

## Chapitre 3

# La mémoire de l'IBM PC

L'IBM PC, comme tous les micro-ordinateurs, utilise des circuits intégrés de mémoire pour sa mémoire vive et le BIOS.

### 3.1 Les circuits intégrés de mémoire

Nous avons vu la notion de mémoire d'un point de vue logiciel. Nous allons ici détailler un peu plus la réalisation physique des circuits intégrés de mémoire.

#### 3.1.1 Les ROM

##### 3.1.1.1 Schéma d'une ROM

On peut voir une puce de mémoire ROM comme un élément ayant  $n$  broches d'entrée et  $m$  broches de sortie. Pour chacune des  $2^n$  combinaisons possibles d'entrée il y a un mot de sortie de  $m$  bits, comme le montre la figure 3.1.

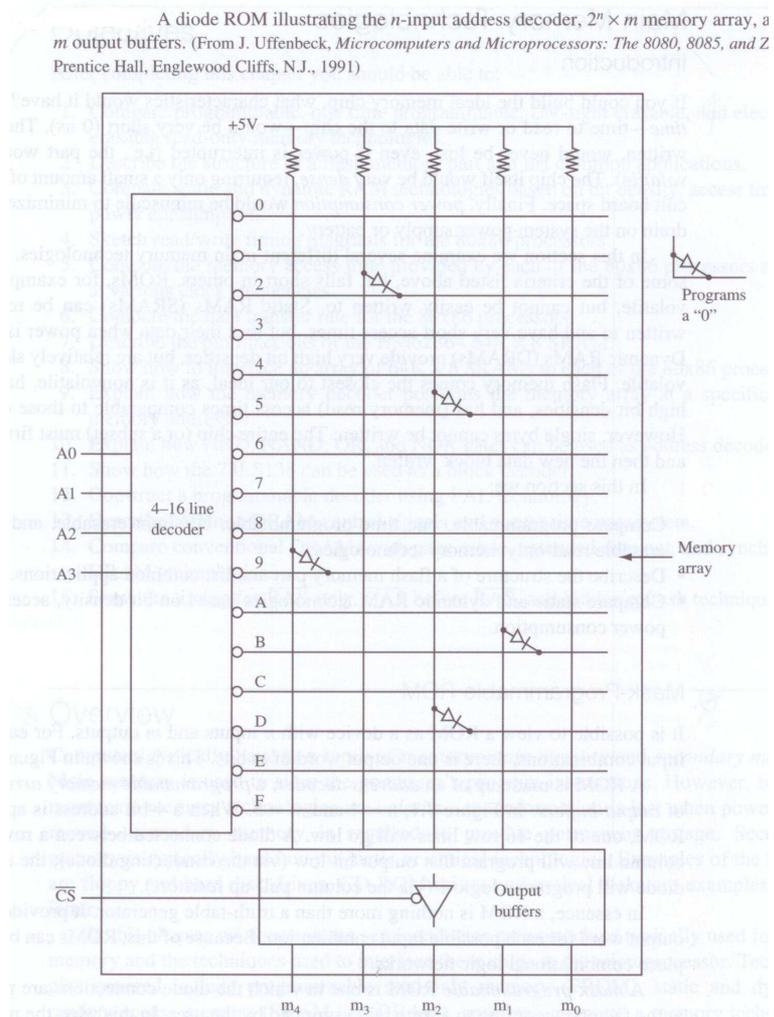


FIGURE 3.1 – ROM programmable par masque

Une puce ROM est constituée d'un **décodeur d'adresse**, d'une **matrice** de mémoire programmable et d'un ensemble de **tamppons** de sortie. Sur la figure 3.1, on a  $n = 4$  et  $m = 5$ . Lorsqu'une adresse de 4 bits est appliquée à la ROM, une des 16 lignes a un niveau bas. Une diode connectée entre une ligne et une colonne donne un bit de sortie de niveau bas. L'absence de ligne donne un niveau haut.

Une puce ROM n'est rien d'autre qu'une table de vérité, donnant un mot en sortie (et non un seul bit). Une ROM peut d'ailleurs remplacer un réseau de circuits combinatoires.

### 3.1.1.2 Technologies des ROM

ROM programmable par masque.- Une **ROM programmable par masque** (en anglais *Mask-Programmable ROM*) est un circuit intégré ROM formé de diodes, comme nous venons de le voir, la présence de diode ou non étant décidée par un masque lors de la conception du circuit intégré.

Une telle ROM ne peut donc être fabriquée qu'en usine et donc à condition de pouvoir vendre plusieurs milliers de circuits intégrés ayant la même table de vérité.

PROM.- L'écriture de la ROM en usine est une contrainte très forte. Il existe des **ROM programmables** ou **PROM** (pour *Programmable ROM*) qui peuvent être écrites par un équipementier ou par l'utilisateur final. La programmation d'une PROM s'appelle *burning ROM* en anglais et exige un appareillage spécial (appelé *ROM burner* ou *ROM programmer*).

Il existe un certain nombre de technologies utilisées pour ces PROM, dont nous allons parler maintenant.

OTPROM.- Un type de PROM utilise des liens fusibles par une intensité suffisamment faible dans la matrice des diodes (figure 3.2).

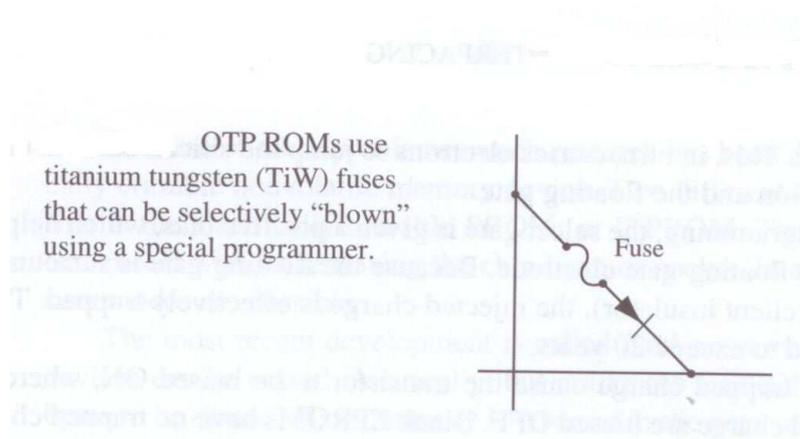


FIGURE 3.2 – PROM à un coup

En appliquant une impulsion ayant l'intensité désirée à l'élément voulu, le fusible fond et on obtient un 1 de façon permanente. Bien entendu on ne peut écrire qu'une seule fois. On parle de **OTPROM** (pour *One-Time PROM*).

**EPROM.**- Un type, plus onéreux, de PROM est l'**EPROM** (pour *Erasable Programmable ROM*), inventé pour permettre sa reprogrammation, ce qui est utile en particulier lors du développement d'un prototype (figure 3.3).

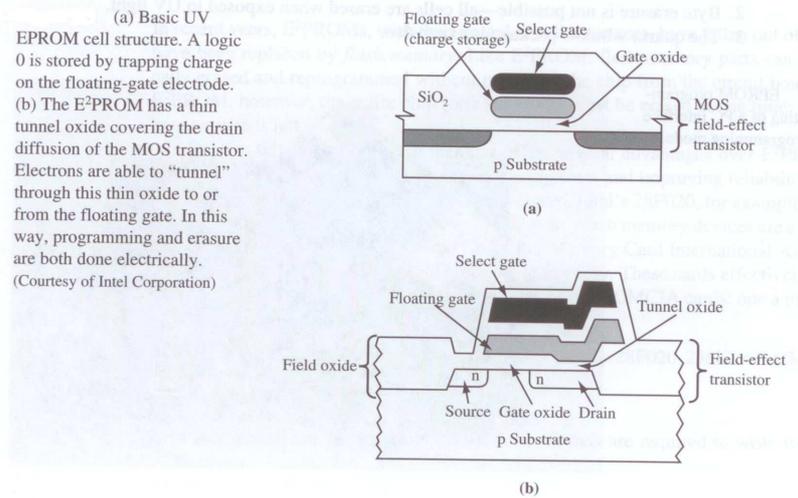


FIGURE 3.3 – EPROM

L'**UV-EPROM** (pour *Ultra-Violet-light Erasable PROM*) peut être effacée et écrite plusieurs fois, en utilisant les rayons ultra-violetes, comme son nom l'indique.

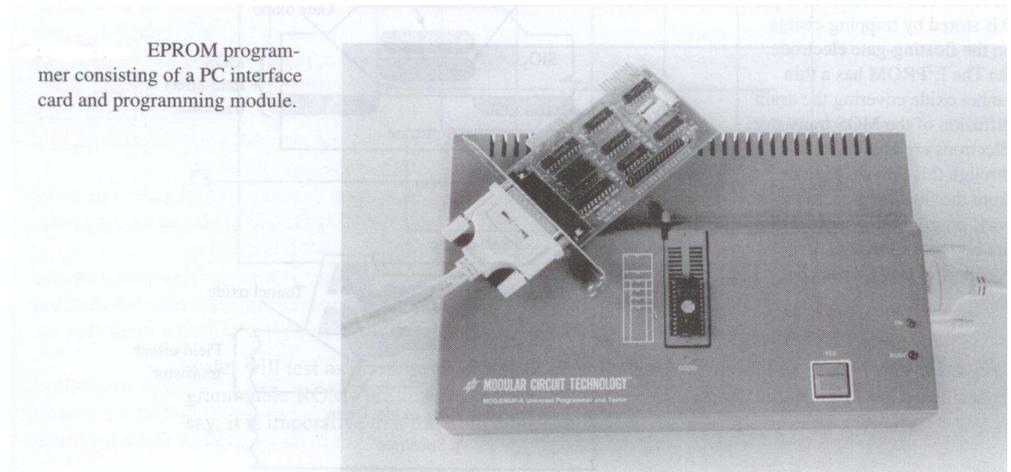


FIGURE 3.4 – Programmation des EPROM

Elle utilise des cellules de transistor **FAMOS** (pour *Floating-gate Avalanche-injection MOS*, figure 3.3 (a)) pour stocker l'information. Appliquer une tension spéciale  $V_{PP}$  a pour effet un champ électrique puissant dans la région du canal du transistor. Ceci, à son tour, permet aux électrons de sauter la barrière de dioxyde de silice et la porte flottante. Durant l'écriture, la porte sélectionnée a un biais positif, ce qui l'aide à attirer les électrons de l'électrode de la porte

flottante. Puisque la porte flottante est entourée de dioxyde de silice (un excellent isolant), la charge injectée est effectivement attirée. La période de stockage est supérieure à 20 ans. Les cellules ayant attiré une charge sont biaisées (et ont donc la valeur logique 0) alors que les autres ne le sont pas (et ont donc la valeur logique 1).

