grâce à des leviers situés de part et d'autre du connecteur.

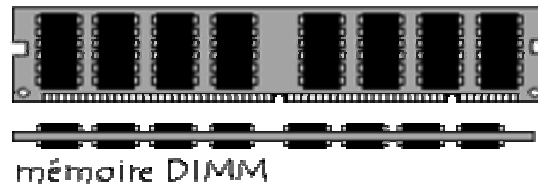


Figure 1.9. Exemple de carte mémoire SIMM.

4.1.1. Reconnaître et choisir une barrette de mémoire

Pour reconnaître une barrette de mémoire, vous devez tout d'abord identifier le type de barrette que vous pouvez mettre dans votre ordinateur. Pour cela, allez jeter un oeil du côté des emplacements mémoire (les bancs).

Si vous avez deux détrompeurs, il s'agit soit de SDRAM soit de RAMBUS. La RAMBUS à ses deux détrompeurs très rapprochés (voir ici), alors que la SDRAM (voir ici) en à deux mais éloignés. La SDRAM-DDR (voir ici) n'a qu'un détrompeur situé presque au milieu des bancs.

Identifiez aussi la barrette : des informations peuvent être inscrites sur une étiquette collée dessus. Dans le cas contraire, On peut aussi se fier à la hauteur des chips : les chips sont très peu épais, environ 1 mm (c'est de la mémoire « récente » : SDRAM, SDRAM-DDR, SDRAM-DDR2, DR-RAM, etc.), si les chips sont assez épais, plus de 2 mm (c'est de la SIMM, SIMM-EDO, etc.). On peut aussi compter le nombre de broches : 30 (c'est de la SIMM-30), 72 (c'est de la SIMM-72 ou SIMM-EDO), 168 (c'est de la SDRAM), 184 (c'est de la SDRAM-DDR, SDRAM-DR2, DR-SDRAM), 240 (c'est de la SDRAM-DDR2). Enfin, pour savoir si votre barrette est une DIMM, SO-DIMM et Micro DIMM, il suffit de mesurez la longueur de la barrette. Avec 133 mm, votre barrette sera une DIMM. 67 mm pour une SO-DIMM 144 broches, 39 mm pour la Micro DIMM 144 broches et 44 mm pour la Micro DIMM 172 broches (note : les mesures sont arrondies au mm le plus proche). Choisir de l'ECC ou de la NON-ECC ?

La mémoire enregistrée ECC (Error Correction Code) est la mémoire à correction d'erreur, elle est à réserver en général aux serveurs ou aux stations de calcul. La mémoire ECC à parité ajoute certaines fonctionnalités pour vérifier si les données en mémoire ont été corrompues. La mémoire ECC étend ces fonctionnalités et tente de corriger certaines erreurs de corruption de bits à la volée. Cette option s'applique surtout aux ordinateurs.

La plupart des autres architectures requièrent simplement de la mémoire ECC ou à parité. Par exemple, le Pegasos I a son chipset VIA qui ne gère que l'ECC. Plusieurs machines ne démarreront même pas avec de la mémoire sans parité. Si vous n'utilisez pas de la mémoire ECC ou à parité, vous aurez peut-être de la corruption de données et d'autres anomalies. Plusieurs fabricants de « mémoire à faible coût pour PC » ne font même pas une variété ECC ! Ce qui vous aiderez à les éviter.

Les fabricants d'ordinateurs vendent souvent plusieurs gammes de produits, organisées en serveurs et en stations de travail. Les serveurs vont contenir de la mémoire ECC dans leur architecture. Les fabricants de stations de travail Unix utilisent de la mémoire à parité et maintenant ECC depuis plusieurs années dans toutes leurs gammes de produits. Cette dernière offre aussi théoriquement un moyen de détecter des modifications semi-aléatoires causées par certains rayonnements (rayonnements alpha).

Utiliser de la mémoire ECC sur un ordinateur dont le contrôleur ne le gère pas, est une aberration. Prenez donc une mémoire NON-ECC qui sera moins chère et tout aussi efficace dans votre cas.

