

ENSEIGNEMENT DE PROMOTION SOCIALE

Cours de
STRUCTURE DES ORDINATEURS
- La mémoire RAM -

H. Schyns

Janvier 2004

Sommaire

5. LA MÉMOIRE RAM

5.1. Introduction

5.2. Cellule de mémoire DRAM

5.3. Cellule de mémoire SRAM

5.4. Structure et fonctionnement de la DRAM

- 5.4.1. L'organisation de la mémoire
- 5.4.2. La lecture de la DRAM
- 5.4.3. L'amplification du signal
- 5.4.4. Le rafraîchissement
- 5.4.5. Lire 8, 16, 32 bits ou plus
- 5.4.6. La correction d'erreurs
- 5.4.7. L'augmentation de vitesse par entrelacement
- 5.4.8. L'adressage par "strobe" (RAS / CAS)
- 5.4.9. La séquence de lecture / écriture
- 5.4.10. Le timing

5.5. Puces de mémoire

- 5.5.1. Description générale
- 5.5.2. Deviner la capacité ?
- 5.5.3. Puces DIP (dual in-line package)
- 5.5.4. Puces SOJ (small outline j-lead)
- 5.5.5. Puces TSOP (thin small outline package)
- 5.5.6. Puces CSP (chip scale package)

5.6. Barettes de mémoire

- 5.6.1. Description générale
- 5.6.2. Disposition des puces
- 5.6.3. Nombre de broches

5.7. Evolutions de la RAM

- 5.7.1. DRAM ou SRAM
- 5.7.2. DRAM PM
- 5.7.3. DRAM FPM
- 5.7.4. Mode de transfert en rafale
- 5.7.5. DRAM EDO
- 5.7.6. SDRAM
- 5.7.7. DDR SDRAM
- 5.7.8. RAMBUS ou DR-SDRAM (Rambus DRAM)

5.8. En résumé

5.9. Installer la mémoire

EXERCICES DU CHAPITRE 5

- ♦ Exercice 5.1
- ♦ Exercice 5.2
- ♦ Exercice 5.3
- ♦ Exercice 5.4

5. La mémoire RAM

5.1. Introduction

La mémoire vive ou RAM (*ang.: Random Access Memory*) est la mémoire principale du système.

L'ordinateur utilise la RAM pour y stocker les instructions des programmes en cours d'exécution ainsi que les données et fichiers sur lesquels il travaille. Ceci permet à l'unité centrale de traitement (*ang.: central processing unit* ou *CPU*) d'y accéder bien plus rapidement que si ces instructions et données étaient stockées sur le disque.

Pour fixer les idées, changeons de perspective temporelle. Comparons le CPU à un travailleur qui effectue une tâche donnée en une seconde. Ce travailleur a besoin de quelques secondes pour prendre les outils et les pièces déposés dans le cache (semblable à son établi). Il a besoin d'une minute pour aller chercher dans la RAM les matériaux nécessaires à son travail (ce qui est déjà une belle perte de productivité). Mais il lui faut un mois pour se procurer des matériaux stockés sur le disque dur !

On comprend qu'une augmentation de la taille de la mémoire RAM permette d'améliorer très sensiblement les performances d'un ordinateur : si le stock du magasin local est grand, le CPU est moins soumis aux délais de livraison des fournisseurs.

Le problème est que la vitesse des processeurs augmente beaucoup plus vite au fil des générations que les performances de la mémoire RAM.

Ainsi, on considère que la vitesse des processeurs double en 18 ou 24 mois, ce qui correspond à un taux de croissance de 40 %/an environ. Par contre, le taux de croissance de la vitesse de la mémoire n'est que de l'ordre de 10 %/an. Autrement dit, l'écart devient de plus en plus important au fil du temps. A l'heure actuelle, les processeurs travaillent à des fréquences supérieures à 1 GHz alors que les mémoires classiques viennent à peine de dépasser 200 MHz ⁽¹⁾.

Aujourd'hui, ce sont les performances de la RAM qui brident les performances d'un PC et non la fréquence de son processeur.

5.2. Cellule de mémoire DRAM

La mémoire DRAM (*ang.: Dynamic Random Access Memory*) est composée d'un ensemble de cellules élémentaires (*ang.: cells*) capables de retenir chacune l'état d'un et un seul bit (0 ou 1).

Chaque cellule élémentaire, appelée cellule 1T/1C, comprend un condensateur et un transistor MOSFET ⁽²⁾. Les dimensions d'une telle cellule sont incroyablement réduites; la largeur est de l'ordre de 100 nm ($100 \cdot 10^{-9} \text{m}$), ce qui correspond à un alignement d'environ 1 000 *atomes*.

1 $1 \text{ GHz} = 10^9 \text{ s}^{-1}$. Un tel processeur effectue une opération élémentaire en 1 milliardième de seconde.

2 Metal Oxide Semiconductor Field Effect Transistor (Transistor à effet de champ à semiconducteur en oxydes métalliques). Grâce à une géométrie simple et une consommation très faible, c'est le composant élémentaire de tous les circuits intégrés à très grande échelle (VLSI).

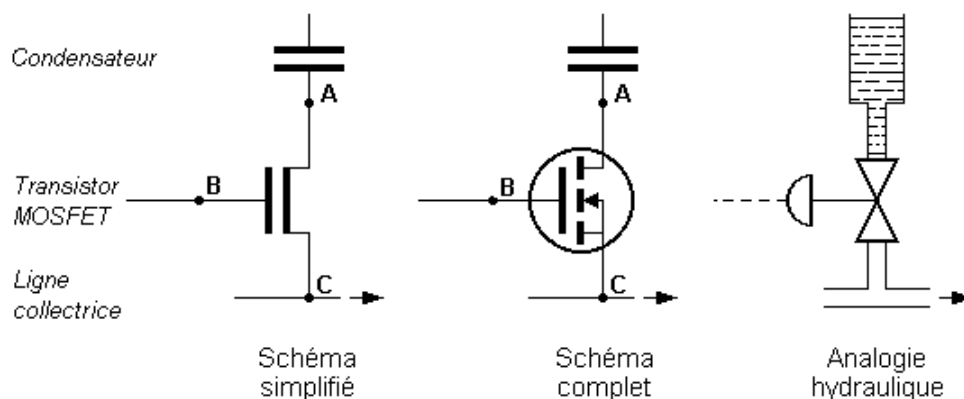


Figure 5.1 Cellule de mémoire RAM et analogie hydraulique

Le rôle du condensateur est de stocker une charge électrique. Si la cellule doit retenir la valeur 1, le condensateur est chargé; si la cellule doit retenir la valeur 0, le condensateur est déchargé. Vu la taille de la cellule, la capacité du condensateur est très faible : 30 fF ($30 \cdot 10^{-15} \text{F}$) ⁽¹⁾.

Ce condensateur est comparable à un réservoir que l'on remplit ou non avec une toute petite quantité de liquide selon qu'il doit retenir la valeur 1 ou 0.

Le transistor MOSFET joue le rôle d'interrupteur. Au repos, il coupe le circuit et isole le condensateur. Par contre, lorsqu'on applique une tension de l'ordre de 1 Volt entre la grille (B) et la ligne collectrice (C), le circuit se ferme. Le courant peut alors passer de A vers C... pour autant que le condensateur soit chargé.

Le transistor est comparable à une vanne qui reste ouverte tant que la commande est mise sous tension.

Si le réservoir est rempli (état 1), on observe l'apparition d'une pression et d'un débit sur la ligne collectrice lorsqu'on ouvre la vanne. Sinon (état 0), la pression dans la ligne collectrice reste inchangée et le débit est nul.

De même, si le condensateur est chargé, l'activation du MOSFET provoque l'apparition d'une tension en C ainsi que le passage d'un courant. Si le condensateur est vide, rien ne se passe.

Comme la capacité du condensateur est très petite, le courant collecté en C est extrêmement faible, de l'ordre de 30 μA ($30 \cdot 10^{-6} \text{A}$). La tension à ses bornes n'est maintenue que pendant une durée très courte, de l'ordre de 10 ns ($10 \cdot 10^{-9} \text{s}$).

Le gros avantage de la DRAM est qu'elle est très simple dans son principe, donc très économique à réaliser. De plus, la taille minuscule de la cellule permet de stocker un grand nombre de bits sur une très petite surface.

Son gros inconvénient est que la lecture est un processus destructeur. En d'autres termes, la simple lecture du bit suffit à effacer (épuiser) la mémoire. En effet, lors de l'activation du MOSFET, la charge du condensateur se dissipe dans la ligne collectrice. De même, tout le liquide du réservoir s'écoule lors de l'ouverture de la vanne. Après la lecture, tant le condensateur que le réservoir ont "oublié" la valeur qu'ils contenaient. Conséquence : une nouvelle lecture du même bit donnera invariablement "0".

1 Sachant que la charge de l'électron est de $1.6 \cdot 10^{-19}$ Coulomb et que la capacité C du condensateur est donnée par $C=q/V$, on déduit que ce condensateur est capable de stocker une charge équivalente à un maximum de 200 000 électrons sous une différence de potentiel de 1 Volts.

La solution consiste à recharger le condensateur ou à remplir le réservoir si on a détecté une tension ou un courant sur la ligne collectrice. Cette partie du fonctionnement - appelé rafraîchissement de la mémoire - n'est pas représentée sur le diagramme ci-dessus. Elle sera développée plus loin. Notons déjà que le processus de lecture est réversible : pour remplir le réservoir, il suffit d'ouvrir la vanne B et de refouler du liquide en appliquant une forte pression en C.

On comprend que, après chaque lecture d'une mémoire, il faudra attendre un certain temps pour laisser au système le temps de rafraîchir sa mémoire.

Le problème se complique encore du fait que, vu la finesse des couches atomiques, le système n'est pas parfaitement étanche. Le condensateur se décharge lentement (en quelques millisecondes) même si on n'active pas le transistor. C'est un peu comme si le réservoir de notre analogie hydraulique était percé. Il faudra dès lors penser à recharger périodiquement les condensateurs.

Ce sont ces cycles de charge/décharge qui ont valu le qualificatif de "dynamique" à ce type de RAM.

5.3. Cellule de mémoire SRAM

La cellule de mémoire SRAM (*ang.*: *Static Random Access Memory*) est beaucoup plus complexe et plus encombrante.

Cette cellule ne renferme aucun condensateur. Par contre, elle contient six transistors MOSFET. Deux d'entre eux servent d'interrupteurs de lecture; les quatre autres constituent une bascule bi-stable alimentée en permanence par le système. Comme l'alimentation est permanente, la phase de rafraîchissement n'est pas nécessaire et les lectures/écritures pourront se succéder à un rythme beaucoup plus rapide que dans une DRAM.

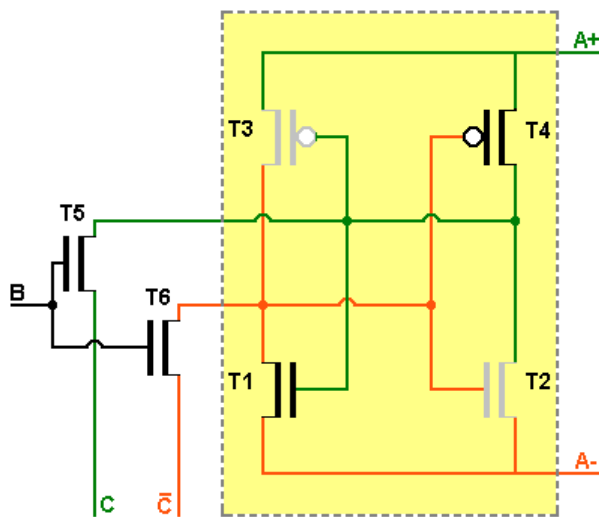


Figure 5.2a Cellule SRAM : Etat 0

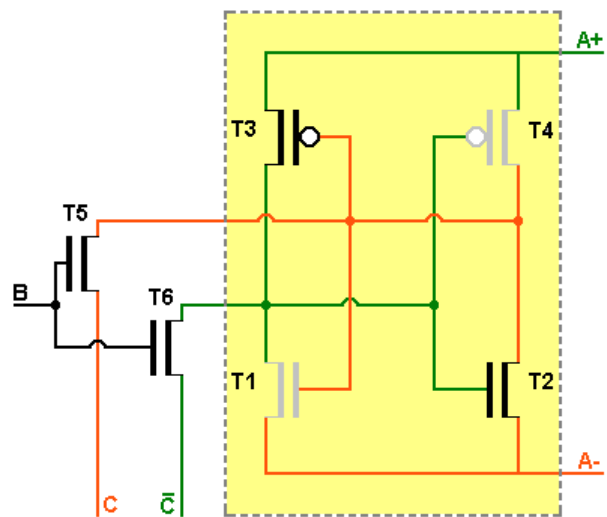


Figure 5.3b Cellule SRAM : Etat 1

Le fonctionnement de la cellule est illustré ci-après. Notons d'abord les transistors MOSFET T5 et T6 qui servent simplement à ouvrir le circuit de lecture/écriture lorsqu'une tension est appliquée en B. La partie entourée de pointillés contient les MOSFET T1 à T4 qui constituent la bascule proprement dite.

Les MOSFET T1 à T4 sont de types différents : T1 et T2 sont activés par une tension positive tandis que T3 et T4 sont activés par une tension nulle ou négative ⁽¹⁾.

1 Remarquez la présence du O sur la grille (activateur) des transistors T3 et T4.

Les transistors fonctionnent par paire : les activateurs (grilles) de T1 et T3 sont mis en commun, de même que les activateurs de T2 et T4. Comme chaque paire est formée de transistors de type différents, l'un est obligatoirement ouvert et l'autre obligatoirement fermé, quel que soit le signe de la tension appliquée à la ligne.

Quand la cellule contient la valeur 0, T1 et T4 sont fermés (conducteurs) et T2 et T3 sont ouverts (bloquants).

La tension d'alimentation venant de la ligne supérieure (A+) est bloquée par T3 mais peut traverser T4 (mais pas T2) et, grâce au point de connexion, atteindre le transistor de sortie T5. En cours de route, cette tension (+) rencontre un autre point de connexion et se reporte sur la grille de T3, qui se voit renforcé dans son rôle bloquant, ainsi que sur la grille de T1, ce qui confirme son rôle conducteur.

D'autre part, la tension d'alimentation (ou la masse) venant de la ligne inférieure (A-) est bloquée par T2 mais peut traverser T1 (mais pas T3) et, grâce au point de connexion, atteindre le transistor de sortie T6. Dans l'autre sens, cette tension (-) se reporte sur la grille de T4, ce qui renforce son rôle conducteur, et sur la grille de T2, qui se voit ainsi confirmé dans son rôle de blocage.