L'UV-EPROM peut être effacée en la soumettant à une lumière ultra-violette de 2 537 angstroms. Les électrons de la porte flottante absorbent alors suffisamment d'énergie pour renverser le processus et retourner au substrat.

Les UV-EPROM sont placées dans des boîtiers spéciaux en céramique avec une fenêtre en quartz pour permettre de l'effacer. Des effaceurs permettent d'effacer en 15 à 20 minutes.

En temps normal la fenêtre doit être recouverte d'un cache opaque, puisque l'éclairage fluorescent pourrait l'effacer (en trois ans, ou en une semaine en cas d'exposition directe au Soleil).

Les UV-EPROM sont écrites avec une tension  $V_{PP}$  de 13 V à la broche correspondante. On utilise les broches des données et les broches des adresses. Les broches subissent des impulsions de 100  $\mu s$ . On utilise pour cela un PC, une carte d'interface, un module pour placer l'EPROM ainsi qu'un logiciel propriétaire. L'écriture s'effectue en 2 à 5 s.

Les UV-EPROM ont un certain nombre d'inconvénients :

- la puce doit être retirée de la carte mère pour être effacée;
- on ne peut pas effacer octet par octet puisque la puce en entier est exposée à la lumière ultra-violette;
- la fenêtre en quartz est onéreuse.

E<sup>2</sup>PROM.- Les inconvénients des UV-EPROM ont conduit à des recherches sur des EPROM effaçables électriquement. Ceci a conduit aux **E<sup>2</sup>PROM** (pour *Electrically Erasable ROM*) et aux mémoires flash. Ces deux types de ROM peuvent être écrites et effacées sans retirer le circuit intégré de la carte mère. De plus on peut écrire un octet à la fois, ou le tout si l'on veut.

Le principe d'une cellule d'E<sup>2</sup>PROM est le suivant (figure 3.3, (b)) : comme pour une cellule FAMOS, il y a une porte flottante, cependant il y a aussi un très fin tunnel d'oxyde sous le drain de diffusion du transistor MOS. Avec une tension positive appliquée à la porte voulue, les électrons sont attirés vers la porte flottante. Appliquer une tension positive au drain terminal décharge (efface) la cellule.

Puisque l'oxyde de la porte sous le drain est très fin, le processus est contrôlé par un effet tunnel au lieu d'une injection d'avalanche. L'effet tunnel a l'avantage qu'une charge importante peut être injectée durant le cycle d'écriture et que seulement une petite quantité de charge est perdue durant un cycle de lecture. La durée de conservation des informations est supérieure à 10 ans.

L'écriture est pratiquement instantanée, on peut choisir le bit à écrire et on peut l'écrire alors que la ROM est sur la carte mère.

Malheureusement, chaque cycle de lecture-écriture a pour conséquence une diminution de la charge attirée par la porte. L'E<sup>2</sup>PROM ne peut plus être réécrite après 10 000 à 1 000 000 cycles de lecture-écriture.

Le temps d'accès en lecture des E<sup>2</sup>PROM est comparable aux autres EPROM mais le temps d'écriture est lent (environ 25 ms). Des états d'attente WAIT sont donc requis au moment de l'écriture.

Mémoire flash.- Les E<sup>2</sup>PROM ont été remplacées par les **mémoires flash** (*flash EEPROM*) dans la seconde moitié des années 1990. Comme les E<sup>2</sup>PROM, les mémoires flash peuvent être effacées électriquement et reprogrammées sans les retirer de la carte mère. Cependant la puce en son entier, ou en fait un sous-bloc, doit être effacé à la fois ; on ne peut pas n'effacer qu'un seul bit.

En dépit de cet inconvénient, les mémoires flash présentent de nombreux avantages par rapport aux E<sup>2</sup>PROM. Une cellule de mémoire flash est beaucoup plus simple à réaliser, ce qui diminue les coûts de fabrication. Les puces peuvent être plus denses. Les mémoires flash peuvent être disponible sous la forme d'une carte de crédit (**PCMCIA** pour *Personal Computer Memory Card International Standard*), qui peuvent stocker 40 Mo de mémoire non volatile. On parle de *disque dur sur silice*.

### 3.1.2 Technologies RAM

Il y a deux types principaux de RAM : les RAM statiques et les RAM dynamiques.

#### 3.1.2.1 RAM statique

Les **RAM statiques**, ou **SRAM**, utilisent une bascule comme élément de stockage de base. La figure 3.5 montre le schéma d'une telle cellule.

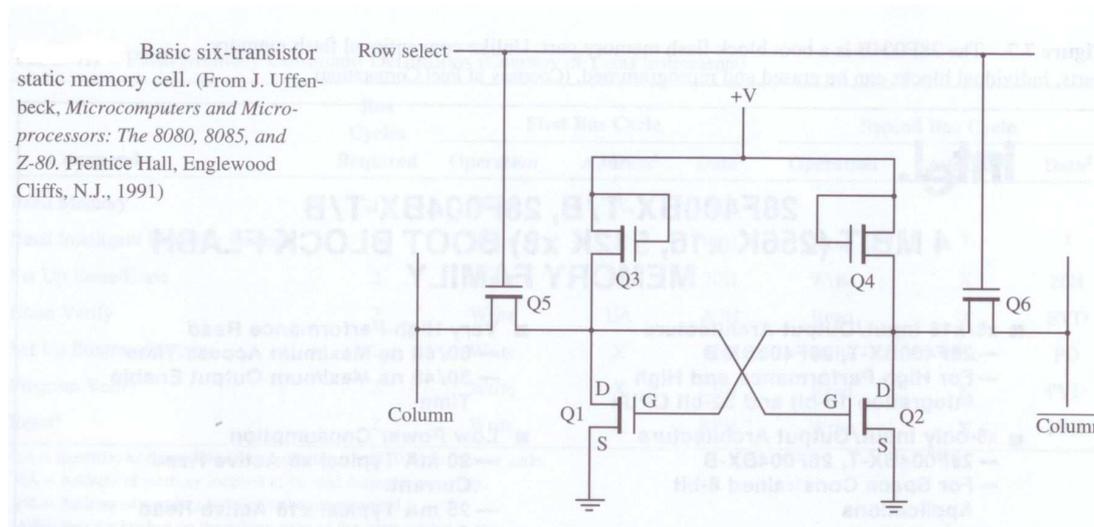


FIGURE 3.5 – RAM statique

Principe.- Dans cette cellule, Q1 et Q2 forment une bascule (*flip-flop*) *cross-coupled*, Q3 et Q4 agissant comme des résistances. Les transistors Q5 et Q6 fournissent un accès à la cellule. Pour écrire une donnée sur la bascule, la ligne de sélection de ligne est rendue active, en actionnant les transistors Q5 et Q6. Pour stocker la valeur logique 1, la ligne COLUMN est à une tension haute et la ligne  $\overline{\text{COLUMN}}$  à une tension basse. La tension haute de la ligne COLUMN est, via Q5, appliquée à la porte de Q2, actionnant ce transistor et forçant son drain à être à niveau bas.

De même un niveau bas sur  $\overline{\text{COLUMN}}$  désactive Q1, et son drain devient haut, forçant le niveau haut appliqué à la porte de Q2. En effet, la cellule se place elle-même dans l'état appliqué. Les transistors de sélection de ligne Q5 et Q6 peuvent maintenant être désactivés, l'information est alors retenue. Pour stocker la valeur 0, on répète ce processus, mais cette fois l'entrée de COLUMN est basse et celle de  $\overline{\text{COLUMN}}$  est haute.

Pour lire la donnée stockée par une cellule SRAM, la ligne de sélection de ligne est encore rendue active, mais cette fois la différence de potentiel entre les lignes de colonnes est testée. Une tension positive entre ces lignes indique une valeur 1. Une tension négative entre ces mêmes lignes indique une valeur 0.

Les données sont conservées sans nécessité de signal de rafraîchissement, d'où le nom de *statique*. Par contre, les données sont perdues en cas de coupure du courant. De plus, lorsque la tension est réappliquée, l'état de chaque bascule est imprévisible.

Applications.- Les SRAM ont des temps d'accès très court. Elles sont surtout utilisées de nos jours pour les mémoires cache, ce qui évite d'insérer des états WAIT.

### 3.1.2.2 RAM dynamique

Six transistors par cellule empêchent la RAM statique d'être un composant ayant une très grande densité. Ceci a conduit à chercher une autre technique.

