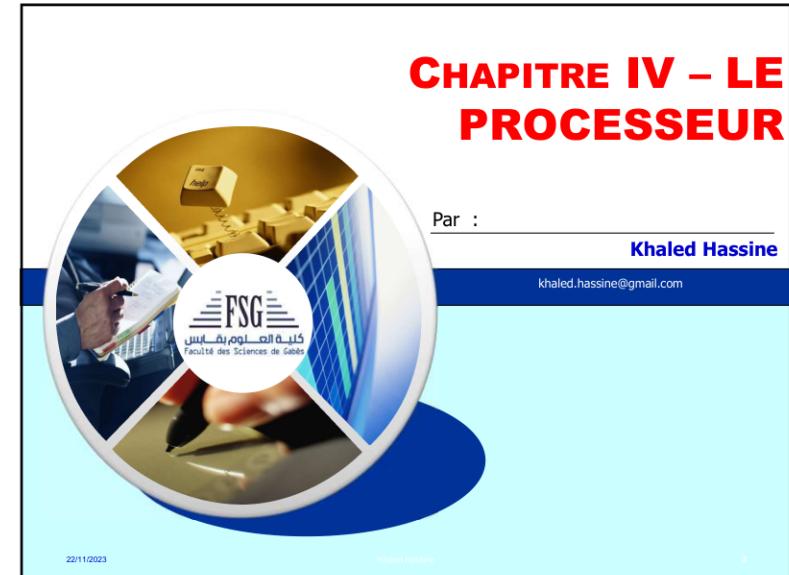
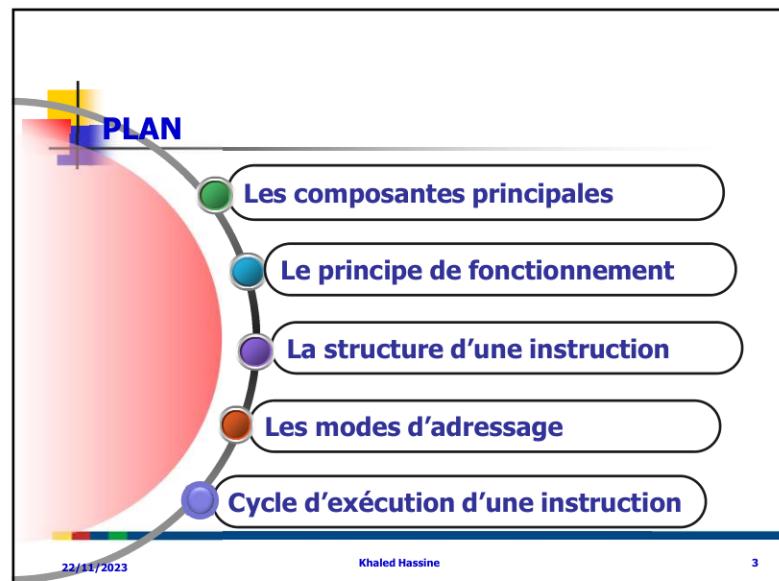




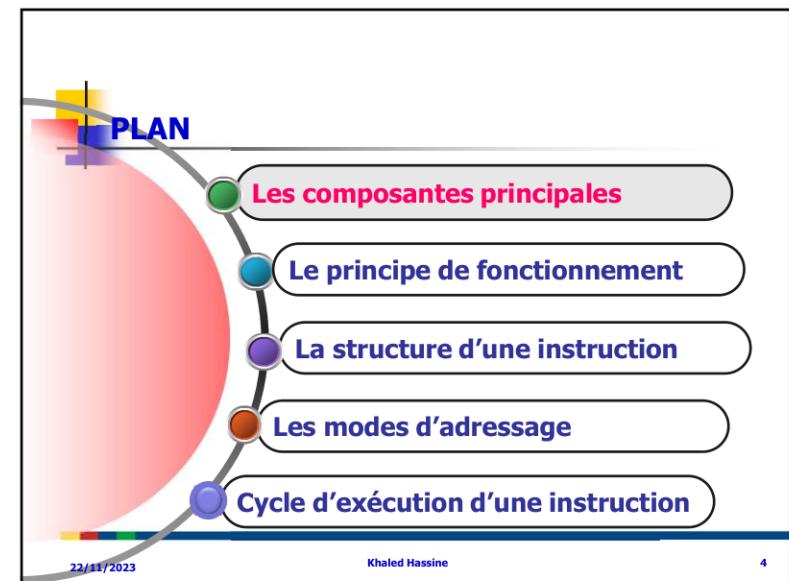
1



2



3



1

Les principales composantes du processeur

- **L'unité de commande** pilote toutes les composantes de l'ordinateur et veiller sur le bon déroulement du programme à exécuter.
- **L'unité de traitement (Unité arithmétique et logique, noté UAL)** : assure la réalisation des opérations arithmétiques et logiques.
- **Registres** : dispositifs de mémorisation internes au processeur.
 - C'est l'unité de base manipulée par l'unité centrale.
 - Aucune opération ne se fait directement sur les cellules mémoire (recopiées dans des registres puis traitées par le processeur).

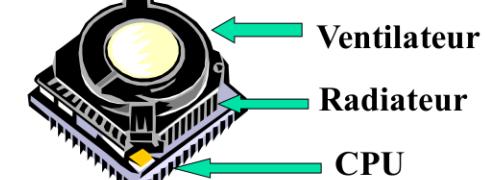
22/11/2023

Khaled Hassine

5

Le processeur : vue externe

- **Cœur de l'ordinateur, dit aussi (CPU : Central Processing Unit)**



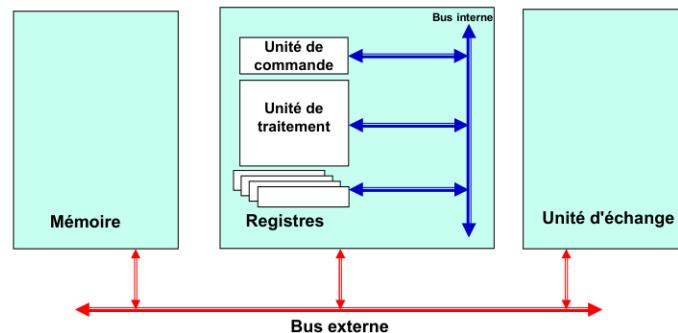
- Traite toutes les informations et gère les entrées sorties
- Il envoie des instructions aux autres composants de l'ordinateur (mémoire, unités de stockage, écran...)

22/11/2023

Khaled Hassine

6

Le processeur : vue interne

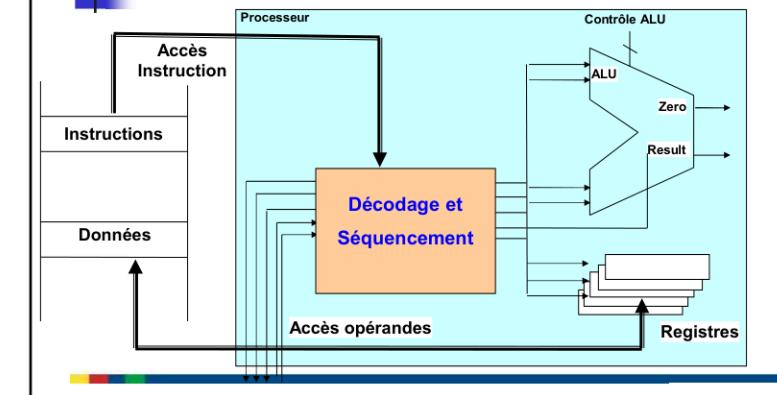


22/11/2023

Khaled Hassine

7

Exécution d'une instruction d'un programme ...



22/11/2023

Khaled Hassine

8

Les composantes principales

Les registres

- Les registres de travail et l'accumulateur
- La pile et le pointeur de pile
- Le registre des drapeaux
- Les registres d'adresses
- Le pointeur et le registre d'instruction

L'UAL

L'Unité des commandes

22/11/2023

Khaled Hassine

9

Les composantes principales

Les registres

- Les registres de travail et l'accumulateur
- La pile et le pointeur de pile
- Le registre des drapeaux
- Les registres d'adresses
- Le pointeur et le registre d'instruction

L'UAL

L'Unité des commandes

22/11/2023

Khaled Hassine

10

9

10

Les registres de travail

- Aussi dits des registres d'ordre général (**General Purpose Registers**)
- Rangement interne des informations afin d'éviter des accès inutiles à la mémoire.
- Un nombre important de ces registres augmente la performance de la machine.
- Remarque : Le nom des registres est définie par le **constructeur**

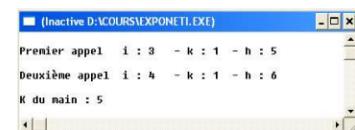
22/11/2023

Khaled Hassine

11

Utilisation des registres dans un programme en C

```
#include <stdio.h>
#include <conio.h>
int i; /* variable globale */
void f1(void)
{
    int k =1;
    static int h = 5;
    printf("%d-%d-%d\n", i, k, h);
    i++;
    k++;
    h++;
}
void main()
{
    register int k = 5;
    i = 3;
    printf("\nPremier appel");
    f1();
    printf("\nDeuxième appel");
    f1();
    printf("\nK du main : %d",k);
}
```



