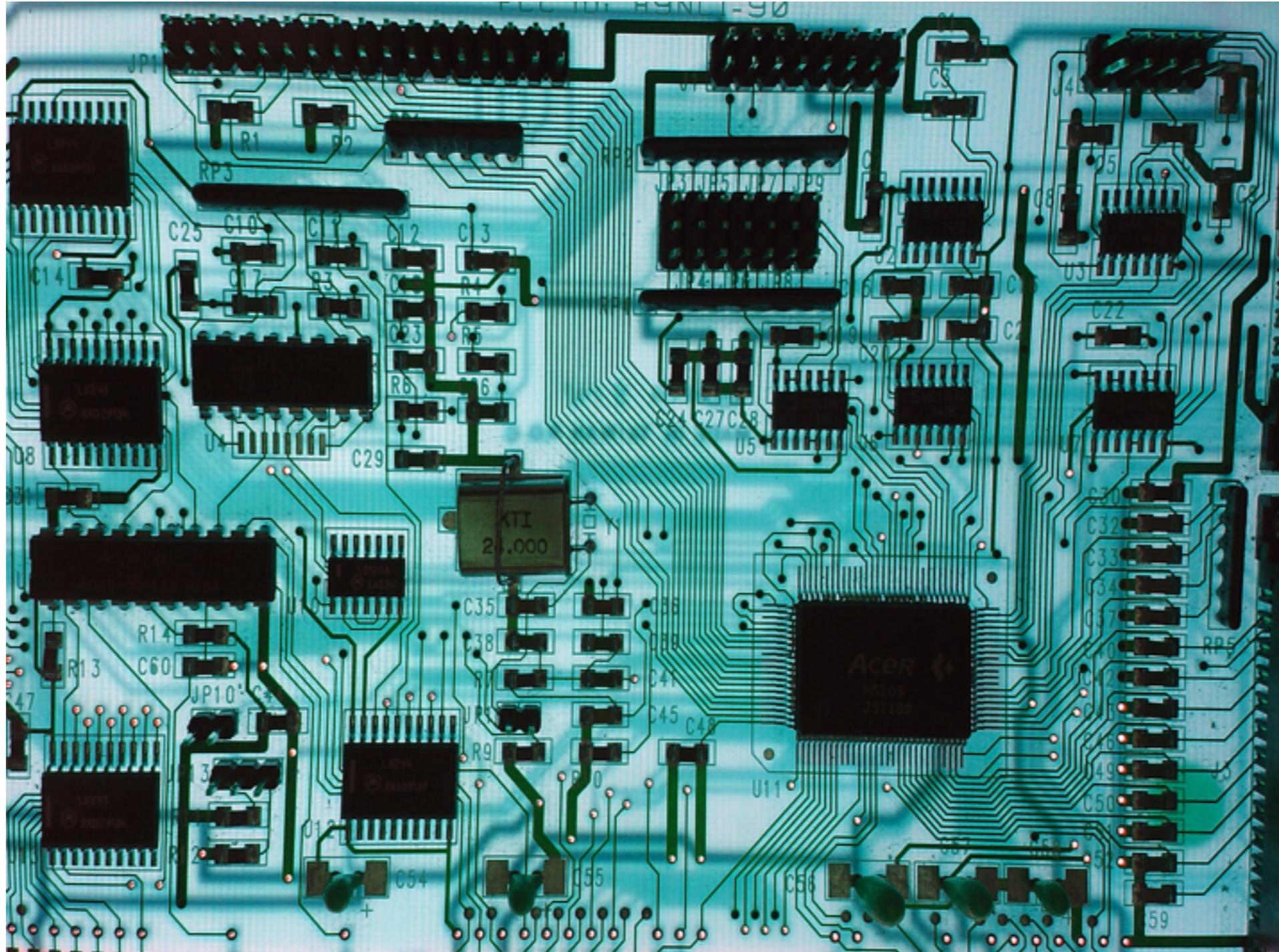


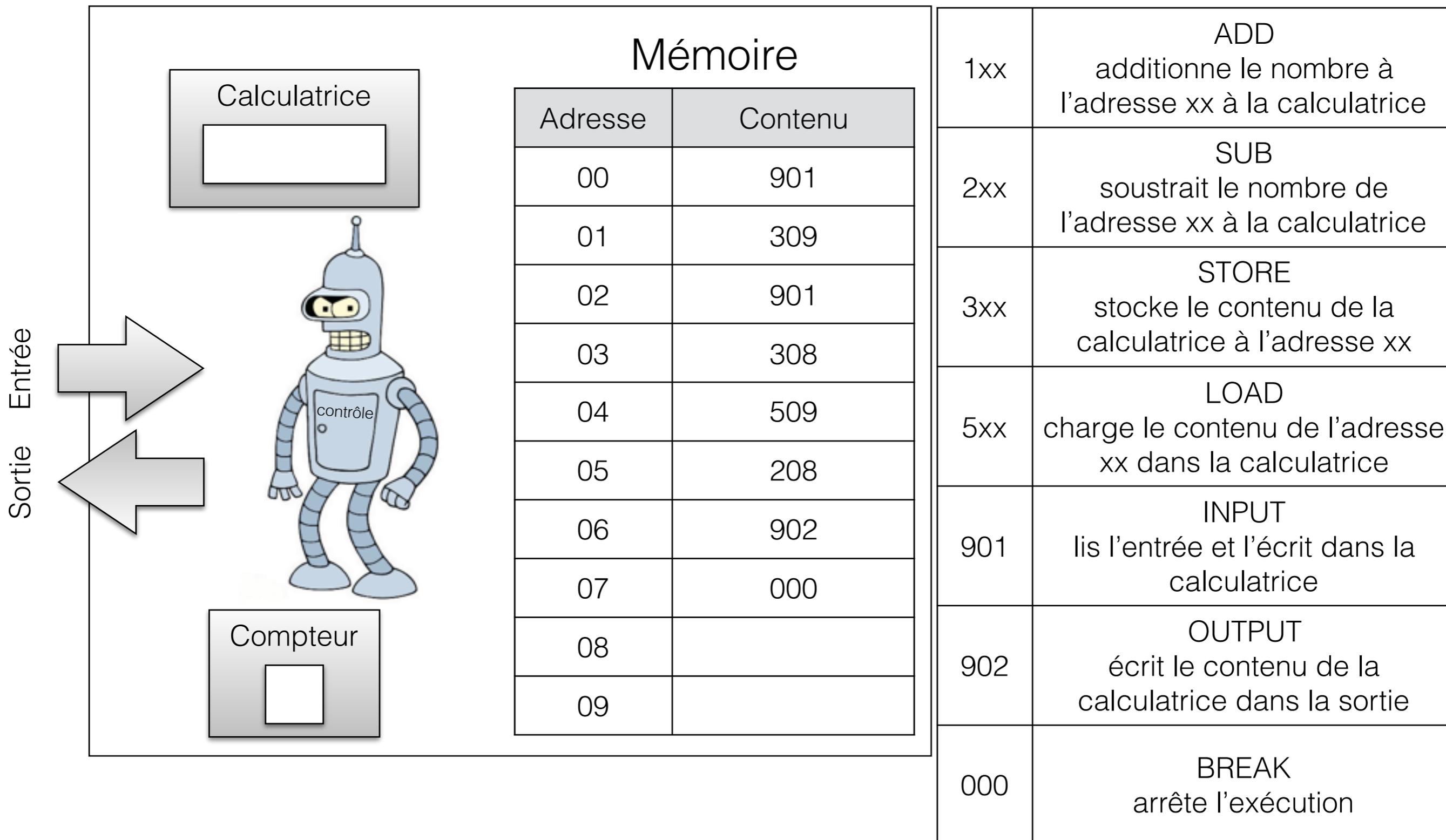
Composantes principales des ordinateurs



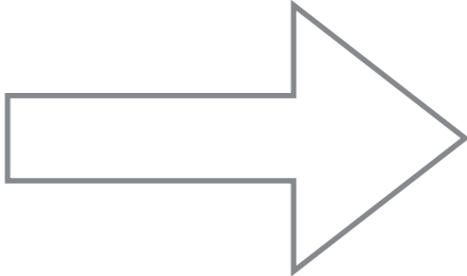
