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

DEUCE PROGRAMME NEWS - No. 24. May, 1958.

1. 64 COLUMN READ & PUNCH.

The information published in reports NS y 82 (DEUCE News No. 17) paragraph 13 and NS u 246 is incomplete and not completely accurate. The following is a revised specification.

Card Layout.

The 64 relevant columns are considered in two fields:

(i) α field - Hollerith columns 17-48.

(ii) β field - Hollerith columns 49-80.

Reading.

Two instructions (a) and (b), each with Source 0 must be obeyed at suitable times in order to successfully read the α, β fields respectively from one row of a card. The instruction (a) or an instruction shortly before (a) must be a stopped instruction, as with 32 column operation, and the reader provides a single shot when the row is in a position to be read.

The changeover of source 0 from α field to β field is initiated at the moment when (a) has been completely replaced in control by the succeeding instruction. This point should be regarded as the "moment of switching".

If (a) enters control in m.c. m (as it would if stored in m.c. m) and has timing number T then the moment of switching is at the end of m.c. m+T+2 (or m+T+34 if W > T) i.e. at the beginning of m.c. m+T+3. Thus m.c. m+T+3 is the first m.c. after the moment of switching, and m.c. m+T+2+n is the nth after the moment of switching.

Source 0 continues to provide field α during the first four m.c.'s after the moment of switching and may be referred to by succeeding instructions if desired. Thereafter the pattern provided is uncertain until the twentieth m.c. after the moment of switching when the β field becomes available with certainty.

Thus we have:

m.c.'s.	Field.
m+T+3 to m+T+6 inclusive.	Still α.
m+T+7 to m+T+21 inclusive.	Uncertain.
m+T+22 onwards.	β

Field β remains available until the current row is no longer accessible at the reading station. The specification for this is unchanged from 32 column operation i.e. the row cannot be read with certainty more than 2 m.s. after the Single Shot (the 2 m.s. being measured from the m.c. specified by the wait number of the stopped instruction i.e. 2 m.s. after the start of the transfer specified by the stopped instruction).

It should be noted that (a) need not necessarily be the stopped instruction released by the Single Shot. In fact (a) may be delayed as long as desired provided sufficient time is left to switch to, and read, the β field if required, before the row is lost (i.e. not more than 2 m.s. after the Single Shot).

Similarly there is no need for (a) to be immediately followed by (b) provided that the transfer called for by (b) is completed within 2 m.s. after the stopper. It is however not permissible for the stopper to be later than the α field source 0 instruction.

A typical pair (a) and (b) might be

1_0 0-14X (a)

1_2 0-15 GO (W $\geqslant$ 18) (b)

Also noteworthy is the fact that instructions of the form 0-24, 0-30 and 0-31 are not instrumental in initiating switching. All other destinations are however; in particular 0-0, which is sometimes used as a dummy instruction and therefore requires care in its application during reading.

Punching.

Two instructions (a) and (c), each with Destination 29, must be obeyed at suitable times in order to successfully punch the α, β fields respectively on one row of a card. There must also be a stopper near (a) as with 32 column operation and the punch provides a Single Shot when it is ready to punch the row.

In order that Destination 29 can receive the second information word without superimposition the Output Staticisers must be switched by a third instruction (b) (of the form 8-24 1) obeyed after (a) yet before (c).

Normally (a) is the stopped instruction and then (a), (b) and (c) may be timed as convenient except that information must be supplied to Destination 29 by c) not more than 4 m.s. after the Single Shot in order to punch successfully on that row.

A typical example might be

1_0 α-29X (a)

1_2 8-24 1 GO (b)

1_4 β-29 GO (c)

As in the case of the reader other instructions may be inserted between (a) and (b) and (b) and (c).

If there is any particular reason for not wanting (a) to be the stopper then the stopper must precede (a) and the transfer called by (c) must again be completed within 4 m.s. after the stopper.

General.

There are no changes in the standard operating specifications (including timings) established for 32 column machines apart from the special effects described above.

For single β field operations on the Reader a dummy Source 0 instruction should be obeyed in place of (a) and (b) should be timed as above.

For single β field operations on the Punch (a) is simply omitted and (b) followed by (c) should be timed as above, with (b) or an instruction rather earlier, being a stopper.

