

VI - LE JEU D'INSTRUCTIONS DU 68000;**VI - 1 - VUE D'ENSEMBLE DU JEU D'INSTRUCTIONS**

Les Instructions ombrées sont privilégiées, elles ne peuvent être exécutées qu'en mode superviseur.

VI - 2 - DÉTAIL DES INSTRUCTIONS

Dans les pages qui suivent, les instructions sont étudiées en détail. Elles sont en principe dans l'ordre alphabétique mais quelques exceptions ont été faites dans un souci de clarté. Certaines instructions aux fonctions très voisines ont été regroupées afin de réduire un peu la pagination de l'ouvrage.

Les colonnes de droite des tableaux des deux pages suivantes, décrivent l'influence de l'exécution de l'instruction sur le registre de Code Condition

A ou N: Bit **A**ffecté par l'instruction ou **N**on affecté par l'instruction

?: Bit mis dans un état indéterminé par l'instruction-

0 ou 1: Bit mis à **0** ou à **1** par l'instruction

MNEMONIQUE	DESCRIPTION	FONCTION	ACTION sur CCR				
			X	N	Z	V	C
ABCD	Addition DCBN	Srcce + Dest + X dans Dest (base 10)	A	?	A	?	A
ADD	Addition Binaire Naturel	Srcce + Dest dans Dest (base 2)	A	?	A	?	A
ADDA	Addition d'adresses	Srcce + Dest dans Dest (dest = An)	N	N	N	N	N
ADDI	Addition Immédiate	Donnée + Dest dans Destination	A	A	A	A	A
ADDQ	Addition Im rapide	Donnée + Dest dans Destination	A	A	A	A	A
ADDX	Addition bin + X	Srcce + Dest + X dans Destination	A	A	A	A	A
AND	ET Logique	Srcce ET Dest dans Destination	N	A	A	0	0
ANDI	ET Log Imméd	Donnée ET Dest dans Destination	N	A	A	0	0
ANDI to CCR	ET dans CCR	Donnée ET CCR dans CCR	A	A	A	A	A
ANDI to SR	ET dans SR	Donnée ET SR dans SR	A	A	A	A	A
ASL ASR	Décalage Arithmétique	Décal à G ou D de n positions	A	A	A	A	A
Bcc	Branch Conditionnel	Si Condit Vraie PC + d dans PC	N	N	N	N	N
BCHG	Test bit et Complémente	Bit testé dans Z puis Comp du bit	N	N	A	N	N
BCLR	Test bit et RAZ	Bit testé dans Z puis RAZ du bit	N	N	A	N	N
BRA	Branchement Incond	PC + Dép dans PC Inconditionnel	N	N	N	N	N
BSET	Teste bit et met à 1	Bit testé dans Z puis bit = 1	N	N	A	N	N
BSR	Branchement à SP	Empile PC puis PC + d dans PC	N	N	N	N	N
BTST	Teste un bit	Bit testé dans Z	N	N	A	N	N
CHK	Teste Source	Compa Srcce à 0 et Dn en dehors Trap	N	A	?	?	?
CLR	Mise à 0	Mise à zéro d'un opérande	N	0	1	0	0
CMP	Comparaison	Dest - Srcce résultat actualise CCR	N	A	A	A	A
CMPA	Comparaison d'adresses	Dest (An) - Srcce actualise CCR	N	A	A	A	A
CMPI	Comparaison Imm	Dest - Donnée actualise CCR	N	A	A	A	A
CMPM	Comparais de Mémoires	Mémoire - Mémoire actualise CCR	N	A	A	A	A
DBcc	Décérmente et Brch Cond	Si cond fausse Décr et Branche	N	N	N	N	N
DIVS	Division signée	Div signée MLong ÷Mot = Mot	N	A	A	A	0
DIVU	Division non signée	Div non signée MLong ÷Mot = Mot	N	A	A	A	A
EOR	OU exclusif	Ou ex bit à bit entre Dn et un opérande	N	A	A	0	0
EORI	OU exclusif immédiat	Donnée immé OUex Opérande	N	A	A	0	0
EORI to CCR	OU exc dans CCR	CCR OU ex Donnée dans CCR	A	A	A	A	A
EORI to SR	OU ex dans SR	SR OU ex Donnée dans SR	A	A	A	A	A
EXG	Permutation	Permutation du contenu de 2 registres	N	N	N	N	N
EXT	Extension de signe	Extension du signe dans un registre	N	A	A	0	0
ILLEGAL	Trap Inst illégale	Trap sur instruction Illégale	N	N	N	N	N
JMP	Saut inconditionnel	Saut inconditionnel, Adresse dans PC	N	N	N	N	N
JSR	Saut à Sous Prog	Saut incond à sous programme	N	N	N	N	N
LEA	Adresse Eff dans An	Charge Adresse Effective dans An	N	N	N	N	N
LINK	Création de Lien	Empil An,SP ds An,SP+d ds SP	N	N	N	N	N
LSL LSR	Décalage Logique	Décalage logique gauche ou droite	A	A	A	0	A
MOVE	Transfert	Transfert de Srcce dans Destination	N	A	A	0	0
MOVEA	Transf d'adresse	Transfert de Srcce dans An	N	N	N	N	N

MNEMONIQUE	DESCRIPTION	FONCTION	ACTION sur CCR				
			X	N	Z	V	C
MOVE to CCR	Transf dans CCR	Transfert de Srce dans CCR	A	A	A	A	A
MOVE fr SR	Transf de SR	Transfert de SR dans Dest	N	N	N	N	N
MOVE to SR	Transf dans SR	Transfert de Srce dans SR	A	A	A	A	A
MOVE USP	Trf de/dans USP	Srce dans USP ou USP dans Dest	N	N	N	N	N
MOVEM	Transfert multiple	Transf de ou dans plusieurs registres	N	N	N	N	N
MOVEP	Transfert périphérique	Transfert avec réorganisation	N	N	N	N	N
MOVEQ	Transfert rapide	Donnée(≤8bits) dans Destination	N	A	A	0	0
MULS	Multiplication Signée	Multiplication signée = 32 ou 64 bits	N	A	A	A	0
MULU	Multiplication non Signée	Multiplic non signée = 32 ou 64 bits	N	A	A	A	0
NBCD	Complément à 10	Complément à 10 d'un octet avec X	A	?	A	?	A
NEG	Complément à 2	Complément à 2 de Dest dans Dest	A	A	A	A	A
NEGX	Complément à 2 avec X	Compl à 2 avec X de Dest ds Dest	A	A	A	A	A
NOP	Pas d'Opération	Pas d'opération	N	N	N	N	N
NOT	Complément à 1	Complément à 1 de Dest dans Dest	N	A	A	0	0
OR	OU logique	Source OU Dest dans Destination	N	A	A	0	0
ORI	OU Immédiat	Donnée OU Dest dans Destination	N	A	A	0	0
ORI to CCR	OU Immédiat dans CCR	Donnée OU CCR dans CCR	A	A	A	A	A
ORI to SR	OU Immédiat dans SR	Donnée OU SR dans SR	A	A	A	A	A
PEA	Push EA	Empile l'Adresse Effective	N	N	N	N	N
RESET	Reset	Effectue un Reset des Périphériques	N	N	N	N	N
ROL ROR	Rotation	Rotation à Gauche ou à Droite	N	A	A	0	A
ROXL ROXR	Rotation par X	Rotation à Gauche ou à Droite par X	A	A	A	0	A
RTE	Retour d'Exception	Restitution de PC et SR	A	A	A	A	A
RTR	Retour et Restaure	Restitution de PC et CCR	A	A	A	A	A
RTS	Retour de SProg	Restitution de PC	N	N	N	N	N
SBCD	Soustraction DCBN X	Dest - Srce - X dans Dest en DCBN	A	?	A	?	A
Scc	Mise à 1 ou 0 Cond	Si condition vraie Dest=FF sinon 00	N	N	N	N	N
STOP	Stop Mpu	Donnée dans SR et Stop	A	A	A	A	A
SUB	Soustraction	Dest - Srce dans Destination	A	A	A	A	A
SUBA	Soustraction d'Adresse	An - Srce dans An	N	N	N	N	N
SUBI	Soustraction Immédiate	Dest - Donnée dans Destination	A	A	A	A	A
SUBQ	Soustraction Im Rapide	Dest - Donnée dans Destination	A	A	A	A	A
SUBX	Soustraction avec X	Dest - Srce - X dans Destination	A	A	A	A	A
SWAP	Permutation de Mots	Permute les 2 mots d'un registre Dn	N	A	A	0	0
TAS	Teste et =1	Teste octet, actual CCR et MSB =1	N	A	A	0	0
TRAP	Trap Incond	Exception Trappe inconditionnelle	N	N	N	N	N
TRAPV	Trap sur V=1	Si V =1 alors exception Trappe	N	N	N	N	N
TST	Test Opérande	Opérande - 0 actualise CCR	N	A	A	0	0
UNLK	Supp Lien	Restitue SP et An	N	N	N	N	N

VI - 2 - DÉTAIL DES INSTRUCTIONS

Dans les pages qui suivent, les instructions sont étudiées en détail. Elles sont en principe dans l'ordre alphabétique mais quelques exceptions ont été faites dans un souci de clarté. Certaines instructions aux fonctions très voisines ont été regroupées afin de réduire un peu la pagination de l'ouvrage.

ABCD

ABCD

ADDITION DCBN AVEC RETENUE

OPERATION: SOURCE₁₀ + DESTINATION₁₀ + X dans DESTINATION

SYNTAXE ASSEMBLEUR:

ABCD Dy,Dx
 ABCD -(Ay),-(Ax)

PORTEE:

Octet de poids faible

FONCTION:

Additionne en DCBN l'octet source à l'octet destination et à l'extension X.
 - Entre registres de données (octets de poids faible)
 Dx = Registre destination
 Dy = Registre source
 - Entre cases mémoire, adressage par registres d'adresses pré-décrémentés

uniquement.

CODES CONDITION:

X	N	Z	V	C
A	?	A	?	A

X identique à C
 N indéfini
 Z mis à 0 si résultat non nul sinon inchangé
 V indéfini
 C mis à 1 si un report décimal est généré sinon mis à 0

FORMAT DU MOT INSTRUCTION

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	N° Rx			1	0	0	0	0	R/M	N° Ry		

R/M = 0 = Opération entre Registres de Données
 = 1 = Opération entre cases mémoire
 N° Rx Numéro du Registre Destination
 N° Ry Numéro du Registre Source

EXEMPLE D'UTILISATION

ABCD D2,D3
 mot instruction: **0xC702**

AVANT EXECUTION

D2	33	44	55	66
D3	44	55	66	77

X	N	Z	V	C
0	0	0	0	0

APRES EXECUTION

D2	33	44	55	66
D3	44	55	66	43

X	N	Z	V	C
1	0	0	0	1

ADD**ADD****ADDITION****OPERATION:** SOURCE + DESTINATION dans DESTINATION**SYNTAXE ASSEMBLEUR:**

ADD <AE>,Dn
ADD Dn,<AE>

PORTEE:

Octet (.B), Mot (.W), Mot long (.L)

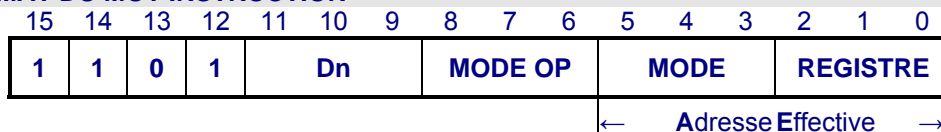
FONCTION:

Additionne en binaire naturel l'opérande source à l'opérande destination et place le résultat dans la destination.

CODES CONDITION:

X	N	Z	V	C
A	A	A	A	A

X identique à C
N affiche le signe du résultat (bit de poids fort)
Z mis à 1 si résultat nul sinon à 0
V mis à 1 si débordement en complément à deux sinon 0
C mis à 1 si un report est généré sinon mis à 0

FORMAT DU MOT INSTRUCTION**Dn** N° du registre de données employé

CHAMP MODE OP			
OCTET	MOT	MOT LONG	OPERATION
000	001	010	<AE> + <Dn> dans <Dn>
100	101	110	<Dn> + <AE> dans <AE>

Le champ Adresse effective est codé, en fonction de l'adressage utilisé, suivant le tableau ci dessous:

NOTA	MODE D'ADRESS	MODE	REGISTRE	NOTA	MODE D'ADRESS	MODE	REGISTRE
(1)	Dn	000	N° Registre		d(An,Xi)	110	N° Registre
(1 + 2)	An	001	N° Registre		Absolu Court	111	000
	(An)	010	N° Registre		Absolu Long	111	001
	(An)+	011	N° Registre	(1)	d(PC)	111	010
	-(An)	100	N° Registre	(1)	d(PC,Xi)	111	011
	d(An)	101	N° Registre	(1)	Immédiat	111	100

NOTA (1) Mode d'adressage utilisable en source seul
(2) Mode n'acceptant pas les opérandes Octet

EXEMPLE D'UTILISATION

	ADD.B				D2,D3				mot instruction: 0xD602						
	AVANT EXECUTION				APRES EXECUTION										
D2	33	44	55	66	D2	33	44	55	66						
D3	44	55	66	77	D3	44	55	66	DD						
	X	N	Z	V	C		X	N	Z	V	C				
	0	0	0	0	0		0	1	0	1	0				

ADDA ADDA

ADDITION DANS REGISTRE D'ADRESSES

OPERATION: SOURCE + DESTINATION dans DESTINATION

SYNTAXE ASSEMBLEUR:

ADDA <AE>,An

PORTEE:

Mot (.W), Mot long (.L)

FONCTION:

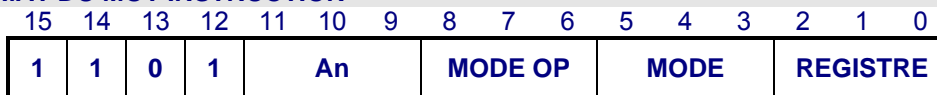
Additionne en binaire naturel l'opérande source à l'opérande destination et place le résultat dans la destination. Bien que l'opération puisse se faire sur un mot ou un mot long, le registre destination dans son entier est influencé. Lorsque l'addition porte sur un mot, le signe du nombre additionné est étendu à 32 bits avant l'opération.

CODES CONDITION:

X	N	Z	V	C
N	N	N	N	N

Cette instruction n'influence pas le registre de codes condition.

FORMAT DU MOT INSTRUCTION



← Adresse Effective →

An N° du registre d'adresses employé

CHAMP MODE OP			
OCTET	MOT	MOT LONG	OPERATION
Néant	011	111	<AE> + <An> dans <An>

Le champ Adresse effective est codé, en fonction de l'adressage utilisé, suivant le tableau ci dessous. Dans cette instruction l'AE est toujours source

NOTA	MODE D'ADRESS	MODE	REGISTRE	NOTA	MODE D'ADRESS	MODE	REGISTRE
	Dn	000	N° Registre		d(An,Xi)	110	N° Registre
	An	001	N° Registre		Absolu Court	111	000
	(An)	010	N° Registre		Absolu Long	111	001
	(An)+	011	N° Registre		d(PC)	111	010
	-(An)	100	N° Registre		d(PC,Xi)	111	011
	d(An)	101	N° Registre		Immédiat	111	100

EXEMPLE D'UTILISATION

<p style="text-align: center;">ADDA.W D2,A3</p> <p style="text-align: center;">AVANT EXECUTION</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 10%; text-align: right;">D2</td> <td style="border: 1px solid black; padding: 2px;">00</td> <td style="border: 1px solid black; padding: 2px;">00</td> <td style="border: 1px solid black; padding: 2px;">80</td> <td style="border: 1px solid black; padding: 2px;">22</td> </tr> <tr> <td style="text-align: right;">A3</td> <td style="border: 1px solid black; padding: 2px;">00</td> <td style="border: 1px solid black; padding: 2px;">10</td> <td style="border: 1px solid black; padding: 2px;">00</td> <td style="border: 1px solid black; padding: 2px;">00</td> </tr> </table> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="padding: 2px;">X</td> <td style="padding: 2px;">N</td> <td style="padding: 2px;">Z</td> <td style="padding: 2px;">V</td> <td style="padding: 2px;">C</td> </tr> <tr> <td style="padding: 2px;">0</td> <td style="padding: 2px;">0</td> <td style="padding: 2px;">0</td> <td style="padding: 2px;">0</td> <td style="padding: 2px;">0</td> </tr> </table>	D2	00	00	80	22	A3	00	10	00	00	X	N	Z	V	C	0	0	0	0	0	<p style="text-align: center;">mot instruction: 0xD6C2</p> <p style="text-align: center;">APRES EXECUTION</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 10%; text-align: right;">D2</td> <td style="border: 1px solid black; padding: 2px;">00</td> <td style="border: 1px solid black; padding: 2px;">00</td> <td style="border: 1px solid black; padding: 2px;">80</td> <td style="border: 1px solid black; padding: 2px;">22</td> </tr> <tr> <td style="text-align: right;">A3</td> <td style="border: 1px solid black; padding: 2px;">00</td> <td style="border: 1px solid black; padding: 2px;">0F</td> <td style="border: 1px solid black; padding: 2px;">80</td> <td style="border: 1px solid black; padding: 2px;">22</td> </tr> </table> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="padding: 2px;">X</td> <td style="padding: 2px;">N</td> <td style="padding: 2px;">Z</td> <td style="padding: 2px;">V</td> <td style="padding: 2px;">C</td> </tr> <tr> <td style="padding: 2px;">0</td> <td style="padding: 2px;">0</td> <td style="padding: 2px;">0</td> <td style="padding: 2px;">0</td> <td style="padding: 2px;">0</td> </tr> </table>	D2	00	00	80	22	A3	00	0F	80	22	X	N	Z	V	C	0	0	0	0	0
D2	00	00	80	22																																					
A3	00	10	00	00																																					
X	N	Z	V	C																																					
0	0	0	0	0																																					
D2	00	00	80	22																																					
A3	00	0F	80	22																																					
X	N	Z	V	C																																					
0	0	0	0	0																																					

ADDI**ADDI****ADDITION IMMEDIATE**

OPERATION: DONNEE IMMEDIATE + DESTINATION dans DESTINATION

SYNTAXE ASSEMBLEUR:

ADDI #<DONNEE>,<AE>

PORTEE:

Octet , Mot , Mot long

FONCTION:

Additionne en binaire naturel la donnée immédiate à l'opérande destination et place le résultat dans la destination. La taille de la donnée immédiate détermine la portée de l'opération.

CODES CONDITION:

X	N	Z	V	C
A	A	A	A	A

X identique à C
 N affiche le signe du résultat (bit de poids fort)
 Z mis à 1 si résultat nul sinon à 0
 V mis à 1 si débordement en complément à deux sinon 0
 C mis à 1 si un report est généré sinon mis à 0

FORMAT DU MOT INSTRUCTION

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	1	1	0	SIZE	MODE	REGISTRE					

← Adresse Effective →

Suivi d'un mot extension, pour la donnée octet (octet de poids faible du mot extension) ou pour la donnée mot et de deux mots extension pour la donnée mot long.

CHAMP SIZE			
OCTET	MOT	MOT LONG	OPERATION
00	01	10	Donnée +<AE> dans <AE>

Le champ Adresse effective est codé, en fonction de l'adressage utilisé, suivant le tableau ci dessous. L'AE est toujours destination, il en découle un certain nombre de restrictions.

