
SYS-S5 Rappel Matériel 1/12

FLORIAN CLIQUET
NOTE TAKING OF ONLINE RESOURCES

9 juin 2024



Plagiarism Mention

We attest that the content of this document is original and stems from our personal reflections.

Sommaire

Introduction	4
1 Rappel Matériel	5
1.1 Composants d'un ordinateur	5
1.1.1 Le processeur	6
1.1.2 La mémoire	10
1.1.3 Les entrées/sorties	10
1.1.4 Les bus	13
1.2 L'architecture d'un système à microprocesseur	17
1.2.1 Etapes d'exécution d'un programme	18

Introduction

This document provides an overview of SYS-S5 concepts written by Cliquet Florian. It's a set of notes on multiple online resources.

1 Rappel Matériel

1.1 Composants d'un ordinateur

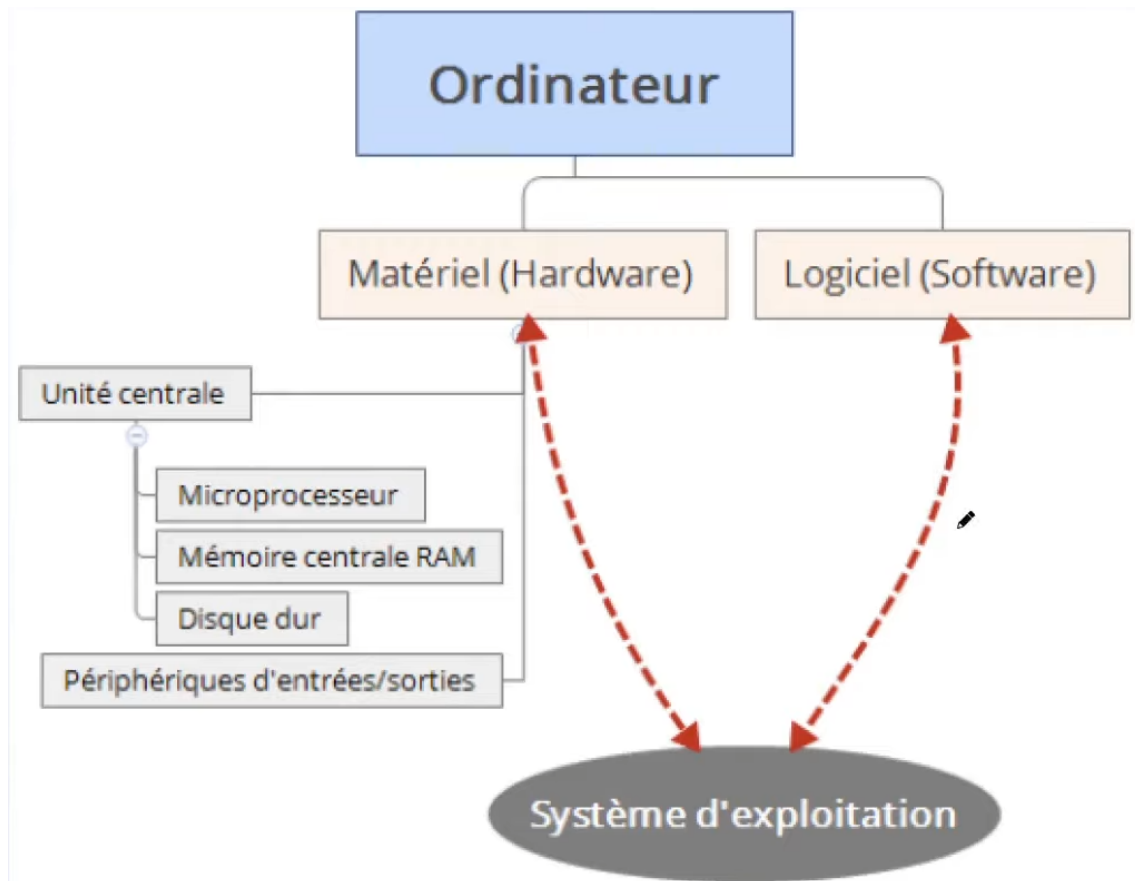


FIGURE 1 – Schéma d'un ordinateur

Le côté matériel (Hardware) parle en binaire (0/1) alors que le logiciel (Software) utilise un langage de programmation plus abstrait. Ici, le système d'exploitation, pour faire simple, sert de "traducteur" entre le matériel et le logiciel.

Il existe différents schémas pour expliquer le fonctionnement de la mémoire, voici celui de **von Neumann** :

