

CC-CM-P - LIF6 Architecture matérielle et logicielle

Nicolas Louvet

Aucun document autorisé, calculatrices et autres moyens de calcul interdits, durée 45 min.

Veillez à bien numéroter vos réponses, en préservant l'ordre du sujet si possible.

I Questions de cours (15 pts)

Q.I.1) - (3 pts) Nous avons vu en cours que l'organisation des ordinateurs peut être décrite selon un modèle en couches. Citer les différentes couches que nous avons décrites, en commençant par les couches de plus bas niveau, et en donnant pour chacune une description.

Niveau 1 – couche micro-architecture A partir des portes logiques, il est possible de concevoir des unités de traitement plus évoluées, telle l'*Unité Centrale de Traitement*. Il est également possible de concevoir des *dispositifs d'interconnexions* pour permettre les échanges d'informations entre différentes unités. En résumé, la *couche micro-architecture* comprend tout le matériel chargé de l'exécution effective des instructions du langage machine.

Niveau 2 – architecture On parle également de couche *ISA* (*Instruction Set Architecture*, architecture au niveau du jeu d'instruction). Elle comporte tous les attributs d'un ordinateur qui sont visibles au programmeur lorsqu'il programme en langage machine.

Niveau 3 – système d'exploitation : Le *système d'exploitation* introduit un niveau d'abstraction entre le programmeur (ou plus généralement l'utilisateur d'une machine), et le matériel. Ce niveau d'abstraction supplémentaire permet de fournir au programmeur des mécanismes de gestion du matériel (mémoire, périphériques, exécution des programmes...) identiques sur de nombreuses architectures différentes.

Niveau 4 – couche langage d'assemblage Ce niveau fournit un langage intermédiaire permettant au programmeur de coder plus facilement en langage machine, appelé *langage d'assemblage*. En effet, les instructions du langage machine ne sont que des suites de bits, facilement exécutables par le processeur, mais illisibles pour le programmeur.

Niveau 5 – couche langage de programmation On trouve à ce niveau les *langages de programmation de haut niveau*, tel le C, le C++ ou le Caml. Ces langages fournissent un haut niveau d'abstraction, et permettent aux programmeurs de réaliser facilement des logiciels complexes. Les programmes codés dans ces langages de hauts niveau sont traduits vers les langages des couches 3 et 4 par des *compilateurs* (par exemple GCC).

Q.I.2) - (2 pt) Décrivez brièvement le modèle dit « de von Neumann ».

La structure des ordinateurs généralistes actuels suit encore le modèle dit de Von Neumann. Dans ce modèle, l'ordinateur se compose :

- d'une *mémoire centrale*, qui contient le programme et les données;
- d'une *Unité Centrale de Traitement* (UCT), qui exécute un programme contenu en mémoire centrale;
- d'une (ou plusieurs) *unité d'entrée-sortie* permettant l'échange d'informations avec l'environnement de l'UCT.

Un *système d'interconnexion* permet l'interaction entre ces unités.

Q.I.3) - (1 pt) Comment sont codées les instructions dans le modèle de von Neumann ?

Chaque instruction est stockée en mémoire sous la forme d'un mot binaire. Le mot codant une instruction se décompose en :

- un champ appelé *code-opération* ou *opcode*, qui spécifie l'opération qui doit être réalisée lors de l'exécution de l'instruction;
- éventuellement, un ou plusieurs champs *opérandes* qui définissent les emplacements où l'instruction doit lire ces sources et écrire son résultat.

Q.I.4) - (1 pt) Quels sont les rôles joués par le registre d'instruction **IR** et par le compteur de programme **PC** ?

L'Unité de Contrôle (UC) se charge de gérer l'exécution des instructions d'un programme. Pour cela, elle contient deux registres importants :

- Le *registre d'instruction* **IR**, qui contient d'instruction en cours d'exécution.
- Le *compteur de programme* **PC**, qui contient l'adresse en mémoire centrale de la prochaine instruction à exécuter.

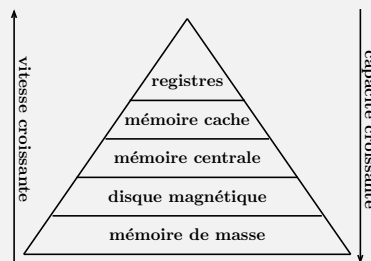
Ces registres ne sont pas accessibles au programmeur.

Q.I.5) - (1 pt) Caractérisez une mémoire à accès séquentiel. Que dire du temps d'accès à une telle mémoire ?

Une mémoire à *accès séquentiel* est organisée sous forme d'enregistrements, chacun possédant un identifiant unique qui constitue son adresse. La mémoire est dotée d'un dispositif mécanique de lecture-écriture : pour accéder à une certaine adresse, le dispositif de lecture-écriture passe en revue les enregistrements du support, et s'arrête sur celui recherché. Les temps d'accès sont donc très variables, et dépendent des accès précédents.

Q.I.6) - (2 pt) Expliquez ce qu'est une hiérarchie mémoire : quel est le but recherché lors de la mise en place d'une telle hiérarchie.

Pour obtenir une mémoire centrale de grande capacité et à faible coût, on est obligé d'en diminuer les performances. Mais pour obtenir des ordinateurs performants, il faut utiliser des mémoires très coûteuses et de faible capacité. . . La clé pour résoudre ce dilemme est de ne pas utiliser un seul type de mémoire, mais d'employer toute une *hiérarchie mémoire*.



Les mémoires rapides, de faibles capacité et très coûteuses sont secondées par des mémoire plus lentes, mais de plus grande capacité et à faible coût. On parvient ainsi à simuler une mémoire centrale :

- de grande capacité,
- présentant de bonnes performances en moyenne,
- pour un coût raisonnable.

