



Chapitre 1: Architecture du Microprocesseur 6809

1. Présentation générale

Le 6809 est le plus évolué des microprocesseurs à 8 bits de la famille Motorola (6800,6802,6805). Il est doté d'une architecture améliorée qui dispose de plusieurs registres supplémentaires, des nouvelles instructions et des modes d'adressage supplémentaires. Il est compatible avec tous les circuits périphériques de la famille 6800 (PIA 6821, ACIA 6850, ...) et il est optimisé pour l'exécution de programmes en langages évolués (Pascal, C, Fortran, ...).

Le microprocesseur 6809 est fabriqué en technologie MOS à canal N et se présente sous la forme d'un boîtier DIL 40 broches. On peut l'utiliser avec trois horloges internes :

1 MHz pour le 6809 ; 1,5 MHz pour le 68A09 et 2 MHz pour le 68B09.

2. Architecture interne du 6809

Le microprocesseur possède 3 bus indépendants (non multiplexés) :

- Un bus de données bidirectionnel de 8 bits,
- Un bus d'adresse de 16 bits
- Un bus de contrôle de 10 bits.

Le bus de données achemine les données échangées par les différents dispositifs du système à base du microprocesseur. Pour le 6809, le bus de données est un bus bidirectionnel de 8 bits (il peut être utilisé dans les deux sens). C'est un bus à 3 états (logique à trois états), ce qui lui permet d'être utilisé pour réaliser un accès DMA (voir plus loin).

Le bus d'adresse émane du microprocesseur et transporte des adresses vers tous les dispositifs branchés sur le bus des données. C'est un bus de 16 bits, ce qui permet au 6809 d'adresser un espace mémoire de 64 K (2^{16}). Une adresse envoyée par le 6809 sert à sélectionner un dispositif ou un registre du dispositif.



Le bus de commande transporte les signaux de synchronisation entre le 6809 et tous les dispositifs branchés sur le bus de données. Ces signaux sont généralement les signaux de lecture, d'écriture, de remise à zéro, d'interruption, ...

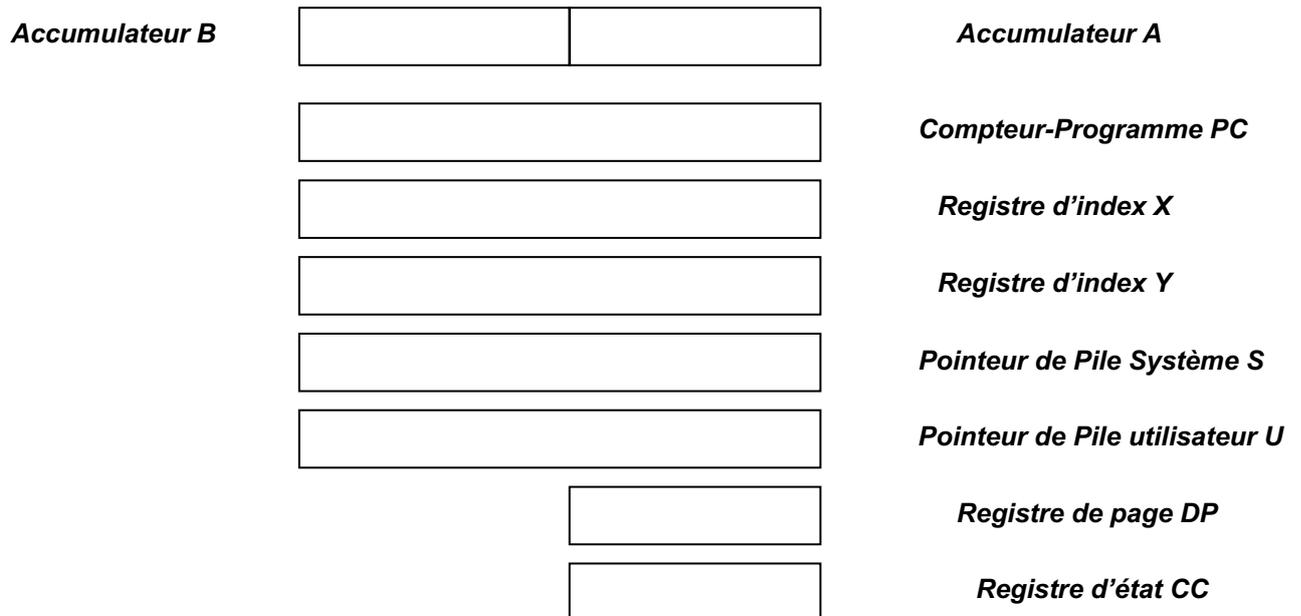
L'unité arithmétique et logique (UAL) contient tous les circuits électroniques qui réalisent effectivement les opérations désirées. Ces opérations sont principalement les opérations arithmétiques (+,-,x,...) et les opérations logiques (ET, OU , XOR, ...).