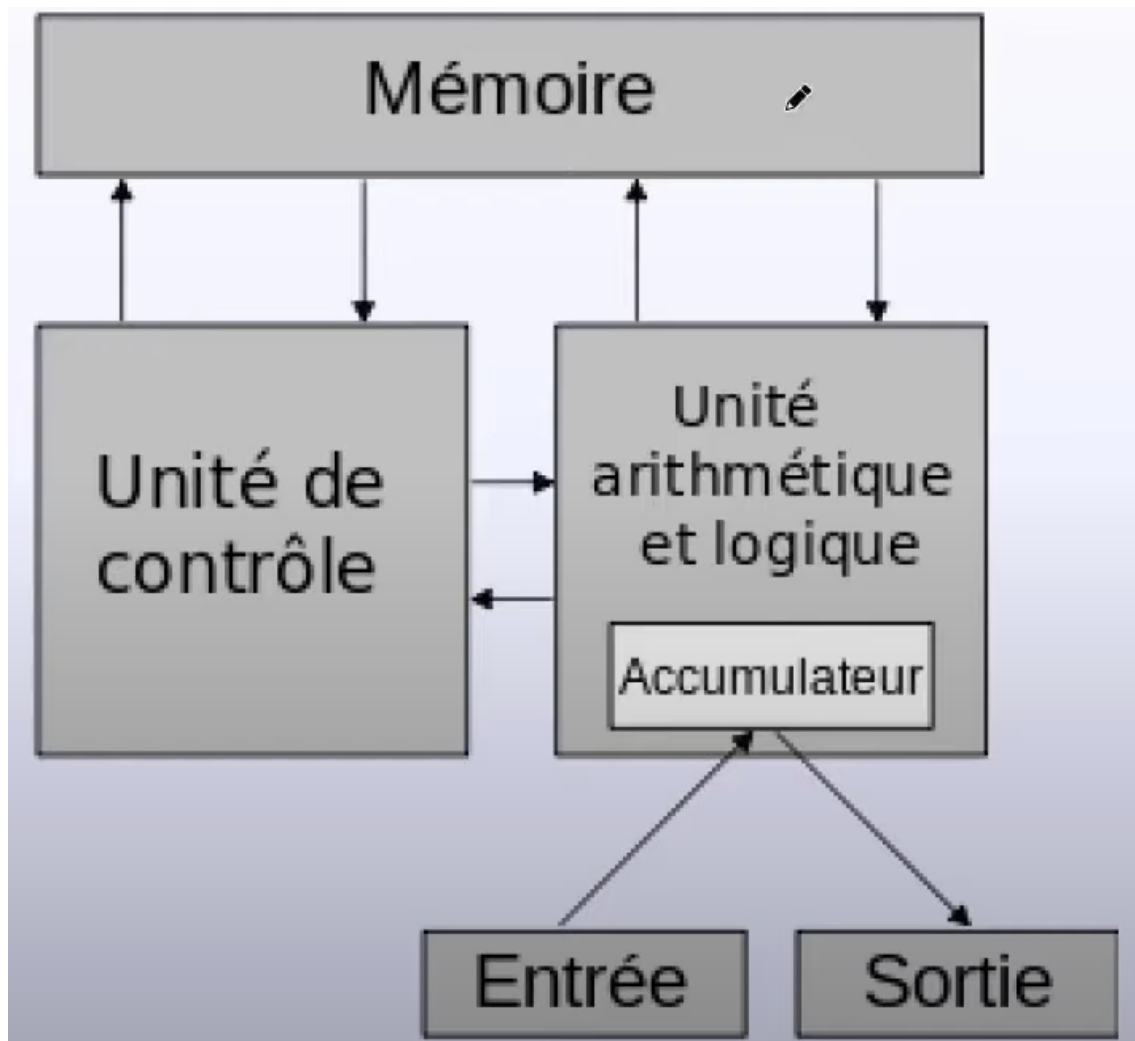


FIGURE 2 – Architecture de von Neumann

Explication des composants de l'architecture :

- **Mémoire** : conserve les données et les instructions.
- **Unité de contrôle** : choisit l'instruction et gère son exécution.
- **Unité arithmétique et logique** : fournit les calculs et les comparaisons nécessaires pour l'exécution de l'instruction.
- **Accumulateur** : effectue les calculs en utilisant l'entrée et la sortie.

1.1.1 Le processeur

Un processeur est le "cerveau" de la machine, il permet d'extraire des instructions de la mémoire et les exécute.

Chaque processeur possède un ensemble spécifique d'instruction exécutables et il

utilise des registres pour accélérer les exécutions en minimisant les copies de la mémoire.

Les registres :

Les registres sont des petites mémoires très rapides de 8, 16, 32, 64 ou 128 bits. Un processeur peut en posséder une dizaine à plusieurs centaines.

Les principaux sont :

- **L'accumulateur (ACC)** : stocke les résultats des opérations arithmétiques et logiques
- **Le compteur ordinal (CO ou PC pour Program Counter)** : contient l'adresse de la prochaine instruction à traiter
- **Le pointeur de pile** : il sert à stocker l'adresse du sommet des piles, qui sont des structures de données généralement utilisées pour gérer des appels de sous-programmes
- **Registre d'instruction (RI)** : contient l'instruction en cours de traitement
- **Registre d'état (PSW pour Processor Status Word)** : stocke des indicateurs sur l'état du système (retenue, dépassement, etc...)
- **Le registre tampon (BUFFER)** : stocke temporairement une donnée

Mémoire cache : Une mémoire cache ou **antémémoire** est une mémoire qui enregistre temporairement des copies de données provenant d'une source (RAM), afin de diminuer le temps d'un accès ultérieur d'un matériel informatique (CPU) à ces données.

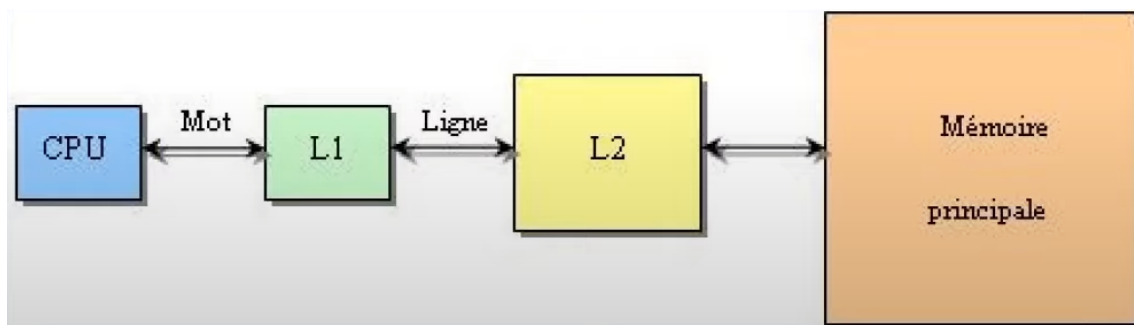


FIGURE 3 – Schéma d'un cache

Avant de regarder ce qu'il y'a dans la mémoire principale, le CPU regarde s'il ne trouve pas ses informations dans le cache (L1 ou L2).