22/11/2023

Khaled Hassine

12

11

12

Exemple : différence en terme de nombre d'accès mémoire entre les deux variantes

```
#include <stdio.h>
#include <conio.h>
void main()
{ register int K = 10;
  int i;
  for (i=0;i<1000;i++)
    K+=5;
}
```

```
#include <stdio.h>
#include <conio.h>
void main()
{
  int i, K = 10;
  for (i=0;i<1000;i++)
    K+=5;
}
```

22/11/2023

Khaled Hassine

13

Récapitulatif des nombres d'accès

	Version 1	Version 2
Initialisation	1	2
Comparaison	1001	1001
Corps de la boucle	0	1000x2
Incrémantation	2000	2000
Total	3002	5003

22/11/2023

Khaled Hassine

14

Le registre Accumulateur

- Certaines opérations portent implicitement sur leur contenu et en reçoivent les résultats (MUL et DIV pour le 80x86).
- Certaines machines sont dites à Accumulateur (toutes les opérations se font sur ce registre)

22/11/2023

Khaled Hassine

15

Les composantes principales

- Les registres
 - Les registres de travail et l'accumulateur
 - La pile et le pointeur de pile
 - Le registre des drapeaux
 - Les registres d'adresses
 - Le pointeur et le registre d'instruction
- L'UAL
- L'Unité des commandes

22/11/2023

Khaled Hassine

16

La pile et le pointeur de pile

La pile (Stack)

- Une zone particulière de la mémoire centrale où les informations sont stockées dans leur ordre d'arrivée et restituées dans l'ordre inverse (**Last In First Out**)
- La pile est manipulée par des instructions spéciales (Empiler **PUSH** et dépiler **POP**)
- Sauvegarde automatique de l'état du processeur (appels de sous-programmes, interruptions, ...).
- Pointeur de pile (Stack Pointer, noté SP)** contient l'adresse du sommet de la pile :
 - celle du mot mémoire qui a l'adresse la plus petite dans la pile
 - où sera mise la prochaine information
- On ne peut pas modifier l'état du sommet de la pile qu'à travers les instructions PUSH et POP.

22/11/2023

Khaled Hassine

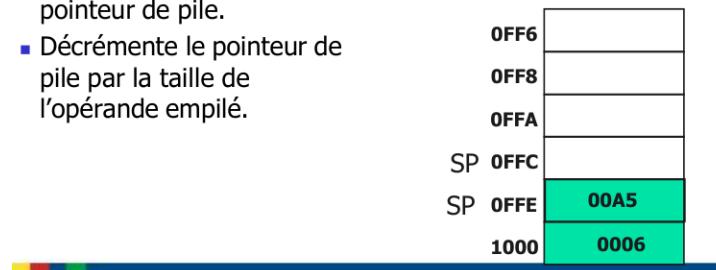
17

17

L'instruction empiler PUSH

- Copie la valeur de l'opérande dans la zone de pile pointée par le pointeur de pile.
- Décrémente le pointeur de pile par la taille de l'opérande empilé.

MOV AX, 00A5H
PUSH AX



22/11/2023

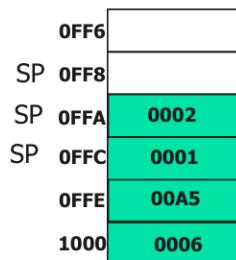
Khaled Hassine

18

18

L'instruction empiler PUSH

- Copie de deux autres valeurs dans la pile



22/11/2023

Khaled Hassine

19

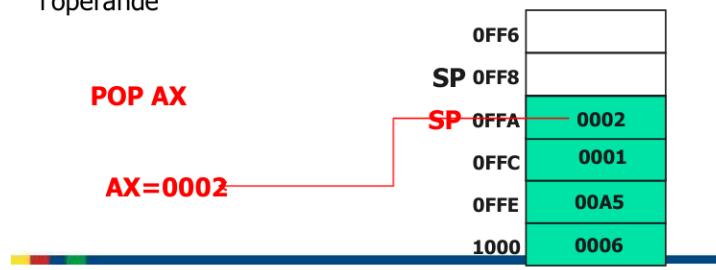
19

L'instruction dépiler POP

- Ajoute n à SP, n est la taille de l'élément déplié.
- Copie le contenu de n octets de la zone mémoire pointée par le pointeur de pile (le sommet de la pile) dans l'opérande

POP AX

AX=0002



22/11/2023

Khaled Hassine

20

20

Les composantes principales

□ Les registres

- Les registres de travail et l'accumulateur
- La pile et le pointeur de pile
- Le registre des drapeaux
- Les registres d'adresses
- Le pointeur et le registre d'instruction

□ L'UAL

□ L'Unité des commandes

22/11/2023

Khaled Hassine

21

Le registre des drapeaux

■ Indicateurs d'états –

- Appelé « Flags (States) register »
- Ces bits indicateurs sont en relation directe avec certaines instructions arithmétiques, logiques, ...
- Utilisés par les instructions de branchement conditionnel.
- Il existe des instructions spécialisées qui modifient leurs contenus

■ Exemples des drapeaux :

- Retenue (C **Carry**)
- Nullité (Z **Zero**)
- Parité (P **Parity**)
- Signe (S **Sign**)
- Débordement (O **Overflow**)

22/11/2023

Khaled Hassine

22

Les composantes principales

□ Les registres

- Les registres de travail et l'accumulateur
- La pile et le pointeur de pile
- Le registre des drapeaux
- Les registres d'adresses
- Le pointeur et le registre d'instruction

□ L'UAL

□ L'Unité des commandes

22/11/2023

Khaled Hassine

23

Les registres d'adresses

■ Permettent l'adressage des informations.

- Peuvent être des registres de travail ou des registres spécialisés.

■ Exemples :

- des registres **d'indirection** (le pointeur)
- des registres **d'indexation** dont le contenu sont considérés comme des index permettant de parcourir les éléments d'un tableau,
- des registres de **base** fournissant une adresse de référence dans les calculs adresse,
- les registres des **segments**, contenant les adresses début des parties code, données, pile, ...

22/11/2023

Khaled Hassine

24

23

24

Les composantes principales

■ Les registres

- Les registres de travail et l'accumulateur
- La pile et le pointeur de pile
- Le registre des drapeaux
- Les registres d'adresses
- Le pointeur et le registre d'instruction

■ L'UAL

■ L'Unité des commandes

22/11/2023

Khaled Hassine

25

Le pointeur d'instructions

- **Le pointeur d'instruction** (aussi dit **compteur ordinal** noté **CO** ou **Program Counter PC**)
 - contient instantanément l'adresse de l'instruction à exécuter.
 - Initialement, il contient l'adresse de la première instruction du programme à exécuter.
- C'est un :
 - **Compteur** : auto-incrémation
 - **Ordinal** : ordre d'exécution
 - **Programmable** il est possible de modifier son contenu sans respecter la séquence de compte (traitement conditionnel, appel de sous-programme, etc.).
- La taille du compteur ordinal définit la taille de **la mémoire directement adressable** (16 bits permet d'accéder à 2^{16} mots).

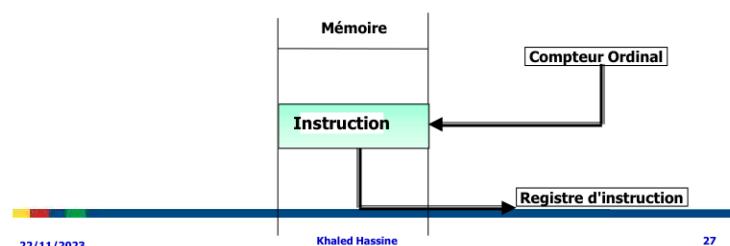
22/11/2023

Khaled Hassine

26

Les registre d'instruction

- **Le registre d'instruction**, noté **RI**, contient instantanément l'instruction à exécuter.
- Le mot mémoire dont l'adresse est fournie par le CO est amenée dans le registre d'instruction pour décodage.



