

Module : Microprocesseurs et API

Partie 1 :

Microprocesseur

1. Architecture de base d'un microprocesseur

1.1 définition d'un μP

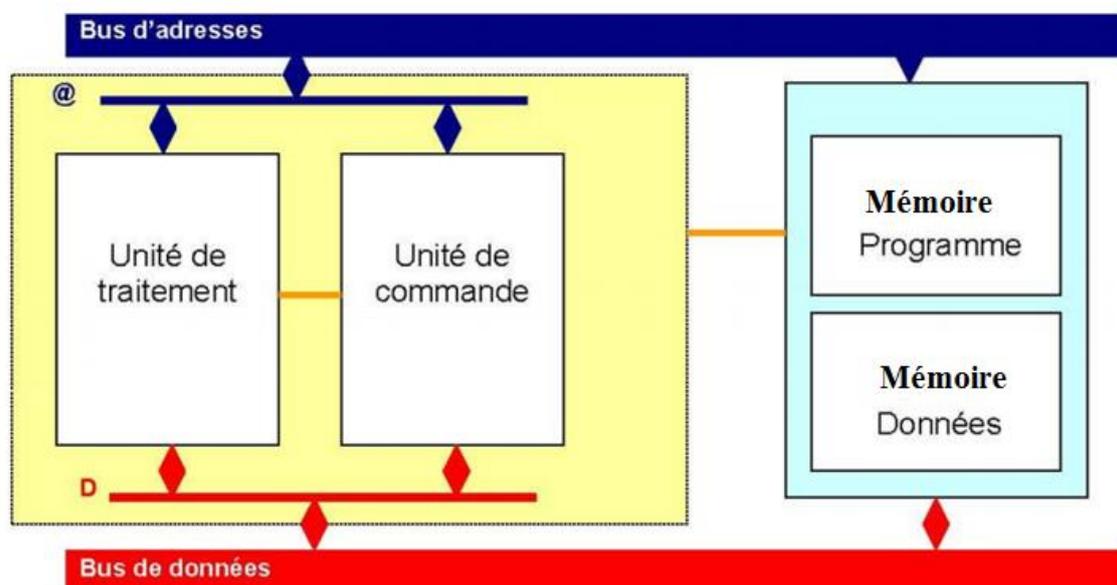
Le microprocesseur (processeur) est le composant hardware le plus connu d'un système micro-programmé, noté aussi M.P.U. (Microprocessor unit) ou encore C.P.U. (Central Processing Unit). C'est l'unité intelligente de traitement des informations. Son travail consiste à lire des programmes (des suites d'instructions), à les décoder et à les exécuter. Les années 80 voyaient l'émergence de ces circuits avec les Z80, 6800 de Motorola, le 8085 de Intel qui est souvent utilisé en tant que microcontrôleur.

Il doit aussi prendre en compte les informations extérieures au système et assurer leur traitement.

1.2 Composition d'un μP

Un microprocesseur est constitué de :

- une unité de commande qui lit les instructions et les décode;
- une unité de traitement (UAL - unité arithmétique et logique) qui exécute les instructions;
- d'un ensemble de mémoire appelés registres;
- d'un bus de données externe;
- d'un bus d'adresse externe;
- d'un bus de commande externe;
- d'un bus de données interne reliant l'unité de commande l'UAL et les registres.



Lorsque tous ces éléments sont assemblés sur une même puce, on parle alors de microprocesseur.

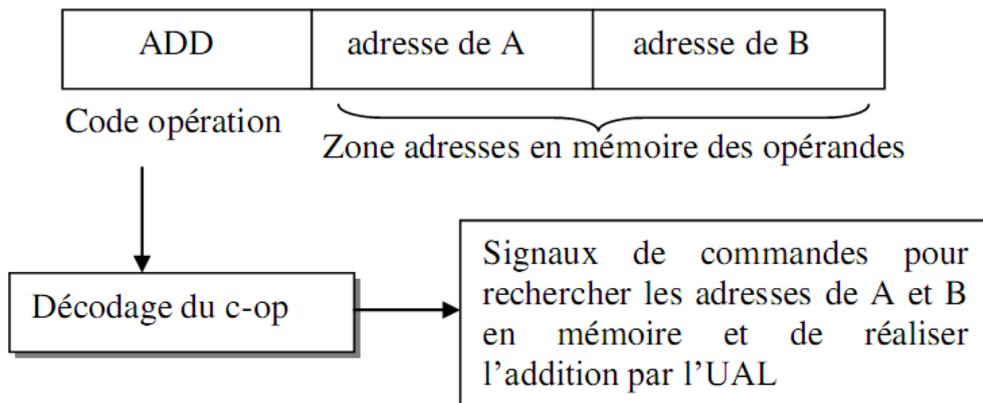
1.2.1 Unité de commande

Elle permet de "séquencer" le déroulement des instructions. Elle effectue la recherche en mémoire du programme, le décodage, le pilotage de l'exécution des instructions via les signaux de l'horloge ainsi que la préparation de l'instruction suivante. L'unité de commande élabore tous les signaux de synchronisation internes ou externes (bus des commandes) au microprocesseur.