Récapitulatif :

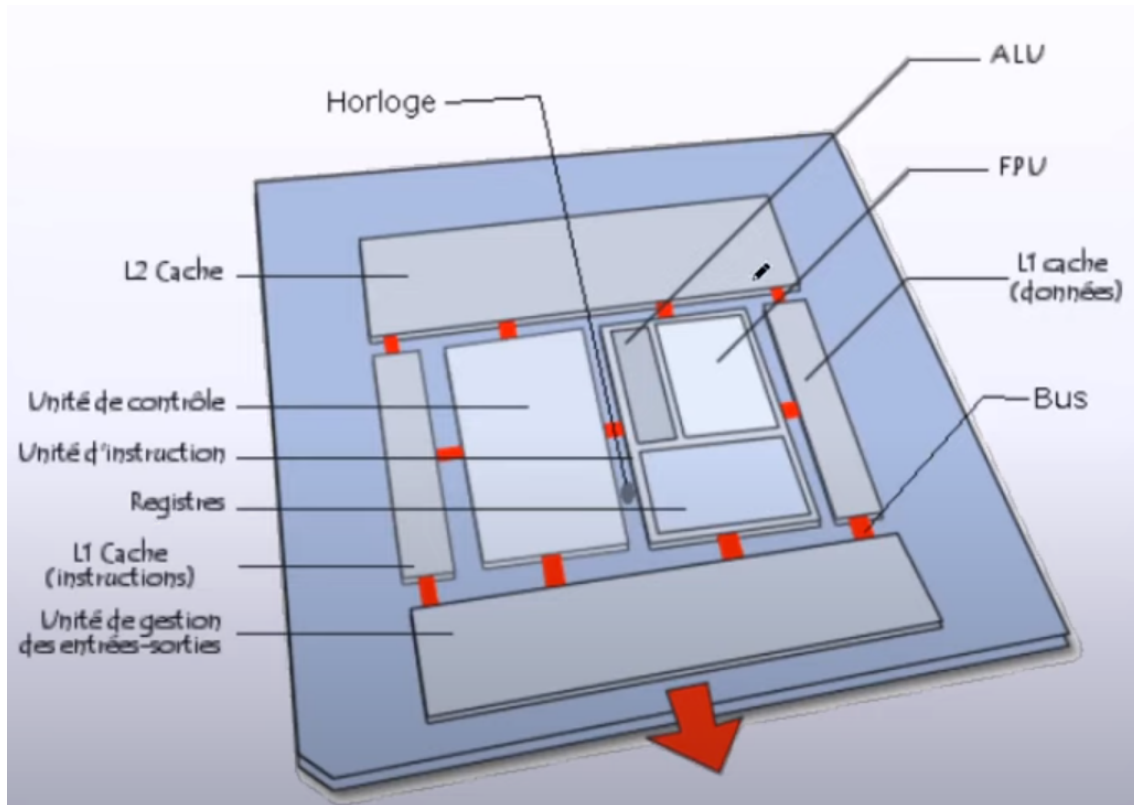


FIGURE 4 – Schéma simplifié d'un processeur

- **Horloge** : fait pour synchroniser l'exécution des instructions
- **ALU** : unité arithmétique et logique
- **FPU** : unité de calcul en virgule flottante
- **Bus** : unité de communication entre les composants

Phases d'exécution des instructions :

- **LI** : Lecture de l'instruction (FETCH instruction)
- **DI** : Décodage de l'instruction (DECODE instruction) et recherche des opérandes (Registre ou valeurs immédiate)
- **EX** : Exécution de l'instruction (EXECUTE instruction) (ADD = somme, SUB = soustraction ,etc...)
- **MEM** : Accès mémoire (MEMORY access), écriture dans la mémoire si nécessaire ou chargement depuis la mémoire
- **ER** : Ecriture (Write instruction) de la valeur calculée dans les registres

Pipeline et processeurs superscalaires : Une **pipeline** permet d'exécuter les instructions plus rapidement. Le principe est que dès qu'une instruction sort d'une de ses phases d'exécution, une autre instruction entre dans cette même phase.

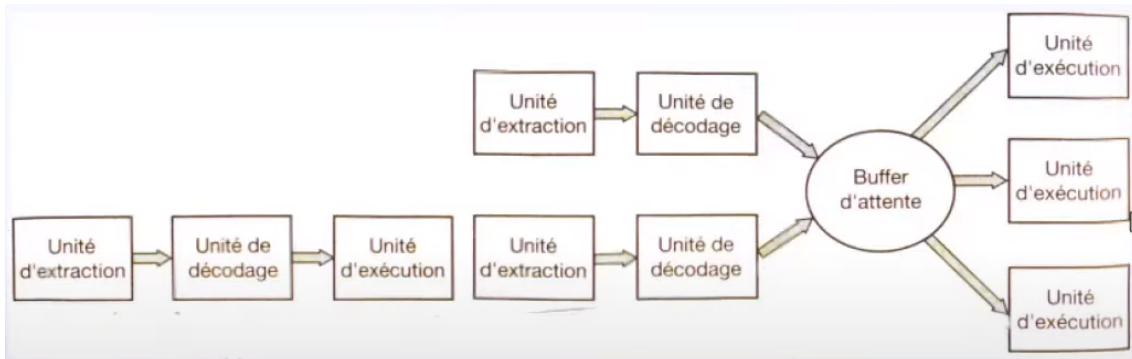


FIGURE 5 – Pipeline

Multicoeurs :

Un cœur représente une unité arithmétique et logique. Un processeur multicoeur permet donc d'effectuer plusieurs calculs simultanément.