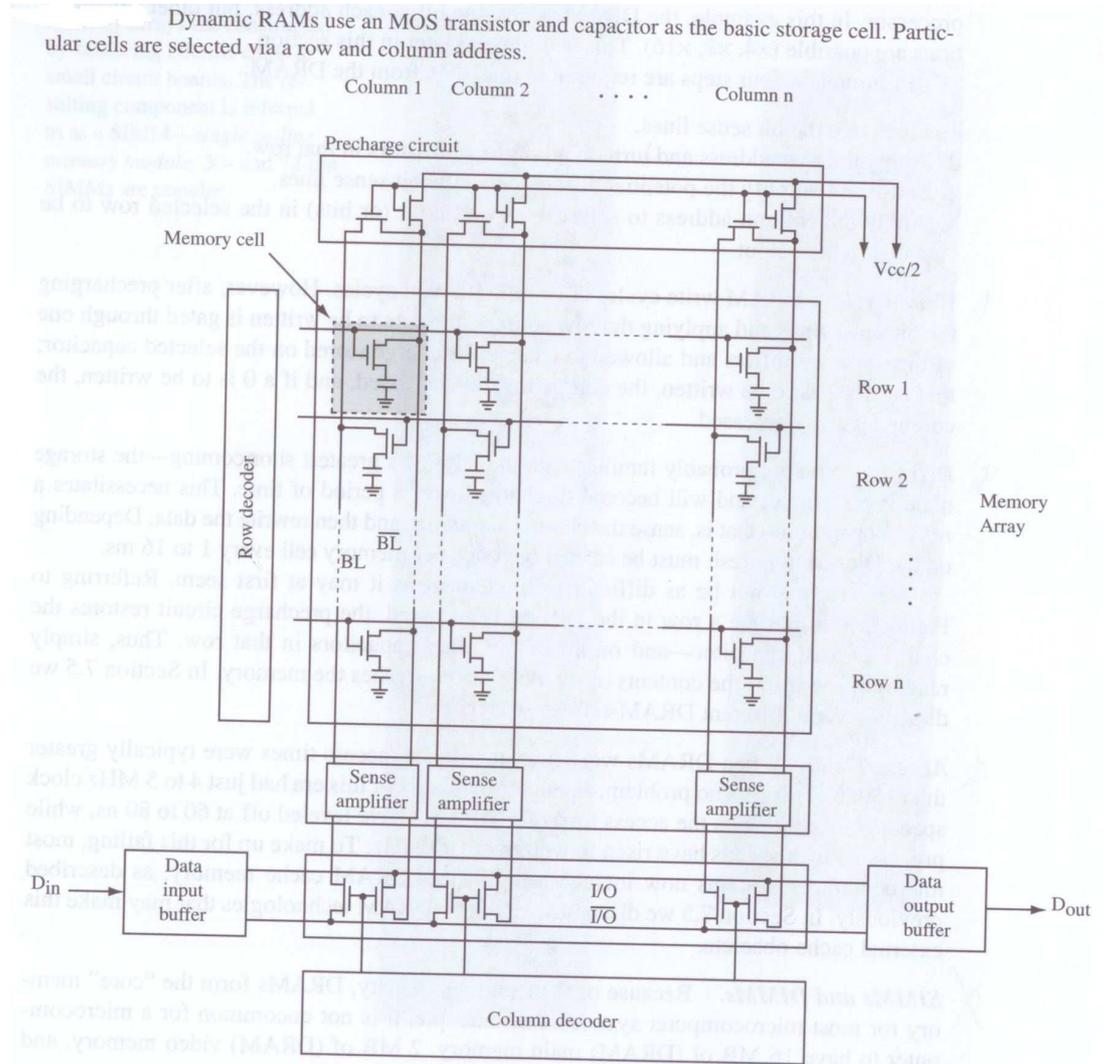


FIGURE 3.6 – RAM dynamique

**Principe.**- La figure 3.6 montre la structure interne d'une **RAM dynamique**, ou **DRAM**. Une cellule de mémoire comprend seulement un transistor et un condensateur. Inventée par *Intel* en 1970, ceci fut son premier type de gloire. Chaque cellule est identifiée par son adresse de ligne et son adresse de colonne.

Une donnée est lue en testant la différence de potentiel du condensateur de la cellule. Le décodeur de l'adresse de ligne actionne tous les transistors d'une ligne donnée. La charge de chacun des condensateurs ainsi sélectionnés est alors placée sur les fils de test (BL et  $\overline{BL}$ ). Cependant ces

charges sont très petites ; on commence donc par placer une précharge de  $V_{CC}/2$  volts. Lorsque les transistors de la ligne sont actionnés, ce niveau est soit accru (lorsque le condensateur est chargé), soit décré (lorsque le condensateur est vide). La quantité concernée est très faible, de l'ordre de 100 mV. Des amplificateurs différentiels sont donc nécessaires pour convertir cet accroissement à des niveaux logiques valides (0 V et 5 V).

On complète le cycle de lecture en appliquant une adresse de colonne utilisée pour sélectionner un bit de la ligne active maintenant stockée dans les amplificateurs. La sortie de l'amplificateur sélectionné est alors appliquée à un tampon de sortie, où elle peut être lue par le microprocesseur.

Un cycle de lecture nécessite donc quatre étapes :

- précharger les lignes ;
- appliquer l'adresse de ligne et actionner tous les transistors de cette ligne ;
- tester et amplifier la différence de potentiel des lignes ;
- appliquer l'adresse de colonne pour sélectionner le bit (ou les bits) de la ligne active.

Le cycle d'écriture des DRAM est analogue à celui du cycle de lecture. Cependant, après que les fils de test aient été pré-chargés et l'adresse de ligne appliquée, la donnée à écrire est placée dans des amplificateurs de signal, ce qui permet de changer la charge stockée dans le condensateur choisi. C'est-à-dire que si c'est un 1 qui doit être écrit, le niveau de la charge est augmenté, et si c'est un 0 le niveau de la charge est diminué.

Rafraîchissement des DRAM.- Le principal inconvénient des DRAM est que le stockage n'est pas parfait. Une opération de **rafraîchissement**, c'est-à-dire de test de la charge, de son amplification et de sa réécriture, est donc nécessaire régulièrement. Une cellule doit être rafraîchie toutes les 1 à 16 ms.

En fait le rafraîchissement n'est pas une opération si difficile que cela. À chaque fois qu'on accède à une ligne, le circuit de précharge restaure la charge des condensateurs. Il suffit donc de lire ou d'écrire pour rafraîchir.

Temps d'accès.- Lorsque les DRAM ont été introduites, leur temps d'accès était de 150 ns. Ceci n'était pas un problème, puisque les microprocesseurs d'alors avaient une cadence de 4 à 5 MHz. Le temps d'accès est passé à 80 ou 60 ns lorsque la cadence des microprocesseurs est passée à 100 MHz. Ceci a exigé l'utilisation de mémoire cache. Puis de nouvelles technologies RAM ont rendu obsolète l'utilisation de mémoire cache.

Applications.- Grâce à leur haute densité en bits, les DRAM sont utilisées pour la mémoire centrale de beaucoup de micro-ordinateurs. Bien entendu la complexité d'un cycle de lecture ou d'écriture rend ces mémoires moins rapides que les SRAM.

### 3.1.3 Caractéristiques d'un circuit intégré de mémoire

Capacité.- La capacité d'un circuit de mémoire (*chip capacity*) est exprimée en bits (kilobits, mégabits) contrairement à la capacité des ordinateurs qui sont exprimés en octets.

Organisation des circuits mémoire.- L'**organisation** des puces de mémoire concerne la façon dont le nombre de cellules mémoire (donc le nombre de bits) est réparti. En effet lorsqu'on spécifie une adresse sur les broches A0, A1, ... du circuit intégré, celui-ci renvoie une donnée sur les broches D0, D1, ... Il se peut très bien qu'il n'y ait qu'une seule broche de donnée D0, si bien qu'on reçoit un bit. On peut aussi avoir quatre broches de données D0, D1, D2, D3 pour recevoir un demi-octet et ainsi de suite. Autrement dit, on peut accéder, par exemple, à une puce de 16 Mbits de plusieurs façons :

- 16M × 1 bit,
- 4M × 4 bits,
- 2M × 8 bits (2 Mio),
- 1M × 16 bits (1 M mots).

Pour résumer :

- Chaque circuit intégré de mémoire contient  $2^x$  cellules, où  $x$  est le nombre de broches d'adresse sur le circuit intégré.
- Chaque cellule mémoire contient  $y$  bits, où  $y$  est le nombre de broches de données sur le circuit intégré.
- Le circuit intégré détient  $2^x \times y$  bits.

Les SRAM et les ROM sont presque toujours × 8 ; on parle d'un octet de **largeur**.

Temps d'accès.- Une des caractéristiques les plus importantes d'un circuit intégré de mémoire est la vitesse à laquelle on peut y accéder aux données. Le **temps d'accès** (*access time*) est la durée entre le moment où l'on présente l'adresse sur les broches d'adresse et le moment où on peut récupérer les données sur les broches de données.

Il est de l'ordre de la nanoseconde.

Accès à la DRAM en deux temps.- Il y a un problème pour adresser un très grand nombre de cellules sur un seul circuit intégré. Par exemple, pour une puce 64K×1, on doit avoir 16 broches d'adresse et une broche de donnée, en plus évidemment des broches  $V_{CC}$ , de terre et de contrôle de lecture et d'écriture.

Pour réduire le nombre de broches nécessaires pour l'adresse, on utilise un procédé de multiplexage/démultiplexage. On découpe l'adresse en deux moitiés et on envoie chacune des deux moitiés à travers les mêmes broches. En interne, la structure de la DRAM est divisée en un carré de lignes et de colonnes. Par exemple, dans le cas d'une DRAM 64K×1, la première moitié de l'adresse est envoyée à travers les 8 broches A0 – A7 et, en activant la broche RAS (pour *Row Address Strobe*), les bascules internes retiennent la première moitié de l'adresse. Ensuite la seconde moitié de l'adresse est envoyée à travers les mêmes broches et, en activant la broche CAS (pour *Column Address Strobe*), d'autres bascules internes retiennent la seconde moitié de l'adresse. On a donc utilisé dix broches (A0 – A7, RAS et CAS) au lieu de seize.

## 3.2 Interfaçage de la mémoire

À l'époque où IBM conçut son PC, les DRAM de 16 KiBits étaient largement utilisés, d'où leur utilisation sur la carte mère.

### 3.2.1 Décodage de l'adresse

Le microprocesseur présente une adresse sur ses broches d'adresse mais il a plus de broches d'adresse que les circuits intégrés de mémoire. On doit donc concevoir un circuit pour déterminer sur quelle puce aller chercher la donnée et l'adresse sur celle-ci.

Les puces de mémoire ont une broche, appelée CS (*Chip Select*) ou CE (*Chip Enable*), que l'on active pour accéder à son contenu. Les signaux de contrôle MEMR et MEMW sont reliés aux broches respectives RD et WR de la puce de mémoire. Pour lire une donnée sur la puce il faut présenter l'adresse (partielle) sur les broches d'adresse de la puce et activer les broches CS et RD. C'est une combinaison des broches d'adresse du microprocesseur, déterminant l'intervalle d'adresses correspondant à la puce mémoire, qui activera CS.

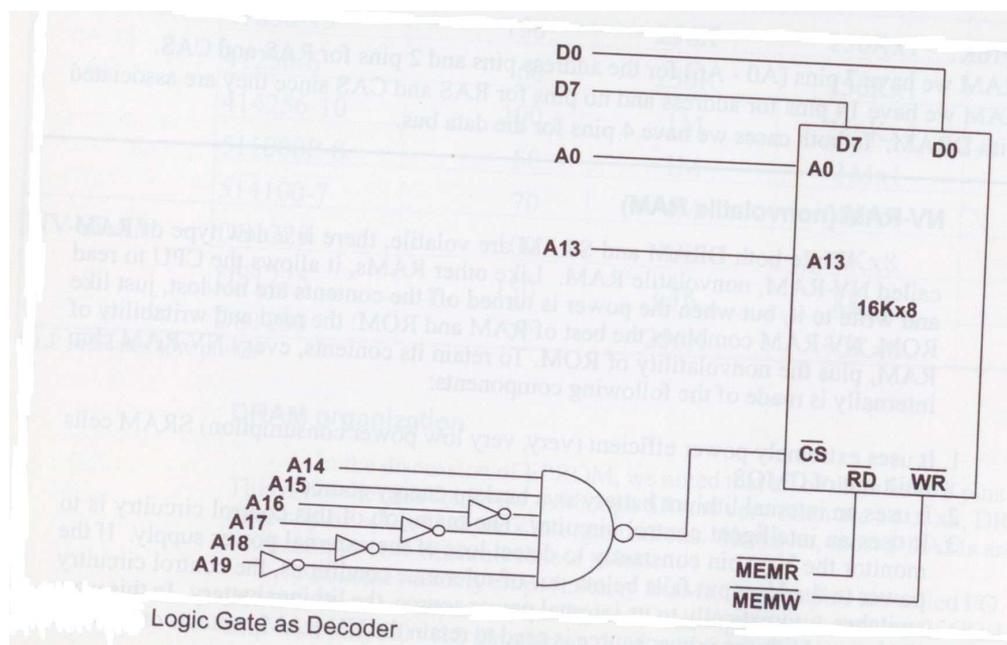


FIGURE 3.7 – Principe du décodage d'adresse

Principe d'un décodeur d'adresse.- La figure 3.7 montre le principe d'un décodeur d'adresse, en prenant l'exemple d'une puce de mémoire ayant 14 broches d'adresse et des broches  $\overline{CS}$ ,  $\overline{RD}$  et  $\overline{WR}$  actives à niveau bas.

