

# Architecture des Ordinateurs

Vue d'Ensemble

Fabrice BOISSIER – [fabrice.boissier@epita.fr](mailto:fabrice.boissier@epita.fr)



# Architecture des Ordinateurs

- Objectifs :
  - *Découvrir ce qu'est l'Architecture des Ordinateurs*
  - Connaître les composants d'un ordinateur
  - Comprendre le format des données manipulées par un processeur
- N'apprenez pas par cœur les schémas de cette séance...  
*(certains sont simplifiés pour mieux comprendre les concepts)*
- ...cette séance sert à vous donner un aperçu de ce cours et des enjeux de l'ensemble des cours touchant au bas niveau

# Les « Ordinateurs » : Historique

- Un peu plus que des calculatrices
  - Ne fait pas que du calcul
- Suite de l'évolution technique dans l'industrie
  - Machines mécaniques / vapeur & transmission de force
  - Machines électro-mécaniques
  - Machines électroniques
- Usages des ordinateurs à l'époque :
  - Gestion & Recherche

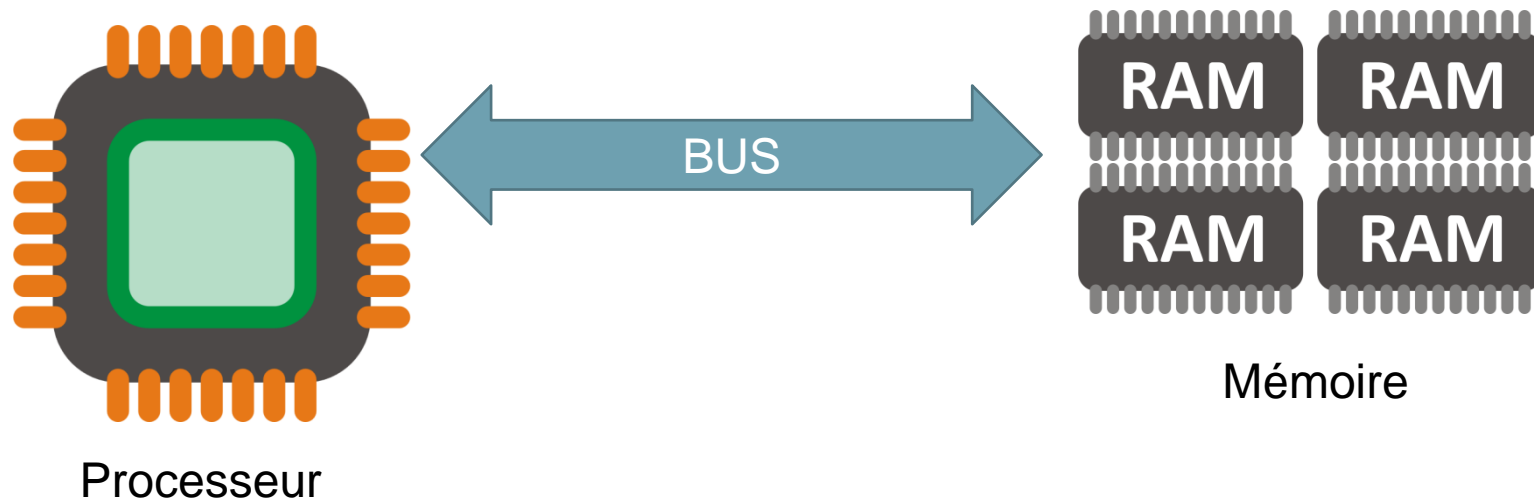
# Les « Ordinateurs » : Historique

*[Voir vidéo SICOB / M.Patron]*

# Les « Ordinateurs » : Définition

- Mot « **Ordinateur** » a été introduit en **1955** par **François Girard** chez **IBM France**
- Définition Wikipédia :  
*« Un ordinateur est une machine électronique qui fonctionne par la lecture séquentielle d'un ensemble d'instructions, organisées en programmes, qui lui font exécuter des opérations logiques et arithmétiques sur des chiffres binaires. »*

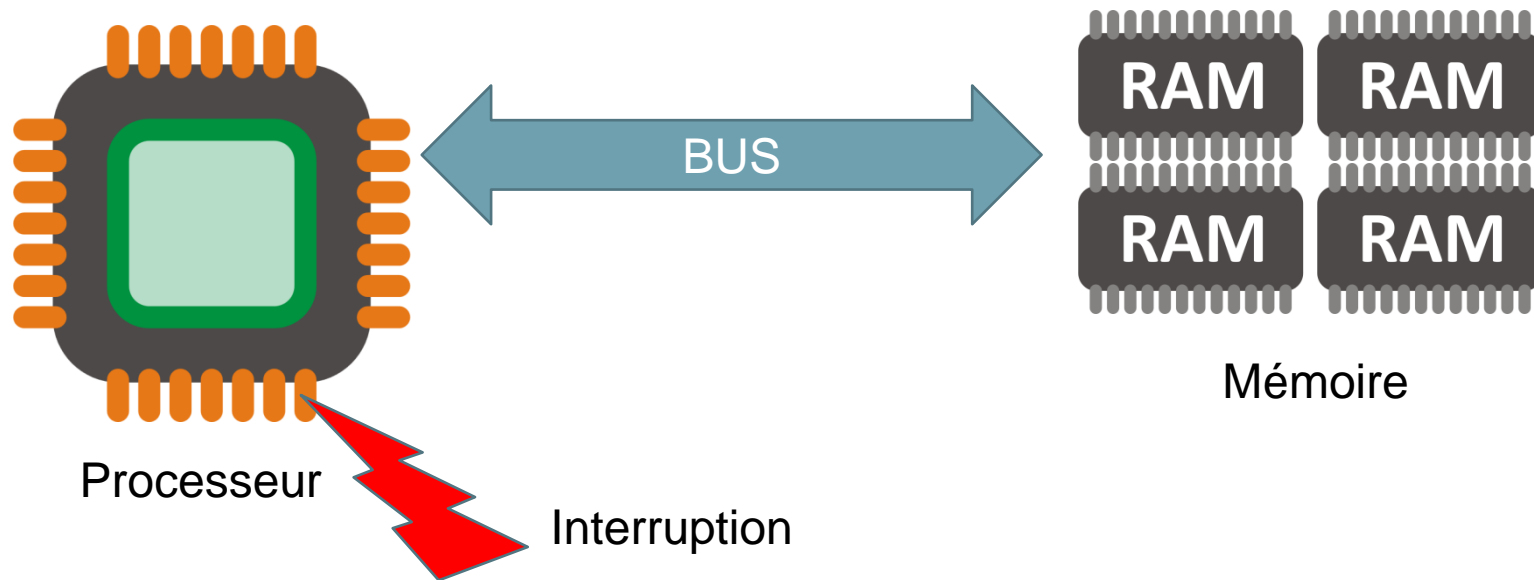
# Les « Ordinateurs »



On sait maintenant compter, déplacer des valeurs, en respectant les ordres

Mais on se sent vite seul...

# Les « Ordinateurs »

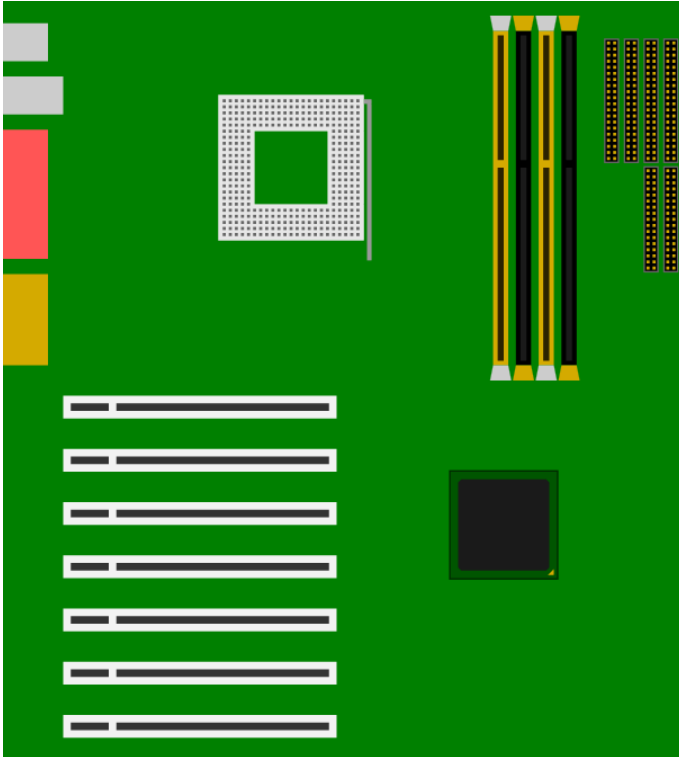


On sait maintenant compter, déplacer des valeurs, en respectant les ordres

On peut appeler quelqu'un d'autre, et se faire appeler aussi

Mais on se sent vite seul...

