

Fonctions non entièrement définies

Exemple

Les fonctions non entièrement définies sont des fonctions logiques où certaines combinaisons d'entrées n'ont pas de sortie définie. Cela peut se produire lorsque certaines combinaisons d'entrées ne sont pas possibles ou ne sont pas pertinentes pour l'application considérée. Ces états indéfinis sont souvent représentés par des tirets ou des "don't care" dans les tables de vérité.

Circuits Combinatoires vs Séquentiels

Les circuits logiques peuvent être classés en deux grandes catégories : les circuits combinatoires et les circuits séquentiels.

- **Circuits Combinatoires** : La sortie dépend uniquement des entrées actuelles. Ils n'ont pas de mémoire et leur comportement peut être entièrement décrit par une table de vérité. Exemples : additionneurs, décodeurs, multiplexeurs.
- **Circuits Séquentiels** : La sortie dépend non seulement des entrées actuelles, mais aussi de l'état précédent du circuit. Ils ont une mémoire interne, souvent implémentée avec des bascules (flip-flops). Exemples : registres, compteurs, machines d'état fini.

Exemple de Table de Vérité

Une table de vérité est un tableau qui affiche les sorties d'une fonction logique pour toutes les combinaisons possibles d'entrées. Voici un exemple de table de vérité pour une fonction logique à 4 variables $f(a, b, c, d)$.

a	b	c	d	f
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1

Table de Karnaugh

La table de Karnaugh est un outil graphique utilisé pour simplifier les expressions booléennes. Elle permet de regrouper les termes pour obtenir une expression minimale. Voici un exemple de table de Karnaugh pour une fonction logique à 4 variables.

$ab \backslash cd$	00	01	11	10
00	1	0	1	1
01	1	0	0	1
11	0	1	1	0
10	0	0	1	0

L'expression simplifiée de cette table de Karnaugh est :

$$F = \bar{a}\bar{d} + abd + bcd$$

Simplification des fonctions logiques

Règles de Simplification

La simplification des fonctions logiques est essentielle pour réduire la complexité des circuits numériques. Voici deux règles de base :

- **Règle 1** : Il est possible d'ajouter un terme déjà existant dans une équation logique. Par exemple, si on a :

$$f = abc + abc + ab\bar{c} + ab\bar{c} + abc + abc$$

Les termes ajoutés ne changent pas la valeur de la fonction, et on peut simplifier en :

$$f = bc + ac + ab$$

- **Règle 2** : Il est possible de supprimer un terme existant qui résulte de l'union de deux autres termes. Par exemple, si on a :

$$f = ab + \bar{b}c + \bar{c}$$

On peut simplifier en utilisant les propriétés de l'algèbre booléenne.

Lois de De Morgan

Les lois de De Morgan sont des règles fondamentales en algèbre booléenne qui permettent de transformer des expressions logiques. Elles sont particulièrement utiles pour simplifier les expressions contenant des négations.

- **Première loi de De Morgan** :

$$\overline{a + b} = \bar{a} \cdot \bar{b}$$

- **Deuxième loi de De Morgan** :

$$\overline{a \cdot b} = \bar{a} + \bar{b}$$

Exemple d'Application des Lois de De Morgan

Considérons l'expression :

$$a + \bar{a}b$$

En appliquant les lois de De Morgan, on peut la transformer comme suit :

$$\begin{aligned} a + \bar{a}b &= \overline{\overline{a + \bar{a}b}} \\ &= \overline{\bar{a} \cdot \overline{\bar{a}b}} \\ &= \overline{\bar{a} \cdot (a + \bar{b})} \\ &= \overline{\bar{a} \cdot a + \bar{a} \cdot \bar{b}} \\ &= \overline{0 + \bar{a} \cdot \bar{b}} \\ &= \overline{\bar{a} \cdot \bar{b}} \end{aligned}$$

Axiomes et Théorèmes

Axiomes et Théorèmes Fondamentaux

Voici quelques axiomes et théorèmes importants en algèbre booléenne :

Idempotence	$a + a + \dots + a = a$ $a \cdot a \cdot \dots \cdot a = a$
Absorption	$a + a \cdot b = a$ $a \cdot (a + b) = a$
Lois de De Morgan	$\overline{a + b} = \bar{a} \cdot \bar{b}$ $\overline{a \cdot b} = \bar{a} + \bar{b}$
Éléments Neutres	$a + 1 = 1$ $a \cdot 0 = 0$

Portes Logiques et Circuits Numériques

Fonction Logique

Une fonction logique est une fonction dérivée de la combinaison logique d'une table de vérité. Elle détermine chaque combinaison d'entrées produisant une sortie vraie (TRUE) ou fausse (FALSE).

x	b	c	f
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Portes Logiques de Base

Voici quelques portes logiques de base :

- **Porte NON (NOT)** : Inverse l'entrée.
- **Porte ET (AND)** : Sortie vraie si toutes les entrées sont vraies.
- **Porte OU (OR)** : Sortie vraie si au moins une entrée est vraie.
- **Porte NON-ET (NAND)** : Inverse la sortie d'une porte ET.
- **Porte NON-OU (NOR)** : Inverse la sortie d'une porte OU.
- **Porte OU Exclusif (XOR)** : Sortie vraie si un nombre impair d'entrées est vrai.

Exemple de Porte Logique : OU

La porte OU est une porte logique qui produit une sortie vraie si au moins une de ses entrées est vraie. Voici sa table de vérité :

a	b	$a + b$
0	0	0
0	1	1
1	0	1
1	1	1

Multiplexeurs et Démultiplexeurs

Multiplexeur

Un multiplexeur est un circuit combinatoire qui sélectionne une entrée parmi plusieurs et la transmet à une sortie unique. Il est contrôlé par des signaux de sélection.

c_1	c_0	s
0	0	e_1
0	1	e_2
1	0	e_3
1	1	e_4

Démultiplexeur

Un démultiplexeur est un circuit combinatoire qui prend une entrée unique et la dirige vers une des plusieurs sorties, en fonction des signaux de sélection.

c_1	c_0	s_1	s_2	s_3	s_4
0	0	e	0	0	0
0	1	0	e	0	0
1	0	0	0	e	0
1	1	0	0	0	e

Unité Arithmétique et Logique (ALU)

Fonctionnement de l'ALU

L'Unité Arithmétique et Logique (ALU) est un composant clé des processeurs qui effectue des opérations arithmétiques et logiques. Elle prend deux entrées, A et B , et produit une sortie Y .

- **Addition** : $A + B$
- **Soustraction** : $A - B$
- **ET Logique** : $A \wedge B$
- **OU Logique** : $A \vee B$

Additionneur

Un additionneur est un circuit qui effectue l'addition de deux nombres binaires. Voici la table de vérité d'un demi-additionneur :

a_0	b_0	R_0	S_0
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Comparateurs

Comparateur de 2 Mots

Un comparateur est un circuit qui compare deux mots binaires de même taille et détermine leur relation (égalité, supérieur, inférieur).

a_1	a_0	b_1	b_0	$A = B$	$A > B$
0	0	0	0	1	0
0	0	0	1	0	0
0	0	1	0	0	0
0	0	1	1	0	0
0	1	0	0	0	1
0	1	0	1	1	0
0	1	1	0	0	0
0	1	1	1	0	0
1	0	0	0	0	1
1	0	0	1	0	1
1	0	1	0	1	0
1	0	1	1	0	0
1	1	0	0	0	1
1	1	0	1	0	1
1	1	1	0	0	1
1	1	1	1	1	0

Codage et Décodage

Codeur

Un codeur est un circuit qui convertit 2^n entrées en n sorties, où chaque combinaison d'entrée active correspond à une combinaison unique de sorties.

Décodeur

Un décodeur est un circuit qui convertit n entrées en 2^n sorties, où chaque combinaison d'entrée active une sortie unique.

Transcodeurs

Exemple de Transcodeur : Binaire vers Code Gray

Un transcodeur convertit un code en un autre. Voici un exemple de table de transcodage du binaire naturel vers le code Gray :

a	b	c	d	N
0	0	0	0	0
0	0	0	1	1
0	0	1	1	2
0	0	1	0	3
0	1	1	0	4
0	1	1	1	5
0	1	0	1	6
0	1	0	0	7
1	1	0	0	8
1	1	0	1	9
1	1	1	1	10
1	1	1	0	11
1	0	1	0	12
1	0	1	1	13
1	0	0	1	14
1	0	0	0	15

Registres

Fonction des Registres

Les registres sont des mémoires ultra-rapides situées dans le processeur. Ils stockent temporairement les données pendant leur traitement. Les registres sont essentiels pour améliorer les performances des opérations arithmétiques et logiques.

Les registres peuvent avoir différentes tailles, selon l'architecture du processeur :

- **8 bits** : Utilisés pour des opérations simples et des manipulations de données de base (ex : un caractère est codé sur 8 bits).
- **16 bits** : Utilisés dans les anciens processeurs ou pour des opérations spécifiques nécessitant 16 bits.
- **32 bits** : Très répandus dans les processeurs modernes pour des opérations plus complexes.
- **64 bits** : Utilisés dans les architectures modernes pour des calculs avancés et une meilleure gestion de la mémoire.

