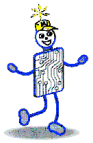
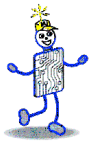
CPU et Mémoire



Design, améliorations, et implémentations

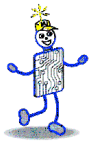
Techniques et caractéristiques modernes qui permettent de donner aux ordinateurs actuels toute leur puissance

Architectures CPU



- Design de l'architecture CPU
  - Architecture traditionnelle
  - VLIW (Transmeta) – Very Long Instruction Word
  - EPIC (Intel) – Explicitly Parallel Instruction Computer
- Architectures CPU
  - IBM System/360
  - Famille Intel x86
  - Famille IBM POWER/PowerPC
  - Famille Sun SPARC

Architectures traditionnelles



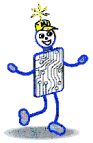
Architecture CISC & RISC

- Architecture traditionnelle des microprocesseurs se composent de deux grandes familles:

- CISC – Complex Instruction Set Computer
- RISC – Reduced Instruction Set Computer

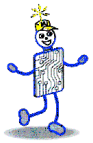
- Chacune de ces deux architectures est consistante avec les caractéristiques d'une architecture selon Von Neumann

Microprocesseur



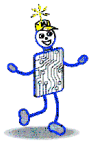
- **L'architecture CISC (famille Intel)**
  - Motivation
    - La mémoire travaillait très lentement => soumettre au microprocesseur des instructions complexes qui demanderaient autant d'accès mémoire que plusieurs petites instructions
    - Le développement des langages de haut niveau posa de nombreux problèmes quand à la conception de compilateurs. On a donc eu tendance à incorporer au niveau processeur des instructions plus proches de la structure de ces langages.

Microprocesseur



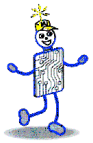
- **L'architecture CISC**
  - Grand nombre d'instructions ou le microprocesseur doit exécuter des taches complexes par instruction unique.
  - Pour une tache donnée, une machine CISC exécute ainsi un petit nombre d'instructions mais chacune nécessite un plus grand nombre de cycles d'horloge.
  - Le code machine de ces instructions varie d'une instruction à l'autre et nécessite donc un décodeur complexe (microcode)

Microprocesseur



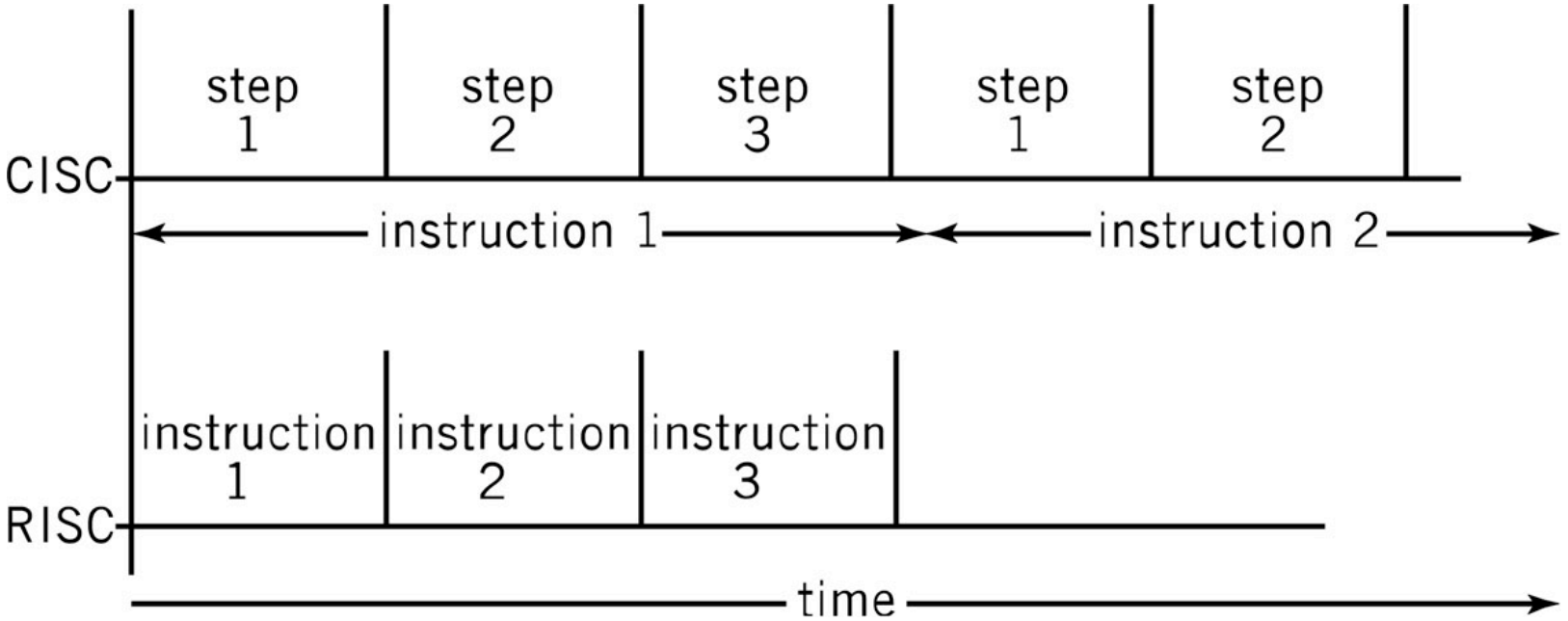
- **L'architecture RISC (**Power PC, Sun Sparc, Motorola 68000)
  - Motivation
    - Statistique: 80% des traitements des langages de haut niveau faisaient appel a seulement 20% des instructions du microprocesseur => réduire le jeu d'instructions à celles le plus couramment utilisées et d'en améliorer la vitesse de traitement.

Microprocesseur

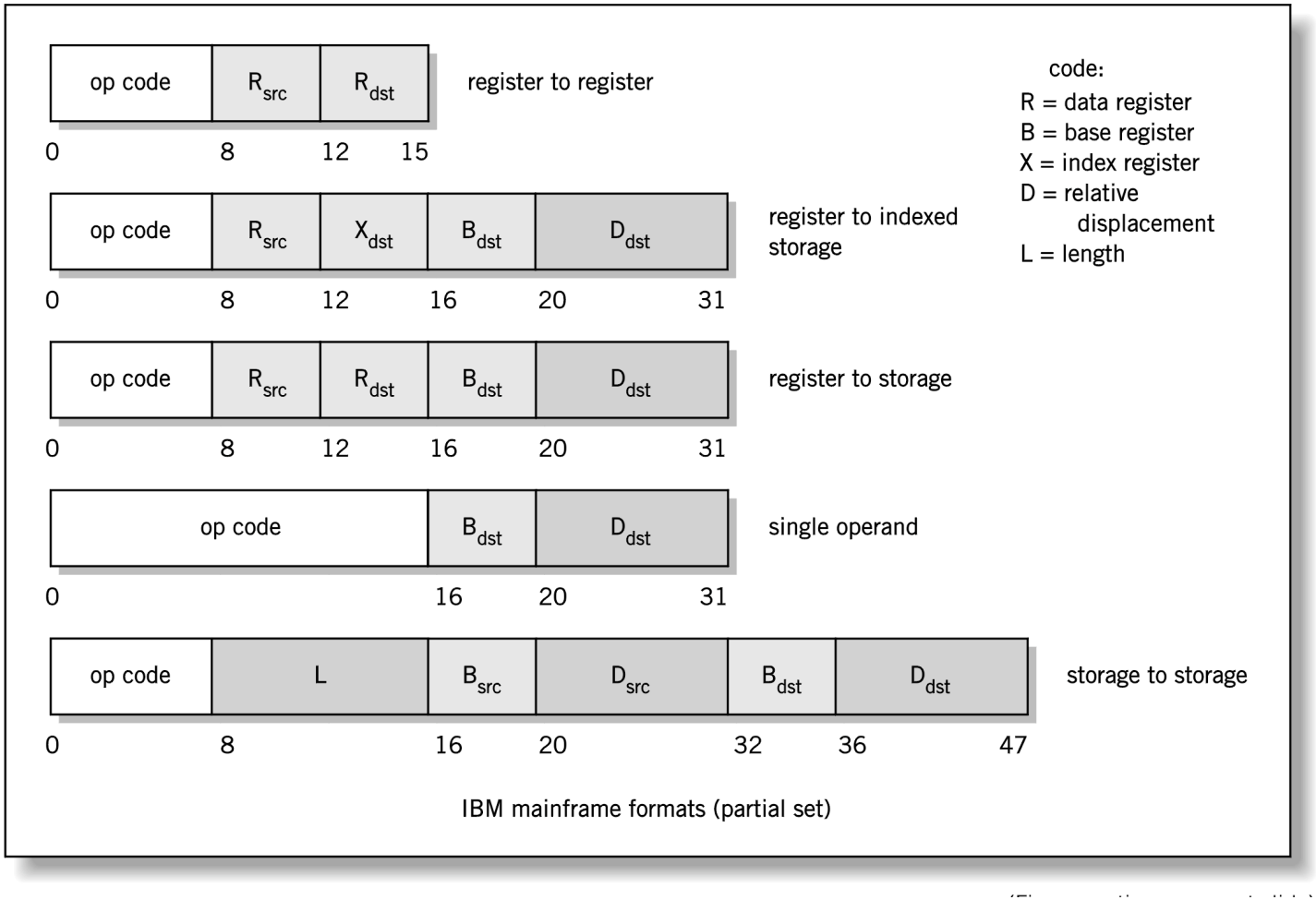


- **L'architecture RISC**
  - Les instructions sont en nombre réduit => une diminution de la complexité de la partie unité de commande
  - Une implantation d'instructions de longueurs fixes
  - Chacune de ces instructions s'exécutent ainsi en un cycle d'horloge