22/11/2023

Khaled Hassine

27

Les composantes principales

■ Les registres

- Les registres de travail et l'accumulateur
- La pile et le pointeur de pile
- Le registre des drapeaux
- Les registres d'adresses
- Le pointeur et le registre d'instruction

■ L'UAL

■ L'Unité des commandes

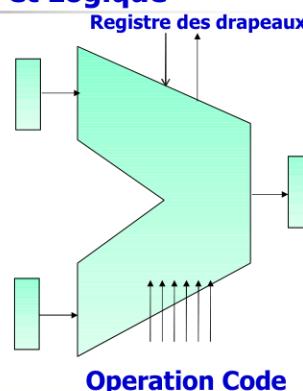
22/11/2023

Khaled Hassine

28

L'Unité Arithmétique et Logique

- L'UAL réalise une opération élémentaire (addition, soustraction, multiplication, ...) et regroupe **les circuits** qui assurent ces fonctions .
- Le registre d'état** est mis à jours (modifiés) après la fin de l'exécution d'une opération par l'UA



22/11/2023

Khaled Hassine

29

Les composantes principales

- Les registres**
 - Les registres de travail et l'accumulateur
 - La pile et le pointeur de pile
 - Le registre des drapeaux
 - Les registres d'adresses
 - Le pointeur et le registre d'instruction
- L'UAL**
- L'Unité des commandes**

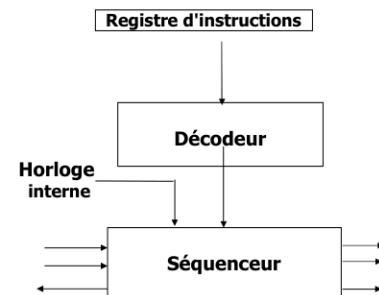
22/11/2023

Khaled Hassine

30

L'unité de contrôle / commande

- Le rôle de l'unité de contrôle / commande est de **coordonner** le travail de toutes les autres unités (UAL , mémoire, ...).
- Décodeur : Déchiffrement des instruction
- Séquenceur : câblée ou microprogrammée
 - enchainement des opérations élémentaires nécessaires à l'exécution d'une instruction
 - il génère les signaux nécessaires pour exécuter une instruction.
- L'horloge interne : la vitesse du processeur et assure la **synchronisation** de l'ensemble

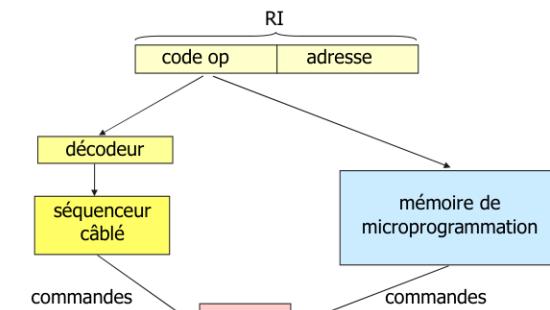


22/11/2023

Khaled Hassine

31

Approches câblée et microprogrammée



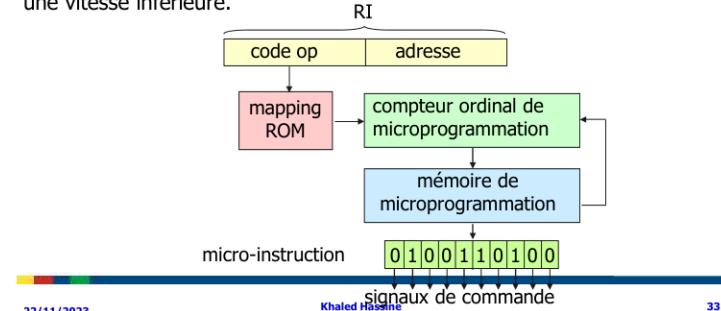
22/11/2023

Khaled Hassine

32

Séquenceur microprogrammée

- Le microprogramme (firmware) peut être stocké dans une ROM ou une EPROM.
- Cette solution est plus flexible que la logique câblée. Le prix à payer est une vitesse inférieure.

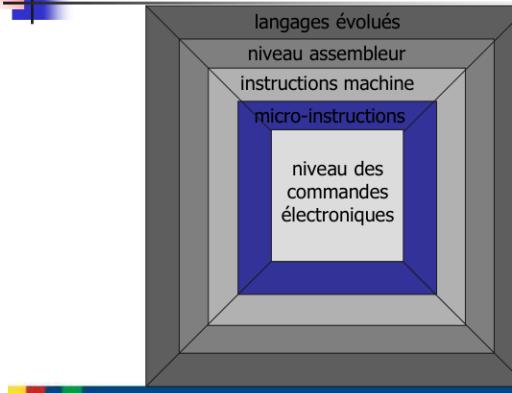


22/11/2023

Khaled Hassine

33

Niveaux de programmation

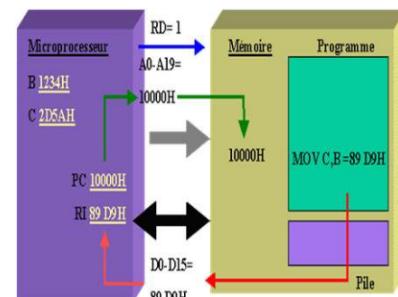


22/11/2023

Khaled Hassine

34

Voir ce film



22/11/2023

Khaled Hassine

35

Histoire d'Intel

- Fondé en 1971, c'est actuellement le premier constructeur de processeur dans le monde.
- Il a successivement nommé les microprocesseurs de sa gamme dite x86 en Intel 8086, Intel 80186, Intel 80286, Intel 80386 et Intel 80486.
- La cinquième génération devait s'appeler logiquement 80586 (ou Intel 80586, abrégé i586), Intel a préféré la nomination Pentium.

22/11/2023

Khaled Hassine

36

Histoire des processeurs de la famille 80X86

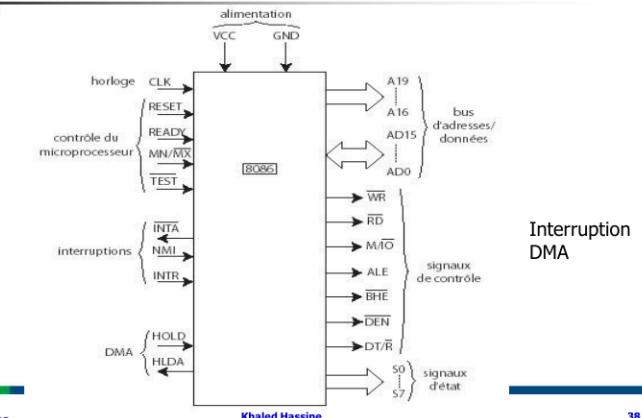
μ -processeur	Adresses	Données	Mémoire	Fréquence
8086	20	16	1 Mo	4,77 MHz
80286	24	16	16 Mo	6 MHz
80386	32	32	4 Go	16 MHz
80486	32	32	4 Go	33 MHz
Pentium	32	32	4 Go	60 MHz
Pentium Pro	32	64	4 Go	200 MHz
Pentium II	32	64	4 Go	300 MHz
Pentium III	32	64	4 Go	400 MHz
Pentium IV	32	64 (x2)	4 Go	3+ GHz

22/11/2023

Khaled Hassine

37

Le 8086 : Broches externes

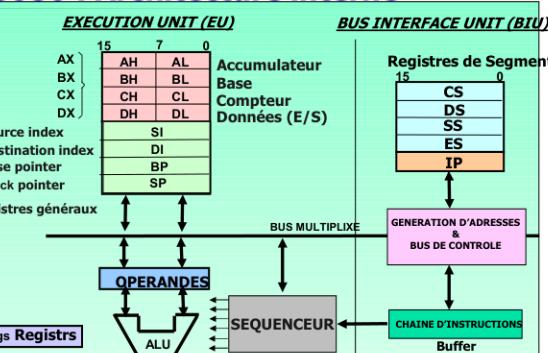


22/11/2023

Khaled Hassine

38

Le 8086 : Architecture interne

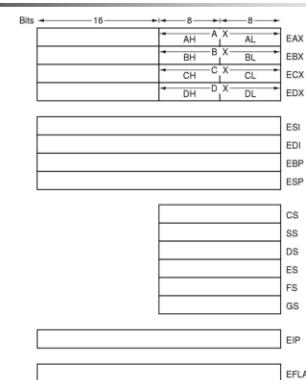


22/11/2023

Khaled Hassine

39

Principaux registres du Pentium



22/11/2023

Khaled Hassine

40

Les registres généraux du 8086

- 4 registres décomposables à usage général AX, BX, CX et DX
 - Décomposables en poids fort et faible nommé AX : AH- AL - BX : BH-BL - CX : CH-CL - DX : DH-DL
 - Pour le Pentium, ces registres sont sur 32 bits nommés EAX, EBX, ECX et EDX (E : Extended)
- Utilisations particulières des GPR :
 - AX : **accumulateur** utilisé comme registre implicite pour la multiplication et la division
 - BX : (Base) registre d'adresse pour **les adressages indirect et basé**.
 - CX : **compteur** (Counter) pour l'instruction **LOOP** (itération).
 - DX : (Data) **extension à AX** pour multiplication sur 16 bits ou le reste de la division dans le cas d'une division sur 16 bits.

22/11/2023

Khaled Hassine

41

Le registre des drapeaux du 8086

- OF** (Overflow Flag) : dépassement ou sous-passement de capacité
- SF** (Sign Flag, signe) : indique le signe du résultat
- ZF** (Zero Flag, zéro) : indique que le résultat est nul
- CF** (Carry Flag, retenue) : indique si le calcul a engendré une retenue ou un emprunt sur le bit le plus significatif
- DF** (Direction Flag) : incrémentation (DF = 0) ou la décrémentation (DF = 1) automatique des index
- AF** (Auxiliary carry Flag, retenue auxiliaire) : entre le poids faible et le poids fort d'un octet, d'un mot ou d'un double mot
- PF** (Parity Flag) : indique si les 8 bits de poids faible du résultat comportent un nombre pair de 1

				OF	DF	IF	TF	SF	ZF		AF		PF		CF
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

22/11/2023

Khaled Hassine

42

Le registre des drapeaux du 8086

- IF** (Interrupt enable Flag)
 - sert à empêcher les appels d'interruptions lorsqu'il est positionné à 1.
 - toutes les interruptions ne sont pas « *masquables* ».
 - programmé grâce aux instructions STI, CTI.
- TF** (single sTep Flag)
 - permet le débogage des programmes.
 - si ce bit est positionné, le programme s'arrête après l'exécution de chaque instruction.

