With the Author's compliments.

TECH. NOTE
M.S.31

TECH. NOTE
M.S.31

ROYAL AIRCRAFT ESTABLISHMENT
FARNBOROUGH, HANTS

TECHNICAL NOTE No: M.S.31

# THE ASSEMBLY OF LARGE PROGRAMMES FOR THE AUTOMATIC COMPUTER DEUCE

by

J.M.WATT, B.A.

JULY, 1956

MINISTRY OF SUPPLY, LONDON, W.C.2

UNCLASSIFIED

U.D.C. No. 518.5 : 518.12 : (DEUCE)

Technical Note No. M.S.31

July, 1956

ROYAL AIRCRAFT ESTABLISHMENT, FARNBOROUGH

# The assembly of large programmes for the automatic computer DEUCE

by

J. M. Watt, B.A.

---

SUMMARY

Details are given of a new method of building up a complete programme from sub-programmes. The method is contrasted with that developed at the National Physical Laboratory, and each is found to be advantageous in particular applications.

---

## LIST OF CONTENTS

## 1 Introduction

This note describes two schemes at present being used to combine sub-programmes (or bricks) to form a complete programme. One of the schemes was developed at N.P.L. by M. Woodger and B. W. Munday, whilst the other has been developed by the author at R.A.E. The N.P.L. method is not described in detail here, but for the benefit of R.A.E. staff D. G. Williams has compiled a note (Computing Note MS/B/206) on Deuce Schemes A and B giving details of it and describing the bricks available.

Although the two schemes are almost identical in effect, each has a particular field in which it is best used. The N.P.L. scheme is best for programmes which have to be constructed from standard bricks, but which will be used a few times only; in such cases the ease of coding of the problems is important and the slight extra amount of computing and input time unimportant. On the other hand when it is required to combine several sections or bricks together into a permanent programme the R.A.E. scheme is better, as it is more flexible, and the slight extra time needed to draw up a programme will be justified by the saving in input and computation time.

The R.A.E. scheme is perhaps at its best when a large programme is split into its logically distinct parts and each part is treated as a brick to simplify the organisation of the programme. As an example of this, a programme to calculate correlation coefficients could be split into several sections. The first brick would read the data from cards and store it on the drum. The next brick would consist of two sections, the first computing the product sums and the second using these to compute the correlation coefficients and store them on the drum. The third brick would then punch out the results, ready for tabulation. Another way to do this would be to construct a master routine to read the data, use the second brick above as a sub-routine to calculate the correlation coefficients, and then punch out the results.

The bricks to be used in these schemes must satisfy certain simple conditions which are not the same for both schemes, but are so very similar that it is no hardship to design bricks that will satisfy the conditions of both schemes simultaneously.

## 2 Conditions to be satisfied by bricks

With the R.A.E. scheme (using R.A.E. 180*)

(1) If all the instructions of a brick cannot be stored in the delay lines at the same time, the brick must be divided into several self-contained sections, each of which can be placed in the delay lines and obeyed before the next section is brought down.

(2) Each section of the brick must occupy a consecutive sequence of delay lines at the lower end of the store, using up to DL10 if necessary.

(3) It must preserve $12_{29,30,31}$.

(4) If the brick consists of one section only this section must end by clearing T.S.13, transferring the contents of DL12 to DL1 and taking the next instruction from $1_{30}$.

---

*R.A.E. 180 is an improved version of R.A.E. 137. This latter is thus made obsolete.

(5) If the brick consists of more than one section, the final section must end as above, but the previous sections must end by planting $P_{32}$ in TS13 and a certain parameter in $21_2$ before transferring the contents of DL12 to DL1 and taking the next instruction from $1_{30}$.

If the section occupies r delay lines and is entered in $N_t$, and the section that follows occupies r' delay lines and is entered in $N'_{t'}$, this parameter must be

$$(N' - N)\, P_2 + r\, P_5 + (t' - t)\, P_{17} + (r' - r)\, P_{22} + (t' - t)\, P_{26}.$$

(6) Order of cards. The sections of a brick must be arranged in their natural order, but the triads within a section in the reverse order.

