

## *Définition:*

**Un microcontrôleur (MCU) est un composant électronique qui intègre:**

- ✓ **Un microprocesseur (MPU),**
- ✓ **De la mémoire de donnée ,**
- ✓ **De la mémoire programme ,**
- ✓ **Des interfaces parallèles pour la connexion des entrées / sorties,**
- ✓ **Des interfaces séries (synchrone ou asynchrone) pour le dialogue avec d'autres unités,**
- ✓ **Des Timers pour générer ou mesurer des signaux avec une grande précision temporelle,**

## *Définition:*

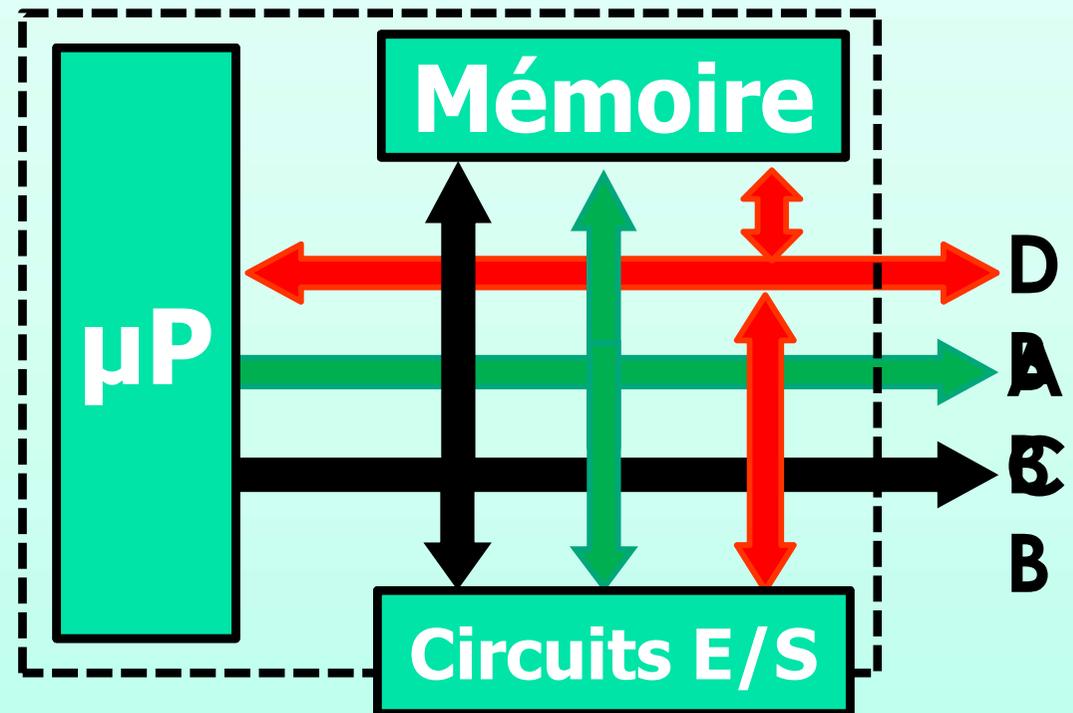
Le microcontrôleur est dimensionné pour l'embarqué. Il est beaucoup plus facile à mettre en œuvre qu'un microprocesseur

### Avantages

- Encombrement réduit
- Circuit imprimé peu complexe
- Faible consommation
- Coût réduit

### Inconvénients

- Système de développement onéreux
- Programmation nécessitant un matériel adapté



## *Description:*

### **C.P.U. (MICROPROCESSEUR):**

**Exécute le programme qui donne vie aux applications. Elle est composée de plusieurs éléments internes :**

- ✓ **Accumulateurs,**
- ✓ **Registres auxiliaires,**
- ✓ **Registres d'index,**
- ✓ **Compteur programme,**
- ✓ **Unité arithmétique et logique (ALU),**
- ✓ **Registre code condition**

**CPU = UAL + unité de contrôle CPU + registres CPU**

## *Description:*

### **MÉMOIRES PROGRAMMES:**

**Ce dispositif contient les instructions du programme que doit exécuter le microprocesseur.**

- ✓ **ROM** : dont le contenu est programmé lors de sa fabrication.
- ✓ **PROM** : programmable électriquement une seule fois par le développeur ;
- ✓ **EPROM** : programmable électriquement et effaçable aux U-V,
- ✓ **EEPROM** programmable et effaçable électriquement.

## *Description:*

### **MÉMOIRES DONNÉES:**

**Ce dispositif permet de mémoriser temporairement les données générées par le microprocesseur pendant les différentes phases du traitement numérique (résultats d'opérations, états des capteurs...).**

**Ces mémoires sont accessibles en écriture et en lecture.**

*Les applications font plus souvent appel à des manipulations d'entrées/sorties.*

## *Description:*

### **INTERFACE PARALLÈLE.**

Ce type d'interface, répartie sur plusieurs ports (maximum 8 bits), permet de prendre en compte des états logiques appliqués en entrée (état de capteurs) ou de générer des signaux binaires en sortie (commande d'actionneurs).

Les broches de ces ports peuvent donc être configurées en entrée ou en sortie.

