

# L'ordinateur

Prof d'info

Lycée Thiers

1 L'ordinateur

2 La mémoire

- Architecture des ordinateurs [ce cours](#) de Stéphane Crozat, Marc Damie

- Architecture des ordinateurs [ce cours](#) de Stéphane Crozat, Marc Damie
- Registres : [Wikipedia](#)

# 1 L'ordinateur

## 2 La mémoire

# Modèle de Von Neuman

- Les ordinateurs ont une conception héritée du modèle théorique appelé *Architecture Von Neuman* (milieu des années 40).

# Modèle de Von Neuman

- Les ordinateurs ont une conception hérité du modèle théorique appelé *Architecture Von Neuman* (milieu des années 40).
- Idée principale : les programmes à exécuter sont stockés au même endroit que les données qu'ils manipulent.

# Architecture Von Neuman

Il y a 3 composantes dans ce modèle :

- Une *Unité de Calcul et de Traitement* qui réalise les opérations demandées par le programmeur. Stocke une très petite quantité d'information à la fois, en binaire et la traite très vite.



# Architecture Von Neuman

Il y a 3 composantes dans ce modèle :

- Une *Unité de Calcul et de Traitement* qui réalise les opérations demandées par le programmeur. Stocke une très petite quantité d'information à la fois, en binaire et la traite très vite.
- Une *mémoire vive* (dite aussi RAM -Random Access Memory-), qui permet de stocker une plus grosse quantité d'informations. C'est là que sont déposés les résultats des calculs effectués et les nouvelles données à traiter.

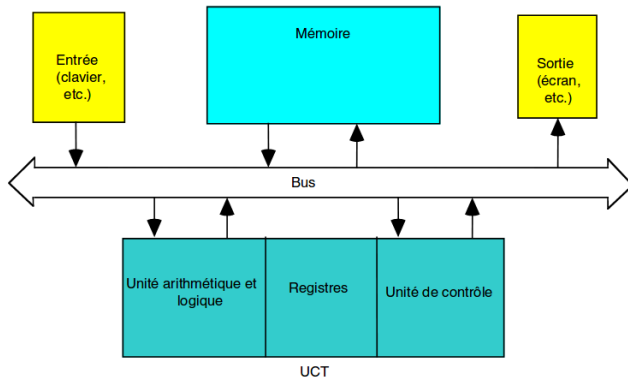
# Architecture Von Neuman

Il y a 3 composantes dans ce modèle :

- Une *Unité de Calcul et de Traitement* qui réalise les opérations demandées par le programmeur. Stocke une très petite quantité d'information à la fois, en binaire et la traite très vite.
- Une *mémoire vive* (dite aussi RAM -Random Access Memory-), qui permet de stocker une plus grosse quantité d'informations. C'est là que sont déposés les résultats des calculs effectués et les nouvelles données à traiter.
- Des *périphériques* d'entrées/sorties qui permettent à l'ordinateur d'interagir avec le monde (entrée : clavier, carte perforée etc. ; sortie : écran, imprimante, fax etc.)

# Schéma simplifié d'un ordinateur

FIGURE – Schéma général d'un ordinateur



# UCT, processeur, microprocesseur

- L'UCT est aussi appelée *processeur*.

# UCT, processeur, microprocesseur

- L'UCT est aussi appelée *processeur*.
- Lorsque les 3 composantes de l'UCT (Unité de contrôle, UAL, registres) sont rassemblées dans un seul circuit électronique, on parle de *microprocesseur*.  
Un bus interne (non représenté ici) permet la communication entre les différentes parties de l'UCT.

# UCT

L'Unité de Contrôle et de Traitement contient :

- L'*Unité de commande* (UC) : récupère en mémoire la prochaine opération à effectuer et les données sur lesquelles elle s'applique. Envoie le tout à l'UAL.

# UCT

L'Unité de Contrôle et de Traitement contient :

- L'*Unité de commande* (UC) : récupère en mémoire la prochaine opération à effectuer et les données sur lesquelles elle s'applique. Envoie le tout à l'UAL.
- L'UAL (Unité Arithmétique et Logique) : commandée par l'UC. Chargée de réaliser toutes les opérations de base : les opérations arithmétiques sur les nombres entiers (addition, multiplication, etc.) et des opérations booléennes (OR, AND, etc.).

# UCT

L'Unité de Contrôle et de Traitement contient :

- L'*Unité de commande* (UC) : récupère en mémoire la prochaine opération à effectuer et les données sur lesquelles elle s'applique. Envoie le tout à l'UAL.
- L'UAL (Unité Arithmétique et Logique) : commandée par l'UC. Chargée de réaliser toutes les opérations de base : les opérations arithmétiques sur les nombres entiers (addition, multiplication, etc.) et des opérations booléennes (OR, AND, etc.).
- Des *registres* : il s'agit de la mémoire du microprocesseur. Il stockent en particulier des adresses et des données utilisées lors de l'exécution d'un programme. Il en existe de différente sorte :



# UCT

L'Unité de Contrôle et de Traitement contient :

- L'*Unité de commande* (UC) : récupère en mémoire la prochaine opération à effectuer et les données sur lesquelles elle s'applique. Envoie le tout à l'UAL.
- L'UAL (Unité Arithmétique et Logique) : commandée par l'UC. Chargée de réaliser toutes les opérations de base : les opérations arithmétiques sur les nombres entiers (addition, multiplication, etc.) et des opérations booléennes (OR, AND, etc.).
- Des *registres* : il s'agit de la mémoire du microprocesseur. Il stockent en particulier des adresses et des données utilisées lors de l'exécution d'un programme. Il en existe de différente sorte :
  - Certains (les *registres généraux*) sont affectés à des opérations d'ordre général et peuvent être accessibles au programmeur (enfin, celui qui maîtrise l'*assembleur*) ;