Types de Registres

Les registres peuvent être classés en plusieurs types selon leur fonction :

- **Registres de données** : Stockent les données pendant les opérations.
- **Registres d'adresse** : Stockent les adresses mémoire.
- **Registres généraux** : Utilisés pour des opérations variées.
- **Registres spéciaux** : Comme le compteur ordinal (PC) et le registre d'état (PSW).

Unité de Contrôle

Fonction de l'Unité de Contrôle

L'unité de contrôle est le "cerveau" du processeur. Elle gère l'exécution des instructions en coordonnant les opérations entre l'ALU, les registres et la mémoire. Elle détermine la priorité des opérations dans les systèmes

multitâches et assure la sécurité des instructions au niveau du noyau.

1. Récupérer l'instruction depuis la mémoire.
2. Décoder l'instruction pour déterminer les opérations nécessaires.
3. Exécuter l'instruction en utilisant l'ALU ou les registres.

Stratégies de Réduction de la Consommation Énergétique

Pour réduire la consommation énergétique, deux stratégies principales sont utilisées :

- **Clock Gating** : Désactive le signal d'horloge pour les composants inactifs, réduisant ainsi leur consommation énergétique.
- **Dynamic Voltage Scaling** : Ajuste dynamiquement la tension d'alimentation des transistors, réduisant la consommation et la dissipation thermique.

Microprocesseurs

Histoire et Évolution

Les microprocesseurs ont connu une évolution rapide depuis leur invention :

- **1965** : Gordon Moore prédit que le nombre de transistors sur une puce doublerait tous les deux ans (Loi de Moore).
- **1971** : Intel lance le premier microprocesseur, l'Intel 4004, avec une fréquence de 740 kHz et une mémoire de 5120 bits.
- **Années 1980-2000** : Miniaturisation continue et augmentation de la puissance de calcul.

Architecture de Base

L'architecture de base d'un microprocesseur comprend :

- **Unité Arithmétique et Logique (ALU)** : Effectue les opérations arithmétiques et logiques.
- **Unité de Contrôle** : Gère l'exécution des instructions.
- **Registres** : Stockent temporairement les données.
- **Mémoire Cache** : Mémoire rapide pour accélérer l'accès aux données fréquemment utilisées.
- **Bus Système** : Transfère les données entre les différents composants.

Comparaison des Architectures ARM et x86

Les architectures ARM et x86 sont les deux architectures de microprocesseurs les plus répandues :

- **ARM** : Conçue pour une faible consommation énergétique, idéale pour les appareils mobiles et embarqués.
- **x86** : Conçue pour des performances élevées, utilisée dans les ordinateurs personnels et les serveurs.

Microcontrôleurs

Définition et Fonctionnement

Un microcontrôleur est un système complet intégré sur une seule puce. Contrairement aux microprocesseurs, qui nécessitent des composants externes pour fonctionner, un microcontrôleur intègre la mémoire, les entrées/sorties, et d'autres périphériques nécessaires pour des tâches spécifiques.

- **Intégration** : CPU, mémoire (RAM, ROM, EEPROM), entrées/sorties, et périphériques sur une seule puce.
- **Faible consommation** : Conçus pour des applications nécessitant une faible consommation énergétique.
- **Coût réduit** : Idéal pour les applications embarquées où le coût est un facteur critique.

Histoire et Évolution

L'évolution des microcontrôleurs a été marquée par plusieurs étapes clés :

- **1971** : Invention du premier microcontrôleur, le TMS 1000, par Texas Instruments.
- **1976** : Introduction du microcontrôleur 8048 par Intel, utilisé dans de nombreux systèmes embarqués.
- **Années 1980-1990** : Développement de microcontrôleurs avec des mémoires EEPROM, permettant une programmation et une reprogrammation faciles.
- **Années 2000** : Miniaturisation continue et intégration de périphériques spécialisés (ADC, DAC, interfaces de communication).

Exemples de Microcontrôleurs

Voici quelques exemples de microcontrôleurs populaires :

- **Atmel ATmega32u4** : Utilisé dans les cartes Arduino, idéal pour les projets de prototypage.
- **Microchip PIC16F84** : Un des premiers microcontrôleurs largement utilisés dans l'industrie.
- **Espressif ESP32** : Connu pour ses capacités Wi-Fi et Bluetooth intégrées.
- **Texas Instruments MSP430** : Utilisé pour des applications nécessitant une très faible consommation énergétique.

Systèmes Embarqués

Définition et Applications

Un système embarqué est un système informatique conçu pour effectuer des tâches spécifiques. Il est souvent intégré dans un dispositif plus large et peut inclure du matériel et des logiciels dédiés.

- **Électronique grand public** : Téléviseurs, lave-linge, réfrigérateurs intelligents.
- **Automobile** : Systèmes de contrôle moteur, systèmes de navigation.
- **Médical** : Appareils de surveillance, stimulateurs cardiaques.
- **Industriel** : Automates programmables, systèmes de contrôle de processus.

Architecture Typique

L'architecture typique d'un système embarqué comprend :

- **Microcontrôleur/Microprocesseur** : Cœur du système embarqué.
- **Mémoire** : RAM pour les données temporaires, ROM/Flash pour le stockage permanent.
- **Périphériques d'entrée/sortie** : Capteurs, actionneurs, interfaces utilisateur.
- **Logiciel embarqué** : Firmware ou système d'exploitation en temps réel (RTOS).

Exemples de Systèmes Embarqués

Voici quelques exemples de systèmes embarqués :

- **Systèmes de contrôle automobile** : Gestion du moteur, freinage antiblocage (ABS).
- **Appareils médicaux** : Moniteurs de fréquence cardiaque, pompes à insuline.
- **Dispositifs IoT** : Capteurs connectés, systèmes domotiques.

Comparaison entre Microprocesseurs et Microcontrôleurs

Voici une comparaison entre les microprocesseurs et les microcontrôleurs :

Caractéristique	Microprocesseur	Microcontrôleur
Complexité	Élevée	Faible à modérée
Consommation énergétique	Élevée	Faible
Coût	Élevé	Faible
Applications	Ordinateurs, serveurs	Systèmes embarqués, IoT
Intégration	Nécessite des composants externes	Tout intégré sur une puce

Bus Système

Définition et Fonctionnement

Un bus système est un ensemble de connexions physiques qui permettent la communication entre les différents composants d'un système informatique. Il transporte les données, les adresses et les signaux de contrôle entre le processeur, la mémoire et les périphériques.

- **Bus de données** : Transfère les données entre les composants.
- **Bus d'adresses** : Transmet les informations d'adressage pour identifier les emplacements mémoire ou les périphériques.
- **Bus de contrôle** : Gère les signaux de contrôle pour synchroniser les opérations.

Architectures de Bus

Il existe plusieurs architectures de bus, notamment :

- **Bus unique** : Tous les composants partagent un seul bus.
- **Bus hiérarchique** : Plusieurs niveaux de bus interconnectés pour améliorer les performances.
- **Bus dédié** : Chaque périphérique a son propre bus pour éviter les conflits.

Exemples de Bus

Voici quelques exemples de bus couramment utilisés :

- **PCI (Peripheral Component Interconnect)** : Utilisé pour connecter des périphériques internes comme les cartes graphiques.
- **USB (Universal Serial Bus)** : Utilisé pour connecter des périphériques externes comme les clés USB et les imprimantes.
- **I2C (Inter-Integrated Circuit)** : Utilisé pour la communication entre microcontrôleurs et périphériques embarqués.
- **SPI (Serial Peripheral Interface)** : Utilisé pour les communications haute vitesse entre microcontrôleurs et périphériques.

Interfaces de Communication

Définition et Importance

Les interfaces de communication permettent aux systèmes numériques de communiquer entre eux. Elles sont essentielles pour échanger des données entre différents composants ou systèmes.

- **Interfaces parallèles** : Transfèrent plusieurs bits simultanément.
- **Interfaces série** : Transfèrent les bits un par un, souvent utilisées pour les communications longue distance.

Interfaces Série

Les interfaces série sont largement utilisées en raison de leur simplicité et de leur efficacité pour les communications longue distance. Voici quelques exemples :

- **UART (Universal Asynchronous Receiver/Transmitter)** : Utilisé pour les communications asynchrones.
- **I2C (Inter-Integrated Circuit)** : Protocole de communication synchrone multi-maître, multi-esclave.
- **SPI (Serial Peripheral Interface)** : Protocole de communication synchrone haute vitesse.
- **CAN (Controller Area Network)** : Utilisé principalement dans les systèmes automobiles.