La méthode la plus simple consiste à utiliser une porte NAND et des inverseurs. Dans notre cas,  $\overline{CS}$  est activé pour  $A19 - A14 = 000011$ , ce qui donne l'intervalle d'adresses 0C000h à 0FFFFh pour cette puce.

Bien entendu lorsque CS est actif à niveau haut, on utilise une porte AND au lieu d'une porte NAND.

Utilisation d'un décodeur 74xx138.- On utilise en fait un décodeur comme le 74LS138 (voir figure 3.8).

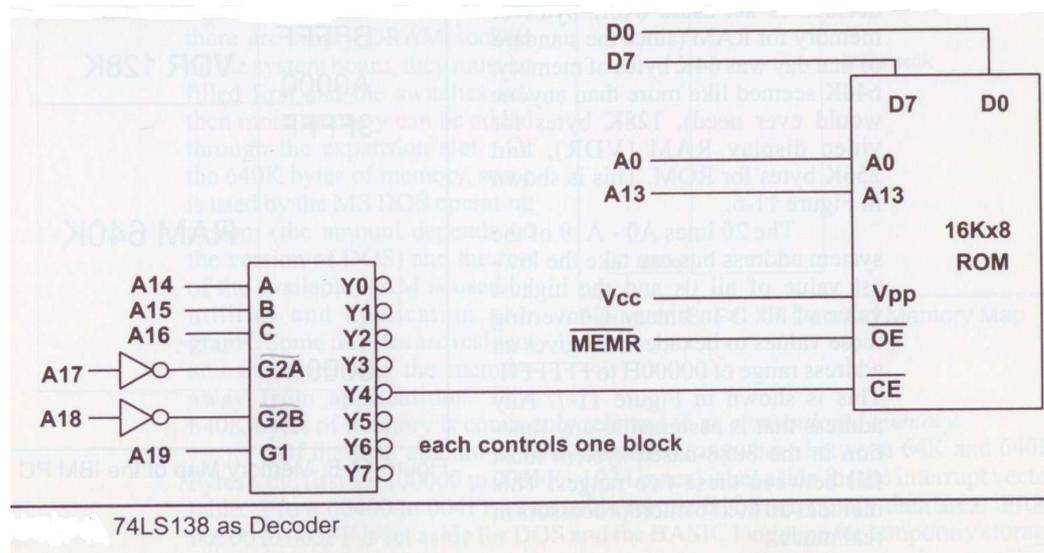


FIGURE 3.8 – Le 74LS138 utilisé pour le décodage d'adresse

Les trois broches d'entrée A, B et C activent l'une des sorties Y0 – Y7 à niveau bas : CBA est la représentation en binaire du numéro de la sortie. Chaque broche Y est reliée à la broche CS d'une puce mémoire, ce qui permet de contrôler jusqu'à 8 puces mémoire avec un seul 74LS138. Les broches d'entrée G1 (active à niveau haut),  $\overline{G2A}$  et  $\overline{G2B}$  (actives à niveau bas) sont reliées à des broches d'adresse pour sélectionner un intervalle d'adresses, éventuellement reliées en permanence à VCC et à la terre.

Dans l'exemple de la figure 3.8, l'intervalle d'adresses pour Y4 est F00000h à F3FFFh.

### 3.3 Cas de l'IBM PC

#### 3.3.1 Organisation générale de la mémoire en trois secteurs

Rappelons que le microprocesseur 8088, avec ses vingt broches, peut adresser jusqu'à un MiO, soit 1024 KiO, d'adresses comprises entre 00000h et FFFFFh. Les concepteurs de l'IBM-PC ont décidé de partager cet espace adressable en trois parties :

- Les 640 KiO compris entre 00000h et 9FFFFh sont réservés pour la *mémoire vive* proprement dite, située sur des circuits intégrés de mémoire RAM ; toutes les barrettes nécessaires ne sont pas nécessairement présentes (les premiers PC étaient livrés avec 64 KiO). On parle de **mémoire classique** (en anglais *conventional memory* d'où quelquefois la mauvaise traduction française *mémoire conventionnelle* (sic)).
- Les 128 KiO suivants, compris entre A0000h et BFFFFh, sont réservés à la **mémoire graphique** (ou VDR pour l'anglais *Video Display RAM*), contenant ce qui est affiché à l'écran. On utilise souvent pour cela des circuits intégrés de mémoire plus rapides (pour permettre le rafraîchissement de l'écran).

FFFFF	ROM 256 KiO
C0000	
BFFFF	VDR 128 KiO
A0000	
9FFFF	RAM 640 KiO
00000	

- Les 256 KiO restants, compris entre C0000h et FFFFFh, sont réservés au BIOS et à l'interpréteur BASIC, situés sur des circuits intégrés de mémoire non volatile de type ROM.

### 3.3.1.1 Implémentation de la ROM

256 KiO de ROM sont prévus pour l'IBM-PC, dont seules les adresses physiques F4000h à FFFFFh sont reliées sur la carte mère du PC/XT, comme le montre la figure 3.9. Six puces de mémoire ROM de 8K×8 sont utilisées dont les plages d'adresses sont déterminées par un 74LS138, ce qui donne un total de 8 KiO.

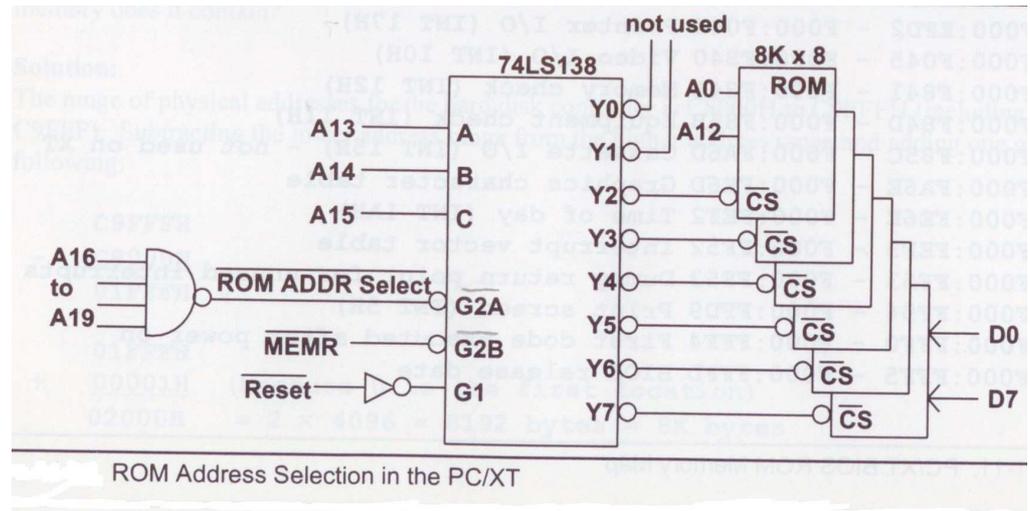


FIGURE 3.9 – Sélection des adresses de ROM pour l'IBM PC

Pour chacune des ces puces, les lignes d'adresse A0 – A12 sont reliées directement au bus des adresses, après avoir été renforcées par un 74LS244. Les lignes d'adresse A13 – A15 du bus des adresses sont reliées aux entrées A, B et C du 74LS138, ce qui permet de sélectionner l'une de ses sorties Y0 – Y7, reliée à la broche  $\overline{CS}$  de la puce ROM voulue. En fait Y0 et Y1 ne sont pas utilisées. La broche  $\overline{G2A}$ , correspondant à  $\overline{ROM\ ADDR\ Select}$ , est activée par les broches A16 – A19 du bus des adresses. La broche  $\overline{G2B}$  est activée par  $\overline{MEMR}$ . La broche G1 est activée par  $\overline{Reset}$  : lorsque le microprocesseur charge des informations, la broche RESET a un niveau bas qui est inversé pour passer à haut ; RESET est active à niveau haut ainsi lorsqu'elle est activée elle force le microprocesseur à arrêter de charger des informations à partir de la ROM.

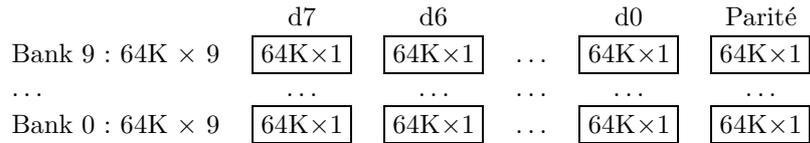
Utilisation de la ROM.- Les 256 KiO d'adresses allant de C0000h à FFFFFh ont été conçu au départ pour être de la mémoire ROM, répartie sur plusieurs barrettes. Il s'agit de quatre barrettes allouant la mémoire conformément au schéma suivant :

Emplacement mémoire	description
C000:0000 - C000:7FFF	extension 32 KiO
C000:8000 - C000:C9FF	contrôle du disque dur
C000:A000 - F000:3FFF	extension (XT) de 168 KiO
F000:4000 - F000:5FFF	ROM utilisateur de 8 KiO (SPARE)
F000:6000 - F000:DFFF	Interpréteur BASIC 32 KiO
F000:E000 - F000:FFFF	BIOS ROM 8 KiO

On s'aperçoit que deux emplacements, appelés extensions, ne sont pas utilisés par le BIOS. On parle de **mémoire haute** (ou UMB pour l'anglais *Upper Memory Block*). Dans certaines versions de PC, on y a placé des barrettes de mémoire RAM et on a utilisé ces emplacements pour étendre la mémoire vive.

