

Cours Microcontrôleurs Avancés

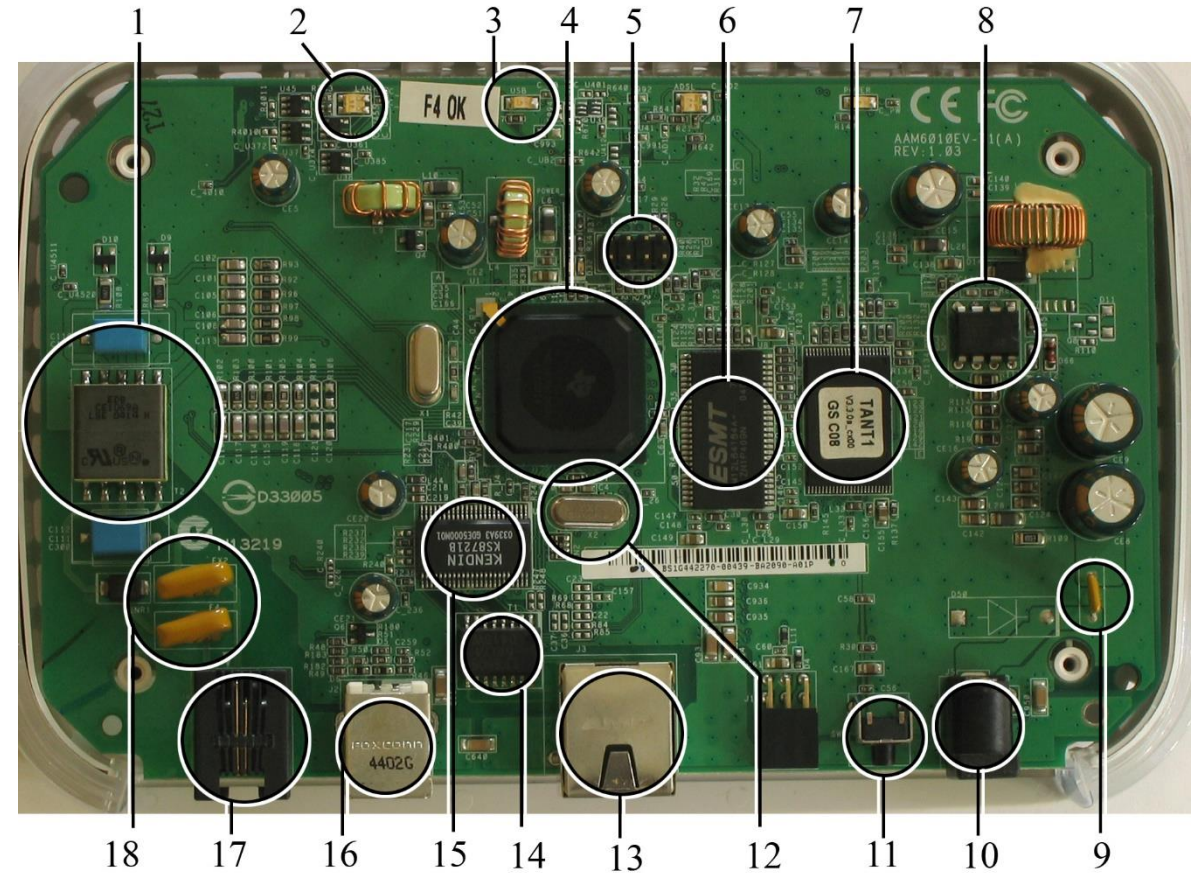
CH1: Généralités

Par:

Hassan EL FADIL

elfadil.h@gmail.com

- Un **système embarqué** est un système électronique et informatique autonome,
- Il est souvent temps réel, spécialisé dans une tâche précise.
- Le terme désigne aussi bien le matériel informatique que le logiciel utilisé.
- Encombrement réduit
- Consommation d'énergie limitée.



Intérieur d'un modem/routeur ADSL. Un exemple de système embarqué. Les parties numérotées comprennent : le microprocesseur (4), la RAM (6) et la mémoire flash (7).

- Les systèmes embarqués modernes sont souvent basés sur des microcontrôleurs (c'est-à-dire des microprocesseurs avec mémoire intégrée et interfaces périphériques),
- mais les microprocesseurs ordinaires (utilisant des puces externes pour la mémoire et les circuits d'interface périphérique) sont également courants, en particulier dans les systèmes plus complexes.
- Dans les deux cas, le ou les processeurs utilisés peuvent être de types allant de ceux à usage général à ceux spécialisés dans une certaine classe de calculs, ou même conçus sur mesure pour l'application en question.
- Une classe standard commune de processeurs dédiés est le processeur de signal numérique (DSP).

- Lecteurs MP3, PDA, téléphones portables, cartes a puce,
- Consoles de jeux
- Avionique / Automobile / Ferroviaire :
- Contrôle de vitesse, Pilote automatique, Tableau de bord
- Telecommunications
- Modems, routeurs, satellites
- Medical
- Pacemaker, imagerie, robots
- Militaire
- Teleguidage, radar, drones



- **Ressources contraintes**: énergie, mémoire, capacités calcul (Ex: pas de FPU)
- **Systèmes critiques**: garanties de sûreté, déterminisme et tolérance aux fautes
- **Systèmes temps réel**: Ex: contrôle de vitesse d'un train
- **Interfaçage avec des périphériques externes**: capteurs de température, de vitesse, communications

- Un système sur une puce, *system on a chip* : SoC,
- Système complet embarqué sur un seul circuit intégré (« puce »), pouvant comprendre de la mémoire, un ou plusieurs microprocesseurs, des périphériques d'interface, ou tout autre composant nécessaire à la réalisation de la fonction attendue.
- Il peut également comprendre de la logique, de la mémoire, des dispositifs (capteurs) mécaniques, opto-électroniques, chimiques ou biologiques et des circuits radio.
- SoC: circuit intégré qui combine différentes fonctions d'un système informatique en une seule puce.



Puce ARM Exynos sur le smartphone Nexus S de Samsung.

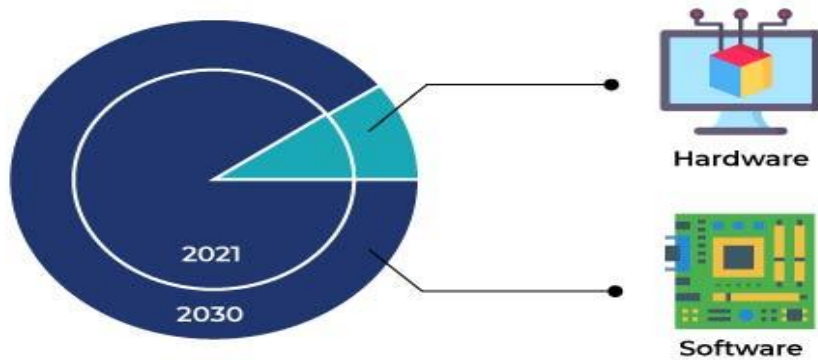
GLOBAL EMBEDDED SYSTEMS MARKET FORECAST 2022-2030