On trouve donc la **tension positive à la sortie de T5 (C)** et la **tension négative (ou nulle) à la sortie de T6 (/C)**.

Quand la cellule contient la valeur 1, c'est l'inverse : T1 et T4 sont ouverts (bloquants) tandis que T2 et T3 sont fermés (conducteurs). Cette fois, on retrouve la **tension négative (ou nulle) à la sortie de T5 (C)** et la **tension positive à la sortie de T6 (/C)**.

Comme dans le cas de la DRAM, le processus est réversible : il suffit d'ouvrir T5 et T6 et de "refouler" les tensions voulues en C et /C pour stabiliser la bascule dans l'une ou l'autre position.

Notons que le courant qui s'écoule dans la cellule est fourni par l'alimentation générale appliquée en A+/A-. Il n'y a donc aucun risque d'épuisement ! Par contre, si l'alimentation est coupée, les MOSFET se dépolarisent et l'information de la cellule est perdue.

D'un autre côté, comme le maintien de l'état des MOSFET nécessite une tension mais très peu de courant, les cellules SRAM dissipent peu de puissance. Elles seront donc utilisées chaque fois que la vitesse et l'économie d'énergie sont essentielles : en mémoire cache du processeur et dans les PC portables (*ang.*: *laptops*) Elles sont par contre très complexes et très onéreuses.

5.4. Structure et fonctionnement de la DRAM

5.4.1. L'organisation de la mémoire

Le cœur d'un chip de mémoire est une grille (*ang.*: *grid*) composé de lignes (*ang.*: *rows*) et de colonnes (*ang.*: *columns*). A l'intersection de chaque ligne et de chaque colonne, se trouve une cellule de mémoire capable de mémoriser 1 bit.

Dans le graphique ci-dessous, nous avons mis en évidence les quatre premières colonnes, constituées de fils verticaux numérotés de 0 à 3, ainsi que les quatre premières lignes, constituées de fils horizontaux numérotés de 0 à 3. Ces fils définissent 16 cellules de mémoire, soit une capacité de 16 bits.

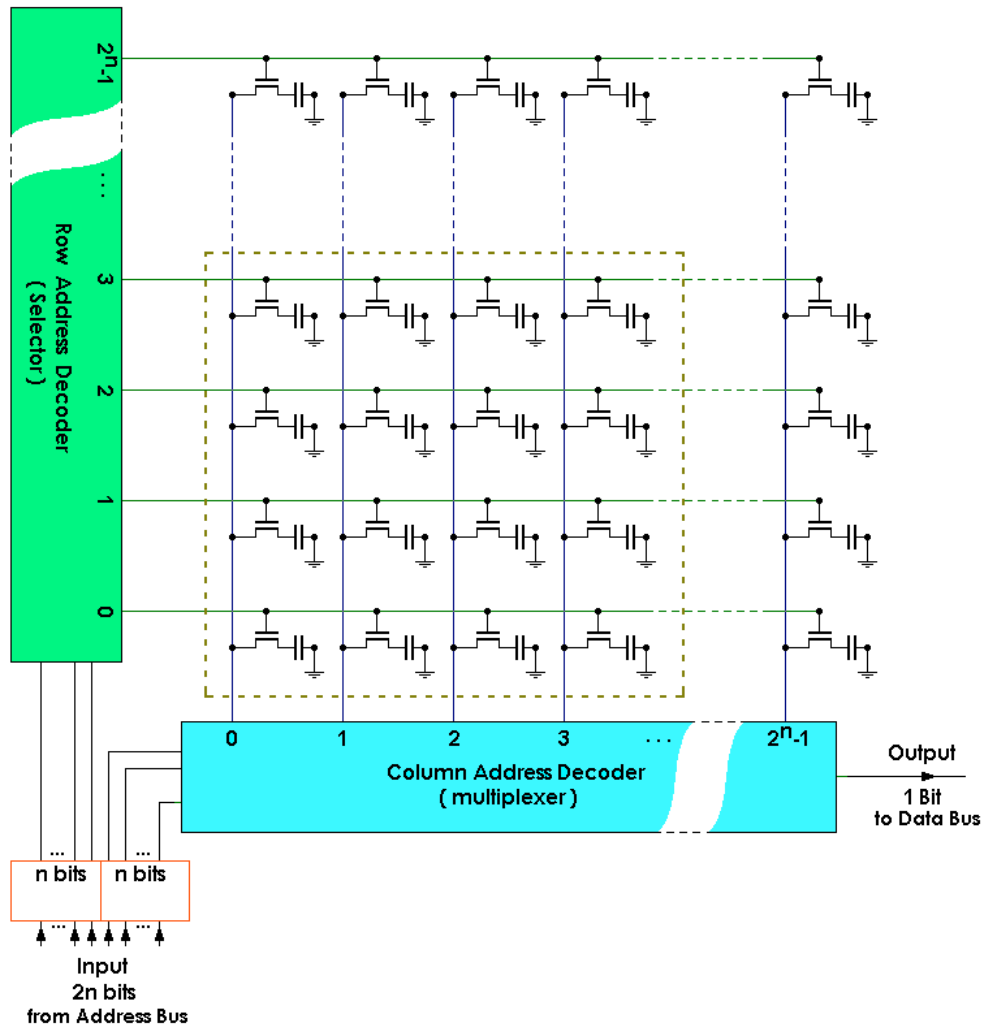


Figure 5.3 Principe de fonctionnement d'un chip de mémoire DRAM

Notez que, sur ce schéma, les fils des lignes et des colonnes se croisent mais ne sont pas connectés entre eux. Par contre, le fil de la ligne est connecté à l'activateur du MOSFET de la cellule de mémoire, tandis que le fil de la colonne collecte la charge éventuelle du condensateur. L'autre extrémité du condensateur est reliée à la masse.

En réalité, un chip de mémoire contient plusieurs centaines de fils verticaux et horizontaux. Pour simplifier le traitement, le nombre de lignes et de colonnes est généralement une puissance de 2. Un chip qui aurait 128 lignes et 128 colonnes (numérotées de 0 à 127) aurait une capacité de 16 384 bits, soit 16kbits ($2^7 \cdot 2^7$). On peut bien sûr imaginer d'autres géométries pour héberger 16 384 bits mais en pratique, on préfère utiliser une matrice carrée car les circuits annexes sont alors beaucoup plus simples ⁽¹⁾. Un tel chip est appelé mémoire 16k x 1 : 16k cellules de 1 bit chacune; le "1" caractérise la "profondeur" de la mémoire. Il existe aussi des chips ayant des profondeurs de 2, 4, ou 8 bits. Nous verrons plus loin comment ils sont organisés.

Le graphique suivant représente une petite zone de mémoire DRAM telle qu'elle apparaît au microscope électronique. Nous y avons mis en évidence les différentes structures. Celles-ci sont réparties sur plusieurs couches.

Les cellules de mémoires sont gravées tête-bêche : une à l'endroit, l'autre à l'envers, comme les figures d'un jeu de carte.

1 Dès lors, le chip de mémoire suivant dans la série aurait une capacité de $28 \cdot 28 = 65\,536$ bits, soit 64 kbits.

Les gros traits horizontaux représentent les conducteurs d'alimentation des cellules. Les traits horizontaux en créneau véhiculent les adresses des lignes. Les traits verticaux sont les conducteurs qui récoltent la charge des condensateurs. Les plaques rectangulaires sont les condensateurs.

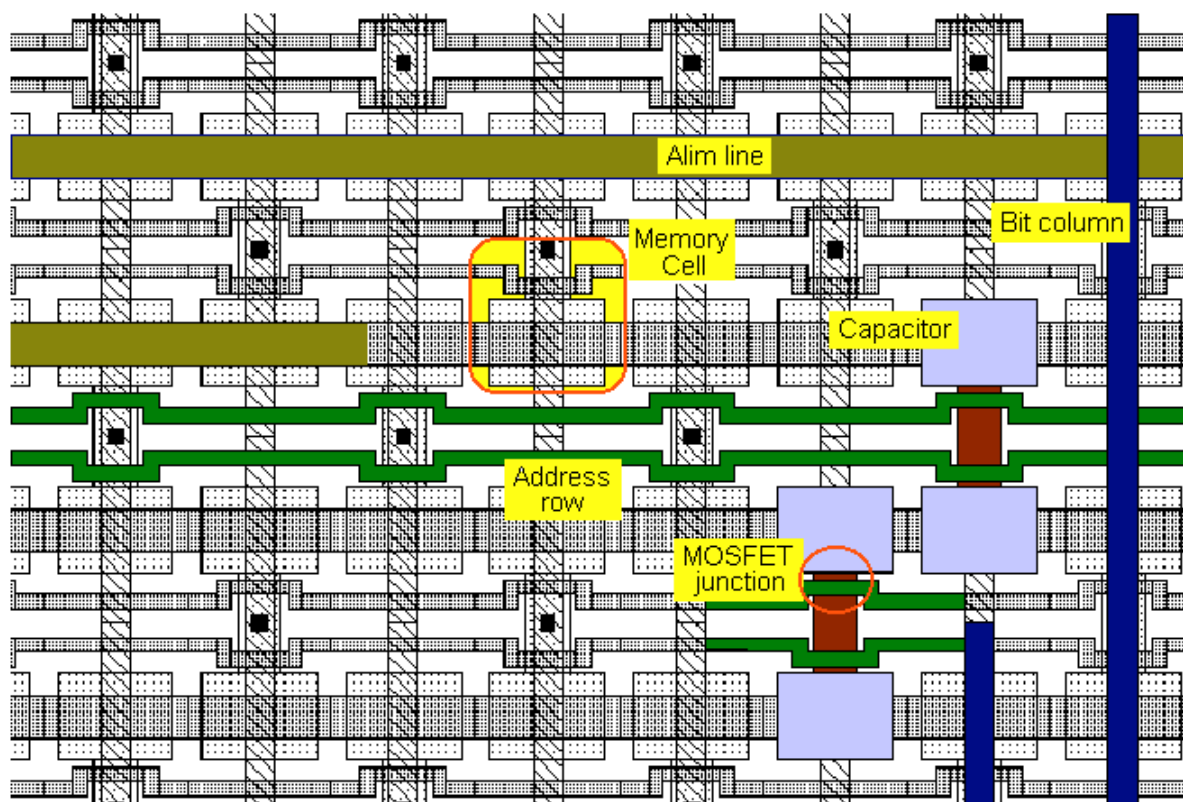


Figure 5.4 Vue partielle de la disposition d'un chip DRAM à cellules de mémoire alternées

5.4.2. La lecture de la DRAM

Chaque cellule de la mémoire est repérée par une adresse constituée du *numéro de la ligne* et du *numéro de la colonne* à laquelle elle est connectée. Pour accéder à une cellule, que ce soit en lecture ou en écriture, nous devons introduire cette référence *ligne-colonne* dans le bus d'adresses.

Comme la zone mise en évidence dans notre schéma contient 4 lignes et 4 colonnes, il nous suffit de 2 bits d'adresse pour spécifier la ligne et 2 bits d'adresse pour spécifier la colonne (¹). Chaque bit supplémentaire permet de doubler le nombre de lignes ou de colonnes de la matrice. La règle générale est qu'il faut n bits d'adresse pour coder un chiffre compris entre 0 et $2^n - 1$. Ainsi, un chip de 16 kbits de 128×128 cellules a besoin de 14 bits d'adresse : 7 pour le numéro de ligne et 7 pour le numéro de colonne.

Admettons que nous veuillons lire le contenu de la cellule située sur la ligne 3 et la colonne 2. Pour cela, nous envoyons l'adresse 1110 ($=14_d$) sur le port d'adresse.

En arrivant dans le chip, l'adresse est coupée en deux. Les deux bits supérieurs 11 ($=3_d$) (bits de poids fort) sont envoyés vers le décodeur de ligne tandis que les deux bits inférieurs 10 ($=2_d$) sont envoyés vers le décodeur de colonne.

Le décodeur de ligne met la ligne 3 sous tension, ce qui active *toutes* les cellules de cette rangée. Chacune de ces cellules s'empresse de délivrer la charge éventuelle

1 Deux bits offrent quatre possibilités de combinaison : 00 01 10 11 qui représentent les nombres de 0 à 3 (et non 1 à 4).

de son condensateur sur la colonne à laquelle elle est connectée. Il y a donc une série de bits (0 ou 1) qui arrivent au décodeur de colonnes. En d'autres mots, bien que l'on ne cherche à lire qu'une seule cellule, ce sont toutes les cellules de la ligne qui envoient leur contenu vers le multiplexeur.

Le rôle du multiplexeur consiste à connecter le fil dont on a donné l'adresse et à envoyer le bit qui y apparaît vers le data bus.

Tout ça pour lire *un* bit alors que la plus petite opération d'un programme impose d'en lire au moins huit !

5.4.3. L'amplification du signal

La tension provenant des cellules de mémoire est de l'ordre de 100 à 200 mV. Elle est tellement faible qu'elle doit être amplifiée avant d'être renvoyée sur le bus de données.

Le problème n'est pas simple car les condensateurs de chaque cellule sont tellement petits qu'on ne peut plus négliger les effets capacitifs dus à la proximité et au parallélisme des conducteurs des lignes et des conducteurs de colonnes.

Dans le même ordre d'idées, les capacités des condensateurs sont tellement faibles qu'il suffit parfois de l'impact d'un rayon cosmique pour modifier leur charge. De même, les matériaux qui constituent le circuit intégré contiennent des traces infimes de Thorium et d'Uranium. Ces éléments ont des isotopes radioactifs qui se décomposent en émettant des particules alpha ou des radiations ionisantes. Le rayonnement émis est infiniment faible mais il est néanmoins suffisant pour modifier la charge de l'un ou l'autre condensateur situé à proximité immédiate.