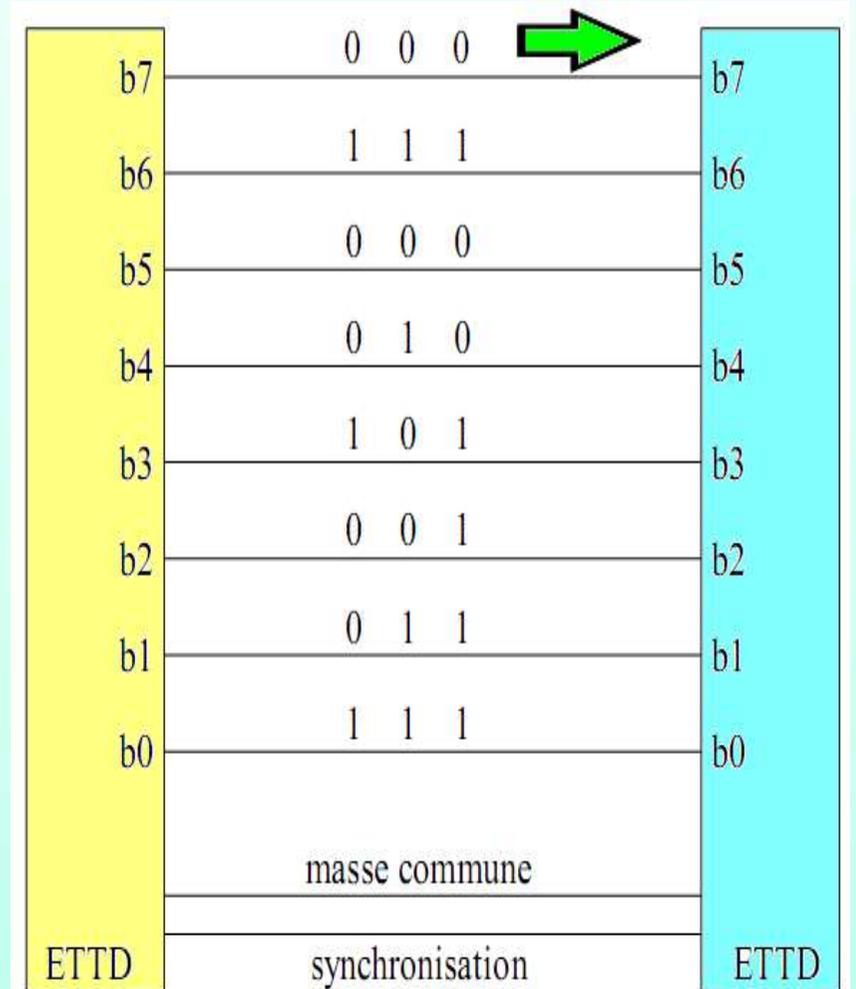
On trouve généralement :

- Un registre de direction pour une configuration en entrée ou en sortie,
- Un registre de donnée recopiant les états logiques de chaque broche de port,
- Un registre d'option permettant plusieurs configurations en entrée ou en sortie.

## Description:

### INTERFACE PARALLÈLE.

- Caractérisée par un transfert simultané de tous les bits d'un même mot.
- Nécessite autant de conducteurs qu'il y a de bits à transmettre et un conducteur commun (liaison asymétrique) ou autant de paires que de fils si la masse n'est pas commune (liaison symétrique).
- Un conducteur supplémentaire peut être utilisé pour transmettre un signal qui assurera la synchronisation entre les intervalles d'émission et ceux de réception
- Autorise une grande vitesse de transmission (débit).
- Un coût élevé (nombre de conducteurs)
- Une distance franchissable limitée réservent la transmission parallèle aux liaisons de processeur à processeur

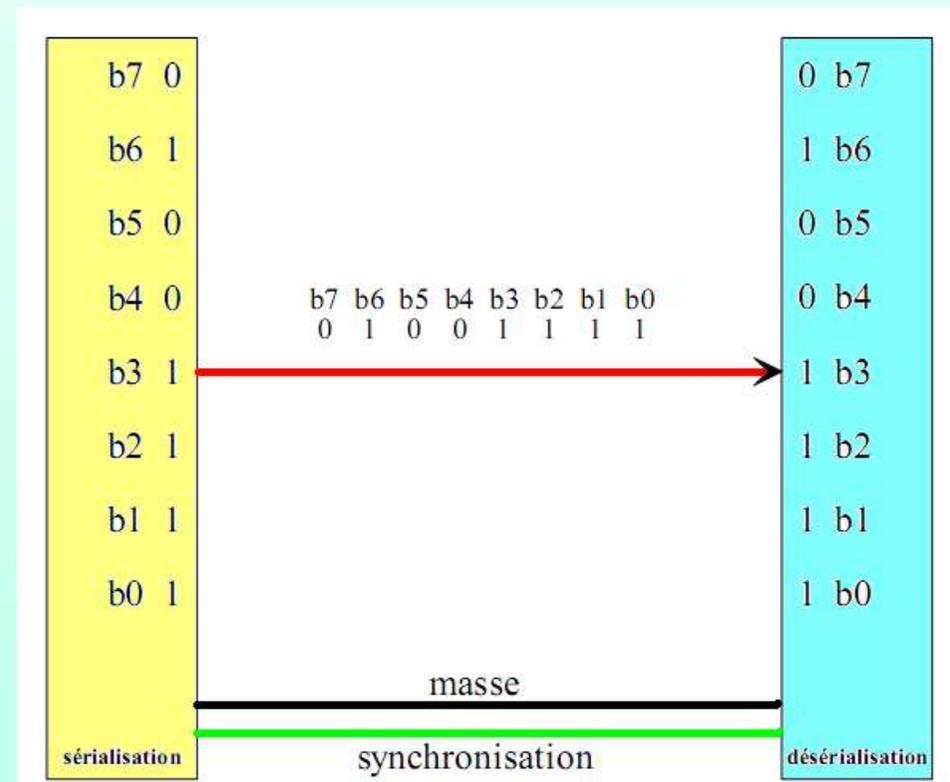


## Description:

### INTERFACE SÉRIE.

Les données envoyées ou reçues se présentent sous la forme d'une succession temporelle (sur un seul bit) de valeurs binaires images d'un mot.