INKWOOD
RESEARCH

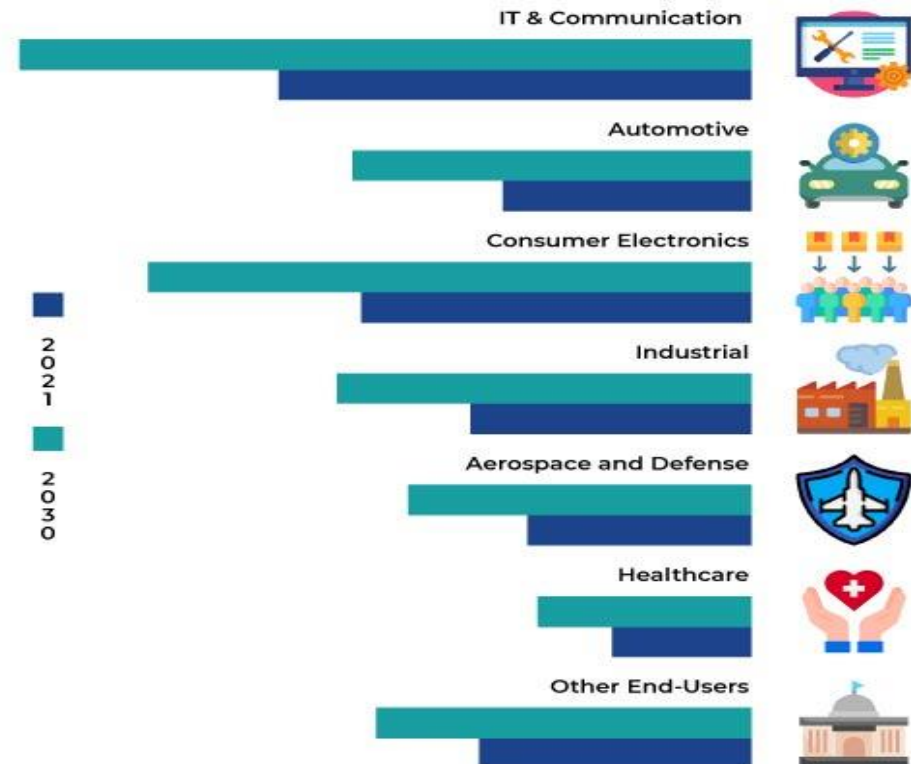
MARKET BY REGION

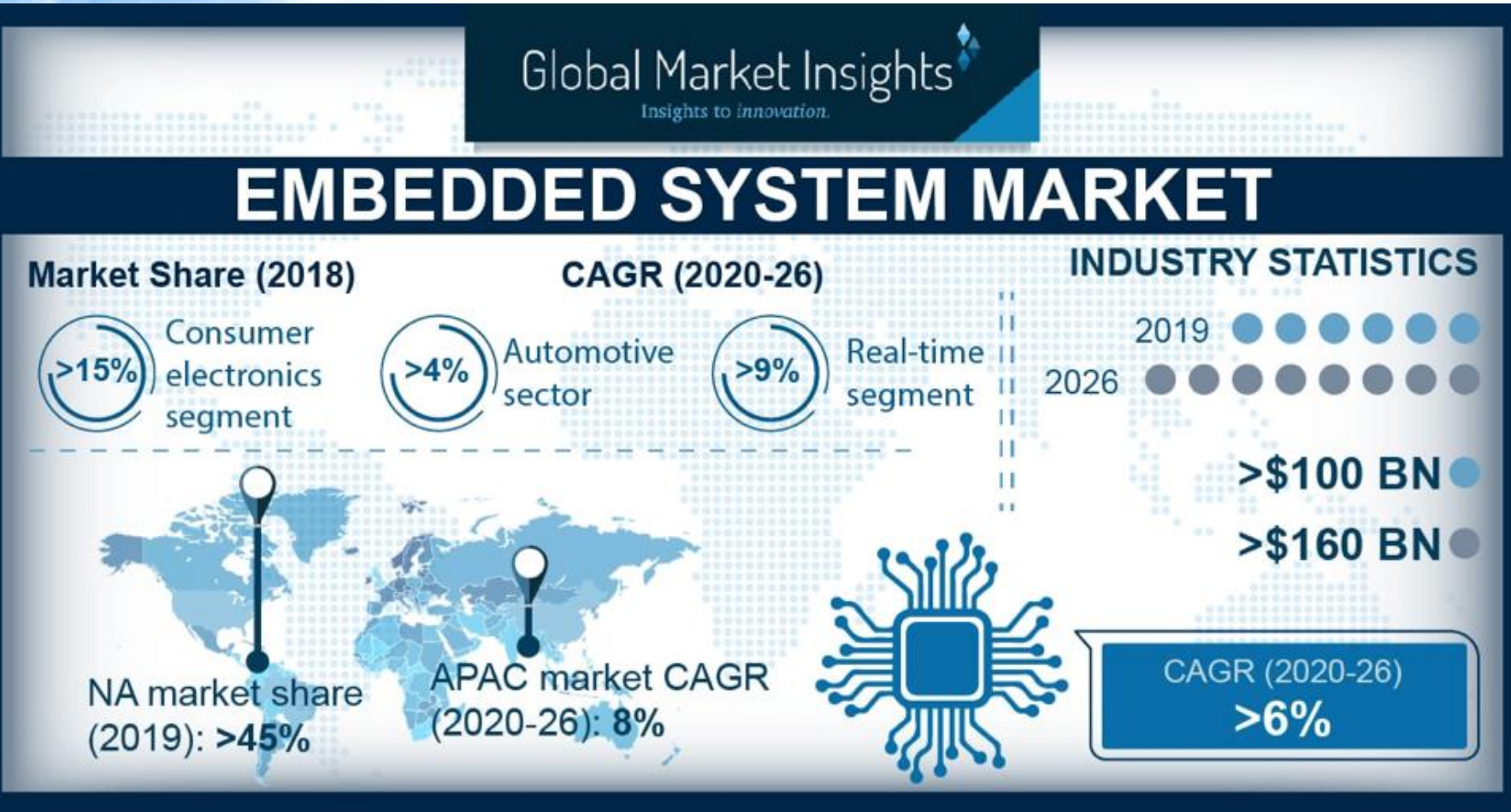


MARKET BY COMPONENT



MARKET BY END-USER

**TOP COMPANIES :** ANALOG DEVICES INC | STMICROELECTRONICS N.V. | NXP SEMICONDUCTORS N.V. | INFINEON TECHNOLOGIES AG



CAGR: « compound annual growth rate »

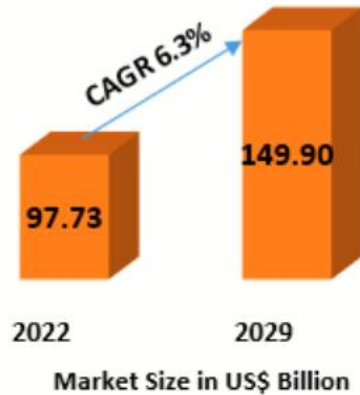
ou encore appelé taux de croissance annuel composé (TCAC)

Embedded System Market

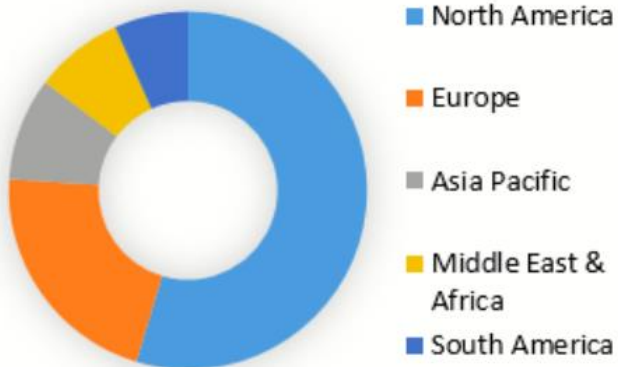


Key Players

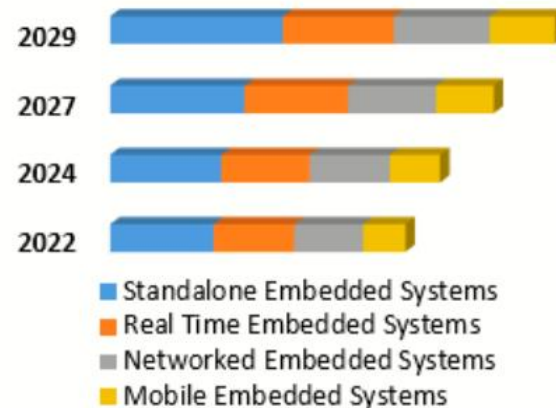
Intel Corporation (U.S.)	Infosys, Ltd. (India)
Microsoft Corporation (U.S.)	Freescale Semiconductor, Inc. (U.S.)
NXP Semiconductors (Netherlands)	Infineon Technologies AG (Germany)
Atmel Corporation (U.S.)	Texas Instruments, Inc. (U.S.)
Renesas Electronics Corporation (Japan)	Philips
HCL Technologies, Ltd. (India)	Motorola