Exécution CISC vs RISC

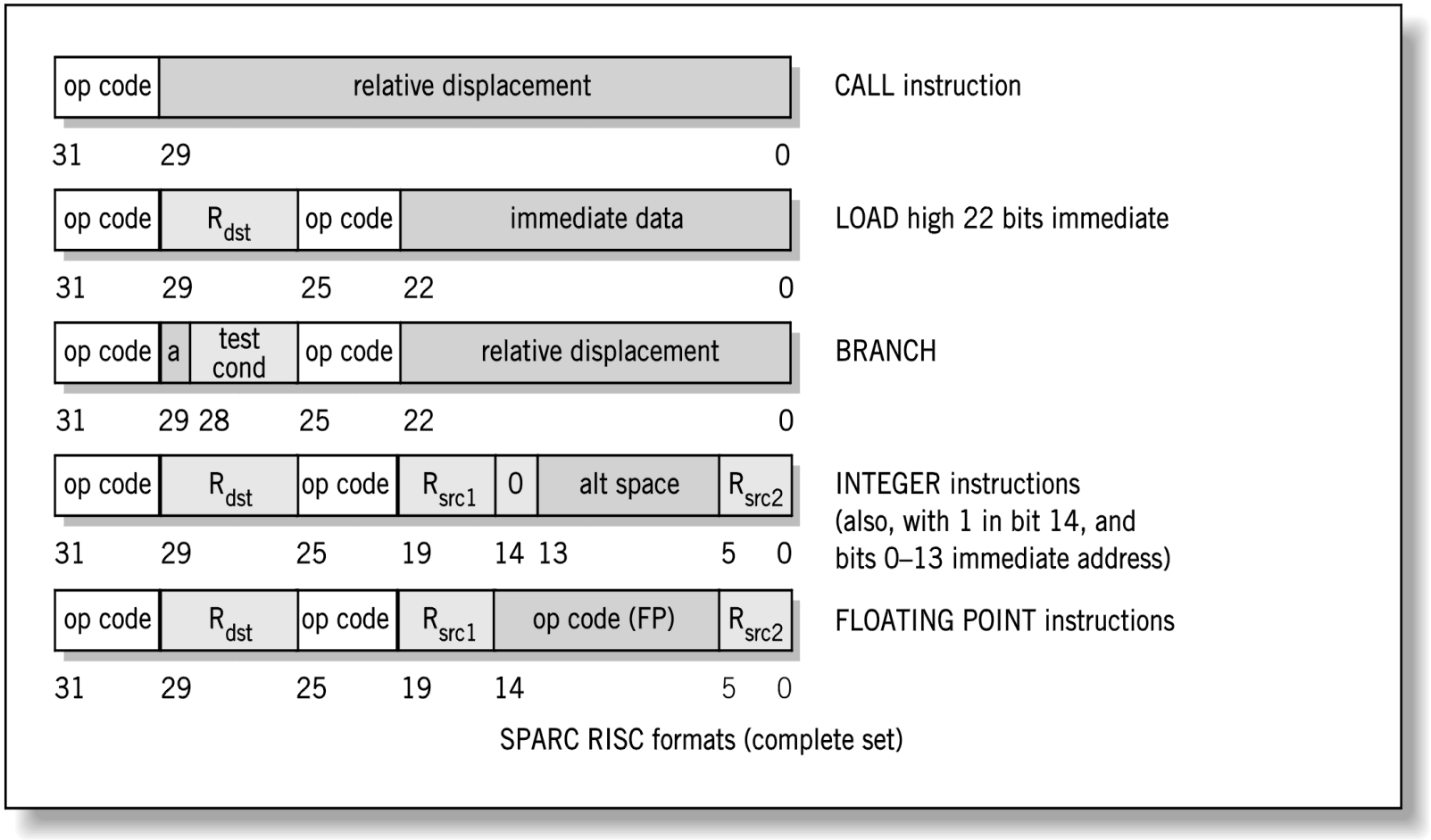


Formats d'instructions: CISC



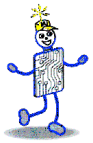
IBM mainframe formats (partial set)

Formats d'instructions: RISC



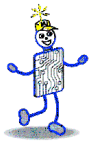
SPARC RISC formats (complete set)

RISC contre CISC



Architecture RISC	Architecture CISC
Instructions simples ne prenant qu'un seul cycle	Instructions complexes prenant plusieurs cycles
Instructions au format fixe	Instructions au format variable
Décodeur simple (câblé)	Décodeur complexe (microcode)
Beaucoup de registres	Peu de registres
Seules les instructions LOAD et STORE ont accès à la mémoire	Toutes les instructions sont susceptibles d'accéder à la mémoire
Peu de modes d'adressage	Beaucoup de modes d'adressage
Compilateur complexe	Compilateur simple

Architectures VLIW - EPIC

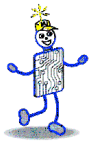


Architecture VLIW - EPIC

- VLIW – Very Long Instruction Word
- EPIC – Explicitely Parallel Instruction Computer

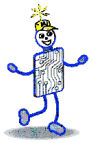
- Le but de ces deux type d'architecture est d'augmenter la vitesse d'exécution du processeur en traitant des instructions/opérations en parallèle

Architecture VLIW



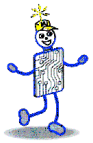
- CPU Transmeta Crusoe
- Caractéristiques:
  - Instruction de longueur 128 bits: molécule
    - Divisée en 4 atomes de 32-bit (atome = instruction)
    - 4 instructions pouvant être exécutées simultanément
  - 64 registres d'usage général
  - Logiciel de Codage Morphing
    - Traduction du code machine d'autre CPU en molécules
    - Traduction est intégrée à l'architecture

Architecture EPIC



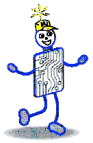
- CPU Intel Itanium
- Même but mais avec caractéristiques différentes:
  - Instruction de longueur 128-bits
    - Comprenne 3 instructions de 41-bits
    - Comprenne 5 bits pour identifier le type d'instruction
  - 128 registres d'usage général de 64-bits
  - 128 registres flottants de 82-bits
  - Instructions famille Intel X86
  - Les 5 bits sont des bits d'informations qui permettent d'identifier les dépendance potentielles entre exécutions

Architectures VLIW vs EPIC



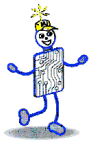
- L'ordonnancement des opérations (+ gestion des priorités, dépendance, etc.) dans l'instruction de 128 bits est
  - Intégré dans l'architecture matériel pour le VLIW
  - Géré par le programmeur ou le compilateur (logiciel) - EPIC

Amélioration des performances



- Augmenter la vitesse des microprocesseurs
  - L'augmentation de la fréquence d'horloge
- Travailler sur l'architecture interne du microprocesseur
- Améliorer les performances du bus de communication entre processeur et mémoire centrale
- Parallélisation de certains opérations; faire plusieurs choses en même temps
  - Le parallélisme au niveau des instructions
  - Le parallélisme au niveau des processeurs

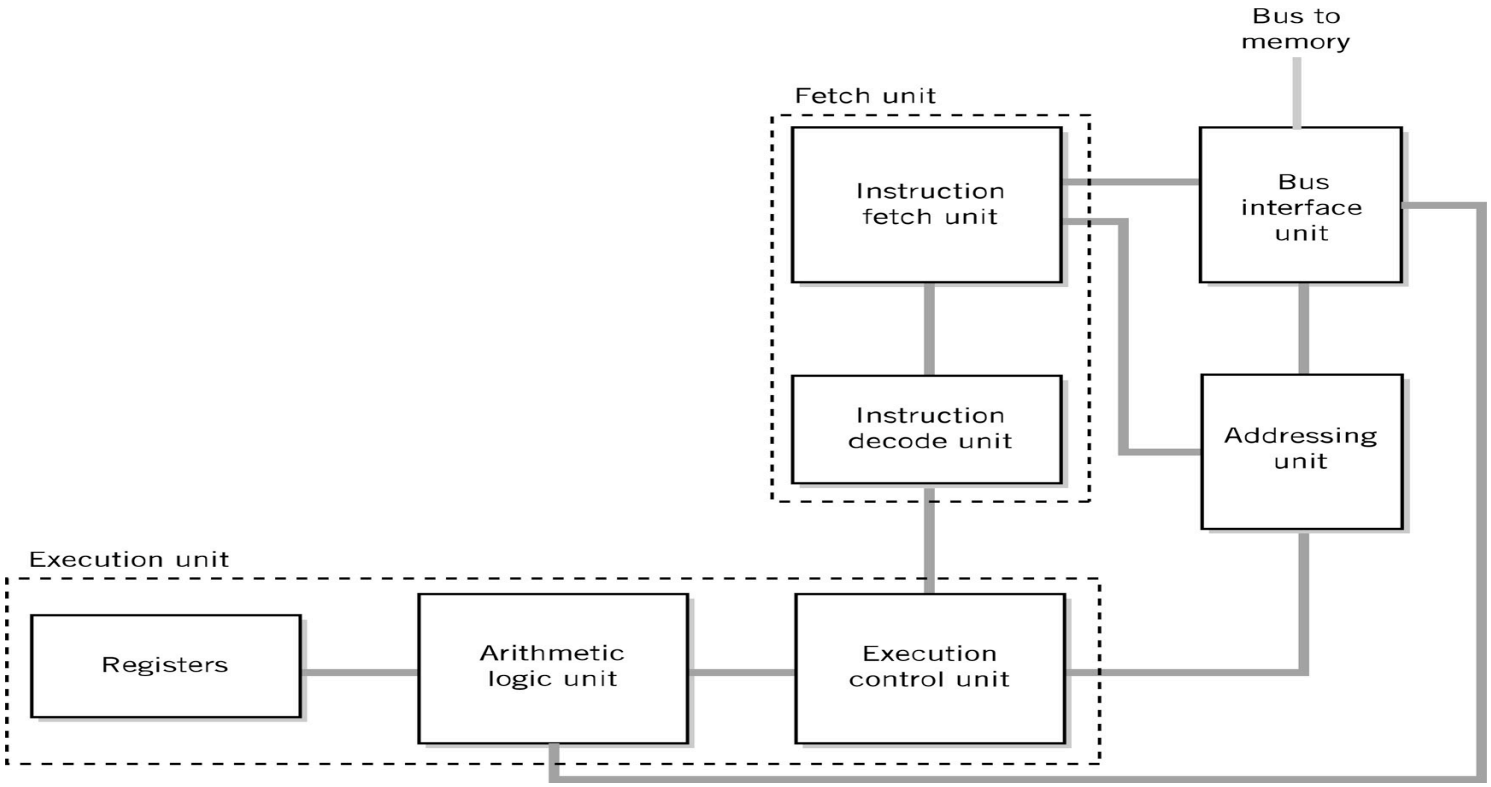
Conception des ordinateurs modernes



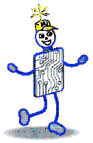
- Unités Fetch et Execute séparées
- Technique du pipeline
- Unités d'exécutions parallèles
- Traitement Scalaire
- Traitement Superscalaire
- Traitement d'instructions de branchement

