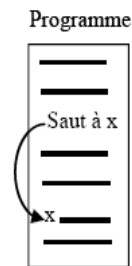


### 3.3.3.5 Les instructions de branchement

Deux types de branchement sont possibles :

- ✓ Branchements inconditionnels
- ✓ Branchements conditionnels

Tous ces branchements provoquent la poursuite de l'exécution du programme à partir d'une nouvelle position du code.



#### a) Branchements inconditionnels

Les branchement inconditionnels permettent de faire un saut vers une autre instruction différente de l'instruction suivante sans aucun test. Les instructions de branchement inconditionnels sont :

##### 1) Instruction JMP :

☞ Syntaxe : `JMP Label` ;

☞ interprétation :  $IP \leftarrow ADR \text{ de } Label = IP + Dép$ ; avec  $Dép \in \begin{cases} [-2^7, +2^7 - 1] \text{ pour un saut court (short)} \\ [-2^{15}, +2^{15} - 1] \text{ pour un saut proche (near)} \\ [-2^{31}, +2^{31} - 1] \text{ pour un saut lointain (far)} \end{cases}$

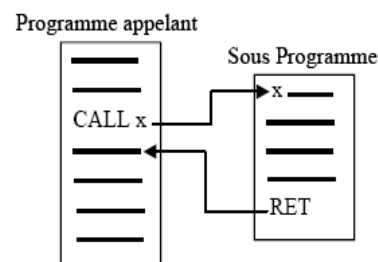
☞ Description : Faire un saut sans condition à l'instruction portant l'étiquette *Label*.

##### 2) Instruction CALL:

☞ Syntaxe : `CALL Label` ;

☞ interprétation :  $\begin{cases} PUSH <ADR \text{ de l'instruction suivante}> \\ IP \bullet <ADR \text{ de } Label> \end{cases}$

☞ Description : Faire un appel à une procédure (sous programme) qui porte le nom *Label*. La position de l'instruction suivante le CALL est empilée pour assurer une poursuite correcte après l'exécution du sous programme.



##### 3) Instruction RET:

☞ Syntaxe : `RET` ;

☞ interprétation :  $\begin{cases} POP IP ; \text{c'est-à-dire :} \\ IP \bullet <ADR \text{ de l'instruction suivante le CALL}> \end{cases}$

☞ Description : Faire un retour de sous programme. L'exécution du programme continue à la position récupérée dans la pile. (un sous programme se termine toujours par l'instruction RET).

##### 4) Instruction INT:

☞ Syntaxe : `INT n` ;

☞ interprétation :  $\begin{cases} PUSH <flags \text{ et } ADR \text{ de l'instruction suivante}> \\ CS:IP \bullet <ADR \text{ de fonction système } N^{\circ} n> \end{cases}$

☞ Description : Faire un appel à l'interruption logicielle (à la fonction système)  $N^{\circ} n$ .

#### b) Branchements conditionnels

Les instructions de branchements conditionnels effectuent un saut (comme JMP) si une certaine condition est vérifiée. Si ce n'est pas le cas, le processeur passe à l'instruction suivante (l'instruction ne fait rien).

☞ Syntaxe : `NomInst Label` ; ☞ interprétation :  $\begin{cases} IP \bullet <ADR \text{ de } Label> = IP + Dép, \text{ avec} \\ Dép \in [-128, +127] \end{cases}$

Les conditions s'expriment en fonction des valeurs des indicateurs. Les instructions de branchement conditionnel s'utilisent en général immédiatement après une instruction de comparaison CMP.

**b.1) Branchements arithmétiques conditionnels :** la liste des instructions de branchement arithmétique conditionnel les plus utiles sont données dans le tableau suivant :

| Condition | Instructions       |                        |                |                          | Explication               |
|-----------|--------------------|------------------------|----------------|--------------------------|---------------------------|
|           | Nombres non signés |                        | Nombres signés |                          |                           |
| =         | JE                 | Jump if equal          | JE             | Jump if Equal            | saut si égal              |
| >         | JA                 | Jump if Above          | JG             | Jump if Greater          | saut si supérieur         |
| <         | JB                 | Jump if Below          | JL             | Jump if Lower            | saut si inférieur         |
| ≥         | JAE                | Jump if Above or equal | JGE            | Jump if Greater or equal | saut si supérieur ou égal |
| ≤         | JBE                | Jump if Below or equal | JLE            | Jump if Lower or equal   | saut si inférieur ou égal |
| ≠         | JNE                | Jump if Not equal      | JNE            | Jump if Not equal        | saut si non égal          |

**b.2) Branchements de test sur les flags:** la liste des instructions de branchement de test sur les flags les plus utiles sont données dans le tableau suivant :

| Indicateur | Instructions        |                |                     |                |
|------------|---------------------|----------------|---------------------|----------------|
|            | Indicateur égal à 1 |                | Indicateur égal à 0 |                |
| ZF         | JZ                  | Jump if ZF = 1 | JNZ                 | Jump if ZF = 0 |
| CF         | JC                  | Jump if CF = 1 | JNC                 | Jump if CF = 0 |
| OF         | JO                  | Jump if OF = 1 | JNO                 | Jump if OF = 0 |
| SF         | JS                  | Jump if SF = 1 | JNS                 | Jump if SF = 0 |
| PF         | JP                  | Jump if PF = 1 | JNP                 | Jump if PF = 0 |