Interfaces Parallèles

Les interfaces parallèles permettent de transférer plusieurs bits simultanément, ce qui peut augmenter la vitesse de communication. Voici quelques exemples :

- **Bus ISA (Industry Standard Architecture)** : Utilisé dans les anciens ordinateurs pour connecter des cartes d'extension.
- **Bus PCI (Peripheral Component Interconnect)** : Utilisé pour connecter des périphériques internes comme les cartes graphiques et les cartes réseau.

Comparaison des Interfaces Série et Parallèle

Voici une comparaison entre les interfaces série et parallèles :

Caractéristique	Interfaces Série	Interfaces Parallèles
Vitesse	Généralement plus lente	Généralement plus rapide
Complexité	Moins complexe	Plus complexe
Distance	Adaptée aux longues distances	Adaptée aux courtes distances
Coût	Moins coûteuse	Plus coûteuse

Exemples Pratiques

Exemple d'Utilisation de l'I2C

L'I2C est un protocole de communication série largement utilisé dans les systèmes embarqués pour connecter des périphériques comme des capteurs et des EEPROMs. Voici un exemple de configuration I2C :

- **Maître** : Le microcontrôleur qui initie la communication.
- **Esclave** : Le périphérique qui répond aux requêtes du maître.

Exemple d'Utilisation de l'UART

L'UART est utilisé pour les communications asynchrones entre deux appareils. Voici un exemple de configuration UART :

- **Baud Rate** : Vitesse de transmission des données (ex. 9600 bauds).
- **Parité** : Méthode de détection d'erreur (ex. paire, impaire, ou aucune).
- **Stop Bits** : Nombre de bits de stop pour indiquer la fin d'une transmission.

Mémoires

Introduction aux Mémoires

Les mémoires sont des composants essentiels dans les systèmes numériques, permettant de stocker des données et des instructions. Elles peuvent être classées en deux grandes catégories : les mémoires volatiles et les mémoires non volatiles.

- **Mémoires volatiles** : Perdent leurs données en l'absence d'alimentation électrique. Exemples : RAM (Random Access Memory).
- **Mémoires non volatiles** : Conservent leurs données même en l'absence d'alimentation électrique. Exemples : ROM (Read-Only Memory), Flash.

Mémoire RAM

La mémoire RAM (Random Access Memory) est une mémoire volatile utilisée pour stocker temporairement les données et les instructions nécessaires au fonctionnement des programmes.

- **SRAM (Static RAM)** : Utilise des bascules pour stocker chaque bit. Elle est rapide et coûteuse.
- **DRAM (Dynamic RAM)** : Utilise des condensateurs pour stocker chaque bit. Elle est moins chère mais plus lente que la SRAM et nécessite un rafraîchissement périodique.

Mémoire ROM

La mémoire ROM (Read-Only Memory) est une mémoire non volatile utilisée pour stocker des données de manière permanente.

- **ROM** : Mémoire en lecture seule, programmée lors de la fabrication.
- **PROM (Programmable ROM)** : Peut être programmée une seule fois par l'utilisateur.
- **EPROM (Erasable Programmable ROM)** : Peut être effacée et reprogrammée à l'aide d'un rayonnement ultraviolet.
- **EEPROM (Electrically Erasable Programmable ROM)** : Peut être effacée et reprogrammée électriquement.
- **Flash** : Type d'EEPROM qui permet des effacements et des reprogrammations par blocs.

Systèmes de Stockage

Introduction aux Systèmes de Stockage

Les systèmes de stockage permettent de conserver des données de manière permanente ou semi-permanente. Ils sont essentiels pour sauvegarder des informations et des programmes.

- **Disques durs (HDD)** : Utilisent des plateaux magnétiques pour stocker les données.
- **Disques SSD (Solid State Drive)** : Utilisent de la mémoire flash pour stocker les données, offrant des vitesses de lecture/écriture plus rapides que les HDD.
- **Clés USB** : Utilisent de la mémoire flash pour un stockage portable.
- **Cartes mémoire** : Utilisées dans les appareils photo, smartphones, et autres appareils portables.

Disques Dur (HDD) vs. Disques SSD

Voici une comparaison entre les disques durs (HDD) et les disques SSD :

Caractéristique	HDD	SSD
Technologie	Plateaux magnétiques	Mémoire flash
Vitesse	Plus lente	Plus rapide
Durabilité	Moins résistante aux chocs	Plus résistante aux chocs
Consommation énergétique	Plus élevée	Plus faible
Coût	Moins cher par Go	Plus cher par Go

Mémoire Cache

La mémoire cache est une mémoire rapide située entre le processeur et la mémoire principale. Elle stocke temporairement les données fréquemment utilisées pour réduire le temps d'accès et améliorer les performances du système.

- **Niveau 1 (L1)** : Le plus rapide et le plus proche du processeur.
- **Niveau 2 (L2)** : Plus grand que le L1 mais un peu plus lent.
- **Niveau 3 (L3)** : Encore plus grand, partagé entre les cœurs du processeur.

Exemple de Hiérarchie de Mémoire

La hiérarchie de mémoire dans un système informatique est organisée de manière à optimiser les performances et les coûts. Voici un exemple de hiérarchie typique :

Niveau	Type de Mémoire	Temps d'Accès
1	Registres	Quelques nanosecondes
2	Cache L1	Moins de 10 nanosecondes
3	Cache L2	Environ 20 nanosecondes
4	Cache L3	Environ 50 nanosecondes
5	RAM	Environ 100 nanosecondes
6	Disque SSD	Quelques microsecondes
7	Disque Dur (HDD)	Quelques millisecondes

Systèmes d'Exploitation Embarqués

Introduction aux Systèmes d'Exploitation Embarqués

Un système d'exploitation embarqué (SE embarqué) est un système d'exploitation conçu pour fonctionner sur des systèmes embarqués. Ces systèmes sont optimisés pour des tâches spécifiques et doivent souvent fonctionner avec des ressources limitées, comme une mémoire réduite et une faible puissance de calcul.

- **Temps réel** : Capacité à répondre aux événements dans des délais déterminés.
- **Faible empreinte mémoire** : Conçu pour fonctionner avec une mémoire limitée.
- **Modularité** : Permet d'ajouter ou de retirer des fonctionnalités selon les besoins.
- **Fiabilité** : Doit fonctionner de manière fiable dans des environnements variés.

Exemples de Systèmes d'Exploitation Embarqués

Voici quelques exemples de systèmes d'exploitation embarqués populaires :

- **FreeRTOS** : Un système d'exploitation temps réel open-source, largement utilisé dans les applications embarquées.
- **VxWorks** : Un système d'exploitation temps réel utilisé dans les applications critiques, comme l'aérospatial et la défense.
- **QNX** : Un système d'exploitation temps réel utilisé dans les systèmes automobiles et médicaux.
- **Embedded Linux** : Une version allégée de Linux adaptée aux systèmes embarqués.

FreeRTOS

FreeRTOS est un noyau temps réel open-source pour les systèmes embarqués. Il est conçu pour être petit et simple, tout en offrant des fonctionnalités avancées de gestion des tâches.

- **Gestion des tâches** : Permet de créer, supprimer et gérer des tâches.
- **Synchronisation** : Utilise des sémaphores, des mutex et des files d'attente pour synchroniser les tâches.
- **Gestion de la mémoire** : Offre des mécanismes pour allouer et libérer de la mémoire dynamiquement.
- **Portabilité** : Peut être porté sur une large gamme de microcontrôleurs.

Applications Pratiques

Applications des Systèmes Embarqués

Les systèmes embarqués sont utilisés dans une multitude d'applications, allant des appareils ménagers aux systèmes industriels complexes.

- **Automobile** : Systèmes de contrôle moteur, systèmes de navigation, systèmes de sécurité (ABS, airbags).
- **Médical** : Appareils de surveillance, stimulateurs cardiaques, pompes à insuline.
- **Industriel** : Automates programmables, systèmes de contrôle de processus.
- **Grand Public** : Téléviseurs, lave-linge, réfrigérateurs intelligents.
- **Aérospatial** : Systèmes de navigation, systèmes de contrôle de vol.

Étude de Cas : Système de Contrôle Moteur

Un système de contrôle moteur est un exemple typique d'application embarquée dans le domaine automobile. Ce système utilise des capteurs pour surveiller divers paramètres du moteur et ajuste les actions en conséquence pour optimiser les performances et réduire les émissions.

- **Capteurs** : Mesurent des paramètres comme la température, la pression, et la position du vilebrequin.
- **Unité de Contrôle Électronique (ECU)** : Un microcontrôleur qui traite les données des capteurs et contrôle les actionneurs.
- **Actionneurs** : Contrôlent des éléments comme les injecteurs de carburant et l'allumage.

Étude de Cas : Appareil Médical Embarqué

Les appareils médicaux embarqués, comme les moniteurs de fréquence cardiaque et les pompes à insuline, sont conçus pour fournir des soins médicaux précis et fiables.

