

3/4.2.1

Le microprocesseur 6809 de Motorola

Fabricants : Motorola, Fairchild, Gould AMI, Hitachi, Thomson-CSF.

3/4.2.1.1

Généralités

Le microprocesseur 6809 de Motorola est un processeur 8 bits de haute gamme, dont l'organisation interne est orientée 16 bits.

Il permet un adressage mémoire de 64 kiloOctets grâce à ses 16 lignes d'adresse.

Il existe deux versions de ce processeur, le 6809 et le 6809E. Leur brochage diffère ainsi que leurs signaux mais leur jeu d'instruction est identique.

Il possède deux accumulateurs A et B de 8 bits qui peuvent être concaténés en un accumulateur D de 16 bits, deux registres X et Y dits d'index (16 bits), deux registres U et S pour la gestion des piles utilisateur et système, un pointeur de page DP (8 bits) servant à l'adressage direct de la mémoire et un registre d'état CC (8 bits). Le rôle de tous ces registres sera présenté plus loin.

Il possède 9 modes d'adressage se complétant et permettant ainsi au programmeur d'optimiser ses routines personnelles.

- L'adressage inhérent
- L'adressage immédiat
- L'adressage direct
- L'adressage étendu
- L'adressage étendu indirect

4.2 Les circuits de la famille des microprocesseurs de la série 68XX

- L'adressage relatif court
- L'adressage relatif long
- L'adressage indexé
- L'adressage indexé indirect.

3/4.2.1.2

Organisation du 6809

Le microprocesseur, fabriqué en technologie MOS Canal N. comporte 40 broches et est du type mono-tension (5 V).

Il a un cycle d'horloge interne de 2 MHz (Fréquence externe divisée par 4) ; il possède trois bus indépendants :

- bus de données sur 8 bits
- bus d'adresses sur 16 bits
- bus de contrôle de 10 bits pour le 6809 et de 12 bits pour le 6809E.

Description des signaux

Le bus de donnée (D0-D7) est totalement bidirectionnel et chaque sortie peut piloter l'équivalent de 8 charges TTL. Ces sorties sont du type trois états et l'inversion du mode entre/sortie est assurée à l'intérieur d'un seul cycle.

Le bus d'adresse (A0-A15) est unidirectionnel, peut piloter 8 charges TTL et est en logique trois états.

Le bus de contrôle comporte des signaux communs aux deux versions de 6809 ; ce sont les suivants :

Le signal lecture-écriture R/\bar{W} (read/write) détermine le sens du transfert des données. Lorsque $R/\bar{W} = 1$, le 6809 fait une lecture ; les données entrent. Lorsque $R/\bar{W} = 0$, le 6809 fait une écriture ; les données sortent du 6809.

Les lignes d'état du bus BA (bus available) et BS (bus state) renseignent les périphériques du 6809 sur la disponibilité des bus données et adresses dans le cas d'applications du type accès direct mémoire (DMA - Direct Memory Acces).

4.2 Les circuits de la famille des microprocesseurs de la série 68XX

Signaux de contrôle du 6809E

Les sorties E et Q sont des horloges destinées aux périphériques du 6809. Elles sont en quadrature de phase.

L'entrée mémoire prête ($\overline{\text{MRDY}}$ — Memory ReaDY) permet l'allongement d'un cycle d'horloge E par fraction de quart de cycle jusqu'à $10\mu\text{s}$. Cela autorise ainsi les mémoires à temps d'accès long.

L'entrée accès direct mémoire ($\overline{\text{DMA/BREQ}}$) suspend le fonctionnement du 6809, permettant ainsi la libération des bus pour des périphériques intelligents ou un rafraîchissement de mémoires dynamiques.

Les broches Ein et Qin sont les entrées d'horloge de 6809E. Ces deux signaux doivent être en quadrature.

La broche de contrôle des trois états bus (TSC - Three State Control) joue le même rôle que $\overline{\text{DMA/BREQ}}$ pour le 6809 et offre les mêmes possibilités.

Le signal d'indication de dernier cycle (LIC - Last Instruction Cycle) est à l'état haut ($\text{LIC} = 1$) à chaque fois que le microprocesseur exécute le dernier cycle machine d'une instruction. Donc lorsqu'il revient à l'état bas, le 6809E effectue une lecture du premier code opératoire de l'instruction suivante.

Le signal de prévention d'accès mémoire (AVMA - Advanced Valid Memory Address) nous prévient en passant à l'état haut au cours du cycle d'un accès bus par le 6809E. En mode HALT, SYNC ou en calcul interne, AVMA est au niveau bas. Cela permet d'optimiser l'échange de ressources communes dans un dispositif multiprocesseur.

La broche BUSY a le même fonctionnement que AVMA mais elle ne passe à l'état haut que pendant l'exécution d'une lecture-écriture d'une donnée numérique.

Conclusion

Les caractéristiques matérielles du 6809 le rendent entièrement compatible avec tous les périphériques de la famille 68XX.

On a pu noter la spécialisation du 6809E dans les applications à plusieurs processeurs.

4.2 Les circuits de la famille des microprocesseurs de la série 68XX

- $BA = BS = 0$: Le 6809 est en fonctionnement, il gère les bus données et adresses.
- $BA = 0 BS = 1$: Le 6809 vient de recevoir une interruption, il recherche le vecteur correspondant.
- $BA = 1 BS = 0$: Le 6809 vient de rencontrer dans le programme l'instruction SYNC. Il attend cette synchronisation sur une ligne d'interruption avec ses bus à haute impédance.
- $BA = BS = 1$: Le 6809 a été arrêté par le signal \overline{HALT} ; les bus sont à haute impédance.

La ligne \overline{HALT} permet d'interrompre de façon matérielle le déroulement d'un programme. Le 6809 termine l'instruction en cours puis positionne BA et BS à 1. Le déroulement reprend dès que la broche \overline{HALT} est à 1 et sans perte d'informations. Les lignes d'interruptions \overline{IRQ} et \overline{FIRQ} sont dévalidées : \overline{RESET} et \overline{NMI} sont valides mais leur traitement ne se fera qu'à la libération du 6809. Les horloges ne sont pas affectées.

La ligne \overline{RESET} permet la réinitialisation complète du 6809. Le travail en cours est totalement perdu. Les IT sont masquées, \overline{NMI} est dévalidée, le 6809 se relance en lecture du vecteur \overline{RESET} aux adresses FFFE-FFFF, il charge ce vecteur dans le compteur programme et se replace en exécution.

La ligne \overline{NMI} (Non Masquable Interrupt) est une ligne d'interruption non masquable. Elle permet d'exécuter une routine d'interruption dont l'adresse est contenue dans le vecteur FFFC-FFFD. Cette ligne est dévalidée par \overline{RESET} et n'est revalidée qu'après un chargement du registre S en mode immédiat. Le contenu de la totalité des registres du 6809 est sauvegardé dans la pile système.

La ligne d'interruption rapide (\overline{FIRQ} : Fast Interrupt Request) a son vecteur situé en FFFA-FFFB, elle est masquée ou non suivant l'état du bit F du registre CC . Les registres du 6809 ne sont sauvegardés que partiellement.

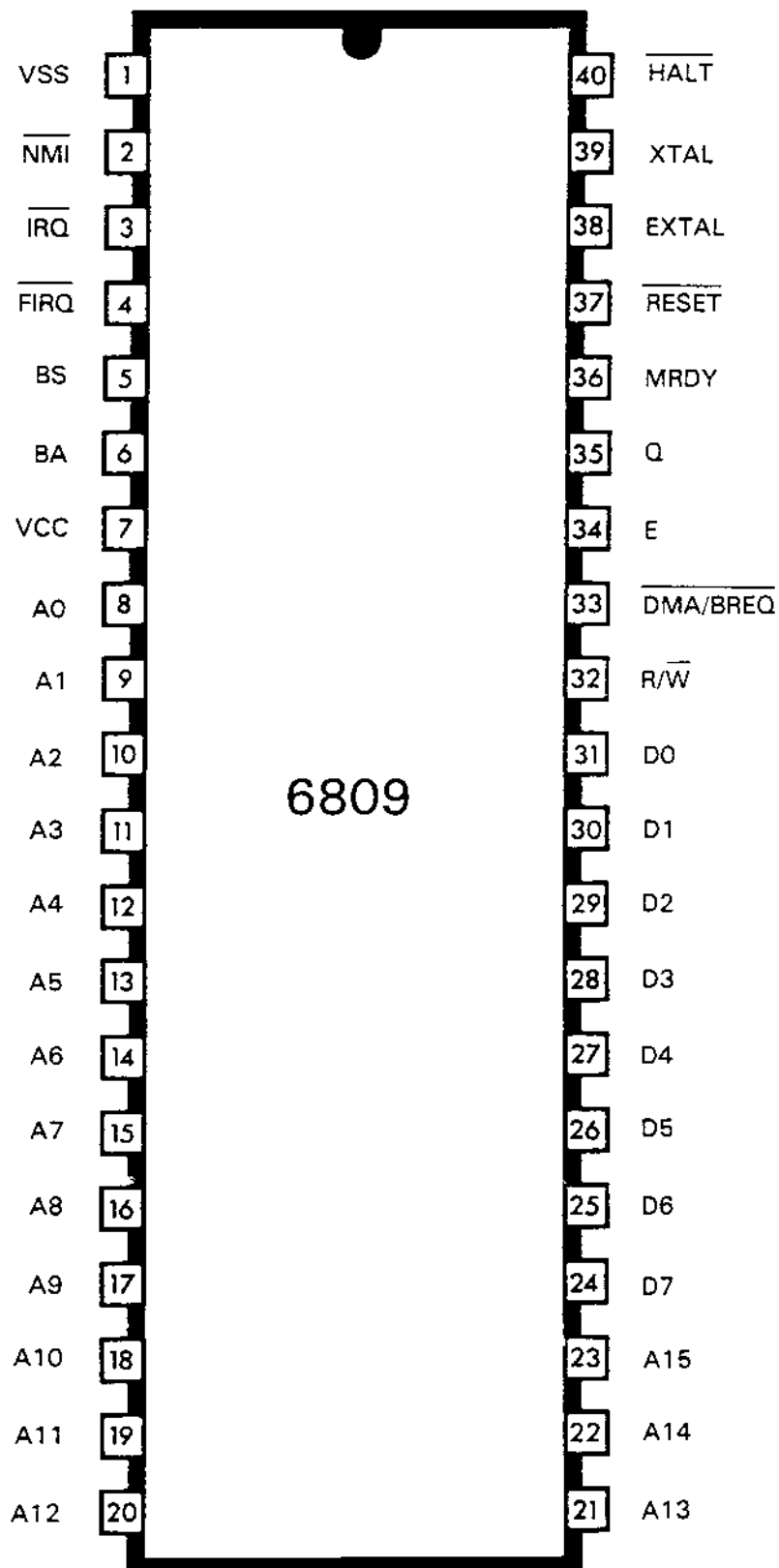
La ligne \overline{IRQ} (Interrupt Request) a son vecteur en FFF8-FFF9 ; elle est conditionnée par le bit 1 de CC et tous les registres sont sauvegardés. Le déroulement de la routine peut être interrompu par \overline{FIRQ} ou \overline{NMI} .

