1. Format de représentation

- taille d'un mot : nb de bit utilisé par un ordinateur pour stocker un entier
- La taille d'un mot limite la taille max de la memoire d'un ordinateur.
- Exemple : Si l'ordi utilise des mots de 32 bits $\Rightarrow$ memoire $= 2^{32}$ bytes $= 4$ Gb
  64 bits $\Rightarrow$ memoire $= 2^{64}$ bytes $= 16$ Eib

2 Ordonnance des bits

Big endians : le mot commence par le bit de poids fort
Little endians : le mot commence par le bit de poids faible

Exemple :

addr	0x100	0x101	0x102	0x103
Big endians	01	23	45	67
Little endians	67	45	23	01

Les ordonnances des bits sont importants lors de:
- la transmission de données
- le debugging d'un programme en assembleur
- traitement de données à bas niveau

3. Representation de texte

- Chaque caractère utilise une séquence de bit définie dans un code.
  Le code le plus utilisé est l'ASCII
- ASCII utilise 7 bits par caractère
- Unicode utilise 16 bits par caractère

4. Representation des images

Deux grandes catégories :

1. Technique de bitmap
2. Technique vectorielle

Bitmap :

- Collection de pixel.
- En noir/blanc : chaque pixel est à 1 ou 0
- En couleur : chaque pixel est représenté par une séquence de bit (RGB)
- Soucis principal : changement d'echelle qui rend l'image pixelisé

5. Représentation des sons

- La methode la plus courante est l'échantillonage à interval regulier
- D'autres methodes existent qui permettent de parametrer l'encodage

Example : .wav propose les options suivantes:

- Num channels : # canaux (mono, stéréo)
- Sample rate : freq d'échantillonage (Hertz)
- ByteRate : # byte à lire par seconde
- ByteBloc : # byte par echantillonage
- BytePerSample : # bit pour l'encodage de chaque echantillon

II. Liens entre le matériel et l'application

1. Niveaux d'abstraction en informatique

1. App
2. Algo
3. language de haut niveau
4. OS
5. architecture de la machine
6. micro architecture
7. Circuit logique
8. dispositif electronique

2. Réalité vs Abstraction

L'abstraction est nécessaire mais ne pas oublier la réalité

Exemple: Est ce que $x^2 \geq 0$?

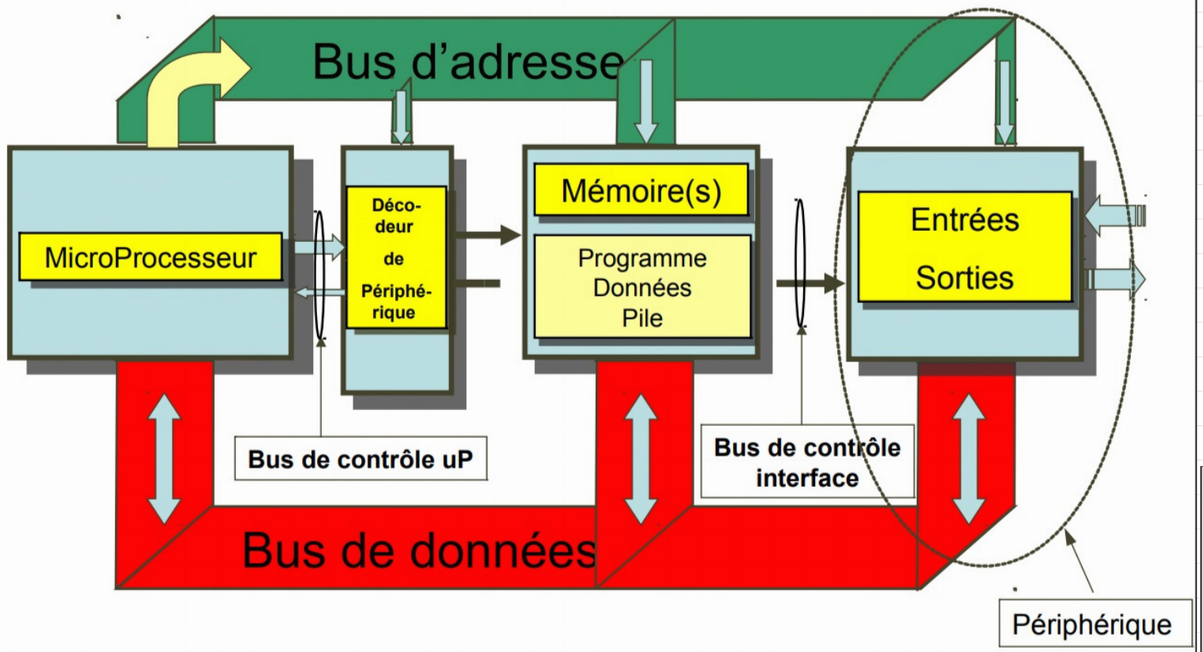
Pour les floats oui

Pour les entiers ?

