

Instrução	Operando 1	Operando 2	
<b>Tratamento de dados (existem diversos modos de endereçamento)</b>			
MOV	OP1	OP2	op1 ← op2
MOVZ	OP1	OP2	op1 ← op2 completando com zero
MOVSZ	OP1	OP2	op1 ← op2 completando com zero e mantendo o sinal
ADD	OP1	OP2	op1 ← op2 + op1
ADC	OP1	OP2	op1 ← op1 + op2 + CF
SUB	OP1	OP2	op1 ← op1 - op2
IMUL	OP1		Realiza a multiplicação considerando sinal e guardo o resultado no acumulador (AX, DX.AX, EDX.EAX).
IDIV	OP1		Realiza a divisão considerando sinal o cociente é armazenado em AL e o resto em AH (ou AX, DX ou EAX, EDX)
MUL	Op1		Realiza a multiplicação sem sinal e guardo o resultado no acumulador (AX, DX.AX, EDX.EAX).
DIV	OP1		Realiza a divisão sem sinal o cociente é armazenado em AL e o resto em AH (ou AX, DX ou EAX, EDX)
INC	OP1		op1 ← op1 + 1
DEC	OP1		op1 ← op1 - 1
NEG	OP1		Op1 recebe a negação em complemento de 2
<b>Comparações</b>			
CMP	OP1	OP2	Atualiza os FLAGS dependendo do resultado de OP1 - OP2 (subtração)
TEST	OP1	OP2	Atualiza os FLAGS dependendo do resultado de OP1 AND OP2
<b>Pulos sem considerar sinal</b>			
JE/JZ	Label		Pula se igual (ZF=1)
JNE/JNZ	Label		Pula se não igual (ZF=0)
JA/JNBE	Label		Pula se superior
JAE/JNB	Label		Pula se superior ou igual
JB/JNAE	Label		Pula se inferior
JBE/JNA	Label		Pula se inferior ou igual
JC	Label		Pula se CF = 1
JNC	Label		Pula se CF=0
<b>Pulos considerando sinal</b>			
JG/JNLE	Label		Pula se superior
JGE/JNL	Label		Pula se superior ou igual
JL/JNGE	Label		Pula se inferior
JLE/JNG	Label		Pula se inferior ou igual
<b>Desvio incondicional</b>			
JMP	Label		Pula
CALL	Label		Pula, empilhando endereço de retorno
RET			Pula retirando o endereço de retorno
INT	OP1		Faz a interrupção indicada pelo operando
<b>Laços</b>			
loop	Label		CX ← CX - 1, pula se CX != 0

Operações Binárias			
AND	OP1	OP2	$op1 \leftarrow op2 \text{ AND } op1$
OR	OP1	OP2	$op1 \leftarrow op2 \text{ OR } op1$
XOR	OP1	OP2	$op1 \leftarrow op2 \text{ XOR } op1$
NOT	OP1		OP1 recebe a negação bit a bit.
SHR	OP1	Valor	Shift para direita (quantidade de bits indicada por <b>valor</b> )
SHL	OP1	Valor	Shift para esquerda (quantidade de bits indicada por <b>valor</b> )
ROR	OP1	Valor	Rotação para direita (quantidade de bits indicada por <b>valor</b> )
ROL	OP1	Valor	Rotação para esquerda (quantidade de bits indicada por <b>valor</b> )
Pilha			
PUSH	OP1		$ESP \leftarrow ESP - (1, 2 \text{ ou } 4), [ESP] \leftarrow op1$
POP	OP2		$op1 \leftarrow [ESP], ESP \leftarrow ESP + (1, 2 \text{ ou } 4)$
ENTER			Cria frame da pilha
LEAVE			Restaura o frame anterior da pilha
Instruções ponto flutuante			
FINIT			Inicializa a FPU
FLD			Move real da memória para o topo da pilha de FPU
FILD			Move inteiro da memória para o topo da pilha de FPU
FINCSTP			Incrementa o ponteiro da FPU
FDECSTP			Decrementa o ponteiro da FPU
FADD/FSUB	OP1?		$st(0) \leftarrow st(0) \text{ +/- } st(1)$ . St(1) pode ser substituído por outra posição da pilha ou endereço de memória indicado mediante o OP1
FMUL/FDIV	OP1?		Similar a FADD/FSUB
FLDZ/FLDI/FLDPI			Carrega o topo da pilha com zero/um/pi