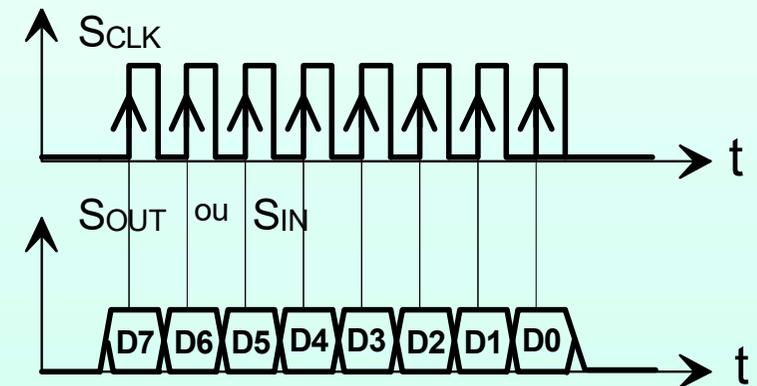
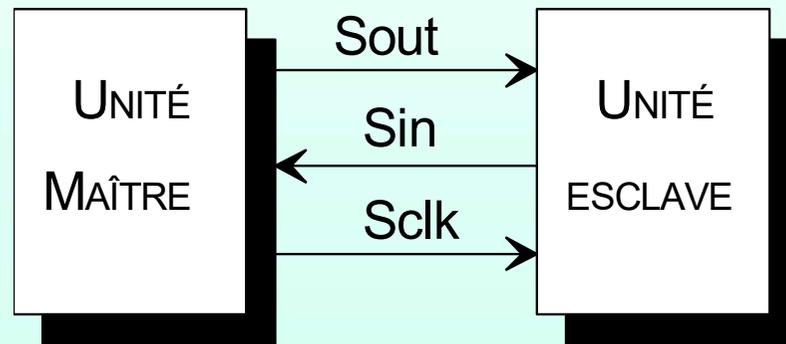
- Nécessite une interface de conversion pour sérialiser les bits à l'émission (conversion parallèle/série) et les désérialiser à la réception (conversion série/parallèle).
- Ne nécessite, pour la transmission des données, que deux conducteurs, d'un coût moins important, elle est utilisée pour les transmissions sur des distances importantes.
- Les vitesses de transmission et de réception doivent être identiques (bits/s ou bauds).
- *Modes simplex, half-duplex et full-duplex*



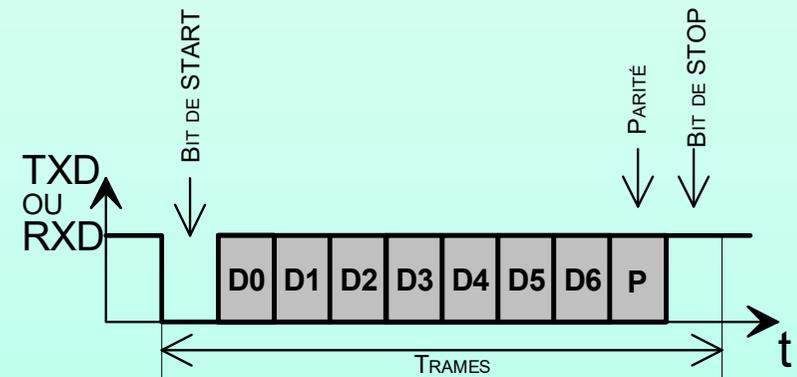
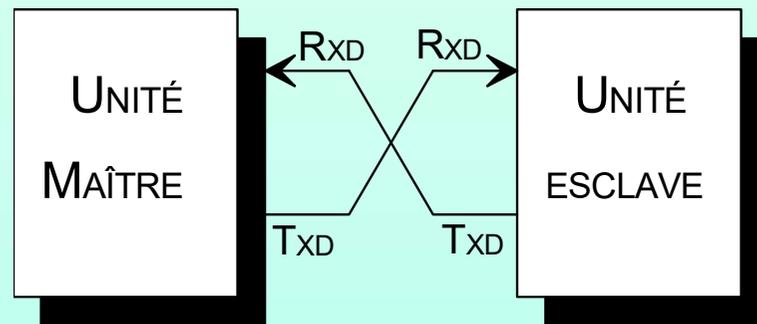
## Description:

### INTERFACE SÉRIE.

#### ○ Synchron



#### ○ Asynchrone



## *Description:*

### **LE CAN:**

Le CAN intégré dans les microcontrôleurs est généralement du type “*Approximations successives*”. Il possède plusieurs entrées multiplexées accessibles via les broches des ports de l’interface parallèle.

Le CAN possède normalement 2 registres :

- ✓ Un registre de données contenant le résultat de la conversion,
- ✓ Un registre de contrôle permettant de lancer et de surveiller la conversion.

## *Description:*

### **LE TIMER:**

**Le Timer permet de réaliser les fonctions suivantes :**

- **Génération d'un signal périodique modulé ou non en largeur d'impulsion,**
- **Génération d'une impulsion calibrée,**
- **Temporisation,**
- **Comptage d'événements.**

**Plusieurs registres associés au Timer permettent de configurer les différents modes décrits précédemment.**

## *Description:*

### **LE CHIEN DE GARDE:**

**Ce dispositif est un système anti-plantage du microcontrôleur. Il s'assure qu'il n'y ait pas d'exécution prolongé d'une même suite d'instructions. Un compteur pré-chargeable se décrémente régulièrement au rythme de la fréquence d'horloge. Si aucun pré-chargement n'est effectué avant qu'il n'atteigne la valeur "0" un Reset est généré relançant ainsi le microcontrôleur.**

*Il faut donc penser à pré-charger régulièrement ce chien de garde par programme lorsqu'il est activé.*

