Module M211
Cours Architecture

Programmation des mécanismes de base d'un système informatique

Marie-Agnès Peraldi-Frati
MCF Informatique

Objectif des modules ASR

- Le champ disciplinaire « **Architecture matérielle - Systèmes d'exploitation - Réseaux** » concerne à la fois les connaissances de base sur le matériel (codage de l'information, fonctionnement interne des ordinateurs), les systèmes d'exploitation professionnels multitâches et multi utilisateurs (utilisation, administration, utilisation des services par programmation), ainsi que les réseaux et leurs normes (organisation, fonctionnement, éléments d'administration, techniques de programmation d'applications réparties ou distribuées).

M211: Organisation et périmètre

- Organisation :
  - Architecture et programmation des mécanismes de base d'un système informatique.
  - 1h Cours, 3h TD/TP par semaine pendant 5 semaines.
  - 1h d'examen en fin de module
  - Utilisation de l'émulateur Proc X86 en TP http://www.emu8086.com
- Périmètre
  - De quoi est composé un ordinateur ?
  - Quel est le lien entre le matériel et le logiciel?
    - Modèles d'exécution (séquentiel, parallèle données, parallèles instructions
    - Modèle de mémoires (registres, cache L1 L2 …)
  - Comment s'exécute un programme ?
  - Comment fonctionnent les divers périphériques ?

Agenda du cours

- Cours 1 :
  - Généralités archi/assembleur
  - Manipulation émulateur
  - **Code, UAL, registres, mémoire**
  - Exécution, visualisation registres
- Cours 2 :
  - Différents **types de mémoires**
  - Accès mémoire (code, données, E/S)
  - Manipulation structure de données en assembleur
- Cours 3 :
  - **Notion de Pile**
  - Appel de procédures
- Cours 4 :
  - **Interruptions**
  - E/S
- Cours 5 :
  - Développement programme
  - E/S , IT,..
- Cours 6 :
  - **Examen**

PPN National

UE21	Informatique approfondie Architecture matérielle - Systèmes d'exploitation - Réseaux	Volume Horaire 8h CM, 10h TD, 12h TP
M2101	Architecture et Programmation des mécanismes de base d'un système informatique	Semestre 2

Objectifs du module :
- Savoir développer des applications simples mettant en œuvre les mécanismes de bas niveau d'un système informatique

Compétences visées :
Compétences citées dans le Référentiel d'activités et de compétences pour les activités :
- FA2-A : Administration de systèmes, de logiciels et de réseaux
- FA2-B : Conseil et assistance technique à des utilisateurs, clients, services
- FA1-B : Conception technique d'une solution informatique

Prérequis :
- M1101
- M1102

Contenus :
- Langages de programmation bas niveau
- Mécanismes de bas niveau d'un système informatique
- Etude d'un système à microprocesseur ou microcontrôleur (réel ou simulé) avec ses composants (mémoires, interfaces, périphériques...)

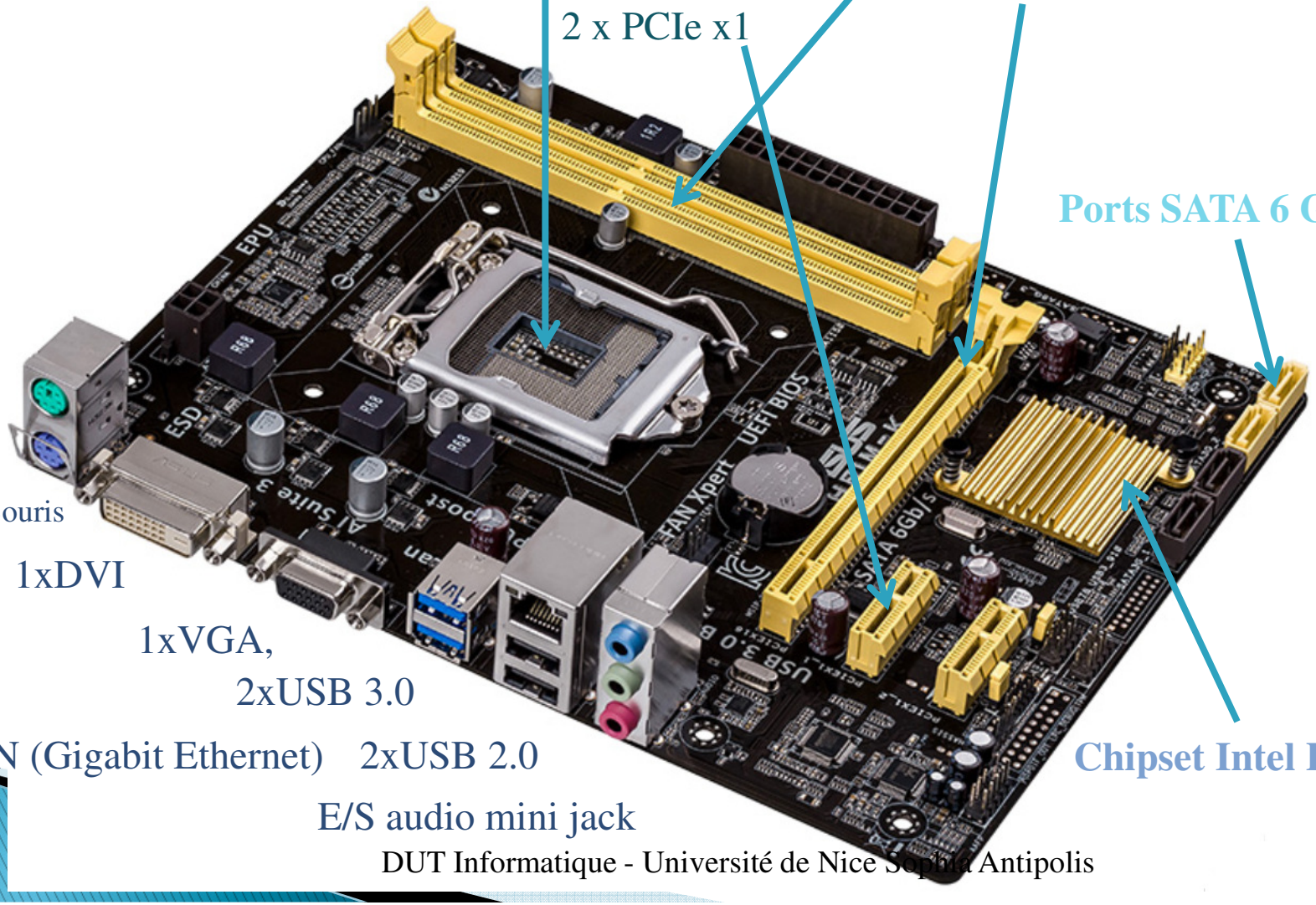
Modalités de mise en œuvre :
- Utilisation du langage C et/ou d'un langage d'assemblage (assembleur)
- Observation de l'exécution pas à pas d'un programme à l'aide d'un outil de simulation/déverminage d'un processeur simple
- Développement de programmes simples permettant d'illustrer les principaux mécanismes de bas niveau d'un système informatique
- Etude des mécanismes de gestion des interruptions