# Les « Ordinateurs »



La « carte mère » (ou n'importe quelle board)  
offre des services au processeur :

Pleins d'interruptions...

« Coucou CPU ! Ici réseau !

- Et clavier !

- Et souris !

- Et... »

Simplification de certaines situations

« J'ai fini de copier la mémoire »

FSB aligne tout le monde

Mais elle limite aussi l'ordinateur :

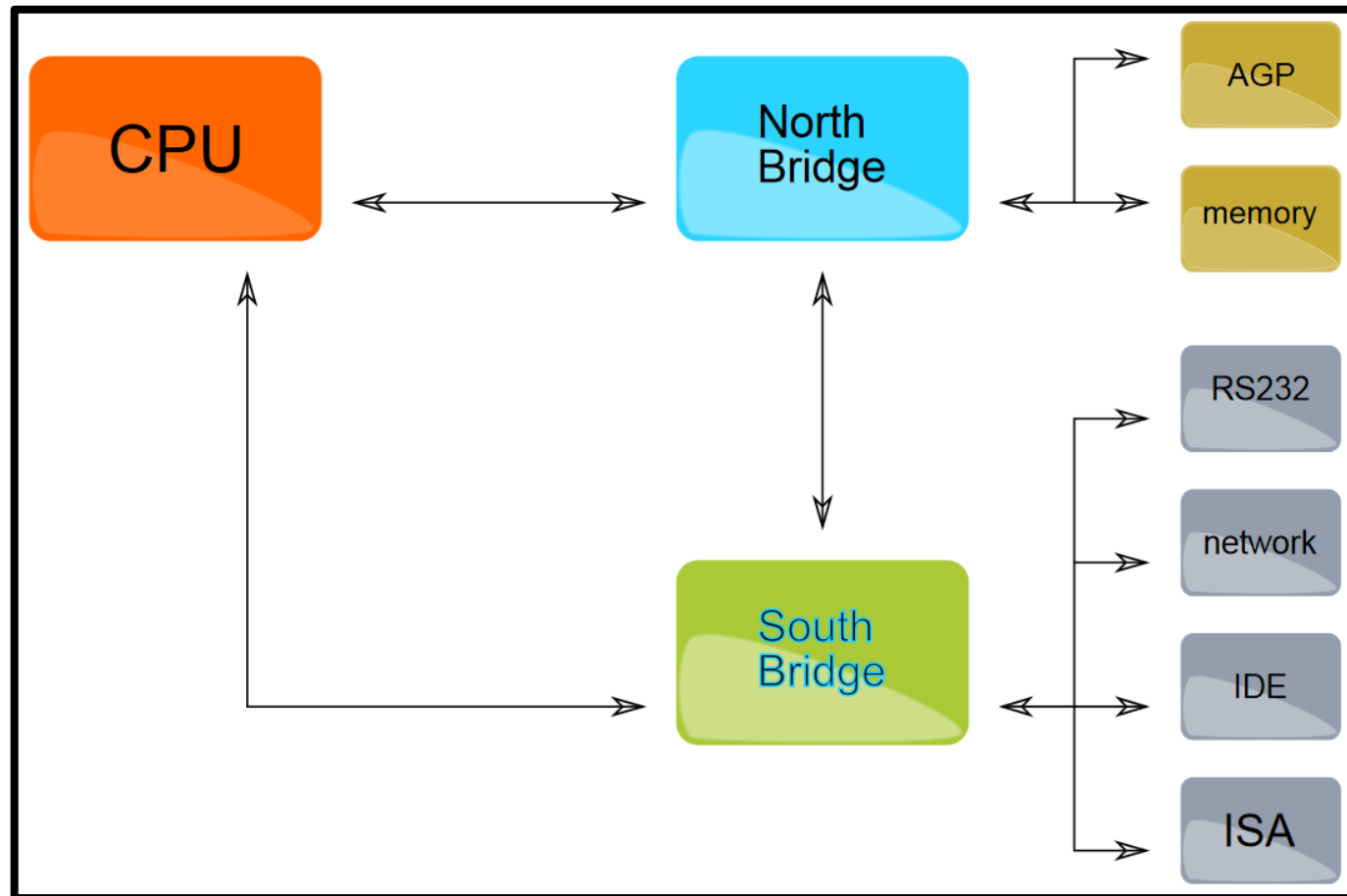
Nombre max de processeurs

Max de mémoire

FSB limite tout le monde...



# Les « Ordinateurs »



Standard IBM-PC  
(IBM 5150)

Basé sur x86

A évolué !

Offre beaucoup de  
canaux de  
communication

Pas cher...

1 processeur au  
centre de tout

Micro Informatique

# Les « Ordinateurs » : Classes historiques

- Évolution au fil du temps des ordinateurs :
  - Mainframes
  - Super-Calculateurs / Super-Ordinateurs
  - Minis
  - Micros

# Les « Ordinateurs » : Classes historiques

- Mainframes

- IBM (z System), Bull (GCOS), Fujitsu, HP, ...
- Dédié I/O et très forte charge
- Un seul référentiel

- Minis

- IBM : System i (AS/400), System p (POWER)
- DEC : VAX (VMS), PDP, ... *[disparu]*
- Plus récemment : HP, Sun/Oracle, SGI, ...
- Innombrables UNIX (AIX, HP-UX, Solaris, ...)
- Se relie les uns aux autres

- Micros

- PC (5150 et « compatibles PC »)
- Atari ST, Commodore, ...
- Bureautique, jeux : graphique

- Super-Ordinateurs  
Super-Calculateurs

- Cray, Blue Gene, ...
- POWER, Xeon, SPARC64, Opteron, GPU NVidia, ...
- Calcul ! Pas I/O !  
Effets d'une bombe atomique, simulation de neurones, ...
- Multiples machines reliées, PS3 en série, ...

# Les « Ordinateurs » : Micro-Ordinateurs



# Les « Ordinateurs » : Mini-Ordinateurs



# Les « Ordinateurs » : Mainframes





# Les « Ordinateurs » : Super-Ordinateurs



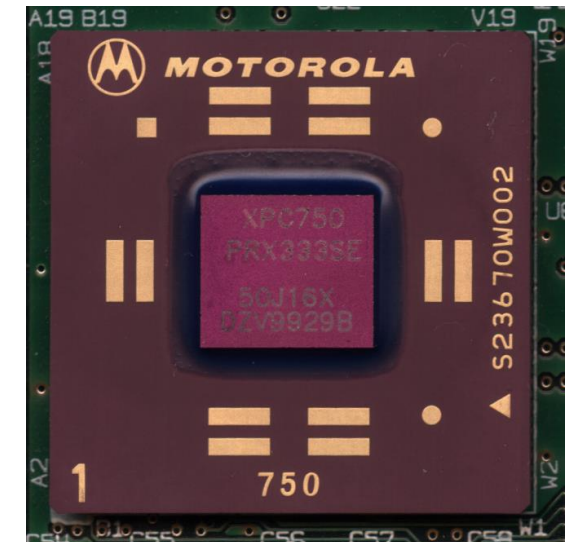
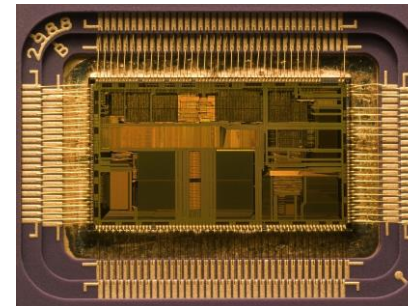
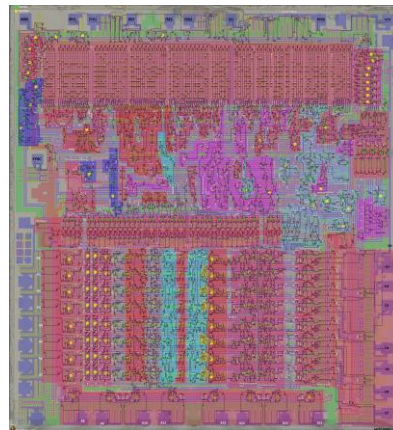
# Les « Ordinateurs » : Classes actuelles