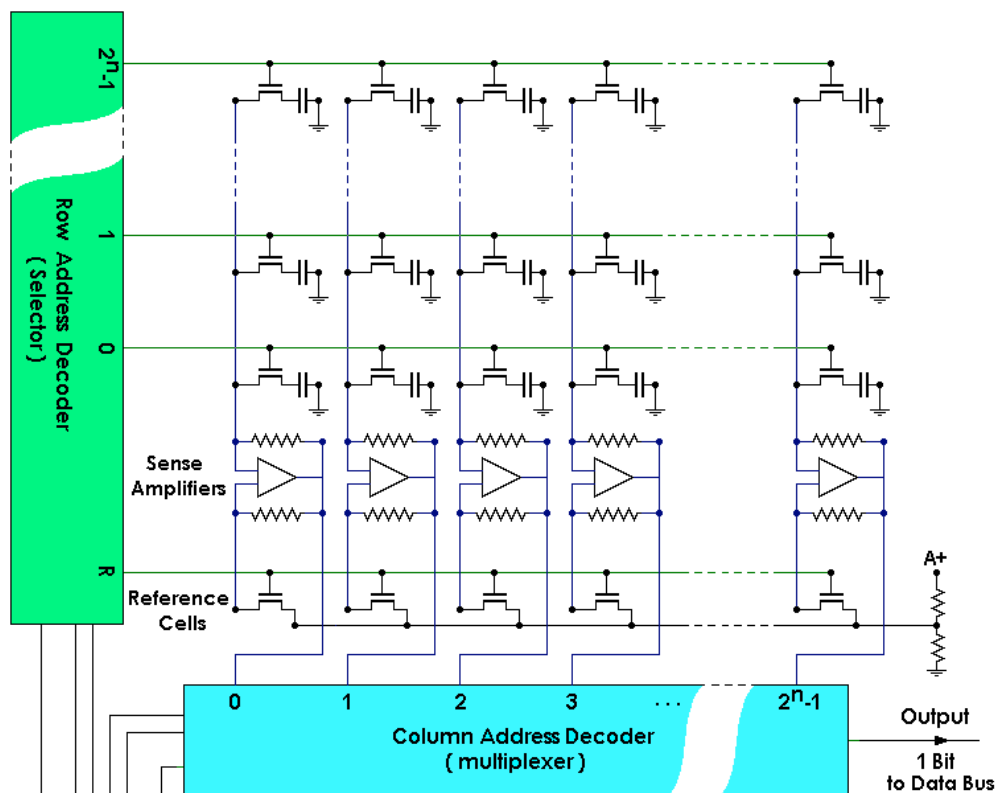


Figure 5.5 Cellules de références (A/2) et amplificateurs

En d'autres mots, on ne peut se fier à un niveau de charge bien défini pour fixer les valeurs 1 et 0.

Pour contourner le problème, on ajoute à la matrice une ligne de cellules de référence (notée R dans le diagramme). La tension sur cette ligne est exactement

égale à la moitié (p.ex. 0.5V) de celle que l'on observe normalement aux bornes d'un condensateur chargé (p.ex. 1.0V).

Quand veut procéder à une lecture, et la ligne des cellules de référence :

- le décodeur de ligne active la ligne de cellules choisies.
La tension venant des cellules de mémoire descend le long des colonnes pour arriver à l'entrée d'un amplificateur de sensibilité (ang.: sense amplifier).
- le décodeur de ligne active en même temps la ligne des cellules de référence.
La tension venant des cellules de référence (ang.: reference cells) remonte le long de la colonne et arrive à l'autre entrée de l'amplificateur
- L'amplificateur constate que la tension venant du condensateur est supérieure (1V) ou inférieure (0V) à la tension de référence. Il en déduit que la valeur stockée était 1 ou 0. Dans le cas "1", il envoie un signal amplifié (3.3 V ou 5 V) vers le multiplexeur des colonnes.

Grâce à cette technique, le système tolère des petites variations de charge autour des valeurs normales.

5.4.4. Le rafraîchissement

5.4.4.1 Rafraîchissement après lecture

Ainsi qu'il a été dit plus haut, le seul fait d'activer une ligne de cellules mémoire suffit à décharger tous les condensateurs de la ligne. Heureusement, la décharge n'est pas instantanée et le signal a le temps d'atteindre les amplificateurs de sensibilité avant de s'effacer.

Admettons que l'amplificateur ait perçu la valeur "1". Dans ce cas, il émet une tension élevée (3.3 V ou 5 V) et un courant dont une partie peut traverser la résistance supérieure et remonter dans la ligne de colonne (n'oublions pas que, la charge descendante s'étant dissipée entre-temps, rien ne s'oppose plus à ce phénomène). Sur son parcours, ce courant rencontre les transistors **MOSFET** de toutes les cellules de la colonne. Tous sont fermés sauf celui de la cellule qui vient d'être activée. Le courant s'y engouffre et recharge le condensateur de la cellule.

Admettons maintenant que l'amplificateur ait perçu la valeur "0". Dans ce cas, il n'émet aucune tension (0V). Aucun courant ne remonte la ligne et le condensateur de la cellule qui contenait le "0" reste déchargé.

Le même mécanisme se passe au même moment dans toutes les colonnes de la matrice. On peut donc dire que la lecture d'une ligne provoque le rafraîchissement de tous ses condensateurs.

Malheureusement, ce système ingénieux n'est pas instantané. La phase de rafraîchissement prend environ **15 ns**. Pendant ce temps, il est interdit de faire une nouvelle lecture. En effet, lire une autre ligne reviendrait à ouvrir un deuxième transistor **MOSFET** sur chaque colonne. Le courant remontant se disperserait alors dans deux cellules qui ne doivent pas nécessairement héberger la même valeur. Pour éviter tout problème, le contrôleur de mémoire utilise le bus de contrôle pour envoyer un signal au processeur en lui demandant de patienter quelques instants.

Notons au passage que le principe du rafraîchissement sera aussi utilisé pour écrire une valeur dans une cellule. La matrice fonctionne en sens inverse : la sortie (ang. *output*) vers le bus de données devient une entrée (ang.: *input*). Cette valeur entrée dans le multiplexeur remonte dans la colonne choisie et charge le condensateur indépendamment de ce qui s'y trouvait inscrit à l'instant précédent.

5.4.4.2 Rafraîchissement périodique

Nous avons déjà signalé que la cellule de mémoire n'est pas parfaitement étanche. A cause de la minceur des couches atomiques, les condensateurs se déchargent lentement même en absence de toute lecture.

En réalité, les condensateurs ne retiennent leur charge que pendant un temps de l'ordre d'une milliseconde. Passé ce délai, la charge tombe sous un niveau où les amplificateurs de sensibilité deviennent incapables de différencier une valeur "1" d'une valeur "0". Autrement dit, l'information est perdue ⁽¹⁾. Il est donc impératif de rafraîchir tous les condensateurs à titre préventif et à intervalles de temps réguliers.

Le principe de ce rafraîchissement périodique est très simple : puisque la lecture d'une cellule provoque l'épuisement et le rafraîchissement automatique de tous les bits de la ligne, il suffit de faire croire à la matrice de mémoire qu'on veut lire successivement toutes les lignes de son tableau.

En pratique, chaque chip de mémoire contient une horloge (*ang.: timer*) calée sur la fréquence critique du rafraîchissement (p.ex.: 1 kHz, soit un délai de 1 ms).

A chaque "top", le chip envoie un signal au processeur pour lui dire qu'il se met hors service. Il enclenche ensuite un compteur qui boucle sur toutes les lignes du tableau, activant chaque fois sa lecture et attendant chaque fois les 15 ns nécessaires au rafraîchissement.

Par exemple, dans une puce mémoire de 16 Mbits organisée en 4 096 lignes et 4 096 colonnes, le compteur activera successivement chacune des 4 096 rangées afin de rafraîchir toute la puce. On parle alors de puce "**4k refresh**". Le temps *minimum* pour accomplir ce travail est de 61 µs ($4\,096 \times 15\text{ ns}$). En réalité, il est beaucoup plus long car l'activation de ligne doit traverser le décodeur de ligne et il faut aussi laisser un certain temps de stabilisation avant de passer à la ligne suivante.

Lorsque la boucle est terminée, le chip envoie un signal au processeur pour lui annoncer qu'il est à nouveau disponible.

Evidemment, tout ce processus de rafraîchissement prend du temps et abaisse parfois considérablement les performances du système. C'est ce point qui est au cœur de toutes les améliorations apportées aux différentes versions des mémoires DRAM.

5.4.5. Lire 8, 16, 32 bits ou plus

Un tableau de mémoire DRAM ne renvoie qu'un bit à la fois vers le bus de données. Or, la plus petite unité de mémoire utilisée par les langages de programmation est le byte (8 bits). Devrons-nous collecter ces huit bits l'un après l'autre en interrogeant successivement plusieurs colonnes ?

La solution est plus simple et plus élégante : il suffit de mettre huit tableaux en parallèle sur le même décodeur d'adresse. Un tel ensemble est appelé **banc de mémoire**. Le principe est illustré sur la figure suivante :

L'adresse qui entre dans le registre d'adresse mémoire (*ang.: Memory Address Register*) est envoyée simultanément sur les décodeurs de ligne et sur les décodeurs de colonnes de huit tableaux. Chacun de ses tableaux renvoie un seul bit

¹ C'est ce qui se passe en cas de coupure de courant : plus d'énergie, plus de rafraîchissement, toute la mémoire est remise à 0.

mais chacun de ces bits est envoyé vers le bus de données par une ligne différente. Le bus de données reçoit donc huit bits en parallèle.

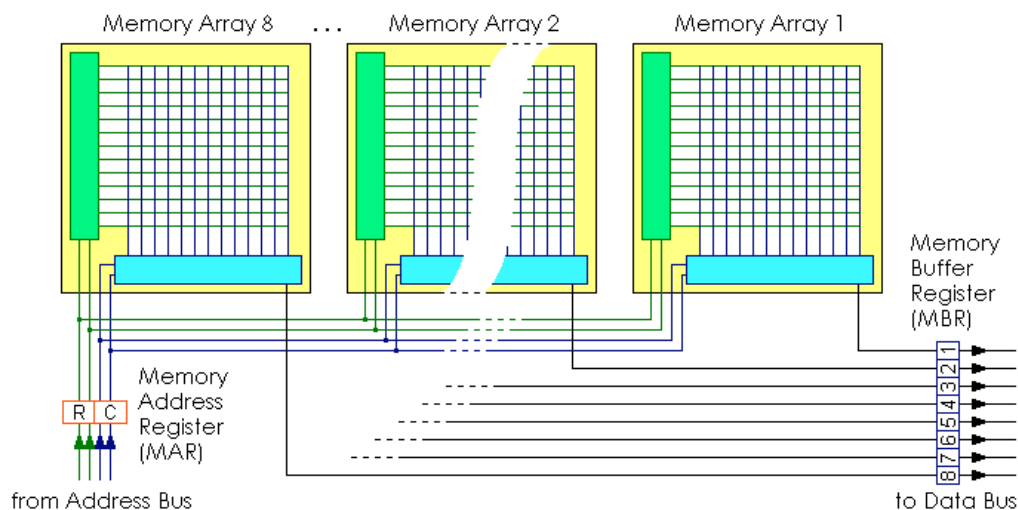


Figure 5.6 La mise en parallèle de huit tableaux renvoie 1 byte

En appliquant le même principe à des bancs de 8, 16, 32 tableaux, il est aisé d'alimenter un bus de données de 8, 16, 32 bits. Chaque fois que le nombre de tableaux mis en parallèle double, la bande passante du bus de données double également. En effet, la collecte de 32 bits en parallèle ne prend pas plus de temps que la collecte d'un seul bit.

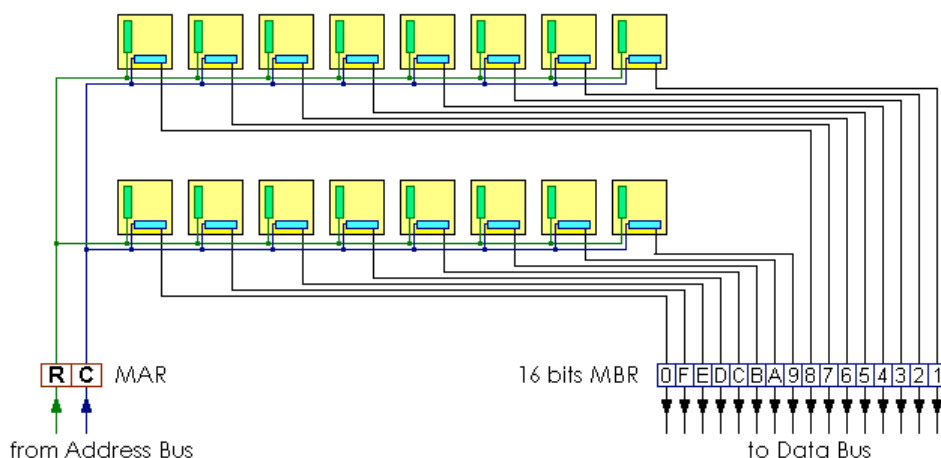


Figure 5.7 Mise en parallèle de plusieurs bancs

Notons aussi que l'élargissement du bus de données est absolument indépendant de la largeur du bus d'adresse.

5.4.6. La correction d'erreurs

Le principe de la mise en parallèle de plusieurs tableaux ouvre d'intéressantes possibilités en matière de correction d'erreurs. Comment savoir avec certitude que les données n'ont pas été altérées entre le moment de leur écriture et celui de leur lecture. Qu'est-ce qui nous assure qu'un rayon cosmique ou une particule alpha ne sont pas passés par-là ?

Certaines mémoires possèdent des mécanismes permettant garantir l'intégrité des données qu'elles contiennent, voire même de corriger les erreurs qui seraient survenues. Ce type de mémoire est généralement utilisé sur des systèmes travaillant sur des données critiques tels que les serveurs.

5.4.6.1 Le bit de parité

Le bit de parité est un bit supplémentaire qui, selon une convention, est mis à 1 si la somme des bits inscrits en mémoire est paire et est mis à 0 si la somme est impaire (parité paire)⁽¹⁾.

A titre d'exemple, si le byte à stocker est 1010 1110 ou 1101 0110, le bit de parité vaudra 0. Si le byte est 0000 0000 ou 1111 0011, le bit de parité vaudra 1.

Gérer un bit de parité est relativement simple : il suffit d'ajouter un neuvième (ou un 17^{ème} ou un 33^{ème}) tableau aux huit qui servent à conserver l'octet.

Lors du stockage, un simple circuit logique placé dans le chip définit la valeur du bit de parité à partir des huit bits fournis et la stocke dans le neuvième tableau. Lors de la lecture, un circuit similaire recalcule la valeur du bit de parité des huit bits lus et la compare avec la valeur stockée dans le neuvième tableau. Si les résultats correspondent, le byte lu est considéré comme correct; sinon, un message est envoyé, signalant qu'une erreur de mémoire a été détectée.

La technique du bit de parité permet de détecter un byte erroné mais il est incapable de le corriger. Impossible en effet de savoir quel bit a été modifié. Ironie du sort, il se pourrait que le byte soit correct et que ce soit le bit de parité qui ait été modifié. Bref, le bit de parité ne détecte pas *toutes* les erreurs mais il permet déjà d'en signaler quelques-unes.

Le gros inconvénient du bit de parité est qu'il exige une capacité de 9 Mb pour stocker 8 Mb de données "utiles", ce qui rend ces mémoires proportionnellement plus chères. Du temps des premiers PC pas trop fiables, les barrettes avec parité étaient assez fréquentes. Aujourd'hui elles ne sont quasiment plus installées que dans certains serveurs.

