

- I) Les instructions logiques :

- AND

Cette instruction réalise un ET logique bit à bit entre le contenu d'un accumulateur (A ou B) et une case mémoire, puis place le résultat dans ce même accumulateur.

Voici les bits du CCR qui sont modifiés :

- N : passe à 1 si le résultat à son bit de poids fort à 1.
- Z : passe à 1 si le résultat est nul.
- V : passe à 0

- ORA

Cette instruction réalise un OU logique bit à bit entre le contenu d'un accumulateur (A ou B) et une case mémoire, puis place le résultat dans ce même accumulateur.

Voici les bits du CCR qui sont modifiés :

- N : passe à 1 si le résultat à son bit de poids fort à 1.
- Z : passe à 1 si le résultat est nul.
- V : passe à 0

- EOR

Cette instruction réalise un OU EXCLUSIF logique bit à bit entre le contenu d'un accumulateur (A ou B) et une case mémoire, puis place le résultat dans ce même accumulateur.

Voici les bits du CCR qui sont modifiés :

- N : passe à 1 si le résultat à son bit de poids fort à 1.
- Z : passe à 1 si le résultat est nul.
- V : passe à 0

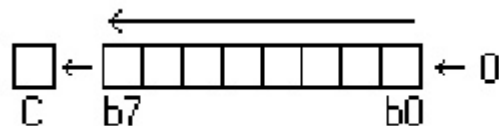
- COM

Cette instruction réalise une COMPLEMENTATION bit à bit d'un accumulateur (A ou B) ou d'une case mémoire, puis place le résultat dans ce même accumulateur, ou case mémoire.

Voici les bits du CCR qui sont modifiés :

- N : passe à 1 si le résultat à son bit de poids fort à 1.
- Z : passe à 1 si le résultat est nul.
- V : passe à 0
- C : passe à 1
- II) Décalages et rotations :
 - o ASL

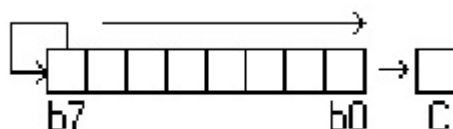
Cette instruction réalise un DECALAGE vers la gauche d'un accumulateur (A ou B) ou d'une case mémoire, puis place le résultat dans ce même accumulateur, ou case mémoire, il rentre un 0 dans le poids faible



Voici les bits du CCR qui sont modifiés :

- N : passe à 1 si le résultat à son bit de poids fort à 1.
- Z : passe à 1 si le résultat est nul.
- V : passe à 1 si après l'opération, N est à 1 et C est à 0 ou si N est à 0 et C est à 1.
- C : passe à 1 si, avant l'opération le bit de poids fort de l'opérande était égal à 1.
- o ASR

Cette instruction réalise un DECALAGE vers la droite d'un accumulateur (A ou B) ou d'une case mémoire, puis place le résultat dans ce même accumulateur, ou case mémoire, b0 (poids faible) sort dans la CARRY et b7 reste inchangé

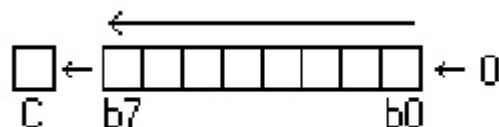


Voici les bits du CCR qui sont modifiés :

- N : passe à 1 si le résultat à son bit de poids fort à 1.
- Z : passe à 1 si le résultat est nul.
- V : passe à 1 si après l'opération, N est à 1 et C est à 0 ou si N est à 0 et C est à 1.
- C : passe à 1 si, avant l'opération le bit de poids faible de l'opérande était égal à 1.

o LSL

Cette instruction réalise un DECALAGE vers la gauche d'un accumulateur (A ou B) ou d'une case mémoire, puis place le résultat dans ce même accumulateur, ou case mémoire, b7 (poids fort) sort dans la CARRY et un 0 rentre dans b0 (poids faible), cette instruction est identique à ASL.