Conception des ordinateurs modernes

- Unités Fetch et Execute séparées

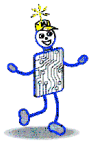


Unités Fetch et Execute séparées

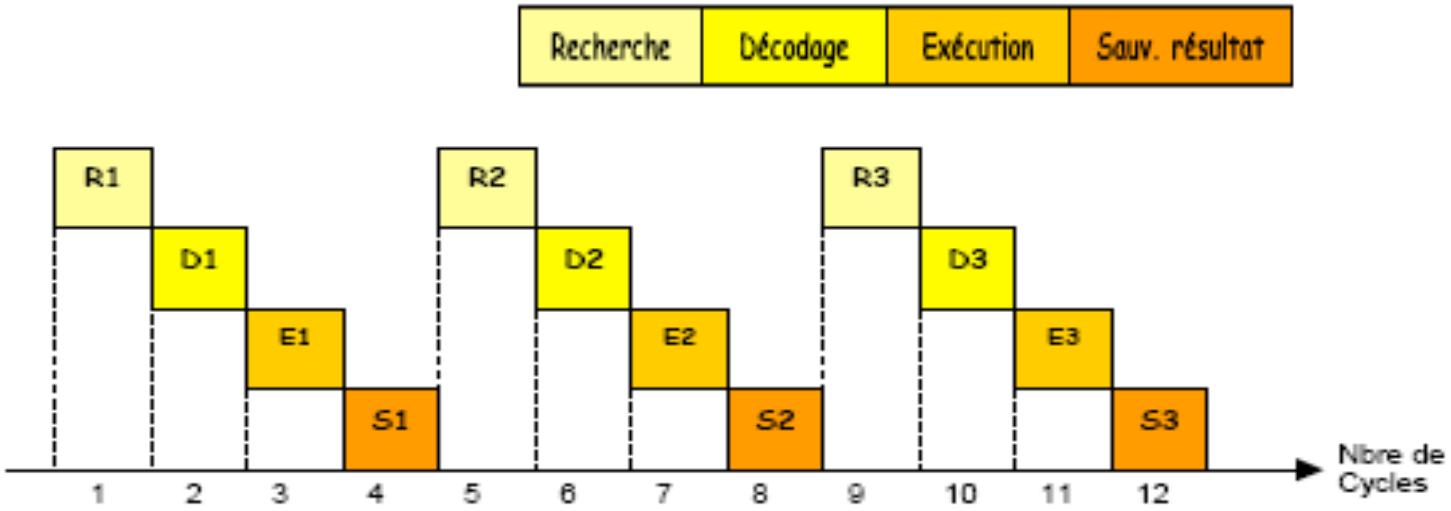


- Unité Fetch
  - Unité de recherche de l'instruction
  - Unité de décodage
    - Détermine un code d'opération
    - Identifier type de l'instruction et les opérandes
  - Plusieurs instructions sont recherchées en parallèle et placées dans un tampon
  - IP – registre « Instruction Pointer » – pointe à l'instruction en cours de traitement
- Unité Execute
  - Reçoive les instructions de l'unité de décodage
  - Unité d'exécution appropriée serve l'instruction

Technique du pipeline

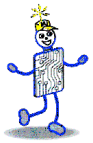


- Exemple de l'exécution en 4 phases d'une instruction



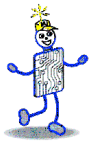
- Le fonctionnement d'un microprocesseur simple n'est pas efficace

Technique du pipeline

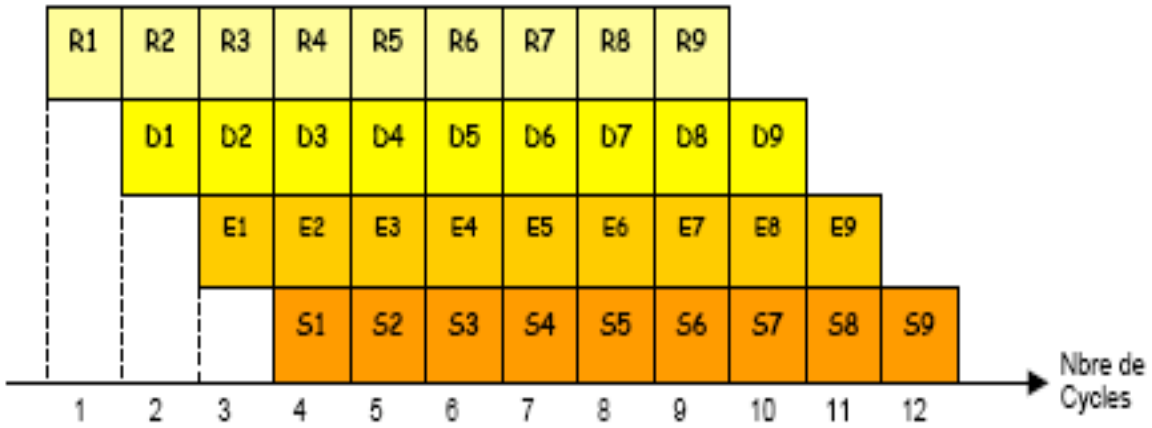


- Technique de pipeline
  - Idée inspirée de l'organisation du travail à la chaîne
  - L'exécution d'une instruction peut être décomposée en plusieurs phases qui s'exécutent indépendamment les unes des autres si l'on dispose d'unités fonctionnelles (du matériel) le permettant
  - La séquence d'instructions fetch/execute est exécutée comme si celle-ci était dans une chaîne de montage.

Technique du pipeline

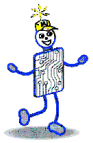


- Exemple de l'exécution en 4 phases d'une instruction



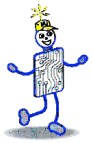
- Le temps d'exécution d'une instruction n'est pas réduit mais le débit d'exécution des instructions est considérablement augmenté.

Technique du pipeline



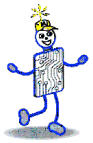
- Si la machine débute l'exécution d'une instruction à chaque cycle et le pipeline est pleinement occupé à partir du quatrième cycle. Le gain obtenu dépend donc du nombre d'étages du pipeline

Technique du pipeline



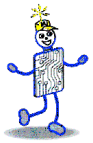
- Pour exécuter **n** instructions, en supposant que chaque instruction s'exécute en **k** cycles d'horloge, il faut :
  - **n** * **k** cycles d'horloge pour une exécution séquentielle
  - **k** cycles d'horloge pour exécuter la première instruction puis **n-1** cycles pour les **n-1** instructions suivantes si on utilise un pipeline de **k** étages
- Gain: G = (n*k)/(k+n-1)

Technique du pipeline



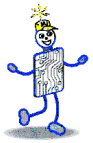
- Le temps de traitement dans chaque unité doit être a peu près égal sinon les unités rapides doivent attendre les unités lentes.
- Exemples :
  - L'Athlon d'AMD comprend un pipeline de 11 étages.
  - Les Pentium 2, 3 et 4 d'Intel comprennent respectivement un pipeline de 12, 10 et 20 étages.

Technique du pipeline



- Facteurs de ralentissement du pipeline
  - Les conflits de ressources
  - Les dépendances des données
  - Les conflits liés au contrôle
    - Les instructions de sauts inconditionnels et de branchement
- Malgré ces circonstances la technique de pipeline reste très efficace

Technique du pipeline

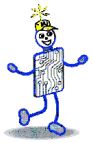


- Problèmes pour les branchements
  - Pipelines séparés pour les deux possibilités
  - Prédiction basée sur les branchements effectués à l'exécution précédente
- Problèmes de dépendance entre instructions
  - Réarrangement de la séquence d'instructions pour maintenir le pipeline plein

Plusieurs unités d'exécutions

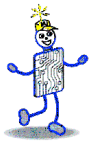
- Différentes instructions ont différent nombre des étapes dans leur cycle
- Différences dans chaque étape
- Chaque unité d'exécutions est optimisée pour un type général de l'instruction
- Plusieurs opérations sont traitées à la fois

Architecture superscalaire

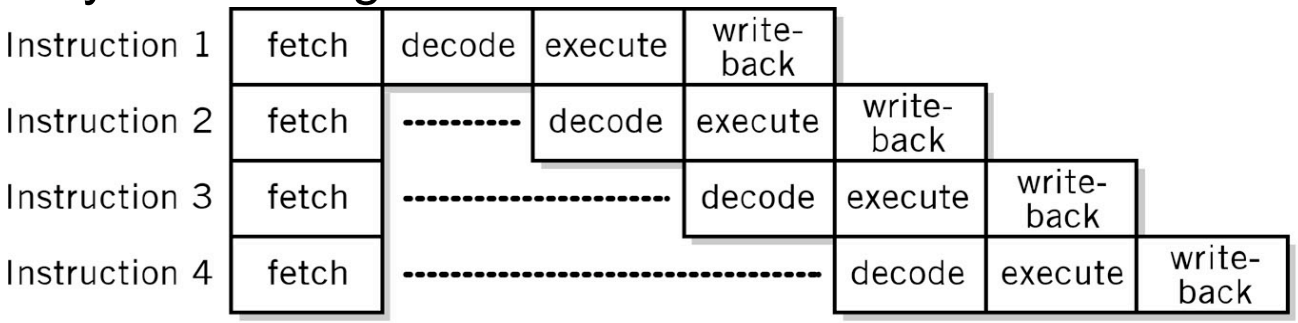


- Exécute plus qu'une instruction par un cycle d'horloge
- Séparer cycles d'extraction de l'instruction et d'exécution
- Garder les données de phases Extraction et Décodage
- Disposer dans le pipeline de plusieurs unités d'exécutions