3.3.1.2 Implémentation de la RAM

Banques de mémoire.- Le PC utilise de la RAM dynamique. Les circuits intégrés de mémoire sont répartis sur la carte mère s'appelle **banques de mémoire** (*memory bank*). Le premier PC utilise des circuits de mémoire 16K×1 puis IBM est passé à du 64K×1. Chaque octet stocké en mémoire vive possède un bit de parité, d'où la nécessité d'utiliser un circuit intégré de mémoire supplémentaire par banque, ce qui donne l'organisation suivante avec dix banques de 64 KiO pour arriver à 640 KiO :



Des 20 lignes d'adresse du microprocesseur, 16 d'entre elles, A0 – A15, arrivent aux circuits intégrés de DRAM 64K×1 à travers des multiplexeurs. Quatre d'entre elles, A16 – A19, sont utilisées pour sélectionner l'une des dix banques (bien qu'il y ait seize possibilités) :

A19	A18	A17	A16	A15 – A12	A11 – A8	A7 – A4	A3 – A0	Bank	Hex
0	0	0	0	0000	0000	0000	0000	Bank 0	00000
0	0	0	0	1111	1111	1111	1111		0FFFF
0	0	0	1	0000	0000	0000	0000	Bank 1	10000
0	0	0	1	1111	1111	1111	1111		1FFFF
0	0	1	0	0000	0000	0000	0000	Bank 2	20000
0	0	1	0	1111	1111	1111	1111		2FFFF
0	0	1	1	0000	0000	0000	0000	Bank 3	30000
0	0	1	1	1111	1111	1111	1111		3FFFF
0	1	0	0	0000	0000	0000	0000	Bank 4	40000
0	1	0	0	1111	1111	1111	1111		4FFFF
...	...	...	...	...	...	...	...	...	...
0	0	0	0	0000	0000	0000	0000	Bank 0	00000
0	0	0	0	1111	1111	1111	1111		0FFFF

La figure 3.10 montre comment les quatre premières banques sont décodées en utilisant un 74LS138.

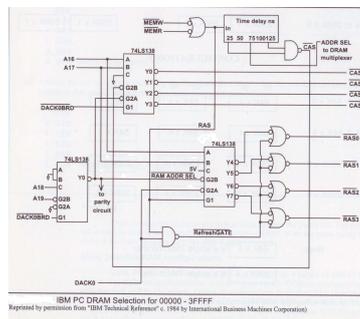


FIGURE 3.10 – Sélection des adresses de DRAM pour l'IBM PC

Puisque la DRAM nécessite de recevoir l'adresse de ligne d'abord, suivie de l'adresse de colonne, il faut un délai entre l'envoi de RAS et de CAS. Ce délai dépend de l'organisation et de la vitesse de la DRAM. Sur la figure 3.10, RAS0 et CAS0 sont reliés à RAS et CAS de la banque 0 de circuits intégrés de mémoire, RAS1 et CAS1 à la banque 1, et ainsi de suite.

Alors que l'accès aux données de la DRAM exige l'envoi de RAS puis de CAS, le rafraîchissement de la DRAM ne nécessite que l'envoi de RAS puisque le rafraîchissement s'effectue une ligne à la fois. Il y a donc deux sources de RAS : une depuis le microprocesseur pour accéder à la mémoire et l'autre depuis le DMAC pour rafraîchir la mémoire.

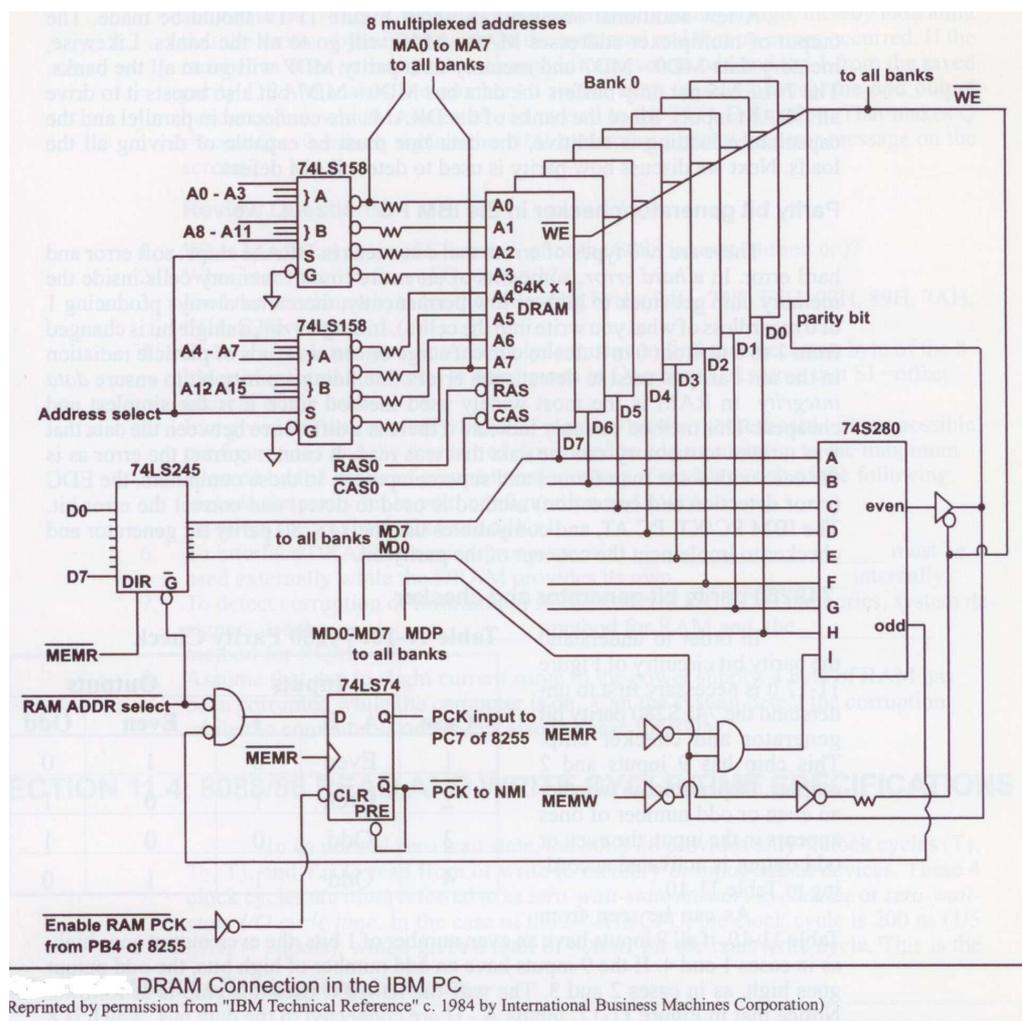


FIGURE 3.11 – Connexion de la DRAM sur l'IBM PC

La figure 3.11 montre le montage de la DRAM et du circuit du bit de parité pour la banque 0. Remarquons l'utilisation d'un 74LS158 pour multiplexer les 16 lignes d'adresse A0 – A15, les changeant en 8 lignes d'adresse MA0 – MA7 (*Multiplexed Address*). Les résistances servent de terminaison des lignes de bus car les entrées de la DRAM doivent être comprises entre 20 et 50 ohms. Dans le cas du PC à 4,7 MHz, il s'agit de 30 ohms.

Utilisation du bit de parité de la RAM.- Il y a deux types d'erreurs fréquentes pouvant survenir sur un circuit de DRAM. Une *erreur matérielle* est due au fait que certains bits restent à niveau haut ou bas en permanence. Une *erreur logicielle* est due au fait qu'un 1 se transforme en 0 (ou l'inverse) sous l'effet de la radiation de particules dans l'air. Un bit de parité est utilisé pour (essayer de) détecter de telles erreurs.

L'IBM PC/XT utilise un générateur et vérificateur de parité 74S280. Ce circuit intégré possède 9 broches d'entrée et deux broches de sortie. Les sorties **even** et **odd** sont activées comme le montre le tableau suivant :

Case	Inputs		Outputs	
	A – H	I	Even	Odd
1	Even	0	1	0
2	Even	1	0	1
3	Odd	0	0	1
4	Odd	1	1	0

Sur l'IBM-PC, les entrées A – H sont reliées au bus des données. L'entrée I est utilisée comme bit de parité. Lorsqu'un octet est écrit à un endroit donné de la DRAM, le bit de parité pair est engendré par le 74S280 et sauvegardé sur le neuvième circuit intégré de de DRAM en utilisant le signal  $\overline{\text{MEMW}}$ . Lorsqu'un octet est lu à partir de ce même endroit, le bit de parité sauvegardé parvient à la broche I du 74S280 *via*  $\overline{\text{MEMW}}$ . Cette fois, la sortie odd est prise en compte à travers un 74LS74. S'il y a une différence entre la donnée écrite et la donnée lue, la sortie Q (appelée **PCK** pour *Parity Bit Checker*) du 74LS74 est activée et  $\overline{\text{Q}}$  active la broche NMI du microprocesseur, le prévenant ainsi qu'il y a une erreur de parité. Le BIOS affiche alors un message d'erreur de parité.

Quantité de mémoire disponible.- Sur les premiers PC, seuls 64 KiO à 256 KiO étaient situés sur la carte mère, le reste étant éventuellement obtenu en ajoutant une carte d'extension (d'ajout de mémoire). Dans ce cas, pour qu'elle soit prise en compte, il fallait modifier des interrupteurs pour indiquer au BIOS et au DOS que de la mémoire avait été ajoutée.

Les adresses disponibles doivent nécessairement être consécutives : lorsqu'on rajoute de la mémoire on doit nécessairement placer un circuit intégré de mémoire pour les adresses les moins élevées non encore pourvues.

## 3.4 Commentaire du BIOS : test de la mémoire

Au démarrage, le BIOS vérifie l'intégrité du contenu de la mémoire ROM et de la mémoire RAM.

### 3.4.1 Vérification de la mémoire ROM

#### 3.4.1.1 Somme de contrôle

Introduction.- Pour s'assurer de l'intégrité du contenu de la mémoire ROM, le PC effectue un calcul de la *somme de contrôle* (en anglais *checksum*). Ce calcul permet de détecter (pas réellement à coup sûr) l'altération de données. Une des causes d'altération est la surcharge électrique, lors du démarrage de l'ordinateur en particulier.

Principe.- L'**octet de somme de contrôle** est un octet supplémentaire ajouté à la fin d'une suite d'octets de données. Le calcul de l'octet de somme de contrôle d'une suite d'octets de données s'effectue de la façon suivante :

1. on ajoute les octets en ne tenant pas compte des retenues ;
2. on prend le complément à 2 de la somme.

Pour effectuer l'opération de vérification, on ajoute donc tous les octets, y compris l'octet de somme de contrôle. Le résultat doit être zéro. S'il n'est pas nul, un ou plusieurs octets ont été changés (altérés).

Exemple.- Soient les quatre octets 25h, 62h, 3Fh et 52h.