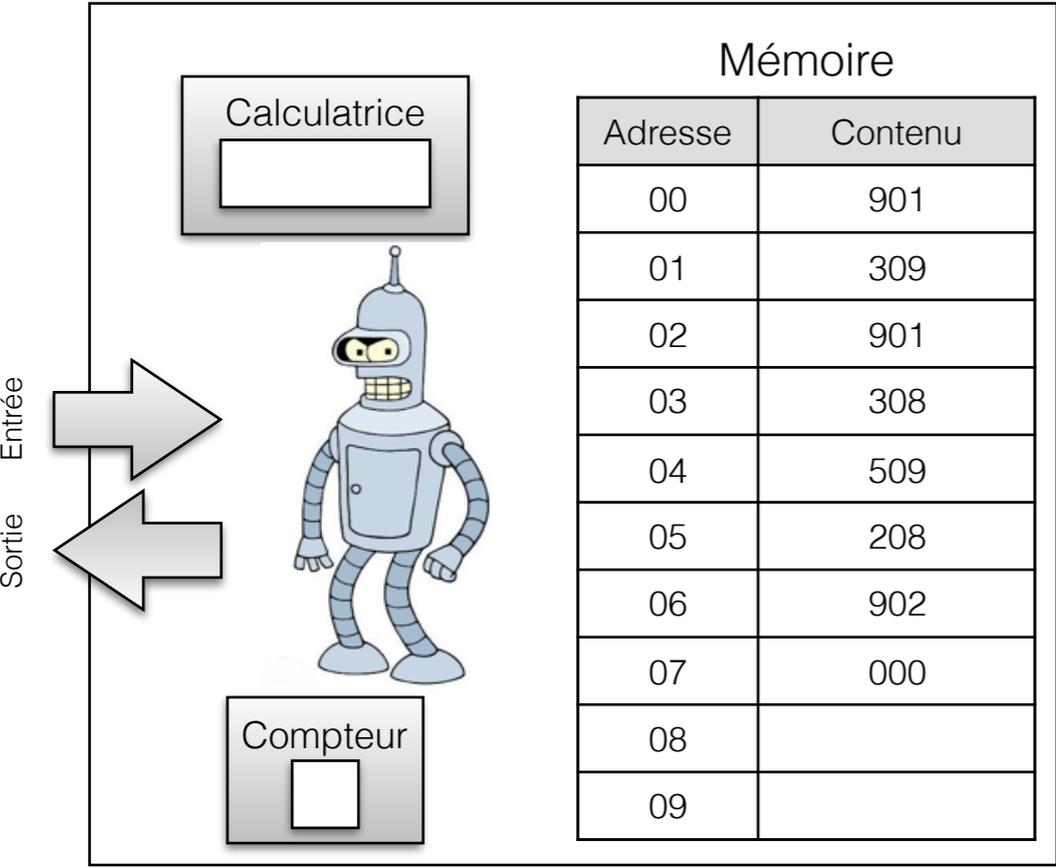
GIF-1001: Ordinateurs: Structure et Applications
Jean-François Lalonde, Hiver 2017

Rappel: ordinateur simplifié

Instructions (3 chiffres)



