CEGELEC
INDUSTRIAL CONTROLS

Special Functions for Multiplication and Division of Variables

MULT T10
DIV T11

DESCRIPTION

The MULT and DIV special functions perform multiplication and division of two variables. Their outputs will be correctly signed (in other words, they give four quadrant operation).

If your GEM80 controller has floating point number facilities, then MULT and DIV can operate on either integer or floating point numbers. If your controller can only handle integers, then obviously MULT and DIV will operate in integer only.

SPECIFICATION

MULT (T10)

Control language:-

```
         MULT
X--------SPEC--------------------Q
         T10
      +-<   >
      |
Y-----+
```

Equation: $Q = X \times Y$

where:

X	=	rung input, upper branch. May be either integer or floating point.
Y	=	rung input, lower branch. May be either integer or floating point.
Q	=	rung output. If both X and Y are integer, then Q will be integer. If either X or Y is floating point, then Q will be floating point.

Notes:

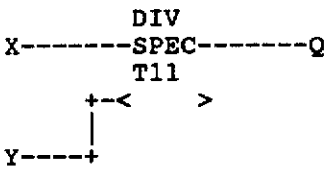
1. Unlike some other special functions, MULT requires no VALUE after it. If you write a VALUE after it in your program, then, on newer models of GEM80 controller, you will get a compiler error and your program won't run. On older models of GEM80 controller that store only one version of program and have no compiler (for instance GEM80/130's), a VALUE after MULT will cause the rung output Q to be held at zero.

2. When both X and Y are integers and Q is therefore an integer, Q is limited to the range ±32767.

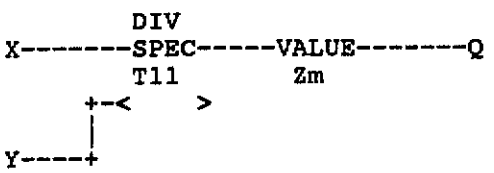
3. When Q is floating point, you can get either overflow or underflow. With overflow, the result could be too large a number to be represented, in which case it will be limited to the maximum representable number. With underflow, the result could be too small a number to be represented, in which case Q will be set to zero. You can detect either of these occurences using the FPFLAGS special function S78; refer to Software Data Sheet T1305 for this function.

DIV (T11)

Control Language (floating point)



Control Language (integer)



Equation: Q = X/Y (floating point)
Q = (X - R)/Y (integer)

Where:

X = rung input, upper branch. May be either integer or floating point.
Y = rung input, lower branch. May be either integer or floating point.
Q = rung output. If both X and Y are integer, then Q will also be integer. If either X or Y is floating point, then Q will be floating point.
R = remainder. Only used when both X and Y are integer. Not used with floating point. The sign of R is the same as the sign of the X input

Notes:

1. DIV requires a VALUE after it only when both rung inputs are integer, as shown above.

2. On newer models of GEM80 controller that support floating point numbers, you will get a compiler error if you include a VALUE when it is not needed (on floating point division), or omit the VALUE when it is needed (on integer division), and your program won't run.

   On newer models of controller that only support integer numbers, you must always include the VALUE, and you will get a compilation error if you omit it.

   On older models of controller that store only one version of program and have no compiler (for instance, GEM80/130's), omitting the VALUE will cause the rung output to be held at zero.

3. When you use integer division (that is, both X and Y are integers and Q is therefore an integer), Q is limited to the range ±32767, and R is limited to the range ±32766.

4. With integer division, the number below the VALUE, shown above as Zm, must be the address of a table location, e.g. W100. On newer models of controller, if you key in a number instead of an address, or the address of a table location in write-protected memory, or a table not usable by Special Functions, then you will get a compiler error and your program won't run. On older models of controller that have no compiler, the rung output Q will be held at zero.

5. If Y is zero with integer division, both the rung output Q and the remainder R will be set to zero.

   If Y is zero with floating point division, then, if X is not also zero, Q will be set to the largest representable number. In floating point representation, zero has a sign (+0 and -0 are represented differently), so the sign of Q will follow the normal rules for division.

