

ASL Arithmetic Shift Left



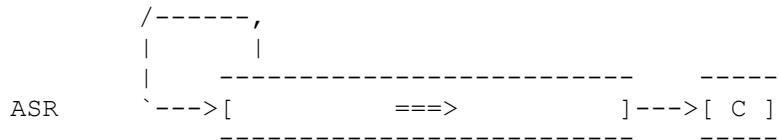
Descript.: Provoque un décalage de l'opérande de source de un ou plusieurs bits à gauche (direction MSB). Les bits des poids les plus faibles sont remplacés par des zéros. L'opération est équivalente à SL, mais l'indicateur de dépassement peut être modifié autrement. Si le signe du résultat est différent de celui de l'opérande de source, l'indicateur de dépassement doit être positionné.

Syntaxe: ASL.id sourcedestination
 ASL.id amplitude, sourcedestination
 ASL.id amplitude, source, destination
 ASL amplitude.id, sourcedestination.id
 ASL amplitude.id, source.id, destination.id

Comment.: Une amplitude négative signifie, que le décalage se fait à droite. Un décalage à gauche multiplie par deux. ASL.16 sourcedestination est équivalent à SL.A16 sourcedestination.

Exemple: ASL.16 CL,BX ; iAPX86

ASR Arithmetic Shift Right



Descript.: Provoque un décalage de l'opérande de source de un ou plusieurs bits à droite (direction LSB), mais le bit de poids le plus fort (MSB, signe) est conservé et dupliqué vers la droite. L'opération est équivalente à une division par deux pour des nombres positifs (les nombres négatifs ne sont pas arrondis dans la même direction).

Syntaxe: ASR.id sourcedestination
 ASR.id amplitude, sourcedestination
 ASR.id amplitude, source, destination
 ASR amplitude.id, sourcedestination.id
 ASR amplitude.id, source.id, destination.id

Comment.: Une amplitude négative signifie, que le décalage se fait à gauche. ASR.16 sourcedestination est équivalent à SR.A16 sourcedestination.

Exemples: ASR B ; 6809 (ASR #1,B)
 ASR.16 CL,BX ; iAPX86

CALL CALL

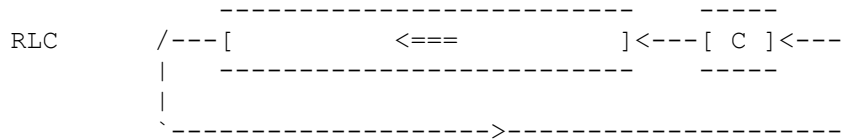
Descript.: Sauve l'adresse de retour sur la pile et saute au sous-programme à l'adresse donnée.

Syntaxe: CALL adresse_de_saut
 CALL,c adresse_de_saut

Comment.: L'instruction RET termine le sous-programme.

Exemple: CALL ADRESSE

RLC Rotate Left with Carry



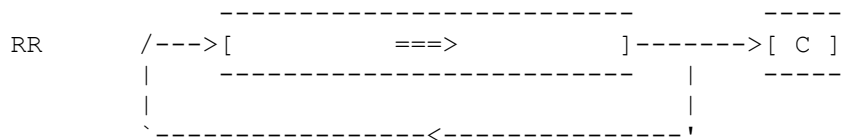
Descript.: Effectue un décalage de l'opérande de source d'un ou plusieurs bits à gauche. A chaque décalage, le LSB est remplacé par l'indicateur de report C et l'indicateur de report C par le MSB.

Syntaxe: RLC.id sourcedestination
 RLC.id amplitude, sourcedestination
 RLC.id amplitude, source, destination
 RLC amplitude.id, sourcedestination.id
 RLC amplitude.id, source.id, destination.id

Comment.: Une amplitude négative signifie, que le décalage se fait à droite.

Exemples: RLC B ; 6809 (RLC #1,B), ...
 RLC.16 CL,BX ; iAPX86 (RLC CL.8,BX.16)

RR Rotate Right



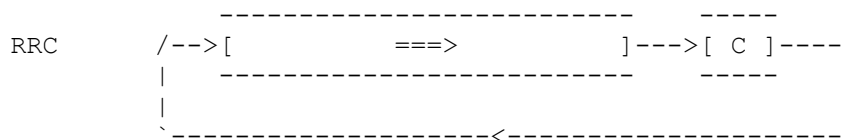
Descript.: Effectue un décalage de l'opérande de source d'un ou plusieurs bits à droite. A chaque décalage, le MSB est remplacé par le LSB.

Syntaxe: RR.id sourcedestination
 RR.id amplitude, sourcedestination
 RR.id amplitude, source, destination
 RR amplitude.id, sourcedestination.id
 RR amplitude.id, source.id, destination.id

Comment.: Une amplitude négative signifie, que le décalage se fait à gauche.

