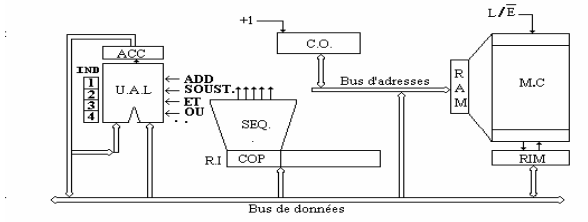
Chapitre 9 : La machine MIASM

- Introduction .
- Architecture générale de MIASM.
- Format d'une instruction et modes d'adressage de MIASM
- Jeu d'instructions de MIASM .
- Programmation en langage MIASM .

Introduction

- Le but de ce chapitre est de montrer le fonctionnement complet d'un ordinateur.
- Nous allons travailler sur une machine fictive (pas réelle) dite "MACHINE PEDAGOGIQUE", que nous appellerons "MIASM".
- Cette machine est très simplifiée, et n'est, donc pas une machine réelle.
- Cependant elle possède tous les composants et les caractéristiques d'un véritable ordinateur.

1. Structure générale de MIASM



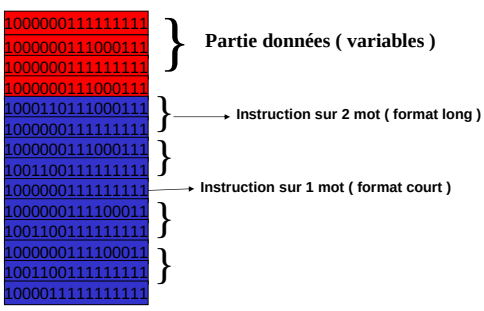
La taille de la mémoire est de 2048 mots → bus d'adresse sur 11 bits
La taille d'un mot est de 16 bits → bus de données sur 16 bits
La taille de l'accumulateur , le RIM et le RI est de 16 bits
La taille du CO et de RAM est de 11 bits
La machine possède 4 indicateurs (flags)

Les indicateurs

- **L'indicateur N°1** : est mis à 1 si un **débordement** de capacité se produit dans une opération , il est mis à 0 dans le cas normal.
- **L'indicateur N°2** : est mis à 1 si opération dégage **une retenue**, à zéro sinon.
- **L'indicateur N°3** : est mis à 1 si le contenu de l'accumulateur est **égal à zéro**. Il est mis à zéro si le contenu de l'accu est non nul.
- **L'indicateur N°4** : est mis à 1 si le contenu de l'accu **est Négatif**, il est à zéro sinon.

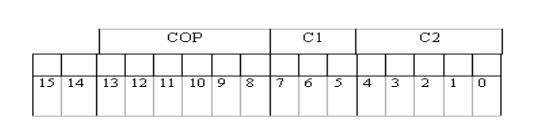
2. Format d'une instruction

- MIASM est une machine à une adresse.
- Les instructions doivent être représentées en binaire sur un ou plusieurs mots.
- La machine MIASM dispose de deux types de format d'instructions
  - Les instructions format LONG:Ce type d'instruction occupe deux mots mémoire :
    - Le premier mot comporte le code opération, le type d'adressage,…….
    - le deuxième mot comporte l'adresse de l'opérande.
  - Les instructions format COURT: Elles occupent un seul mot (comporte le code opération, le type d'adressage,…….). Ce format est utilisé par les instructions qui ne comportant pas une partie adresse(Exemple : entrées/sorties).

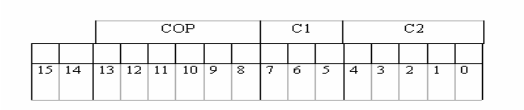


Format d'une instruction
2.1 Le premier mot

- Ce mot est commun aux deux types d'instructions.
- Il tient sur 16 bits et il est divisé en plusieurs champs.



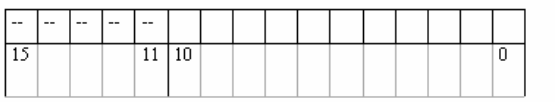
Les bits 15 et 14: Servent à indiquer le type d'adressage:
- **00** : adressage direct .
- **01** : adressage indirect.
- **10** : adressage immédiat.
- **11** : Configuration interdite.



