

	Permet de/d'...	Exemple	Modes d'adressage	Note	Drapeaux affectés												
MOV	Déplacer des données, de OP2 vers OP1	MOV EBX, dword ptr [0x100] Lit 4 octets en mémoire depuis l'adresse 0x100, et les transfère dans le registre EBX  MOV byte ptr [RAX+RSI-4], 0xFF Ecrit l'octet 0xFF en mémoire, à l'endroit pointé par RAX+RSI-4	<table><tr><th>Op.1</th><th>Op.2</th></tr><tr><td>reg</td><td>imm</td></tr><tr><td>mem</td><td>imm</td></tr><tr><td>reg</td><td>reg</td></tr><tr><td>reg</td><td>mem</td></tr><tr><td>mem</td><td>reg</td></tr></table>	Op.1	Op.2	reg	imm	mem	imm	reg	reg	reg	mem	mem	reg	Il n'est pas permis de combiner des accès à la mémoire pour les deux opérandes.	Aucun.
Op.1	Op.2																
reg	imm																
mem	imm																
reg	reg																
reg	mem																
mem	reg																
XCHG	Echanger le contenu de ses deux OP.	XCHG AL, AH Permuter les 8 bits de poids faible et de poids fort du registre AX  XCGH EAX, EAX N'a aucun effet. S'abrévie en NOP (No OPeration)	<table><tr><th>Op.1</th><th>Op.2</th></tr><tr><td>reg</td><td>reg</td></tr><tr><td>reg</td><td>mem</td></tr><tr><td>mem</td><td>reg</td></tr></table>	Op.1	Op.2	reg	reg	reg	mem	mem	reg		Aucun.				
Op.1	Op.2																
reg	reg																
reg	mem																
mem	reg																
ADD	Ajouter OP2 à OP1.	ADD R10, -1 Décrémente le contenu de R10 d'une unité	Idem MOV.	La même instruction peut être employée pour des nombres non signés ou bien représentés par complément à deux.	CF, ZF, SF et OF.												
SUB	Soustraire OP2 à OP1	SUB R10, 1 Même effet que précédente	Idem ADD.		Idem ADD												
CMP	(Même opération que SUB, mais ne modifie pas OP1) Mettre à jour les drapeaux. → effectuer des comparaisons de valeurs, pouvant servir de base à des décisions dans les instructions suivantes.	CMP EAX, EBX Calcule la diff Δ=EAX-EBX On a alors: ZF=1 ssi Δ=0, ie. EAX = EBX SF=1 ssi Δ<0, ie. EAX < EBX	Idem ADD.		Idem ADD.												
INC/DEC	Incrémenter (INC) ou décrémenter (DEC) leur OP, qui sert donc de source et de destination.	INC byte ptr [RBX] Ajoute 1 à l'octet pointé par RBX	<table><tr><th>Op.1</th></tr><tr><td>reg</td></tr><tr><td>mem</td></tr></table>	Op.1	reg	mem		CF est préservé  ZF, SF et OF sont mis à jour.									
Op.1																	
reg																	
mem																	

OPX: Opérande n°X

	Permet de/d'...	Exemple	Modes d'adressage	Note	Drapeaux affectés
<b>MUL</b>	<b>Multiplier</b> deux nombres non-signés de n bits, avec $n \in \{8, 16, 32, 64\}$ . Le résultat est représenté sur 2n bits.	<code>MUL dword ptr [0x1234]</code> Multiplie l'entier non-signé de 32 bits pointé par 0x1234 par le contenu de EAX. Les 32 bits de poids fort du résultat sont écrit dans EDX, les 32 bits de poids faible dans EAX	Op.1 imm reg mem	Contrairement à l'addition, le fait que les nombres sont représentés de façon signée ou non signée influence la multiplication.	CF et OF sont mis à 0 si le résultat de l'opération est représentable sur n (=taille des OP) bits et à 1 sinon. Autres drapeaux → modifiés de façon arbitraire.
<b>IMUL</b>	Similaire à MUL, mais calcule le produit de deux nombres <b>signés</b> .	Idem MUL.	Idem MUL.	Il existe d'autres formes de cette instruction (à deux et trois OP), que nous n'étudierons pas.	Idem MUL.
<b>AND/OR/XOR</b>	Appliquer une <b>opération booléenne</b> bit par bit à leurs deux OP, et écrire le résultat dans OP1  <ul style="list-style-type: none"> <li>• AND: Le résultat est égal à 1 ssi <math>OP1 = OP2 = 1</math> ("et" logique).</li> <li>• OR: Le résultat est égal à 1 ssi au moins <math>OP1=1</math> ou <math>OP2 = 1</math> ("ou" inclusif).</li> <li>• XOR: Le résultat est égal à 1 ssi exactement une <math>OP = 1</math> ("ou" exclusif).</li> </ul> Ces instructions permettent de forcer à 0 (AND), forcer à 1 (OR) ou d'inverser (XOR) des bits à des positions données dans une valeur.	<code>AND byte ptr [0x100], 0xFC</code> Force à 0 les deux bits de poids faible de l'octet situé à l'adresse 0x100  <code>OR AL, 0xF0</code> Force à 1 les quatre bits de poids fort du registre AL  <code>XOR RBX, 0xFF00</code> Complémente les bits 8 à 15 du registre RBX	Idem ADD.		CF et OF mis à 0.  ZF et SF mis à jour en fonction du résultat de l'opération.
<b>NOT</b>	<b>Inverser</b> tous les bits de cette opérande (càd de la remplacer par son complément à un).	Si le registre DX contient initialement l'instruction 0x5A, alors <code>NOT DX</code> Lui attribuera la valeur de 0xFFA5	Op.1 reg mem	Cette instruction est équivalente à <code>XOR DX, 0xFFFF</code>	Aucun.

