CONSTRUCTION AND USE OF THE DEUCE COMPUTOR

© David R. Hill 1962, 2009

1. Introduction

The acronym DEUCE stands for: Digital Electronic Universal Computing Engine. Deuce is a stored programme digital computor capable of carrying out a high speed operation of arithmetic, logic and information transfer as directed by instructions in the stored programme or, occasionally, instructions manually set up on the computor control panel. The term stored programme means that all the instructions required for solving a problem are held within the machine. Once such a machine has been loaded with a programme and data it can proceed through the steps of a problem at very high speed without the need for human intervention. The term 'digital' refers to the manner in which information is held within the computor and means 'as digits', i.e. in discrete steps, rather than as continuous variables, such as are used in 'analogue' computors.

The difference is perhaps made clear by considering an abacus and a slide rule. To represent 8 on an abacus 8 beads are used and there is no doubt that 8 rather than 7 or 9 is represented, though 8.5 could not be represented except by having a second abacus to represent 0.5 with 5 beads. There is no ambiguity in an abacus, whatever the accuracy, and the accuracy is only limited by space in which to mount rows of beads. To represent 8 on a slide rule (which is a form of analogue computor) a length is used. A number such as 8.5 could be equally easily represented on the same scale but there could be doubt as to whether the number was 8.4, 8.5 or 8.6. The accuracy of an analogue machine is limited by accuracy of setting and accuracy of reading both of which are affected by the scale used. There is always ambiguity in an analogue machine whatever the accuracy.

The discrete steps used in Deuce represent the digits of the binary number code which has two possible digit states, '0' or '1'. This is easier to contrive electronically than the ten possible digit states of the decimal code, though to represent a given number about three times as many digits will normally be required.

In Deuce data and machine instructions are of the same form and thus either may be stored anywhere in the machine and arithmetical operations may be carried out on instructions. The machine has no way of distinguishing between data and instructions and if some data were sent to the machine's control circuits it would be 'obeyed' just as though it were an instruction, even if obeying it produced no result or a meaningless result. Data and instructions will therefore be referred to by the common title of "words".

A word consists of 32 binary digits (bits). Thus 2^{32} numbers may be represented. In Deuce the 32 bit binary words are represented by trains of pulses for ones and spaces for noughts. The digits follow one another in time sequence, the least significant or P1 digit appearing first and the most significant or P32 digit appearing last. The next digit to appear would be the first of the same,or another number, the timing circuits of Deuce preventing the numbers from getting mixed up. Thus we have;

P1, P2, P3 ……. P31, P32

It should be noticed that the order of significance is the opposite to that of numbers in general use; i.e. the least significant digit is on the left.

In Deuce the time interval between successive pulses is one microsecond, thus one word is thirty two microseconds long. Words are stored in Deuce by circulating them through a delay line which requires a finite time to pass through. For example, in a delay of 32 μsecs. a complete Deuce word could be stored. The first digit of a word in such a store would reappear every 32 μsecs, and this is called a minor cycle—a m.c.—of the machine's operation. Words are also stored as a series of dipoles on the surface of a magnetic drum.

2. Main Components

Reader and Punch. A modified Hollerith balancing tabulator is used for reading data from punched cards into 'Deuce'. A modified Hollerith gang punch is used to punch data fed out by Deuce. Both are connected. to Deuce by cables.

Main Frame. The electronics have about 1,450 valves [tubes] on about

68 standard chassis each chassis having 20 or 30 valves and being self contained apart from power supplies. Each is attached to the main skeleton by standard plugs and bolts,the power plugs being standard for all chassis locations. Some chassis are identical—e.g. there are 22 chassis associated with the 22 high speed stores. There is a magnetic drum store mounted on a non-standard chassis.

Power Unit. The power unit takes a 3-phase supply and provides 7 main D.C. voltages for Deuce (-300 -200, -100, 0, +100, +200, +300) and two special D.C. supplies for the reader and punch. The valve heaters are supplied with A. C. from transformers on the main frame.

Delay Line Stores. The twelve long delay line stores are in a thermostatic chamber together with two amplifiers per delay line, one for input one for output.

The control panel. The control panel on the front of the main frame has a number of keys, lights and buttons which are only used for testing programmes or checking machine faults. They consist of:

(i) 32 keys and white lights for the input of single words—the Input Dynamicisor or ID;

(ii) 32 white lights for the output of single words—the Output Staticisor or OS lights;

(iii) An alarum buzzer and red warning light;

(iv) Two cathode ray tubes for the display of storage.