22/11/2023

Khaled Hassine

43

Le registre des drapeaux du Pentium, complément

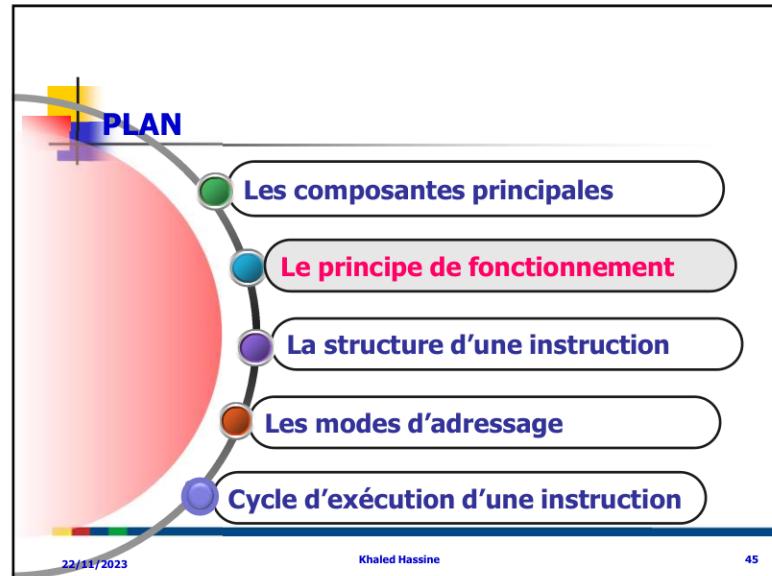
- I01-2** Niveau de priorité des I/O
- NT** Nested Task - contrôle IRET
- RF** Resume Flag : utilisé avec les registres de débogage (Reprise)
- VM** Virtual Mode : Mode virtuel

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NT	I01	I02	OF	DF	IF	TF	SF	ZF		AF		PF		CF	