Signaux de contrôle du 6809

Les entrées d'horloge $XTAL$ et $EXTAL$ permettent deux fonctionnements différents. On connecte un quartz en parallèle sur ces broches (osc. interne) ou on injecte un signal TTL d'horloge sur $EXTAL$ en reliant $XTAL$ au OV .

4.2 Les circuits de la famille des microprocesseurs de la série 68XX

Brochage du 6809



4.2 Les circuits de la famille des microprocesseurs de la série 68XX

3/4.2.1.3

Architecture interne du 6809

Les registres

Le 6809 comporte 9 registres 8 ou 16 bits que nous allons décrire.

Les accumulateurs A et B du 6809 sont des registres de calcul de 8 bits dans lesquels toutes les opérations arithmétiques et logiques peuvent s'effectuer. Ils sont pour cela entièrement identiques et mis à part les instructions ABX et DAA, ils jouent exactement le même rôle sur 8 bits.

Le double accumulateur D est en fait la concaténation de A et B, le registre A représentant les poids forts et B les poids faibles. D peut donc être le résultat d'une opération sur A ou sur B avec B ou A constituant une partie constante.

Les registres X et Y sont dits registres d'index car ils permettent d'adresser tout l'espace mémoire avec en plus la capacité d'être pré-décrémenté ou post-incrémenté pour faciliter le traitement de variables en tables.

Les registres U et S sont des pointeurs de pile qui fonctionnent tous deux identiquement. Ils opèrent en mode dernier entré-premier sorti (LIFO - Last In - First out) négatif. Ces deux registres peuvent à l'occasion servir de registre d'index avec la totalité des possibilités de X et Y. Le registre S (System) est utilisé par le 6809 pour toutes les opérations de sauvegarde en cas d'interruption ou de saut à un sous-programme (Adresse de retour). Le registre U (User) est entièrement réservé à l'utilisateur.

Le registre PC (Program Counter) détermine l'adresse mémoire à laquelle le 6809 doit exécuter une instruction. Il peut dans certains cas être — avec grande précaution — utilisé comme registre d'index.

Le registre de page DP (Direct Page) forme la partie haute de l'adresse à pointer dans le cas d'un adressage direct. Il est automatiquement remis à 00 par un RESET.

Le registre CC (Condition Codes) définit à tout instant l'état du 6809 résultant d'une instruction. Il comporte 8 bits jouant chacun un rôle important pour les instructions de sauts ou de branchements conditionnels.

Rôle des indicateurs du registre CC

Le bit C (Carry - BO) est positionné lors d'une opération arithmétique uniquement. Il est à UN dans le cas d'une addition dont le

résultat est supérieur à 255 (\$ FF) ou lorsque le résultat d'une soustraction (SUB, NEG, CMP, SBC) est positif. Dans le cas de l'instruction MUL (A multiplié par B, résultat sur 16 bits dans D), CARRY est égal au bit B7 de D.

Le bit V (oVerflow - B1) est le résultat d'un OU-EXCLUSIF entre les retenues arithmétiques de B6 et de B7. Les opérations logiques de chargements, de stockages et de transferts positionnent V à 0. Seules les opérations comme ADD, ADC, SUB, SBC, NEG et CMP positionnent V à la valeur appropriée.

Le bit Z (Zéro - B2) est l'indicateur d'un résultat nul. Toutes les instructions positionnent ce bit.

Le bit N (NEGATIVE - B3) indique un résultat négatif. Pour toutes les instructions, N prend la valeur du bit de poids fort de l'opérande ou de l'accumulateur en mouvement. Il est très utile lors de travaux signés sur des entiers compris entre + 127 et - 128.

Le bit I (Interrupt - B4) autorise lorsque il est à 0 le traitement des interruptions IRQ et les dévalide quand il est à 1. Ce bit est uniquement manipulé par l'utilisateur mais est positionné à 1 par un $\overline{\text{RESET}}$

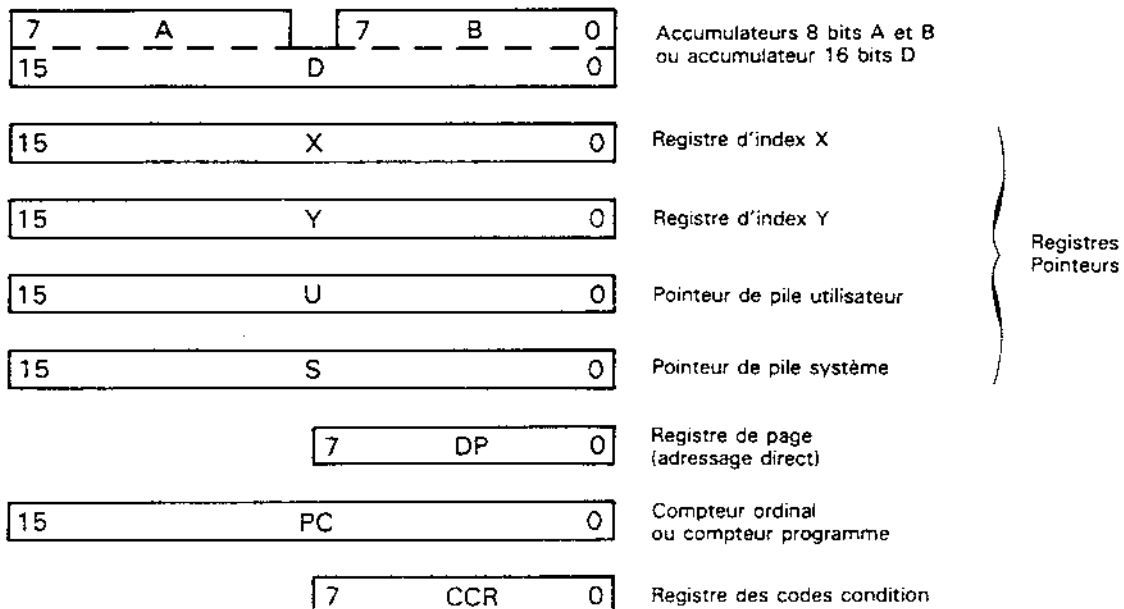
Le bit H (Half-carry - B5) est un bit de demi-retenue entre le quartet faible et le quartet fort de A ou B. Il n'y a pas d'instruction de branchement conditionnel testant ce bit mais il faut savoir que l'on peut le tester en utilisant TFR CC, A pour ne pas modifier CC et tester A après ou en utilisant directement ANDCC \$20, ce qui transfère H dans Z.

Le bit F (Fast interrupt) - B6) conditionne le traitement de la ligne $\overline{\text{FIRQ}}$. Il est à 0 après un $\overline{\text{RESET}}$ et dans ce cas $\overline{\text{FIRQ}}$ est dévalidée. Si l'utilisateur le force à 1, $\overline{\text{FIRQ}}$ est traitée.