NOTA	MODE D'ADRESS	MODE	REGISTRE	NOTA	MODE D'ADRESS	MODE	REGISTRE
	Dn	000	N° Registre		d(An,Xi)	110	N° Registre
(3)	An				Absolu Court	111	000
	(An)	010	N° Registre		Absolu Long	111	001
	(An)+	011	N° Registre	(3)	d(PC)		
	-(An)	100	N° Registre	(3)	d(PC,Xi)		
	d(An)	101	N° Registre	(3)	Immédiat		

NOTA (3) Mode d'adressage ne pouvant être utilisé

EXEMPLE D'UTILISATION

ADDI.W				#0XA987,D3	mot extension 0XA987			
mot instruction: 0X0643								
AVANT EXECUTION				APRES EXECUTION				
D2	XX	XX	XX	XX	XX	XX	XX	XX
D3	44	55	88	77	44	55	31	FE
	X	N	Z	V	C			
	0	0	0	0	0			
	X	N	Z	V	C			
	1	0	0	1	1			

ADDQ

ADDQ

ADDITION IMMEDIATE RAPIDE

OPERATION: DONNEE IMMEDIATE + DESTINATION dans DESTINATION

SYNTAXE ASSEMBLEUR:

ADDQ #<DONNEE>, <AE>

PORTEE:

Octet (.B), Mot (.W), Mot long(.L)

FONCTION:

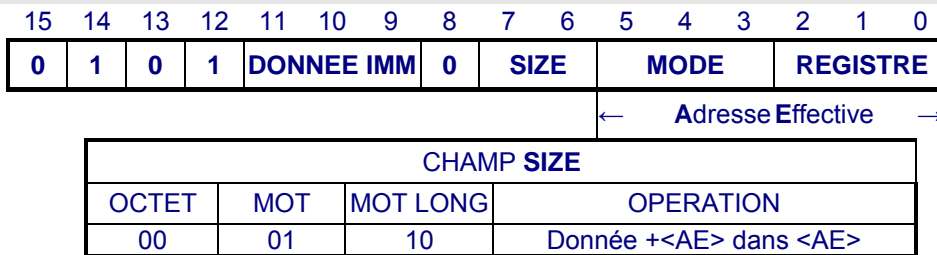
Additionne en binaire naturel la donnée immédiate à l'opérande destination et place le résultat dans la destination. L'opération peut être effectuée sur un registre d'adresses mais alors, elle **ne peut porter sur un octet et n'influence pas CCR**

CODES CONDITION:

X	N	Z	V	C
A	A	A	A	A

- X identique à C
- N affiche le signe du résultat (bit de poids fort)
- Z mis à 1 si résultat nul sinon à 0
- V mis à 1 si débordement en complément à deux sinon 0
- C mis à 1 si un report est généré sinon mis à 0

FORMAT DU MOT INSTRUCTION



Le champ Adresse effective est codé, en fonction de l'adressage utilisé, suivant le tableau ci dessous. L'AE est toujours destination, il en découle un certain nombre de restrictions.

NOTA	MODE D'ADRESS	MODE	REGISTRE	NOTA	MODE D'ADRESS	MODE	REGISTRE
	Dn	000	N° Registre		d(An,Xi)	110	N° Registre
(2)	An	001	N° Registre		Absolu Court	111	000
	(An)	010	N° Registre		Absolu Long	111	001
	(An)+	011	N° Registre	(3)	d(PC)		
	-(An)	100	N° Registre	(3)	d(PC,Xi)		
	d(An)	101	N° Registre	(3)	Immédiat		

- NOTA (2) Mode d'adressage n'acceptant pas les opérandes octet
 (3) Mode d'adressage ne pouvant être utilisé

EXEMPLE D'UTILISATION

<p style="text-align: center;">ADDQ.W</p> <p style="text-align: center;">AVANT EXECUTION</p> <table border="1" style="width: 100%;"> <tr> <td>D2</td> <td>XX</td> <td>XX</td> <td>XX</td> <td>XX</td> </tr> <tr> <td>D3</td> <td>44</td> <td>55</td> <td>FF</td> <td>FF</td> </tr> </table> <table border="1" style="width: 100%;"> <tr> <td>X</td> <td>N</td> <td>Z</td> <td>V</td> <td>C</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> </table>	D2	XX	XX	XX	XX	D3	44	55	FF	FF	X	N	Z	V	C	0	0	0	0	0	<p style="text-align: center;">#1,D3</p> <p style="text-align: center;">mot instruction: 0X5243</p> <p style="text-align: center;">APRES EXECUTION</p> <table border="1" style="width: 100%;"> <tr> <td>D2</td> <td>XX</td> <td>XX</td> <td>XX</td> <td>XX</td> </tr> <tr> <td>D3</td> <td>44</td> <td>55</td> <td>00</td> <td>00</td> </tr> </table> <table border="1" style="width: 100%;"> <tr> <td>X</td> <td>N</td> <td>Z</td> <td>V</td> <td>C</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>0</td> <td>1</td> </tr> </table>	D2	XX	XX	XX	XX	D3	44	55	00	00	X	N	Z	V	C	1	0	1	0	1
D2	XX	XX	XX	XX																																					
D3	44	55	FF	FF																																					
X	N	Z	V	C																																					
0	0	0	0	0																																					
D2	XX	XX	XX	XX																																					
D3	44	55	00	00																																					
X	N	Z	V	C																																					
1	0	1	0	1																																					

ADDX

ADDX

AND ET LOGIQUE AND

OPERATION: SOURCE ET DESTINATION dans DESTINATION

SYNTAXE ASSEMBLEUR:

AND <AE>,Dn
AND Dn,<AE>

PORTEE:

Octet (.B), Mot (.W), Mot long (.L)

FONCTION:

Effectue un ET logique entre la source et la destination , résultat dans la destination.

Le contenu d'un registre d'adresses ne peut être employé dans cette instruction.

CODES CONDITION:

X	N	Z	V	C
N	A	A	0	0

X n'est pas affecté
N affiche le signe du résultat (bit de poids fort)
Z mis à 1 si résultat nul sinon 0
V mis à 0
C mis à 0

FORMAT DU MOT INSTRUCTION

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	Dn				MODE OP		MODE		REGISTRE			

← Adresse Effective →

Le champ Dn reçoit le N° du registre Dn utilisé

CHAMP MODE OP			
OCTET	MOT	MOT LONG	OPERATION
000	001	010	<AE> ET <Dn> dans <Dn>
100	101	110	<Dn> ET <AE> dans <AE>

Le champ **Adresse effective** est codé, en fonction de l'adressage utilisé, suivant le tableau ci dessous:

NOTA	MODE D'ADRESS	MODE	REGISTRE	NOTA	MODE D'ADRESS	MODE	REGISTRE
(1)	Dn	000	N° Registre		d(An,Xi)	110	N° Registre
(3)	An				Absolu Court	111	000
	(An)	010	N° Registre		Absolu Long	111	001
	(An)+	011	N° Registre	(1)	d(PC)	111	010
	-(An)	100	N° Registre	(1)	d(PC,Xi)	111	011
	d(An)	101	N° Registre	(1)	Immédiat	111	100

NOTA (1) Mode d'adressage utilisable en source seul

(3) Mode d'adressage ne pouvant être utilisé ni en source ni en destination.

EXEMPLE D'UTILISATION

AND.L D2,D3

(lorsque <AE> est de la forme Dn , <AE> est source donc ici <AE> ET Dn dans Dn)

mot instruction: **0XC682**

AVANT EXECUTION

D2	00	00	FF	00
D3	44	55	66	77

APRES EXECUTION

D2	00	00	FF	00
D3	00	00	66	00

X	N	Z	V	C
0	0	0	0	0

X	N	Z	V	C
0	0	0	0	0

ANDI**ANDI****ET LOGIQUE IMMEDIAT****OPERATION:** DONNEE IMMEDIATE ET DESTINATION dans DESTINATION**SYNTAXE ASSEMBLEUR:**

ANDI #Donnée,<AE>

PORTEE:

Octet (.B), Mot (.W), Mot long (.L)

FONCTION:

Effectue un ET logique entre la donnée immédiate et la destination , résultat dans la destination. Le contenu d'un registre d'adresses ne peut être employé dans cette instruction.

CODES CONDITION:

X	N	Z	V	C
N	A	A	0	0

X n'est pas affecté
 N affiche le signe du résultat (bit de poids fort)
 Z mis à 1 si résultat nul sinon 0
 V mis à 0
 C mis à 0

FORMAT DU MOT INSTRUCTION

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	1	0	SIZE	MODE	REGISTRE					

← Adresse Effective →

Le mot instruction est suivi d'un mot extension pour les opérations sur un octet (octet de poids faible de ce mot) ou sur un mot et de deux mots extension pour les opérations sur mots longs

CHAMP SIZE			
OCTET	MOT	MOT LONG	OPERATION
00	01	10	Source ET <AE> dans <AE>

Le champ **Adresse effective** est codé, en fonction de l'adressage utilisé, suivant le tableau ci dessous:

NOTA	MODE D'ADRESS	MODE	REGISTRE	NOTA	MODE D'ADRESS	MODE	REGISTRE
	Dn	000	N° Registre		d(An,Xi)	110	N° Registre
(3)	An				Absolu Court	111	000
	(An)	010	N° Registre		Absolu Long	111	001
	(An)+	011	N° Registre	(3)	d(PC)		
	-(An)	100	N° Registre	(3)	d(PC,Xi)		
	d(An)	101	N° Registre	(3)	Immédiat		

NOTA (3) Mode d'adressage ne pouvant être utilisé.

EXEMPLE D'UTILISATION**ANDI.L #0X0F,D3**mot instruction: **0x0283** 1er mot extension: **0000** 2ème mot extension: **000F**

AVANT EXECUTION

D2	XX	XX	XX	XX
D3	44	55	66	77

APRES EXECUTION

D2	XX	XX	XX	XX
D3	00	00	00	07

X	N	Z	V	C
0	0	0	0	0

X	N	Z	V	C
0	0	0	0	0

ANDI to CCR ET LOGIQUE IMMEDIAT AVEC CCR

OPERATION: DONNEE IMMEDIATE ET CCR dans CCR

SYNTAXE ASSEMBLEUR:

ANDI # Donnée,CCR

PORTEE:

Octet

FONCTION:

Effectue un ET logique entre la donnée immédiate et le registre de codes condition, résultat dans la partie basse du registre d'état (CCR).

CODES CONDITION:

X	N	Z	V	C
A	A	A	A	A

X mis à 0 si le bit correspondant de la donnée est à 0 sinon inchangé
N, Z, V, C identiques à X

FORMAT DU MOT INSTRUCTION 0X023C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	1	0	0	0	1	1	1	1	0	0

Le mot instruction est suivi d'un mot extension dont l'octet de poids faible contient la donnée immédiate . L'octet de poids fort est mis à 0.

ANDI to SR ET LOGIQUE IMMEDIAT AVEC SR

INSTRUCTION PRIVILEGIEE

OPERATION: Si le processeur est dans l'état superviseur
DONNEE IMMEDIATE ET SR dans SR
Si non TRAP

SYNTAXE ASSEMBLEUR:

ANDI #Donnée,SR

PORTEE:

Mot

FONCTION:

Si le microprocesseur est dans le mode superviseur, effectue un ET logique entre la donnée immédiate et le registre d'état, résultat dans le registre d'état (SR). Si le microprocesseur n'est pas dans le mode superviseur déclenchement d'une exception viol de privilège (Vecteur N° 8)

CODES CONDITION:

X	N	Z	V	C
A	A	A	A	A

X mis à 0 si le bit correspondant de la donnée est à 0 sinon inchangé
N, Z, V, C identiques à X

FORMAT DU MOT INSTRUCTION 0X027C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	1	0	0	1	1	1	1	1	0	0

Le mot instruction est suivi d'un mot extension qui contient la donnée immédiate .

ASL DECALAGE ARITHMETIQUE ASL ASR (ARITHMETIC SHIFT LEFT, RIGHT) ASR

OPERATION: DESTINATION DECALEE DE <NOMBRE> dans DESTINATION

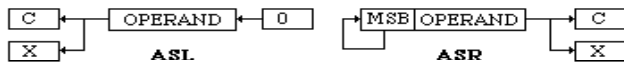
SYNTAXE ASSEMBLEUR:

ASL	Dx,Dy	ASR	Dx,Dy
ASL	#Donnée,Dy	ASR	#Donnée,Dy
ASL	<AE>	ASR	<AE>

PORTEE:

Octet (.B), Mot (.W), Mot long (.L)

FONCTION:



ASL décalage arithmétique à gauche, ASR à droite suivant les croquis ci-dessus. Le nombre de décalages peut être précisé de deux façons:

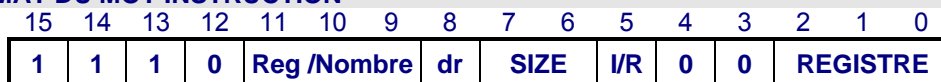
- 1/ Le nombre de décalage est précisé dans l'instruction , de 1 à 8.
 - 2/ Le nombre est contenu dans un registre spécifié dans l'instruction le décalage est modulo 64.
- Le décalage d'une case mémoire ne peut être que d'une unité.

CODES CONDITION:

X	N	Z	V	C
A	A	A	A	A

X et C	Voir les croquis ci-dessus	Z	mis à 1 si résultat nul sinon mis à 0
N	affiche le signe du résultat (bit de poids fort)	V	mis à 1 si le MSB a été modifié pendant décalage sinon 0

FORMAT DU MOT INSTRUCTION



Champ **Reg/Nombre** Si I/R = 0 ce champ précise le nombre de décalages = 1 ce champ indique le N° de Dx

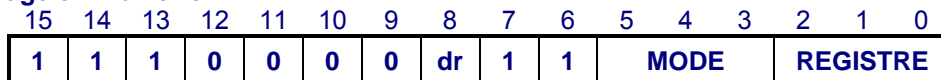
Champ **dr** indique le sens du décalage: 0 à droite 1 à gauche

Champ **SIZE** Octet :00 Mot :01 Mot long : 10

Champ **I/R** 0 décalage immédiat 1 Décalage par registre (Dx)

REGISTRE N° de Dy

Décalage en mémoire:



← Adresse Effective →

Le champ **Adresse effective** est codé, en fonction de l'adressage utilisé, suivant le tableau ci dessous, dr est identique au précédent.

NOTA	MODE D'ADRESS	MODE	REGISTRE	NOTA	MODE D'ADRESS	MODE	REGISTRE
(3)	Dn				d(An,Xi)	110	N° Registre
(3)	An				Absolu Court	111	000
	(An)	010	N° Registre		Absolu Long	111	001
	(An)+	011	N° Registre	(3)	d(PC)		
	-(An)	100	N° Registre	(3)	d(PC,Xi)		
	d(An)	101	N° Registre	(3)	Immédiat		

NOTA (3) Mode d'adressage ne pouvant être utilisé

EXEMPLE D'UTILISATION ASR.B

#1,D3

mot instruction: **0XE203**

AVANT EXECUTION

APRES EXECUTION

D3

44	55	66	83	
X	N	Z	V	C
0	0	0	0	0

D3

44	55	66	C1	
X	N	Z	V	C
1	1	0	0	1

BCC à BVS**BCC à BVS****BRANCHEMENT CONDITIONNEL
(BRANCH CONDITIONALLY)****OPERATION:** SI CONDITION VRAIE ALORS: PC + d dans PC**SYNTAXE ASSEMBLEUR:**

Bcc <Etiquette>

PORTEE:

Octet , Mot , Mot long

FONCTION:

Si la condition spécifiée (cc) voir ci-dessous est vérifiée alors le branchement est effectué (PC + d dans PC) le déplacement est calculé par l'assembleur , il correspond au nombre d'octets en complément à 2 à sauter, la valeur de PC est l'adresse de l'instruction plus 2

Conditions de Branchement

Précisions sur le branchement basé sur le résultat d'une comparaison ou une soustraction

INST	CONDITION	Codage	si résultat DEST - SOURCE	Système de Numération	Condition booléenne
BCC	Carry Clear	0100	≥ à 0	Bin Nat	C = 0
BCS	Carry Set	0101	< à 0	Bin Nat	C = 1
BEQ	Equal	0111	= 0	Quelconque	Z = 1
BGE	Greater or Equal	1100	≥ à 0	Comp à 2	N ⊕ V = 0
BGT	Greater Than	1110	> à 0	Comp à 2	Z nor (N ⊕ V) = 1
BHI	High	0010	> à 0	Bin Nat	C ou Z = 0
BLE	Less or Equal	1111	≤ à 0	Comp à 2	Z ou N ⊕ V = 1
BLS	Lower or Same	0011	≤ à 0	Bin Nat	C ou Z = 1
BLT	Less Than	1101	< à 0	Comp à 2	N ⊕ V = 1
BMI	Minus	1011	Négatif	Quelconque	N = 1
BNE	Not Equal	0110	Différent de 0	Quelconque	Z = 0
BPL	Plus	1010	Positif	Quelconque	N = 0
BVC	Overflow Clear	1000	Signe Cohérent	Comp à 2	V = 0
BVS	Overflow Set	1001	Signe non cohérent	Comp à 2	V = 1

CODES CONDITION:

X	N	Z	V	C
N	N	N	N	N

Le registre de Codes condition n'est pas affecté par cette instruction

FORMAT DU MOT INSTRUCTION

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	0	CONDITION				DEPLACEMENT							

Si le déplacement tient sur 8 bits il est logé dans le champ déplacement du mot instruction.
Si le déplacement tient sur 16 bits (et ne tient pas sur 8) le champ déplacement est mis à \$00 et le déplacement est logé dans un mot extension.

La suite de la description de cette instruction est reportée dans le bas de la page suivante

BRA**BRA**

BRANCHEMENT INCONDITIONNEL (BRANCH ALWAYS)

OPERATION: PC + d dans PC

SYNTAXE ASSEMBLEUR:

BRA <Etiquette>

PORTEE:

Octet , Mot , Mot long

FONCTION:

Lorsque le microprocesseur rencontre cette instruction, le branchement est toujours effectué (PC + d dans PC). Le déplacement est calculé par l'assembleur , il correspond au nombre d'octets en complément à 2 à sauter, la valeur de PC est l'adresse de l'instruction plus 2

CODES CONDITION:

X	N	Z	V	C
N	N	N	N	N

Le registre de Codes condition n'est pas affecté par cette instruction

FORMAT DU MOT INSTRUCTION

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	0	0	0	0	0	DEPLACEMENT							

Si le déplacement tient sur 8 bits il est logé dans le champ déplacement du mot instruction.

Si le déplacement tient sur 16 bits (et ne tient pas sur 8) le champ déplacement est mis à \$00 et le déplacement est logé dans un mot extension.

Suite de **BCC** à **BVS**

CONDITION		TEST	CONDITION		TEST
CC	Carry Clear	$C = 0$	LS	Lower or Same	$C + Z = 0$
CS	Carry Set	$C = 1$	LT	Less Than	$N \oplus V = 0$
EQ	Equal	$Z = 1$	MI	Minus	$N = 1$
GE	Greater or Equal	$N \oplus V = 0$	NE	Not Equal	$Z = 0$
GT	Greater Than	$(N \oplus V) + Z = 0$	PL	Plus	$N = 0$
HI	High	$C + Z = 0$	VC	Overflow Clear	$V = 0$
LE	Less or Equal	$(N \oplus V) + Z = 1$	VS	Overflow Set	$V = 1$

BCHG - BCLR – BSET- BTST TESTE UN BIT ET : (BCHG)CHANGE, (BCLR)MET A 0, (BSET)MET A 1, (BTST) NE CHANGE PAS

OPERATION: Teste le bit concerné dans le mot long d'un registre de données ou dans un octet mémoire, en copie le complément dans le bit Z de CCR puis suivant l'instruction:

BCHG Remplace ce bit par son complément (TEST BIT AND CHANGE)
 BCLR Remplace ce bit par 0 (TEST BIT AND CLEAR)
 BSET Remplace ce bit par 1 (TEST BIT AND SET)
 BTST Ne modifie pas le bit testé (TEST BIT)

SYNTAXE ASSEMBLEUR:

BCHG Dn,<AE>	BCHG <N°>,<AE>
BCLR Dn,<AE>	BCLR <N°>,<AE>
BSET Dn,<AE>	BSET <N°>,<AE>
BTST Dn,<AE>	BTST <N°>,<AE>

PORTEE:

Octet en mémoire
 Mot long dans un registre (AE = Dn)

CODES CONDITION:

X	N	Z	V	C
N	N	A	N	N

Seul le bit Z est influencé dans le registre de Codes condition suivant la description ci dessus.

FORMAT DU MOT INSTRUCTION

Le N° du bit à tester est dans un registre Dn

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	Dn			1	INS	MODE			REGISTRE			

← Adresse Effective →

Le N° du bit à tester est fourni en donnée immédiate

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	1	0	0	0	INS	MODE			REGISTRE			

← Adresse Effective →

dans ce dernier cas le N° est dans un mot extension Octet Haut \$00 Octet Bas N° du bit modulo 32
 Le champ INS dépend de l'instruction

BCHG	BCLR	BSET	BTST
01	10	11	00

Le champ Dn reçoit le N° du registre Dn utilisé

Le champ **Adresse effective** est codé, en fonction de l'adressage utilisé, suivant le tableau ci dessous:

NOTA	MODE D'ADRESS	MODE	REGISTRE	NOTA	MODE D'ADRESS	MODE	REGISTRE
(4)	Dn	000	N° Registre		d(An,Xi)	110	N° Registre
(3)	An				Absolu Court	111	000
	(An)	010	N° Registre		Absolu Long	111	001
	(An)+	011	N° Registre	(3)	d(PC)		
	-(An)	100	N° Registre	(3)	d(PC,Xi)		
	d(An)	101	N° Registre	(3)	Immédiat		

NOTA (3) Ce mode d'adressage ne peut être utilisé

(4) Seul mode portant sur un mot long les autres modes portent sur un octet seul

VOIR LES EXEMPLES D'APPLICATION EN BAS DE LA PAGE SUIVANTE

BSR BSR

BRANCHEMENT A UN SOUS-PROGRAMME (BRANCH TO SUBROUTINE)

OPERATION:

$$(PC) \rightarrow -(SP) \quad (PC) + d \rightarrow PC$$
SYNTAXE ASSEMBLEUR:

BSR <Étiquette>

PORTEE:

Short(.S), Long(.L).

