

TD 2 ADOintro. Architecture de Von Neumann

Exercice 1 : Modes d'adressage (Complément de cours)

Les modes d'adressage

Un mode d'adressage est une méthode permettant d'interpréter, d'accéder à un opérande (aux données) lors de l'exécution d'une instruction. Par exemple l'assembleur MC68000 de Motorola présente 6 modes d'adressage :

1. *Adressage Direct* : l'opérande est un registre de données ou d'adresse.
2. *Adressage Indirect* : l'opérande est désigné :
 - soit par le contenu d'un registre d'adresse,
 - soit par l'addition du contenu d'un registre d'adresse et d'une constante (*offset*) et/ou du contenu d'un registre de donnée ou d'adresse (*index*).
3. *Adressage Immédiat* : la donnée est fournie dans le code instruction.
4. *Adressage Absolu* : l'adresse de la donnée est fournie dans le code instruction.
5. *Adressage Relatif* : l'adresse de la donnée est calculée par addition du contenu du Compteur Ordinal et d'un *offset* et/ou d'un *index*.
6. *Adressage Implicite* : les registres impliqués sont les registres de contrôle (Registre d'Etat, Compteur Ordinal, Pile).

Motorola donne le tableau 4.1 récapitulant les modes d'adressage avec les codes associés. An et Dn désignent respectivement les registres d'adresse et de donnée.

Mode	Code	Champ registre	Syntaxe
Direct	000	Num. reg.	Dn
Direct	000	Num. reg.	An
Indirect	010	Num. reg.	(An)
Indirect	011	Num. reg.	(An) +
Indirect	100	Num. reg.	-(An)
Indirect	101	Num. reg.	d(An)
Indirect	110	Num. reg.	d(An, Rm)
Absolu	111	000	xxxx
Absolu	111	001	xxxxxxxx
Relatif	111	010	Rel. CO
Relatif	111	011	Rel. CO+Rm
Immédiat	111	100	#xxxx

Table 4.1: Modes d'adressage du M68000.

Exemple 1 (Instructions de transfert)

`move, w d3, -(a4)` : les 16 bits de poids faible (extension ,w) du registre d3 sont transférés à l'adresse donnée par le contenu du registre a4 décrétementé de 1 avant transfert. Le rangement en mémoire se fait octet de poids faible d'abord

Exemple 2 (Instructions de comparaison)

`cmp, l d4, d2` : compare d4 à d2 en effectuant la soustraction d2-d4. Les drapeaux (sauf X) sont modifiés en conséquence.

Exemple 3 (Instructions de branchement inconditionnel)

`bra plus_loin` : l'argument du branchement est un déplacement sur 8 ou 16 bits qui est ajouté au contenu du compteur ordinal (celui-ci contient alors l'adresse de l'instruction qui suit).

`jmp (a3)` : branchement à l'adresse donnée par le contenu de a3.

L'instruction ASSEMBLEUR INTEL 8086 :

AND AX, 06 (AX est un registre accumulateur 16 bits)

a pour code Machine :

25 06 00_H (convention INTEL Little Endian)

Elle est implantée à l'adresse 01 00_H.

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

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

Exercice 2 : Programme ASSEMBLEUR (ASM)

Soit l'extrait de programme ASSEMBLEUR INTEL 8086 suivant, stocké à l'adresse 01 00_H (via le code ASM **ORG 100h**) avec les valeurs initiales : **AX = 00 00_H**, **BX = 00 00_H** et l'état de pile (STACK) suivant :

STACK : **FF FE_H : 00 00_H**
 FF FC_H : 00 00_H
 FF FA_H : 00 00_H
 ...

Code ASM	Code Machine	Commentaire (AX) signifie : contenu de AX
MOV AX, 0100h	B8 00 01	Ecrit 01 00 _H dans le registre AX ; (AX) = 01 00 (convention INTEL Little Endian)
MOV BX, 0304h	BB 04 03	Ecrit 03 04 _H dans le registre BX ; (BX) = 03 04 (convention INTEL Little Endian)
Boucle: ADD AL, 1	04 01	Ajoute 1 à l'octet de poids faible de AX noté AL : (AL) = (AL) + 1
CMP AL, 2	3C 02	Compare (AL) à 2 ; place le bit de Flag Z à 1 en cas d'égalité de la comparaison
JNE Boucle	75 FA	Saut à l'étiquette Boucle si le bit Z = 0 (\equiv s'il n'y a pas égalité) (<i>Jump Not Equal</i>)
PUSH AX	50	Empile le contenu de AX dans la pile (STACK) : (AX) → STACK
PUSH BX	53	Empile le contenu de BX dans la pile (STACK) : (BX) → STACK

Attention : ne pas introduire de caractère « espace » dans l'étiquette « Boucle: »

a) Compléter ce tableau lors de l'exécution complète et pas à pas du programme, en indiquant le contenu des registres spécifiés :

Instruction ASM	IP	AX	BX	Flag Z	SP	STACK (FFFF,FFFE,FFFD,FFFC,FFFB,FFFA)
Etat initial				0		
MOV AX, 0100h						

TP 2 ADOintro. Architecture de Von Neumann

b) Vérifier les résultats de la question précédente en assemblant (menu *emulate*) le programme précédent avec le simulateur Assembler with Microprocessor Simulator 8086 (fichier **emu8086v408r.exe**) téléchargeable à l'adresse :

http://pcwin.com/Software_Development/Debugging/Assembler_with_Microprocessor_Simulator_8086/index.htm (rubrique *download*)