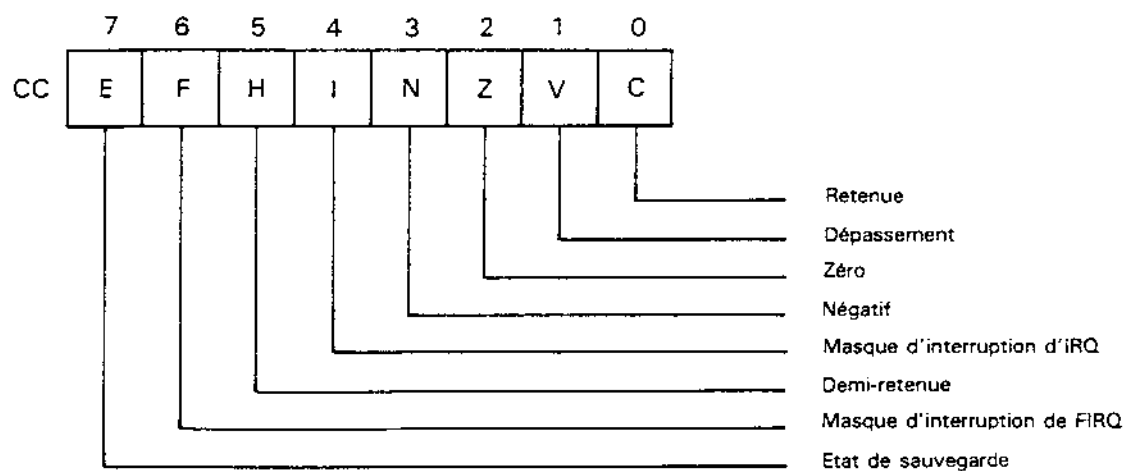
Le bit E (B7) est l'indicateur de sauvegarde partielle lié à la ligne d'interruption $\overline{\text{FIRQ}}$. Lorsque il est à 0 après un RESET seulement, une partie des registres internes sont sauvegardés pendant le traitement de l'interruption. Lorsque il est à 1, tous les registres sont sauvegardés dans la pile système.

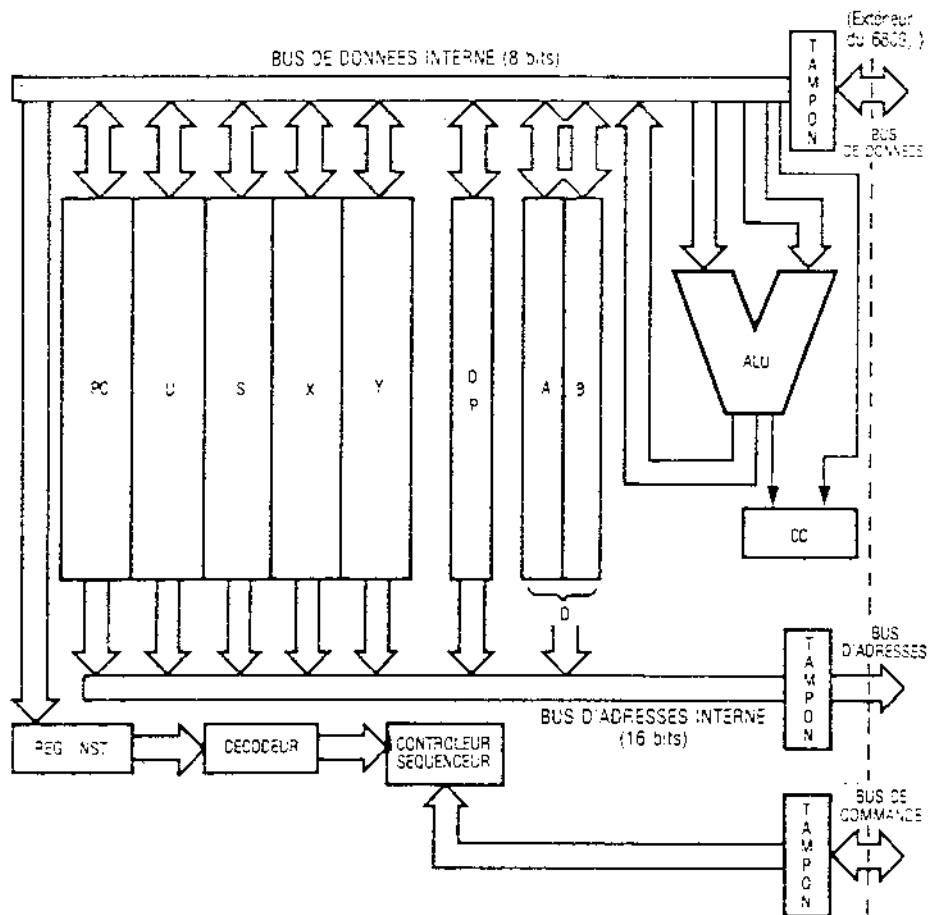
4.2 Les circuits de la famille des microprocesseurs de la série 68XX

Les registres du 6809

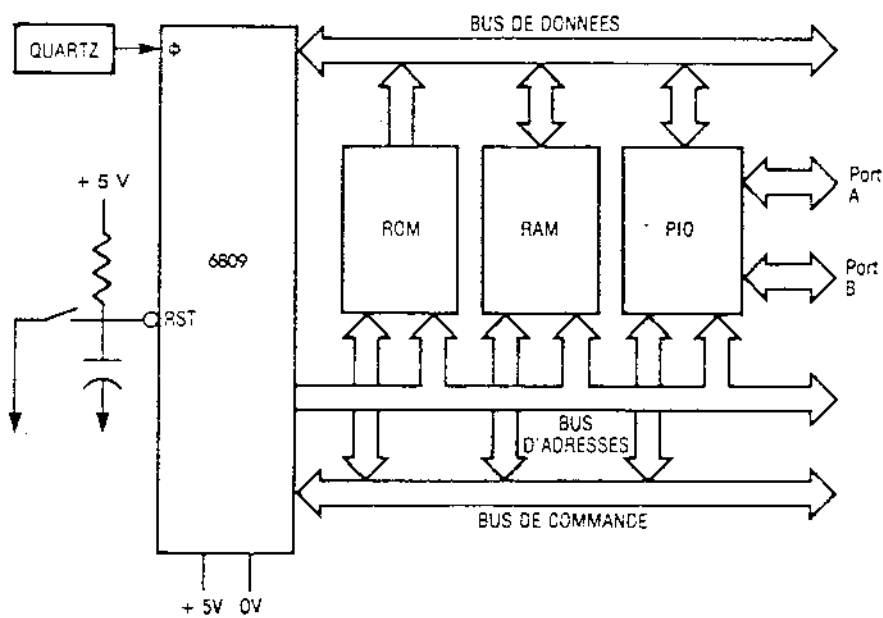


Le registre des codes condition CC





Organisation interne du 6809



Système 6809 standard

3/4.2.1.4

Modes d'adressage du 6809

Le 6809 possède 9 modes d'adressage qui, utilisés en conjonction avec ses 59 instructions de base, procurent 1464 instructions possibles.

L'adressage inhérent

Le code opération contient toute l'information nécessaire à l'exécution de l'instruction. Ces instructions sont codées

— soit sur un seul octet.

Ex : ABX ASLA RORA NEGA COMA DECA...

— soit sur deux octets.

Ex : SWI2 TFR PSHS PULU CWAY...

L'adressage immédiat

Le code opératoire est directement suivi par un opérande de un ou deux octets. Cet opérande est une valeur qui va être chargée dans, comparée à ou additionnée à un registre de 6809.

Ex : LDA 25 → Chargement de A avec 25 (décimal)

CMPX # \$ 4A03 → Comparaison de X avec 4A03 (Hexa)

ADDD # %0 010010 → Addition de 010010 (Binaire) à D

Ces instructions peuvent atteindre quatre octets comme LDS ou CMPU.

On retrouvera toujours le signe « # » dans le langage d'assemblage qui notifie le mode immédiat.

L'adressage direct

Ce mode de travail permet en collaboration avec le registre DP de charger un accumulateur avec le contenu de n'importe quelle adresse mémoire. De plus, il n'occupe que deux octets sauf pour les instructions liées à Y et S.

Si DP a été préalablement chargé avec \$ 24 :

— LDA < \$ 00 charge A avec le contenu de l'adresse \$ 2400

— CMPX < \$ 56 compare X avec le contenu des adresses \$ 2456 - \$ 2457.

Le symbole « < » précise le mode direct en langage d'assemblage.

Si l'utilisateur veut travailler sur une autre page, il lui suffit d'écrire :

— LDA # \$ 18 → numéro de page \$ 18 dans A

— TFR A.DP → transfert de A dans DP.

4.2 Les circuits de la famille des microprocesseurs de la série 68XX

Adressage étendu

Cela prend quatre octets qui permettront d'en gagner beaucoup plus dans le cas d'un travail important dans la page \$ 1800-\$ 18 FF.

Ce mode permet d'atteindre toute la mémoire mais avec un opérateur à deux octets. Cet opérateur représente l'adresse mémoire dont le contenu va être manipulé.

Ex : LDA \$ 1000 → charge A avec le contenu de l'adresse \$ 1000
SBCB \$ 29 D0 → soustrait à B le contenu de l'adresse \$ 29 D0 et le contenu du bit C de CC.

CMPX \$ 4800 → compare X au contenu de \$ 4800-\$ 4801.

Il n'y a pas de symbole d'assemblage pour l'adressage étendu.

Adressage étendu indirect

C'est ici le contenu de l'adresse citée comme opérateur qui va indiquer l'adresse dans laquelle se trouve le nombre à manipuler.