FONCTION:

Le contenu de PC, adresse de l'instruction suivante, est sauvé dans la pile active (utilisateur ou superviseur suivant l'état du bit S), puis PC est chargé avec le résultat de (PC) + déplacement

Le déplacement en code complément à deux peut porter sur ± 127 octets, le déplacement se code alors sur un octet et ne nécessite pas de mot extension ce sera un BSR.S ou sur ± 32 k Octets avec un BSR.L. Dans ce dernier cas, le déplacement calculé par l'assembleur est logé dans un mot extension et le champ réservé au déplacement dans le mot instruction est mis à 0

CODES CONDITION:

X	N	Z	V	C
N	N	N	N	N

Le registre de Code Condition n'est pas influencé par cette instruction

FORMAT DU MOT INSTRUCTION

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	0	0	0	0	1	DEPLACEMENT							

EXEMPLES D'UTILISATION DES INSTRUCTIONS DE LA PAGE PRECEDENTE

BCLR #7,D3 mot instruction: **0X0883** mot extension: **0X0007**

AVANT EXECUTION

D2	XX	XX	XX	XX
D3	44	55	FF	FF

APRES EXECUTION

D2	XX	XX	XX	XX
D3	44	55	FF	7F

X	N	Z	V	C
0	0	0	0	0

X	N	Z	V	C
0	0	0	0	0

BCHG D2,D3 mot instruction: **0X0543**

AVANT EXECUTION

D2	00	00	00	0F
D3	44	55	00	FF

APRES EXECUTION

D2	00	00	00	0F
D3	44	55	80	FF

X	N	Z	V	C
0	0	0	0	0

X	N	Z	V	C
0	0	1	0	0

CHK

CHK

TEST ENTRE DEUX LIMITES (CHECK REGISTER AGAINST BOUNDS)

OPERATION:

Si Dn<0 ou Dn>source alors Trappe

SYNTAXE ASSEMBLEUR:

CHK <AE>,Dn

PORTEE:

Mot(.W)

FONCTION:

Le contenu d'un registre de données spécifié dans l'instruction est testé entre deux limites. Si le contenu du registre en complément à 2 est supérieur à la source ou inférieur à 0 alors une exception CHK est initiée vecteur N° 6

CODES CONDITION:

X	N	Z	V	C
N	A	?	?	?

- X Non affecté
- N Mis à 1 si Dn<0 mis à 0 si Dn>source Autrement indéfini
- Z Indéfini
- V Indéfini
- C Indéfini

FORMAT DU MOT INSTRUCTION

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	REG Dn			1	1	0	MODE			REGISTRE		

← Adresse Effective →

Le champ Registre Dn précise le registre de données qui contient l'opérande à tester.

Le champ **Adresse effective** est codé, en fonction de l'adressage utilisé, suivant le tableau ci dessous

NOTA	MODE D'ADRESS	MODE	REGISTRE	NOTA	MODE D'ADRESS	MODE	REGISTR E
	Dn	000	N° Registre		d(An,Xi)	110	N° Registre
(3)	An				Absolu Court	111	000
	(An)	010	N° Registre		Absolu Long	111	001
	(An)+	011	N° Registre		d(PC)	111	010
	-(An)	100	N° Registre		d(PC,Xi)	111	011
	d(An)	101	N° Registre		Immédiat	111	100

NOTA (3) ce mot d'adressage ne peut pas être utilisé

EXEMPLE D'UTILISATION

CHK D2,D3
mot instruction: 0X4702

AVANT EXECUTION

D2	XX	XX	22	33
----	----	----	----	----

APRES EXECUTION

D2	XX	XX	22	33
----	----	----	----	----

CAS	Contenu de D3	Trappe	Bit N
1er CAS (0<D3<D2)	XX XX 11 22	NON	N = indéfini
2ème CAS (D3>D2)	XX XX 33 22	OUI	N = 0
3ème CAS (D3<0)	XX XX 88 22	OUI	N = 1

CLR**CLR****MISE A 0 D'UN OPERANDE
(CLEAR AN OPERAND)****OPERATION:**

0X00 dans Destination

SYNTAXE ASSEMBLEUR:

CLR <AE>

PORTEE:

Octet (.B), Mot(.W), Mot long(.L).

FONCTION:

La destination est mise à 0

CODES CONDITION:

X	N	Z	V	C
N	0	1	0	0

X Non affecté
 N Toujours mis à 0
 Z Toujours mis à 1
 V Toujours mis à 0
 C Toujours mis à 0

FORMAT DU MOT INSTRUCTION

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	0	0	1	0	SIZE	MODE	REGISTRE					
										← Adresse Effective →					

Le champ SIZE spécifie la taille de l'opération

CHAMP SIZE			
OCTET	MOT	MOT LONG	OPERATION
00	01	10	0 dans Destination

Le champ **Adresse effective** est codé, en fonction de l'adressage utilisé, suivant le tableau ci dessous

NOTA	MODE D'ADRESS	MODE	REGISTRE	NOTA	MODE D'ADRESS	MODE	REGISTRE
	Dn	000	N° Registre		d(An,Xi)	110	N° Registre
(3)	An				Absolu Court	111	000
	(An)	010	N° Registre		Absolu Long	111	001
	(An)+	011	N° Registre	(3)	d(PC)		
	-(An)	100	N° Registre	(3)	d(PC,Xi)		
	d(An)	101	N° Registre	(3)	Immédiat		

NOTA (3) Ce mode d'adressage ne peut être utilisé

EXEMPLE D'UTILISATION**CLR.W D3**mot instruction: **0X4243****AVANT EXECUTION**

D2	XX	XX	XX	XX
D3	44	55	66	77

X	N	Z	V	C
1	1	0	1	1

APRES EXECUTION

D2	XX	XX	XX	XX
D3	44	55	00	00

X	N	Z	V	C
1	0	1	0	0

CMP

COMPARAISON (COMPARE)

CMP

OPERATION:

Destination - Source

SYNTAXE ASSEMBLEUR:

CMP <AE>,Dn

PORTEE:

Octet (.B), Mot(.W), Mot long(.L).

FONCTION:

Soustrait l'opérande source au contenu d'un registre de données et positionne CCR en fonction des résultats obtenus. Le contenu du registre Dn est inchangé (c'est ce qui fait différer cette instruction de la soustraction)

CODES CONDITION:

X	N	Z	V	C
N	A	A	A	A

- X Non affecté
- N Copie du bit de poids fort du résultat de la soustraction
- Z Mis à 1 si le résultat de la soustraction donne 0
- V Mis à 1 si le résultat de la soustraction donne une incohérence de signe
- C Mis à 1 si le résultat de la soustraction génère un report

FORMAT DU MOT INSTRUCTION

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	1	1	REG Dn			MODE OP			MODE			REGISTRE		

" ← Adresse Effective →

Le champ **Registre Dn** le registre de données utilisé

Le champ **Op Mode** dépend de la taille de l'opération

CHAMP MODE OP			
OCTET	MOT	MOT LONG	OPERATION
000	001	010	<Dn> - <AE> positionne CCR

Le champ **Adresse effective** est codé, en fonction de l'adressage utilisé, suivant le tableau ci dessous

NOTA	MODE D'ADRESS	MODE	REGISTRE	NOTA	MODE D'ADRESS	MODE	REGISTRE
	Dn	000	N° Registre		d(An,Xi)	110	N° Registre
(1)	An	001	N° Registre		Absolu Court	111	000
	(An)	010	N° Registre		Absolu Long	111	001
	(An)+	011	N° Registre		d(PC)	111	010
	-(An)	100	N° Registre		d(PC,Xi)	111	011
	d(An)	101	N° Registre		Immédiat	111	100

NOTA (1) Ce mode d'adressage n'accepte pas les opérandes octet

Lorsque la destination est un registre d'adresses CMPA est à utiliser, lorsqu'une donnée immédiate est utilisée comme comparaison l'instruction devient CMPI et entre mémoires CMPM. Certains assembleurs le font automatiquement.

EXEMPLE D'UTILISATION

CMP.W

AVANT EXECUTION

D2	00	00	77	00
D3	44	55	66	77

D2,D3 mot instruction: **0XB642**

APRES EXECUTION

D2	00	00	77	00
D3	44	55	66	77

X	N	Z	V	C
0	0	0	0	0

X	N	Z	V	C
0	1	0	0	1

CMPA

CMPA

COMPARAISON D'ADRESSES (COMPARE ADDRESS)

OPERATION:

Destination - Source

SYNTAXE ASSEMBLEUR:

CMPA <AE>,An

PORTEE:

Mot(.W), Mot long(.L).

FONCTION:

Soustrait l'opérande source au contenu d'un registre d'adresses et positionne CCR en fonction des résultats obtenus. Le contenu du registre An est inchangé (c'est ce qui fait différer cette instruction de la soustraction).Lorsque l'opération est spécifiée sur un mot, l'opérande de comparaison voit son signe étendu à 32 bits et la comparaison se fait sur toute l'étendue du registre d'adresses.

CODES CONDITION:

X	N	Z	V	C
N	A	A	A	A

X Non affecté

N Copie du bit de poids fort du résultat de la soustraction

Z Mis à 1 si le résultat de la soustraction donne 0

V Mis à 1 si le résultat de la soustraction donne une incohérence de signe

C Mis à 1 si le résultat de la soustraction génère un report

FORMAT DU MOT INSTRUCTION

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	1	1	REG An				MODE OP			MODE			REGISTRE	

← Adresse Effective →

Le champ Registre An le registre d'adresses utilisé

Le champ Op Mode dépend de la taille de l'opération

CHAMP MODE OP			
OCTET	MOT	MOT LONG	OPERATION
Néant	011	111	<An> - <AE> positionne CCR

Le champ Adresse effective est codé, en fonction de l'adressage utilisé, suivant le tableau ci dessous

NOTA	MODE D'ADRESS	MODE	REGISTRE	NOTA	MODE D'ADRESS	MODE	REGISTRE
	Dn	000	N° Registre		d(An,Xi)	110	N° Registre
	An	001	N° Registre		Absolu Court	111	000
	(An)	010	N° Registre		Absolu Long	111	001
	(An)+	011	N° Registre		d(PC)	111	010
	-(An)	100	N° Registre		d(PC,Xi)	111	011
	d(An)	101	N° Registre		Immédiat	111	100

EXEMPLE D'UTILISATION

CMPA.W

AVANT EXECUTION

D2	00	00	80	00
A3	00	10	00	00

X	N	Z	V	C
0	0	0	0	0

CMPI

D2,A3 mot instruction: **0XB6C2**

APRES EXECUTION

D2	00	00	80	00
A3	00	10	00	00

X	N	Z	V	C
0	0	0	0	1

CMPI

COMPARAISON IMMEDIATE (COMPARE IMMEDIATE)

OPERATION:

Destination - Donnée immédiate

SYNTAXE ASSEMBLEUR:

CMPI #Donnée,<AE>

PORTEE:

Octet (.B), Mot(.W), Mot long(.L).

FONCTION:

Soustrait la donnée immédiate au contenu de l'Adresse Effective et positionne CCR en fonction du résultat obtenu. Le contenu de l'Adresse Effective est inchangé (c'est ce qui fait différer cette instruction de la soustraction).

CODES CONDITION:

X	N	Z	V	C
N	A	A	A	A

- X Non affecté
- N Copie du bit de poids fort du résultat de la soustraction
- Z Mis à 1 si le résultat de la soustraction donne 0
- V Mis à 1 si le résultat de la soustraction donne une incohérence de signe
- C Mis à 1 si le résultat de la soustraction génère un report

FORMAT DU MOT INSTRUCTION



Le mot instruction est suivi d'un ou deux mots extension qui supportent la donnée immédiate
Le champ Size spécifie la taille de l'opération.

CHAMP SIZE			
OCTET	MOT	MOT LONG	OPERATION
00	01	10	Destination - Donnée actualise CCR

Le champ Adresse effective est codé, en fonction de l'adressage utilisé, suivant le tableau ci dessous

NOTA	MODE D'ADRESS	MODE	REGISTRE	NOTA	MODE D'ADRESS	MODE	REGISTRE
	Dn	000	N° Registre		d(An,Xi)	110	N° Registre
(3)	An				Absolu Court	111	000
	(An)	010	N° Registre		Absolu Long	111	001
	(An)+	011	N° Registre	(3)	d(PC)		
	-(An)	100	N° Registre	(3)	d(PC,Xi)		
	d(An)	101	N° Registre	(3)	Immédiat		

NOTA (3) Ce mode d'adressage ne peut être utilisé.

EXEMPLE D'UTILISATION

CMPI.W #0X100,D3

mot instruction: **0X0C43** mot extension: **0X0100**

AVANT EXECUTION

APRES EXECUTION

D2	XX	XX	XX	XX
D3	00	01	00	00

D2	XX	XX	XX	XX
D3	00	01	00	00

X	N	Z	V	C
0	0	0	0	0

X	N	Z	V	C
0	1	0	0	1

CMPM**CMPM****COMPARAISON MEMOIRE
(COMPARE MEMORY)****OPERATION:**

Destination - Source

SYNTAXE ASSEMBLEUR:

CMPM (Ay)+,(Ax)+

PORTEE:

Octet (.B), Mot(.W), Mot long(.L).

FONCTION:

Soustrait l'opérande source de l'opérande destination et positionne CCR en fonction des résultats obtenus. L'opérande destination est inchangé. Cette instruction n'utilise que le mode d'adressage indirect par registre d'adresses post-incrémenté.

CODES CONDITION:

X	N	Z	V	C
N	A	A	A	A

X Non affecté

N Copie du bit de poids fort du résultat de la soustraction

Z Mis à 1 si le résultat de la soustraction donne 0

V Mis à 1 si le résultat de la soustraction donne une incohérence de signe

C Mis à 1 si le résultat de la soustraction génère un report

FORMAT DU MOT INSTRUCTION

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	1	1	Ax			1	SIZE	0	0	1	Ay			

← Destination →

← Source →

Le champ Registre Ax spécifie le registre d'adresses utilisé pour spécifier la destination

Le champ Registre Ay spécifie le registre d'adresses utilisé pour spécifier la source

Le champ définit la taille de l'opération

CHAMP SIZE			
OCTET	MOT	MOT LONG	OPERATION
00	01	10	Destination - Source actualise CCR

EXEMPLE D'UTILISATION**CMPM.L****(A2)+,(A3)+ mot instruction: 0XB78A****MEMOIRES**

ADRESSES	CONTENU
00 01 80 04	2B
00 01 80 03	55
00 01 80 02	44
00 01 80 01	33
00 01 80 00	11
00 01 7F FF	57

ADRESSES	CONTENU
00 01 00 04	AB
00 01 00 03	55
00 01 00 02	44
00 01 00 01	33
00 01 00 00	11
00 00 FF FF	17

AVANT EXECUTION

A2	00	01	80	00
A3	00	01	00	00

APRES EXECUTION

A2	00	01	80	04
A3	00	01	00	04

X	N	Z	V	C
0	0	0	0	0

X	N	Z	V	C
0	0	1	0	0

DBcc à DECREMENTATION ET DBcc DBvs BRANCHEMENT CONDITIONNEL DBvs (TEST CONDITION, DECREMENT, AND BRANCH)

OPERATION:

Si la condition **n'est pas vraie** alors
 Dn est décrémenté, si Dn ≠ -1 alors branchement (PC + d dans PC)
 si Dn = -1 alors (PC + 2 dans PC)
 Si la condition **est vraie** alors pas d'opération (PC + 2 dans PC)

SYNTAXE ASSEMBLEUR:

DBcc Dn,<Étiquette>

PORTEE:

Mot

FONCTION:

Cette instruction permet de réaliser une boucle dont la sortie est obtenue par la condition devenue vraie ou par la décrémentation d'un registre . L'Instruction commence par tester la condition afin de vérifier si la sortie de boucle est obtenue. Si c'est le cas aucune opération n'est réalisée et le MPU passe à l'instruction suivante. Si la condition n'est pas vraie, le registre Dn est décrémenté d'une unité, si celui ci est parvenu à la valeur -1 il y aura également sortie de boucle. Le nombre chargé dans le registre Dn indique donc le nombre maximum (+1) de boucles exécutées ce nombre pouvant être réduit par l'obtention de la condition vraie

Pour le branchement et l'interprétation plus fine des conditions voir l'instruction Bcc.

CONDITION		CODE	TEST	CONDITION		CODE	TEST
CC	Carry Clear	0100	C = 0	LS	Low or Same	0011	C ou Z = 1
CS	Carry Set	0101	C = 1	LT	Less Than	1101	$N \oplus V = 1$
EQ	Equal	0111	Z = 1	MI	Minus	1011	N = 1
GE	Greater or Equal	1100	$N \oplus V = 0$	NE	Not Equal	0110	Z = 0
GT	Greater Than	1110	$(N \oplus V)$ ou Z=0	PL	Plus	1010	N = 0
HI	High	0010	C ou Z = 0	VC	Overflow Clear	1000	V = 0
LE	Less or Equal	1111	$(N \approx V)$ ou Z=1	VS	Overflow set	1001	V = 1

CODES CONDITION:

X	N	Z	V	C
N	N	N	N	N

Le registre de codes condition n'est pas affecté par cette instruction

FORMAT DU MOT INSTRUCTION

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	1	CONDITION			1	1	0	0	1	REGISTRE			

Le mot instruction est suivi d'un mot contenant le déplacement à ajouter à PC pour effectuer le branchement.

Le champ **condition** indique la valeur prise dans le tableau ci-dessus

Le champ **registre** précise le N° du registre de données employé comme compteur

NOTES:

L'obtention de la condition peut être interprétée comme l'expression UNTIL des langages évolués par exemple Décrémente et branche UNTIL (jusqu'à) Minus(Négatif)

Si aucune condition n'est requise, seulement n boucles à exécuter, on pourra prendre la (non)condition DBF, certains assembleurs acceptent DBRA, ainsi, seul le passage à -1 de Dn permettra la sortie de boucle.