(v) Keys for the manual imitations of some of Deuce Automatic functions:

  (a) TT key which gives continuous transfer between the selected source and destination,

  (b) TCI key which gives continuous entry into TS COUNT, the control store, of whatever is on the instruction highway,

  (c) 13 Special keys and associated indicator lights which can be usd to set up manually single instructions for the machine to obey. The lights will display the source destination and next instruction word in TS COUNT, and are kown as the Instruction Staticisor or IS lights,

(d) External tree/request stop key used to bring the previous 13 keys into effect under different circumstances;

(vi) A key to start and stop the reader with an indicator light, and a key and light for the punches;

(vii) A key to force local discriminations in a programme (positive/negative or zero/non-zero) along a particular path, regardless of the logic;

(viii) A similar key to force a choice when the machine tries to discriminate between the TIL signal on/off (see Appendix I, Triggers)

(ix) A stop/normal/augmented stop key.

(x) A key used in conjunction with (ix), to make the machine punch out each instruction as it obeys it;

(xi) Various lights to indicate machine function or malfunctions;

(xiii) A key to release the machine when stopped on augmented stop;

(xiv) Buttons and indicated lights controlling the power supplies to Deuce and its ancilliary equipment.

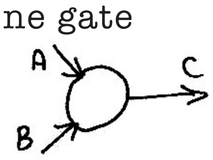
3. Logical diagrams and waveforms

Notation

Wide and narrow pulses: the pulse interval is always 1 μsec but the width of a pulse may vary according to where the pulse train is in the machine. Narrow pulses are approximately 0.3 μsec. wide. Thus there is a clear gap between successive ‘ones’ and so they can be transmitted using a.c, coupling and are normally used to connect units. Wide pulses have a width of 1 μsec. thus a row of ones or a row of noughts are represented by different D.C. levels. They are normally used in arithmetical units where successive digits are combined to form—say—the sum. Wide pulses eliminate a lot of error due to timing discrepancies since 0.2 μsec. difference in wide pulse arrival times would give 20% distortion whereas for narrow pulses the figure would be 66.7%.

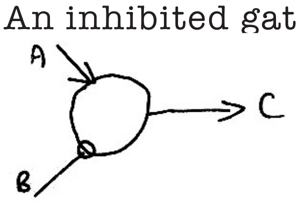
4. Logical diagram symbols_

(a) A one gate



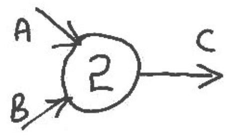
A signal appears at C from either or both A and B. e.g. A0101 B0011-C0111

(b) An inhibited gate



The signal at C equals the signal at A unless there is a signal at B.

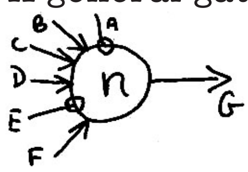
(c) A two gate



There is a signal at C only when there is one at both A and B (One of which must normally be wide to avoid timing problems)

e.g. A: 0101 B: 0011 --> C: 0001

(d) A general gate

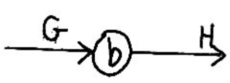


There is an output signal at G only if n normal inputs have a signal and no inhibiting inputs have a signal

(e) A unit delay

A is reproduced at B except that each digit is 1 μsec. late. Longer or shorter delays may be indicated.

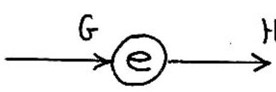
(f) A beginning element



H gives a single pulse outut corresponding to the beginning of a pulse at G

eg. (G --> 0 1 0 0 1 1 0 0 0 1 1 1 0
(H --> 0 1 0 0 1 0 0 0 0 1 0 0 0

(g) An end element

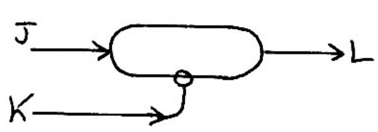


H gives a single pulse output corresponding to the end of a pulse at G (i.e. when O's start)

e.g. (G --> 0 1 0 0 1 1 0 0 0 1 1 1 0
(H --> 0 0 1 0 0 0 1 0 0 0 0 0 1

[(f) and (g) work only for wide pulses]

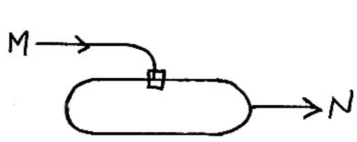
(h) A trigger



A signal at J starts a signal from L. A signal at K stops the signal at L.

e.g.
(J --> 1 0 0 0 0 1 0
(K --> 0 0 0 1 0 0 1
(L --> 1 1 1 0 0 1 0

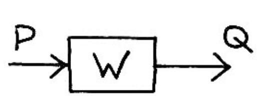
(j) A changeover trigger



The trigger is put 'ON', and 'OFF' alternately by pulses at K.

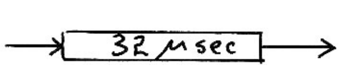
e.g. (M 1 0 0 1 0 1 0 1
(N 1 1 1 0 0 1 1 0

(k) A widener



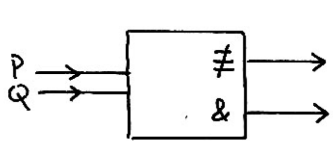
Changes narrow pulses at P into wide pulses at Q.

