x86 and amd64 instruction reference

Derived from the December 2023 version of the Intel® 64 and IA-32 Architectures Software Developer's Manual. Last updated 2024-02-18.

THIS REFERENCE IS NOT PERFECT. It's been mechanically separated into distinct files by a dumb script. It may be enough to replace the official documentation on your weekend reverse engineering project, but for anything where money is at stake, go get the official and freely available documentation.

Core Instructions

Mnemonic	Summary
AAA	ASCII Adjust After Addition
AAD	ASCII Adjust AX Before Division
AAM	ASCII Adjust AX After Multiply
AAS	ASCII Adjust AL After Subtraction
ADC	Add With Carry
ADCX	Unsigned Integer Addition of Two Operands With Carry Flag
ADD	Add
ADDPD	Add Packed Double Precision Floating-Point Values
ADDPS	Add Packed Single Precision Floating-Point Values
ADDSD	Add Scalar Double Precision Floating-Point Values
ADDSS	Add Scalar Single Precision Floating-Point Values
ADDSUBPD	Packed Double Precision Floating-Point Add/Subtract
ADDSUBPS	Packed Single Precision Floating-Point Add/Subtract
ADOX	Unsigned Integer Addition of Two Operands With Overflow Flag
AESDEC	Perform One Round of an AES Decryption Flow
AESDEC128KL	Perform Ten Rounds of AES Decryption Flow With Key Locker Using 128-BitKey
AESDEC256KL	Perform 14 Rounds of AES Decryption Flow With Key Locker Using 256-Bit Key
AESDECLAST	Perform Last Round of an AES Decryption Flow
AESDECWIDE128KL	Perform Ten Rounds of AES Decryption Flow With Key Locker on 8 BlocksUsing 128-Bit Key
AESDECWIDE256KL	Perform 14 Rounds of AES Decryption Flow With Key Locker on 8 BlocksUsing 256-Bit Key
AESENC	Perform One Round of an AES Encryption Flow
AESENC128KL	Perform Ten Rounds of AES Encryption Flow With Key Locker Using 128-Bit Key
AESENC256KL	Perform 14 Rounds of AES Encryption Flow With Key Locker Using 256-Bit Key
AESENCLAST	Perform Last Round of an AES Encryption Flow

AESENCWIDE128KL	Perform Ten Rounds of AES Encryption Flow With Key Locker on 8 BlocksUsing 128-Bit Key
AESENCWIDE256KL	Perform 14 Rounds of AES Encryption Flow With Key Locker on 8 BlocksUsing 256-Bit Key
AESIMC	Perform the AES InvMixColumn Transformation
AESKEYGENASSIST	AES Round Key Generation Assist
AND	Logical AND
ANDN	Logical AND NOT
ANDNPD	Bitwise Logical AND NOT of Packed Double Precision Floating-Point Values
ANDNPS	Bitwise Logical AND NOT of Packed Single Precision Floating-Point Values
ANDPD	Bitwise Logical AND of Packed Double Precision Floating-Point Values
ANDPS	Bitwise Logical AND of Packed Single Precision Floating-Point Values
ARPL	Adjust RPL Field of Segment Selector
BEXTR	Bit Field Extract
BLENDPD	Blend Packed Double Precision Floating-Point Values
BLENDPS	Blend Packed Single Precision Floating-Point Values
BLENDVPD	Variable Blend Packed Double Precision Floating-Point Values
BLENDVPS	Variable Blend Packed Single Precision Floating-Point Values
BLSI	Extract Lowest Set Isolated Bit
BLSMSK	Get Mask Up to Lowest Set Bit
BLSR	Reset Lowest Set Bit
BNDCL	Check Lower Bound
BNDCN	Check Upper Bound
BNDCU	Check Upper Bound
BNDLDX	Load Extended Bounds Using Address Translation
BNDMK	Make Bounds
BNDMOV	Move Bounds
BNDSTX	Store Extended Bounds Using Address Translation
BOUND	Check Array Index Against Bounds
BSF	Bit Scan Forward
BSR	Bit Scan Reverse
BSWAP	Byte Swap
BT	Bit Test
BTC	Bit Test and Complement
BTR	Bit Test and Reset
BTS	Bit Test and Set
BZHI	Zero High Bits Starting with Specified Bit Position
CALL	Call Procedure
CBW	Convert Byte to Word/Convert Word to Doubleword/Convert Doubleword toQuadword

CDQ	Convert Word to Doubleword/Convert Doubleword to Quadword
CDQE	Convert Byte to Word/Convert Word to Doubleword/Convert Doubleword toQuadword
CLAC	Clear AC Flag in EFLAGS Register
CLC	Clear Carry Flag
CLD	Clear Direction Flag
CLDEMOTE	Cache Line Demote
CLFLUSH	Flush Cache Line
CLFLUSHOPT	Flush Cache Line Optimized
CLI	Clear Interrupt Flag
CLRSSBSY	Clear Busy Flag in a Supervisor Shadow Stack Token
CLTS	Clear Task-Switched Flag in CR0
CLUI	Clear User Interrupt Flag
CLWB	Cache Line Write Back
CMC	Complement Carry Flag
CMOVcc	Conditional Move
CMP	Compare Two Operands
CMPPD	Compare Packed Double Precision Floating-Point Values
CMPPS	Compare Packed Single Precision Floating-Point Values
CMPS	Compare String Operands
CMPSB	Compare String Operands
CMPSD	Compare String Operands
CMPSD (1)	Compare Scalar Double Precision Floating-Point Value
CMPSQ	Compare String Operands
CMPSS	Compare Scalar Single Precision Floating-Point Value
CMPSW	Compare String Operands
CMPXCHG	Compare and Exchange
CMPXCHG16B	Compare and Exchange Bytes
CMPXCHG8B	Compare and Exchange Bytes
COMISD	Compare Scalar Ordered Double Precision Floating-Point Values and Set EFLAGS
COMISS	Compare Scalar Ordered Single Precision Floating-Point Values and Set EFLAGS
CPUID	CPU Identification
CQO	Convert Word to Doubleword/Convert Doubleword to Quadword
CRC32	Accumulate CRC32 Value
CVTDQ2PD	Convert Packed Doubleword Integers to Packed Double Precision Floating-PointValues
CVTDQ2PS	Convert Packed Doubleword Integers to Packed Single Precision Floating-PointValues

CVTPD2DQ	Convert Packed Double Precision Floating-Point Values to Packed DoublewordIntegers
CVTPD2PI	Convert Packed Double Precision Floating-Point Values to Packed Dword Integers
CVTPD2PS	Convert Packed Double Precision Floating-Point Values to Packed Single PrecisionFloating-Point Values
CVTPI2PD	Convert Packed Dword Integers to Packed Double Precision Floating-Point Values
CVTPI2PS	Convert Packed Dword Integers to Packed Single Precision Floating-Point Values
CVTPS2DQ	Convert Packed Single Precision Floating-Point Values to Packed SignedDoubleword Integer Values
CVTPS2PD	Convert Packed Single Precision Floating-Point Values to Packed Double PrecisionFloating-Point Values
CVTPS2PI	Convert Packed Single Precision Floating-Point Values to Packed Dword Integers
CVTSD2SI	Convert Scalar Double Precision Floating-Point Value to Doubleword Integer
CVTSD2SS	Convert Scalar Double Precision Floating-Point Value to Scalar Single PrecisionFloating-Point Value
CVTSI2SD	Convert Doubleword Integer to Scalar Double Precision Floating-Point Value
CVTSI2SS	Convert Doubleword Integer to Scalar Single Precision Floating-Point Value
CVTSS2SD	Convert Scalar Single Precision Floating-Point Value to Scalar Double PrecisionFloating-Point Value
CVTSS2SI	Convert Scalar Single Precision Floating-Point Value to Doubleword Integer
CVTTPD2DQ	Convert with Truncation Packed Double Precision Floating-Point Values toPacked Doubleword Integers
CVTTPD2PI	Convert With Truncation Packed Double Precision Floating-Point Values to PackedDword Integers
CVTTPS2DQ	Convert With Truncation Packed Single Precision Floating-Point Values to PackedSigned Doubleword Integer Values
CVTTPS2PI	Convert With Truncation Packed Single Precision Floating-Point Values to PackedDword Integers
CVTTSD2SI	Convert With Truncation Scalar Double Precision Floating-Point Value to SignedInteger
CVTTSS2SI	Convert With Truncation Scalar Single Precision Floating-Point Value to Integer
CWD	Convert Word to Doubleword/Convert Doubleword to Quadword
CWDE	Convert Byte to Word/Convert Word to Doubleword/Convert Doubleword toQuadword
DAA	Decimal Adjust AL After Addition
DAS	Decimal Adjust AL After Subtraction
DEC	Decrement by 1
DIV	Unsigned Divide
DIVPD	Divide Packed Double Precision Floating-Point Values

DIVPS	Divide Packed Single Precision Floating-Point Values
DIVSD	Divide Scalar Double Precision Floating-Point Value
DIVSS	Divide Scalar Single Precision Floating-Point Values
DPPD	Dot Product of Packed Double Precision Floating-Point Values
DPPS	Dot Product of Packed Single Precision Floating-Point Values
EMMS	Empty MMX Technology State
ENCODEKEY128	Encode 128-Bit Key With Key Locker
ENCODEKEY256	Encode 256-Bit Key With Key Locker
ENDBR32	Terminate an Indirect Branch in 32-bit and Compatibility Mode
ENDBR64	Terminate an Indirect Branch in 64-bit Mode
ENQCMD	Enqueue Command
ENQCMDS	Enqueue Command Supervisor
ENTER	Make Stack Frame for Procedure Parameters
EXTRACTPS	Extract Packed Floating-Point Values
F2XM1	Compute 2x–1
FABS	Absolute Value
FADD	Add
FADDP	Add
FBLD	Load Binary Coded Decimal
FBSTP	Store BCD Integer and Pop
FCHS	Change Sign
FCLEX	Clear Exceptions
FCMOVcc	Floating-Point Conditional Move
FCOM	Compare Floating-Point Values
FCOMI	Compare Floating-Point Values and Set EFLAGS
FCOMIP	Compare Floating-Point Values and Set EFLAGS
FCOMP	Compare Floating-Point Values
FCOMPP	Compare Floating-Point Values
FCOS	Cosine
FDECSTP	Decrement Stack-Top Pointer
FDIV	Divide
FDIVP	Divide
FDIVR	Reverse Divide
FDIVRP	Reverse Divide
FFREE	Free Floating-Point Register
FIADD	Add
FICOM	Compare Integer
FICOMP	Compare Integer
FIDIV	Divide
FIDIVR	Reverse Divide