DIVS DIVISION SIGNEE ET NON SIGNEE DIVS DIVU (SIGNED AND UNSIGNED DIVIDE) DIVU

OPERATION:

Destination ÷ Source dans Destination

SYNTAXE ASSEMBLEUR:

DIVS <AE>,Dn

DIVU <AE>,Dn

PORTEE:

Dividende 32 bits

Diviseur Quotient et Reste 16 bits

FONCTION:

Divise l'opérande destination par l'opérande source (AE) et range le résultat dans Destination. L'opération est réalisée en arithmétique **signée (DIVS) non signée (DIVU)**. La Destination est toujours un Registre de Données. Pour DIVS, la division respecte la règle des signes et le reste est donné avec le signe du dividende (voir les exemples de la page suivante)

Division d'un **Mot Long** (Destination) par un **Mot** (Source) Le résultat remplace le dividende. Le mot de poids fort contient le reste et le mot de poids faible le quotient

Une division par **ZERO** entraîne le déclenchement d'une exception (Vecteur N°5)

Si le quotient ne tient pas dans l'emplacement qui lui est réservé, le bit d'**Overflow** est positionné mais l'opération n'est pas exécutée et les opérandes sont inchangés

CODES CONDITION:

X	N	Z	V	C
N	A	A	A	0

X Non affecté

N Mis à 1 si le quotient est négatif, 0 autrement, indéfini si overflow

Z Mis à 1 si le quotient = 0 , 0 autrement, indéfini si overflow ou div par 0

V Mis à 1 si le quotient dépasse la capacité d'accueil 0 autrement

C Toujours mis à 0

FORMAT DU MOT INSTRUCTION

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	0	0	REG Dn			S	1	1	MODE			REGISTRE		

← Adresse Effective →

Le champ **Reg Dn** précise le registre de données qui contient l'opérande à diviser (Dest)Le bit **S** = 1 pour division signée, 0 pour non signéeLe champ **Adresse effective** qui précise le diviseur est codé, en fonction de l'adressage utilisé, suivant le tableau ci dessous

NOTA	MODE D'ADRESS	MODE	REGISTRE	NOTA	MODE D'ADRESS	MODE	REGISTR E
	Dn	000	N° Registre		d(An,Xi)	110	N° Registre
(3)	An				Absolu Court	111	000
	(An)	010	N° Registre		Absolu Long	111	001
	(An)+	011	N° Registre		d(PC)	111	010
	-(An)	100	N° Registre		d(PC,Xi)	111	011
	d(An)	101	N° Registre		Immédiat	111	100

NOTA (3) Ce mode d'adressage ne peut être utilisé

VOIR LES EXEMPLES D'APPLICATION PAGE SUIVANTE

EXEMPLES D'UTILISATION DES INSTRUCTIONS DIVU ET DIVS

1er EXEMPLE DIVU D2,D3
mot instruction: **0X86C2**

AVANT EXECUTION

D2	00	00	00	02
D3	00	00	00	05

APRES EXECUTION

D2	00	00	00	02
D3	00	01	00	02

← Reste → ← Quotient →

X	N	Z	V	C
0	0	0	0	0

X	N	Z	V	C
0	0	0	0	0

2ème EXEMPLE DIVU D2,D3

AVANT EXECUTION

D2	00	00	00	02
D3	00	40	00	05

APRES EXECUTION

D2	00	00	00	02
D3	00	40	00	05

Le résultat ne tient pas sur 16 bits le dividende n'est pas modifié et le bit V est mis à 1

X	N	Z	V	C
0	0	0	0	0

X	N	Z	V	C
0	0	0	1	0

3ème EXEMPLE DIVS D2,D3
mot instruction: **0X87C2**

AVANT EXECUTION

D2	00	00	00	02
D3	00	00	00	05

APRES EXECUTION

D2	00	00	00	02
D3	00	01	00	02

← Reste → ← Quotient →

(division de +5 par +2 , quotient +2 reste +1)

X	N	Z	V	C
0	0	0	0	0

X	N	Z	V	C
0	0	0	0	0

4ème EXEMPLE DIVS D2,D3

AVANT EXECUTION

D2	00	00	00	02
D3	FF	FF	FF	FB

APRES EXECUTION

D2	00	00	00	02
D3	FF	FF	FF	FE

← Reste → ← Quotient →

(division de - 5 (FFFFFFFB) par +2 , quotient - 2 (FFFE) reste - 1 (FFFF))

X	N	Z	V	C
0	0	0	0	0

X	N	Z	V	C
0	1	0	0	0

5ème EXEMPLE DIVS D2,D3

AVANT EXECUTION

D2	00	00	FF	FE
D3	00	00	00	05

APRES EXECUTION

D2	00	00	FF	FE
D3	00	01	FF	FE

← Reste → ← Quotient →

(division de +5 par - 2 (FFFE), quotient - 2 (FFFE) reste + 1)

X	N	Z	V	C
0	0	0	0	0

X	N	Z	V	C
0	1	0	0	0

6ème EXEMPLE DIVS D2,D3

AVANT EXECUTION

D2	00	00	FF	FE
D3	FF	FF	FF	FB

APRES EXECUTION

D2	00	00	FF	FE
D3	FF	FF	00	02

← Reste → ← Quotient →

(division de - 5 (FFFFFFFB) par - 2 (FFFE), quotient +2 reste - 1 (FFFF))

X	N	Z	V	C
0	0	0	0	0

X	N	Z	V	C
0	0	0	0	0

EOR OU EXCLUSIF (DILEME) (EXCLUSIVE OR LOGICAL) EOR

OPERATION:

Source ⊕ Destination → Destination

SYNTAXE ASSEMBLEUR:

EOR Dn,<AE>

PORTEE:

Octet(.B), Mot(.W), Mot long(.L).

FONCTION:

OU exclusif bit à bit entre la source qui est toujours un registre de Données et le contenu de l'adresse effective résultat dans celle-ci.

CODES CONDITION:

X	N	Z	V	C
N	A	A	0	0

- X Non affecté
- N Copie du bit de poids fort du résultat
- Z Mis à un si le résultat = 00
- V Mis à 0
- C Mis à 0

FORMAT DU MOT INSTRUCTION

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	1	1	REG Dn		MODE OP		MODE		REGISTRE					

← Adresse Effective →

Le champ **Registre Dn** précise le registre de données qui contient l'opérande source

Le champ **Op Mode** spécifie la taille de l'opération

CHAMP MODE OP			
OCTET	MOT	MOT LONG	OPERATION
100	101	110	<Dn> ⊕ <AE> dans <AE>

Le champ **Adresse effective** est codé, en fonction de l'adressage utilisé, suivant le tableau ci dessous

NOTA	MODE D'ADRESS	MODE	REGISTRE	NOTA	MODE D'ADRESS	MODE	REGISTRE
	Dn	000	N° Registre		d(An,Xi)	110	N° Registre
(3)	An				Absolu Court	111	000
	(An)	010	N° Registre		Absolu Long	111	001
	(An)+	011	N° Registre	(3)	d(PC)		
	-(An)	100	N° Registre	(3)	d(PC,Xi)		
	d(An)	101	N° Registre	(3)	Immédiat		

NOTA (3) Ce mode d'adressage ne peut être utilisé

EXEMPLE D'UTILISATION

EOR.B
mot instruction:
D2,D3
0XB503

AVANT EXECUTION

D2	00	00	00	F0
D3	11	22	33	44

X	N	Z	V	C
1	0	0	0	1

APRES EXECUTION

D2	00	00	00	F0
D3	11	22	33	B4

X	N	Z	V	C
1	0	0	0	0

EORI

EORI

OU EXCLUSIF IMMEDIAT (DILEME) (EXCLUSIVE OR IMMEDIATE)

OPERATION:

Donnée immédiate \approx Destination \oplus Destination

SYNTAXE ASSEMBLEUR:

EORI #Donnée,<AE>

PORTEE:

Octet(.B), Mot(.W), Mot long(.L).

FONCTION:

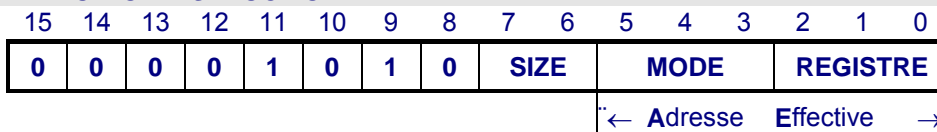
OU exclusif bit à bit entre la Donnée Immédiate et le contenu de l'adresse effective résultat dans celle-ci. La taille de la donnée immédiate doit correspondre à la portée de l'instruction

CODES CONDITION:

X	N	Z	V	C
N	A	A	0	0

- X Non affecté
- N Copie du bit de poids fort du résultat
- Z Mis à un si le résultat = 00
- V Mis à 0
- C Mis à 0

FORMAT DU MOT INSTRUCTION



Suivi d'un ou deux mots contenant la donnée immédiate
Le champ **SIZE** spécifie la taille de l'opération

CHAMP SIZE			
OCTET	MOT	MOT LONG	OPERATION
00	01	10	Donnée \approx Destination dans Destination

Le champ **Adresse effective** est codé, en fonction de l'adressage utilisé, suivant le tableau ci dessous

NOTA	MODE D'ADRESS	MODE	REGISTRE	NOTA	MODE D'ADRESS	MODE	REGISTRE
	Dn	000	N° Registre		d(An,Xi)	110	N° Registre
(3)	An				Absolu Court	111	000
	(An)	010	N° Registre		Absolu Long	111	001
	(An)+	011	N° Registre	(3)	d(PC)		
	-(An)	100	N° Registre	(3)	d(PC,Xi)		
	d(An)	101	N° Registre	(3)	Immédiat		

NOTA (3) Ce mode d'adressage ne peut être utilisé

EXEMPLE D'UTILISATION

EORI.W #0X100,D3 mot instruction: **0X0A43** mot extension: **0X0100**

<p>AVANT EXECUTION</p> <table border="1" style="width: 100%; text-align: center;"> <tr> <td>D2</td><td>XX</td><td>XX</td><td>XX</td><td>XX</td> </tr> <tr> <td>D3</td><td>00</td><td>00</td><td>01</td><td>00</td> </tr> </table>	D2	XX	XX	XX	XX	D3	00	00	01	00	<p>APRES EXECUTION</p> <table border="1" style="width: 100%; text-align: center;"> <tr> <td>D2</td><td>XX</td><td>XX</td><td>XX</td><td>XX</td> </tr> <tr> <td>D3</td><td>00</td><td>00</td><td>00</td><td>00</td> </tr> </table>	D2	XX	XX	XX	XX	D3	00	00	00	00
D2	XX	XX	XX	XX																	
D3	00	00	01	00																	
D2	XX	XX	XX	XX																	
D3	00	00	00	00																	

<table border="1" style="width: 100%; text-align: center;"> <tr> <td>X</td><td>N</td><td>Z</td><td>V</td><td>C</td> </tr> <tr> <td>0</td><td>0</td><td>0</td><td>0</td><td>0</td> </tr> </table>	X	N	Z	V	C	0	0	0	0	0	<table border="1" style="width: 100%; text-align: center;"> <tr> <td>X</td><td>N</td><td>Z</td><td>V</td><td>C</td> </tr> <tr> <td>0</td><td>0</td><td>1</td><td>0</td><td>0</td> </tr> </table>	X	N	Z	V	C	0	0	1	0	0
X	N	Z	V	C																	
0	0	0	0	0																	
X	N	Z	V	C																	
0	0	1	0	0																	

EORI to CCR

OU EXCLUSIF IMMEDIAT AVEC CCR

OPERATION: DONNEE IMMEDIATE \oplus CCR dans CCR

SYNTAXE ASSEMBLEUR:

EORI #Donnée,CCR

PORTEE:

Octet

FONCTION:

Effectue un OU Exclusif entre la donnée immédiate et le registre de codes condition, résultat dans la partie basse du registre d'état (CCR).

CODES CONDITION:

X	N	Z	V	C
A	A	A	A	A

X Complémenté si le bit correspondant de la donnée est à 1 sinon inchangé

N,Z,V,C identiques à X

FORMAT DU MOT INSTRUCTION (0X0A3C)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	1	0	1	0	0	0	1	1	1	1	0	0

Le mot instruction est suivi d'un mot extension dont l'octet de poids faible contient la donnée immédiate . L'octet de poids fort est mis à 0.

EORI to SR OU EXCLUSIF IMMEDIAT AVEC SR

INSTRUCTION PRIVILEGIEE

OPERATION: Si le processeur est dans l'état superviseur
DONNEE IMMEDIATE \oplus SR dans SR
Si non TRAP

SYNTAXE ASSEMBLEUR:

EORI #Donnée,SR

PORTEE:

Mot

FONCTION:

Si le microprocesseur est dans le mode superviseur, effectue un OU Exclusif entre la donnée immédiate et le registre d'état, résultat dans le registre d'état (SR). Si le microprocesseur n'est pas dans le mode superviseur déclenchement d'une exception viol de privilège (Vecteur N° 8)

CODES CONDITION:

X	N	Z	V	C
A	A	A	A	A

X Complémenté si le bit correspondant de la donnée est à 1 sinon inchangé

N,Z,V,C identiques à X

FORMAT DU MOT INSTRUCTION (0X0A7C)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	1	0	1	0	0	1	1	1	1	1	0	0

Le mot instruction est suivi d'un mot extension qui contient la donnée immédiate .

EXG

EXG

ECHANGE DE REGISTRES (EXCHANGE REGISTERS)

OPERATION:

Rx ↔ Ry

SYNTAXE ASSEMBLEUR:

EXG Dx,Dy
EXG Ax,Ay
EXG Dx,Ay

PORTEE:

Mot long(.L).

FONCTION:

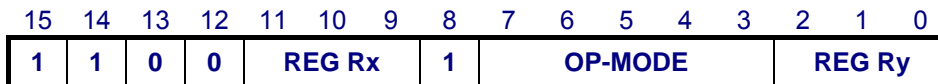
Échange le contenu de deux registres, cet échange porte sur les 32 bits . Il peut se faire entre deux registres de données , d'adresse ou données et adresse.

CODES CONDITION:

X	N	Z	V	C
N	N	N	N	N

Le Registre de Codes Condition n'est pas affecté par cette instruction.

FORMAT DU MOT INSTRUCTION



Les champs **Registre Rx et Ry** précisent les N° des registres concernés
Le champ **Op Mode** spécifie le mode d'échange

Dx,Dy	Ax,Ay	Dx,Ay
01000	01001	10001

EXEMPLE D'UTILISATION

EXG D2,D3
mot instruction: **0XC543**

AVANT EXECUTION

D2	11	22	33	44
D3	55	66	77	88

APRES EXECUTION

D2	55	66	77	88
D3	11	22	33	44

X	N	Z	V	C
0	0	0	0	0

X	N	Z	V	C
0	0	0	0	0

EXT**EXT****EXTENSION DU SIGNE
(SIGN EXTEND)****OPERATION:**

Extension du signe à 16 ou 32 bits

SYNTAXE ASSEMBLEUR:

EXT.W Dn Extension d'Octet à Mot
 EXT.L Dn Extension de Mot à Mot Long

PORTEE:

Mot (.W), Mot Long(.L).

FONCTION:

Etend le bit de signe d'un registre de données d'un octet à un mot, d'un mot à un mot long.

CODES CONDITION:

X	N	Z	V	C
N	A	A	0	0

X Non affecté
 N Copie du bit de poids fort du résultat
 Z Mis à un si le résultat = 0
 V Mis à 0
 C Mis à 0

FORMAT DU MOT INSTRUCTION

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	1	0	0	Op-Mode			0	0	0	REG Dn		

Le champ **Registre Dn** précise les N° du registre concernéLe champ **Op Mode** spécifie la taille de l'extension

7 Æ 15	15 Æ 31
010	011

EXEMPLE D'UTILISATION

		EXT.L D3 mot instruction: 0X48C3										
		AVANT EXECUTION				APRES EXECUTION						
D2		XX	XX	XX	XX	D2	XX	XX	XX	XX		
D3		00	00	80	00	D3	FF	FF	80	00		
		X	N	Z	V	C		X	N	Z	V	C
		0	0	0	0	0		0	1	0	0	0

ILLEGAL**ILLEGAL**

EXECUTER UNE EXCEPTION TRAPPE INSTRUCTION ILLEGALE (TAKE ILLEGAL INSTRUCTION TRAP)

OPERATION:

SSP - 4 dans SSP puis PC DANS LA PILE
SSP - 2 dans SSP puis SR DANS LA PILE

SYNTAXE ASSEMBLEUR:

Cette pseudo instruction s'emploie en déposant dans le programme le code de l'instruction

PORTEE:

Sans

FONCTION:

Ce mot instruction déclenche une exception Instruction Illégale (N° vecteur \$4). Toutes les autres configurations illégales ne doivent pas être utilisées seule cette instruction est garantie ILLEGALE

CODES CONDITION:

X	N	Z	V	C
N	N	N	N	N

Le Registre de Codes Condition n'est pas affecté par cette instruction.

FORMAT DU MOT INSTRUCTION (0X4AFC)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	1	0	1	0	1	1	1	1	1	1	0	0

JMP
JSR

SAUT (JUMP)
SAUT A SOUS PROGRAMME
(JUMP TO SUBROUTINE)

JMP
JSR

OPERATION:

JUMP Adresse de Destination dans PC
JSR SP - 4 dans SP puis PC dans la Pile
Puis Adresse de Destination dans PC

SYNTAXE ASSEMBLEUR:

JMP <AE>
JSR <AE>

PORTEE:

Sans

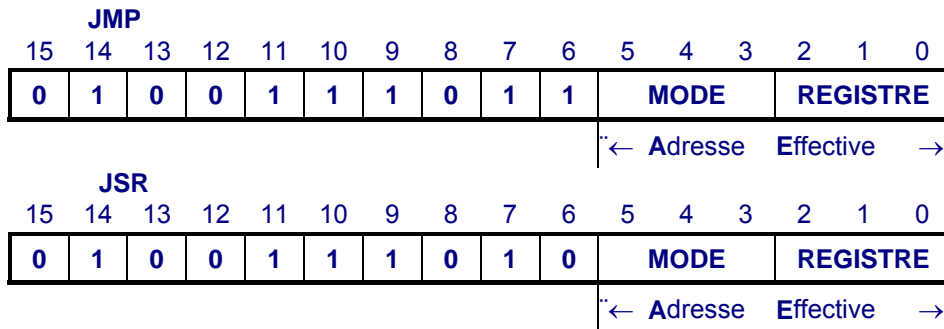
FONCTION:

JMP Le 68000 continue l'exécution du programme à l'adresse indiquée.
JSR Le 68000 continue l'exécution du programme à l'adresse indiquée après avoir sauvegardé l'adresse de retour dans la pile (adresse de l'instruction qui suit le JSR). La pile utilisée dépend de l'état du bit S
S=0 pile utilisateur
S=1 pile superviseur

CODES CONDITION:

X	N	Z	V	C
N	N	N	N	N

Le Registre de Codes Condition n'est pas affecté par ces instructions.

FORMAT DU MOT INSTRUCTION

Le champ **Adresse effective** est codé, en fonction de l'adressage utilisé, suivant le tableau ci dessous

NOTA	MODE D'ADRESS	MODE	REGISTRE	NOTA	MODE D'ADRESS	MODE	REGISTRE
(3)	Dn				d(An,Xi)	110	N° Registre
(3)	An				Absolu Court	111	000
	(An)	010	N° Registre		Absolu Long	111	001
(3)	(An)+				d(PC)	111	010
(3)	-(An)				d(PC,Xi)	111	011
	d(An)	101	N° Registre	(3)	Immédiat		

NOTA (3) Ce mode d'adressage ne peut être utilisé

LEA

LEA

**CHARGEMENT DE L'ADRESSE EFFECTIVE
(LOAD EFFECTIVE ADDRESS)**

OPERATION:

<AE> → An

SYNTAXE ASSEMBLEUR:

LEA <AE>, An

PORTEE:

Mot Long

FONCTION:

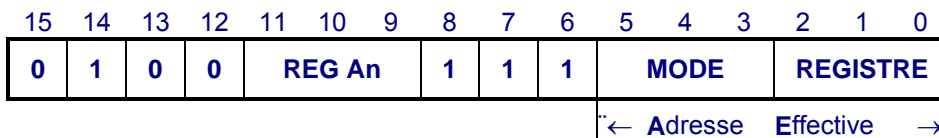
L'adresse effective est chargée dans le registre d'adresse spécifié dans l'instruction. Les 32 bits sont concernés par l'opération

CODES CONDITION:

X	N	Z	V	C
N	N	N	N	N

Le Registre de Codes Condition n'est pas affecté par cette instruction.

FORMAT DU MOT INSTRUCTION



Le champ **Adresse effective** est codé, en fonction de l'adressage utilisé, suivant le tableau ci dessous

NOTA	MODE D'ADRESS	MODE	REGISTRE	NOTA	MODE D'ADRESS	MODE	REGISTR E
(3)	Dn				d(An,Xi)	110	N° Registre
(3)	An				Absolu Court	111	000
	(An)	010	N° Registre		Absolu Long	111	001
(3)	(An)+				d(PC)	111	010
(3)	-(An)				d(PC,Xi)	111	011
	d(An)	101	N° Registre	(3)	Immédiat		

NOTA (3) Ce mode d'adressage ne peut être utilisé

EXEMPLE D'UTILISATION

LEA 4(A3),A3
mot instruction: **0X47EB** mot extension: **0X0004**

AVANT EXECUTION

D2	XX	XX	XX	XX
A3	00	01	00	00

APRES EXECUTION

D2	XX	XX	XX	XX
A3	00	01	00	04

X	N	Z	V	C
0	0	0	0	0

X	N	Z	V	C
0	1	0	0	1

LINK

LINK

**CREATION D'UN LIEN
(LINK AND ALLOCATE)**

OPERATION:

SP - 4 → SP; An → (SP);
SP → An; SP + d → SP;

SYNTAXE ASSEMBLEUR:

LINK An,#<Déplacement>

PORTEE:

Mot, Mot Long

FONCTION:

Le contenu du registre d'adresse spécifié dans l'instruction est sauvé dans la pile puis le contenu du pointeur de pile est chargé dans ce même registre et le déplacement spécifié est ajouté au contenu du pointeur de pile. Si le déplacement est spécifié sur un mot, son signe sera étendu à 32 bits.

Si le déplacement est négatif, ce qui est généralement le cas, il réserve un espace dans la pile qui est situé entre le registre An (ex SP-4) et la nouvelle valeur de SP (SP-4-d) . Cet espace pourra être utilisé pour la communication de paramètres entre programme principal et sous programme.