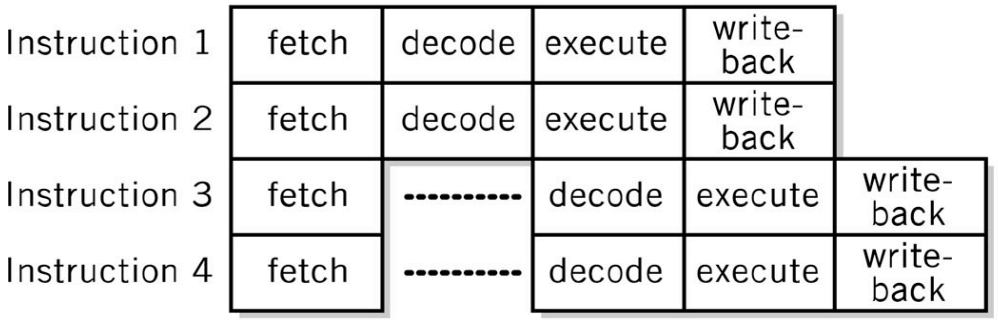
Scalaire vs. Superscalaire



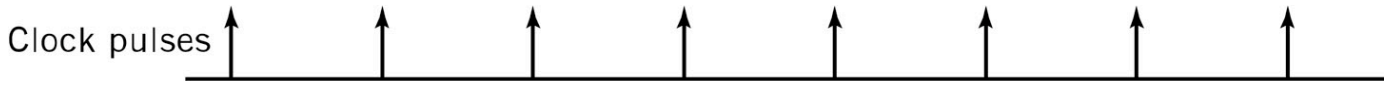
- Processeur Scalaire
  - Processeur pour lequel la vitesse d'exécution moyenne d'une instruction égale approximativement à la vitesse d'un cycle d'horloge



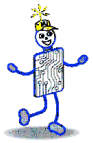
a. Scalar

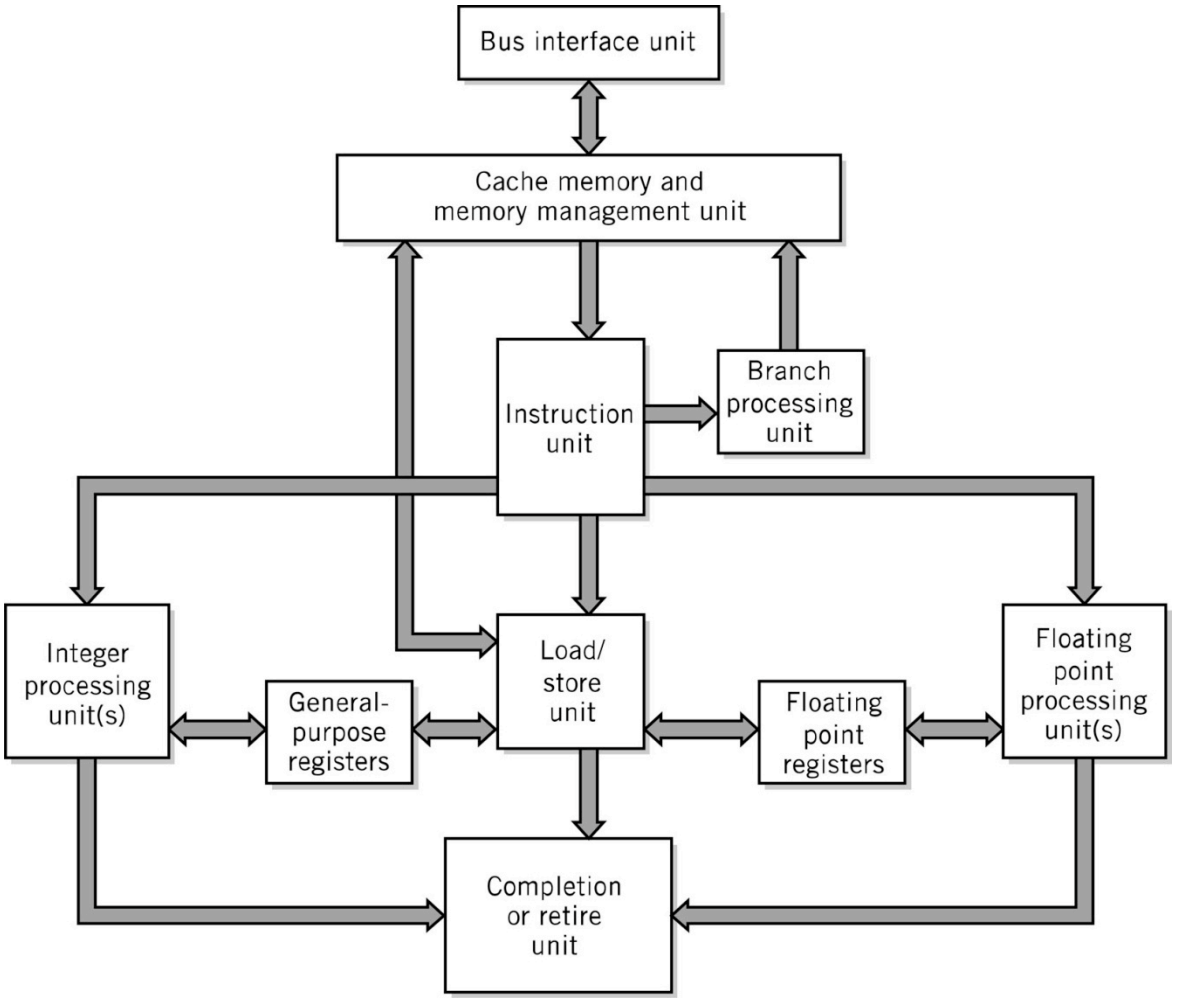


b. Superscalar

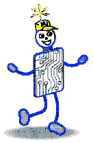


Bloc Diagramme du CPU superscalaire



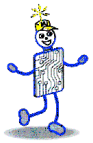


Problèmes de traitement Superscalaire



- Traitement “Out-of-order” – dépendances (hazards)
- Dépendances de données
- Dépendances de branchement et une exécution spéculative
- Exécutions spéculatives parallèles ou prédiction de branchement
- Table historique de branchement
- Conflit d’accès aux registres
  - Renommer ou utiliser les registres logiques

Mémoires



- Les éléments de mémoire d'un ordinateur se répartissent en plusieurs niveaux caractérisés par
  - leur capacité
  - Leur temps d'accès
- Hiérarchie
  - Les niveaux sont ordonnés en fonction
    - Temps d'accès
    - Capacité et coût par bit

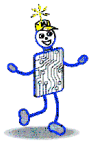
Hiérarchie des mémoires

Increasing storage capacity

Device	*Typical access times*
CPU registers	0.25 nsec
Cache memory (SRAM)	1-10 nsec
Conventional memory (DRAM)	10-50 nsec
Flash memory	120 μsec
Magnetic disk drive	10-50 msec
Optical disk drive	100-500 msec
Magnetic tape	0.5 and up sec

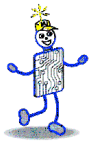
Increasing access times

Mémoire centrale



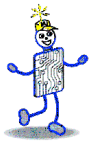
- Mémoire centrale ou principale contient les instructions et les données des programmes que l'on désire exécuter, ainsi qu'une partie du système d'exploitation nécessaire au bon fonctionnement de l'ordinateur
- Depuis le début des années 70, les mémoires à semi-conducteurs constituent les éléments de base de toute mémoire centrale

Mémoires à semi-conducteurs



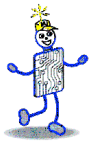
- RAM - Mémoire à accès aléatoire
  - Le temps d'accès est indépendant du numéro de la cellule adressée
- On distingue différents types de mémoires RAM
  - DRAM – « Dynamic RAM », mémoire vive dynamique
  - SRAM – « Static RAM », mémoire vive statique

DRAM



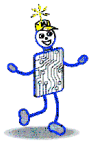
- Pas chères, consommation électrique fiable, grande densité d'intégration
- Les boitiers de mémoire dynamique enferment une pastille de silicium sur laquelle est intégré un très grand nombre de cellules binaires. Chaque cellule binaire est réalisée à partir d'un transistor relié à un petit condensateur.
- L'inconvénient de cette technique simple est que le condensateur se décharge seul au cours du temps (courants de fuite). Il est donc nécessaire de rafraichir tous les condensateurs

SRAM



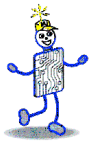
- Les mémoires statiques n'utilisent pas de condensateurs : chaque cellule binaire est réalisée à l'aide de 4 transistors formant un *bistable*
- Les SRAM permettent des temps d'accès plus court que les DRAM, mais sont plus couteuses car leur construction demande 4 fois plus de transistors que les DRAM.
- Les SRAM sont utilisées lorsque l'on désire maximiser les performances
  - mémoires caches

Mémoires non volatiles



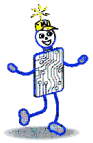
- *ROM*
  - « Read-only Memory » - Mémoire à lecture seul
- EEPROM
  - « Electrically Erasable Programmable ROM » - ROM programmable et Électriquement Effaçable.
- *Mémoire Flash*
  - Basée sur le principe des EEPROM
  - Plus performantes que les disques mais chères
  - Se programme électriquement par blocs