Elle est chargée également à travers le décodeur d'instruction de décomposer et analyser l'instruction se trouvant dans le registre d'instructions, selon le code opération, elle envoie la nature des opérations à effectuer à l'unité de commande et précisément le séquenceur qui va ensuite générer les microcommandes nécessaires aux différents composants participant à l'exécution de l'instruction en cours.

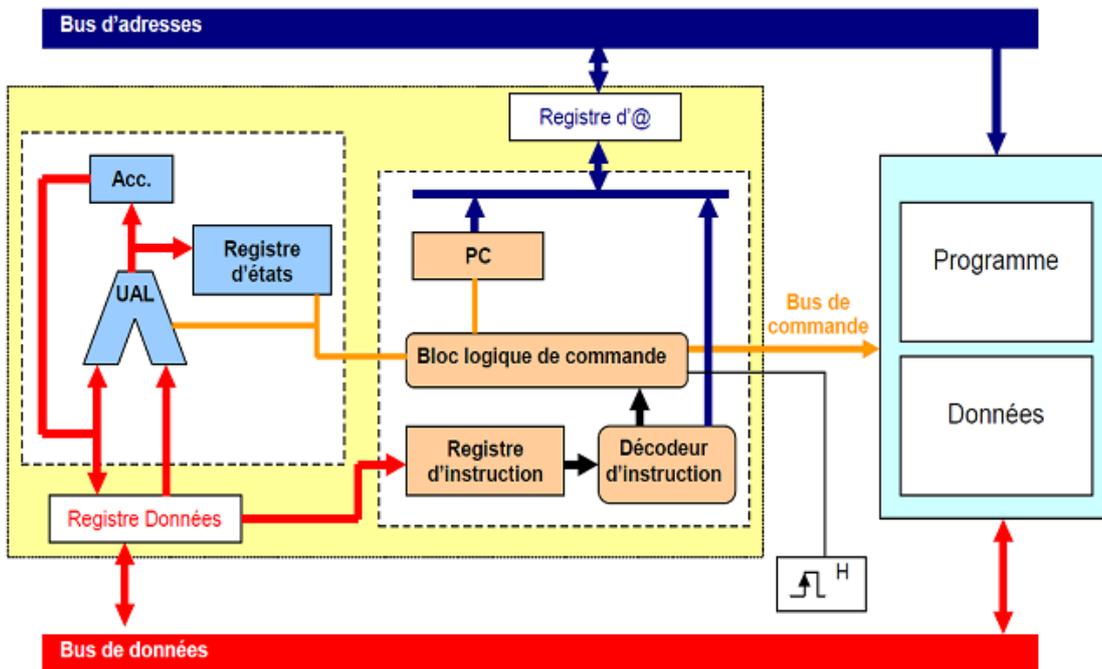
Exemple :

Pour un microprocesseur ayant un format d'instructions à deux adresses, l'instruction d'addition de deux variables en mémoire A et B est représenté dans le registre d'instructions comme suit :

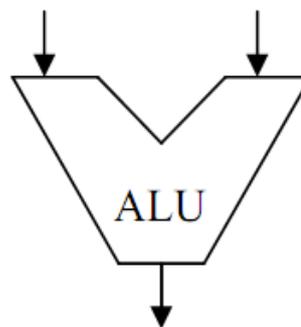


1.2.2 Unité de traitement

C'est le cœur du microprocesseur. Elle regroupe les circuits qui assurent les traitements nécessaires à l'exécution des instructions.



- **L'unité arithmétique et logique (UAL)** : Elle assure les fonctions logiques (ET, OU, Comparaison, Décalage, etc...) ou arithmétique (Addition, soustraction...). Toute instruction qui modifie une donnée fait toujours appel à l'ALU.



-
Symbole de l'Unité arithmétique et logique

- **L'accumulateur** : C'est un registre nommé A il représente l'une des deux entrées de l'UAL. Il est impliqué dans presque toutes les opérations réalisées par l'UAL. Certains constructeurs ont des microprocesseurs à deux accumulateurs (Motorola : 6800).

Exemple: A étant l'accumulateur et B un registre, on peut avoir : $A+B$ (ADD A,B : addition du contenu du registre A avec celui du registre B, le résultat étant mis dans A)

- **Le registre d'état** : à chaque opération, le microprocesseur positionne un certain nombre de bascules d'état. Ces bascules sont appelées aussi indicateurs d'état ou drapeaux (status, flags). Par exemple, si une soustraction donne un résultat nul, l'indicateur de zéro (Z) sera mis à 1. Ces bascules sont regroupées dans le registre d'état. On peut citer comme indicateurs:

- retenue (carry : C)
- retenue intermédiaire (Auxiliary-Carry : AC)
- signe (Sign : S)
- débordement (Overflow : O)
- zéro (Z)
- parité (Parity : P)

➤ **Le Retenue : (carry : C)**

Flag indiquant si une retenue a eu lieu d'une addition ou d'une soustraction. Si une retenue a été générée, ce bit passe à 1.