5. En résumé :

- Le fonctionnement de la mémoire centrale repose essentiellement sur la fourniture de l'adresse à travers le registre d'adresse du mot à accéder à travers le registre d'adresse, ainsi l'ordre d'accès en lecture et en écriture. En cas d'accès en lecture le résultat sera rangé dans le registre mot, inversement en cas d'écriture la valeur à ranger dans ce même registre.
- A l'intérieur de la mémoire centrale, on trouve les bistables avec des portes logiques qui permettent de les scinder en mots et de les commander soit en lecture ou en écriture.
- La mémoire centrale se caractérise par des facteurs qui sont: Capacité, Temps d'accès, Cycle mémoire, Débit, Volatilité.
- Pour réaliser des mémoires centrales de grande taille il y a des pratiques d'assemblage de blocs mémoires (circuits intégrés) qui permettent d'augmenter soit de la longueur des mots, soit le nombre de mots.
- Le rôle de la mémoire cache est de minimiser les temps d'attente du processeur.
- La mémoire cache est gérée par un microprogramme intégré dans le microprocesseur. Cette gestion se fait selon une stratégie bien déterminée. La stratégie de gestion de la mémoire cache dépend de sa taille et de son emplacement (interne, externe ou les deux en même temps)

6. Série d'Exercices

Exercice 1 :

Soit un fichier de taille 8.3 Méga octets. Le chargement depuis le disque vers la mémoire centrale s'effectue via un bus de 64 bits. Sachant que la vitesse du bus ne dépassant pas les 66 Mhz, déterminer le temps nécessaire pour charger le fichier (du disque vers la MC).

Exercice 2 :

Soit une mémoire centrale de 1 Mmots de 32 bits (soit 4Mo), réalisée avec des puces de 16Kbits. Cette mémoire peut être organisée selon plusieurs principes :

1. un bit par puce : un mot est constitué de 32x1 bit provenant chacun d'une puce différente; 32 puces sont nécessaires pour former un mot.
2. 16 bits par puce : un mot est constitué de 2x16 bits provenant chacun d'une puce différente; 2 puces sont nécessaires pour former un mot.
3. 32 bits par puce : les 32 bits proviennent de la même puce.

Donner le nombre de pattes de chaque puce utilisées pour l'adressage et pour les données dans chacun des cas.

Exercice 3 :

Si le registre d'adresse d'une mémoire comporte 32 bits, calculer :

- Le nombre de mots adressables si 1 mot = 1 byte ;
- La plus haute adresse possible pour ces mots de 1 byte ;
- Le nombre de mots adressables si 1 mot = 32 bits ;
- La plus haute adresse possible pour ces mots de 32 bits.

Exercice 4 :

On considère une mémoire centrale de 2 Mbytes, où chaque byte est adressable séparément :
Calculer l'adresse, en octal, du sixième élément d'un tableau dont l'adresse du premier élément est 77_8 , et dont tous les éléments sont composés de 16 bits.

Calculer, en décimal, le nombre de bytes précédents l'adresse 77_8 .

Exercice 5 :

On considère une machine avec la configuration suivante :

- mémoire centrale de taille 1Moctets
- mot mémoire de taille 2 octets
- bus d'adresse (ou registre adresse) de taille 20 bits.

Calculer la taille minimale du bus d'adresse qui permet d'accéder à cette mémoire.

Déterminer la plage d'adressage de cette mémoire (adresse minimale et adresse maximale).

En fait, cette mémoire est constituée de deux blocs séparés (2 puces différentes) comme le montre la figure ci dessous. Le premier est une DRAM de taille 512 Koctets adressable à partir de l'adresse $(00000)_{16}$ et le deuxième est une SRAM de taille 512Koctets adressable à partir de l'adresse $(60000)_{16}$.

Déterminer les deux plages d'adressage respectivement de la DRAM et la SRAM.

Quelle est la taille maximale que peut avoir la mémoire centrale sur cette machine ?

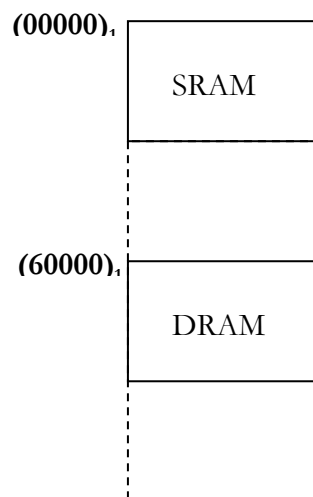
Justifier pourquoi la mémoire centrale de cet ordinateur est extensible ? Déterminer la taille de la mémoire d'extension ?

Justifier pourquoi la mémoire d'extension doit être organisée en au moins deux blocs mémoires (c.a.d. deux puces différentes).

Le cycle mémoire de la SRAM vaut 50 ns et celui de la DRAM vaut 250 ns

Justifier pourquoi le cycle mémoire de la SRAM est plus court que celui de la DRAM.

Sachant que le cycle de recherche d'une instruction revient à deux accès à la mémoire centrale et que le cycle d'exécution est négligeable devant le premier, déterminer le cycle moyen d'instruction dans le cas où le programme est chargé en mémoire statique puis dans le cas où il est chargé en mémoire dynamique.



