

# Introduction au processeurs

## I- Préambule

**Microprocesseur et microcontrôleur** : éléments incontournables en instrumentation.

Progressivement complétés par d'autres composants numériques.

Fausse Idée : marché de l'informatique → principal débouché du microprocesseur.

L'informatique (monde PC) : dominé par 3 fabricants (**Intel**, ...).

Marché du microprocesseur : plusieurs centaines de fabricants → tout autre ordre de grandeur.

En 1998 : 99% des microprocesseurs → destinés au marché des systèmes embarqués (c.à.d de l'instrumentation et du contrôle).

## II- Les processeurs

Développement de l'électronique numérique

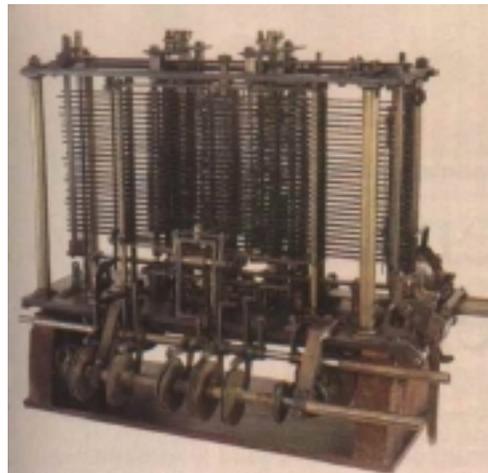
→ Apparition de plusieurs types de composants très puissants en instrumentation :

- **Microprocesseur** : généraliste (peut tout faire, mais n'est optimisé pour rien)
  - Employé préférentiellement dans les systèmes informatiques.
  - Traite des données de 8, 16, 32, 64 et même 128 bits.
  - En instrumentation on lui préférera des composants plus spécialisés (plus optimisés et plus performants).
- **Microcontrôleur** → adapté aux applications embarquées
  - Comporte sur sa puce un certain nombre d'interfaces n'existant pas sur un microprocesseur.
  - Par contre : généralement moins puissant en terme de rapidité ou de taille de mémoire.
  - Plus souvent cantonné aux données de 8 ou 16 bits.
- **Processeur de signal** : beaucoup plus spécialisé.
  - Optimisé pour le traitement du signal.
  - Non prévu pour accomplir des tâches généralistes de bas niveau.

- Composant **ASIC ou FPGA** : Exploités complémentaires.
  - Un ASIC : très spécifique d'une application → spécialement développé pour celle-ci.
  - Le FPGA : destiné à faire n'importe quoi ← l'utilisateur qui devra le configurer
  - Le FPGA sert à la mise au point, et un fondeur de silicium transcrit ses fonctionnalités en un ASIC figé

### III- Historique

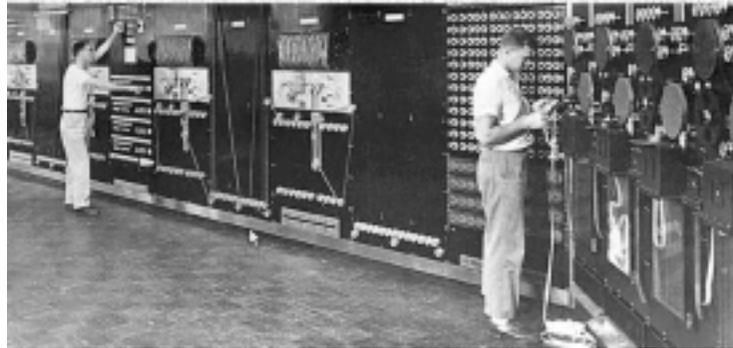
- **1632** Oughtred (GB) invente la **règle à calcul**
- **1642** Pascal (F) imagine sa **machine à calculer**
- **1666** Moreland (GB) identifie la **multiplication** comme une **suite d'additions**
- **1770** Hahn (D) crée une **machine à cylindre**
- **1820** Babbage (GB) initie ses **premiers travaux** sur **l'automatisation des calculs**
- **1833** Babbage invente le **concept de calculateur programmé** (c.à.d enchaînement automatique des opérations)



*La machine analytique de Charles Babbage*

- **1847-1854** Boole (GB) imagine **l'algèbre de Boole**
- **1880** Hollerith (USA) invente la **carte perforée** à l'occasion du premier recensement
- **1885** création d'une **petite compagnie** qui deviendra **IBM**

- **1897** Braun (USA) invente l'**oscilloscope**
- **1906** Lee de Forest (USA) invente la **triode**
- **1920** Morrison (USA) imagine l'**horloge à quartz**
- **1938** Couffignal (F) le père de la **numération binaire**
- **1944** **Création de l'ENIAC** : première machine électronique (monstre d'environ 18000 tubes électroniques!)



*L'ENIAC*

- **1948** Bardeen, Brattain, Shockley (USA) découvrent l'**effet transistor** aux Bell Labs
- **1949** première **machine IBM à cartes perforées**



*Pupitre de contrôle d'un ordinateur IBM705*

- **1960 IBM 7070**
- **1962** Dreyfus (F) invente le **mot informatique** (techniques de traitement automatique de l'information)
- **1968** Dreyfus (F) première exploitation d'un **système multiprocesseurs** (Centre de Calcul Scientifique de l'Armement à Arcueil, machine UNIVAC 7108)
- **1972** la compagnie **INTEL** (USA) **invente le microprocesseur** 2 ans après le premier circuit intégré
- **1986** la même lance le **80386**
- **1990** les premiers circuits **périphériques configurables** (selon le concept **FPGA**)
- **2000** la **barrière du GHz** est franchie pour des circuits à base de silicium (AMD et Intel).

