

OR—Logical Inclusive OR

Opcode	Instruction	64-Bit Mode	Compat/Leg Mode	Description
0C <i>ib</i>	OR AL, <i>imm8</i>	Valid	Valid	AL OR <i>imm8</i> .
0D <i>iw</i>	OR AX, <i>imm16</i>	Valid	Valid	AX OR <i>imm16</i> .
0D <i>id</i>	OR EAX, <i>imm32</i>	Valid	Valid	EAX OR <i>imm32</i> .
REX.W + 0D <i>id</i>	OR RAX, <i>imm32</i>	Valid	N.E.	RAX OR <i>imm32</i> (sign-extended).
80 /1 <i>ib</i>	OR <i>r/m8</i> , <i>imm8</i>	Valid	Valid	<i>r/m8</i> OR <i>imm8</i> .
REX + 80 /1 <i>ib</i>	OR <i>r/m8*</i> , <i>imm8</i>	Valid	N.E.	<i>r/m8</i> OR <i>imm8</i> .
81 /1 <i>iw</i>	OR <i>r/m16</i> , <i>imm16</i>	Valid	Valid	<i>r/m16</i> OR <i>imm16</i> .
81 /1 <i>id</i>	OR <i>r/m32</i> , <i>imm32</i>	Valid	Valid	<i>r/m32</i> OR <i>imm32</i> .
REX.W + 81 /1 <i>id</i>	OR <i>r/m64</i> , <i>imm32</i>	Valid	N.E.	<i>r/m64</i> OR <i>imm32</i> (sign-extended).
83 /1 <i>ib</i>	OR <i>r/m16</i> , <i>imm8</i>	Valid	Valid	<i>r/m16</i> OR <i>imm8</i> (sign-extended).
83 /1 <i>ib</i>	OR <i>r/m32</i> , <i>imm8</i>	Valid	Valid	<i>r/m32</i> OR <i>imm8</i> (sign-extended).
REX.W + 83 /1 <i>ib</i>	OR <i>r/m64</i> , <i>imm8</i>	Valid	N.E.	<i>r/m64</i> OR <i>imm8</i> (sign-extended).
08 / <i>r</i>	OR <i>r/m8</i> , <i>r8</i>	Valid	Valid	<i>r/m8</i> OR <i>r8</i> .
REX + 08 / <i>r</i>	OR <i>r/m8*</i> , <i>r8*</i>	Valid	N.E.	<i>r/m8</i> OR <i>r8</i> .
09 / <i>r</i>	OR <i>r/m16</i> , <i>r16</i>	Valid	Valid	<i>r/m16</i> OR <i>r16</i> .
09 / <i>r</i>	OR <i>r/m32</i> , <i>r32</i>	Valid	Valid	<i>r/m32</i> OR <i>r32</i> .
REX.W + 09 / <i>r</i>	OR <i>r/m64</i> , <i>r64</i>	Valid	N.E.	<i>r/m64</i> OR <i>r64</i> .
0A / <i>r</i>	OR <i>r8</i> , <i>r/m8</i>	Valid	Valid	<i>r8</i> OR <i>r/m8</i> .
REX + 0A / <i>r</i>	OR <i>r8*</i> , <i>r/m8*</i>	Valid	N.E.	<i>r8</i> OR <i>r/m8</i> .
0B / <i>r</i>	OR <i>r16</i> , <i>r/m16</i>	Valid	Valid	<i>r16</i> OR <i>r/m16</i> .
0B / <i>r</i>	OR <i>r32</i> , <i>r/m32</i>	Valid	Valid	<i>r32</i> OR <i>r/m32</i> .
REX.W + 0B / <i>r</i>	OR <i>r64</i> , <i>r/m64</i>	Valid	N.E.	<i>r64</i> OR <i>r/m64</i> .

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

Performs a bitwise inclusive OR operation between the destination (first) and source (second) operands and stores the result in the destination operand location. The source operand can be an immediate, a register, or a memory location; the destination operand can be a register or a memory location. (However, two memory operands cannot be used in one instruction.) Each bit of the result of the OR instruction is set to 0 if both corresponding bits of the first and second operands are 0; otherwise, each bit is set to 1.

This instruction can be used with a LOCK prefix to allow the instruction to be executed atomically.

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

DEST ← DEST OR SRC;

Flags Affected

The OF and CF flags are cleared; the SF, ZF, and PF flags are set according to the result. The state of the AF flag is undefined.

Protected Mode Exceptions

#GP(0)	If the destination operand points to 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.
#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 but the destination is not a memory operand.

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.

#UD If the LOCK prefix is used but the destination is not a memory operand.

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 but the destination is not a memory operand.

Compatibility Mode Exceptions

Same as for protected mode exceptions.

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 but the destination is not a memory operand.