If, with floating point division, Y is zero and X is also zero, then the result is indeterminate. Q will be set to a value that does not represent a valid number, and this value will not be usable in further calculations. The mathematics co-processor flags this up as an invalid operation.

6. When Q is floating, you can get either overflow or underflow. With overflow, the result could be too large a number to be represented, in which case it will be limited to the maximum representable number. With underflow, the result could be too small a number to be represented, in which case Q will be set to zero. You can detect either of these occurences using the FPFLAGS special function S78; refer to Software Data Sheet T1305 for this function.

Examples of the Results of Integer Division:

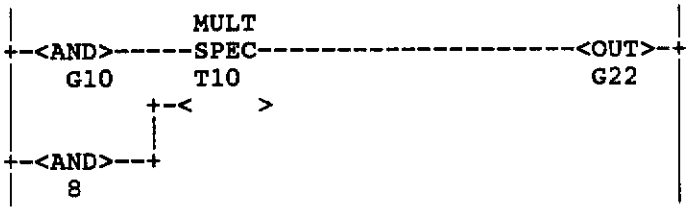
Dividend	Divisor	Quotient	Remainder
X	Y	Q	R
17	3	5	2
17	18	0	17
17	-3	-5	2
-17	3	-5	-2
-17	-3	5	-2

Examples of the Results of Floating Point Division for the same Inputs:

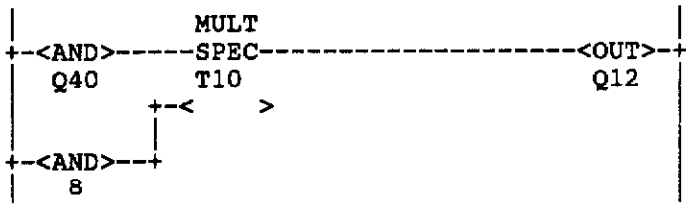
X	Y	Q
17	3	5.66667
17	18	9.44444×10^{-1}
17	-3	- 5.66667
-17	3	- 5.66667
-17	-3	5.66667

APPLICATION NOTES

Multiplication by a constant

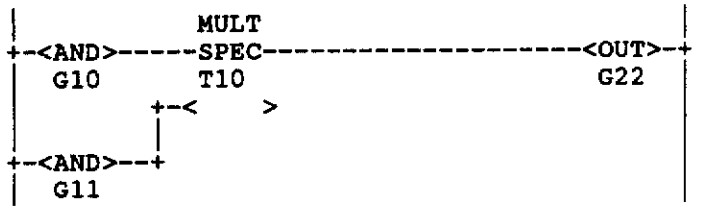


The above rung makes the contents of G22 equal to 8 times the contents of G10 (subject to the result being within the integer number range, otherwise it will be limited to ±32767)

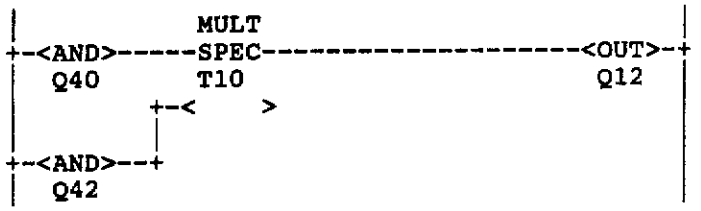


The above rung is virtually the same as the previous one, except that it deals with floating point numbers. It makes the contents of Q12 and Q13 equal to 8 times the contents of Q40 and Q41 (subject to the result being within the floating point number range, otherwise it will be limited to approximately ±3.4 x 1038).

Multiplication by a variable

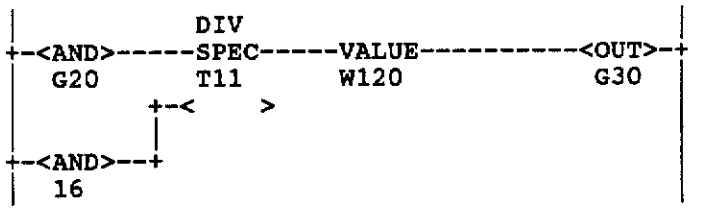


The above rung makes the contents of G22 equal to G11 times the contents of G10 (subject to the result being within the integer number range, otherwise it will be limited to ±32767).