# UCT

L'Unité de Contrôle et de Traitement contient :

- L'*Unité de commande* (UC) : récupère en mémoire la prochaine opération à effectuer et les données sur lesquelles elle s'applique. Envoie le tout à l'UAL.
- L'UAL (Unité Arithmétique et Logique) : commandée par l'UC. Chargée de réaliser toutes les opérations de base : les opérations arithmétiques sur les nombres entiers (addition, multiplication, etc.) et des opérations booléennes (OR, AND, etc.).
- Des *registres* : il s'agit de la mémoire du microprocesseur. Il stockent en particulier des adresses et des données utilisées lors de l'exécution d'un programme. Il en existe de différente sorte :
  - Certains (les *registres généraux*) sont affectés à des opérations d'ordre général et peuvent être accessibles au programmeur (enfin, celui qui maîtrise l'*assembleur*) ;
  - d'autres ne servent qu'à des opérations très spécifiques ;

# Bus

Pour info :

- Les *bus* sont des pistes électroniques parfois sous forme de câbles qui sont reliées au microprocesseur.

# Bus

Pour info :

- Les *bus* sont des pistes électroniques parfois sous forme de câbles qui sont reliées au microprocesseur.
- Le microprocesseur est relié à trois types de bus : le bus d'adresse, le bus de données et le bus de commande. Ces bus assurent la communication interne et externe du microprocesseur.

# Bus

Pour info :

- Les *bus* sont des pistes électroniques parfois sous forme de câbles qui sont reliées au microprocesseur.
- Le microprocesseur est relié à trois types de bus : le bus d'adresse, le bus de données et le bus de commande. Ces bus assurent la communication interne et externe du microprocesseur.
  - le *bus d'adresse* sert à envoyer un code d'adresse à la mémoire et à communiquer avec des dispositifs extérieurs.

# Bus

Pour info :

- Les *bus* sont des pistes électroniques parfois sous forme de câbles qui sont reliées au microprocesseur.
- Le microprocesseur est relié à trois types de bus : le bus d'adresse, le bus de données et le bus de commande. Ces bus assurent la communication interne et externe du microprocesseur.
  - le *bus d'adresse* sert à envoyer un code d'adresse à la mémoire et à communiquer avec des dispositifs extérieurs.
  - toutes les données entrantes et sortantes du processeur transitent par le *bus de données*.

# Bus

Pour info :

- Les *bus* sont des pistes électroniques parfois sous forme de câbles qui sont reliées au microprocesseur.
- Le microprocesseur est relié à trois types de bus : le bus d'adresse, le bus de données et le bus de commande. Ces bus assurent la communication interne et externe du microprocesseur.
  - le *bus d'adresse* sert à envoyer un code d'adresse à la mémoire et à communiquer avec des dispositifs extérieurs.
  - toutes les données entrantes et sortantes du processeur transitent par le *bus de données*.
  - le *bus de commande* sert à coordonner les échanges d'informations. Il veille à la synchronisation.

# Bus

Pour info :

- Les *bus* sont des pistes électroniques parfois sous forme de câbles qui sont reliées au microprocesseur.
- Le microprocesseur est relié à trois types de bus : le bus d'adresse, le bus de données et le bus de commande. Ces bus assurent la communication interne et externe du microprocesseur.
  - le *bus d'adresse* sert à envoyer un code d'adresse à la mémoire et à communiquer avec des dispositifs extérieurs.
  - toutes les données entrantes et sortantes du processeur transitent par le *bus de données*.
  - le *bus de commande* sert à coordonner les échanges d'informations. Il veille à la synchronisation.
    - Il introduit notamment un délai d'attente lorsque des informations sont envoyées/reçues vers/depuis un périphérique qui présente une vitesse de traitement réduite.



# Bus

Pour info :

- Les *bus* sont des pistes électroniques parfois sous forme de câbles qui sont reliées au microprocesseur.
- Le microprocesseur est relié à trois types de bus : le bus d'adresse, le bus de données et le bus de commande. Ces bus assurent la communication interne et externe du microprocesseur.
  - le *bus d'adresse* sert à envoyer un code d'adresse à la mémoire et à communiquer avec des dispositifs extérieurs.
  - toutes les données entrantes et sortantes du processeur transitent par le *bus de données*.
  - le *bus de commande* sert à coordonner les échanges d'informations. Il veille à la synchronisation.
    - Il introduit notamment un délai d'attente lorsque des informations sont envoyées/reçues vers/depuis un périphérique qui présente une vitesse de traitement réduite.
    - évite les conflits de bus lorsque 2 éléments cherchent à communiquer en même temps.

# UCT et microprocesseur

- La notion théorique d'UCT est matérialisée par un *microprocesseur*. Celui-ci contient, outre les registres, l'UAL et l'UC :

# UCT et microprocesseur

- La notion théorique d'UCT est matérialisée par un *microprocesseur*. Celui-ci contient, outre les registres, l'UAL et l'UC :
  - un bus interne (qui traverse les 3 parties),

# UCT et microprocesseur

- La notion théorique d'UCT est matérialisée par un *microprocesseur*. Celui-ci contient, outre les registres, l'UAL et l'UC :
  - un bus interne (qui traverse les 3 parties),
  - un *contrôleur de bus*,

# UCT et microprocesseur

- La notion théorique d'UCT est matérialisée par un *microprocesseur*. Celui-ci contient, outre les registres, l'UAL et l'UC :
  - un bus interne (qui traverse les 3 parties),
  - un *contrôleur de bus*,
  - une *horloge* (dans dont les « bips » servent à la synchronisation,

# UCT et microprocesseur

- La notion théorique d'UCT est matérialisée par un *microprocesseur*. Celui-ci contient, outre les registres, l'UAL et l'UC :
  - un bus interne (qui traverse les 3 parties),
  - un *contrôleur de bus*,
  - une *horloge* (dans dont les « bips » servent à la synchronisation,
  - un *séquenceur* (dans l'UC) décompose les instructions complexes en micro-commandes.