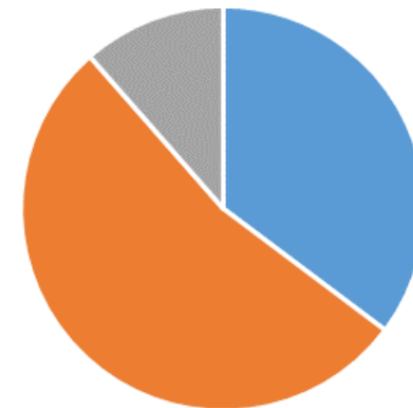
Regional Analysis in 2022 (%)



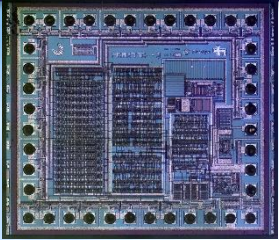
Functionality Segment Overview



Embedded System Market, by Microcontroller 2022 (%)



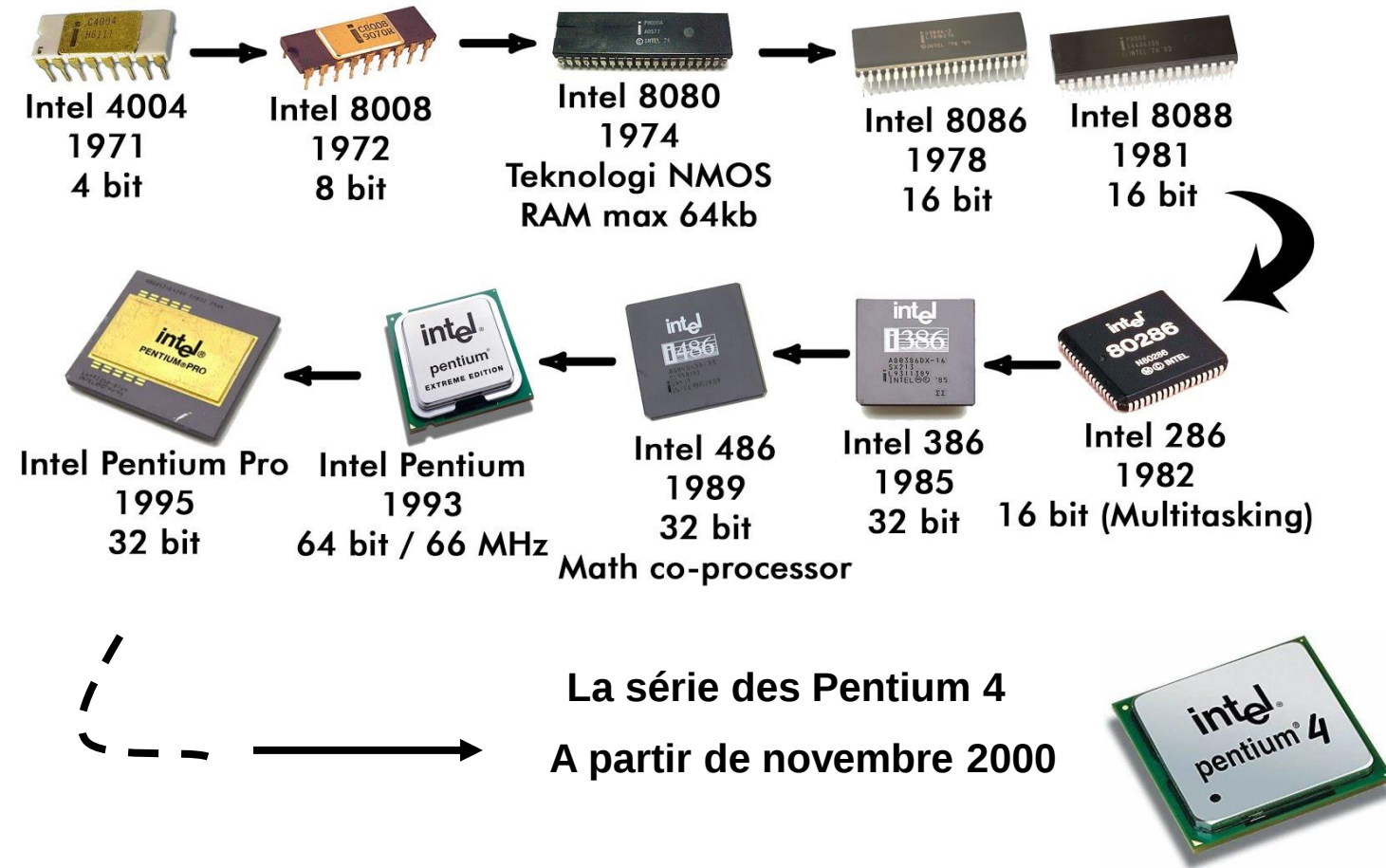
- Small Scale Embedded Systems
- Medium Scale Embedded Systems
- Large Scale Embedded Systems



Le **circuit intégré (CI)** est un **composant électronique**, basé sur un **semi-conducteur**, reproduisant une ou plusieurs fonctions électroniques plus ou moins complexes, intégrant souvent plusieurs types de **composants électroniques de base** dans un volume réduit (sur une petite plaque), rendant le circuit facile à mettre en œuvre

Nom	Signification	Année de sortie	Nombre de transistors	Nombre de portes logiques par boîtier
SSI	small-scale integration	1964	1 à 10	1 à 12
MSI	medium-scale integration	1968	10 à 500	13 à 99
LSI	large-scale integration	1971	500 à 20 000	100 à 9 999
VLSI	very large-scale integration	1980	20 000 à 1 000 000	10 000 à 99 999
ULSI	ultra large-scale integration	1984	1 000 000 et plus	100 000 et plus

Cas des Processeurs: d'Intel



À partir de 2005, Intel a progressivement abandonné l'architecture **NetBurst** des Pentium 4 (gardant les instructions 64 bits) à cause de pipelines trop longs, de performances trop faibles et de problèmes de surchauffe.