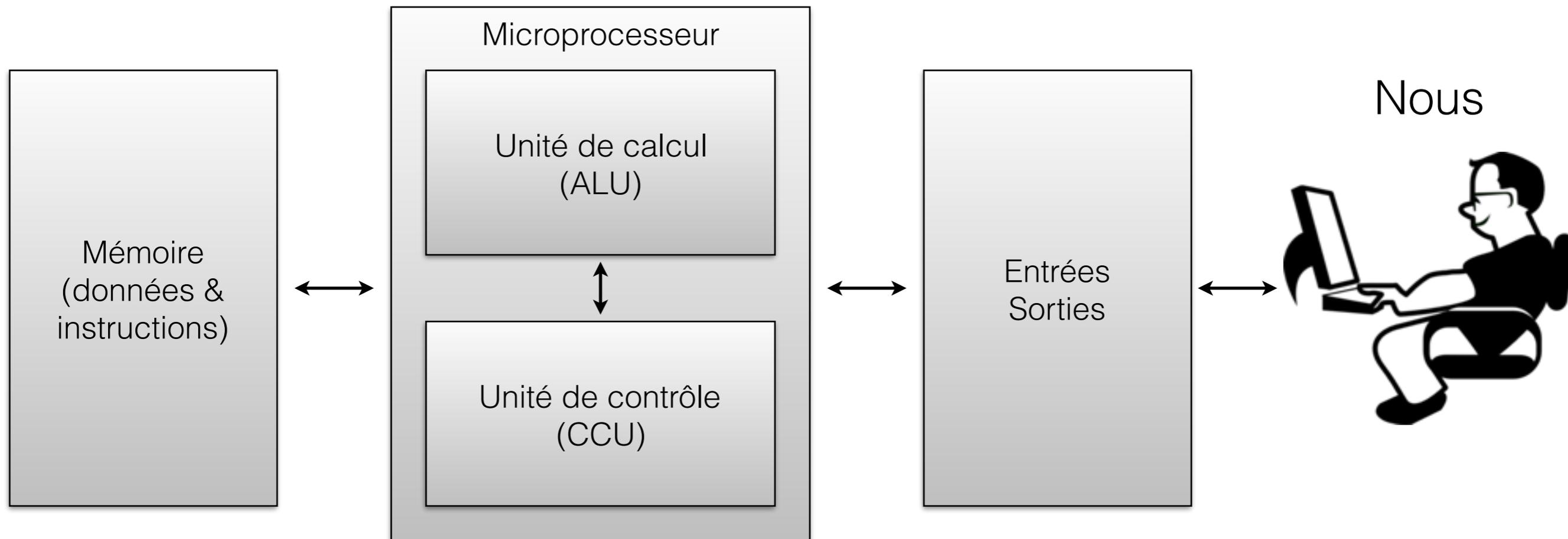
Aujourd'hui



?