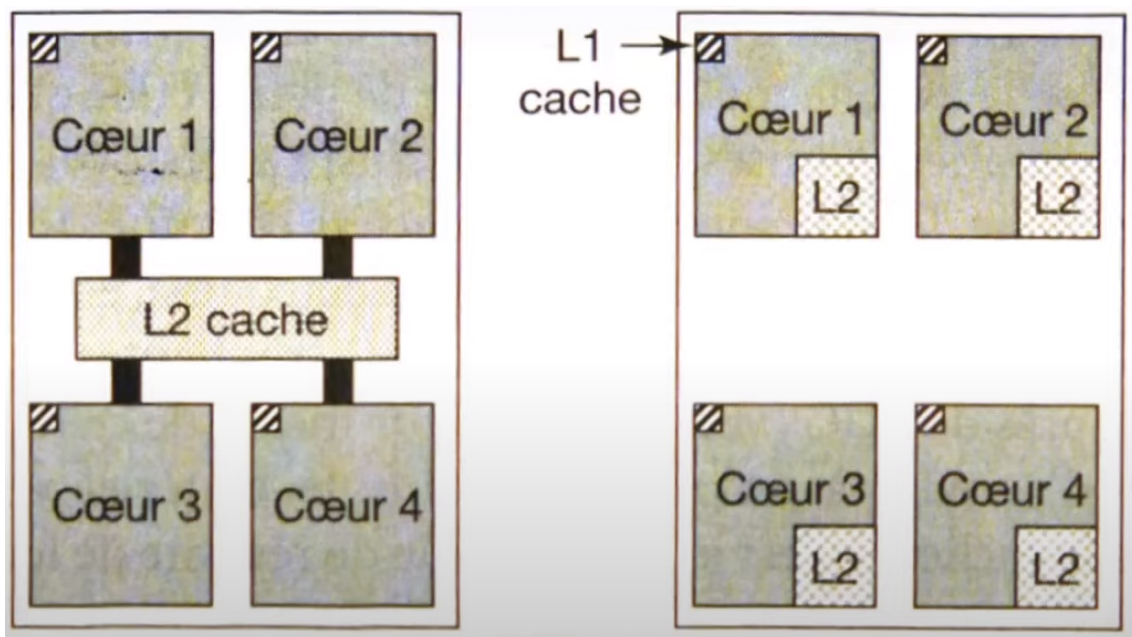


FIGURE 6 – multicoeurs

Sur ce schéma, on remarque que chaque cœur peut soit partager un cache L2, soit avoir son propre cache L2. Le tout est contrôlé par l'unité de contrôle du processeur.

1.1.2 La mémoire

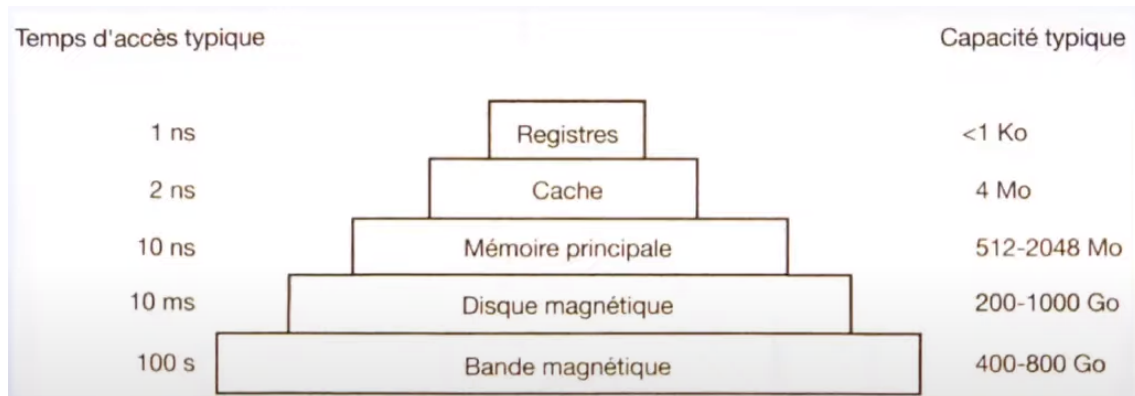


FIGURE 7 – différents types de mémoire

- **Bande / Disque magnétique** : Mémoire de stockage avec une grande capacité typique mais aussi un grand temps d'accès
- **Mémoire principale (RAM)** : mémoire utilisé afin de procéder aux calculs d'instruction.
- **Cache / Registres** : Expliqué au dessus

1.1.3 Les entrées/sorties

Les périphériques d'E/S se présentent généralement en deux parties : un **contrôleur** (unité d'échange) et le périphérique lui-même.

Le contrôleur possède une partie "numérique" pour communiquer avec le processeur et la mémoire et une partie pour envoyer et recevoir des signaux périphériques.