## *Familles génériques:*

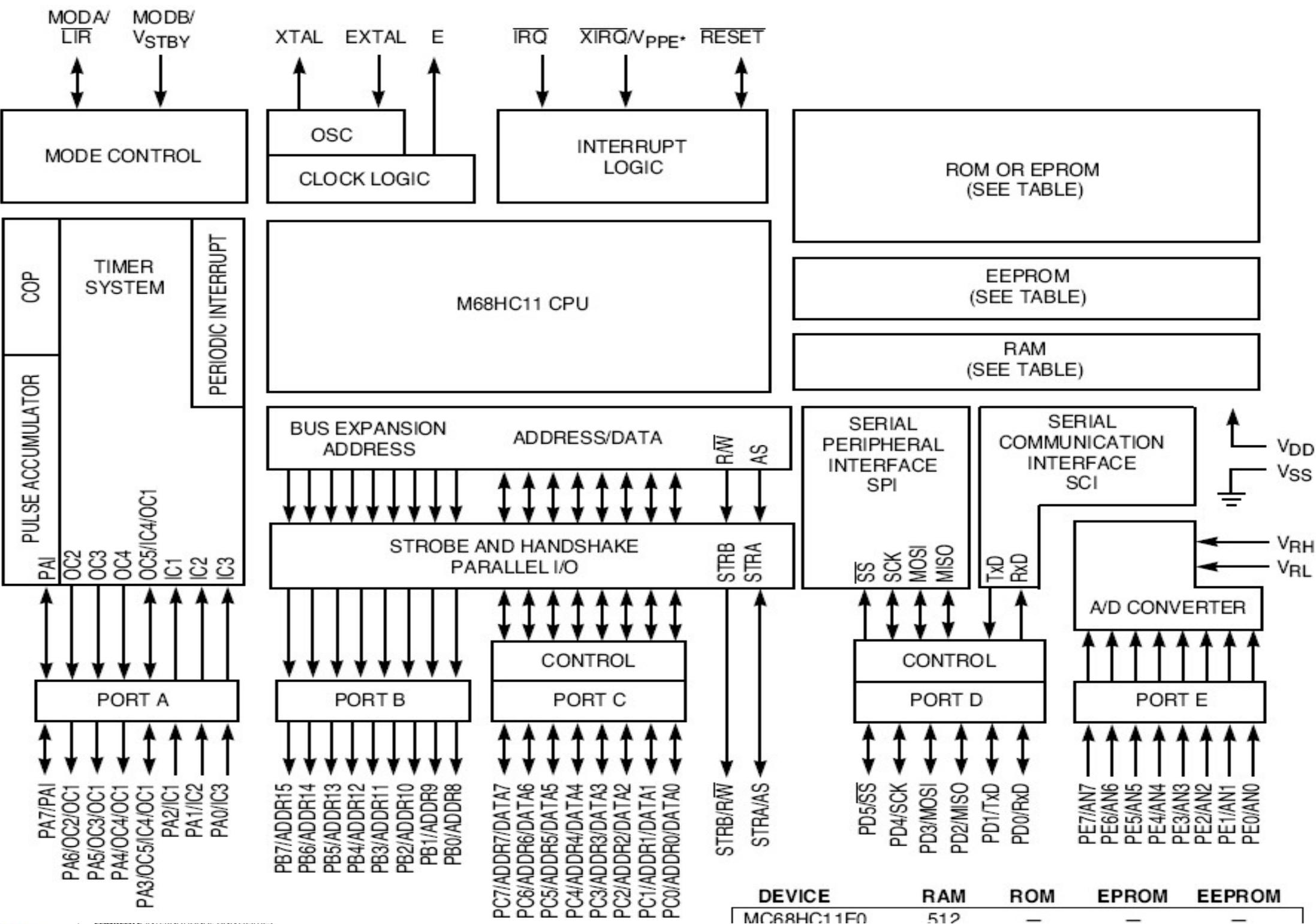
RÉFÉRENCE	FABRICANT	VITESSE	RAM	ROM / EPROM / FLASH	EEPROM	E / S LOGIQUES	TIMER	ENTRÉES ANALOGIQUES
8051	Intel	12 Mhz	128 o	4 Ko	X	32	2	0
16C71	Microchip	20 Mhz	36 o	1Kx14	X	13	1	4
6805 S2	Motorola	4 MHz	64	1 Ko	X	16	2	8
68HC11 A1	Motorola	8 MHz	256 o	X	512	22	1	8
AT90S 8515	Atmel	20 MHz	512 o	4 Ko	512	32	3	8
ST 6265	Thomson	8 MHz	128 o	4 Ko	64 o	21	2	13

## *Présentation:*

**La famille 68HC11 est la plus puissante des familles de microcontrôleurs 8 bits de Motorola.**

**Selon les versions:**

- ✓ **RAM : de 256 octets à 1Ko,**
  - ✓ **ROM : de 8 à 24 Ko (ou EPROM)**
  - ✓ **EEPROM : de 512 à 2048 octets.**
- 
- **Sa particularité est de posséder une ALU et des registres internes sur 16 bits mais le bus de données reste sur 8 bits.**
  - **Il adresse jusqu'à 64 Ko de mémoire.**
  - **Son jeu d'instructions est dérivé de ces ancêtres (68xx)**

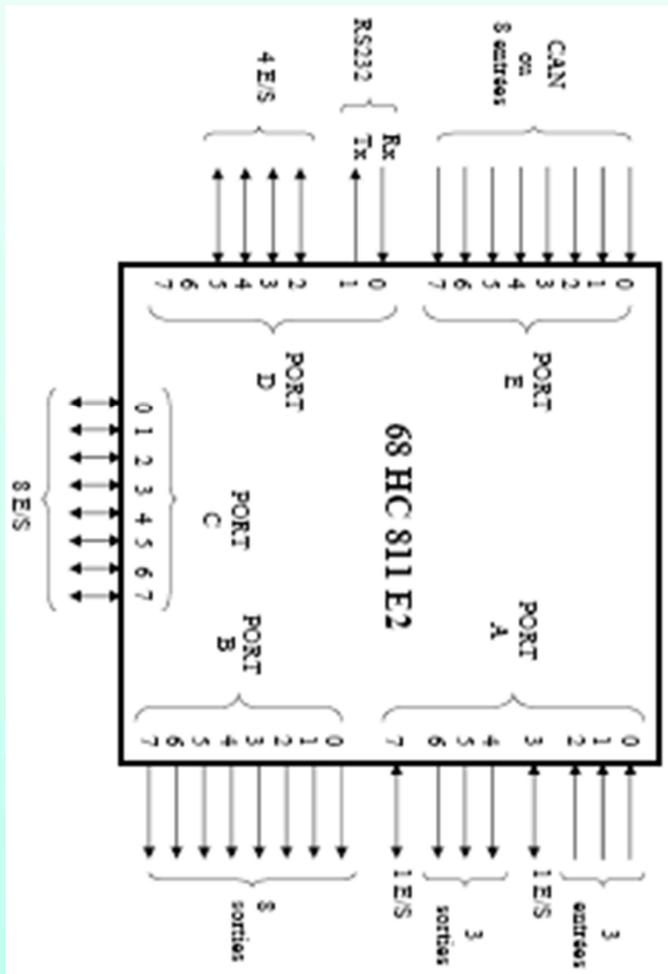


DEVICE	RAM	ROM	EPROM	EEPROM
MC68HC11F0	512	-	-	-

# Le microcontrôleur 68HC11:

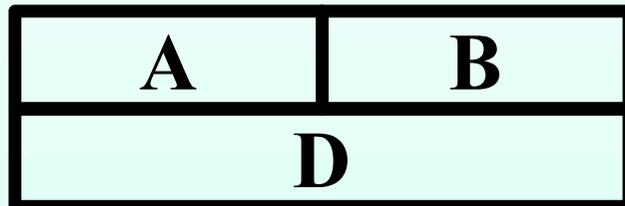
(3)