- Aujourd'hui on distingue plutôt :
  - Super-Calculateurs / HPC
  - Serveurs
  - Ordinateurs fixes / portables
  - Smartphones / Tablettes / SmartTV / ... *[appareils personnels]*
  - Embarqué & Temps Réel & IoT *[appareils non accessibles]*



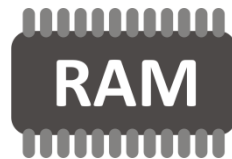
# Composants d'un ordinateur

- Processeur :
  - Exécute des instructions (en *langage machine*)
  - Accède à la mémoire
  - Envoie des signaux (interruptions) aux autres composants
  - Se programme avec du code écrit en langage *assembleur*

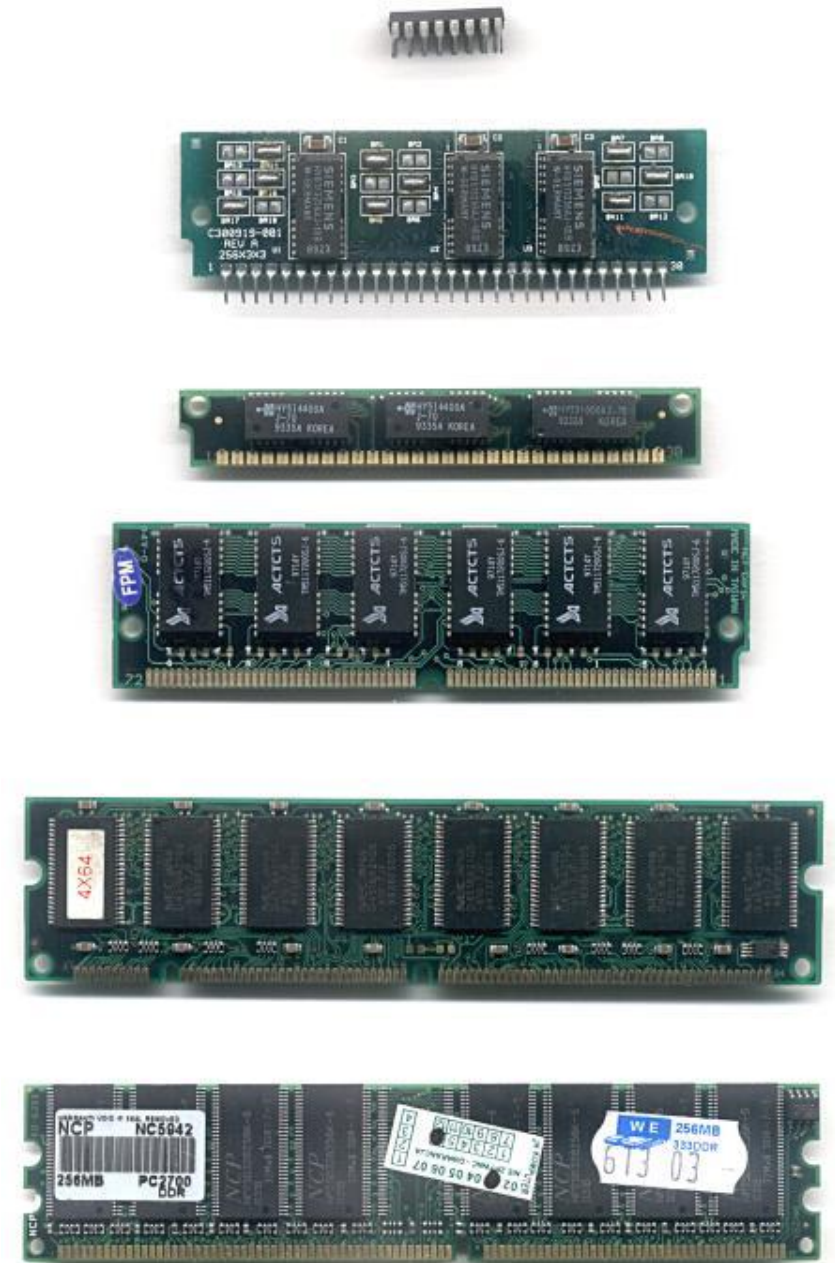
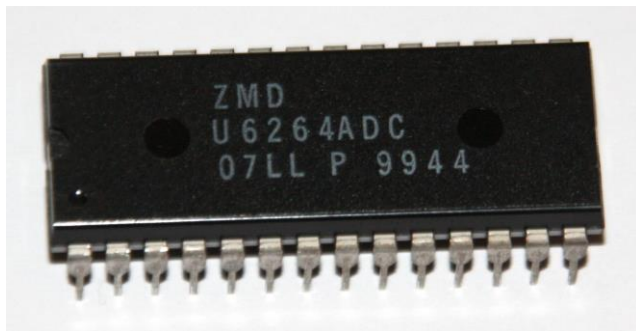


(merci Wikipedia pour les images)

# Composants d'un ordinateur



- Mémoire :
  - Contient des adresses (@)
  - Chaque adresse contient de la donnée

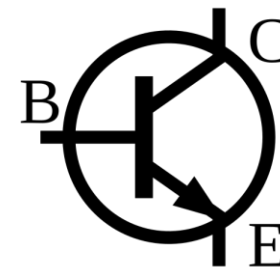
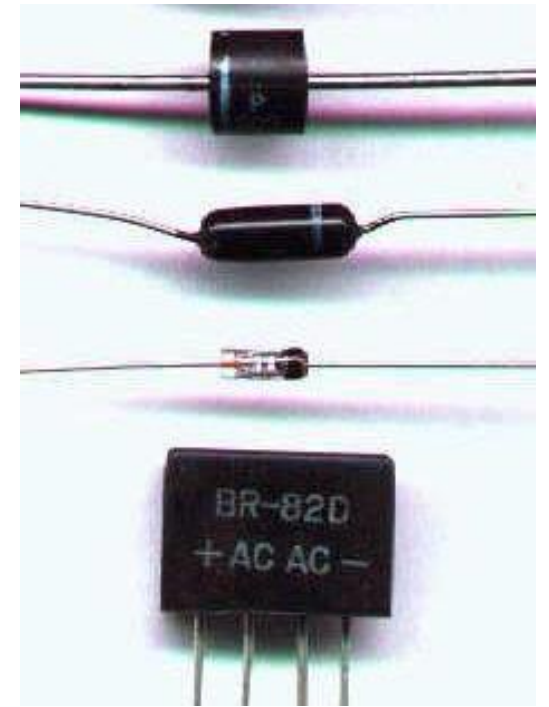
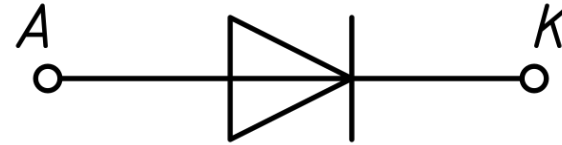


(merci Wikipedia pour les images)



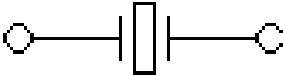
# Autres composants

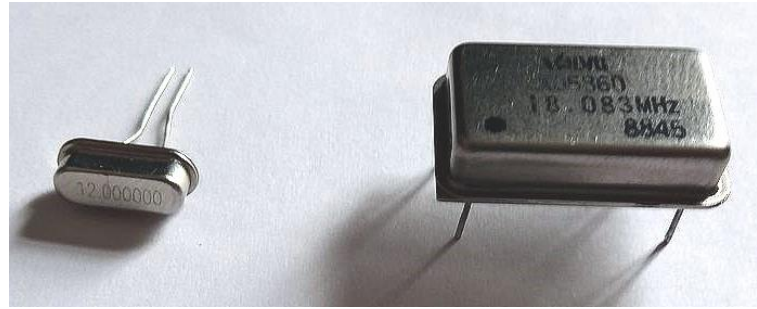
- Diode :
  - Composant électronique semi-conducteur dipôle
  - Permet de ne laisser passer le courant que dans un sens
- Transistor :
  - Composant électronique semi-conducteur à 3 électrodes
  - Permet de faire varier la sortie selon ses entrées
  - Grande diversité de types de transistors
  - Est la base de toutes les sondes/capteurs



(merci Wikipedia pour les images)

# Autres composants

- Quartz : 
  - Composant électronique générant des signaux réguliers
  - Permet de synchroniser tous les composants
- Microcontrôleur :
  - Composant électronique contenant un processeur et de la mémoire
  - Processeur simple et petite mémoire
  - Parfait pour contrôler quelques composants et faire un petit appareil
- Périphérique :
  - Assemblage de processeurs, mémoires, microcontrôleurs, etc...