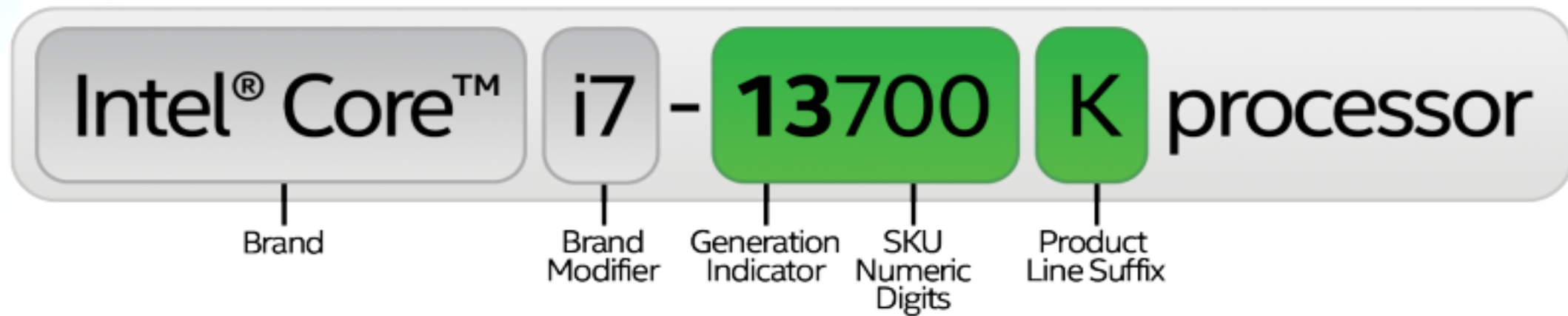
Notion de cœurs (2, 4): Core En 2006

Dual-core = processeur à 2 cœurs physiques
 Quad-core = 4 cœurs physiques

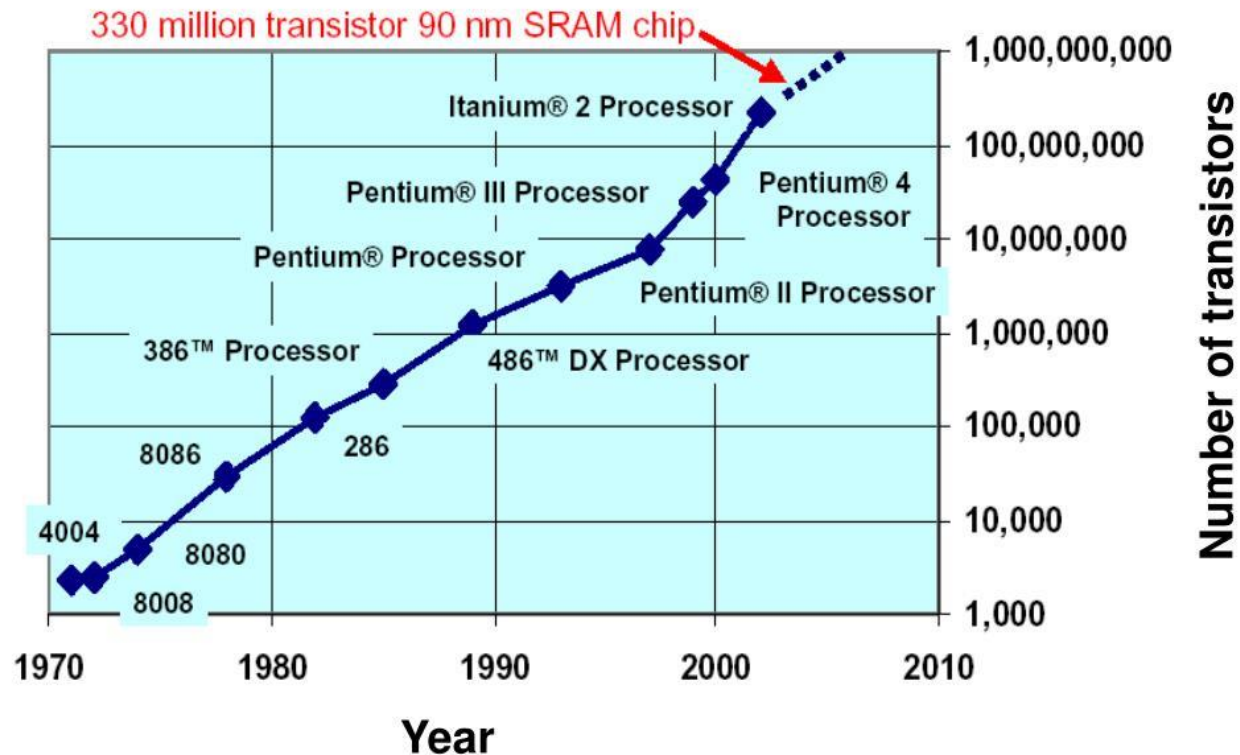


Noms et numéros des processeurs Intel:

- Les modèles: i3, i5, i7, i9
- Notion de génération



INTEL microprocessors



Le premier **microprocessor Intel 4004**:
2 300 transistors.

La capacité d'intégration des transistors et l'augmentation de la finesse de gravure ont amélioré les performances des processeurs.

Loi de Moore:

Le nombre de transistors des microprocesseurs double tous les deux ans

Core i9-13900K intègre 14,2 milliards de transistors

Intel envisage **1000 milliards** de transistors en 2030.

- La famille **x86**, développée principalement par les entreprises Intel (fabricant du Pentium)
- Le microprocesseur **Zilog Z80**
- la famille **Motorola 68000**
- Les microprocesseurs **PowerPC d'IBM**
- les processeurs **d'architecture MIPS** : Microprocessor without Interlocked Pipeline Stages
- La famille **ARM** est de nos jours utilisée principalement dans les systèmes embarqués, dont de nombreux **PDA** et **smartphones**.

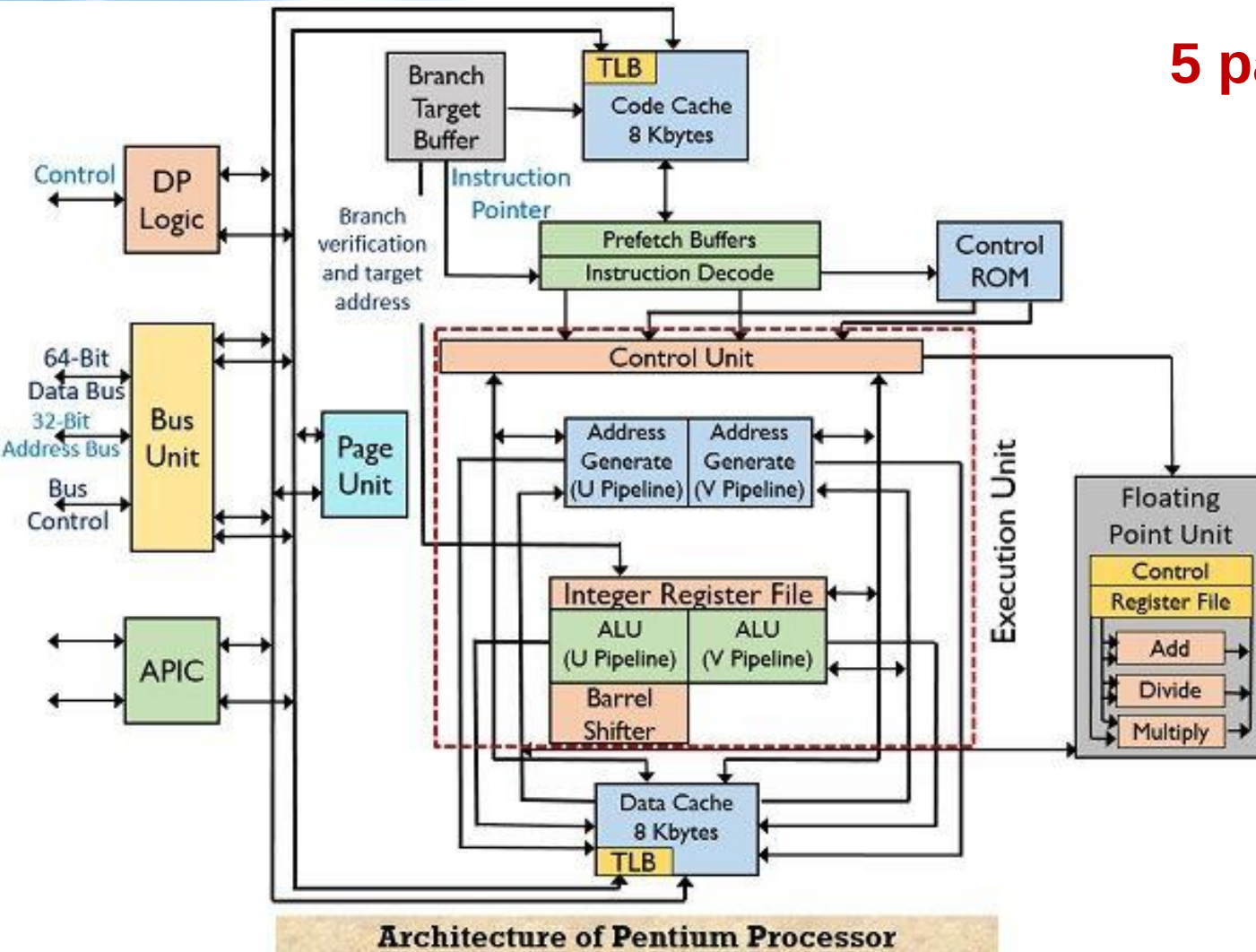


Architecture de base d'un microprocesseur

5 parties essentielles:

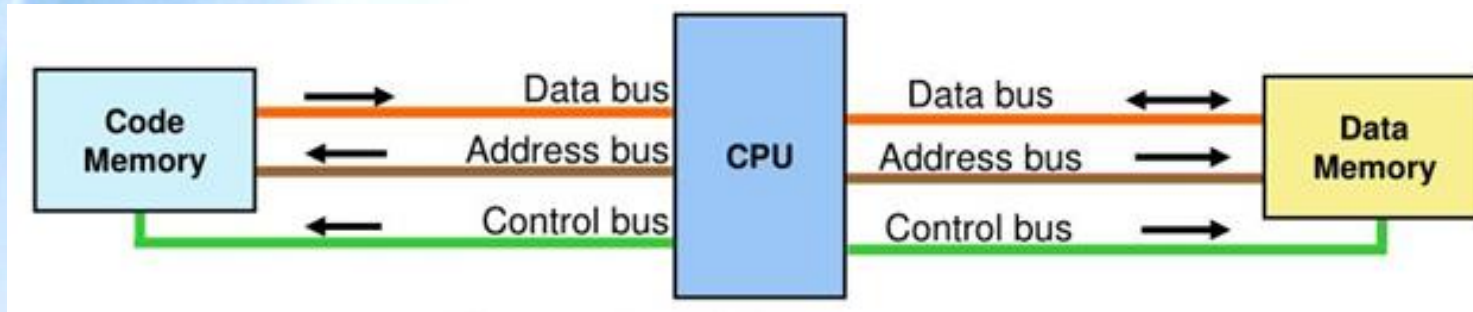
- **Unité d'entrée/sortie**: gère la façon dont les instructions de données entrent et quittent le processeur
- **Unité arithmétique et logique** (ALU) – effectue des comparaisons et des calculs
- **Unité de contrôle**: gère le fonctionnement interne du processeur
- **Unité de mémoire**: contrôle la façon dont les informations sont stockées dans la mémoire interne ou le cache
- **Le coprocesseur** (FPU: Floating point Unit): pour les calculs complexes.

Remarque: Le processeur contient également plusieurs registres internes: exemple registre d'état (status register)



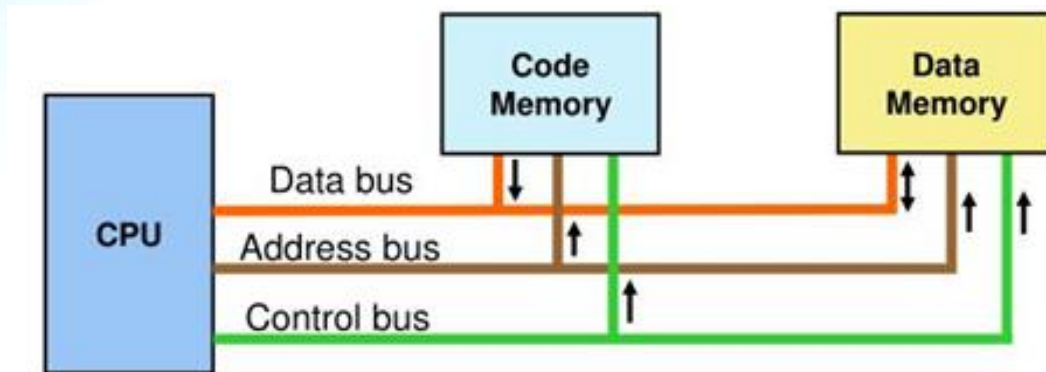
Architecture Von Neumann et Architecture Harvard

Harvard Architecture



L'**architecture de type Harvard** sépare physiquement la mémoire de données et la mémoire programme. L'accès à chacune des deux mémoires s'effectue via deux **bus** distincts.

Von Neumann Architecture



L'**architecture de von Neumann** utilise une structure de stockage unique pour conserver à la fois les **instructions** et les **données** demandées ou produites par le calcul. Utilise un bus commun.

*L'architecture von Neumann est l'architecture la plus couramment utilisée aujourd'hui. De nombreux processeurs modernes, notamment les processeurs **ARM7** et **Pentium**, sont basés sur l'architecture von Neumann.*

● CISC: Complex instruction set computer:

- ▶ Propose de nombreux modes d'adressage, dont certains sont souvent complexes.
- ▶ Le jeu d'instructions comporte souvent de nombreuses instructions complexes qui seront réalisées en plusieurs cycles.

● RISC: Reduced instruction set computer:

- ▶ Propose un jeu d'instructions relativement réduit.
- ▶ Chacune de ces instructions est censée être exécutée en un seul cycle.
- ▶ Les modes d'adressage sont plus simples que dans une architecture CISC

CISC versus RISC

// Example:

- CISC:
 - // MULT [ADDR1], [ADDR2]
- RISC
 - // LOAD A, [ADDR1]
 - // LOAD B, [ADDR2]
 - // MULT A, B
 - // STORE [ADDR1], A

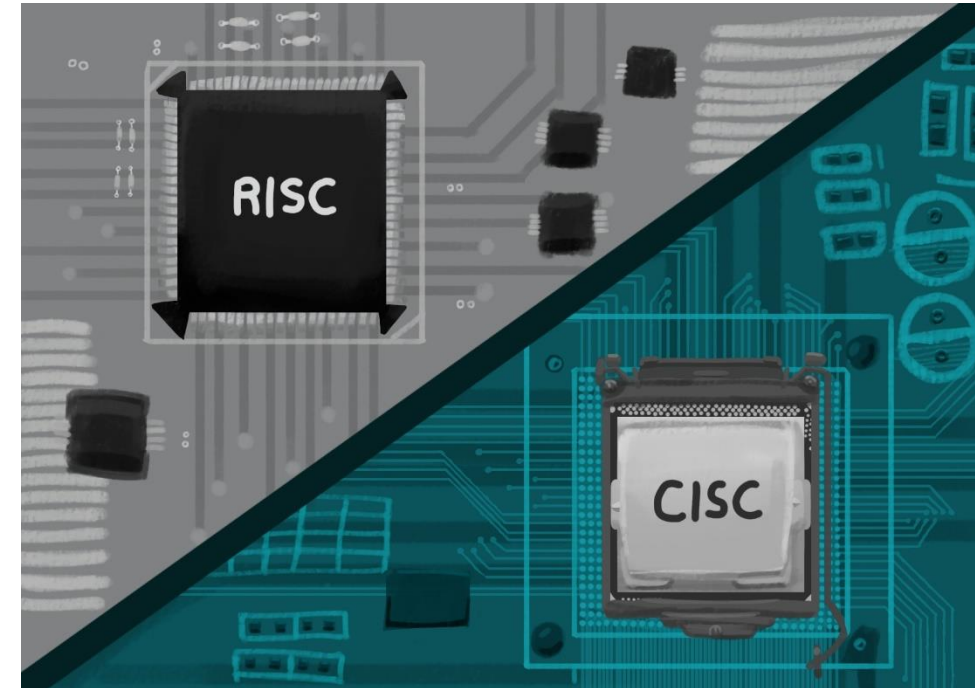
REMARQUE: En termes de complexité matérielle, les architectures CISC sont plus complexes que les architectures RISC: le matériel doit être capable de décoder et d'exécuter un grand nombre d'instructions différentes.