- **Capteurs** : Mesurent des paramètres physiologiques comme la fréquence cardiaque et la glycémie.
- **Microcontrôleur** : Traite les données des capteurs et contrôle les actionneurs.
- **Interface Utilisateur** : Permet aux utilisateurs et aux professionnels de santé d'interagir avec l'appareil.
- **Communication** : Permet de transmettre les données à d'autres appareils ou systèmes de surveillance.

Étude de Cas : Domotique

Les systèmes domotiques utilisent des microcontrôleurs et des capteurs pour automatiser et contrôler divers aspects d'une maison, comme l'éclairage, le chauffage, et la sécurité.

- **Capteurs** : Détectent des paramètres comme la température, la lumière, et le mouvement.
- **Microcontrôleur** : Traite les données des capteurs et contrôle les actionneurs.
- **Actionneurs** : Contrôlent des éléments comme les interrupteurs de lumière et les thermostats.
- **Interface Utilisateur** : Permet aux utilisateurs de configurer et de contrôler le système.

Protocoles de Communication Avancés

Introduction aux Protocoles de Communication Avancés

Les protocoles de communication avancés sont essentiels pour permettre une communication efficace et sécurisée entre les systèmes embarqués et d'autres dispositifs. Ces protocoles sont souvent utilisés dans des applications nécessitant une haute fiabilité et une faible latence.

- **Protocoles filaires** : Utilisent des câbles pour transmettre les données.
- **Protocoles sans fil** : Utilisent des ondes radio pour transmettre les données.

Protocoles Filaires

Les protocoles filaires sont largement utilisés dans les systèmes embarqués pour leur fiabilité et leur vitesse de transmission élevée.

- **CAN (Controller Area Network)** : Utilisé principalement dans les systèmes automobiles pour la communication entre les différents modules électroniques.
- **Ethernet** : Utilisé pour les communications à haut débit dans les réseaux locaux.
- **MODBUS** : Protocole de communication série utilisé dans les systèmes industriels pour la communication entre automates programmables.

Protocoles Sans Fil

Les protocoles sans fil sont de plus en plus utilisés dans les systèmes embarqués pour leur flexibilité et leur capacité à permettre une communication à distance.

- **Wi-Fi** : Utilisé pour les communications locales sans fil à haut débit.
- **Bluetooth** : Utilisé pour les communications à courte portée entre appareils.
- **Zigbee** : Protocole de communication sans fil à faible consommation, utilisé dans les applications IoT.
- **LoRaWAN** : Protocole de communication longue portée et à faible consommation, utilisé pour les applications IoT à grande échelle.

Comparaison des Protocoles de Communication

Voici une comparaison entre les protocoles de communication filaires et sans fil :

Caractéristique	Protocoles Filaires	Protocoles Sans Fil
Vitesse	Généralement plus rapide	Généralement plus lente
Fiabilité	Très fiable	Sujette aux interférences
Latence	Faible	Plus élevée
Portée	Limitée par la longueur des câbles	Potentiellement illimitée
Flexibilité	Moins flexible	Très flexible

Tendances Futures

Évolutions Technologiques

Les systèmes embarqués et les microcontrôleurs évoluent rapidement grâce aux avancées technologiques. Voici quelques tendances futures :

- **Intelligence Artificielle (IA)** : Intégration de l'IA dans les systèmes embarqués pour permettre des fonctionnalités avancées comme la reconnaissance vocale et la vision par ordinateur.
- **Internet des Objets (IoT)** : Développement de systèmes embarqués connectés pour des applications domestiques, industrielles et médicales.
- **5G** : Utilisation de la technologie 5G pour des communications ultra-rapides et à faible latence.
- **Sécurité** : Renforcement de la sécurité des systèmes embarqués pour protéger contre les cyberattaques.
- **Edge Computing** : Traitement des données au plus près de leur source pour réduire la latence et améliorer l'efficacité.

Intelligence Artificielle dans les Systèmes Embarqués

L'intégration de l'IA dans les systèmes embarqués permet d'ajouter des fonctionnalités intelligentes, comme la reconnaissance de motifs et la prise de décision autonome.

- **Reconnaissance d'images** : Utilisée dans les systèmes de surveillance et les véhicules autonomes.
- **Reconnaissance vocale** : Utilisée dans les assistants vocaux et les systèmes de commande vocale.
- **Maintenance prédictive** : Utilisée dans les systèmes industriels pour anticiper les pannes et optimiser la maintenance.

Internet des Objets (IoT)

L'IoT est une tendance majeure dans les systèmes embarqués, permettant de connecter divers appareils pour échanger des données et automatiser des tâches.

- **Maison intelligente** : Automatisation des systèmes d'éclairage, de chauffage et de sécurité.
- **Villes intelligentes** : Optimisation de la gestion des ressources et des services urbains.
- **Industrie 4.0** : Automatisation et optimisation des processus industriels.

Sécurité des Systèmes Embarqués

La sécurité des systèmes embarqués est un enjeu crucial, surtout avec l'augmentation des cyberattaques. Voici quelques mesures de sécurité :

- **Chiffrement** : Utilisation de techniques de chiffrement pour protéger les données.
- **Authentification** : Mécanismes pour vérifier l'identité des utilisateurs et des appareils.
- **Mises à jour logicielles** : Mises à jour régulières pour corriger les vulnérabilités.
- **Sécurité physique** : Protection contre les accès non autorisés aux dispositifs physiques.

Edge Computing

L'Edge Computing consiste à traiter les données au plus près de leur source, réduisant ainsi la latence et améliorant les performances des systèmes embarqués.

- **Réduction de la latence** : Traitement des données localement plutôt que dans le cloud.
- **Économie de bande passante** : Moins de données à transmettre vers le cloud.
- **Amélioration de la sécurité** : Les données sensibles peuvent être traitées localement.

Multicore and Manycore Systems

Examples

- **Intel entry to the multicore CPU on the mass market in 2006:** Intel a marqué un tournant dans l'informatique grand public avec le lancement des processeurs multicœurs en 2006. Cela a permis aux utilisateurs de bénéficier de performances accrues grâce à la capacité de traiter plusieurs tâches simultanément.
 - **Intel Core 2 Duo (2006):** Premier processeur grand public à deux cœurs, permettant une meilleure gestion des tâches multithreads.
- **Intel Core i9-13900K (2022):** Ce processeur possède 24 cœurs, ce qui permet une gestion encore plus efficace des tâches parallèles, idéale pour les applications gourmandes en ressources comme le montage vidéo ou les jeux vidéo.
 - **Source:** <https://www.intel.com/content/www/us/en/products/processors/core/i9-processors/i9-13900k.html>
- **AMD Threadripper PRO 9995WX (2025):** Avec 96 cœurs, ce processeur est conçu pour les stations de travail professionnelles, offrant des performances exceptionnelles pour les tâches de calcul intensif.
 - **Source:** <https://www.amd.com/en/products/cpu/amd-threadripper-pro-9995wx>

How it works

Chaque cœur d'un processeur multicœur peut exécuter son propre thread ou partager une charge de travail. Cela permet de réaliser du multitâche de manière efficace.

- **Multitâche:** Les processeurs multicœurs permettent d'exécuter plusieurs tâches simultanément, ce qui améliore la productivité et réduit le temps d'attente pour les utilisateurs.
- **Parallélisme:** En répartissant les tâches entre plusieurs cœurs, les processeurs multicœurs optimisent l'utilisation des ressources et réduisent la consommation d'énergie par tâche.
 - **Source:** <https://www.computerhope.com/jargon/m/multicore.htm>

What is a Multicore Processor?

Un processeur multicœur est une puce contenant deux cœurs de CPU ou plus, chacun capable de fonctionner de manière indépendante. Cela permet d'exécuter plusieurs tâches simultanément, améliorant ainsi les performances globales du système.

Behind the Idea

Auparavant, les performances des CPU étaient améliorées en augmentant la fréquence d'horloge. Cependant, cette approche a atteint ses limites en raison de la consommation d'énergie et de la dissipation thermique accrues. Plutôt que de rendre un seul cœur plus rapide, les fabricants ont commencé à ajouter plus de cœurs pour gérer davantage de tâches en parallèle.

Pipeline Architecture and Superscalar Processing

Parallelism = Performance + Energy Savings

- **Plus de travail par cycle:** Moins besoin d'augmenter la fréquence d'horloge, ce qui réduit la consommation d'énergie.