- Les bits 13 à 8: donnent (sur 6 bits) Le CODE OPERATION à effectuer.
- Le bit 13, premier bit du code opération, indique le format de l'instruction :
  - bit 13=0 ==> instruction format court.
  - bit 13=1 ==> instruction format long.
- Les bits 7 à 5: Définissent une zone ou champ appelé C1 dont l'utilisation dépend de l'opération.
- Les bits 4 à 0: Définissent une zone ou champ appelé C2 dont l'utilisation dépend de l'opération.

Format d'une instruction
2.2. Le deuxième mot

- Propre aux instructions en format long, il contient la partie adresse de l'instruction. Cette adresse tient sur 11 bits (bit 0 au bit 10)



- Exemple1 : déroulement de l'instruction d'addition en mode immédiat ACC←(ACC)+ Valeur sur MIASM
  - Phase 1 : (rechercher l'instruction à traiter)
    - Mettre le contenu du CO dans le registre RAM RAM← (CO)
    - Commande de lecture à partir de la mémoire
    - Transfert du contenu du RIM dans le registre RI RI← (RIM)
    - Analyse et décodage
  - Phase 2 : (traitement)
    - CO ← (CO)+ 1
    - Transfert de l 'adresse du 2eme mot dans le registre RAM RAM← (CO)
    - Commande de lecture à partir de la mémoire
    - Transfert de l'opérande vers l'UAL UAL ← (RIM)
    - Commande de l'exécution de l'opération (addition)
  - Phase 3 : (passer à l'instruction suivante)
    - CO ← (CO)+ 1

- **Exemple 2 :** déroulement de l'instruction d'addition en mode direct ACC←(ACC)+ (ADR) sur MIASM
  - **Phase 1 :** (rechercher l'instruction à traiter)
    - Mettre le contenu du CO dans le registre RAM RAM← (CO)
    - Commande de lecture à partir de la mémoire
    - Transfert du contenu du RIM dans le registre RI RI← (RIM)
    - Analyse et décodage
  - **Phase 2 :** (traitement)
    - CO ← (CO)+ 1
    - Transfert de l 'adresse du 2eme mot dans le registre RAM RAM ← (CO)
    - Commande de lecture à partir de la mémoire
    - Transfert de l'adresse de l'opérande vers le RAM RAM ← (RIM)
    - Commande de lecture à partir de la mémoire
    - Transfert du contenu du RIM (l'opérande)vers l'UAL UAL ←(RIM)
    - Commande de l'exécution de l'opération (addition)
  - **Phase 3 :** (passer à l'instruction suivante)
    - CO ← (CO)+ 1

- Exemple1 : déroulement de l'instruction au format court sur MIASM (exemple entrées /sorties).
  - Phase 1 : (rechercher l'instruction à traiter)
    - Mettre le contenu du CO dans le registre RAM RAM← (CO)
    - Commande de lecture à partir de la mémoire
    - Transfert du contenu du RIM dans le registre RI RI← (RIM)
    - Analyse et décodage
  - Phase 2 : (traitement)
    - Commande de l'exécution de l'opération
  - Phase 3 : (passer à l'instruction suivante)
    - CO ← (CO)+ 1

3. LE JEU D'INSTRUCTIONS DE MIASM

- Pour pouvoir faire des programmes exécutables sur la machine MIASM, on dispose d'un certain nombre d'instructions qui forment le langage de la machine.
- Ce langage est un langage ASSEMBLEUR.

```
      ORG   X'100'
X     RM    1
Y     RC    10
DEBUT ENT   01
      ADM   Y
      RGM   X
      SOR   02
      STOP
END   DEBUT
```

→

```
1110000011101011
0010110011101011
0010000011101011
1110000011101011
1100000011101011
1000000011101011
1110000011101011
```

Langage assembleur — Langage machine

3.1 Les instructions d'échange entre l'accumulateur et la mémoire centrale

- Instruction : **RANGEMENT (RGM)**
- Effet : Le contenu de l'accumulateur est écrit en mémoire centrale à l'adresse figurant dans l'instruction. Le contenu de l'accumulateur n'est pas modifié.
- Format : Long
- Adressage : Direct ou Indirect

- Exemple :
  - RGM A (mode direct)
  - RGM *B (mode indirect)

- Instruction : **CHARGEMENT IMMEDIAT (CHI)**
- Effet : La partie adresse de l'instruction est chargée dans l'accumulateur. Le contenu précédant de l'accumulateur est détruit .
- Format : Long
- Adressage : Immédiat
- Les indicateurs 3 et 4 de l'UAL sont positionnés selon l'information chargée.