64 Column General Punch Subroutine.

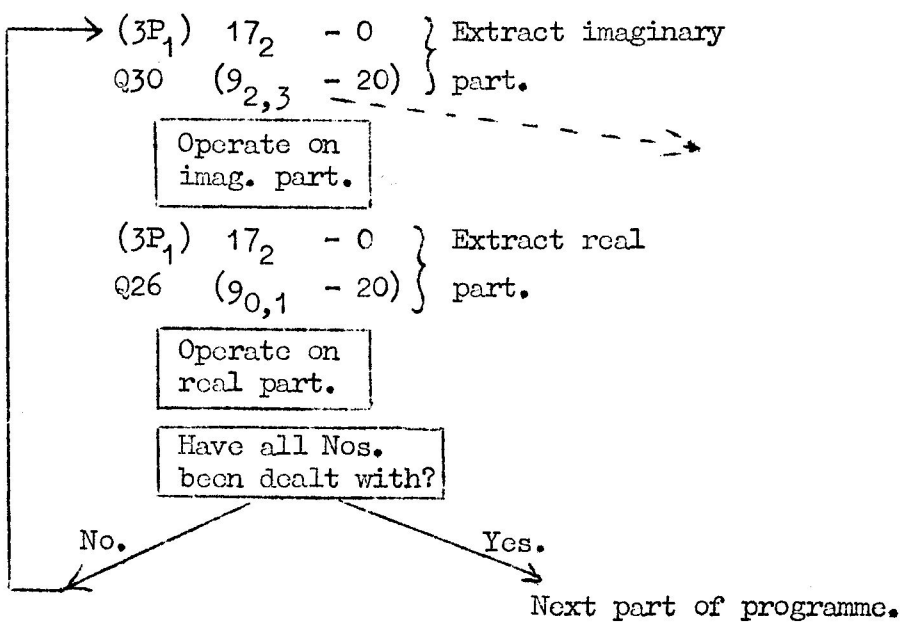
A subroutine has been produced at E.E. (N.R.L. Stafford) which allows the punching of more or less any pattern of decimal numbers in the 64 column DEUCE field. It allows nearly all of the 116 ms. between cards (when the punch is running continuously) for programming. The subroutine punches one card late i.e. while converting the set of binary numbers for one card to decimal it is punching the previous card. For this reason the subroutine has two entries, FIRST & NORMAL, e.g. To punch two sets of numbers (i.e. two cards) plant the first set and enter at FIRST, then plant the second set and enter at NORMAL. Finally enter at NORMAL again. (This simply punches the second set while uselessly converting them again). There must be at least 2 and not more than 9 decimal digits in each number and the inequality $5n + 4d < 352$ must be satisfied where n is the number of numbers on one card and d is the total number of columns to be punched on one card. Blank columns are allowed except in the first column of each of the α, β fields. The numbers presented to the subroutine must all be to 27 b.p. and in modulus less than 10 and the subroutine assumes that they are correctly rounded off. If there are 16 numbers or less for each card the subroutine occupies $2\frac{1}{2}$ instruction D.L.'s and uses D.L.'s 9 and 10 for working space.

2. PROGRAMMING DEVICES.

(a) People who use ZP38 (No. 354) for filling the drum may be interested in the following. (ZP38 is the drum filling programme described in the original programming manual and has come to be known as the Humby Card).

Each triad to be read to the drum is first read into D.L.7 so that after the last triad has been read to the drum it is also in D.L.7. This could then be entered at 7_t by punching NIS 7 and timing number t+1 in the 1-row of the first card of the last triad. By this means it is not necessary to follow the drum triads with any for the D.L.'s i.e. all the programme goes on the drum.

(b) The following device suggested by Mr. A.K. Rhodes of Marconi's Wireless Telegraph Co. Ltd. could be used to extract or store double pairs of numbers from a D.L., e.g. when dealing with the real and complex parts of complex floating numbers.



The point is that $(3P_1)$ means add $2P_{17}$ to the instruction in 17_2 but since this is obeyed 4 m.c.'s earlier the effect is to fetch the previous pair on obeying it the second time in each loop.

It may also be possible to spill (as indicated by the dotted arrow) after the appropriate number of loops, the final 9-20 instruction being a useless one.

The method could be used to fetch or store blocks of 4 numbers. Other applications will probably occur to individual programmers.

3. SEMI-FLOATING ARITHMETIC.

The following scheme has been suggested by Mr. O. Amble of Norway to extract more accuracy from semi-floating arithmetic.

When using floating point arithmetic a number is represented as $a.2^b$ where a, b are given for each number with certain restrictions imposed, generally of the form

$-1 \leqslant a < -\frac{1}{2}$ or $\frac{1}{2} \leqslant a < 1$ with a to p b.p. and $-2^q \leqslant b < 2^q$.

In standard floating binary p=30, q=31 and a single length word is used for the storage of each of a and b. This gives good accuracy (30 significant binary digits), over a large range of numbers and it is rarely, if ever, necessary to bother with the limits of numbers.