- **Utilisation efficace des unités CPU:** Évite le gaspillage des ressources.
- **Importance dans les environnements mobiles et cloud:** Les économies d'énergie sont cruciales pour prolonger la durée de vie des batteries et réduire les coûts opérationnels.
 - **Source:** <https://www.arm.com/why-arm/architecture/pipeline-processing>

Superscalar Processing

- **Exécution simultanée d'instructions indépendantes:** Les processeurs superscalaires peuvent exécuter plusieurs instructions en parallèle, à condition qu'elles soient indépendantes.
- **Exemple:** Un CPU avec 2 ALUs (Arithmetic Logic Units) peut effectuer deux opérations mathématiques simultanément, comme l'addition et la soustraction de différents nombres.
- **Presque tous les CPU modernes sont superscalaires:** Cela permet d'améliorer les performances sans augmenter la fréquence d'horloge.
 - **Source:** https://en.wikipedia.org/wiki/Superscalar_processor

Pipeline

- **Réduction du temps et de la consommation d'énergie:** Le pipelining permet d'améliorer la vitesse de traitement sans augmenter la fréquence d'horloge.
- **Exemple concret:** Dans une chaîne de montage de sandwiches, chaque étape (préparation du pain, ajout des légumes, etc.) est effectuée en parallèle pour différents sandwiches, ce qui accélère la production globale.
 - **Source:** <https://www.geeksforgeeks.org/pipelining-in-computer-architecture/>

Real Life Example

- **Exemple de pipeline:** Dans un restaurant, un cuisinier prépare le pain, un autre ajoute les légumes, un autre ajoute la sauce, etc. Chaque étape est effectuée en parallèle pour différents sandwiches, ce qui accélère le service.

Back to our Initial Problem

- **Problème initial:** Comment effectuer plus d'une instruction à la fois dans un seul cœur de CPU?
- **Solutions:**
 - **Pipeline:** Superposition des instructions pour maximiser l'utilisation des unités de traitement.
 - **Exécution superscalaire:** Exécution de plusieurs instructions simultanément, mais de manière différente du pipelining.

Conclusion

Les processeurs multicœurs et les architectures superscalaires ont révolutionné l'informatique en permettant une exécution plus rapide et plus efficace des tâches. Ces technologies sont essentielles pour répondre aux besoins croissants en puissance de calcul tout en optimisant la consommation d'énergie.

Von Neumann and Harvard Architectures

The Von Neumann Architecture

Key Features

L'architecture Von Neumann est basée sur les principes suivants :

- **Mémoire unique** : Une seule mémoire est utilisée pour stocker à la fois les instructions et les données.
- **Bus partagé** : Le CPU utilise le même bus pour récupérer les instructions et lire/écrire les données.
- **Goulot d'étranglement** : Comme le CPU et la mémoire partagent un seul bus, une seule opération peut se produire à la fois entre le CPU et la mémoire. Cela signifie que le CPU doit attendre pendant que les données sont lues ou écrites.

– Source : https://en.wikipedia.org/wiki/Von_Neumann_architecture

Historical Context

L'architecture Von Neumann a vu le jour en 1945 dans le "First Draft of a Report on the EDVAC" (Electronic Discrete Variable Automatic Computer). Ce rapport décrivait pour la première fois un système informatique composé de "blocs" ou "organes" :

- **Unité centrale de traitement (CPU)** : Pour effectuer les calculs.
- **Mémoire** : Pour stocker les instructions et les données.
- **Unités d'entrée/sortie** : Pour interagir avec l'extérieur.

– Source : <https://history.computer.org/pioneers/neumann.html>

The Harvard Architecture

Key Features

L'architecture Harvard se distingue par :

- **Mémoires séparées** : Les instructions et les données sont stockées dans des mémoires distinctes.
- **Bus séparés** : Le CPU utilise des bus différents pour accéder aux instructions et aux données, ce qui permet un accès simultané.
- **Exécution parallèle** : Comme les instructions et les données peuvent être accédées simultanément, il n'y a pas de temps d'attente, ce qui accélère l'exécution.

– Source : https://en.wikipedia.org/wiki/Harvard_architecture

Historical Context

L'architecture Harvard a été développée dans le cadre d'un projet secret de l'IBM à la fin de la Seconde Guerre mondiale. Howard Aiken, un pionnier de l'informatique, a conçu cette architecture pour construire une machine de calcul puissante.

Comparison

Von Neumann vs. Harvard

Caractéristique	Von Neumann	Harvard
Système de mémoire	Partagé	Séparé
Bus d'instructions et de données	Unique (partagé)	Séparé
Vitesse	Plus lente (goulot d'étranglement)	Plus rapide (accès parallèle)
Complexité	Plus simple, moins chère	Plus complexe, plus rapide
Cas d'utilisation	Informatique générale	Systèmes embarqués, microcontrôleurs
Flexibilité	Plus facile à programmer et gérer	Nécessite une gestion mémoire personnalisée

Explication

- **Von Neumann** : Idéale pour les systèmes où la flexibilité et la simplicité sont prioritaires, comme les ordinateurs personnels et les serveurs.
- **Harvard** : Préférée pour les systèmes embarqués et les applications en temps réel où la rapidité et l'efficacité énergétique sont cruciales.

In Reality, It's More Complex

Hybrid Architectures

Dans la pratique, les architectures modernes sont souvent hybrides :

- **Processus matériel** : Les instructions et les données sont gérées via des chemins physiques distincts.
- **Niveau d'abstraction de la programmation** : Un modèle de mémoire unifié où le code et les données partagent le même espace d'adressage.
- **Exemple** : Les processeurs x86 modernes interprètent le code dans une architecture hybride, combinant les avantages des architectures Von Neumann et Harvard.

– Source : <https://www.intel.com/content/www/us/en/architecture-and-technology/64-ia-32-architecture.html>

The Harvard Architectures

Applications

L'architecture Harvard est principalement utilisée dans des applications spécifiques :

- **Systèmes embarqués** : Comme les microcontrôleurs.
- **Applications en temps réel** : Où la rapidité et la fiabilité sont essentielles.
- **Processeurs de signaux numériques (DSPs)** : Pour le traitement rapide des signaux.

– Source : <https://www.ti.com/dsp>

Why Hasn't Harvard Taken Over?

Historical Reasons

Bien que l'architecture Harvard semble plus performante, elle n'a pas remplacé l'architecture Von Neumann pour plusieurs raisons :

- **Coût** : L'architecture Von Neumann était moins chère à produire.
- **Compatibilité** : Les processeurs x86, basés sur Von Neumann, dominaient déjà le marché des PC.
- **Exemple** : Le processeur Intel 8086, sorti en 1978, utilisait une architecture Von Neumann et a marqué le début de la domination des PC compatibles x86.

– **Source** : <https://www.intel.com/content/www/us/en/history/museum-story-of-intel-4004.html>

Conclusion

Les architectures Von Neumann et Harvard ont chacune leurs avantages et inconvénients. Alors que l'architecture Von Neumann domine dans les systèmes informatiques généraux, l'architecture Harvard est préférée pour les systèmes embarqués et les applications en temps réel. Les architectures modernes combinent souvent des éléments des deux pour optimiser les performances et l'efficacité énergétique.

Types of Memory: RAM, ROM, Flash, Cache

ROM (Read-Only Memory)

Examples of Use

La mémoire ROM est utilisée dans divers contextes où la stabilité et la permanence des données sont essentielles :

- **Dispositifs mobiles** : Stocke des éléments critiques comme les chargeurs de démarrage pour démarrer le système.
- **Systèmes automobiles** : Empêche les falsifications ou les pertes de données, comme les compteurs kilométriques.
- **Consoles de jeux vidéo** : Les anciennes cartouches (ex. : Nintendo NES) stockaient toutes les instructions de jeu dans des mémoires ROM.
- **CD, DVD, Blu-ray** : Ce sont des formes de ROM optiques.

– **Source** : <https://www.explainthatstuff.com/how-cd-drives-work.html>

Characteristics

- **Non-volatile** : Conserve les données même lorsque l'alimentation est coupée.
- **Lecture seule** : Généralement, elle ne peut pas être modifiée (sauf avec difficulté ou par flashage).
- **Stocke des instructions permanentes** : Comme le firmware ou le code de démarrage.
- **Analogie** : La ROM est comme "un livre en verre", une fois que quelque chose y est écrit, il y reste jusqu'à ce que le livre se brise.

– **Source** : <https://www.computerhope.com/jargon/r/rom.htm>

RAM (Random Access Memory)

Power Use

La RAM est une mémoire volatile qui consomme plus d'énergie que la ROM ou la Flash :

- **Consommation moyenne** : Environ 5 watts par barrette de RAM.
- **Variation** : Les RAM plus rapides peuvent consommer plus.
- **Paramètres de tension** : Les réglages de tension peuvent augmenter la consommation.
- **Exemple** : La LPDDR (Low Power Double Data Rate) peut passer de 2.5V à 1.8V pour réduire la consommation.

– Source : <https://www.kingston.com/en/blog/pc-memory/dram-power-consumption>

Examples of Use

La RAM est utilisée pour stocker temporairement des données actives :

- **Navigateurs web** : Stocke les données de chaque onglet ouvert.
- **Jeux vidéo** : Stocke les modèles 3D, les informations comme la santé, le compteur de balles, l'inventaire.
- **Cybersécurité** : Les clés de déchiffrement pour les connexions sans fil peuvent être stockées en RAM.

– Source : <https://www.crucial.fr/articles/about-memory/what-does-ram-do>

Characteristics

- **Mémoire volatile** : Les données sont perdues lorsque l'alimentation est coupée.
- **Très rapide** : Accès direct pour le CPU pendant l'exécution des programmes.
- **Stockage temporaire** : Contient les données et instructions des programmes en cours d'exécution.