L'unité de commande s'occupe de gérer l'exécution des instructions d'un programme. Les principaux dispositifs de l'unité de commande sont :

- Le registre instruction (RI) qui reçoit l'instruction qui doit être exécutée.
- Le décodeur du code opération, qui détermine quelle opération doit être effectuée, parmi toutes les opérations possibles ;
- Le séquenceur qui génère les signaux nécessaires à l'exécution de chaque instruction d'un programme. Cette unité a besoin d'une horloge pour enchaîner les commandes.

3. Registres internes du 6809

Le microprocesseur 6809 dispose de neuf registres internes accessibles à l'utilisateur (figure n°2).



Les registres sont les suivants :

- **Accumulateurs A,B,D**

Le 6809 possède deux accumulateurs indépendants A et B de longueur 8 bits qui peuvent être associés pour constituer un seul accumulateur 16 bits nommé D. Le registre A constitue l'octet de poids fort de D, le registre B celui de poids faible.

Les accumulateurs A, B, D sont utilisés pour les calculs arithmétiques et les manipulations de données.

- **Registres d'index X et Y**

Les registres d'index X et Y sont utilisés pour les modes d'adressage indexé lors de rangement de données en mémoire sous forme de table.

- **Pointeurs de pile S et U**

Le 6809 dispose de deux pointeurs de pile S et U sur 16 bits.

Le pointeur de pile S (hardware Stack Pointer) est utilisé par le μ processeur pour gérer la sauvegarde de ses registres internes lors des appels des sous-programmes et des demandes d'interruptions. Il contient une adresse à 16 bits qui pointe le sommet de la pile. Cette dernière peut être implantée n'importe où dans l'espace mémoire RAM adressé par le μ processeur (voir plus loin).

Le pointeur de pile U (User Stack Pointer) est laissé entièrement à la disposition de l'utilisateur pour transférer ses propres arguments lors de l'appel des sous programmes.

Les pointeurs de pile S et U peuvent être utilisés comme registres d'index au même titre que X et Y.

- **Compteur programme PC**

Le compteur programme PC est un registre à 16 bits qui contient l'adresse de la prochaine instruction à exécuter (qu'il faut aller chercher en mémoire). Il s'incrémente automatiquement à chaque lecture d'un octet en mémoire à moins qu'un branchement ou un saut ne l'oblige à prendre une autre valeur particulière.