Exemples: RR B ; 6809 (RR #1,B), ...
 RR.16 CL,BX ; iAPX86 (RR CL.8,BX.16)

RRC Rotate Right with Carry



Descript.: Effectue un décalage de l'opérande de source d'un ou plusieurs bits à droite. A chaque décalage, le MSB est remplacé par l'indicateur de report C et l'indicateur de report C par le LSB.

Syntaxe: RRC.id sourcedestination
 RRC.id amplitude, sourcedestination
 RRC.id amplitude, source, destination
 RRC amplitude.id, sourcedestination.id
 RRC amplitude.id, source.id, destination.id

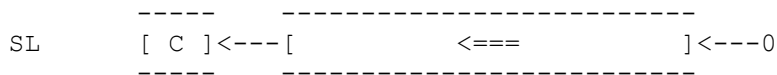
Comment.: Une amplitude négative signifie, que le décalage se fait à gauche.

Exemples: RRC B ; 6809 (RRC #1,B), ...
 RRC.16 CL,BX ; iAPX86 (RRC CL.8,BX.16)

SET SET
 Descript.: Initialise l'opérande de destination avec des uns (tous les bits sont mis à un). Instruction équivalente: MOVE #-1,destination.
 Syntaxe: SET.id destination
 Comment.: Si une instruction SET est conditionnelle, l'opérande est initialisé avec des uns si la condition est vraie, et initialisé avec des zéros si la condition est fausse (voir aussi CLR).
 Exemples: SET.8,EQ {A0} ; 68000
 SET {HL}:#4 ; Z80
 SETC ; 8080, Z80, ...

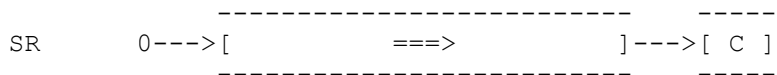
SKIP SKIP
 Descript.: Saute par dessus l'instruction suivante.
 Syntaxe: SKIP,c
 Comment.: L'instruction SKIP est pratiquement toujours combinée avec une condition.
 Exemple: SKIP,EQ DJ.16,NMO D0,ADRESSE ; 68000

SL Shift Left



Descript.: Effectue un décalage de l'opérande de source d'un ou plusieurs bits à gauche. A chaque décalage, le LSB est remplacé par un zéro.
 Syntaxe: SL.id sourcedestination
 SL.id amplitude, sourcedestination
 SL.id amplitude, source, destination
 SL amplitude.id, sourcedestination.id
 SL amplitude.id, source.id, destination.id
 Comment.: Une amplitude négative signifie, que le décalage se fait à droite. Un décalage à gauche multiplie par deux.
 Exemples: SL B ; 6809 (SL #1,B), ...
 SL.16 CL,BX ; iAPX86 (SL CL.8,BX.16)

SR Shift Right



Descript.: Effectue un décalage de l'opérande de source d'un ou plusieurs bits à droite. A chaque décalage, le MSB est remplacé par un zéro.
 Syntaxe: SR.id sourcedestination
 SR.id amplitude, sourcedestination
 SR.id amplitude, source, destination
 SR amplitude.id, sourcedestination.id
 SR amplitude.id, source.id, destination.id
 Comment.: Une amplitude négative signifie, que le décalage se fait à gauche. Un décalage à droite divise par deux.
 Exemples: SR B ; 6809 (SR #1,B), ...
 SR.16 CL,BX ; iAPX86 (SR CL.8,BX.16)

SUB SUBtract
 Descript.: Soustrait le premier opérande du second opérande.
 Syntaxe: SUB.id source, sourcedestination
 SUB.id source1, source2, destination
 SUB source.id, sourcedestination.id
 SUB source1.id, source2.id, destination.id
 Exemples: SUB #2'1010,A ; 8080, Z80, ...
 SUB.32 {A6}+400,D6 ; 68000
 SUB.F64 F0,{SB}+10 ; NS32000

SUBC SUBtract with Carry
 Descript.: Soustrait le premier opérande et l'indicateur de report C du second opérande.
 Syntaxe: SUBC.id source, sourcedestination
 SUBC.id source1, source2, destination
 SUBC source.id, sourcedestination.id
 SUBC source1.id, source2.id, destination.id
 Exemples: SUBC #2'1010,A ; 8080, Z80, ...
 SUBX.32 {-A4},{-A3} ; 68000 (X est ici C)
 SUBC.32 R0,{SB}+10 ; NS32000

SWAP SWAP
 Descript.: Echange les deux moitiés de l'opérande. La taille associée se réfère à la taille entière de l'opérande.
 Syntaxe: SWAP.id sourcedestination
 Comment.: L'instruction SWAP est nécessaire, si l'on peut pas accéder indépendamment aux sous-unités d'un opérande.
 Exemple: SWAP.32 D4 ; 68000

