Instruction	Effect	Description
movb S,D	D ← S	Move byte
movw S,D		Move word
movl S,D		Move double word
movq S,D		Move quad word
movzbw S,R	R ← ZeroExt(S)	Move zero-extended byte to word
movzbl S,R		
movzwl S,R		
movzbq S,R		
movzwq S,R		
movsbw S,R	R ← SignExt(S)	Move sign-extended byte to word
movsbl S,R		
movswl S,R		
movsbq S,R		
movswq S,R		

Instruction	Effect	Description
pushq S	R[%rsp] ← R[%rsp] - 8; M[R[%rsp]] ← S	Push quad word
popq D	D ← M[R[%rsp]]; R[%rsp] ← R[%rsp] + 8	Pop quad word

Instruction	Effect	Description
leaq S,D	D ← &S	Load effective address
inc D	D ← D + 1	
dec D	D ← D - 1	
neg D	D ← -D	
not D	D ← ~D	
add S,D	D ← D + S	
sub S,D	D ← D - S	
imul S,D	D ← D * S	
xor S,D	D ← D ⊕ S	
sal k,D	D ← D << k	Equivalent to shl k,D
sar k,D	D ← D >> k	Arithmetic right shift
shr k,D	D ← D >> k	Logical right shift

Instruction	Effect	Description
imulq S	R[%rdx]:R[%rax] ← S * R[%rax]	Signed full multiply
mulq S	R[%rdx]:R[%rax] ← S * R[%rax]	Unsigned full multiply
cqto S	R[%rdx]:R[%rax] ← SignExt(R[%rax])	To oct word
idivq S	R[%rdx] ← R[%rdx]:R[%rax] mod S R[%rax] ← R[%rdx]:R[%rax] ÷ S	Signed divide
divq S		Unsigned divide

CF: Carry flag. Most recent operation generated carry from most significant bit. Detects overflow in unsigned operations.

ZF: Zero flag. Most recent operation yielded 0.

SF: Sign flag. Most recent operation yielded a negative value.

OF: Overflow flag. Most recent operation caused a two's complement overflow.

Instruction	Effect	Description
cmp S_1,S_2	$S_2 - S_1$	Set flags
test S_1,S_2	S_1 & S_2	Set flags

Instruction	Synonym	Condition	Description
jmp		1	Direct jump
je	jz	ZF	Equal/zero
jne	jnz	~ZF	Not equal/not zero
js		SF	Negative
jns		~SF	Nonnegative
jg	jnle	~(SF ^ OF) & ~ZF	Signed >
jge	jnl	~(SF ^ OF)	Signed >=
jl	jnge	SF ^ OF	Signed <
jle	jng	(SF ^ OF) | ZF	Signed <=
ja	jnbe	~CF & ~ZF	Unsigned >
jae	jnb	~CF	Unsigned >=
jb	jnae	CF	Unsigned <
jbe	jna	CF | ZF	Unsigned <=

There are analogous cmov (conditional move) and set (for accessing conditional codes) instructions. Other important operations include:

call: Calls a function at specified label.

ret: Return control to address on top of stack. Synonymous with retq.

rep: Or repz, can be ignored.

nop: Effectively does nothing.

leave: Move %rbp to %rsp, then pops %rbp.

Equivalent floating point operations for movement, conversion, comparison, arithmetic, etc.