Prolongements possibles :
- Programmation des Systèmes Embarqués
- Processus de compilation
- Etude du fonctionnement d'un système d'exploitation (*OS* : *Operating System*) minimal embarqué

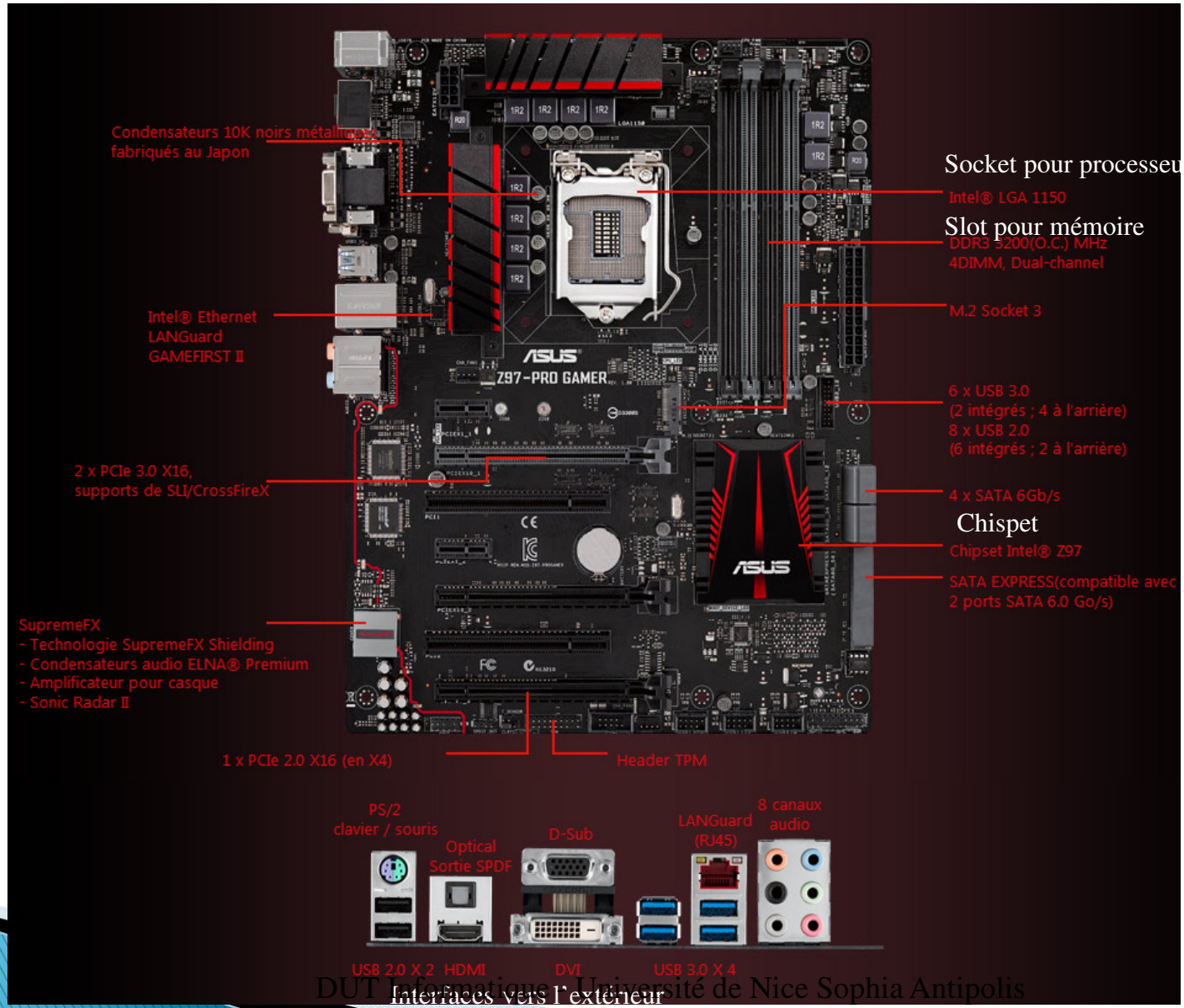
Un ordinateur

- Unité centrale
  - Carte mère
    - Processeur
    - Mémoire vive /mémoire de stockage
    - Chipset
  - Carte graphique
  - Alimentation
- Périphériques d'entrée : clavier, souris
- Périphériques de sortie : Ecrans HP
- Périphériques de stockage disques, lecteurs CD, DVD

Carte mère ASUS H61M-K



Exemple Carte mère ATX ASUS pro Gamer



Interfaces vers l'extérieur

MotherBoard inside !

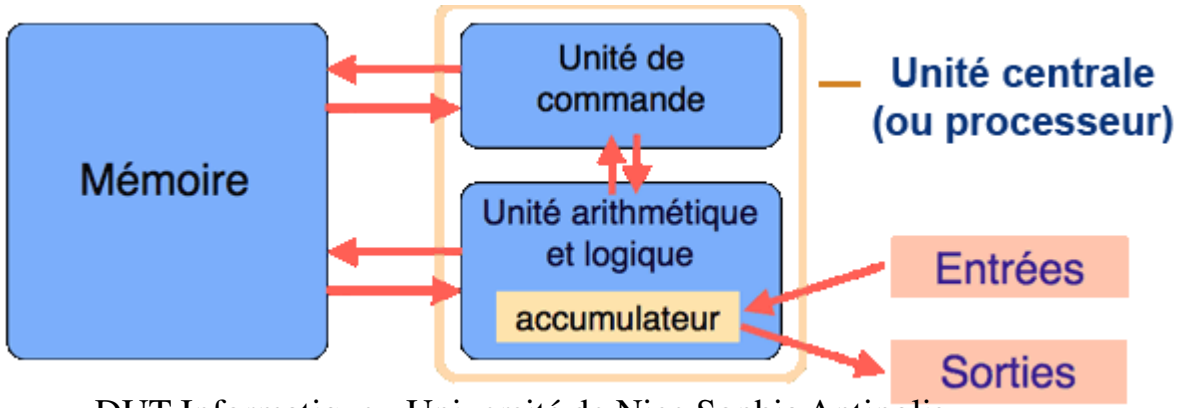
- CPU = Processeur (Central Processing Unit)
  - Qui exécute les instructions de votre programme
- Processeurs dédiés présents sur une carte spécifique ou intégrés sur une carte-mère
  - GPU = (Graphics Processing Unit) assure les fonctions de calcul de l'affichage
  - PPU = (Physics Processing Unit) calcul de la dynamique physique (fluides, structures complexes …)

CPU inside !