For example, the cards of a brick of two sections, the first occupying DL's 1,2,3 and the second DL's 1,2 should be arranged

| | | | | | |
|---|---|---|---|---|---|
| DL | 3 | Section | 1 | cards | 1 - 3 |
| " | 2 | " | 1 | " | 4 - 6 |
| " | 1 | " | 1 | " | 7 - 9 |
| " | 2 | " | 2 | " | 10 - 12 |
| " | 1 | " | 2 | " | 13 - 15 |

(7) It is recommended that all bricks be entered in $1_0$.

(8) N.B. When a new section of programme is brought down the contents of all stores, except TS13, TS14, TS15, DS21, DL11, and those delay lines occupied by the new section of programme, are preserved.

If the brick is also to be used with the N.P.L. scheme:-

(9) The only parameters allowed are positive integers $< 256$, the first three of which should be planted in $18_{0,1,2}$ as multiples of $P_5$. If more parameters are needed they should be planted in $17_{0,1,2}$ in a similar way.

(10) The filler instructions of the triads must be punched as though the programme was to be read directly into the delay lines, and the point of entry to each section must be given by the filler instructions of the last triad, in the normal way.

If a brick is to be used only with the N.P.L. scheme the delay lines occupied by a section need not be consecutive, the parameter need not be planted in $21_2$, and the order of triads within a section does not matter. However the extra restrictions required by either scheme compared with the other are not very severe. Most of the existing linear algebra bricks conform to both schemes, and it is hoped that future bricks will be designed to do so if possible.

## 3 The R.A.E. scheme of assembling programme from bricks

In this scheme a master routine must be constructed to supply parameters to the bricks and to R.A.E. 180 (the programme-changing routine) which is used to transfer the sections of programme from the magnetic drum into the delay lines. The master routine can also be used to count the number of repetitions of cycles, and to do sections of the computation.

To call in a brick

(1) The master routine must either leave the instructions

```
1 14 - 30   0  0
1 15 - 31   0 29
1 11 - 1  1 4  3
```

undisturbed in $12_{29,30,31}$, or replace them if they have been destroyed. Initially these instructions can be planted by R.A.E.181 which is specially designed for that purpose, or by the master routine. They will not be destroyed by any of the bricks.

(2) It must plant the parameters for the brick. (In the N.P.L. matrix algebra programmes the a,b,c parameters are planted in $18_{0,1,2}$ respectively as multiples of $P_5$).

(3) The 'cue' and 'link' must be planted in $21_2$ and $21_3$ respectively.

(4) R.A.E. 180 must be transferred to DL1 from track 254 of the drum, and entered in minor cycle 2.

The cue and link mentioned above are parameters for R.A.E. 180. They specify the position and number of tracks on the drum that are to be transferred to the fast store, and the point of entry to the programme transferred.

The cue in $21_2$ is $NP_2 + mP_5 + t\,P_{17} + (r-1)\,P_{22} + t\,P_{26}$

and the link in $21_3$ is $N'P_2 + m'P_5 + t'\,P_{17} + (r'-1)\,P_{22} + t'\,P_{26}$ .

These parameters will cause R.A.E. 180 to transfer the first section of r tracks of the brick stored in track m onwards to the delay lines and to enter the brick in $N_t$. When this brick has been finished, the programme stored in track m' onwards whose first section occupies r' delay lines and is entered at $N'_{t'}$ will be transferred to the delay lines and obeyed. This programme will normally be the master routine to which control will be transferred between each of the bricks.

It is sometimes convenient to treat R.A.E. 180 as DL1 of the master routine, so that it is brought down from the drum with the rest of the master routine and need not be transferred separately.

At the beginning of a programme the master routine itself can be transferred from the drum using R.A.E. 180. In this case R.A.E. 138 (initial entry to Programme changing routine) can be used to call down R.A.E. 180 into DL1 and supply its parameters. This method can also be used if the programme contains only one brick, which is transferred to the delay lines using R.A.E. 138 and R.A.E. 180.

Example

Suppose it is required to construct a programme from two bricks 'A' and 'B' which are to be obeyed one after the other. When finished the programme is to restart so that fresh data can be dealt with.

