

## DESCRIPTION

The CMPGRP function treats a group of adjacent data table locations as a one dimensional array of bits. This block of tables is referred to as a GROUP. The elements within an Array are referenced by a bit number, commencing at bit 0.

The CMPGRP function compares its two input Groups and returns the bit position of the first difference detected and the state of this bit in the first input group. The comparison can commence at bit positions other than the beginning of the Group so that all differences can be detected in successive scans.

## FEATURES

- Input Group locations specified by the user
- User programmable Group length
- User programmable bit start position
- Indication when input Groups are identical

## SPECIFICATION

Control language:-

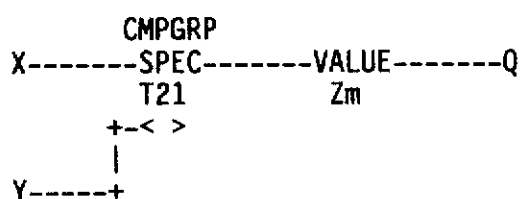


Table 1 Program rung data

| Sym | Location                  | Use                                        | Range                                                |
|-----|---------------------------|--------------------------------------------|------------------------------------------------------|
| X   | Rung input (upper branch) | Internal GEM 80 address of 1st input Group | Addresses of data table locations obtained by LOCATE |
| Y   | Rung input (lower branch) | Internal GEM 80 address of 2nd input Group |                                                      |
| Zm  | Value location            | Start of parameter list (3 locations)      | Any write-enabled table usable by specials           |
| Q   | Rung output               | Result of comparison                       | See 'Output Data' and table 3                        |

Table 2 Value data

| Sym | Location | Use                                  | Range                                                                                                  |
|-----|----------|--------------------------------------|--------------------------------------------------------------------------------------------------------|
| EC  | Zm       | Error code or bit output             | See table 3                                                                                            |
| GL  | Zm + 1   | Group length                         | 1 to 128 data table locations                                                                          |
| BP  | Zm + 2   | Bit position for start of comparison | 0 to (GL*16-1)<br>Updated on exit with the position after the 1st difference detected (0 if identical) |

## Output Data

As long as there are no programming faults then the value of Q reflects the result of the comparison.

If the groups are identical from the starting position (Zm + 2) to the end of the group, then Q = -1.

If the groups differ, then Q = bit number where first difference detected, and bit 15 of Zm will be set to the same state as this bit in the first input group (the input group defined by the X input).

With no programming faults, Zm will therefore have the value either 0 or @8000 (~32768) depending on whether bit 15 is 0 or 1.

Table 3 Fault conditions

| Condition                                                                                                                                                          | Result Q | Code in Zm<br>Numeric Value | States ON |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|-----------------------------|-----------|
| Programming error.<br>Group length invalid.                                                                                                                        | 0        | 3                           | 0 & 1     |
| Programming error.<br>Invalid Group address:-<br>(a) 1st input Group not in table usable by special.<br>(b) 1st input Group extends beyond end of data table area. | 0        | 5                           | 0 & 2     |
| Programming error.<br>Invalid Group address:-<br>(a) 2nd input Group not in table usable by special.<br>(b) 2nd input Group extends beyond end of data table area. | 0        | 9                           | 0 & 3     |

Table 3 (continued)

| Condition                                                 | Result Q | Code in Zm<br>Numeric Value | States ON |
|-----------------------------------------------------------|----------|-----------------------------|-----------|
| Programming error.<br>Start bit position not within Group | 0        | 33                          | 0 & 5     |

## APPLICATION NOTES

Comparison of two groups defined by G100-G102 and G110-G112.

```

LOCATE      CMPGRP
-<AND>--SPEC--VALUE---SPEC--VALUE--(OUT)
  0   S20   G100   T21   G0   G10
                        +<   >
LOCATE      |
-<AND>--SPEC--VALUE--+
  0   S20   G110

```

Initially G1 = 3, G2 = 0

G100 = @0002, G101 = @0000, G102 = @5A00

G110 = @0000, G111 = @8000, G112 = @5A00

After the first scan of the program the outputs would be:-

G0 = @8000, G10 = 1, (G100.1 ≠ G110.1)

G2 = 2

After the second scan the outputs would be:-

G0 = 0, G10 = 31, (G101.15 ≠ G111.15)

G2 = 32

After the third scan:-

G0 = 0, G10 = -1 (remainder of groups identical)

G2 = 0





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

Note: The Company's policy is one of continuous development. Products supplied may differ from those described herein.

©1991 CEGELEC CONTROLS Ltd