Exemple :

$$\begin{array}{r}
 \mathbf{11111100} \\
 + \mathbf{10000010} \\
 \text{carry : } \mathbf{1} = \mathbf{01111110}
 \end{array}
 \qquad
 \begin{array}{r}
 \mathbf{FCH} \\
 + \mathbf{82H} \\
 \text{carry : } \mathbf{1} = \mathbf{7EH}
 \end{array}$$

De la même manière lors d'une opération de décalage ou de rotation ce bit peut être positionné en cas de débordement. Par exemple, soit le décalage à gauche sur d'un bit de cet octet : 10010110, après le décalage nous avons 00101100. En ce qui concerne le carry, il va être positionné à 1, puisque le bit expulsé est égal à 1.

➤ **Retenue intermédiaire : (Auxiliary Carry : AC)**

Flag fonctionnant comme le bit de Carry, sauf qu'ici la surveillance de la retenue s'exerce non pas sur l'octet entier, mais sur le premier demi-octet. Ce flag se positionne à 1 si une retenue est générée du bit 3 (bit de poids fort du quartet inférieur) vers le bit 0 du quartet supérieur. Il est utile pour corriger le résultat d'opérations effectuées en code BCD.

➤ **Signe: (S)**

Ce bit est mise à **1** lorsque le résultat de l'opération est négatif (MSB: bit de plus fort poids du résultat: à 1).

➤ **Débordement : (overflow : O)**

Cet indicateur est mis **1**, lorsqu'il y a un dépassement de capacité pour les opérations arithmétiques en complément à 2. Sur 8 bits, on peut coder de -128 (1000 0000) à +127 (0111

1111).

$$\begin{array}{r}
 \mathbf{104} \quad \mathbf{0110\ 1000} \quad \mathbf{-\ 18} \quad \mathbf{1110\ 1110} \\
 + \mathbf{26} \quad + \mathbf{0001\ 1010} \quad \mathbf{-\ 118} \quad \mathbf{1000\ 1010} \\
 = \mathbf{130} \quad = \mathbf{1000\ 0010\ (-126)} \quad \mathbf{-136} \quad \mathbf{0111\ 1000\ (120)\ \text{avec } C=1}
 \end{array}$$

➤ **Le bit Zéro : (Zéro : Z)**

Ce bit est mis à **1** lorsque le résultat de l'opération est nul.

➤ **Le bit de parité : (P)**

Ce bit est mis à **1** lorsque le nombre de 1 de l'accumulateur est pair. Remarque : La plupart des instructions modifient le registre d'état.

1.3 Les registres

Il y'a deux type de registres : les registres d'usage général, et les registres d'adresses (pointeurs).

1.3.1 Les registres d'usage général

Ce sont des mémoires rapides, à l'intérieur du microprocesseur, qui permettent à l'UAL de manipuler des données à vitesse élevée. Ils sont connectés au bus de données interne au microprocesseur. L'adresse d'un registre est associée à son nom (on donne généralement comme nom une lettre) A, B,C...

Exemple : MOV C,B: transfert du contenu du registre B dans le registre C.

1.3.2 Les registres d'adresses (pointeurs)

Ce sont des registres connectés sur le bus d'adresses. On peut citer comme registre:

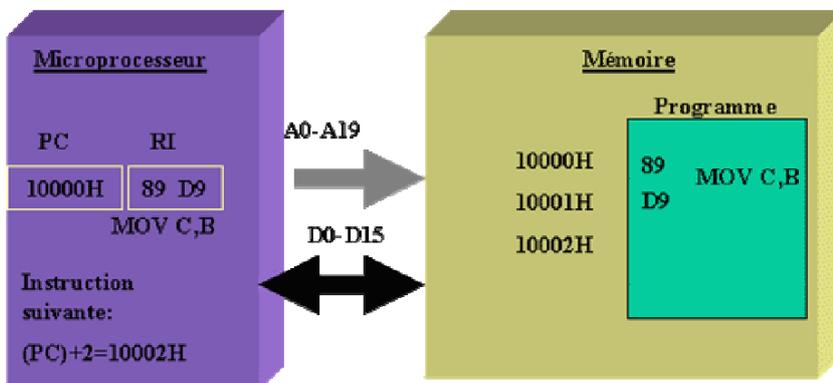
- Le compteur ordinal (pointeur de programme PC : Programme Counter) ;
- Le pointeur de pile (stack pointer SP) ;
- Les registres d'index (index source SI et index destination DI).

1.3.2.1 Le compteur ordinal

Il contient l'adresse de l'instruction à rechercher en mémoire. L'unité de commande incrémente le compteur ordinal (PC) du nombre d'octets sur lequel l'instruction, en cours d'exécution, est codée. Le compteur ordinal contiendra alors l'adresse de l'instruction suivante.

Prenons l'exemple d'un microprocesseur 8086 de INTEL :

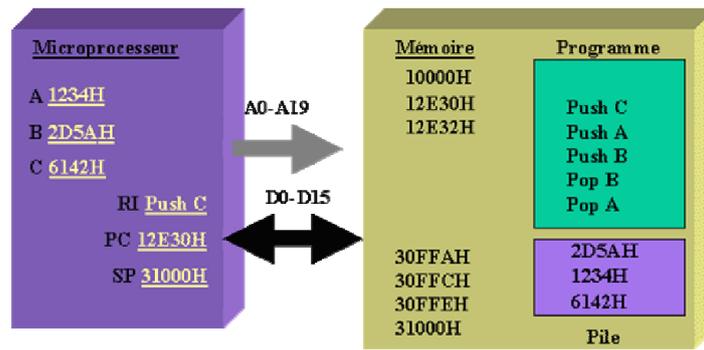
Exemple : (PC)=10000H ; il pointe la mémoire qui contient l'instruction MOV CX,BX qui est codée sur deux octets (89 D9H) ; l'unité de commande incrémentera de deux le contenu du PC : (PC) = 10002H (la mémoire sera supposée être organisée en octets).