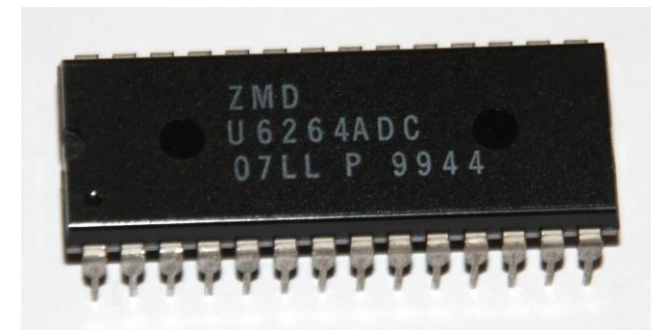
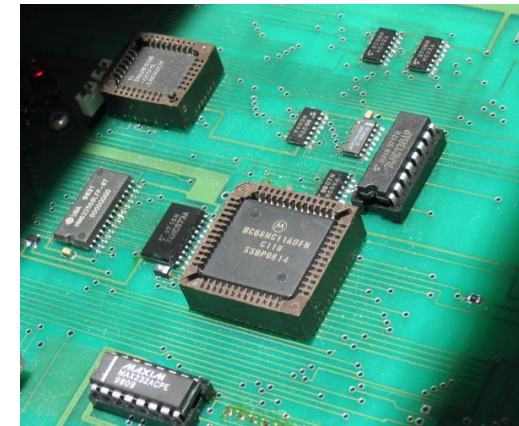


(merci Wikipedia pour les images)

# Autres composants

- Les « puces électroniques » (*chip* en anglais) ou « circuits intégrés » (*integrated circuits* en anglais) sont simplement un emballage miniaturisé d'un ou plusieurs composants

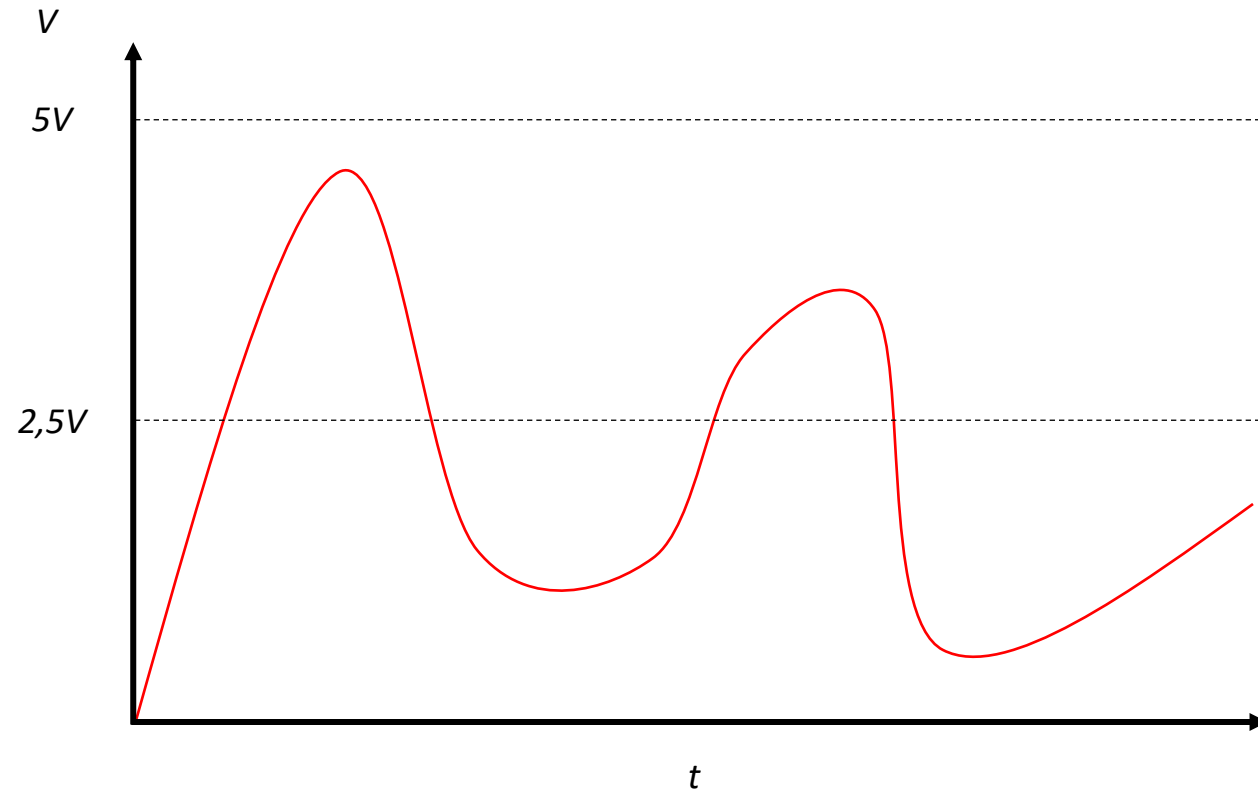
- Processeur
- Mémoire
- Microcontrôleur
- Portes logiques
- Amplificateurs
- ...



(merci Wikipedia pour les images)

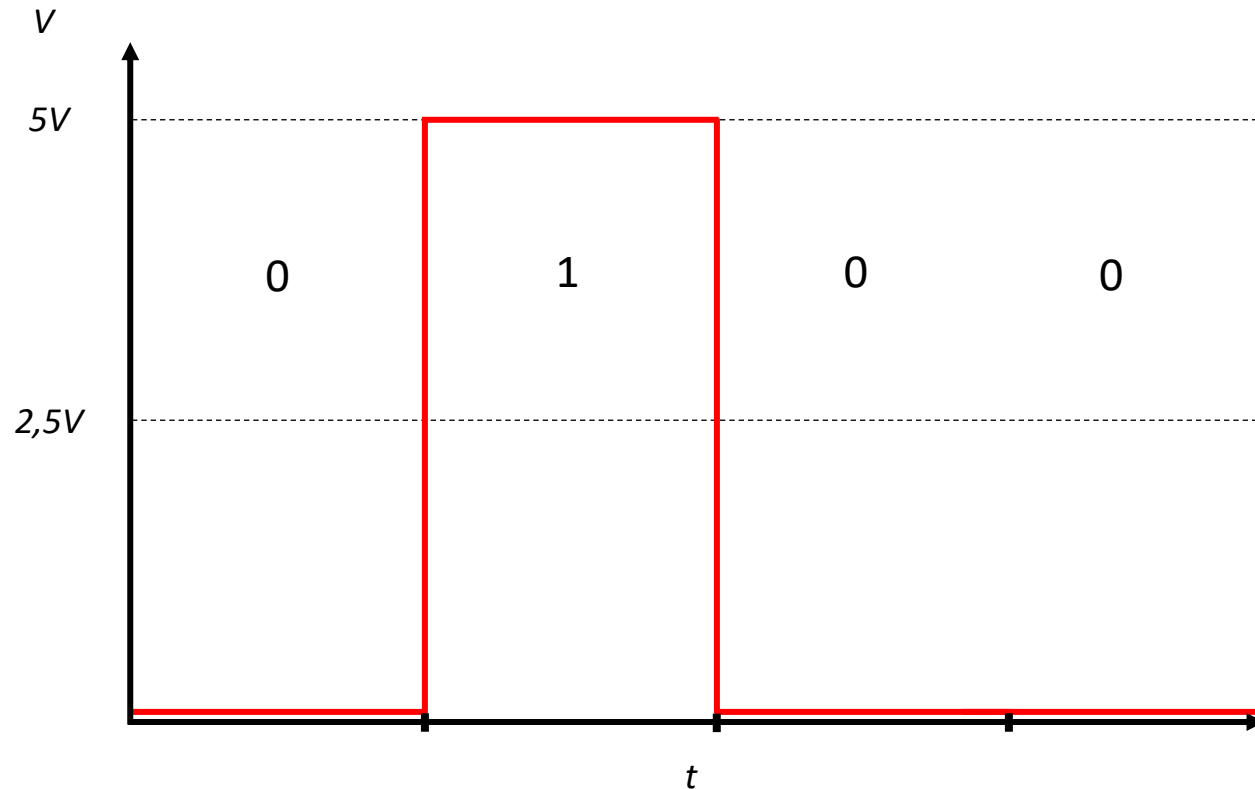
# Signaux analogiques

- Signal Analogique (*Analog Signal*) : signal = grandeur



# Signaux numériques

- Signal Numérique (*Digital Signal*) : signal = bits



# Signaux

- Il existe des composants « analogiques » ou « numériques » (ou les 2)
  - Ex : les DSP sont des processeurs manipulant des signaux
- Dans le cadre de l'informatique, on s'intéresse surtout aux composants numériques...
- ...mais les sondes émettant certaines données peuvent être analogiques (capteur de lumière, micro, ...)
- Ce sont les transistors qui contribuent à transformer ces signaux analogiques en numériques