## Principales caractéristiques:



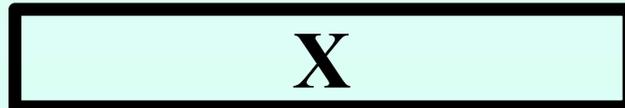
DEVICE	RAM	ROM	EPROM	EEPROM
MC68HC11E0	512	-	-	-
MC68HC11E1	512	-	-	512
MC68HC11E8	512	12K	-	-
MC68HC11E9	512	12K	-	512
MC68HC11E20	768	20K	-	512
MC68HC711E20	768	-	20K	512
MC68HC811E20	256	-	-	2048

## *Les registres:*



Accumulateurs A et B : 8 bits

Accumulateurs D (A:B) : 16 bits



Registre d'index X : 16 bits



Registre d'index Y : 16 bits



Pointeur de pile S: 16 bits



Compteur programme : 16 bits



Registre code condition: 8 bits

## *Les accumulateurs:*



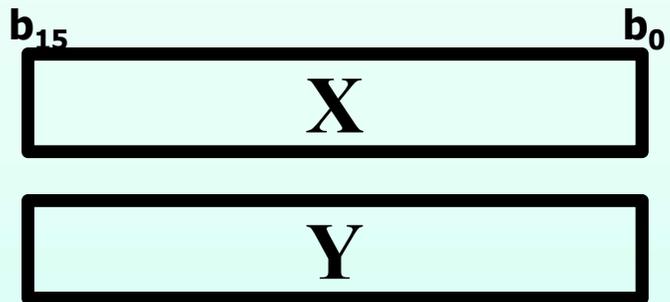
Il existe deux registres appelés accumulateurs correspondant à un octet.

Ils sont notés A et B et servent à faire toutes sortes de manipulations sur des données de 8 bits.

On peut y stocker temporairement une valeur, faire une opération sur cette valeur à l'intérieur de l'accumulateur ou encore utiliser cet Acc pour ranger la donnée en mémoire.

La concaténation des accumulateurs A et B constituent ce qui est appelé le registre D

## *Les registres Index:*



### *Le registre X :*

A l'instar de A et B, c'est un registre à tout faire mais il peut manipuler une donnée de deux octets. Par ailleurs, il est très utile en tant qu'index lorsque le  $\mu$ P exécute une instruction utilisant un adressage indexé.

### *Le registre Y :*

Il joue le même rôle que X si ce n'est que toute instruction utilisant Y plutôt que X se code sur un octet supplémentaire donc son exécution est alors plus lente.

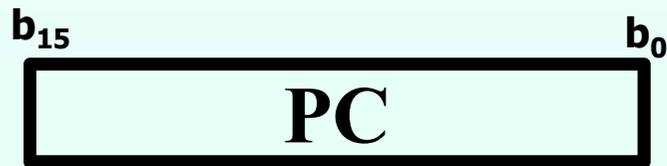
## *Les registres SP:*



### *Le registre SP (Stack Pointer) :*

- C'est le pointeur de pile.
- Il indique en permanence la prochaine adresse libre de la pile. La pile est une zone de mémoire que l'on peut utiliser pour ranger momentanément des données avant de s'en servir à nouveau selon un principe LIFO.
- On peut donc empiler puis dépiler des données ce qui signifie que la pile est de taille variable.
- Le registre SP indique en quelque sorte la hauteur de la pile.
- Il existe des instructions pour empiler puis dépiler des données.

## *Le registre PC:*



### *Le registre PC (Program Counter) :*

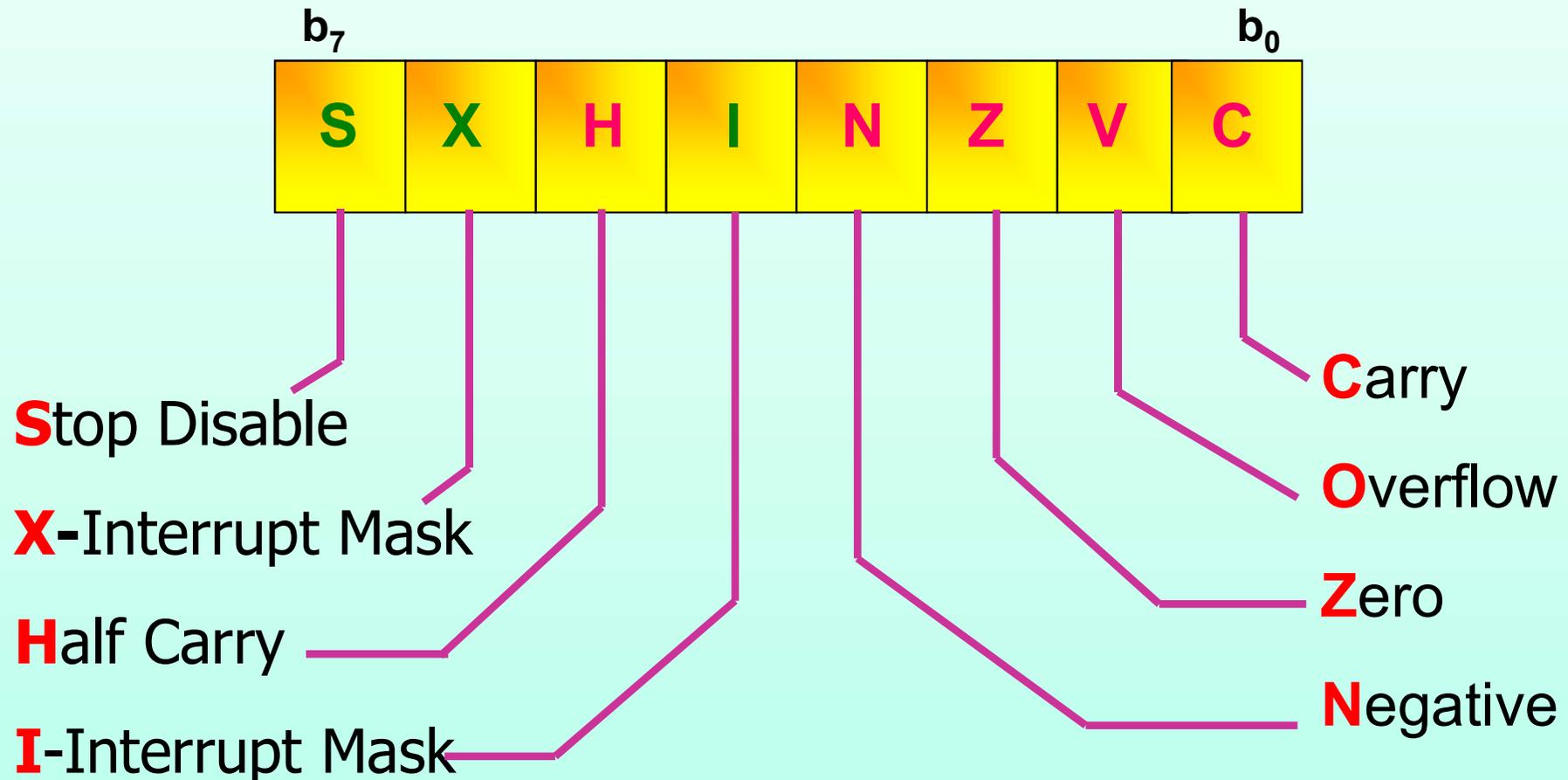
**Ce registre appelé compteur ordinal contient l'adresse de la prochaine instruction du programme qui sera exécutée.**