5.4.6.2 Le codage de correction d'erreur

En utilisant plusieurs bits de parité, il est possible non seulement de détecter les erreurs mais aussi de les corriger. Le principe est le même que celui des digits de contrôle qui servent à valider les numéros de carte d'identité ou de compte bancaire ⁽²⁾. Ce type de traitement est appelé "codage de correction d'erreur" (*ang.*: *Error Correction Coding* ou **ECC**).

Ces mémoires, utilisées principalement dans les serveurs, sont naturellement plus onéreuses que les mémoires classiques. Elles sont aussi plus lentes puisque l'exécution du contrôle et de la correction éventuelle prend "un certain temps".

5.4.7. L'augmentation de vitesse par entrelacement

L'entrelacement conduit à une réduction sensible des périodes pendant laquelle un chip de mémoire est indisponible. Cette technique est souvent retenue dans les grands systèmes tels que les serveurs et les stations de travail. Son principe est assez simple.

Les cellules mémoire à l'intérieur d'une puce DRAM sont divisées en deux ou quatre bancs indépendants, activables séparément.

1 Il va de soi que la convention inverse - parité impaire - est tout aussi valable. L'important est de se mettre d'accord sur la convention prise.

2 Dans un numéro de compte bancaire de la forme 123-4567890-XX, XX est le reste de la division de 1 234 567 890 par 97. On peut montrer que remplacer l'un quelconque des chiffres du numéro de compte par un autre chiffre quelconque conduit chaque fois à un digit de contrôle différent. Il est donc possible d'identifier le chiffre qui a été changé et de retrouver la valeur correcte.

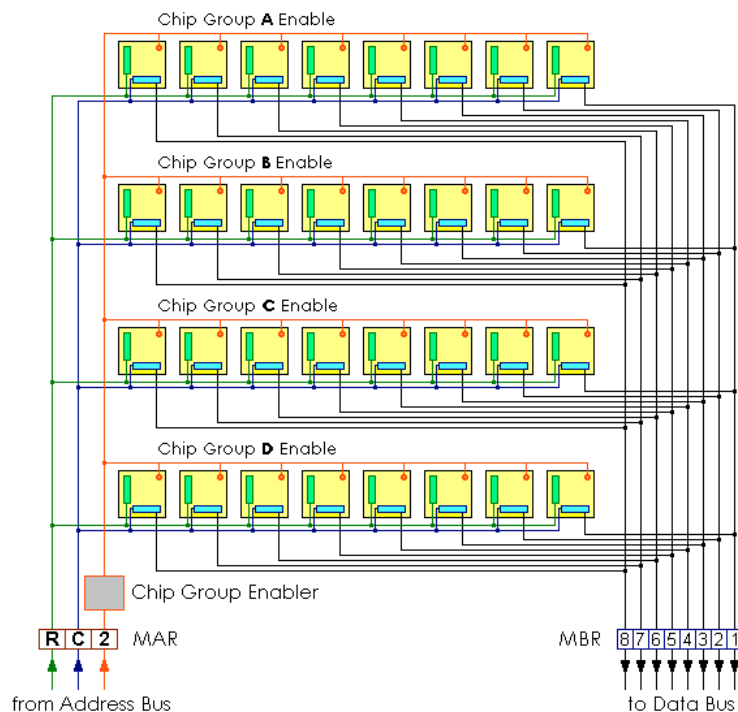


Figure 5.8 Système à quatre bancs entrelacés de huit bits

Chaque fois que le CPU a demandé une donnée à un banc de mémoire, celui-ci doit se mettre hors service pendant un certain temps afin de rafraîchir ses lignes.

Une manière intelligente de stocker les données est de faire appel à plusieurs bancs en parallèle et de pratiquer l'**entrelacement** : les données successives sont enregistrées dans les bancs successifs et non séquentiellement dans le même banc.

| <u>Bancs séquentiels</u> | <u>A</u> | <u>B</u> | <u>C</u> | <u>D</u> |
|---------------------------------|-----------------|-----------------|-----------------|-----------------|
| <i>Donnée</i> | 1 | 11 | 21 | 31 |
| | 2 | 12 | 22 | 32 |
| | 3 | 13 | 23 | 33 |
| | ... | ... | ... | ... |
| | 10 | 20 | 30 | 40 |
| <u>Bancs entrelacés</u> | <u>A</u> | <u>B</u> | <u>C</u> | <u>D</u> |
| <i>Donnée</i> | 1 | 2 | 3 | 4 |
| | 5 | 6 | 7 | 8 |
| | 9 | 10 | 11 | 12 |
| | 13 | 14 | ... | ... |
| | ... | ... | 39 | 40 |

Dans une configuration entrelacée, le CPU n'est pas (souvent) pénalisé par l'arrêt d'un banc. Il peut directement demander la donnée suivante au banc suivant. L'entrelacement permet d'obtenir un flux de données pratiquement continu. Cela écourte le cycle mémoire total et autorise des transferts de données plus rapides.

Evidemment, il se pourrait que le CPU doive précisément accéder à deux données situées dans le banc "A", auquel cas il n'y aurait aucun gain. Mais, en vertu du principe de localisation, on sait que le processeur accède généralement à des données (ou des instructions) qui se suivent.

L'entrelacement est géré soit au niveau de la barrette de mémoire, qui dispose d'un circuit (*ang.*: *Chip Group Enabler*) qui active le chip désiré en fonction de l'adresse qui lui est fournie, soit par le contrôleur de mémoire qui répartit les données sur différentes barrettes.

5.4.8. L'adressage par "strobe" (RAS / CAS)

Considérons un chip de 256 Mbytes structuré en 256 M x 8 bits, quelle doit être la largeur du bus d'adresses ?

Comme 1 byte contient 8 bits et que la profondeur de la mémoire est précisément de 8 bits, le chip est composé de 8 tableaux de 256 Mbits. Chaque tableau contient donc 268 435 456 éléments ($256 \times 1024 \times 1024 = 256 \text{ M}$). Les éléments étant disposés dans une matrice carrée, la matrice contient 16 384 lignes et 16 384 colonnes numérotées de 0 à 16 383.

Comme $16\,384 = 2^{14}$, le bus d'adresse doit disposer de 14 bits pour numéroté toutes les lignes et 14 bits pour numéroté toutes les colonnes, soit 28 bits. C'est beaucoup, mais du côté de la carte mère, ça ne pose pas vraiment de problème.

Le problème est du côté du chip : pour lire les 28 bits du bus d'adresses, le chip doit disposer de 28 pattes (*ang.*: *pins*). Ajoutez à cela les 8 pattes du bus de données, quelques pattes pour le bus de contrôle et deux pattes d'alimentation électrique, ça commence à faire beaucoup ! A quoi bon essayer de miniaturiser la mémoire si le nombre de pattes qui devient de plus en plus encombrant impose un gros volume ? Bref, pour réduire la taille du chip, il faut impérativement réduire le nombre de pattes.

L'astuce consiste à multiplexer le numéro des lignes et le numéro des colonnes sur *les mêmes* pattes. En d'autres mots, on utilise 14 pattes pour envoyer le numéro de la ligne et les mêmes 14 pattes pour envoyer le numéro de la colonne.

Pour que le chip puisse s'y retrouver, on envoie aussi deux signaux par le bus de contrôle. Le premier, appelé **RAS** (*ang.*: **Row Address Strobe**) signale qu'on a envoyé le numéro de la ligne; le second, appelé **CAS** (*ang.*: **Column Address Strobe**) est envoyé sur une autre ligne du bus de contrôle et signale qu'on vient d'envoyer l'adresse de colonne. Le processus sera détaillé au paragraphe suivant.

Avec cette technique, le chip de 256 MB n'a donc plus besoin que de

- 14 pins d'adresse,
- 2 pins de contrôle RAS et CAS,
- 8 pins de données,
- 2 pins d'alimentation électrique.

C'est pratiquement la moitié de ce qui était nécessaire antérieurement. La taille du chip peut donc être réduite de moitié. Le prix à payer : on passe d'un processus en parallèle à un processus en série, ce qui est souvent une perte de temps.

5.4.9. La séquence de lecture / écriture

5.4.9.1 Fonctionnement en lecture

Le graphique ci-dessous représente la suite des opérations de lecture :

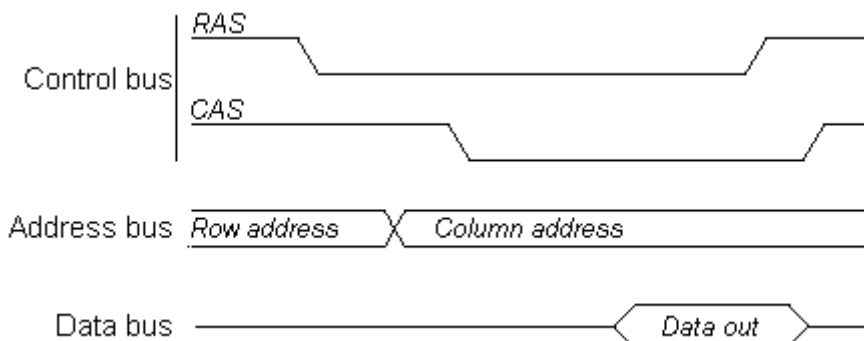


Figure 5.9 Séquence de lecture

- le contrôleur envoie et maintient l'adresse de **ligne** (*ang.: Row address*) sur le bus d'adresses,
- il active la ligne **RAS** du bus de contrôle et laisse passer quelques nanosecondes,
- il envoie et maintient l'adresse de **colonne** (*ang.: Column address*) sur le bus d'adresse,
- il active la ligne **CAS** du bus de contrôle et laisse passer quelques nanosecondes,
- le chip DRAM envoie et maintient les **données** sur le bus d'adresse (*ang.: Data out*),
- le contrôleur supprime le signal RAS pour indiquer que les données sont disponibles,
- le processeur lit les données disponibles sur le bus de données,
- le contrôleur supprime le signal CAS pour clôturer l'opération de lecture.

5.4.9.2 Fonctionnement en écriture

Le processus d'écriture est presque identique :

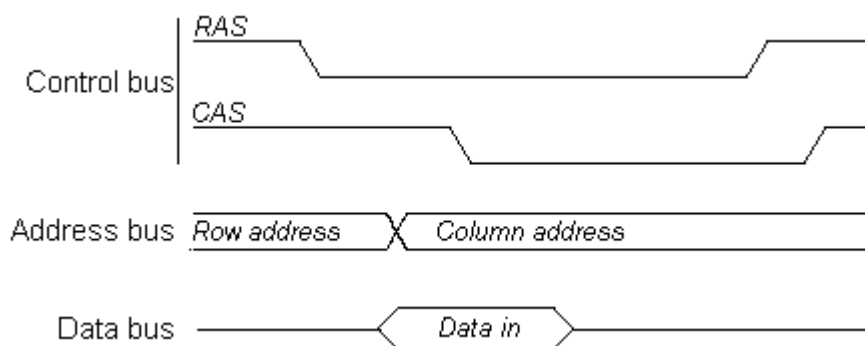


Figure 5.10 Séquence d'écriture

- le contrôleur envoie et maintient l'adresse de ligne (*ang.: Row address*) sur le bus d'adresses,
- il active la ligne RAS du bus de contrôle et laisse passer quelques nanosecondes,
- le processeur envoie les données à mémoriser sur le bus de données (*ang.: Data in*),
- le contrôleur envoie et maintient l'adresse de colonne (*ang.: Column address*) sur le bus d'adresse,
- il active la ligne CAS du bus de contrôle et laisse passer quelques nanosecondes,
- les données sont "absorbées" par le chip DRAM,
- le contrôleur supprime le signal RAS pour indiquer la fin de l'écriture,
- il supprime le signal CAS pour clôturer l'opération.

5.4.9.3 Fonctionnement en rafraîchissement

Nous avons vu qu'une cellule de mémoire DRAM ne conserve les données que pendant une milliseconde environ. Elle doit être rafraîchie avant l'expiration de ce délai. D'un autre côté, nous avons vu que, grâce aux amplificateurs de sensibilité (*ang.: sense amplifiers*) le rafraîchissement était automatique après chaque lecture.

Il suffit donc de faire croire au chip qu'on veut lire une de ses données :

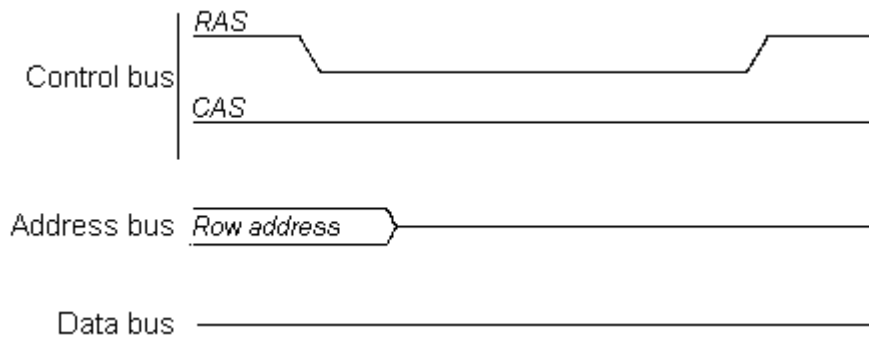


Figure 5.11 Séquence de rafraîchissement

- le contrôleur envoie et maintient l'adresse de ligne (*ang.: Row address*) sur le bus d'adresses;
- il active la ligne RAS du bus de contrôle. Comme dans les cas précédents, ceci provoque immédiatement l'ouverture de tous les transistors CMOS de la ligne. Les charges électriques arrivent dans tous les amplificateurs et la recharge de tous les condensateurs qui contenaient la valeur "1" est activée;
- le contrôleur n'a pas besoin d'envoyer une adresse de colonne puisque toutes les colonnes ont été rafraîchies à l'étape précédente;
- de même, il n'est pas nécessaire d'activer la ligne CAS du bus de contrôle,
- de même, le bus de donnée reste vide car aucune donnée ne doit être réellement lue ou écrite.
- après un temps nécessaire au rafraîchissement de la ligne de mémoire, le contrôleur supprime le signal RAS pour indiquer la fin de l'opération.

5.4.10. Le timing

Dans les paragraphes précédents, nous avons détaillé le fonctionnement de la mémoire DRAM et la succession des opérations nécessaires à la lecture d'une série de bits. Vous aurez compris que l'accès au byte choisi est loin d'être instantané !