# Standards / Définitions du numérique

- 1 bit peut prendre 2 valeurs :
  - 0
  - 1
  - (binaire)
- 8 bits = 1 octet = 1 Byte
- 1 octet peut prendre 256 valeurs ( $2^8$ ) :
  - de 0 à 255 s'il est « non-signé »
  - de -128 à +127 s'il est « signé »

# Standards / Définitions du numérique

|                             |                |                   |
|-----------------------------|----------------|-------------------|
| • Kilo octet / KiloByte     | = 1000 o / B   | = $10^3$ o / B    |
| • Méga octet / MegaByte     | = 1000 ko / kB | = $10^6$ o / B    |
| • Giga octet / GigaByte     | = 1000 Mo / MB | = $10^9$ o / B    |
| • Téra octet / TeraByte     | = 1000 Go / GB | = $10^{12}$ o / B |
| • Péta octet / PetaByte     | = 1000 To / TB | = $10^{15}$ o / B |
| • Exa octet / ExaByte       | = 1000 Po / PB | = $10^{18}$ o / B |
| • Zetta octet / ZettaByte   | = 1000 Eo / EB | = $10^{21}$ o / B |
| • Yotta octet / YottaByte   | = 1000 Zo / ZB | = $10^{24}$ o / B |
| • Ronna octet / RonnaByte   | = 1000 Yo / YB | = $10^{27}$ o / B |
| • Quetta octet / QuettaByte | = 1000 Ro / RB | = $10^{30}$ o / B |

# Standards / Définitions du numérique

- Ne confondez pas les préfixes du SI (en décimaux)...
  - Kilo (k), Méga (M), Giga (G), Téra (T), ...
  - 1.000 , 1.000.000 , 1.000.000.000 , 1.000.000.000.000 , ...
- Avec les préfixes binaires !
  - Kibi (Ki), Mébi (Mi), Gibi (Gi), Tébi (Ti), ...
  - 1.024 ( $2^{10}$ ) , 1.048.576 ( $2^{20}$ ) , 1.073.741.824 ( $2^{30}$ ) ,  $2^{40}$  , ...
  - ..., Pébi (Pi), Exbi (Ei), Zébi (Zi), Yobi (Yi)
  - ...,  $2^{50}$  ,  $2^{60}$  ,  $2^{70}$  ,  $2^{80}$

# La Mémoire

- Circuit intégré
  - Dispose d'un premier bus pour sélectionner une adresse (@)
  - Dispose d'un second bus pour transférer la donnée vers/depuis l'extérieur de la mémoire
- Volatilité
  - Volatile : perte des données en cas de coupure de courant
  - Non-Volatile : données conservées hors tension
- Attention, la mémoire n'est pas faite pour le « stockage »
  - Disque dur, bande magnétique, Blu-Ray, flash, ...

# La Mémoire

- RAM
  - *Random Access Memory* (ou mémoire vive)
  - Lecture/Écriture une fois mise sous tension
  - Généralement volatile
  - Exemples : DRAM (FPM, EDO, SDRAM, DDR, ...), SRAM, ...
- ROM
  - *Read-Only Memory* (ou mémoire morte)
  - Modifiable uniquement dans certaines conditions (voire pas du tout)
  - Généralement non-volatile
  - ROM, PROM, EPROM, EEPROM, ...

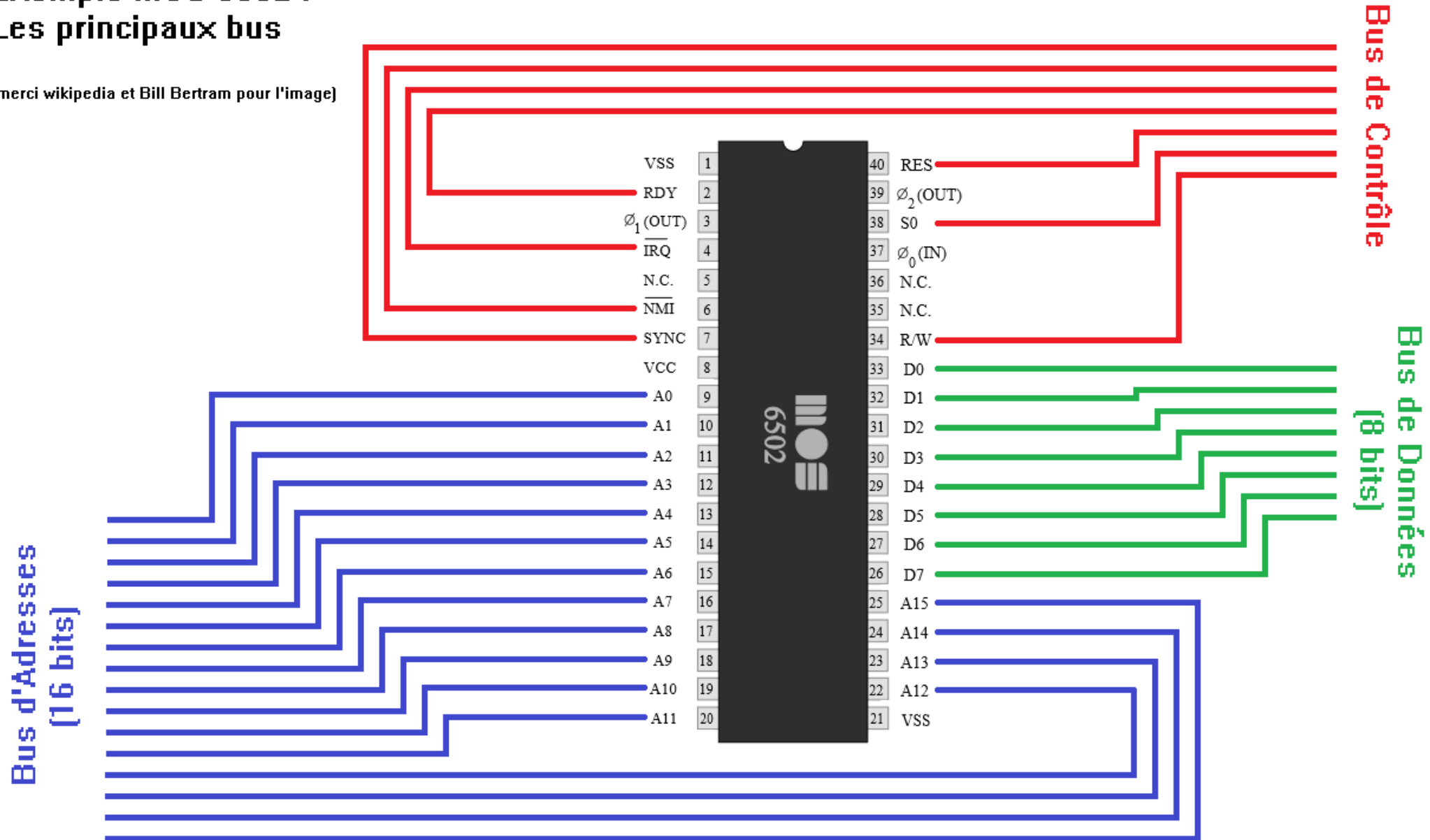
# Les 3 Bus

- Bus d'adresse
  - Sélectionne une adresse
- Bus de données
  - Écrit un mot
  - Lit un mot
- Bus de contrôle
  - Gestion des interruptions
  - Gestion des composants externes