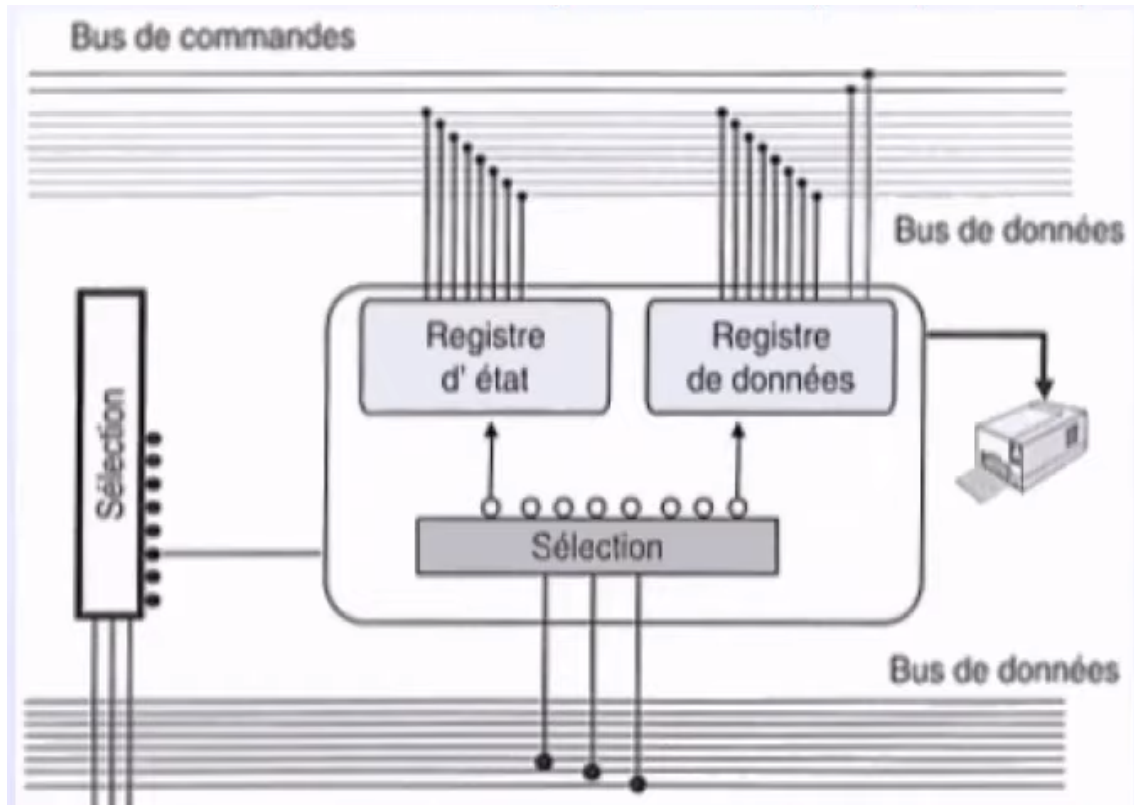


FIGURE 8 – Exemples d'échange E/S

Lors d'une opération d'entrée-sortie, le programme en cours d'exécution cède le processeur et ses registres à un programme du système d'exploitation situé à la RAM (**Pilote (driver)** : petit logiciel créé par le fabricant du matériel pour communiquer avec le contrôleur et le SE).

La manière dont se fait cette opération donne lieu à trois modèles de gestion des entrées-sorties :

1. La liaison programmée :

- **Description** : Dans cette méthode, le processeur effectue des boucles d'attente (polling) jusqu'à ce que le périphérique signale qu'il est prêt pour une opération d'entrée-sortie. Le processeur vérifie continuellement l'état du périphérique pour déterminer quand il peut lire ou écrire des données.
- **Avantages** :
 - Simplicité de mise en œuvre.
 - Contrôle direct du processeur sur les opérations I/O.
- **Inconvénients** :
 - Très inefficace en termes de temps, car le processeur passe beaucoup de cycles à attendre, ce qui le rend indisponible pour d'autres tâches.

- Gaspillage des ressources processeur, ce qui peut ralentir l'ensemble du système.

2. Entrées-sorties pilotées par les interruptions :

- **Description :** Dans cette méthode, le périphérique envoie une interruption au processeur lorsqu'il est prêt à effectuer une opération d'entrée-sortie. Le processeur interrompt son activité courante pour exécuter une routine de service d'interruption (ISR) qui traite l'opération I/O. Souvent, cela implique l'écriture de données dans un tampon (BUFFER), puis le processeur reprend son activité normale.
- **Avantages :**
 - Améliore l'efficacité du processeur, car celui-ci peut effectuer d'autres tâches pendant que l'opération I/O est en attente.
 - Réduction du gaspillage des cycles processeur comparée à la liaison programmée.
- **Inconvénients :**
 - Peut encore être relativement lent si les interruptions sont fréquentes, car le changement de contexte (context switch) associé aux interruptions consomme des ressources.
 - Complexité accrue dans la gestion des interruptions et la synchronisation des tampons.

3. Accès direct à la mémoire (DMA) :

- **Description :** Dans cette méthode, un contrôleur DMA gère directement les transferts de données entre le périphérique et la mémoire, sans intervention continue du processeur. Le processeur configure le contrôleur DMA pour transférer une certaine quantité de données, puis continue son travail. Une fois le transfert terminé, le contrôleur DMA signale au processeur par une interruption.
- **Avantages :**
 - Très efficace, car il libère le processeur des opérations de transfert de données, permettant un multitâche plus efficace.
 - Réduit le temps de latence et augmente la bande passante pour les opérations I/O intensives.
- **Inconvénients :**
 - Complexité accrue dans la configuration et la gestion du contrôleur DMA.
 - Peut nécessiter un matériel supplémentaire dédié au contrôle DMA.

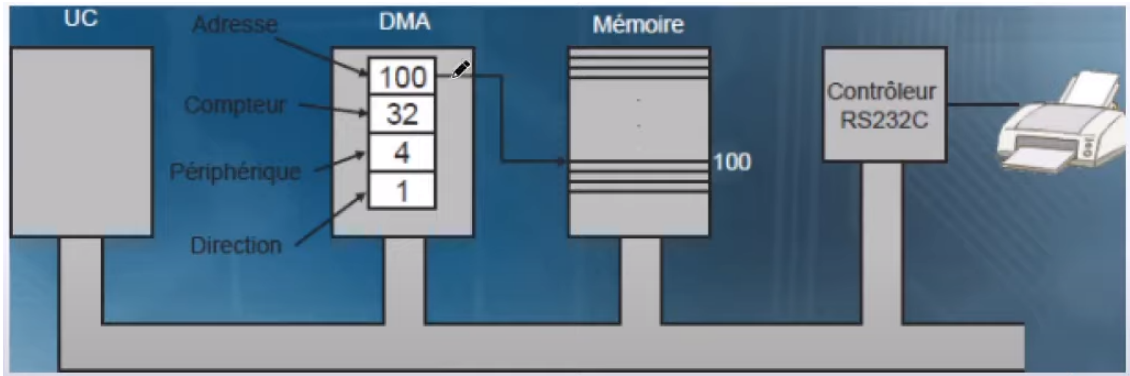


FIGURE 9 – Exemple DMA

1.1.4 Les bus

Les **bus** servent à lier les différents composants entre eux. Il existe différents types de bus :

