

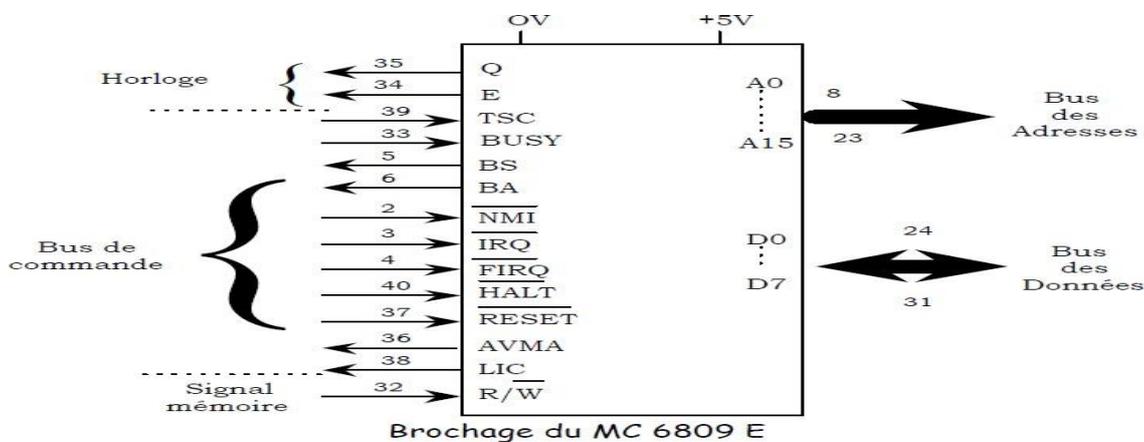
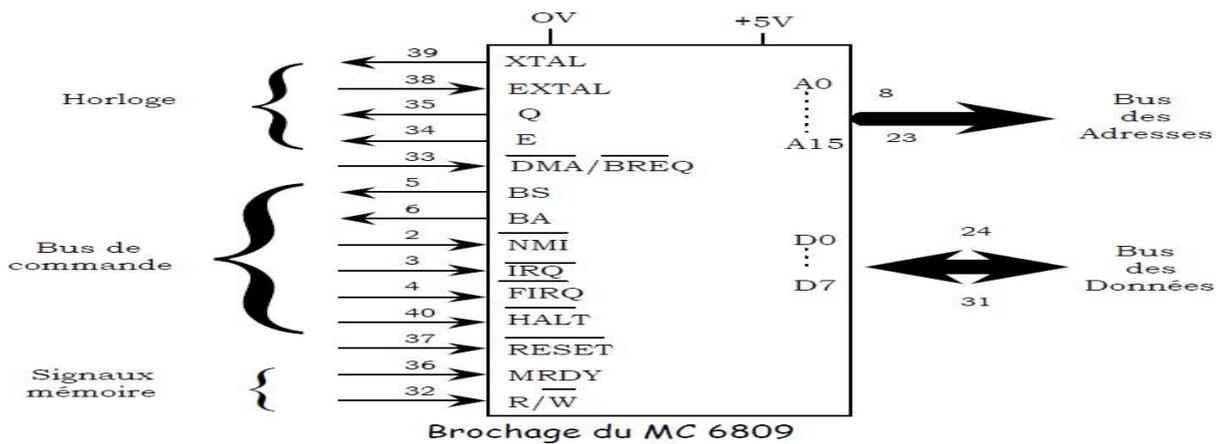
Dans ce chapitre, nous allons présenter les différents moyens ou les différentes liaisons dont dispose un processeur pour communiquer avec l'extérieur et les composants qui l'entourent. On se basera toujours sur le processeur de Motorola le 6809 pour mieux atteindre cet objectif.

1. Communication du processeur avec l'extérieur

1.1. Bus de données (bi-directionnel.), d'adresse (uni-directionnel.) et de contrôle (bi-directionnel.)

Rappel Architecture du 6809

Le microprocesseur 6809 est un processeur 8 bits dont l'organisation interne est orientée 16 bits. Il est fabriqué en technologie MOS (*Metal Oxide Semiconductor*) canal N et se présente sous la forme d'un boîtier DIL (*Dual In Line*) 40 broches. Il est monotension (5V). Il existe deux versions différenciées par l'horloge, le 6809 est rythmé par une horloge interne (f=1 MHz, 1.5 MHz et 2MHz). le 6809E est rythmé par une horloge externe. Ce dernier est adapté aux applications multiprocesseurs. Il présente la particularité de pouvoir être synchronisé par une horloge extérieure. Cependant, il y'a une compatibilité complète entre les 2 versions.