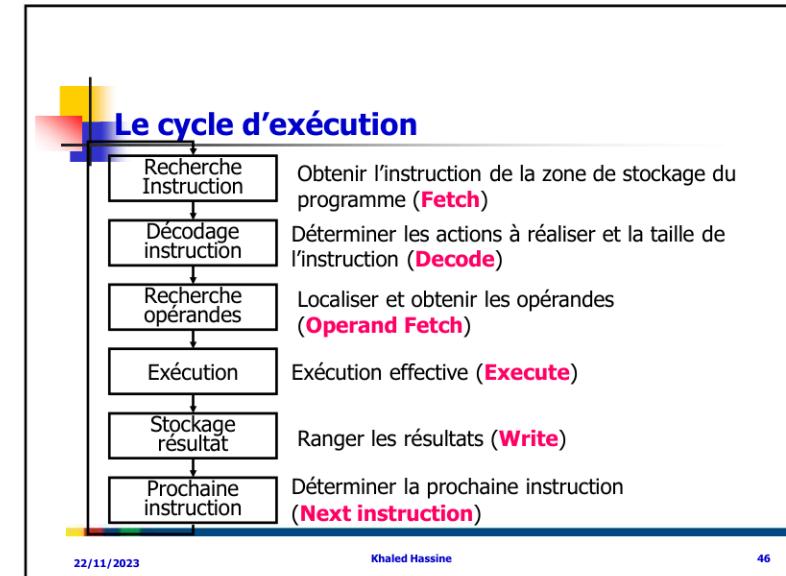
22/11/2023

Khaled Hassine

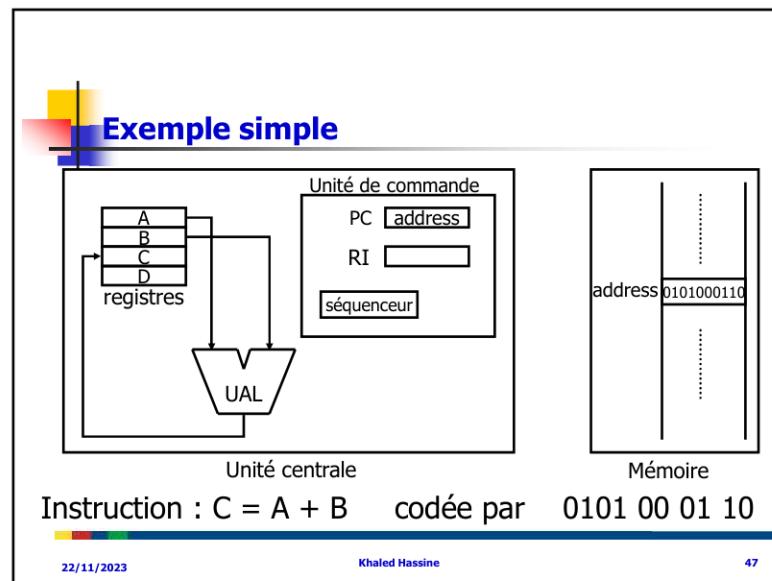
44



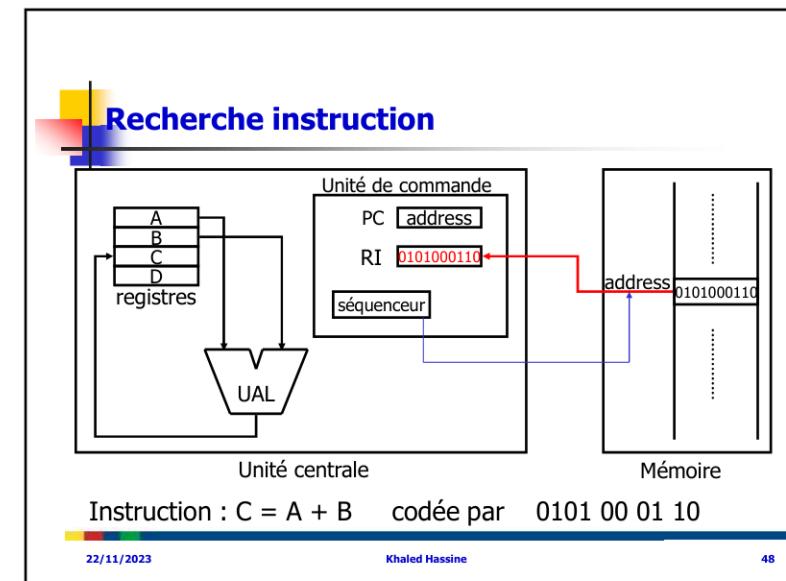
45



46

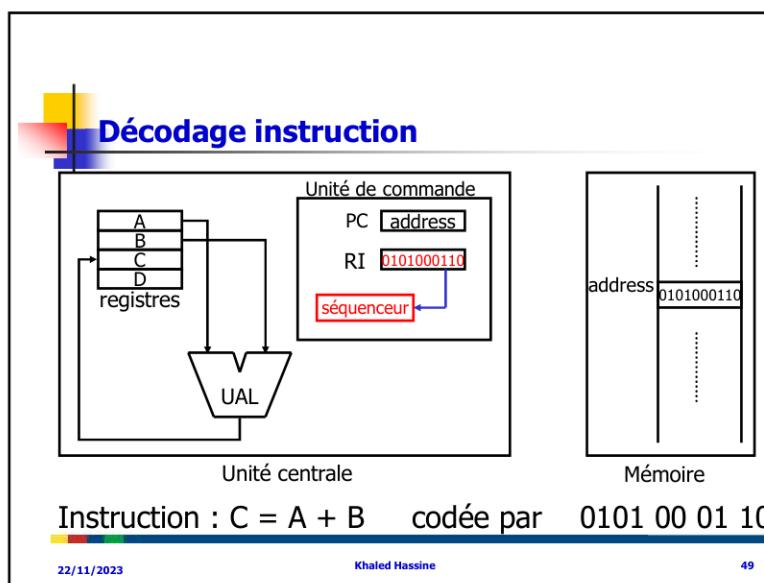


47

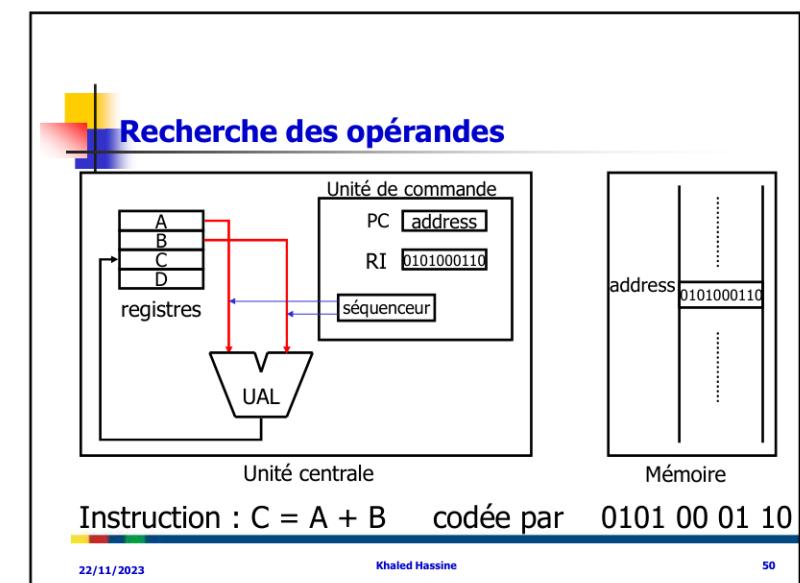


12

Décodage instruction



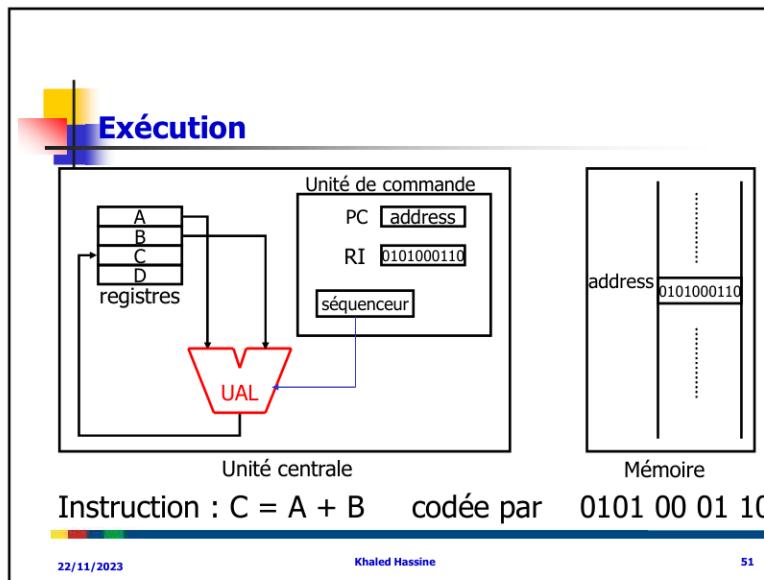
Recherche des opérandes



49

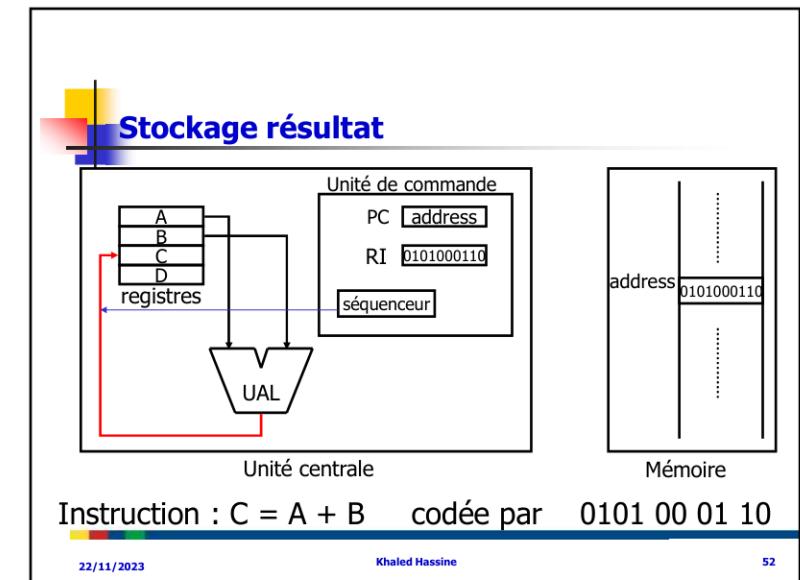
50

Exécution



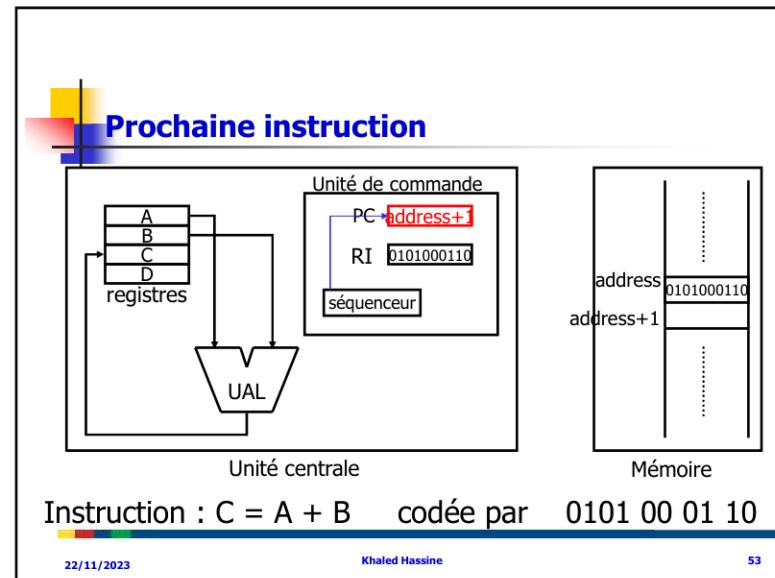
51

Stockage résultat

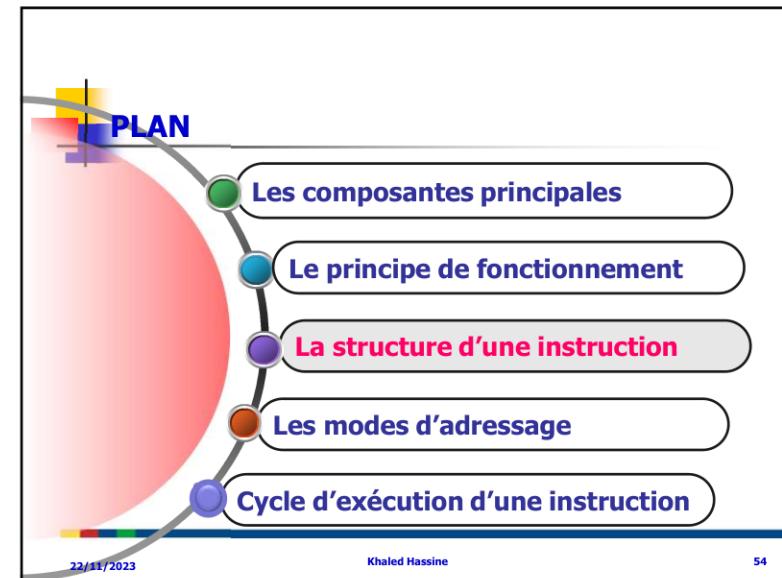


52

13



53



54

Rappel

- Tous service demandé à un ordinateur se traduit par l'exécution d'un **programme**.
- Un programme est **une suite d'instructions** définissant les opérations élémentaires que doit exécuter le processeur.
- Chaque **instruction** est codée en binaire selon un format défini **par le constructeur**. L'utilisateur a rarement l'occasion d'accéder à la représentation interne des instructions.
- L'ensemble des instructions offertes par un processeur forme **le jeu d'instructions** de la machine.

22/11/2023 Khaled Hassine 55

55

Langage assembleur

- Le constructeur d'un processeur fournit :
 - des **mémoires** désignant les instructions (DIV pour diviser, ADD pour additionner, ...)
 - des **identificateurs** prédéfinis pour les différents registres (AX, R1, ...).
 - la **grammaire** à suivre pour écrire les instructions.
- Ceux-ci forment **le langage assembleur** de la machine, qualifié de **bas niveau** en raison de sa proximité de l'architecture interne de la machine.
- Il y a un langage d'assemblage différent pour chaque type de machine ou processeur : Pentium, PowerPC, Alpha, Sparc, etc. L'assembleur n'est donc pas un langage **portable**.

22/11/2023 Khaled Hassine 56

56

Propriétés voulues des jeux d'instructions

- **Complétude** : pouvoir évaluer n'importe quelle fonction calculable.
- **Efficacité** : rapide, peu d'instructions pour les fonctions fréquemment utilisées.
- **Régularité** : contenir les fonctionnalités attendues.
- **Compatibilité** : économie de logiciel et éventuellement de matériel.

22/11/2023

Khaled Hassine

57

57

Classification des instructions

- Les instructions peuvent être classées selon **le type de traitement** en 5 catégories :
 - Les instructions de **transfert** des données entre les registres et la mémoire
 - Les instructions **arithmétiques** (ADD, DIV, SUB, ...)
 - Les instructions **binaires** (**logique**, **Rotation**, **décalage**)
 - Les instructions de **branchement** (conditionnelles, inconditionnelles, avec retour).
 - Les instructions **d'entrées sorties** (IN, OUT, ...).