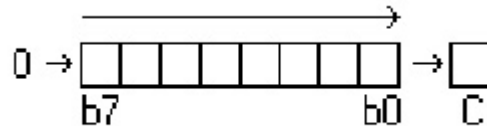


Voici les bits du CCR qui sont modifiés :

- N : passe à 1 si le résultat à son bit de poids fort à 1.
- Z : passe à 1 si le résultat est nul.
- V : passe à 1 si après l'opération, N est à 1 et C est à 0 ou si N est à 0 et C est à 1.
- C : passe à 1 si, avant l'opération le bit de poids fort de l'opérande était égal à 1.

o LSR

Cette instruction réalise un DECALAGE vers la droite d'un accumulateur (A ou B) ou d'une case mémoire, puis place le résultat dans ce même accumulateur, ou case mémoire, b0 (poids faible) sort dans la CARRY et un 0 rentre dans b7 (poids fort)

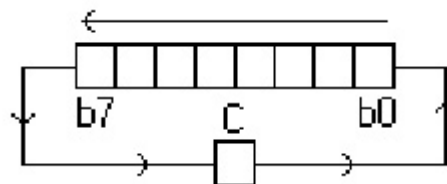


Voici les bits du CCR qui sont modifiés :

- N : passe à 0.
- Z : passe à 1 si le résultat est nul.
- V : passe à 1 si après l'opération, N est à 1 et C est à 0 ou si N est à 0 et C est à 1.
- C : passe à 1 si, avant l'opération le bit de poids faible de l'opérande était égal à 1.

o ROL

Cette instruction réalise un DECALAGE vers la gauche d'un accumulateur (A ou B) ou d'une case mémoire, puis place le résultat dans ce même accumulateur, ou case mémoire, b7 (poids fort) sort dans la CARRY et la CARRY précédente rentre dans b0 (poids faible)

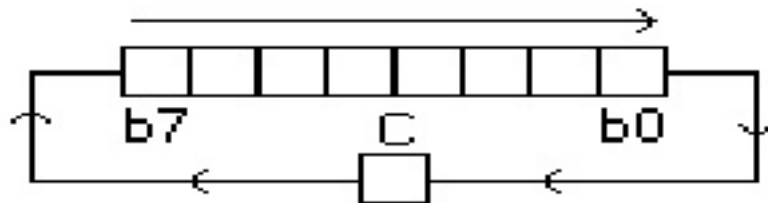


Voici les bits du CCR qui sont modifiés :

- N : passe à 1 si le résultat à son bit de poids fort à 1.
- Z : passe à 1 si le résultat est nul.
- V : passe à 1 si après l'opération, N est à 1 et C est à 0 ou si N est à 0 et C est à 1.
- C : passe à 1 si, avant l'opération le bit de poids fort de l'opérande était égal à 1.

o ROR

Cette instruction réalise un DECALAGE vers la droite d'un accumulateur (A ou B) ou d'une case mémoire, puis place le résultat dans ce même accumulateur, ou case mémoire, b0 (poids faible) sort dans la CARRY et la CARRY précédente rentre dans b7 (poids fort)



Voici les bits du CCR qui sont modifiés :

- N : passe à 1 si le résultat à son bit de poids fort à 1.
- Z : passe à 1 si le résultat est nul.
- V : passe à 1 si après l'opération, N est à 1 et C est à 0 ou si N est à 0 et C est à 1.
- C : passe à 1 si, avant l'opération le bit de poids faible de l'opérande était égal à 1.

• III) Tests et manipulations de bits :

o BIT

Cette instruction réalise un ET logique entre le contenu d'un accumulateur (A ou B) et le contenu d'une case mémoire, puis positionne les bits du CCR s'il y a lieu, sans modifier le contenu de l'accumulateur ni de la mémoire.

Voici les bits du CCR qui sont modifiés :

- N : passe à 1 si le résultat de poids fort du ET est à 1.
- Z : passe à 1 si le résultat du ET est nul.
- V : passe à 0.

o BCLR

Cette instruction réalise un POSITIONNEMENT à 0 des bits spécifiés dans le masque par des 1, les autres bits ne subissent aucune modification.

Voici les bits du CCR qui sont modifiés :

- N : passe à 1 si le résultat de poids fort du ET est à 1.
- Z : passe à 1 si le résultat est nul.
- V : passe à 0.

o BSET

Cette instruction réalise un POSITIONNEMENT à 1 des bits spécifiés dans le masque par des 1, les autres bits ne subissent aucune modification.