Brick A is stored in tracks 100 to 109; it occupies DL's 1 to 10 and is entered in $4_0$. Brick B is stored on tracks 110 to 122, and consists of two sections, the first occupying DL's 1 to 8 and the second DL's 1 to 5. Brick B requires parameters 100 $P_5$ in 180 and 5 $P_5$ in $18_1$.

The master routine could be stored on track 130, and coded:-

| | | |
|---|---|---|
| Initial entry. Entry from B. → | $1_0$ $1_{2,3}$ - 21 | Plant cue and link for A. |
| Entry from A → | $1_4$ $1_{8,9}$ - $18_{0,1}$ | Plant parameters for B. |
| | $1_{10}$ $1_{12,13}$ - 21 | Plant cue and link for B. |
| | $1_{20}$ 15 - 31<br>$1_{22}$ 14 - 30<br>$1_{29}$ 11 - 1 (32 m.c.)<br>$1_2$ | Transfer R.A.E. 180 from track 254 to DL1, and enter in $1_2$. |

Content of $1_2$ $4.P_2 + 100.P_5 + 0.P_{17} + 9.P_{22} + 0.P_{26}$ cue for A.

" " $1_3$ $1.P_2 + 130.P_5 + 4.P_{17} + 0.P_{22} + 4.P_{26}$ link for A.

" " $1_8$ $100.P_5$

" " $1_9$ $5.P_5$ } Parameters for B.

" " $1_{12}$ $2.P_2 + 110.P_5 + 17.P_{17} + 7.P_{22} + 17.P_{26}$ cue for B.

" " $1_{13}$ $1.P_2 + 130.P_5 + 0.P_{17} + 0.P_{22} + 0.P_{26}$ link for B.

The master routine would be called in initially by R.A.E. 138 with parameter $1.P_2 + 130.P_5 + 0.P_{17} + 0.P_{22} + 0.P_{26}$, punched in row 2. The cards would be assembled as in section 4, but in this case item 9 would not be needed.

## 4 Assembly of the cards of the complete programme

The cards are best assembled in the following order:-

(1) Initial card.

(2) R.A.E.134 (clear drum) (Optional).

(3) R.A.E.135 (clock track).

(4) R.A.E.136 or ZP04 to read the programme to the drum.

(5) The bricks required.

If these are to be read by R.A.E.136, the initial card preceding each brick should be removed and the Y row of the first card of each brick punched with $P_{15} + mP_{17}$, where m is the first of the sequence of tracks on which this brick is to be stored. The Y row of the first cards of all other triads must be blank. (This will normally be the case but may not be if some sections store programme in DL's 9 and 10.)

If ZP04 is being used to read the programme to the drum, the appropriate parameter card must be placed in front of each of the bricks.

(6) The master routine prepared in the same way as the bricks.

(7) R.A.E.180 to be read up to track 254 of the drum, and to be the last triad read. If R.A.E.136 is used this means that the Y row of the first card must be punched with $P_{15} + 254\ P_{17} + P_{31}$.

(8) R.A.E.181 to synchronise the remaining input with the information already on the drum, and plant the three orders in DL12.

(9) Any other instructions or data to occupy the delay lines at the beginning of the programme.

(10) R.A.E.138 to bring down the first section of programme into the mercury store and enter this programme. This must be punched on row 2 with the appropriate parameter for R.A.E.180.

Item 9 is not often needed and sometimes 8 can also be omitted.

Provided the instructions on the drum have not been destroyed, the programme can be restarted by running in the cards of 8, 9 and 10 above with the initial input key.

Programmes R.A.E.134, 135 and 136 have been described in Computing Note M.S./B/203. Specification sheets and flow diagrams of programmes R.A.E.180, 181 and 138 are appended to this note.