- Exemple :
  - CHI 12
  - CHI 0

- Instruction : **CHARGEMENT MOT (CHM)**
- Effet : Le contenu du mot mémoire référencé par la partie adresse de l'instruction est chargé dans l'accumulateur. Le contenu précédant de l'accumulateur est détruit.
- Format : Long
- Adressage : Direct ou Indirect
- Les indicateurs 3 et 4 de l'UAL sont positionnés selon l'information chargée.
- Exemple :
  - CHM A (mode direct)
  - CHM *B (mode indirect)

3.2 Instructions d'opérations arithmétiques

- Instruction : **ADDITION/SOUSTRACTION IMMEDIATE (ADI / SI)**
- Effet : La partie adresse de l'instruction est additionnée/soustraite au/du contenu de l'accumulateur. Le résultat est dans l'accumulateur
- Format : Long
- Adressage : Immédiat
- Observations : Les indicateurs 1,2,3 et 4 de l'UAL sont positionnés selon l'information chargée.

- Exemple :

```
ADI  12
SI   13
```


- Instruction : **ADDITION/SOUSTRACTION MOT (ADM / SM)**
- Effet : Le contenu du mot mémoire référencé par la partie adresse de l'instruction est additionné/soustrait au/du contenu de l'accumulateur. Le résultat est dans l'accumulateur.
- Format : Long
- Adressage : Direct ou Indirect
- Observations : Les indicateurs 1,2,3 et 4 de l'UAL sont positionnés selon l'information chargée.

- **Exemple :**

```
ADM  A
SM   *B
```

3.3 Instructions d'opérations logiques

- Instruction : **ET MOT**
- Effet : Un ET logique est effectué entre le contenu de l'accumulateur et le contenu du mot adresse par la partie adresse de l'instruction. Le résultat est dans l'accumulateur.
- Format : Long
- Adressage : Direct ou Indirect
- Observations : Les indicateurs 3 et 4 de l'UAL sont positionnés selon le résultat trouvé.

- Exemple :

  ET A

  ET *B

- Instruction : **OU/OUX MOT**
- Effet : Un OU/OUX logique est effectué entre le contenu de l'accumulateur et le contenu du mot adressé par la partie adresse de l'instruction. Le résultat est dans l'accumulateur.
- Format : Long
- Adressage : Direct ou Indirect
- Observations : Les indicateurs 3 et 4 de l'UAL sont positionnés selon le résultat trouvé.

- Exemple :

  OU A

  OU *B

- Instruction : **NON MOT**
- Effet : Tous les bits du contenu de l'accumulateur sont inverses.
- Format : Long
- Adressage : Direct ou Indirect
- Observations : Les indicateurs 3 et 4 de l'UAL sont positionnés selon le résultat trouvé.

3.4 Instructions d'entrées/sorties

- Instruction : **ENTREE DE DONNEES ENT**
- Effet : Une donnée est entrée à partir d'un périphérique dans l'accumulateur.
- Format : Court
- Adressage : Immédiat
- Observations : - Le champ C1 n'est pas utilisé
- Le champ C2 donne le numéro du périphérique (**le périphérique 01 indique le clavier)**

Exemple :

ENT 01
RGM A

- Instruction : **SORTIE DE DONNEES SOR**
- Effet : Une donnée est sortie de l'accumulateur vers un périphérique.
- Format : Court
- Adressage : Immédiat.
- Observations : - Le champ C1 n'est pas utilisé
- Le champ C2 donne le numéro du périphérique (**le périphérique 02 indique l'ecran)**

- Exemple

  CHM A

  SOR 02

3.5 Instructions d'arrêt du calculateur

- Instruction : **ARRET DU CALCULATEUR (STOP)**
- Effet : Provoque un arrêt du programme en cours d'exécution.
- Format : Court
- Adressage : Immédiat

Instruction	Mnémonique	Cod. Op.	Format	Adr
Rangement mot	RGM	20	L	D/I
Chargement mot	CHM	22	L	D/I
Chargement immédiat	CHI	21	L	Imm
Addition immédiate	ADI	23	L	Imm
Addition mot	ADM	24	L	D/I
Soustraction immédiate	SI	25	L	Imm
Soustraction mot	SM	26	L	D/I
ET mot	ET	27	L	D/I
OU mot	OU	28	L	D/I
OU Exclusif mot	OUX	29	L	D/I
NON mot	NON	2A	L	D/I
Branchement si condition vraie	BCV	2B	L	D/I
Branchement si cond fausse	BCF	2C	L	D/I
Entrée	ENT	01	C	--
Sortie	SOR	02	C	--
Stop	STOP	00	C	--

L : Long, C : Cours;

D : Direct; I : Indirect; Imm. : Immédiat.