## IV- Informatique industrielle

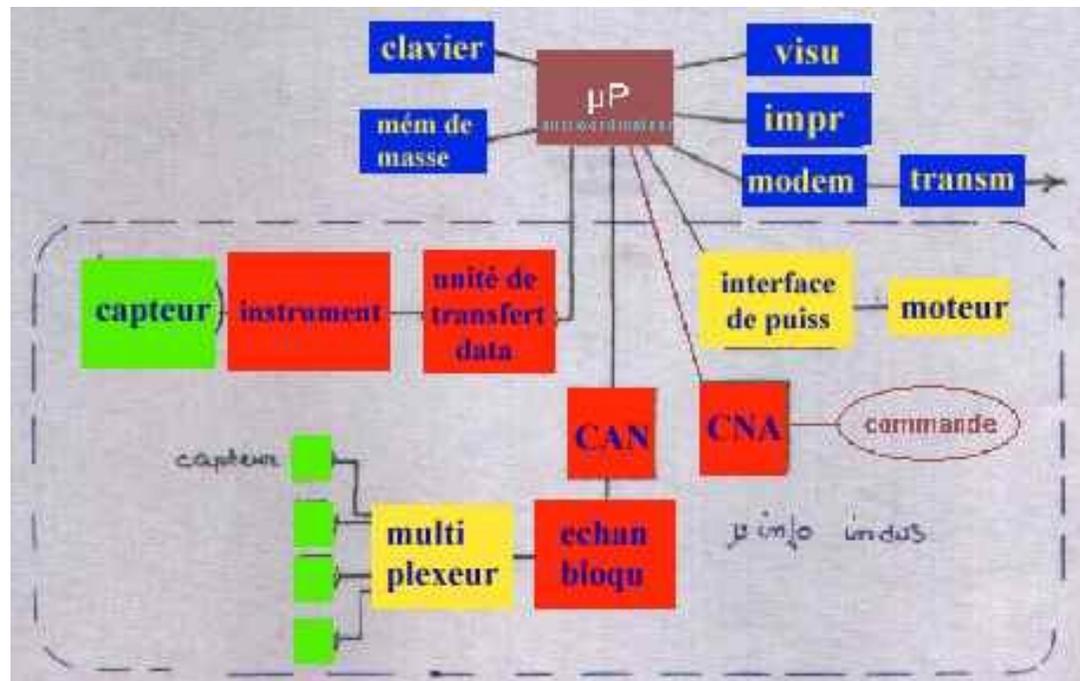
Pas de différence de principe entre ordinateur et un micro-système utilisé en informatique industrielle.

L'emploi qu'on en fait est différent :

Un CRAY est employé à du calcul scientifique ou à de la gestion bancaire,

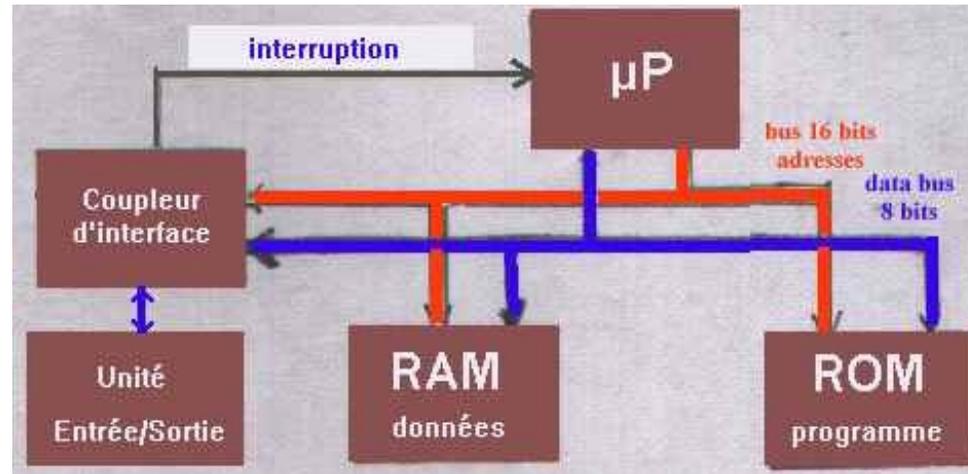
Un micro-ordinateur peut certes faire des calculs, assure la gestion d'un cabinet médical, sera aussi employé à des applications d'informatique industrielle.

Un micro-ordinateur peut servir à contrôler un processus industriel ← des périphériques spécifiques : micro-système ou système à microprocesseur.



Micro-système

Composition d'un système minimal.



## V- Environnement d'un $\mu P$

Microprocesseur  $\rightarrow$  composant essentiel d'un micro-système. Présente divers avantages :

- moins de composants
- modification aisée du programme
- consommation réduite.

Tout **seul un  $\mu P$  ne peut rien faire**, c'est comme un moteur de voiture sans roues, sans boîte de vitesse et sans carrosserie... Il comporte diverses broches d'entrée/sortie sous forme de **bus**.

### 1- L'environnement du $\mu P$

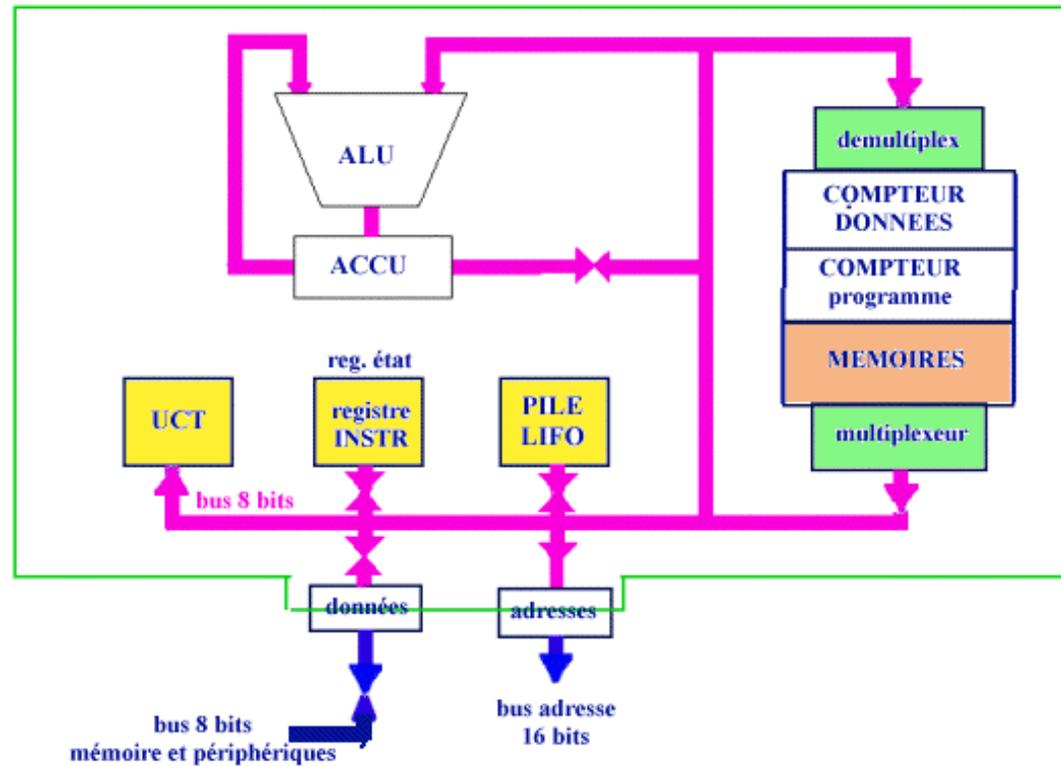
- Mémoires ROM et RAM : stockage de données et du programme applicatif
- Périphériques : autres dispositifs reliés au  $\mu P$  par le biais d'un coupleur d'interface.

## 2- Fonctionnement simplifié du $\mu P$

### a- Constitution

Un  $\mu P$   $\rightarrow$  3 parties principales :

- une unité de contrôle (**UCT**),
- une unité arithmétique et logique (**ALU**),
- un certain nombre de registres, le tout relié par un réseau complexe de bus interne.