- **Registre de page DP**

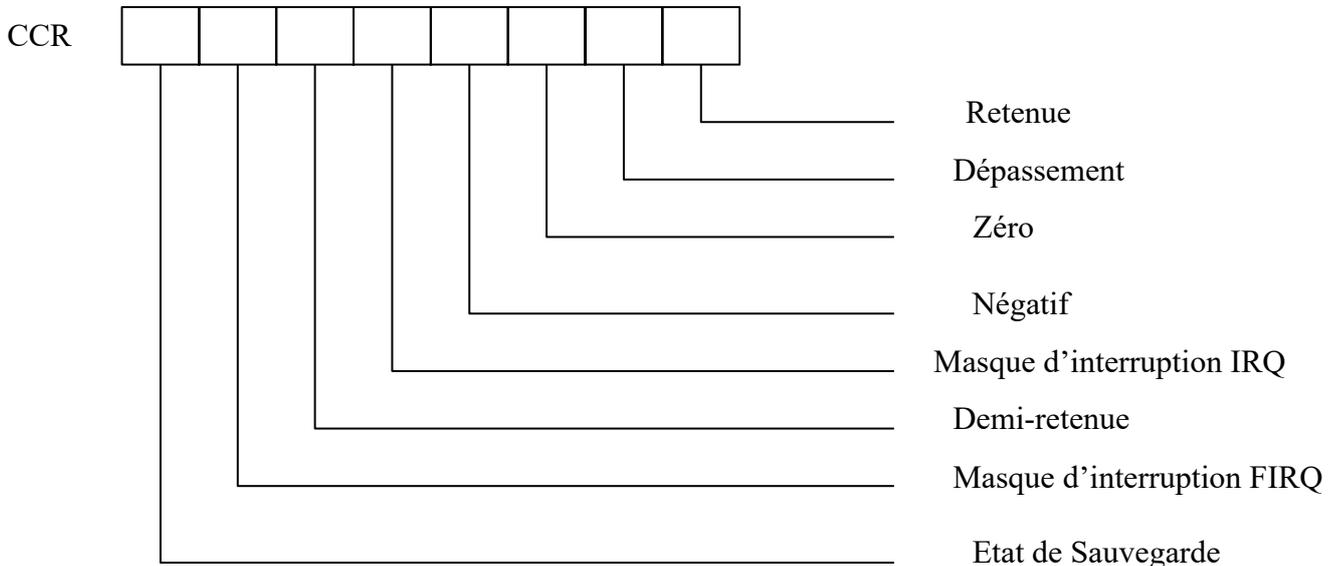
Le registre de page DP est un registre à 8 bits. Le 6809 s'en sert pour le mode d'adressage direct.

Au cours de l'exécution d'une instruction en mode d'adressage direct, le registre DP fournit l'octet de poids fort de l'adresse. Celui de poids faible suit le code opération de l'instruction. Ainsi, le programmeur peut explorer une zone mémoire de 256 octets sans changer le contenu du registre DP. Pour atteindre une autre zone, le programmeur doit redéfinir le registre DP.

Pour rester compatible avec la famille 6800, le DP est mis à 0 à l'initialisation du μ p

Registre d'état CCR (Condition Code Register)

Le registre d'état CCR est un registre à 8 bits qui contient les 8 indicateurs du μ p. Il définit après chaque opération arithmétique ou logique l'état du μ p et l'aide à prendre des décisions lors de l'exécution des instructions de branchement.



Bit C (Retenue)

L'indicateur de retenue C est positionné à 1 chaque fois qu'une retenue est générée. Voici des illustrations :

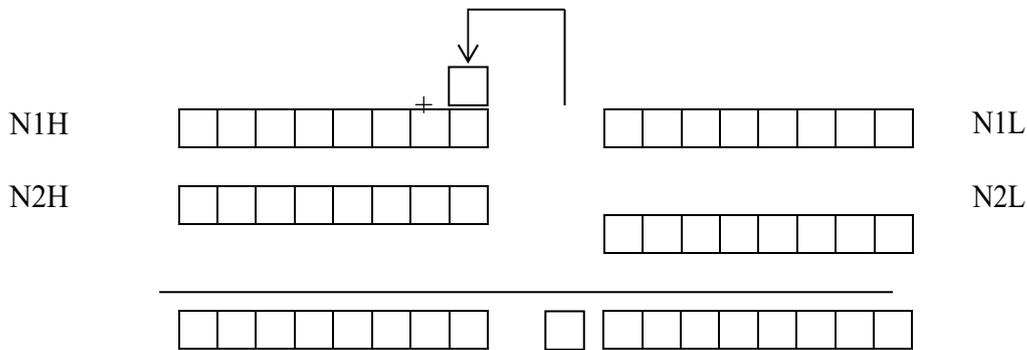
Exemple : Addition

ADDA #\$20

Addition avec retenue du contenu de l'accumulateur A et de la valeur hexadécimale \$20

L'indicateur C est utilisé pour réaliser des opérations arithmétiques sur des nombres supérieurs à 8 bits.

