

CMPXCHG--Compare and Exchange

Opcode	Instruction	Description
0F B0/r	CMPXCHG <i>r/m8,r8</i>	Compare AL with <i>r/m8</i> . If equal, ZF is set and <i>r8</i> is loaded into <i>r/m8</i> . Else, clear ZF and load <i>r/m8</i> into AL.
0F B1/r	CMPXCHG <i>r/m16,r16</i>	Compare AX with <i>r/m16</i> . If equal, ZF is set and <i>r16</i> is loaded into <i>r/m16</i> . Else, clear ZF and load <i>r/m16</i> into AL.
0F B1/r	CMPXCHG <i>r/m32,r32</i>	Compare EAX with <i>r/m32</i> . If equal, ZF is set and <i>r32</i> is loaded into <i>r/m32</i> . Else, clear ZF and load <i>r/m32</i> into AL.

Description

Compares the value in the AL, AX, or EAX register (depending on the size of the operand) with the first operand (destination operand). If the two values are equal, the second operand (source operand) is loaded into the destination operand. Otherwise, the destination operand is loaded into the AL, AX, or EAX register.

This instruction can be used with a LOCK prefix to allow the instruction to be executed atomically. To simplify the interface to the processor's bus, the destination operand receives a write cycle without regard to the result of the comparison. The destination operand is written back if the comparison fails; otherwise, the source operand is written into the destination. (The processor never produces a locked read without also producing a locked write.)

Intel(R) Architecture Compatibility

This instruction is not supported on Intel(R) processors earlier than the Intel486™ processors.

Operation

```
(* accumulator ← AL, AX, or EAX, depending on whether *)
(* a byte, word, or doubleword comparison is being performed*)

IF accumulator ← DEST
THEN
ZF ← 1
DEST ← SRC
ELSE
ZF ← 0
accumulator ← DEST
FI;
```

Flags Affected

The ZF flag is set if the values in the destination operand and register AL, AX, or EAX are equal; otherwise it is cleared. The CF, PF, AF, SF, and OF flags are set according to the results of the comparison operation.

Protected Mode Exceptions

#GP(0) - If the destination is located in a nonwritable 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.

#SS(0) - If a memory operand effective address is outside the SS segment limit.

#PF(fault-code) - If a page fault occurs.

#AC(0) - If alignment checking is enabled and an unaligned memory reference is made while the current privilege level is 3.

Real-Address Mode Exceptions

#GP - If a memory operand effective address is outside the CS, DS, ES, FS, or GS segment limit.

#SS - If a memory operand effective address is outside the SS segment limit.

Virtual-8086 Mode Exceptions

#GP(0) - If a memory operand effective address is outside the CS, DS, ES, FS, or GS segment limit.

#SS(0) - If a memory operand effective address is outside the SS segment limit.

#PF(fault-code) - If a page fault occurs.

#AC(0) - If alignment checking is enabled and an unaligned memory reference is made.

*For details, see [Volume 2A](#) and [Volume 2B](#) of the **Intel(R) 64 and IA-32 Intel Architecture Software Developer's Manual**. For the latest updates on the instruction set information, go to [the web site](#).*