

LE MICROPROCESSEUR ET SES PÉRIPHÉRIQUES

I/ GÉNÉRALITÉS

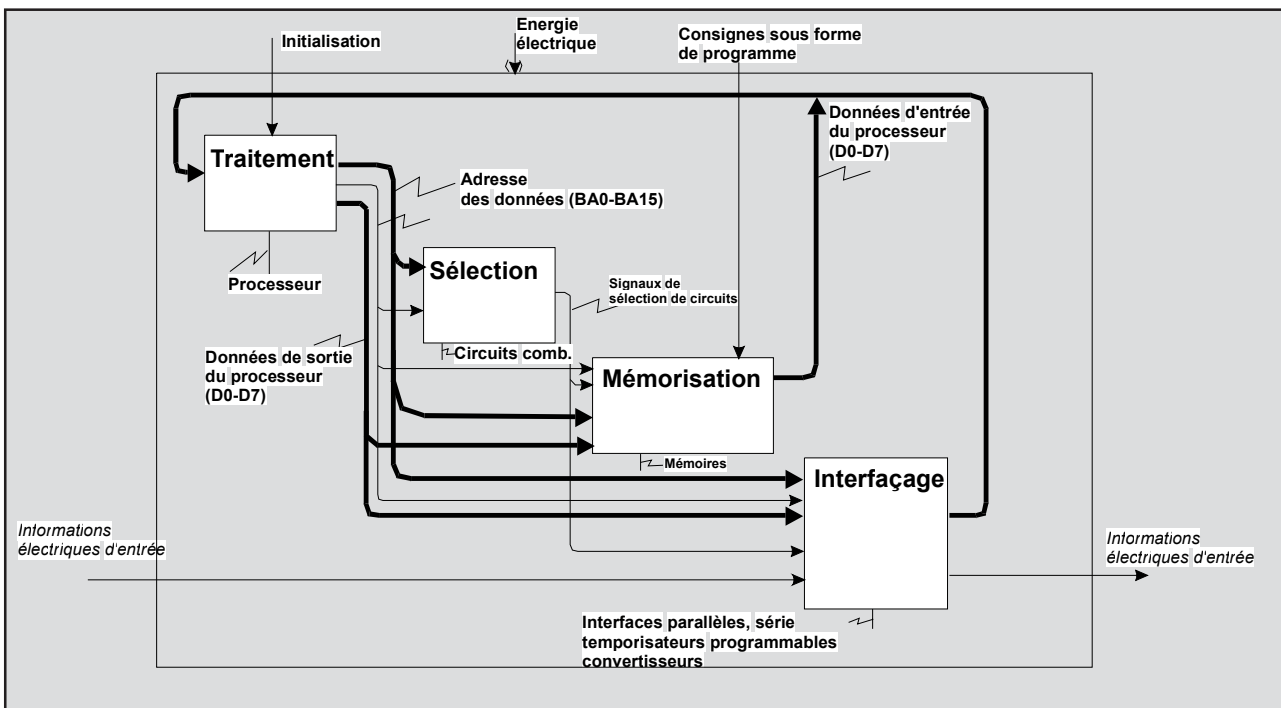
La plupart des dispositifs électroniques reposent aujourd'hui sur une structure minimale microprogrammée assurant le séquençement de tâches plus ou moins complexes selon un algorithme défini.

Selon la complexité ou l'importance des actions à réaliser cette structure peut être :

- ♦ un seul circuit intégré de forte intégration (*LSI : Large Scale Integration*) appelé **microcontrôleur**
- ♦ un ensemble de circuits qui dialoguent ensemble et qui constituent un **système minimal** pour lequel le microprocesseur serait le "cerveau".

L'avantage d'une structure microprogrammée est qu'elle est *adaptable* à n'importe quel système travaillant dans des contraintes technologiques identiques (temps de réponse, encombrement, température, etc...) et qu'elle est capable de gérer un grand nombre de variables logiques ou numériques. Seul ses liaisons avec l'extérieur et son programme de traitement seront changés. On utilise par exemple le même microcontrôleur pour gérer un lave-linge ou l'ordinateur de bord d'une voiture. Enfin son *coût de revient est faible*.