CODES CONDITION:

X	N	Z	V	C
N	N	N	N	N

Le Registre de Codes Condition n'est pas affecté par cette instruction.

FORMAT DU MOT INSTRUCTION

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	1	1	1	0	0	1	0	1	0	REG An		

Suivi de un ou deux mots contenant le déplacement.

EXEMPLE D'UTILISATION

	LINK	A5,#0XFFFA									
	mot instruction:	0X4E55	mot extension: 0XFFFA								
	AVANT EXECUTION		APRES EXECUTION								
SP	<table border="1"><tr><td>00</td><td>01</td><td>80</td><td>04</td></tr></table>	00	01	80	04	SP	<table border="1"><tr><td>00</td><td>01</td><td>7F</td><td>FA</td></tr></table>	00	01	7F	FA
00	01	80	04								
00	01	7F	FA								
A5	<table border="1"><tr><td>00</td><td>00</td><td>22</td><td>44</td></tr></table>	00	00	22	44	A5	<table border="1"><tr><td>00</td><td>01</td><td>80</td><td>00</td></tr></table>	00	01	80	00
00	00	22	44								
00	01	80	00								

MEMOIRES

ADRESSES	CONTENU		ADRESSES	CONTENU
00 01 80 04	AB	SP →	00 01 80 04	AB
00 01 80 03	XX		00 01 80 03	44
00 01 80 02	XX		00 01 80 02	22
00 01 80 01	XX		00 01 80 01	00
00 01 80 00	XX		00 01 80 00	00
00 01 7F FF	XX	A5 →	00 01 7F FF	XX
00 01 7F FE	XX		00 01 7F FE	XX
00 01 7F FD	XX		00 01 7F FD	XX
00 01 7F FC	XX		00 01 7F FC	XX
00 01 7F FB	XX		00 01 7F FB	XX
00 01 7F FA	XX	SP →	00 01 7F FA	XX
00 01 7F F9	XX		00 01 7F F9	XX

LSL, LSR **DECALAGE LOGIQUE A GAUCHE** LSL **DECALAGE LOGIQUE A DROITE** LSR **(LOGICAL SHIFT LEFT,RIGHT)**

OPERATION: Destination décalée de <nombre> dans Destination

SYNTAXE ASSEMBLEUR:

LSL Dx,Dy	LSR Dx,Dy
LSL #<donnée>,Dy	LSR #<donnée>,Dy
LSL <EA>	LSR <EA>

PORTEE:

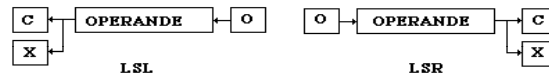
Octet(.B), Mot(.W), Mot long(.L).

FONCTION:

Les bits de l'Opérande contenus dans le registre Dy sont décalés à droite (LSR) ou à gauche (LSL) du nombre de rang spécifié dans la donnée immédiate ou dans le registre Dx

Si le décalage est en donnée immédiate, celle-ci sera logée dans le mot instruction elle pourra être de 1 à 8 dans ce dernier cas le codage sera 000. Lorsque le nombre de décalage est spécifié dans le registre de données il peut aller jusqu'à 64 le nombre contenu dans le registre est interprété comme modulo 64

Si l'opérande est contenu dans une adresse mémoire <EA> il ne peut être décalé que d'un rang



CODES CONDITION:

X et C	voir ci-dessus	Z	Mis à un si le résultat = 00
N	Copie du bit de poids fort du résultat	V	Mis à 0

FORMAT DU MOT INSTRUCTION

L'opérande est dans le registre Dy

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	0	Nbr ou Dx			dr	SIZE	i/r	0	1	REG Dy			

L'opérande est dans une case mémoire

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	0	0	0	1	dr	1	1	MODE			REGISTRE		

← Adresse Effective →

- Le champ **Nombre ou Dx** précise le nombre de décalage de 1 à 8 (de 001 à 000) ou le registre de données qui contient le nombre de décalage à effectuer (modulo 64)
- **dr** indique le sens du décalage 0 à droite 1 à gauche
- Le champ **SIZE** spécifie la taille de l'opération Octet:00 Mot:01 Mot long :10
- Le bit **i/r** précise si le nombre de décalage est dans le mot opération (0) ou dans le registre Dx (1)
- Le champ **Adresse effective** est codé, en fonction de l'adressage utilisé, suivant le tableau ci dessous

NOTA	MODE D'ADRESS	MODE	REGISTRE	NOTA	MODE D'ADRESS	MODE	REGISTRE
(3)	Dn				d(An,Xi)	110	N° Registre
(3)	An				Absolu Court	111	000
	(An)	010	N° Registre		Absolu Long	111	001
	(An)+	011	N° Registre	(3)	d(PC)		
	-(An)	100	N° Registre	(3)	d(PC,Xi)		
	d(An)	101	N° Registre	(3)	Immédiat		

NOTA (3) Ce mode d'adressage ne peut être utilisé

EXEMPLE D'UTILISATION LSR.B #1,D3

mot instruction: **0XB20B**

AVANT EXECUTION

D3	44	55	66	83	
	X	N	Z	V	C
	0	0	0	0	0

APRES EXECUTION

D3	44	55	66	41	
	X	N	Z	V	C
	1	0	0	0	1

MOVE**MOVE****TRANSFERT DE SOURCE DANS DESTINATION
(MOVE DATA FROM SOURCE TO DESTINATION)****OPERATION:**

Source dans Destination

SYNTAXE ASSEMBLEUR:

MOVE <AE>, <AE>

PORTEE:

Octet(.B), Mot(.W), Mot long(.L).

FONCTION:

Transfère le contenu de source dans destination. La donnée transférée est examinée et le CCR est positionné en conséquence

CODES CONDITION:

X	N	Z	V	C
N	A	A	0	0

X Non affecté
 N Copie du bit de poids fort du résultat
 Z Mis à un si le résultat = 00
 V Mis à 0
 C Mis à 0

FORMAT DU MOT INSTRUCTION

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	SIZE	REGISTRE	MODE	MODE	REGISTRE									

← Adresse Effect Dest → ← Adresse Effect Source →

Le champ **SIZE** spécifie la taille de l'opération

CHAMP SIZE			
OCTET	MOT	MOT LONG	OPERATION
01	11	10	Transfert de Source dans Destination

Le champ **Adresse effective** est codé, en fonction de l'adressage utilisé, suivant le tableau ci dessous aussi bien en source qu'en destination

NOTA	MODE D'ADRESS	MODE	REGISTRE	NOTA	MODE D'ADRESS	MODE	REGISTRE
	Dn	000	N° Registre		d(An, Xi)	110	N° Registre
(1 + 2)	An	001	N° Registre		Absolu Court	111	000
	(An)	010	N° Registre		Absolu Long	111	001
	(An)+	011	N° Registre	(1)	d(PC)	111	010
	-(An)	100	N° Registre	(1)	d(PC, Xi)	111	011
	d(An)	101	N° Registre	(1)	Immédiat	111	100

NOTA (1) Ce mode d'adressage ne peut être utilisé que pour l'opérande source

(2) Le transfert ne peut porter sur un octet

Voir aussi MOVEA pour un transfert dans un registre d'adresse (les assembleurs acceptent souvent un move classique) et également MOVEQ pour les transferts immédiats rapides dans les registres de données.

EXEMPLE D'UTILISATION

MOVE.W				D2, D3	mot instruction: 0X3602						
AVANT EXECUTION					APRES EXECUTION						
D2	88	99	AA	BB	D2	88	99	AA	BB		
D3	11	22	33	44	D3	11	22	AA	BB		
	X	N	Z	V	C		X	N	Z	V	C
	1	0	0	0	1		1	1	0	0	0

**MOVE
to CCR**

**TRANSFERT DANS CCR
(MOVE TO CCR)**

**MOVE
to CCR**

OPERATION:

SOURCE Æ CCR;

SYNTAXE ASSEMBLEUR:

MOVE <AE>,CCR

PORTEE:

Mot (.W obligatoire)

FONCTION:

MOVE to CCR: Transfert du contenu de source dans le registre de code condition, l'opérande source est le mot mais seuls les 5 bits de poids faible sont introduits dans le registre positionnant les bits de CCR. L'octet de poids fort est ignoré il est cependant véhiculé sur le bus.

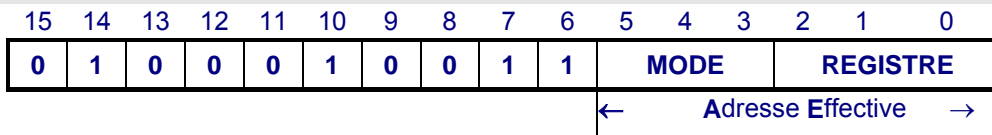
CODES CONDITION:

Move to CCR

X	N	Z	V	C
A	A	A	A	A

Le Registre de Codes Condition est affecté dans son entier par cette instruction, chaque bit est chargé par le bit correspondant de "source".

FORMAT DU MOT INSTRUCTION



Le champ **Adresse effective** est codé, en fonction de l'adressage utilisé, suivant le tableau ci dessous

NOTA	MODE D'ADRESS	MODE	REGISTRE	NOTA	MODE D'ADRESS	MODE	REGISTRE
	Dn	000	N° Registre		d(An,Xi)	110	N° Registre
(3)	An				Absolu Court	111	000
	(An)	010	N° Registre		Absolu Long	111	001
	(An)+	011	N° Registre		d(PC)	111	010
	-(An)	100	N° Registre		d(PC,Xi)	111	011
	d(An)	101	N° Registre		Immédiat	111	100

(3) Ce mode d'adressage ne peut être utilisé

EXEMPLE D'UTILISATION

<p style="text-align: center;">MOVE.W mot instruction:</p> <p style="text-align: center;">AVANT EXECUTION</p> <table border="1" style="width: 100%; text-align: center;"> <tr><td>D2</td><td>XX</td><td>XX</td><td>XX</td><td>XX</td></tr> <tr><td>D3</td><td>XX</td><td>XX</td><td>XX</td><td>XX</td></tr> </table> <table border="1" style="width: 100%; text-align: center;"> <tr><td>X</td><td>N</td><td>Z</td><td>V</td><td>C</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> </table>	D2	XX	XX	XX	XX	D3	XX	XX	XX	XX	X	N	Z	V	C	0	0	0	0	0	<p style="text-align: center;">#0X11,CCR 0X44FC mot extension: 0X0011</p> <p style="text-align: center;">APRES EXECUTION</p> <table border="1" style="width: 100%; text-align: center;"> <tr><td>D2</td><td>XX</td><td>XX</td><td>XX</td><td>XX</td></tr> <tr><td>D3</td><td>XX</td><td>XX</td><td>XX</td><td>XX</td></tr> </table> <table border="1" style="width: 100%; text-align: center;"> <tr><td>X</td><td>N</td><td>Z</td><td>V</td><td>C</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td></tr> </table>	D2	XX	XX	XX	XX	D3	XX	XX	XX	XX	X	N	Z	V	C	1	0	0	0	1
D2	XX	XX	XX	XX																																					
D3	XX	XX	XX	XX																																					
X	N	Z	V	C																																					
0	0	0	0	0																																					
D2	XX	XX	XX	XX																																					
D3	XX	XX	XX	XX																																					
X	N	Z	V	C																																					
1	0	0	0	1																																					

MOVE
from SR
to SR *

**TRANSFERT DU REGISTRE D'ETAT
TRANSFERT DANS REGISTRE D'ETAT
(MOVE FROM OR TO SR)**

MOVE
from SR
* to SR

***INSTRUCTION PRIVILEGEE (TO)**

OPERATION: **MOVE FROM SR:** SR → DESTINATION
MOVE TO SR: SI ETAT SUPERVISEUR SOURCE → SR; SI NON TRAPPE

SYNTAXE ASSEMBLEUR: MOVE SR,<AE>
MOVE <AE>,SR

PORTEE: Mot,

FONCTION:

MOVE from SR: Transfert le contenu du registre d'état dans l'emplacement destination qui ne peut être un registre d'adresse.

MOVE to SR: Si le bit S est positionné à 1 (mode superviseur), transfert du contenu de source dans le registre d'état, l'opérande source est le mot, les bits de SR sont positionnés en conséquence. Si S est à 0, déclenchement d'une exception Viol de privilège (N° vecteur 8).

CODES CONDITION:

Move from SR: Le Registre de Codes Condition n'est pas affecté par cette instruction.

Move to SR:

X	N	Z	V	C
A	A	A	A	A

Le Registre de Codes Condition est affecté dans son entier par cette instruction, chaque bit est chargé par le bit correspondant de "source". Il en est de même pour l'octet de poids fort.

FORMAT DU MOT INSTRUCTION

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	0	SENS	0	1	1	MODE			REGISTRE			
										← Adresse Effective →					

Le champ **SENS** spécifie le sens du transfert

From SR	To SR
00	11

Le champ **Adresse effective** est codé, en fonction de l'adressage utilisé, suivant le tableau ci dessous

NOTA	MODE D'ADRESS	MODE	REGISTRE	NOTA	MODE D'ADRESS	MODE	REGISTRE
	Dn	000	N° Registre		d(An,Xi)	110	N° Registre
(3)	An				Absolu Court	111	000
	(An)	010	N° Registre		Absolu Long	111	001
	(An)+	011	N° Registre	(1)	d(PC)	111	010
	-(An)	100	N° Registre	(1)	d(PC,Xi)	111	011
	d(An)	101	N° Registre	(1)	Immédiat	111	100

(1) Ce mode d'adressage ne peut être utilisé que dans MOVE to SR

(3) Ce mode d'adressage ne peut être utilisé

EXEMPLE D'UTILISATION (PASSAGE DU MASQUE D'INTERRUPTION DE 1 À 4)

MOVE #0X2400,SR

mot instruction: **0X46FC** mot extension: **0X2400**

SR AVANT EXECUTION

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	X	1	X	X	0	0	1	X	X	X	1	0	0	0	1

SR APRES EXECUTION

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	X	1	X	X	1	0	0	X	X	X	0	0	0	0	0

MOVE
from USP
to USP

**TRANSFERT DU SP UTILISATEUR
TRANSFERT DANS LE SP UTILISATEUR
(MOVE FROM OR TO SR)**

MOVE
from USP
to USP

INSTRUCTIONS PRIVILEGIEES

OPERATION:

MOVE FROM USP: SI ETAT SUPERVISEUR:
USP → REGISTRE D'ADRESSE; SI NON TRAPPE
MOVE TO USP: SI ETAT SUPERVISEUR:
REGISTRE D'ADRESSE → SR; SI NON TRAPPE

SYNTAXE ASSEMBLEUR:

MOVE USP,An
MOVE An,USP

PORTEE:

Mot Long,

FONCTION:

MOVE from USP: Si le microprocesseur est dans l'état superviseur (bit S=1), transfère le contenu du **Pointeur de Pile Utilisateur** dans le registre d'adresses spécifié. Si S est à 0, déclenchement d'une exception Viol de privilège (N° vecteur 8).

MOVE to USP: Si le microprocesseur est dans l'état superviseur (bit S=1), transfert du contenu du registre d'adresse dans le **Pointeur de Pile Utilisateur**. Si S est à 0, déclenchement d'une exception Viol de privilège (N° vecteur 8).

L'usage de cette instruction ne se justifie qu'en mode Superviseur puisque dans ce mode USP n'est pas accessible autrement. En mode Utilisateur on utilisera le MOVEA vers A7 pour charger USP

CODES CONDITION:

Le Registre de Codes Condition n'est pas affecté par cette instruction.

X	N	Z	V	C
N	N	N	N	N

FORMAT DU MOT INSTRUCTION

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	1	1	1	0	0	1	1	0	Dr	REG An		

Dr= 0 transfert de An dans USP
Dr = 1 transfert de USP dans An

EXEMPLE D'UTILISATION

	MOVE.L mot instruction:	A1,USP 0X4E61	
	AVANT EXECUTION		APRES EXECUTION
A1	00 01 00 00	A1	00 01 00 00
A7	00 11 22 44	A7	00 11 22 44
USP	00 00 55 66	USP	00 01 00 00

MOVEA**MOVEA****TRANSFERT DANS UN REGISTRE D'ADRESSE
(MOVE ADDRESS)****OPERATION:**

SOURCE → DESTINATION

SYNTAXE ASSEMBLEUR:

MOVE <AE>,An

PORTEE:

Mot (.W) Mot Long (.L)

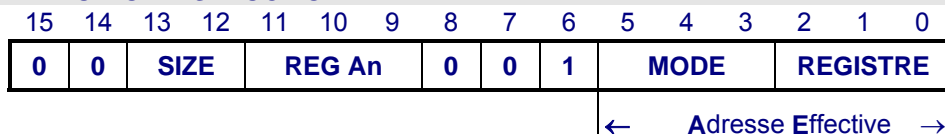
FONCTION:

Transfert le contenu de l'adresse effective dans un registre d'adresse. Le transfert s'effectue sur un mot ou un mot long. Dans le premier cas le signe est étendu à 32 bits. La plupart des assembleurs acceptent l'instruction MOVE pour ce type de transfert, cependant c'est le code ci-dessous qui résultera de l'assemblage.

CODES CONDITION:

X	N	Z	V	C
N	N	N	N	N

Le Registre de Codes Condition n'est pas affecté par cette instruction.

FORMAT DU MOT INSTRUCTION

Le champ **SIZE** spécifie la taille de l'opération

CHAMP SIZE			
OCTET	MOT	MOT LONG	OPERATION
	11	10	Transfert de <AE> dans An

Le champ **Adresse effective** est codé, en fonction de l'adressage utilisé, suivant le tableau ci dessous

	MODE D'ADRESS	MODE	REGISTRE		MODE D'ADRESS	MODE	REGISTRE
	Dn	000	N° Registre		d(An,Xi)	110	N° Registre
	An	001	N° Registre		Absolu Court	111	000
	(An)	010	N° Registre		Absolu Long	111	001
	(An)+	011	N° Registre		d(PC)	111	010
	-(An)	100	N° Registre		d(PC,Xi)	111	011
	d(An)	101	N° Registre		Immédiat	111	100

EXEMPLE D'UTILISATION

	MOVEA.W				D2,A3										
	mot instruction:				0X3642										
	AVANT EXECUTION														
D2	11	22	33	44											
A3	00	2B	00	2B											
	<table border="1"> <tr> <td>X</td> <td>N</td> <td>Z</td> <td>V</td> <td>C</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> </table>					X	N	Z	V	C	0	0	0	0	0
X	N	Z	V	C											
0	0	0	0	0											

	APRES EXECUTION														
D2	11	22	33	44											
A3	00	00	33	44											
	<table border="1"> <tr> <td>X</td> <td>N</td> <td>Z</td> <td>V</td> <td>C</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> </table>					X	N	Z	V	C	0	0	0	0	0
X	N	Z	V	C											
0	0	0	0	0											

MOVEM TRANSFERT DE MOVEM PLUSIEURS REGISTRES(MOVE MULTIPLE REGISTERS)

OPERATION:

REGISTRES dans DESTINATION
SOURCE dans REGISTRES

SYNTAXE ASSEMBLEUR:

MOVEM LISTE DE REGISTRES,<AE>
MOVEM <AE>,LISTE DE REGISTRES

PORTEE:

Mot (.W) Mot long (.L)

FONCTION:

Les registres sélectionnés sont transférés de ou vers la mémoire. Les registres concernés sont codés dans un mot extension suivant la structure décrite ci-dessous. Lorsqu'il s'agit d'un transfert **de la mémoire vers les registres, si l'opération porte sur un mot, le bit de signe est étendu à 32 bits même pour les registres de données.**

Si le mode d'adressage est pré-décrémenté, seuls les transferts de registres vers la mémoire sont possibles et l'ordre de transfert est A7 → A0 et D7 → D0.

Si le mode d'adressage est post-incrémenté, seuls les transferts de mémoire vers les registres sont possibles et l'ordre de transfert est le suivant: D0 → D7 et A0 → A7

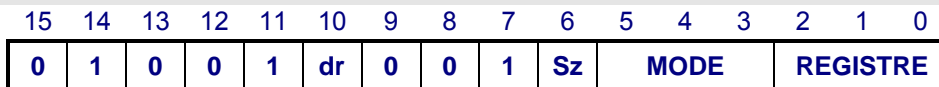
Dans les autres modes d'adressage, les deux sens de transfert sont possibles et l'ordre est D0 Æ D7 puis A0 Æ A7. Les transferts vers la mémoire commencent à l'adresse spécifiée par <AE> et se poursuivent vers les adresses croissantes.