- **Bus de contrôle** : Ce bus est utilisé pour transmettre les signaux de contrôle entre le processeur et les autres composants du système. Il gère les opérations telles que la lecture, l'écriture, et l'interruption.
- **Bus de données** : Ce bus permet le transfert des données entre le processeur, la mémoire et les périphériques. La largeur du bus de données (par exemple, 8, 16, 32, ou 64 bits) détermine la quantité de données pouvant être transférées simultanément.
- **Bus d'adresse** : Ce bus est utilisé pour envoyer les adresses de mémoire où les données doivent être lues ou écrites. La largeur du bus d'adresse détermine la quantité maximale de mémoire adressable par le système.

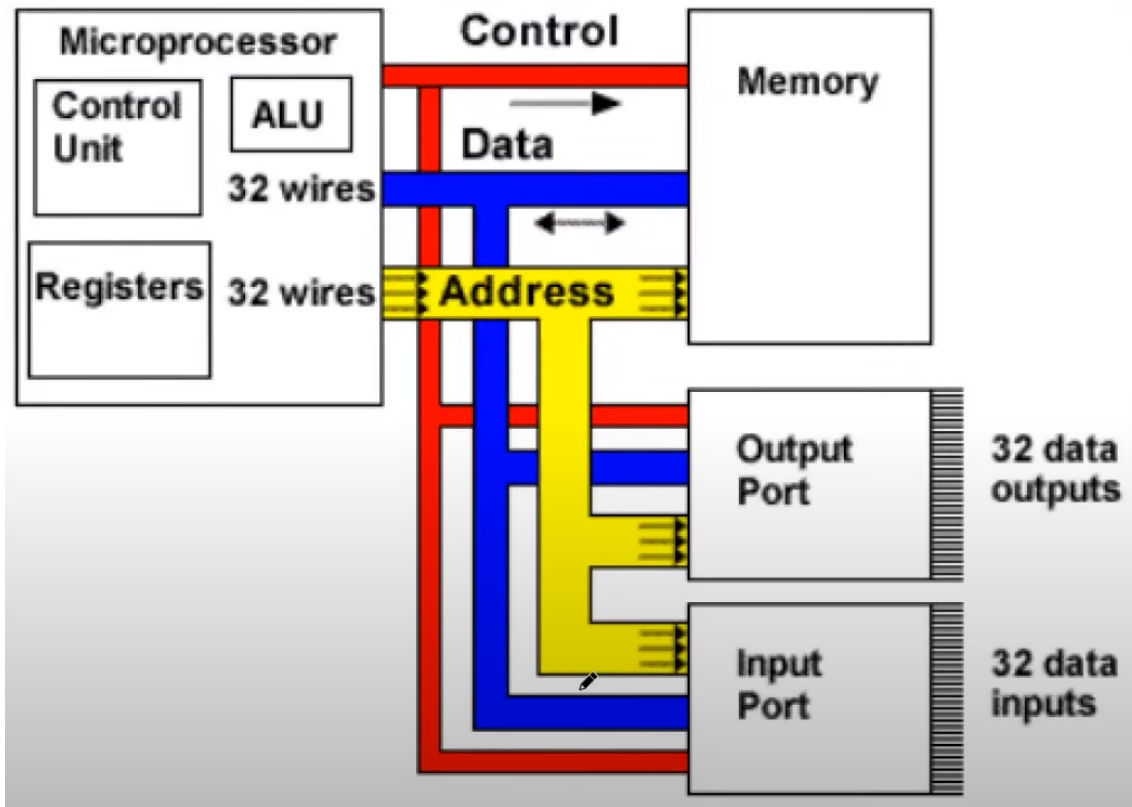


FIGURE 10 – Schéma bus

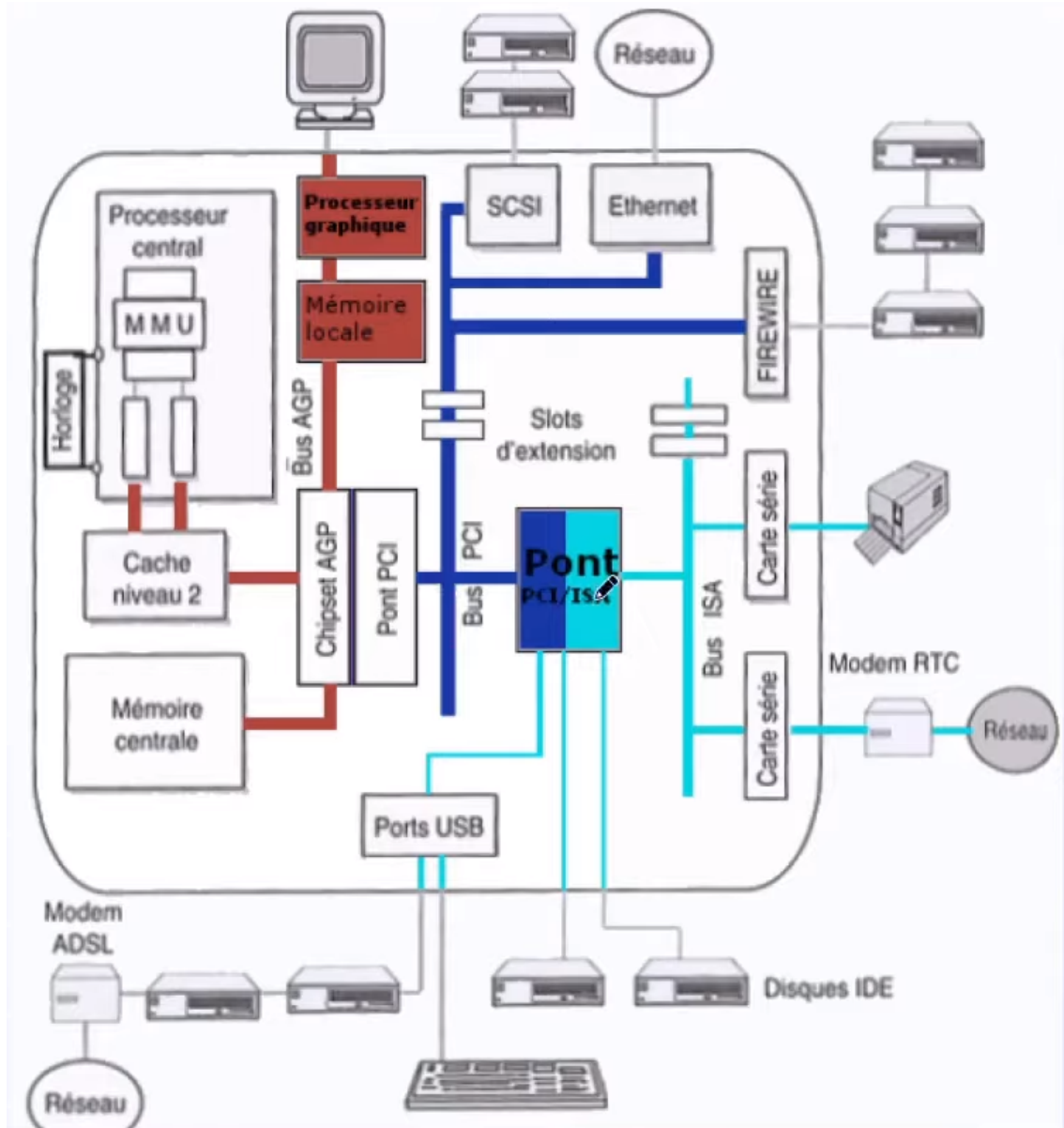


FIGURE 11 – Architecture des bus dans une machine

Dans ce schéma, on remarque de nouveaux types de bus :

- **Bus AGP (Accelerated Graphics Port)** : Ce bus est conçu spécifiquement pour les cartes graphiques. Il offre un accès direct à la mémoire système pour les textures et autres données graphiques, permettant des performances graphiques améliorées par rapport aux bus PCI. L'AGP est capable de transférer des données à des vitesses élevées, ce qui est crucial pour les applications graphiques intensives telles que les jeux vidéo et la modélisation 3D.
