Chapitre II : Principaux composants d'un ordinateur

1. Introduction

Un ordinateur est une machine électronique dédiée au traitement de l'information. Pour cela, il doit posséder un nombre d'unités capables de collecter l'information, de la sauvegarder, de la traiter, puis enfin de la diffuser. En effet, tout ordinateur doit posséder :

- Un processeur (ou unité centrale de traitement ou CPU)
- Une mémoire centrale
- Des unités d'entrées/sorties (ou unités d'échanges)
- Des périphériques d'entrées/sorties

2. Principes de fonctionnement

Les deux principaux constituants d'un ordinateur sont la mémoire centrale qui stocke les informations (programmes+données) et le processeur qui exécute les instructions composants un programme.

Un programme est une suite d'instructions qui vont être exécutées par le processeur.

2.1 Le processeur

Le processeur est un circuit électronique qui exécute chaque instruction en quelques cycles d'horloges exprimés en MHZ ou GHZ (Ex : le pentium 133 possédait une fréquence d'horloge de 133 MHZ (133.10^6 opérations/seconde)). C'est le cerveau de l'ordinateur. Schématiquement, le processeur effectue les opérations suivantes :

-Lire en MC l'instruction à exécuter

-Effectuer le traitement

-Passer à l'instruction suivante.

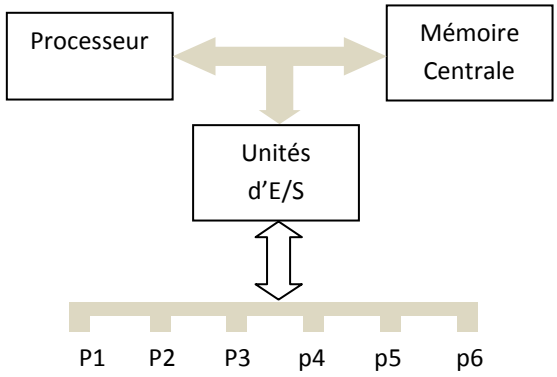


Fig 1. Structure générale d'un ordinateur

Le processeur est composé de :

L'unité de commande et de contrôle : est responsable de la lecture en mémoire et du décodage des instructions. Dirige le fonctionnement de l'UAL, de la mémoire et des E/S ; Elle va chercher, une par une, des instructions en mémoire (et les données qu'elles utilisent), décode chaque instruction, et envoie un signal à l'UAL pour déclencher l'exécution de l'instruction. L'UCC dispose des principaux éléments suivant :

- Le Compteur Ordinal qui est un registre
- Le Registre d'Instruction qui est aussi un registre
- Le décodeur qui est un circuit combinatoire qui détermine quelle opération doit être effectué.
- Le séquenceur qui génère les signaux de commandes, et qui est un circuit séquentiel (câblé) généralement ou un microprogramme stocké dans une mémoire morte.
- L'horloge : qui est un système logique qui émet régulièrement des impulsions calibrées (ou signaux périodiques). L'intervalle de temps entre deux impulsions est appelé temps de cycle. Un cycle machine ou cycle de base est un cycle des signaux périodiques générés par l'horloge. De ce fait, l'horloge synchronise toutes les actions du processeur.

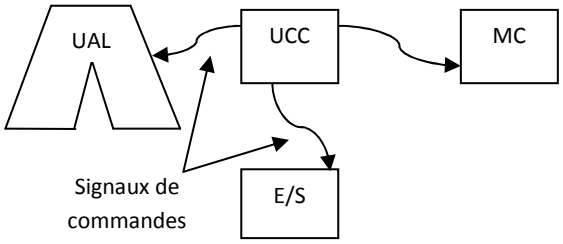


Fig 2. Signaux de commandes émis par l'UCC

L'unité arithmétique et logique : exécute les instructions arithmétique (+,-,*,/) et logiques (AND , OR, NOT, etc.). Pour cela, elle est dotée de circuits logiques capables de réaliser des fonctions logiques et des opérations arithmétiques.

