MATHEMATICS DEPARTMENT.

DEUCE PROGRAMME NEWS - No. 29. November, 1958.

1. 64 COLUMN DL FILLING INSTRUCTIONS.(D.J.Ozanne, E.E., N.R.L., Stafford).

There are 3 ways of reading triads into the DL's with the reader on 64 column.

(i) Using ZP39T (No. 358).

(ii) Reading only the α -field with filling instructions as for 32 column operation.

(iii) Using special fillers such as the set below. These assume that 1_0 is blank and the higher numbered DL is punched in the α -field. while the x-row is being read.

For 2 NIS DL's, calling N_T at exit.

```
Y
X  1, 0 - α 1 27 26 GO      β, 0 - α 30 30 X
0  α, 0 - β 1 28 27 GO      α, 0 - β 30 31 GO
1  β, 0 - α    30 30 X      N, 0 - β 30 (T-1)GO
```

For 2 non-NIS, or for 1 NIS and 1 non-NIS DL, using buffer DL's for both, calling N_T at exit.

```
Y
X  1, 0 - A 1 27 26 GO      B, 0 - α 30 30 X
0  1, 0 - B 1 29 28 GO      A, 0 - β 30 31 GO
1  A, 0 - B    27 28 GO      N, 0 - β 30 (T-1)GO
```

For this set $A = \alpha$, $B = \beta$ are allowable if

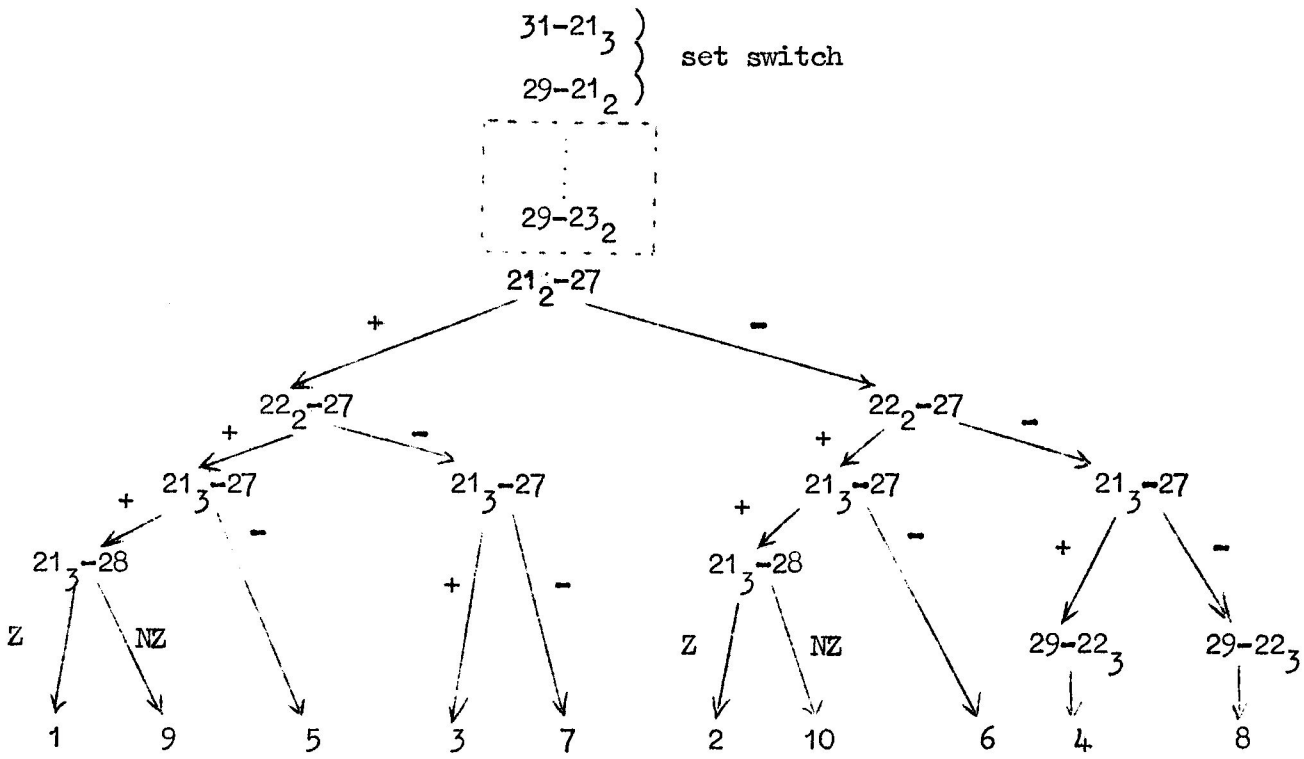
$A \neq 1$, $B \neq 1$

A, B are the buffer DL's for α , β respectively.