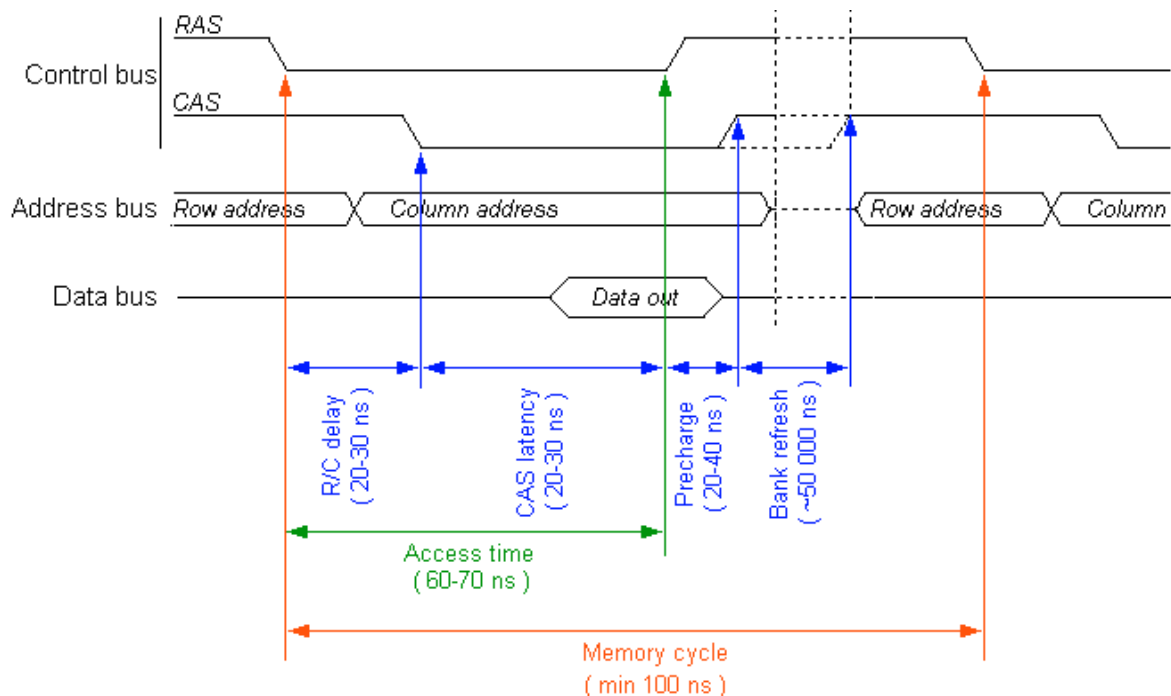


Figure 5.12 Timing de la DRAM



Figure 5.13 Puce de DRAM-EDO.

Le chiffre 60 dans la référence indique un temps d'accès de 60 ns

Commençons par distinguer deux concepts usuels :

- **le temps d'accès** (ang.: *access time*)
Temps qui s'écoule entre l'introduction d'une demande de lecture (ou écriture) via le bus d'adresse (strobe RAS) et le moment où les données sont disponibles sur le bus de données (fin de strobe RAS).
Pour une DRAM classique, le temps d'accès est de 70 ns. Il tombe à 60 ns pour une DRAM-EDO, à 10 ns pour une SDRAM et 5 à 6 ns pour une DDR-SDRAM.
- **le cycle de mémoire** (ang.: *memory cycle*)
Temps qui doit s'écouler entre deux accès successifs à la mémoire.
Typiquement, le cycle de mémoire est de l'ordre de 100 ns pour une DRAM classique.

Le temps d'accès comprend deux phases :

- **le délai ligne / colonne** (ang.: *Row/Column delay*) ou délai de cycle
Temps qui sépare l'envoi du numéro de ligne de l'envoi du numéro de colonne.
Ce temps est de deux cycles d'horloge, soit de l'ordre de 20 à 30 ns pour une DRAM classique.
- **le temps de latence CAS** (ang.: *CAS latency time*)
Délai qui s'écoule entre le moment où le processeur envoie le numéro de colonne par le bus d'adresse (*signal CAS*) et le moment où les données sont disponibles dans le bus de données. De fait, l'adresse venant du bus doit percoler dans les circuits de décodage du numéro de ligne et du numéro de colonne, ce qui prend "un certain temps"
En général, le temps de latence CAS dure deux ou trois cycles d'horloge (CAS 2 ou CAS 3) ce qui correspond à 20 ou 30 ns sur un bus à 100 MHz mais tombe beaucoup plus bas dans le cas de la SDRAM.

Bien que les barrettes de mémoire mentionnent généralement le temps d'accès, en pratique, c'est le cycle de mémoire qui est la donnée importante.

En effet, après chaque lecture, un **temps de précharge** est nécessaire pour recharger les condensateurs, rafraîchir la ligne de cellules et se préparer à une nouvelle lecture. Ce temps est loin d'être négligeable; il est de l'ordre de 20 à 40 ns selon le type et l'organisation de la mémoire.

Or, nous savons que les condensateurs ne retiennent leur charge que pendant un temps de l'ordre d'une milliseconde. Dès lors, à ce temps de précharge, il faut ajouter de temps à autres un **temps de rafraîchissement du banc** de mémoire. C'est ici que se trouve l'étape limitante du processus car le rafraîchissement du banc prend énormément de temps : environ 50 μ s ou 50 000 ns pour un chip "4k refresh".

Il découle de ce décompte que la vitesse de la mémoire est un facteur essentiel de la performance de l'ordinateur.

Naguère, un ordinateur qui travaillait avec une fréquence d'horloge élevée (p.ex. 200 Mhz) et des mémoires lentes (temps d'accès de 60ns), devait effectuer une

dizaine de cycles d'attente (*ang.*: *wait state*) chaque fois qu'il voulait accéder à la mémoire.

L'apparition de processeurs à fréquence très élevée, supérieure à 1 GHz, a imposé l'usage de la SDRAM, plus rapide. Notez cependant que le premier accès est encore toujours beaucoup plus long que les accès suivants. De plus, le processus de rafraîchissement du banc est toujours nécessaire et constitue encore une étape limitante.

5.5. Puces de mémoire

5.5.1. Description générale

Les tableaux de mémoire et les circuits annexes sont gravés sur une plaquette de silicium qui est ensuite placée dans un boîtier hermétique. Celui-ci est formé d'un matériau isolant, généralement un polymère (dérivé de la bakélite, polymère fluoré, etc.)

Le boîtier est pourvu de pattes (*ang.*: *pins*) extérieures qui récupèrent les informations en provenance du bus d'adresse et du bus de contrôle et qui renvoient les bits lus dans les tableaux vers le bus de données. Ce sont de minuscules fils d'or qui assurent les connexions entre les pattes du chip et les circuits gravés dans le silicium.

L'un des petits côtés du boîtier présente une encoche (*ang.*: *notch*). La patte numéro 1 est toujours située à gauche de l'encoche. Les autres pattes sont numérotées dans le sens inverse des aiguilles d'une montre.

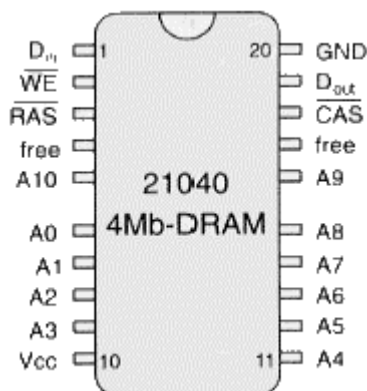


Figure 5.14 Puce DRAM de 4 Mbits et attribution des pattes

Il n'y a pas de règle stricte pour l'usage attribué aux différentes pattes d'une puce (*ang.*: *pin assignment*). Il faut toujours se reporter à la notice technique. Ces notices sont assez facilement accessibles via Internet lorsque l'on connaît le nom du fabricant et le numéro de référence de la puce. A titre d'exemple, voici le rôle attribué aux pattes de la puce représentée ci-dessus :

- **Pin 10 = Vcc (Input)**
Fournit la tension nécessaire au fonctionnement de la puce (généralement +5 V ou +3.3 V).
- **Pin 20 = GND**
Sert de mise à la masse (généralement 0 V).
- **Pins 5-9, 11-16 = A0 - A10 (Input)**
11 pattes qui fournissent le numéro de ligne et de colonne de la cellule mémoire à laquelle on veut accéder. Ici, on peut adresser 2^{11} lignes et 2^{11} colonnes, soit $2\ 048 \times 2\ 048 = 4\ \text{M}$ adresses.

- **Pin 3 = RAS (Input)**
Lorsqu'un signal est envoyé sur cette patte, la DRAM lit les informations fournies sur les 11 pattes d'adresses et les utilise comme numéro de ligne.
- **Pin 18 = CAS (Input)**
Lorsqu'un signal est envoyé sur cette patte, la DRAM lit les informations fournies sur les 11 pattes d'adresses et les utilise comme numéro de colonne.
- **Pin 19 = Dout (Output)**
Renvoie le bit lu dans la DRAM lors d'une demande de lecture.
- **Pin 1 = Din (Input)**
Reçoit la donnée à inscrire dans la DRAM lors d'une demande d'écriture. C'est un cas particulier; d'habitude, les données utilisent la même patte en écriture et en lecture.
- **Pin 2 = WE (Input)**
Lorsque le processeur désire écrire une donnée dans la DRAM (*ang.: write enable*), il envoie un signal sur cette patte. Sinon, la DRAM comprend qu'il s'agit d'un ordre de lecture.

5.5.2. Deviner la capacité ?

Est-il possible de deviner la capacité d'une puce de mémoire quand on ne dispose pas de la notice technique ?

La réponse est non. Tout au plus peut-on estimer la capacité maximale de la puce si toutes les pattes étaient utilisées dans la meilleure combinaison possible.

Reprenons la puce ci-dessus. Elle est dotée de 20 pattes. Retirons les deux pattes d'alimentation Vcc et GND, les deux pattes de RAS et CAS, et une patte d'écriture WE. Il nous reste 15 pattes à répartir entre bus d'adresses et bus de données, ce qui donne les combinaisons suivantes :

| Adresse | Données | Tableaux | bits | bytes |
|---------|---------|---------------------------------|-------|-------|
| 14 | 1 | $2^{14} \times 2^{14} \times 1$ | 256 M | 32 M |
| 13 | 2 | $2^{13} \times 2^{13} \times 2$ | 128 M | 16 M |
| 11 | 4 | $2^{11} \times 2^{11} \times 4$ | 16 M | 2 M |
| 7 | 8 | $2^7 \times 2^7 \times 8$ | 128 k | 16 k |

La capacité de la puce ci-dessus ne peut dépasser 32 Mbytes. Or, elle ne fait que 4 Mbits. Sans notice technique, il nous est impossible de deviner que deux pattes sont inutilisées et que la lecture et l'écriture se font sur des pattes différentes...

5.5.3. Puces DIP (dual in-line package)

Les premiers circuits intégrés se présentaient sous la forme de boîtiers DIP (*ang.: Dual in-line package*). Ces boîtiers avaient une forme rectangulaire et étaient pourvus de deux rangées de longues pattes.

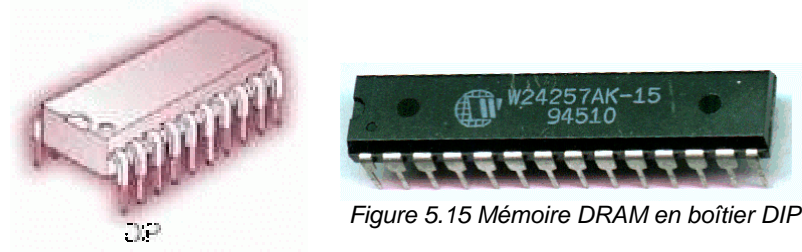


Figure 5.15 Mémoire DRAM en boîtier DIP

Les longues pattes s'inséraient dans une embase qui, elle, était directement soudée à la carte mère.

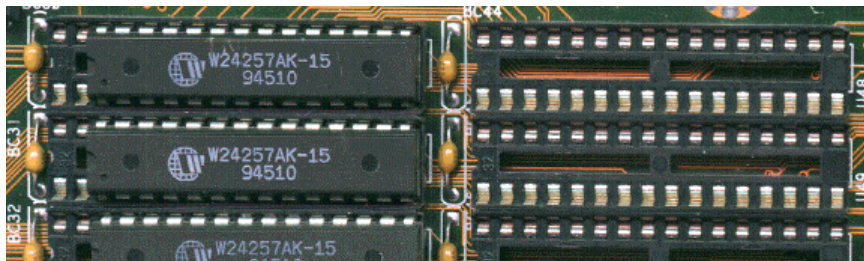


Figure 5.16 Embase occupée par une DRAM de type DIP et embase libre

Ce système a été en vogue jusqu'au moment où l'espace sur la carte est devenu un problème. On a alors trouvé la solution de souder les puces mémoire sur une petite carte modulaire, nommée barrette, enfichable dans un connecteur prévu à cet effet.

5.5.4. Puces SOJ (small outline j-lead)

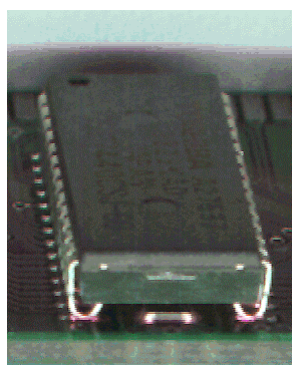
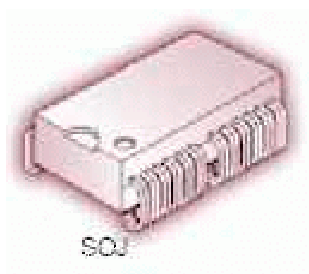


Fig 5.17 Chip SOJ sur une barrette DRAM

Les chips de mémoire plus récents sont montés directement à la surface de la barrette de mémoire. Il s'agit de boîtiers SOJ (*ang.: small outline j-lead*) qui doivent leur dénomination à leurs pattes en forme de "J". Cette forme a deux avantages : elle assure une certaine élasticité et permet une meilleure dissipation de la chaleur.

5.5.5. Puces TSOP (thin small outline package)

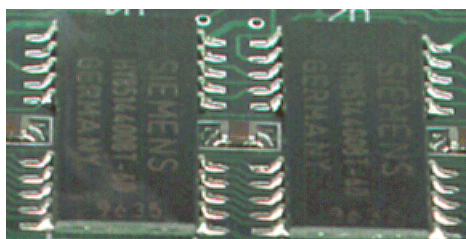


Fig 5.18 Chip TSOP sur une barrette SDRAM

Le boîtier TSOP (*ang.: thin small outline package*), est un concept plus récent pour le montage en surface. Il doit son nom à son boîtier bien plus mince (*ang.: thin*) que celui du SOJ. Les boîtiers TSOP ont été initialement utilisés pour confectionner les modules minces destinées aux micro-ordinateurs portables et aux cartes de crédit. L'épaisseur étant plus mince, le gradient de température dans la puce est également réduit.