Pour mieux illustrer :

Supposons qu'à l'adresse \$ 1000 de la mémoire il y ait \$ 23 et qu'à l'adresse \$ 1001 il y ait \$ D9.

LDA [\$ 1000] → CHARGE A avec le contenu de l'adresse \$ 23D9...

Cela est très efficace lorsqu'une routine unique doit traiter différentes valeurs. La routine n'a pas besoin d'être doublée, ce sont les « points de repère » qui bougent.

On y trouve aussi une grande utilité dans l'emploi de tables d'adressage. Les signes « [] » sont la notation de ce mode de travail.

Adressage relatif court

Ce mode d'adressage est employé pour les instructions de branchement. Il permet de « brancher » le programme à une instruction se trouvant à + 127 ou - 128 octets de distance.

Adr.	Code	Mnémonique	
\$ 0FFC	86 40	LDA # \$ 40	
\$ 0FFE	20 XX	BRA \$ 1010	Branchement à l'adresse \$ 1010 la valeur de XX se calcule à partir du 1 ^{er} octet suivant
\$ 1000	
\$ 1010	7E 0000	LDX # 0	

On soustrait l'adresse qui suit l'opérateur de l'instruction de branchement de l'adresse de destination.

$$\$ 1010 - \$ 1000 = \$ 0010$$

On obtient la valeur de l'octet XX = \$ 10 qui sera ajoutée au compteur programme lors de l'exécution.

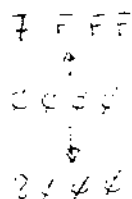
8 8 7 F (+127)

8 8 8 8

F F 8 8 (-128)

4.2 Les circuits de la famille des microprocesseurs de la série 68XX

Adressage relatif long



Il fonctionne suivant le même principe que le précédent mais le déplacement est codé sur deux octets.

Il permet d'atteindre toute la mémoire par un déplacement de + 32767 à - 32768.

Il occupe quatre octets mais reste très pratique pour la mise en œuvre de routines devant pouvoir s'implanter n'importe où dans la mémoire.

Ex : LBEQ \$ 2000 → Branche le déroulement à l'adresse \$ 2000 si le bit Z de CC est positionné.

Le « L » devant l'instruction BEQ signale en langage d'assemblage le mode relatif long.

L'adressage indexé

Dans ce mode d'adressage, un registre d'index de 16 bits spécifie une base à laquelle on ajoute un déplacement signé de 5, 8 ou 16 bits.

Ce déplacement peut être défini par une constante faisant partie de l'instruction ou par le contenu d'un des autres registres du 6809.

Enfin ce mode permet aussi la pré-décrémentation et la post-incrémentation simple ou double. Toutes ces possibilités sont précisées par le post-octet qui suit le code opératoire.

Adressage indexé déplacement nul

C'est le fonctionnement le plus simple de ce mode d'adressage ; le registre d'index contient l'adresse effective de l'octet à manipuler.

Ex : LDA,X → charge A avec le contenu de l'adresse pointée par le valeur de X.

Adressage indexé avec auto-incrémentation décrémentation

Dans ce type de travail le contenu du registre d'index est décrémenté avant de pointer l'adresse effective ou incrémenté après l'avoir fait.

Ex : Si X contient \$ 1000

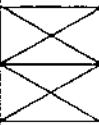
LDA,X + → Charge A avec le contenu de l'adresse \$ 1000, mais X contiendra \$ 1001 après l'instruction terminée.

LDB , -X → Commence par décrémenter X. donc $X = \$ 0FFF$ et charge B avec le contenu de \$ 0FFF.


LDD ,X++ → Charge D avec les contenus de \$ 1000 - \$ 1001 et incrémente deux fois X après. donc $X = \$ 1002$.

4.2 Les circuits de la famille des microprocesseurs de la série 68XX

Signification des bits du registre post-octet

Bit du registre post-octet								Mode d'adressage indexé
7	6	5	4	3	2	1	0	AE = , Base (R) \pm Déplacement
0	R		DÉPLACEMENT					AE = , R \pm 4 bits
1	R		0	0	0	0	0	AE = , R +
1	R		1	0	0	0	1	AE = , R ++
1	R		0	0	0	1	0	AE = , - R
1	R		1	0	0	1	1	AE = , -- R
1	R		1	0	1	0	0	AE = , R \pm 0
1	R		1	0	1	0	1	AE = , R \pm Acc B
1	R		1	0	1	1	0	AE = , R \pm Acc A
1	R		1	1	0	0	0	AE = , R \pm 8 bits
1	R		1	1	0	0	1	AE = , R \pm 16 bits
1	R		1	1	0	1	1	AE = , R \pm D (Acc A + Acc B)
1			1	1	1	0	0	AE = PC \pm 7 bits
1			1	1	1	0	1	AE = PC \pm 15 bits
1	R		1	1	1	1	1	AE = , Adresse

REMARQUE : Les bits 5 et 6 du post-octet permettent de définir la base.

Base R	b6	b5
Index X	0	0
Index Y	0	1
Pointeur U	1	0
Pointeur S	1	1
Compteur Programme		

→ Indifférent, la sélection de la base PC se fait à l'aide des bits 2 et 3 (1, 1).

2^e Complément

4.2 Les circuits de la famille des microprocesseurs de la série 68XX

	<p>Le travail en auto-incrémentation/décrémentation simple facilite le traitement de tables de données sur 8 bits et le double de tables sur 16 bits.</p>
Adressage indexé à déplacement constant	<p>Le post-octet définit la puissance du déplacement par le positionnement de son bit B7.</p> <p>B7 = 0 : Le déplacement est codé sur 5 bits (B0 – B4) et sera additionné en valeur algébrique au registre d'index avant le pointage de l'adresse effective.</p> <p>B7 = 1 : Le déplacement est stipulé par un ou deux octets qui suivront le post-octet.</p> <p>Dans le cas d'un déplacement sur 8 bits, les bits B0-B3 du post-octet vaudront % 1000 et dans le cas d'un déplacement sur 16 bits ils vaudront % 1001.</p> <p>Ex : LDA 4,X → Charge A avec le contenu de l'adresse X + 4 CMPD -5,U → Compare D avec le contenu des adresses U-5, U-4.</p> <p>Les bits B5 et B6 du post-octet désignent le registre devant servir d'index.</p>
Adressage indexé déplacement accumulateur	<p>Dans ce cas c'est le contenu des accumulateurs A, B ou D qui est additionné au registre d'index avant de pointer l'adresse effective. Le bit B7 du post-octet est à 1, les bits B5-B6 précisent l'index et les bits B0-B3 précisent l'accumulateur concerné :</p> <ul style="list-style-type: none"> — %1010 ACC A — % 1001 ACC B — % 1011 ACC D <p>Ex : LDA A,Y → Additionne X et A pour obtenir l'adresse effective où l'on doit trouver le nouveau contenu de A.</p>
Adressage indexé avec PC comme base	<p>Ce mode permet seulement un adressage indexé à déplacement constant sur 8 ou 16 bits.</p> <p>Les bits B5-B6 du post-octet n'ont plus de sens, ce sont les bits B2-B3 qui forcent ce mode de travail :</p> <p>B2 = B3 = 1 → Index = PC</p> <p>Le déplacement est précisé par un ou deux octets qui suivent le post-octet.</p>
Adressage indexé indirect	<p>Ce mode d'adressage est une combinaison du mode indexé et du mode indirect.</p>

4.2 Les circuits de la famille
des microprocesseurs de la série 68XX

L'adressage index

Déplacement																							
L'adressage index		Nul				5 bits		8 bits		16 bits		Accu. A		Accu. B		Accu. D (A : B)		Pré-décrémentation		Post-incrémentation			
		S		I		S		I		S		I		S		I		S		I			
		S		I		S		I		S		I		S		I		S		I			
S = Adressage indexé simple uniquement I = Adressage indexé simple ou indirect		Index X		Index Y		Pointeur S		Pointeur U		Compteur programme PC													
		I	S	I	S	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	
		S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	
		I	S	I	S	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	
		S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	
		I	S	I	S	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	
		S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	
		I	S	I	S	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	
		S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	
		I	S	I	S	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	
		S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	
		I	S	I	S	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	
		S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	
		I	S	I	S	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	
		S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	
		I	S	I	S	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	
		S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	
		I	S	I	S	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	
		S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	
		I	S	I	S	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	
		S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	
		I	S	I	S	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	
		S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	
		I	S	I	S	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	
		S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	
		I	S	I	S	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	
		S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	
		I	S	I	S	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	
		S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	
		I	S	I	S	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	
		S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	
		I	S	I	S	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	
		S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	
		I	S	I	S	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	
		S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	
		I	S	I	S	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	
		S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	
		I	S	I	S	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	
		S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	
		I	S	I	S	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	
		S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	
		I	S	I	S	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	
		S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	
		I	S	I	S	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	
		S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	
		I	S	I	S	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	
		S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	
		I	S	I	S	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	
		S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	
		I	S	I	S	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	
		S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	
		I	S	I	S	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	
		S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	
		I	S	I	S	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	
		S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	
		I	S	I	S	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	
		S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	
		I	S	I	S	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	
		S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	
		I	S	I	S	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	
		S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	
		I	S	I	S	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	
		S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	
		I	S	I	S	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	
		S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	
		I	S	I	S	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	
		S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	
		I	S	I	S	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	
		S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	
		I	S	I	S	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	
		S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	
		I	S	I	S	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	
		S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	
		I	S	I	S	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	
		S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	
		I	S	I	S	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	
		S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	
		I	S	I	S	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	
		S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	
		I	S	I	S	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	
		S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	
		I	S	I	S	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	
		S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	
		I	S	I	S	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	
		S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	
		I	S	I	S	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	
		S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	
		I	S	I	S	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	
		S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	
		I	S	I	S	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	
		S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	
		I	S	I	S	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	
		S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	
		I	S	I	S	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	
		S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	
		I	S	I	S	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	
		S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	
		I	S	I	S	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	
		S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	
		I	S	I	S	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	
		S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	
		I	S	I	S	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	
		S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	I	S	
		I	S	I	S	I	I	I	I	I	I	I	I	I	I	I	I</						

4.2 Les circuits de la famille
des microprocesseurs de la série 68XX

C'est le contenu de l'adresse pointée par l'index qui va servir d'adresse effective.