Voici les bits du CCR qui sont modifiés :

- N : passe à 1 si le résultat de poids fort du ET est à 1.
- Z : passe à 1 si le résultat est nul.
- V : passe à 0.

• IV) Opérations arithmétiques :

* D'addition

o ABA

Cette instruction réalise une addition de A et B et place le résultat dans A.

Voici les bits du CCR qui sont modifiés :

- H : passe à 1 s'il y a retenue du bit 3 vers le bit 4.
- N : passe à 1 si le bit de poids fort du résultat est à 1.
- Z : passe à 1 si le résultat est nul.
- V : passe à 1 s'il y a un débordement arithmétique en complément à 2.
- C : passe à 1 s'il y a retenue depuis le bit 7.

o ABX

Cette instruction réalise une addition de B (considéré comme non signé) et X et place le résultat dans X.
Les bits du CCR ne sont pas modifiés.

o ABY

Cette instruction réalise une addition de B (considéré comme non signé) et Y et place le résultat dans Y.
Les bits du CCR ne sont pas modifiés.

o ADCA

Cette instruction réalise une addition de A, de la Carry et d'une case mémoire et place le résultat dans A
Voici les bits du CCR qui sont modifiés :

- H : passe à 1 s'il y a retenue du bit 3 vers le bit 4.
- N : passe à 1 si le bit de poids fort du résultat est à 1.
- Z : passe à 1 si le résultat est nul.
- V : passe à 1 s'il y a un débordement arithmétique en complément à 2.
- C : passe à 1 s'il y a retenue depuis le bit 7.

o ADCB

Cette instruction réalise une addition de B, de la Carry et d'une case mémoire et place le résultat dans B
Voici les bits du CCR qui sont modifiés :

- H : passe à 1 s'il y a retenue du bit 3 vers le bit 4.
- N : passe à 1 si le bit de poids fort du résultat est à 1.
- Z : passe à 1 si le résultat est nul.
- V : passe à 1 s'il y a un débordement arithmétique en complément à 2.
- C : passe à 1 s'il y a retenue depuis le bit 7.

o ADDA

Cette instruction réalise une addition de A et du contenu d'une case mémoire et place le résultat dans A.

Voici les bits du CCR qui sont modifiés :

- H : passe à 1 s'il y a retenue du bit 3 vers le bit 4.
- N : passe à 1 si le bit de poids fort du résultat est à 1.
- Z : passe à 1 si le résultat est nul.
- V : passe à 1 s'il y a un débordement arithmétique en complément à 2.
- C : passe à 1 s'il y a retenue depuis le bit 7.

o ADDB

Cette instruction réalise une addition de B et du contenu d'une case mémoire et place le résultat dans B.

Voici les bits du CCR qui sont modifiés :

- H : passe à 1 s'il y a retenue du bit 3 vers le bit 4.
- N : passe à 1 si le bit de poids fort du résultat est à 1.
- Z : passe à 1 si le résultat est nul.
- V : passe à 1 s'il y a un débordement arithmétique en complément à 2.
- C : passe à 1 s'il y a retenue depuis le bit 7.

* De soustraction

o SBCA

Cette instruction soustrait la Carry et le contenu d'une case mémoire et place le résultat dans A.

Voici les bits du CCR qui sont modifiés :

- N : passe à 1 si le bit de poids fort du résultat est à 1.
- Z : passe à 1 si le résultat est nul.
- V : passe à 1 s'il y a un débordement arithmétique en complément à 2.
- C : passe à 1 si la valeur absolue de la case mémoire ajoutée à la Carry est supérieure à la valeur de A.

o SBCB

Cette instruction soustrait la Carry et le contenu d'une case mémoire et place le résultat dans B.

Voici les bits du CCR qui sont modifiés :

- N : passe à 1 si le bit de poids fort du résultat est à 1.
- Z : passe à 1 si le résultat est nul.
- V : passe à 1 s'il y a un débordement arithmétique en complément à 2.
- C : passe à 1 si la valeur absolue de la case mémoire ajoutée à la Carry est supérieure à la valeur de B.

o SUBA

Cette instruction soustrait le contenu d'une case mémoire et place le résultat dans A.