Amélioration reposant sur l'accès mémoire



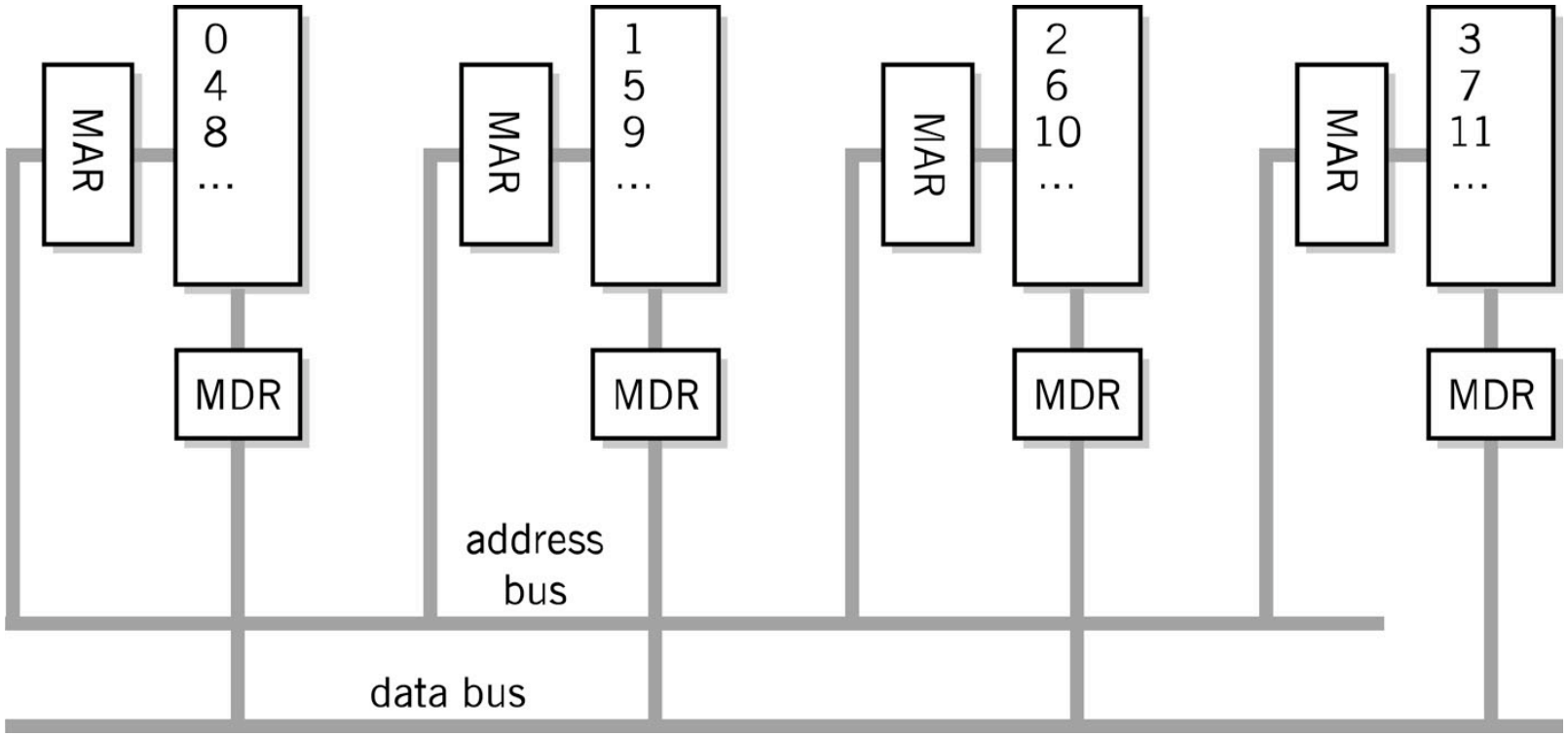
- Les échanges entre le processeur et la mémoire sont très nombreux
  - Un programme et ses données doivent êtres placés en mémoire centrale afin d'être exécutés par le processeur

Amélioration reposant sur l'accès mémoire

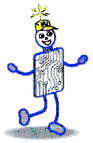


- Un accès mémoire est lent comparativement à la vitesse de processeur
  - CPU 2Ghz = 1 cycle en 0.5 ns
  - 30ns DRAM = 1 accès en 50 cycles
- Méthodes pour diminuer le temps d'accès à la mémoire
  - Accès à la mémoire avec un bus de données plus grand
    - Extraire plusieurs octets au lieu de 1 octet chaque fois
  - Entrelacement de Mémoire
    - Partitionner une mémoire en sousections, chaque avec ses registres de données et d'adresse
  - Mémoire Cache

Entrelacement de mémoire

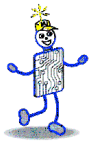


Les mémoires caches



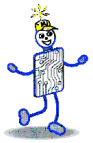
- **Pourquoi a-t-on besoin de mémoire cache ?**
  - CPU 2Ghz = 1 cycle en 0.5 ns
  - 30ns DRAM = 1 accès en 50 cycles
  - Même le plus rapide des disques dur a un temps d'accès de 10 millisecondes
    - Avec un CPU de 2 GHz, le CPU attendant 10 ms **gaspille 20 millions de cycles d'horloge!**
    - Le processeur ne fonctionne pas au meilleur rythme

Les mémoires caches

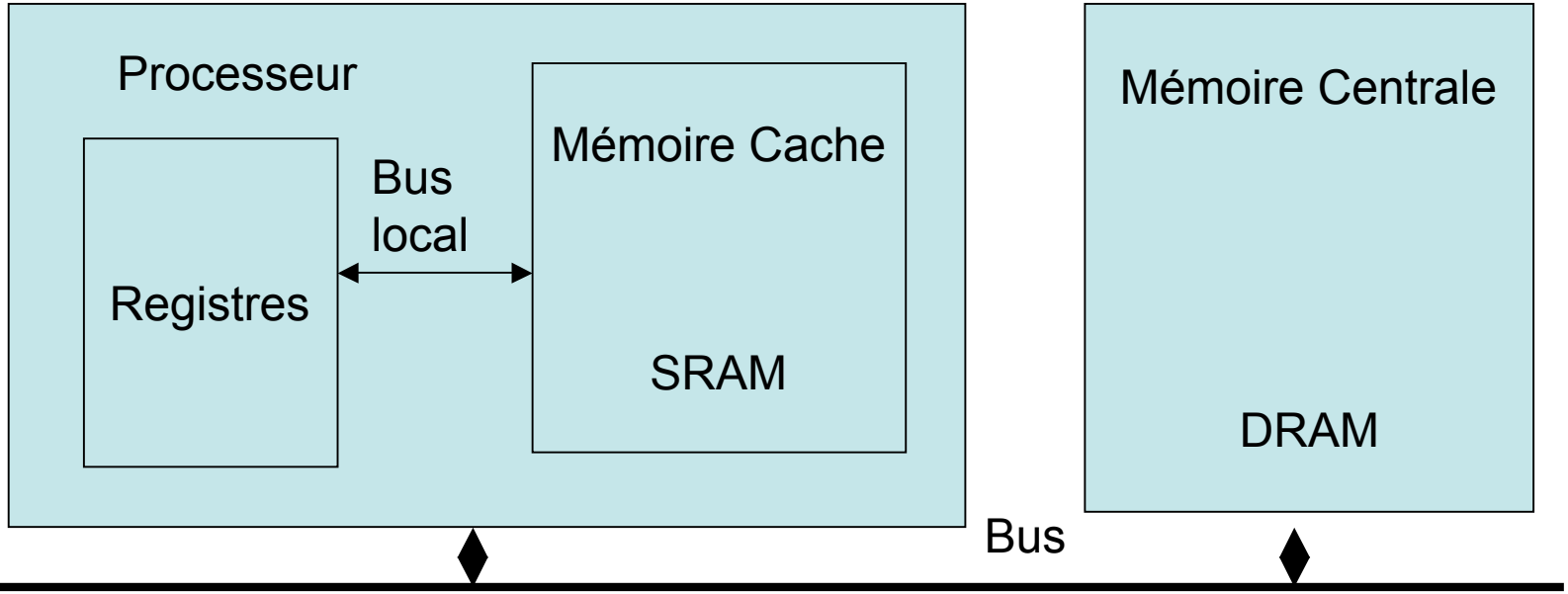


- Typiquement, 90 % du temps d'exécution d'un programme est dépensé dans juste 10 % du code => principe de localité
  - Localité Temporelle
    - Une cellule mémoire référencé a plus de chance d'être référencée encore une autre fois
  - Localité Spatiale
    - Une cellule mémoire voisine a plus de chance d'être référencée (données stockées continûment)

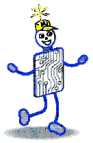
Les mémoires caches



- Ajout d'un bloc mémoire rapide dans le CPU
- Principe de fonctionnement
  - Faire coopérer des mémoires de faible capacité très rapides et à proximité du processeur avec des mémoires plus lentes et de grandes capacités

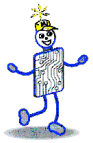


Les mémoires caches



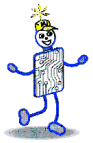
- Lecture d'un mot
  - Si l'information est présente dans le cache on parle de succès (cache hit)
  - L'information n'est pas dans le cache – un échec (cache miss)
- L'efficacité du cache dépend de son taux de succès