**Il permet donc au  $\mu$ P de gérer la séquence des instructions du programme, en particulier lors des boucles.**

**Il est incrémenté automatiquement par la taille de l'instruction.**

## Le registre CCR:

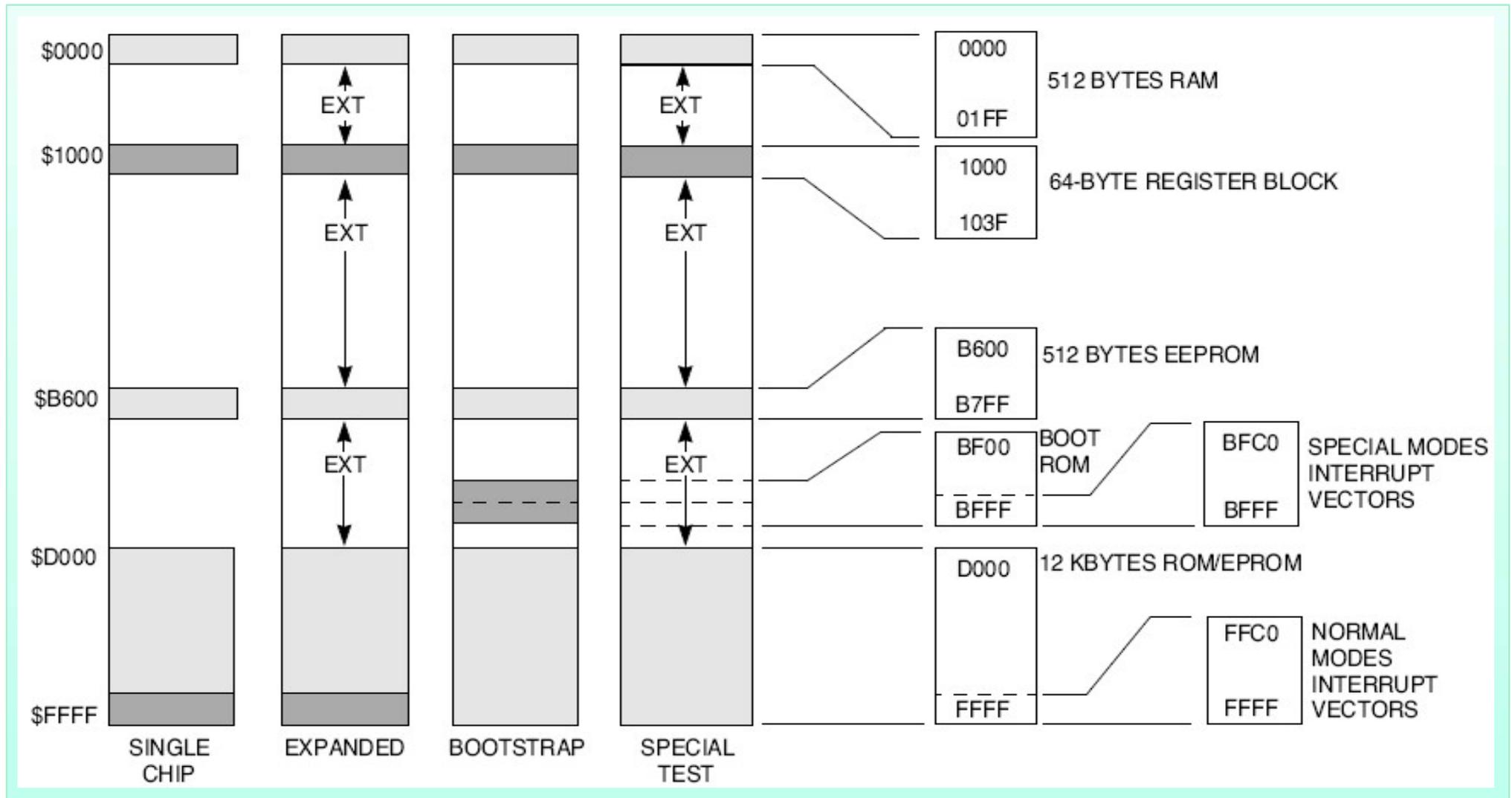
### Code Condition Register :