FILD	Load Integer
FIMUL	Multiply
FINCSTP	Increment Stack-Top Pointer
FINIT	Initialize Floating-Point Unit
FIST	Store Integer
FISTP	Store Integer
FISTTP	Store Integer With Truncation
FISUB	Subtract
FISUBR	Reverse Subtract
FLD	Load Floating-Point Value
FLD1	Load Constant
FLDCW	Load x87 FPU Control Word
FLDENV	Load x87 FPU Environment
FLDL2E	Load Constant
FLDL2T	Load Constant
FLDLG2	Load Constant
FLDLN2	Load Constant
FLDPI	Load Constant
FLDZ	Load Constant
FMUL	Multiply
FMULP	Multiply
FNCLEX	Clear Exceptions
FNINIT	Initialize Floating-Point Unit
FNOP	No Operation
FNSAVE	Store x87 FPU State
FNSTCW	Store x87 FPU Control Word
FNSTENV	Store x87 FPU Environment
FNSTSW	Store x87 FPU Status Word
FPATAN	Partial Arctangent
FPREM	Partial Remainder
FPREM1	Partial Remainder
FPTAN	Partial Tangent
FRNDINT	Round to Integer
FRSTOR	Restore x87 FPU State
FSAVE	Store x87 FPU State
FSCALE	Scale
FSIN	Sine
FSINCOS	Sine and Cosine
FSQRT	Square Root
FST	Store Floating-Point Value

FSTCW	Store x87 FPU Control Word
FSTENV	Store x87 FPU Environment
FSTP	Store Floating-Point Value
FSTSW	Store x87 FPU Status Word
FSUB	Subtract
FSUBP	Subtract
FSUBR	Reverse Subtract
FSUBRP	Reverse Subtract
FTST	TEST
FUCOM	Unordered Compare Floating-Point Values
FUCOMI	Compare Floating-Point Values and Set EFLAGS
FUCOMIP	Compare Floating-Point Values and Set EFLAGS
FUCOMP	Unordered Compare Floating-Point Values
FUCOMPP	Unordered Compare Floating-Point Values
FWAIT	Wait
FXAM	Examine Floating-Point
FXCH	Exchange Register Contents
FXRSTOR	Restore x87 FPU, MMX, XMM, and MXCSR State
FXSAVE	Save x87 FPU, MMX Technology, and SSE State
FXTRACT	Extract Exponent and Significand
FYL2X	Compute y * log2x
FYL2XP1	Compute y * log2(x +1)
GF2P8AFFINEINVQB	Galois Field Affine Transformation Inverse
GF2P8AFFINEQB	Galois Field Affine Transformation
GF2P8MULB	Galois Field Multiply Bytes
HADDPD	Packed Double Precision Floating-Point Horizontal Add
HADDPS	Packed Single Precision Floating-Point Horizontal Add
HLT	Halt
HRESET	History Reset
HSUBPD	Packed Double Precision Floating-Point Horizontal Subtract
HSUBPS	Packed Single Precision Floating-Point Horizontal Subtract
IDIV	Signed Divide
IMUL	Signed Multiply
IN	Input From Port
INC	Increment by 1
INCSSPD	Increment Shadow Stack Pointer
INCSSPQ	Increment Shadow Stack Pointer
INS	Input from Port to String
INSB	Input from Port to String

INSD	Input from Port to String
INSERTPS	Insert Scalar Single Precision Floating-Point Value
INSW	Input from Port to String
INT n	Call to Interrupt Procedure
INT1	Call to Interrupt Procedure
INT3	Call to Interrupt Procedure
INTO	Call to Interrupt Procedure
INVD	Invalidate Internal Caches
INVLPG	Invalidate TLB Entries
INVPCID	Invalidate Process-Context Identifier
IRET	Interrupt Return
IRETD	Interrupt Return
IRETQ	Interrupt Return
JMP	Jump
Jcc	Jump if Condition Is Met
KADDB	ADD Two Masks
KADDD	ADD Two Masks
KADDQ	ADD Two Masks
KADDW	ADD Two Masks
KANDB	Bitwise Logical AND Masks
KANDD	Bitwise Logical AND Masks
KANDNB	Bitwise Logical AND NOT Masks
KANDND	Bitwise Logical AND NOT Masks
KANDNQ	Bitwise Logical AND NOT Masks
KANDNW	Bitwise Logical AND NOT Masks
KANDQ	Bitwise Logical AND Masks
KANDW	Bitwise Logical AND Masks
KMOVB	Move From and to Mask Registers
KMOVD	Move From and to Mask Registers
KMOVQ	Move From and to Mask Registers
KMOVW	Move From and to Mask Registers
KNOTB	NOT Mask Register
KNOTD	NOT Mask Register
KNOTQ	NOT Mask Register
KNOTW	NOT Mask Register
KORB	Bitwise Logical OR Masks
KORD	Bitwise Logical OR Masks
KORQ	Bitwise Logical OR Masks
KORTESTB	OR Masks and Set Flags
KORTESTD	OR Masks and Set Flags

KORTESTQ	OR Masks and Set Flags
KORTESTW	OR Masks and Set Flags
KORW	Bitwise Logical OR Masks
KSHIFTLB	Shift Left Mask Registers
KSHIFTLD	Shift Left Mask Registers
KSHIFTLQ	Shift Left Mask Registers
KSHIFTLW	Shift Left Mask Registers
KSHIFTRB	Shift Right Mask Registers
KSHIFTRD	Shift Right Mask Registers
KSHIFTRQ	Shift Right Mask Registers
KSHIFTRW	Shift Right Mask Registers
KTESTB	Packed Bit Test Masks and Set Flags
KTESTD	Packed Bit Test Masks and Set Flags
KTESTQ	Packed Bit Test Masks and Set Flags
KTESTW	Packed Bit Test Masks and Set Flags
KUNPCKBW	Unpack for Mask Registers
KUNPCKDQ	Unpack for Mask Registers
KUNPCKWD	Unpack for Mask Registers
KXNORB	Bitwise Logical XNOR Masks
KXNORD	Bitwise Logical XNOR Masks
KXNORQ	Bitwise Logical XNOR Masks
KXNORW	Bitwise Logical XNOR Masks
KXORB	Bitwise Logical XOR Masks
KXORD	Bitwise Logical XOR Masks
KXORQ	Bitwise Logical XOR Masks
KXORW	Bitwise Logical XOR Masks
LAHF	Load Status Flags Into AH Register
LAR	Load Access Rights Byte
LDDQU	Load Unaligned Integer 128 Bits
LDMXCSR	Load MXCSR Register
LDS	Load Far Pointer
LDTILECFG	Load Tile Configuration
LEA	Load Effective Address
LEAVE	High Level Procedure Exit
LES	Load Far Pointer
LFENCE	Load Fence
LFS	Load Far Pointer
LGDT	Load Global/Interrupt Descriptor Table Register
LGS	Load Far Pointer
LIDT	Load Global/Interrupt Descriptor Table Register

LLDT	Load Local Descriptor Table Register
LMSW	Load Machine Status Word
LOADIWKEY	Load Internal Wrapping Key With Key Locker
LOCK	Assert LOCK# Signal Prefix
LODS	Load String
LODSB	Load String
LODSD	Load String
LODSQ	Load String
LODSW	Load String
LOOP	Loop According to ECX Counter
LOOPcc	Loop According to ECX Counter
LSL	Load Segment Limit
LSS	Load Far Pointer
LTR	Load Task Register
LZCNT	Count the Number of Leading Zero Bits
MASKMOVDQU	Store Selected Bytes of Double Quadword
MASKMOVQ	Store Selected Bytes of Quadword
MAXPD	Maximum of Packed Double Precision Floating-Point Values
MAXPS	Maximum of Packed Single Precision Floating-Point Values
MAXSD	Return Maximum Scalar Double Precision Floating-Point Value
MAXSS	Return Maximum Scalar Single Precision Floating-Point Value
MFENCE	Memory Fence
MINPD	Minimum of Packed Double Precision Floating-Point Values
MINPS	Minimum of Packed Single Precision Floating-Point Values
MINSD	Return Minimum Scalar Double Precision Floating-Point Value
MINSS	Return Minimum Scalar Single Precision Floating-Point Value
MONITOR	Set Up Monitor Address
MOV	Move
MOV (1)	Move to/from Control Registers
MOV (2)	Move to/from Debug Registers
MOVAPD	Move Aligned Packed Double Precision Floating-Point Values
MOVAPS	Move Aligned Packed Single Precision Floating-Point Values
MOVBE	Move Data After Swapping Bytes
MOVD	Move Doubleword/Move Quadword
MOVDDUP	Replicate Double Precision Floating-Point Values
MOVDIR64B	Move 64 Bytes as Direct Store
MOVDIRI	Move Doubleword as Direct Store
MOVDQ2Q	Move Quadword from XMM to MMX Technology Register
MOVDQA	Move Aligned Packed Integer Values
MOVDQU	Move Unaligned Packed Integer Values