---

Attached: Appendix I
Detachable Abstract Cards

Advance Distribution:

DGSR(A)
DWR(D)
TPA/TIL - 100

ARDE (Dr. Maccoll)
RRE (Mr. Taylor)
NPL (Maths. Div.) - 5

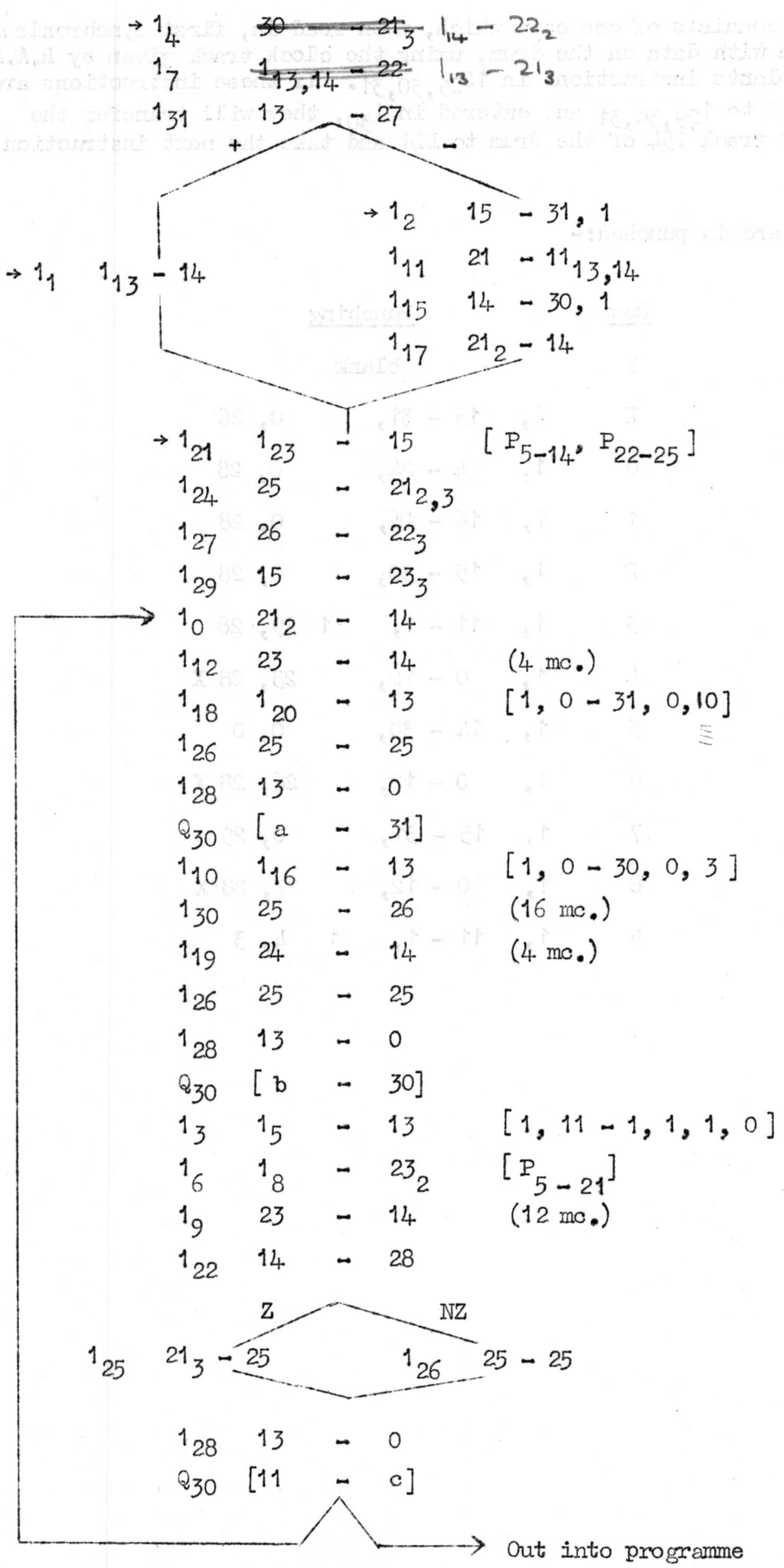
DRAE
DDRAE(A)
RAE Aero Dept
GW "
Arm "
Str. "
RPD "
Trials Dept.
Library

The routine uses TS13, TS14, TS15, DS21 and DL11. No other stores are disturbed except those delay-lines into which new programme is transferred.

At exit $(m + r)P_5 - P_{22}$, is left in $21_2$.

---

## Flow diagram of R.A.E.180



## R.A.E.181 - Synchronise and plant instructions in DL12

This consists of one card which, when read in, first synchronises its action with data on the drum, using the clock track given by R.A.E.135 and then plants instructions in $12_{29,30,31}$. If these instructions are transferred to $1_{29,30,31}$ and entered in $1_{30}$, they will transfer the content of track 254 of the drum to DL1 and take the next instruction from $1_4$.