Ex : LDB [- 6.S] → Va chercher à l'adresse S-6, S-5 l'adresse effective à partir de laquelle on va charger B.

La notation « [] » précise le mode indirect et l'indexation offre les mêmes possibilités que le mode indexé simple.

Conclusion

La multitude des combinaisons entre les 9 modes d'adressages et les 59 instructions de base du 6809, fait de ce microprocesseur le meilleur des microprocesseurs de 8 bits sur le marché.

3/4.2.1.5

Le jeu d'instruction du 6809

Bien que le nombre de mnémoniques soit limité, le jeu d'instruction du 6809 est très performant. Le rôle de l'instruction, son effet sur les indicateurs, les modes d'adressage utilisables et les différentes formes du mnémonique vont être décrits dans cette partie. Les instructions seront abordées par ordre alphabétique.

ABX — Addition non signée de B dans X

Mode d'adr →	INHERENT		
	Code objet	Nbre cycles	Nbre octets
ABX	3A	3	1

$X \leftarrow (X) + (B)$; Ajoute le contenu de B au registre X,
Résultat dans X.

	H		N		Z	V	C
CCR	x	x	x	x	x	x	x

ADC — Addition d'un octet mémoire avec un accumulateur et avec retenue.

	IMMEDIAT			DIRECT			ETENDU			INDEXE		
	Co	Cy	Oc	Co	Cy	Oc	Co	Cy	Oc	Co	Cy	Oc
ADCA	89	2	2	99	4	2	B9	5	3	A9	4	2
ADCB	C9	2	2	D9	4	2	F9	5	3	E9	4	2

$R \leftarrow (R) + \text{Donnée} + (C)$; ajoute à l'accumulateur la donnée puis le contenu du bit de retenue Carry de CC

	H		N		Z	V	C
CCR	x	x	H	x	N	Z	V

ADD — Addition d'une donnée à un accumulateur

4.2 Les circuits de la famille des microprocesseurs de la série 68XX

	IMMEDIAT			DIRECT			ETENDU			INDEXE		
	Co	Cy	Oc	Co	Cy	Oc	Co	Cy	Oc	Co	Cy	Oc
ADDA	8B	2	2	9B	4	2	BB	5	3	AB	4	2
ADDB	CB	2	2	DB	4	2	FB	5	3	EB	4	2
ADDD	C3	4	3	D3	6	2	F3	7	4	E3	6	2

$R \leftarrow (R) \div (M)$; ajoute le contenu de l'adresse mémoire à l'accumulateur, résultat dans l'acc.

$$D \leftarrow (A:B) \div (M:M+1) ; \text{ Idem sur 16 bits}$$

				H	N	Z	V	C
ADDA/ADDB	x	x	H	x	N	Z	V	C

					H		N	Z	V	C
ADDD	x	x	x	x	N		Z	V	C	

AND — Et logique entre la mémoire et un registre.

[illegible]

R ← (R). Donnée ; effectue un ET logique, le résultat est dans le registre.

				H	N	Z	V	C
ANDA/ANDB	x	x	x	x	N	Z	O	x

				H		N	Z	V	C
ANDCC	x	x	H	x	N	Z	V	C	

ASL— Décalage arithmétique vers la gauche.

[illegible]

4.2 Les circuits de la famille
des microprocesseurs de la série 68XX

6809

$C \leftarrow b7$; Décale le contenu de l'octet de un bit vers la
 $b0 \leftarrow 0$; gauche, le bit 7 va dans le Carry, le bit 0 est mis à 0.

CC

				H		N	Z	V	C
x	x	x	x	N	Z	*	C		

* $V = N + C$ après décalage

ASR — Décalage arithmétique à droite.

	IMMEDIAT			DIRECT			ETENDU			INDEXE		
	Co	Cy	Oc	Co	Cy	Oc	Co	Cy	Oc	Co	Cy	Oc
ASR	xx	xx	xx	07	6	2	77	7	3	67	6	2
ASRA	47	2	1									
ASRB	57	2	1									

$C \leftarrow b0$; Décale le contenu de l'octet de un bit vers la
 $b7 \leftarrow b7$; droite, le bit 0 va dans le carry, le bit 7 n'est pas modifié.

BCC — Branchement si pas de retenue.

	RELATIF		
	Co	Cy	Oc
BCC	24	3	2
LBCC	10.24	5	4

Effectue un branchement si Carry = 0

BCS — Branchement si retenue.

	RELATIF		
	Co	Cy	Oc
BCS	25	3	2
LBCS	10.25	5	4

Effectue un branchement si Carry = 1.

4.2 Les circuits de la famille
des microprocesseurs de la série 68XX

BEQ — Branchement si égal à zéro.

	RELATIF		
	Co	Cy	Oc
BEQ	27	3	2
LBEQ	10.27	5	4

Effectue un branchement si l'indicateur $Z = 1$.

BGE — Branchement si supérieur ou égal à zéro (signé)

	RELATIF		
	Co	Cy	Oc
BGE	2C	3	2
LBGE	10.2C	5	4

BGT — Branchement si supérieur (signé)

	RELATIF		
	Co	Cy	Oc
BGT	2E	3	2
LBGT	10.2E	5	4

Effectue un branchement si : $Z + (N*V) = 0$

* symbolise le OU EXCLUSIF

BHI — Branchement si supérieur (non signé)

	RELATIF		
	Co	Cy	Oc
BHI	22	3	2
LBHI	10.22	5	4

Effectue un branchement si : $C + Z = 0$

BHS — Branchement si supérieur ou égal (non signé)

	RELATIF		
	Co	Cy	Oc
BHS	24	3	2
LBHS	10.24	5	4

Effectue un branchement si : $C = 0$ BLO — Branchement si inférieur (non signé)

	RELATIF		
	Co	Cy	Oc
BLO	25	3	2
LBLO	10.25	5	4

Effectue un branchement si : $C = 1$ BIT — Test sur 1 bit.

	IMMEDIAT			DIRECT			ETENDU			INDEXE		
	Co	Cy	Oc	Co	Cy	Oc	Co	Cy	Oc	Co	Cy	Oc
BITA	85	2	2	95	4	2	B5	5	3	A5	4	2
BITB	C5	2	2	D5	4	2	F5	5	3	E5	4	2

Fait un « ET » logique entre l'opérande et l'accumulateur

BIT

H				N Z V C			
x	x	x	x	N	Z	O	x

BLE — Branchement si inférieur ou égal

	RELATIF		
	Co	Cy	Oc
BLE	2F	3	2
LBLE	10.2F	5	4

Effectue un branchement si : $Z + (N * V) = 0$

4.2 Les circuits de la famille
des microprocesseurs de la série 68XX

BLS — Branchement si inférieur ou égal (non signé)

	RELATIF		
	Co	Cy	Oc
BLS	23	3	2
LBLS	10.23	5	4

Effectue un branchement si : $Z + C = 1$

BLT — Branchement si inférieur (signé)

	RELATIF		
	Co	Cy	Oc
BLT	2D	3	2
LBLT	10.2D	5	4

Effectue un branchement si : $N * V = 1$

BMI — Branchement si négatif

	RELATIF		
	Co	Cy	Oc
BMI	2B	3	2
LBMI	10.2B	5	4

Effectue un branchement si : $N = 1$

BNE — Branchement si différent

	RELATIF		
	Co	Cy	Oc
BNE	26	3	2
LBNE	10.26	5	4

Effectue un branchement si : $Z = 0$

BPL — Branchement si positif

	RELATIF		
	Co	Cy	Oc
BPL	2A	3	2
LBPL	10.2A	5	4

Effectue un branchement si : $N = 0$ BRA — Branchement inconditionnel

	RELATIF		
	Co	Cy	Oc
BRA	20	3	2
LBRA	10.20	5	4

Effectue un branchement relatif

BRN — Branchement nul

	RELATIF		
	Co	Cy	Oc
BRN	21	3	2
LBRN	10.21	5	4

N'effectue aucun branchement (par opposition à BRA)

BSR — Branchement vers un sous-programme

	RELATIF		
	Co	Cy	Oc
BSR	8D	3	2
LBSR	17	5	4

Effectue un branchement vers un sous-programme
Le compteur programme est sauvé dans la pile S

4.2 Les circuits de la famille
des microprocesseurs de la série 68XX

BVC — Branchement si pas de débordement

	RELATIF		
	Co	Cy	Oc
BVC	28	3	2
LBVC	10.28	5	4

Effectue un branchement si : $V = 0$

BVS — Branchement si débordement

	RELATIF		
	Co	Cy	Oc
BVS	29	3	2
LBVS	10.29	5	4

Effectue un branchement si : $V = 1$

CLR — Mise à zéro.