5.5.6. Puces CSP (chip scale package)

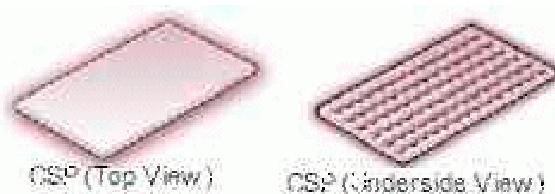


Figure 5.19 Puces CSP (Rambus)

Contrairement aux boîtiers DIP, SOJ, et TSOP, le boîtier CSP n'utilise pas de broches pour connecter le chip. Au lieu de cela, les connexions électriques traversent un réseau de billes BGA (ang.: *Ball Grid Array*) monté sur la partie inférieure du boîtier. Les chips de Rambus DRAM sont configurés selon ce principe.

5.6. Barettes de mémoire

5.6.1. Description générale

Aujourd'hui, la mémoire vive se présente presque toujours sous la forme de barrettes amovibles qui se clipsent dans des connecteurs spécifiques de la carte mère. Cette technique a permis de réduire considérablement l'encombrement de la mémoire.

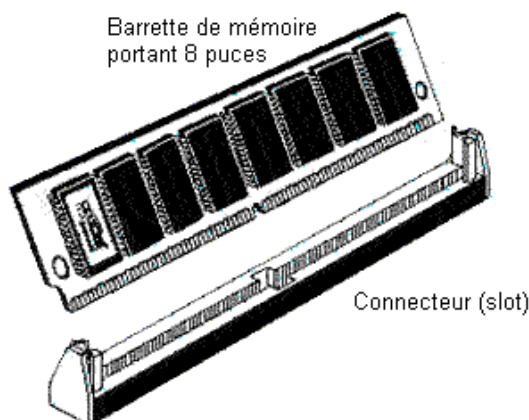


Figure 5.20 Barrette de mémoire et connecteur

Les connecteurs sont aisément identifiables. De plus, comme le transport des informations entre la mémoire et le processeur est un facteur critique pour les performances d'une machine, ces connecteurs sont presque toujours situés à proximité du CPU.

La barrette de mémoire se compose d'une **carte a circuit imprime** (ang.: *Printed circuit board*) sur laquelle les puces de mémoire sont enfichées et soudées.

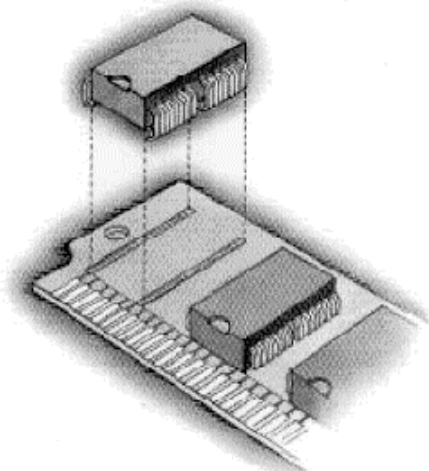


Figure 5.21 Enfichage des chips sur une barrette

La carte comprend souvent **plusieurs couches** (ang.: *multi-layers*). Chaque couche supporte les pistes et les circuits qui assurent le transfert de données. La largeur, la longueur, l'écartement et la courbure de ces pistes affectent à la fois la vitesse et la fiabilité du module complet. Ce sont des concepteurs expérimentés qui agencent ou et configurent les pistes afin de réduire les phénomènes d'interférence.

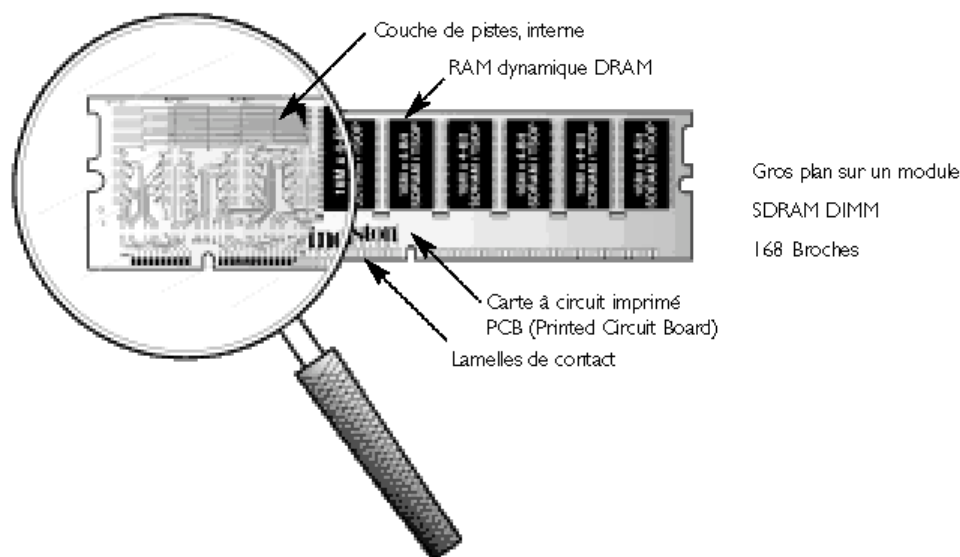


Figure 5.22 Constitution d'une barrette

En règle générale, les modules de mémoire de haute qualité utilisent des cartes ayant un grand nombre de couches. Plus le nombre de couches d'une carte est élevé, plus l'espace entre pistes est important ce qui réduit d'autant les risques d'interférences.

Les **puces** (ang.: *chip*) soudés sur barrettes de mémoire sont presque toujours de type DRAM. En effet, ainsi qu'il a été dit plus haut, les mémoires SRAM, plus rapides mais beaucoup plus chères sont réservées aux caches.

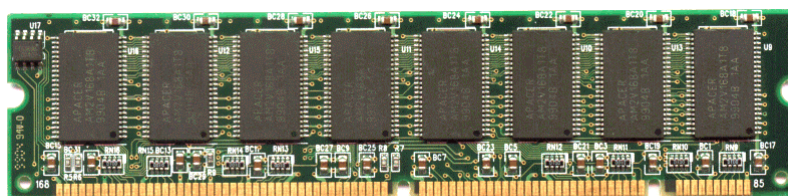


Figure 5.23 Barrette (taille réelle)

Le bas de barrette de mémoire est pourvu d'un certain nombre de **lamelles de contact** parfois appelées "connecteurs" ou "broches". Ce sont elles qui assurent le transfert de l'information (address bus, data bus, control bus) entre la carte mère et les différentes puces. Le *nombre de contacts* est caractéristique de l'organisation de la mémoire ainsi qu'on le verra au point suivant.

Sur certaines barrettes, les conducteurs sont en étain, sur d'autres, ils sont en or. L'or a l'avantage d'être un excellent conducteur qui ne s'oxyde pas. L'étain a l'avantage d'être beaucoup moins cher mais il a tendance à s'oxyder. Ce petit problème est aisément résolu en frottant les connecteurs avec une gomme avant de les insérer dans le support. Il est conseillé de choisir des barrettes dont le métal est identique à celui du support. On évitera ainsi d'éventuels couples électrochimiques qui pourraient produire des tensions parasites.

5.6.2. Disposition des puces

On peut distinguer les barrettes de RAM en fonction de la position des puces. Il existe deux géométries : **SIMM** et **DIMM**. Notez bien que des deux appellations ne

concernent que la *disposition* des puces et n'ont aucun rapport avec le *fonctionnement* de la mémoire.

Sur les barrettes de format **SIMM** (ang.: **Single Inline Memory Module**), les puces de mémoire se trouvent sur une seule face. Ces barrettes se déclinent en plusieurs versions en fonction du nombre de broches : 30, 72 ou 168.

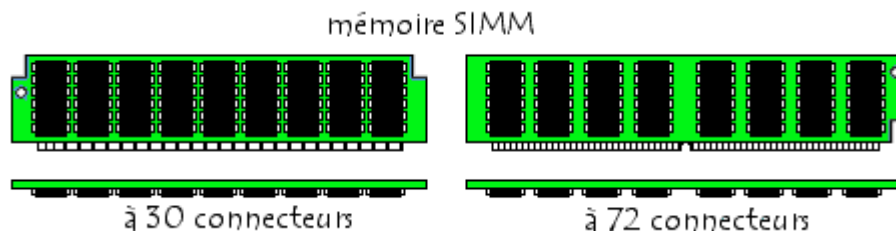


Figure 5.24

Sur les barrettes de format **DIMM** (ang.: **Dual Inline Memory Module**), les puces de mémoire et les contacts se trouvent sur les deux faces. Ces barrettes se déclinent également en plusieurs versions en fonction du nombre de broches : 72, 168, etc.

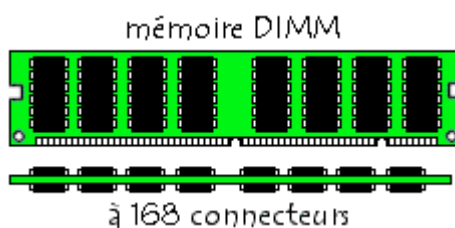


Figure 5.25

5.6.3. Nombre de broches

C'est le nombre de broches qui est caractéristique du *fonctionnement* de la mémoire ⁽¹⁾.

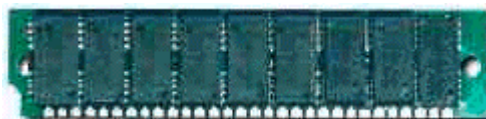


Figure 5.26 DRAM PM 30 broches avec bit de parité (9 chips)

Les barrettes à **30 broches** sont des mémoires **DRAM PM** de **8 bits** qui équipaient les premières générations de PC (286, 386) et les imprimantes. Certaines, comme celle de l'illustration, conservaient un 9^{ème} bit de parité. Leur capacité allait de 256 kBytes à 8 Mbytes selon le modèle. Lorsque les PC ont commencé à utiliser des bus de 32 bits, il était encore possible d'utiliser ces barrettes, à condition d'en placer dans les quatre slots de la carte mère.

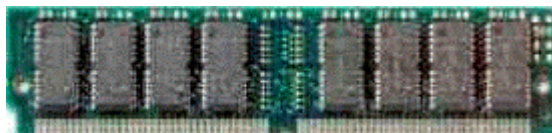


Figure 5.27 DRAM EDO 72 broches

Les barrettes à **72 broches**, plus récentes, sont des mémoires **DRAM FPM** ou **DRAM EDO** à **32 bits** (ou 36 bits avec les bits de parité). Ces barrettes équipaient les PC allant du 386DX aux premiers pentiums. Leur capacité va de 4 MBytes à 32 Mbytes. Comme les pentiums travaillent avec un bus de données d'une largeur de **64 bits**, il était impératif que les slots de la carte mère contiennent des *paires* de barrettes identiques (deux barrettes de 32 bits en parallèle = 1 barrette de 64 bits).

1 Les sigles PM, FPM, EDO, SDRAM, etc utilisés dans ce paragraphe seront définis au point suivant.

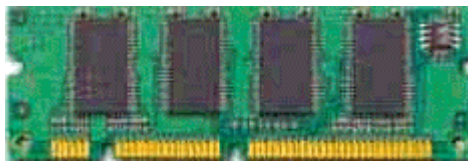


Figure 5.28 DRAM à 100 broches

On trouve plus rarement des barrette à **100 broches** (50 par face). Ce sont des mémoires à **64 bits** (ou 72 dont bits de parité) utilisées sur les imprimantes. Leur capacité va de 16 à 128 MBytes. Selon le modèle, il s'agit de DRAM **FPM**, DRAM **EDO** ou **SDRAM**.

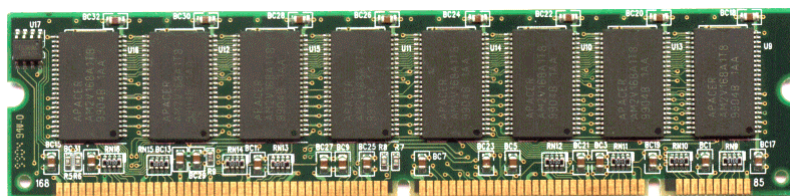


Figure 5.29 SDRAM 168 broches

Les barrettes plus récentes présentent **84 connecteurs** de chaque côté, ce qui les dote d'un total de **168 broches**. Ce sont des mémoires **64 bits** (ou 72 ou 80 avec le contrôle de parité et les technique ECC). Elles peuvent donc être utilisées seules avec un processeur pentium. Leur capacité d'étend de 16 à 1024 Mbytes. Ces barrettes présentent de nombreuses variantes et caractéristiques qui ne sont pas nécessairement compatibles avec tous les PC. Ainsi, les premiers modèles utilisaient encore les techniques DRAM FPM et DRAM EDO mais très rapidement, ce type de barrette est devenu le standard de la **SDRAM**.

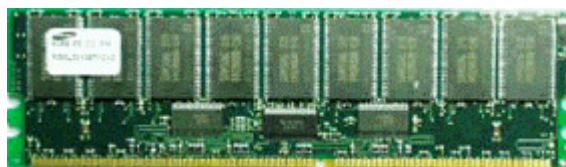


Figure 5.30 DDR SDRAM 184 broches

La **DDR SDRAM** est un type de mémoire très récent. La barrette compte **184 broches** et offre une largeur de bus de **64 bits**. Le transfert de données se fait à 200 ou 266 MHz avec une alimentation de 2.5V. Les capacités offertes vont de 64 MBytes à 1 Gbyte.



Figure 5.31 Rambus DRAM 184 broches

La mémoire **Rambus DRAM**, ou **RIMM**, se présente sous la forme de barrettes de **184 broches** pourvues d'un boîtier métallique qui sert de refroidisseur. La Rambus est un nouveau concept d'organisation de la mémoire : elle travaille sur 64 bits répartis en quatre canaux de 16 bits.

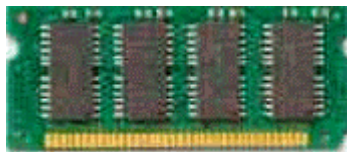


Figure 5.32 SO DIMM EDO 72 broches

L'encombrement des composants est un point critique de la construction des PC portables. Ceux-ci utilisent généralement des barrettes de mémoire plus compactes comme cette mémoire **SO DIMM** (Small Outline DIMM) à **72 broches** (36 par face). Il s'agit d'une mémoire sur **64 bits** qui utilise la technique **FPM** ou **EDO**. La capacité va de 8 à 32 Mbytes.

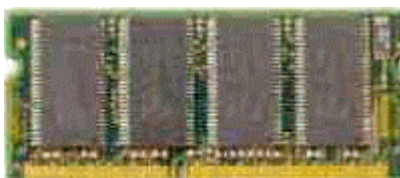


Figure 5.33 SO DIMM SDRAM 144 broches

Les portables plus récents utilisent des barrettes SO DIMM à **144 broches** (72 par face). Elles existent en version à **64 bits** (ou 72 bits avec bit de parité). Il s'agit d'une mémoire **SDRAM** qui offre une capacité allant de 16 à 256 MBytes selon les modèles.