*Structure interne d'un microprocesseur*

## b- Cycle Fetch

Un  $\mu$ P sous tension va travailler ← ordre en "levant un drapeau".

Le  $\mu$ P peut démarrer.

Soit le CP contienne la valeur 0800H (en hexadécimal sur 16 bits) : adresse de la première instruction à effectuer.

Pour identifier cet ordre le  $\mu$ P → aller le chercher (*to fetch* en anglais).

En interne le bus du  $\mu$ P est sur 8 bits  $\Rightarrow$  2 phases pour transmettre une adresse de 16 bits. Soit :

- transfert de l'octet de poids faible du CP dans le registre d'adresse,
- transfert de l'octet de poids fort du CP dans le registre d'adresse,
- envoi d'un ordre de lecture à la mémoire centrale (RD),
- attente de WAIT puis,
- transfert du contenu du poids faible du CP vers l'ALU,
- incrémentation de 1 et retour dans le CP,
- transfert du poids fort du CP dans l'ALU et addition de la retenue éventuelle de l'opération précédente, puis retour dans le CP,
- simultanément la donnée est transférée de la mémoire vers le registre de données via le bus de données.

Ensemble d'opérations : **premier cycle machine** ou cycle **FETCH**.

# Les processeurs Motorola

## I- Avant-Propos

Les premiers produits créés par : Intel, Motorola, Rockwell, Zilog, Texas, NS, Fairchild et bien d'autres  
2 géants : Intel et Motorola (concept **CISC** (*Complex Instruction Set Computer*)). Coté Motorola :

- Les processeurs : 6800 - 6802, 6809, 68000, 68020, 68030 ...
- Les circuits de la périphérie : PIA 6821, ACIA 6850, DUART 68681, USART ...
- Les microcontrôleurs : 6805, 68HC11, 68302, 68331, 68332, 68340 ...

## II- Introduction

### A- Qu'est-ce qu'un microprocesseur CISC?

Les **CISC** : principale caractéristique → mettre à disposition du programmeur un jeu d'instructions très développé et un choix de modes d'adressage important.

### B- Architecture de Communication

#### 1- Les signaux

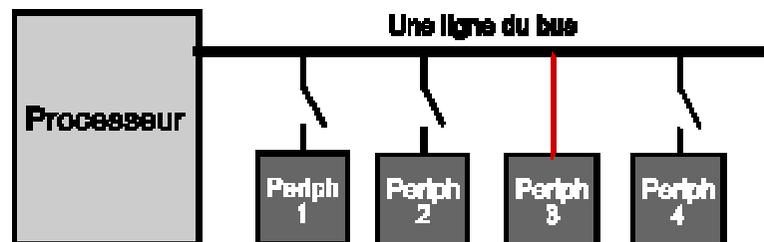


Figure 1

Le processeur communique avec son environnement en échangeant des données codées sur n bits par exemple :

- le niveau électrique 0 V (0 binaire),
- le niveau électrique 5 V (1 binaire)

⇒ besoin de faire appel à n lignes électriques pour véhiculer les données (**Bus de données**).

n caractérise ce que l'on a coutume d'appeler la « taille du microprocesseur ».

Bus de données est bidirectionnel.

Le processeur échange avec tout périphérique en le considérant comme un emplacement en mémoire.

Afin d'entamer un dialogue avec un périphérique, le processeur doit fournir l'adresse de ce dernier.

L'adresse est unique ⇒ une combinaison binaire unique, présente sur le **bus d'adresse**, un seul emplacement.

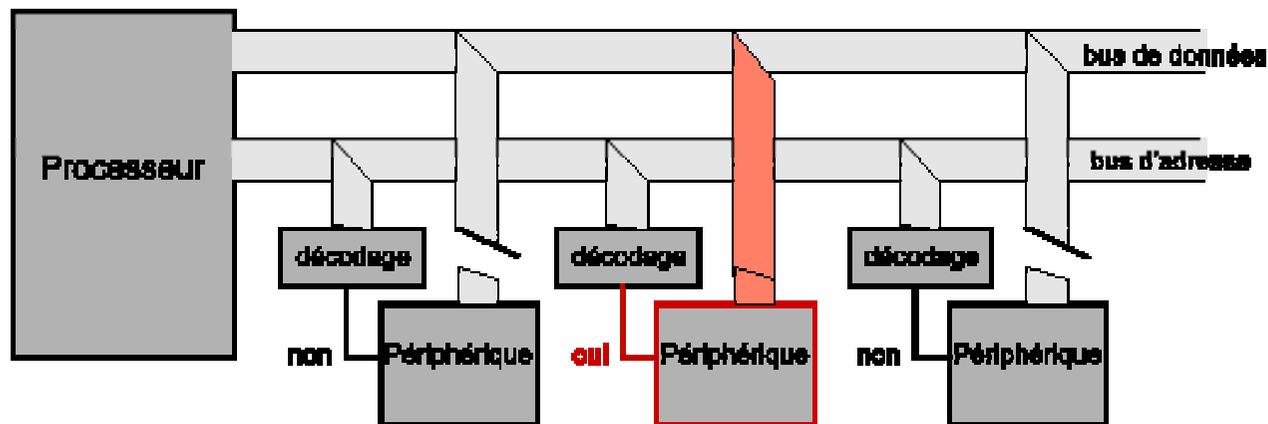


Figure 2

Un troisième bus appelé **bus de contrôle** assure le contrôle directionnel, temporel et événementiel des échanges.

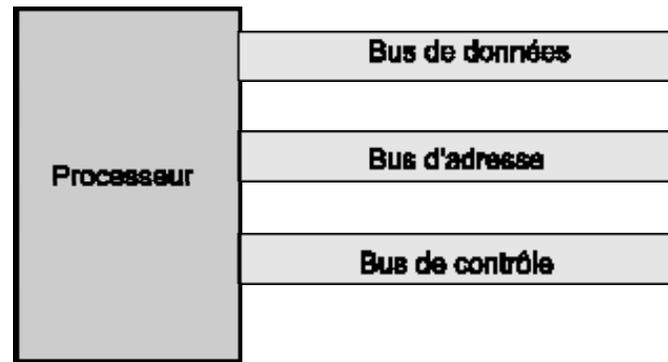


Figure 3- Architecture de communication

## 2- Le système à microprocesseur

Un système est une combinaison entre un microprocesseur et divers périphériques.

Son fonctionnement est décrit par un **programme** dont deux types :

- Le programme général (figé) définissant, le fonctionnement des périphériques,
- Le programme applicatif (libre) caractérisant l'application en cours de fonctionnement sur la machine.

Exemple : micro ordinateur.

- Lors de sa mise sous tension, exécute un programme (lié au matériel) qui initialise tous les périphériques,
- Ensuite charge, depuis le disque, un système d'exploitation.

Le programme figé → mémoire morte **ROM** (Read only memory) ;

Le programme libre pour une application donnée → mémoire vive **RAM** (Random access memory).