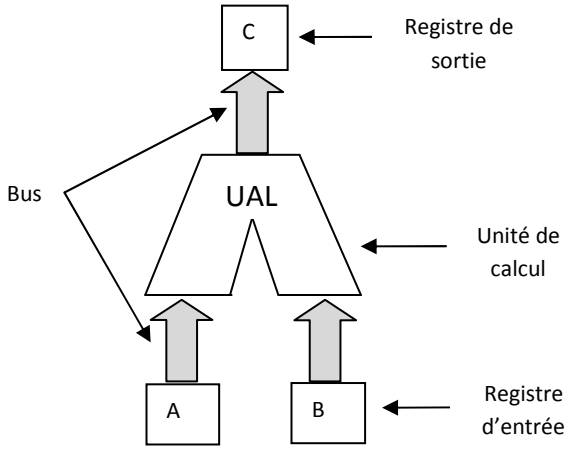


Fig 3. Schéma type d'une UAL (A et B : Opérandes, C : Résultat)

Les circuits qui réalisent les opérations logiques et arithmétiques constituent l'unité de calcul de l'UAL, des bus permettent de véhiculer les données et le résultat entre les registres et l'unité de calcul. **Les registres de l'UAL sont accessibles au programmeur, contrairement aux registres de l'UCC.**

On appelle l'ensemble des instructions (opérations) que peut exécuté l'UAL un **jeu d'instruction**.

2.2 Les bus:

Est un ensemble de fils conducteurs utilisés pour transporter des signaux binaires. Ils servent à lier et faire communiquer les différentes unités.

Il existe trois types de bus :

Le bus d'adresses : est un bus unidirectionnel, seul le processeur envoie des adresses vers la mémoire centrale. Il permet de transporter l'adresse de l'instruction à exécuter ainsi que les adresses des opérandes (données et résultats). Si le bus est composé de N fils, la mémoire possède 2^N adresses comprises entre 0 et 2^N-1.

Le bus de données : c'est un bus bi-directionnel qui transporte les données et les instructions entre les différentes unités.

On distingue le bus de données interne qui transporte les données et les instructions entre l'UAL et les registres du processeur, et le bus de données externe qui transporte les données et les instructions entre la MC et le processeur.

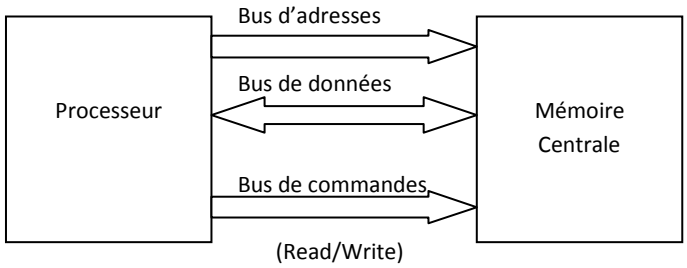


Fig 4. Bus reliant Processeur et MC

Le bus de commandes ou de contrôle : permet de transférer les signaux de commandes émises de l'UCC vers les autres unités. Ces commandes sont généralement :

- Des commandes de lecture en MC
- Des commandes d'écriture en MC
- Des commandes d'exécution à l'UAL
- Des commandes d'E/S aux unités d'E/S

Les signaux de commandes permettent au processeur de communiquer avec les autres circuits, en particulier les signaux Read/Write.

2.3 Registres du processeur:

Ce sont des petites mémoires très rapides d'accès internes au processeur utilisés pour stocker une donnée, une adresse, un résultat intermédiaire ou une instruction. Le nombre de registres diffère d'un ordinateur à un autre, cela dépend de son architecture, mais certains registres fondamentaux existent dans la majorité des machines.

Un registre peut être un registre mot ou un registre d'adresse;

Registre mot: contient le contenu d'un mot mémoire, et a donc la taille d'un mot (longueur classique actuelle: 32 ou 64 bits).

Registre adresse: contient l'adresse d'un mot dont la longueur = $\log_2$ (Nombre_de_mots_en_MC).

