

GENERAL INTERPRETIVE PROGRAM 5

DESCRIPTION

The programme (GIP 5) gives a simple method of linking together several other programmes so that sequences of operations can easily be performed on the quantities in store. This is achieved by a system of 3-address orders given in a special code.

The orders use a section of store 96 words long, and this store is in effect treated as part of a distinct 3-address computer, the orders being interpreted by the main programme GIP 5. This "computer" has its own rules, described below, and it is not necessary for the user to be an expert DEUCE programmer to use the scheme.

The orders can call down any programme from the drum or can carry out operations on their own special store, it being possible to carry out repeated cycles of operations with them. There are several discrimination and instruction modification facilities, and these are rather more flexible than the current DEUCE ones.

Finally, single DEUCE instructions can be included amongst the orders. These are obeyed with absolute timing; the Wait No. specifies the mc of transfer.

The GIP 5 programme itself is a single pack of 46 cards with no breaks. It is followed by a data card with NxP17 on the Y row, and the N programmes required. It then reads in a set of 32 orders into the "computer" stores 0 to 31, and starts by obeying the order in store 0. The orders are punched on 3 cards starting on row 2 of the first card.

Each order consists of 4 parts: a (P1-P8); b (P9-P16); c (P17-P24); r (P25-P30); together with P31, the presence of which identifies a DEUCE instruction, and P32, used to modify meaning of r when $r > 32$. Except where $r = 33, 34, 35, 36, 37, 41, 42, 43, 45, 46, 47$ the orders are obeyed consecutively.

In the case of simple programmes for which no addresses need be given, the whole order except c is left blank, and programme number c is called down and obeyed. (In calling down such simple programmes nothing is inserted in QS18). Programmes that require 3 addresses (e.g. Scheme B) are called by using the full order code, a, b, c, r. Programme r is called down, and has a, b and c each supplied to it xP5 via QS18. Such programmes are also given rxP17 in the spare store of QS18 for reference purposes. Programmes find in DL's and TS's what was left by previous programme except DL1; DL11; TS13; and QS18 (when $r = 0$, QS18 also preserved, as noted above). In particular, except in DL's 1-6, 11, TS13 and QS18, the first programme called will find what was left by any bricks passed through the reader before GIP 5 was inserted.

The total number of programmes that can be stored is 62; where the number r of the programme being called down is greater than 31, it is necessary to supply a P32 to the order. With 21 tracks occupied by GIP 5, all the remaining 235 tracks are available for storage of programmes and numerical data.

With no P32 and values of r greater than 31, an order is interpreted in a special way, and when $r > 32$ the addresses a, b and c refer to the 96 words of order store. The facilities provided by such instructions are listed on the reverse of this sheet. Some stores with addresses greater than 95 are used for special purposes, described here.

Stores 120-127 are special in that they are arranged to contain permanently certain useful constants. They contain respectively zero; P1; P9; P1 + P9; P17; P1 + P17; P9 + P17; P1 + P9 + P17. Stores 96-99 contain whatever was left in QS17 by the last programme called.

Any order with a P31, regardless of r, is obeyed as a DEUCE instruction with absolute timing (i.e. the Wait number specifies the mc of transfer), and any sources or destinations may be used so long as they do not destroy the main programme itself in DL's 1-8 and 12. During a succession of such instructions no TS's, DS's or QS's are interfered with by the main programme. Where there is no P31 obeying an order with $r > 32$ and no P32 will destroy TS15, TS16 and DS21, and obeying one with $r \leq 32$, or with a P32, will destroy all such stores except TS13 and TS14, which are preserved in stores 110 and 111 respectively.

A P32 on the ID will make all orders except those called by function 46 stoppers, i.e. the order about to be obeyed will appear on the OPS, the machine stopping at 1-1X. If an order has its address xP17 on the ID, this order only becomes a stopper and remains so until the ID is altered. A one-shot will make the machine obey the order on the OPS when the TIL key is normal. With the TIL key down the order on the OPS will be replaced by whatever is on the ID. A further one-shot with TIL off will make the machine obey the new order. An order with a P32 cannot be inserted in this way. With a P32 on the ID and TIL on, a one-shot will have no effect whatsoever, the OPS remaining unchanged, and the machine remaining at 1-1X.

DEUCE type orders normally have NIS and timing numbers zero, but putting a timing number of 31 will make an order return to itself in a closed loop, unless there is a successful discrimination (destination 27 or 28), when the next order will be obeyed normally. This is useful when looking for TIL with a 2-24X instruction, as the time between consecutive DEUCE orders is about 8 major cycles.

A successful discrimination with destination 27 or 28 and zero timing number will cause a jump to the next but one order, as also will any other DEUCE type order with timing number one.

Any completely blank order is treated as a waste-time, and has no effect on any store location.

LIST OF FUNCTIONS

- r = 32 a \neq 0; b = 0; c \neq 0. Replace the c programmes a to (a+c-1) by programmes in the reader.
 b = 0; c = 0. Read NxP17 on data card then carry out preceding operation with a = 1, b = 0, c = N.
 b \neq 0. Pass one programme through the reader and obey it. (a, b \neq 0 and c are available in QS18).
- 33 Jump to order c.
34 Jump to c if A negative, otherwise jump to b (where A is contents of store a).
35 Jump to c if A non-zero, otherwise jump to b.
36 Jump to c if B > A, otherwise continue normally.
37 Jump to c if B \neq A, otherwise continue normally.
38 A + B \rightarrow C.
39 A - B \rightarrow C.
40 A \rightarrow C.
41 Obey A + B, add P1 to B, jump to c.
42 Obey A + B, add P9 to B, jump to c.
43 Obey A + B, add P17 to B, jump to c.
44 Obey A + B, add C to B, continue normally.
45 Obey A + B, jump to c
46 Obey the orders in stores a to b inclusive, then jump to c.
47 Read a set of 32 orders into store 32c onwards (where c = 0,1 or 2) jump to b.
48 Put a, b, c in QS17 for next programme called (auxiliary codes).