

Exit Print View

# ORACLE<sup>®</sup> x86 Assembly Language Reference Manual

## Floating-Point Instructions

The floating point instructions operate on floating-point, integer, and binary coded decimal (BCD) operands.

### Data Transfer Instructions (Floating Point)

The data transfer instructions move floating-point, integer, and BCD values between memory and the floating point registers.

**Table 3-13 Data Transfer Instructions (Floating-Point)**

Solaris Mnemonic	Intel/AMD Mnemonic	Description	Notes
fbld	FBLD	load BCD	
fbstp	FBSTP	store BCD and pop	
fcmovb	FCMOVB	floating-point conditional move if below	
fcmovbe	FCMOVBE	floating-point conditional move if below or equal	
fcmove	FCMOVE	floating-point conditional move if equal	
fcmovnb	FCMOVNB	floating-point conditional move if not below	
fcmovnbe	FCMOVNBE	floating-point conditional move if not below or equal	
fcmovne	FCMOVNE	floating-point conditional move if not equal	
fcmovnu	FCMOVNU	floating-point conditional move if unordered	
fcmovu	FCMOVU	floating-point conditional move if unordered	
fild	FILD	load integer	
fist	FIST	store integer	
fistp	FISTP	store integer and pop	
fld	FLD	load floating-point value	
fst	FST	store floating-point value	
fstp	FSTP	store floating-point value and pop	
fxch	FXCH	exchange registers	

### Basic Arithmetic Instructions (Floating-Point)

The basic arithmetic instructions perform basic arithmetic operations on floating-point and integer operands.

**Table 3-14 Basic Arithmetic Instructions (Floating-Point)**

Solaris Mnemonic	Intel/AMD Mnemonic	Description	Notes
fabs	FABS	absolute value	
fadd	FADD	add floating-point	
faddp	FADDP	add floating-point and pop	
fchs	FCHS		

		change sign	
<code>fdiv</code>	<code>FDIV</code>	divide floating-point	
<code>fdivp</code>	<code>FDIVP</code>	divide floating-point and pop	
<code>fdivr</code>	<code>FDIVR</code>	divide floating-point reverse	
<code>fdivrp</code>	<code>FDIVRP</code>	divide floating-point reverse and pop	
<code>fiadd</code>	<code>FIADD</code>	add integer	
<code>fidiv</code>	<code>FIDIV</code>	divide integer	
<code>fidivr</code>	<code>FIDIVR</code>	divide integer reverse	
<code>fimul</code>	<code>FIMUL</code>	multiply integer	
<code>fisub</code>	<code>FISUB</code>	subtract integer	
<code>fisubr</code>	<code>FISUBR</code>	subtract integer reverse	
<code>fmul</code>	<code>FMUL</code>	multiply floating-point	
<code>fmulp</code>	<code>FMULP</code>	multiply floating-point and pop	
<code>fprem</code>	<code>FPREM</code>	partial remainder	
<code>fprem1</code>	<code>FPREM1</code>	IEEE partial remainder	
<code>frndint</code>	<code>FRNDINT</code>	round to integer	
<code>fscale</code>	<code>FSCALE</code>	scale by power of two	
<code>fsqrt</code>	<code>FSQRT</code>	square root	
<code>fsub</code>	<code>FSUB</code>	subtract floating-point	
<code>fsubp</code>	<code>FSUBP</code>	subtract floating-point and pop	
<code>fsubr</code>	<code>FSUBR</code>	subtract floating-point reverse	
<code>fsubrp</code>	<code>FSUBRP</code>	subtract floating-point reverse and pop	
<code>fextract</code>	<code>FEXTRACT</code>	extract exponent and significand	

## Comparison Instructions (Floating-Point)

The floating-point comparison instructions operate on floating-point or integer operands.