- Processeur : Circuit électronique cadencé par une horloge interne (quartz)
- **Fréquence d'horloge (cycle)** = Nbre d'impulsion de quartz /secondes (Hertz)
- A chaque impulsion d'horloge le processeur va lire une instruction stockée dans un registre d'instruction et l'exécute.
- Une **instruction** s'exécute en un ou plusieurs cycles
- **Puissance** du processeur = Nbre d'instructions par seconde. MIPS (Millions Instruction par seconde)

Architecture logique d'un processeur Von Neumann(1945)

- Unité de commande (UC), qui organise le séquencement des instructions
  - Compteur ordinal contient l'adresse de l'instruction en cours d'exécution, *(IP)*
    - soit incrémenté
    - soit modifié lors des sauts
  - Registre d'état : ensemble de flags qui donne l'état du calcul *(AF CF OF SF PF ZF DF IF TF)*
- Unité arithmétique UAL, exécute les instructions arithmétiques et logiques
  - Accumulateur (ou registre) : mémorise les résultats intermédiaires des calculs avant E/S *(AX BX CX DX)*
- Mémoire peut contenir les programmes et les données
  - Programme : suite d'instructions codées selon un format conventionnel (jeu d'instruction) *(CS)*
  - Données : valeurs encodées selon un format conventionnel (entier, chaîne, flottants …) *(DS)*



Architecture logique d'un processeur
Evolutions de AVN à nos jours

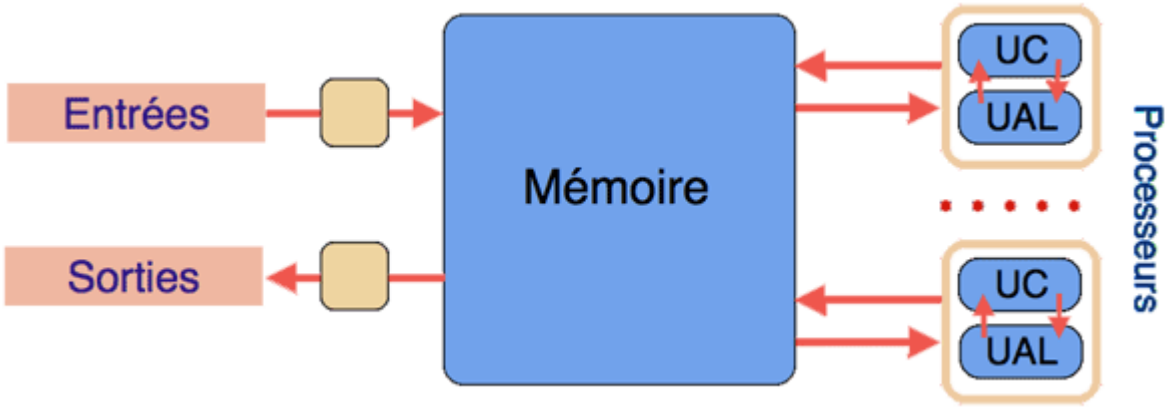
- Processeurs multiples, qu'il s'agisse d'unités séparées ou de « cœurs » multiples.

cela permet d'atteindre des puissances de calcul élevées non atteignable sur un seul procésseu

- Les entrées-sorties, initialement commandées par l'unité centrale, sont désormais sous le

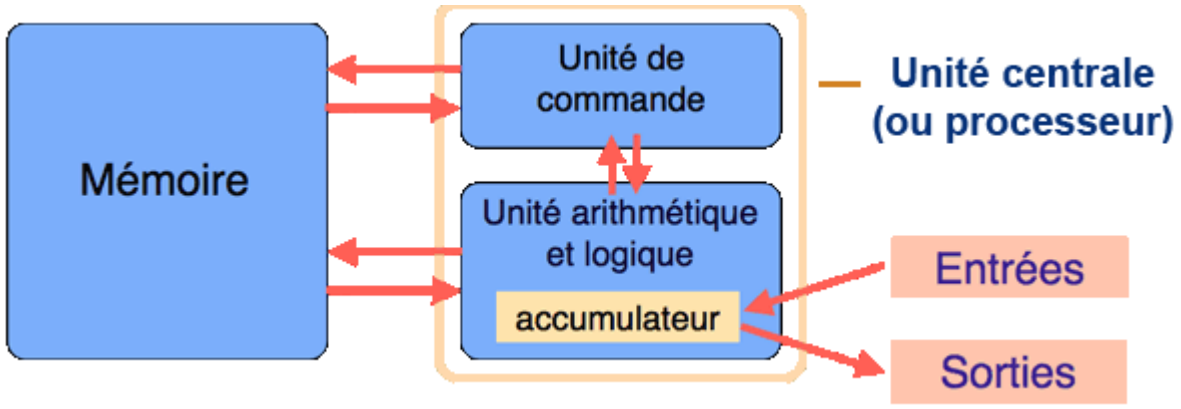
contrôle de processeurs autonomes (canaux d'entrée-sortie et mécanismes assimilés).

- Multiprogrammation: OS et temps partagé entre plusieurs programmes,



Revenons à l'archi de Von Neumann

- Unité de commande (UC), qui organise le séquencement des instructions
  - Compteur ordinal contient l'adresse de l'instruction en cours d'exécution,
    - soit incrémenté
    - modifié lors des sauts
  - Registre d'état
- Unité arithmétique UAL, exécute les instructions arithmétiques et logiques
- Accumulateur (ou registres) : mémorise les résultats intermédiaires des calculs avant E/S
- Mémoire peut contenir les programmes et les données
  - Programme : suite d'instructions codées selon un format conventionnel (jeu d'instruction)
  - Données : valeurs encodées selon un format conventionnel (entier, chaîne, flottants …)



Une instruction c"est quoi ?

