]dec...[.][dec...][exp_char[][dec...]]

, but not both

dec...a required sequence of 1 or more decimal digits

[.]a single optional ª. º

[dec...]an optional sequence of 1 or more decimal digits

[exp_char]an optional exponent delimiter character (see the following table)

The type specification character, flt_char, specifies the type and representation of the constructed r
type specification characters with the processor architecture, as shown here:

Unary minus: the result is the two's complement of the operand

~One's complement: the result is the one's complement of the operand

!Logical negation: the result is 0 if the operand is non-zero, and 1 if the operand is 0

The assembler recognizes the following binary operators:

+Addition: the result is the arithmetic addition of the two operands

Subtraction: the result is the arithmetic subtraction of the two operands

*Multiplication: the result is the arithmetic multiplication of the two operands

/Division: the result is the arithmetic division of the two operands this is integer division, which tr

%Modulus: the result is the remainder that's produced when the first operand is divided by the sec
applies only to integral operands)

>>Right shift: the result is the value of the first operand shifted to the right, where the second oper
number of bit positions by which the first operand is to be shifted (this operator appli
operands). This is always an arithmetic shift since all operators operate on signed op

<<Left shift: the result is the value of the first operand shifted to the left, where the second operan
of bit positions by which the first operand is to be shifted (this operator applies only t

&Bitwise AND: the result is the bitwise AND function of the two operands (this operator applies
operands)

^Bitwise exclusive OR: the result is the bitwise exclusive OR function of the two operands (this o
integral operands)

|Bitwise inclusive OR: the result is the bitwise inclusive OR function of the two operands (this op
integral operands) this operator can't be used on the M68000 microprocessor family,
character is used there to mark the start of a comment

<Less than: the result is 1 if the first operand is less than the second operand, and 0 otherwise

>Greater than: the result is 1 if the first operand is greater than the second operand, and 0 otherwis

(x * y + z).

·A term preceded by a unary operator (for example, ~var). Multiple unary operators may be used i
!~var).

varis not relocatable. The expression can't be linked by adding var's offset to it.

var+dat+5is a relocatable expression if both var and dat are both defined in some sectionÐthat is,
This form of relocatable expression is used for position-independent code and is supp
Release 3.3 and later.