MOVHLPS	Move Packed Single Precision Floating-Point Values High to Low
MOVHPD	Move High Packed Double Precision Floating-Point Value
MOVHPS	Move High Packed Single Precision Floating-Point Values
MOVLHPS	Move Packed Single Precision Floating-Point Values Low to High
MOVLPD	Move Low Packed Double Precision Floating-Point Value
MOVLPS	Move Low Packed Single Precision Floating-Point Values
MOVMSKPD	Extract Packed Double Precision Floating-Point Sign Mask
MOVMSKPS	Extract Packed Single Precision Floating-Point Sign Mask
MOVNTDQ	Store Packed Integers Using Non-Temporal Hint
MOVNTDQA	Load Double Quadword Non-Temporal Aligned Hint
MOVNTI	Store Doubleword Using Non-Temporal Hint
MOVNTPD	Store Packed Double Precision Floating-Point Values Using Non-Temporal Hint
MOVNTPS	Store Packed Single Precision Floating-Point Values Using Non-Temporal Hint
MOVNTQ	Store of Quadword Using Non-Temporal Hint
MOVQ	Move Doubleword/Move Quadword
MOVQ (1)	Move Quadword
MOVQ2DQ	Move Quadword from MMX Technology to XMM Register
MOVS	Move Data From String to String
MOVSB	Move Data From String to String
MOVSD	Move Data From String to String
MOVSD (1)	Move or Merge Scalar Double Precision Floating-Point Value
MOVSHDUP	Replicate Single Precision Floating-Point Values
MOVSLDUP	Replicate Single Precision Floating-Point Values
MOVSQ	Move Data From String to String
MOVSS	Move or Merge Scalar Single Precision Floating-Point Value
MOVSW	Move Data From String to String
MOVSX	Move With Sign-Extension
MOVSXD	Move With Sign-Extension
MOVUPD	Move Unaligned Packed Double Precision Floating-Point Values
MOVUPS	Move Unaligned Packed Single Precision Floating-Point Values
MOVZX	Move With Zero-Extend
MPSADBW	Compute Multiple Packed Sums of Absolute Difference
MUL	Unsigned Multiply
MULPD	Multiply Packed Double Precision Floating-Point Values
MULPS	Multiply Packed Single Precision Floating-Point Values
MULSD	Multiply Scalar Double Precision Floating-Point Value
MULSS	Multiply Scalar Single Precision Floating-Point Values
MULX	Unsigned Multiply Without Affecting Flags

MWAIT	Monitor Wait
NEG	Two's Complement Negation
NOP	No Operation
NOT	One's Complement Negation
OR	Logical Inclusive OR
ORPD	Bitwise Logical OR of Packed Double Precision Floating-Point Values
ORPS	Bitwise Logical OR of Packed Single Precision Floating-Point Values
OUT	Output to Port
OUTS	Output String to Port
OUTSB	Output String to Port
OUTSD	Output String to Port
OUTSW	Output String to Port
PABSB	Packed Absolute Value
PABSD	Packed Absolute Value
PABSQ	Packed Absolute Value
PABSW	Packed Absolute Value
PACKSSDW	Pack With Signed Saturation
PACKSSWB	Pack With Signed Saturation
PACKUSDW	Pack With Unsigned Saturation
PACKUSWB	Pack With Unsigned Saturation
PADDB	Add Packed Integers
PADDD	Add Packed Integers
PADDQ	Add Packed Integers
PADDSB	Add Packed Signed Integers with Signed Saturation
PADDSW	Add Packed Signed Integers with Signed Saturation
PADDUSB	Add Packed Unsigned Integers With Unsigned Saturation
PADDUSW	Add Packed Unsigned Integers With Unsigned Saturation
PADDW	Add Packed Integers
PALIGNR	Packed Align Right
PAND	Logical AND
PANDN	Logical AND NOT
PAUSE	Spin Loop Hint
PAVGB	Average Packed Integers
PAVGW	Average Packed Integers
PBLENDVB	Variable Blend Packed Bytes
PBLENDW	Blend Packed Words
PCLMULQDQ	Carry-Less Multiplication Quadword
PCMPEQB	Compare Packed Data for Equal
PCMPEQD	Compare Packed Data for Equal
PCMPEQQ	Compare Packed Qword Data for Equal

PCMPEQW	Compare Packed Data for Equal
PCMPESTRI	Packed Compare Explicit Length Strings, Return Index
PCMPESTRM	Packed Compare Explicit Length Strings, Return Mask
PCMPGTB	Compare Packed Signed Integers for Greater Than
PCMPGTD	Compare Packed Signed Integers for Greater Than
PCMPGTQ	Compare Packed Data for Greater Than
PCMPGTW	Compare Packed Signed Integers for Greater Than
PCMPISTRI	Packed Compare Implicit Length Strings, Return Index
PCMPISTRM	Packed Compare Implicit Length Strings, Return Mask
PCONFIG	Platform Configuration
PDEP	Parallel Bits Deposit
PEXT	Parallel Bits Extract
PEXTRB	Extract Byte/Dword/Qword
PEXTRD	Extract Byte/Dword/Qword
PEXTRQ	Extract Byte/Dword/Qword
PEXTRW	Extract Word
PHADDD	Packed Horizontal Add
PHADDSW	Packed Horizontal Add and Saturate
PHADDW	Packed Horizontal Add
PHMINPOSUW	Packed Horizontal Word Minimum
PHSUBD	Packed Horizontal Subtract
PHSUBSW	Packed Horizontal Subtract and Saturate
PHSUBW	Packed Horizontal Subtract
PINSRB	Insert Byte/Dword/Qword
PINSRD	Insert Byte/Dword/Qword
PINSRQ	Insert Byte/Dword/Qword
PINSRW	Insert Word
PMADDUBSW	Multiply and Add Packed Signed and Unsigned Bytes
PMADDWD	Multiply and Add Packed Integers
PMAXSB	Maximum of Packed Signed Integers
PMAXSD	Maximum of Packed Signed Integers
PMAXSQ	Maximum of Packed Signed Integers
PMAXSW	Maximum of Packed Signed Integers
PMAXUB	Maximum of Packed Unsigned Integers
PMAXUD	Maximum of Packed Unsigned Integers
PMAXUQ	Maximum of Packed Unsigned Integers
PMAXUW	Maximum of Packed Unsigned Integers
PMINSB	Minimum of Packed Signed Integers
PMINSD	Minimum of Packed Signed Integers
PMINSQ	Minimum of Packed Signed Integers

PMINSW	Minimum of Packed Signed Integers
PMINUB	Minimum of Packed Unsigned Integers
PMINUD	Minimum of Packed Unsigned Integers
PMINUQ	Minimum of Packed Unsigned Integers
PMINUW	Minimum of Packed Unsigned Integers
PMOVMSKB	Move Byte Mask
PMOVSX	Packed Move With Sign Extend
PMOVZX	Packed Move With Zero Extend
PMULDQ	Multiply Packed Doubleword Integers
PMULHRSW	Packed Multiply High With Round and Scale
PMULHUW	Multiply Packed Unsigned Integers and Store High Result
PMULHW	Multiply Packed Signed Integers and Store High Result
PMULLD	Multiply Packed Integers and Store Low Result
PMULLQ	Multiply Packed Integers and Store Low Result
PMULLW	Multiply Packed Signed Integers and Store Low Result
PMULUDQ	Multiply Packed Unsigned Doubleword Integers
POP	Pop a Value From the Stack
POPA	Pop All General-Purpose Registers
POPAD	Pop All General-Purpose Registers
POPCNT	Return the Count of Number of Bits Set to 1
POPF	Pop Stack Into EFLAGS Register
POPFD	Pop Stack Into EFLAGS Register
POPFQ	Pop Stack Into EFLAGS Register
POR	Bitwise Logical OR
PREFETCHW	Prefetch Data Into Caches in Anticipation of a Write
PREFETCHh	Prefetch Data Into Caches
PSADBW	Compute Sum of Absolute Differences
PSHUFB	Packed Shuffle Bytes
PSHUFD	Shuffle Packed Doublewords
PSHUFHW	Shuffle Packed High Words
PSHUFLW	Shuffle Packed Low Words
PSHUFW	Shuffle Packed Words
PSIGNB	Packed SIGN
PSIGND	Packed SIGN
PSIGNW	Packed SIGN
PSLLD	Shift Packed Data Left Logical
PSLLDQ	Shift Double Quadword Left Logical
PSLLQ	Shift Packed Data Left Logical
PSLLW	Shift Packed Data Left Logical
PSRAD	Shift Packed Data Right Arithmetic

PSRAQ	Shift Packed Data Right Arithmetic
PSRAW	Shift Packed Data Right Arithmetic
PSRLD	Shift Packed Data Right Logical
PSRLDQ	Shift Double Quadword Right Logical
PSRLQ	Shift Packed Data Right Logical
PSRLW	Shift Packed Data Right Logical
PSUBB	Subtract Packed Integers
PSUBD	Subtract Packed Integers
PSUBQ	Subtract Packed Quadword Integers
PSUBSB	Subtract Packed Signed Integers With Signed Saturation
PSUBSW	Subtract Packed Signed Integers With Signed Saturation
PSUBUSB	Subtract Packed Unsigned Integers With Unsigned Saturation
PSUBUSW	Subtract Packed Unsigned Integers With Unsigned Saturation
PSUBW	Subtract Packed Integers
PTEST	Logical Compare
PTWRITE	Write Data to a Processor Trace Packet
PUNPCKHBW	Unpack High Data
PUNPCKHDQ	Unpack High Data
PUNPCKHQDQ	Unpack High Data
PUNPCKHWD	Unpack High Data
PUNPCKLBW	Unpack Low Data
PUNPCKLDQ	Unpack Low Data
PUNPCKLQDQ	Unpack Low Data
PUNPCKLWD	Unpack Low Data
PUSH	Push Word, Doubleword, or Quadword Onto the Stack
PUSHA	Push All General-Purpose Registers
PUSHAD	Push All General-Purpose Registers
PUSHF	Push EFLAGS Register Onto the Stack
PUSHFD	Push EFLAGS Register Onto the Stack
PUSHFQ	Push EFLAGS Register Onto the Stack
PXOR	Logical Exclusive OR
RCL	Rotate
RCPPS	Compute Reciprocals of Packed Single Precision Floating-Point Values
RCPSS	Compute Reciprocal of Scalar Single Precision Floating-Point Values
RCR	Rotate
RDFSBASE	Read FS/GS Segment Base
RDGSBASE	Read FS/GS Segment Base
RDMSR	Read From Model Specific Register
RDPID	Read Processor ID
RDPKRU	Read Protection Key Rights for User Pages