Le bus des données 8 bits (Do à D7)

Ces huit broches sont bidirectionnelles. Elles permettent la communication avec le bus des données interne du microprocesseur. Chaque broche peut "piloter" 1 charge **TTL** (*Transistor-Transistor Logic*) et 8 entrées de circuits appartenant à la famille 6800. Bus en logique 3 états.

Le bus des adresses 16 bits (A0 à A15)

Ces broches unidirectionnelles transfèrent l'adresse 16 bits fournie par le microprocesseur au bus d'adresse du système. Mêmes caractéristiques électriques que pour le bus des données. Bus en logique 3 états. les adresses sont validées sur le front montant de Q.

Le bus de contrôle

La broche **Read/Write**. Cette broche indique le sens de transfert des données sur le bus des données. Ligne à logique 3 états. **R/W = 1** lecture en cours (D0 - D7 sont des entrées) **R/W = 0** écriture en cours (D0 - D7 sont des sorties) Cette ligne est validée sur le front montant de Q.

Les lignes d'état du bus (BA BS)

BA (Bus available) et **BS** (Bus state). Ces lignes informent sur l'état du microprocesseur à tout moment.

BA	BS	Etat
0	0	normal
0	1	reconnaissance d'interruption
1	0	reconnaissance de synchronisation externe
1	1	arrêt bus disponible

Selon le Tableau, nous pouvons avoir quatre états des lignes BA et BS

Premier Cas : BA BS = 0 0

Le microprocesseur est en fonctionnement normal, il gère les bus d'adresses et de données

Deuxième cas : BA BS = 0 1

Le microprocesseur est en phase de reconnaissance d'interruption pendant deux cycles. Cet état correspond à la recherche des vecteurs d'interruption : **Reset, NMI, IRQ, SW1,2 et 3.**

Troisième cas : BA BS = 1 0

Ce signal apparaît lorsque le microprocesseur rencontre l'instruction de synchronisation externe (niveau bas sur SYNC). Il attend alors cette synchronisation sur une des lignes d'interruption. Les bus sont en haute impédance pendant ce temps.

Quatrième cas : BA BS = 1 1

Le microprocesseur est à l'arrêt (niveau bas sur HALT). Le microprocesseur cède la gestion des bus des données et des adresses à un circuit annexe (contrôleur de DMA). Les bus sont en haute impédance.

1.2. Dynamique de la communication (DDE : Echange Dynamique de Données)

1.2.1. Transfert synchrone ou asynchrone des données

Nous distinguons deux types de transmissions, **asynchrone et synchrone**. Dans une transmission **asynchrone**, l'unité de données de transmission est le **caractère** alors que dans une transmission **synchrone**, les données sont transmises sous forme de *un bloc de caractères* appelées paquet ou trame.

Sans trop rentrer dans les détails nous pouvons dire que dans le cas d'une transmission asynchrone, le début de transmission est signalée par un état qui dure la durée d'un bit : Start, et à l'arrêt de la transmission le signal revient à l'état repos durant au moins un bit, un bit et demi ou deux bits : Stop.

Le passage de l'état repos au Start permet la synchronisation des horloges émettrice et réceptrice pour la réception du caractère.

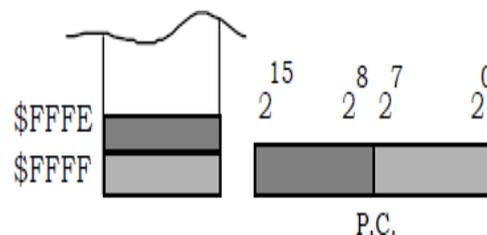
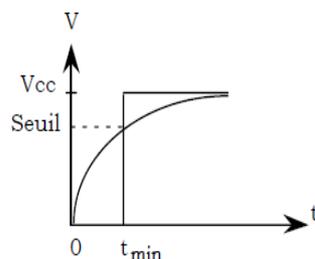
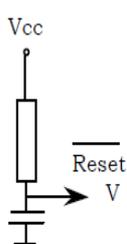
Dans la transmission synchrone, un caractère au début du bloc permet la synchronisation.

Mais les horloges peuvent se déphaser durant le bloc. Il est alors nécessaire de resynchroniser.

1.2.2. RESET du microprocesseur

Un niveau bas sur la broche d'initialisation **RESET** entraîne une réinitialisation complète du circuit, ceci entraîne les actions suivantes :

- 1 Arrêt de l'exécution de l'instruction en cours,
- 2 registre de pagination (DP) mis à zéro
- 3 interruptions **IRQ** et **FIRQ** sont masquées et l'interruption non masquable **NMI** est désarmée.
- 4 Remarque : Pour être réactivée cette ligne doit être maintenue à un niveau bas durant un temps suffisamment long (plusieurs cycles d'horloge).
- 5 Le compteur programme, PC, est initialisé avec le contenu des vecteurs d'initialisation qui se trouvent aux adresses **\$FFFE** et **\$FFFF**. Ce contenu représente l'adresse du début du programme qui sera exécuté par le microprocesseur.