RISC Vs. CISC

Parameter	RISC	CISC
Instruction Types	Simple	Complex
No of Instructions	Reduced(30-40)	Extended(100-200)
Duration of an Instruction	One Cycle	More Cycles (4-120)
Instruction Format	Fixed	Variable
Instruction execution	In Parallel(Pipeline)	Sequential
Addressing Modes	Simple	Complex
Instruction Accessing the memory	Two: Load and Store	Almost all from set
Register set	Multiple	Unique
Complexity	In compiler	In CPU

- L'architecture RISC est plus **économe en énergie**, ce qui la rend adaptée aux appareils à faible consommation, tandis que l'architecture CISC excelle dans le calcul haute performance.
- RISC se concentre sur les **logiciels** pour l'utilisation des performances tandis que CISC se concentre sur le **matériel**
- Les processeurs RISC ont besoin de plus de **RAM** que CISC car ils exécutent une instruction, agissent dessus, puis passent à la suivante.

- **Z80** (8 bits, **CISC**) ;
- **x86** (16, puis 32 bits, **CISC**) ;
- **VAX** (32 bits, **CISC**) ;
- **Alpha** (64 bits, **RISC**) ;
- **SPARC** (32, puis 64 bits, **RISC**) ;
- **MIPS** (32 puis 64 bits, **RISC**) ;
- **68000** (16/32 puis 32 bits, **CISC**) ;
- **AMD64** (version 64 bits de x86 par AMD, aujourd'hui la plus répandue de la famille x86, **CISC**) ;
- **PowerPC** (32, puis 64 bits, **RISC**) ;
- **ARM** (16, 32, puis 64 bits, **RISC**) ;

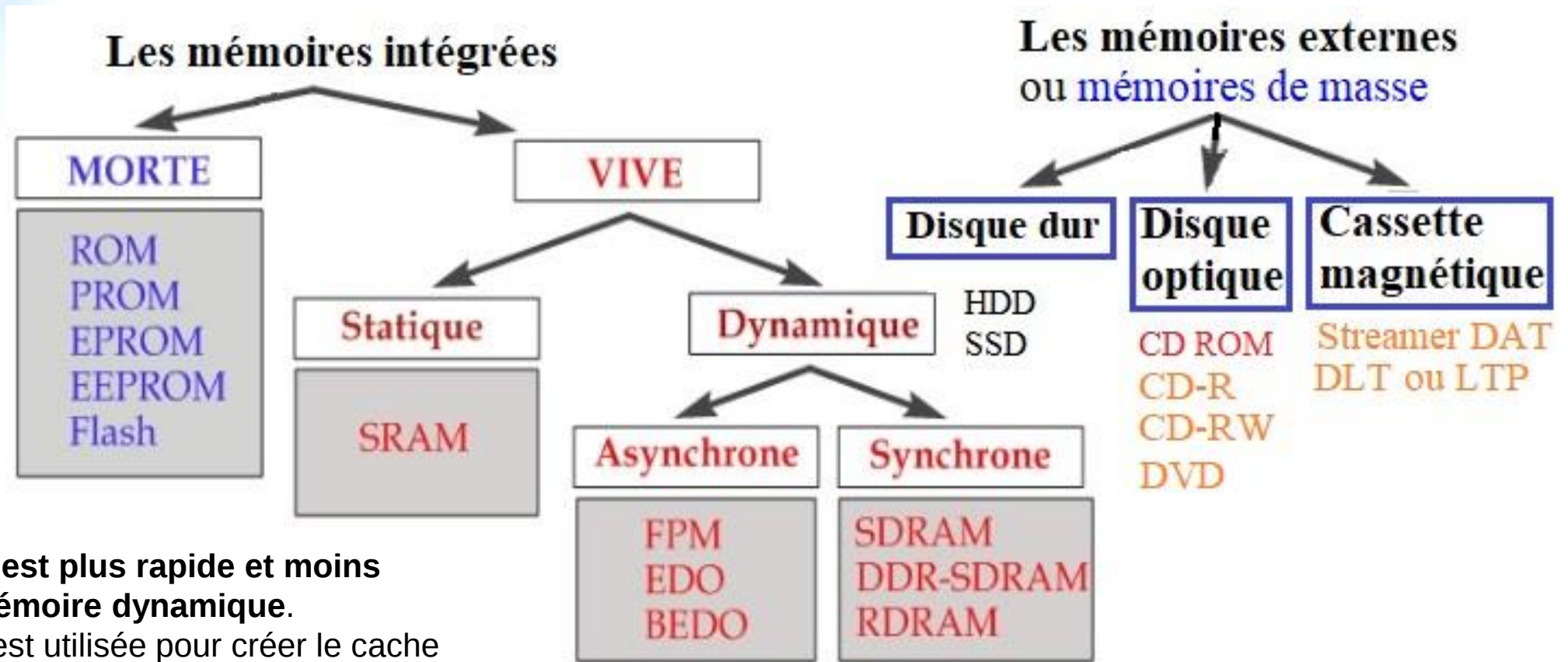


- Le **jeu d'instructions** est l'ensemble des instructions-machine qu'un processeur peut exécuter.
- Ces instructions-machines permettent d'effectuer des opérations élémentaires (**addition, AND, OR, ...**) ou plus complexes (**division**, passage en mode basse consommation...).
- Le jeu d'instructions précise aussi quels sont les registres du processeur manipulables par le programmeur (**les registres architecturaux**).

```
LOAD R0, B ; charge B dans le registre R0
LOAD R1, C ; charge C dans le registre R1
ADD R2, R0, R1 ;  $R2 \leftarrow R0 + R1$ 
STORE R2, A ; stocke R2 à l'adresse A
```

Format d'instruction fixe et variable :

- **Format d'instruction fixe (Cas des RISC):**
 - Dans le contexte des architectures de processeurs, un format d'instruction fixe signifie que chaque instruction machine a une longueur prédéterminée.
 - Chaque instruction occupe le même nombre d'octets en mémoire.
 - Cela simplifie le décodage des instructions, mais cela peut conduire à un gaspillage d'espace si certaines instructions n'ont pas besoin de tous les bits alloués.
- **Format d'instruction variable (Cas des CISC):**
 - À l'inverse, un format d'instruction variable signifie que les instructions peuvent avoir des longueurs différentes.
 - Les **opcodes** (codes opératoires) et les opérandes peuvent être encodés de manière flexible pour économiser de l'espace mémoire.
 - Cela peut être plus efficace en termes d'utilisation de la mémoire, mais le décodage des instructions peut être plus complexe.



la mémoire statique est plus rapide et moins énergivore que la mémoire dynamique.

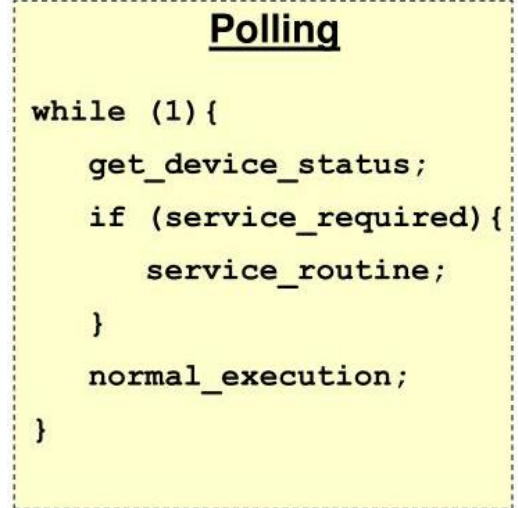
La mémoire statique est utilisée pour créer le cache sensible à la vitesse de la CPU, tandis que la mémoire dynamique forme l'espace RAM le plus vaste.