Les mémoires caches

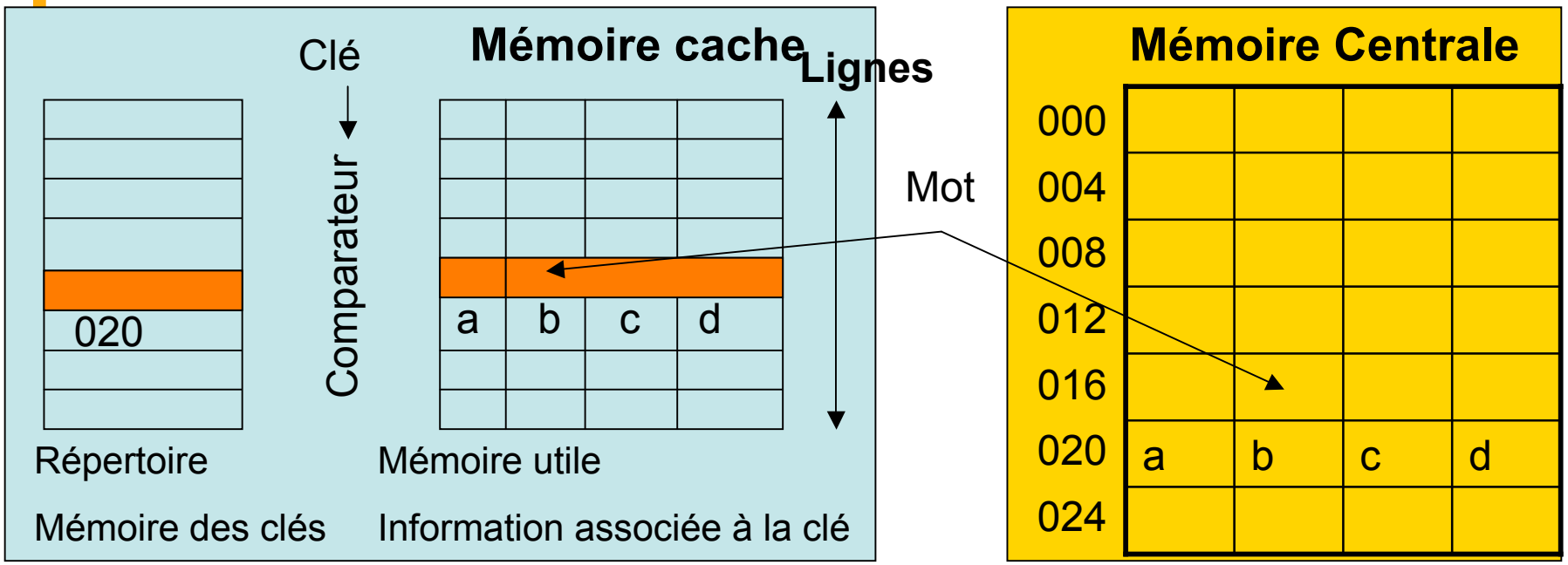


- Écriture d'un mot
  - Accéder au cache pour vérifier si l'information est présente dans le cache et éventuellement la modifier
  - Une écriture dans le cache modifiant une information => la modification de cette information dans la mémoire centrale
    - Écriture immédiate (Write through)
      - On écrit simultanément dans le cache et la mémoire principale
    - Écriture différée (Write Back)
      - Mettre à jour la mémoire centrale quand l'information de la mémoire cache doit être remplacée
      - Le bus de communication est libre

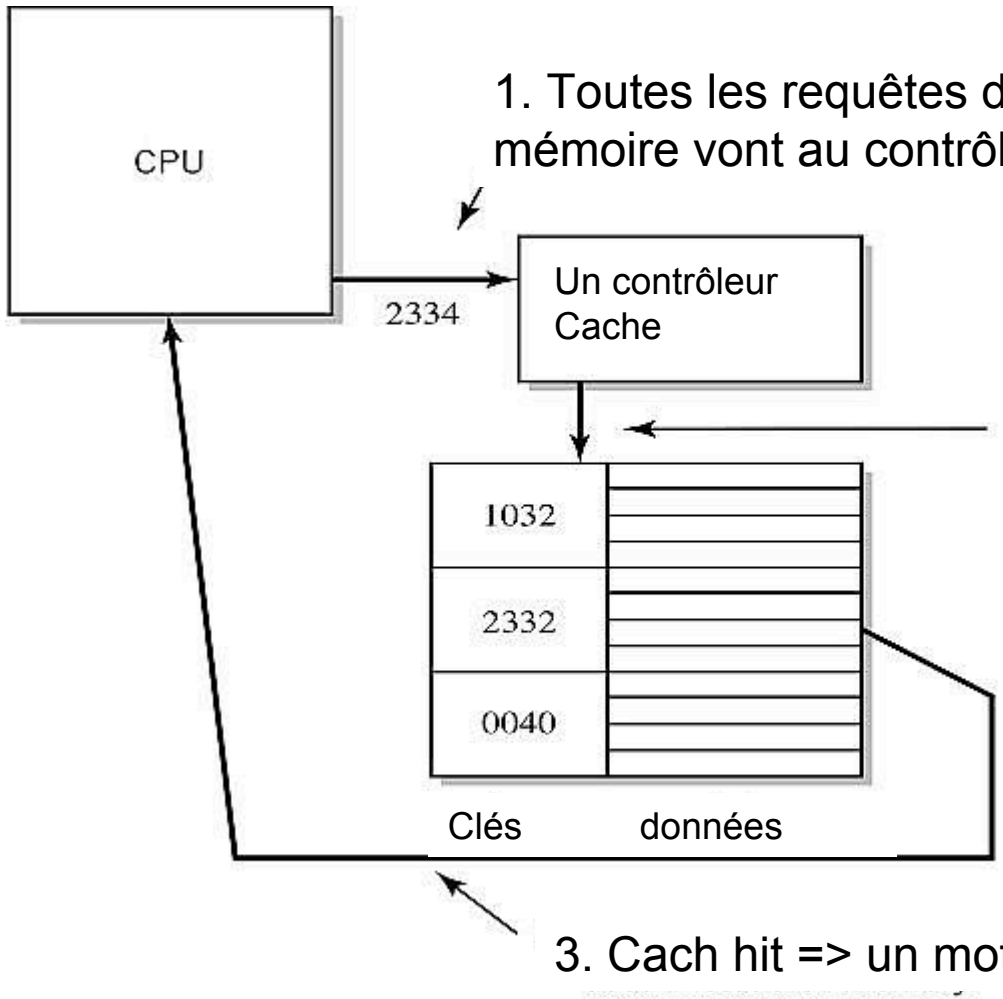
Les mémoires caches



- Organisation et fonctionnement
- Le principe de localité => considérer la mémoire centrale comme une suite de blocs mémoires : les lignes de mémoires



Cache hit

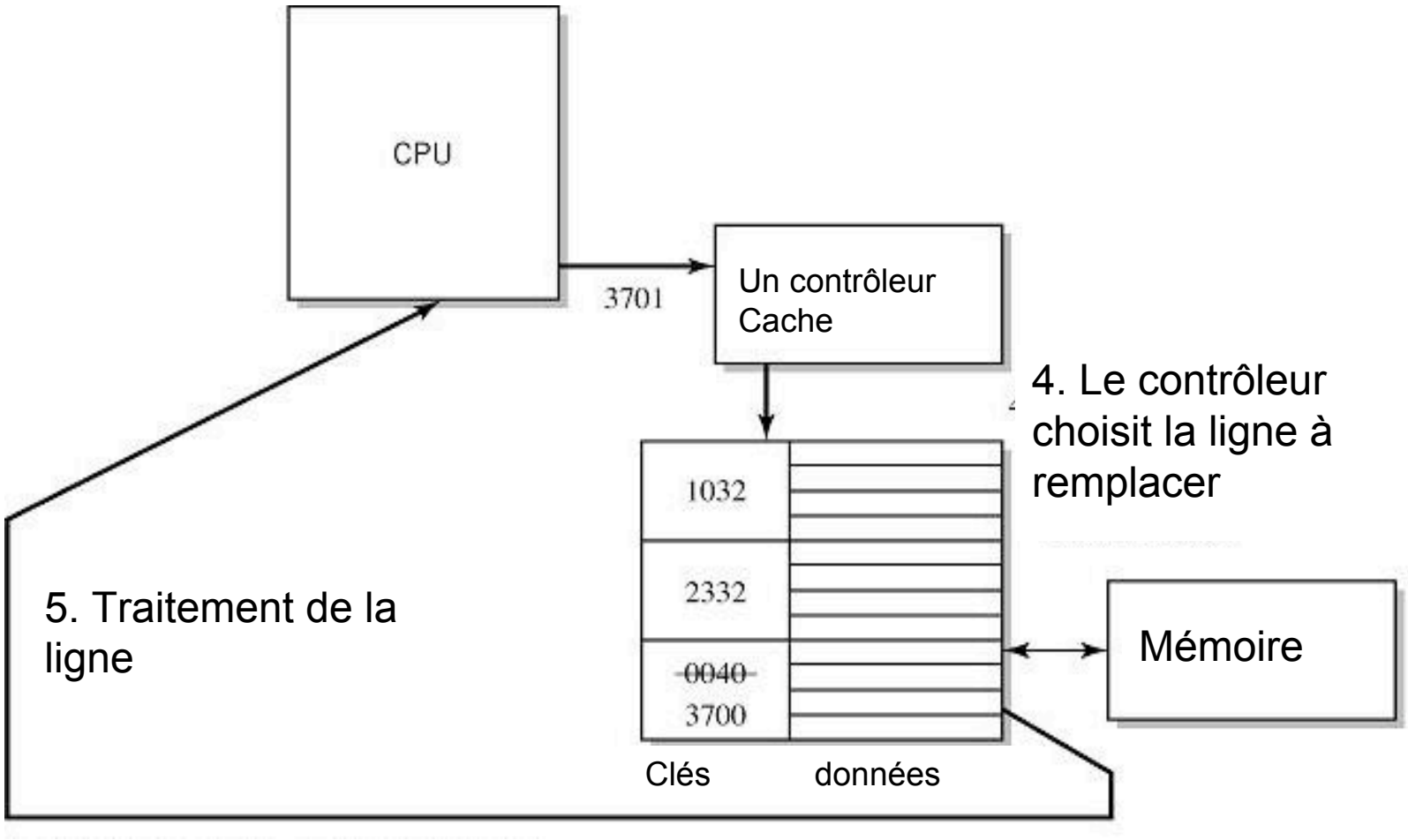


1. Toutes les requêtes de l'accès à la mémoire vont au contrôleur Cache

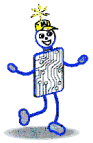
2. Le contrôleur vérifie une clés si la ligne demandée est dans le cache