1.2.3 HALT du processeur

Un niveau bas sur la broche **HALT** provoque l'arrêt du microprocesseur, mais à la différence avec le Reset à la fin de l'exécution de l'instruction en cours sans perte des données. (BA = BS = 1).

Ceci entraîne les actions suivantes :

- interruption **IRQ** et **FIRQ** sont inhibées
- Accès direct (**DMA**) à la mémoire autorisée.
- interruptions **RESET** et **NMI** prises en compte mais leur traitement est différé.

1.2.4 Les interruptions

Les interruptions et les exceptions sont deux types de rupture de séquence « exceptionnelles ». Il ne faut pas confondre les interruptions et les exceptions. Les interruptions surviennent de façon asynchrones. Elles sont commandées par le matériel (interruption externe) ou par le système (interruption système). Les interruptions n'ont aucune relation avec les instructions en cours d'exécution.

IRQ : On nomme **IRQ** (de l'anglais *Interrupt Request*, en français: *demande d'interruption*) une interruption qui est déclenchée par une ligne d'entrée-sortie matérielle d'un microprocesseur ou d'un microcontrôleur (*la mémoire, les cartes d'extension, les périphériques, etc*). Tous les registres sont sauvegardés. Le déroulement de la routine peut être interrompu par **FIRQ** ou **NMI**.

Exemples d'interruptions :

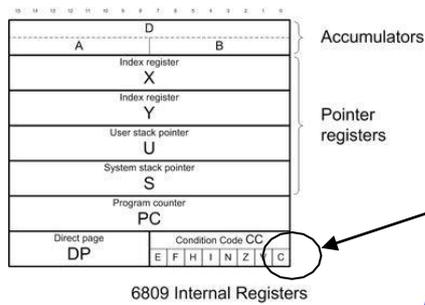
- Le clavier informe le système qu'on appuie sur une touche
- L'horloge système informe le processeur qu'il faut mettre à jour l'heure

FIRQ (*Fast Interrupt Request*) : c'est une interruption masquable ou non. Les registres du 6809 ne sont sauvegardés que partiellement.

NMI : La ligne **NMI** (*Non Masquable Interrupt*) est une ligne d'interruption non masquable. Le contenu de la totalité des registres du 6809 est sauvegardé dans la **pile** système (structure de données fondée sur le principe « *dernier arrivé, premier sorti* »).

Lors du fonctionnement de certaines parties du système d'exploitation, il peut être nécessaire de ne pas permettre les interruptions, soit parce que celles-ci perturberaient un compte serré du temps, soit parce que des structures de données sont en cours de modification (on réalise ainsi une sorte de verrou d'exclusion mutuelle dans un système mono-processeur). Aussi, on peut généralement bloquer (on dit souvent *masquer*) les interruptions. Dans la plupart des systèmes, les interruptions **bloquées** sont accumulées, c'est-à-dire qu'elles sont exécutées dès qu'elles sont démasquées

La ligne **IRQ** (Interrupt Request) a son vecteur en FFF8-FFF9. Elle est conditionnée par le bit 1 de CC



1.2.5 Les Exceptions

Il ne faut pas confondre les interruptions et les exceptions, car ce sont deux types de rupture de séquence mais pas de la même nature et pas avec les mêmes conséquences. On vu dans le paragraphe précédent que les interruptions surviennent à la demande d'un matériel (interruption externe) ou par le système (interruption système) et sont entièrement indépendantes du programme en cours d'exécution.

Cependant, les exceptions sont déclenchées par des « accidents » dans l'exécution du programme tel que : un débordement arithmétique, des erreurs de bus, une tentative d'utilisation d'instructions réservées, des erreurs d'adressage, un défaut dans la mémoire cache, un défaut de page, une instruction erronée, accès à une zone mémoire inexistante, etc.

Les exceptions provoquent un traitement spécifique : appel à une routine spécialisée. On parle de gestion précise des exceptions si l'état de la machine qui résulterait de l'exécution séquentielle de toutes les instructions antérieures à l'instruction provoquant l'exception peut être reconstruit dans tous les cas. Il est alors possible après le traitement de l'exception de reprendre l'exécution à l'instruction même ayant provoqué l'exception. Dans le cas contraire, les exceptions sont gérées de façon imprécise.

1.2.6 Protections

Il existe deux modes d'exécution pour le microprocesseur : Mode Utilisateur et Mode Superviseur

Mode superviseur

- Exécution du code du système d'exploitation
- Accès à toutes les instructions du processeur.