Un registre peut être aussi un registre général ou un registre spécialisé ;

Registres généraux: Registres d'usage général interchangeables (parfois notés R0, R1, etc.), peuvent stocker indifféremment adresses, entiers, flottants, etc. et permettent de limiter l'accès à la MC en stockant les informations utilisées fréquemment par le processeur tel que les résultats intermédiaires.

Registres spécialisés : Ces registres sont spécialisés et ne peuvent contenir qu'un type bien précis de données. Ils permettent d'effectuer les opérations de décalages et les opérations arithmétiques à virgule flottante, etc… On peut citer comme exemple de ces registres : CO, PSW, Pointeur de pile.

On peut distinguer les registres suivants :

- **Accumulateur**: est un registre de l'UAL qui sert essentiellement à contenir le résultat de l'opération effectuée, comme il peut contenir aussi l'un des deux opérandes avant l'exécution de l'opération et recevoir le résultat après.
  En général, la taille de l'accumulateur est équivalente à la taille du *mot mémoire*.
- **RD (MDR: Memory Data Register)** : Registre tampon de l'UAL : Stocke l'une des deux opérandes d'une instruction arithmétique.
- **Registre d'état** (PSW : Program Status Word): stocke des bits indicateurs appelés drapeaux (flags) qui indiquent des états particuliers (ex : retenue, signe du résultat, dépassement de capacité, etc…)
- **RI** : Registre d'instruction, contient le code de l'instruction en cours d'exécution (Via le bus de données).
- **CO** : Compteur ordinal (IP : Instruction Pointer, PC : Program Counter), contient l'adresse de la prochaine instruction à exécuter.
- **RA (MAR: Memory Address Register)** : Registre d'adresse, utilisé pour accéder en lecture ou écriture à une donnée/instruction en mémoire.

Fonctionnement ou comment s'exécute une instruction:

Le fonctionnement peut être décrit de la façon suivante :

L'UCC va chercher en MC une instruction en envoyant une adresse par le bus d'adresse, et une commande de lecture. L'instruction enregistrée à l'adresse donnée est transférée par bus de données vers le RI, où son décodage permet de déterminer le type d'opération à effectuer, de même que les opérandes (données) nécessaires à

Registres spécialisés : Ces registres sont spécialisés et ne peuvent co

l'opération.

Les opérandes sont ensuite transférés vers l'UAL via le bus de données de la même manière que l'instruction. L'UAL exécute l'opération et met le résultat dans l'accumulateur, ensuite ce résultat sera transféré de l'accumulateur à la MC.

3. Mémoire

Une mémoire est un système capable d'acquérir, de conserver et de restituer des informations binaires dans un ordinateur, et tout cela par des mécanismes adaptés.

La mémoire est physiquement divisée en cellules de tailles fixes et est utilisée pour stocker les instructions et les données (mots mémoires). La longueur classique actuelle d'un mot mémoire= 32 ou 64 bits. Chaque cellule contient le même nombre de bits et est identifiée ou repérée par un numéro (adresse). L'adresse est souvent donnée en hexadécimal.

Avec une adresse de N bits il est possible de référencer au plus 2^N cases mémoire. La capacité (taille) de la mémoire est la quantité d'information qu'elle peut stocker, exprimée en bits ou en mots de 2^N bits :

1 mot de 8 bits= 1 octet. 1 kilo-octet=2^{10} octets. 1 méga-octet=2^{10} KO. 1 giga-octet=2^{10} MO. 1 Tera=2^{10} GO.

Le temps d'accès est le temps qui s'écoule entre le lancement d'une opération d'accès (lecture ou écriture) et son accomplissement. Cette vitesse peut également s'exprimer comme une fréquence d'horloge caractéristique de la mémoire, égale à l'inverse du temps d'accès et mesurée en hertz (Hz). Ainsi un temps d'accès égal à 10 nanosecondes correspond à une fréquence de 100 Mhz (1 Mhz = 10^6 Hz). Pour une mémoire électronique qui est une

mémoire très rapide (RAM, ROM, registre…) ce temps se mesure en nanosecondes (milliardième de seconde: 10^{-9} s). Pour des mémoires magnétiques ou optiques (mémoire de masse) ce temps se mesure en millisecondes (millième de seconde : 10^{-3} s).