22/11/2023

Khaled Hassine

58

58

Questions de base

- Quelles sont les **opérations** exécutées par le processeur ?
- Quel est le **nombre** d'opérandes de chaque opération ?
- Quels sont les types de données supportés ?
- Comment **accéder** aux opérandes pour chaque opération ?
- Comment coder (et décoder) les différentes opérations dans un ou plusieurs **formats** d'instruction cohérent ?

22/11/2023

Khaled Hassine

59

59

Instructions machines

- Le choix du format des instructions est très dépendant des facteurs suivants :
 - La taille des instructions : fixe ou variable (la longueur d'une instruction doit être un multiple des mots de base),
 - Le nombre total d'instructions,
 - La variété des instructions,
 - Le nombre de registres, les modes d'adressage, la taille de la mémoire.
 - La taille des champs
- Ces choix influent sur :
 - La rapidité du décodage
 - La taille du code généré
 - L'espace mémoire directement adressable.

22/11/2023

Khaled Hassine

60

60

Taille de l'instruction

- La **taille** d'une instruction : nombre de bits nécessaires pour coder l'instruction et les opérandes.
- Deux politiques possibles :
 - Fixe
 - Variable
- Les machines à taille d'instruction **fixe** sont **plus rapides** que celles à taille variable
 - Recherche d'instructions plus simple
 - Aucun contrôle supplémentaire n'est nécessaire
- Les machines à taille d'instruction **variable** génère un **code moins volumineux**

22/11/2023

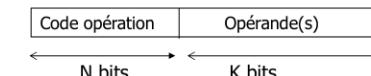
Khaled Hassine

61

61

Codage d'une instruction

- Deux parties :
 - **Code opération** : indique l'opération à faire.
 - Le ou les **opérandes** : qui contient la donnée ou une référence à (adresse de) la donnée.
- Le **format** d'une instruction peut ne pas être le même pour toutes les instructions.



22/11/2023

Khaled Hassine

62

62

Codage des instructions

- Deux types de champs pour coder une instruction :
- **Code opération** (obligatoire) :
 - sa taille indique le nombre maximal d'instructions dans le jeu d'instruction du processeur
 - **Opérandes** (dépend de l'opération):
 - coder le nombre, le mode d'adressage : une valeur constante, l'adresse d'une cellule mémoire, le numéro d'un registre, etc.
 - la taille de chaque opérande définit les limites des valeurs ou des adresses possibles selon le mode d'adressage utilisé.

22/11/2023

Khaled Hassine

63

63

Format des instructions

- Le **jeu d'instructions** d'un processeur peuvent être regroupées en classes (ou familles). Les instructions appartenant à une même classe ont la même structure de codage, connue sous le nom de **format d'instructions**.
- Le **nombre de formats d'instructions agit considérablement sur la performance de la machine**. Par exemple, une machine qui a quatre formats d'instruction dispose d'une unité de décodage nettement plus simple, donc plus rapide, qu'une machine à seize formats d'instruction.
- Le concepteur d'un processeur souhaite bien **réduire le nombre de formats** d'instructions, mais par nature même, certaines instructions exigent plus d'arguments que d'autres, et par conséquent des formats différents.

22/11/2023

Khaled Hassine

64

64

Codage des modes d'adressage

- Attribuer des **codes opérations différents** selon que les opérandes de l'instruction sont des valeurs immédiates, des registres, des zones mémoires, etc.
- Rajouter un sous champ (une partie de l'opérande), dit **spécificateur d'accès**, permettant de préciser la nature de l'opérande. Dans ce cas, le nombre de bits alloués à ce spécificateur donne le nombre des modes d'adressage de l'opérande.

22/11/2023

Khaled Hassine

65

Codage des modes d'adressage

- Utilisation des spécificateur d'accès au niveau des opérandes**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Numéro de bits	
Code opération	Spéc. 1				Opérande 1				Spéc. 2				Opérande 2				Instruction
0 1	0	0	0	0	0	0	0	1	0	1	0	0	0	1	ADD RL, 1		
0 1	0	0	0	0	0	0	0	0	1	0	0	0	0	1	ADD RL, R1		
0 1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	ADD RL, [I]	

- Utilisation des codes opération différents**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Numéro de bits	
Code opération	Spéc. 1				Opérande 1				Spéc. 2				Opérande 2				Instruction
0 1	0	0	0	0	0	0	0	0	1	0	0	0	0	1	ADD RI, 1		
0 1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	ADD RI, R1		
0 1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	ADD RI, [I]	

22/11/2023

Khaled Hassine

66

Exemple de classification des instructions

Instructions à un argument		Instructions à deux arguments	
Code	Rôle	Code	Rôle
INC AX	Incrémantion	ADD AX, BX	Addition
DEC BX	Décrémentation	SUB AX, BX	Soustraction
PUSH AX	Empiler	MOV AX, BX	Transfert
POP AX	Dépiler	...	
JMP étiquette	Branchement		
CALL SP	Appel de sous-programme		

22/11/2023

Khaled Hassine

67

Exemple de codage en fonction de la classification

- Pour les instructions à un opérande, il faut distinguer les instructions de saut et de branchement des autres :
- Les instructions de branchement présentent la contrainte d'avoir comme opérande **une adresse mémoire**.
- Le nombre de bits réservés à ce champ détermine l'espace mémoire directement adressable par ces instructions qu'on souhaite maximiser.
- On a intérêt alors à favoriser ce champ en lui allouant un nombre maximal de bits.

22/11/2023

Khaled Hassine

68

Exemple : 3 Formats

- On prend comme hypothèse que les instructions de branchement commencent par 1 et toutes les autres commencent par 0.
- On peut définir à priori les trois formats suivants:
 - Format I : Pour les instructions de branchement
 - Format J : Pour les instructions à un opérande
 - Format K : Pour les instructions à 2 opérandes
- Le choix de coder (01xx...) les instructions de classe K pour les distinguer des instructions de format J. Cette distinction apporte à ce niveau de conception un avantage de **lisibilité**.
- Cependant, les instructions commençant par un 1 de classe I est **un choix impératif**. Ce choix permet de réduire le nombre d'instructions de cette classe et par conséquent **d'augmenter l'espace mémoire directement adressable**.

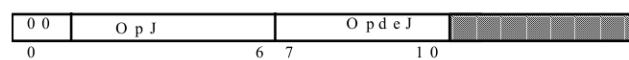
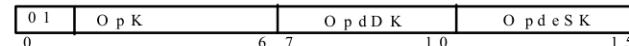
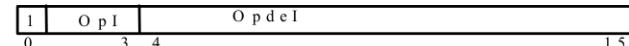
22/11/2023

Khaled Hassine

69

69

Exemple : 3 Formats ...



22/11/2023

Khaled Hassine

70

70

Classification des machines

- À 0 adresse : machine à pile
- À 1 adresse : machine à accumulateur
- À 2 adresses : machine à registres généraux
- À 3 adresses : machine à chargement-Rangement

22/11/2023

Khaled Hassine

71

71

Machine à pile

- Se base sur deux opérations élémentaires :
 - Empiler
 - Dépiler
- Etapes nécessaires :
 - Empiler les opérandes
 - Effectuer l'opération
 - Dépiler le résultat

22/11/2023

Khaled Hassine

72

72

Machine à 1 adresse

- Dans ce type de machine pour chaque instruction il faut préciser uniquement l'adresse du **deuxième opérande**.
 - Le **premier opérande** est implicite un registre généralement l'**accumulateur**.
 - Le **résultat** est mis dans le **registre accumulateur**.
 - Exemple :
 - ADD A
 - Équivalent algorithmique : $ACC \leftarrow (ACC) + A$

22/11/2022

Khaled Hassim

7

Machine à 2 adresses

- Dans de type de machine, pour chaque instruction, il faut préciser :
 - l'adresse du premier opérande
 - du deuxième opérande ,
 - l'adresse du résultat est implicitement l'adresse du deuxième opérande.

Code opération	Opérande1	Opérande2
-----------------------	------------------	------------------

- Exemple : ADD A, B (A \leftarrow A + B)

73

74

Machine à 3 adresses

- Dans ce type de machine pour chaque instruction, il faut préciser :
 - Le premier opérande
 - Le deuxième opérande
 - L'emplacement du résultat

Code opération	Opérande1	Opérande2	Résultat
----------------	-----------	-----------	----------

- Exemple : ADD A,B,C (C \leftarrow A+B)
 - Dans ce type de machine la taille de l'instruction est plus grande.