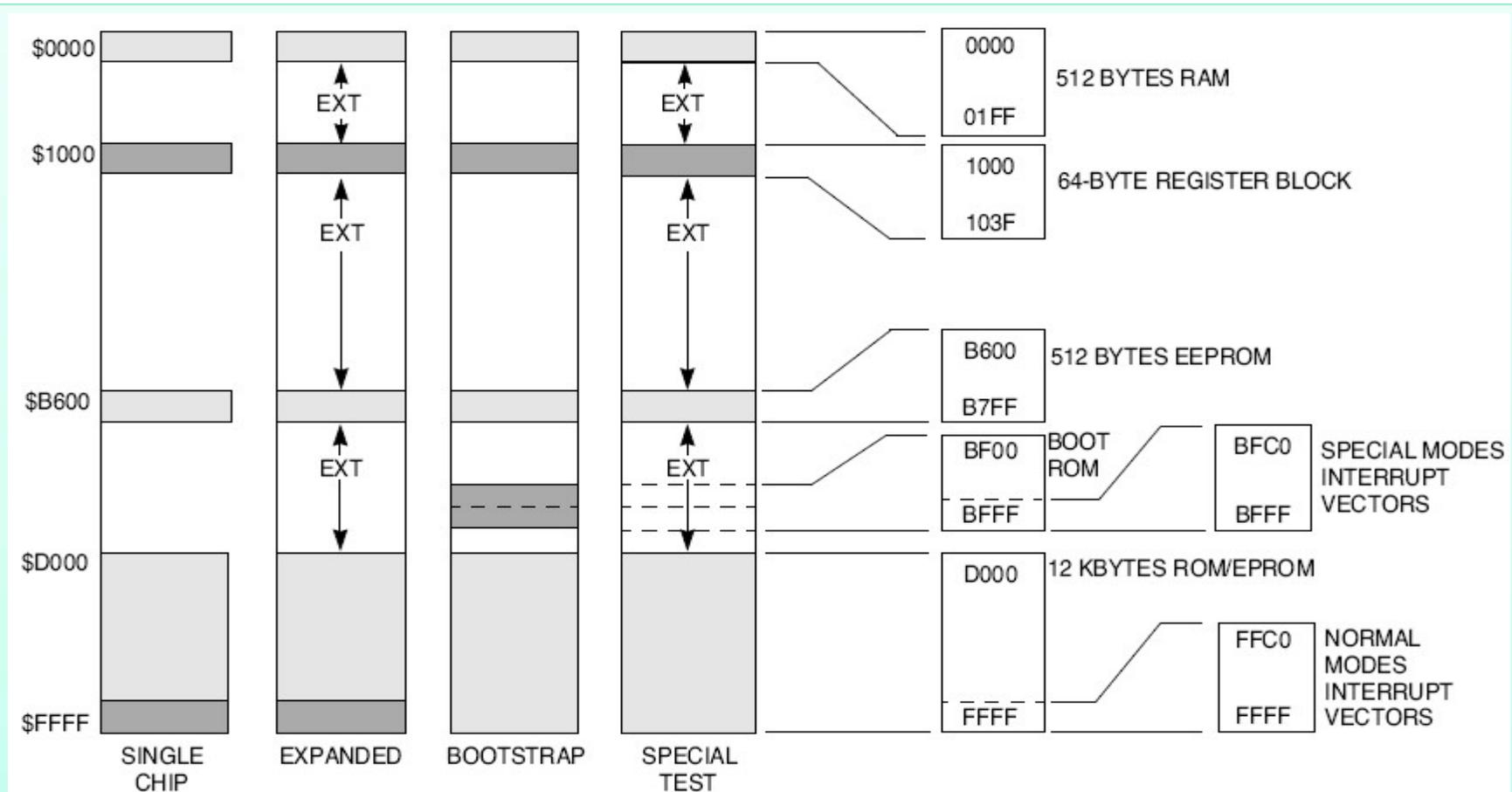
# Le microcontrôleur 68HC11:

[ 10 ]