Exercice 6 :

Soit une mémoire centrale de 1 Mmots de 32 bits réalisée avec des puces de 16 Kbits. Cette mémoire peut être organisée plusieurs principes ; nous considérons les trois suivants :

Un bit par puce : un mot est constitué de 32×1 bit provenant chacun d'une puce différente, donc 32 puces sont nécessaires pour réaliser un mot ;

16 bits par puce : un mot est constitué de 2×16 bits. Deux puces, fournissant chacune 16 bits, sont nécessaires pour former un mot de 32 bits ;

32 bits par puce : un mot est constitué de 1×32 bits provenant de la même puce.

Calculer :

Le nombre de bits nécessaires pour adresser toute la mémoire dans chacun des cas ;

Le nombre de pattes de chaque puce utilisées pour l'adressage et pour les données dans chacun des cas.

Proposer un montage pour cette mémoire dans les trois cas de figure.

Exercice 7 :

Soit une mémoire de capacité totale 1 Goctets. Sachant que cette mémoire est adressable par mot et que le nombre de bits sur ses lignes de données est de 32 bits :

Q1/ calculez la capacité de cette mémoire en Mots.

Q2/ donnez le nombre d'entrée d'adresse de cette mémoire.

Q3/ si le temps de lecture d'un bit est de 10^{-12} s calculer le temps nécessaire pour Lire un kilo octets de données.

Exercice 8 :

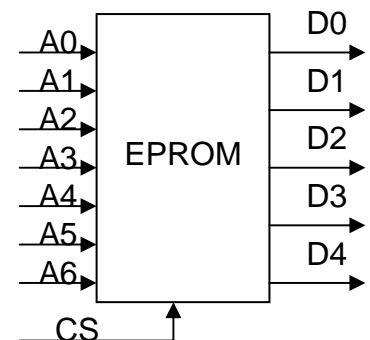
Supposant que le prix d'achat d'une EPROM de 128 Koctets est $3^{DT}200$ et que le prix d'achat d'une SDRAM de 256 Moctets est $50^{DT}000$ laquelle de ces deux mémoires possède un prix de revient meilleur ?

Exercice 9 :

Supposant qu'on dispose d'une EPROM représentée par le schéma suivant :

Q1/ proposé un montage qui nous permet d'augmenter la taille du mot mémoire pour qu'il passe de 5 bits à 10 bits.

Q2/ proposé un montage qui nous permet d'augmenter la taille de la mémoire pour qu'elle soit doublée.



Exercice 10 :

Soit le schéma suivant qui représente un montage de barrettes mémoire qui possède les caractéristiques suivantes :

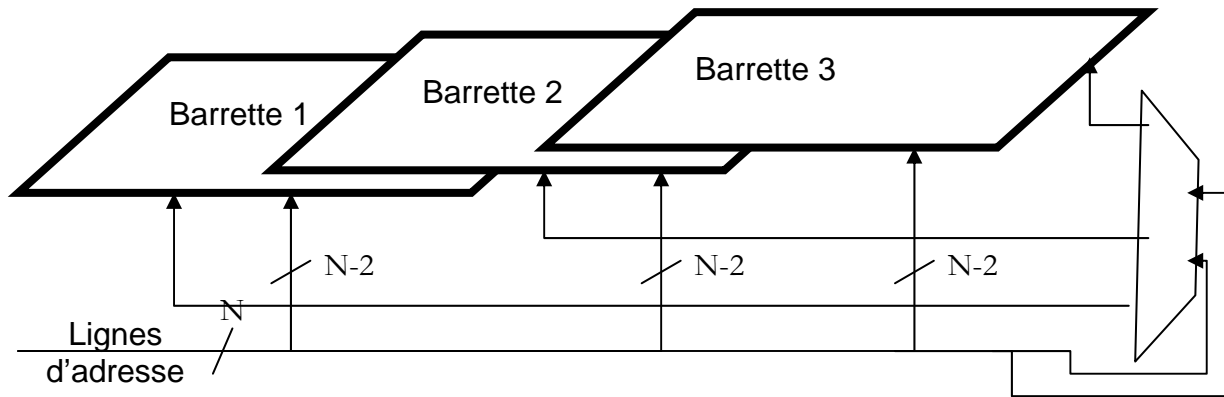
→ Chaque barrette est formée de 4 circuits mémoires chacune de capacité 64 Moctets

→ Tous les circuits mémoires d'une même barrette sont adressés et activés ensemble.

→ La taille d'un mot mémoire de chaque circuit est 64 bits.