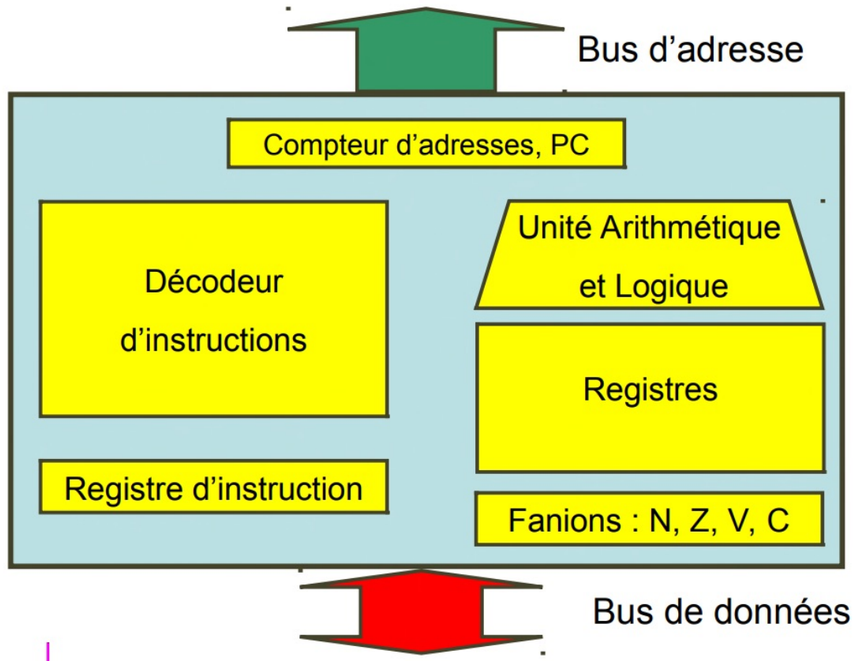
$40'000 * 40'000 = 16 \cdot 10^8$

$50'000 * 50'000 = ??$

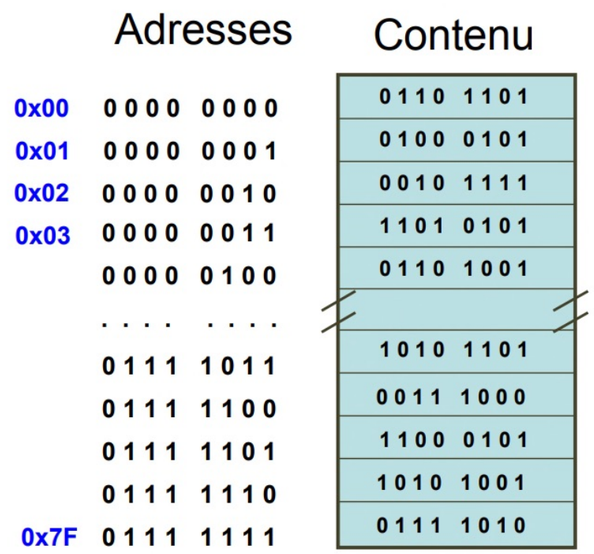
Architecture d'un système informatique



Modèle simple d'un processeur



Modèle d'une mémoire



Programme en assembleur

```
0x00000100      mov R0, #2
0x00000104      ldr R1, [R4]
0x00000108      add R2, R0, R1
0x0000010C      and R2, R2, #15
0x00000110      str R2, [R4]
0x00000114      mov R2, #3

  debut_for:
0x00000118      ldr R0, [R3]
0x0000011C      add R0, R0, R1
0x00000120      str R0, [R3]
0x00000124      add R3, R3, #4
0x00000128      subs R2, R2, #1
0x0000012C      bne debut_for
0x00000130      mul R2, R1, R0
```