- 1<sup>o</sup>) Calculons l'octet de somme de contrôle. Commençons par calculer la somme de ces octets :

```

25h
+ 62h
+ 3Fh
+ 52h
-----
1 18h

```

La somme est donc 118h. Ne tenons pas compte de la retenue, ce qui donne 18h. L'octet de somme de contrôle est le complément à 2 de 18h, soit E8h.

- 2<sup>o</sup>) Effectuons l'opération de vérification pour nous assurer de l'intégrité des données. Effectuons la somme, pour cela, des cinq octets :

```

25h
+ 62h
+ 3Fh
+ 52h
+ E8h
-----
2 00h

```

ce qui donne bien zéro, en enlevant la retenue.

- 3°) Montrer que si le second octet a été changé en 22h, la vérification de la somme détecte l'erreur. Effectuons la somme des cinq octets :

```

25h
+ 22h
+ 3Fh
+ 52h
+ E8h
-----
1 C0h

```

ce qui ne donne pas zéro mais C0h.

### 3.4.1.2 Programme de vérification du BIOS

Le code du BIOS d'IBM consacré à la vérification de l'intégrité de la ROM commence ligne 387 :

```

EOC7 8CC8          387      MOV     AX,CS           ; SETUP SS SEG REG
EOC9 8ED0          388      MOV     SS,AX
EOCB 8ED8          389      MOV     DS,AX           ; SET UP DATA SEG TO POINT TO
                               390                      ; ROM ADDRESS
EOCD FC           391      CLD                     ; SET DIRECTION FLAG TO INC.
                               392      ASSUME  SS:CODE
EOCE BB00ED       393      MOV     BX,0E000H       ; SETUP STARTING ROS ADDR
EOD1 BC16E0       394      MOV     SP,OFFSET C1    ; SETUP RETURN ADDRESS
EOD4 E91B18       395      JMP     ROS_CHECKSUM
EOD7 7704         396  C11:  JNE     ERR01           ; HALT SYSTEM IF ERROR

```

Les segments de pile (lignes 387 et 388) et de données (lignes 389 et 390) sont les mêmes que le segment de code, c'est-à-dire qu'ils correspondent à la ROM BIOS, comme nous l'avons vu au chapitre 2.

On parcourt les données dans l'ordre croissant (ligne 391).

On place dans BX l'adresse de début de la ROM BIOS (ligne 393).

On fait appel à une sous-routine pour vérifier la ROM BIOS (ligne 395), l'adresse de retour étant celle de C1. S'il y a une erreur dans la vérification, on arrête tout bonnement le système (ligne 396).

La sous-routine ROS\_CHECKSUM est définie à partir de ligne 5232 :

```

                               5232
                               5233 ;-----
                               5234 ;      ROS CHECKSUM SUBROUTINE      :
                               5235 ;-----
F8F2          5236 ROS_CHECKSUM  PROC  NEAR           ; NEXT_ROS_MODULE
F8F2 B90020    5237      MOV     CX,8192           ; NUMBER OF BYTES TO ADD
F8F5          5238 ROS_CHECKSUM_CNT: ; ENTRY FOR OPTIONAL ROS TEST
F8F5 32C0      5239      XOR     AL,AL
F8F7          5240 C26:
F8F7 0207      5241      ADD     AL,DS:[BX]
F8F9 43        5242      INC     BX                     ; POINT TO NEXT BYTE
F8FA E2FB      5243      LOOP   C26                     ; ADD ALL BYTES IN ROS MODULE
F8FC 0AC0      5244      OR     AL,AL                     ; SUM = 0?
F8FE C3        5245      RET
                               5246 ROS_CHECKSUM  ENDP

```

Autrement dit, le nombre d'octets à vérifier est celui de la ROM BIOS, soit 8 KiO (ligne 5237) ; on initialise le registre AL, dans lequel on va effectuer la somme, à zéro (ligne 5239) ; on lui ajoute les octets de la ROM (lignes 5241 à 5243) et on renvoie 0 dans l'indicateur CF si la somme est nulle et 1 sinon (ligne 5244).

### 3.4.2 Zone de la mémoire vive réservée au BIOS

Le BIOS a besoin de mémoire vive. Le premier KiO est réservé à la table des interruptions, comme exigé par le microprocesseur. Le reste est défini au tout début du BIOS :

```

35 ;-----
36 ;      8088 INTERRUPT LOCATIONS      :
37 ;-----
38
----
39 ABSO  SEGMENT  AT 0
0000 40 STG_LOCO LABEL BYTE
0008 41          ORG  2*4
0008 42 NMI_PTR LABEL WORD
0014 43          ORG  5*4
0014 44 INT5_PTR LABEL WORD
0020 45          ORG  8*4
0020 46 INT_ADDR LABEL WORD
0020 47 INT_PTR LABEL DWORD
0040 48          ORG 10H*4
0040 49 VIDEO_INT LABEL WORD
0074 50          ORG 1DH*4
0074 51 PARM_PTR LABEL DWORD          ; POINTER TO VIDEO PARMS
0060 52          ORG 1BH*4
0060 53 BASIC_PTR LABEL WORD          ; ENTRY POINT FOR CASSETTE BASIC
0078 54          ORG 01EH*4          ; INTERRUPT 1EH
0078 55 DISK_POINTER LABEL DWORD
007C 56          ORG 01FH*4          ; LOCATION OF POINTER
007C 57 EXT_PTR LABEL DWORD          ; POINTER TO EXTENSION
0400 58          ORG 400H
0400 59 DATA_AREA LABEL BYTE          ; ABSOLUTE LOCATION OF DATA SEGMENT
0400 60 DATA_WORD LABEL WORD
0500 61          ORG 0500H
0500 62 MFG_TEST_RTN LABEL FAR
7C00 63          ORG 7C00H
7C00 64 BOOT_LOCN LABEL FAR
----
65 ABSO  ENDS
66

```

Autrement dit :

- On spécifie quelques vecteurs d'interruption de la table des vecteurs d'interruption (lignes 40 à 56).
- La zone 400h à 4FFh est réservée à la *zone de communication du BIOS*, sur laquelle nous allons revenir à la prochaine section.
- La zone 500h à 5FFh est réservée au code supplémentaire spécifié par le constructeur du clavier, comme nous le verrons plus tard.
- La zone de démarrage, que l'on récupérera sur le disque, sera placée à l'adresse 7C00h.

### 3.4.3 Zone de communication du BIOS

Zones du BIOS en mémoire vive.- La mémoire classique n'est pas entièrement disponible pour les programmes et les données. Une partie de celle-ci est occupée pour des données temporaires du BIOS et pour le système d'exploitation, données qui ne peuvent pas se trouver en ROM.

Les 1 536 premiers octets sont occupés par des données du BIOS :

- le premier KiO (00000h à 003FFh) est réservé à la *table des vecteurs d'interruption*;
- les 256 octets suivants de 00400h à 004FFh sont réservés à des données temporaires utilisées par le BIOS, ce qui est appelé **zone de communication du BIOS** (*ROM BIOS Data Area*).

Zone de communication du BIOS.- La zone de communication du BIOS est structurée suivant le tableau (partiel) ci-dessous (lignes 75 à 215 du listing du BIOS de l'IBM PC-XT) :

Adresses mémoire	Octets	Description
0000:0400 - 0000:0401	2	adresse du port de com1
0000:0402 - 0000:0403	2	adresse du port de com2
0000:0404 - 0000:0405	2	adresse du port de com3
0000:0406 - 0000:0407	2	adresse du port de com4
0000:0408 - 0000:0409	2	adresse du port de lpt1
0000:040A - 0000:040B	2	adresse du port de lpt2
0000:040C - 0000:040D	2	adresse du port de lpt3
0000:040E - 0000:040F	2	adresse du port de lpt4
0000:0410 - 0000:0411	2	liste du matériel installé
0000:0412 - 0000:0412	1	initialisation du registre des indicateurs
0000:0413 - 0000:0414	2	taille de la mémoire (en KiO)
0000:0415 - 0000:0416	2	mémoire (éventuelle) dans les canaux d'entrée-sortie
0000:0417 - 0000:0418	2	indicateur du statut du clavier
0000:0419 - 0000:0419	1	stockage de l'entrée d'une touche spéciale
0000:041A - 0000:041B	2	pointeur du tampon du clavier (début)
0000:041C - 0000:041D	2	pointeur du tampon du clavier (fin)
0000:041E - 0000:043D	32	tampon du clavier
0000:043E - 0000:0448	11	secteur des données de la disquette
0000:0449 - 0000:0449	1	mode graphique en cours
0000:044A - 0000:044B	2	nombre de colonnes à l'écran
0000:044C - 0000:044D	2	longueur du tampon de régénération (en octets)
0000:044E - 0000:044F	2	décalage du début du tampon de régénération
0000:0450 - 0000:045F	16	position du curseur, pages 1-8
0000:0460 - 0000:0460	1	ligne de la fin du curseur
0000:0461 - 0000:0461	1	ligne de début du curseur
0000:0462 - 0000:0462	1	numéro de la page en cours
0000:0463 - 0000:0464	2	adresse de base du moniteur actif
0000:0465 - 0000:0465	1	registre du mode CRT pour MDA et CGA
0000:0466 - 0000:0466	1	registre pour CGA
0000:0467 - 0000:046B	5	secteur des données pour le lecteur de cassette
0000:046C - 0000:0470	5	secteur des données du minuteur

