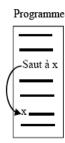
#### 3.3.3.5 Les instructions de branchement

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

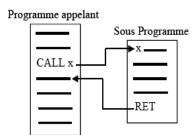
Les branchement inconditionnels permettent de faire un saut vers une autre instruction différente de l'instruction suivante sans aucun test. Les instructions de branchement inconditionnels sont :

### 1) Instruction JMP:

- ☞ Syntaxe : JMP Label ;
- Syntaxe: JMP Label; interprétation:  $IP \leftarrow ADR$  de Label = IP + Dép; avec  $Dép \in \begin{cases} [-2^7, +2^7 1] \ pour \ un \ saut \ court \ (short) \\ [-2^{15}, +2^{15} 1] \ pour \ un \ saut \ proche \ (near) \\ [-2^{31}, +2^{31} 1] \ pour \ un \ saut \ lointain \ (far) \end{cases}$
- Description: Faire un saut sans condition à l'instruction portant l'étiquette Label.

### 2) Instruction CALL:

- Syntaxe : CALL Label ;
- $\Rightarrow$  interprétation :  $\begin{cases} PUSH < ADR \ de \ l'instruction \ suivante > \\ IP \bullet \quad < ADR \ de \ Label > \end{cases}$
- Tescription : 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:

- $\P$  interprétation :  $\begin{cases} POP \ IP \ ; \ c'est-\`{a}-dire \ : \\ IP \bullet & <ADR \ de \ l'instruction \ suivante \ le \ CALL > \end{cases}$ Syntaxe: RET;
- 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:

- interprétation :  $\begin{cases} PUSH < flags \ et \ ADR \ de \ l'instruction \ suivante > \\ CS:IP \bullet < ADR \ de \ fonction \ système \ N^{\circ} \ n > \end{cases}$ Syntaxe: INT n;
- Description: Faire un appel à l'interruption logicielle (à la fonction système) N° 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).

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
Condition	Nombres non signés			Nombres signés	Explication
=	JE	Jump if equal	JE	Jump if Equal	saut si égal
>	JA	Jump if Above	JG	Jump if Greater	saut si supérieur
<	JВ	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
<b>S</b>	JBE	Jump if Below or equal	JLE	Jump if Lower or equal	saut si inférieur ou égal
<b>≠</b>	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	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	JР	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	$CX \cdot CX - 1$ puis saut si $CX \neq 0$			
LOOPE / LOOPZ	$CX \cdot CX - 1$ puis saut si $CX \neq 0$ et $ZF = 1$			
LOOPNE / LOOPNZ	CX • CX -1 puis saut si CX $\neq$ 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				
Indicateur	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 • CF	Inverser CF.
LAHF	Load AH from Flags	AH • PSW[0-7]	Charge l'octet bas du registre d'état dans AH.
SAHF	Store AH into Flags	PSW[0-7] • 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,

<sup>•</sup> 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**:  $\begin{cases} \text{Byte [DI]} \bullet & \text{Byte [SI]}. \\ \text{si DF} = 0 \text{ alors SI} \bullet & \text{SI} + 1 \text{ et DI} \bullet & \text{DI} + 1 \\ \text{sinon (DF} = 1) \text{ SI} \bullet & \text{SI} 1 \text{ et DI} \bullet & \text{DI} 1 \end{cases}$
- **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 ;
- $\text{ word [DI] } \cdot \text{ Word [SI]}.$   $\text{si DF} = 0 \text{ alors SI} \cdot \text{ SI} + 2 \text{ et DI} \cdot \text{ DI} + 2$   $\text{sinon (DF} = 1) \text{ SI} \cdot \text{ SI} 2 \text{ et DI} \cdot \text{ DI} 2$
- 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 :  $\begin{cases} AL \cdot [SI]. \\ si DF = 0 \text{ alors } SI \cdot SI + 1 \text{ et } DI \cdot DI + 1 \\ sinon (DF = 1) SI \cdot SI 1 \text{ et } DI \cdot DI 1 \end{cases}$
- 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 :  $\begin{cases} AX \cdot [SI]. \\ si DF = 0 \text{ alors } SI \cdot SI + 2 \text{ et } DI \cdot DI + 2 \\ sinon (DF = 1) SI \cdot SI 2 \text{ et } DI \cdot DI 2 \end{cases}$
  - **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 :  $\begin{cases} [DI] \cdot AL. \\ si DF = 0 \text{ alors } SI \cdot SI + 1 \text{ et } DI \cdot DI + 1 \\ sinon (DF = 1) SI \cdot SI 1 \text{ et } DI \cdot DI 1 \end{cases}$

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

6) Instruction STOSW:

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

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

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

7) Instruction CMPSB:

- **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;

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: 
$$\begin{cases} IDIJ - AL. \\ si DF = 0 \text{ alors } SI \cdot SI + 1 \text{ et } DI \cdot DI + 1 \\ sinon (DF = 1) SI \cdot SI - 1 \text{ et } DI \cdot DI - 1 \end{cases}$$

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

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.

# 3.3.3.8 Les instructions d'adressage

## 1) Instruction LEA:

Syntaxe: LEA OpD, OpS;
interprétation: OpD • Adresse(OpS)

Description: Charge l'offset de l'opérande source OpS dans le registre OpD.

F Exemples: LEA AX, [BX+124] LEA BX, Variable

General Combination of Combinatio

LEA Registre 16, Mémoire Charge l'offset d'une case mémoire dans un registre 16 bits.

☞ Combinaisons interdites :

LEA Registre8, Mémoire Charge l'offset d'une case mémoire dans un registre 8 bits.

❖ Les indicateurs affectés : l'instruction LEA ne modifie aucun indicateur du PSW.

# 2) Instruction LDS:

Description: Le mot pointé par adr est recopié dans l'opérande OpD et le mot suivant est recopié dans DS.

© Exemples: LDS BX, [9A00]

3) Instruction LES:

Description: Le mot pointé par adr est recopié dans l'opérande OpD et le mot suivant est recopié dans ES.

### 3.3.3.9 Les instructions d'entrée-sortie

## 1) Instruction IN:

Syntaxe : IN AL, adresse port E/S;

**Description :** Lire un octet depuis un port d'entrée-sortie et le transférer dans le registre AL. adresse port E/S peut être DX ou bien une constante de 8 bits.

Feemple: IN AL, 4Dh MOV DX, 378h IN AL, DX

### 2) Instruction OUT:

Syntaxe : OUT adresse port E/S, AL;

**Description :** Ecrire dans le Port d'entrée-sortie la valeur contenue dans le registre AL. adresse port E/S est une constante ou bien DX.

\* Exemple : OUT 4Dh, AL MOV DX, 378h OUT DX, AL

**Remarque :** L'adresse port E/S ne s'agit pas d'une adresse en mémoire principale, mais c'est une adresse d'une interface d'entrées/sorties.

### 3.3.3.10 Les instructions de retour au DOS

A la fin d'un programme en assembleur, on souhaite en général que l'interpréteur de commandes du DOS reprenne le contrôle du PC. Pour cela, on utilisera la séquence de deux instructions :

MOV AH, 4C INT 21