3.1 Types de mémoires

3.1.1 Les mémoires vives (RAM: Random Acces Memory ou Mémoire à accès aléatoire)

Une mémoire vive sert au stockage temporaire de données. Elle doit avoir un temps de cycle très court pour ne pas ralentir le microprocesseur. Les mémoires vives sont en général volatiles: elles perdent leurs informations en cas de coupure d'alimentation. Ce sont des mémoires à lecture ou écriture où le temps d'accès est indépendant de la place de l'information dans la mémoire. Il existe deux grandes familles de mémoires RAM:

- Les RAM statiques ou SRAM: Le bit mémoire est composé d'une bascule. Chaque bascule contient entre 4 et 6 transistors.
- Les RAM dynamiques ou DRAM: l'information est mémorisée sous la forme d'une charge électrique stockée dans un condensateur qui est relié à un transistor. Un point mémoire nécessite environ quatre fois moins de transistors que dans une mémoire statique, sa consommation s'en retrouve donc aussi très réduite. Mais l'information est perdue si on ne la régénère pas périodiquement (charge du condensateur). Les DRAM doivent donc être rafraîchies régulièrement pour entretenir la mémorisation. D'autre part, la lecture de l'information est destructive. En effet, elle se fait par décharge de la capacité du point mémoire lorsque celle-ci est chargée. Donc toute lecture doit être suivie d'une réécriture. Les RAM utilisées dans le passé et actuellement, DDR1 SDRAM-DDR4 SDRAM (Double Data Rate Synchronous) sont des mémoires DRAM.

En général les DRAM, qui offrent une plus grande densité d'information et un coût par bit plus faible, sont utilisées pour la MC, alors que les SRAM, plus rapides, sont utilisées lorsque le facteur vitesse est critique, notamment pour des mémoires de petite taille comme les caches et les registres.

3.1.2 Les mémoires mortes (ROM: Read Only Memory ou Mémoire à lecture seule)

On utilise ces mémoires quand il est nécessaire de pouvoir conserver des informations de façon permanente même lorsque l'alimentation électrique est interrompue. Ces mémoires sont non volatiles et contrairement aux RAM, ne peuvent être que lue. L'inscription en mémoire des données reste possible mais est appelée programmation.

Elle permet de stocker typiquement le programme de *bootstrap* dont l'objet est de charger au démarrage de l'ordinateur, à partir du disque magnétique, le noyau du système d'exploitation qui nous permet d'accéder aux ressources matérielles et logicielles de l'ordinateur.

Il existe donc plusieurs types de ROM :

- ROM : Elle est programmée par le fabricant et son contenu ne peut plus être ni modifié ni effacé par l'utilisateur.
- La PROM (Programmable ROM): Elle peut être programmée une seule fois par l'utilisateur. La programmation est réalisée à l'aide d'une machine spéciale.
- L'EPROM ou UV-EPROM (Erasable PROM) : est une PROM qui peut être effacée et reprogrammée. L'exposition d'une vingtaine de minutes à un rayonnement ultra-violet permet de forcer tous les bits à une même valeur.
- L'EEPROM (Electically EPROM) est une mémoire programmable et effaçable électriquement sans intervention d'un rayonnement ultraviolet. Elle répond ainsi à l'inconvénient principal de l'EPROM et peut être programmée sur place.
- La FLASH: La mémoire Flash s'apparente à la technologie de l'EEPROM. Elle est programmable et effaçable électriquement comme

les EEPROM. Le temps d'écriture est similaire à celui d'un DD. Son cycle de vie est limité (100 000 écriture). La mémoire flash a connu un essor très important ces dernières années avec le boom de la téléphonie portable et des appareils multimédia (PDA, appareil photo numérique, lecteur MP3, etc...).