Q1/ Donnez la taille en bits d'un mot mémoire de ce montage.

Q2/ Donnez alors la capacité totale de ce montage.



Exercice 11 :

On veut utiliser deux exemplaires de mémoire LIFO dans une calculatrice réalisant les 4 opérations $+$, $-$, \times , $/$ et $^$. La première pile sert au stockage des données, tandis que la deuxième sert au stockage des opérations. La calculatrice peut traiter des expressions arithmétiques non parenthésées contenant une ou plusieurs opérations se terminant par l'opérateur $=$.

1. Etudier le fonctionnement de cette calculatrice. Justifier l'emploi de deux piles.
2. Décrire les états successifs des piles lors de l'exécution des expressions suivantes:

$$4 - 6 - 12 + 36 =$$

$$3 \times 7 - 41 + 18 / 6 - 12 =$$

$$4 \times 3 / 6 + 15 - 9 / 3 =$$

3. Déterminer la profondeur maximale atteinte par chaque pile.

Exercice 12 :

On considère une mémoire cache de 4K mots, placées en tampon entre le système central et la mémoire centrale de 64K mots. On décide d'organiser la mémoire en blocs de 64 mots. Décrire la structure du cache mémoire et le découpage en champs de l'adresse d'un mot, pour une correspondance directe, puis associative.

Exercice 13 :

Soit une machine disposant d'une mémoire principale de 2^{18} mots de 32 bits, adressable à l'octet. Après initialisation, l'unité centrale effectue des opérations de lecture sur les adresses suivantes (ces adresses sont données en hexadécimal, dans l'ordre chronologique) :

10803₁₆ 2354F₁₆ 6BFA0₁₆ 129F4₁₆ 21507₁₆ 23548₁₆ 23596₁₆ 6BF08₁₆ 362B1₁₆ 45678₁₆ FF021₁₆
 98765₁₆ 14BB7₁₆ 32767₁₆ 98764₁₆ 45555₁₆ FACDE₁₆ 19980₁₆ 8EFCD₁₆ 93861₁₆

L'unité centrale utilise une mémoire cache d'une taille de 16 blocs, chaque bloc faisant 16 octets.

1. Quelle est la taille d'une adresse manipulée par cette machine ?
2. On veut réaliser la mémoire principale avec des boîtiers de mémoire $2^{16} \times 4$ bits. Combien de boîtiers sont nécessaires pour réaliser cette mémoire principale ?
3. En considérant une organisation à correspondance directe, donnez l'état de la table des étiquettes après les accès précédents.
4. En considérant un organisation associative basée sur une technique LRU (Least Recently Used) pour le remplacement des pages, donnez ci-dessous l'état de la table des numéros de page après les accès précédents (lorsque la table est vide, on charge dans l'ordre des entrées croissantes).

Exercice 14 :

Soit la portion de programme suivante :

Lire (A)

Lire (B)

$A = A + B$

$B = A * B$

On dispose d'une mémoire qui gère des mots de 32 bits et un bus de données ayant une fréquence de 100 Mhz.

On suppose que chaque instruction est chargée en mémoire centrale sur 2 octets.

A= 130 se trouvant à l'adresse 125 H et B se trouve à l'adresse suivante en mémoire.

- 1) Ecrire le programme en mémoire à partir de l'adresse 128H
- 2) Calculer le temps de chargement du programme de la mémoire centrale vers le processeur sachant que le temps d'accès de la mémoire est de 10 ns.
- 3) Le temps d'exécution d'une instruction par le processeur est de 2 ns. Quel est le temps nécessaire pour exécuter ce programme ?
- 4) supposons qu'on dispose d'une mémoire cache, sachant que le temps d'accès de la mémoire cache est de 2 ns. Calculer le temps d'exécution du programme. Conclure.

Exercice 15 :

Soit le programme suivant qui est chargé dans la mémoire centrale afin qu'il soit exécuté :

Pour $i := 0$ à 100 faire

$X \leftarrow i * Y$

$Y \leftarrow Y/2$

Fin Pour

- 1) Calculer le temps moyen d'accès à la mémoire centrale après une exécution totale de ce programme.
- 2) Supposons qu'on dotera notre système d'une mémoire cache de niveau 1 qui permet une lecture et une écriture à vitesses égales. Cette vitesse est vingt fois plus réduite que la lecture ou l'écriture dans la mémoire principale.

Calculer alors le temps d'accès moyen si on travaille avec une stratégie de remplissage du cache qui consiste à la mise d'une copie à chaque nouvelle instruction ou données référencée pour la première fois.