

LEA—Load Effective Address

Opcode	Instruction	64-Bit Mode	Compat/ Leg Mode	Description
8D /r	LEA <i>r16,m</i>	Valid	Valid	Store effective address for <i>m</i> in register <i>r16</i> .
8D /r	LEA <i>r32,m</i>	Valid	Valid	Store effective address for <i>m</i> in register <i>r32</i> .
REX.W + 8D /r	LEA <i>r64,m</i>	Valid	N.E.	Store effective address for <i>m</i> in register <i>r64</i> .

Description

Computes the effective address of the second operand (the source operand) and stores it in the first operand (destination operand). The source operand is a memory address (offset part) specified with one of the processors addressing modes; the destination operand is a general-purpose register. The address-size and operand-size attributes affect the action performed by this instruction, as shown in the following table. The operand-size attribute of the instruction is determined by the chosen register; the address-size attribute is determined by the attribute of the code segment.

Table 3-66. Non-64-bit Mode LEA Operation with Address and Operand Size Attributes

Operand Size	Address Size	Action Performed
16	16	16-bit effective address is calculated and stored in requested 16-bit register destination.
16	32	32-bit effective address is calculated. The lower 16 bits of the address are stored in the requested 16-bit register destination.
32	16	16-bit effective address is calculated. The 16-bit address is zero-extended and stored in the requested 32-bit register destination.
32	32	32-bit effective address is calculated and stored in the requested 32-bit register destination.

Different assemblers may use different algorithms based on the size attribute and symbolic reference of the source operand.

In 64-bit mode, the instruction's destination operand is governed by operand size attribute, the default operand size is 32 bits. Address calculation is governed by address size attribute, the default address size is 64-bits. In 64-bit mode, address size of 16 bits is not encodable. See Table 3-67.

Table 3-67. 64-bit Mode LEA Operation with Address and Operand Size Attributes

Operand Size	Address Size	Action Performed
16	32	32-bit effective address is calculated (using 67H prefix). The lower 16 bits of the address are stored in the requested 16-bit register destination (using 66H prefix).
16	64	64-bit effective address is calculated (default address size). The lower 16 bits of the address are stored in the requested 16-bit register destination (using 66H prefix).
32	32	32-bit effective address is calculated (using 67H prefix) and stored in the requested 32-bit register destination.
32	64	64-bit effective address is calculated (default address size) and the lower 32 bits of the address are stored in the requested 32-bit register destination.
64	32	32-bit effective address is calculated (using 67H prefix), zero-extended to 64-bits, and stored in the requested 64-bit register destination (using REX.W).
64	64	64-bit effective address is calculated (default address size) and all 64-bits of the address are stored in the requested 64-bit register destination (using REX.W).

Operation

```

IF OperandSize = 16 and AddressSize = 16
  THEN
    DEST ← EffectiveAddress(SRC); (* 16-bit address *)
  ELSE IF OperandSize = 16 and AddressSize = 32
    THEN
      temp ← EffectiveAddress(SRC); (* 32-bit address *)
      DEST ← temp[0:15]; (* 16-bit address *)
    FI;
  ELSE IF OperandSize = 32 and AddressSize = 16
    THEN
      temp ← EffectiveAddress(SRC); (* 16-bit address *)
      DEST ← ZeroExtend(temp); (* 32-bit address *)
    FI;
  ELSE IF OperandSize = 32 and AddressSize = 32
    THEN
      DEST ← EffectiveAddress(SRC); (* 32-bit address *)
    FI;
  ELSE IF OperandSize = 16 and AddressSize = 64
    THEN
      temp ← EffectiveAddress(SRC); (* 64-bit address *)

```