De plus, le temps d'accès pour la mémoire RAM est de l'ordre de 10 fois plus long que pour les registres. Les concepteurs essaient de compenser cette lenteur relative en introduisant une mémoire rapide entre les registres de l'UCT et la mémoire centrale. Il s'agit de la *mémoire cache*, qui sert de zone de stockage temporaire des données : les mots les plus souvent utilisés y sont conservés.

Q.I.7) - (2 pt) On considère une mémoire de 32 Kio. Sur combien de bits devront être codées en binaire les adresses si on suppose :

7(a) - que cette mémoire est adressée par octets ?

7(b) - que cette mémoire est adressée par mots de 32 bits ?

Justifiez vos réponses.

$$32 \text{ Kio} = 2^5 \times 2^{10} \text{ o} = 2^{15} \text{ o}.$$

7(a) - adressée par octets : $(2^{15} \text{ o}) / (1 \text{ o})$ donne 2^{15} adresses, donc 15 bits par adresses.

7(b) - adressée par mots de 16 bits : $(2^{15} \text{ o}) / (4 \text{ o})$ donne 2^{13} adresses, donc 11 bits par adresses.

Q.I.8) - (3 pts) On considère à nouveau une mémoire de 32 Kio, et on suppose ici qu'elle est adressable par octet. On souhaite adjoindre à cette mémoire une petite mémoire cache à correspondance directe de 8 entrées pouvant stocker chacune 32 o de données. Comme dans le cours, on va supposer que les adresses en mémoire centrale sont découpées comme suit (bits de poids faibles à droite) :

INDICATEUR	LIGNE	OCTET
------------	-------	-------

8(a) - En combien de lignes de cache se décompose la mémoire centrale ? Combien de bits sont nécessaires pour identifier une ligne de cache ?

8(b) - A quoi sert le champ OCTET ? Quelle est sa taille en nombre de bits dans l'exemple ?

8(c) - A quoi sert le champ LIGNE ? Quelle est sa taille en nombre de bits dans l'exemple ?

8(d) - A quoi sert le champ INDICATEUR ? Quelle est sa taille en nombre de bits dans l'exemple ?

Justifiez vos réponses.

8(a) - $(32 \text{ Kio}) / (32 \text{ o}) = 2^{10}$, d'où 2^{10} lignes de cache. Il faut donc 10 bits pour identifier une ligne de cache.

8(b) - Le champ OCTET : permet d'identifier l'un des octets d'une ligne de cache, ou d'une entrée dans le cache. Comme on a 32 o par ligne de cache, il faut un champ OCTET sur 5 bits.

8(c) - Le champ LIGNE détermine dans quelle entrée du cache sera rangée la ligne de cache à laquelle l'adresse appartient. Ici, on a 8 entrées dans le cache, donc il faut un champ LIGNE sur 3 bits.

8(d) - Le mot formé par la concaténation des mots INDICATEUR et LIGNE identifie une ligne de cache dans la mémoire centrale. Il faut 10 bits pour identifier une ligne de cache, et 3 bits pour le champ ligne : le champ INDICATEUR comporte donc 7 bits.

II Questions à réponses brèves (5 pts)

Il vous est demandé de répondre aux questions suivantes en deux lignes au plus.

Q.II.1) - Quelle capacité mémoire, exprimée en nombre d'octets, représente 128 Tio ?

$Ki = 2^{10}$, $Mi = 2^{20}$, $Gi = 2^{30}$, $Ti = 2^{40}$: $128 \text{ Tio} = 2^{47}$ octets.

Q.II.2) - Quel est l'ordre de grandeur de la taille des transistors gravés sur les processeurs actuels : le pm, le nm, ou le μm ?

Le nm (nano-mètre).

Q.II.3) - Combien d'adresses différentes peut-on coder sur m bits ? Quelles est dans ce cas la plus petite adresse, et quelle est la plus grande ?

2^m adresses, numérotées de 0 à $2^m - 1$.

Q.II.4) - Comment est appelée la séquence de tâches permettant le traitement d'une instruction par l'UCT ?

Le cycle d'instruction.

Le bus, le bus informatique.

Q.II.5) - Si un évènement périodique se produit à la fréquence de 1 GHz, quelle est sa période ?

Si la fréquence est $f = 1 \text{ GHz} = 10^9 \text{ Hz}$, la période est $T = 1/f = 1/10^9 \text{ s} = 10^{-9} \text{ s} = 1 \text{ ns}$.

Q.II.6) - Expliquez ce qui est généralement appelé la « loi de Moore ».

En 1965, Gordon Moore (l'un des fondateurs d'Intel), observait que le nombre de transistors que l'on pouvait intégrer sur puce, en choisissant la technologie la plus économique, doublait environ tous les 18 mois. Cette observation est restée sous le nom de *loi de Moore*.

Q.II.7) - Donnez le nom de deux micro-architectures commerciales citées en cours.

L'Intel Pentium, l'Intel 8080, le Motorola 68000, le PowerPC d'IBM, l'Intel Itanium. . .

Q.II.8) - Donnez le nom de deux architectures citées en cours.

x86, IA-32, IA-64, POWER. . .

Q.II.9) - Enoncez le « principe de localité temporelle ».

Q.II.10) - Un bus système présente une bande passante de 4 Gio/s. Combien de temps faut-il pour que 128 Mio transitent sur ce bus ?

$(128 \text{ Mio}) / (4 \text{ Gio/s}) = (2^7 \times 2^{20} \text{ o}) / (4 \times 2^{30} \text{ o/s}) = 2^{-5} \text{ s} = \frac{1}{32} \text{ s}$.