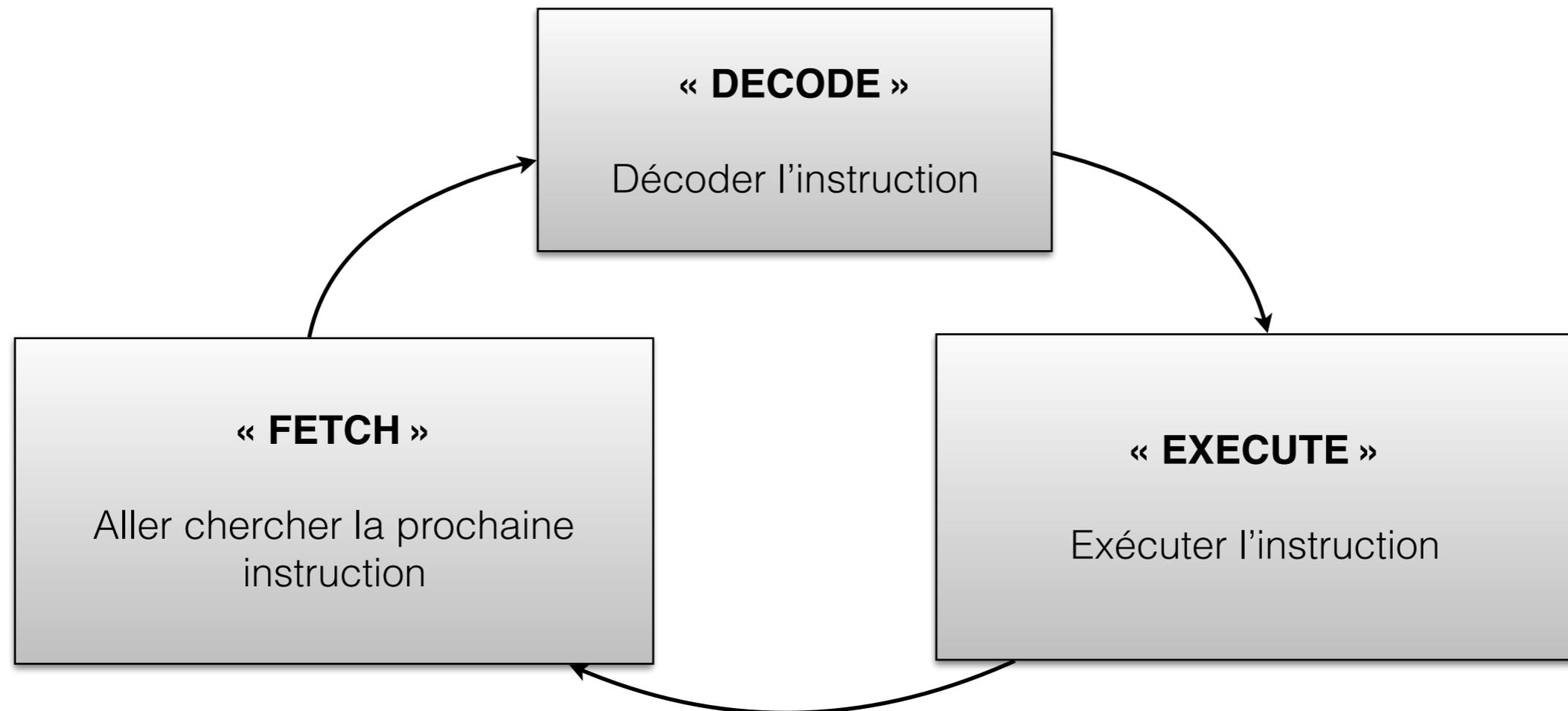
Architecture von Neumann

- Périphériques d'entrée-sorties
 - Nous permettent d'interagir avec l'ordinateur!
 - entrées: clavier, souris, lecteur DVD, etc.
 - sorties: écran, carte graphique, imprimante, etc.
 - entrées-sorties: disque dur, port ethernet, etc.



Cycle d'instructions

- Que fait le microprocesseur?
 - « Fetch »: aller chercher la prochaine instruction
 - « Decode »: décode l'instruction (détermine ce qu'il y a à faire)
 - « Execute »: exécuter l'instruction



Plan

- Nomenclature et définitions
 - mémoire
 - registres
 - bus
 - entrées-sorties
- Fonctionnement d'un ordinateur simplifié
 - mais un peu plus compliqué que la semaine dernière!

Mémoire

- Un emplacement qui contient des données
- Organisé par cellules individuelles. Chaque cellule possède:
 - une taille: combien de bits cette cellule peut stocker?
 - une adresse unique: permet de savoir à quel endroit on veut stocker l'information.

Analogie #1 (mémoire): boîte aux lettres

- chaque boîte à lettres contient:
 - un nombre fixe de lettres (contenu)
 - un numéro (adresse)



Analogie #2 (mémoire): dossiers

- chaque dossier contient:
 - un nombre fixe de feuilles (contenu)
 - un numéro pour pouvoir le retrouver (adresse)



Mémoire

- Un mot de mémoire se retrouve à chaque adresse. Les mots sont constitués de plusieurs bits
- On décrit une mémoire grâce à deux chiffres (indépendants):
 - le nombre d'adresses possibles
 - ici: $2^{16} = 65,536$ adresses
 - la taille des mots de la mémoire
