CMP—Compare Two Operands

Opcode	Instruction	64-Bit Mode	Compat/ Leg Mode	Description
3C *ib*	CMP AL, *imm8*	Valid	Valid	Compare *imm8* with AL.
3D *iw*	CMP AX, *imm16*	Valid	Valid	Compare *imm16* with AX.
3D *id*	CMP EAX, *imm32*	Valid	Valid	Compare *imm32* with EAX.
REX.W + 3D *id*	CMP RAX, *imm32*	Valid	N.E.	Compare *imm32 sign-extended to 64-bits* with RAX.
80 /7 *ib*	CMP *r/m8, imm8*	Valid	Valid	Compare *imm8* with *r/m8.*
REX + 80 /7 *ib*	CMP *r/m8**, *imm8*	Valid	N.E.	Compare *imm8* with *r/m8.*
81 /7 *iw*	CMP *r/m16, imm16*	Valid	Valid	Compare *imm16* with *r/m16.*
81 /7 *id*	CMP *r/m32, imm32*	Valid	Valid	Compare *imm32* with *r/m32.*
REX.W + 81 /7 *id*	CMP *r/m64, imm32*	Valid	N.E.	Compare *imm32 sign-extended to 64-bits* with *r/m64.*
83 /7 *ib*	CMP *r/m16, imm8*	Valid	Valid	Compare *imm8* with *r/m16.*
83 /7 *ib*	CMP *r/m32, imm8*	Valid	Valid	Compare *imm8* with *r/m32.*
REX.W + 83 /7 *ib*	CMP *r/m64, imm8*	Valid	N.E.	Compare *imm8* with *r/m64.*
38 /*r*	CMP *r/m8, r8*	Valid	Valid	Compare *r8* with *r/m8.*
REX + 38 /*r*	CMP *r/m8**, *r8**	Valid	N.E.	Compare *r8* with *r/m8.*
39 /*r*	CMP *r/m16, r16*	Valid	Valid	Compare *r16* with *r/m16.*
39 /*r*	CMP *r/m32, r32*	Valid	Valid	Compare *r32* with *r/m32.*
REX.W + 39 /*r*	CMP *r/m64,r64*	Valid	N.E.	Compare *r64* with *r/m64.*
3A /*r*	CMP *r8, r/m8*	Valid	Valid	Compare *r/m8* with *r8.*
REX + 3A /*r*	CMP *r8**, *r/m8**	Valid	N.E.	Compare *r/m8 with r8.*
3B /*r*	CMP *r16, r/m16*	Valid	Valid	Compare *r/m16* with *r16.*
3B /*r*	CMP *r32, r/m32*	Valid	Valid	Compare *r/m32* with *r32.*
REX.W + 3B /*r*	CMP *r64, r/m64*	Valid	N.E.	Compare *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.

Description

Compares the first source operand with the second source operand and sets the status flags in the EFLAGS register according to the results. The comparison is performed by subtracting the second operand from the first operand and then setting the status flags in the same manner as the SUB instruction. When an immediate value is used as an operand, it is sign-extended to the length of the first operand.

The condition codes used by the J*cc*, CMOV*cc*, and SET*cc* instructions are based on the results of a CMP instruction. Appendix B, "EFLAGS Condition Codes," in the *Intel® 64 and IA-32 Architectures Software Developer's Manual, Volume 1*, shows the relationship of the status flags and the condition codes.

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

Operation

```
temp ← SRC1 – SignExtend(SRC2);
ModifyStatusFlags; (* Modify status flags in the same manner as the SUB instruction*)
```

Flags Affected

The CF, OF, SF, ZF, AF, and PF flags are set according to the result.

Protected Mode Exceptions

#GP(0)	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.
#UD	If the LOCK prefix is used.

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.
#UD	If the LOCK prefix is used.

Compatibility Mode Exceptions

Same exceptions as in protected mode.

64-Bit Mode Exceptions

#SS(0)	If a memory address referencing the SS segment is in a non-canonical form.
#GP(0)	If the memory address is in a non-canonical form.
#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.
#UD	If the LOCK prefix is used.