RDPMC	Read Performance-Monitoring Counters
RDRAND	Read Random Number
RDSEED	Read Random SEED
RDSSPD	Read Shadow Stack Pointer
RDSSPQ	Read Shadow Stack Pointer
RDTSC	Read Time-Stamp Counter
RDTSCP	Read Time-Stamp Counter and Processor ID
REP	Repeat String Operation Prefix
REPE	Repeat String Operation Prefix
REPNE	Repeat String Operation Prefix
REPNZ	Repeat String Operation Prefix
REPZ	Repeat String Operation Prefix
RET	Return From Procedure
ROL	Rotate
ROR	Rotate
RORX	Rotate Right Logical Without Affecting Flags
ROUNDPD	Round Packed Double Precision Floating-Point Values
ROUNDPS	Round Packed Single Precision Floating-Point Values
ROUNDSD	Round Scalar Double Precision Floating-Point Values
ROUNDSS	Round Scalar Single Precision Floating-Point Values
RSM	Resume From System Management Mode
RSQRTPS	Compute Reciprocals of Square Roots of Packed Single Precision Floating-PointValues
RSQRTSS	Compute Reciprocal of Square Root of Scalar Single Precision Floating-Point Value
RSTORSSP	Restore Saved Shadow Stack Pointer
SAHF	Store AH Into Flags
SAL	Shift
SAR	Shift
SARX	Shift Without Affecting Flags
SAVEPREVSSP	Save Previous Shadow Stack Pointer
SBB	Integer Subtraction With Borrow
SCAS	Scan String
SCASB	Scan String
SCASD	Scan String
SCASW	Scan String
SENDUIPI	Send User Interprocessor Interrupt
SERIALIZE	Serialize Instruction Execution
SETSSBSY	Mark Shadow Stack Busy
SETcc	Set Byte on Condition

SFENCE	Store Fence
SGDT	Store Global Descriptor Table Register
SHA1MSG1	Perform an Intermediate Calculation for the Next Four SHA1 Message Dwords
SHA1MSG2	Perform a Final Calculation for the Next Four SHA1 Message Dwords
SHA1NEXTE	Calculate SHA1 State Variable E After Four Rounds
SHA1RNDS4	Perform Four Rounds of SHA1 Operation
SHA256MSG1	Perform an Intermediate Calculation for the Next Four SHA256 MessageDwords
SHA256MSG2	Perform a Final Calculation for the Next Four SHA256 Message Dwords
SHA256RNDS2	Perform Two Rounds of SHA256 Operation
SHL	Shift
SHLD	Double Precision Shift Left
SHLX	Shift Without Affecting Flags
SHR	Shift
SHRD	Double Precision Shift Right
SHRX	Shift Without Affecting Flags
SHUFPD	Packed Interleave Shuffle of Pairs of Double Precision Floating-Point Values
SHUFPS	Packed Interleave Shuffle of Quadruplets of Single Precision Floating-Point Values
SIDT	Store Interrupt Descriptor Table Register
SLDT	Store Local Descriptor Table Register
SMSW	Store Machine Status Word
SQRTPD	Square Root of Double Precision Floating-Point Values
SQRTPS	Square Root of Single Precision Floating-Point Values
SQRTSD	Compute Square Root of Scalar Double Precision Floating-Point Value
SQRTSS	Compute Square Root of Scalar Single Precision Value
STAC	Set AC Flag in EFLAGS Register
STC	Set Carry Flag
STD	Set Direction Flag
STI	Set Interrupt Flag
STMXCSR	Store MXCSR Register State
STOS	Store String
STOSB	Store String
STOSD	Store String
STOSQ	Store String
STOSW	Store String
STR	Store Task Register
STTILECFG	Store Tile Configuration
STUI	Set User Interrupt Flag
SUB	Subtract

SUBPD	Subtract Packed Double Precision Floating-Point Values
SUBPS	Subtract Packed Single Precision Floating-Point Values
SUBSD	Subtract Scalar Double Precision Floating-Point Value
SUBSS	Subtract Scalar Single Precision Floating-Point Value
SWAPGS	Swap GS Base Register
SYSCALL	Fast System Call
SYSENTER	Fast System Call
SYSEXIT	Fast Return from Fast System Call
SYSRET	Return From Fast System Call
TDPBF16PS	Dot Product of BF16 Tiles Accumulated into Packed Single Precision Tile
TDPBSSD	Dot Product of Signed/Unsigned Bytes with DwordAccumulation
TDPBSUD	Dot Product of Signed/Unsigned Bytes with DwordAccumulation
TDPBUSD	Dot Product of Signed/Unsigned Bytes with DwordAccumulation
TDPBUUD	Dot Product of Signed/Unsigned Bytes with DwordAccumulation
TEST	Logical Compare
TESTUI	Determine User Interrupt Flag
TILELOADD	Load Tile
TILELOADDT1	Load Tile
TILERELEASE	Release Tile
TILESTORED	Store Tile
TILEZERO	Zero Tile
TPAUSE	Timed PAUSE
TZCNT	Count the Number of Trailing Zero Bits
UCOMISD	Unordered Compare Scalar Double Precision Floating-Point Values and Set EFLAGS
UCOMISS	Unordered Compare Scalar Single Precision Floating-Point Values and Set EFLAGS
UD	Undefined Instruction
UIRET	User-Interrupt Return
UMONITOR	User Level Set Up Monitor Address
UMWAIT	User Level Monitor Wait
UNPCKHPD	Unpack and Interleave High Packed Double Precision Floating-Point Values
UNPCKHPS	Unpack and Interleave High Packed Single Precision Floating-Point Values
UNPCKLPD	Unpack and Interleave Low Packed Double Precision Floating-Point Values
UNPCKLPS	Unpack and Interleave Low Packed Single Precision Floating-Point Values
VADDPH	Add Packed FP16 Values
VADDSH	Add Scalar FP16 Values
VALIGND	Align Doubleword/Quadword Vectors
VALIGNQ	Align Doubleword/Quadword Vectors
VBLENDMPD	Blend Float64/Float32 Vectors Using an OpMask Control

VBLENDMPS	Blend Float64/Float32 Vectors Using an OpMask Control
VBROADCAST	Load with Broadcast Floating-Point Data
VCMPPH	Compare Packed FP16 Values
VCMPSH	Compare Scalar FP16 Values
VCOMISH	Compare Scalar Ordered FP16 Values and Set EFLAGS
VCOMPRESSPD	Store Sparse Packed Double Precision Floating-Point Values Into DenseMemory
VCOMPRESSPS	Store Sparse Packed Single Precision Floating-Point Values Into Dense Memory
VCOMPRESSW	Store Sparse Packed Byte/Word Integer Values Into DenseMemory/Register
VCVTDQ2PH	Convert Packed Signed Doubleword Integers to Packed FP16 Values
VCVTNE2PS2BF16	Convert Two Packed Single Data to One Packed BF16 Data
VCVTNEPS2BF16	Convert Packed Single Data to Packed BF16 Data
VCVTPD2PH	Convert Packed Double Precision FP Values to Packed FP16 Values
VCVTPD2QQ	Convert Packed Double Precision Floating-Point Values to Packed QuadwordIntegers
VCVTPD2UDQ	Convert Packed Double Precision Floating-Point Values to Packed UnsignedDoubleword Integers
VCVTPD2UQQ	Convert Packed Double Precision Floating-Point Values to Packed UnsignedQuadword Integers
VCVTPH2DQ	Convert Packed FP16 Values to Signed Doubleword Integers
VCVTPH2PD	Convert Packed FP16 Values to FP64 Values
VCVTPH2PS	Convert Packed FP16 Values to Single Precision Floating-PointValues
VCVTPH2PSX	Convert Packed FP16 Values to Single Precision Floating-PointValues
VCVTPH2QQ	Convert Packed FP16 Values to Signed Quadword Integer Values
VCVTPH2UDQ	Convert Packed FP16 Values to Unsigned Doubleword Integers
VCVTPH2UQQ	Convert Packed FP16 Values to Unsigned Quadword Integers
VCVTPH2UW	Convert Packed FP16 Values to Unsigned Word Integers
VCVTPH2W	Convert Packed FP16 Values to Signed Word Integers
VCVTPS2PH	Convert Single-Precision FP Value to 16-bit FP Value
VCVTPS2PHX	Convert Packed Single Precision Floating-Point Values to Packed FP16 Values
VCVTPS2QQ	Convert Packed Single Precision Floating-Point Values to Packed SignedQuadword Integer Values
VCVTPS2UDQ	Convert Packed Single Precision Floating-Point Values to Packed UnsignedDoubleword Integer Values
VCVTPS2UQQ	Convert Packed Single Precision Floating-Point Values to Packed UnsignedQuadword Integer Values
VCVTQQ2PD	Convert Packed Quadword Integers to Packed Double Precision Floating-PointValues
VCVTQQ2PH	Convert Packed Signed Quadword Integers to Packed FP16 Values
VCVTQQ2PS	Convert Packed Quadword Integers to Packed Single Precision Floating-PointValues

VCVTSD2SH	Convert Low FP64 Value to an FP16 Value
VCVTSD2USI	Convert Scalar Double Precision Floating-Point Value to Unsigned DoublewordInteger
VCVTSH2SD	Convert Low FP16 Value to an FP64 Value
VCVTSH2SI	Convert Low FP16 Value to Signed Integer
VCVTSH2SS	Convert Low FP16 Value to FP32 Value
VCVTSH2USI	Convert Low FP16 Value to Unsigned Integer
VCVTSI2SH	Convert a Signed Doubleword/Quadword Integer to an FP16 Value
VCVTSS2SH	Convert Low FP32 Value to an FP16 Value
VCVTSS2USI	Convert Scalar Single Precision Floating-Point Value to Unsigned DoublewordInteger
VCVTTPD2QQ	Convert With Truncation Packed Double Precision Floating-Point Values toPacked Quadword Integers
VCVTTPD2UDQ	Convert With Truncation Packed Double Precision Floating-Point Values toPacked Unsigned Doubleword Integers
VCVTTPD2UQQ	Convert With Truncation Packed Double Precision Floating-Point Values toPacked Unsigned Quadword Integers
VCVTTPH2DQ	Convert with Truncation Packed FP16 Values to Signed Doubleword Integers
VCVTTPH2QQ	Convert with Truncation Packed FP16 Values to Signed Quadword Integers
VCVTTPH2UDQ	Convert with Truncation Packed FP16 Values to Unsigned DoublewordIntegers
VCVTTPH2UQQ	Convert with Truncation Packed FP16 Values to Unsigned Quadword Integers
VCVTTPH2UW	Convert Packed FP16 Values to Unsigned Word Integers
VCVTTPH2W	Convert Packed FP16 Values to Signed Word Integers
VCVTTPS2QQ	Convert With Truncation Packed Single Precision Floating-Point Values toPacked Signed Quadword Integer Values
VCVTTPS2UDQ	Convert With Truncation Packed Single Precision Floating-Point Values toPacked Unsigned Doubleword Integer Values
VCVTTPS2UQQ	Convert With Truncation Packed Single Precision Floating-Point Values toPacked Unsigned Quadword Integer Values
VCVTTSD2USI	Convert With Truncation Scalar Double Precision Floating-Point Value toUnsigned Integer
VCVTTSH2SI	Convert with Truncation Low FP16 Value to a Signed Integer
VCVTTSH2USI	Convert with Truncation Low FP16 Value to an Unsigned Integer
VCVTTSS2USI	Convert With Truncation Scalar Single Precision Floating-Point Value toUnsigned Integer
VCVTUDQ2PD	Convert Packed Unsigned Doubleword Integers to Packed Double PrecisionFloating-Point Values
VCVTUDQ2PH	Convert Packed Unsigned Doubleword Integers to Packed FP16 Values
VCVTUDQ2PS	Convert Packed Unsigned Doubleword Integers to Packed Single PrecisionFloating-Point Values
VCVTUQQ2PD	Convert Packed Unsigned Quadword Integers to Packed Double PrecisionFloating-Point Values