Trois critères principaux définissent l'importance d'une mémoire: **La capacité**, **le temps d'accès** et **le coût**. La mémorisation de l'information dans un ordinateur ne se fait pas en un lieu unique mais est organisée au travers d'une **hiérarchie de mémoires**. Il se trouve que les mémoires de grande capacité sont souvent très lentes et que les mémoires rapides sont très chères. On utilise des mémoires de faible capacité mais très rapides pour stocker les informations dont le microprocesseur se sert le plus et on utilise des mémoires de capacité importante mais beaucoup plus lente pour stocker les informations dont le microprocesseur se sert le moins. Ainsi, plus on s'éloigne du microprocesseur et plus la capacité et le temps d'accès des mémoires vont augmenter.

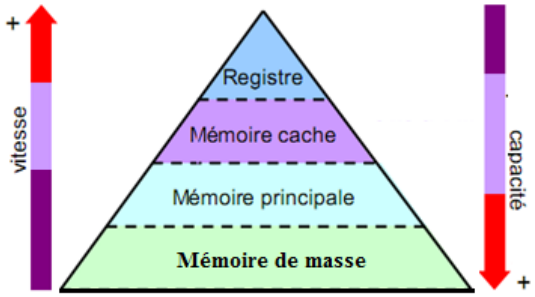


Fig 5. Hiérarchie mémoire

Toutes les mémoires ne jouent pas le même rôle et on peut les classer en deux grandes catégories :

– les mémoires de travail désignent les mémoires électroniques qui sont actives dans l'exécution d'un programme. On y trouve les registres du processeur, la MC, la mémoire cache, la mémoire morte.

– les mémoires de stockage ont pour objet de conserver de manière permanente de grandes quantités d'informations. Les informations qui y sont stockées ne participent pas directement à l'exécution d'un programme mais doivent êtres chargées en MC pour être exploitées par le processeur. Ce sont des mémoires de masses, elles sont de type magnétique ou optique.

3.2 Mémoire centrale ou mémoire principale ou la RAM

C'est la mémoire de travail de l'ordinateur, pour qu'un programme s'exécute il faut qu'il soit chargé dans la MC. Elle contient aussi une partie du système d'exploitation de l'ordinateur.

Capacité= élevé, Temps d'accès= rapide, Coût= Très élevé.

3.3 Mémoire de masse

Ce sont des mémoires auxiliaires permanentes ou des mémoires de stockage ou encore d'archivage. Elles peuvent être :

- Magnétiques : Exemple ; Disque dur et bandes magnétiques.
- Optiques : Exemple : CD-R et CD-ROM.

Capacité= Très élevé, Temps d'accès= lent, Coût= faible.

3.4 Mémoire cache

Appelée aussi antémémoire, c'est une mémoire intermédiaire entre le processeur et la MC.

L'écart de performance entre le microprocesseur et la MC ne cesse de s'accroître. Ainsi, le temps de cycle processeur décroît plus vite que le temps d'accès mémoire entraînant un **goulot d'étranglement –Goulot de Von Neumann-**. La MC n'est plus en mesure de délivrer des informations aussi rapidement que le processeur est capable de les traiter.

On considère en effet que les performances des processeurs doublent tout les 1 an et demi (loi de Moore) alors que celles des mémoires doublent tous les 10 ans.

Depuis le début des années 80, une des solutions utilisées pour masquer cette latence est de disposer une mémoire très rapide entre le microprocesseur et la MC. Elle est appelée mémoire cache. On permet ainsi microprocesseur d'acquérir les données à sa vitesse propre.

Elle fait maintenant partie intégrante du microprocesseur et se décline même sur plusieurs niveaux.

Les mémoires caches permettent donc de « rapprocher » du processeur les informations qu'il doit traiter. En effet c'est souvent le temps de transfert par le bus qui est un facteur de ralentissement.

Capacité= Faible, Temps d'accès= Très rapide, Coût= Très élevé.