Lors d'une interruption, le processeur passe en mode superviseur

Mode utilisateur

- Exécution des programmes « Utilisateur »
- Certaines instructions sont interdites

1.2.5. Multiprocesseurs

Les systèmes multiprocesseurs sont moins complexes que les systèmes multicœurs, parce qu'il s'agit de processeurs uniques connectés les uns aux autres. L'inconvénient des systèmes multiprocesseurs est qu'ils sont coûteux parce qu'ils nécessitent plusieurs circuits, ce qui revient plus cher qu'une solution à circuit unique.

Avant, les systèmes à base de multiprocesseurs contenaient plusieurs processeurs qui ne se trouvaient pas forcément sur le même circuit. Ces systèmes étaient utilisés aux serveurs de communication IT. À l'heure actuelle, les multiprocesseurs se trouvent habituellement sur la même carte physique et sont connectés via une interface de communication haute vitesse.

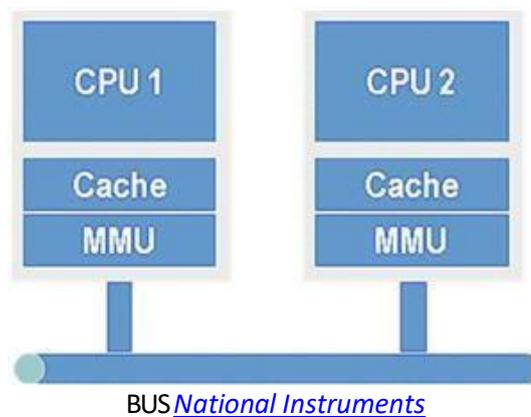


Figure n°1. Le système multiprocesseur a une mémoire cache divisée et une unité de gestion mémoire avec des interconnexions longues

Les systèmes multiprocesseurs sont l'un des deux types :

1- Multithreading

Les processus légers sont typiquement utilisés avec l'[interface graphique](#) ("GUI") d'un [programme](#). En effet, les interactions de l'utilisateur avec le processus, par l'intermédiaire des [périphériques](#) d'entrée, sont gérées par un processus léger, tandis que les calculs lourds (en termes de temps de calcul) sont gérés par un ou plusieurs autres processus légers. Cette technique de [conception de logiciel](#) est avantageuse dans ce cas, car l'utilisateur peut continuer d'interagir avec le programme même lorsque celui-ci est en train d'exécuter une tâche. Une application pratique se retrouve dans les traitements de texte où la correction orthographique est exécutée tout en permettant à l'utilisateur de continuer à entrer son texte.

L'utilisation des processus légers permet donc de rendre l'utilisation d'une application plus fluide, car il n'y a plus de blocage durant les phases de traitements intenses.

2- Hyperthreading

L'hyperthreading est une technologie qui a été introduite par Intel, avec l'objectif premier d'améliorer le support du code multithread. Dans certaines conditions de travail, la technologie de l'hyperthreading garantit une utilisation plus efficace des ressources processeurs en exécutant les threads en parallèle sur un seul et unique processeur.

Un processeur équipé en hyperthreading prétend faire office de deux processeurs "logiques" pour le système d'exploitation hôte, permettant au système d'exploitation de prévoir deux tâches (threads) ou deux processus simultanément. Les avantages de l'hyperthreading sont les suivants : un support amélioré pour le code multithread, il permet à plusieurs threads de fonctionner simultanément, et offre un temps de réaction et de réponse amélioré.

2.2.7 Processeurs double-cœurs et multicœurs

Les processeurs double-cœurs sont deux processeurs sur un seul et unique circuit. Les processeurs multicœurs constituent une famille de processeurs qui contiennent n'importe quel nombre de processeurs sur un seul et unique circuit, comme par exemple 2, 4 ou 8. Le défi avec les processeurs multicœurs se trouve dans le domaine du développement logiciel. L'accélération des performances est directement liée à la façon dont le code source d'une application a été écrit en parallèle via le multithread. La génération actuelle de microprocesseur est celle de Intel® Core™2 Duo Processor

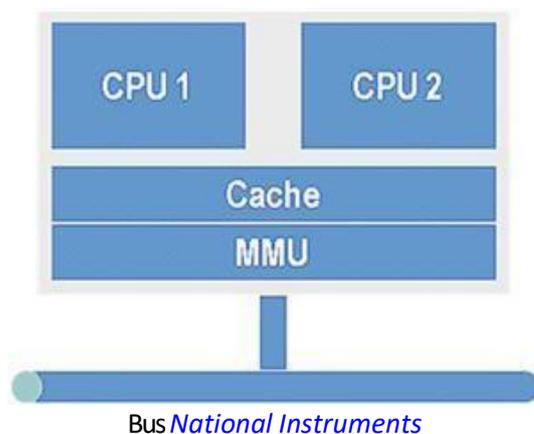


Figure n°2. Les processeurs multicœurs partagent la mémoire cache et l'unité de gestion mémoire avec des interconnexions courtes.