Etat initial

R0 : 1234

R1 : 5678

R2 : 0x00000002

R3 : 0x40001000

R4 : 0xE0028008

Contenu mémoire

0x40001000 0x00003F00

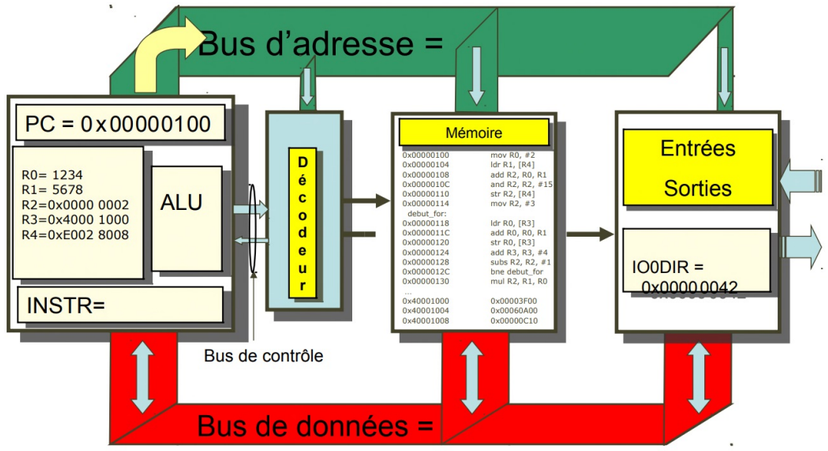
0x40001004 0x00060A00

0x40001008 0x00000C10

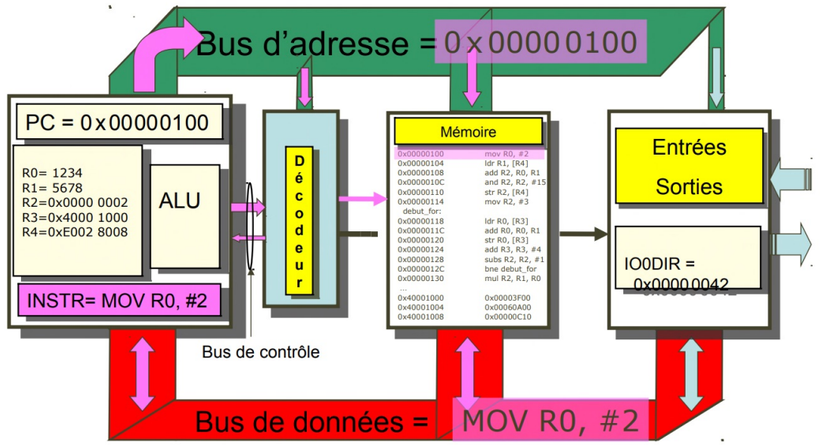
IO0DIR = 0x00000042

(le registre IO0DIR se trouve à l'adresse 0xE0028008)

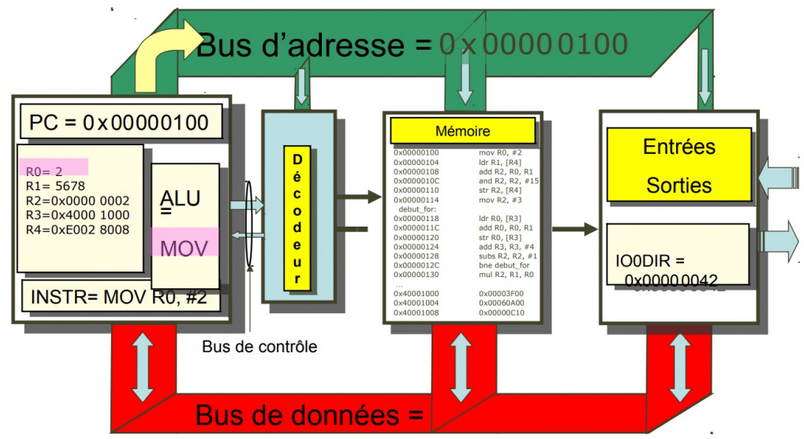
Architecture: État initial



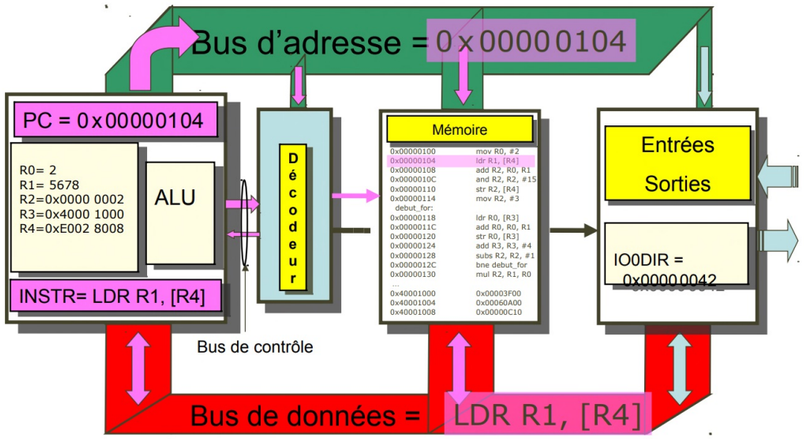
Chargement première instruction



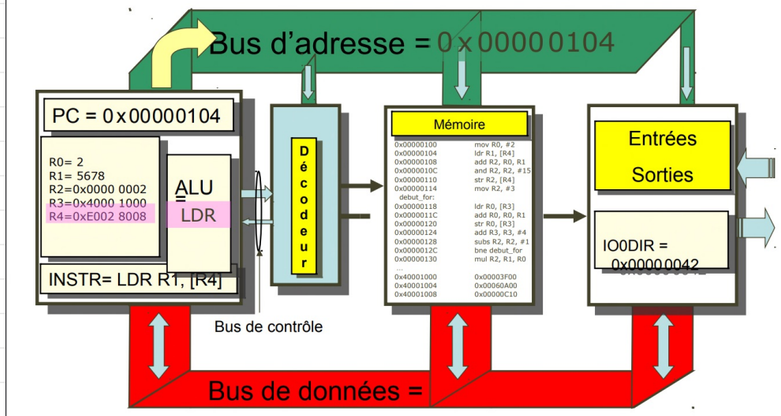
Décodage et exécution première instruction



Chargement seconde instruction



Décodage seconde instruction



Chargement IO0DIR --> R1