Voilà ce que dit le BIOS :

```

76 ;-----
77 ; ROM BIOS DATA AREAS :
78 ;-----
79
----
80 DATA SEGMENT AT 40H
0000 (4 81 RS232_BASE DW 4 DUP(?) ; ADDRESSES OF RS232 ADAPTERS
    ????)
    )
0008 (4 82 PRINTER_BASE DW 4 DUP(?) ; ADDRESSES OF PRINTERS
    ????)
    )
0010 ????) 83 EQUIP_FLAG DW ? ; INSTALLED HARDWARE
0012 ?? 84 MFG_TST DB ? ; INITIALIZATION FLAG
0013 ????) 85 MEMORY_SIZE DW ? ; MEMORY SIZE IN K BYTES
0015 ?? 86 MFG_ERR_FLAG DB ? ; SCRATCHPAD FOR MANUFACTURING
0016 ?? 87 DB ? ; ERROR CODES
88
89 ;-----
90 ; KEYBOARD DATA AREAS ;
91 ;-----
92
0017 ?? 93 KB_FLAG DB ?
94
95 ;----- SHIFT FLAG EQUATES WITHIN KB_FLAG
96
0080 97 INS_STATE EQU 80H ; INSERT STATE IS ACTIVE
0040 98 CAPS_STATE EQU 40H ; CAPS LOCK STATE HAS BEEN TOGGLED
0020 99 NUM_STATE EQU 20H ; NUM LOCK STATE HAS BEEN TOGGLED
0010 100 SCROLL_STATE EQU 10H ; SCROLL LOCK STATE HAS BEEN TOGGLED
0008 101 ALT_SHIFT EQU 08H ; ALTERNATE SHIFT KEY DEPRESSED
0004 102 CTL_SHIFT EQU 04H ; CONTROL SHIFT KEY DEPRESSED
0002 103 LEFT_SHIFT EQU 02H ; LEFT SHIFT KEY DEPRESSED
0001 104 RIGHT_SHIFT EQU 01H ; RIGHT SHIFT KEY DEPRESSED
105
0018 ?? 106 KB_FLAG_1 DB ?
107
0080 108 INS_SHIFT EQU 80H ; INSERT KEY IS DEPRESSED
0040 109 CAPS_SHIFT EQU 40H ; CAPS LOCK KEY IS DEPRESSED
0020 110 NUM_SHIFT EQU 20H ; NUM LOCK KEY IS DEPRESSED
0010 111 SCROLL_SHIFT EQU 10H ; SCROLL LOCK KEY IS DEPRESSED
0008 112 HOLD_STATE EQU 08H ; SUSPEND KEY HAS BEEN TOGGLED
113
0019 ?? 114 ALT_INPUT DB ? ; STORAGE FOR ALTERNATE KEYPARD ENTRY
001A ????) 115 BUFFER_HEAD DW ? ; POINTER TO HEAD OF KEYBOARD BUFFER
001C ????) 116 BUFFER_TAIL DW ? ; POINTER TO TAIL OF KEYBOARD BUFFER
001E (16 117 KB_BUFFER DW 16 DUP(?) ; ROOM FOR IS ENTRIES
    ????)
    )
003E 118 KB_BUFFER_END LABEL WORD
119
120 ;----- HEAD = TAIL INDICATES THAT THE BUFFER IS EMPTY
121
0045 122 NUM_KEY EQU 69 ; SCAN CODE FOR NUMBER LOCK
0046 123 SCROLL_KEY EQU 70 ; SCROLL LOCK KEY
0038 124 ALT_KEY EQU 56 ; ALTERNATE SHIFT KEY SCAN CODE
001D 125 CTL_KEY EQU 29 ; SCAN CODE FOR CONTROL KEY
003A 126 CAPS_KEY EQU 58 ; SCAN CODE FOR SHIFT LOCK
002A 127 LEFT_KEY EQU 42 ; SCAN CODE FOR LEFT SHIFT
0036 128 RIGHT_KEY EQU 54 ; SCAN CODE FOR RIGHT SHIFT
0052 129 INS_KEY EQU 82 ; SCAN CODE FOR INSERT KEY
0053 130 DEL_KEY EQU 83 ; SCAN CODE FOR DELETE KEY
131
132 ;-----
133 ; DISKETTE DATA AREAS :
134 ;-----
003E ?? 135 SEEK_STATUS DB ? ; DRIVE RECALIBRATION STATUS
136 ; BIT 3-0 = DRIVE 3-0 NEEDS RECAL
137 ; BEFORE NEXT SEEK IF BIT IS = 0
138
0080 139 INT_FLAG EQU 080H ; INTERRUPT OCCURRENCE FLAG
003F ?? 140 MOTOR_STATUS DB ? ; MOTOR STATUS
141 ; BIT 3-0 = DRIVE 3-0 IS CURRENTLY

```



```

??
)
210 ;-----
211 ;      ADDITIONAL KEYBOARD DATA AREA      :
212 ;-----
0080 ???? 213 BUFFER_START DW      ?
0082 ???? 214 BUFFER_END   DW      ?
----      215 DATA      ENDS
216 ;-----
217 ;      EXTRA DATA AREA                    :
218 ;-----
----      219 XXDATA SEGMENT AT 50H
0000 ??   220 STATUS_BYTE DB      ?
----      221 XXDATA ENDS

```

Cet emplacement mémoire est structuré en plusieurs zones que nous allons passer en revue l'une après l'autre.

### 3.4.3.1 Périphériques présents

Description.- Les dix-huit premiers octets permettent de connaître des renseignements sur les périphériques présents. Cela débute par les adresses des ports séries et parallèles :

Adresses mémoire	Octets	Description
0000:0400 - 0000:0401	2	adresse du port de com1
0000:0402 - 0000:0403	2	adresse du port de com2
0000:0404 - 0000:0405	2	adresse du port de com3
0000:0406 - 0000:0407	2	adresse du port de com4
0000:0408 - 0000:0409	2	adresse du port de lpt1
0000:040A - 0000:040B	2	adresse du port de lpt2
0000:040C - 0000:040D	2	adresse du port de lpt3
0000:040E - 0000:040F	2	adresse du port de lpt4
0000:0410 - 0000:0411	2	liste du matériel installé

Les seize bits d'adresse 410h donnent la liste du matériel installé conformément au schéma suivant :

Bits	Matériel
15-14	Nombre de ports parallèles
11-9	Nombre de ports séries RS232
7-6	Nombre de lecteurs de disquettes 00 = 1, 01 = 2, 10 = 3, 11 = 4
5-4	Mode graphique initial : 00 = non utilisé, 01 = 40 x 25 couleur 10 = 80 x 25 couleur 11 = 80 x 25 monochrome
2	souris installée si 1
1	coprocasseur mathématique si 1
0	lecteur de disquette si 1

Détermination du contenu.- Commentons par un exemple de zone de communication du BIOS. Déterminons le contenu grâce à `debug` :

```
C:>debug
-d 0000:0400
0000:0400  F8 03 F8 02 00 00 00 00-78 03 00 00 00 00 0C 02  .....x.....
0000:0410  27 C4 00 80 02 00 00 20-00 00 2E 00 2E 00 30 0B  '.....0.
0000:0420  08 0E 08 0E 08 0E 34 05-30 0B 30 0B 0D 1C 30 0B  .....4.0.0...0.
0000:0430  30 0B 30 0B 30 0B 3A 34-30 0B 30 0B 34 05 00 80  0.0.0.:40.0.4...
0000:0440  00 00 C0 00 00 00 00 00-00 03 50 00 00 10 00 00  .....P.....
0000:0450  00 0C 00 00 00 00 00 00-00 00 00 00 00 00 00 00  .....
0000:0460  0E 0D 00 D4 03 29 30 A4-17 D8 84 FF DE 9F 10 00  .....)0.....
0000:0470  00 00 00 00 00 01 00 00-14 14 14 3C 01 01 01 01  .....<....
-d
0000:0480  1E 00 3E 00 18 10 00 60-F9 11 0B C8 58 00 00 01  ..>....'....X...
0000:0490  07 07 00 00 00 00 10 12-00 00 00 00 00 00 00 00  .....
0000:04A0  00 00 00 10 12 00 C0 00-A2 03 00 C0 10 10 00 00  .....
0000:04B0  00 00 00 01 D1 00 44 00-00 00 04 00 00 00 00 00  .....D.....
0000:04C0  00 00 00 00 00 00 0A 00-00 00 00 00 00 00 00 00  .....
0000:04D0  00 00 00 00 00 00 00 00-DC 04 00 00 00 00 00 00  .....
0000:04E0  00 00 00 00 00 00 00 80-07 00 08 00 00 00 00 00  .....
0000:04F0  AA 00 00 00 00 00 00 00-1C 00 50 01 00 00 00 03  .....P.....
-
```

puis commentons-le.

Nous pouvons en tirer les renseignements suivants :

- les deux premiers octets, `F8h` et `03h`, donnent l'adresse du port de COM1, ici `03F8h`.
- les deux octets suivants donnent l'adresse du port de COM2, ici `02F8h`.
- les quatre octets suivants sont nuls, ce qui indique qu'il n'y a que deux ports séries présents sur notre système.
- les deux octets suivants, `78h` et `03h`, donnent l'adresse du port de LPT1, ici `0378h`.
- les quatre octets suivants sont nuls, ce qui indique qu'il n'y a ni port LPT2 ni port LPT3.
- les deux octets suivants donnent l'adresse du port LPT4, ici `020Ch`.
- les deux octets d'adresse `410h`, ici `27h` et `C4h`, donnent la liste du matériel. On a `C427h = 00 11 010 0 00 10 0 1 1 1b`, soit deux ports parallèles (LPT1 et LPT4 d'après ce que nous avons vu ci-dessus), deux ports séries (COM1 et COM2), un lecteur de disquette, un mode graphique initial de 25 lignes de 40 colonnes en couleur, une souris, un coprocesseur mathématique et au moins un lecteur de disquettes.

### 3.4.3.2 Valeur d'initialisation du registre des indicateurs

Description.- L'octet suivant donne la valeur d'initialisation du registre des indicateurs :

Adresses mémoire	Octets	Description
0000:0412 - 0000:0412	1	initialisation du registre des indicateurs

Exemple.- Dans notre cas, l'octet d'adresse `412h` vaut `00h`, ce qui montre que le registre des indicateurs est initialisé à zéro.

**3.4.3.3 Taille mémoire**

Description.- On a ensuite des indications sur la taille de la mémoire :

Adresses mémoire	Octets	Description
0000:0413 - 0000:0414	2	taille de la mémoire (en KiO)
0000:0415 - 0000:0416	2	mémoire sur carte d'extension (en KiO)

Sur les premiers modèles d'IBM PC, la valeur indiquée était fondée sur les petits interrupteurs et le BIOS pouvait ainsi indiquer moins que la mémoire réellement installée.

Exemple.- Dans notre cas, les deux octets d'adresses 413h et 414h donnent pour quantité de mémoire 8002h, qu'il faut interpréter ici comme  $32\ 768\ \text{KiO} \times 2$ , soit 64 MiO.

Les deux octets suivants montrent qu'il n'y a pas de mémoire sur carte d'extension (cette façon d'ajouter de la mémoire était réservée aux premiers PC).

**3.4.3.4 Premier secteur des données du clavier**

Description.- Les octets 417h à 43Dh constituent un premier secteur concernant le clavier, conformément au tableau suivant :

Adresses mémoire	Octets	Description
0000:0417 - 0000:0418	2	indicateur du statut du clavier
0000:0419 - 0000:0419	1	stockage de l'entrée d'une touche spéciale
0000:041A - 0000:041B	2	pointeur du tampon du clavier (début)
0000:041C - 0000:041D	2	pointeur du tampon du clavier (fin)
0000:041E - 0000:043D	32	tampon du clavier

L'octet 417h donne le statut des touches spéciales selon le tableau suivant :

Bit	Signification
7	Insertion active
6	Majuscule bloquée activée
5	Nombre bloqué actif
4	Scroll Lock actif
3	Alt pressé
2	Ctrl pressé
1	Left shift pressé
0	Right shift pressé

L'octet 418h continue à donner le statut des touches spéciales conformément au tableau suivant :

Bit	Signification
7	Insertion pressée
6	Majuscule bloquée pressé
5	Nombre bloqué pressé
4	Scroll Lock pressé
3	Ctrl/NumLock pressé
2	SysReq pressé
1	Left Alt pressé
0	Right Ctrl pressé

« Pressé » signifie que l'on vient d'appuyer sur la touche correspondante alors que « Actif » signifie que l'on a appuyé un nombre impair de fois sur la touche et donc que la fonction est activée.