4. Structure générale d'un programme en langage MIASM

- Un programme écrit afin d'être exécuter sur MIASM est composé de deux partie :
  - partie données
  - et partie instruction

- Exemple :

```
        ORG   X'100'
X       RM    1
Y       RC    10
DEBUT   ENT   01
        ADM   Y
        RGM   X
        SOR   02
        STOP
END     DEBUT
```

•**Adresse début du programme en mémoire**

•**Réservation d'un mot mémoire**

•**Réserver un mot mémoire et l'initialiser avec la valeur 10**

Les instructions

4.1 Partie données

- Pour les données on utilise les deux directives RM et RC :
  - RM permet de réserver une zone mémoire de N mots mémoire.
  - RC permet de réserver une zone mémoire avec initialisation.
  -
- Exemple

```
   ORG  100
X  RM    1                  réserver un seul  mot
Y  RC    23                 réserver un mot et l'initialiser par la valeur 23
Z  RM    4                  réserver 4 mots mémoire
T  RC    X'AB'  X'10'  X'23'  réserver 3 mots mémoire initialisés avec les valeurs
                              hexadécimales 'AB' , '10' et '23'
```


4.2 Partie instructions

- La partie instruction contient l'ensemble des instructions (dans l'ordre) qui détermine la logique du programme.
- Dans cette partie on peut trouver les instructions de :
  - des instruction arithmétique ,
  - logique ,
  - entrées /sorties ,
  - ……
  - ……

Exemple 1

- Exemple :
- Soit l'algorithme suivant :

Lire (B)
Lire (C)
A← (B+C)-123
Écrire (A)

```
        ORG   100
A       RM    1
B       RM    1
C       RM    1
DEBUT   ENT   01
        RGM   B
        ENT   01
        RGM   C
        CHM   B
        ADM   C
        SI    123
        RGM   A
        SOR   02
        STOP
        END   DEBUT
```


Exercice : Quel est le contenu du Mot " RESU " à la fin de l'exécution du programme suivant:

```
        ORG   0
DON     RC    X ' ABCD '
RESU    RM    1
DEBUT   CHI   X ' F00F '
        ET    DON
        ADI   X ' 2FFD '
        RGM   RESU
        CHI   X ' 0FF0 '
        ET    DON
        ADI   X ' 00F0 '
        OU    RESU
        RGM   RESU
        END   DEBUT
```

5. Instructions de branchement

- Dans un programme les instructions sont exécutées dans l'ordre , généralement séquentielles .
- Dans quelques cas les instructions à exécutées après la vérification d'une condition .
- Dans ce cas la prochaine instruction à exécuter dépend de la valeur de la condition → pas forcément la prochaine instruction.
- Si la condition n'est pas vérifiée alors il faut faire un branchement (saut vers une autre instruction).
- Le deuxième mot de l'instruction contient l'adresse de branchement (adresse de l'instruction à exécuter si la condition n'est pas vérifiée)
- Pour tester la condition on utilise les indicateurs.

- **Exemple :** déroulement de l'instruction de branchement si la condition est vérifiée (exemple tester l'indicateur 4 s'il est égale à 1)

  – **Phase 1 :** (rechercher l'instruction à traiter)
    - Mettre le contenu du CO dans le registre RAM RAM← (CO)
    - Commande de lecture à partir de la mémoire
    - Transfert du contenu du RIM dans le registre RI RI← (RIM)
    - Analyse et décodage

  – **Si condition vérifiée (valeur de l'indicateur 4 est égale à 1)**
    - **Phase 2 :** (traitement)
      - CO ← (CO)+ 1
      - Transfert de l 'adresse du 2ème mot dans **le registre RAM** RAM ← (CO)
      - Commande de lecture à partir de la mémoire
      - Transfert de l'adresse de l'instruction vers le **CO** CO ← (RIM)