- En C

```
#include <stdio.h>
int a=1;
int b=2;
void main(){
    b=15;   /* Ceci est un instruction C d'affectation */
    a=b+10;
}
```

- En Assembleur

```
DONNEES     SEGMENT
        intA                Db        1
        intB                Db        2
DONNEES     ENDS
CODE        SEGMENT
MAIN    PROC     FAR
  ; L'affectation dans une var nécessite 2 instructions en
ASM
        mov Al, 15  ; hex=0fh or bin=00001111b
        mov intB, Al
        ; intA reçoit intB+10
        mov Bl, 10 ; ; hex=0ah or bin=00001010b
        add Al,Bl ; Al <- Al+Bl
        mov intA,Al
    RET
MAIN        ENDP
CODE        ENDS
```

Une instruction c'est quoi ?

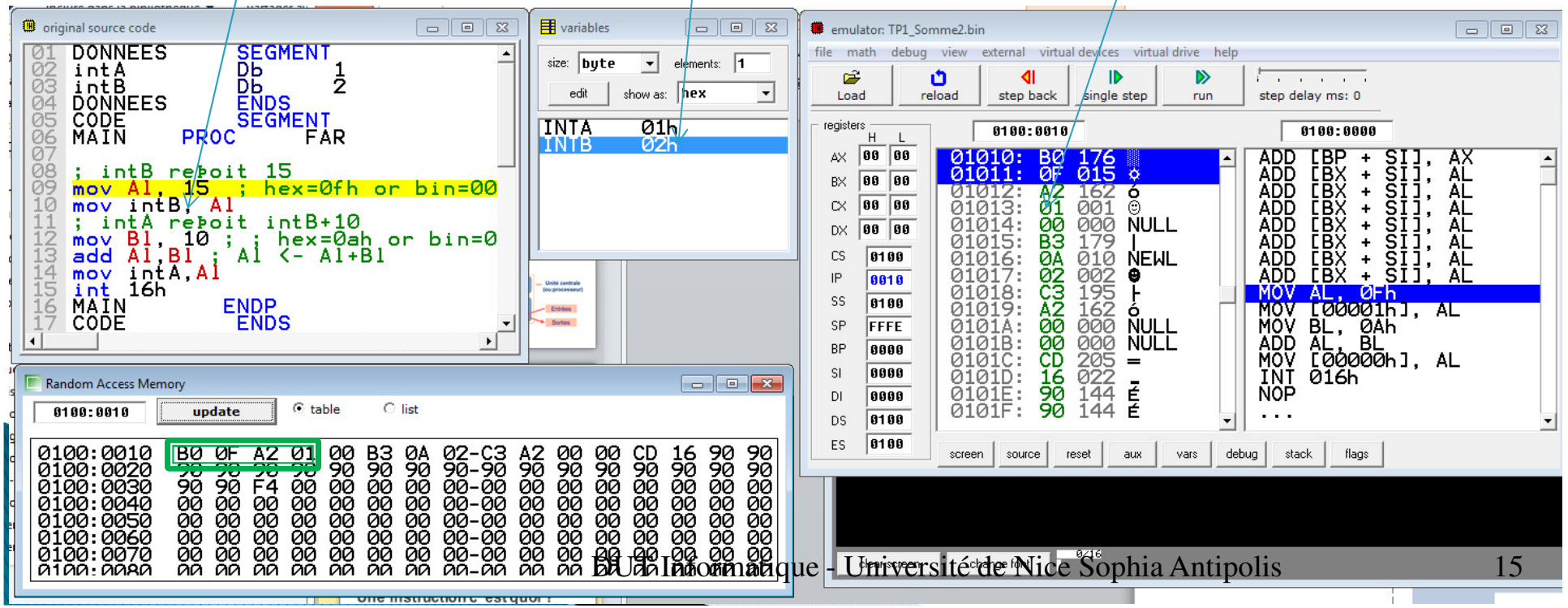
- En code machine 80x86

  Dans le jeu d'instruction assembleur, à chaque instruction ASM correspond un code machine.
  Exemple : MOV AL, 0Fh donne : B0 0F
  Avec B0 *le code signifiant « déplacer une valeur dans AL » et* 0F *étant cette valeur.*
  Le code machine qui correspond à b=15; est stocké dans la RAM et vaut B0 0F A2 01