2. PROGRAMMING DEVICES.

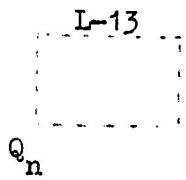
(a) The following are three methods of doing more or less the same thing, namely dividing a programme after a subroutine. (i) is the least efficient as far as instructions are concerned but may provoke ideas. (ii) is the universally used system, while (iii) is the most efficient for a large number of entries. The dotted rectangle represents the subroutine.

(i) (J. Boothroyd, E.E., N.R.L. Stafford.)



Three instructions are used to "change the points", a common 29-23$_2$ within the routine from which the several exits are required and two 29-22$_3$ instructions on the exit branches 4 and 8 . TCB is assumed to be OFF. 14 instructions for 10 exit paths.

(ii) (Author not known).



A different link is planted each time

1 instruction per exit path.

(iii) (E.N. Hawkins, E.E., N.R.L. Stafford.)

The same link, if any, is planted each time

P$_{26}$-25

1$_{28}$ 13-0

Q30(13-I)

the exit points being in consecutive m.c's of a DL. 5 instructions (inc. 13-0) for up to 32 exit paths.

(b) (R.A.Smith, E.E., N.R.L. Stafford.)

In DEUCE News No. 8 paragraph 8(ii) there appeared a device which enables two instructions to be saved on the sign correction to be applied to the result of a multiplication. A number of programmers either have not seen this or do not realise its full possibilities. There is in fact no need for the discrimination to come outside the multiplication time as shown in the paragraph in DEUCE News 8. Thus in place of the sign correction given in the Programming Manual we have the following:

"a"-21_3

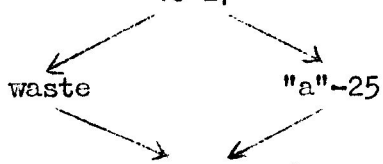
"b"-16

0-21_2

0-24

21_2-13 (in first possible m.c. after stim.mult.)

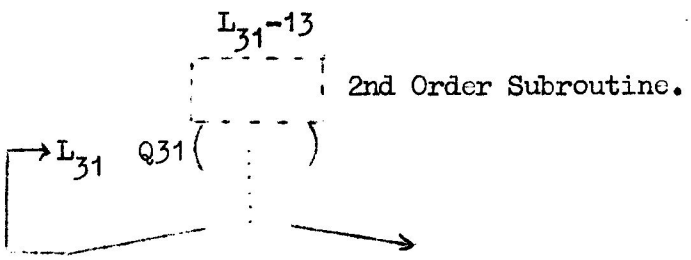
16-27



13-23_3 (when mult. has finished.)

(c) (Author not known).

It is sometimes necessary to re-enter a programme at the link of a subroutine. This can be done without loss of instructions if the link is obeyed from the same m.c. as it is stored in. e.g.



3. NOTATIONS AND DEFINITIONS.

(a) Annotating Flow Diagrams. (P.J. Winnall, E.E., Bradford.)

Most people have their own ideas about how to lay out instructions and these vary according to the problem and after some experience has been gained practically However, it improves the readability of a flow diagram if the tracks appearing on each page are listed in some convenient place on that page. It is also a help if the head positions applying to each page are also attached.

(b) Constants Which Are Not. (R.A.Smith, E.E., N.R.L. Stafford.)

Occasionally in a subroutine an instruction is modified by the addition of some quantity which for example causes the instruction, on being obeyed a second time, to lead to a different instruction. This quantity is not necessarily a constant, in the case of a library subroutine although it may look like one. A good example of this occurred in R19F/1 (No. 269). The "constant" in 3_{19} is used to modify the NIS of 2_{22} and this "constant" does in fact vary with the position of the subroutine. Thus $3P_2$ should read 3, 0-0, 0, 0 X, which will be modified accordingly when the subroutine is copied into the other positions. This sort of situation has occurred several times in the past but has so far always been discovered in time. However there is always a chance that it will not, and

if a programmer were to use ZP03 (No. 4.) to copy one of his own subroutines he might make a mistake in this way. Note that this only applies to quantities which modify an instruction and often not even then.

(c) "Bit Binary" and Similar Terms. (R.A. Smith, E.E., N.R.L. Stafford.)

The use of terms "bit binary", "binary coded decimal" etc. is growing particularly in commercial work and in connection with the 80 column reader. These terms nearly always apply to a character, as for example that represented by a card column (punched alphanumerically) converted to binary. These characters in binary form are usually stored more than one per minor cycle. The following terms are in use

Binary Coded Decimal (B.C.D.). Usually meaning a decimal digit (0-9) in binary form.

4-Bit Binary (4 B.B.)
5-Bit Binary (5 B.B.)
6-Bit Binary (6 B.B.)
— A character represented by a 4 digit/ 5 digit/ 6 digit binary integer.