# Les 3 Bus

## Exemple MOS 6502 : Les principaux bus

(merci wikipedia et Bill Bertram pour l'image)



# La taille des bus

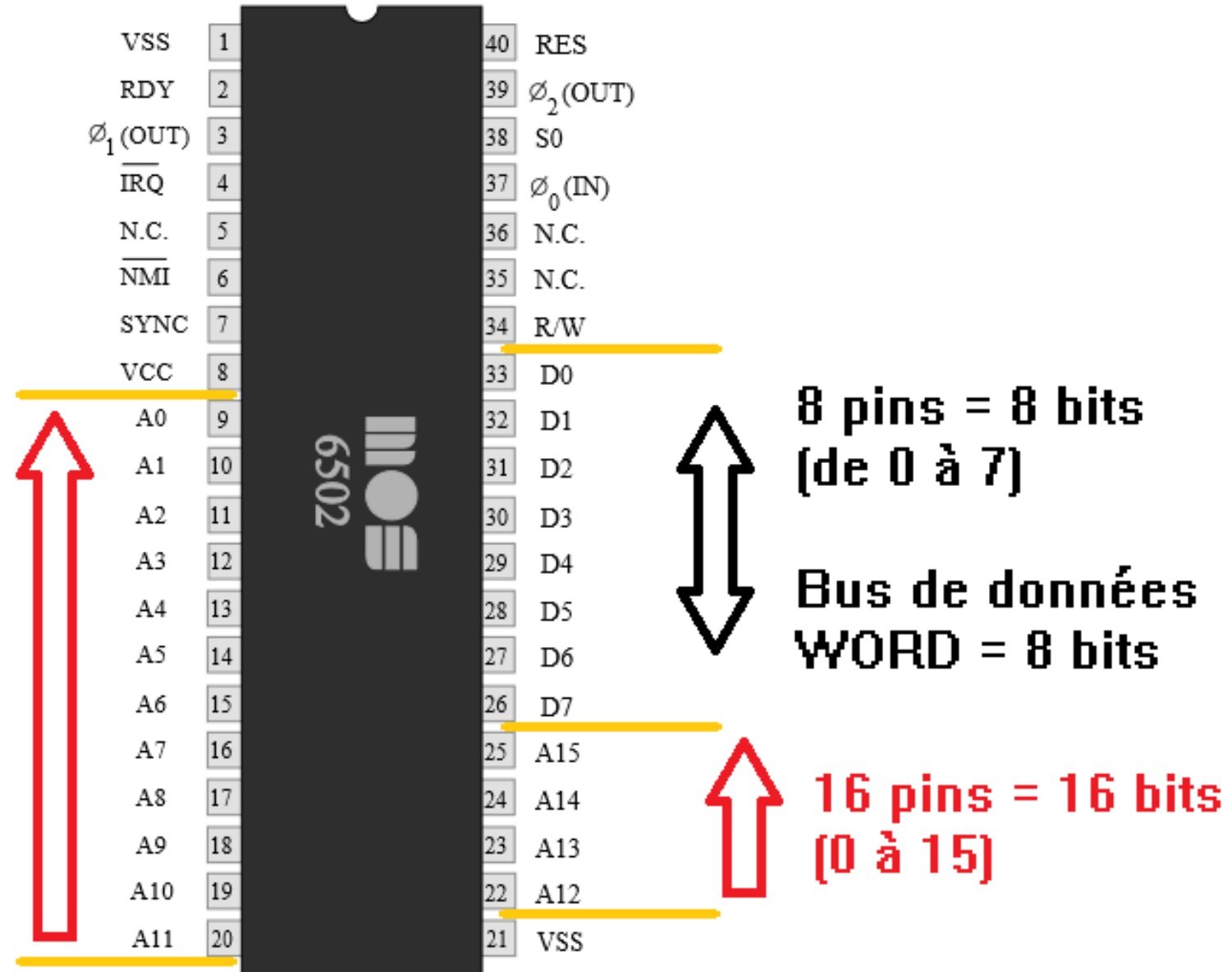
Exemple MOS 6502 :

(merci wikipedia et Bill Bertram pour l'image)

Taille des bus

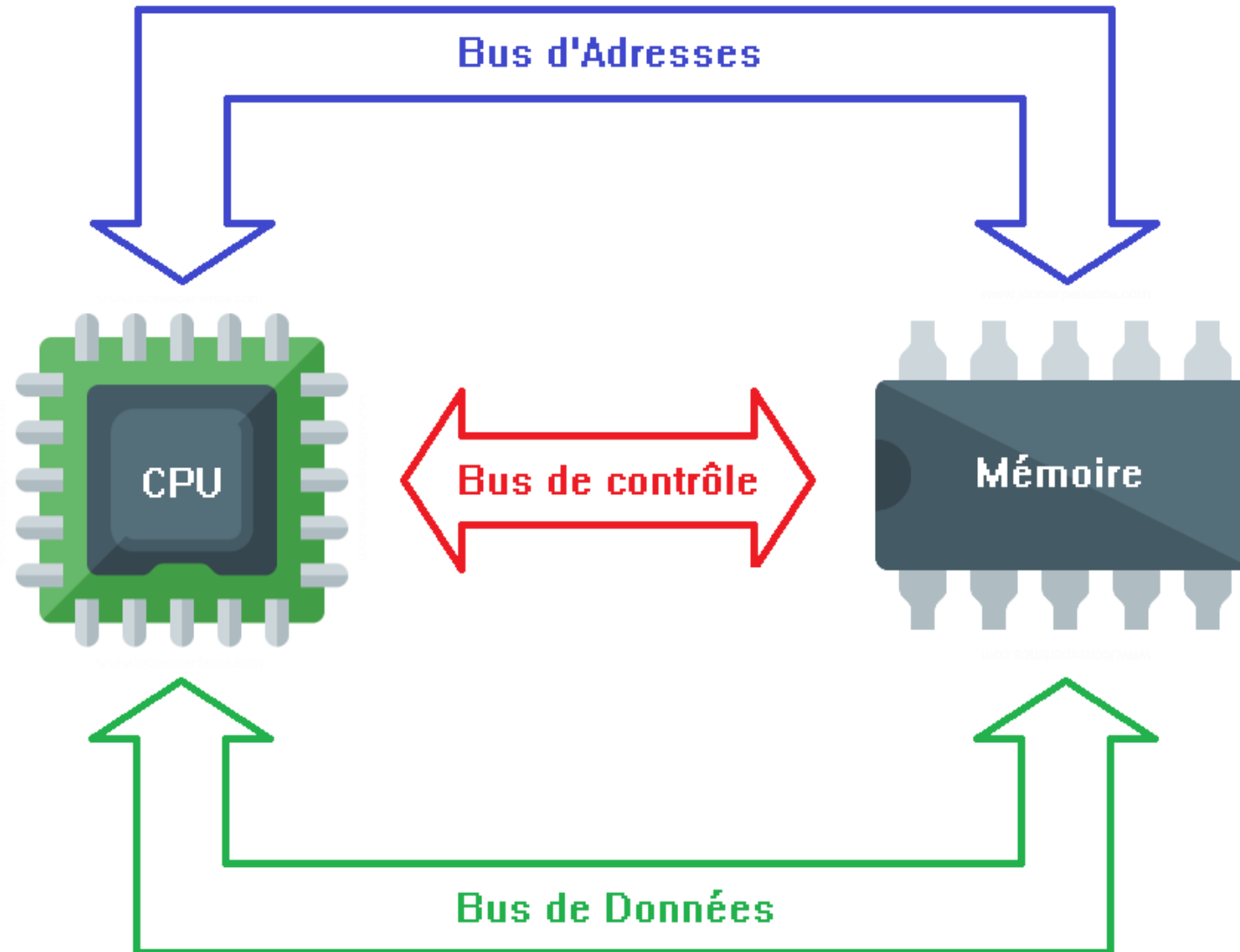
Bus d'@ de  
taille 16 bits

Soit 65.536  
adresses  
accessibles





# La Mémoire



# La Mémoire

- Lecture :

- 1 - Le processeur met sur le bus d'@, l'adresse qu'il souhaite atteindre

- 1 - Le processeur indique sur le bus contrôle qu'il souhaite lire (READ)

- 2 - La mémoire charge l'adresse et récupère la donnée

- 3 - Le processeur indique sur le bus de contrôle qu'il est prêt à recevoir la donnée