 - ici: 8 bits = 1 octet)
- Les mémoires qui peuvent se lire et s'écrire possèdent au moins trois signaux de contrôle du microprocesseur:
 - Lecture de la mémoire;
 - Écriture de la mémoire;
 - Activation (Enable) de la mémoire.

mémoire de 2^{16} adresses

Adress	b ₇	b ₆	b ₅	b ₄	b ₃	b ₂	b ₁	b ₀
0x0000								
0x0001								
0x0002								
0x0003								
0x0004								
0x0005								
0x0006								
0x0007								
0x0008								
0x0009								
0x0010								
0x0011								
0x0012								
0x0013								
0x0014								
0x0015								
...								
...								
0xFFFF								

taille des mots = 8 bits = 1 octet

Questions

1. Une mémoire stocke des mots de 8 bits (1 octet) et possède 2^{16} cases mémoires. Quelle est la taille totale de la mémoire en kilo-octets (Ko)?
2. Une mémoire stocke des mots de 16 bits (2 octets) et nécessite 8 bits pour les adresser. Quelle est la taille totale de la mémoire en octets?
3. Une mémoire possède une taille totale de 32 Mo et peut stocker des mots de 32 bits.
 - a. Combien de bits a-t-on besoin pour représenter les adresses dans cette mémoire?
 - b. Quelles sont les adresses minimales et maximales de cette mémoire exprimées en hexadécimal?

Réponses

1. Une mémoire stocke des mots de 8 bits (1 octet) et possède 2^{16} cases mémoires. Quelle est la taille totale de la mémoire en kilo-octets (Ko)?

1. 64Ko

2. Une mémoire stocke des mots de 16 bits (2 octets) et nécessite 8 bits pour les adresser. Quelle est la taille totale de la mémoire en octets?