The card is punched:-

| Row | Punching |
|---|---|
| Y | blank |
| X | 1, 15 - 31, 0, 26 |
| 0 | 1, 4 - 24, 0, 28 |
| 1 | 1, 16 - 16, 0, 28 |
| 2 | 1, 15 - 30, 0, 28 |
| 3 | 1, 11 - 1, 1 29, 28 |
| 4 | 1, 0 - 12, 25, 28 X |
| 5 | 1, 14 - 30, 0, 0 |
| 6 | 1, 0 - 12, 26, 28 X |
| 7 | 1, 15 - 31, 0, 29 |
| 8 | 1, 0 - 12, 27, 28 X |
| 9 | 1, 11 - 1, 1 4, 3 |

## R.A.E.138 - Initial entry to programme changing routine

This card, when run in either at the end of a programme pack, or by the initial-input key, brings down the programme-changing routine from track 254 of the drum, transfers it to DL1, and enters it in minor cycle 2, with a given parameter in both minor cycles of DS21. The parameter must be punched on row 2 of the card.

The card is punched:-

| Row | Punching | | | | | |
|---|---|---|---|---|---|---|
| Y | blank | | | | | |
| X | 1, | 15 - 31, | | 0, | 26 | |
| 0 | 1, | 4 - 24, | | 0, | 28 | |
| 1 | 1, | 0 - 21, | 2 | 0, | 28 | X |
| 2 | parameter | | | | | |
| 3 | 1, | 16 - 16, | | 0, | 28 | |
| 4 | 1, | 15 - 30, | | 0, | 28 | |
| 5 | 1, | 0 - 0, | | 0, | 28 | |
| 6 | 1, | 11 - 1, | 1 | 29, | 28 | |
| 7 | 1, | 14 - 30, | | 0, | 28 | |
| 8 | 1, | 11 - 1, | 1 | 31, | 30 | X |
| 9 | blank | | | | | P54 |

UNCLASSIFIED

Royal Aircraft Est. Tech. Note No. M.S.31 518.5:
1956.7 518.12:
Watt, J. M. (DEUCE)

THE ASSEMBLY OF LARGE PROGRAMMES FOR THE AUTOMATIC COMPUTER DEUCE

Details are given of a new method of building up a complete programme from sub-programmes. The method is contrasted with that developed at the National Physical Laboratory, and each is found to be advantageous in particular applications.

UNCLASSIFIED

UNCLASSIFIED

Royal Aircraft Est. Tech. Note No. M.S.31 518.5:
1956.7 518.12:
Watt, J. M. (DEUCE)

THE ASSEMBLY OF LARGE PROGRAMMES FOR THE AUTOMATIC COMPUTER DEUCE

Details are given of a new method of building up a complete programme from sub-programmes. The method is contrasted with that developed at the National Physical Laboratory, and each is found to be advantageous in particular applications.

UNCLASSIFIED

UNCLASSIFIED

Royal Aircraft Est. Tech. Note No. M.S.31 518.5:
1956.7 518.12:
Watt, J. M. (DEUCE)

THE ASSEMBLY OF LARGE PROGRAMMES FOR THE AUTOMATIC COMPUTER DEUCE

Details are given of a new method of building up a complete programme from sub-programmes. The method is contrasted with that developed at the National Physical Laboratory, and each is found to be advantageous in particular applications.

UNCLASSIFIED

UNCLASSIFIED

Royal Aircraft Est. Tech. Note No. M.S.31 518.5:
1956.7 518.12:
Watt, J. M. (DEUCE)

THE ASSEMBLY OF LARGE PROGRAMMES FOR THE AUTOMATIC COMPUTER DEUCE

Details are given of a new method of building up a complete programme from sub-programmes. The method is contrasted with that developed at the National Physical Laboratory, and each is found to be advantageous in particular applications.

UNCLASSIFIED

DETACHABLE ABSTRACT CARDS

These abstract cards are inserted in RAE Reports and Technical Notes for the convenience of Librarians and others who need to maintain an Information Index.