1.3.2.2 Le pointeur de pile (stack pointer SP)

Il contient l'adresse de la pile. Celle-ci est une partie de la mémoire, elle permet de stocker des informations (le contenu des registres) relatives au traitement des interruptions et des sous-programmes. La pile est gérée en LIFO : (Last IN First Out) dernier entré premier sorti. Le fonctionnement est identique à une pile d'assiettes. Le pointeur de pile SP pointe le haut de la pile (31000H voir figure en dessous), il est décrémenté avant chaque empilement, et incrémenté après chaque dépilement. Il existe deux instructions pour empiler et dépiler: PUSH et POP. exemple: PUSH A empilera le registre A et

POP A le dépilera. La figure suivante montre l'évolution du pointeur de pile durant l'exécution du programme commençant en 12E30H.



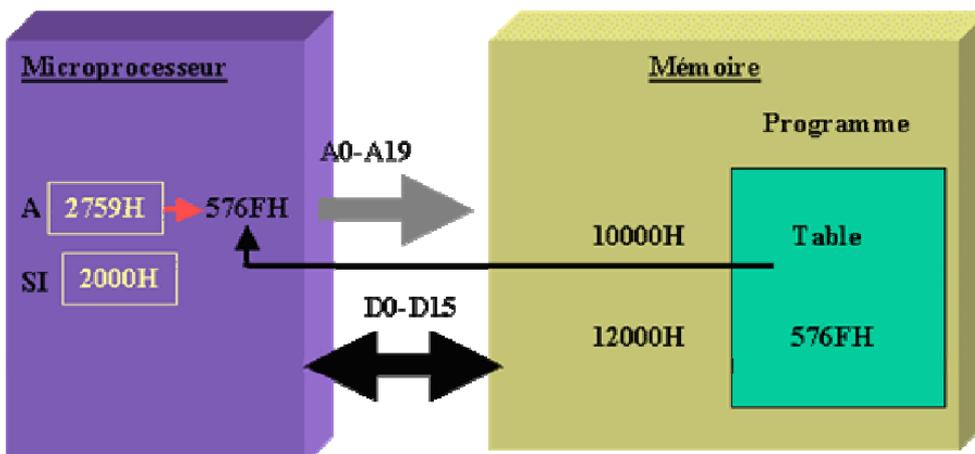
Sur cette figure le programme commence par sauvegarder le contenu de C dans la pile (PUSH C). Pour cela (SP) est décrémenté de deux ((SP)=31000H-2=30FFEH), puis on effectue l'écriture de (C) dans la mémoire à l'adresse (SP) : (30FFEH) = 6142H. Pour PUSH A on obtient : (30FFCH)=1234H, et pour PUSH B : (30FFAH)=2D5AH. Pour l'instruction POP B, (SP) est chargé dans le registre B (SP=30FFAH ; B=2D5AH) puis (SP) est incrémenté de deux (SP= 30FFAH+2=30FFCH). Enfin, pour POP A on obtient : A=1234H et (SP=30FFCH + 2 = 30FFEH).

1.3.2.3 Les registres d'index (index source SI et index destination DI)

Les registres d'index permettent de mémoriser une adresse particulière (par exemple : début d'un tableau). Ces registres sont aussi utilisés pour adresser la mémoire de manière différente. C'est le mode d'adressage indexé.

Exemple :

MOV A,[SI+10000H] place le contenu de la mémoire d'adresse 10000H + le contenu de SI, dans le registre A.



1.4 Principe d'exécution d'une instruction

Dans cette architecture standard, l'exécution d'une instruction se fait en trois étapes:

- Recherche de l'instruction (Fetch) ;

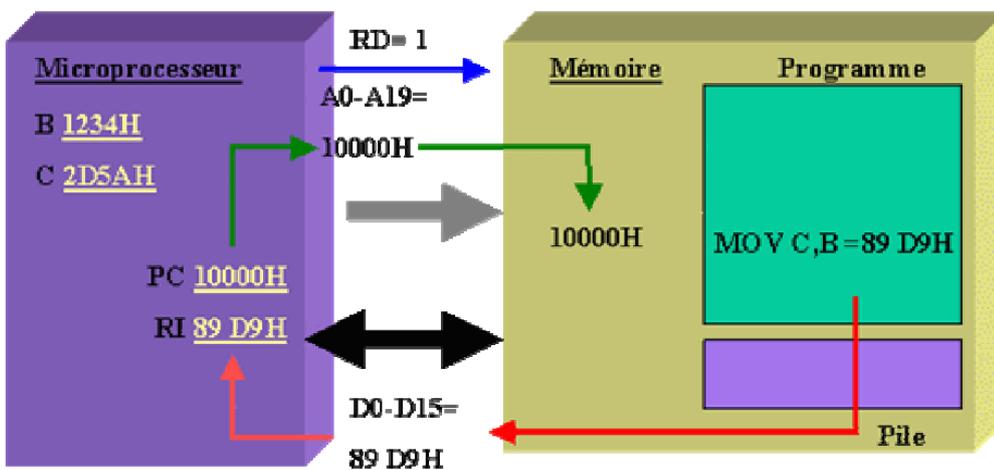
-Décodage (décode) ;

-Exécution (exécute).