1. 512 octets

3. Une mémoire possède une taille totale de 32 Mo et peut stocker des mots de 32 bits.

a. Combien de bits a-t-on besoin pour représenter les adresses dans cette mémoire?

a. 23 bits

b. Quelles sont les adresses minimales et maximales de cette mémoire exprimées en hexadécimal?

a. minimale: 0x0, maximale: 0x7FFFFFFF

Réponses et démarche

- Question 1

- Taille d'un mot: 1 octet
- Nombre total de mots: 2^{16}
- Taille mémoire = $1 \times 2^{16} = 2^{16}$ octets = 2^6 Ko = 64 Ko.

- Question 2

- Taille d'un mot: 2 octets
- Nombre total de mots: $2^8 = 256$
- Taille mémoire = $2 \times 256 = 512$ octets

- Question 3-a)

- Taille mémoire: 32Mo
- Taille d'un mot: 32 bits = 4 octets
- Nombre total de mots = $32 \text{ Mo} / 4 \text{ o} = 32 \times 2^{20} / 4 = 2^{25} / 2^2 = 2^{23}$.
- Donc, nous avons besoin de $\log_2(2^{23}) = 23$ bits pour représenter les adresses.

- Question 3-b)

- Les adresses minimales et maximales sont:
- minimale: 0b000000000000000000000000 (23 bits) = 0x000000
- maximale: 0b111111111111111111111111 (23 bits) = 0x7FFFFFF

Types de mémoires

- Les mémoires peuvent être:
 - volatiles: perdent leur contenu lorsqu'elles perdent leur alimentation;
 - ou non-volatiles: conservent leur contenu même sans alimentation.
- Les mémoires volatiles peuvent être:
 - statiques: n'ont pas besoin d'être lues pour conserver leurs valeurs
 - dynamiques: nécessitent un rafraîchissement de leur données de façon périodique. Si les données d'une mémoire dynamique ne sont pas "lues" régulièrement, elles s'effacent.
- Les mémoires
 - ROM: ne peuvent pas être écrites (Read Only Memory)
 - RAM: peuvent être écrites (Random Access Memory);
- Les noms sont donnés aux mémoires en fonction de ces caractéristiques. Par exemple, SRAM est de la RAM Statique.