**Table 3-15 Comparison Instructions (Floating-Point)**

<b>Solaris Mnemonic</b>	<b>Intel/AMD Mnemonic</b>	<b>Description</b>	<b>Notes</b>
<code>fcom</code>	<code>FCOM</code>	compare floating-point	
<code>fcomi</code>	<code>FCOMI</code>	compare floating-point and set <code>%eflags</code>	
<code>fcomip</code>	<code>FCOMIP</code>	compare floating-point, set <code>%eflags</code> , and pop	
<code>fcomp</code>	<code>FCOMP</code>	compare floating-point and pop	
<code>fcompp</code>	<code>FCOMPP</code>	compare floating-point and pop twice	
<code>ficom</code>	<code>FICOM</code>	compare integer	
<code>ficomp</code>	<code>FICOMP</code>	compare integer and pop	
<code>ftst</code>	<code>FTST</code>	test floating-point (compare with 0.0)	
<code>fucom</code>	<code>FUCOM</code>	unordered compare floating-point	

fucomi	FUCOMI	unordered compare floating-point and set %eflags
fucomip	FUCOMIP	unordered compare floating-point, set %eflags, and pop
fucomp	FUCOMP	unordered compare floating-point and pop
fucompp	FUCOMPP	compare floating-point and pop twice
fxam	FXAM	examine floating-point

## Transcendental Instructions (Floating-Point)

The transcendental instructions perform trigonometric and logarithmic operations on floating-point operands.

**Table 3-16 Transcendental Instructions (Floating-Point)**

Solaris Mnemonic	Intel/AMD Mnemonic	Description	Notes
f2xm1	F2XM1	computes $2^x - 1$	
fcos	FCOS	cosine	
fpatan	FPATAN	partial arctangent	
fptan	FPTAN	partial tangent	
fsin	FSIN	sine	
fsincos	FSINCOS	sine and cosine	
fyl2x	FYL2X	computes $y * \log_2 x$	
fyl2xp1	FYL2XP1	computes $y * \log_2(x+1)$	

## Load Constants (Floating-Point) Instructions

The load constants instructions load common constants, such as  $\pi$ , into the floating-point registers.

**Table 3-17 Load Constants Instructions (Floating-Point)**

Solaris Mnemonic	Intel/AMD Mnemonic	Description	Notes
fld1	FLD1	load +1.0	
fldl2e	FLDL2E	load $\log_2 e$	
fldl2t	FLDL2T	load $\log_2 10$	
fldlg2	FLDLG2	load $\log_{10} 2$	
fldln2	FLDLN2	load $\log_e 2$	
fldpi	FLDPI	load $\pi$	
fldz	FLDZ	load +0.0	

## Control Instructions (Floating-Point)

The floating-point control instructions operate on the floating-point register stack and save and restore the floating-point state.

**Table 3-18 Control Instructions (Floating-Point)**

Solaris Mnemonic	Intel/AMD Mnemonic	Description	Notes
fclex	FCLEX	clear floating-point exception flags after	

		checking for error conditions	
fdecstp	FDECSTP	decrement floating-point register stack pointer	
ffree	FFREE	free floating-point register	
fincstp	FINCSTP	increment floating-point register stack pointer	
finit	FINIT	initialize floating-point unit after checking error conditions	
fldcw	FLDCW	load floating-point unit control word	
fldenv	FLDENV	load floating-point unit environment	
fnclex	FNCLEX	clear floating-point exception flags without checking for error conditions	
fninit	FNINIT	initialize floating-point unit without checking error conditions	
fnop	FNOP	floating-point no operation	
fnsave	FNSAVE	save floating-point unit state without checking error conditions	
fnstcw	FNSTCW	store floating-point unit control word without checking error conditions	
fnstenv	FNSTENV	store floating-point unit environment without checking error conditions	
fnstsw	FNSTSW	store floating-point unit status word without checking error conditions	
frstor	FRSTOR	restore floating-point unit state	
fsave	FSAVE	save floating-point unit state after checking error conditions	
fstcw	FSTCW	store floating-point unit control word after checking error conditions	
fstenv	FSTENV	store floating-point unit environment after checking error conditions	
fstsw	FSTSW	store floating-point unit status word after checking error conditions	
fwait	FWAIT	wait for floating-point unit	
wait	WAIT	wait for floating-point unit	