Université Gustave Eiffel

ESIEE PARIS

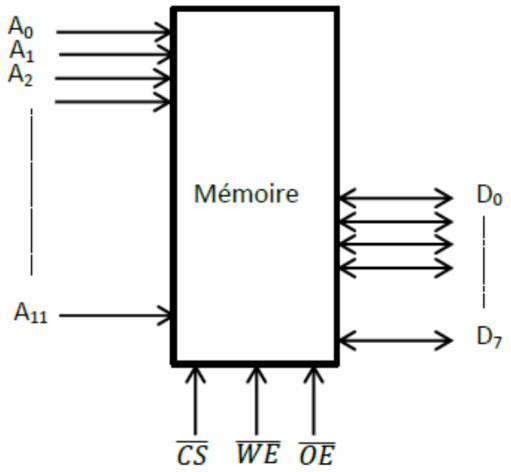
Architecture des ordinateurs
TD 2 - La mémoire centrale

Halim Djerroud

révision 1.0

Exercice 1 :

La mémoire d'un ordinateur est constituée d'un assemblage de plusieurs circuits mémoires, comme celui représenté sur la figure ci-après :



Les entrées A_i codent l'adresse d'un mot mémoire. Les entrées/sorties D_j communiquent avec le bus de données (écriture ou lecture d'un mot en mémoire). Ce boîtier a 3 entrées de commande CS, WE, et OE, actives en inverse.

Lors d'une opération de lecture, le rôle de signal CS (chip select) est de sélectionner un des boîtiers : pour un boîtier donné, cette entrée autorise la lecture ou l'écriture. Dans ce cas, WE (write enable) provoque l'écriture, tandis qu'OE (output enable) provoque la lecture. La taille d'une mémoire est exprimée en Kilos (1 $Kilo$, noté $1K = 2^{10} = 1024_{(10)}$).

— Déterminer la taille des mots mémoire et la capacité de ce boîtier en Ko.
— En assemblant des boîtiers de ce type : comment réaliser un espace mémoire de $4K$ mots de $16bits$
— Quelle est la taille de bus d'adresse nécessaire afin de couvrir un espace mémoire de $8K$ mots de $16bits$ si on suppose que le bus des données est de taille de $16bits$.

Exercice 2 :

1) On souhaite réaliser une mémoire de 8 GO avec des mots de 32 Bits.
— Quelle est la taille du bus d'adresse et la taille du bus de données nécessaires ?
— Quelle est la taille du bus d'adresse nécessaire si on souhaite augmenter la capacité de la mémoire à 16 GO ?
— Donnez deux signaux faisant partie du bus de contrôle de la mémoire. Pour chaque signal, indiquez son sens (depuis ou vers la mémoire) et son rôle ?

Exercice 3 :

Soit un ordinateur dont les mots mémoire sont composés de 32 bits. Cet ordinateur dispose de 4 Mo de mémoire.

- Un entier étant codé sur un mot, combien de mots cet ordinateur peut-il mémoriser simultanément ?
- Quelle est la plus grande valeur entière (décimale) que cet ordinateur peut mémoriser, cette valeur étant représentée par son codage binaire pur ?
- Donner un ordre de grandeur du nombre de chiffres en codage décimal.

Exercice 1 : Représentations big endian et little endian

On considère deux ordinateurs dotés d'une mémoire centrale dont les mots mémoire font 4 octets. Le processeur d'un des ordinateurs adopte la représentation big endian, celui de l'autre ordinateur la représentation little endian.

- Rappelez l'organisation des octets d'un mot mémoire dans chacune de ces représentations. Rappelez également les contraintes d'alignement qui existent quand on accède à un mot mémoire, en lecture ou en écriture.

On suppose qu'on stocke en mémoire un enregistrement d'une base de données décrivant une personne. L'enregistrement donne, dans cet ordre : le nom de la personne (chaine de caractère terminée par '\0'), son âge (entier sur un mot), le numéro de service dans lequel elle travaille (entier sur un mot).

- Donnez, sur chacun des deux ordinateurs, la représentation en mémoire de l'enregistrement décrivant « Luc Duval », 31 ans, travaillant dans le service 260.

On effectue un transfert de cet enregistrement de l'ordinateur big endian vers l'ordinateur little endian.

- On suppose que le transfert est fait de façon brute, i.e. octet par octet, du premier au dernier. Comment est interprété l'enregistrement reçu sur l'ordinateur little endian ?
- L'ordre des octets dans un mot étant inversé sur les deux ordinateurs, on décide maintenant d'inverser, en réception, les octets de chaque mot reçu. Quel est le résultat obtenu ?
- Quelle solution proposer pour résoudre ce problème, sachant par ailleurs qu'il est difficile de savoir quel est le type (little / big endian) de l'ordinateur avec lequel on correspond ?
- Pourquoi le problème étudié dans cet exercice se manifeste rarement quand on fait un transfert de données brut entre deux machines d'un même cluster ?

Exercice 2 : TD sur machine

1. Écrire une programme en C qui permet de déclarer variables suivante : `char a, shor int b, int c , long int d , long long e, float f, double g, long double h`
   - Pour chaque variable afficher sa taille en mémoire
   - Pour chaque variable afficher son adresse
   - Déplacer les variables dans la pile (variables locales) et afficher de nouveau leurs adresses, que constatez vous ? et pourquoi ?
2. Écrire un programme qui permet d'afficher l'octet du poids fort et l'octet du poids faible d'un entier.
3. Écrire un programme qui permet de vérifier si votre machine adopte la représentation Big Endian ou Little Endian.