XCHG—Exchange Register/Memory with Register

Opcode	Instruction	Op/ En	64-Bit Mode	Compat/ Leg Mode	Description
90+*rw*	XCHG AX, *r16*	A	Valid	Valid	Exchange *r16* with AX.
90+*rw*	XCHG *r16*, AX	B	Valid	Valid	Exchange AX with *r16.*
90+*rd*	XCHG EAX, *r32*	A	Valid	Valid	Exchange *r32* with EAX.
REX.W + 90+*rd*	XCHG RAX, *r64*	A	Valid	N.E.	Exchange *r64* with RAX.
90+*rd*	XCHG *r32*, EAX	B	Valid	Valid	Exchange EAX with *r32.*
REX.W + 90+*rd*	XCHG *r64*, RAX	B	Valid	N.E.	Exchange RAX with *r64.*
86 /*r*	XCHG *r/m8, r8*	C	Valid	Valid	Exchange *r8* (byte register) with byte from *r/m8.*
REX + 86 /*r*	XCHG *r/m8*, r8**	C	Valid	N.E.	Exchange *r8* (byte register) with byte from *r/m8.*
86 /*r*	XCHG *r8, r/m8*	D	Valid	Valid	Exchange byte from *r/m8* with *r8* (byte register).
REX + 86 /*r*	XCHG *r8*, r/m8**	D	Valid	N.E.	Exchange byte from *r/m8* with *r8* (byte register).
87 /*r*	XCHG *r/m16, r16*	C	Valid	Valid	Exchange *r16* with word from *r/m16.*
87 /*r*	XCHG *r16, r/m16*	D	Valid	Valid	Exchange word from *r/m16* with *r16.*
87 /*r*	XCHG *r/m32, r32*	C	Valid	Valid	Exchange *r32* with doubleword from *r/m32.*
REX.W + 87 /*r*	XCHG *r/m64, r64*	C	Valid	N.E.	Exchange *r64* with quadword from *r/m64.*
87 /*r*	XCHG *r32, r/m32*	D	Valid	Valid	Exchange doubleword from *r/m32* with *r32.*
REX.W + 87 /*r*	XCHG *r64, r/m64*	D	Valid	N.E.	Exchange quadword from *r/m64* with *r64.*

NOTES:

* In 64-bit mode, r/m8 can not be encoded to access the following byte registers if a REX prefix is used: AH, BH, CH, DH.

Instruction Operand Encoding

Op/En	Operand 1	Operand 2	Operand 3	Operand 4
A	AX/EAX/RAX (r, w)	reg (r, w)	NA	NA
B	reg (r, w)	AX/EAX/RAX (r, w)	NA	NA
C	ModRM:r/m (r, w)	ModRM:reg (r, w)	NA	NA
D	ModRM:reg (r, w)	ModRM:r/m (r, w)	NA	NA

Description

Exchanges the contents of the destination (first) and source (second) operands. The operands can be two general-purpose registers or a register and a memory location. If a memory operand is referenced, the processor's locking protocol is automatically implemented for the duration of the exchange operation, regardless of the presence or absence of the LOCK prefix or of the value of the IOPL. (See the LOCK prefix description in this chapter for more information on the locking protocol.)

This instruction is useful for implementing semaphores or similar data structures for process synchronization. (See "Bus Locking" in Chapter 8 of the *Intel® 64 and IA-32 Architectures Software Developer's Manual, Volume 3A*, for more information on bus locking.)

The XCHG instruction can also be used instead of the BSWAP instruction for 16-bit operands.

In 64-bit mode, the instruction's default operation size is 32 bits. Using a REX prefix in the form of REX.R permits access to additional registers (R8-R15). Using a REX prefix in the form of REX.W promotes operation to 64 bits. See the summary chart at the beginning of this section for encoding data and limits.

Operation

```
TEMP ← DEST;
DEST ← SRC;
SRC ← TEMP;
```

Flags Affected

None.

Protected Mode Exceptions

#GP(0)	If either operand is in a non-writable segment.
	If a memory operand effective address is outside the CS, DS, ES, FS, or GS segment limit.
	If the DS, ES, FS, or GS register contains a NULL segment selector.