1.4.1 Recherche de l'instruction

Le contenu de PC (compteur ordinal) est placé sur le bus d'adresse (c'est l'unité de commande qui établit la connexion). L'unité de commande (UC) émet un ordre de lecture (READ=RD=1). Au bout d'un certain temps (temps d'accès à la mémoire), le contenu de la case mémoire sélectionnée est disponible sur le bus des données. L'unité de commande charge la donnée dans le registre d'instruction pour décodage.

Le microprocesseur place le contenu de PC (10000H) sur le bus d'adresse et met RD à 1 (cycle de lecture). La mémoire met sur le bus de données le contenu de sa mémoire n° 10000H (ici 89D9H qui est le code de MOV C,B). Le microprocesseur place dans son registre d'instruction le contenu du bus de données (89D9H). L'unité de commande décode et exécute l'instruction MOV C,B.

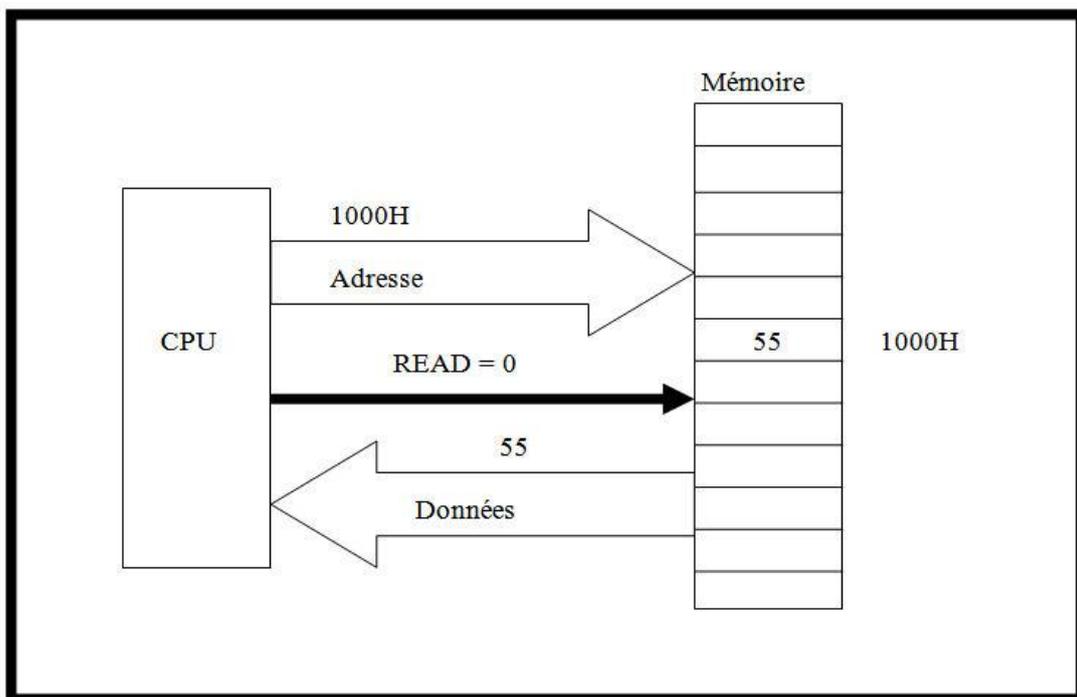
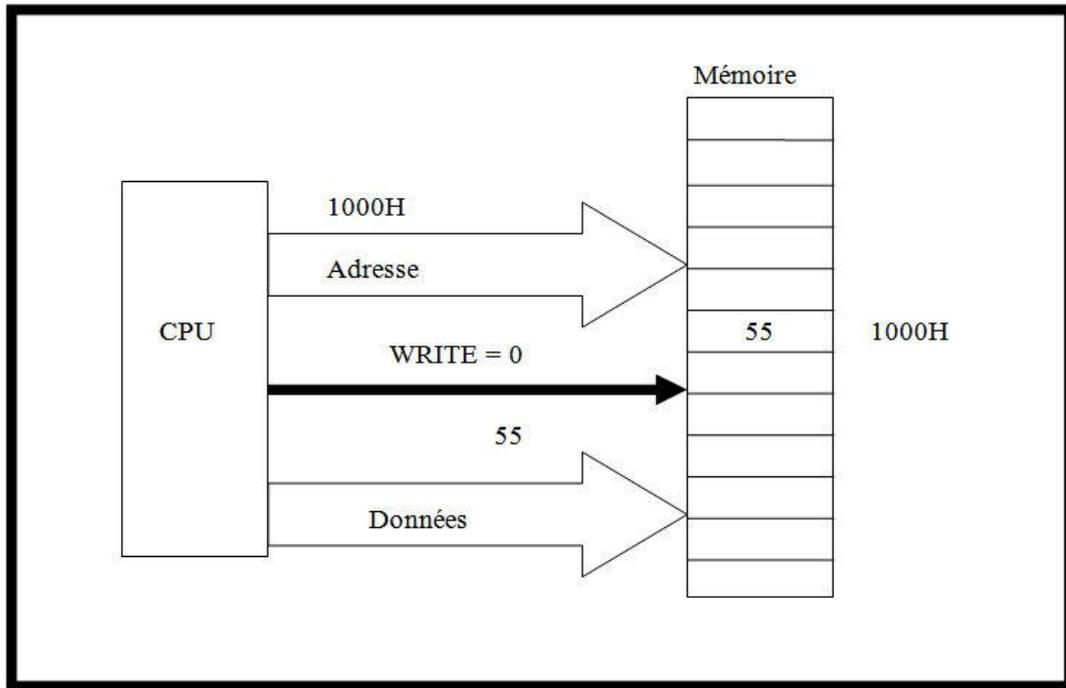


