

Op Code	R ₁ M ₃	B ₂ DL ₁	DL ₁ DL ₁	DH ₂ DH ₂	Op Code
---------	-------------------------------	--------------------------------	---------------------------------	---------------------------------	---------

Load on condition (**LOC**) loads (or not) a fullword from memory designated by operand 2 into the register designated by operand 1. The fullword is loaded into the register depending on a combination of the mask bits in operand 3, and the current condition code.

The mask field consists of four bits representing conditions equal/zero, low/minus, high/plus, and overflow in sequence from left to right. A 1 bit in the mask indicates the load should occur if the corresponding condition exists. For example, mask B'1000' represents condition equal/zero, while mask B'1010' represents conditions equal/zero or high/plus, and mask B'0111' represents conditions low/minus, high/plus, and overflow. If the current condition code matches any bit in the mask, the fullword is loaded into R1, otherwise no load occurs.

For example, assume the current condition code is high. Consider the following code,

```
LOC      5, X, B' 0111'
```

Since the high bit is indicated in the above mask (along with low/minus and overflow), the load of fullword X into register 5 will occur. The above code is equivalent to the following lines,

```
BE      THERE
L       5, X
THERE   EQU      *
```

Notice that control falls through the BE on conditions low/minus, high/plus, and overflow. As a result, the load occurs on these conditions.

The example above illustrates when to choose LOC over L. If the condition code is set and the load is conditional based on the current value of the condition code, a judicious use of LOC can mean one less branch instruction in your program.

HLASM provides relief from having to code a binary mask by providing suffixes that can be appended to LOC to indicate the conditions under which the load should occur. An earlier version of HLASM provided the following suffixes.

Suffix	Condition	Effective M ₃ Value
E	Equal	B'1000'
L	Low	B'0100'
H	High	B'0010'
NE	Not equal	B'0111'
NL	Not low	B'1011'
NH	Not high	B'1101'

Using these suffixes you can code this,

```
LOCH 3,X
```

instead of this,

```
LOC 3,X,B'0010'
```

The most current version of HLASM provides the following additional suffixes

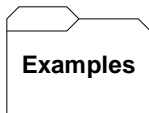
Suffix	Condition	Effective M ₃ Value
Z	Zero	B'1000'
M	Minus	B'0100'
P	Plus	B'0010'
NZ	Not zero	B'0111'
NM	Not minus	B'1011'
NP	Not plus	B'1101'
NO	Not overflow	B'1110'

Below is an assembly listing of LOC instructions using all possible suffixes. The opcode, EBF2, occupies the first and last bytes in the object code. The operand 1 register is 3. The base displacement address C1FE00 (which when rearranged in proper sequence is C001FE) is the base/displacement address of X. The only difference in these instructions occurs in the fourth digit of the object code and corresponds to the suffix appended to LOC.

Loc	Object Code	Addr1	Addr2	Stmt	Source Statement
00002C	EB38 C1FE 00F2		00204	63	LOCE R3,X
000032	EB34 C1FE 00F2		00204	64	LOCL R3,X
000038	EB32 C1FE 00F2		00204	65	LOCH R3,X
00003E	EB37 C1FE 00F2		00204	66	LOCNE R3,X
000044	EB3B C1FE 00F2		00204	67	LOCNL R3,X
00004A	EB3D C1FE 00F2		00204	68	LOCNH R3,X
000050	EB38 C1FE 00F2		00204	69	LOCZ R3,X
000056	EB34 C1FE 00F2		00204	70	LOCM R3,X

00005C	EB32	C1FE	00F2	00204	71	LOCP	R3,X
000062	EB37	C1FE	00F2	00204	72	LOCNZ	R3,X
000068	EB3B	C1FE	00F2	00204	73	LOCNM	R3,X
00006E	EB3D	C1FE	00F2	00204	74	LOCNP	R3,X
000074	EB3E	C1FE	00F2	00204	75	LOCNO	R3,X

There are two binary masks for which there is no corresponding suffix. B'0000' would indicate that the load should never occur, and if no exceptional condition exists, makes the LOC function as a NOP. B'1111' indicates the load should always occur, and if no exceptional condition exists, makes the LOC function as L. In both cases it is better to avoid using LOC.



Assume the following declaration for all the examples below,

```

Y          DC    F'100'

          LOC     R6,Y,B'0010'  LOAD Y INTO R6 IF CC = HIGH/PLUS
          LOCH    R6,Y          EQUIVALENT TO THE INSTRUCTION ABOVE

          LOC     R9,Y,B'1011'  LOAD Y INTO R9 IF CC NOT = LOW/MINUS
          LOCNL   R9,Y          EQUIVALENT TO THE INSTRUCTION ABOVE

          LOC     R7,Y,B'1000'  LOAD Y INTO R7 IF CC = EQUAL/ZERO
          LOCZ    R7,Y          LOAD Y INTO R7 IF CC = EQUAL/ZERO
          LOCE    R7,Y          LOAD Y INTO R7 IF CC = EQUAL/ZERO

          LOC     R7,Y,B'0111'  LOAD Y INTO R7 IF CC NOT = EQUAL/ZERO
          LOCNZ   R7,Y          EQUIVALENT TO THE INSTRUCTION ABOVE

```

Tips

1. Use LOC instead of L in situations where the load occurs conditionally. This avoids coding an additional branch instruction.
2. Avoid the use of masks B'0000' and B'1111' with LOC. Choose NOP and L instead.
3. STOC is the sister instruction to LOC. It works in an analogous way to LOC but stores a fullword in memory instead of loading.
4. Consult the Principles of Operation for situations in which LOC can provide significant performance improvement.