

Architecture des ordinateurs

Introduction

Définitions

- ▶ Qu'est-ce qu'un ordinateur ?
- ▶ Qu'est-ce qu'un programme ?
- ▶ Qu'est-ce qu'une information ?

Qu'est-ce qu'un ordinateur ?

- ▶ Une machine "calculateur universel"
- ▶ composée de plusieurs "unités" spécialisées (mémoire, processeur, ...)
- ▶ permettant de traiter (électroniquement) une ou des informations de départ (sous forme binaire)
- ▶ suivant une suite d'instructions nommée « programme »
- ▶ et de donner un résultat en sortie.
- ▶ interagit avec l'environnement à travers des périphériques (clavier, écran, ...)
- ▶ L'information en sortie peut servir à son tour
 - ▶ de donnée de départ pour un autre traitement ou
 - ▶ de programme pour un autre traitement
- ▶ Les résultats peuvent être fournis au fur et à mesure : effectuer une rétroaction

Qu'est-ce qu'un programme ?

- ▶ suite d'instructions stockées sur un support
 - ▶ destiné à être exécuté par un ordinateur particulier
 - ▶ rédigées dans un langage indépendant d'un quelconque contexte
 - ▶ permettant de traiter une ou des données de départ
 - ▶ en vu d'obtenir un ou des résultats
-
- ▶ un programme peut être considéré comme une donnée

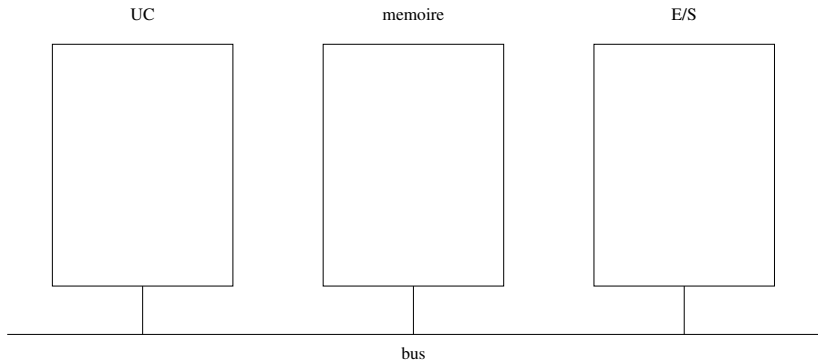
Qu'est-ce qu'une information ?

- ▶ donnée de n'importe quel type (texte, chiffres, signal électrique, onde sonore)
- ▶ renseignant sur l'état d'un objet, individu, système, etc...

- ▶ subjectivité des informations,
- ▶ elle dépendent de leur forme (audio, texte, etc...),
- ▶ du "point de vue" de celui qui a émis l'information
- ▶ des limitations du codage utilisé pour les conserver.
- ▶ implique des conventions de codage pour pouvoir les relire.

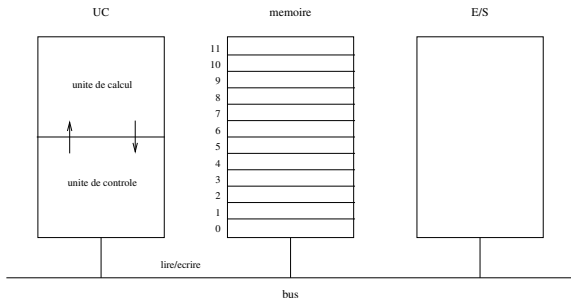
Organes d'un ordinateur

- ▶ Unité centrale UC ou CPU (Central Processing Unit)
- ▶ Mémoire
- ▶ Entrées/Sorties (E/S) : périphériques
- ▶ => canal de communication : un bus
une seule information circule à la fois



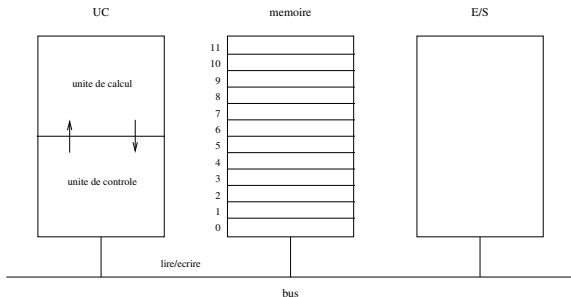
Memoire

- ▶ cases numérotées : **mots** mémoire
- ▶ numéros des cases : **adresses** mémoire
- ▶ Valeur stockée : n'importe quoi, nécessite une convention de codage (1000001 => 65 ou A ou ...)
- ▶ opérations : écrire ou lire
- ▶ stocke les programmes aussi bien que les données : caractéristique essentielle de nos ordinateurs



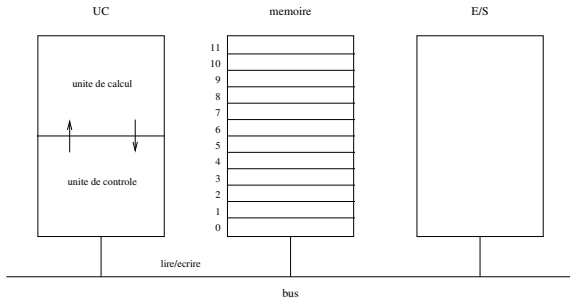
UC / CPU

- ▶ l'unité de calcul (UAL : unité arithmétique et logique)
 - ▶ Opérations simples : addition, ...
 - ▶ Ensemble d'opérations défini par le constructeur de l'UC.
- ▶ l'unité de commande (appelée aussi unité de contrôle).
 - ▶ lire une instruction du programme en mémoire
 - ▶ fournir une l'opération et un/des opérande(s)
 - ▶ la faire réaliser par l'unité de calcul
 - ▶ passer à l'instruction suivante



Entrées/Sorties

- ▶ E/S, Input/Output, IO
 - ▶ Lecture seule : clavier, ...
 - ▶ Écriture seule : l'écran, ...
 - ▶ Les deux : l'adaptateur réseau, ...