## *Le plan mémoire:*

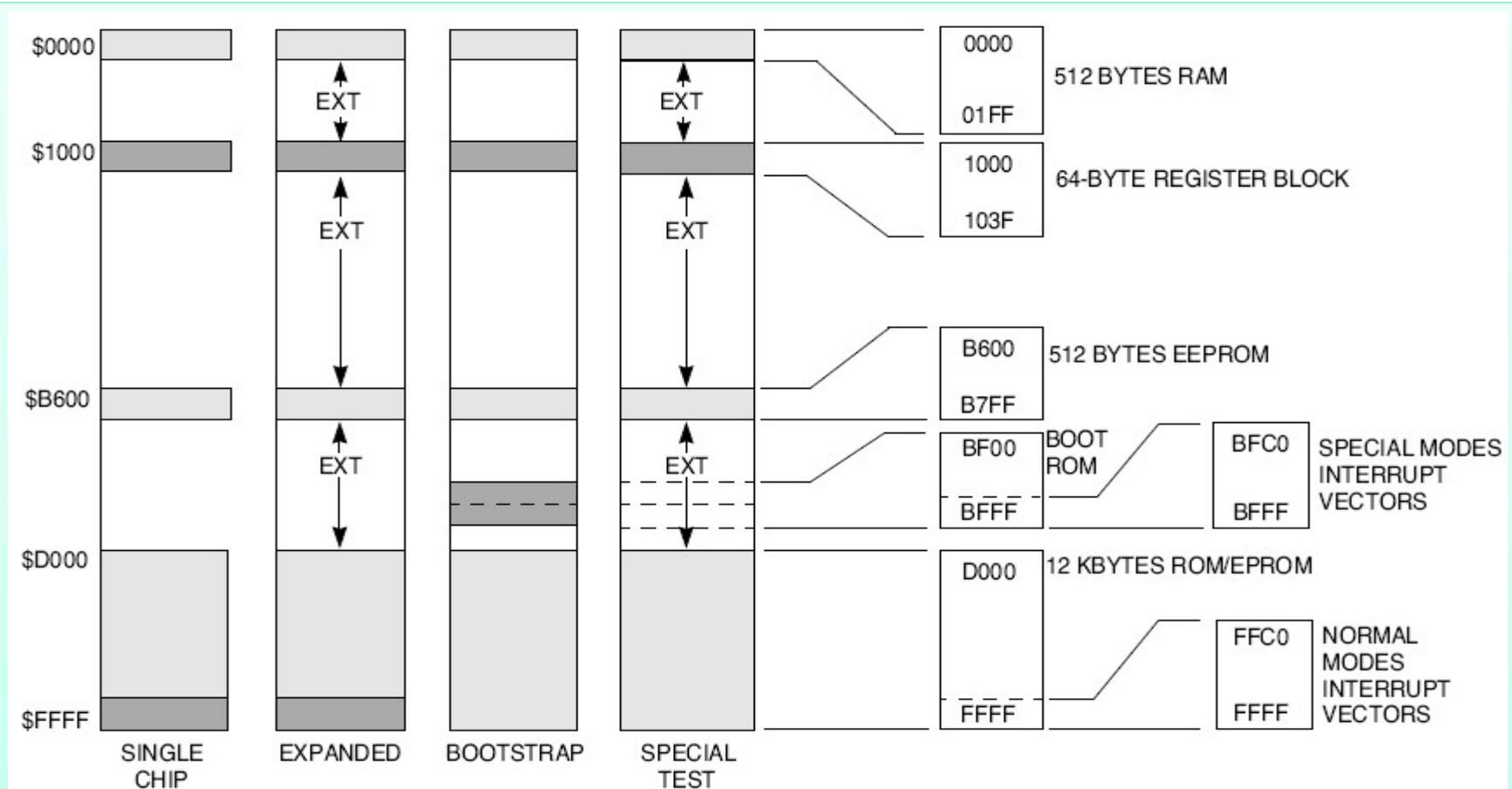


## Le plan mémoire:



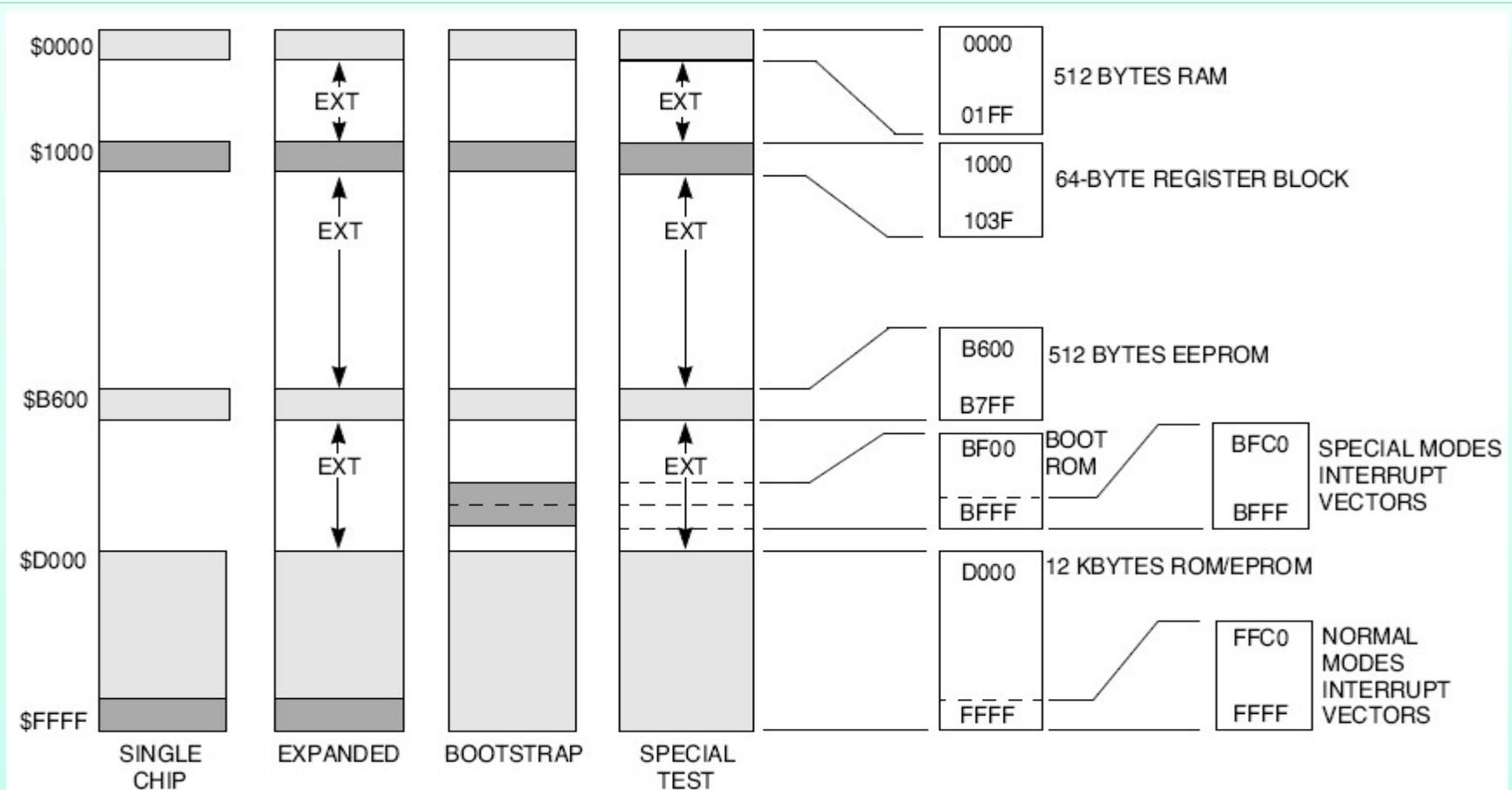
**Single Chip:** fonctionnement autonome, tous les ports sont disponibles par contre la mémoire est limitée à la mémoire interne.

## Le plan mémoire:



**Expanded** : ce mode permet d'étendre la capacité mémoire ainsi que les périphériques par l'adjonction de boîtiers externes

## Le plan mémoire:



**Special test** : destiné au départ aux tests de production de Motorola. Ce mode peut être utilisé pour le développement en particulier pour l'émulation, il permet entre autre de modifier le registre CONFIG après le RESET