VCVTUQQ2PH	Convert Packed Unsigned Quadword Integers to Packed FP16 Values
VCVTUQQ2PS	Convert Packed Unsigned Quadword Integers to Packed Single PrecisionFloating-Point Values
VCVTUSI2SD	Convert Unsigned Integer to Scalar Double Precision Floating-Point Value
VCVTUSI2SH	Convert Unsigned Doubleword Integer to an FP16 Value
VCVTUSI2SS	Convert Unsigned Integer to Scalar Single Precision Floating-Point Value
VCVTUW2PH	Convert Packed Unsigned Word Integers to FP16 Values
VCVTW2PH	Convert Packed Signed Word Integers to FP16 Values
VDBPSADBW	Double Block Packed Sum-Absolute-Differences (SAD) on Unsigned Bytes
VDIVPH	Divide Packed FP16 Values
VDIVSH	Divide Scalar FP16 Values
VDPBF16PS	Dot Product of BF16 Pairs Accumulated Into Packed Single Precision
VERR	Verify a Segment for Reading or Writing
VERW	Verify a Segment for Reading or Writing
VEXPANDPD	Load Sparse Packed Double Precision Floating-Point Values From Dense Memory
VEXPANDPS	Load Sparse Packed Single Precision Floating-Point Values From Dense Memory
VEXTRACTF128	Extract Packed Floating-Point Values
VEXTRACTF32x4	Extract Packed Floating-Point Values
VEXTRACTF32x8	Extract Packed Floating-Point Values
VEXTRACTF64x2	Extract Packed Floating-Point Values
VEXTRACTF64x4	Extract Packed Floating-Point Values
VEXTRACTI128	ExtractPacked Integer Values
VEXTRACTI32x4	ExtractPacked Integer Values
VEXTRACTI32x8	ExtractPacked Integer Values
VEXTRACTI64x2	ExtractPacked Integer Values
VEXTRACTI64x4	ExtractPacked Integer Values
VFCMADDCPH	Complex Multiply and Accumulate FP16 Values
VFCMADDCSH	Complex Multiply and Accumulate Scalar FP16 Values
VFCMULCPH	Complex Multiply FP16 Values
VFCMULCSH	Complex Multiply Scalar FP16 Values
VFIXUPIMMPD	Fix Up Special Packed Float64 Values
VFIXUPIMMPS	Fix Up Special Packed Float32 Values
VFIXUPIMMSD	Fix Up Special Scalar Float64 Value
VFIXUPIMMSS	Fix Up Special Scalar Float32 Value
VFMADD132PD	Fused Multiply-Add of Packed DoublePrecision Floating-Point Values
VFMADD132PH	Fused Multiply-Add of Packed FP16 Values
VFMADD132PS	Fused Multiply-Add of Packed SinglePrecision Floating-Point Values
VFMADD132SD	Fused Multiply-Add of Scalar DoublePrecision Floating-Point Values
VFMADD132SH	Fused Multiply-Add of Scalar FP16 Values

VFMADD132SS	Fused Multiply-Add of Scalar Single PrecisionFloating-Point Values
VFMADD213PD	Fused Multiply-Add of Packed DoublePrecision Floating-Point Values
VFMADD213PH	Fused Multiply-Add of Packed FP16 Values
VFMADD213PS	Fused Multiply-Add of Packed SinglePrecision Floating-Point Values
VFMADD213SD	Fused Multiply-Add of Scalar DoublePrecision Floating-Point Values
VFMADD213SH	Fused Multiply-Add of Scalar FP16 Values
VFMADD213SS	Fused Multiply-Add of Scalar Single PrecisionFloating-Point Values
VFMADD231PD	Fused Multiply-Add of Packed DoublePrecision Floating-Point Values
VFMADD231PH	Fused Multiply-Add of Packed FP16 Values
VFMADD231PS	Fused Multiply-Add of Packed SinglePrecision Floating-Point Values
VFMADD231SD	Fused Multiply-Add of Scalar DoublePrecision Floating-Point Values
VFMADD231SH	Fused Multiply-Add of Scalar FP16 Values
VFMADD231SS	Fused Multiply-Add of Scalar Single PrecisionFloating-Point Values
VFMADDCPH	Complex Multiply and Accumulate FP16 Values
VFMADDCSH	Complex Multiply and Accumulate Scalar FP16 Values
VFMADDRND231PD	Fused Multiply-Add of Packed Double-Precision Floating-Point Valueswith rounding control
VFMADDSUB132PD	Fused Multiply-AlternatingAdd/Subtract of Packed Double Precision Floating-Point Values
VFMADDSUB132PH	Fused Multiply-AlternatingAdd/Subtract of Packed FP16 Values
VFMADDSUB132PS	Fused Multiply-AlternatingAdd/Subtract of Packed Single Precision Floating-Point Values
VFMADDSUB213PD	Fused Multiply-AlternatingAdd/Subtract of Packed Double Precision Floating-Point Values
VFMADDSUB213PH	Fused Multiply-AlternatingAdd/Subtract of Packed FP16 Values
VFMADDSUB213PS	Fused Multiply-AlternatingAdd/Subtract of Packed Single Precision Floating-Point Values
VFMADDSUB231PD	Fused Multiply-AlternatingAdd/Subtract of Packed Double Precision Floating-Point Values
VFMADDSUB231PH	Fused Multiply-AlternatingAdd/Subtract of Packed FP16 Values
VFMADDSUB231PS	Fused Multiply-AlternatingAdd/Subtract of Packed Single Precision Floating-Point Values
VFMSUB132PD	Fused Multiply-Subtract of Packed DoublePrecision Floating-Point Values
VFMSUB132PH	Fused Multiply-Subtract of Packed FP16 Values
VFMSUB132PS	Fused Multiply-Subtract of Packed SinglePrecision Floating-Point Values
VFMSUB132SD	Fused Multiply-Subtract of Scalar DoublePrecision Floating-Point Values
VFMSUB132SH	Fused Multiply-Subtract of Scalar FP16 Values
VFMSUB132SS	Fused Multiply-Subtract of Scalar SinglePrecision Floating-Point Values
VFMSUB213PD	Fused Multiply-Subtract of Packed DoublePrecision Floating-Point Values
VFMSUB213PH	Fused Multiply-Subtract of Packed FP16 Values
VFMSUB213PS	Fused Multiply-Subtract of Packed SinglePrecision Floating-Point Values
VFMSUB213SD	Fused Multiply-Subtract of Scalar DoublePrecision Floating-Point Values

VFMSUB213SH	Fused Multiply-Subtract of Scalar FP16 Values
VFMSUB213SS	Fused Multiply-Subtract of Scalar SinglePrecision Floating-Point Values
VFMSUB231PD	Fused Multiply-Subtract of Packed DoublePrecision Floating-Point Values
VFMSUB231PH	Fused Multiply-Subtract of Packed FP16 Values
VFMSUB231PS	Fused Multiply-Subtract of Packed SinglePrecision Floating-Point Values
VFMSUB231SD	Fused Multiply-Subtract of Scalar DoublePrecision Floating-Point Values
VFMSUB231SH	Fused Multiply-Subtract of Scalar FP16 Values
VFMSUB231SS	Fused Multiply-Subtract of Scalar SinglePrecision Floating-Point Values
VFMSUBADD132PD	Fused Multiply-AlternatingSubtract/Add of Packed Double Precision Floating-Point Values
VFMSUBADD132PH	Fused Multiply-AlternatingSubtract/Add of Packed FP16 Values
VFMSUBADD132PS	Fused Multiply-AlternatingSubtract/Add of Packed Single Precision Floating-Point Values
VFMSUBADD213PD	Fused Multiply-AlternatingSubtract/Add of Packed Double Precision Floating-Point Values
VFMSUBADD213PH	Fused Multiply-AlternatingSubtract/Add of Packed FP16 Values
VFMSUBADD213PS	Fused Multiply-AlternatingSubtract/Add of Packed Single Precision Floating-Point Values
VFMSUBADD231PD	Fused Multiply-AlternatingSubtract/Add of Packed Double Precision Floating-Point Values
VFMSUBADD231PH	Fused Multiply-AlternatingSubtract/Add of Packed FP16 Values
VFMSUBADD231PS	Fused Multiply-AlternatingSubtract/Add of Packed Single Precision Floating-Point Values
VFMULCPH	Complex Multiply FP16 Values
VFMULCSH	Complex Multiply Scalar FP16 Values
VFNMADD132PD	Fused Negative Multiply-Add of PackedDouble Precision Floating-Point Values
VFNMADD132PH	Fused Multiply-Add of Packed FP16 Values
VFNMADD132PS	Fused Negative Multiply-Add of PackedSingle Precision Floating-Point Values
VFNMADD132SD	Fused Negative Multiply-Add of ScalarDouble Precision Floating-Point Values
VFNMADD132SH	Fused Multiply-Add of Scalar FP16 Values
VFNMADD132SS	Fused Negative Multiply-Add of ScalarSingle Precision Floating-Point Values
VFNMADD213PD	Fused Negative Multiply-Add of PackedDouble Precision Floating-Point Values
VFNMADD213PH	Fused Multiply-Add of Packed FP16 Values
VFNMADD213PS	Fused Negative Multiply-Add of PackedSingle Precision Floating-Point Values
VFNMADD213SD	Fused Negative Multiply-Add of ScalarDouble Precision Floating-Point Values
VFNMADD213SH	Fused Multiply-Add of Scalar FP16 Values