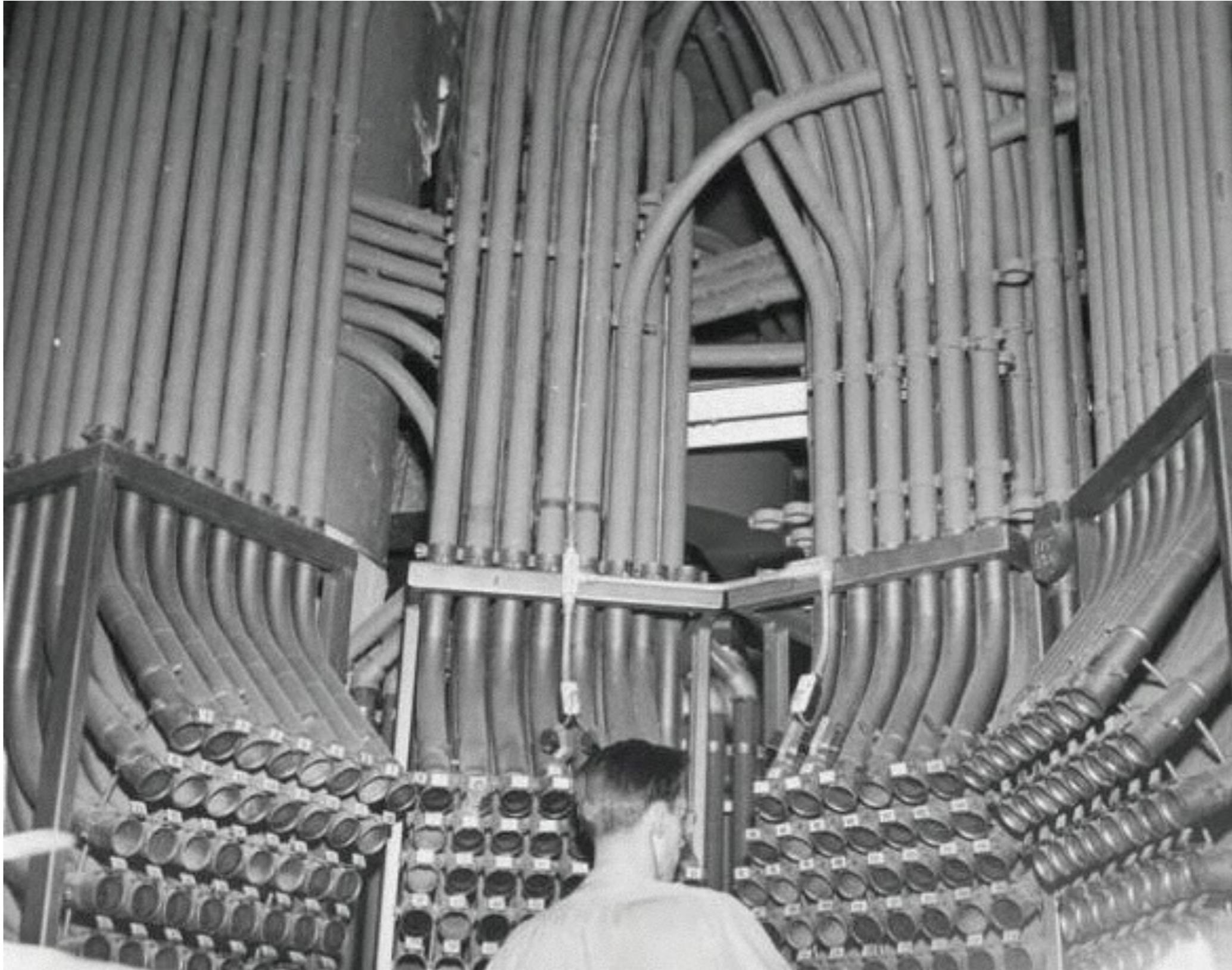
Types de mémoires

- Pour le moment, les deux types de mémoires qui nous intéressent sont:
 - ROM: ne peuvent **pas** être écrites (Read Only Memory)
 - doivent être écrites au préalable, une fois écrite on ne peut plus modifier leur contenu!
 - RAM: peuvent être écrites (Random Access Memory);

Entrées-sorties

- Un emplacement où on peut prendre ou mettre des données
- La mémoire peut être vue comme une sorte d'entrée/sortie pour le processeur

Analogie #3 (entrées-sorties): un tube



Défis des entrées-sorties

- L'information arrive à des moment surprenants
- L'information arrive à des débits très rapides
- L'information doit être envoyée à des débits très rapides

Registres

- L'unité de contrôle à l'intérieur du microprocesseur contient des emplacements mémoire très rapides dédiés à des fonctions particulières: les registres
- Registres généraux:
 - permettent d'entreposer temporairement des données pour exécuter les programmes
- Registres particuliers:
 - PC: "Program Counter", stocke l'adresse (en mémoire) de la prochaine instruction à exécuter
 - IR: "Instruction Register", stocke l'instruction à décoder, puis à exécuter
 - ACC: "accumulator", stocke les résultats temporaires pour les calculs effectués par l'ALU
 - MAR: "Memory Address Register", détermine la prochaine adresse lue en mémoire
 - MDR: "Memory Data Register", emmagasine temporairement le contenu de la mémoire
 - Statut: "Status Register(s)", stocke des drapeaux (carry, overflow, zero, erreurs, etc.)

Bus

- Un bus est un groupe de lignes électriques qui relie le CPU aux autres composantes. Chaque ligne peut transférer un bit d'information à la fois.
- Il y a trois types de bus: données, adresse, et contrôle
- Le **bus de données** permet le transfert des données. Les données peuvent circuler dans les deux sens, mais elles circulent dans *un seul sens à la fois*.
 - La taille du bus de données (le nombre de lignes) détermine la grandeur maximale des mots pouvant être transférés d'un coup
- Le **bus d'adresse** indique l'emplacement de la mémoire ou des périphériques visé par la transaction sur le bus. Il est contrôlé par le microprocesseur.
 - La taille du bus d'adresse (le nombre de lignes) détermine la quantité maximum de mémoire ou d'entrées-sorties que le CPU peut utiliser
- Le **bus de contrôle** contrôle l'utilisation des bus de données et d'adresse. Par exemple, il permet de gérer la direction des données sur le bus des données (lecture, écriture).
 - Le bus de contrôle a aussi une horloge, qui détermine la vitesse à laquelle les données peuvent être transférées et qui synchronise les opérations

Analogie #4 (bus): autoroutes

- Sauf que les voitures...
 - démarrent et arrivent à destination en même temps
 - ne peuvent circuler que dans un sens à la fois