La mémoire statique est **plus onéreuse** et **moins dense** que la mémoire dynamique,

- Pour dialoguer avec ces périphériques le microprocesseur a trois façons de communiquer avec ces derniers :
 - En questionnant de façon continue le périphérique pour vérifier que des données peuvent être lues ou écrites (**Polling**).
 - En l'interrompant lorsqu'un périphérique est prêt à lire ou écrire des données (**Interruption**)
 - En établissant une communication directe entre deux périphériques (**DMA : Direct memory acces**).

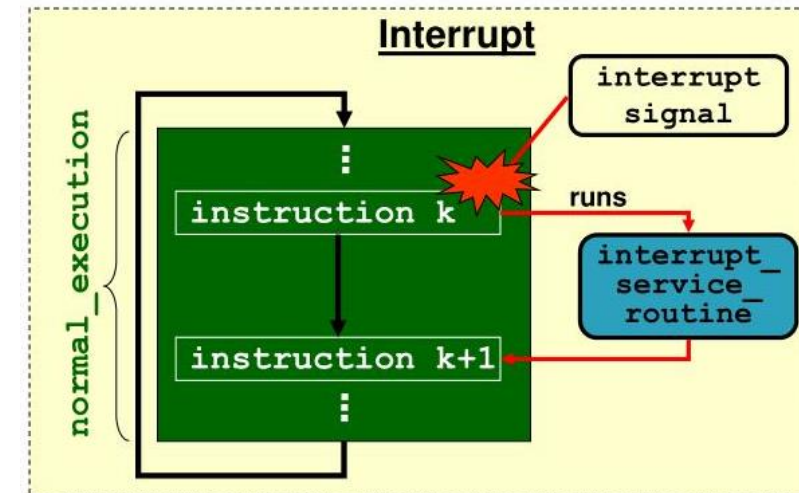
● Polling:

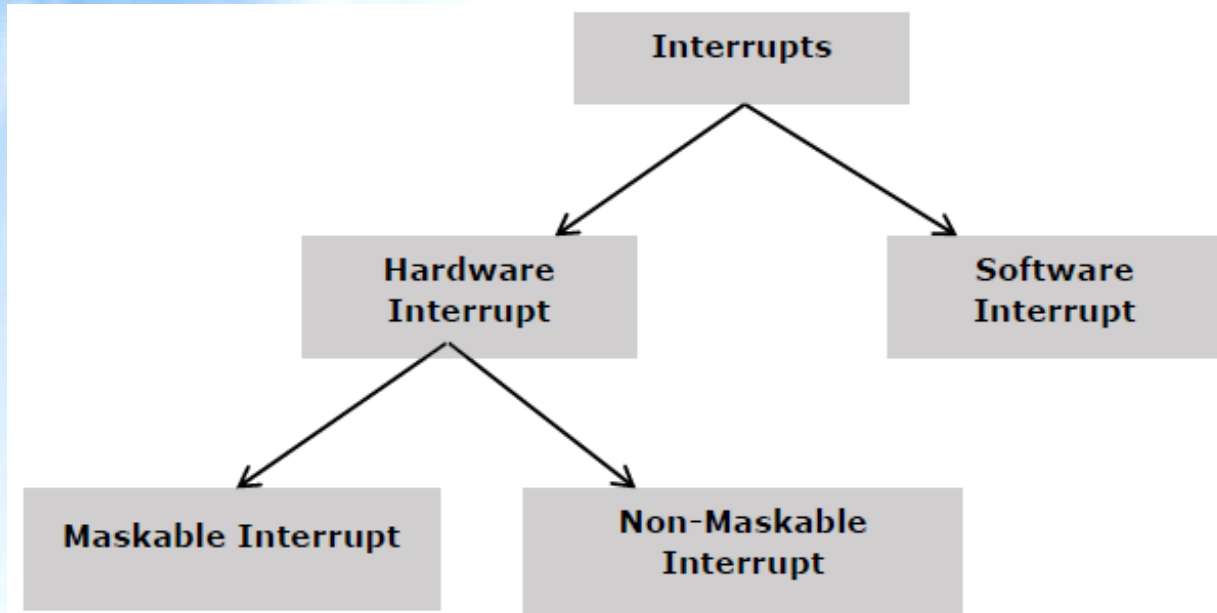
- Le CPU surveille en permanence tous les appareils, à la recherche d'un « indicateur de demande de service »
- Chaque fois qu'il s'agit d'une demande, il sert l'appareil et continue ensuite le polling
- Le CPU est toujours « occupé » et le polling effectue une boucle.



● Interrupts:

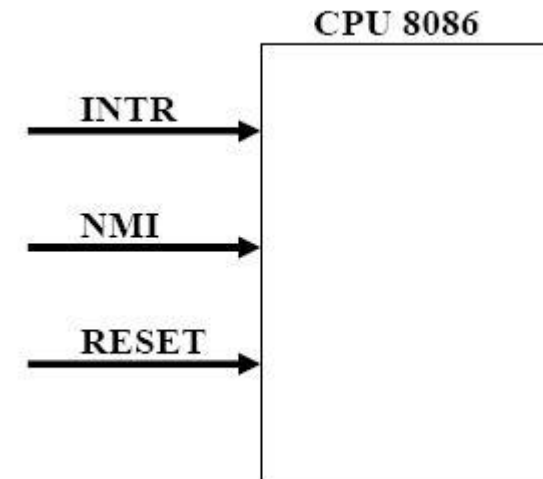
- Lorsqu'un périphérique est prêt et nécessite une attention particulière, il en informe le CPU.
- Le CPU abandonne tout ce qu'il faisait et sert l'appareil, puis revient à sa tâche d'origine
- Le CPU est toujours « libre », lorsqu'il ne sert aucune interruption