	IMMEDIAT			DIRECT			ETENDU			INDEXE		
	Co	Cy	Oc	Co	Cy	Oc	Co	Cy	Oc	Co	Cy	Oc
CLR	xx	xx	xx	0F	6	2	7F	7	3	6F	6	2
CLRA	4F	2	1									
CLRB	5F	2	1									

Met à zéro le contenu de la mémoire ou de l'accumulateur.

4.2 Les circuits de la famille
des microprocesseurs de la série 68XX

6809

CLR

H				N Z V C			
x	x	x	x	0	1	0	0

CMP — Comparaison de la mémoire avec un accumulateur

	IMMEDIAT			DIRECT			ETENDU			INDEXE		
	Co	Cy	Oc	Co	Cy	Oc	Co	Cy	Oc	Co	Cy	Oc
CMPA	81	2	2	91	4	2	B1	5	3	A1	4	2
CMPB	C1	2	2	D1	4	2	F1	5	3	E1	4	2
CMPD	10.83	5	4	10.93	7	3	10.B3	8	4	10.A3	7	3
CMPS	11.8C	5	4	11.9C	7	3	11.BC	8	4	11.AC	7	3
CMPU	11.83	5	4	11.93	7	3	11.B3	8	4	11.A3	7	3
CMPY	10.8C	5	4	10.9C	7	3	10.BC	8	4	10.AC	7	3
CMPX	8C	4	3	9C	7	3	BC	7H	3	AC	7	3

Compare le contenu de la mémoire avec le contenu d'un accumulateur.

CMP

H				N Z V C			
x	x	x	x	N	Z	V	C

COM — Complémentation à un.

	IMMEDIAT			DIRECT			ETENDU			INDEXE		
	Co	Cy	Oc	Co	Cy	Oc	Co	Cy	Oc	Co	Cy	Oc
COM	xx	xx	xx	03	6	2	73	7	3	63	6	2
COMA	43	2	1									
COMB	53	2	1									

COM

H				N Z V C			
x	x	x	x	N	Z	0	0

CWAY — Attente d'une interruption

	INHERENT		
	Co	Cy	Oc
CWAY	3C	20	2

Effectue un ET logique entre le contenu du registre d'état et l'octet immédiat. Cela permet de bloquer le déroulement du programme et d'attendre une interruption IRQ, FIRQ ou NMI.

4.2 Les circuits de la famille
des microprocesseurs de la série 68XX

	H		N		Z	V	C
CWAY	x	x	x	x	x	x	x

DAA — Ajustement décimal

	INHERENT		
	Co	Nc	No
DAA	19	2	1

Ajuste le contenu de l'accumulateur A en décimal en fonction des indicateurs résultant de l'addition de deux nombres en BCD.

On ajoute 6 au quartet faible si :

- il est plus grand que 9
- le bit H est à 1.

On ajoute 6 au quartet fort si :

- il est plus grand que 9
- le bit C est à 1
- il est plus grand que 8 et le quartet faible a été corrigé.

	H		N		Z	V	C
DAA	x	x	x	x	N	Z	x

DEC — Décrémentation

	INHERENT			DIRECT			ETENDU			INDEXE		
	Co	Cy	Oc	Co	Cy	Oc	Co	Cy	Oc	Co	Cy	Oc
DEC	xx	xx	xx	OA	6	2	7A	7	3	6A	6	2
DECA	4A	2	1									
DECB	5A	2	1									

Soustrait 1 à l'opérande

	H		N		Z	V	C
DEC	x	x	x	x	N	Z	*

* V = 1 si l'opérande était \$80

* V = 0 dans tous les autres cas

EOR — OU EXCLUSIF

	IMMEDIAT			DIRECT			ETENDU			INDEXE		
	Co	Cy	Oc	Co	Cy	Oc	Co	Cy	Oc	Co	Cy	Oc
EORA	88	2	2	98	4	2	B8	5	3	A8	4	2
EORB	C8	2	2	D8	4	2	F8	5	3	E8	4	2

Effectue un OU EXCLUSIF entre l'accumulateur et l'opérande.

	H			N			Z	V	C
EOR	x	x	x	x	N	Z	O	x	x

EXG — Echange de contenu entre deux registres.

	INHERENT		
	Co	Nc	No
EXG	1E	7	2

Echange les contenus des registres désignés par le post octet.

Les deux quartets du post octet désignent chacun un registre.

16 Bits	8 Bits
0 → D	8 → A
1 → X	9 → B
2 → Y	A → CC
3 → U	B → DP
4 → S	
5 → PC	

	H			N			Z	V	C
EXG	x	x	x	x	x	x	x	x	x

INC — Incréméntation

	INHERENT			DIRECT			ETENDU			INDEXE		
	Co	Cy	Oc	Co	Cy	Oc	Co	Cy	Oc	Co	Cy	Oc
INC	xx	xx	xx	OC	6	2	7C	7	3	6C	6	2
INCA	4C	2	1									
INCB	5C	2	1									

Ajoute 1 à l'opérande

4.2 Les circuits de la famille des microprocesseurs de la série 68XX

	H		N		Z	V	C	
INC	x	x	x	x	N	Z	*	x

* V = 1 si l'opérande était \$7F

* V = 0 dans tous les autres cas

JMP — Saut inconditionnel à une adresse effective

	DIRECT			ETENDU			INDEXE		
	Co	Nc	No	Co	Nc	No	Co	Nc	No
JMP	OE	3	2	7E	4	3	6E	3	2

Transfère l'exécution du programme à l'adresse désignée par l'instruction.

	H		N		Z	V	C
JMP	x	x	x	x	x	x	x

JSR — Saut à un sous-programme

	DIRECT			ETENDU			INDEXE		
	Co	Nc	No	Co	Nc	No	Co	Nc	No
JSR	9D	7	2	BD	8	3	AD	7	2

Empile l'adresse effective suivant l'instruction JSR dans la pile système S et transfère l'exécution à l'adresse désignée par l'instruction.

	H		N		Z	V	C
JSR	x	x	x	x	x	x	x

LD — Chargement d'un registre

	IMMEDIAT			DIRECT			ETENDU			INDEXE		
	Co	Cy	Oc	Co	Cy	Oc	Co	Cy	Oc	Co	Cy	Oc
LDA	86	2	2	96	4	2	B6	5	3	A6	4	2
LDB	C6	2	2	D6	4	2	F6	5	3	E6	4	2
LDD	CC	3	3	DC	5	2	FC	6	3	EC	5	2
LDX	8E	3	3	9E	5	2	BE	6	3	AE	5	2
LDU	CE	3	3	DE	5	2	FE	6	3	EE	5	2
LDY	10.8E	4	4	10.9E	6	3	10.BE	7	4	10.AE	6	3
LDS	10.CE	4	4	10.DE	6	3	10.FE	7	4	10.EE	6	3

Charge dans le registre la valeur spécifiée ou le contenu mémoire.

	H		N		Z	V	C	
LD	x	x	x	x	N	Z	O	x

LEA — Chargement d'un registre 16 Bits avec une adresse

	INDEXE		
	Co	Nc	No
LEAS	32	4	2
LEAU	33	4	2
LEAX	30	4	2
LEAY	31	4	2

Calcule l'adresse effective en fonction de l'adressage indexé et charge cette valeur dans le registre.

			H		N		Z	V	C
LEAX/LEAY	x	x	x	x	x	Z	x	x	

	H		N		Z	V	C
LEAU/LEAS	x	x	x	x	x	x	x

LSL — Décalage logique à gauche

	INHERENT			DIRECT			ETENDU			INDEXE		
	Co	Cy	Oc	Co	Cy	Oc	Co	Cy	Oc	Co	Cy	Oc
LSL	xx	xx	xx	08	6	2	78	7	3	68	6	2
LSLA	48	2	1									
LSLB	58	2	1									

$$C \leftarrow b7.b6.b5.b4.b3.b2.b1.b0 \leftarrow 0$$

Effectue un décalage à gauche $b0 \leftarrow 0$

$$C \leftarrow b7$$

	H		N		Z	V	C
LSL	x	x	x	x	N	Z	* C

$$V = N * C \text{ (* représente un OU EXCLUSIF)}$$

4.2 Les circuits de la famille
des microprocesseurs de la série 68XX

LSR — Décalage logique à droite

	INHERENT			DIRECT			ETENDU			INDEXE		
	Co	Cy	Oc	Co	Cy	Oc	Co	Cy	Oc	Co	Cy	Oc
LSR	xx	xx	xx	04	6	2	74	7	3	64	6	2
LSRA	44	2	1									
LSRB	54	2	1									

$C \leftarrow b7.b6.b5.b4.b3.b2.b1.b0 \leftarrow 0$

Effectue un décalage à gauche $b0 \leftarrow 0$

$C \leftarrow b7$

	H				N Z V C			
LSR	x	x	x	x	0	Z	x	C

MUL — Multiplication entre registres

	INHERENT		
	Co	Nc	No
MUL	3D	11	1

Effectue une multiplication entre A et B et place le résultat dans D.