CODES CONDITION:

X	N	Z	V	C
N	N	N	N	N

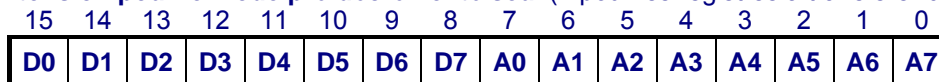
Le registre de code condition n'est pas affecté par cette instruction.

FORMAT DU MOT INSTRUCTION

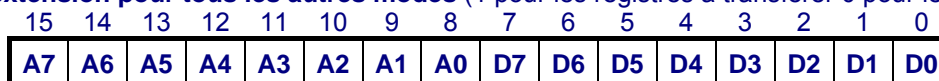


← Adresse Effective →

Mot extension pour le mode pré-décrémenté seul (1 pour les registres à transférer 0 pour les autres)



Mot extension pour tous les autres modes (1 pour les registres à transférer 0 pour les autres)



Le transfert s'effectue du bit 0 vers le bit 15

Le champ **dr** indique le sens du transfert 0 Registres vers Mémoire 1 Mémoire vers Registres

Le champ **Sz** précise la taille du transfert. 0 = Mots , 1 = Mots Longs

Le champ **Adresse effective** est codé, en fonction de l'adressage utilisé, suivant le tableau ci dessous

NOTA	MODE D'ADRESS	MODE	REGISTRE	NOTA	MODE D'ADRESS	MODE	REGISTRE
(3)	Dn	000	N° Registre		d(An,Xi)	110	N° Registre
(3)	An	001	N° Registre		Absolu Court	111	000
	(An)	010	N° Registre		Absolu Long	111	001
(1)	(An)+	011	N° Registre	(1)	d(PC)	111	010
(2)	-(An)	100	N° Registre	(1)	d(PC,Xi)	111	011
	d(An)	101	N° Registre	(3)	Immédiat		

- NOTA (1) Sens Mémoire vers Registres uniquement
 (2) Sens Registres vers Mémoire uniquement
 (3) Ce mode d'adressage ne peut être utilisé

EXEMPLE D'UTILISATION MOVEM

		MOVEM.L				D2/A2,-(A3)			
		mot instruction: 0X48E3				mot extension: 0X2020			
		AVANT EXECUTION				APRES EXECUTION			
D2	11	22	33	44	D2	11	22	33	44
D3	55	66	77	88	D3	55	66	77	88
A2	00	01	23	45	A2	00	01	23	45
A3	00	01	80	F0	A3	00	01	80	E8

MEMOIRES

ADRESSES	CONTENU		ADRESSES	CONTENU
00 01 80 F0	XX	A3	00 01 80 F0	XX
00 01 80 EF	XX		00 01 80 EF	45
00 01 80 EE	XX		00 01 80 EE	23
00 01 80 ED	XX		00 01 80 ED	01
00 01 80 EC	XX		00 01 80 EC	00
00 01 80 EB	XX		00 01 80 EB	44
00 01 80 EA	XX		00 01 80 EA	33
00 01 80 E9	XX		00 01 80 E9	22
00 01 80 E8	XX		A3→ 00 01 80 E8	11
00 01 80 E7	XX		00 01 80 E7	XX
00 01 80 E6	XX		00 01 80 E6	XX

EXEMPLE D'UTILISATION MOVEP

		MOVEP.L D2,(0,A3)				mot instruction: 0X05CB				mot extension: 0X0000																			
		AVANT EXECUTION				APRES EXECUTION																							
D2	11	22	33	44	D2	11	22	33	44																				
A3	00	01	7F	FD	A3	00	01	7F	FD																				
<table border="1"> <tr><td>X</td><td>N</td><td>Z</td><td>V</td><td>C</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> </table>					X	N	Z	V	C	0	0	0	0	0	<table border="1"> <tr><td>X</td><td>N</td><td>Z</td><td>V</td><td>C</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> </table>					X	N	Z	V	C	0	0	0	0	0
X	N	Z	V	C																									
0	0	0	0	0																									
X	N	Z	V	C																									
0	0	0	0	0																									

MEMOIRE CONNECTEE SUR LA PARTIE BASSE DU BUS DE DONNEES

ADRESSES	CONTENU		ADRESSES	CONTENU
00 01 80 05	XX		00 01 80 05	XX
00 01 80 03	XX		00 01 80 03	44
00 01 80 01	XX		00 01 80 01	33
00 01 7F FF	XX		00 01 7F FF	22
00 01 7F FD	XX		00 01 7F FD	11
00 01 7F FB	XX		00 01 7F FB	XX

(Le terme mémoire doit être pris au sens large, il peut s'agir des registres internes d'un périphérique placés à des adresses consécutives)

MOVEP TRANSFERT DE DONNEES DE MOVEP ou VERS PERIPHERIQUES (MOVE PERIPHERAL DATA)

OPERATION:

SOURCE DANS DESTINATION

SYNTAXE ASSEMBLEUR:

MOVEP Dx,(d,Ay)

MOVEP (d,Ay),Dx

PORTEE:

Mot (.W) Mot long (.L)

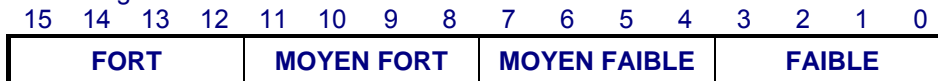
FONCTION:

Cette instruction est utilisée pour transférer des données de ou vers un périphérique dont le port 8 bits est connecté sur un bus 16 bits. L'adresse du périphérique est spécifiée dans le mode d'adressage indirect par registre d'adresse avec déplacement sur 16 bits. Le transfert s'effectue Octet de poids fort d'abord à l'adresse indiquée puis le moyen fort à l'adresse +2 moyen faible ad+4 et en dernier octet de poids faible à l'adresse +6. Si l'adresse est paire (even) le transfert s'effectue sur la partie haute du bus, si l'adresse est impaire (odd) c'est la partie basse du bus qui sera utilisée.

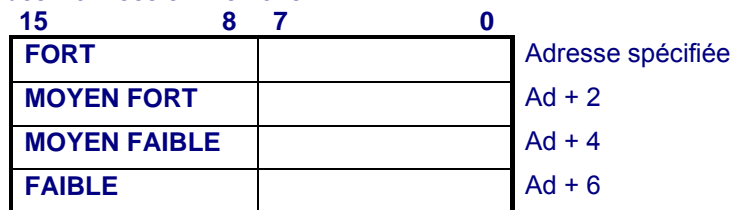
Exemples:

Transfert d'un mot long de ou vers la mémoire

Structure du Registre

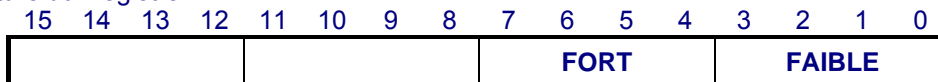


Organisation des Données en mémoire

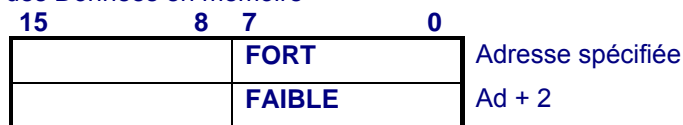


Transfert d'un mot de ou vers la mémoire à une adresse impaire

Structure du Registre



Organisation des Données en mémoire

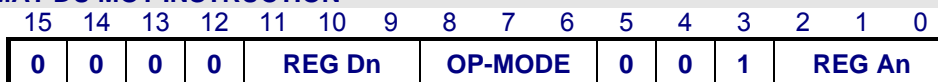


CODES CONDITION:

X	N	Z	V	C
N	N	N	N	N

Le registre de code condition n'est pas affecté par cette instruction.

FORMAT DU MOT INSTRUCTION



Le Champ **REGISTRE Dn** spécifie le registre de données

Le Champ **REGISTRE An** spécifie le registre d'adresse

Le rôle du champ **OP-MODE** est précisé dans le tableau ci-dessous

Mots Mem → Dn	M Longs Mem → Dn	Mots Dn → Mem	M Longs Dn → Mem
100	101	110	111

MOVEQ**MOVEQ****TRANSFERT RAPIDE
(MOVE QUICK)****OPERATION:**

Donnée immédiate dans Destination

SYNTAXE ASSEMBLEUR:

MOVEQ #Donnée,Dn

PORTEE:

Octet étendu à Mot long

FONCTION:

Transfert la donnée immédiate dans le Registre de Données. La donnée immédiate est fournie sous la forme d'un Octet mais son bit de signe est étendu à 32 bits. Le registre est donc affecté dans son ensemble.

CODES CONDITION:

X	N	Z	V	C
N	A	A	0	0

- X Non affecté
- N Copie du bit de poids fort du résultat
- Z Mis à un si le résultat = 0 sinon mis à 0
- V Mis à 0
- C Mis à 0

FORMAT DU MOT INSTRUCTION

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	1	REGISTRE			0	DONNEE IMMEDIATE							

Le champ **REGISTRE** spécifie le N° du Registre qui doit recevoir la donnée.-

EXEMPLE D'UTILISATION

		MOVEQ		#4,D3	
		mot instruction:		0X7604	
		AVANT EXECUTION		APRES EXECUTION	
D2	XX	XX	XX	XX	D2
D3	44	55	66	77	D3
	X	N	Z	V	C
	0	0	0	0	0
	X	N	Z	V	C
	0	0	0	0	0

**MULS
MULU****MULTIPLICATION SIGNEE
MULTIPLICATION NON SIGNEE
(SIGNED OR UNSIGNED MULTIPLY)****MULS
MULU****OPERATION:**

Source multipliée par Destination dans Destination

SYNTAXE ASSEMBLEUR:MULS <AE>,Dn
MULU <AE>,Dn**PORTEE:**

Opérandes: Mot(.W), Résultat: Mot long(.L).

FONCTION:Effectue le produit signé (MULS) ou non signé (MULU) de deux mots résultat sur 32 .
MUL(S ou U).W effectue le produit d'un mot, contenu de l'Adresse Effective, et du mot de poids faible du registre Destination résultat sur 32 bits dans le registre Destination.**CODES CONDITION:**

X	N	Z	V	C
N	A	A	0	0

X Non affecté
 N Copie du bit de poids fort du résultat
 Z Mis à un si le résultat = 0
 V Mis à 0
 C Mis à 0

FORMAT DU MOT INSTRUCTION

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	REG Dn		U/S	1	1	MODE		REGISTRE				

← Adresse Effective →

Le champ **Dn** indique le N° du registre de données utiliséLe champ **U/S** spécifie la nature de l'opération 0= Non signée 1= SignéeLe champ **Adresse effective** est codé, en fonction de l'adressage utilisé, suivant le tableau ci dessous

NOTA	MODE D'ADRESS	MODE	REGISTRE	NOTA	MODE D'ADRESS	MODE	REGISTR
	Dn	000	N° Registre		d(An,Xi)	110	N° Registr
(3)	An				Absolu Court	111	000
	(An)	010	N° Registre		Absolu Long	111	001
	(An)+	011	N° Registre		d(PC)	111	010
	-(An)	100	N° Registre		d(PC,Xi)	111	011
	d(An)	101	N° Registre		Immédiat	111	100

NOTA (3) Ce mode d'adressage ne peut être utilisé

EXEMPLES D'UTILISATION DES INSTRUCTIONS MULU ET MULS

1er EXEMPLE **MULU D2,D3**

mot instruction: **0XC6C2**

AVANT EXECUTION

D2	11	22	00	02
D3	33	44	00	02

X	N	Z	V	C
0	0	0	0	0

APRES EXECUTION

D2	11	22	00	02
D3	00	00	00	04

X	N	Z	V	C
0	0	0	0	0

2ème EXEMPLE **MULU D2,D3** (ici -2 x -2)

AVANT EXECUTION

D2	00	00	FF	FE
D3	00	00	FF	FE

X	N	Z	V	C
0	0	0	0	0

APRES EXECUTION

D2	00	00	FF	FE
D3	FF	FC	00	04

X	N	Z	V	C
0	1	0	0	0

Les nombres négatifs ne sont pas reconnus par la multiplication non signée

3ème EXEMPLE **MULS D2,D3**

mot instruction: **0XC7C2**

AVANT EXECUTION

D2	00	00	00	02
D3	00	00	00	02

X	N	Z	V	C
0	0	0	0	0

APRES EXECUTION

D2	00	00	00	02
D3	00	00	00	04

X	N	Z	V	C
0	0	0	0	0

(multiplication de +2 par +2 = +4)

4ème EXEMPLE **MULS D2,D3**

AVANT EXECUTION

D2	00	00	FF	FE
D3	00	00	00	02

X	N	Z	V	C
0	0	0	0	0

APRES EXECUTION

D2	00	00	FF	FE
D3	FF	FF	FF	FC

X	N	Z	V	C
0	1	0	0	0

(multiplication de -2 (FFFE) par +2 = (FFFFFFFC (-4))

5ème EXEMPLE **MULS D2,D3**

AVANT EXECUTION

D2	00	00	00	02
D3	00	00	FF	FE

X	N	Z	V	C
0	0	0	0	0

APRES EXECUTION

D2	00	00	00	02
D3	FF	FF	FF	FC

X	N	Z	V	C
0	1	0	0	0

(multiplication de +2 par -2 (FFFE) = (FFFFFFFC (-4))

6ème EXEMPLE **MULS D2,D3**

AVANT EXECUTION

D2	00	00	FF	FE
D3	00	00	FF	FE

X	N	Z	V	C
0	0	0	0	0

APRES EXECUTION

D2	00	00	FF	FE
D3	00	00	00	04

X	N	Z	V	C
0	0	0	0	0

(multiplication de -2 (FFFE) par -2 (FFFE) = +4)

NBCD

NBCD

**COMPLEMENT A DIX AVEC EXTENSION
(NEGATE DECIMAL WITH EXTEND)**

OPERATION:

0 - Destination (10) - X dans Destination

SYNTAXE ASSEMBLEUR:

NBCD <AE>

PORTEE:

Octet (.B)

FONCTION:

L'opérande et le bit X sont soustraits à 00 (soustraction en DCBN).

De même qu'il est possible de transformer un soustraction de nombres en binaire naturel en une addition du complément à deux, on peut transformer une soustraction DCBN en addition du complément à dix. Pour réaliser une soustraction sur des nombres de plusieurs décades, le complément à dix de la décade de poids faible doit être réalisé d'abord avec X=0. Cette première opération générera une retenue entraînant X=1 et le complément de la décade suivante tiendra compte de ce report.

CODES CONDITION:

X	N	Z	V	C
A	?	A	?	A

- X Copie du bit C
- N Indéfini
- Z Mis à 0 si le résultat non nul, si non inchangé
- V Indéfini
- C Mis à 1 si une retenue décimale est générée

FORMAT DU MOT INSTRUCTION

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	1	0	0	0	0	0	MODE			REGISTRE		

← Adresse Effective →

Le champ **Adresse effective** est codé, en fonction de l'adressage utilisé, suivant le tableau ci dessous

NOTA	MODE D'ADRESS	MODE	REGISTRE	NOTA	MODE D'ADRESS	MODE	REGISTRE
	Dn	000	N° Registre		d(An,Xi)	110	N° Registre
(3)	An				Absolu Court	111	000
	(An)	010	N° Registre		Absolu Long	111	001
	(An)+	011	N° Registre	(3)	d(PC)		
	-(An)	100	N° Registre	(3)	d(PC,Xi)		
	d(An)	101	N° Registre	(3)	Immédiat		

NOTA (3) Ce mode d'adressage ne peut être utilisé

EXEMPLE D'UTILISATION

NBCD

mot instruction:

D3

0X4803

AVANT EXECUTION

D2	XX	XX	XX	XX
D3	44	55	66	77

APRES EXECUTION

D2	XX	XX	XX	XX
D3	44	55	66	23

X	N	Z	V	C
0	0	0	0	0

X	N	Z	V	C
1	0	0	0	1

NEG COMPLEMENT A DEUX SANS NEG NEGX OU AVEC EXTENSION NEGX (NEGATE AND NEGATE WITH EXTEND)

OPERATION:

0 moins Destination dans Destination
0 moins Destination moins X dans Destination

SYNTAXE ASSEMBLEUR:

NEG <AE>
NEGX <AE>

PORTEE:

Octet (.B), Mot (.W), Mot Long (.L)

FONCTION:

NEG L'opérande est soustrait à 00, cette soustraction est effectuée en binaire naturel.(opération identique à inversion des bits + 1 ou complément à deux)

NEGX L'opérande et le bit X sont soustraits à 00.

CODES CONDITION:

X	N	Z	V	C
A	A	A	?	A

X et C Mis à 1 si une retenue est générée Z Mis à un si le résultat = 0 sinon inchangé
N Copie du bit de poids fort du résultat V Indéfini

FORMAT DU MOT INSTRUCTION

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	0	F	0	0	SIZE	MODE	REGISTRE					
									←	Adresse Effective	→				

Le bit **F** est à **1** pour l'instruction **NEG** et à **0** pour **NEGX**

Le champ **SIZE** spécifie la taille de l'opération

CHAMP SIZE			
OCTET	MOT	MOT LONG	OPERATION
00	01	10	Complément de Destination dans Destination

Le champ **Adresse effective** est codé, en fonction de l'adressage utilisé, suivant le tableau ci dessous

NOTA	MODE D'ADRESS	MODE	REGISTRE	NOTA	MODE D'ADRESS	MODE	REGISTRE
	Dn	000	N° Registre		d(An,Xi)	110	N° Registre
(3)	An				Absolu Court	111	000
	(An)	010	N° Registre		Absolu Long	111	001
	(An)+	011	N° Registre	(3)	d(PC)		
	-(An)	100	N° Registre	(3)	d(PC,Xi)		
	d(An)	101	N° Registre	(3)	Immédiat		

NOTA (3) Ce mode d'adressage ne peut être utilisé

EXEMPLE D'UTILISATION

NEG.B D3

mot instruction: **0X4403**

		AVANT EXECUTION					
D2		XX	XX	XX	XX		
D3		44	55	66	77		

X	N	Z	V	C
0	0	0	0	0

		APRES EXECUTION					
D2		XX	XX	XX	XX		
D3		44	55	66	89		

X	N	Z	V	C
1	1	0	0	1

NOP**NOP****PAS D'OPERATION
(NO OPERATION)****OPERATION:**

Néant

SYNTAXE ASSEMBLEUR:

NOP

PORTEE:

Sans

FONCTION:

Le processeur passe à l'instruction suivante

CODES CONDITION:

X	N	Z	V	C
N	N	N	N	N

Le registre de code condition n'est pas affecté par cette opération.