Exemple.- Dans le cas de notre exemple, nous obtenons les données suivantes :

- l'octet 417h est à 20h donc NumLock est actif;
- l'octet suivant est nul, donc aucune touche spéciale n'a été pressée;
- l'octet suivant est nul, donc on n'a pas pressé de touche permettant de modifier le code de la touche suivante;
- les deux octets suivants sont 2Eh et 00h, ce qui donne l'adresse 002Eh du début du tampon du clavier;
- les deux octets suivants ont la même valeur, ce qui donne l'adresse de fin du tampon du clavier. Il n'y a donc ici qu'un seul caractère frappé au clavier non encore utilisé;
- les 32 octets suivants permettent de stocker le tampon du clavier. Un seul octet nous intéresse ici, comme nous venons de le voir, celui d'adresse 42Eh, soit 30h, c'est-à-dire le caractère '0'.

### 3.4.3.5 Secteur des données des lecteurs de disquettes

Description.- On a la structure suivante :

Adresses mémoire	Octets	Description
0000:043E	1	statut de recherche
0000:043F	1	statut du moteur
0000:0440	1	compteur d'arrêt
0000:0441	1	statut du disque
0000:0442 - 0000:0448	7	statut du contrôleur

L'octet 43Eh concerne le *statut de recherche du disque*, le bit numéro 1 faisant référence au lecteur A, le bit numéro 2 à B, le numéro 3 à C et le numéro 4 à D. Un bit de valeur 0 signifie que la prochaine recherche consiste à se repositionner sur le cylindre 0 pour recalibrer le lecteur.

L'octet 43Fh concerne le *statut du moteur*. Si le bit 7 est à 1, c'est qu'une opération en écriture est en train de s'effectuer. Le bit numéro 0 fait référence au lecteur A, le bit numéro 1 à B, le bit numéro 2 à C, le numéro 3 à D. Un bit de valeur 0 indique que le moteur est en marche.

L'octet 440h est un compteur faisant connaître dans combien de temps il faudra arrêter le moteur.

L'octet 441h concerne le statut du disque, indiquant l'éventuel type d'erreur lors de la dernière opération sur un lecteur de disquettes :

Valeur	Erreur
00h	Pas d'erreur
01h	Paramètre de disque non valide
02h	Adresse non trouvée
03h	Erreur de protection en écriture
04h	Secteur non trouvé
08h	DMA surchargée
09h	Essai d'aller au-delà de 64 KiO en DMA
0Ch	Type de média non trouvé
10h	Erreur CRC en lecture
20h	Erreur contrôleur
40h	Échec de la recherche
80h	Lecteur non prêt

Les octets 42h à 48h concernent le statut du contrôleur de lecteur de disquettes.

Remarque.- Il n'est prévu qu'un seul type de contrôleur de lecteurs de disquettes et jusqu'à quatre lecteurs de disquettes (pour les modèles XT, les modèles à partir du modèle AT ne pouvant être pourvu que de deux lecteurs de disquettes).

Exemple.- Dans le cas de notre exemple, nous avons les valeurs suivantes :

- l'octet 43Eh est nul, il n'y a donc de recherche en cours sur aucun des lecteurs de disquettes ;
- l'octet 43Fh vaut 80h, ce qui signifie *a priori* qu'on est en train d'écrire sur une disquette, mais aucun moteur ne fonctionne ce qui est curieux ;
- l'octet 440h est nul, c'est-à-dire qu'il n'y a pas d'attente avant que le moteur du lecteur de disquettes soit arrêté, ce qui est normal puisqu'aucun moteur n'est en fonctionnement ;

- l'octet 441h est nul, ce qui indique qu'il n'y a pas eu d'erreur lors de la dernière tentative d'accéder à la disquette.

### 3.4.3.6 Première zone de données graphiques

Description.- La structure de la première zone de données graphiques est la suivante :

Adresses mémoire	Octets	Description
0000:0449 - 0000:0449	1	mode graphique actuel
0000:044A - 0000:044B	2	nombre de colonnes à l'écran
0000:044C - 0000:044D	2	taille du tampon graphique (en octets)
0000:044E - 0000:044F	2	décalage du début du tampon graphique
0000:0450 - 0000:045F	16	positions du curseur, pages 0-7
0000:0460 - 0000:0460	1	ligne de la fin du curseur
0000:0461 - 0000:0461	1	ligne de début du curseur
0000:0462 - 0000:0462	1	numéro de la page en cours
0000:0463 - 0000:0464	2	adresse du port du moniteur actif
0000:0465 - 0000:0465	1	valeur du registre de mode graphique
0000:0466 - 0000:0466	1	palette des couleurs (pour CGA)

Le mode graphique de l'octet 449h est indiqué par un bit :

Bit	Mode
7	Monochrome MDA
6	640 x 200 monochrome
5	320 x 200 monochrome
4	320 x 200 couleur
3	80 x 25 couleur
2	80 x 25 monochrome
1	40 x 25 couleur
0	40 x 25 monochrome

La valeur du mot d'adresse 463h est 03B4h pour une carte monochrome MDA et 03D4h pour une carte couleur CGA.

Exemple.- Dans le cas de notre exemple, les valeurs sont les suivantes :

- l'octet 449h vaut 03h, nous sommes donc en mode couleur de 25 lignes de 80 colonnes ;
- le mot 450h vaut 0050h, il y a donc 80 colonnes à l'écran ;
- le mot 44Ch vaut 1000h, la taille du tampon de page est donc de 4 096 octets ;
- le mot 44Eh est égal à 0000h, d'où un décalage de 0 dans le tampon de page ;
- le mot 450h vaut 0C00h, d'où la position du curseur de la page numéro 0 ; les autres pages ne sont pas utilisées, le curseur est donc au tout début de page ;
- les octets 460h et 461h valent 0Eh et 0Dh, le curseur occupe donc deux lignes ;
- l'octet 462h est nul, ce qui indique que la page affichée est la page numéro 0 ;
- le mot 463h vaut 03D4h, ce qui donne l'adresse du port du moniteur actif, qui est donc un moniteur couleur ;
- l'octet 465h vaut 29h et l'octet 466h vaut 30h, ce qu'il est difficile d'interpréter pour l'instant.

**3.4.3.7 Zone des données système**

Description.- La zone des données système est structurée de la façon suivante :

Adresses mémoire	Octets	Description
0000:0467 - 0000:046B	5	secteur des données pour le lecteur de cassettes

Puis :

Adresses mémoire	Octets	Description
0000:0467h - 0000:0468h	2	Data-edge time count
0000:0469h - 0000:046Ah	2	Cyclical redundancy check (CRC) register
0000:046Bh	1	Last input value

On a ensuite :

Adresses mémoire	Octets	Description
0000:046C - 0000:0470	5	secteur des données du temporisateur

Plus précisément :

Adresses mémoire	Octets	Description
0000:046C - 0000:046D	2	Partie inférieure du temporisateur
0000:046E - 0000:046F	2	Partie supérieure du temporisateur
0000:0470	1	Timer overflow (1 if timer has passed midnight)
0000:0471	1	Ctrl+Break keys set bit 7 to 1
0000:0472 - 0000:0473	2	Memory reset flag

En ce qui concerne le dernier mot, celui d'adresse 472h, si son contenu est 1234h, le raccourci clavier Ctrl+Alt+Del provoquera un redémarrage « à chaud ».

Exemple.-  
à faire

### 3.4.3.8 Zone des données du disque dur

Description.- La structure de la zone des données du disque dur est la suivante :

Adresses mémoire	Octets	Description
0000:0474	1	Statut de la dernière opération sur le disque dur
0000:0475	1	Nombre de disques durs montés

Le code de l'octet 474h doit être compris de la façon suivante :

Code	Statut
00h	Pas d'erreur
01h	Mauvaise commande (non reconnue par le contrôleur)
02h	Adresse non trouvée sur le disque
03h	Essai d'écrire sur un disque protégé
04h	Piste/secteur non valide
05h	Échec de l'opération de réinitialisation
06h	Disquette retirée depuis le dernier accès
07h	Mauvais paramètres du lecteur
08h	DMA surchargée (données à entrer trop rapides)
09h	Essai de DMA au-delà de 64 KiO en lecture-écriture
10h	Rencontre d'un mauvais CRC en lecture (indique une donnée corrompue)
20h	Échec du contrôleur (échec matériel)
40h	Échec de recherche (échec matériel)
80h	Le périphérique ne répond pas disquette : porte ouverte ou pas de disquette, disque dur : problème de temps
AAh	Lecteur non prêt
BBh	Erreur non définie
CCh	Problème en écriture

Exemple.- Dans notre cas, les valeurs sont les suivantes :

- l'octet 474h est nul, ce qui indique qu'il n'y a pas eu d'erreur lors du dernier essai d'accès au disque dur ;
- l'octet 475h vaut 1, c'est-à-dire qu'il n'y a qu'un seul disque dur de monté.

### 3.4.3.9 Zone des données de sortie de session des ports

Description.- Les octets 478h à 47Bh donnent le dernier caractère sorti (en anglais *time-out*) sur les ports parallèles (LPT1 à LPT4).

Les octets 47Ch à 47Fh donnent le dernier caractère sorti sur les ports séries (COM1 à COM4).

Exemple.- Dans notre cas, les données sont les suivantes :

- les octets 478h à 47Ah sont égaux à 14h et l'octet 47Bh à 3Ch ;
- les octets 47Ch à 47Fh sont tous égaux à 01h.

#### 3.4.3.10 Deuxième zone de données du clavier

Description.- Le mot d'adresse 480h donne le décalage (*offset*) de l'adresse du début du tampon du clavier. Le mot d'adresse 482h donne le décalage de l'adresse de la fin du tampon du clavier.

Exemple.- Dans le cas de notre exemple, le mot d'adresse 480h est égal à 001Eh, celui d'adresse 482h est égal à 003E.

#### 3.4.4 Test de la RAM

Nous l'étudierons au chapitre 10 car le code fait intervenir le PIP que nous verrons au chapitre suivant.