II/ SCHÉMA FONCTIONNEL D'UN SYSTÈME MINIMUM



III/ DÉFINITION DES ÉLÉMENTS CONSTITUTIFS

L'architecture VON NEUMANN a été la plus utilisée dans les premiers systèmes microprogrammés.

III.1/ Fonction Traitement

L'unité centrale de traitement est composée de trois sous-ensembles qui sont :

- l'unité de calcul (UAL : unité arithmétique et logique)
- l'unité de contrôle
- les registres internes

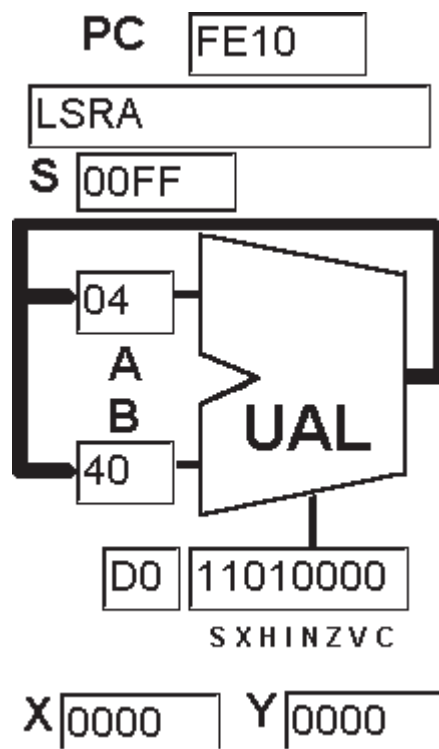
•**L'unité de calcul** : permet de réaliser les opérations logiques ou arithmétiques - demandées par le programme - entre les données (appelées *OPE-RANDE*) et des registres privilégiés appelés ACCUMULATEURS (Ici A et B). Les résultats de l'opérations sont placés dans ces mêmes accumulateurs.

Certaines opérations sont effectuées directement avec la mémoire.

Selon le microprocesseur, les instructions (le jeu d'instruction) ne sont pas les mêmes.

•**L'unité de contrôle ou de séquençement**: elle gère le fonctionnement interne du l'unité centrale : recherche de la prochaine instruction à exécuter, décodage de cette instruction puis exécution. Son fonctionnement est totalement transparent pour l'utilisateur.

•**Les registres internes** : ce sont des mémoires de capacité réduite (un octet ou deux octets pour des microprocesseurs 8 bits) indispensables à l'exécution correct d'un programme.



Dans le cas du 68HC11 les registres sont:

- A et B** :accumulateurs 8 bits qui peuvent être associés pour former un registre 16 bits (utile pour certaines instructions) appelé **D**
- PC** : compteur de programme sur 16 bits: contient en permanence l'adresse de la case mémoire contenant le ou les codes de la prochaine instruction à exécuter.
- S** : pointeur de pile sur 16 bits: il contient l'adresse d'une zone mémoire appelée pile et utilisée pour stocker des données temporaires (une sorte de *brouillon*)
- X et Y** : registres d'index sur 16 bits : registres utilisés pour contenir l'adresse d'une case mémoire contenant une donnée stratégique du programme exécuté ou d'une case ou zone mémoire.
- CC** : *Code Condition* ou *drapeau* ou *Flag* sur 8 bits : ce registre donne des informations sur le résultat d'une opération (*résultat nul* ou *résultat négatif* ou *résultat supérieur à 255*, etc...). Les plus courants sont :



- Z : zéro : _____
- C : retenue : _____
- N : négatif : _____

L'unité de traitement est intégré dans un **MICROPROCESSEUR** et travaille toujours avec des **mots codés en binaire**, cependant pour l'utilisateur, pour des raisons de simplification d'écriture, ces mots sont **transcrits en hexadécimal**.