Voici les bits du CCR qui sont modifiés :

- N : passe à 1 si le bit de poids fort du résultat est à 1.
- Z : passe à 1 si le résultat est nul.
- V : passe à 1 s'il y a un débordement arithmétique en complément à 2.
- C : passe à 1 si la valeur absolue de la case mémoire est supérieure à la valeur absolue de A.

o SUBB

Cette instruction soustrait le contenu d'une case mémoire et place le résultat dans B.

Voici les bits du CCR qui sont modifiés :

- N : passe à 1 si le bit de poids fort du résultat est à 1.
- Z : passe à 1 si le résultat est nul.
- V : passe à 1 s'il y a un débordement arithmétique en complément à 2.
- C : passe à 1 si la valeur absolue de la case mémoire est supérieure à la valeur absolue de B.

o NEG

Cette instruction réalise un changement de signe ou complément à 2 de l'opérande spécifié (A, B ou le contenu d'une case mémoire)

* D'incréméntation et décrémentation :

o INC

Cette instruction augmente de 1 A (INCA), B (INCB) ou une case mémoire.

Voici les bits du CCR qui sont modifiés :

- N : passe à 1 si le bit de poids fort du résultat est à 1.
- Z : passe à 1 si le résultat est nul.
- V : passe à 1 s'il y a un débordement arithmétique en complément à 2.

o INX

Cette instruction augmente de 1 le contenu de X.

Voici le bit du CCR qui est modifié :

- Z : passe à 1 si le résultat est nul.

o INY

Cette instruction augmente de 1 le contenu de Y.

Voici le bit du CCR qui est modifié :

- Z : passe à 1 si le résultat est nul.

o INS

Cette instruction augmente de 1 le contenu de S.

Aucun bit du CCR n'est modifié

o DEC

Cette instruction diminue de 1 A (DECA), B (DECB) ou une case mémoire.

Voici les bits du CCR qui sont modifiés :

- N : passe à 1 si le bit de poids fort du résultat est à 1.
- Z : passe à 1 si le résultat est nul.
- V : passe à 1 s'il y a un débordement arithmétique en complément à 2.

o DEX

Cette instruction diminue de 1 le contenu de X.

Voici le bit du CCR qui est modifié :

- Z : passe à 1 si le résultat est nul.

o DEY

Cette instruction diminue de 1 le contenu de Y.
Voici le bit du CCR qui est modifié :

- Z : passe à 1 si le résultat est nul.

o DES

Cette instruction diminue de 1 le contenu de S.
Aucun bit du CCR n'est modifié

* De comparaison

o CMP

Cette instruction compare un accumulateur (A ou B) et une valeur (ou d'une adresse pointant sur une case mémoire).
Pour cela, est retranchée la valeur de la case mémoire à celle de l'accumulateur (A - M ou B - M).
Voici les bits du CCR qui sont modifiés :

- N : passe à 1 si le bit de poids fort de la soustraction est à 1.
- Z : passe à 1 si le résultat est nul.
- V : passe à 1 s'il y a un débordement arithmétique en complément à 2.
- C : passe à 1 si la valeur absolue de la case mémoire est supérieure à la valeur de l'accumulateur.

o CPD

Cette instruction compare l'accumulateur D et une valeur 16 bits (ou d'une adresse pointant sur deux cases mémoire).
Pour cela, est retranchée la valeur des cases mémoire à celle de l'accumulateur D.
Voici les bits du CCR qui sont modifiés :

- N : passe à 1 si le bit de poids fort de la soustraction est à 1.
- Z : passe à 1 si le résultat est nul.
- V : passe à 1 s'il y a un débordement arithmétique en complément à 2.
- C : passe à 1 si la valeur absolue de la case mémoire est supérieure à la valeur de l'accumulateur.

o CPX

Cette instruction compare le registre X et une valeur 16 bits (ou d'une adresse pointant sur deux cases mémoire).

Pour cela, est retranchée la valeur des cases mémoire à celle du registre X.

Voici les bits du CCR qui sont modifiés :

- N : passe à 1 si le bit de poids fort de la soustraction est à 1.
- Z : passe à 1 si le résultat est nul.
- V : passe à 1 s'il y a un débordement arithmétique en complément à 2.
- C : passe à 1 si la valeur absolue de la case mémoire est supérieure à la valeur de l'accumulateur.