CEGELEC
INDUSTRIAL CONTROLS

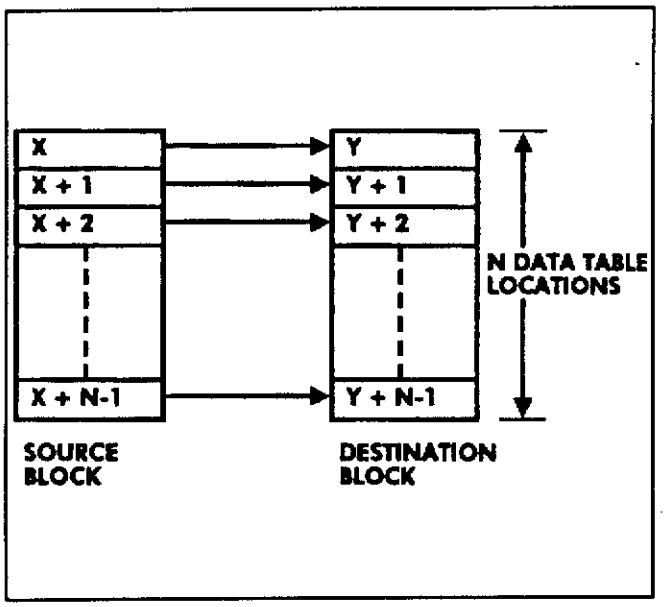
Move Function

MOVE T20

DESCRIPTION

The pair of functions LOCATE and MOVE can be used to copy the data from one block of data table locations into a different block of locations.

The MOVE function transfers a defined number of data items from the source table to the destination table.



FEATURES

- Source and destination locations specified by user (LOCATE function S20)
- User programmable data block size
- Copies data between different data tables, or can be used to shift data within the same table.

SPECIFICATION

Control language:-

a) Fixed block size (N locations)

```
         MOVE
X--------SPEC--------VALUE--------Q
         T20           N
       +-<  >
       |
Y------+
```

b) Variable block size (location Zm contains block size N)

```
         MOVE
X--------SPEC--------VALUE--------Q
         T20           Zm
       +-<  >
       |
Y------+
```

Table 1 **Program rung & value data**

Sym	Location	Use	Range
X	Rung Input (upper branch)	Internal GEM80 address of first source location	Addresses of data table locations as obtained by LOCATE function. Destination must not be write protected.
Y	Rung Input (lower branch)	Internal GEM80 address of first destination location	
N	Value integer	Block size	0 to 9999
Zm	Value location	Block size N	Any table usable by special functions
Q	Rung output	Fault code (5 locations)	See table 2 (Zero = no fault)

Table 2 **Fault codes and flags**

Conditions	Rung output	
	Numeric Value	States ON
Programming error. No VALUE after SPEC	1	0
Programming error or block size N too large. Invalid location in source block: X + N beyond end of data tables memory area	3	0 & 1
Programming error or block size N too large. Invalid location in destination block: a) Y in write protected memory b) Y + N beyond end of data tables memory area	5	0 & 2
Fault detected by LOCATE function used for source location i.e. X = −1	9	0 & 3
Fault detected by LOCATE function used for destination location i.e. Y = −1	17	0 & 4
Block size N<0 or >9999	33	0 & 5

No data transfer occurs if any of above conditions occurs.

APPLICATION NOTES

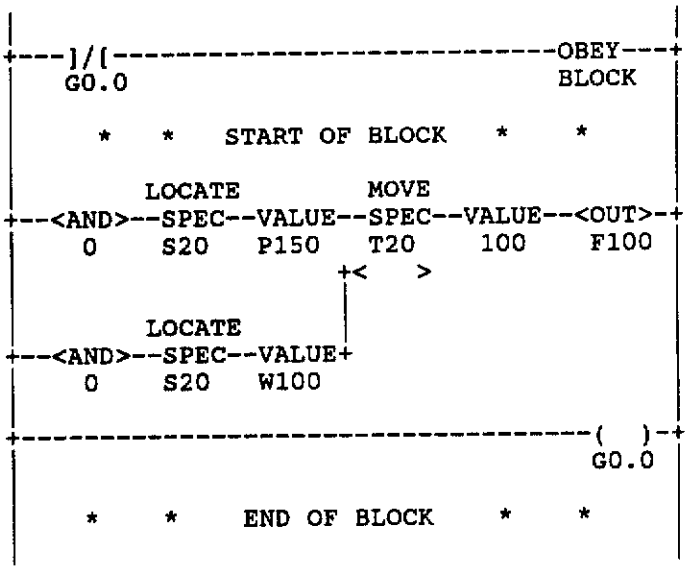
1. POTENTIAL MISUSE

Since the MOVE function is intended for transferring large quantities of data, the effect of a programming mistake can be to corrupt a large area of the data tables. The user is advised to take particular care to prove that MOVE rungs work as intended.

A typical GEM80 controller has data tables A,B,C,D,G,J,K,L,M and W as a contiguous block in alphabetical order in memory (E,F,R,P in 3 separate blocks). If the user attempts to MOVE too many locations, the effect will be to run over the adjacent data tables. A fault code will be generated only if the block extends beyond the space allocated to the whole group of tables. The use of a fixed block size is preferable whenever practical, since this reduces the chance of a mistake elsewhere in the user program causing MOVE to corrupt data tables.

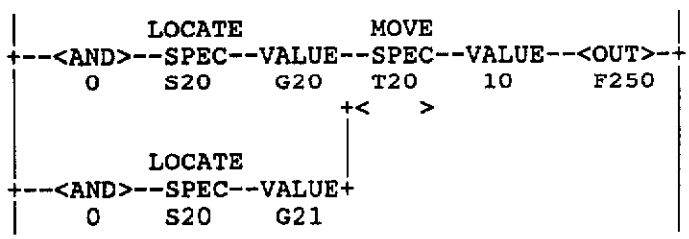
2. EXAMPLE OF COPYING BETWEEN TABLES

Typical application of LOCATE and MOVE for special function "value" data. Preset "value" data. Preset "values" in P table (P150 to P249) are to be copied to working W table (W100 to W199) on start up. G0.0 is set OFF automatically on start up, so this is used as condition for executing MOVE rung.



3. EXAMPLE OF SHIFTING DATA IN SAME TABLE

The MOVE function includes checks to ensure satisfactory execution if the source and destination blocks overlap, i.e. MOVE may be used to shift a block of data up or down a table. A typical application is in history recording, and the example below shows a rung for storing ten previous values of variable G20 in locations G21 to G31.



CEGELEC
INDUSTRIAL CONTROLS

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

Publication No. T323 Issue 4