# UCT et microprocesseur

- La notion théorique d'UCT est matérialisée par un *microprocesseur*. Celui-ci contient, outre les registres, l'UAL et l'UC :
  - un bus interne (qui traverse les 3 parties),
  - un *contrôleur de bus*,
  - une *horloge* (dans dont les « bips » servent à la synchronisation,
  - un *séquenceur* (dans l'UC) décompose les instructions complexes en micro-commandes.
- Les microprocesseurs actuels intègrent d'autres composants comme une carte graphique ou encore la UVF (Unité de Calcul à Virgule flottante) ou FPU (*Floating Point Unity*).

# UCT et microprocesseur

- La notion théorique d'UCT est matérialisée par un *microprocesseur*. Celui-ci contient, outre les registres, l'UAL et l'UC :
  - un bus interne (qui traverse les 3 parties),
  - un *contrôleur de bus*,
  - une *horloge* (dans dont les « bips » servent à la synchronisation,
  - un *séquenceur* (dans l'UC) décompose les instructions complexes en micro-commandes.
- Les microprocesseurs actuels intègrent d'autres composants comme une carte graphique ou encore la UVF (Unité de Calcul à Virgule flottante) ou FPU (*Floating Point Unity*).
- Ils sont insérés sur la *carte mère*.



# Unité de contrôle

L'unité de contrôle (UC) est reliée aux autres composants de l'ordinateur par un bus interne.

- L'UC contient

# Unité de contrôle

L'unité de contrôle (UC) est reliée aux autres composants de l'ordinateur par un bus interne.

- L'UC contient
  - Deux registres pour la manipulation des instructions : le *compteur ordinal* CO, le *registre d'instruction* RI ;

# Unité de contrôle

L'unité de contrôle (UC) est reliée aux autres composants de l'ordinateur par un bus interne.

- L'UC contient
  - Deux registres pour la manipulation des instructions : le *compteur ordinal* CO, le *registre d'instruction* RI ;
  - le décodeur et le séquenceur ;

# Unité de contrôle

L'unité de contrôle (UC) est reliée aux autres composants de l'ordinateur par un bus interne.

- L'UC contient
  - Deux registres pour la manipulation des instructions : le *compteur ordinal* CO, le *registre d'instruction* RI ;
  - le décodeur et le séquenceur ;
  - le registre d'adresse RAD et le registre de données RDO permettant la communication avec les autres modules.

# Unité de contrôle

L'unité de contrôle (UC) est reliée aux autres composants de l'ordinateur par un bus interne.

- L'UC contient
  - Deux registres pour la manipulation des instructions : le *compteur ordinal* CO, le *registre d'instruction* RI ;
  - le décodeur et le séquenceur ;
  - le registre d'adresse RAD et le registre de données RDO permettant la communication avec les autres modules.
- Via le *bus de commande* l'UC permet la lecture/écriture depuis/vers la RAM.

## Contenu de l'UC

- Le compteur ordinal (CO) : registre d'adresse. Contient l'adresse de la prochaine instruction à réaliser. Il est mis à jour dès que l'instruction courante a fini de s'exécuter.

## Contenu de l'UC

- Le compteur ordinal (CO) : registre d'adresse. Contient l'adresse de la prochaine instruction à réaliser.
- le registre d'instruction (RI). Registre de données, contient l'instruction courante.

## Contenu de l'UC

- Le compteur ordinal (CO) : registre d'adresse. Contient l'adresse de la prochaine instruction à réaliser.
- le registre d'instruction (RI). Registre de données, contient l'instruction courante.
- Le *décodeur* : ensemble de circuits qui indentifient l'instruction contenue dans RI, puis indique au séquenceur la nature de cette instruction.



## Contenu de l'UC

- Le compteur ordinal (CO) : registre d'adresse. Contient l'adresse de la prochaine instruction à réaliser.
- le registre d'instruction (RI). Registre de données, contient l'instruction courante.
- Le *décodeur* : ensemble de circuits qui indentifient l'instruction contenue dans RI, puis indique au séquenceur la nature de cette instruction.
- Le *séquenceur* : Ensemble de circuits qui décomposent, rythmé par l'horloge, l' instruction complexe à exécuter (exemple : une addition) en *micro-commandes* successives exécutables par les circuits de l'UAL.

## Contenu de l'UC

- Le compteur ordinal (CO) : registre d'adresse. Contient l'adresse de la prochaine instruction à réaliser.
- le registre d'instruction (RI). Registre de données, contient l'instruction courante.
- Le *décodeur* : ensemble de circuits qui indentifient l'instruction contenue dans RI, puis indique au séquenceur la nature de cette instruction.
- Le *séquenceur* : Ensemble de circuits qui décomposent, rythmé par l'horloge, l' instruction complexe à exécuter (exemple : une addition) en *micro-commandes* successives exécutables par les circuits de l'UAL.
- Le *registre RAD* : registre d'adresse connecté au bus d'adresse. L'adresse contenu dans RAD est placée sur le bus d'adresse et devient la valeur d'entrée du *circuit de sélection*. Ce dernier va, à partir de cette entrée, sélectionner le mot mémoire correspondant.