This is not so with the semi-floating system. For instance in the system used in A11 and A11/1 (Nos. 191, and 192), p=20, q=9 and a, b are in the same single length word. No meaningful arithmetical operations can be done directly on these semi-floating numbers. These are performed in the subroutine whose first part "UNPACK" separates a and b. It then does the operation and finally in the "PACK" section brings the result back to semi-floating form. 20 significant binary digits correspond to 6 significant decimal digits which very often will be reduced to 5 or even less by rounding-off errors etc. and this is insufficient accuracy in many cases. However this scheme stores two redundant digits above the binary point, since a number x satisfying the above conditions has both its digits above the binary point the same and the first digit below the binary point different from these two i.e. the top 3 digits are ... 100 or ... 011. Thus if a number were shifted up two places the new top digit determines the two lost digits. A few extra instructions in the UNPACK section would take care of this, e.g.

$30\text{-}21_2$

$x\text{-}21_3$

22-21 (d)

$29\text{-}22_3$

reproduces the sign digit and x can then be shifted down further to separate a and b as A11 does already.

The number zero would have to be specially represented since a blank m.c. now represents -1 (-1×2^0). Zero could in fact be taken as the smallest number possible and this picked out in UNPACK.

Accuracy can be gained at the expense of a loss in the range by increasing p and decreasing q. This can be done in A11 by altering a few instructions in the UNPACK section.

The above points are suggested for consideration by anyone who is considering making semi-floating arithmetic subroutines. Similarly the idea contained in A17 (No. 240).

4. SORTING BY COMPUTER.

DEUCE News No. 17, paragraph 10, contained a short note on sorting programmes stating that some programmes were being prepared and making an estimate of the time for one case. The purpose of this note is:-

I. To define some of the terms used.

II. To mention some of the differences that occur in sorting problems.

I. (a) When it is necessary to sort items of information it is commonly desired to arrange the information in increasing order according to some reference which is included in each of the individual items. This reference is commonly called the key.

I. (b) Each item of information which is to be sorted is sometimes called a block.

I. (c) In some sorting operations it is desired to sort large blocks of information. In this case it is usual:

(i) to detach the key from the remainder of the block;

(ii) to attach to the key an address identifying where the remainder of the block is, then

(iii) sort the address plus key and finally

(iv) use the address to put the remainder of the block in the correct place.

I. (d) A string is a number of blocks which are in order within themselves, thus at any time the total number of blocks which have to be sorted consist of one or more strings.

I. (e) There are various methods of getting information from one order to another. These go under the collectivename of sorting, but are distinct in several ways. Some of these are:

(i) Pigeon holing, e.g. putting letters in a letter rack where there is no concern about the order of the letters within a pigeon hole.

(ii) Diverging sorting. This is an extension of the pigeon holing method. An example of this method is the use of the punched card sorter which achieves a new order by progressive separations column by column. This method destroys any existing order in the blocks.

(iii) Merging sorting. This is the opposite of the divergingprinciple and achieves a new order by putting strings together so as to form longer strings and, by a repeating process, only one string. The nearest punched card equivalent to this process is that of collation. This method takes advantage of any existing order in the blocks.

(iv) Substitution. In this method the process is to look through the entire number of items to be sorted picking out the one with the smallest/largest key and then swopping that item with the first/last block; then repeat the process on the reduced number of blocks until the desired order is achieved.

II. There are a number of variables which have to be considered before the best method can be determined in any circumstance. Some of these are:

II.(a) Is there a known number of blocks to be sorted?

II.(b) Have the keys a known range in value? For example, are they always positive and/or non-zero? If the keys can have any value then it will in general be necessary to know how many blocks there are as clearly there must be some means of knowing when the end has been reached.

II.(c) Are the blocks uniform in size? If yes, then the size of block is probably determined. If not, it may be best to choose a size of block which covers the majority of cases and allows for continuous blocks for the few which exceed this limit.

II.(d) Is the requirement to sort needed as a subroutine or is it required more as a brick? Linked with this is the consideration of the number of blocks to be sorted.

If the number of blocks is small and the requirement is for a subroutine then it is probable that the most suitable method is that of substitution because:

(i) the programme is smaller and

(ii) the method does not need extra storage space.

If the number of blocks is large then it is probable that either the diverging or the merging method is more suitable.

If the key has a small range of possible values in comparison with the number of blocks, then diverging will be more advantageous than merging; conversely if the key has a large range of possible values in comparison with the number of blocks then merging will be the better method. Both the diverging and the merging methods need extra storage space of intermediate results, and in order to make them fast they need nearly all the mercury store.

The time for substitution is proportional to N^2

diverging	$N \log K$
merging	$N \log N$

where N is the number of blocks and

K is the range of the key.

If the block is large (8 m.c. appears to be the break-even point) then it will in general be advantageous to detach the key from the block and do the sorting on the key and address only.

The key should, if at all possible, be kept less than 2 minor cycles in size as beyond this the complications increase very considerably.