22/11/202

Khaled Hassim

7

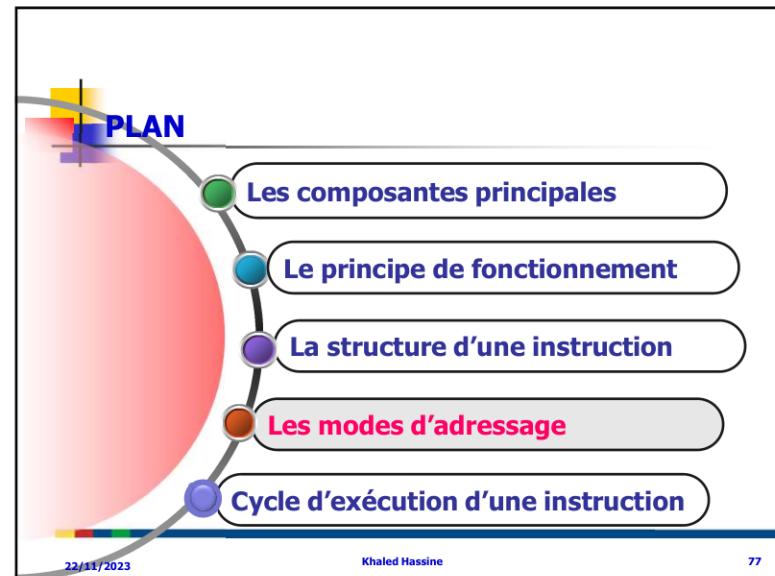
Exemple pour $C = A + B$

Pile	Accumulateur	Registre - mémoire	Registre - Registré
Push A	Load A	Load R1, A	Load R1, A
Push B	Add B	Add R1, B	Load R2, B
Add	Store C	Store C, R1	Add R3, R1, R2
Pop C			Store C, R3

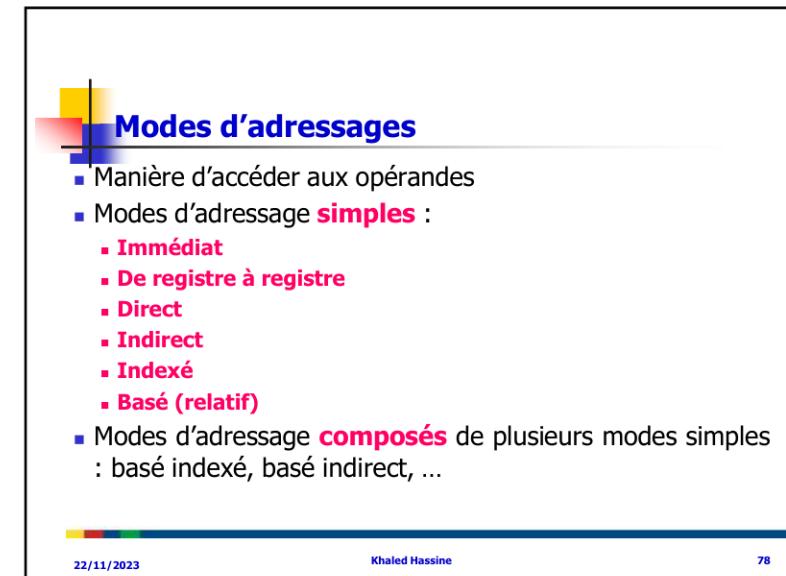
1

Khaled Hassine

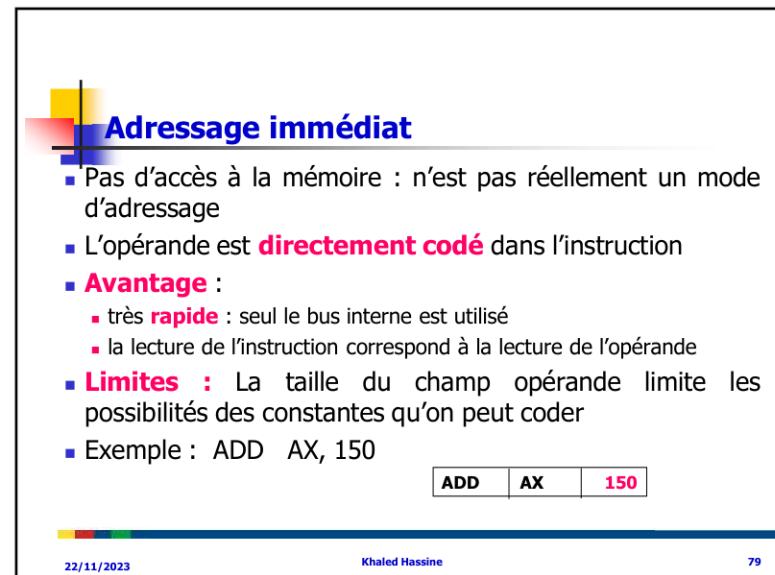
76



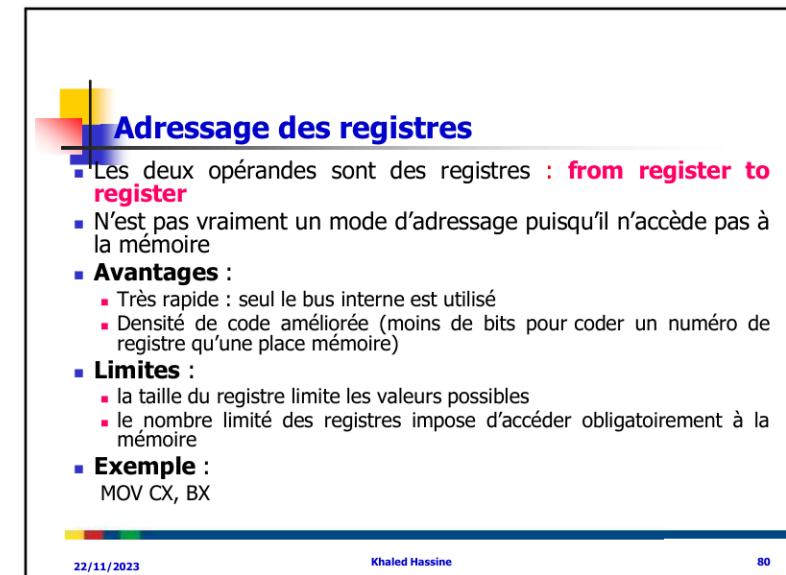
77



78



79



80

20

Adressage direct

- Le champs opérande de l'instruction contient **l'adresse effective de l'opérande** (emplacement en mémoire) : Pour réaliser l'opération, il faut récupérer (lire) l'opérande à partir de la mémoire
- Avantage** : Le plus simple et le plus rapide des modes d'adressage qui accèdent effectivement à la mémoire
- Limite** : L'espace mémoire **directement adressable** dépend de la taille du champ opérande

22/11/2023

Khaled Hassine

81

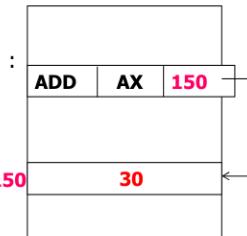
Adressage direct : Exemple

$$AX \leftarrow AX + (ADR)$$

Si l'accumulateur contient 20

A la fin de l'exécution on obtient :

50 ($20 + 30$)



22/11/2023

Khaled Hassine

82

Adressage indirect

Le champ opérande contient **l'adresse de l'adresse** de l'opérande.

Pour réaliser l'opération, il faut :

- Récupérer **l'adresse de l'opérande** à partir de la mémoire.
- Chercher la **valeur** de l'opérande à partir de la mémoire



Exemple : $AX \leftarrow AX + ((ADR))$

- Initialement l'accumulateur contient la valeur 20
- Il faut récupérer l'adresse (150) de l'instruction.
- Récupérer l'adresse de l'opérande à partir de la cellule 150 (**la valeur 200**)
- Récupérer la valeur de l'opérande à partir de la cellule 200 (**la valeur 40**)
- Ajouter la valeur 40 avec le contenu de l'accumulateur (20) pour obtenir **60**



22/11/2023

Khaled Hassine

83

Adressage Indirect

- Permet de réaliser la notion de **pointeur** et la **gestion dynamique de la mémoire** dans les langages de haut niveau

- Accéder à **un espace mémoire plus large** que le mode d'adressage direct

- Nécessite **n+1 accès** à la mémoire n étant le niveaux d'indirection :

- n accès pour la détermination de **l'adresse effective**
- 1 accès pour déterminer la valeur

22/11/2023

Khaled Hassine

84

Adressage indexé

- L'adresse effectif de l'opérande est **relatif** à une adresse mémoire qui se trouve généralement dans un registre spécial (**registre d'index** avec **auto incrémentation et auto décrémentation**).
- Permet l'accès à des zones mémoires contigües (structures répétitives)
- Adresse effective** = Opérande + (index)

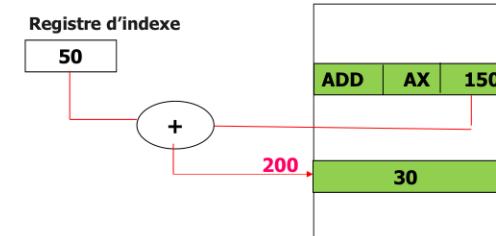
22/11/2023

Khaled Hassine

85

Adressage indexé

- Adresse effective** = Opérande + (index)



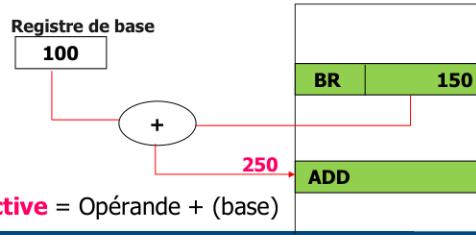
22/11/2023

Khaled Hassine

86

Adressage Relatif / Basé

- L'adresse effectif de l'opérande est relatif à une zone mémoire.
- L'adresse de cette zone se trouve dans un registre spécial (**registre de base**).
- Ce mode d'adressage est utilisée pour les instructions de branchements.