**b.2) Branchements de test sur le registre CX :** la liste des instructions de branchement de test sur le registre CX les plus utiles sont données dans le tableau suivant :

| Instructions    | Interprétation                           |
|-----------------|--|
| JCXZ            | Jump if CX equal Zero (saut si CX = 0)   |
| LOOP            | CX • CX -1 puis saut si CX ≠ 0           |
| LOOPE / LOOPZ   | CX • CX -1 puis saut si CX ≠ 0 et ZF = 1 |
| LOOPNE / LOOPNZ | CX • CX -1 puis saut si CX ≠ 0 et ZF = 0 |

### 3.3.3.6 Les instructions de manipulation des flags (indicateurs)

La liste des instructions agissant sur les indicateurs du registre d'état les plus utiles sont données dans les deux tableaux suivants :

| Indicateur | Instructions |                               |            |                             |
|------------|--------------|-------------------------------|------------|-----------------------------|
|            | Mettre à 0   |                               | Mettre à 1 |                             |
| CF         | CLC          | Clear Carry flag (CF • 0)     | STC        | Set Carry flag (CF • 1)     |
| DF         | CLD          | Clear Direction flag (DF • 0) | STD        | Set Direction flag (DF • 1) |
| IF         | CLI          | Clear Interrupt flag (IF • 0) | STI        | Set Interrupt flag (IF • 1) |

| Instructions |                       | Interprétation           | Description                                       |
|--------------|-----------------------|--------------------------|---|
| CMC          | Complement Carry flag | $CF \cdot \overline{CF}$ | Inverser CF.                                      |
| LAHF         | Load AH from Flags    | $AH \cdot PSW[0-7]$      | Charge l'octet bas du registre d'état dans AH.    |
| SAHF         | Store AH into Flags   | $PSW[0-7] \cdot AH$      | Transfert AH dans l'octet bas du registre d'état. |
| PUSHF        | PUSH Flags            | PUSH PSW                 | Empile le registre d'état,                        |
| POPF         | POP Flags             | POP PSW                  | Dépile le registre d'état,                        |

☞ **Remarque** : toutes ces instructions sont sans opérandes.

### 3.3.3.7 Les instructions de manipulation de chaînes

Les instructions de manipulation de chaînes sont au nombre de 5 classifiées en deux versions, l'une se fait entre deux octets (MOVSB, LODSB, STOSB, CMPSB et SCASB), l'autre se fait entre 2 words de 16 bits (MOVSW, LODSW, STOSW, CMPSW et SCASW) :

#### 1) Instruction MOVSB :

☞ **Syntaxe** : MOVSB ;      ☞ **interprétation** :  $\begin{cases} \text{Byte [DI]} \cdot \text{Byte [SI].} \\ \text{si DF} = 0 \text{ alors SI} \cdot \text{SI} + 1 \text{ et DI} \cdot \text{DI} + 1 \\ \text{sinon (DF} = 1) \text{ SI} \cdot \text{SI} - 1 \text{ et DI} \cdot \text{DI} - 1 \end{cases}$

☞ **Description** : Copier un octet depuis la case mémoire source [DS:SI] vers la case destination [ES:DI] puis auto inc/decrémenter les registre SI et DI.

#### 2) Instruction MOVSW :

☞ **Syntaxe** : MOVSW ;      ☞ **interprétation** :  $\begin{cases} \text{Word [DI]} \cdot \text{Word [SI].} \\ \text{si DF} = 0 \text{ alors SI} \cdot \text{SI} + 2 \text{ et DI} \cdot \text{DI} + 2 \\ \text{sinon (DF} = 1) \text{ SI} \cdot \text{SI} - 2 \text{ et DI} \cdot \text{DI} - 2 \end{cases}$

☞ **Description** : Copier deux octets depuis la case mémoire source [DS:SI] vers la case destination [ES:DI] puis auto inc/decrémenter de 2 les registre SI et DI.

**Remarque** : Grâce au préfixe de répétition REP, les instructions MOVSB et MOVSW sont répétées CX fois ce qui permet de copier une zone mémoire dans une autre.

#### 3) Instruction LODSB :

☞ **Syntaxe** : LODSB ;      ☞ **interprétation** :  $\begin{cases} AL \cdot [SI]. \\ \text{si DF} = 0 \text{ alors SI} \cdot \text{SI} + 1 \text{ et DI} \cdot \text{DI} + 1 \\ \text{sinon (DF} = 1) \text{ SI} \cdot \text{SI} - 1 \text{ et DI} \cdot \text{DI} - 1 \end{cases}$

☞ **Description** : Copier un octet depuis la case mémoire source [DS:SI] vers le registre AL puis auto inc/decrémenter les registre SI et DI.

