

Architecture des Ordinateurs

TD7 : Processeur

Exercice 1 : Architecture du processeur

Quels sont les caractéristiques de chacun des deux modèles ?

Modèle de Von Neumann

- 1 seule mémoire pour les instructions et les données
- 1 bus d'adresse, 1 bus de donnée
- Avantage : gestion unifiée de la mémoire
- Inconvénient : logique complexe pour lire/écrire tour à tour les données et les instructions, goulot d'étranglement pour l'accès à la mémoire

Modèle Harvard

- 2 mémoires : 1 pour les instructions, 1 pour les données
- 2 bus d'instructions (adresses / instructions), 2 bus de données (adresses / données)
- Avantage de Harvard : plus grande rapidité d'accès car données et programme peuvent transiter simultanément.
- Inconvénients de Harvard : le prix, à cause du doublement des bus + gestion plus complexe car le contrôle des accès données et programme différencié

4) Quels sont les principaux registres utilisés ?

Un registre est un emplacement de mémoire interne à un processeur.

Les registres constituent la mémoire la plus rapide d'un ordinateur, mais dont le coût de fabrication est le plus élevé car la place dans un microprocesseur est limitée.

Leur capacité dépasse donc rarement quelques dizaines d'octets.

AX : Accumulateur

IP (ou PC ou CO) : Instruction Pointer (Program Counter, Compteur Ordinal)

IR : Instruction Register

Pile (Stack en anglais)

SP : Stack Pointer

Exercice 3 : (Exemple d'instruction assembleur INTEL 8086)

L'instruction suivante (avec AX un accumulateur 16 bits) **AND AX, 06** a pour code machine : 25 06 00_H. Cette instruction est implantée à l'adresse 01 00_H.

1) Indiquer le contenu des registres IR/RI (*Instruction Register ou Registre d'Instruction*) et IP (ou PC ou CO, *Instruction Pointer ou Program Counter ou Compteur Ordinal*) juste avant l'exécution de l'instruction. IR :

2) Indiquer le contenu du registre IP juste après exécution de l'instruction.

NB : Le microprocesseur 8086 utilise un bus d'adresse de 20 bits, ce qui lui permet d'atteindre $2^{20} = 1M$ cases mémoire de 1 octet chacune.

a) Indiquer le contenu des registres IR et IP juste avant l'exécution de l'instruction

a. Juste avant

IR : 25 06 00

IR contient le code Machine de la prochaine instruction à exécuter

IP : 01 00

$01\ 00_H$ = adresse d'implantation du programme = contenu de IP, noté (IP) =
adresse de la prochaine instruction à exécuter

Mémoire

01 00 _H	25
01 01 _H	06
01 02 _H	00
01 03 _H	...

Ecrivez le code machine de cette instruction dans la mémoire, sachant que
chaque case mémoire possède une capacité de 1 octet.

b) Indiquer le contenu du registre IP juste après exécution de l'instruction

b. Juste après

IP : 01 03

$01\ 00_H + 3\ \text{octets} = 01\ 03_H$

Exercice 4 : Jeux d'instruction

Soit l'extrait de programme ASSEMBLEUR INTEL 8086 suivant, stocké à l'adresse 0100_H (via le code (ASM ORG 100h) avec les valeurs initiales :

AX = 0000_H, BX = 0000_H, et le Flag Z = 0.

L'état de pile : FFFE_H = 0000_H, FFFC_H = 0000_H, FFFA_H = 0000_H.

SP (Pointeur de pile) = FFFE_H

1	MOV AX, 0100h	B8 00 01	Ecrit 01 00h dans AX : (AX) = 01 00 – (AH) = 01 ; (AL) = 00
2	MOV BX, 0304h	BB 04 03	Ecrit 03 04h dans BX : (BX) = 03 04 – (BH) = 03 ; (BL) = 04
3	Boucle : ADD AL, 1	04 01	Ajoute 1 à l'octet de poids faible de AX noté AL : (AL) = (AL)+1
4	CMP AL, 2	3C 02	Compare AL à 2 ; place Z à 1 en cas d'égalité
5	JNE Boucle	75 FA	Saut à l'étiquette Boucle si Z=0 (si pas d'égalité)
6	PUSH AX	50	Empile le contenu de AX dans la pile (STACK) AX -> STACK
7	PUSH BX	52	Empile le contenu de BX dans la pile (STACK) BX -> STACK

	Instruction ASM	IP	AX	BX	Flag Z	SP	STACK FFFF, FFFE, FFFD, FFFC, FFFB, FFFA	
0	Estat Initial	01 00	00 00	00 00	0	FF FE	00 00 00 00 00 00	1 MOV AX, 0100h 88 00 01 2 MOV BX, 0304h 8B 04 03 3 Boucle: ADD AL, 1 04 01 4 CMP AL, 2 3C 02 5 JNE Boucle 75 FA 6 PUSH AX 50 7 PUSH BX 53
1	MOV AX, 0100h (B8 00 01)	01 03	01 00	00 00	0	FF FE	00 00 00 00 00 00	
2	MOV BX, 0304h (BB 04 03)	01 06	01 00	03 04	0	FF FE	00 00 00 00 00 00	
3	ADD AL, 1 (04 01)	01 08	01 01	03 04	0	FF FE	00 00 00 00 00 00	0100 B8 0101 00 0102 01 0103 BB 0104 04 0105 03 0106 04 0107 01 0108 3C 0109 02 010A 75 010B FA 010C 50 010D 53
4	CMP AL, 2 (3C 02)	01 0A	01 01	03 04	0	FF FE	00 00 00 00 00 00	
5	JNE Boucle (75 FA)	01 06	01 01	03 04	0	FF FE	00 00 00 00 00 00	
6	ADD AL, 1 (04 01)	01 08	01 02	03 04	0	FF FE	00 00 00 00 00 00	
7	CMP AL, 2 (3C 02)	01 0A	01 02	03 04	1	FF FE	00 00 00 00 00 00	
8	JNE Boucle (75 FA)	01 0C	01 02	03 04	1	FF FE	00 00 00 00 00 00	
9	PUSH AX (50)	01 0D	01 02	03 04	1	FF FC	00 00 01 02 00 00	
10	PUSH BX (53)	01 0E	01 02	03 04	1	FF FA	00 00 01 02 03 04	