FORMAT DU MOT INSTRUCTION**0X 4E71**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	1	1	1	0	0	1	1	1	0	0	0	1

NOT NOT COMPLEMENT A UN OU COMPLEMENT LOGIQUE (LOGICAL COMPLEMENT)

OPERATION:

Inversion des bits de Destination dans Destination

SYNTAXE ASSEMBLEUR:

NOT <AE>

PORTEE:

Octet (.B), Mot (.W), Mot Long (.L)

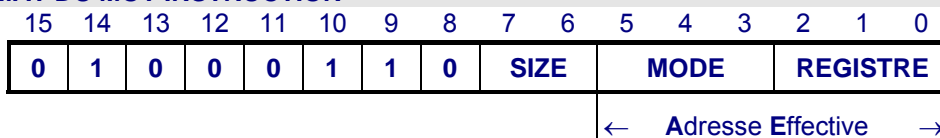
FONCTION:

Les bits de l'Opérande sont inversés et chargés dans destination. Cette fonction est équivalente à une soustraction en binaire naturel de l'opérande à FF

CODES CONDITION:

X	N	Z	V	C
N	A	A	0	0

X Non affecté
 N Copie du bit de poids fort du résultat
 Z Mis à un si le résultat = 0
 V Toujours mis à 0
 C Toujours mis à 0

FORMAT DU MOT INSTRUCTIONLe champ **SIZE** spécifie la taille de l'opération

CHAMP SIZE			
OCTET	MOT	MOT LONG	OPERATION
00	01	10	Complément de Destination dans Destination

Le champ **Adresse effective** est codé, en fonction de l'adressage utilisé, suivant le tableau ci dessous

NOTA	MODE D'ADRESS	MODE	REGISTRE	NOTA	MODE D'ADRESS	MODE	REGISTRE
	Dn	000	N° Registre		d(An,Xi)	110	N° Registre
(3)	An				Absolu Court	111	000
	(An)	010	N° Registre		Absolu Long	111	001
	(An)+	011	N° Registre	(3)	d(PC)		
	-(An)	100	N° Registre	(3)	d(PC,Xi)		
	d(An)	101	N° Registre	(3)	Immédiat		

NOTA (3) Ce mode d'adressage ne peut être utilisé

EXEMPLE D'UTILISATION

<p style="text-align: center;">NOT.W mot instruction:</p> <p style="text-align: center;">D3 0X4643</p> <p>AVANT EXECUTION</p> <table border="1" style="width: 100%; text-align: center;"> <tr> <td>D2</td> <td>XX</td> <td>XX</td> <td>XX</td> <td>XX</td> </tr> <tr> <td>D3</td> <td>44</td> <td>55</td> <td>66</td> <td>77</td> </tr> </table> <table border="1" style="width: 100%; text-align: center;"> <tr> <td>X</td> <td>N</td> <td>Z</td> <td>V</td> <td>C</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> </table>	D2	XX	XX	XX	XX	D3	44	55	66	77	X	N	Z	V	C	0	0	0	0	0	<p>APRES EXECUTION</p> <table border="1" style="width: 100%; text-align: center;"> <tr> <td>D2</td> <td>XX</td> <td>XX</td> <td>XX</td> <td>XX</td> </tr> <tr> <td>D3</td> <td>44</td> <td>55</td> <td>99</td> <td>88</td> </tr> </table> <table border="1" style="width: 100%; text-align: center;"> <tr> <td>X</td> <td>N</td> <td>Z</td> <td>V</td> <td>C</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> </tr> </table>	D2	XX	XX	XX	XX	D3	44	55	99	88	X	N	Z	V	C	0	1	0	0	0
D2	XX	XX	XX	XX																																					
D3	44	55	66	77																																					
X	N	Z	V	C																																					
0	0	0	0	0																																					
D2	XX	XX	XX	XX																																					
D3	44	55	99	88																																					
X	N	Z	V	C																																					
0	1	0	0	0																																					

OR

OR

**OU LOGIQUE (INCLUSIF)
(INCLUSIVE OR LOGICAL)**

OPERATION:

Source **OU** Destination dans Destination

SYNTAXE ASSEMBLEUR:

OR <AE>,Dn
OR Dn,<AE>

PORTEE:

Octet (.B), Mot (.W), Mot Long (.L)

FONCTION:

Un OU logique est effectué entre les Opérandes Source et Destination bit à bit, le résultat est chargé dans la Destination

CODES CONDITION:

X	N	Z	V	C
N	A	A	0	0

- X Non affecté
- N Copie du bit de poids fort du résultat
- Z Mis à un si le résultat = 0
- V Toujours mis à 0
- C Toujours mis à 0

FORMAT DU MOT INSTRUCTION

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	0	0	REGISTRE			MODE OP			MODE			REGISTRE		

← Adresse Effective →

Le champ **REGISTRE** indique le N° du registre de données employé

Le champ **OPMODE** spécifie la taille de l'opération suivant l'opération à effectuer

CHAMP MODE OP			
OCTET	MOT	MOT LONG	OPERATION
000	001	010	<AE> + <Dn> dans <Dn>
100	101	110	<Dn> + <AE> dans <AE>

Le champ **Adresse effective** est codé, en fonction de l'adressage utilisé, suivant le tableau ci dessous

NOTA	MODE D'ADRESS	MODE	REGISTRE	NOTA	MODE D'ADRESS	MODE	REGISTRE
(1)	Dn	000	N° Registre		d(An,Xi)	110	N° Registre
(3)	An				Absolu Court	111	000
	(An)	010	N° Registre		Absolu Long	111	001
	(An)+	011	N° Registre	(1)	d(PC)	111	010
	-(An)	100	N° Registre	(1)	d(PC,Xi)	111	011
	d(An)	101	N° Registre	(1)	Immédiat	111	100

(1) <AE> Source seulement

(3) Ce mode d'adressage ne peut être utilisé

EXEMPLE D'UTILISATION

OR.W D2,D3
mot instruction: **0X8642**

AVANT EXECUTION

D2	11	22	33	44
D3	44	55	00	00

X	N	Z	V	C
0	0	0	0	0

APRES EXECUTION

D2	11	22	33	44
D3	44	55	33	44

X	N	Z	V	C
0	0	0	0	0

ORI OU LOGIQUE (INCLUSIF) ORI

(INCLUSIVE OR LOGICAL)

OPERATION:

Donnée immédiate OU Destination dans Destination

SYNTAXE ASSEMBLEUR:

ORI #<Donnée>,<AE>

PORTEE:

Octet (.B), Mot (.W), Mot Long (.L)

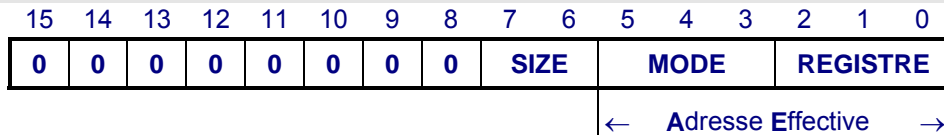
FONCTION:

Un OU logique est effectué entre la donnée immédiate et l'Opérande Destination, le résultat est chargé dans la Destination. La taille de la donnée doit correspondre à la taille de l'opération.

CODES CONDITION:

X	N	Z	V	C
N	A	A	0	0

X Non affecté C
 N Copie du bit de poids fort du résultat
 Z Mis à un si le résultat = 0
 V Toujours mis à 0
 C Toujours mis à 0

FORMAT DU MOT INSTRUCTION

CHAMP SIZE			
OCTET	MOT	MOT LONG	OPERATION
00	01	10	Donnée OU Destination dans Destination

Le champ **Adresse effective** est codé, en fonction de l'adressage utilisé, suivant le tableau ci dessous

NOTA	MODE D'ADRESS	MODE	REGISTRE	NOTA	MODE D'ADRESS	MODE	REGISTRE
	Dn	000	N° Registre		d(An,Xi)	110	N° Registre
(3)	An				Absolu Court	111	000
	(An)	010	N° Registre		Absolu Long	111	001
	(An)+	011	N° Registre	(3)	d(PC)		
	-(An)	100	N° Registre	(3)	d(PC,Xi)		
	d(An)	101	N° Registre	(3)	Immédiat		

NOTA (3) Ce mode d'adressage ne peut être utilisé

EXEMPLE D'UTILISATION

ORI.B mot instruction:				#0XFF,D3 0X0003 mot extension	0X00FF						
AVANT EXECUTION				APRES EXECUTION							
D2	XX	XX	XX	D2	XX	XX	XX	XX			
D3	44	55	66	D3	44	55	66	FF			
	X	N	Z	V	C		X	N	Z	V	C
	0	0	0	0	0		0	1	0	0	0

**ORI
to CCR**

**OU LOGIQUE IMMEDIAT
AVEC CCR**

**ORI
to CCR**

OPERATION: DONNEE IMMEDIATE OU CCR dans CCR

SYNTAXE ASSEMBLEUR:

ORI #Donnée,CCR

PORTEE:

Octet

FONCTION:

Effectue un OU logique entre la donnée immédiate et le registre de codes condition, résultat dans la partie basse du registre d'état (CCR).

CODES CONDITION:

X	N	Z	V	C
A	A	A	A	A

X mis à 0 si le bit correspondant de la donnée est à 0 sinon inchangé
N, Z, V, C identiques à X

FORMAT DU MOT INSTRUCTION

0X003C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0

Le mot instruction est suivi d'un mot extension dont l'octet de poids faible contient la donnée immédiate . L'octet de poids fort est mis à 0.

**ORI
to SR**

**OU LOGIQUE IMMEDIAT
AVEC SR**

**ORI
to SR**

INSTRUCTION PRIVILEGIEE

OPERATION: Si le processeur est dans l'état superviseur
DONNEE IMMEDIATE OU SR dans SR
Sinon TRAP

SYNTAXE ASSEMBLEUR:

ORI #Donnée,SR

PORTEE:

Mot

FONCTION:

Si le microprocesseur est dans le mode superviseur, effectue un OU logique entre la donnée immédiate et le registre d'état, résultat dans le registre d'état (SR). Si le microprocesseur n'est pas dans le mode superviseur, déclenchement d'une exception viol de privilège (Vecteur N° 8)

CODES CONDITION:

X	N	Z	V	C
A	A	A	A	A

X mis à 0 si le bit correspondant de la donnée est à 0 sinon inchangé
N, Z, V, C identiques à X

FORMAT DU MOT INSTRUCTION

0X007C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0

Le mot instruction est suivi d'un mot extension qui contient la donnée immédiate .

PEA**PEA**

EMPILEMENT DE L'ADRESSE EFFECTIVE (PUSH EFFECTIVE ADDRESS)

OPERATION:

SP - 4 dans SP; AE dans (SP)

SYNTAXE ASSEMBLEUR:

PEA <AE>

PORTEE:

Mot Long (.L)

FONCTION:

L'adresse effective est calculée, le pointeur de pile est décrémenté de 4 unités et l'adresse effective est poussée dans la pile

CODES CONDITION:

X	N	Z	V	C
N	N	N	N	N

Le registre de Codes Condition n'est pas affecté par cette instruction

FORMAT DU MOT INSTRUCTION

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	1	0	0	0	0	1	MODE			REGISTRE		
										← Adresse Effective →					

Le champ **Adresse effective** est codé, en fonction de l'adressage utilisé, suivant le tableau ci dessous

NOTA	MODE D'ADRESS	MODE	REGISTRE	NOTA	MODE D'ADRESS	MODE	REGISTRE
(3)	Dn				d(An,Xi)	110	N° Registre
(3)	An				Absolu Court	111	000
	(An)	010	N° Registre		Absolu Long	111	001
(3)	(An)+				d(PC)	111	010
(3)	-(An)				d(PC,Xi)	111	011
	d(An)	101	N° Registre	(3)	Immédiat		

(3) Ce mode d'adressage ne peut être utilisé

EXEMPLE D'UTILISATION

		PEA			8(A3)				
		mot instruction:		0X486B		mot extension: 0X0008			
		AVANT EXECUTION				APRES EXECUTION			
A3	00	02	20	00	A3	00	02	20	00
SP	00	01	80	04	SP	00	01	80	00

MEMOIRE

ADRESSES	CONTENU		ADRESSES	CONTENU
00 01 80 04	XX	SP	00 01 80 04	XX
00 01 80 03	XX		00 01 80 03	08
00 01 80 02	XX		00 01 80 02	20
00 01 80 01	XX		00 01 80 01	02
00 01 80 00	XX	SP→	00 01 80 00	00
00 01 7F FF	XX		00 01 7F FF	XX

RESET**RESET**

RESET PERIPHERIQUES (RESET EXTERNAL DEVICES)

INSTRUCTION PRIVILEGIEE
OPERATION:

Si en état superviseur active la ligne RESET sinon Trappe

SYNTAXE ASSEMBLEUR:

RESET

PORTEE:

Sans

FONCTION:

La ligne Reset passe à l'état actif causant le reset des circuits périphériques connectés à cette ligne. L'état interne du 68000 n'est pas affecté par cette instruction (à l'exception de PC qui est incrémenté) . Si le microprocesseur n'est pas dans l'état superviseur une exception pour viol de privilège est entreprise.

CODES CONDITION:

X	N	Z	V	C
N	N	N	N	N

Le registre de Codes Condition n'est pas affecté par cette instruction

FORMAT DU MOT INSTRUCTION**0X4E70**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	1	1	1	0	0	1	1	1	0	0	0	0

ROL ROTATION A GAUCHE(ROTATE LEFT) ROL
ROR ROTATION A DROITE(ROTATE RIGHT) ROR

OPERATION:

Rotation de Destination de <nombre>de case dans Destination

SYNTAXE ASSEMBLEUR:

ROL Dx,Dy ROL #<donnée>,Dy ROL <EA>
 ROR Dx,Dy ROR #<donnée>,Dy ROR <EA>

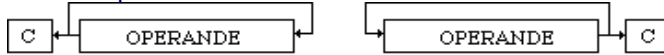
PORTEE: Octet(.B), Mot(.W), Mot long(.L).

FONCTION:

Une rotation des bits de l'Opérande contenus dans le registre Dy est effectuée à droite (ROR) ou à gauche (ROL) du nombre de rang spécifié dans la donnée immédiate ou dans le registre Dx

Si la rotation est en donnée immédiate, celle-ci sera logée dans le mot instruction elle pourra être de 1 à 8, dans ce dernier cas le codage sera 000. Lorsque le nombre de rotation est spécifié dans le registre de données il peut aller jusqu'à 64 le nombre contenu dans le registre est interprété comme modulo 64

Si l'opérande est contenu dans une adresse mémoire <EA> il ne peut être décalé que d'un rang



ROL
CODES CONDITION:

ROR

X	N	Z	V	C
N	A	A	0	A

- X Non affecté
- N Copie du bit de poids fort du résultat
- Z Mis à un si le résultat = 00
- V Mis à 0
- C Copie du dernier MSB décalé (ROL) LSB (ROR)

FORMAT DU MOT INSTRUCTION

L'opérande est dans le registre Dy

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	0	Nbre ou Dx		dr	SIZE	i/r	1	1	Registre Dy				

L'opérande est dans une case mémoire

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	0	0	1	1	dr	1	1	MODE		REGISTRE			

← Adresse Effective →

- Le champ **Nombre ou Dx** précise le nombre de rotation de 1 à 8 (de 001 à 000) ou le registre de données qui contient le nombre de rotation à effectuer (modulo 64)
- **dr** indique le sens du rotation 0 à droite 1 à gauche
- Le champ **SIZE** spécifie la taille de l'opération 00:OCTET 01:MOT 10 MOT LONG
- Le bit **i/r** précise si le nombre de rotation est contenu dans le mot opération(0) ou dans le registre Dx (1)
- Le champ **Adresse effective** est codé, en fonction de l'adressage utilisé, suivant le tableau ci dessous

NOTA	MODE D'ADRESS	MODE	REGISTRE	NOTA	MODE D'ADRESS	MODE	REGISTRE
(3)	Dn				d(An,Xi)	110	N° Registre
(3)	An				Absolu Court	111	000
	(An)	010	N° Registre		Absolu Long	111	001
	(An)+	011	N° Registre	(3)	d(PC)		
	-(An)	100	N° Registre	(3)	d(PC,Xi)		
	d(An)	101	N° Registre	(3)	Immédiat		

NOTA (3) Ce mode d'adressage ne peut être utilisé

EXEMPLE D'UTILISATION ROL.B

#1,D3

mot instruction: **0XE31B**

AVANT EXECUTION

D3	44	55	66	83
	X	N	Z	V
	0	0	0	0

APRES EXECUTION

D3	44	55	66	07
	X	N	Z	V
	0	0	0	1

SBCD

SBCD

SOUSTRACTION DECIMALE AVEC RETENUE (SUBSTRACT DECIMAL WITH EXTEND)

OPERATION:DESTINATION₁₀ - SOURCE₁₀ - X dans DESTINATION**SYNTAXE ASSEMBLEUR:**

SBCD Dy,Dx

SBCD -(Ay),-(Ax)

PORTEE:

Octet de poids faible

FONCTION:

Soustraction en DCBN de l'octet source et de l'extension X à l'octet destination .

- Entre registres de données (octets de poids faible)

Dx = Registre Destination

Dy = Registre Source

- Entre cases mémoire, adressage par registres d'adresses pré-décrémentés uniquement

CODES CONDITION:

X	N	Z	V	C
A	?	A	?	A

Le registre de Codes Condition est affecté par la restitution de SR

X Identique à C

N Indéfini

Z mis à 0 si résultat non nul sinon inchangé

V Indéfini

C Mis à 1 si un report décimal est généré sinon mis à 0

FORMAT DU MOT INSTRUCTION

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	0	0	N° Rx			1	0	0	0	0	R/M	N° Ry		

R/M = 0 = Opération entre registres de données

= 1 = Opération entre cases mémoire

N° Rx Numéro du registre Destination

N° Ry Numéro du registre Source

EXEMPLE D'UTILISATION

SBCD D2,D3
mot instruction: **0X8702**

AVANT EXECUTION

D2	XX	XX	XX	77
D3	44	55	66	82

X	N	Z	V	C
0	0	0	0	0

APRES EXECUTION

D2	XX	XX	XX	77
D3	44	55	66	05

X	N	Z	V	C
0	0	0	0	0

Scc POSITIONNEMENT D'UN OCTET SUR CONDITION **Scc**
à **Svs** (SET ACCORDING TO CONDITION) à **Svs**

OPERATION:

Si la condition est vraie alors FF dans Destination sinon 00

SYNTAXE ASSEMBLEUR:

Scc <AE>

PORTEE:

Octet

FONCTION:

La condition spécifiée est testée, si la condition est vraie, l'octet spécifié est mis à FF sinon mis à 00.

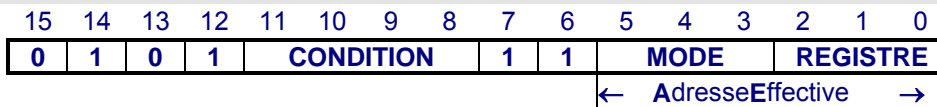
Instruction	Codage	Résultat d'une Comparaison Destination moins Source	Système de Numération	Condition booléenne
SCC	0100	≥ à 0	Binaire Naturel	C = 0
SCS	0101	< à 0	Binaire Naturel	C = 1
SEQ	0111	= 0	Quelconque	Z = 1
SGE	1100	≥ à 0	Comp à 2	N ≈ V = 0
SGT	1110	> à 0	Comp à 2	(Z nor N) ≈ V = 1
SHI	0010	> à 0	Binaire Naturel	C ou Z = 0
SLE	1111	≤ à 0	Comp à 2	Z ou N ≈ V = 1
SLS	0011	≤ à 0	Binaire Naturel	C ou Z = 1
SLT	1101	< à 0	Comp à 2	N ≈ V = 1
SMI	1011	Négatif	Quelconque	N = 1
SNE	0110	Différent de 0	Quelconque	Z = 0
SPL	1010	Positif	Quelconque	N = 0
SVC	1000	Signe Cohérent	Comp à 2	V = 0
SVS	1001	Signe non cohérent	Comp à 2	V = 1

CODES CONDITION:

X	N	Z	V	C
N	N	N	N	N

Le registre de Codes Condition n'est pas affecté par cette instruction

FORMAT DU MOT INSTRUCTION



- Le champ **condition** est chargé avec le code du tableau ci-dessus en fonction du test à effectuer.

- Le champ **Adresse effective** est codé, en fonction de l'adressage utilisé, suivant le tableau ci dessous

NOTA	MODE D'ADRESS	MODE	REGISTRE	NOTA	MODE D'ADRESS	MODE	REGISTRE
	Dn	000	N° Registre		d(An,Xi)	110	N° Registre
(3)	An				Absolu Court	111	000
	(An)	010	N° Registre		Absolu Long	111	001
	(An)+	011	N° Registre	(3)	d(PC)		
	-(An)	100	N° Registre	(3)	d(PC,Xi)		
	d(An)	101	N° Registre	(3)	Immédiat		

NOTA (3) Ce mode d'adressage ne peut être utilisé

STOP

STOP

CHARGEMENT DU REGISTRE D'ETAT ET STOP (LOAD STATUS REGISTER AND STOP)

INSTRUCTION PRIVILEGIEE

OPERATION:

Si dans l'état superviseur:
Donnée immédiate dans le registre d'état puis Stop
Sinon Trappe

SYNTAXE ASSEMBLEUR:

STOP #<donnée>

PORTEE:

Sans

FONCTION:

La donnée immédiate est chargée dans le registre d'état, le compteur de programme pointe l'instruction suivante puis le processeur cesse les recherches et l'exécution des instructions.

L'exécution des instructions reprend lorsque survient une exception: Trace, Interruption ou Reset. Une exception Trace ne peut survenir que si le bit T était actif avant le début de l'exécution de l'instruction Stop. Pour qu'une interruption soit prise en compte, elle doit être de niveau supérieur à celui contenu dans la donnée immédiate. Un Reset externe entraîne toujours une exception reset.

Dans la donnée immédiate fournie, le bit S doit être à 1, dans le cas contraire, l'exécution de l'instruction entraînerait une exception viol de privilège.

CODES CONDITION:

X	N	Z	V	C
A	A	A	A	A

Le registre de Codes Condition est affecté par le chargement de SR avec la donnée immédiate

FORMAT DU MOT INSTRUCTION

0X4E72

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	1	1	1	0	0	1	1	1	0	0	1	0

Le code instruction est suivi d'un mot contenant la donnée immédiate.