Lorsqu'elle est associée à de la mémoire ou/et des périphériques d'entrée-sortie dans un même circuit, celui-ci est appelé **MICROCONTROLEUR**.

III.2/ Mémorisation

Cette fonction permet de stocker les données numériques organisées en mots de 8 bits (pour des microprocesseurs de 8 bits) à une adresse imposée (sur 16 bits pour des microprocesseurs de faible capacité)

- Une partie de la mémoire sera à lecture seule : on l'appelle **Mémoire morte** ou **Mémoire passive** ou **ROM (Read Only Memory)**.

Elle stocke le programme et les données permanentes nécessaires au programme (constantes, etc...). On l'appelle parfois la *Mémoire programme*.

Les ROM sont de type:

- Permanentes programmées en usine: on les appelle ROM
- EPROM : Effaçable Programmable : mémoires programmables électriquement et effaçable par UV. On les reconnaît par leur petite fenêtre de quartz qui permet aux UV d'atteindre le silicium
- EEPROM : effaçable électriquement PROM : plus souple que les précédentes
- OTP ROM : One time programming ROM : identique aux ROM mais elles peuvent être programmées par l'utilisateur une seule fois.

- Une deuxième partie pourra être lue ou écrite : on l'appelle **Mémoire vive** ou **Mémoire active** ou **RAM (Random Access Memory)**.

Elle contient toutes les variables et les données et les données temporaires nécessaires au programme pour son bon fonctionnement (*Mémoire données*).

Les deux catégories essentielles sont les

- SRAM : RAM statique : accès rapide mais coûteuses donc peu adaptées à des mémoires massives (type ordinateur PC)
- DRAM : RAM dynamique : bon rapport capacité/prix, mais il est nécessaire de rafraîchir régulièrement les données mémorisées ce qui entraîne une diminution des performances en temps.

Pour une mémoire vive le signal Read/Write (R/W) provenant du microprocesseur définit si le microprocesseur lit le contenu (R/W = 1) de la RAM ou écrit dans la RAM (R/W = 0).

La capacité d'une mémoire est exprimée en Kilo-Octets.



1 Koctet = 1024 octets = 2¹⁰ octets = 1024 x 8 bits

Exemple : Une mémoire possédant 12 fils d'adresse (A0 - A11) et 8 fils de données contient _____

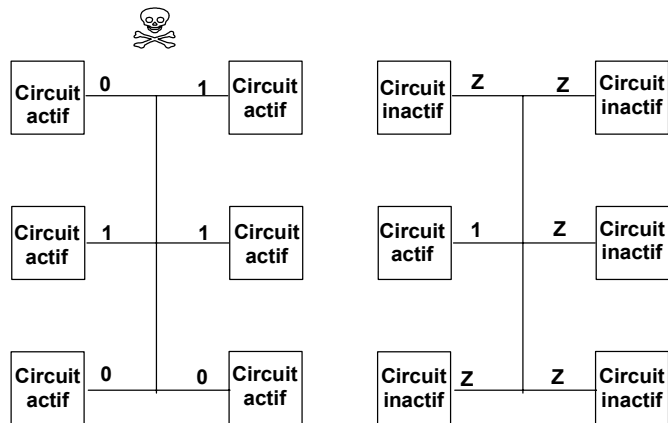
III.3/ Sélection

Cette fonction appelée parfois "*Décodage d'adresse*" permet les connections entre le microprocesseur et ses circuits périphériques (mémorisation, interfaçage, etc...).