## Contenu de l'UC

- Le compteur ordinal (CO) : registre d'adresse. Contient l'adresse de la prochaine instruction à réaliser.
- le registre d'instruction (RI). Registre de données, contient l'instruction courante.
- Le *décodeur* : ensemble de circuits qui indentifient l'instruction contenue dans RI, puis indique au séquenceur la nature de cette instruction.
- Le *séquenceur* : Ensemble de circuits qui décomposent, rythmé par l'horloge, l' instruction complexe à exécuter (exemple : une addition) en *micro-commandes* successives exécutables par les circuits de l'UAL.
- Le *registre RAD* : registre d'adresse connecté au bus d'adresse. L'adresse contenu dans RAD est placée sur le bus d'adresse et devient la valeur d'entrée du *circuit de sélection*. Ce dernier va, à partir de cette entrée, sélectionner le mot mémoire correspondant.
- Le *registre RDO* permet l'échange d'information (mot mémoire) entre la mémoire centrale et le processeur.

## Exécution d'une instruction par l'UC

Lorsque le processeur doit exécuter une instruction, il


- déplace l'adresse contenue dans le CO vers le RAD via le bus d'adresse interne et le circuit de sélection (une nouvelle adresse est alors placée dans le registre CO) ;

Le *chemin de données* est donc CO, RAD, commande de lecture, RDO, RI, décodage et séquençage. Charger ensuite les *opérandes* de l'instruction. 

## Exécution d'une instruction par l'UC

Lorsque le processeur doit exécuter une instruction, il

- déplace l'adresse contenue dans le CO vers le RAD via le bus d'adresse interne et le circuit de sélection (une nouvelle adresse est alors placée dans le registre CO) ;
- déclenche une commande de lecture mémoire via le bus de commande Processeur/Mémoire ;

Le *chemin de données* est donc CO, RAD, commande de lecture, RDO, RI, décodage et séquençage. Charger ensuite les *opérandes* de l'instruction. 

## Exécution d'une instruction par l'UC

Lorsque le processeur doit exécuter une instruction, il


- déplace l'adresse contenue dans le CO vers le RAD via le bus d'adresse interne et le circuit de sélection (une nouvelle adresse est alors placée dans le registre CO) ;
- déclenche une commande de lecture mémoire via le bus de commande Processeur/Mémoire ;
- reçoit dans le registre RDO (via bus de données Processeur/Mémoire) le mot mémoire de l'instruction ;

Le *chemin de données* est donc CO, RAD, commande de lecture, RDO, RI, décodage et séquençage. Charger ensuite les *opérandes* de l'instruction. ↻ 🔍 🔗

## Exécution d'une instruction par l'UC

Lorsque le processeur doit exécuter une instruction, il


- déplace l'adresse contenue dans le CO vers le RAD via le bus d'adresse interne et le circuit de sélection (une nouvelle adresse est alors placée dans le registre CO) ;
- déclenche une commande de lecture mémoire via le bus de commande Processeur/Mémoire ;
- reçoit dans le registre RDO (via bus de données Processeur/Mémoire) le mot mémoire de l'instruction ;
- Place le contenu du RDO dans le registre d'instruction RI (via bus interne) ;

Le *chemin de données* est donc CO, RAD, commande de lecture, RDO, RI, décodage et séquençage. Charger ensuite les *opérandes* de l'instruction. 

## Exécution d'une instruction par l'UC

Lorsque le processeur doit exécuter une instruction, il

- déplace l'adresse contenue dans le CO vers le RAD via le bus d'adresse interne et le circuit de sélection (une nouvelle adresse est alors placée dans le registre CO) ;
- déclenche une commande de lecture mémoire via le bus de commande Processeur/Mémoire ;
- reçoit dans le registre RDO (via bus de données Processeur/Mémoire) le mot mémoire de l'instruction ;
- Place le contenu du RDO dans le registre d'instruction RI (via bus interne) ;
- Le décodeur analyse l'instruction stockée dans RI, la reconnaît et indique au séquenceur sa nature ;


Le *chemin de données* est donc CO, RAD, commande de lecture, RDO, RI, décodage et séquençage. Charger ensuite les *opérandes* de l'instruction. 



## Exécution d'une instruction par l'UC

Lorsque le processeur doit exécuter une instruction, il

- déplace l'adresse contenue dans le CO vers le RAD via le bus d'adresse interne et le circuit de sélection (une nouvelle adresse est alors placée dans le registre CO) ;
- déclenche une commande de lecture mémoire via le bus de commande Processeur/Mémoire ;
- reçoit dans le registre RDO (via bus de données Processeur/Mémoire) le mot mémoire de l'instruction ;
- Place le contenu du RDO dans le registre d'instruction RI (via bus interne) ;
- Le décodeur analyse l'instruction stockée dans RI, la reconnaît et indique au séquenceur sa nature ;
- le séquenceur déclenche, au rythme de l'horloge, la séquence de micro-commandes nécessaires à la réalisation de cette instruction.

Le *chemin de données* est donc CO, RAD, commande de lecture, RDO, RI, décodage et séquençage. Charger ensuite les *opérandes* de l'instruction. 

## Lecture d'une donnée

Pour lire un opérande de l'instruction, le processeur :

- Place l'adresse de la donnée dans le registre RAD ;

Le *chemin de données* est donc CO, RAD, commande de lecture, RDO, RI.

## Lecture d'une donnée

Pour lire un opérande de l'instruction, le processeur :

- Place l'adresse de la donnée dans le registre RAD ;
- déclenche une commande de lecture mémoire ;

Le *chemin de données* est donc CO, RAD, commande de lecture, RDO, RI.