Un algorithme

- ▶ racine carré r d'un nombre n selon l'algorithme dit de *Newton-Raphson*.

début

$r \leftarrow 1$

 répéter

$r \leftarrow (r + n / r) / 2$

fin

- ▶ E/S : lire la valeur de n , écrire les valeurs successives de r
- ▶ arrêter le calcul : boucler un nombre B de fois ou comparer n avec r^2

Programme traduit pour l'ordinateur 1/2

- ▶ Travail d'un **interpréteur** ou **compilateur**.
- ▶ L'ordinateur ne peut effectuer que des opérations élémentaires : 1 opérateur et 2 opérandes

```
r <- 1
```

```
répéter
```

```
  r <- (r + n / r) / 2
```

- (1) placer 1 dans r
- (2) diviser n par r
- (3) ajouter r et le résultat du calcul précédent
- (4) diviser le résultat du calcul précédent par 2
- (5) placer le résultat dans r
- (6) recommencer avec l'instruction 2

Programme traduit pour l'ordinateur 2/2

- ▶ "Résultat du calcul précédent" => traduire en adresse d'un mot mémoire
- ▶ Prenons les mots mémoire t1, t2, t3.

- (a) placer 1 dans r
- (b) diviser n par r, placer le résultat dans t1
- (c) additionner r et t1, placer le résultat dans t2
- (d) placer 2 dans t3
- (e) diviser t2 par t3, placer le résultat dans r
- (f) recommencer avec l'instruction b

Instructions élémentaires

- ▶ (a) et (d) : placer une valeur dans un mot
- ▶ (b) et (e) : diviser le contenu d'un mot par celui d'un autre mot, placer le résultat dans un troisième mot
- ▶ (c) : additionner le contenu d'un mot par celui d'un autre mot, placer le résultat dans un troisième mot
- ▶ recommencer avec telle instruction

(a) placer 1 dans r

(b) diviser n par r, placer le résultat dans t1

(c) additionner r et t1, placer le résultat dans t2

(d) placer 2 dans t3

(e) diviser t2 par t3, placer le résultat dans r

(f) recommencer avec l'instruction b

Choix d'un codage arbitraire

opération	code opération	2ème mot	3ème mot	4ème mot
écrire le nombre x dans le mot adr_1	1	x	adr_1	?
écrire à adr_3 la somme des contenus de adr_1 et adr_2	4	adr_1	adr_2	adr_3
écrire à adr_3 le résultat de la division des contenus de adr_1 et adr_2	2	adr_1	adr_2	adr_3
continuer avec l'instruction qui se trouve à adr_1	3	adr_1	?	?

L'unité de contrôle PC

- ▶ Unité de contrôle : contrôle l'exécution du programme
 - ▶ Program counter (compteur ordinal) : registre, case mémoire d'accès très rapide
 - ▶ Exécute en boucle un programme câblé (un circuit / un automate)
- 1. charger le contenu du mot dont l'adresse est dans PC
- 2. ajouter 1 au contenu de PC
- 3. décoder l'instruction,
 - ▶ charger les opérandes si nécessaire,
 - ▶ faire effectuer l'opération par l'unité de calcul
 - ▶ stocker le résultat à l'adresse spécifiée dans l'instruction
 - ▶ => opérations encore plus élémentaires :
micro-instructions

L'unité de contrôle PC

- ▶ Unité de contrôle : contrôle l'exécution du programme
 - ▶ Program counter (compteur ordinal) : registre, case mémoire d'accès très rapide
 - ▶ Exécute en boucle un programme câblé (un circuit / un automate)
- 1. charger le contenu du mot dont l'adresse est dans PC (code opération)
- 2. ajouter 1 au contenu de PC
- 3. décoder l'instruction,
 - ▶ charger les opérandes si nécessaire,
 - ▶ faire effectuer l'opération par l'unité de calcul
 - ▶ stocker le résultat à l'adresse spécifiée dans l'instruction
 - ▶ => opérations encore plus élémentaires :
micro-instructions

Micro-instructions pour décoder une instruction

- ▶ Si le code de l'opération vaut 1
 1. lire le mot dont l'adresse est dans PC,
 2. et ajouter 1 à PC
 3. lire le mot dont l'adresse est dans PC,
 4. et ajouter 2 à PC
 5. placer la valeur lue en 1 au mot dont on a lu l'adresse en 3.

- ▶ Si le code de l'opération vaut 2
 1. lire le mot dont l'adresse est dans PC
 2. et ajouter 1 à PC
 3. lire le mot dont l'adresse est dans PC
 4. et ajouter 1 à PC
 5. placer les valeurs lues en 1 et 3 dans le diviseur
 6. lire le mot dont l'adresse est dans PC
 7. et ajouter 1 à PC
 8. placer la sortie du diviseur dans à l'adresse lue en 6 .

- ▶ Si le code de l'opération vaut 3
 1. lire le mot dont l'adresse est dans PC
 2. et placer sa valeur dans PC

Question

Réfléchir à ce qui se passe si :

- ▶ $PC = 2997$
- ▶ l'adresse 2997 contient la valeur 3
(branchement inconditionnel)
- ▶ l'adresse 2998 contient la valeur 2997