3. Cach hit => un mot du cache est utilisé

Cache miss

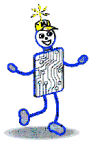

CPU
3701
Un contrôleur Cache
4. Le contrôleur choisit la ligne à remplacer
1032
2332
0040
3700
Mémoire
5. Traitement de la ligne
Clés
données

Cache



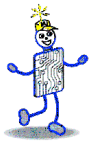
- Facteurs déterminants une bonne conception d'un cache
  - La taille
  - La longueur des lignes de cache
  - Le mode de gestion du cache
    - Minimiser le temps de vérification de présence de l'information
  - Le nombre et la localisation du ou des caches

Cache



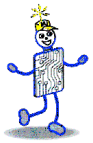
- Types de cache
  - Cache direct
  - Cache purement associatif
  - Cache mixte

Cache

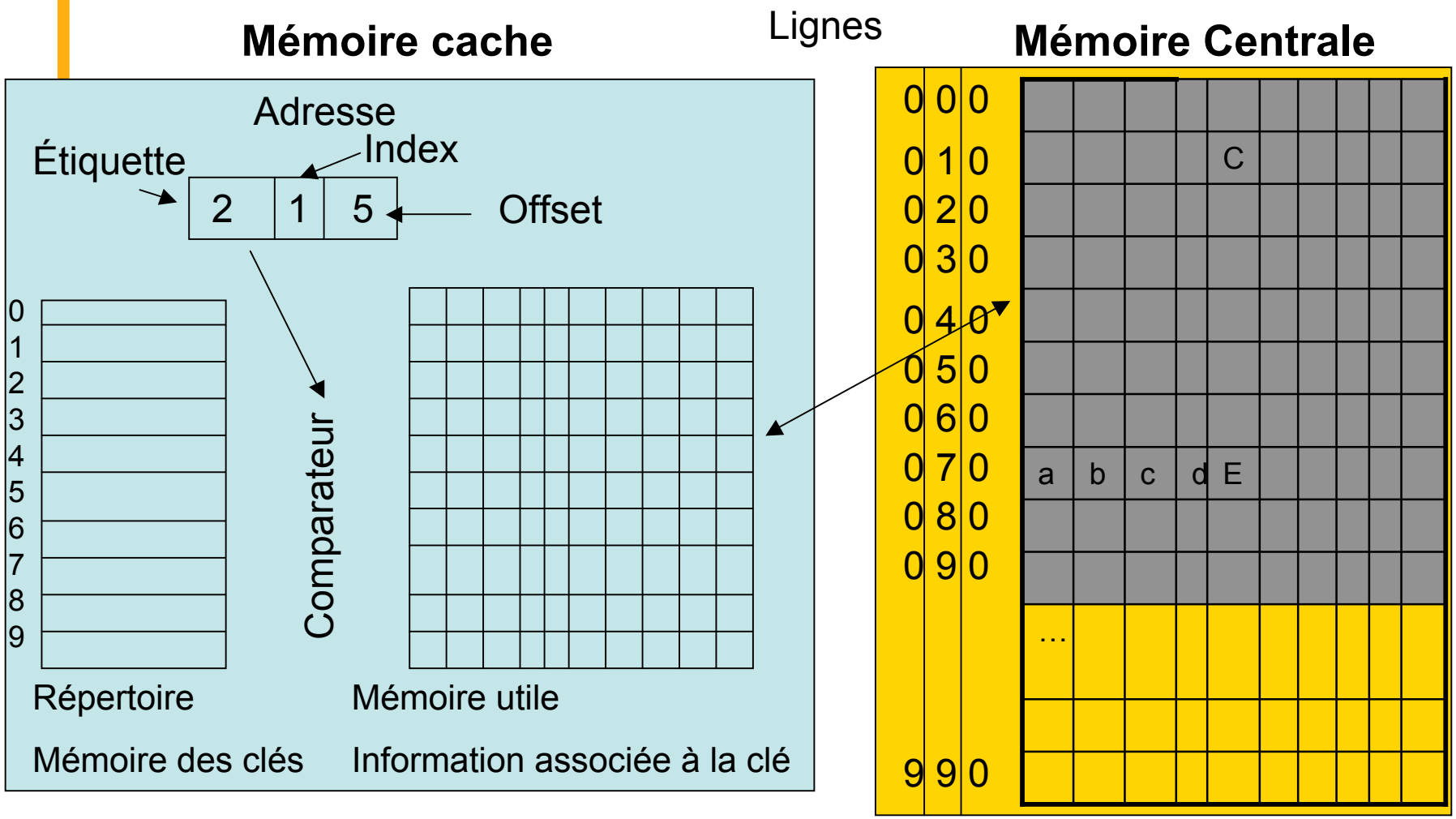


- Cache à correspondance directe
  - Cache le plus simple
  - On va affecter à chaque ligne de notre mémoire cache une zone de mémoire RAM fixe et de taille fixe
  - Il y a une correspondance directe entre mémoire RAM et mémoire cache

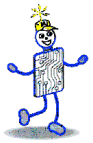
Cache direct



- Lorsqu'une adresse est présentée au cache, le contrôleur de cache décompose cette adresse

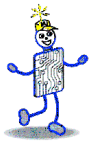


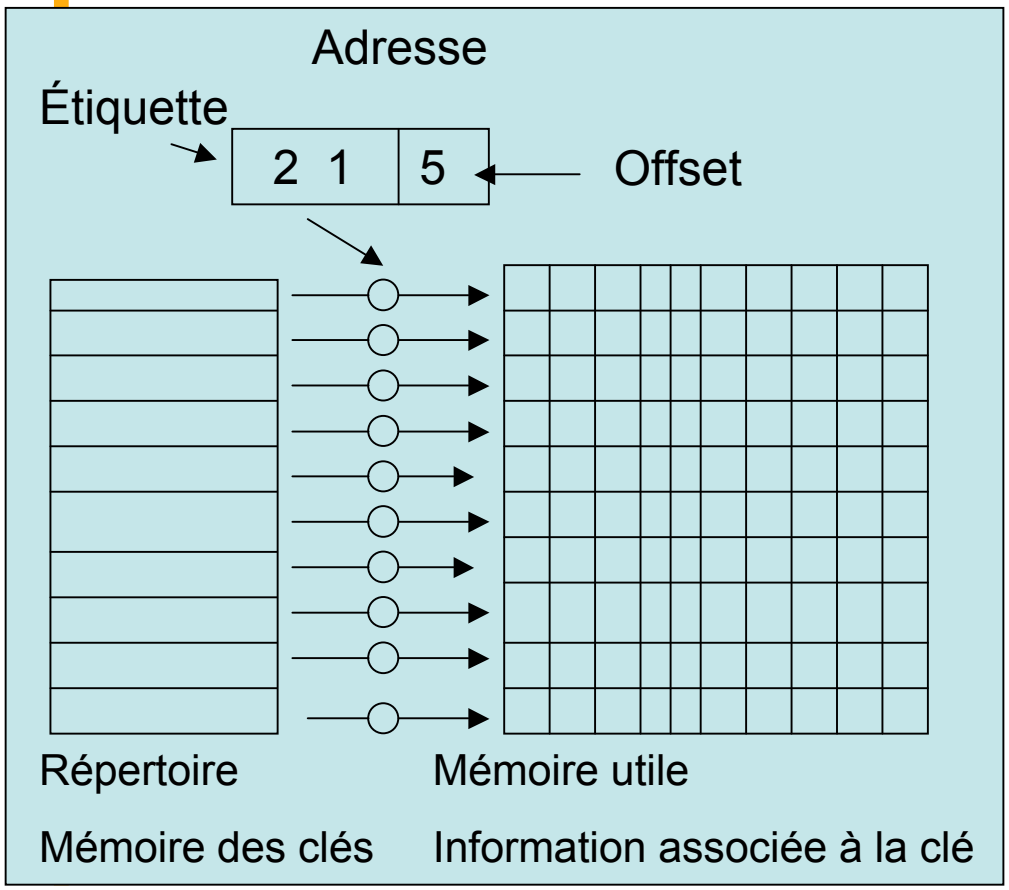
Cache

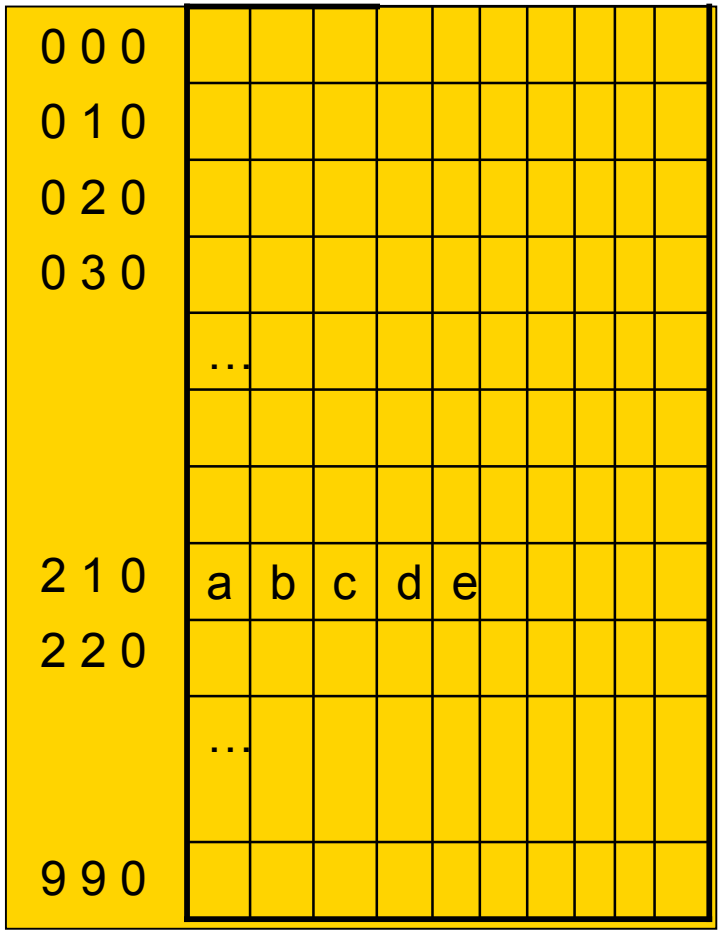


- Cache purement associatif
  - Plus complexe et plus cher à construire
  - Il y a autant de comparateurs que de lignes de cache
  - Une ligne de données entre dans n'importe quelle entrée libre du cache
  - L'adresse est interprétée comme une étiquette et un offset