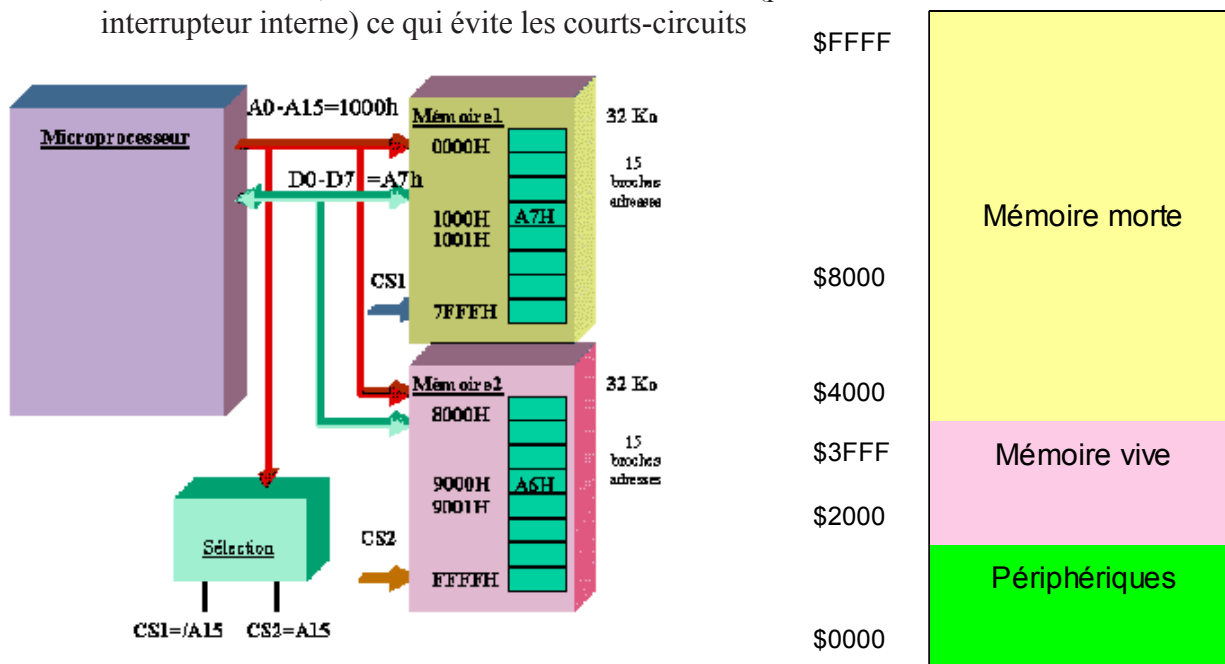
Les circuits périphériques et les mémoires sont connectés au microprocesseur par des

^{"bus"}
1 : On appelle **bus** un ensemble de fils électriques synchronisés entre eux et répondant à une même fonction (exemple : véhiculer des informations).

Ainsi, un même fil de bus est relié à plusieurs sorties pouvant être à des niveaux logiques différents. Pour éviter des courts-circuits sur ces fils de bus, il est nécessaire de définir quel circuit ou périphérique envoie son information sur le bus. La fonction **Sélection** à partir de l'adresse transmise par le µP, active des signaux logiques qui valident ou non la sortie des circuits périphériques.



Ces signaux logiques appelés souvent **Chip Enable (validation de circuit)**, **Output Enable (validation de sortie)** ou **Enable (validation)** sont reliés aux entrées du même nom des circuits périphérique. Si un tel signal est actif, le circuit place les signaux électriques de sortie sur le bus, sinon il se déconnecte du bus (par un interrupteur interne) ce qui évite les courts-circuits



III.4/ Interfaçage

Cette fonction permet la communication entre le microprocesseur et les dispositifs périphériques (*clavier, affichage, ligne téléphonique, etc...*).

Les circuits d'interface sont de différents types :

- **parallèle** : ils assurent l'échange des informations parallèle.
Application: clavier, afficheur 7 segment, etc...
- **série** : ils assurent l'échange des informations sérielle.
Application : liaison série sur longue distance
- **Temporisateur programmable** : ils solutionnent les applications temporelles.
- **Convertisseurs N/A-A/N** : ils assurent la liaison entre le microprocesseur et les signaux analogiques.