– Source : <https://www.techopedia.com/definition/24998/random-access-memory-ram>

Flash Memory

Examples of Use

La mémoire Flash est utilisée pour le stockage de données persistantes et portables :

- **Clés USB** : Pour le transport de fichiers.
- **Cartes mémoire** : Utilisées dans les appareils photo et smartphones.
- **SSD** : Pour le stockage interne des ordinateurs.

– Source : <https://www.explainthatstuff.com/how-flash-memory-works.html>

Characteristics

- **Non-volatile** : Conserve les données sans alimentation.
- **Réinscriptible** : Peut être effacée et réécrite un nombre limité de fois.
- **Rapide et compacte** : Plus rapide que les disques durs mécaniques, mais plus lente que la RAM.

– Source : <https://www.computerhope.com/jargon/f/flashmem.htm>

Cache Memory

Examples of Use

La mémoire cache est utilisée pour accélérer l'accès aux données fréquemment utilisées :

- **Processeurs** : Stocke les instructions et données fréquemment utilisées pour réduire les temps d'accès à la RAM.
- **Systèmes embarqués** : Pour améliorer les performances des microcontrôleurs.
 - Source : <https://www.intel.com/content/www/us/en/support/articles/000005657/processors.html>

Characteristics

- **Très rapide** : Plus rapide que la RAM.
- **Petite capacité** : Généralement de quelques kilo-octets à quelques méga-octets.
- **Volatile** : Les données sont perdues lorsque l'alimentation est coupée.
 - Source : <https://www.techopedia.com/definition/19/cache-memory>

Comparison

Caractéristique	RAM	ROM	Flash
Volatile	Oui	Non	Non
Réinscriptible	Oui	Non (ou rarement)	Oui
Vitesse	Très rapide	Modérée	Plus lente que la RAM
Cas d'utilisation	Applications actives	Firmware, code de démarrage	Stockage de fichiers, applications mobiles

Why the Use of Memory?

Memory Hierarchy

Les systèmes numériques utilisent divers types de mémoire pour équilibrer vitesse, coût et capacité :

- **Registers** : Très rapides, mais de très petite taille.
- **Cache** : Rapide, petite capacité.
- **RAM** : Rapide, capacité modérée.
- **Flash/SSD** : Plus lente, grande capacité, non volatile.
- **HDD/Cloud** : Très lente, très grande capacité, non volatile.
 - Source : <https://www.geeksforgeeks.org/memory-hierarchy-in-computer-architecture/>

Power Usage / Exercise

Power Consumption

- **Accès à la RAM** : Consomme plus d'énergie que l'accès au cache.
- **Accès au SSD/HDD** : Beaucoup plus lent et énergivore que l'accès à la RAM.
- **Accès au Cloud** : Encore plus lent et énergivore.
 - Source : <https://www.anandtech.com/show/6326/memory-power-consumption-on-intel-snbivb>

Conclusion

Les différents types de mémoire jouent des rôles complémentaires dans les systèmes informatiques modernes. La RAM est utilisée pour les données actives, la ROM pour les instructions permanentes, la Flash pour le stockage portable et persistant, et le cache pour accélérer l'accès aux données fréquemment utilisées. Comprendre ces différences est essentiel pour optimiser les performances et l'efficacité énergétique des systèmes.

Bus Systems and Communication Protocols

How to Recognize USB Types

Les connecteurs USB peuvent être reconnus par leur forme, leurs symboles et leurs couleurs :

- **Type A** : Rectangulaire, utilisé principalement pour les périphériques hôtes comme les ordinateurs.
- **Type B** : Carré avec des coins biseautés, utilisé principalement pour les imprimantes et scanners pour garantir un branchement correct (à la fois pour les données et l'alimentation).
- **Type C** : Symétrique et réversible, utilisé pour les appareils modernes comme les smartphones et les ordinateurs portables.

– Source : <https://www.usb.org/>

Not All USBs Are Equal

Les normes USB varient en termes de vitesse de bus et de capacité de transfert de données :

Système	Vitesse du Bus	Nom Commun
USB 1.1	12 Mbps	Full Speed
USB 2.0	480 Mbps	High-Speed
USB 3.0/3.1	5–10 Gbps	SuperSpeed
USB 3.2/4.0	20–40 Gbps	SuperSpeed+
USB-C	Type de connecteur (pas une vitesse)	Supports USB 3.2, 4, Thunderbolt

- **Note** : USB-PD (Power Delivery) supporte jusqu'à 240W de livraison de puissance, et certaines versions dans les tablettes supportent même 230V.

– Source : <https://www.usb.org/documents>

Focus on USB

Universal Serial Bus

- **Conçu pour remplacer de nombreux ports hérités** : Avec un connecteur standard et flexible (rappelons que l'USB-C peut gérer les données, la vidéo, etc.).
 - **Communication série** : Transfère les données un bit à la fois, mais à haute vitesse.
 - **Pourquoi l'USB a-t-il conquis le monde si rapidement ?**
 - **Plug-and-play** : Facilité d'utilisation sans configuration complexe.
 - **Hot-swappable** : Peut être branché et débranché sans redémarrer l'ordinateur.
 - **Alimentation + transfert de données** : Un seul connecteur pour les deux.
 - **Compatibilité ascendante** : Bien que certains problèmes de pilotes persistent, notamment avec les imprimantes.
- Source : <https://www.explainthatstuff.com/how-usb-works.html>

Bus System

Recap

Un bus est un chemin pour transférer des données, des adresses et des signaux de contrôle entre les composants d'un système informatique. Il peut être :

- **Série** : Une seule ligne de données.
- **Parallèle** : Plusieurs lignes ou lignes dédiées.
- **Partagé** : Plusieurs périphériques "parlent" dessus.

Bandwidth and Latency

La bande passante (quantité de données transmises) et la latence (délai entre la cause et l'effet) dépendent du protocole utilisé sur le bus et de ses caractéristiques physiques.

Types of I/O Devices and Interfaces

Power Consumption Example (using ESP32)

- **WiFi Power Consumption** :
 - **Mode WiFi actif** : 240 mA lors de la transmission/réception.
 - **Fonctionnement normal** : 43 mA lorsqu'il est connecté mais non actif.
 - **Mode veille profonde** : 8-15 μ A lorsqu'il n'est pas nécessaire.
- **Bluetooth Power Consumption** :
 - **Mode BLE actif** : 10-50 mA (moins que le WiFi).
 - **Mode veille profonde** : 8-15 μ A (similaire au WiFi en veille profonde).
- **Dans les appareils IoT** : L'utilisation d'une interface comme BLE peut être 4 à 24 fois plus efficace.
 - **Source** : <https://www.espressif.com/en/products/socs/esp32>

When to Use Each One?

- **WiFi** : Utilisé lorsque vous avez besoin de :
 - Accès à Internet (navigation web, streaming, téléchargements).
 - Transfert de données à haute vitesse, connexions à longue portée.
 - Partage de périphériques multiples.
- **Bluetooth** : Utilisé lorsque vous avez besoin de :
 - Connexions entre appareils.
 - Faible consommation d'énergie (appareils sur batterie).
 - Partage simple de données (streaming audio).

Wireless Interfaces

- **Wi-Fi (Wireless Fidelity) :**
 - **Wi-Fi 7 (2020) :** Bande passante maximale de 46 Gbps, portée maximale de 100 mètres dans des installations optimisées.
- **Bluetooth :**
 - **Note :** Bluetooth et BLE (Low Energy) sont deux spécifications pour des usages différents.
 - **Débit maximal :** 2-3 Mbps.
 - **Portée :** De contact rapproché à 100 mètres.
 - **Source :** <https://www.bluetooth.com/>

Network Interfaces

- **Interfaces filaires :**
 - **PoE (Power over Ethernet) :** Permet d'alimenter un appareil via un câble Ethernet.
 - * **15 W / 48 V** pour le PoE standard et **30 W / 48 V** pour le PoE+.
 - **SFP (Small Form-factor Pluggable) :** Connecteur plus petit qui peut être utilisé pour le câblage en cuivre ou en fibre optique. Souvent trouvé sur les routeurs industriels.
 - * **1-10 Gbps** de débit.
 - * Peut couvrir plusieurs kilomètres (120 km testés) au lieu de 100+ mètres.
 - **Immunisé contre les interférences électromagnétiques.**
 - **Source :** <https://www.fs.com/>
- **Interfaces sans fil :**
 - **Wi-Fi :** Trouvé dans les ordinateurs portables, téléphones, réfrigérateurs intelligents.
 - **Bluetooth :** Pour des connexions à plus courte portée, comme les écouteurs, les manettes de jeu, les montres intelligentes.