Si les mémoires cache permettent d'accroître les performances, c'est en partie grâce à deux principes:

- **Localité spatiale**: le code d'un programme s'exécute toujours à l'intérieur de petites zones répétées de mémoire (des blocs correspondant à des boucles ou/et des sous-programmes). Ainsi quand une instruction s'exécute, il est très probable que la prochaine instruction à exécuter soit dans le mot suivant de la mémoire.

  Aussi, l'accès à une donnée située à une adresse X va probablement être suivi d'un accès à une zone tout proche de X.
- **Localité temporelle**: les blocs s'exécutent en séquences très proches (il y a plus de chances d'accéder à une position de mémoire utilisée il y a 10 cycles qu'à une autre utilisée il y a 10000 cycles). L'accès à une zone mémoire à un instant donné a de fortes chances de se reproduire dans la suite du programme.

Donc, lorsqu'une donnée ou une instruction doit être chargée dans un cache il serait intéressant de charger également les données (et/ou instructions) qui sont proches en MC. On augmente ainsi la probabilité que le processeur trouve la prochaine donnée (ou instruction) dans le cache. Le principe de cache est très simple: le microprocesseur n'a pas conscience de sa présence et lui envoie toutes ses requêtes comme s'il agissait de la MC:

- Soit la donnée ou l'instruction requise est présente dans le cache et elle est alors envoyée directement au microprocesseur. On parle de succès de cache (cache hit).
- Soit la donnée ou l'instruction n'est pas dans le cache, et le contrôleur de cache envoie alors une requête à la MC. Une fois l'information récupérée, il la renvoie au microprocesseur tout en la stockant dans le cache. On parle de défaut de cache ou de d'échec de cache (cache miss).

Bien entendu, le cache mémoire n'apporte un gain de performance que dans le premier cas. Sa performance est donc entièrement liée à son taux de succès (hit rate). Ce taux dépend de la taille de la cache et de l'algorithme exécuté par le contrôleur de cache. Il est courant de rencontrer des taux de succès moyen de 80% à 90%.

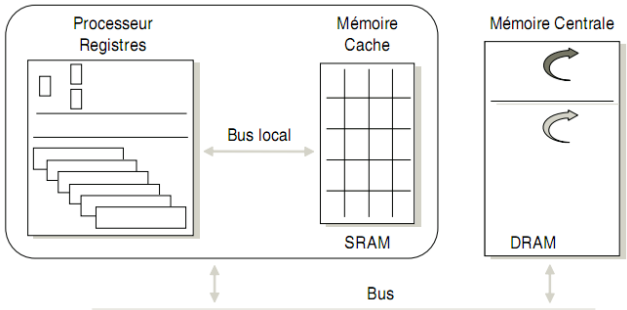


Fig 6. Mémoire cache

Remarques

- Un cache utilisera un répertoire de clés qui est une sorte de carte pour savoir quels sont les mots de la MC dont il possède une copie.
- Il existe dans le système deux copies de la même information: l'originale dans la MC et la copie dans le cache. Si le microprocesseur modifie la donnée présente dans le cache, il faudra prévoir une mise à jour de la MC.
- Une ligne (ou bloc) de cache est la plus petite portion de la cache avec une étiquette unique. C'est le plus petit élément de données qui peut être transféré entre la mémoire cache et la MC. Un mot est le plus petit élément de données qui peut être transféré entre le processeur et la mémoire cache.

On distingue en fonction de leur taille et de leur localité :

- la mémoire cache de premier niveau (L1) qui est formée de deux blocs séparés, l'un servant au

stockage des données, l'autre servant au stockage des instructions. On distingue donc :

- le cache de données L1D
- le cache d'instructions L1I

- la mémoire cache de second niveau (L2) dont la taille est plus grande que le cache L1
- la mémoire cache de troisième niveau (L3) parfois qualifié de LLC dont la taille est plus grande que le cache L2.