IV/ Mode d'échange et initialisation

IV.1/ Bus de donnée

Les données (8 bits pour un μP de 8 bits) arrivent et partent de l'unité de traitement par 8 fils véhiculant chacun 1 niveau logique donc un bit. Ces 8 fils sont appelés **BUS de DONNEES**, ils sont bidirectionnels et sont notés D0 à D7 en général.

IV.2/ Bus d'adresse

Les données sont stockées par exemple dans une partie de la mémoire. Chaque donnée occupe une place qui lui est propre et qui est identifiée par son *adresse* sur 16 bits (en général). Cette adresse est spécifiée par le microprocesseur grâce au **BUS D'ADRESSE** qui regroupe 16 fils électriques véhiculant chacun un seul bit. Ce bus est unidirectionnel et maîtrisé entièrement par l'unité de traitement.

Ainsi un microprocesseur possédant un bus d'adresse de 16 bits (A0 à A15) est capable d'accéder (*d'adresser*) à 2^{16} données soit 65536 adresses donc un espace de 64 koctets.

IV.3/ Bus de contrôle

Le bus de contrôle véhicule des signaux logiques utiles au fonctionnement et à la synchronisation des différents circuits entre eux. Parmi ceux-ci notons :

- **R/W** : signal à l'état logique 1 lorsque le μP procède à une lecture de périphérique et à l'état 0 lors d'une écriture
- **E** : signal d'horloge assurant le cadencement et la synchronisation des circuits périphériques avec le microprocesseur
- **RESET** : signal actif à 0. Il assure l'initialisation du μP et des périphériques auxquels il est connecté.
- **IRQ** : Signal d'interruption permettant d'interrompre le programme en cours pour exécuter des tâches estimées plus prioritaires par le concepteur.

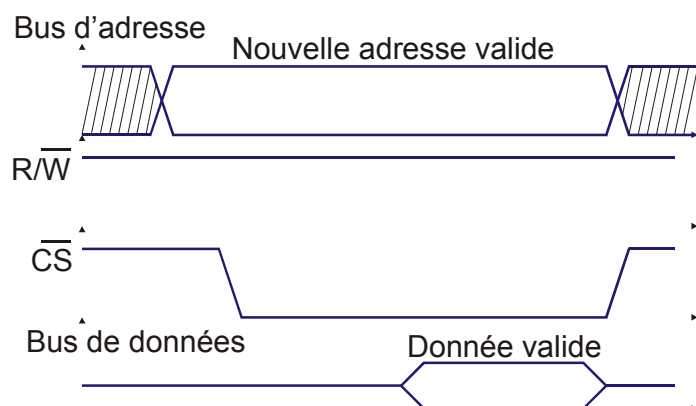
IV.4/ Exemple de dialogue entre μP et mémoire

Pour réaliser une lecture d'une case mémoire, les étapes sont les suivantes :

- le μP met l'adresse sur le bus d'adresse
- le μP précise qu'il veut réaliser une lecture : R/ W est mis à 1
- la structure de sélection valide au regard de l'adresse le circuit contenant la donnée recherchée

-le circuit devient actif et restitue sur le bus de donnée l'information recherchée.

-le μP récupère l'information et procède à une nouvelle écriture ou lecture



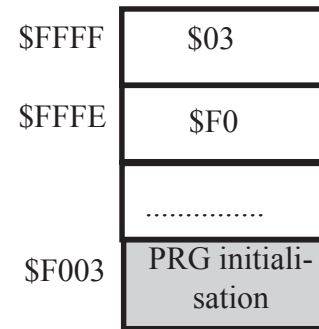
IV.5/ Initialisation

Lors de l'initialisation du microprocesseur (RESET), dans le cas d'un microprocesseur MOTOROLA (68HC11, 6805, 6809, etc...) l'unité de commande va chercher l'adresse présente dans les deux dernières cases mémoires.

Cette adresse constitue l'adresse de la première instruction à exécuter.

