LOG10 S73
LOGE S74
EXP S84
POWER T70
FSCALE T73

Special Functions for Powers & Logarithms

1. Description

The Special Functions described in this leaflet provide the following arithmetic operations:-

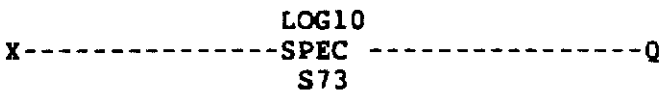
- Taking logarithms to base 10.
- Taking logarithms to base e.
- Raising a value to a specified power.
- Raising e to a specified power.
- Multiplying a value by a specified power of 10.

All these arithmetic functions are available only on GEM80 Controllers with floating point mathematics facilities. Their rung outputs are always floating point numbers, but their inputs may be either floating point or integer.

2. Specifications

LOG10 (S73)

Control language:



Equation:

$Q = \log_{10}(X)$

where:

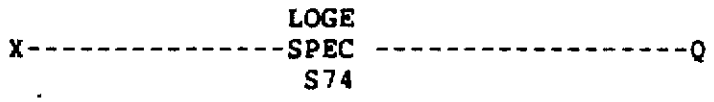
X = rung input, either integer or floating point, must be positive.

Q = rung output (floating point, up to the maximum values that correspond to the maximum and minimum representable values of the rung input X)

If X is negative, the output Q will be of a form that does not represent a valid number, and will not be usable in any subsequent calculations. This error can be detected using the S78 Special Function FPFLAGS; refer to the Software Date Sheet of this function for details.

LOGE (S74)

Control language:



Equation:

$Q = \log_e (X)$

where:

X = rung input, either integer or floating point, must be positive.

Q = rung output (floating point, up to ± the maximum values that correspond to the maximum and minimum representable values of the rung input X)

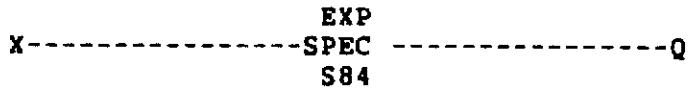
e = the base for natural logarithms

= 2.7182818285...

If X is negative, the output Q will be of a form that does not represent a valid number, and will not be usable in any subsequent calculations. This error can be detected using the S78 Special Function FPFLAGS; refer to the Software Data Sheet of this function for details.

EXP (S84)

Control language:



Equation:

$Q = e^X$

where:

X = rung input (either integer or floating point, up to ± the maximum representable number of either type)

Q = rung output (floating point, always positive, up to the maximum representable floating point number)

Notes:

1. If you output Q directly to Q-table and X lies outside the range of approximately +88.7 to -87.3, the message 'not a number' is displayed against the appropriate Q-table in the data table list.

   When X reaches about +88.7, Q will reach the maximum floating point number that can be stored in Q-table.

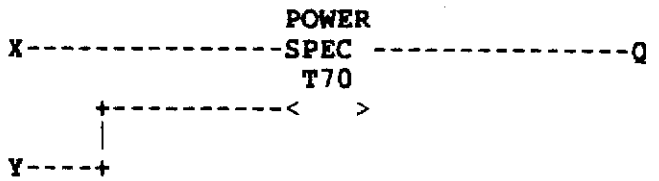
   When X reaches about -87.3, Q will reach the minimum non-zero floating point number that can be stored in Q-table.

2. Where Q is not output directly to Q-table but follows the EXP function with further floating point functions in the same rung of the ladder diagram, X may range up to around ±11,000 before overflow or underflow occurs on the EXP output. It is still possible to get overflow or underflow, however, when the end of the rung is reached and the value is output to Q-table.

3. For the detection of underflow and overflow errors, see the Software Data Sheet for the S78 Special Function FPFLAGS.

POWER (T70)

Control language:



Equation:

$Q = X^Y$

where:

X = rung input, upper branch (either integer or floating point, up to ± the maximum representable number of either type)

Y = rung input, lower branch (either integer or floating point, up to ± the maximum representable number of either type)

Q = rung output (floating point, up to the maximum representable floating point number)

Notes:

1. If Q is output directly to Q-table, it is quite easy to get overflow or underflow if Y is allowed to be too large a positive or negative number respectively.

   For instance, if X = 2, Q will reach the maximum floating point number that can be stored in Q-table if Y = 128; any further increase in Y does not make any difference to the stored value.

   Similarly, if X = 2, Q will reach the minimum non-zero floating point number that can be stored in Q-table if Y = 120; making X any more negative makes the stored value become zero.

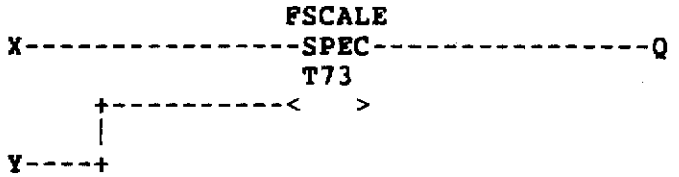
2. Where Q is not output directly to Q-table but follows the POWER function with further floating point functions in the same rung of the ladder diagram, then, for X = 2, Y may range up to around +16,000 before overflow or underflow occurs on the POWER output. It is still possible to get overflow or underflow, however when the end of the rung is reached and the value is output to Q-table.