Le monde extérieur interagit avec l'application → **Interfaces d'entrée/sortie** (I/O qui signifie Input Output).

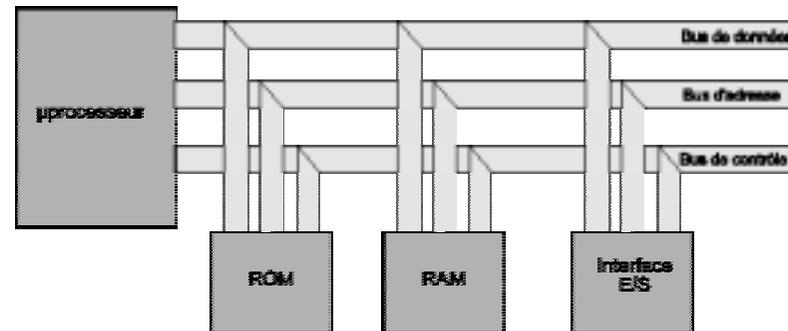


Figure 4- Système à microprocesseur

## C- Le modèle générique de programmation

La programmation est une succession d'appels d'instructions auxquelles on attribue des opérandes :

**Instruction Opérande1, Opérande2, ..., Opérande n**

Une instruction → micro câblage électronique → délivre en sortie des signaux dépendants des signaux d'entrée.

L'ensemble des câblages d'instructions → regroupé dans l'**ALU** (Arithmétique and Logic Unit).

Une instruction → repérée par un code → la localisation ligne-colonne du câblage de l'instruction dans L'ALU.

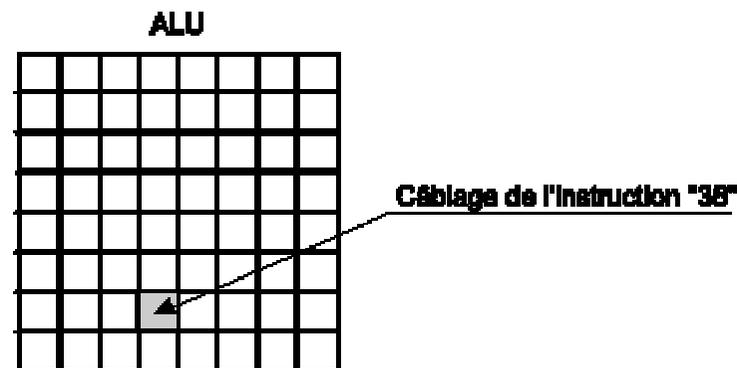


Figure 5- Structure de l'ALU

Exemple d'instruction :

Un processeur CISC possède un unique bus de données, il ne peut donc effectuer l'opération suivante :

***Additionner le contenu de la mémoire d'adresse 100 avec le contenu de l'interface de clavier et mettre le résultat dans l'interface d'écran***

⇒ faut, simultanément **trois** emplacements, or il n'en voit qu'un par l'intermédiaire de son bus de données.

⇒ Il utilise pour résoudre ce problème une mémoire interne appelée Registre de données.

Soit l'instruction ci-dessus qui est décomposée de la manière suivante :

- ***Transférer le contenu de la mémoire d'adresse 100 dans le registre de données***
- ***Additionner le contenu de l'interface clavier au registre de données***
- ***Copier le contenu du registre de données dans l'interface écran***

Un processeur peut posséder plusieurs registres de données.

Exemple d'algorithme :

***POUR  $i=2000$  A  $2100$  FAIRE***  
***Copier le contenu de l'adresse  $i$  dans l'adresse  $i+1000$***   
***FINFAIRE***

La boucle contient deux définitions d'adresses variables ⇒ calcul d'une adresse.

L'adresse ne peut pas figurer dans le programme.

Pour effectuer le calcul on se place dans un **Registre d'adresse**.

Autre Exemple d'algorithme (prise de décision) :

***SI condition rempli ALORS***  
***Effectuer traitement 1***  
***SINON***  
***Effectuer traitement 2***  
***FINSI***

Afin de permettre la prise de décisions (conditions : <, >, =, >0, <0, etc...), un registre est indispensable : **Status**.

Le **Status** contient des témoins (flag) → A tester pour décider de la poursuite en séquence ou de la rupture.

Exemples (témoin de zéro Z) :

***Copier le contenu d'une adresse dans un registre***

Met Z à 1 si contenu de l'adresse recopiée est 0.

***Soustraire une valeur d'un registre***

Met Z à 1 si résultat de la soustraction est 0.

⇒ possibilité de comparer une valeur calculée à une autre valeur.

***Effectuer un calcul dans un registre***  
***Soustraire la valeur 100 au contenu du registre***  
***SI Z=1 ALORS***  
    ***Effectuer Traitement 1***  
***SINON***  
    ***Effectuer traitement 2***  
***FINSI***

→ En assembleur : décomposer la structure SI en instructions supportées par le processeur concerné :

**Brancher à Tr2 SI Z=0**  
**<Traitement 1>**  
**Brancher à FINSI**  
**Tr2: <Traitement 2>**  
**FINSI:**

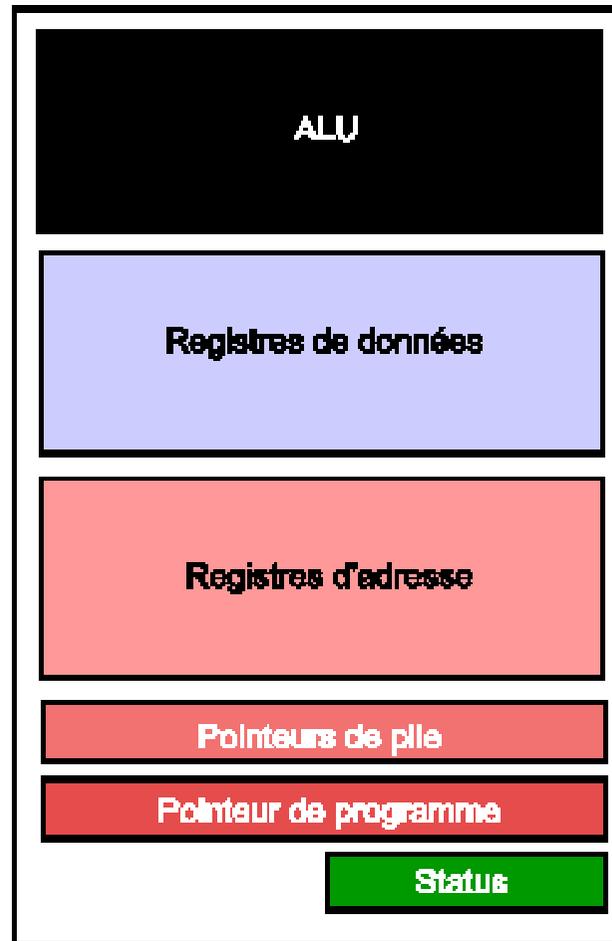


Figure 6- Modèle générique d'un processeur CISC

### III- Les microprocesseurs CISC et les microcontrôleurs de Motorola

#### A- LES CISC

## 1- Le 6800 Le premier né :

Architecture interne :

- 1 ALU.
- 2 registres de données sur 8 bits (accumulateurs A et B).
- 1 registre d'adresse sur 16 bits (index X).
- 1 pointeur de pile sur 16 bits.
- 1 pointeur de programme sur 16 bits.
- 1 status sur 8 bits dont 6 utiles.