- **Bus PCI (Peripheral Component Interconnect)** : Ce bus est utilisé pour

connecter des périphériques internes tels que des cartes réseau, des cartes son, et des cartes graphiques. Le PCI est un bus parallèle qui permet des transferts de données rapides et une large compatibilité avec divers périphériques. Il a été largement utilisé dans les ordinateurs des années 1990 et 2000 avant d'être remplacé par le PCI Express (PCIe), qui offre des vitesses de transfert plus élevées et une meilleure flexibilité.

- **Bus ISA (Industry Standard Architecture) :** Ce bus est l'un des plus anciens types de bus d'ordinateur, utilisé principalement dans les PC des années 1980 et début des années 1990. Bien que plus lent et moins performant que le PCI, l'ISA était essentiel pour connecter des périphériques tels que les cartes son, les modems et les cartes d'extension de mémoire. Sa conception simple et robuste a permis une large adoption initiale.
- **Ponts PCI/ISA :** Ces ponts sont utilisés pour connecter des bus PCI et ISA dans un même système. Ils permettent aux périphériques utilisant le bus ISA plus ancien de communiquer avec les composants plus modernes utilisant le bus PCI. Cela est particulièrement utile pour maintenir la compatibilité avec les anciens périphériques tout en tirant parti des avantages des nouvelles technologies.