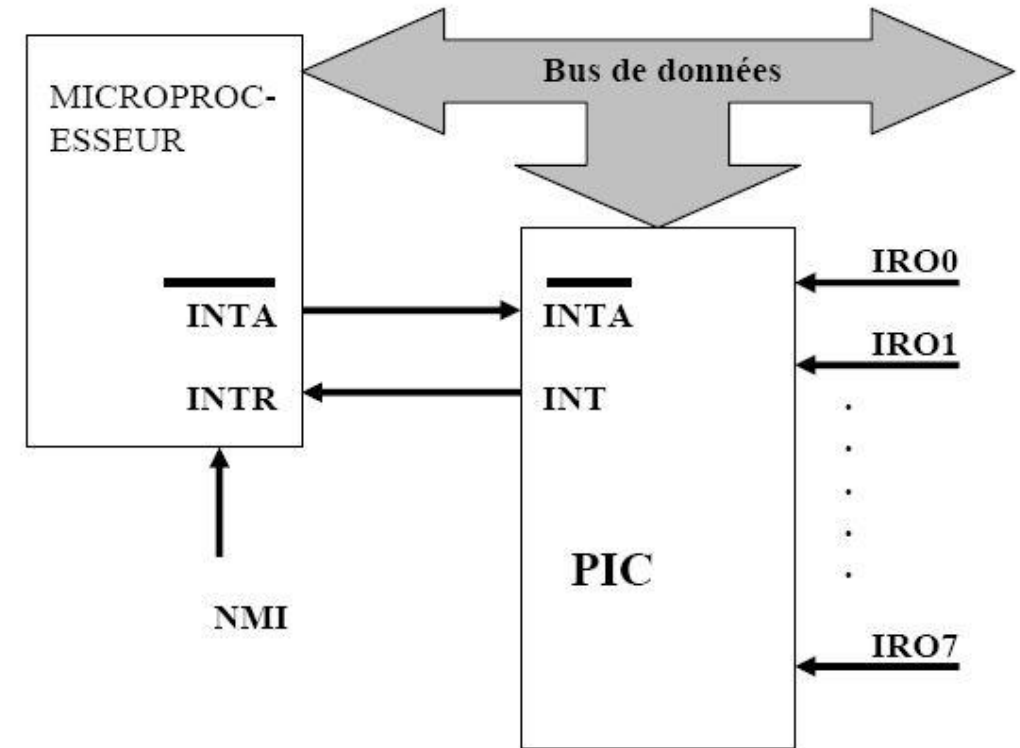


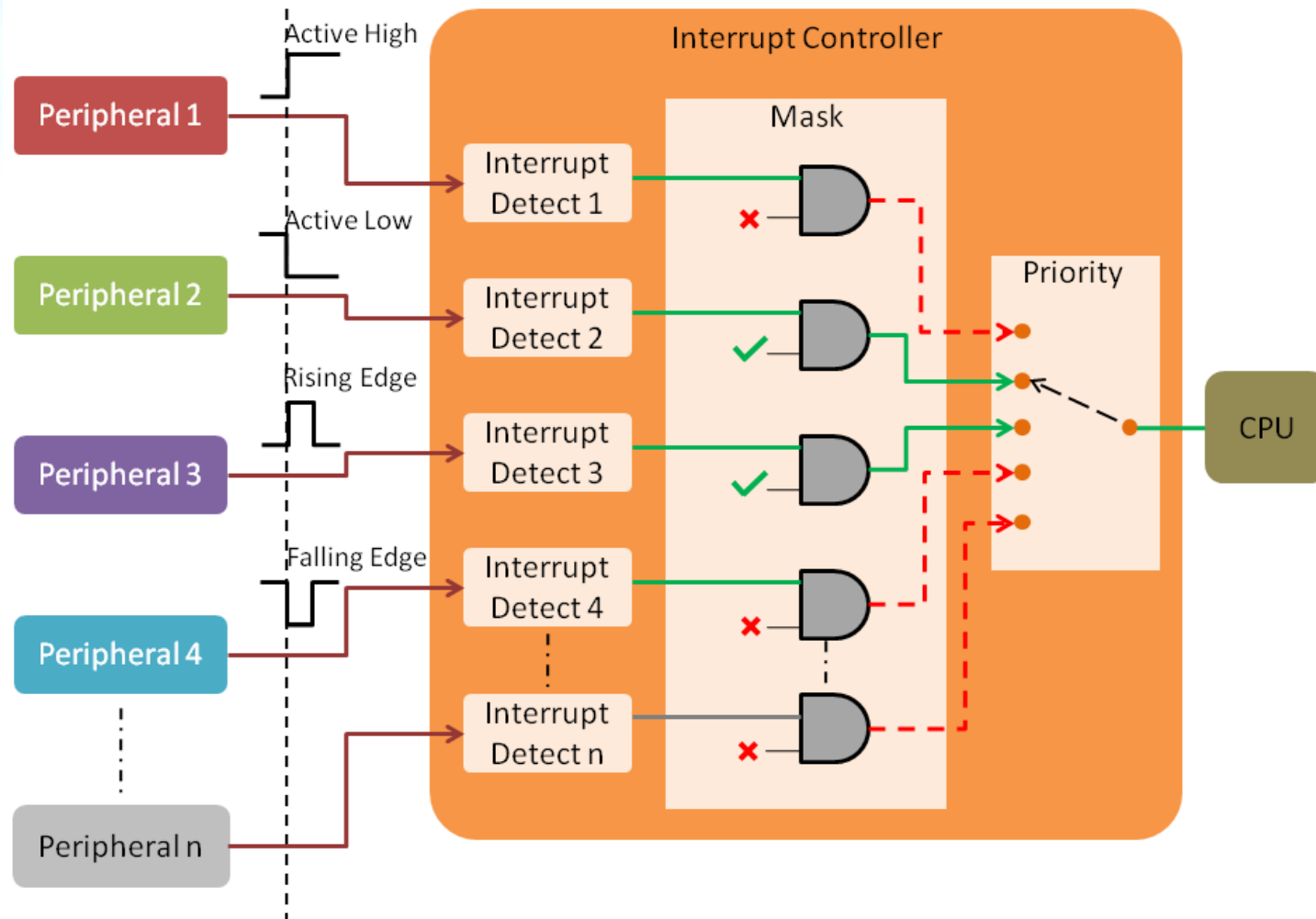
Interruptions logicielles: Ce sont des instructions insérées dans le programme pour générer des interruptions. Il y a 256 interruptions logicielles dans le microprocesseur 8086.

- **Les interruptions matérielles:** sont émises par des périphériques matériels tels que des disques, des cartes réseau, des claviers, des horloges, etc. Chaque périphérique ou ensemble de périphériques aura sa propre ligne **IRQ (Interrupt ReQuest)**.
- Le pilote qui gère l'interruption est exécuté sur le CPU.
- L'interruption matérielle **interrompt directement** le processeur.



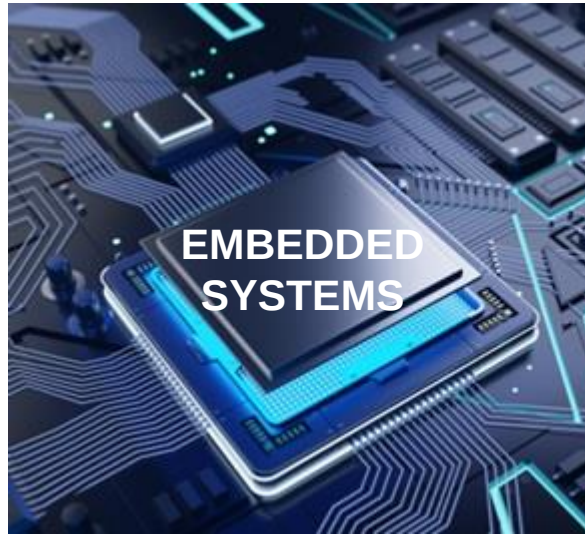
- ❖ **Plusieurs interruptions:** Le fonctionnement des périphériques se fait en général d'une manière asynchrone, donc plusieurs interruptions peuvent être déclenchées en même temps,
- ❖ Le contrôleur d'interruptions est un circuit spécial, extérieur au processeur, dont le rôle est de distribuer et de mettre en attente les demandes d'interruptions provenant des différents périphériques
- ❖ **Ordre de priorité:** La priorité d'interruption définit laquelle d'un ensemble d'interruptions en attente est traitée en premier.





Interrupt Handler, also known as an Interrupt Service Routine or ISR

- Dans la programmation des systèmes informatiques, un gestionnaire d'interruption (**Interrupt Handler**), également connu sous le nom de routine de service d'interruption (**Interrupt Service Routine: ISR**), est un bloc de code spécial associé à une condition d'interruption spécifique.
- Si une condition d'interruption est détectée,
 - l'indicateur d'interruption (**IF: Interrupt Flag**) est activé.
 - La logique d'interruption vérifie ensuite si le bit IE (**Interrupt Enable**) est activé,
 - et si tel est le cas et qu'aucune autre interruption avec une priorité plus élevée n'est en attente, alors l'interruption est déclenchée, c'est-à-dire que l'ISR est appelé.



FIN !
Questions ?