L'architecture de communication repose sur :

- 1 bus de données sur 8 bits.
- 1 bus d'adresse sur 16 bits.
- 1 bus de contrôle.

## 2- Le 6802

Ne diffère du 6800 que par 2 points :

- 128 octets de RAM interne adressée de \$00 à \$7F.
- Possède un état de repli en faible consommation dans lequel la RAM interne est maintenue.

## 3- Le 6809

Progrès considérable par rapport au 6800.

Architecture interne :

- 1 ALU
- 1 registre de données sur 16 bits (accumulateur D), ou utilisé comme 2 registres sur 8 bits (accumulateur A : octet fort et accumulateur B : octet faible).
- 2 registres d'adresse d'usage général sur 16 bits (index X et Y).
- 1 registre d'adresse sur 16 bits soit d'usage général, soit comme pointeur de pile.
- 1 pointeur de pile dédié sur 16 bits.
- 1 pointeur de programme sur 16 bits.
- 1 status sur 8 bits.

L'architecture de communication repose sur :

- 1 bus de données sur 8 bits.
- 1 bus d'adresse sur 16 bits.
- 1 bus de contrôle.

#### 4- Le 68000

Plus de compatibilité avec les processeurs antérieurs.

Modèle de programmation :

- 1 ALU.
- 8 registres de données sur 32 bits (utilisables sur 16 ou 8 bits).
- 7 registres d'adresse sur 32 bits (24 seulement sont utilisables).
- 2 pointeurs de pile sur 32 bits (24 seulement sont utilisables).
- 1 pointeur de programme sur 32 bits (24 seulement sont utilisables).
- 1 status sur 16 bits.

L'architecture de communication repose sur :

- 1 bus de données sur 16 bits.
- 1 bus d'adresse sur 23 bits permettant d'adresser  $2^{24}$  octets.
- 1 bus de contrôle.

#### 5- Le 68010

Passé quelque peu inaperçu alors qu'il apporte par rapport au 68000 des fonctionnalités intéressantes.

Son modèle de programmation comporte 3 registres nouveaux par rapport au 68000 :

- 1 registre de Base Vecteur (VBR).
- 2 registres de transfert entre espaces (SFC, DFC).

L'architecture de communication est identique à celle du 68000.

## 6- Le 68020

Au modèle de programmation du 68010 on ajoute :

- 1 troisième pointeur de pile
- 2 registres pour contrôler et adresser la mémoire cache.

L'architecture de communication comporte :

- 1 bus de données est sur 32 bits (donc un vrai 32 bits).
- 1 bus d'adresse est sur 32 bits.
- 1 bus de contrôle.

## 7- Le 68030

Apporte par rapport au 68020 :

- 1 unité de gestion de mémoire paginée (PMMU) intégrée,
- 1 bus synchrone, et
- 1 mémoire cache de données interne de 256 octets.

Son modèle de programmation comporte :

- les mêmes éléments que le 68020 augmentés,
- du registre de contrôle de mémoire paginée, et
- du registre d'adressage de cette même mémoire paginée.

## 8- Evolution du modèle de programmation

Les schémas des *figure 7* et *8* présentent les modèles de tous les processeurs présentés.