Toutes ces barrettes ont des dimensions différentes. Il est donc impossible de les confondre. De plus, la rangée de broches présente un ou deux trous selon le modèle. Ces trous, placés à des endroits bien précis sont nommés *détrompeurs*.

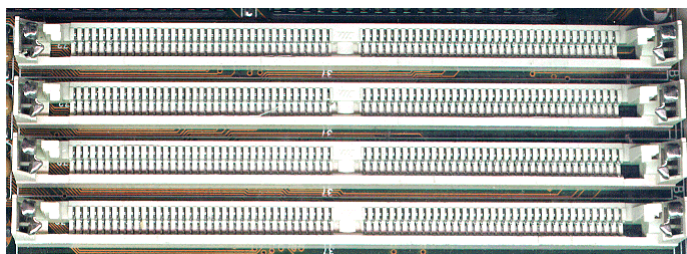


Figure 5.34 Slots pour 4 barrettes DRAM EDO à 72 broches.
On voit nettement la place du détrompeur central

De leur côté, les connecteurs de la carte mère sont conçus pour accepter un et un seul type de barrette. Il est donc impossible d'installer par erreur une barrette qui n'a pas le nombre de broches adéquat.

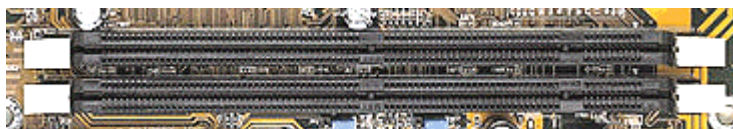


Figure 5.35 Slots pour 2 barrettes SDRAM à 168 broches.
On voit la place des deux détrompeurs

Malheureusement, aucune caractéristique physique ne permet de différencier les barrettes SDRAM qui fonctionnent à des fréquences différentes, ce qui peut évidemment poser quelques problèmes.

5.7. Evolutions de la RAM

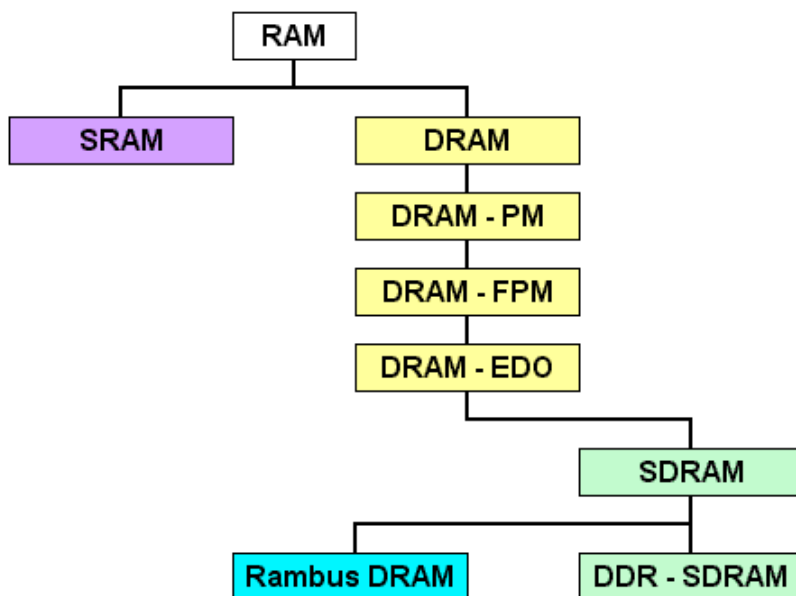


Figure 5.36 Les familles de mémoires

5.7.1. DRAM ou SRAM

Ainsi qu'il a déjà été dit à plusieurs reprises, DRAM et SRAM désignent des conceptions différentes de la cellule de mémoire.

L'avantage de la DRAM est qu'elle est dense, rapide et relativement bon marché. Sa densité, exprimée en nombre de bits ou de bytes par unité de surface est très élevée. Sa vitesse, représentée par le temps d'accès à une information est inférieur à 100 ns, tant en lecture qu'en écriture.

Son principal défaut est qu'en cas de coupure de l'alimentation électrique les données sont perdues en quelques millisecondes. Dès lors, la DRAM a besoin d'une alimentation électrique permanente ainsi que de circuits de contrôle additionnels pour assurer le rafraîchissement périodique de la mémoire. Leur consommation électrique est relativement élevée.

Dans la SRAM, l'information est aussi volatile. Cependant, cette mémoire ne nécessite pas de rafraîchissement périodique. Il suffit d'une faible tension pour maintenir l'état 0 ou 1 de la cellule. Ceci a deux conséquences : puisqu'il n'y a pas de rafraîchissement, la mémoire elle est toujours accessible et donc plus rapide. D'autre part, puisqu'il suffit de maintenir une tension, elle ne consomme qu'une puissance infime et une petite batterie ou un condensateur suffisent pour maintenir l'information pendant des mois (p.ex.: pile sur la carte mère).

L'inconvénient de la SRAM est son encombrement. La structure complexe de la cellule de mémoire nécessite six transistors par bit, ce qui conduit à des puces de taille relativement grande, à une diminution de la densité de l'information et à un coût de production nettement plus élevé.

Pour cette raison la DRAM est surtout utilisée dans les systèmes qui ont besoin d'une grande quantité de mémoire et qui sont reliés en permanence au secteur.

La SRAM est utilisée dans les systèmes où l'autonomie des batteries est cruciale. La SRAM est aussi préférée pour les mémoires qui doivent être très rapides car le système accède fréquemment. C'est notamment le cas des caches et des tables de traduction d'adresses.

5.7.2. DRAM PM

La DRAM PM (*ang.: Paged Mode - fr.: mode paginé*) est le mode "classique" de fonctionnement de la DRAM. Nous savons que la mémoire est structurée en lignes et colonnes. Des tableaux de 2048 ou 4096 lignes et colonnes ne sont pas rares. Or, nous savons que l'accès à une cellule se fait en deux temps :

- envoi du numéro de ligne
- envoi du numéro de colonne

Dès lors, on peut voir l'envoi du numéro de ligne comme l'ouverture d'une "page" de 2048 ou 4096 cellules de mémoire qui seront ensuite repérées par leur numéro de colonne. D'où l'expression "mode paginé"

En mode paginé, le temps d'accès à une cellule quelconque est au minimum de 5 cycles d'horloge. Sur un bus cadencé à 66 MHz ⁽¹⁾, un cycle d'horloge dure 15 ns. Le temps d'accès est au minimum de 75 ns.

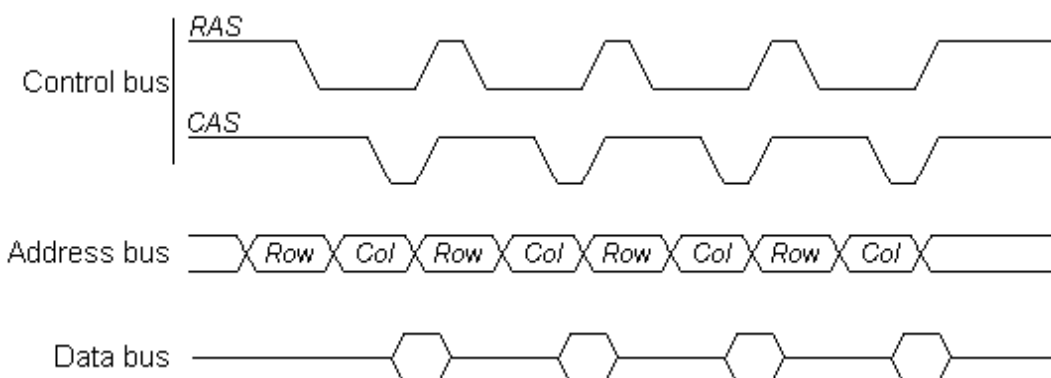


Figure 5.37 Timing et signaux d'accès à quatre données quelconques en mode PM

On pourrait comparer (approximativement) ce fonctionnement à celui d'un garçon de café qui, voyant une personne lui faire signe à une table, se rend auprès d'elle, prend la commande puis revient au bar pour préparer son plateau.

5.7.3. DRAM FPM

Le fonctionnement FPM (*ang.: Fast Paged Mode - fr.: mode paginé rapide*) tire parti du fait que les demandes successives du processeur concernent généralement des données qui sont rangées consécutivement en mémoire. En d'autres mots, les données demandées sont situées sur la même ligne (ou page de mémoire) et dans des colonnes successives.

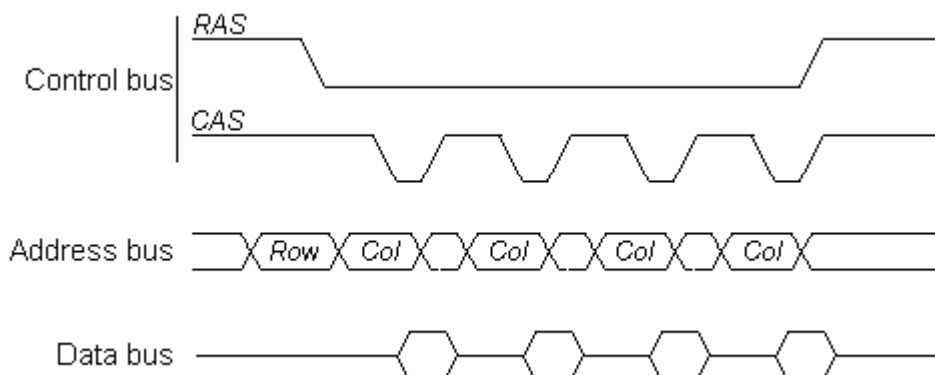


Figure 5.38 Timing et signaux d'accès à quatre données consécutives en mode FPM

1 Le bus à 66 MHz a été la configuration la plus habituelle pendant des années

Le mode FPM tient compte du fait que, si les données sont situées sur la même ligne, le numéro de ligne ne doit pas être répété à chaque demande. Il suffit de fournir uniquement le nouveau numéro de colonne, ce qui fait gagner deux cycles d'horloge pour chaque donnée supplémentaire.

Il est habituel de mesurer la performance du fonctionnement de la mémoire en comptant le nombre de cycles d'horloge nécessaires pour obtenir une série de quatre données contiguës :

- pour une DRAM PM, la performance serait du type 5-5-5-5
- pour une DRAM FPM, la performance devient 5-3-3-3

Le gain est appréciable. Dans notre comparaison approximative, le garçon de café prend la commande de la personne qui lui a fait signe et en profite pour faire le tour des tables voisines. Il revient ensuite au bar pour préparer toutes les commandes.

5.7.4. Mode de transfert en rafale

Le mode rafale (*ang.*: *burst mode* ou *bursting*) est une autre astuce qui permet de gagner du temps : au lieu de rechercher et d'envoyer les informations une à une, la mémoire prépare un paquet de données correspondant à plusieurs adresses consécutives en mémoire. La taille du paquet (*ang.*: *burst length*) correspond généralement au contenu de quatre adresses.

En d'autres mots, la mémoire anticipe une demande probable du processeur en lui envoyant des données supplémentaires. Ces données seront stockées dans un cache SRAM en attendant leur utilisation. Un peu comme si notre garçon de café apportait directement quatre bières au lieu d'une à un client dont il connaît les habitudes.

Sur le plan statistique, la démarche est pleinement justifiée : il est très probable que la prochaine adresse demandée par le processeur sera contiguë à celle qu'il vient de réclamer. Cela représente un gain de temps puisque les trois données consécutives à la première sont fournies sans temps de latence supplémentaire.

Le mode rafale fonctionne aussi bien en lecture qu'en écriture. Il est compatible avec différents types de mémoire et différents types de fonctionnement.

5.7.5. DRAM EDO

La DRAM EDO (*ang.*: **Extended Data Out** - *fr.*: *sortie de données étendue*) est presque identique au mode FPM.

Dans un cycle de lecture classique, les données sont envoyées et maintenues sur le data bus jusqu'à ce que le signal CAS (du bus de contrôle) revienne à son niveau normal (ce qui signale la fin du cycle).

En mode EDO, les données sont maintenues sur le data bus pendant un certain temps après la fin normale du cycle. Ceci n'empêche nullement l'envoi d'une nouvelle adresse de colonne sur le bus d'adresse. On crée ainsi un chevauchement des accès qui permet de gagner un peu de temps sur chaque cycle : le processeur peut encore lire les données demandées alors qu'il envoie déjà une nouvelle demande. C'est un peu comme si le garçon de café de notre comparaison prenait la commande d'un client pendant qu'il apporte les consommations à un autre.

Par rapport au fonctionnement FPM, le gain est de l'ordre de 10 ns par cycle de lecture d'une nouvelle colonne. Le temps d'accès à la mémoire en mode EDO est

de l'ordre de 50 à 60 nanosecondes pour une fréquence de bus de 66 MHz. Couplé avec un mode burst, il permet d'obtenir des cycles de la forme 5-2-2-2.

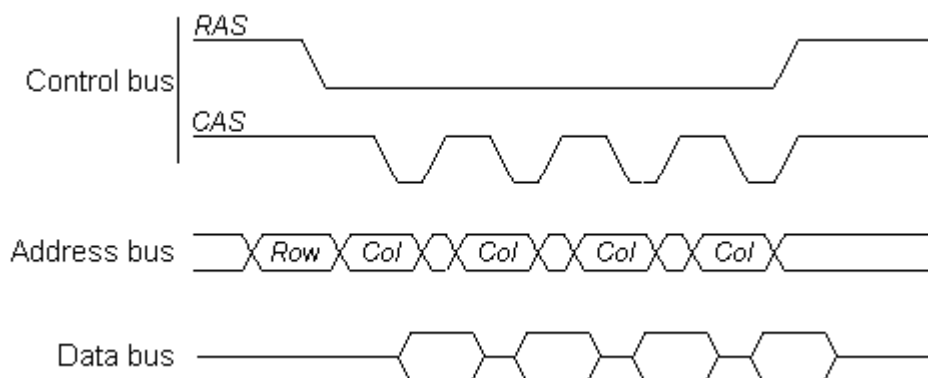


Figure 5.39 Timing et signaux d'accès à quatre données consécutives en mode EDO

Avec l'EDO, la DRAM avait épuisé toutes les possibilités d'amélioration du fonctionnement asynchrone. La mémoire EDO n'acceptant pas des fréquences supérieures à 66 Mhz, elle a disparu au bénéfice de la SDRAM.

5.7.6. SDRAM

Dans les cas examinés ci-dessus - PM, FPM, EDO - la mémoire DRAM est cadencée par sa propre horloge. Cette horloge est totalement indépendante de celle de la carte mère. C'est pourquoi ces fonctionnements sont qualifiés d'asynchrones.