1.2 L'architecture d'un système à microprocesseur

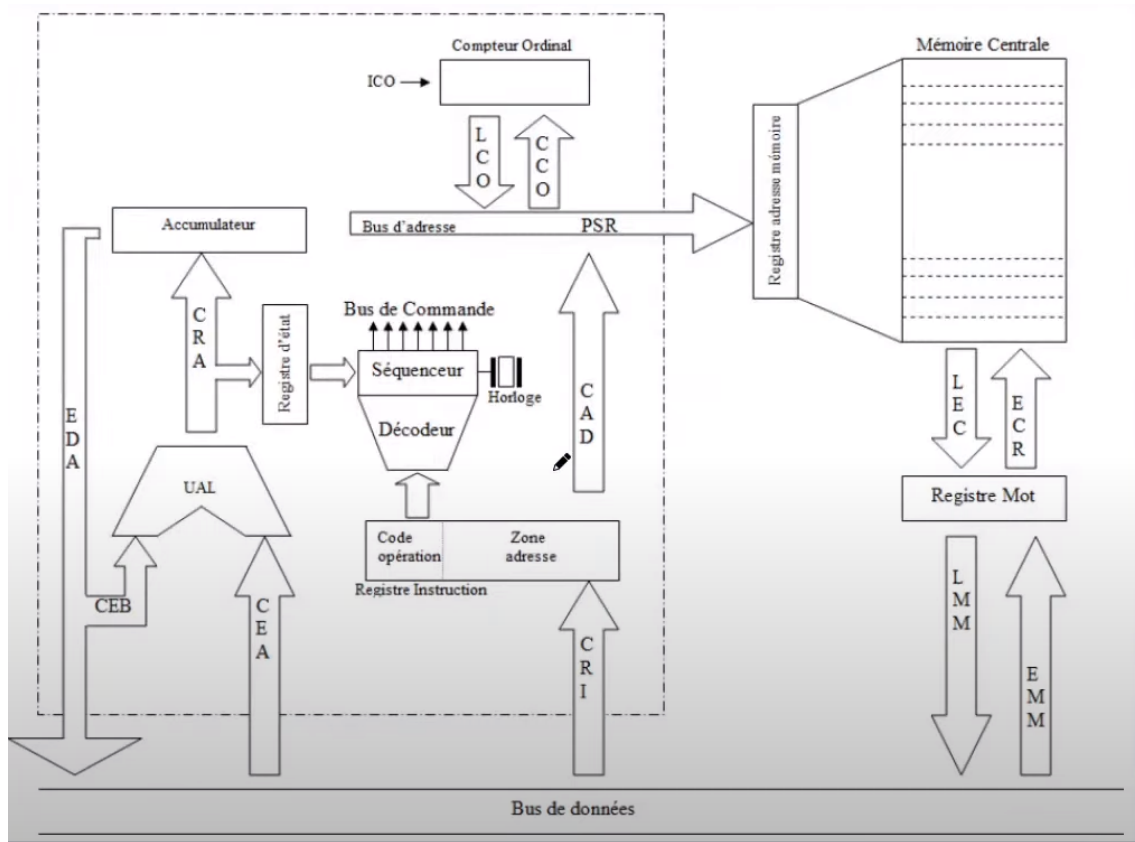


FIGURE 12 – Schéma d'exécution d'une instruction

Quelques définitions :

- **LCO (Lecture du Compteur Ordinal) :** Transfère le contenu du compteur ordinal sur le bus d'adresses
- **CCO (Chargement du Compteur Ordinal) :** Transfère le contenu du bus d'adresses dans le compteur ordinal
- **PSR (Pointage Sur Registre) :** Transfère le contenu du bus d'adresses dans le registre adresse mémoire
- **LEC (Lecture) :** Transfère le contenu de l'emplacement mémoire dans le registre mot (mémoire tampon)
- **ECR (Ecriture) :** Transfère le contenu du registre mot mémoire dans l'emplacement mémoire repéré par l'adresse stockée dans le registre adresse mémoire
- **LMM (Lecture Mot Mémoire) :** Transfère le contenu du registre mot mémoire sur le bus de données

- **EMM (Ecriture Mot Mémoire)** : Transfère le contenu du bus de données dans le registre mot mémoire
- **CRI (Chargement Registre Instruction)** : Transfère le contenu du bus de données dans le registre instruction
- **CAD (Chargement Adresse)** : Transfère la partie adresse (opérande) de l'instruction sur le bus d'adresse
- **CRA (Chargement Registre Accumulateur)** : Le résultat d'une opération dans l'UAL est transféré dans l'accumulateur
- **CEA (Chargement Entrée A)** : Transfère le contenu du bus de données sur l'entrée A de l'UAL
- **CEB (Chargement Entrée B)** : Transfère le contenu de l'accumulateur sur l'entrée B de l'UAL
- **EDA (Envoi de Données Accumulateur)** : Transfère le contenu de l'accumulateur sur le bus de données
- **ICO** : Incrémentation (+1) du Compteur Ordinal

1.2.1 Etapes d'exécution d'un programme

Soit le programme suivant :

```
entier P = 10000001
entier L,S
Lire(L) //lire du clavier
S = L+P
Afficher(S)
```

1. Chargement du programme en mémoire (partie instructions et partie données)
Le compteur ordinal contient 0001 (adresse de 1ere instruction)
2. l'instruction à l'adresse au compteur ordinal (Lire(L)) passe par LEC, Registre Mot , Bus de données, et puisque c'est une instruction, elle passe par CRI vers Registre Instruction
3. Elle sera séparée en code opération (lecture) et adresse (clavier)
4. l'opération est décodée, puis, si c'est une opération complexe, le séquenceur la divisera en plusieurs opérations simples qui s'effectueront en plusieurs cycles.
Avec le bus de commande, on écoutera le bus de données pour une donnée arrivant du clavier et via le bus d'adresse la variable L est ouverte (enable) en écriture.
5. la donnée arrivant du clavier passera par EMM, Registre Mot , ECR et sera écrite dans L
6. Le compteur ordinal est incrémenté (0002)

7. $S = L + P$ passera du même chemin sauf que le séquenceur la divisera en quatre opérations :
- récupérer la valeur de L de la mémoire (Ouvrir L en lecture, sa valeur passe par LEC, Registre Mot , LMM, Bus de données, et puisque c'est une donnée, elle passera par CEA à l'UAL comme l'opération est une addition, il manque une deuxième opérande, la valeur de L passera par CRA et sera sauvegardée temporairement dans l'accumulateur)
 - récupérer la valeur de P de la mémoire (même chose sauf que quand elle arrivera à l'UAL, la valeur de L passera par EDA puis CEB)
 - Calculer la somme (L'UAL calcule la somme et met le résultat dans l'accumulateur)
 - écrire le résultat de la somme dans S (ouvrir S en écriture, la valeur dans l'accumulateur passe par EDA, Bus de données, EMM, Registre Mot , ECR et sera écrite dans S)
8. Le compteur ordinal est incrémenté (0003)
9. Afficher (S) passe jusqu'au registre instruction. La mémoire tampon de l'écran est ouverte en écriture, S passera par LEC , Registre Mot ,LMM , Bus de données et sera écrite dans la mémoire tampon de l'écran puis affichée.