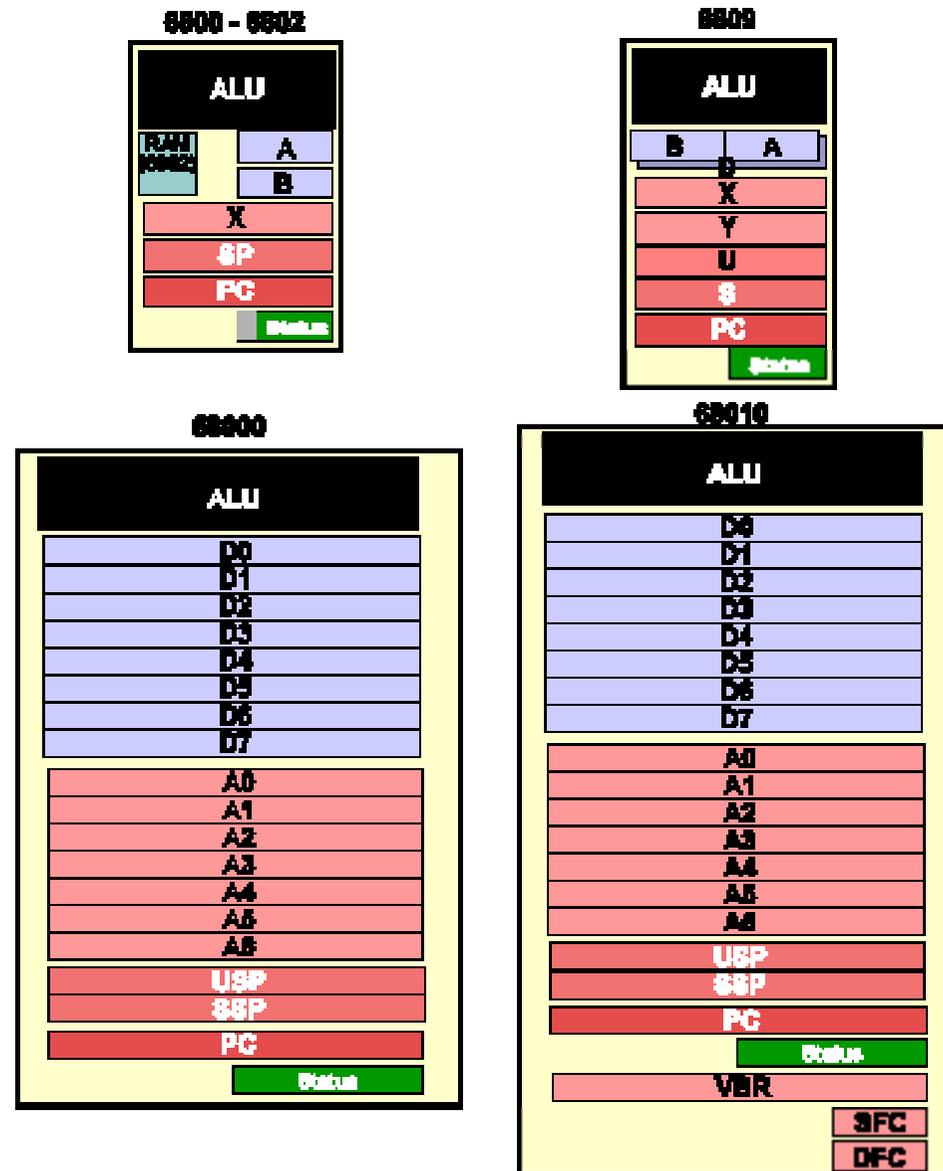


Figure 7- Registres internes : 6800, 6802, 6809, 68000, 68010

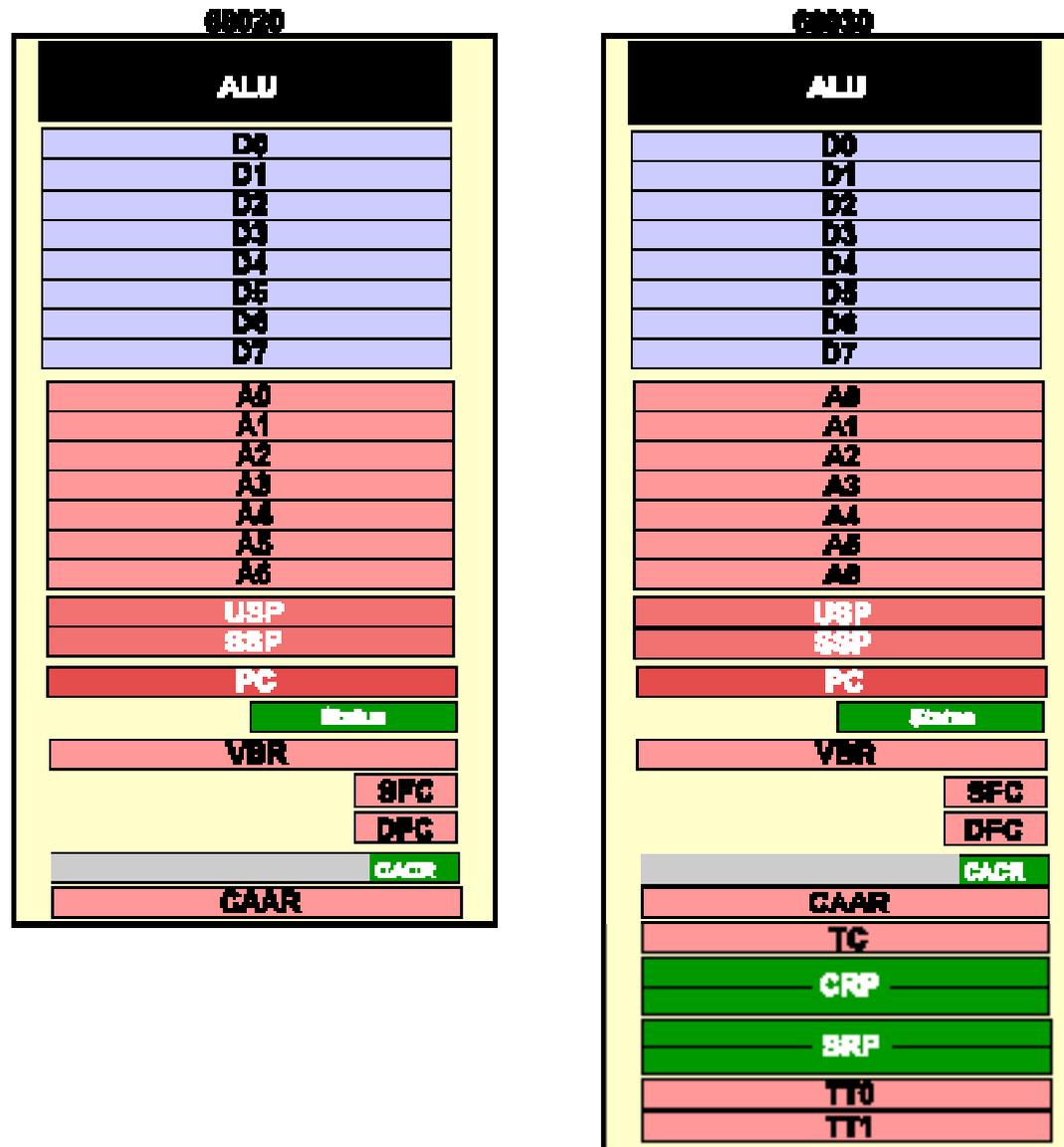


Figure 8- Registres internes : 68020, 68030

## B- La notion de Microcontrôleur

L'objectif d'un microcontrôleur → diminuer les interconnexions de circuits.

⇒ Intégrer sur la même puce de silicium le microprocesseur et un certain nombre de périphériques.

Les périphériques intégrées :

- La mémoire RAM
- La mémoire ROM ou plus souvent OTPROM
- La mémoire EEPROM
- Des ports de liaisons parallèles
- Des ports de liaison série
- Des compteurs temporisateurs
- Des convertisseurs numériques analogiques et analogiques numériques
- Des contrôleurs de DMA, etc...

## C- La familles des microcontrôleurs

La famille la plus ancienne de Motorola : famille 6805 (le plus connu, le 68705).

Parallèlement au développement de la famille 68HC11, Motorola a développé des microcontrôleurs dotés d'un cœur 16 et 32 bits (68000 68020), c'est la famille des 683XX.

Pour les microcontrôleurs nous distinguerons 3 approches :

- Le modèle de programmation du cœur → modèle de programmation du microprocesseur intégré.
- Le modèle de programmation applicatif, qui définit l'organisation des périphériques internes.
- Le modèle de communication qui peut être de 2 natures :
  - Communication avec des périphériques externes.
  - Communication avec l'application.

## 1- Le 6805

Plus simple que le cœur du 6800

Le modèle de programmation comporte :

- Un registre de données sur 8 bits (l'Accumulateur A)
- Un registre d'adresse sur 8 bits (l'Index X)
- Un pointeur de programme sur 11 bits (PC)
- Un pointeur de pile sur 5 bits (SP)
- Un status sur 5 bits.