VFNMADD213SS	Fused Negative Multiply-Add of ScalarSingle Precision Floating-Point Values
VFNMADD231PD	Fused Negative Multiply-Add of PackedDouble Precision Floating-Point Values
VFNMADD231PH	Fused Multiply-Add of Packed FP16 Values
VFNMADD231PS	Fused Negative Multiply-Add of PackedSingle Precision Floating-Point Values
VFNMADD231SD	Fused Negative Multiply-Add of ScalarDouble Precision Floating-Point Values
VFNMADD231SH	Fused Multiply-Add of Scalar FP16 Values
VFNMADD231SS	Fused Negative Multiply-Add of ScalarSingle Precision Floating-Point Values
VFNMSUB132PD	Fused Negative Multiply-Subtract ofPacked Double Precision Floating-Point Values
VFNMSUB132PH	Fused Multiply-Subtract of Packed FP16 Values
VFNMSUB132PS	Fused Negative Multiply-Subtract ofPacked Single Precision Floating-Point Values
VFNMSUB132SD	Fused Negative Multiply-Subtract ofScalar Double Precision Floating-Point Values
VFNMSUB132SH	Fused Multiply-Subtract of Scalar FP16 Values
VFNMSUB132SS	Fused Negative Multiply-Subtract ofScalar Single Precision Floating-Point Values
VFNMSUB213PD	Fused Negative Multiply-Subtract ofPacked Double Precision Floating-Point Values
VFNMSUB213PH	Fused Multiply-Subtract of Packed FP16 Values
VFNMSUB213PS	Fused Negative Multiply-Subtract ofPacked Single Precision Floating-Point Values
VFNMSUB213SD	Fused Negative Multiply-Subtract ofScalar Double Precision Floating-Point Values
VFNMSUB213SH	Fused Multiply-Subtract of Scalar FP16 Values
VFNMSUB213SS	Fused Negative Multiply-Subtract ofScalar Single Precision Floating-Point Values
VFNMSUB231PD	Fused Negative Multiply-Subtract ofPacked Double Precision Floating-Point Values
VFNMSUB231PH	Fused Multiply-Subtract of Packed FP16 Values
VFNMSUB231PS	Fused Negative Multiply-Subtract ofPacked Single Precision Floating-Point Values
VFNMSUB231SD	Fused Negative Multiply-Subtract ofScalar Double Precision Floating-Point Values
VFNMSUB231SH	Fused Multiply-Subtract of Scalar FP16 Values
VFNMSUB231SS	Fused Negative Multiply-Subtract ofScalar Single Precision Floating-Point Values
VFPCLASSPD	Tests Types of Packed Float64 Values
VFPCLASSPH	Test Types of Packed FP16 Values

VFPCLASSPS	Tests Types of Packed Float32 Values
VFPCLASSSD	Tests Type of a Scalar Float64 Value
VFPCLASSSH	Test Types of Scalar FP16 Values
VFPCLASSSS	Tests Type of a Scalar Float32 Value
VGATHERDPD	Gather Packed Double Precision Floating-Point Values UsingSigned Dword/Qword Indices
VGATHERDPD (1)	Gather Packed Single, Packed Double with Signed Dword Indices
VGATHERDPS	Gather Packed Single Precision Floating-Point Values UsingSigned Dword/Qword Indices
VGATHERDPS (1)	Gather Packed Single, Packed Double with Signed Dword Indices
VGATHERQPD	Gather Packed Double Precision Floating-Point Values UsingSigned Dword/Qword Indices
VGATHERQPD (1)	Gather Packed Single, Packed Double with Signed Qword Indices
VGATHERQPS	Gather Packed Single Precision Floating-Point Values UsingSigned Dword/Qword Indices
VGATHERQPS (1)	Gather Packed Single, Packed Double with Signed Qword Indices
VGETEXPPD	Convert Exponents of Packed Double Precision Floating-Point Values to DoublePrecision Floating-Point Values
VGETEXPPH	Convert Exponents of Packed FP16 Values to FP16 Values
VGETEXPPS	Convert Exponents of Packed Single Precision Floating-Point Values to SinglePrecision Floating-Point Values
VGETEXPSD	Convert Exponents of Scalar Double Precision Floating-Point Value to DoublePrecision Floating-Point Value
VGETEXPSH	Convert Exponents of Scalar FP16 Values to FP16 Values
VGETEXPSS	Convert Exponents of Scalar Single Precision Floating-Point Value to SinglePrecision Floating-Point Value
VGETMANTPD	Extract Float64 Vector of Normalized Mantissas From Float64 Vector
VGETMANTPH	Extract FP16 Vector of Normalized Mantissas from FP16 Vector
VGETMANTPS	Extract Float32 Vector of Normalized Mantissas From Float32 Vector
VGETMANTSD	Extract Float64 of Normalized Mantissa From Float64 Scalar
VGETMANTSH	Extract FP16 of Normalized Mantissa from FP16 Scalar
VGETMANTSS	Extract Float32 Vector of Normalized Mantissa From Float32 Scalar
VINSERTF128	Insert PackedFloating-Point Values
VINSERTF32x4	Insert PackedFloating-Point Values
VINSERTF32x8	Insert PackedFloating-Point Values
VINSERTF64x2	Insert PackedFloating-Point Values
VINSERTF64x4	Insert PackedFloating-Point Values
VINSERTI128	Insert PackedInteger Values
VINSERTI32x4	Insert PackedInteger Values
VINSERTI32x8	Insert PackedInteger Values
VINSERTI64x2	Insert PackedInteger Values
VINSERTI64x4	Insert PackedInteger Values

VMASKMOV	Conditional SIMD Packed Loads and Stores
VMAXPH	Return Maximum of Packed FP16 Values
VMAXSH	Return Maximum of Scalar FP16 Values
VMINPH	Return Minimum of Packed FP16 Values
VMINSH	Return Minimum Scalar FP16 Value
VMOVDQA32	Move Aligned Packed Integer Values
VMOVDQA64	Move Aligned Packed Integer Values
VMOVDQU16	Move Unaligned Packed Integer Values
VMOVDQU32	Move Unaligned Packed Integer Values
VMOVDQU64	Move Unaligned Packed Integer Values
VMOVDQU8	Move Unaligned Packed Integer Values
VMOVSH	Move Scalar FP16 Value
VMOVW	Move Word
VMULPH	Multiply Packed FP16 Values
VMULSH	Multiply Scalar FP16 Values
VP2INTERSECTD	Compute Intersection Between DWORDS/QUADWORDS to aPair of Mask Registers
VP2INTERSECTQ	Compute Intersection Between DWORDS/QUADWORDS to aPair of Mask Registers
VPBLENDD	Blend Packed Dwords
VPBLENDMB	Blend Byte/Word Vectors Using an Opmask Control
VPBLENDMD	Blend Int32/Int64 Vectors Using an OpMask Control
VPBLENDMQ	Blend Int32/Int64 Vectors Using an OpMask Control
VPBLENDMW	Blend Byte/Word Vectors Using an Opmask Control
VPBROADCAST	Load Integer and Broadcast
VPBROADCASTB	Load With Broadcast Integer Data From General Purpose Register
VPBROADCASTD	Load With Broadcast Integer Data From General Purpose Register
VPBROADCASTM	Broadcast Mask to Vector Register
VPBROADCASTQ	Load With Broadcast Integer Data From General Purpose Register
VPBROADCASTW	Load With Broadcast Integer Data From General Purpose Register
VPCMPB	Compare Packed Byte Values Into Mask
VPCMPD	Compare Packed Integer Values Into Mask
VPCMPQ	Compare Packed Integer Values Into Mask
VPCMPUB	Compare Packed Byte Values Into Mask
VPCMPUD	Compare Packed Integer Values Into Mask
VPCMPUQ	Compare Packed Integer Values Into Mask
VPCMPUW	Compare Packed Word Values Into Mask
VPCMPW	Compare Packed Word Values Into Mask
VPCOMPRESSB	Store Sparse Packed Byte/Word Integer Values Into DenseMemory/Register
VPCOMPRESSD	Store Sparse Packed Doubleword Integer Values Into Dense Memory/Register

VPCOMPRESSQ	Store Sparse Packed Quadword Integer Values Into Dense Memory/Register
VPCONFLICTD	Detect Conflicts Within a Vector of Packed Dword/Qword Values Into DenseMemory/ Register
VPCONFLICTQ	Detect Conflicts Within a Vector of Packed Dword/Qword Values Into DenseMemory/ Register
VPDPBUSD	Multiply and Add Unsigned and Signed Bytes
VPDPBUSDS	Multiply and Add Unsigned and Signed Bytes With Saturation
VPDPWSSD	Multiply and Add Signed Word Integers
VPDPWSSDS	Multiply and Add Signed Word Integers With Saturation
VPERM2F128	Permute Floating-Point Values
VPERM2I128	Permute Integer Values
VPERMB	Permute Packed Bytes Elements
VPERMD	Permute Packed Doubleword/Word Elements
VPERMI2B	Full Permute of Bytes From Two Tables Overwriting the Index
VPERMI2D	Full Permute From Two Tables Overwriting the Index
VPERMI2PD	Full Permute From Two Tables Overwriting the Index
VPERMI2PS	Full Permute From Two Tables Overwriting the Index
VPERMI2Q	Full Permute From Two Tables Overwriting the Index
VPERMI2W	Full Permute From Two Tables Overwriting the Index
VPERMILPD	Permute In-Lane of Pairs of Double Precision Floating-Point Values
VPERMILPS	Permute In-Lane of Quadruples of Single Precision Floating-Point Values
VPERMPD	Permute Double Precision Floating-Point Elements
VPERMPS	Permute Single Precision Floating-Point Elements
VPERMQ	Qwords Element Permutation
VPERMT2B	Full Permute of Bytes From Two Tables Overwriting a Table
VPERMT2D	Full Permute From Two Tables Overwriting One Table
VPERMT2PD	Full Permute From Two Tables Overwriting One Table
VPERMT2PS	Full Permute From Two Tables Overwriting One Table
VPERMT2Q	Full Permute From Two Tables Overwriting One Table
VPERMT2W	Full Permute From Two Tables Overwriting One Table
VPERMW	Permute Packed Doubleword/Word Elements
VPEXPANDB	Expand Byte/Word Values
VPEXPANDD	Load Sparse Packed Doubleword Integer Values From Dense Memory/Register
VPEXPANDQ	Load Sparse Packed Quadword Integer Values From Dense Memory/Register
VPEXPANDW	Expand Byte/Word Values
VPGATHERDD	Gather Packed Dword Values Using Signed Dword/Qword Indices
VPGATHERDD (1)	Gather Packed Dword, Packed Qword With Signed Dword Indices
VPGATHERDQ	Gather Packed Dword, Packed Qword With Signed Dword Indices
VPGATHERDQ (1)	Gather Packed Qword Values Using Signed Dword/Qword Indices
VPGATHERQD	Gather Packed Dword Values Using Signed Dword/Qword Indices