#### 4) Instruction LODSW :

☞ **Syntaxe** : LODSW ;      ☞ **interprétation** :  $\begin{cases} AX \cdot [SI]. \\ \text{si DF} = 0 \text{ alors SI} \cdot \text{SI} + 2 \text{ et DI} \cdot \text{DI} + 2 \\ \text{sinon (DF} = 1) \text{ SI} \cdot \text{SI} - 2 \text{ et DI} \cdot \text{DI} - 2 \end{cases}$

☞ **Description** : Copier deux octets depuis la case mémoire source [DS:SI] vers le registre AX puis auto inc/decrémenter de 2 les registre SI et DI.

#### 5) Instruction STOSB :

☞ **Syntaxe** : STOSB ;      ☞ **interprétation** :  $\begin{cases} [DI] \cdot AL. \\ \text{si DF} = 0 \text{ alors SI} \cdot \text{SI} + 1 \text{ et DI} \cdot \text{DI} + 1 \\ \text{sinon (DF} = 1) \text{ SI} \cdot \text{SI} - 1 \text{ et DI} \cdot \text{DI} - 1 \end{cases}$

☞ **Description** : Copier AL dans la case destination [ES:DI] puis auto inc/decrémenter les registre SI et DI.

**6) Instruction STOSW :**

☞ **Syntaxe** : STOSW ;      ☞ **interprétation** : 
$$\begin{cases} [DI] \leftarrow AX. \\ \text{si } DF = 0 \text{ alors } SI \leftarrow SI + 2 \text{ et } DI \leftarrow DI + 2 \\ \text{sinon } (DF = 1) \text{ } SI \leftarrow SI - 2 \text{ et } DI \leftarrow DI - 2 \end{cases}$$

☞ **Description** : Copier AX dans la case destination [ES:DI] puis auto inc/decrémenter de 2 les registre SI et DI.

**Remarque** : Avec le préfixe de répétitions REP, Cette instruction permet d'initialiser une chaîne (une zone mémoire) avec le même caractère.

**7) Instruction CMPSB :**

☞ **Syntaxe** : CMPSB ;      ☞ **interprétation** : 
$$\begin{cases} \text{Byte } [DI] - \text{Byte } [SI]. \\ \text{si } DF = 0 \text{ alors } SI \leftarrow SI + 1 \text{ et } DI \leftarrow DI + 1 \\ \text{sinon } (DF = 1) \text{ } SI \leftarrow SI - 1 \text{ et } DI \leftarrow DI - 1 \end{cases}$$

☞ **Description** : Comparer l'octet de la case mémoire source [DS:SI] avec l'octet de la case destination [ES:DI], positionner les indicateurs puis auto inc/decrémenter les registre SI et DI.

**8) Instruction CMPSW :**

☞ **Syntaxe** : CMPSW ;      ☞ **interprétation** : 
$$\begin{cases} \text{Word } [DI] - \text{Word } [SI]. \\ \text{si } DF = 0 \text{ alors } SI \leftarrow SI + 2 \text{ et } DI \leftarrow DI + 2 \\ \text{sinon } (DF = 1) \text{ } SI \leftarrow SI - 2 \text{ et } DI \leftarrow DI - 2 \end{cases}$$

☞ **Description** : Comparer les deux octets de la case mémoire source [DS:SI] avec les deux octets de la case destination [ES:DI], positionner les indicateurs puis auto inc/decrémenter de 2 les registre SI et DI.

**Remarque1** : Avec le préfixe de répétition **REPZ** (repeat while Z), cette instruction est répétée CX fois tant que Z=1, et avec le préfixe de répétition **REPNZ** (repeat while not Z), cette instruction est répétée CX fois tant que Z=0.

**Remarque2** : La répétition de l'instruction CMPSB permet par exemple la comparaison de deux chaînes de caractères.

**9) Instruction SCASB :**

☞ **Syntaxe** : SCASB ;      ☞ **interprétation** : 
$$\begin{cases} [DI] - AL. \\ \text{si } DF = 0 \text{ alors } SI \leftarrow SI + 1 \text{ et } DI \leftarrow DI + 1 \\ \text{sinon } (DF = 1) \text{ } SI \leftarrow SI - 1 \text{ et } DI \leftarrow DI - 1 \end{cases}$$

☞ **Description** : Comparer AL avec l'octet de la case destination [ES:DI], positionner les indicateurs puis auto inc/decrémenter les registre SI et DI.

**10) Instruction SCASW :**

☞ **Syntaxe** : SCASW ;      ☞ **interprétation** : 
$$\begin{cases} [DI] - AX. \\ \text{si } DF = 0 \text{ alors } SI \leftarrow SI + 2 \text{ et } DI \leftarrow DI + 2 \\ \text{sinon } (DF = 1) \text{ } SI \leftarrow SI - 2 \text{ et } DI \leftarrow DI - 2 \end{cases}$$

☞ **Description** : Comparer AX avec les deux octets de la case destination [ES:DI], positionner les indicateurs puis auto inc/decrémenter de 2 les registre SI et DI.

**Remarque** : La répétition de l'instruction SCASB permet par exemple de chercher l'occurrence d'une valeur dans une chaîne.