	H				N Z V C			
MUL	x	x	x	x	x	Z	x	*

* C = b7 du registre B après multiplication

NEG — Complémentation à 2

	INHERENT			DIRECT			ETENDU			INDEXE		
	Co	Cy	Oc	Co	Cy	Oc	Co	Cy	Oc	Co	Cy	Oc
NEG	xx	xx	xx	00	6	2	70	7	3	60	6	2
NEGA	40	2	1									
NEGB	50	2	1									

Complémente à deux le contenu de la mémoire ou de l'accumulateur.

	H				N Z V C			
NEG	x	x	x	x	N	Z	*	x

* V = 1 si l'opérande était \$80

* V = 0 dans tous les autres cas

NOP — Pas d'opération

	INHERENT		
	Co	Nc	No
NOP	12	2	1

	H		N		Z	V	C
NOP	x	x	x	x	x	x	x

OR — OU entre accumulateur et mémoire

	IMMEDIAT			DIRECT			ETENDU			INDEXE		
	Co	Cy	Oc	Co	Cy	Oc	Co	Cy	Oc	Co	Cy	Oc
ORA	8A	2	2	9A	4	2	BA	5	3	AA	4	2
ORB	CA	2	2	DA	4	2	FA	5	3	EA	4	2
ORCC	1A	3	2									

Effectue un OU logique entre l'accumulateur et l'opérande.

	H		N		Z	V	C
OR	x	x	x	x	N	Z	O

PSH — Empilement des registres

	INHERENT		
	Co	Nc	No
PSHS	34	5	2
PSHU	36	5	2

Empile les registres désignés par la valeur des bits du post-octet.

	b7	b6	b5	b4	b3	b2	b1	b0
Post-O	PC	US	X	Y	DP	B	A	CCR

Ordre d'empilement

	H		N		Z	V	C
PSH	x	x	x	x	x	x	x

4.2 Les circuits de la famille des microprocesseurs de la série 68XX

PUL — Dépilement des registres

	INHERENT		
	Co	Nc	No
PULS	35	5	2
PULU	37	5	2

Dépile les registres désignés par la valeur des bits du post-octet.

Post-O	b7	b6	b5	b4	b3	b2	b1	b0
	PC	US	X	Y	DP	B	A	CCR

Ordre de dépilement.

PUL	H N Z V C							
	x	x	x	x	N	Z	O	x

ROL — Rotation à gauche

	INHERENT			DIRECT			ETENDU			INDEXE		
	Co	Cy	Oc	Co	Cy	Oc	Co	Cy	Oc	Co	Cy	Oc
ROL	xx	xx	xx	09	6	2	79	7	3	69	6	2
ROLA	49	2	1									
ROLB	59	2	1									

Effectue une rotation à gauche de une position.

$$C \leftarrow b7.b6.b5.b4.b3.b2.b1.b0 \leftarrow C$$

ROL	H N Z V C							
	x	x	x	x	N	Z	*	C

$V = N * C$ après décalage (* représente un OU EXCLUSIF)

4.2 Les circuits de la famille
des microprocesseurs de la série 68XX

6809

ROR — Rotation à droite

	INHERENT			DIRECT			ETENDU			INDEXE		
	Co	Cy	Oc	Co	Cy	Oc	Co	Cy	Oc	Co	Cy	Oc
ROR	xx	xx	xx	06	6	2	76	7	3	66	6	2
RORA	46	2	1									
RORB	56	2	1									

Effectue une rotation à gauche de une position.

 $C \rightarrow b7.b6.b5.b4.b3.b2.b1.b0 \rightarrow C$

ROR

		H		N	Z	V	C
x	x	x	x	N	Z	x	C

RTI — Retour d'interruption

	INHERENT		
	Co	Nc	No
RTI	3B	15	1

Cette instruction est utilisée pour retourner au programme principal depuis une routine d'interruption. Il y a dépilement du contexte initial avant le retour.

RTS — Retour d'un sous-programme

	INHERENT		
	Co	Nc	No
RTS	39	5	1

Cette instruction dépile l'adresse de retour et transfère l'exécution à cette adresse.

NB : Adresse empilée par BSR ou JSR en fonctionnement normal.

RTS

H				N	Z	V	C
x	x	x	x	x	x	x	x

4.2 Les circuits de la famille des microprocesseurs de la série 68XX

SBC — Soustraction avec Carry (retenue)

	IMMEDIAT			DIRECT			ETENDU			INDEXE		
	Co	Cy	Oc	Co	Cy	Oc	Co	Cy	Oc	Co	Cy	Oc
SBCA	82	2	2	92	4	2	B2	5	3	A2	4	2
SBCB	C2	2	2	D2	4	2	F2	5	3	E2	4	2

Soustrait l'opérande à l'accumulateur et retire un au résultat si Carry était positionné à un.

	H				N Z V C			
SBC	x	x	x	x	N	Z	V	C

SEX — Extension du signe

	INHERENT		
	Co	Nc	No
SEX	10	2	1

Etend le signe de B (b7) à l'accumulateur D

b7=1 → B négatif → A=\$ FF → D=\$ FFxx
b7=0 → B positif → A=\$ 00 → D=\$ 00xx

	H				N Z V C			
SEX	x	x	x	x	N	Z	x	x

ST — Stockage d'un registre en mémoire

	IMMEDIAT			ETENDU			INDEXE		
	Co	Cy	Oc	Co	Cy	Oc	Co	Cy	Oc
STA	97	4	2	B7	5	3	A7	4	2
STB	D7	4	2	F7	5	3	E7	4	2
STD	DD	5	2	FD	6	3	ED	5	2
STX	9F	5	2	BF	6	3	AF	5	2
STU	DF	5	2	FF	6	3	EF	5	2
STY	10.9F	6	3	10.BF	7	4	10.AF	6	3
STS	10.DF	6	3	10.FF	7	4	10.EF	6	3

Stocke le contenu du registre dans l'adresse mémoire.

	H				N Z V C			
ST	x	x	x	x	N	Z	0	C

SUB — Soustraction

	IMMEDIAT			DIRECT			ETENDU			INDEXE		
	Co	Cy	Oc	Co	Cy	Oc	Co	Cy	Oc	Co	Cy	Oc
SUBA	80	2	2	90	4	2	BO	5	3	AO	4	2
SUBB	CO	2	2	DO	4	2	FO	5	3	EO	4	2
SUBD	83	4	3	93	6	2	B3	7	3	A3	6	2

Soustrait l'opérande à l'accumulateur.

	H		N		Z	V	C	
SUB	x	x	x	x	N	Z	V	C

SWI — Interruption logicielle

	INHERENT		
	Co	Nc	No
SWI	3 F	19	
SWI	3 F	19	1
SWI2	10.3 F	20	2
SWI3	11.3 F	20	2

Instruction permettant le saut après empilement du contexte à une routine d'interruption.

	H		N		Z	V	C
SWI	x	x	x	x	x	x	x

SYNC — Synchronisation extérieure

	INHERENT		
	Co	Nc	No
SYNC	13	2	1

Stoppe le déroulement du programme et se met en attente d'une interruption matérielle.

Le déroulement reprend immédiatement derrière SYNC.

	H		N		Z	V	C
SYNC	x	x	x	x	x	x	x

TFR — Transfert de registre à registre

	INHERENT		
	Co	Nc	No
TFR R1, R2	1F	7	2

4.2 Les circuits de la famille des microprocesseurs de la série 68XX

Transfère le contenu registre source dans le registre de destination ;
le post-octet précise source et destination.

b7.b6.b5.b4	SOURCE	0 → D	8 → A
b3.b2.b1.b0	DESTINATION	1 → X	9 → B
		2 → Y	A → CCR
		3 → U	B → DP
		4 → S	
		5 → PC	

TFR

	H		N	Z	V	C
x	x	x	x	x	x	x

TST — Test sur un octet.

	IMMEDIAT			DIRECT			ETENDU			INDEXE		
	Co	Cy	Oc	Co	Cy	Oc	Co	Cy	Oc	Co	Cy	Oc
TST	xx	xx	xx	OD	6	2	7D	7	3	6D	6	2
TSTA	4D	2	1									
TSTB	5D	2	1									

Positionne les indicateurs suivant le contenu de l'octet.

	H		N	Z	V	C
TST	x	x	x	x	N	Z
					O	x

Nous venons de décrire, aussi précisément que possible, le jeu d'instruction du 6809 de MOTOROLA.

Ceci vous sera très utile car il ne faut jamais oublier que seul le langage-machine est en mesure pour des raisons de « timing » de gérer des périphériques rapides.

Les langages dits évolués tels que Basic, Pascal, Forth et bien d'autres permettent de structurer des programmes importants impossibles à réaliser en assembleur.

N'oublions pas toutefois que ces langages sont écrits en assembleur...

Bibliographie

MC6809-MC6809E, Microprocessor Programing Manual, Motorola Inc.
 Programming the 6809, Rodnay Zaks and William Labiak, *Edit. Sybex*.
 Microprocesseurs et mémoires, Thomson-Csf Composants.
 Le microprocesseur 6809, Claude Dardanne, *Edit. Eyrolles*.