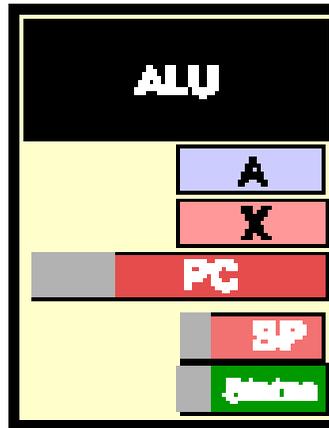


Figure 9- Le modèle de programmation 6805

Le modèle applicatif fait intervenir les circuits périphériques internes (*figure 10*) :

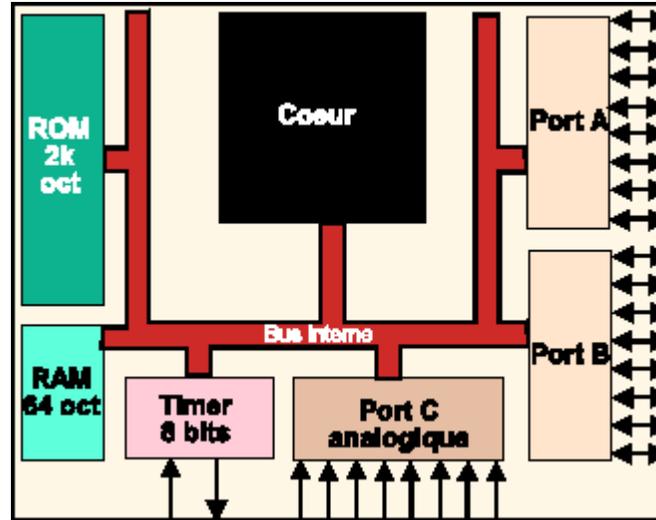


Figure 10- Modèle applicatif du 6805

**Pas** de modèle de communication avec les périphériques externes.

La seule communication se fait avec l'application elle-même, via :

- les ports A et B,
- les entrées analogiques du port C, ou
- l'entrée de comptage et
- la sortie temporisée du timer.

## 2- Le 68HC11

Les 68HC11 possèdent des bus d'extension optionnels, et peuvent donc enrichir l'application.

Le modèle de programmation du 68HC11 se schématise sur la figure 11 :



Figure 11- Le modèle de programmation du 68HC11

Le modèle applicatif du 68HC11 schématisé sur la *figure 12* est composé de :

- mémoire ROM,
- mémoire RAM,
- mémoire EEPROM,
- 1 compteur temporisateur sur 16 bits,
- 5 ports parallèles Tout Ou Rien (TOR) pouvant être transformés en :
  - Bus d'extension (adresse et données)
  - Port série synchrone
  - Port série Asynchrone
  - Port d'entrées analogiques
  - Entrées de comptage
  - Sorties événementielles.

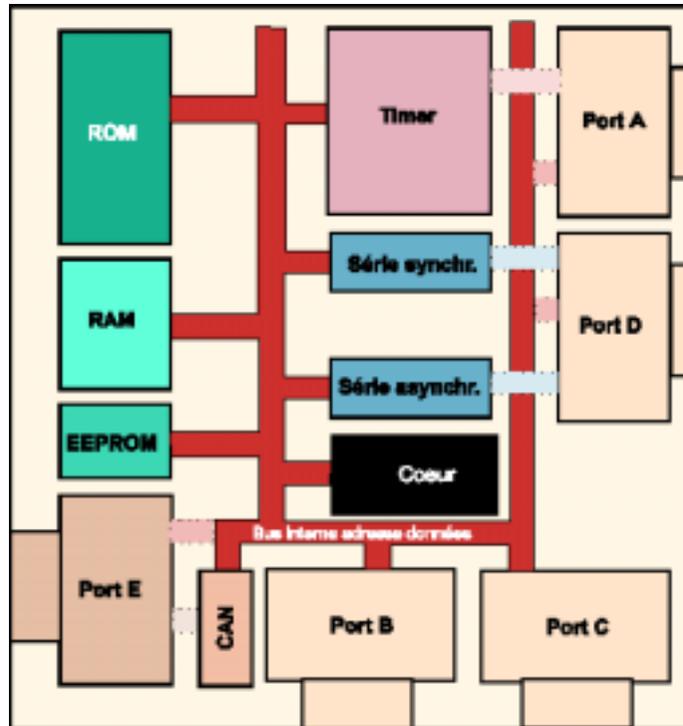


Figure 12- Modèle applicatif du 68HC11

### 3- Le 68302

Le 68302 : microcontrôleur doté d'un cœur 68000.

Son modèle de programmation est celui du 68000.

Le modèle applicatif se décompose en 3 parties :

- Le cœur 68000
- Le bloc du système intégré
- Le processeur de communication

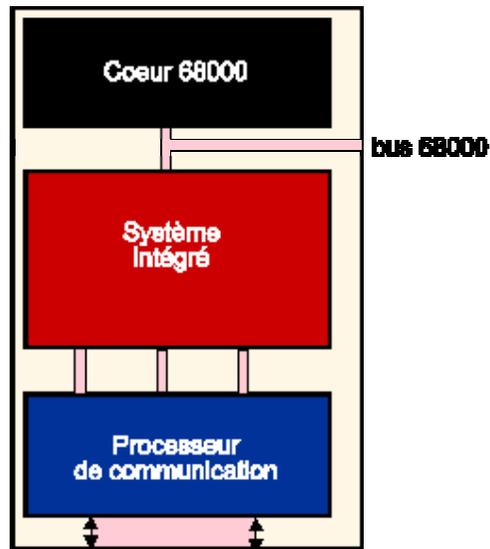


Figure 13- Schéma bloc du 68302

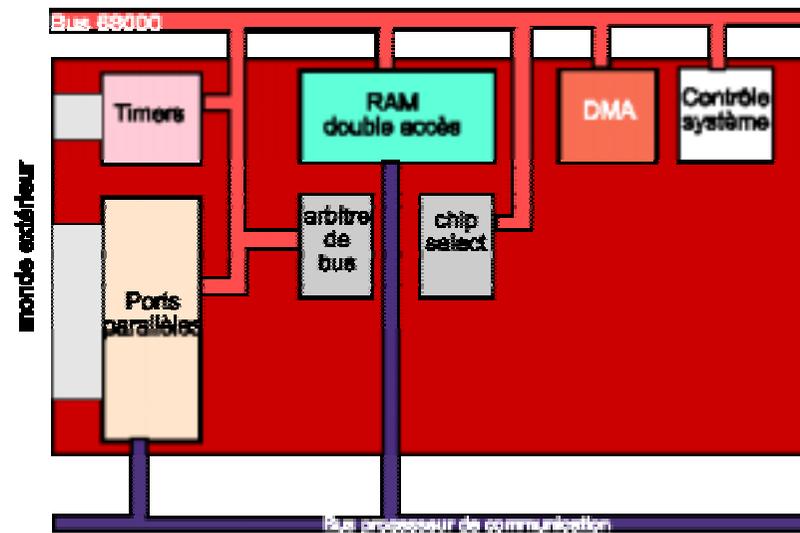


Figure 14- Le système intégré du 68302

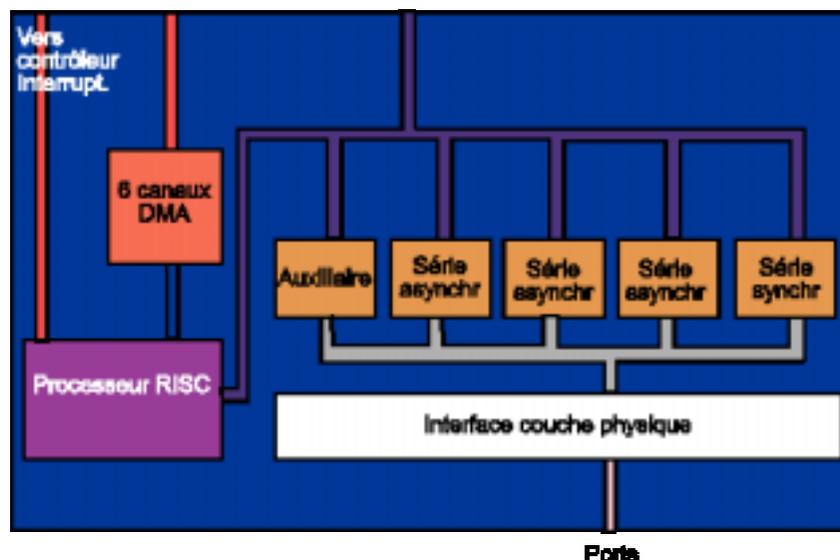


Figure 15- Processeur de communication du 68302

#### 4- Le 68331

Possède un cœur 68020.