Code assembleur

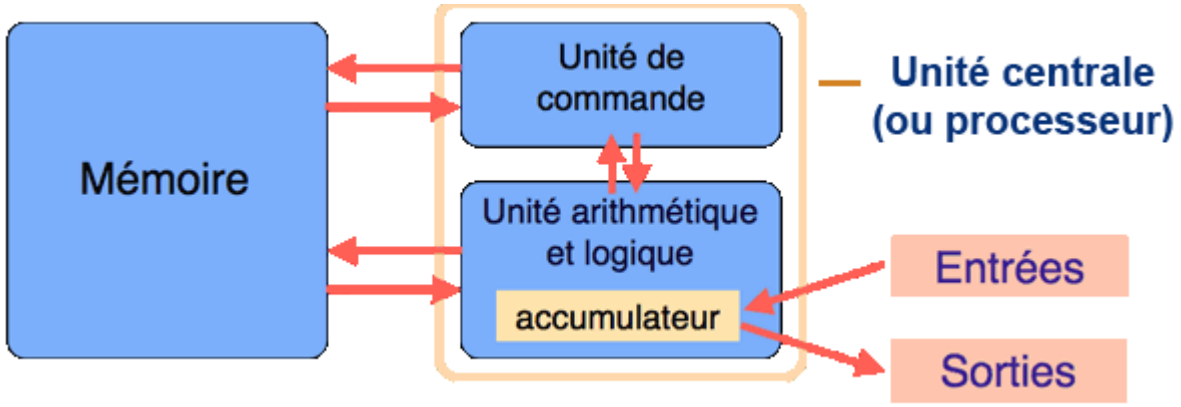
Variable b

Code machine



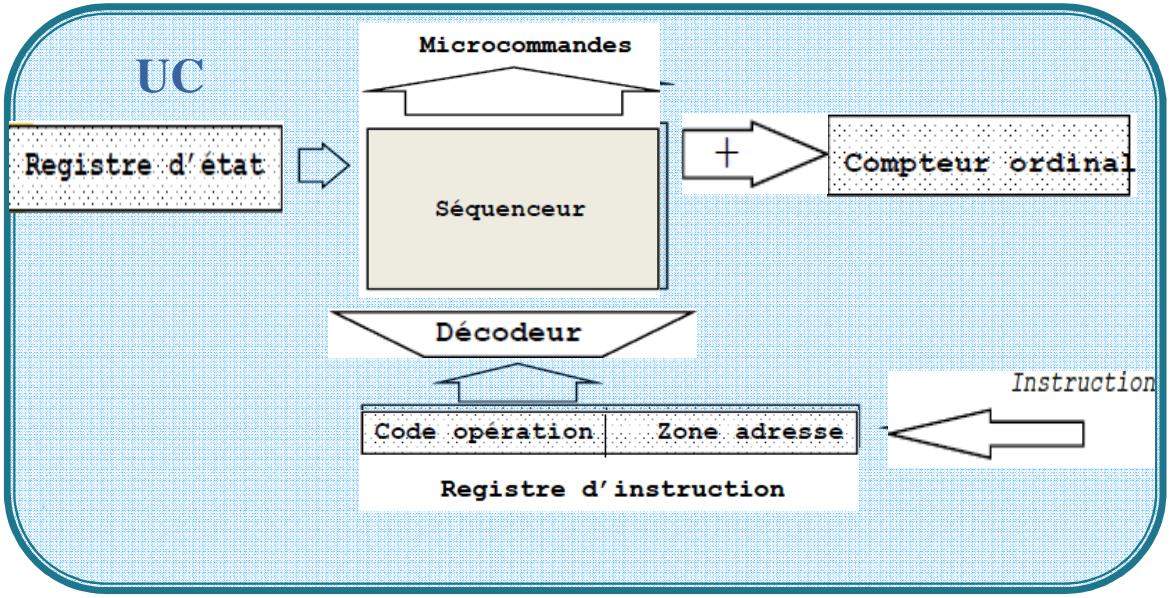
Revenons à l'archi de Von Neumann

- Unité de commande (UC), qui organise le séquencement des instructions
  - Compteur ordinal contient l'adresse de l'instruction en cours d'exécution,
    - soit incrémenté
    - modifié lors des sauts
  - Registre d'état
- Unité arithmétique UAL, exécute les instructions arithmétiques et logiques
- Accumulateur (ou registre) : mémorise les résultats intermédiaires des calculs avant E/S
- Mémoire peut contenir les programmes et les données
  - Programme : suite d'instructions codées selon un format conventionnel (jeu d'instruction)
  - Données : valeurs encodées selon un format conventionnel (entier, chaîne, flottants …)



Unité de commande :
organise l'exécution des instructions

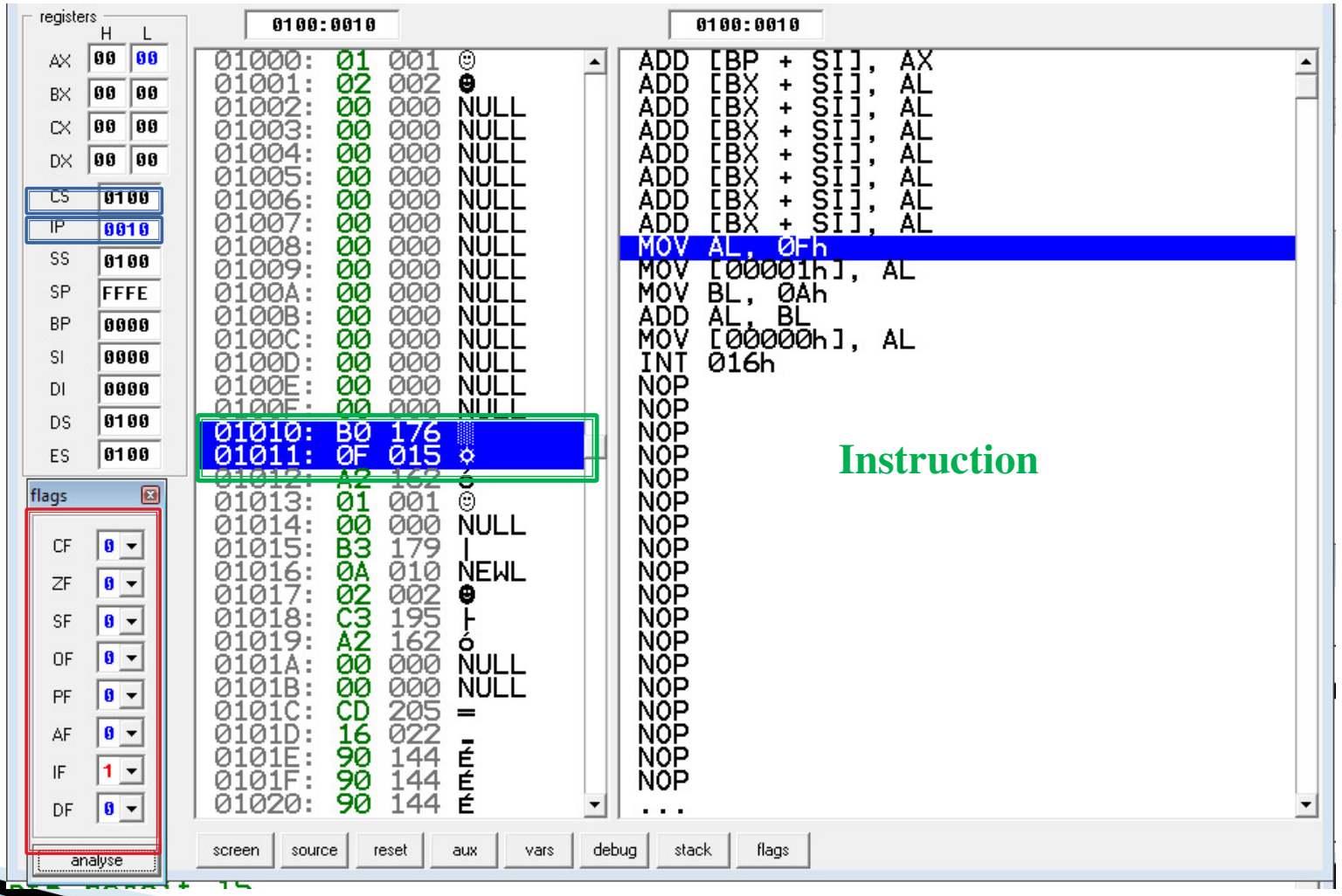
- Le compteur Ordinal
  - contient l'adresse mémoire de la prochaine instruction
  - Au chargement d'un programme par l'OS le CO récupère l'adresse mémoire de la première instruction du programme.
- Le séquenceur va à la recherche en mémoire de l'instruction
- L'Instruction stockée dans le registre d'instruction
- Elle est décodée par le décodeur
- les registres d'état sont lus
- Le séquenceur génère les microcommandes pour exécuter l'instruction (vers ALU, vers mémoire …)
- A la fin de l'instruction le CO est incrémenté pour passer à l'instruction suivante