- 4 - La mémoire prend la donnée, et l'envoie sur le bus de données

# La Mémoire

- Ecriture :

- 1 - Le processeur met sur le bus d'@, l'adresse qu'il souhaite atteindre

- 1 - Le processeur indique sur le bus contrôle qu'il souhaite écrire (WRITE)

- 2 - La mémoire charge l'adresse

- 3 - Le processeur indique, via le bus de contrôle, qu'il est prêt à émettre la donnée

- 3 - Le processeur charge la donnée sur le bus de données

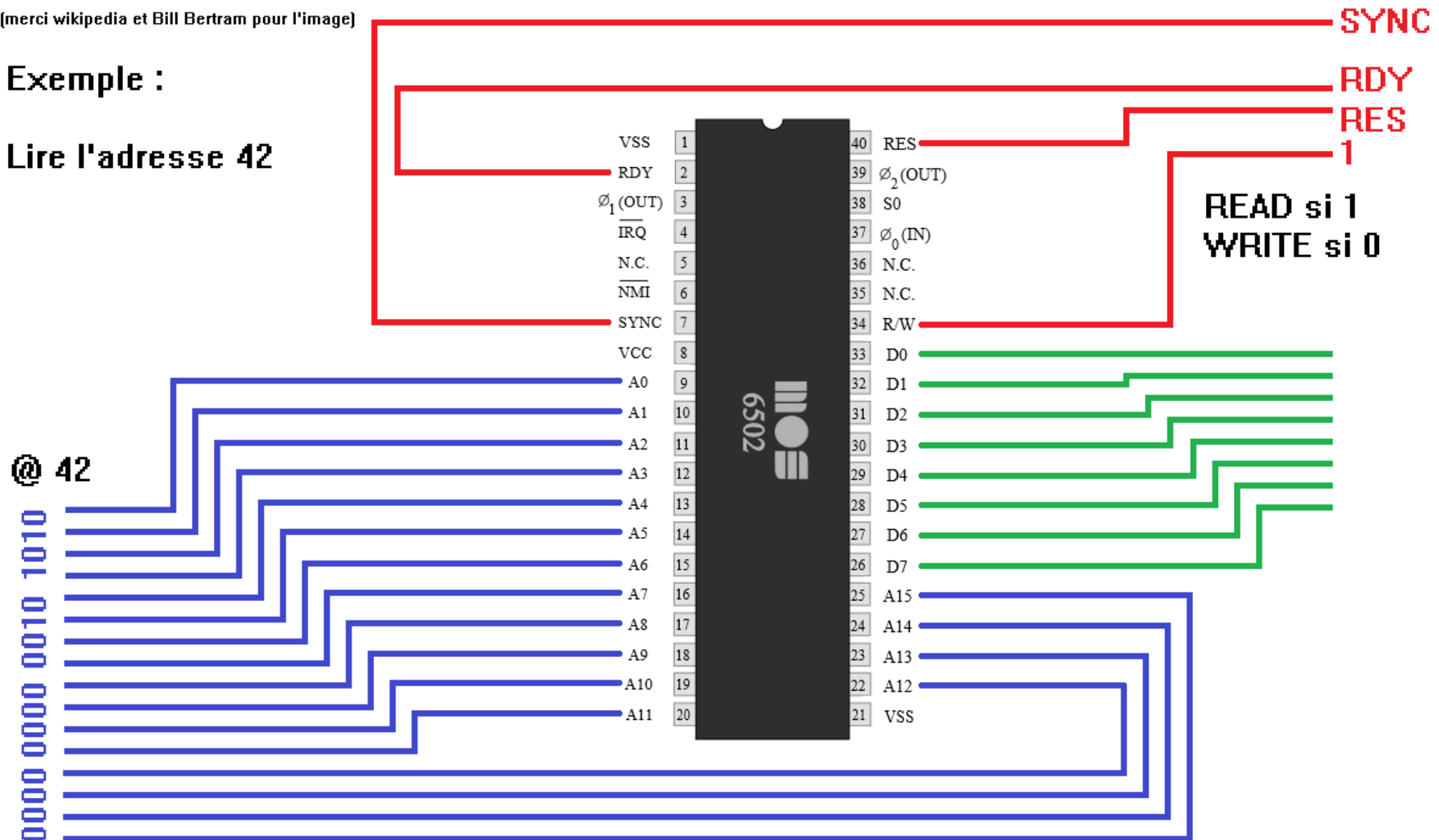
- 4 - La mémoire prend la donnée, et l'écrit à l'adresse indiquée

# Exemple de Lecture en Mémoire

(merci wikipedia et Bill Bertram pour l'image)

Exemple :

Lire l'adresse 42

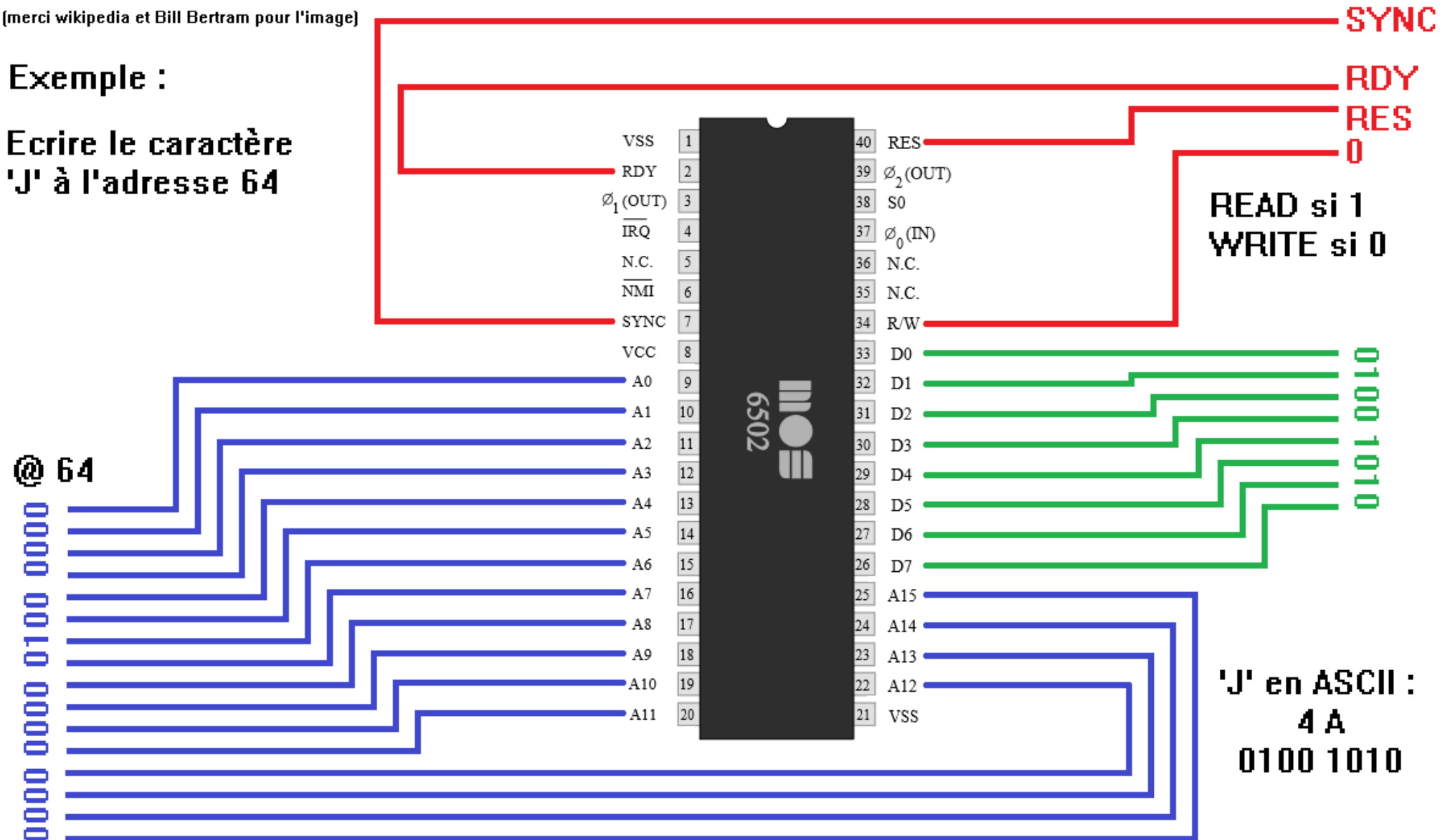


# Exemple d'Écriture en Mémoire

(merci wikipedia et Bill Bertram pour l'image)