Video Interfaces: HDMI, DisplayPort, VGA/DVI

- **GPPI (General Purpose Media Interface) :** Nouvelle technologie conçue en Chine.
 - **De l'Alliance Shenzhen 8K UHD Video Industry Cooperation :** Une alliance de 50 entreprises technologiques (Lenovo, Xiaomi, Hisense, etc.).
 - **Supporte un format USB-C et un format propriétaire pour les professionnels.**
 - **Capacités :**
 - * Alimenter des appareils jusqu'à 480 W.
 - * Aller jusqu'à 8K/120FPS.
 - * Supporter une bande passante jusqu'à 192 GB/s.
 - **Note :** Les spécifications Thunderbolt 4 sont encore plus élevées.
 - **Source :** <https://www.8kalliance.org/>
- **USB-C :** Supporte également le transfert vidéo, mais tous les câbles USB-C ne sont pas identiques.
 - **Vérifiez :**

- * Mode alternatif DisplayPort (DP Alt Mode).
- * Support Thunderbolt 3.
- * Mode de compatibilité USB-C.
- **Capacités :**
 - * Enchaînement de moniteurs.
 - * Jusqu'à 4K/60FPS.
 - * 40 Gb/s.
- **Note :** Les spécifications Thunderbolt 4 sont encore plus élevées.
- **VGA/DVI :** Interfaces héritées.
 - **VGA :** Connecteur analogique plus ancien.
 - **DVI :** Connecteur numérique plus ancien.
 - **Progressivement abandonnés** dans la plupart des appareils modernes, résolution plus faible et pas de support audio.
 - **Devaient être verrouillés** pour garantir la transmission correcte du signal dans certaines situations.
- **DisplayPort :**
 - **Bande passante plus élevée** que le HDMI (meilleur pour les moniteurs PC).
 - **Pas de licence nécessaire** pour l'utiliser, mécanisme de verrouillage natif.
 - **Supporte l'enchaînement de moniteurs** (DP out).
 - **DisplayPort 2.1 :**
 - * Supporte 8K@80fps et 16K@80fps.
 - * Supporte le HDR dynamique.
 - * Bande passante de 77 Gb/s.
 - * Fonction de chiffrement native DPCP.
- **HDMI (High-Definition Multimedia Interface) :**
 - **Combine audio + vidéo** en un seul câble.
 - **Largement utilisé** pour les téléviseurs, moniteurs, consoles de jeux.
 - **Nécessite une licence** pour être utilisé.
 - **Pas de système de verrouillage natif.**
 - **HDMI 2.1B :**
 - * Supporte 8K@80fps et 4K@120fps.
 - * Supporte le HDR dynamique.
 - * Bande passante de 48 Gb/s.

Dedicated Output Devices

Parfois, vous avez besoin de périphériques de sortie si spécialisés qu'il faut créer un domaine entier de nouveaux appareils. Voici quelques exemples :

- **Moniteurs :** Sortie visuelle, LCD, LED, microLED, gamma.
- **Imprimantes :** Papier, autocollants.
- **Haut-parleurs/écouteurs :** Sortie audio, sonar (si en complément d'un capteur).

General Output Devices

- **Imprimantes 3D** : Utilisent du filament, de l'encre ou de la résine pour produire des objets 3D.
- **Microphones 3D** : Utilisent des combinaisons de projecteurs à haute gamme et à grande vitesse pour induire un effet 3D sans besoin de lunettes 3D.

Dedicated Input Devices

Parfois, vous avez besoin de périphériques d'entrée si spécialisés qu'il faut créer un domaine entier de nouveaux appareils. Voici quelques exemples :

- **Claviers/souris/trackpads** : Détectent les mouvements et les pressions de touches.
- **Écrans tactiles** : Combinent entrée et sortie.
- **Caméras et microphones** : Entrée pour les appels vidéo, enregistrements, vision/speech AI, sécurité.
- **Manettes de jeu** : Entrées pour les consoles de jeux et les PC.

General Input Devices

Certains peuvent être spécialisés tout en utilisant des technologies précédemment listées :

- **Scanners 3D** : Utilisent un système de matrice pour des scans 3D à haute résolution.
- **Souris MMO** : Ajoutent 12 boutons sur le côté pour vos barres de raccourcis.

What Are I/O Devices?

I/O signifie Inputs/Outputs, cette désignation peut être trouvée dans :

- **Ordinateurs**
- **Microélectronique**
- **Automatisation**
- **Robotique**

Conclusion

Les systèmes de bus et les protocoles de communication sont essentiels pour assurer une communication efficace entre les différents composants d'un système informatique. Les normes comme l'USB, le WiFi, le Bluetooth, et les interfaces vidéo comme HDMI et DisplayPort jouent un rôle crucial dans la connectivité et la performance des appareils modernes.

Specialized Architectures (GPU, TPU, FPGA)

General Comparison

Caractéristique	CPU	GPU	TPU	FPGA
Type de tâche	Usage général	Hautement parallèle, non mono-tâche	Spécifique à l'IA	Reconfigurable
Vitesse	Moyenne	Élevée pour les tâches parallèles	Très élevée pour l'IA	Moyenne
Flexibilité	Élevée	Modérée à faible	Faible (fonction fixe)	Élevée
Consommation d'énergie	Élevée	Élevée à extrêmement élevée	Faible à moyenne	Faible
Cas d'utilisation	Applications, OS	Jeux, IA, édition, 3D	Inférence/entraînement d'IA	Périphériques embarqués

GPUs (Graphics Processing Units)

Overview

Les GPU ont été initialement conçus pour rendre les graphiques en gérant des calculs massivement parallèles (jeux vidéo, modélisation 3D, effets cinématographiques, simulations de particules pour l'aérodynamisme, etc.).

Characteristics

- **Cœurs multiples** : Les GPU ont des centaines, voire des milliers de cœurs plus petits et plus simples, dédiés à des tâches spécifiques.
- **Efficacité dans le traitement des matrices et des vecteurs** : Idéal pour les opérations de base en IA et en apprentissage automatique.
- **Utilisation moderne** : Les GPU sont utilisés pour l'inférence et l'entraînement des modèles d'IA, ainsi que pour le traitement d'images et de vidéos.

– Source : <https://www.nvidia.com/en-us/data-center/gpu-accelerated-applications/>

Examples

- **NVIDIA CUDA Cores** : Utilisés pour accélérer les calculs en virgule flottante et en entiers.
- **AMD ROCm** : Technologie similaire à CUDA pour les GPU AMD.

– Source : <https://rocmdocs.amd.com/en/latest/>

TPUs (Tensor Processing Units)

Overview

Les TPU sont des circuits intégrés spécifiques à une application (ASIC) construits par Google, conçus spécifiquement pour les opérations de tenseurs utilisées dans les réseaux de neurones.

Characteristics

- **Matériel spécialisé** : Les TPU sont optimisés pour les calculs de tenseurs, ce qui les rend très efficaces pour les tâches d'IA.
- **Efficacité énergétique** : Selon Google, les TPU sont 2,5 fois plus efficaces que les GPU de coût similaire pour différentes opérations.
- **Utilisation énergétique** : 3 à 10 fois plus efficace pour les mêmes calculs.

– Source : <https://cloud.google.com/tpu/docs/overview>

Examples

- **Google Colab** : Offre un accès gratuit aux TPU pour les utilisateurs.

– Source : <https://colab.research.google.com/notebooks/tpu.ipynb>

FPGAs (Field-Programmable Gate Arrays)

Overview

Les FPGA sont des puces que l'on peut reprogrammer au niveau matériel après leur fabrication. Ils combinent la vitesse du matériel avec la flexibilité du logiciel.

Characteristics

- **Reprogrammables** : Les FPGA peuvent être configurés pour exécuter différentes tâches matérielles.
- **Utilisation flexible** : Idéal pour le prototypage rapide et les applications nécessitant des mises à jour fréquentes du matériel.
- **Applications** : Traitement du signal, cryptage, inférence légère d'IA, aérospatial, automatisation industrielle, IA de bordure.
 - Source : <https://www.xilinx.com/products/silicon-devices/fpga/what-is-an-fpga.html>

Examples

- **Altera FP300** : Premier modèle industriel produit en 1983 par Altera (une filiale d'Intel).
- **Utilisation moderne** : Les FPGA sont utilisés dans les systèmes embarqués pour des tâches spécifiques nécessitant une faible consommation d'énergie et une personnalisation.

NPU (Neural Processing Unit)

Overview

Les NPU sont une technologie relativement nouvelle, principalement utilisée dans les ordinateurs portables, smartphones et certaines cartes mères. Ils sont spécialisés dans les calculs liés aux réseaux de neurones.

Characteristics

- **Matériel spécialisé** : Conçu pour imiter les fonctions de traitement du cerveau humain.
- **Calculs de réseaux de neurones** : Optimisé pour les opérations de tenseurs, vecteurs et scalaires.
- **Utilisation énergétique** : Souvent combiné avec des systèmes sur puce (SoC).
- **Précision réduite** : Utilise souvent des calculs moins précis (opérations sur 8 bits) mais est spécialement conçu pour la résolution de problèmes et l'utilisation de modèles proches du langage humain.