Unité de commande :
organise l'exécution des instructions

Compteur Ordinal

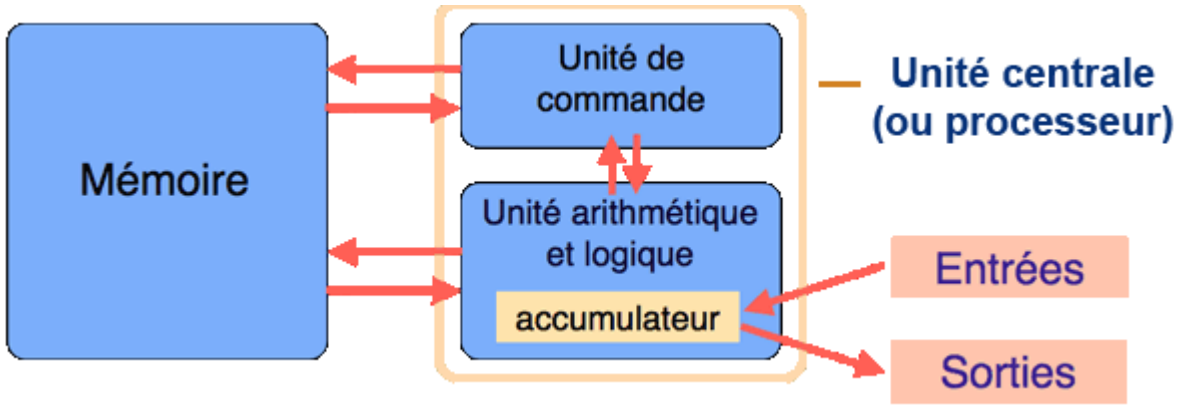
Registres d'état



Instruction

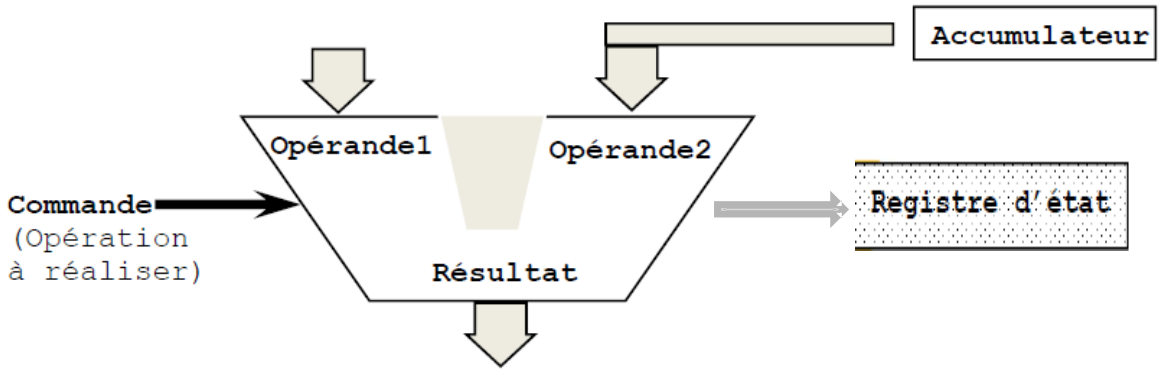
Revenons à l'archi de Von Neumann

- Unité de commande (UC), qui organise le séquencement des instructions
- Compteur ordinal contient l'adresse de l'instruction en cours d'exécution,
  - soit incrémenté
  - modifié lors des sauts
- Unité arithmétique UAL, exécute les instructions arithmétiques et logiques
- Accumulateur (ou registres) : mémorise les résultats intermédiaires des calculs avant E/S
- Mémoire peut contenir les programmes et les données
  - Programme : suite d'instructions codées selon un format conventionnel (jeu d'instruction)
  - Données : valeurs encodées selon un format conventionnel (entier, chaîne, flottants …)



Unité Arithmétique et logique

- Reçoit de L'UAL les **opérations** à effectuer et les **opérandes** associées.
- Composée de circuits logiques qui réalisent :
  - des opérations **arithmétiques classique pour un CPU** : + - * …**logique**: or and not
  - Des **opérations plus complexes :** pour un FPU/GPU : inverse, racine carré, produit scalaire, vectoriel



UAL : Exemple d'opération

- Opération de **comparaison** du contenu d'un registre avec 0

CMP AL, 0

  - Flag ZF du registre d'état
- Une UAL fait un nombre limité d'opérations. Fonctions manquantes à programmer

Registres d'état

Indicateur (généralement composé de 8 bits) dont l'état dépend du résultat de la dernière opération effectuée par l'UAL.
Indicateurs d'état ou flags.

- **ZF (Zero Flag)**
  - mis à 1 lorsque le résultat de la dernière opération est zéro.
  - sinon, ZF est positionné à 0.

- **CF (Carry Flag)**
  - Indicateur de report (retenue). Il est positionné par les instructions ADD, SUB et CMP (entiers naturels).
  - CF = 1 s'il y a une retenue

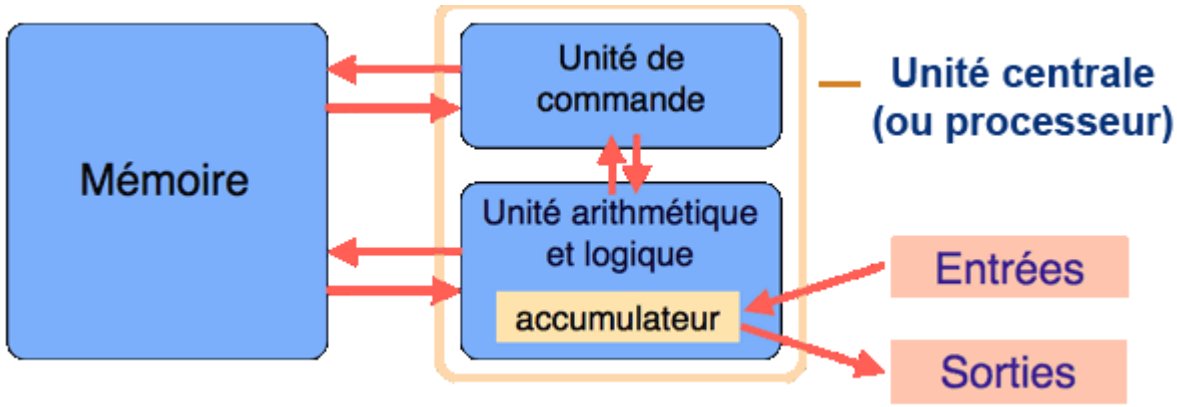
Registres d'état

- **SF (Sign Flag)**
  - Positionné à 1 si le bit de poids fort du résultat d'une addition ou soustraction est 1; sinon SF=0. Utile pour la manipulation des entiers relatifs, le bit de poids fort donne le signe du résultat.