1.5 Les BUS

On appelle Bus, en informatique, un ensemble de liaisons physiques (câbles, pistes de circuits imprimés, ...) pouvant être exploitées en commun par plusieurs éléments matériels afin de communiquer. Dans le cas où la ligne sert uniquement à la communication de deux composants matériels, on parle parfois de port (port série, port parallèle, ...). Un Bus est caractérisé par le volume d'informations transmises simultanément (exprimé en bits), correspondant au nombre de lignes sur lesquelles les données sont envoyées de manière simultanée. Une nappe de 32 fils permet ainsi de transmettre 32 bits en parallèle. On parle ainsi de "largeur de bus" pour désigner le nombre de bits qu'il peut transmettre simultanément. D'autre part, la vitesse du bus est également définie par sa fréquence (exprimée en Hertz), c'est-à-dire le nombre de paquets de données envoyés ou reçus par seconde. On parle de cycle pour désigner chaque envoi ou réception de données. De cette façon, il est possible de connaître la bande passante d'un bus, c'est-à-dire le débit de données qu'il peut transporter, en multipliant sa largeur par sa fréquence. Un Bus d'une largeur de 16 bits, cadencé à une fréquence de 133 Mhz possède donc une bande passante égale à :

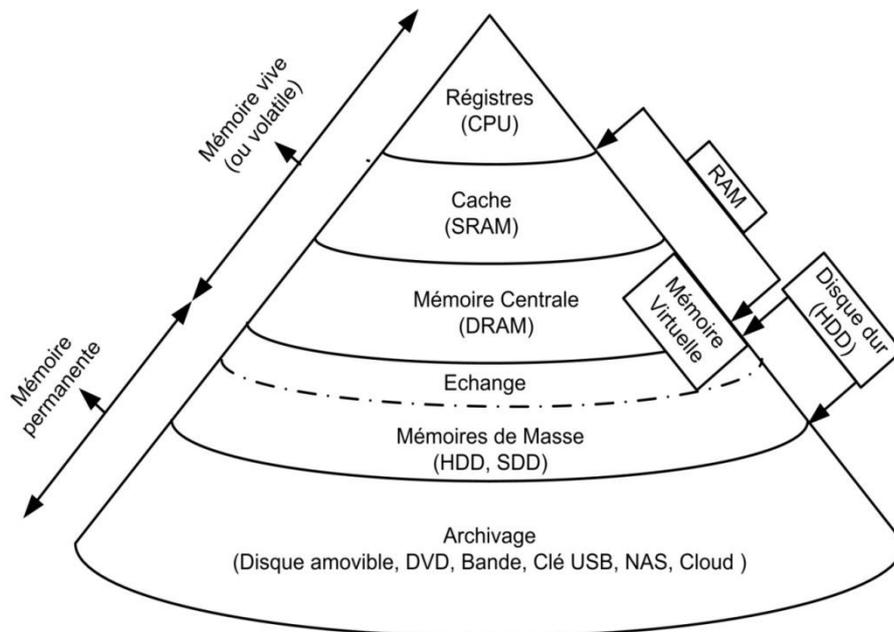
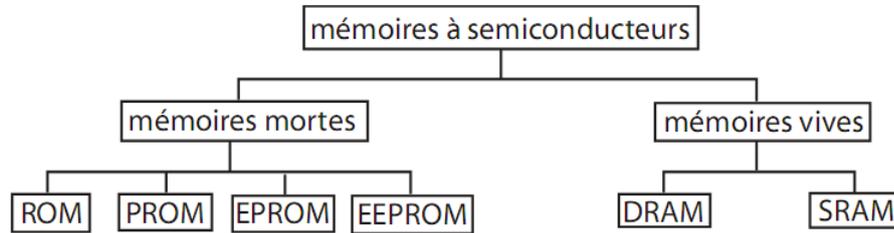
$16 * 133.10^6 = 2128 * 10^6 \text{ bit/s,}$
 soit $2128 * 10^6 / 8 = 266 * 10^6 \text{ octets/s}$
 soit $266 * 10^6 / 1024 = 259.7 * 10^3 \text{ Ko/s}$
 soit $259.7 * 10^3 / 1024 = 253.7 \text{ Mo/s}$

Pour la communication, un microprocesseur a besoin en général de trois Bus. Un Bus de données, un Bus d'adresse et un Bus de commande. Sur la figure suivante nous présentons un système à base de microprocesseur classique avec ces trois Bus.