Pour un déroulement correct de cette procédure et pour éviter un «plantage» du microprocesseur, l'adresse \$F003 (dans l'exemple ci-contre) doit exister et doit comporter une suite d'instruction cohérente.

Il faut, par ailleurs, qu'après une remise en route de la machine, les données soient toujours présentes dans la mémoire. Il faudra donc qu'elles soient stockées dans de la mémoire morte.



VI/ LES CIRCUITS PÉRIPHÉRIQUES

Appelés souvent *coupleurs*, ces structures (externe pour le microprocesseur, interne pour le micro-contrôleur) permettent l'interfaçage entre le microprocesseur (l'unité de traitement) et les structures de dialogue avec l'homme (clavier, visualisation...) ou d'autres dispositifs électriques (données numériques d'un disque dur, tension issue d'un capteur...).

On rencontre en général trois types de coupleurs numériques :

- les coupleurs parallèles
- les coupleurs série
- les temporisateur programmable


Remarque : il existe également des circuits périphériques assurant la conversion de tensions analogiques vers des données numériques et réciproquement. Ils seront étudiés ultérieurement.

V.1/ Coupleurs parallèles

On réalise une communication parallèle lorsque plusieurs bits sont transférés sur des fils différents d'un émetteur vers un récepteur.

Le rôle du coupleur parallèle est d'assurer cette liaison entre "l'extérieur" et le microprocesseur. Pour ce faire, il dispose généralement d'un registre de configuration, d'un registre d'entrée sortie, et est en liaison avec le microprocesseur par l'intermédiaire du bus de donnée et du bus d'adresse.

Exemple du coupleur parallèle C interne au 68HC11 :

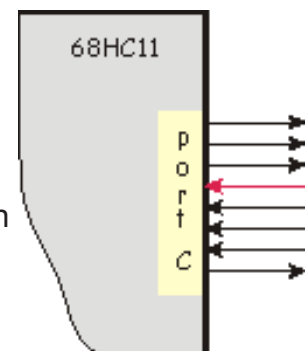
 Ce port est _____ ce qui signifie que ses fils de sortie peuvent recevoir des signaux logiques, mais aussi affecter des niveaux logiques en sortie.

Le registre de direction DDRC permet de _____

Un bit à 0 configure le fil en _____ Un bit à 1 le configure en _____

Le registre d'entrée-sortie PortC permet _____

DDRC=%10000111



Exemple :

Mettre \$F1 dans le DDRA, signifie que les 4 dernières lignes du port A ainsi que la première sont en sorties et les autres en entrée.

Pour lire les états logiques placés sur le port, il suffit d'aller lire le registre d'entrée-sortie correspondant.

De la même manière pour fixer des niveaux logiques sur un port configuré en sortie, il suffit d'écrire l'octet souhaité dans le registre d'entrée-sortie.

V.2/ Temporisateur programmable

Le temporisateur programmable assure une gestion du temps par le microprocesseur. C'est lui qui pourra être utilisé pour réaliser un chronomètre, une horloge, un générateur de fréquence, ou pour faire une mesure de durée d'impulsion...

Comme le coupleur parallèle, il est en relation avec le microprocesseur par l'intermédiaire du bus de données et d'adresse. Selon le type de temporisateur, le nombre de registre interne est plus ou moins important. La complexité et les possibilités de ces circuits ne permettent pas de généraliser leur fonctionnement. Seule une lecture attentive de la documentation permet leur bonne utilisation.

VI.5/ Interface série

Dans une transmission série, les bits sont transmis un à un entre l'émetteur et le récepteur.

La difficulté d'une transmission de ce type réside dans la synchronisation des informations entre l'émetteur et le récepteur. Selon le type de communication, la transmission peut être synchrone ou asynchrone.

Le coupleur série permet cette transmission (en général selon les deux modes), évitant au microprocesseur de devoir gérer les protocoles de transfert.

Remarque : Les transmissions série et parallèle seront vues plus en détail dans le cours consacrés au dialogue.