Bit-Binary - A loose term usually meaning any alphanumeric character(s) in binary form.

4. CONVERSION OF LIBRARY TO 64 COLUMN. (R.A.Smith, E.E., N.R.L. Stafford.)

The N.R.L. Stafford Library conversion to 64 column has now taken place. So far there have been very few complications. The main points are as follows:

(i) Programmers are inclined to grumble on finding that odd packs of cards which they have tucked away in a drawer have not converted themselves! The amount of work involved in making the change however is usually very slight since a reproducer board is kept permanently plugged for converting old 32 Col. to 64 ∝ field. Column 54 is plugged to Column 1 and Column 1 to Column 54 if the former is in the Y or X row - this being necessary for routines like ZP48M.

(ii) The design of the 64 Column instruction card has been found lacking in some respects. It is hoped to improve this.

(iii) The effect which moving the DEUCE field down 4 columns would have on the non-DEUCE-field part of the Alphacode layout was forgotten until a rather late hour. The layout of the card was however easily re-arranged.

(iv) Paragraph 12 shows a list of the a mendments to all programmes which need to be altered. It will be seen that some work remains to be done (generally on little used programmes) and any assistance given in this will be gratefully received.

(v) It is anticipated that for a time at least it will be necessary to do a small amount of work with the old 32 Column system.

501/23 NELSON RESEARCH LABORATORIES
STAFFORD E. E. CO. LTD.
MATHEMATICS DEPARTMENT.

5. ANOTHER INITIAL CARD - CLEAR DRUM. (I. Hughes, E.E., Warton.)

The following is a version of the initial card clear drum by J.P.O'Brien (DEUCE News 27, paragraph 4(i), which leaves the high speed store clear.

Y	0	0 - 0		0	0	X
X	0	0 - 8	1	27	26	X
0	0	0 - 8	2	6	3	X
1	0	8 -17		1	4	
2	$12P_1$	17 - 0	1	3	3	
3	0	30 -17	2	2	19	
4	0	30 - 8	1	8	7	
5	0	21 -28	2	0	3	
6	0	30 -21	2	0	20	
7	0	12 -24		0	19	
8	0	0 -24		0(1)	22	
9	0	0 -31	1	0(0)	26	

Anyone who is interested is advised to follow through the workings of this routine as it contains a number of points of interest and may provoke further ideas - it has two disadvantages over the O'Brien version, namely (i) it is not a genuine STOP detector (the card would have to be re-read) and (ii) it is not so easy to line up the scope.

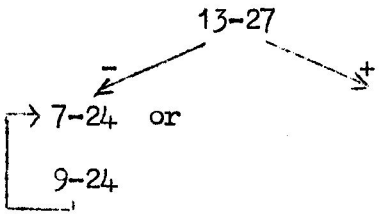
6. FAILURE INSTRUCTIONS IN SUBROUTINES AND PROGRAMMES. (R.A.Smith, E.E., N.R.L. Stafford).

A concensus of opinion at N.R.L. Stafford on this subject prøduced very little in the way of firm conclusions.

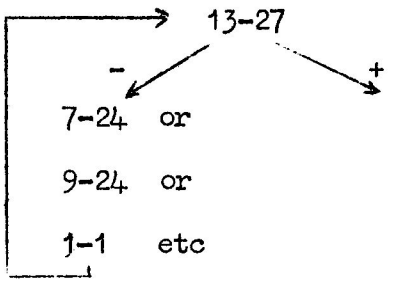
There was some support for and no opposition to the idea that failures in library subroutines should not be left blank (for the programmer to punch in). This was on the grounds that parameter punching should be minimised and that if the failure were forgotten the subroutine would at least not "drop out". On the other hand there is the point that if the programmer has to plant a failure instruction he is unlikely to be confronted with unfamiliar failures while programme testing.

The following methods are all possible but no one of them seems to cover conveniently all cases which arise in practice.

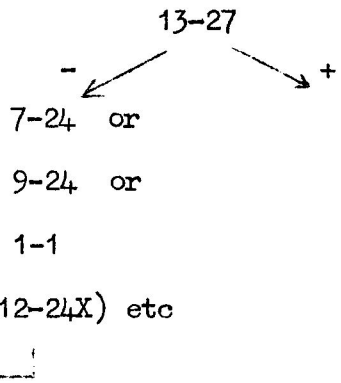
(i)



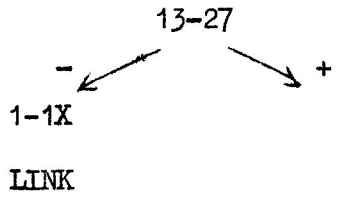
(ii)



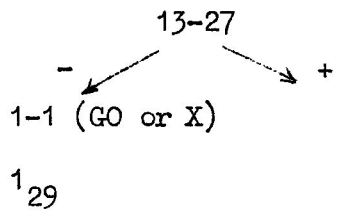
(iii)