– Source : <https://www.qualcomm.com/news/onq/2021/05/18/what-neural-processing-unit-and-why-it-m>

Use Cases

- **Robotique et véhicules autonomes** : Pour la fusion et l'interprétation des données lidar, ultrasoniques et vidéo.
- **Informatique de bordure** : Réduction de la charge sur les centres de données avec des modèles de langage (LM) embarqués.
- **Services comme Windows Recall** : Utilisent ce matériel pour offrir des fonctionnalités avancées aux utilisateurs.

Optimizing for Parallel Tasks

Type	Utilisé par	Exemple
Parallélisme de tâche	CPU multicœurs	Exécuter Krita + Firefox + Discord
Parallélisme de données	GPU multicœurs	Appliquer des filtres à un grand ensemble d'images

Explanation

- **Parallélisme** : Permet une exécution plus efficace à fréquence plus basse.
- **GPU et puces multicœurs spécialisées** : Peuvent offrir des performances par watt plus élevées.
- **Accélérateurs d'IA multicœurs** : Permettent une inférence à faible puissance à la périphérie du système, réduisant la charge sur les serveurs centraux pour une informatique globalement plus économe en énergie.

ASIC (Application-Specific Integrated Circuit)

Overview

Les ASIC sont des puces conçues pour des tâches spécifiques. Ils sont souvent utilisés dans des applications où la performance et l'efficacité énergétique sont critiques.

Examples

- **Mineurs de cryptomonnaie** : Matériel dédié aux opérations spécifiques d'une cryptomonnaie et de son algorithme de hachage.
- **Évolution** : Au début du boom des cryptomonnaies, les GPU étaient utilisés en raison de leur flexibilité et disponibilité, mais ont été remplacés par du matériel dédié (maintenant remplacé par l'IA).

Manycore Systems

Overview

Les systèmes manycore sont conçus pour traiter un grand nombre de tâches similaires en parallèle, ce qui est connu sous le nom de parallélisme de données.

Characteristics

- **GPU** : Utilisés dans le gaming, l'apprentissage automatique et le calcul scientifique.
- **Accélérateurs d'IA** : Comme les TPU de Google et les cœurs CUDA de NVIDIA.
- **Calcul haute performance (HPC)** : Utilisé dans les supercalculateurs.

Examples

Modèle	RTX 3070	RTX 3070 Ti
Année de lancement	2020	2021
ROPs	96	96
TMUs	184	280
Tensor Cores	184	280
Ray Tracing Cores	46	70

Energy Consumption / Cost

Single Core

- **Temps en heures** : $2.778 \times 3,500 = 0.772$ heures.
- **Énergie \times Puissance** : $65 \times 0.772 = 0.502$ Wh = 0.0502 kWh.
- **Tarif en kWh** : $0.0502 \times 0.12 = 0.0060$ euros.

Multi-core

- **Temps en heures** : $347 \times 3,600 = 0.096$ heures.
- **Énergie \times Puissance** : $125 \times 0.096 = 0.012$ kWh = 0.012 kWh.
- **Tarif en kWh** : $0.012 \times 0.12 = 0.0014$ euros.

Results Analysis

- **Efficacité énergétique** : Le multicœur est plus économe en énergie car, malgré une consommation instantanée plus élevée, il termine la tâche beaucoup plus rapidement.
- **Économies d'énergie** : $0.0502 - 0.012 = 0.0382$ kWh économisés (réduction de 76%).
- **Économies annuelles** : $(0.0080 \times 0.0014) \times 365 = 1.68$ euros par an.
- **À grande échelle** : Bien que cela puisse sembler peu, à l'échelle de millions/milliards de CPU avec des charges croissantes, les économies d'énergie s'accumulent rapidement.

Conclusion

Les architectures spécialisées comme les GPU, TPU, FPGA et NPU jouent un rôle crucial dans l'optimisation des tâches spécifiques, offrant des performances et une efficacité énergétique supérieures par rapport aux CPU traditionnels. Leur utilisation est essentielle dans des domaines tels que l'IA, le calcul haute performance, et les systèmes embarqués.

Persistent Storage (HDDs, SSDs, NVMe)

Overview

Le stockage persistant est essentiel pour conserver les données de manière permanente, même lorsque l'alimentation est coupée. Les principales technologies de stockage persistant incluent les disques durs (HDDs), les disques SSD (Solid State Drives), et les disques NVMe (Non-Volatile Memory Express).

HDD vs SSD

HDD (Hard Disk Drive)

- **Technologie** : Utilise des plateaux magnétiques en rotation et une tête de lecture/écriture pour accéder aux données.
- **Caractéristiques** :
 - **Mécanique** : Composants mobiles qui peuvent être sensibles aux chocs.
 - **Vitesse** : Plus lente que les SSDs en raison des limitations mécaniques.
 - **Capacité** : Généralement plus grande capacité de stockage à moindre coût.
 - **Utilisation** : Idéal pour le stockage de masse et les sauvegardes.
- **Exemple** : Temps de démarrage de Windows 10 (64 bits) sur un HDD : plus de 32,7 secondes.
 - **Source** : <https://www.seagate.com/fr/fr/tech-insights/hdd-vs-ssd-master-ti/>

SSD (Solid State Drive)

- **Technologie** : Utilise des puces de mémoire flash pour stocker les données, sans pièces mobiles.
- **Caractéristiques** :
 - **Électronique** : Pas de pièces mobiles, donc plus résistant aux chocs.
 - **Vitesse** : Accès aux données plus rapide que les HDDs.
 - **Consommation d'énergie** : Moins énergivore que les HDDs.
 - **Coût** : Plus cher par gigaoctet que les HDDs.
- **Exemple** : Temps de démarrage de Windows 10 (64 bits) sur un SSD : environ 12,5 secondes.
 - **Source** : <https://www.kingston.com/fr/blog/pc-performance/ssd-vs-hdd>

NVMe (Non-Volatile Memory Express)

Overview

NVMe est un protocole qui connecte les SSDs directement au CPU via le bus PCIe, offrant une bande passante plus élevée et une latence plus faible que les SSDs traditionnels connectés via SATA.

Characteristics

- **Interface** : Utilise le bus PCIe pour une communication plus rapide avec le CPU.
- **Vitesse** : Offre des vitesses de lecture/écriture bien supérieures à celles des SSDs SATA.
- **Latence** : Réduit la latence grâce à une connexion directe au CPU.

Comparison

Type	Interface	Vitesse de Lecture (Mo/s)
HDD	SATA	Environ 100
SSD SATA	SATA	Environ 500
SSD NVMe	PCIe	3000-7000+

- **Source** : <https://www.pny.com/fr-FR/ssd/nvme-vs-sata>

Memory Hierarchy and Access Speed

Power Use

- **Accès à la RAM** : Consomme plus d'énergie que l'accès au cache.
- **Accès au SSD ou HDD** : Plusieurs ordres de grandeur plus lent et plus énergivore que l'accès à la RAM.
- **Accès au Cloud** : Encore plus lent et énergivore.
- **Optimisation** : Une utilisation optimisée de la mémoire conduit à de meilleures performances et à une consommation d'énergie réduite.
 - **Source** : <https://www.anandtech.com/show/6326/memory-power-consumption-on-intel-snbivb>

Memory Pyramid

Niveau	Type	Emplacement	Vitesse	Taille	Coût/bit
Registres	Registres	À l'intérieur du CPU	La plus rapide	La plus petite	Très élevé
Cache (L1-L3)	Cache	À l'intérieur du CPU	Très rapide	Petite	Très élevé
RAM (Mémoire principale)	DRAM	Externe	Très rapide	Modérée	Élevé
Flash/SSD	Stockage secondaire	Externe	Lente	Grande	Modéré
HDD/Cloud	Stockage tertiaire	Externe	La plus lente	Très grande	Le moins ch

To Remember

- **RAM** : Comme votre bureau où vous travaillez, rapide, temporaire, volatile.
- **ROM** : Comme un manuel d'instructions scellé au mur, toujours accessible, immuable.
- **Flash** : Comme une clé USB dans votre poche, réinscriptible et portable, mais plus lente que votre bureau.

Comparison

Caractéristique	RAM	ROM	Flash
Volatile	Oui	Non	Non
Réinscriptible	Oui	Non (ou rarement)	Oui
Vitesse	La plus rapide	Modérée	Plus lente que la RAM
Cas d'utilisation	Applications actives	Firmware, code de démarrage	Stockage de fichiers, applications mobiles

Conclusion

Les technologies de stockage persistant comme les HDDs, SSDs et NVMe jouent un rôle crucial dans les systèmes informatiques modernes. Les HDDs offrent une grande capacité à moindre coût, tandis que les SSDs et NVMe offrent des performances supérieures en termes de vitesse et d'efficacité énergétique. Comprendre ces différences est essentiel pour choisir la technologie de stockage appropriée en fonction des besoins spécifiques en termes de vitesse, capacité et coût.