22/11/2023

Khaled Hassine

87

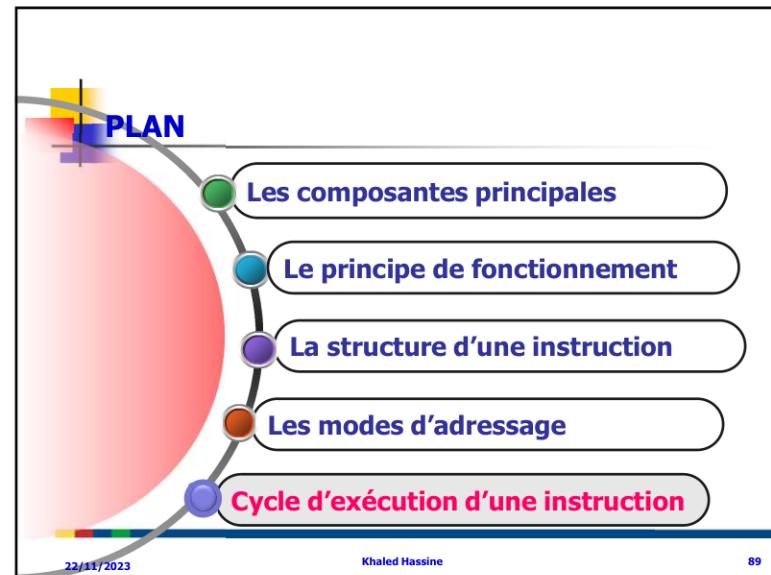
Modes d'adressage composés

- Association de plusieurs modes d'adressage simples.
- Surmonter les limites des modes d'adresses simples : accéder à un espace mémoire plus large
- On doit souvent donner une priorité d'un mode par rapport à un autre :
 - Indirect indexé : post indexé, pré-indexé
 - Basé indexé
 - ...
- Inconvénients : temps supplémentaire pour le calcul complexe de l'adresse.

22/11/2023

Khaled Hassine

88



89

Etat de la mémoire et des registres

Etat des registres		Etat de la mémoire			
		Adresse	Contenu	Adresse	Contenu
AX = 1122H	BX = 0006H	20020H	80H	FFF00H	00H
CX = 0104H	DX = 0000H	20021H	70H	FFF01H	02H
IP = 00FFH	SP = FFEEH	20022H	60H	FFF02H	01H
CS = 1000H	DS = 2000H	20023H	50H	FFF03H	50H
BP = 0010H	SI = 0000H	20024H	40H	FFF04H	04H
SS = FFF0H	DI = 0002H	20025H	30H	FFF05H	FFH
ES = 4F00H		20026H	20H	FFF06H	ACH
		20027H	10H	FFF07H	00H
		20028H	09H	FFF08H	20H
	

N° bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Drapeau					OF	DF	IF	TF	SF	ZF		AF		PF		CF
Etat					0	0	1	1	0	1	1	1	0	1		1

22/11/2023 Khaled Hassine 90

90

Exécution pas à pas des instructions

Donner les différentes phases (microcommande et micro-instructions correspondantes) nécessaires pour l'exécution de l'instruction suivante :

ADD BX, X (BX ← BX+X).

X est l'identificateur d'une variable mémoire (mot de 2 octets) implantée à l'adresse 24H dans le segment des données :

X DW 3040H

22/11/2023 Khaled Hassine 91

91

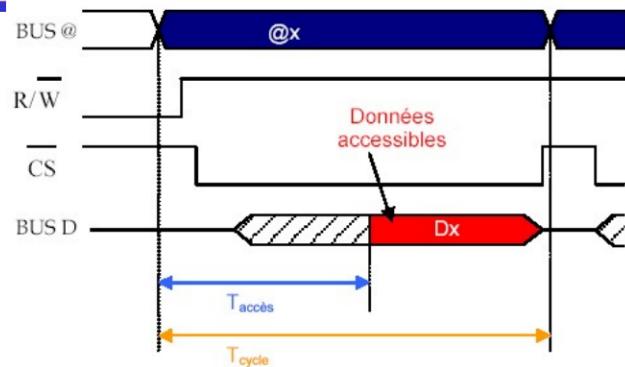
Demander l'instruction de la mémoire

- Chaque **phase** comporte un certain nombre d'opérations élémentaires (**micro-instructions**) exécutées dans un ordre bien précis (suite à des microcommandes **générées par le séquenceur**).
- Les étapes nécessaires pour cette première phase sont les mêmes étapes qu'une lecture à partir de la mémoire
- On commence par le calcul d'adresse qui peut se faire par l'UAL ou par une unité de calcul d'adresse dédiée.

22/11/2023 Khaled Hassine 92

23

Rappel Cycle de lecture à partir de la mémoire



93

Demander l'instruction de la mémoire

- Φ_1 : Demande de l'instruction de la mémoire.
 - μ_{11} : Calcul de l'adresse de l'instruction : $CS*16+IP = 1000H \times 10H + 00FFH = 100FFH$
 - μ_{12} : Déposer l'adresse de l'instruction sur le bus d'adresse.
 - μ_{13} : Ordre de lecture vers la mémoire (via le bus de Contrôle/commande).

22/11/2023 Khaled Hassine 94

94

Incrémantion du CO

- En attendant que la mémoire **décode l'adresse** et **dépose son contenu sur le bus des données**, le processeur procède à l'incrémentation du CO.
- Ceci permet aussi de surmonter la différence de vitesse entre le processeur et la mémoire.
- En général, on prend comme incrément **la taille la plus fréquente**, égal à 2 pour le 8086, le précurseur du Pentium
- **En cas d'erreur** (instruction de taille différente de 2), la correction se fait lors de la phase suivante de décodage.

22/11/2023

Khaled Hassine

95

Incrémantion du CO

- Φ_2 : Incrémentation du compteur ordinal
 - μ_{21} : $CO \leftarrow CO + \text{Taille d'une instruction}$.
 $IP = IP + 2 = 00FFH + 2 = 0101H$

22/11/2023 Khaled Hassine 96

96

Lecture effective et décodage de l'instruction

- φ_3 : Lecture effective de l'instruction et son décodage.
- μ_{31} : Transfert du contenu du bus de données vers le Registre d'Instruction.
- Décodage de l'instruction. A ce niveau, l'UC connaît qu'on a à faire à une instruction d'addition de BX avec la cellule mémoire X.

22/11/2023

Khaled Hassine

97

97

Préparation à l'exécution

- φ_4 : Préparation à l'exécution.
- μ_{41} : Calcul de l'adresse de X :
 $DS*16+@X = 2000H \times 10H + 0024H = 20024H$
- μ_{42} : Mettre l'adresse de X sur le bus d'adresse.
- μ_{43} : Ordre de lecture vers la mémoire.
- μ_{44} : Mettre le contenu de BX dans la première entrée de l'UAL.
- μ_{45} : Transfert du contenu du bus de données vers la deuxième entrée de l'UAL (**3040H**).

22/11/2023

Khaled Hassine

99

99

Lecture effective et décodage de l'instruction

- Remarquons qu'on a connu que l'instruction est une addition qu'à ce niveau, on en déduit que **les trois premières phases sont identiques quelque soit l'instruction**.
- A ce niveau, on **corrige éventuellement le compteur ordinal** (si la taille de l'instruction est différente de 2) et on ramène le cas échéant le reste de l'instruction.

22/11/2023

Khaled Hassine

98

98

Exécution effective

- φ_5 : Exécution effective et sauvegarde du résultat.
- μ_{51} : Ordre vers l'UAL pour effectuer l'opération d'addition et mise à jour du registre des drapeaux.
- $BX(0006H)+[20024H] = 0006H+3040H = \text{3046H}$
- μ_{52} : Déposer la sortie de l'UAL vers le registre BX
- Le nouveau contenu de BX est **3046H**.

22/11/2023

Khaled Hassine

100

100

Remarques

- Phases 1, 2 et 3 sont identiques quelque soit l'instruction (on ne connaît l'instruction qu'à la fin de la 3^{ème} phase, après le décodage).
- Phase 4 (préparation à l'exécution) dépend de la nature des opérandes et les modes d'adressages utilisés.
- La phase 5 (exécution effective) dépend du code opération et de la nature des opérandes et leurs modes d'adressage.

22/11/2023

Khaled Hassine

101

Exercice : exécution des instructions

- Quelles sont les modifications à apporter à la réponse précédente pour l'exécution des instructions suivantes :
 - ADD X, BX ($X \leftarrow BX + X$).
 - MOV AX, X ($AX \leftarrow X$).
 - MOV X, AX ($X \leftarrow AX$).
 - MUL BL ($AX \leftarrow AL * BL$).
 - MUL Byte Ptr [BX]
 - MUL Word Ptr [BX]
 - JMP Label
 - JE Label

22/11/2023

Khaled Hassine

102

101

102



103