  – **Si condition non vérifiée**
    – **Phase 3 :** (passer à l'instruction suivante)
      - CO ← (CO)+ 1

Instructions de branchement

- Instruction : **BRANCHEMENT SI CONDITION VERIFIEE (BCV,ind)**
- Effet : Les trois bits du champ C1 donnent un numéro de condition de 0 à 4 à tester :
  – Si le n°de la condition est 0 : exécution d'un bran chement à l'adresse effective AE.
  – Si le n°de la condition est 1.2.3 ou 4 : test de l' indicateur correspondant et exécution d'un branchement à l'adresse effective AE si l'indicateur est à 1. Si l'indicateur est à 0, poursuite en séquence (non branchement).

- Exemple

  BCV,4 branchement si l'indicateur 4 est à 1 (le résultat est négative)

  BCV,3 branchement si indicateur 3 est à 1 (le résultat est nul)

- Instruction : BRANCHEMENT SI CONDITION FAUSSE (BCF,ind)
- Effet : Les trois bits du champ C1 donnent un numéro de condition de 0 à 4 à tester.
  – Si le n°de la condition est 0 : exécution d'un bran chement à l'adresse effective AE.
  – Si le n°de la condition est 1.2.3 ou 4 : test de l' indicateur correspondant et exécution d'un branchement à l'adresse effective AE si l'indicateur est à 0. Si l'indicateur est à 1 poursuite en séquence (non branchement).

- Exemple

  BCF,4 branchement si l'indicateur 4 est à 0 (le résultat n'est pad négative)

  BCF,3 branchement si indicateur 3 est à 0 (le résultat n'est pas nul)

Exemple

Exemple
Si A > B alors Z← A+B;

A← A+1;

Exemple
Si (A - B) > 0 alors Z← A+B;

A← A+1;

En langage MIASM :

```
        CHM   A
        SM    B
        BCV,4 suite
        CHM   A
        ADM   B
        RGM   Z
Suite   CHM   A
        ADI   1
        RGM   A
```


La forme SI …… SINON

- Si **cond** alors Action 1
  sinon Action 2

- Évaluation de la **condition**
  – Si condition est fausse branchement A2
  – Exécuter Action 1
  – Branchement inconditionnelle à suite
- A2 : exécuter Action 2

Exemple

Si A > B alors Max ← A
Sinon Max ← B

→

Si (A – B) > 0 alors Max ← A
Sinon Max ← B

En MAISM

```
        CHM     A
        SM      B
        BCV,4   A2
        CHM     A
        RGM     MAX
        BCV,0   suite
A2      CHM     B
        RGM     MAX
Suite   CHM     MAX
        SOR     02
```


Condition composée

Si (A > B) et (A < C) alors
RES←C

```
        CHM     A
        SM      B
        BCV,4   suite
        CHM     A
        SM      C
        BCF,4   suite
        CHM     C
        RGM     RES
suite
```


Condition composée

Si A > B ou A < C alors
Res← B+C

```
        CHM A
        SM B
        BCF,4   action
        CHM A
        SM C
        BCF,4   suite
Action  CHM     B
        ADM     C
        RGM     RES
Suite ……
```


Forme Tant que

Exemple
res ← 1+2+3+4+5+6+7+8+9

K←1
RES← 0
Tant que K< 10 faire
Début
RES← RES + K;
K←K+1;
End

```
Org     X'100'
K       RM  1
RES     RM  1
Debut   CHI 0
        RGM RES
        CHI 1
        RGM K
BOUCLE  SI  10
        BCF,4   FIN
        CHM     RES
        ADM     K
        RGM     RES
        CHM     K
        ADI     1
        RGM     K
        BCV,0   BOUCLE
FIN     CHM     RES
        SOR     02
        STOP
END     Debut
```


Forme répéter

Exemple
res ← 1+2+3+4+5+6+7+8+9

K←1
RES← 0
Répéter
RES← RES + K;
K←K+1;
Jusqu'à K=10

```
Org     X'100'
K       RM  1
RES     RM  1
Debut   CHI 0
        RGM RES
        CHI 1
        RGM K
BOUCLE  CHM RES
        ADM K
        RGM RES
        CHM K
        ADI 1
        RGM K
        SI  10
        BCV,3   FIN
        BCV,0   BOUCLE
FIN     CHM     RES
        SOR     02
        STOP
END     Debut
```