

HINTS ON PROGRAMME TESTING ON DEUCE

by

Miss E. R. de Bourcier, M.A.
D. G. Burnett-Hall, B.A.

As the title implies this note does not claim to be an exhaustive account of methods for testing programmes on DEUCE. The general emphasis is to encourage the programme-tester to save time on the machine by getting it to punch out information he can take away and study.



HINTS ON PROGRAMME TESTING

If a programme is to be tested satisfactorily it is essential to have calculated a test case in advance. This may be worked out in decimal with the numbers converted to binary at strategic points in the programme. The programme should be divided into its main sections and the results at the end of each section should be known. Later, if any section is not correct, it can be subdivided and the numbers evaluated by hand for the new points of division. It is often useful to know the binary equivalents of data read in decimal to check that the scaling factors are correct.

At present the only way to run the programme a section at a time and come to a stop at the end of a section is to put "stoppers" into the Go digit holes of certain instructions before the cards are read. This should be done at the points at which the results can be checked; it is advisable to mark such stopped instructions on the cards to facilitate the removal of the stoppers when testing has been completed. It is often useful to make the links stopped instructions. The machine will stop before obeying these instructions, and on receiving a single-shot should proceed to the next "stopper." If it fails to do so, one quick method of finding where the machine has stopped (or where it is looping) is to look at the links in 1₃₀, 1₃₁, and 1₀. These will indicate which was the last subroutine used. If the machine stops within an inner loop, examine the count number.

Another method which does not involve time-wasting while reading numbers from the 'scope is to run the programme in again, proceed to the last stopper it reached successfully, and then use "programme display". If the instructions punched out are not what was expected, examine the programme cards to make sure there is no punching error. This should never happen as "Check detailed coding" should be used on every programme, however trivial, and the flow diagram should be reconstructed.

Some of the more common faults are:-

1. Errors in counting loops.
2. Wrong parameters in subroutines, or parameters omitted completely.
3. Numbers getting out of capacity through wrong scaling.
4. $|a| > |b|$ in division subroutines.
5. Altering writing head position but not reading head position, or vice versa.

If the error is not obvious time should not be spent on DEUCE working through a long programme with the single-shot key; instead use should be made of the Programme Testing Aid, R.A.E. 145, as follows:-

Put the single-card programmes R.A.E. 134 and 135 between the initial card and the rest of the programme. These clear the drum and put up a clock track on to track 15/15. The reader stops for six seconds after reading the first card.

At any point of the programme when information is required the programme testing aid will punch out the short tanks if desired, the contents of selected delay lines and of non-zero tracks on the drum. To use it,

- (i) Stop the machine.
- (ii) Clear I.D. and put 0, 0-0, 1 on external tree keys. (i.e. all keys level).
- (iii) External tree on, give a single-shot, external tree off, stop key to normal.
- (iv) Stim. read, put remainder of programme testing aid in reader, and press "Run in" key, not "Initial input." Cards 0-4 are read in and the machine stops on 1, 0-15.

- (v) Put any combination $P_1 \dots P_{12}$ on the I.D., putting up the lights corresponding to the delay lines you wish to have punched. Put up a P_{32} if the short tanks should be punched.
- (vi) Give a single-shot. The delay lines requested are punched out. Then cards 5-7 are read in and all non-zero tracks on the drum are punched out.

If P_{32} is put on the I.D. the machine punches out first a triad of cards with the first four rows blank; the rows following are the contents of $13, 14, 15, 16, 17_0, 17_1, 17_2, 17_3$ (end of first card), $18_0, 18_1, 18_2, 18_3, 19_2, 19_3, 20_2, 20_3, 21_2, 21_3$, with two blank rows on the second card and a third card which is blank.

The programme will punch out the contents of delay lines 1 to 12 (as selected by putting some of P_1 to P_{12} on the I.D.) on triads of cards; each triad has the number of the delay line $X P_j$ on the top row of the first card, followed by three blank rows and the 32 minor cycles punched in their correct positions.

When a track of the drum is punched the track number $X P_{17}$ appears on the top row of the first card, followed by a P_{17} on the second row, two blank rows and the 32 minor cycles in their correct positions.

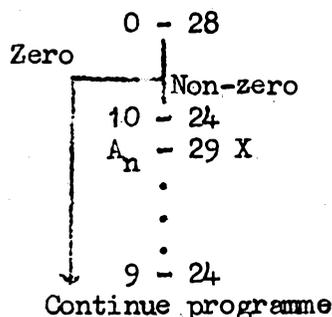
If it is unnecessary to punch out the drum, run cards 5-7 out of the reader while the delay lines are being punched out.

Tracks 15/12 and 15/13 are used by this programme. One minor cycle of delay line 1 will have been made zero when the programme is read in, and will be punched as a blank row if D.L.1 is requested.

It is possible to examine this evidence away from the machine. It will show whether instructions have been altered since they were read in, or whether tracks have not been brought down from the drum at the right time. It will also give the values of expressions already found, and the numbers to be used in the next part of the programme.

If the whole of the programme, including a clock track, and perhaps an extra track to bring down the first part of the programme, is read up to the drum, the single card programme R.A.E. 144 can be used to start the programme. Then to restart the programme again it is necessary to read only this last card with the initial input key.

If there is an iterative loop and there is any doubt about its convergence it is useful to insert the following group of instructions:-



If any number is put on the I.D. the machine will punch out the contents of previously specified locations in the store. Once the programme has been tested this device will cause a delay of at most one

instruction. It is then advisable to punch 30 in the source position of the O-28 instruction so that the I.D. does not have to be clear when the programme is used.

REMEMBER THAT THE MACHINE'S TIME IS MORE VALUABLE THAN YOURS.

