

MUL—Unsigned Multiply

Opcode	Instruction	64-Bit Mode	Compat/Leg Mode	Description
F6 /4	MUL <i>r/m8</i>	Valid	Valid	Unsigned multiply ($AX \leftarrow AL * r/m8$).
REX + F6 /4	MUL <i>r/m8</i> *	Valid	N.E.	Unsigned multiply ($AX \leftarrow AL * r/m8$).
F7 /4	MUL <i>r/m16</i>	Valid	Valid	Unsigned multiply ($DX:AX \leftarrow AX * r/m16$).
F7 /4	MUL <i>r/m32</i>	Valid	Valid	Unsigned multiply ($EDX:EAX \leftarrow EAX * r/m32$).
REX.W + F7 /4	MUL <i>r/m64</i>	Valid	N.E.	Unsigned multiply ($RDX:RAX \leftarrow RAX * r/m64$).

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 an unsigned multiplication of the first operand (destination operand) and the second operand (source operand) and stores the result in the destination operand. The destination operand is an implied operand located in register AL, AX or EAX (depending on the size of the operand); the source operand is located in a general-purpose register or a memory location. The action of this instruction and the location of the result depends on the opcode and the operand size as shown in Table 3-69.

The result is stored in register AX, register pair DX:AX, or register pair EDX:EAX (depending on the operand size), with the high-order bits of the product contained in register AH, DX, or EDX, respectively. If the high-order bits of the product are 0, the CF and OF flags are cleared; otherwise, the flags are set.

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.

Table 3-69. MUL Results

Operand Size	Source 1	Source 2	Destination
Byte	AL	<i>r/m8</i>	AX
Word	AX	<i>r/m16</i>	DX:AX
Doubleword	EAX	<i>r/m32</i>	EDX:EAX
Quadword	RAX	<i>r/m64</i>	RDX:RAX

Operation

```

IF (Byte operation)
    THEN
         $AX \leftarrow AL * SRC;$ 
    ELSE (* Word or doubleword operation *)
        IF OperandSize = 16
            THEN
                 $DX:AX \leftarrow AX * SRC;$ 
            ELSE IF OperandSize = 32
                THEN  $EDX:EAX \leftarrow EAX * SRC; FI;$ 
            ELSE (* OperandSize = 64 *)
                 $RDX:RAX \leftarrow RAX * SRC;$ 
        FI;
    FI;

```

Flags Affected

The OF and CF flags are set to 0 if the upper half of the result is 0; otherwise, they are set to 1. The SF, ZF, AF, and PF flags are undefined.

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.
#UD	If the LOCK prefix is used.

Virtual-8086 Mode Exceptions

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

INSTRUCTION SET REFERENCE, A-M

#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.