Les mémoires sont connectées à un bus d'adresse de n bits, un bus de données de p ($p=1$ c'est-à-dire 8bits) bits et des lignes de commandes. Pour pouvoir communiquer avec le microprocesseur, on va relier leurs bus ensemble. Pour cela, il est nécessaire d'avoir adéquation entre le nombre de bits des bus de données et d'adresse de la mémoire et du microprocesseur.

ROM : Read Only Memory. Mémoire à lecture seule, sans écriture.



La mémoire vive, généralement appelée RAM (Random Access Memory, mémoire à accès aléatoire), est la mémoire principale du système, c'est-à-dire qu'il s'agit d'un espace permettant de stocker de manière temporaire des données lors de l'exécution d'un programme. En effet le stockage de données dans la mémoire vive est temporaire, contrairement au stockage de données sur une mémoire de masse telle que le disque dur, car elle permet uniquement de stocker des données tant qu'elle est alimentée électriquement. Ainsi, à chaque fois que l'ordinateur est éteint, toutes les données présentes en mémoire sont irrémédiablement effacées.

Types de RAM

Les mémoires statiques SRAM (Static RAM) : rapides mais coûteuses, elles sont utilisées comme mémoires caches pour les processeurs.

Définition : La mémoire cache a pour fonction principale d'optimiser les accès aux différentes instructions et données que le microprocesseur a besoin lors de l'exécution des

programmes. Elle permet de mettre dans la mémoire cache qui est beaucoup plus rapide que la mémoire centrale, les informations les plus utilisées et que le microprocesseur a besoin fréquemment.

Les performances du microprocesseur sont très liés à la validité des prédictions faites par le contrôleur de mémoire cache. Il existe deux niveaux de mémoire cache. La cache de niveau 1 est toujours intégrée au microprocesseur, sa taille est de quelques kilooctets et peut atteindre 256 KO dans certains microprocesseurs. La cache de second niveau est généralement sur la carte mère, sa taille varie de 256 KO jusqu'à 1MO. L'accès aux informations sera donc plus rapide et l'exécution des programmes est plus optimale.

Les mémoires dynamiques DRAM : peu coûteuses avec une grande capacité de stockage par rapport aux SRAM mais moins rapides, elles sont utilisées principalement pour la mémoire centrale des ordinateurs. Plusieurs types de mémoires vives dynamiques existes : SDRAM qui fonctionnent à 100MHz, les RDRAM, les DDR-SDRAM, DDR2 ou DDR3.

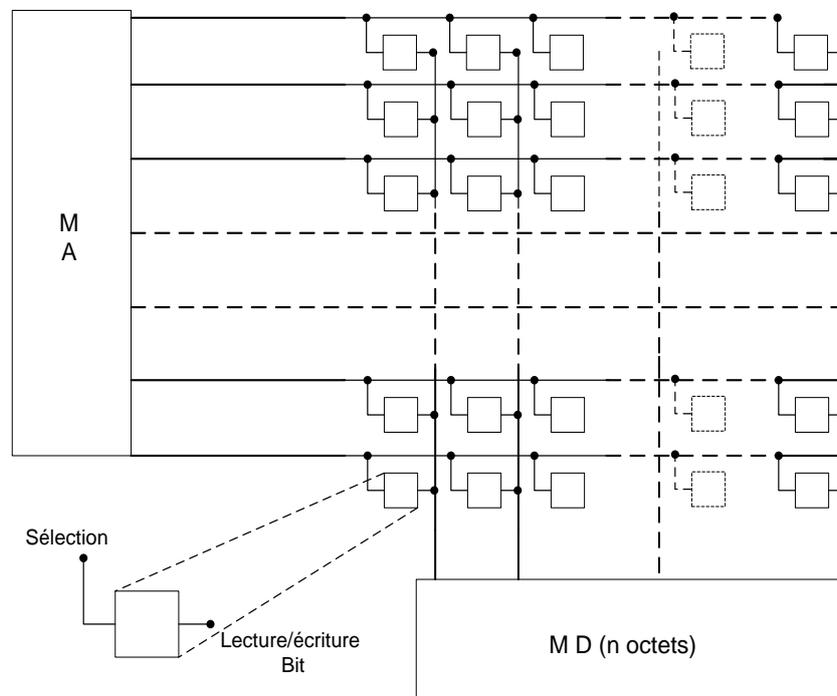
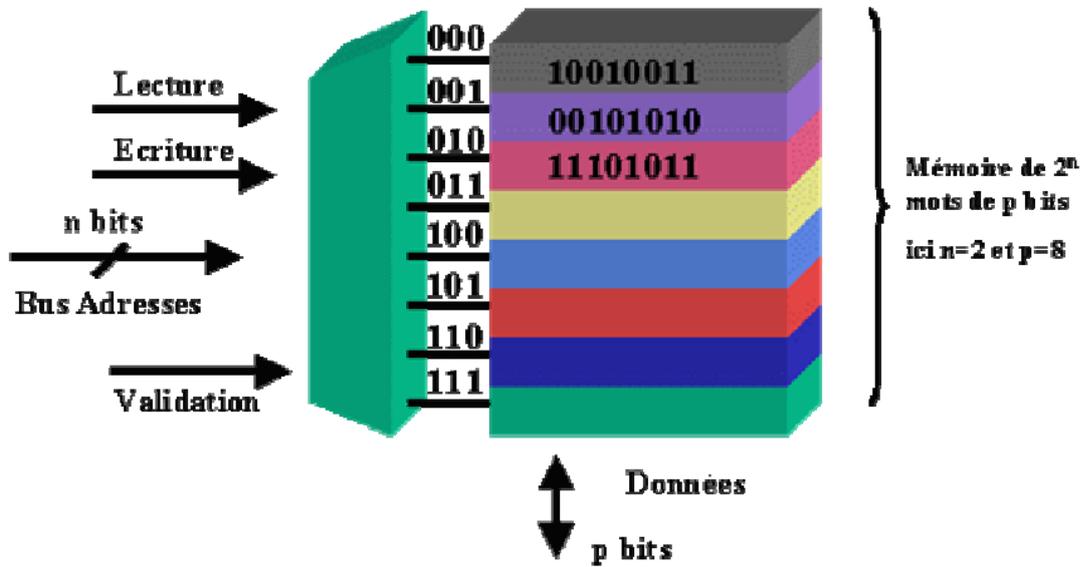