## Lecture d'une donnée

Pour lire un opérande de l'instruction, le processeur :

- Place l'adresse de la donnée dans le registre RAD ;
- déclenche une commande de lecture mémoire ;
- reçoit la donnée dans le registre RDO ;

Le *chemin de données* est donc CO, RAD, commande de lecture, RDO, RI.

## Lecture d'une donnée

Pour lire un opérande de l'instruction, le processeur :

- Place l'adresse de la donnée dans le registre RAD ;
- déclenche une commande de lecture mémoire ;
- reçoit la donnée dans le registre RDO ;
- place le contenu du RDO dans l'un des registres du microprocesseur (soit un registre général, soit un registre de d'entrée de l'UAL).

Le *chemin de données* est donc CO, RAD, commande de lecture, RDO, RI.

# UAL

Contient des registres et des circuits électroniques

- *Registres de données* : chargés avec des valeurs provenant de la mémoire. L'UC indique quelle opération doit être effectuée avec ces données.

# UAL

Contient des registres et des circuits électroniques

- *Registres de données* : chargés avec des valeurs provenant de la mémoire. L'UC indique quelle opération doit être effectuée avec ces données.
- *Accumulateur* : registre spécial contenant le résultat du calcul demandé par l'UC et réalisé par l'UAL avec le contenu des registres de données.

# UAL

Contient des registres et des circuits électroniques

- *Registres de données* : chargés avec des valeurs provenant de la mémoire. L'UC indique quelle opération doit être effectuée avec ces données.
- *Accumulateur* : registre spécial contenant le résultat du calcul demandé par l'UC et réalisé par l'UAL avec le contenu des registres de données.
- L'UAL possède des circuit permettant de faire :



# UAL

Contient des registres et des circuits électroniques

- *Registres de données* : chargés avec des valeurs provenant de la mémoire. L'UC indique quelle opération doit être effectuée avec ces données.
- *Accumulateur* : registre spécial contenant le résultat du calcul demandé par l'UC et réalisé par l'UAL avec le contenu des registres de données.
- L'UAL possède des circuit permettant de faire :
  - des opérations arithmétiques (+,-,...)

# UAL

Contient des registres et des circuits électroniques

- *Registres de données* : chargés avec des valeurs provenant de la mémoire. L'UC indique quelle opération doit être effectuée avec ces données.
- *Accumulateur* : registre spécial contenant le résultat du calcul demandé par l'UC et réalisé par l'UAL avec le contenu des registres de données.
- L'UAL possède des circuit permettant de faire :
  - des opérations arithémétiques (+,-,...)
  - des opérations logiques (et,ou, complément à 1...)

# UAL

Contient des registres et des circuits électroniques

- *Registres de données* : chargés avec des valeurs provenant de la mémoire. L'UC indique quelle opération doit être effectuée avec ces données.
- *Accumulateur* : registre spécial contenant le résultat du calcul demandé par l'UC et réalisé par l'UAL avec le contenu des registres de données.
- L'UAL possède des circuit permettant de faire :
  - des opérations arithémétiques (+,-,...)
  - des opérations logiques (et,ou, complément à 1...)
  - des comparaisons ( $\leq$ ,  $\geq$ ,  $<$ ,  $>$ ,  $==$ , ...)

# UAL

Contient des registres et des circuits électroniques

- *Registres de données* : chargés avec des valeurs provenant de la mémoire. L'UC indique quelle opération doit être effectuée avec ces données.
- *Accumulateur* : registre spécial contenant le résultat du calcul demandé par l'UC et réalisé par l'UAL avec le contenu des registres de données.
- L'UAL possède des circuit permettant de faire :
  - des opérations arithmétiques (+,-,...)
  - des opérations logiques (et,ou, complément à 1...)
  - des comparaisons ( $\leq$ ,  $\geq$ ,  $<$ ,  $>$ ,  $==$ , ...)
  - Des opérations sur les bits (décalages, rotations)

# UAL

Contient des registres et des circuits électroniques

- *Registres de données* : chargés avec des valeurs provenant de la mémoire. L'UC indique quelle opération doit être effectuée avec ces données.
- *Accumulateur* : registre spécial contenant le résultat du calcul demandé par l'UC et réalisé par l'UAL avec le contenu des registres de données.
- L'UAL possède des circuit permettant de faire :
  - des opérations arithmétiques (+,-,...)
  - des opérations logiques (et,ou, complément à 1...)
  - des comparaisons ( $\leq$ ,  $\geq$ ,  $<$ ,  $>$ ,  $==$ , ...)
  - Des opérations sur les bits (décalages, rotations)
  - Des copies depuis ou vers la mémoire.

# UAL

Contient des registres et des circuits électroniques

- *Registres de données* : chargés avec des valeurs provenant de la mémoire. L'UC indique quelle opération doit être effectuée avec ces données.
- *Accumulateur* : registre spécial contenant le résultat du calcul demandé par l'UC et réalisé par l'UAL avec le contenu des registres de données.
- L'UAL possède des circuit permettant de faire :
  - des opérations arithmétiques (+,-,...)
  - des opérations logiques (et,ou, complément à 1...)
  - des comparaisons ( $\leq$ ,  $\geq$ ,  $<$ ,  $>$ ,  $==$ , ...)
  - Des opérations sur les bits (décalages, rotations)
  - Des copies depuis ou vers la mémoire.
- Parfois une opération se passe mal (dépassement de capacité, division par zéro etc.). Un registre spécial, le PSW (*Program Status Word*), contient des informations sur le bon déroulement (ou non) de l'instruction. L'UAL peut alors envoyer un signal adapté (division par

# Mémoire et périphériques, DMA

- Les échanges entre l'UCT et la mémoire se font grâce à des bus.