Ses bus de communication avec les périphériques sont configurables.

Données échangées avec l'extérieur sur 16 et non 32 bits, quand aux adresses elles sont sur 24 bits.

On peut distinguer 4 grands blocs dans le modèle applicatif :

- Le cœur 68020.
- Le module sériel (QSM : Queued Serial Module)
- Le module compteur temporisateur (GPT : General-Purpose Timer)
- Le module de gestion des bus (SIM : System Integrated Module)

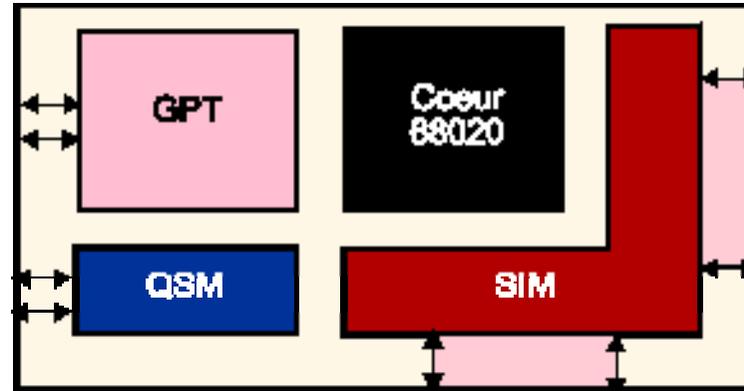


Figure 16- Modèle applicatif du 68331

## 5- Le 68340

Pas de mémoire interne.

Cœur : 68020.

Le modèle applicatif comporte 6 modules principaux :

- Cœur 68020.
- 2 modules compteurs temporisateurs.
- 1 module double liaison série asynchrone comparable à un DUART.
- 1 module contrôleur de DMA 2 canaux.
- 1 module de gestion de bus comparable au SIM du 68331.

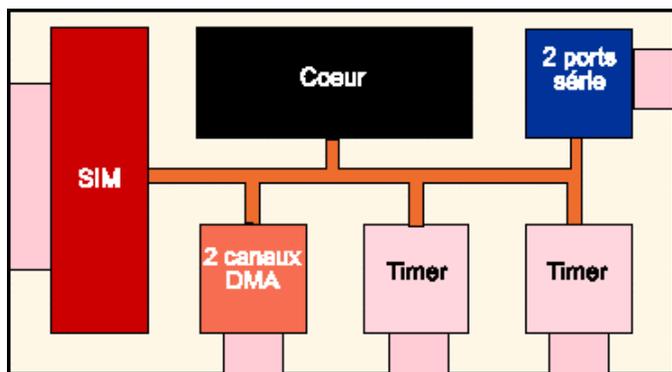


Figure 17- Modèle applicatif du 68340

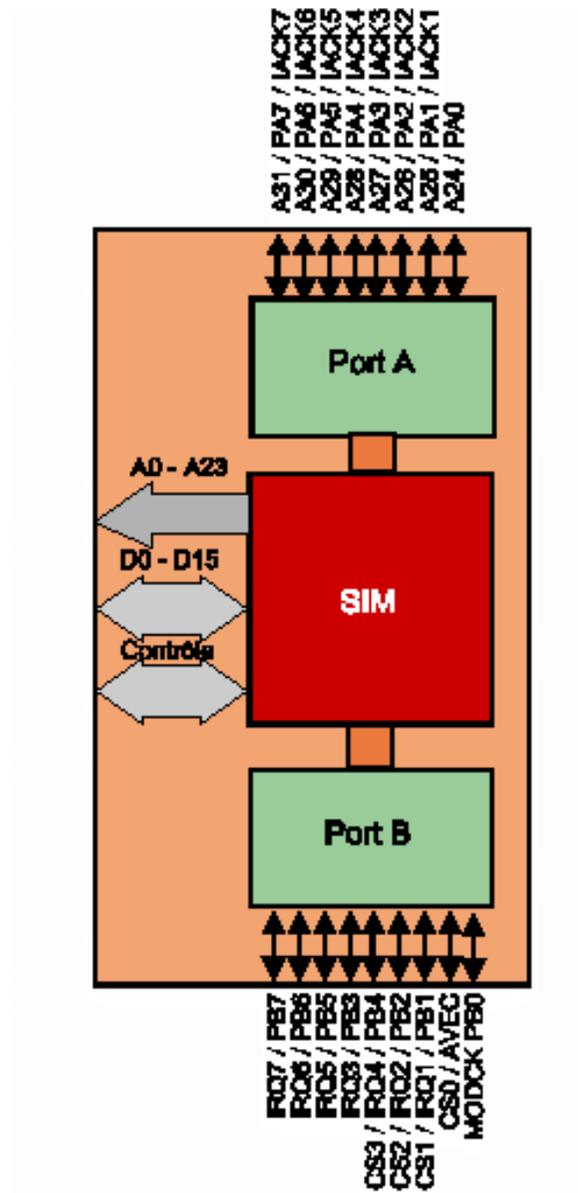


Figure 18- Le SIM du 68340

## 6- Exemples d'applications à base de microcontrôleurs

- Le 6805 (version REPROM 68705) utilisé dans des terminaux pour les réglages sur automates programmables.
- Testé sur un module de réglage des fonctions d'un ampli HI-FI.
- 68HC11 : Développement des applications tels des modules de gestion de distribution électrique et de réglage d'appareils de chauffage et d'éclairage.
- Modules utilisant un 68HC11 gèrent les différents mode de paiement (cartes à pré-paiement, carte Visa).
- Application très intéressante avec un 68302 : carte de communication intelligente pouvant être utilisée comme concentrateur de réseau.
- Le 68331 est utilisé au cœur d'une unité centrale d'automate programmable dont les modules sont reliés par un réseau FIP.
- Le 68332 est très utilisé dans le domaine de l'automobile, ....
- Le 68340 unité centrale d'automate programmable.