instruction	effect	description
leal s,d	d ← &s	load effective address
inc_ d	d ← d + 1	increment
dec_ d	d ← d - 1	decrement
neg_ d	d ← -d	negate
not_ d	d ← ~d	complement (bitwise)
add_ s,d	d ← d + s	add
sub_ s,d	d ← d - s	subtract
imul_ s,d	d ← d * s	multiply (32-bit)
xor_ s,d	d ← d ^ s	exclusive-or (bitwise)
or_ s,d	d ← d | s	or (bitwise)
and_ s,d	d ← d & s	and (bitwise)
sal_ k,d	d ← d << k	left shift
shl_ k,d	d ← d << k	left shift (same as sal_)
sar_ k,d	d ← d >> k	arithmetic right shift
shr_ k,d	d ← d >> k	logical right shift

type	gas form	operand value	addressing mode
immediate	$imm	imm	immediate
register	%r	R[r]	register
memory	imm	M[imm]	absolute
	(%r)	M[R[r]]	indirect
	imm(%r)	M[imm+R[r]]	base+displacement
	(%rb,%ri)	M[R[rb]+R[ri]]	indexed
	imm(%rb,%ri)	M[imm+R[rb]+R[ri]]	indexed
	(,%r,s)	M[R[r]*s]	scaled (by 1,2,4,8) indexed
	imm(,%r,s)	M[imm+R[r]*s]	scaled (by 1,2,4,8) indexed
	(%rb,%ri,s)	M[R[rb]+R[ri]*s]	scaled (by 1,2,4,8) indexed
	imm(%rb,%ri,s)	M[imm+R[rb]+R[ri]*s]	scaled (by 1,2,4,8) indexed

instruction		synonym	jump condition	description
jmp	label		1	direct jump
jmp	*operand		1	indirect jump
je	d	jz	zf	equal / zero
jne	d	jnz	~zf	not equal / not zero
js	d		sf	negative
jns	d		~sf	nonnegative
jg	d	jnle	~(sf ^ of) & ~zf	greater than (signed >)
jge	d	jnl	~(sf ^ of)	greater or equal (signed >=)
jl	d	jnge	sf ^ of	less than (signed <)
jle	d	jng	(sf ^ of) | zf	less or equal (signed <=)
ja	d	jnbe	~cf & ~zf	above (unsigned >)
jae	d	jnb	~cf	above or equal (unsigned >=)
jb	d	jnae	cf	below (unsigned <)
jbe	d	jna	cf | zf	below or equal (unsigned <=)