# Mémoire et périphériques, DMA

- Les échanges entre l'UCT et la mémoire se font grâce à des bus.
- Il y a aussi un bus dédié aux échanges directs d'information entre la mémoire et les périphériques SANS PASSER par le processeur.



# Mémoire et périphériques, DMA

- Les échanges entre l'UCT et la mémoire se font grâce à des bus.
- Il y a aussi un bus dédié aux échanges directs d'information entre la mémoire et les périphériques SANS PASSER par le processeur.
- Le bus qui réalise cet accès direct à la mémoire est un circuit électronique appelé DMA (Direct Memory Access)

# Périphériques d'entrées/sorties

- Connectés à l'ordinateur par des circuits électroniques appelés *ports d'entrée-sortie* sur lesquels il est possible d'envoyer ou recevoir des données.

## Périphériques d'entrées/sorties

- Connectés à l'ordinateur par des circuits électroniques appelés *ports d'entrée-sortie* sur lesquels il est possible d'envoyer ou recevoir des données.
- L'accès à ces ports se fait à travers des emplacements mémoires occupant une adresse prédéfinie.

## Périphériques d'entrées/sorties

- Connectés à l'ordinateur par des circuits électroniques appelés *ports d'entrée-sortie* sur lesquels il est possible d'envoyer ou recevoir des données.
- L'accès à ces ports se fait à travers des emplacements mémoires occupant une adresse prédéfinie.
- Lire ou recevoir des données depuis/vers un périphérique consiste à lire ou écrire dans la RAM à ces emplacements.

## Périphériques d'entrées/sorties

- Connectés à l'ordinateur par des circuits électroniques appelés *ports d'entrée-sortie* sur lesquels il est possible d'envoyer ou recevoir des données.
- L'accès à ces ports se fait à travers des emplacements mémoires occupant une adresse prédéfinie.
- Lire ou recevoir des données depuis/vers un périphérique consiste à lire ou écrire dans la RAM à ces emplacements.
- Pour connaître l'état d'un périphérique, l'UCT peut :

## Périphériques d'entrées/sorties

- Connectés à l'ordinateur par des circuits électroniques appelés *ports d'entrée-sortie* sur lesquels il est possible d'envoyer ou recevoir des données.
- L'accès à ces ports se fait à travers des emplacements mémoires occupant une adresse prédéfinie.
- Lire ou recevoir des données depuis/vers un périphérique consiste à lire ou écrire dans la RAM à ces emplacements.
- Pour connaître l'état d'un périphérique, l'UCT peut :
  - soit consulter régulièrement l'emplacement associé;

## Périphériques d'entrées/sorties

- Connectés à l'ordinateur par des circuits électroniques appelés *ports d'entrée-sortie* sur lesquels il est possible d'envoyer ou recevoir des données.
- L'accès à ces ports se fait à travers des emplacements mémoires occupant une adresse prédéfinie.
- Lire ou recevoir des données depuis/vers un périphérique consiste à lire ou écrire dans la RAM à ces emplacements.
- Pour connaître l'état d'un périphérique, l'UCT peut :
  - soit consulter régulièrement l'emplacement associé ;
  - soit être prévenu d'un changement par le périphérique via un mécanisme spécial d'*interruption*.

## Périphériques d'entrées/sorties

- Connectés à l'ordinateur par des circuits électroniques appelés *ports d'entrée-sortie* sur lesquels il est possible d'envoyer ou recevoir des données.
- L'accès à ces ports se fait à travers des emplacements mémoires occupant une adresse prédéfinie.
- Lire ou recevoir des données depuis/vers un périphérique consiste à lire ou écrire dans la RAM à ces emplacements.
- Pour connaître l'état d'un périphérique, l'UCT peut :
  - soit consulter régulièrement l'emplacement associé ;
  - soit être prévenu d'un changement par le périphérique via un mécanisme spécial d'*interruption*.
- Une fois interrompu, l'UCT peut aller consulter le contenu des ports.



# Algorithme d'exécution d'une instruction

## Listing 1 – Algorithme simplifié

```
debut
  Charger l'instruction à exécuter depuis la RAM vers le RI;
  Modifier le CO pour qu'il pointe vers la prochaine instruction;
  Décoder l'instruction contenu dans le RI;
  Charger les éventuelles données dans les registre;
  Exécuter les micro-commandes dans l'ordre
    donné par le séquenceur;
fin
```

# Algorithme d'exécution d'un programme machine

## Listing 2 – Algorithme simplifié

---

```
1 debut  
2   Exécuter la 1ere instruction du programme;  
3   tant_que ce n'est pas la dernière instruction faire  
4     exécuter l'instruction;  
5   fin  
6 fin
```

---

---

1 L'ordinateur

2 La mémoire

# Les différents types de mémoire

## Classement par usage

- *mémoire vive* (RAM) mémoire où chaque information stockée peut à tout moment être consultée, ou modifiée. **La mémoire centrale des ordinateurs est la plupart du temps une mémoire vive volatile.**

# Les différents types de mémoire

## Classement par usage

- *mémoire vive* (RAM) mémoire où chaque information stockée peut à tout moment être consultée, ou modifiée. La mémoire centrale des ordinateurs est la plupart du temps une mémoire vive volatile.
- *mémoire morte* (ROM) mémoire où les informations sont écrites une fois mais ne peuvent pas être modifiées.

# Les différents types de mémoire

## Classement par usage

- *mémoire vive* (RAM) mémoire où chaque information stockée peut à tout moment être consultée, ou modifiée. La mémoire centrale des ordinateurs est la plupart du temps une mémoire vive volatile.
- *mémoire morte* (ROM) mémoire où les informations sont écrites une fois mais ne peuvent pas être modifiées.
  - Les mémoires mortes sont utilisées par exemple pour stocker définitivement des logiciels enfouis.