VPGATHERQD (1)	Gather Packed Dword, Packed Qword with Signed Qword Indices
VPGATHERQQ	Gather Packed Qword Values Using Signed Dword/Qword Indices
VPGATHERQQ (1)	Gather Packed Dword, Packed Qword with Signed Qword Indices
VPLZCNTD	Count the Number of Leading Zero Bits for Packed Dword, Packed Qword Values
VPLZCNTQ	Count the Number of Leading Zero Bits for Packed Dword, Packed Qword Values
VPMADD52HUQ	Packed Multiply of Unsigned 52-Bit Unsigned Integers and Add High 52-BitProducts to 64-Bit Accumulators
VPMADD52LUQ	Packed Multiply of Unsigned 52-Bit Integers and Add the Low 52-Bit Productsto Qword Accumulators
VPMASKMOV	Conditional SIMD Integer Packed Loads and Stores
VPMOVB2M	Convert a Vector Register to a Mask
VPMOVD2M	Convert a Vector Register to a Mask
VPMOVDB	Down Convert DWord to Byte
VPMOVDW	Down Convert DWord to Word
VPMOVM2B	Convert a Mask Register to a VectorRegister
VPMOVM2D	Convert a Mask Register to a VectorRegister
VPMOVM2Q	Convert a Mask Register to a VectorRegister
VPMOVM2W	Convert a Mask Register to a VectorRegister
VPMOVQ2M	Convert a Vector Register to a Mask
VPMOVQB	Down Convert QWord to Byte
VPMOVQD	Down Convert QWord to DWord
VPMOVQW	Down Convert QWord to Word
VPMOVSDB	Down Convert DWord to Byte
VPMOVSDW	Down Convert DWord to Word
VPMOVSQB	Down Convert QWord to Byte
VPMOVSQD	Down Convert QWord to DWord
VPMOVSQW	Down Convert QWord to Word
VPMOVSWB	Down Convert Word to Byte
VPMOVUSDB	Down Convert DWord to Byte
VPMOVUSDW	Down Convert DWord to Word
VPMOVUSQB	Down Convert QWord to Byte
VPMOVUSQD	Down Convert QWord to DWord
VPMOVUSQW	Down Convert QWord to Word
VPMOVUSWB	Down Convert Word to Byte
VPMOVW2M	Convert a Vector Register to a Mask
VPMOVWB	Down Convert Word to Byte
VPMULTISHIFTQB	Select Packed Unaligned Bytes From Quadword Sources
VPOPCNT	Return the Count of Number of Bits Set to 1 in BYTE/WORD/DWORD/QWORD

VPROLD	Bit Rotate Left
VPROLQ	Bit Rotate Left
VPROLVD	Bit Rotate Left
VPROLVQ	Bit Rotate Left
VPRORD	Bit Rotate Right
VPRORQ	Bit Rotate Right
VPRORVD	Bit Rotate Right
VPRORVQ	Bit Rotate Right
VPSCATTERDD	Scatter Packed Dword, PackedQword with Signed Dword, Signed Qword Indices
VPSCATTERDQ	Scatter Packed Dword, PackedQword with Signed Dword, Signed Qword Indices
VPSCATTERQD	Scatter Packed Dword, PackedQword with Signed Dword, Signed Qword Indices
VPSCATTERQQ	Scatter Packed Dword, PackedQword with Signed Dword, Signed Qword Indices
VPSHLD	Concatenate and Shift Packed Data Left Logical
VPSHLDV	Concatenate and Variable Shift Packed Data Left Logical
VPSHRD	Concatenate and Shift Packed Data Right Logical
VPSHRDV	Concatenate and Variable Shift Packed Data Right Logical
VPSHUFBITQMB	Shuffle Bits From Quadword Elements Using Byte Indexes Into Mask
VPSLLVD	Variable Bit Shift Left Logical
VPSLLVQ	Variable Bit Shift Left Logical
VPSLLVW	Variable Bit Shift Left Logical
VPSRAVD	Variable Bit Shift Right Arithmetic
VPSRAVQ	Variable Bit Shift Right Arithmetic
VPSRAVW	Variable Bit Shift Right Arithmetic
VPSRLVD	Variable Bit Shift Right Logical
VPSRLVQ	Variable Bit Shift Right Logical
VPSRLVW	Variable Bit Shift Right Logical
VPTERNLOGD	Bitwise Ternary Logic
VPTERNLOGQ	Bitwise Ternary Logic
VPTESTMB	Logical AND and Set Mask
VPTESTMD	Logical AND and Set Mask
VPTESTMQ	Logical AND and Set Mask
VPTESTMW	Logical AND and Set Mask
VPTESTNMB	Logical NAND and Set
VPTESTNMD	Logical NAND and Set
VPTESTNMQ	Logical NAND and Set
VPTESTNMW	Logical NAND and Set
VRANGEPD	Range Restriction Calculation for Packed Pairs of Float64 Values

VRANGEPS	Range Restriction Calculation for Packed Pairs of Float32 Values
VRANGESD	Range Restriction Calculation From a Pair of Scalar Float64 Values
VRANGESS	Range Restriction Calculation From a Pair of Scalar Float32 Values
VRCP14PD	Compute Approximate Reciprocals of Packed Float64 Values
VRCP14PS	Compute Approximate Reciprocals of Packed Float32 Values
VRCP14SD	Compute Approximate Reciprocal of Scalar Float64 Value
VRCP14SS	Compute Approximate Reciprocal of Scalar Float32 Value
VRCPPH	Compute Reciprocals of Packed FP16 Values
VRCPSH	Compute Reciprocal of Scalar FP16 Value
VREDUCEPD	Perform Reduction Transformation on Packed Float64 Values
VREDUCEPH	Perform Reduction Transformation on Packed FP16 Values
VREDUCEPS	Perform Reduction Transformation on Packed Float32 Values
VREDUCESD	Perform a Reduction Transformation on a Scalar Float64 Value
VREDUCESH	Perform Reduction Transformation on Scalar FP16 Value
VREDUCESS	Perform a Reduction Transformation on a Scalar Float32 Value
VRNDSCALEPD	Round Packed Float64 Values to Include a Given Number of Fraction Bits
VRNDSCALEPH	Round Packed FP16 Values to Include a Given Number of Fraction Bits
VRNDSCALEPS	Round Packed Float32 Values to Include a Given Number of Fraction Bits
VRNDSCALESD	Round Scalar Float64 Value to Include a Given Number of Fraction Bits
VRNDSCALESH	Round Scalar FP16 Value to Include a Given Number of Fraction Bits
VRNDSCALESS	Round Scalar Float32 Value to Include a Given Number of Fraction Bits
VRSQRT14PD	Compute Approximate Reciprocals of Square Roots of Packed Float64 Values
VRSQRT14PS	Compute Approximate Reciprocals of Square Roots of Packed Float32 Values
VRSQRT14SD	Compute Approximate Reciprocal of Square Root of Scalar Float64 Value
VRSQRT14SS	Compute Approximate Reciprocal of Square Root of Scalar Float32 Value
VRSQRTPH	Compute Reciprocals of Square Roots of Packed FP16 Values
VRSQRTSH	Compute Approximate Reciprocal of Square Root of Scalar FP16 Value
VSCALEFPD	Scale Packed Float64 Values With Float64 Values
VSCALEFPH	Scale Packed FP16 Values with FP16 Values
VSCALEFPS	Scale Packed Float32 Values With Float32 Values
VSCALEFSD	Scale Scalar Float64 Values With Float64 Values
VSCALEFSH	Scale Scalar FP16 Values with FP16 Values
VSCALEFSS	Scale Scalar Float32 Value With Float32 Value
VSCATTERDPD	Scatter Packed Single, PackedDouble with Signed Dword and Qword Indices
VSCATTERDPS	Scatter Packed Single, PackedDouble with Signed Dword and Qword Indices
VSCATTERQPD	Scatter Packed Single, PackedDouble with Signed Dword and Qword Indices
VSCATTERQPS	Scatter Packed Single, PackedDouble with Signed Dword and Qword Indices
VSHUFF32x4	Shuffle Packed Values at 128-BitGranularity
VSHUFF64x2	Shuffle Packed Values at 128-BitGranularity
VSHUFI32x4	Shuffle Packed Values at 128-BitGranularity