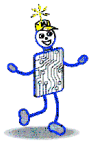
Cache purement associatif





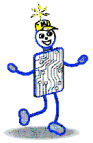


Cache purement associatif



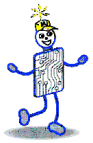
- Gestion plus complexe
  - Vérification si le cache est plein
  - Algorithme de remplacement
    - LRU (remplacer la ligne la moins récemment utilisée)
    - FIFO
    - LFU (remplacer la ligne la moins fréquemment utilisée)

Cache



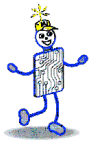
- Cache mixte
  - Utilise les techniques des deux caches précédents
  - Le cache est divisé en blocs gérés comme des caches directs
  - En cas d'échec, la ligne de mémoire correspondante doit être chargée dans une des lignes référencées
  - Utilisation d'un algorithme de remplacement

Nombre et localisation des caches

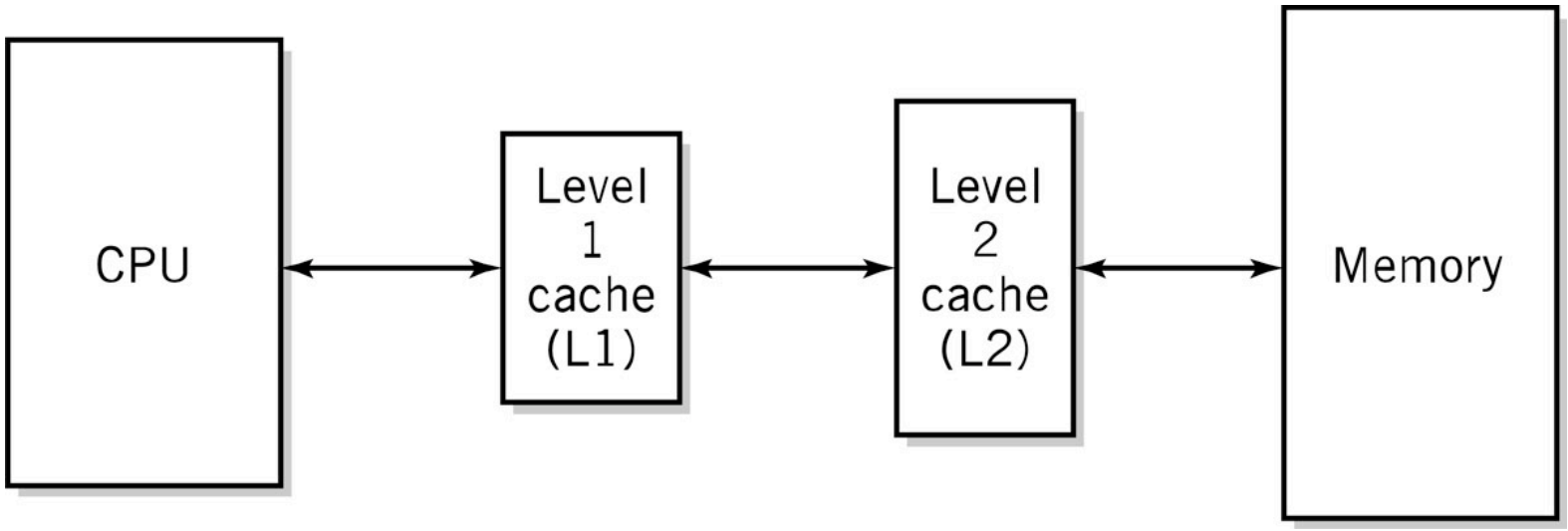


- Hiérarchie de mémoires cache
  - Le premier niveau de mémoire cache, petite et très rapide, est placé dans le processeur (cache de niveau 1)
  - Le deuxième niveau, de capacité plus importante et d'accès également rapide, est mis à l'extérieur du processeur (cache de niveau 2)
  - Le troisième niveau est constitué par la mémoire centrale

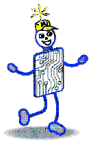
Caches de deux niveaux



- Pour être utile, le second niveau de cache doit avoir plus de mémoire que le premier niveau

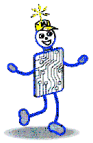


Performance de la mémoire cache



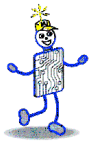
- “Hit ratios” de 90% couramment obtenu
- Gain de plus de 50 % en rapidité d’exécution
- Technique de mémoire cache utilisée dans les disques durs

Parallélisme au niveau du processeur



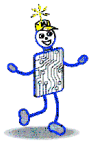
- Raisons
  - Le parallélisme au niveau des instructions
    - Multiplie la vitesse par un facteur de 5 à 10
  - Plus de parallélisme
    - Développement des ordinateurs équipés de plusieurs unités centrales
    - Gains en vitesse de cinquante, cent et plus
- Les Systèmes Multiprocesseurs
  - CPU multiples dans un ordinateur
  - Processeurs multicœurs - CPUs sont intégrés sur un seul chip

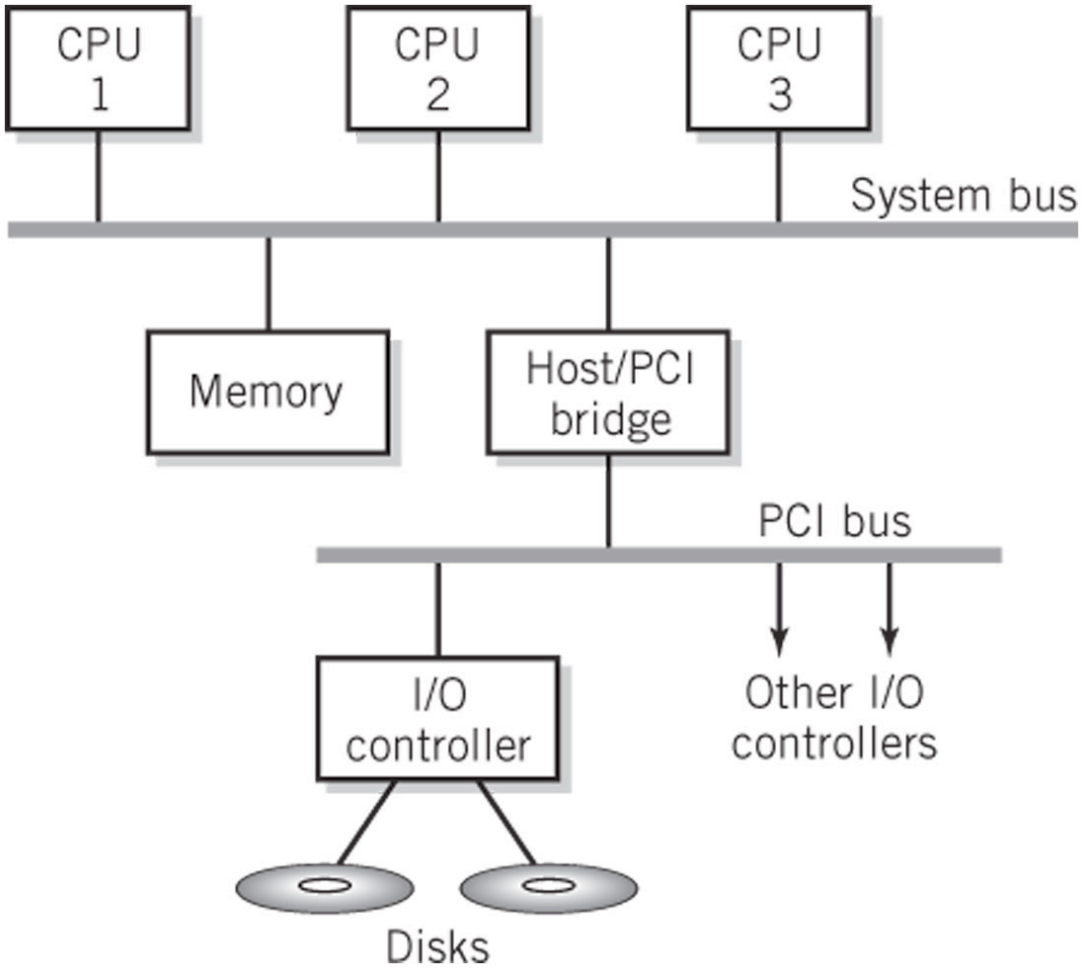
Les Systèmes Multiprocesseurs



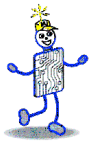
- Accès identique aux programmes, données, mémoire partagée, I/O, etc.
- Facilement étendent exécution multitâche, et exécution redondante de programmes
- Deux méthodes de configuration
  - Maître-esclave – approche centralisée
  - Symétrique - approche distribuée
    - Symmetrical multiprocessing - SMP

Configuration typique du système multiprocesseurs



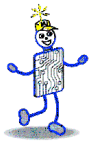


Approche centralisée



- CPU Maître
  - centralise et gère les appels systèmes
  - ordonnancement (cherche à équilibrer la charge)
- Avantages
  - Simplicité
  - Protection de système et données
- Désavantages
  - CPU Maître peut saturer
  - Fiabilité – si CPU Maître tombe en panne tout le système tombe en panne

Approche distribuée



- Tous les processeurs sont équivalents. Ils peuvent tous exécuter le système.
- Désavantages
  - Conflits de ressources - mémoire, i/o, etc.
  - Implémentation complexe
- Avantages
  - Fiabilité accrue
  - Support de tolérance aux fautes est simple
  - Charge équilibrée