# Les différents types de mémoire

## Classement par usage

- *mémoire vive* (RAM) mémoire où chaque information stockée peut à tout moment être consultée, ou modifiée. La mémoire centrale des ordinateurs est la plupart du temps une mémoire vive volatile.
- *mémoire morte* (ROM) mémoire où les informations sont écrites une fois mais ne peuvent pas être modifiées.
  - Les mémoires mortes sont utilisées par exemple pour stocker définitivement des logiciels enfouis.
  - La mémoire morte fournit des données invariables nécessaires au démarrage de l'ordinateur.

# Les différents types de mémoire

## Classement par usage

- *mémoire vive* (RAM) mémoire où chaque information stockée peut à tout moment être consultée, ou modifiée. La mémoire centrale des ordinateurs est la plupart du temps une mémoire vive volatile.
- *mémoire morte* (ROM) mémoire où les informations sont écrites une fois mais ne peuvent pas être modifiées.
  - Les mémoires mortes sont utilisées par exemple pour stocker définitivement des logiciels enfouis.
  - La mémoire morte fournit des données invariables nécessaires au démarrage de l'ordinateur.
  - De nos jours, seul un système minimaliste tout juste suffisant au démarrage, le BIOS, a été maintenu. Son remplaçant l'UEFI, lui, est modifiable.



# Classement par coût

- Pour des raisons économiques, les mémoires sont en général divisées en plusieurs familles.

# Classement par coût

- Pour des raisons économiques, les mémoires sont en général divisées en plusieurs familles.
- Elles sont traitées, la plupart du temps, différemment par le système d'exploitation.

# Classement par coût

- Pour des raisons économiques, les mémoires sont en général divisées en plusieurs familles.
- Elles sont traitées, la plupart du temps, différemment par le système d'exploitation.
- Par ordre de rapport croissant prix/capacité et de vitesses croissantes ce sont la *Mémoire de masse ou mémoire de stockage*, la *La mémoire vive*, la *Mémoire cache* et les *registres*.

# Classement par coût

**Mémoire de masse** (ou mémoire de stockage) : stocker à long terme des grandes quantités d'informations. **Capacité de stockage élevée à faible coût, vitesse inférieure aux autres mémoires. Contenu persistant. Exemple : disque dur**

## Classement par coût

**Mémoire de masse** (ou mémoire de stockage) : stocker à long terme des grandes quantités d'informations. Capacité de stockage élevée à faible coût, vitesse inférieure aux autres mémoires. Contenu persistant. Exemple : disque dur

**Mémoire vive** : espace principal de stockage du microprocesseur. **Contenu volatile, disparaît lors de la mise hors tension de l'ordinateur.**  
Ex : mémoire DDR4

# Classement par coût

**Mémoire de masse** (ou mémoire de stockage) : stocker à long terme des grandes quantités d'informations. Capacité de stockage élevée à faible coût, vitesse inférieure aux autres mémoires. Contenu persistant. Exemple : disque dur

**Mémoire vive** : espace principal de stockage du microprocesseur. Contenu volatile, disparaît lors de la mise hors tension de l'ordinateur. Ex : mémoire DDR4

**Mémoire cache** : conserver un court instant des informations fréquemment consultées. C'est un buffer mémoire vive/processeur  
**But : accélérer la vitesse des opérations de consultation.**  
**Grande vitesse, coût élevé, faible capacité de stockage. Deux versions : interne au processeur ou externe**

# Classement par coût

**Mémoire de masse** (ou mémoire de stockage) : stocker à long terme des grandes quantités d'informations. Capacité de stockage élevée à faible coût, vitesse inférieure aux autres mémoires. Contenu persistant. Exemple : disque dur

**Mémoire vive** : espace principal de stockage du microprocesseur. Contenu volatile, disparaît lors de la mise hors tension de l'ordinateur. Ex : mémoire DDR4

**Mémoire cache** : conserver un court instant des informations fréquemment consultées. C'est un buffer mémoire vive/processeur  
But : accélérer la vitesse des opérations de consultation. Grande vitesse, coût élevé, faible capacité de stockage. Deux versions : interne au processeur ou externe

**Registres** : intégrés au processeur. Très rapide mais très cher donc réservé à une très faible quantité de données.

# Les différents types de mémoire

Classement par comportement vis à vis de l'alimentation électrique

- *Mémoire volatile* : mémoire où les informations sont perdues lors de la mise hors tension de l'appareil.



# Les différents types de mémoire

Classement par comportement vis à vis de l'alimentation électrique

- *Mémoire volatile* : mémoire où les informations sont perdues lors de la mise hors tension de l'appareil.
- *Mémoire rémanente* ou non volatile : mémoire où les informations sont conservées même après la mise hors tension de l'appareil.

# Les différents types de mémoire

Classement par comportement vis à vis de l'alimentation électrique

- *Mémoire volatile* : mémoire où les informations sont perdues lors de la mise hors tension de l'appareil.
- *Mémoire rémanente* ou non volatile : mémoire où les informations sont conservées même après la mise hors tension de l'appareil.
- *Mémoire flash* : mémoire rémanente rapide, dépourvue de pièce mécanique **donc résistante au choc. Pour cette raison, utilisée dans les smartphone. Ex : disque dur SSD.**

# Les différents types de mémoire

## Classement par comportement vis à vis de l'alimentation électrique

- *Mémoire volatile* : mémoire où les informations sont perdues lors de la mise hors tension de l'appareil.
- *Mémoire rémanente* ou non volatile : mémoire où les informations sont conservées même après la mise hors tension de l'appareil.
- *Mémoire flash* : mémoire rémanente rapide, dépourvue de pièce mécanique donc résistante au choc. Pour cette raison, utilisée dans les smartphones. Ex : disque dur SSD.
- *Mémoire virtuelle* : mécanisme qui permet de donner plus de mémoire au processeur pour travailler, en simulant la présence d'un type de mémoire tout en utilisant un autre type. Il est utilisé par exemple pour simuler la présence de mémoire vive en utilisant de la mémoire de masse.