TCLR Test and CLear
 Descript.: Teste et met à zéro le(s) bit(s) d'opérande.
 Syntaxe: TCLR.id sourcedestination (:adresse_de_bit(.id))
 TCLR sourcedestination.id (:adresse_de_bit(.id))
 Comment.: L'indicateur de nullité Z contient souvent la valeur du bit avant l'exécution de l'instruction. La lettre supplémentaire I signale que l'instruction est indivisible. L'indicateur de données, ajouté au code opératoire ou à l'opérande d'adresse d'octet, indique que l'adressage de bit est limité.
 Exemples: TCLR.32 D5:#31 ; 68000 (bit: 0..31)
 TCLR.8 {A6}+CONDITION:#2 ; 68000 (bit: 0..7)
 TCLRI {SB}:{FP}.A8 ; NS32000 (bit: -128..+127)

TEST TEST
 Descript.: Teste le signe et la valeur (si zéro) de l'opérande.
 Syntaxe: TEST.id source (:adresse_de_bit(.id))
 TEST source.da (:adresse_de_bit(.id))
 TEST.id source1, source2
 Comment.: TEST opérande est équivalent à COMP #0,opérande. L'indicateur de nullité Z est positionné si la valeur de l'opérande est zéro. Le MSB est copié dans l'indicateur de signe N. Avec deux opérandes, une opération AND est effectuée avant le test.
 Exemples: TEST B ; 6800, ...
 TEST.32 D5:#31 ; 68000 (bit: 0..31)
 TEST.8 {A6}+CONDITION:#2 ; 68000 (bit: 0..7)
 TEST {SB}:{FP}.A8 ; NS32000 (bit: -128..+127)
 TEST.16 [DS]+{SI},AX ; iAPX86

TNOT Test and NOT

Descript.: Teste et invertit le(s) bit(s) d'opérande.

Syntaxe: TNOT.id sourcedestination (:adresse_de_bit(.id))
TNOT sourcedestination.id (:adresse_de_bit(.id))

Comment.: L'indicateur de nullité Z contient souvent la valeur du bit avant l'exécution de l'instruction. L'indicateur de données après le code opératoire ou l'opérande d'adresse d'octet indique que l'adressage de bit est limité.

Exemples: TNOT.32 D5:#31 ; 68000 (bit: 0..31)
TNOT.8 {A6}+CONDITION:#2 ; 68000 (bit: 0..7)
TNOT {SB}:{FP}.A8 ; NS32000 (bit: -128..+127)

TRAP TRAP

Descript.: Appel de sous-programme spécial.

Syntaxe: TRAP expression
TRAP,c expression

Comment.: Les instructions TRAP peuvent être déclenchées par programmation ou par matériel (p.ex. division par zéro).

Exemples: TRAP ; 6800, ...
TRAP #15 ; 68000
TRAP,VS ; 68000

TSET Test and SET

Descript.: Teste et met à un le(s) bit(s) d'opérande.

Syntaxe: TSET.id sourcedestination (:adresse_de_bit(.id))
TSET sourcedestination.id (:adresse_de_bit(.id))

Comment.: L'indicateur de nullité Z contient souvent la valeur du bit avant l'exécution de l'instruction. La lettre supplémentaire I signale que l'instruction est indivisible. L'indicateur de données, ajouté après le code opératoire ou l'opérande d'adresse d'octet, indique que l'adressage de bit est limité.

Exemples: TSET.32 D5:#31 ; 68000 (bit: 0..31)
TSET.8 {A6}+CONDITION:#2 ; 68000 (bit: 0..7)
TSETI {SB}:{FP}.A8 ; NS32000 (bit: -128..+127)

WAIT WAIT

Descript.: Attend sur une interruption.

Syntaxe: WAIT

Comment.: L'instruction WAIT minimise le temps de réponse à une requête d'interruption. Après une interruption et après l'exécution du sous-programme d'interruption, le processeur exécute l'instruction suivante (après WAIT). Il y a quatre variantes du WAIT: WAIT (CPU attend activement), SLEEP (le courant est coupé des E/S), STOP (l'oscillateur interne est arrêté) et HALT (CPU s'arrête).

Exemple: WAIT

XOR XOR

Descript.: Effectue un OU exclusif bit à bit des deux opérandes de source.

Syntaxe: XOR.id source, sourcedestination
XOR.id source1, source2, destination
XOR source.id, sourcedestination.id
XOR source1.id, source2.id, destination.id

Comment.: Cette instruction sert à invertir des bits.

Exemples: XOR #1,A ; 8080, Z80, ...
XOR.32 #16'FFFF0000,D6 ; 68000