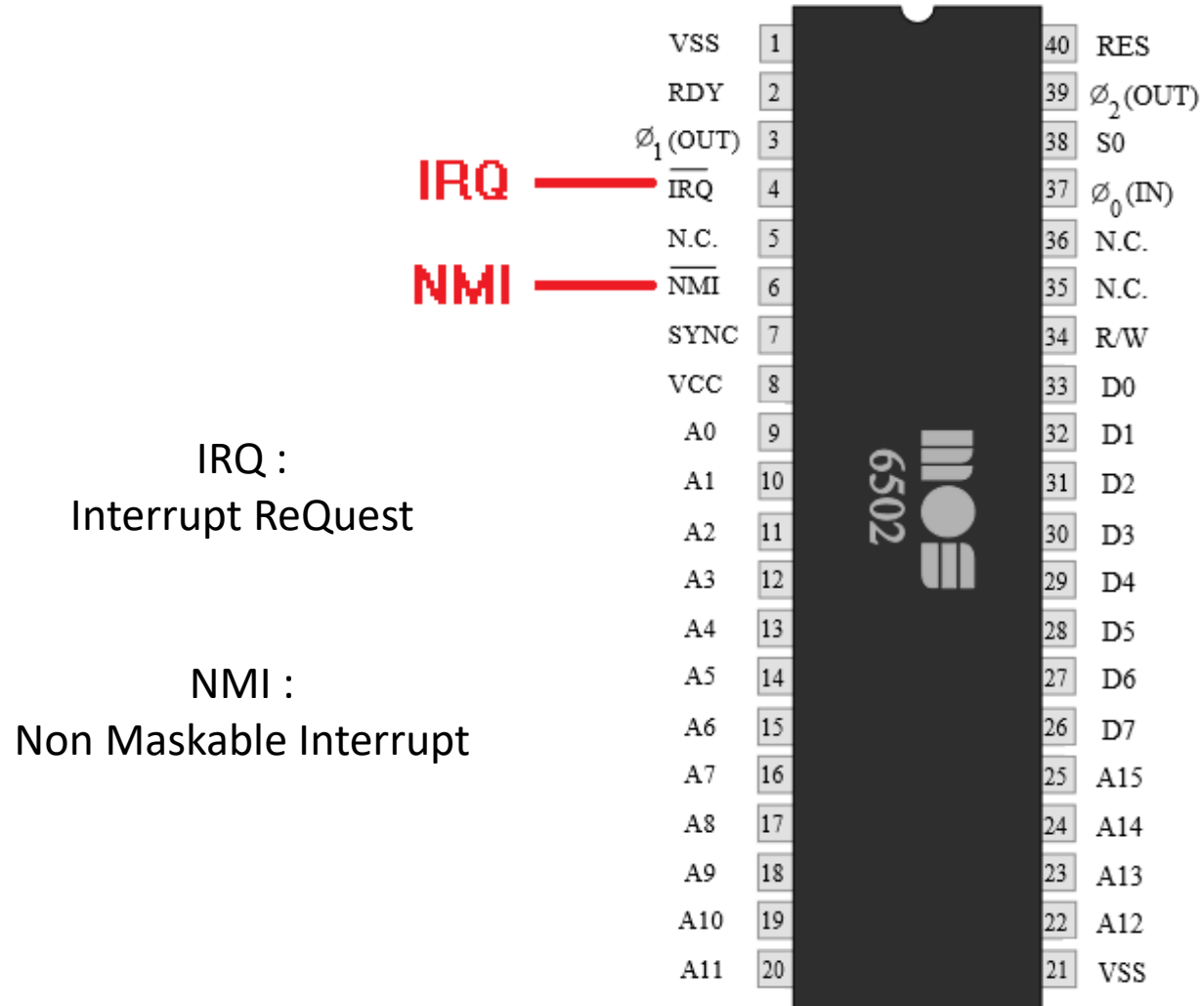
Exemple :

Ecrire le caractère  
'J' à l'adresse 64

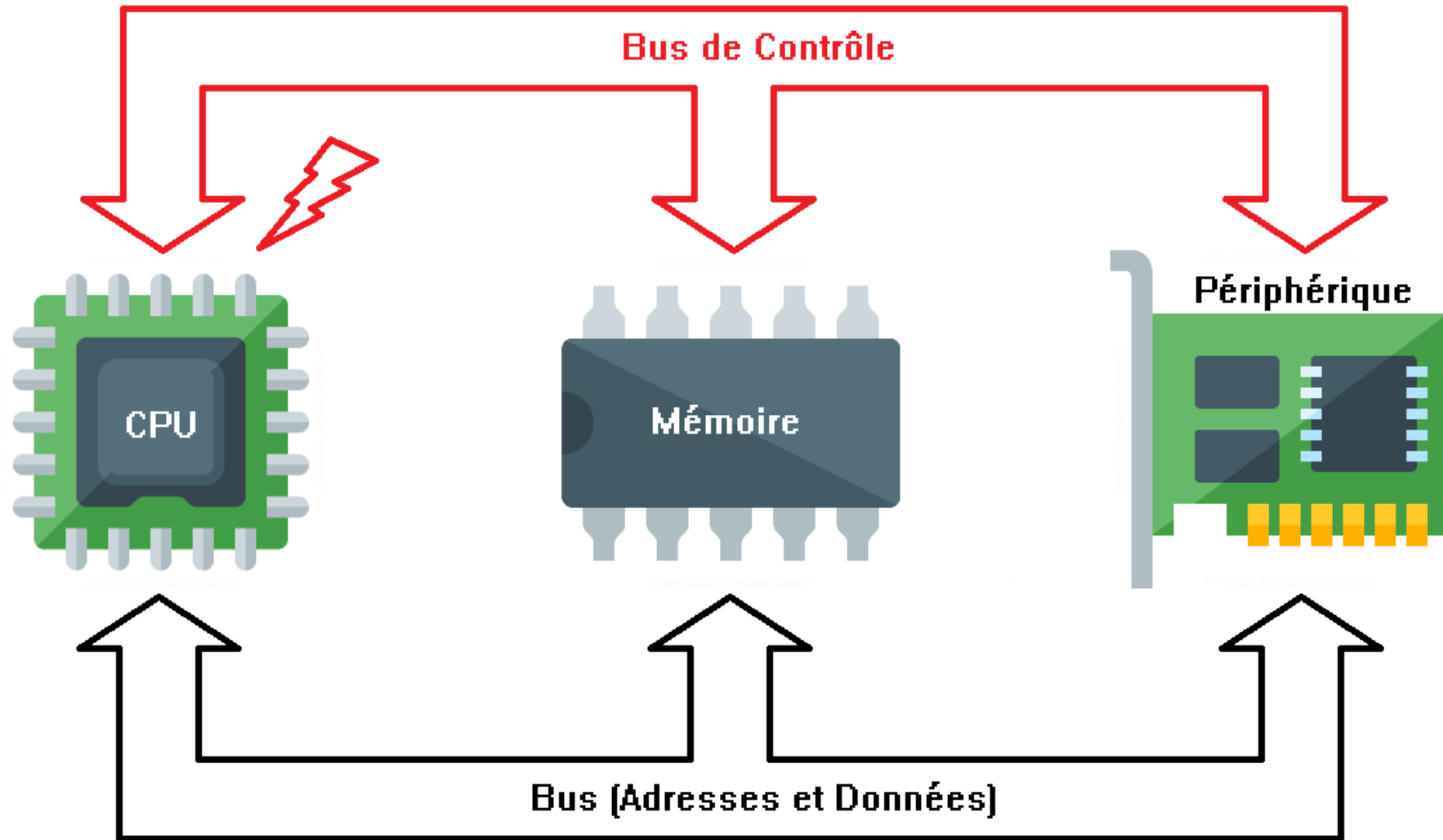


# Les Interruptions

(merci wikipedia et Bill Bertram pour l'image)



# Les Interruptions



# Et maintenant...

- Vous avez maintenant une vision générale simplifiée des ordinateurs
- Ainsi que des composants à l'intérieur
- Et comment ces derniers communiquent
- Il est temps de voir les détails techniques.