# Le langage C et la mémoire

- Avec un langage bas niveau comme C, on gère simplement la RAM mais pas les registres. OCAML n'est pas bas niveau : l'utilisateur n'interagit pas directement avec la RAM.

# Le langage C et la mémoire

- Avec un langage bas niveau comme C, on gère simplement la RAM mais pas les registres. OCAML n'est pas bas niveau : l'utilisateur n'interagit pas directement avec la RAM.
- Seuls les langages *très bas niveau* comme l'*assembleur* gèrent les caches et registres.  
L'assembleur dépend du type de processeur de la machine et n'est pas portable.

## Quelques registres spécialisés

- Registres affectés à des types de données :

## Quelques registres spécialisés

- Registres affectés à des types de données :
  - Registres entiers : chargés de stocker des nombres entiers

## Quelques registres spécialisés

- Registres affectés à des types de données :
  - Registres entiers : chargés de stocker des nombres entiers
  - Registres flottants : stockent les nombres à virgule flottante



## Quelques registres spécialisés

- Registres affectés à des types de données :
  - Registres entiers : chargés de stocker des nombres entiers
  - Registres flottants : stockent les nombres à virgule flottante
  - Registres d'adresses : stockent des adresses (ce rôle peut être rempli par les registres entiers)

## Quelques registres spécialisés

- Registres affectés à des types de données :
  - Registres entiers : chargés de stocker des nombres entiers
  - Registres flottants : stockent les nombres à virgule flottante
  - Registres d'adresses : stockent des adresses (ce rôle peut être rempli par les registres entiers)
  - Registres à prédicats : stockent des résultats de comparaison

## Quelques registres spécialisés

- Registres affectés à des types de données :
  - Registres entiers : chargés de stocker des nombres entiers
  - Registres flottants : stockent les nombres à virgule flottante
  - Registres d'adresses : stockent des adresses (ce rôle peut être rempli par les registres entiers)
  - Registres à prédicats : stockent des résultats de comparaison
- Sur certaines architectures, l'ensemble ou une partie des registres spécialisés sont remplacés par des registres généraux. Ceux-ci peuvent stocker indifféremment des adresses, des entiers, des flottants, etc.

## Quelques registres spécialisés

- Registres affectés à des types de données :
  - Registres entiers : chargés de stocker des nombres entiers
  - Registres flottants : stockent les nombres à virgule flottante
  - Registres d'adresses : stockent des adresses (ce rôle peut être rempli par les registres entiers)
  - Registres à prédicats : stockent des résultats de comparaison
- Sur certaines architectures, l'ensemble ou une partie des registres spécialisés sont remplacés par des registres généraux. Ceux-ci peuvent stocker indifféremment des adresses, des entiers, des flottants, etc.
- Certains registres figurent en un seul exemplaire :

## Quelques registres spécialisés

- Registres affectés à des types de données :
  - Registres entiers : chargés de stocker des nombres entiers
  - Registres flottants : stockent les nombres à virgule flottante
  - Registres d'adresses : stockent des adresses (ce rôle peut être rempli par les registres entiers)
  - Registres à prédicats : stockent des résultats de comparaison
- Sur certaines architectures, l'ensemble ou une partie des registres spécialisés sont remplacés par des registres généraux. Ceux-ci peuvent stocker indifféremment des adresses, des entiers, des flottants, etc.
- Certains registres figurent en un seul exemplaire :
  - Compteur ordinal : indique l'emplacement de la prochaine instruction à effectuer.

## Quelques registres spécialisés

- Registres affectés à des types de données :
  - Registres entiers : chargés de stocker des nombres entiers
  - Registres flottants : stockent les nombres à virgule flottante
  - Registres d'adresses : stockent des adresses (ce rôle peut être rempli par les registres entiers)
  - Registres à prédicats : stockent des résultats de comparaison
- Sur certaines architectures, l'ensemble ou une partie des registres spécialisés sont remplacés par des registres généraux. Ceux-ci peuvent stocker indifféremment des adresses, des entiers, des flottants, etc.
- Certains registres figurent en un seul exemplaire :
  - Compteur ordinal : indique l'emplacement de la prochaine instruction à effectuer.
  - Registre d'état : décrit l'état du processeur. Une utilisation fréquente de ce registre consiste à déterminer si un branchement (saut vers une portion spécifique du code) doit être effectué.

## Quelques registres spécialisés

- Registres affectés à des types de données :
  - Registres entiers : chargés de stocker des nombres entiers
  - Registres flottants : stockent les nombres à virgule flottante
  - Registres d'adresses : stockent des adresses (ce rôle peut être rempli par les registres entiers)
  - Registres à prédicats : stockent des résultats de comparaison
- Sur certaines architectures, l'ensemble ou une partie des registres spécialisés sont remplacés par des registres généraux. Ceux-ci peuvent stocker indifféremment des adresses, des entiers, des flottants, etc.
- Certains registres figurent en un seul exemplaire :
  - Compteur ordinal : indique l'emplacement de la prochaine instruction à effectuer.
  - Registre d'état : décrit l'état du processeur. Une utilisation fréquente de ce registre consiste à déterminer si un branchement (saut vers une portion spécifique du code) doit être effectué.
  - Pointeur de pile : indique la position du prochain emplacement disponible dans la pile mémoire.