3. The X input should generally be positive. It may be negative only if Y is an integer. If a calculation is tried where X is negative and Y is non-integer, the output Q will be of a form that does not represent a valid number, and will not be usable in any subsequent calculations. The calculation is categorized as an invalid operation.

4. To detect underflow, overflow and invalid operation errors, use the S78 Special Function FPFLAGS; refer to the Software Data Sheet of this function for details.

FSCALE (T73)

Control language:



Equation:

$Q = X \times 10^{Y}$

where:

X = rung input, upper branch (either integer or floating point, up to ± the maximum representable number of either type)

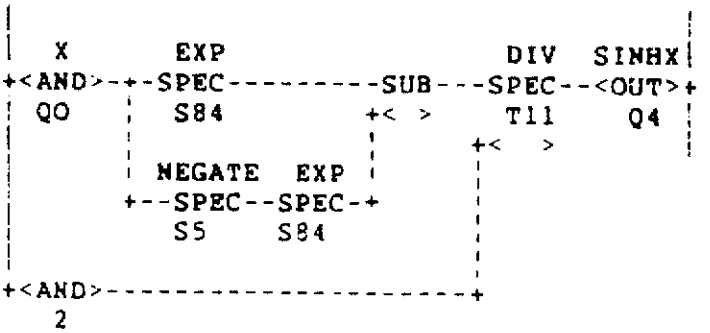
Y = rung input, lower branch (must be an integer, up to ± the maximum representable integer)

Q = rung output (floating point, up to the maximum representable floating point number)

If Y is not an integer, a compilation error is given and the program will not run.

3. Application Notes

EXAMPLE 1

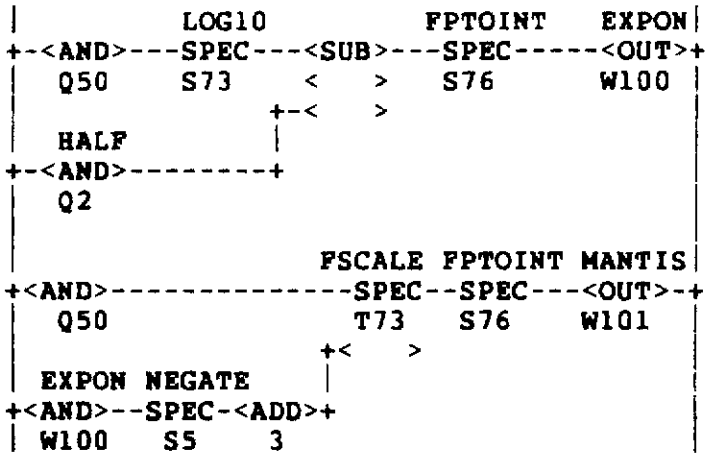


$\sinh x = \frac{e^{x}-e^{-x}}{2}$

The above rung produces a floating point number that is the hyperbolic sine of the input using the above formula. Wherever necessary, the rung input Q0 should be limited to prevent overflow or underflow using the functions T7/T8 (MAX/MIN).

EXAMPLE 2

If it is required to convert a floating point number in table location Q50 into two separate integers in W-table, one giving the value (the mantissa) and the other the power-of-10 multiplier (the exponent), it could be worked out as follows:-



In this example, the value to be converted is held in table locations Q50 and Q51.

First, find the power of 10 by taking logs to base 10. This gives a number that will generally include a fractional part. To remove the fractional part, use the floating point to integer conversion function FPTOINT. However, this function rounds to the nearest integer. If the fractional part were greater than 0.5, it would give the next integer up. To prevent this happening, ensure that the answer is rounded down by first subtracting 0.5 (held in Q2 and Q3) before the conversion. The output in W100 is now the power of 10 required.

In the second rung, the input number is scaled by a power of 10 to bring it into the normal integer range. If it is scaled by the power of 10 found in the first rung, the result will be in the range 1.00000-9.99999. When converted to an integer, the answer would be an integer in the range 1-10. Therefore increase the power of 10 by three (the ADD 3 term), to give an answer in the range 1,000.00 to 9,999.00. When rounded, this will give an integer answer in the range 1,000 to 10,000, which can be stored to better than 0.1% accuracy in W101.

If it is ever required to print, or to display, the contents of W101, on a VDU it must be remembered that it represents the actual value multiplied by a thousand.

CEGELEC
INDUSTRIAL CONTROLS

Kidsgrove, Stoke-on-Trent, ST7 1TW, England
Tel: (0782) 783511 Fax: (0782) 776329 Telex: 36293/4 CICKID G
Registered in England No. 2259095