L'avantage des processus asynchrones est qu'ils permettent de faire fonctionner ensemble deux parties qui travaillent chacune à leur propre rythme : la vitesse à laquelle le client boit sa bière est indépendante de la vitesse à laquelle le garçon prend la commande et la sert. Néanmoins, à un moment donné, les deux parties doivent se rencontrer et ceci pose deux problèmes :

- soit on crée des stocks intermédiaires pour éviter une interruption du service. C'est ce que fait le serveur qui anticipe et apporte plusieurs bières; c'est également ce qu'on fait en créant des mémoires cache.
- soit on admet de longs délais d'attente. C'est ce qui arrive quand le client boit sa bière plus vite que le serveur ne les livre; c'est également ce qui arrive quand le contrôleur de mémoire synchronise les transactions et que le processeur subit de longs cycles d'attente.

Le processus le plus efficace est le processus synchrone. C'est sous cette forme que fonctionnent les chaînes de montage de voiture, les militaires qui marchent au pas et les volontaires qui se passent des seaux d'eau pour éteindre un incendie.

La DRAM peut synchroniser son horloge avec celle de la carte mère. Elle devient alors une DRAM synchrone ou **SDRAM** (*ang.*: **Synchronous DRAM**) ⁽¹⁾. En supprimant les temps d'attente, la SDRAM autorise des transferts extrêmement rapides.

Par ailleurs, les barrettes de SDRAM utilisent fréquemment l'entrelacement des données (*ang.*: *interleaving*) : en répartissant les données sur quatre bancs, on évite d'être bloqué par le rafraîchissement de la ligne (mais on doit quand même rafraîchir les tableaux après 1 ms). On peut adresser une commande de lecture à un des bancs de la puce pendant qu'un autre est déjà en train d'être lu. Dès lors, le cycle en mode rafale est de la forme 5-1-1-1. De plus, comme la cadence peut être

1 Ne pas confondre SDRAM et SRAM !

augmentée de 66 à 133 voire 150 MHz, le meilleur temps d'accès revient à moins de 10 ns.

Mais il ne faut pas se leurrer : en accès aléatoire, où seule la première donnée est utile, une barrette de mémoire SDRAM PC-100 (100 MHz) a un temps d'accès de 40 ns, soit à peine mieux qu'une DRAM-EDO 50. Par contre quand les données sont bien organisées et que le mode rafale donne toute son efficacité, le gain est net.

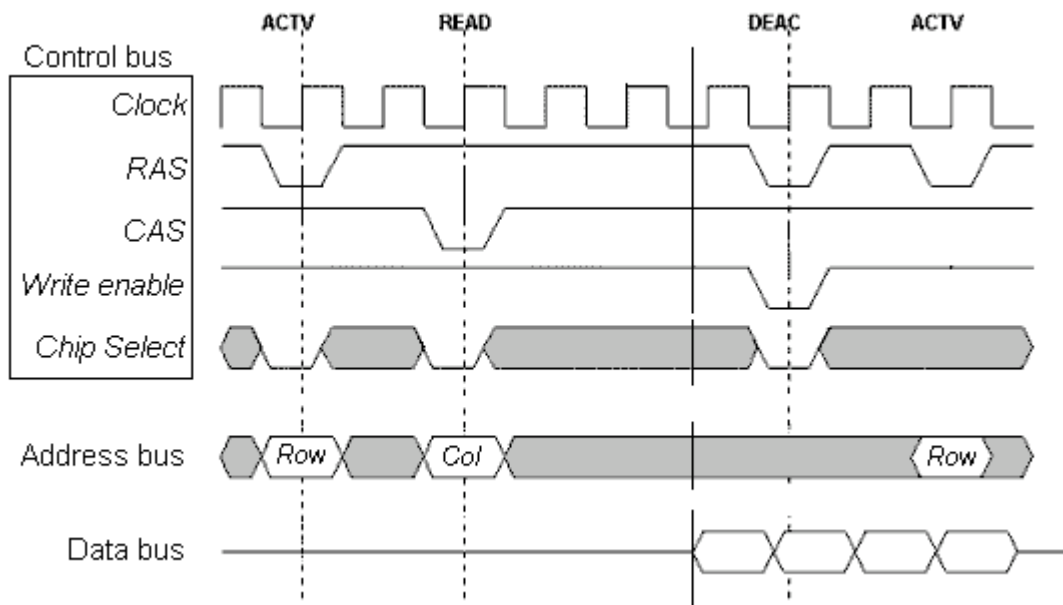


Figure 5.40 Timing et signaux d'accès à une rafale de quatre données consécutives dans la SDRAM

Une adresse de ligne est présentée à la RAM sur le bus d'adresse. Les signaux RAS et CS (chip select) sont activés pendant un cycle d'horloge, ce qui déclenche le processus (**ACTV**).

On laisse passer un cycle puis on présente l'adresse de la colonne et on déclenche la lecture (**READ**) en activant les signaux CAS et CS.

Trois cycles d'horloge plus tard, on récupère les données sur le data bus à raison d'une donnée par cycle. En même temps, on envoie simultanément les signaux RAS, CAS et CS afin de désactiver (**DEAC**) la ligne qui vient d'être lue et de permettre son rafraîchissement.

Comme le CPU sait exactement quand les données seront prêtes, il peut faire autre chose pendant les cycles d'attente.

Signalons au passage que les SDRAM fonctionnent avec une tension d'alimentation de 3.3V alors que la DRAM EDO utilise une tension de 5V.

5.7.7. DDR SDRAM

La DDR-SDRAM (*ang.: Double Data Rate SDRAM*) est assez semblable à la SDRAM : c'est une mémoire synchrone, calée sur une horloge externe fonctionnant à 100 ou 133 MHz.

Mais, alors que la SDRAM n'utilise que le niveau "haut" du cycle d'horloge pour expédier ses données, la DDR SDRAM est capable de détecter et d'utiliser aussi bien le flanc montant que le flanc descendant du cycle d'horloge (*ang.: rising and falling edge*). Par conséquent, la DDR SDRAM a un taux de transfert deux fois plus élevé qu'une SDRAM "normale".

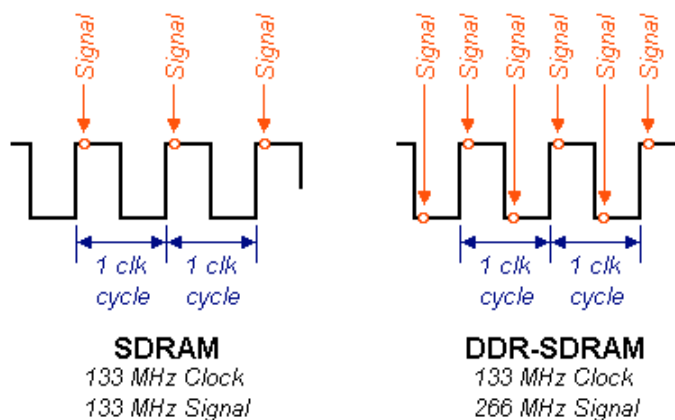


Figure 5.41 Principe du doublement de fréquence d'horloge

Cette astuce permet de doubler le taux de transfert et la bande passante de la SDRAM à fréquence d'horloge égale : un bus à 133 MHz permettra à une DDR SDRAM de travailler à 266 MHz.

Notons au passage qu'une DDR-SDRAM travaille avec une tension de 2.5V

5.7.8. RAMBUS ou DR-SDRAM (Rambus DRAM)

La DR-SDRAM (ang.: *Direct Rambus SDRAM*) encore appelée Rambus, RDRAM ou RIMM fonctionne selon un tout autre principe.

A la base, il s'agit toujours d'une mémoire DRAM fonctionnant en mode synchrone. "normal" ⁽¹⁾. Mais ici, les 64 bits du bus de données sont scindés en **quatre canaux de 16 bits**. Ces quatre canaux fonctionnent de manière indépendante à la fréquence de **800 MHz**, ce qui donne une bande passante totale de 1.6 Gbyte/s.

Il semble que le principal avantage soit de permettre à quatre applications de disposer chacune de son propre accès direct à la zone de mémoire qui lui a été attribuée.

Il n'est pas inutile de signaler que la mémoire RAMBUS est une technologie propriétaire (c-à-d brevetée), développée par Rambus Inc. et adoptée par Intel dans son nouveau chipset i820/i840. Ceci a deux conséquences : d'une part, toute entreprise désirant construire des barrettes selon cette technologie doit acheter une licence aux sociétés RAMBUS et Intel. D'autre part, la documentation disponible sur ce sujet est particulièrement confuse. Selon les fiches techniques, il semble que les chips et les barrettes sont parcourues en série et non en parallèle. Tout le transfert est réglé par l'impédance de chaque barrette qui provoque un certain déphasage dans le signal. Nous en saurons peut-être plus un jour.

5.8. En résumé

Il ne faut pas confondre

- **SRAM** et **DRAM**, qui définissent le *type de la cellule élémentaire* de mémoire. Sauf cas exceptionnel, toutes les barrettes de mémoire sont de type DRAM;
- **SIMM** et **DIMM**, qui définissent la *disposition des puces et des contacts* sur une barrette de mémoire : sur une seule face ou sur les deux faces;
- **DRAM-EDO**, **SDRAM**, **Rambus DRAM**, etc. qui définissent le *type d'exploitation* de la mémoire de la mémoire DRAM : asynchrone, avec diverses astuces, ou synchrone, avec diverses astuces.

1 Mode "normal" signifie ici non-DDR

5.9. Installer la mémoire

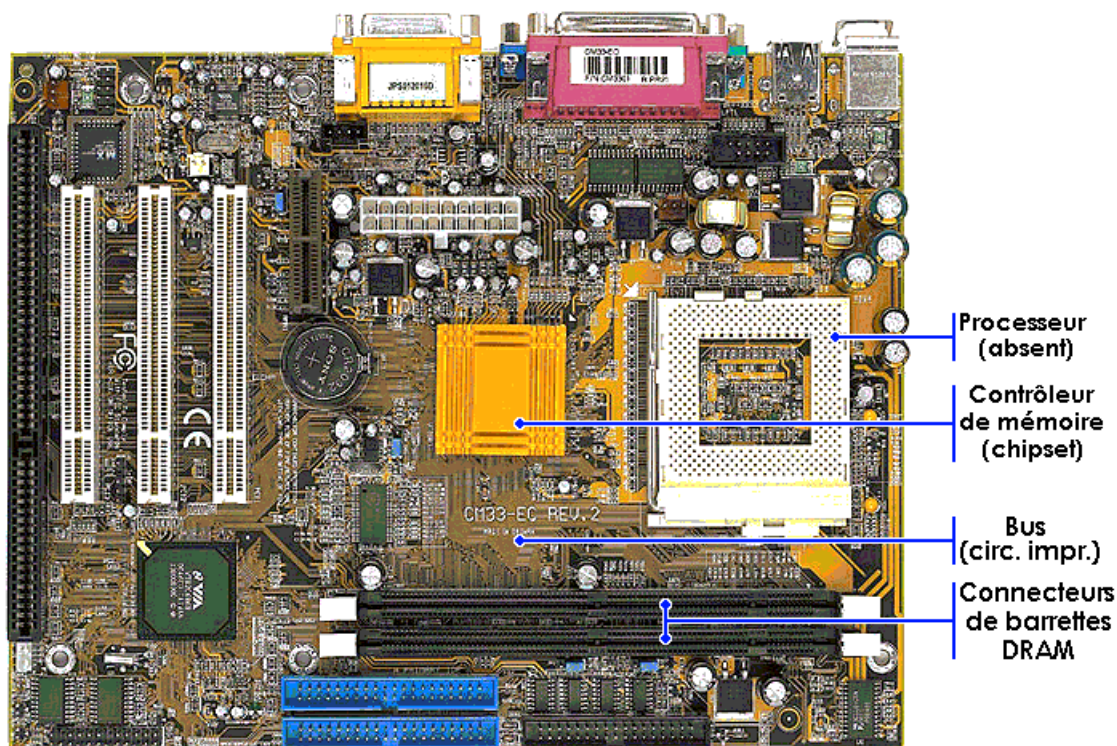


Figure 5.42 Position des connecteurs de bancs de mémoire

Dans un ordinateur, les barrettes de mémoire s'insèrent sur la carte mère, dans des connecteurs disposés en rangées parallèles à proximité du processeur.

Ainsi qu'il a été dit plus haut, ces connecteurs sont conçus pour accepter un et un seul type de barrette. Il est donc impossible d'installer par erreur une barrette qui n'a pas le *nombre de broches* adéquat. Par contre, il est parfaitement possible d'installer des barrettes qui n'ont pas la *fréquence* adéquate (p.ex.: installer une SDRAM 100 MHz sur un bus 66 MHz). Sachez que c'est le contrôleur de mémoire, encore appelé chipset, qui décide du type de mémoire avec lequel il veut bien travailler. Inutile d'essayer de placer une barrette SDRAM 100 MHz si le chipset est prévu pour des barrettes à 66 MHz... même si le nombre de broches est bon.

Il existe parfois - quoique rarement - des règles d'installation : sur certains PC, tous les connecteurs doivent porter des modules de même capacité. Sur d'autres, le premier banc est destiné au module de plus grande capacité... Il sera temps d'y penser si votre ordinateur refuse de démarrer après avoir reçu une barrette de mémoire supplémentaire.

Rien de tel que la consultation du manuel ou de la fiche technique de la carte mère pour savoir ce qui est permis et ce qui ne l'est pas ! Un examen attentif des minuscules inscriptions imprimées sur la carte mère ou gravées sur les chips et les supports de mémoire peut parfois apporter la clé de l'énigme...

Exercices du chapitre 5

♦ Exercice 5.1

Combien de tableaux, de lignes et de colonnes trouve-t-on dans un chip de 64 M x 4 bits. Combien de pins trouve-t-on au minimum sur un tel chip ?

♦ Exercice 5.2

Un constructeur utilise des chips 16 M x 1 bit pour fabriquer des barrettes de mémoire qui stockent les données par paquets de 2 bytes. Combien doit-il placer de chips par barrette ?

Ces barrettes sont utilisées dans un PC dont le bus de données a une largeur de 32 bits. Combien barrettes faut-il au minimum pour satisfaire aux exigences du bus ? Quelle est la capacité de mémoire minimale fournie par cet ensemble de barrettes ?

♦ Exercice 5.3

Expliquez brièvement la différence entre rafraîchissement après lecture et rafraîchissement périodique d'une DRAM. Quelles sont les causes ?

♦ Exercice 5.4

Calculez et comparez les bandes passantes des mémoires suivantes :

- DRAM EDO 66 MHz
- SDRAM PC-133
- DDR-SDRAM PC-266
- Rambus 800