SUB

SUB

SOUSTRACTION (SUBTRACT BINARY)

OPERATION:

DESTINATION - SOURCE dans DESTINATION

SYNTAXE ASSEMBLEUR:

SUB <AE>,Dn
SUB Dn,<AE>

PORTEE:

Octet (.B), Mot (.W), Mot Long (.L)

FONCTION:

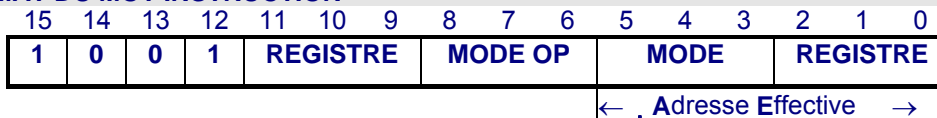
Soustrait en Binaire Naturel l'opérande source à l'opérande destination et place le résultat dans la destination.

CODES CONDITION:

X	N	Z	V	C
A	A	A	A	A

- X Identique à C
- N Affiche le signe du résultat (bit de poids fort)
- Z Mis à 1 si résultat nul sinon à 0
- V Mis à 1 si débordement en complément à 2 sinon 0
- C Mis à 1 si un report décimal est généré sinon mis à 0

FORMAT DU MOT INSTRUCTION



CHAMP MODE OP			
OCTET	MOT	MOT LONG	OPERATION
000	001	010	<Dn> - <AE> dans <Dn>
100	101	110	<AE> - <Dn> dans <AE>

Le champ **Adresse effective** est codé, en fonction de l'adressage utilisé, suivant le tableau ci dessous:

NOTA	MODE D'ADRESS	MODE	REGISTRE	NOTA	MODE D'ADRESS	MODE	REGISTRE
(1)	Dn	000	N° Registre		d(An,Xi)	110	N° Registre
(1 + 2)	An	001	N° Registre		Absolu Court	111	000
	(An)	010	N° Registre		Absolu Long	111	001
	(An)+	011	N° Registre	(1)	d(PC)	111	010
	-(An)	100	N° Registre	(1)	d(PC,Xi)	111	011
	d(An)	101	N° Registre	(1)	Immédiat	111	100

- NOTA (1) Mode d'adressage utilisable en source seul
(2) Mode d'adressage n'acceptant pas les opérandes octet

EXEMPLE D'UTILISATION

SUB.B D2,D3 mot instruction: 0X9602

AVANT EXECUTION

D2	XX	XX	XX	77
D3	44	55	66	82

X	N	Z	V	C
0	0	0	0	0

APRES EXECUTION

D2	XX	XX	XX	77
D3	44	55	66	0B

X	N	Z	V	C
0	0	0	1	0

SUBA

SUBA

SOUSTRACTION D'ADRESSES (SUBSTRACT ADDRESS)

OPERATION:

An - SOURCE dans An

SYNTAXE ASSEMBLEUR:

SUBA <AE>,An

PORTEE:

Mot (.W), Mot Long (.L)

FONCTION:

Soustrait en Binaire Naturel l'opérande source à l'opérande destination et place le résultat dans la destination. Bien que l'opération puisse se faire sur un mot ou un mot long, le registre destination est influencé dans son entier. Lorsque la soustraction porte sur un mot, le signe de celui-ci est étendu à 32 bits avant l'opération.

CODES CONDITION:

X	N	Z	V	C
N	N	N	N	N

Le registre de Codes Condition n'est pas affecté par cette instruction.

FORMAT DU MOT INSTRUCTION

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	0	1	REG An			MODE OP			MODE			REGISTRE		
← . Adresse Effective →															

CHAMP MODE OP			
OCTET	MOT	MOT LONG	OPERATION
Néant	011	111	<An> - <AE> dans <An>

Le champ **Adresse effective** est codé, en fonction de l'adressage utilisé, suivant le tableau ci dessous, dans cette instruction AE est toujours source.

NOTA	MODE D'ADRESS	MODE	REGISTRE	NOTA	MODE D'ADRESS	MODE	REGISTRE
	Dn	000	N° Registre		d(An,Xi)	110	N° Registre
	An	001	N° Registre		Absolu Court	111	000
	(An)	010	N° Registre		Absolu Long	111	001
	(An)+	011	N° Registre		d(PC)	111	010
	-(An)	100	N° Registre		d(PC,Xi)	111	011
	d(An)	101	N° Registre		Immédiat	111	100

EXEMPLE D'UTILISATION

SUBA.W D2,A3
mot instruction: **0X96C2**

AVANT EXECUTION

D2	XX	XX	30	70
A3	00	11	22	33

X	N	Z	V	C
0	0	0	0	0

APRES EXECUTION

D2	XX	XX	30	70
A3	00	10	F1	C3

X	N	Z	V	C
0	0	0	0	0

SUBI

SOUSTRACTION IMMEDIATE (SUBSTRACT IMMEDIAT)

SUBI

OPERATION:

DESTINATION - DONNEE IMMEDIATE dans DESTINATION

SYNTAXE ASSEMBLEUR:

SUBI #DONNEE,<AE>

PORTEE:

Octet(.B), Mot (.W), Mot Long (.L)

FONCTION:

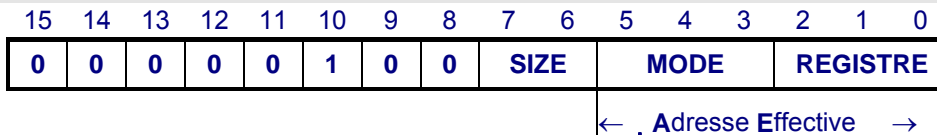
Soustrait en Binaire Naturel la donnée immédiate à l'opérande destination et place le résultat dans la destination. La taille de la donnée immédiate doit être en accord avec la taille de l'opération.

CODES CONDITION:

X	N	Z	V	C
A	A	A	A	A

- X Identique à C
- N Affiche le signe du résultat (bit de poids fort)
- Z Mis à 1 si résultat nul sinon à 0
- V Mis à 1 si débordement en complément à 2 sinon 0
- C Mis à 1 si un report décimal est généré sinon mis à 0

FORMAT DU MOT INSTRUCTION



CHAMP SIZE			
OCTET	MOT	MOT LONG	OPERATION
00	01	10	<AE> - Donnée Immédiate dans AE

Le champ **Adresse effective** est codé, en fonction de l'adressage utilisé, suivant le tableau ci dessous, dans cette instruction AE est toujours destination.

NOTA	MODE D'ADRESS	MODE	REGISTRE	NOTA	MODE D'ADRESS	MODE	REGISTRE
	Dn	000	N° Registre		d(An,Xi)	110	N° Registre
(3)	An				Absolu Court	111	000
	(An)	010	N° Registre		Absolu Long	111	001
	(An)+	011	N° Registre	(3)	d(PC)		
	-(An)	100	N° Registre	(3)	d(PC,Xi)		
	d(An)	101	N° Registre	(3)	Immédiat		

NOTA (3) Mode d'adressage ne pouvant être utilisé

EXEMPLE D'UTILISATION

SUBI.B #0X77,D3
 mot instruction: **0X0403** mot extension: **0X0077**

AVANT EXECUTION

D2	XX	XX	XX	XX
D3	44	55	66	82

APRES EXECUTION

D2	XX	XX	XX	XX
D3	44	55	66	0B

X	N	Z	V	C
0	0	0	0	0

X	N	Z	V	C
0	0	0	1	0

SUBQ SOUSTRACTION IMMEDIATE RAPIDE SUBQ

(SUBSTRACT QUICK)

OPERATION:

DESTINATION - DONNEE IMMEDIATE dans DESTINATION

SYNTAXE ASSEMBLEUR:

SUBQ #<DONNEE>,<AE>

PORTEE:

Octet(.B), Mot (.W), Mot Long (.L)

FONCTION:

Soustrait en Binaire Naturel la donnée immédiate à l'opérande destination et place le résultat dans la destination. La taille de la donnée immédiate est de 1 à 8. L'opération peut être effectuée sur un registre d'adresses mais alors, elle **ne peut porter sur un octet et n'influence pas CCR**.

CODES CONDITION:

X	N	Z	V	C
A	A	A	A	A

X Identique à C

N Affiche le signe du résultat (bit de poids fort)

Z Mis à 1 si résultat nul sinon à 0

V Mis à 1 si débordement en complément à 2 sinon 0

C Mis à 1 si un report décimal est généré sinon mis à 0

FORMAT DU MOT INSTRUCTION

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	1	DONNEE			1	SIZE		MODE			REGISTRE		
← , Adresse Effective →															

CHAMP SIZE			
OCTET	MOT	MOT LONG	OPERATION
00	01	10	<AE> - Donnée Immédiate dans AE

Le champ **Adresse effective** est codé, en fonction de l'adressage utilisé, suivant le tableau ci dessous, dans cette instruction AE est toujours destination.

NOTA	MODE D'ADRESS	MODE	REGISTRE	NOTA	MODE D'ADRESS	MODE	REGISTR E
	Dn	000	N° Registre		d(An,Xi)	110	N° Registre
(2)	An	001	N° Registre		Absolu Court	111	000
	(An)	010	N° Registre		Absolu Long	111	001
	(An)+	011	N° Registre	(3)	d(PC)		
	-(An)	100	N° Registre	(3)	d(PC,Xi)		
	d(An)	101	N° Registre	(3)	Immédiat		

NOTA (2) Mode d'adressage n'acceptant pas les opérandes octet, utilisé, n'influence pas CCR

(3) Mode d'adressage ne pouvant être utilisé

EXEMPLE D'UTILISATION**SUBQ.B #3,D3** mot instruction: **0X5703**

AVANT EXECUTION

D2	XX	XX	XX	XX
D3	44	55	66	82

APRES EXECUTION

D2	XX	XX	XX	XX
D3	44	55	66	7F

X	N	Z	V	C
0	0	0	0	0

X	N	Z	V	C
0	0	0	1	0

SUBX

SUBX

SOUSTRACTION AVEC EXTENSION (SUBTRACT WITH EXTEND)

OPERATION:

DESTINATION - SOURCE - X dans DESTINATION

SYNTAXE ASSEMBLEUR:

SUBX Dy,Dx
SUBX -(Ay),-(Ax)

PORTEE:

Octet(.B), Mot (.W), Mot Long (.L)

FONCTION:

Soustrait en Binaire Naturel l'opérande source et le bit X à l'opérande destination et place le résultat dans la destination. Cette instruction ne peut être exécutée que dans deux modes d'adressage, entre registres de données ou entre cases mémoire en adressage indirect par registres d'adresses pré-décrémenté.

CODES CONDITION:

X	N	Z	V	C
A	A	A	A	A

- X Identique à C
- N Affiche le signe du résultat (bit de poids fort)
- Z Mis à 1 si résultat nul sinon à 0
- V Mis à 1 si débordement en complément à 2 sinon 0
- C Mis à 1 si un report décimal est généré sinon mis à 0

FORMAT DU MOT INSTRUCTION

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	0	1	Rx			1	SIZE	0	0	R/M	Ry			

Le bit **R/M** indique le mode d'adressage 0 Registres de données
1 Entre cases mémoire

Le champ **Ry** N° du registre source (Dy ou Ay)

Le champ **Rx** N° du registre destination (Dx ou Ax)

CHAMP SIZE			
OCTET	MOT	MOT LONG	OPERATION
00	01	10	Destination - Source - X dans Destination

EXEMPLE D'UTILISATION

SUBX.B D2,D3
mot instruction: **0X9702**

AVANT EXECUTION

D2	XX	XX	XX	77
D3	44	55	66	82

X	N	Z	V	C
1	0	0	0	0

APRES EXECUTION

D2	XX	XX	XX	77
D3	44	55	66	0A

X	N	Z	V	C
0	0	0	1	0

SWAP

SWAP

PERMUTATION DES MOTS D'UN REGISTRE (SWAP REGISTERS HALVES)

OPERATION:

Permutation mot de poids fort mot de poids faible dans un registre de données

SYNTAXE ASSEMBLEUR:

SWAP Dn

PORTEE:

Mot long

FONCTION:

Permutation mot de poids fort mot de poids faible dans un registre de données

CODES CONDITION:

X	N	Z	V	C
N	A	A	0	0

- X N'est pas affecté par l'opération
- N Affiche le signe du résultat (bit de poids fort)
- Z Mis à 1 si résultat nul sinon à 0
- V Toujours mis à 0
- C Toujours mis à 0

FORMAT DU MOT INSTRUCTION

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	1	0	0	0	0	1	0	0	0	0	N° reg Dn	

EXEMPLE D'UTILISATION

SWAP D3
mot instruction: **0x4843**

AVANT EXECUTION

D2	XX	XX	XX	XX
D3	11	22	AA	BB

X	N	Z	V	C
0	0	0	0	0

APRES EXECUTION

D2	XX	XX	XX	XX
D3	AA	BB	11	22

X	N	Z	V	C
0	1	0	0	0

TAS

TAS

TESTE ET POSITIONNE UN OPERANDE (TEST AND SET AN OPERAND)

OPERATION:

Positionnement de CCR en fonction de l'Opérande testé
1 dans le MSB de l'opérande

SYNTAXE ASSEMBLEUR:

TAS <AE>

PORTEE:

Octet

FONCTION:

Teste un opérande **octet** et positionne les bits N et Z du registre de codes condition puis positionne le bit de poids fort de l'octet testé à 1.

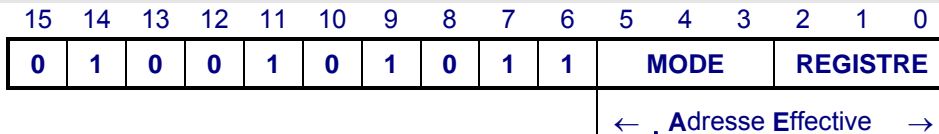
Cette instruction est du type **Lecture Modification Ecriture indivisible**

CODES CONDITION:

X	N	Z	V	C
N	A	A	0	0

- X N'est pas affecté par l'opération
- N Affiche le signe du résultat (bit de poids fort)
- Z Mis à 1 si résultat nul sinon à 0
- V Toujours mis à 0
- C Toujours mis à 0

FORMAT DU MOT INSTRUCTION



Le champ **Adresse effective** est codé, en fonction de l'adressage utilisé, suivant le tableau ci dessous

NOTA	MODE D'ADRESS	MODE	REGISTRE	NOTA	MODE D'ADRESS	MODE	REGISTRE
	Dn	000	N° Registre		d(An,Xi)	110	N° Registre
(3)	An				Absolu Court	111	000
	(An)	010	N° Registre		Absolu Long	111	001
	(An)+	011	N° Registre	(3)	d(PC)		
	-(An)	100	N° Registre	(3)	d(PC,Xi)		
	d(An)	101	N° Registre	(3)	Immédiat		

NOTA (3) Mode d'adressage ne pouvant être utilisé

EXEMPLE D'UTILISATION

TAS D3
mot instruction: **0X4AC3**

AVANT EXECUTION

D2	XX	XX	XX	XX
D3	44	55	66	00

X	N	Z	V	C
0	0	0	0	0

APRES EXECUTION

D2	XX	XX	XX	XX
D3	44	55	66	80

X	N	Z	V	C
0	0	1	0	0

TRAP

TRAP

TRAPPE INCONDITIONNELLE (TRAP)

OPERATION:

SSP - 4 → SSP PC dans la pile

SSP - 2 → SSP SR dans la pile puis Adresse Vecteur dans PC

SYNTAXE ASSEMBLEUR:

TRAP #<N°>

PORTEE:

Sans

FONCTION:

Le processeur amorce une exception Trappe. Le N° de vecteur fourni est utilisé pour rechercher le vecteur TRAP - (32 + N°)*4 - Seize N° de trappe sont utilisables (0 à 15).

CODES CONDITION:

X	N	Z	V	C
N	N	N	N	N

Le registre de Codes Condition n'est pas affecté par cette instruction

FORMAT DU MOT INSTRUCTION

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	1	1	1	0	0	1	0	0	N° TRAPPE			

TRAPV

TRAPV

TRAPPE SUR OVERFLOW (TRAP ON OVERFLOW)

OPERATION:

Si V=1 alors TRAPPE

SYNTAXE ASSEMBLEUR:

TRAPV

PORTEE:

Sans

FONCTION:

Si le bit d'overflow est égal à 1, le processeur amorce une exception Trappe. Le N° de vecteur fourni est 07.

CODES CONDITION:

X	N	Z	V	C
N	N	N	N	N

Le registre de Codes Condition n'est pas affecté par cette instruction

FORMAT DU MOT INSTRUCTION

\$4E76															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	1	1	1	0	0	1	1	1	0	1	1	0

TST

TST

TESTE UN OPERANDE (TEST AN OPERAND)

OPERATION:

Positionnement de CCR en fonction de l'Opérande testé

SYNTAXE ASSEMBLEUR:

TST <AE>

PORTEE:

Octet (.B), Mot (.W), Mot Long (.L)

FONCTION:

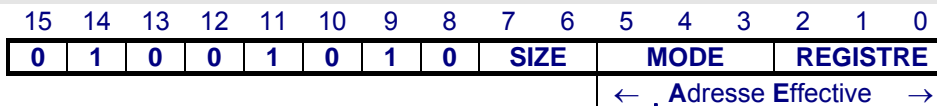
Teste un opérande et positionne les bits N et Z du registre de codes condition.

CODES CONDITION:

X	N	Z	V	C
N	A	A	0	0

- X N'est pas affecté par l'opération
- N Affiche le signe du résultat (bit de poids fort)
- Z Mis à 1 si résultat nul sinon à 0
- V Toujours mis à 0
- C Toujours mis à 0

FORMAT DU MOT INSTRUCTION



CHAMP SIZE			
OCTET	MOT	MOT LONG	OPERATION
00	01	10	TESTE <AE> et positionne CCR

Le champ **Adresse effective** est codé, en fonction de l'adressage utilisé, suivant le tableau ci dessous, dans cette instruction AE est toujours destination.

NOTA	MODE D'ADRESS	MODE	REGISTRE	NOTA	MODE D'ADRESS	MODE	REGISTRE
	Dn	000	N° Registre		d(An,Xi)	110	N° Registre
(3)	An				Absolu Court	111	000
	(An)	010	N° Registre		Absolu Long	111	001
	(An)+	011	N° Registre	(3)	d(PC)		
	-(An)	100	N° Registre	(3)	d(PC,Xi)		
	d(An)	101	N° Registre	(3)	Immédiat		

NOTA (3) Mode d'adressage ne pouvant être utilisé

EXEMPLE D'UTILISATION

TST.B D3
mot instruction: **0X4A03**

AVANT EXECUTION

D2	XX	XX	XX	XX
D3	44	55	66	00

APRES EXECUTION

D2	XX	XX	XX	XX
D3	44	55	66	00

X	N	Z	V	C
0	0	0	0	0

X	N	Z	V	C
0	0	1	0	0

UNLK

UNLK

DEFAIRE UN LIEN (UNLINK)

OPERATION:

An → SP
 (SP) → An
 SP + 4 → SP

SYNTAXE ASSEMBLEUR:

UNLK An

PORTEE:

Sans

FONCTION:

Retour après utilisation de la fonction Link.

Le contenu du registre d'adresses An (ex contenu de SP) est rechargé dans SP, ce qui annule le déplacement qui avait été ajouté à SP dans l'instruction Link, le mot long pointé par SP dans le pile

(ex valeur de An sauvegardée) est réintégré dans An et SP revient à sa position d'origine.

CODES CONDITION:

X	N	Z	V	C
N	N	N	N	N

Le registre de Codes Condition n'est pas affecté par cette instruction

FORMAT DU MOT INSTRUCTION

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	1	1	1	0	0	1	0	1	1	N° regist An		

EXEMPLE D'UTILISATION

	UNLK				A5				
	mot instruction:				0x4E5D				
	AVANT EXECUTION					APRES EXECUTION			
SP	00	01	7F	FA	SP	00	01	80	04
A5	00	01	80	00	A5	00	00	22	44

MEMOIRES

ADRESSES	CONTENU		ADRESSES	CONTENU
00 01 80 04	AB		SP → 00 01 80 04	AB
00 01 80 03	00		00 01 80 03	44
00 01 80 02	00		00 01 80 02	22
00 01 80 01	22		00 01 80 01	00
00 01 80 00	44	← A5	00 01 80 00	00
00 01 7F FF	XX		00 01 7F FF	XX
00 01 7F FE	XX		00 01 7F FE	XX
00 01 7F FD	XX		00 01 7F FD	XX
00 01 7F FC	XX		00 01 7F FC	XX
00 01 7F FB	XX		00 01 7F FB	XX
00 01 7F FA	XX	← SP	00 01 7F FA	XX
00 01 7F F9	XX		00 01 7F F9	XX