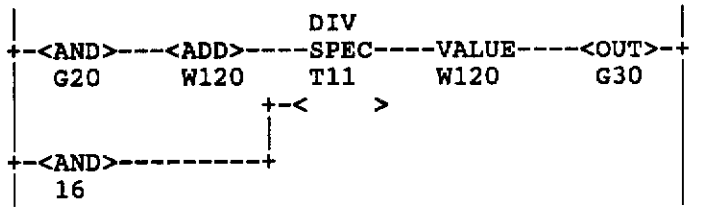


The above rung is virtually the same as the previous one, except that it deals with floating point numbers. It makes the contents of Q12 and Q13 equal to Q42 and Q43 times the contents of Q40 and Q41 (subject to the result being within the floating point number range, otherwise it will be limited to approximately $\pm 3.4 \times 10^{38}$).

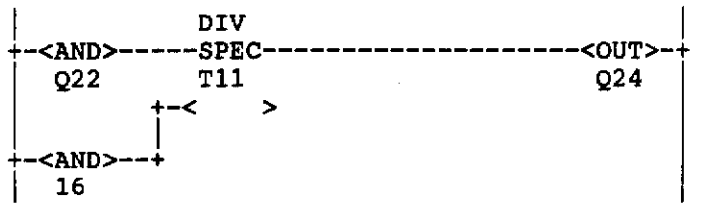
Division by a constant



The above rung performs integer division, and therefore the DIV special function has to be followed by a VALUE. In this rung, the contents of G20 are divided by 16, the result placed in G30, and the remainder placed in W120.

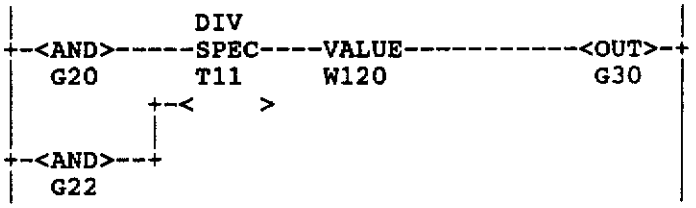


The above rung is a modification of the previous one, where the remainder in W120 calculated on the previous execution of the program is added to the input G20 prior to the division. The DIV function will then recalculate the remainder and place its new value in W120. This method of adding back the remainder avoids cumulative errors with the integer arithmetic in any applications that involve totalisation.

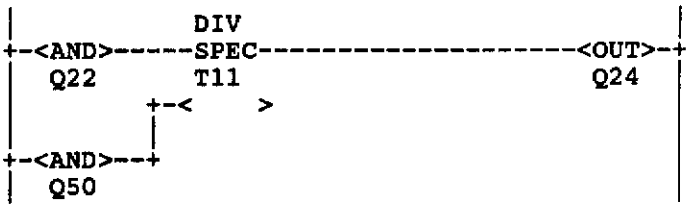


The above rung performs floating point division, and therefore the DIV special function must not have a VALUE after it, as there is no remainder to store. In this rung, the contents of Q22 & Q23 are divided by 16, and the result placed in Q24 & Q25.

Division by a variable



The above rung performs integer division, and therefore the DIV special function has to be followed by a VALUE. In this rung, the contents of G20 are divided by the contents of G22, the result placed in G30, and the remainder placed in W120.



The above rung performs floating point division, and therefore the DIV special function must not have a VALUE after it, as there is no remainder to store. In this rung, the contents of Q22 and Q23 are divided by the contents of Q50 and Q51, and the result placed in Q24 and Q25.

CEGELEC
INDUSTRIAL CONTROLS

Kidsgrove, Stoke-on-Trent, ST7 1TW, England
Tel: (0782) 783511 Fax: (0782) 776329

Publication No. T1307 Issue 7