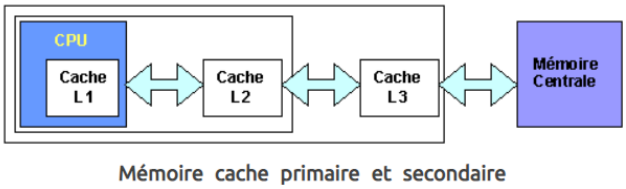


Fig 7: Différents niveaux de mémoire cache

Le cache de niveau 1 est interne (même puce que le processeur) et les caches de niveaux 2 et 3 sont externes.

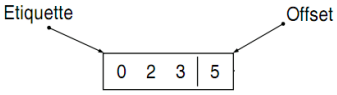
- Si le processeur cherche une donnée, elle va être d'abord recherchée dans le cache de donnée L1 et rapatriée dans un registre adéquat, si la donnée n'est pas présente dans le cache L1, elle sera recherchée dans le cache L2.
- Si la donnée est présente dans L2, elle est alors rapatriée dans un registre adéquat et recopiée dans le bloc de donnée du cache L1. Il en va de même lorsque la donnée n'est pas présente dans le cache L2, elle est alors rapatriée depuis L3.
- Si la donnée ne se trouve pas dans L3, elle est alors rapatriée depuis la mémoire centrale dans le registre adéquat et recopiée dans tous les niveaux de caches.

Le cache de niveau L2 est de capacité plus importante et d'accès moins rapide que le cache L1. Il se retrouve aujourd'hui intégré dans le microprocesseur (sur la carte processeur). Toutefois, s'il est intégré, il n'est pas imbriqué comme le cache L1. Le cache L3 est lui généralement implanté sur la carte mère.

Il existe 3 différents types de cache:

3.2.1 Mémoire cache purement (ou complètement) associative

Dans ce cas, l'antémémoire est une **mémoire associative**. Chaque ligne de la mémoire de niveau supérieur peut être écrite à n'importe quelle adresse de la mémoire cache. Chaque case d'une mémoire associative comprend deux champs correspondant à *la clé* et *à l'information associée à cette clé*. La clé est constituée par l'adresse en mémoire centrale de l'instruction ou la donnée cherchée (son numéro de ligne), et l'information associée est constituée de l'instruction ou la donnée elle-même. Cela donne le format suivant de l'adresse qui est interprétée comme *une étiquette* et *un offset*:



Il faut noter que la recherche par clé dans la mémoire associative ne s'effectue pas de manière séquentielle, mais en parallèle sur toutes les cases de la mémoire associative. Le contrôleur de cache vérifie en une seule opération (c'est pourquoi il y a autant de comparateurs que de lignes) si une étiquette est présente dans une des lignes du répertoire.

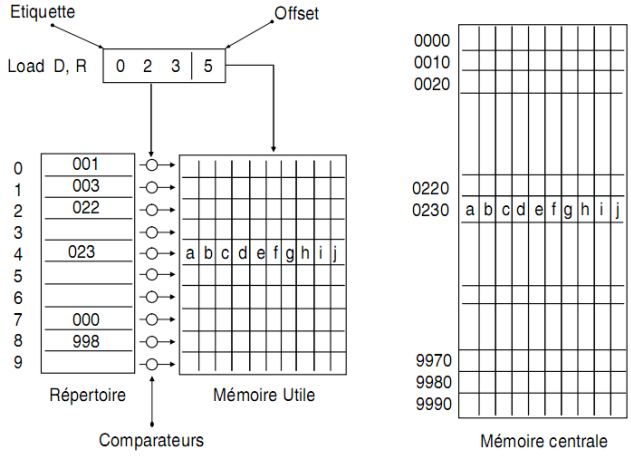


Fig 8: Cache purement associatif

Dans l'exemple ci-dessus, la MC est vue comme une suite de lignes composées chacune de 10 mots (10.000 mots au total). Le cache est organisé autour :

– de la mémoire utile. Elle contient les données. Chaque ligne a une longueur de 10 mots. Il y a 10 lignes;