VSHUFI64x2	Shuffle Packed Values at 128-BitGranularity
VSQRTPH	Compute Square Root of Packed FP16 Values
VSQRTSH	Compute Square Root of Scalar FP16 Value
VSUBPH	Subtract Packed FP16 Values
VSUBSH	Subtract Scalar FP16 Value
VTESTPD	Packed Bit Test
VTESTPS	Packed Bit Test
VUCOMISH	Unordered Compare Scalar FP16 Values and Set EFLAGS
VZEROALL	Zero XMM, YMM, and ZMM Registers
VZEROUPPER	Zero Upper Bits of YMM and ZMM Registers
WAIT	Wait
WBINVD	Write Back and Invalidate Cache
WBNOINVD	Write Back and Do Not Invalidate Cache
WRFSBASE	Write FS/GS Segment Base
WRGSBASE	Write FS/GS Segment Base
WRMSR	Write to Model Specific Register
WRPKRU	Write Data to User Page Key Register
WRSSD	Write to Shadow Stack
WRSSQ	Write to Shadow Stack
WRUSSD	Write to User Shadow Stack
WRUSSQ	Write to User Shadow Stack
XABORT	Transactional Abort
XACQUIRE	Hardware Lock Elision Prefix Hints
XADD	Exchange and Add
XBEGIN	Transactional Begin
XCHG	Exchange Register/Memory With Register
XEND	Transactional End
XGETBV	Get Value of Extended Control Register
XLAT	Table Look-up Translation
XLATB	Table Look-up Translation
XOR	Logical Exclusive OR
XORPD	Bitwise Logical XOR of Packed Double Precision Floating-Point Values
XORPS	Bitwise Logical XOR of Packed Single Precision Floating-Point Values
XRELEASE	Hardware Lock Elision Prefix Hints
XRESLDTRK	Resume Tracking Load Addresses
XRSTOR	Restore Processor Extended States
XRSTORS	Restore Processor Extended States Supervisor
XSAVE	Save Processor Extended States
XSAVEC	Save Processor Extended States With Compaction
XSAVEOPT	Save Processor Extended States Optimized

XSAVES	Save Processor Extended States Supervisor
XSETBV	Set Extended Control Register
XSUSLDTRK	Suspend Tracking Load Addresses
XTEST	Test if in Transactional Execution

SGX Instructions

Mnemonic	Summary
ENCLS	Execute an Enclave System Function of Specified Leaf Number
ENCLS[EADD]	Add a Page to an Uninitialized Enclave
ENCLS[EAUG]	Add a Page to an Initialized Enclave
ENCLS[EBLOCK]	Mark a page in EPC as Blocked
ENCLS[ECREATE]	Create an SECS page in the Enclave Page Cache
ENCLS[EDBGRD]	Read From a Debug Enclave
ENCLS[EDBGWR]	Write to a Debug Enclave
ENCLS[EEXTEND]	Extend Uninitialized Enclave Measurement by 256 Bytes
ENCLS[EINIT]	Initialize an Enclave for Execution
ENCLS[ELDBC]	Load an EPC Page and Mark its State
ENCLS[ELDB]	Load an EPC Page and Mark its State
ENCLS[ELDUC]	Load an EPC Page and Mark its State
ENCLS[ELDU]	Load an EPC Page and Mark its State
ENCLS[EMODPR]	Restrict the Permissions of an EPC Page
ENCLS[EMODT]	Change the Type of an EPC Page
ENCLS[EPA]	Add Version Array
ENCLS[ERDINFO]	Read Type and Status Information About an EPC Page
ENCLS[EREMOVE]	Remove a page from the EPC
ENCLS[ETRACKC]	Activates EBLOCK Checks
ENCLS[ETRACK]	Activates EBLOCK Checks
ENCLS[EWB]	Invalidate an EPC Page and Write out to Main Memory
ENCLU	Execute an Enclave User Function of Specified Leaf Number
ENCLU[EACCEPTCOPY]	Initialize a Pending Page
ENCLU[EACCEPT]	Accept Changes to an EPC Page
ENCLU[EDECCSSA]	Decrements TCS.CSSA
ENCLU[EENTER]	Enters an Enclave
ENCLU[EEXIT]	Exits an Enclave
ENCLU[EGETKEY]	Retrieves a Cryptographic Key
ENCLU[EMODPE]	Extend an EPC Page Permissions
ENCLU[EREPORT]	Create a Cryptographic Report of the Enclave
ENCLU[ERESUME]	Re-Enters an Enclave
ENCLV	Execute an Enclave VMM Function of Specified Leaf Number

ENCLV[EDECVIRTCHILD]	Decrement VIRTCHILDCNT in SECS
ENCLV[EINCVIRTCHILD]	Increment VIRTCHILDCNT in SECS
ENCLV[ESETCONTEXT]	Set the ENCLAVECONTEXT Field in SECS

SMX Instructions

Mnemonic	Summary
GETSEC[CAPABILITIES]	Report the SMX Capabilities
GETSEC[ENTERACCS]	Execute Authenticated Chipset Code
GETSEC[EXITAC]	Exit Authenticated Code Execution Mode
GETSEC[PARAMETERS]	Report the SMX Parameters
GETSEC[SENTER]	Enter a Measured Environment
GETSEC[SEXIT]	Exit Measured Environment
GETSEC[SMCTRL]	SMX Mode Control
GETSEC[WAKEUP]	Wake Up Sleeping Processors in Measured Environment

VMX Instructions

Mnemonic	Summary
INVEPT	Invalidate Translations Derived from EPT
INVVPID	Invalidate Translations Based on VPID
VMCALL	Call to VM Monitor
VMCLEAR	Clear Virtual-Machine Control Structure
VMFUNC	Invoke VM function
VMLAUNCH	Launch/Resume Virtual Machine
VMPTRLD	Load Pointer to Virtual-Machine Control Structure
VMPTRST	Store Pointer to Virtual-Machine Control Structure
VMREAD	Read Field from Virtual-Machine Control Structure
VMRESUME	Launch/Resume Virtual Machine
VMRESUME (1)	Resume Virtual Machine
VMWRITE	Write Field to Virtual-Machine Control Structure
VMXOFF	Leave VMX Operation
VMXON	Enter VMX Operation

Xeon Phi™ Instructions

Mnemonic	Summary
PREFETCHWT1	Prefetch Vector Data Into Caches With Intent to Write and T1 Hint
V4FMADDPS	Packed Single Precision Floating-Point Fused Multiply-Add(4-Iterations)
V4FMADDSS	Scalar Single Precision Floating-Point Fused Multiply-Add(4-Iterations)

V4FNMADDPS	Packed Single Precision Floating-Point Fused Multiply-Add(4-Iterations)
V4FNMADDSS	Scalar Single Precision Floating-Point Fused Multiply-Add(4-Iterations)
VEXP2PD	Approximation to the Exponential 2^x of Packed Double Precision Floating-PointValues With Less Than 2^-23 Relative Error
VEXP2PS	Approximation to the Exponential 2^x of Packed Single Precision Floating-PointValues With Less Than 2^-23 Relative Error
VGATHERPF0DPD	Sparse PrefetchPacked SP/DP Data Values With Signed Dword, Signed Qword Indices Using T0 Hint
VGATHERPF0DPS	Sparse PrefetchPacked SP/DP Data Values With Signed Dword, Signed Qword Indices Using T0 Hint
VGATHERPF0QPD	Sparse PrefetchPacked SP/DP Data Values With Signed Dword, Signed Qword Indices Using T0 Hint
VGATHERPF0QPS	Sparse PrefetchPacked SP/DP Data Values With Signed Dword, Signed Qword Indices Using T0 Hint
VGATHERPF1DPD	Sparse PrefetchPacked SP/DP Data Values With Signed Dword, Signed Qword Indices Using T1 Hint
VGATHERPF1DPS	Sparse PrefetchPacked SP/DP Data Values With Signed Dword, Signed Qword Indices Using T1 Hint
VGATHERPF1QPD	Sparse PrefetchPacked SP/DP Data Values With Signed Dword, Signed Qword Indices Using T1 Hint
VGATHERPF1QPS	Sparse PrefetchPacked SP/DP Data Values With Signed Dword, Signed Qword Indices Using T1 Hint
VP4DPWSSD	Dot Product of Signed Words With Dword Accumulation (4-Iterations)
VP4DPWSSDS	Dot Product of Signed Words With Dword Accumulation and Saturation(4-Iterations)
VRCP28PD	Approximation to the Reciprocal of Packed Double Precision Floating-Point ValuesWith Less Than 2^-28 Relative Error
VRCP28PS	Approximation to the Reciprocal of Packed Single Precision Floating-Point ValuesWith Less Than 2^-28 Relative Error
VRCP28SD	Approximation to the Reciprocal of Scalar Double Precision Floating-Point ValueWith Less Than 2^-28 Relative Error
VRCP28SS	Approximation to the Reciprocal of Scalar Single Precision Floating-Point ValueWith Less Than 2^-28 Relative Error
VRSQRT28PD	Approximation to the Reciprocal Square Root of Packed Double PrecisionFloating-Point Values With Less Than 2^-28 Relative Error
VRSQRT28PS	Approximation to the Reciprocal Square Root of Packed Single PrecisionFloating-Point Values With Less Than 2^-28 Relative Error
VRSQRT28SD	Approximation to the Reciprocal Square Root of Scalar Double PrecisionFloating-Point Value With Less Than 2^-28 Relative Error
VRSQRT28SS	Approximation to the Reciprocal Square Root of Scalar Single Precision Floating-Point Value With Less Than 2^-28 Relative Error
VSCATTERPF0DPD	Sparse PrefetchPacked SP/DP Data Values with Signed Dword, Signed Qword Indices Using T0 Hint With Intentto Write
VSCATTERPF0DPS	Sparse PrefetchPacked SP/DP Data Values with Signed Dword, Signed Qword Indices Using T0 Hint With Intentto Write

VSCATTERPF0QPD	Sparse PrefetchPacked SP/DP Data Values with Signed Dword, Signed Qword Indices Using T0 Hint With Intentto Write
VSCATTERPF0QPS	Sparse PrefetchPacked SP/DP Data Values with Signed Dword, Signed Qword Indices Using T0 Hint With Intentto Write
VSCATTERPF1DPD	Sparse PrefetchPacked SP/DP Data Values With Signed Dword, Signed Qword Indices Using T1 Hint With Intentto Write
VSCATTERPF1DPS	Sparse PrefetchPacked SP/DP Data Values With Signed Dword, Signed Qword Indices Using T1 Hint With Intentto Write
VSCATTERPF1QPD	Sparse PrefetchPacked SP/DP Data Values With Signed Dword, Signed Qword Indices Using T1 Hint With Intentto Write
VSCATTERPF1QPS	Sparse PrefetchPacked SP/DP Data Values With Signed Dword, Signed Qword Indices Using T1 Hint With Intentto Write