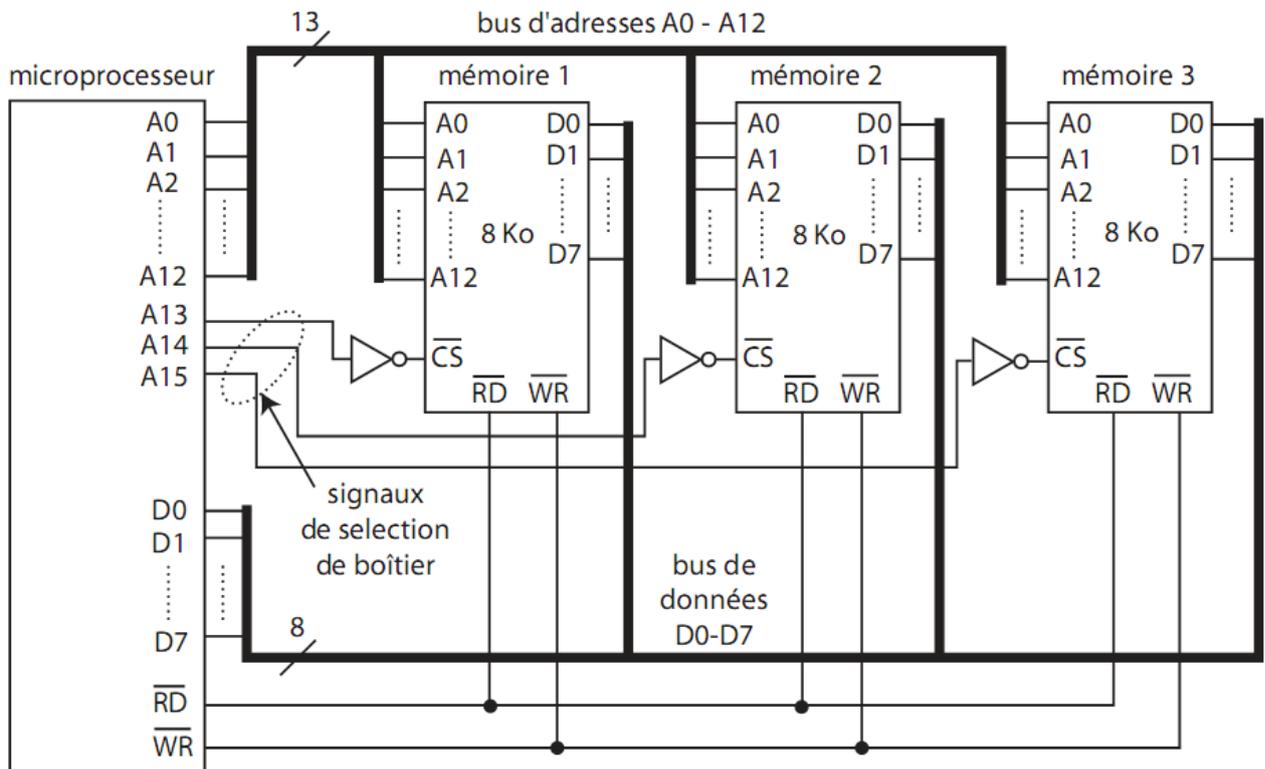
Schéma interne d'une mémoire vive

Sélection d'une mémoire vive

Les boîtiers de mémoires dans les RAM sont connectés chacun à une broche de validation. Cette broche permet de sélectionner un boîtier mémoire parmi plusieurs, d'où son appellation : "chip select". Cette broche permet d'éviter les conflits sur le bus de données. En effet dans le cas général, il existe plusieurs boîtiers mémoire sur la carte, tous branchés sur le même bus de données. Dans ce cas, il est nécessaire de construire un signal qui permettra à un seul boîtier d'accéder au bus de données.



Organisation externe de la mémoire



Connexion de plusieurs boîtiers de mémoires

Interfacage avec le μ P