CODES HEXADÉCIMAUX DES INSTRUCTIONS D'ASSEMBLAGE

CLASSEMENT CROISSANT DES VALEURS HEXADÉCIMALES

00	NEG	Direct
03	COM	Direct
04	LSR	Direct
06	ROR	Direct
07	ASR	Direct
08	ASL	Direct
08	LSL	Direct
09	ROL	Direct
0A	DEC	Direct
0C	INC	Direct
0D	TST	Direct
0E	JMP	Direct
0F	CLR	Direct

1020	LBRA	Relatif
1021	LBRN	Relatif
1022	LBHI	Relatif
1023	LBLS	Relatif
1024	LBCC	Relatif
1024	LBHS	Relatif
1025	LBCS	Relatif
1025	LBLO	Relatif
1026	LBNE	Relatif
1027	LBEQ	Relatif
1028	LBVC	Relatif
1029	LBVS	Relatif
102A	LBPL	Relatif
102B	LBMI	Relatif
102C	LBGE	Relatif
102D	LBLT	Relatif
102E	LBGT	Relatif
102F	LBLE	Relatif
103F	SWI2	Inférent
1083	CMPD	Immédiat
108C	CMPLY	Immédiat
108E	LDY	Immédiat
1093	CMPD	Direct
109C	CMPLY	Direct
109E	LDY	Direct
109F	STY	Direct
10A3	CMPD	Indexé
10AC	CMPLY	Indexé
10AE	LDY	Indexé
10AF	STY	Indexé
10B3	CMPD	Etendu
10BC	CMPLY	Etendu
10BE	LDY	Etendu
10BF	STY	Etendu
10CE	LDS	Immédiat
10DE	LDS	Direct
10DF	STS	Direct
10EE	LDS	Indexé
10EF	STS	Indexé
10FE	LDS	Etendu
10FF	STS	Etendu

12	NOP	Inhérent
13	SYNC	Inférent
17	LBSR	Relatif
19	DAA	Inhérent
1A	ORCC	Immédiat
1C	ANDDCC	Immédiat
1D	SEX	Inhérent
1E	EXG	Immédiat
1F	TFR	Immédiat

20	BRA	Relatif
21	BRN	Relatif
22	BHI	Relatif
23	BLS	Relatif
24	BCC	Relatif
24	BHS	Relatif
25	BCS	Relatif
25	BLO	Relatif
26	BNE	Relatif
27	BEQ	Relatif
28	BVC	Relatif
29	BVS	Relatif
2A	BPL	Relatif
2B	BMI	Relatif
2C	BGE	Relatif
2D	BLT	Relatif
2E	BGT	Relatif
2F	BLE	Relatif

30	LEAX	Indexé
31	LEAY	Indexé
32	LEAS	Indexé
33	LEAU	Indexé
34	PSHS	Immédiat
35	PULS	Immédiat
36	PSHU	Immédiat
37	PULU	Immédiat
39	RTS	Inhérent
3A	ABX	Inhérent
3B	RTI	Inhérent
3C	CWAI	Immédiat
3D	MUL	Inhérent
3F	SWI	Inférent

40	NEGA	Inhérent
43	COMA	Inhérent
44	LSRA	Inhérent
46	RORA	Inhérent
47	ASRA	Inhérent
48	ASLA	Inhérent
48	LSLA	Inhérent
49	ROLA	Inhérent
4A	DECA	Inhérent
4C	INCA	Inhérent
4D	TSTA	Inférent

1 1 3 F	SWI3	Inférent
1 1 8 3	CMPU	Immédiat
1 1 8 C	CMPS	Immédiat
1 1 9 3	CMPU	Direct
1 1 9 C	CMPS	Direct
1 1 A 3	CMPU	Indexé
1 1 A C	CMPS	Indexé
1 1 B 3	CMPU	Etendu
1 1 B C	CMPS	Etendu

6 0	NEG	Indexé
6 3	COM	Indexé
6 4	LSR	Indexé
6 6	ROR	Indexé
6 7	ASR	Indexé
6 7	CLR	Indexé
6 8	ASL	Indexé
6 8	LSL	Indexé
6 9	ROL	Indexé
6 A	DEC	Indexé
6 C	INC	Indexé
6 D	TST	Indexé
6 E	JMP	Indexé

8 0	SUBA	Immédiat
8 1	CMPA	Immédiat
8 2	SBCA	Immédiat
8 3	SUBD	Immédiat
8 4	ANDA	Immédiat
8 5	BITA	Immédiat
8 6	LDA	Immédiat
8 8	EORA	Immédiat
8 9	ADCA	Immédiat
8 A	ORA	Immédiat
8 B	ADDA	Immédiat
8 C	CMPX	Immédiat
8 D	BSR	Relatif
8 E	LDX	Immédiat

A 0	SUBA	Indexé
A 1	CMPA	Indexé
A 2	SBCA	Indexé
A 3	SUBD	Indexé
A 4	ANDA	Indexé
A 5	BITA	Indexé
A 6	LDA	Indexé
A 7	STA	Indexé
A 8	EORA	Indexé
A 9	ADCA	Indexé
A A	ORA	Indexé
A B	ADDA	Indexé
A C	CMPX	Indexé
A D	JSR	Indexé
A E	LDX	Indexé
A F	STX	Indexé

C 0	SUBB	Immédiat
C 1	CMPB	Immédiat
C 2	SBCB	Immédiat
C 3	ADDD	Immédiat
C 4	ANDB	Immédiat
C 5	BITB	Immédiat
C 6	LDB	Immédiat

4 F	CLRA	Inhérent
-----	------	----------

5 0	NEGB	Inhérent
5 3	COMB	Inhérent
5 4	LSRB	Inhérent
5 6	RORB	Inhérent
5 7	ASRB	Inhérent
5 8	ASLB	Inhérent
5 8	LSLB	Inhérent
5 9	ROLB	Inhérent
5 A	DECB	Inhérent
5 C	INCB	Inhérent
5 D	TSTB	Inférent
5 F	CLRB	Inhérent

7 0	NEG	Etendu
7 3	COM	Etendu
7 4	LSR	Etendu
7 6	ROR	Etendu
7 7	ASR	Etendu
7 7	CLR	Etendu
7 8	ASL	Etendu
7 8	LSL	Etendu
7 9	ROL	Etendu
7 A	DEC	Etendu
7 C	INC	Etendu
7 D	TST	Etendu
7 E	JMP	Etendu

9 0	SUBA	Direct
9 1	CMPA	Direct
9 2	SBCA	Direct
9 3	SUBD	Direct
9 4	ANDA	Direct
9 5	BITA	Direct
9 6	LDA	Direct
9 7	STA	Direct
9 8	EORA	Direct
9 9	ADCA	Direct
9 A	ORA	Direct
9 B	ADDA	Direct
9 C	CMPX	Direct
9 D	JSR	Direct
9 E	LDX	Direct
9 F	STX	Direct

B 0	SUBA	Etendu
B 1	CMPA	Etendu
B 2	SBCA	Etendu
B 3	SUBD	Etendu
B 4	ANDA	Etendu
B 5	BITA	Etendu
B 6	LDA	Etendu
B 7	STA	Etendu
B 8	EORA	Etendu
B 9	ADCA	Etendu
B A	ORA	Etendu
B B	ADDA	Etendu
B C	CMPX	Etendu
B D	JSR	Etendu
B E	LDX	Etendu
B F	STX	Etendu

C 8	EORB	Immédiat
C 9	ADCB	Immédiat
C A	ORB	Immédiat
C B	ADDB	Immédiat
C C	LDD	Immédiat
C E	LDU	Immédiat

E 0	SUBB	Indexé
E 1	CMPB	Indexé
E 2	SBCB	Indexé
E 3	ADDD	Indexé
E 4	ANDB	Indexé
E 5	BITB	Indexé
E 6	LDB	Indexé
E 7	STB	Indexé
E 8	EORB	Indexé
E 9	ADCB	Indexé
E A	ORB	Indexé
E B	ADDB	Indexé
E C	LDD	Indexé
E D	STD	Indexé
E E	LDU	Indexé
E F	STU	Indexé

D 0	SUBB	Direct
D 1	CMPB	Direct
D 2	SBCB	Direct
D 3	ADDD	Direct
D 4	ANDB	Direct
D 5	BITB	Direct
D 6	LDB	Direct
D 7	STB	Direct
D 8	EORB	Direct
D 9	ADCB	Direct
D A	ORB	Direct
D B	ADDB	Direct
D C	LDD	Direct
D D	STD	Direct
D E	LDU	Direct
D F	STU	Direct

F 0	SUBB	Etendu
F 1	CMPB	Etendu
F 2	SBCB	Etendu
F 3	ADDD	Etendu
F 4	ANDB	Etendu
F 5	BITB	Etendu
F 6	LDB	Etendu
F 7	STB	Etendu
F 8	EORB	Etendu
F 9	ADCB	Etendu
F A	ORB	Etendu
F B	ADDB	Etendu
F C	LDD	Etendu
F D	STD	Etendu
F E	LDU	Etendu
F F	STU	Etendu