- **OF (Overflow Flag) Indicateur de débordement**
  - OF=1 si le résultat d'une addition ou soustraction donne un nombre qui n'est pas codable en relatif dans l'accumulateur (par exemple si l'addition de 2 nombres positifs donne un codage négatif).

Revenons à l'archi de Von Neumann

- Unité de commande (UC), qui organise le séquencement des instructions
- Compteur ordinal contient l'adresse de l'instruction en cours d'exécution,
  - soit incrémenté
  - modifié lors des sauts
- Unité arithmétique UAL, exécute les instructions arithmétiques et logiques
- Accumulateurs (ou registres) : mémorise les résultats intermédiaires des calculs avant E/S
- Mémoire peut contenir les programmes et les données
  - Programme : suite d'instructions codées selon un format conventionnel (jeu d'instruction)
  - Données : valeurs encodées selon un format conventionnel (entier, chaîne, flottants …)



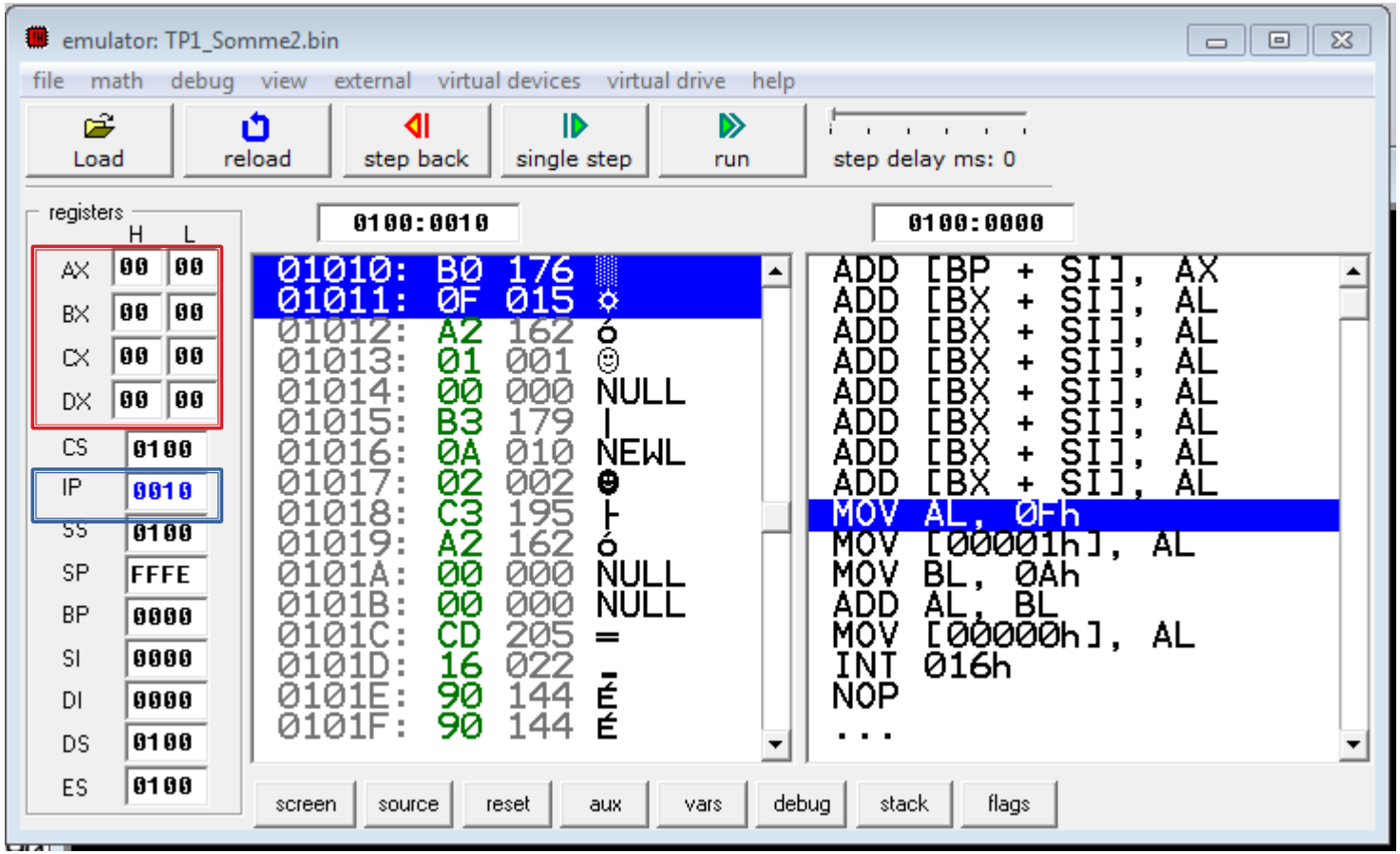
Accumulateurs

- Mémoire interne proche de l'UAL (accès rapide)
- 4 registres de données,
  - Décomposés en deux parties : une partie « haute » et une partie « basse » de 8 bits chacune, ce qui permet au microprocesseur de manipuler des données sur 8 ou 16 bits :
    - AX (décomposable en AH et AL) sert d'accumulateur et est principalement utilisé lors d'opérations arithmétiques et logiques ;
    - BX est la plupart du temps utilisé comme opérande dans les calculs ;
    - CX est utilisé comme compteur dans les structures itératives ;
    - DX, tout comme AX, est utilisé pour les calculs arithmétiques et notamment dans la division et la multiplication. Il intervient également dans les opérations d'entrées/sorties.