Exemple : soit à effectuer $N_1 + N_2$, N_1 et N_2 étant des nombres de 16 bits.



Le μp commence par additionner les deux octets de poids le plus faible de N_1 et N_2 (N_{1L} et N_{2L}), ce qui donne une somme S_L (octet) et une retenue éventuelle C . On additionne ensuite les deux octets de poids le plus faible N_{1H} et N_{2H} en tenant compte de la retenue C . Ceci est automatiquement effectué par l'instruction ADC (addition avec retenue).

Bit V (Overflow – débordement en complément à 2)

Ce bit est positionné à 1 si le résultat d'une opération arithmétique entre deux nombres représentés en complément à 2 déborde ; c'est à dire si le résultat excède l'intervalle -128 à +127.

Les instructions de chargement, de stockage et les opérations logiques positionnent V à 0.

Bit Z (Zéro)

Ce bit est positionné à 1 quand une opération quelconque produit un résultat nul. Il est mis à 0 dans le cas contraire.

Bit N (Négative)

Ce bit est significatif seulement quand le μp manipule des nombres signés. Il recopie l'état du bit de signe du résultat de l'opération effectuée. Quand le résultat est négatif, le bit N est positionné à 1 ; si le résultat est positif, N est positionné à 0.

Bit H (Demi-retendue)

Ce bit n'a de signification que quand le μp manipule des nombres représentés en DCB. Rappelons qu'en code DCB, on fait correspondre à chaque chiffre décimal son équivalent binaire codé sur 4 bits. Il est possible, avec quatre bits, de représenter tous les nombres allant de 0 à 15. Or dans le code DCB, il ne nous est pas permis de dépasser 9. Le code DCB n'utilise donc que 10 quartets parmi 16 qui servent à coder les chiffres décimaux allant de 0 à 9. Les quartets interdits sont : 1010,1011,1100,1101,1110,1111.

Si l'un des quartets interdits se manifeste lors d'une addition, il faut effectuer une correction en y ajoutant $6=(0110)_2$. Cette opération supplémentaire s'appelle ajustement décimal. Ceci est automatiquement effectué par l'instruction DAA qui utilise cet indicateur pour corriger le résultat après une addition ADDA ou ADCA.

Exemples : $6 + 5$

$$\begin{array}{r} 6 \rightarrow \quad 0110 \\ + 5 \rightarrow \quad + \underline{0101} \\ \hline \quad \quad 1010 \end{array}$$

Le résultat de l'addition 1011 est un quartet interdit. Donc, une correction est nécessaire. On ajoute 6 au résultat, ce qui donnera :

$$\begin{array}{r} \quad \quad 1011 \\ \quad + \underline{0110} \\ \hline 0001 \quad 0001 \\ \underbrace{\quad} \quad \underbrace{\quad} \\ \quad 1 \quad 1 \end{array}$$

Le résultat est bien 11.

Bit I (Interrupt Mask – Masque d'interruption)

Ce bit, lorsqu'il est à 1, masque l'interruption matérielle IRQ. On traitera en détail les interruptions (Voir Chap. : Les interruptions).

Bit F (Fast Interrupt Mask – Masque d'interruption rapide)

Ce bit, lorsqu'il est à 1, masque l'interruption matérielle FIRQ. Les deux bits I et F peuvent être mis en place par le programmeur à l'aide des instructions de type ANDCC ou ORCC (Voir Chap. : Le Jeu d'Instructions). Ainsi, le μp ne tient pas compte des demandes d'interruption provenant de la ligne correspondante (ou les deux).



Bit E (sauvegarde des registres internes dans la pile)

Quand une demande d'interruption IRQ a lieu, le bit E se positionne à 1. Dans ce cas, tous les registres internes sont sauvegardés dans la pile.

Quand il s'agit de la demande d'interruption FIRQ, le bit E se positionne à 0. Dans ce cas, seuls les registres PC et CCR sont sauvegardés dans la pile.

Ainsi, l'indicateur E différencie entre les deux mode d'interruption et indique au μp le nombre de registres à dépiler lorsque ce dernier a fini d'exécuter le programme d'interrupti