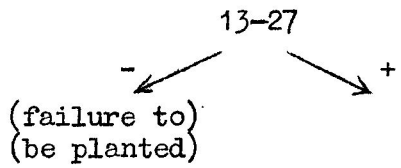
(iv)



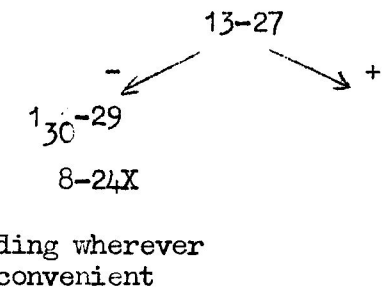
(v)



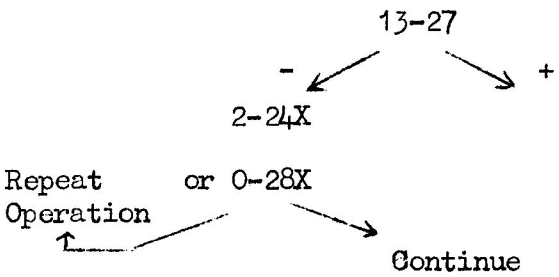
(vi)



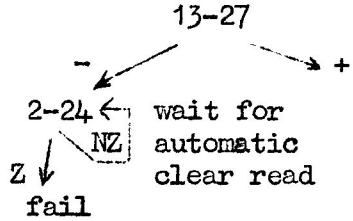
(vii)



(viii)



(ix)



NOTES.

(i) is universally considered undesirable.

(ii) In (ii) if the discrim is forced +ve the subroutine may get stuck in another loop(s) before finally reaching the LINK.

(iii) is good in read routines and sum checks but could be a nuisance (and pointless) otherwise.

(iv) is not desirable if the reader or punch is running.

(v) & (vi) put the onus on the programmer. In the case of (v) every subroutine would presumably lead to the same place unless 1_{29} were changed.

(vii) if universally adopted would mean that all subroutine failures would initially look alike (an advantage?) and later be distinguished.

(viii) In (viii) the programmer has the choice of repeating the operation or continuing.

(ix) is for the special case of the 80 column reader where single shots continue to appear through the complete card cycle.

One suggestion for actual failure instructions is

1-1 X
2-2 X
⋮
12-12X
} Programme Failures.

13-13X
14-14X
⋮
21-21X
} Subroutine Failures.

7. RATIONALIZATION OF MAGNETIC DRUM INTERLOCKS. (A.C.D.Haley, E.E., N.R.L. Stafford.)

The present system of interlocks set up during operations on the Magnetic Drum has the virtues of simplicity both of engineering and usage, so far as estimation of time of operation is concerned. It does, however, result in unnecessary wastage of time in some applications due to interlocking on sequences of instructions which, functionally, do not conflict. For example, the following sequence of operations forms part of a random access type of up-dating operation on a standing file (stored on the drum), using data read in on cards:-

9 - 24 ,
P - 31 ,
H - 30 ,
modify entry
P - 31 , 1
H - 30 , 1
12 - 24

The sequence requires at least 102 major cycles,and in practical applications may result in inability to run the reader at full speed.

A system has accordingly been investigated having, so far as is possible, only those interlocks which are functionally essential. These are as follows:-

Present Operation.		Inhibits.
Shift write heads	(P - 31, 1)	P - 31, 1 H - 30, 1
Shift read heads	(P - 31)	P - 31, 0 H - 30, 0
Write transfer	(H - 30, 1)	S - 11 P - 31, 1 H - 30, 0 or 1
Read transfer	(H - 30, 0)	S - 11, 11 - D (D $\neq$ 24 or 31) P - 31, 0 H - 30, 0 or 1.

As before, a call for shifting heads to the position they currently occupy does not set up a new interlock, but is still held up by an appropriate previous one: i.e. the sequence

0 - 31 ,
0 - 31 ,
H - 30 , 1

is obeyed at times (roughly) of 0, 35 and 35 major cycles, not at the full DEUCE instruction rate.

The sequence of the earlier example can now be re-arranged as follows:-

9 - 24 ,
P - 31 ,
P - 31 , 1
H - 30 ,
modify entry
H - 30 , 1
12 - 24

giving a saving in time of about 35 major cycles since the two head shifting operations proceed concurrently.

It is not expected that any significant benefit will accrue on scientific applications, but comment is invited on the following points:-

(a) Whether the proposed system is thought desirable on ground of time saved, and if so on what programmes or types of work.

(b) Whether the proposed system, regardless of possible saving of time, is thought desirable on grounds of easier programming.

(c) Whether the proposed system is known or thought likely to invalidate any existing programmes where, for example, reader or punch single shots are at present covered by magnetic operation delays.