Accumulateurs
Visualisation sous l'émulateur 8086

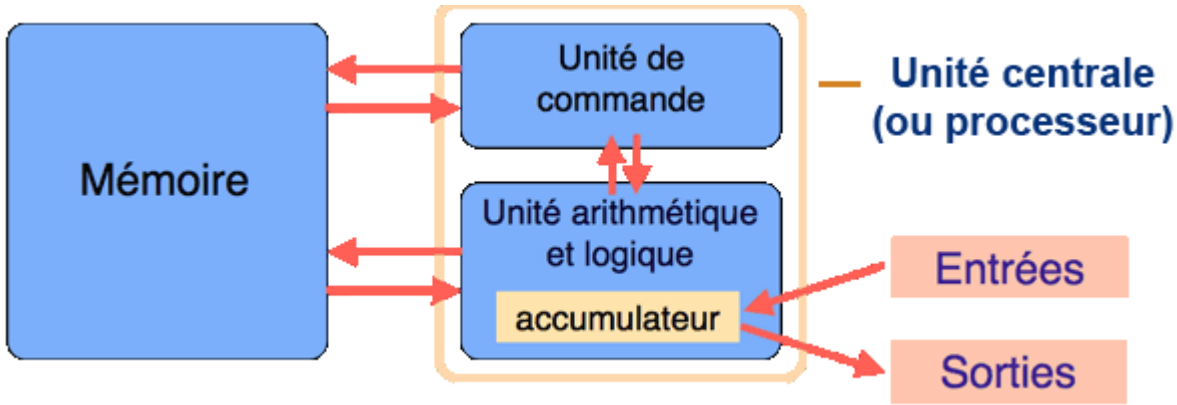
Accu-mulateurs

Compteur Ordinal



Revenons à l'archi de Von Neumann

- Unité de commande (UC), qui organise le séquencement des instructions
- Compteur ordinal contient l'adresse de l'instruction en cours d'exécution,
  - soit incrémenté
  - modifié lors des sauts
- Unité arithmétique UAL, exécute les instructions arithmétiques et logiques
- Accumulateurs (ou registres) : mémorise les résultats intermédiaires des calculs avant E/S
- Mémoire peut contenir les programmes et les données
  - Programme : suite d'instructions codées selon un format conventionnel (jeu d'instruction)
  - Données : valeurs encodées selon un format conventionnel (entier, chaîne, flottants …)



Mémoire

- Une mémoire peut être représentée comme une armoire de rangement constituée de différents tiroirs.
  - un tiroir représente alors une **case mémoire** qui peut contenir un seul élément **un octet.**
  - Une **adresse** identifie la case mémoire et chaque donnée est accessible grâce à son adresse
- On peut accéder simultanément à plusieurs tiroirs via des **bus d'adresses** et de **données**
- Exemple : avec un bus de données 64 bits on peut accéder à 4 tiroirs (4 octets) simultanément

Mémoire

- **Stocke** des informations utilisées par le processeur
- **Juxtaposition** de cellules (la plus petite quantité de mémoire adressable).
  - Chaque cellule est numérotée (adresse)
  - Généralement les mémoires sont adressables par octets
- **Taille** d'une cellule
  - octet : 8 bits (byte)
  - word : 16 bits,
  - Double word : 32 bits,
  - Quad word 64 bits : en fonction de machines (16 ou 32 bits)
- **Notation** :
  - Si M désigne une cellule mémoire
  - [M] représente son contenu

Mémoire : décomposition en segments

- **Principe** : la numérotation sur 16 bits a ses limites (2^16 cases adressables : 65535 octets (un octet = 8 bits)
- Notion de **segments** : consiste à regrouper les cases mémoires par zone plus grande (segment) de 65535 octets
- Exemple sur l'émulateur 8086:

Segment 0000

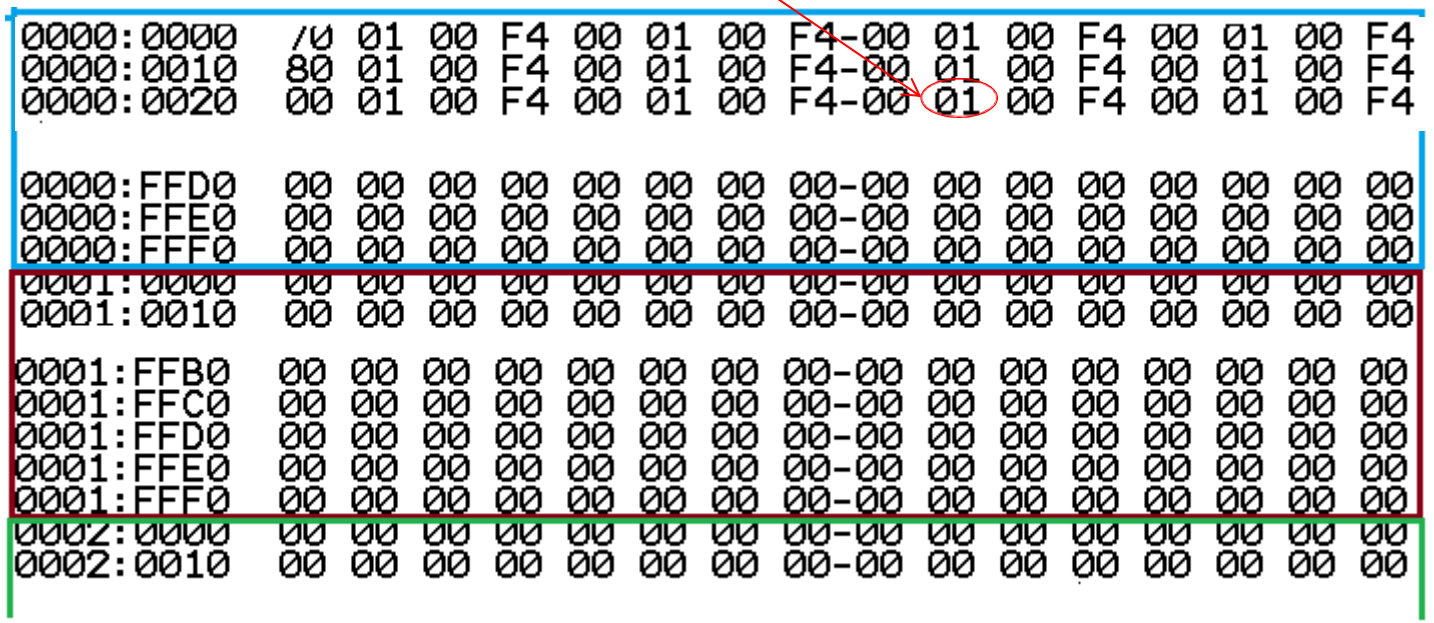
De 0000:0000
à 0000:FFFF

Segment 0001

De 0001:0000
à 0001:FFFF

Segment 0003

De 0002:0000
à 0002:FFFF



Quelle est l' adresse de l'octet cerclé de rouge ? 0000:002A

Mémoire programme, données

Données

Registres Segments

Programme