(l) Delay



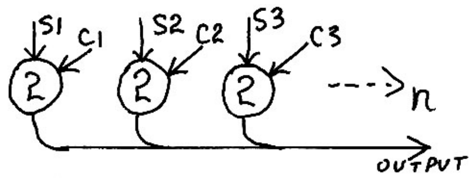
Delays by amount shown in box (32 μsec here)

(m) A composite gate



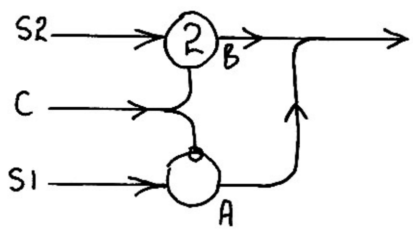
Has two outputs ≢ and '&'. '≢' gives an output when P & Q- differ. '&' gives an output if there is a signal at both P and Q

(n) A multiway gate



This connects one of n inputs (s1, s2, ... sn) to the outputs as selected by controls (c1 ... cn)

(o) A changeover gate



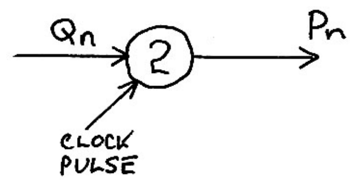
With no control signal, C, X ≡ S1; with a control signal X ≡ S2

Fundamental waveforms

(i) Clock pulses: A continuous sequence of narrow pulses, i.e. 0.3 μsec. wide, at 1 μsec. intervals.

(ii) Q pulses: there are 32 of these (Q1, Q2, ... Q32) corresponding to the successive digits of a word. Each consists of one wide (1 μsec wide) pulse every 32 μsec. They can be used to pick a particular digit from a word or initiate action required at a particular time in a minor cycle. The former is important for interpreting instructions.

(iii) P pulses: these are the narrow equivalent of Q pulses and are generated from them by a clock pulse gate which is a two gate with clock pulses as one input giving P1, P2, ... P32.

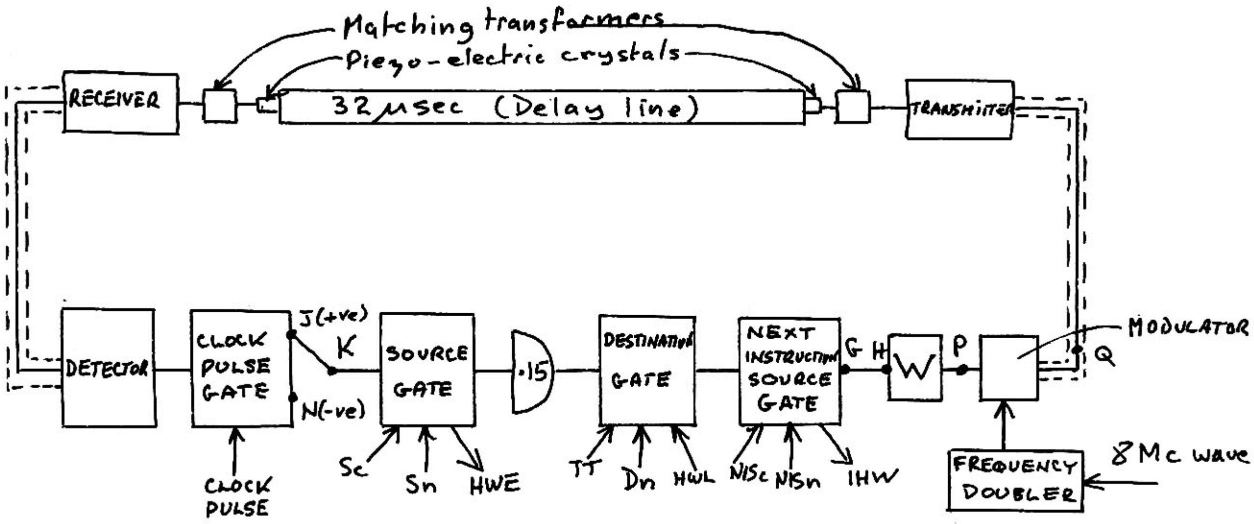


(iv) The fundamental timing waveform is generated by an oscillator circuit, crystal controlled to 1 Mc/s, and controls the timing of digits throughout Deuce. Three connections leave the oscillator: one to a shaper and power amplifier to give clock pulses; one to a divider, shaper, power amplifier and manually adjustable phase shifter (which generates pulses and enables them to be adjusted relative to the clock pulses); and lastly one leads to three frequency doublers in series, followed by a power amplifier, which supplies the stores circuits with an 8 Mc/s wave.

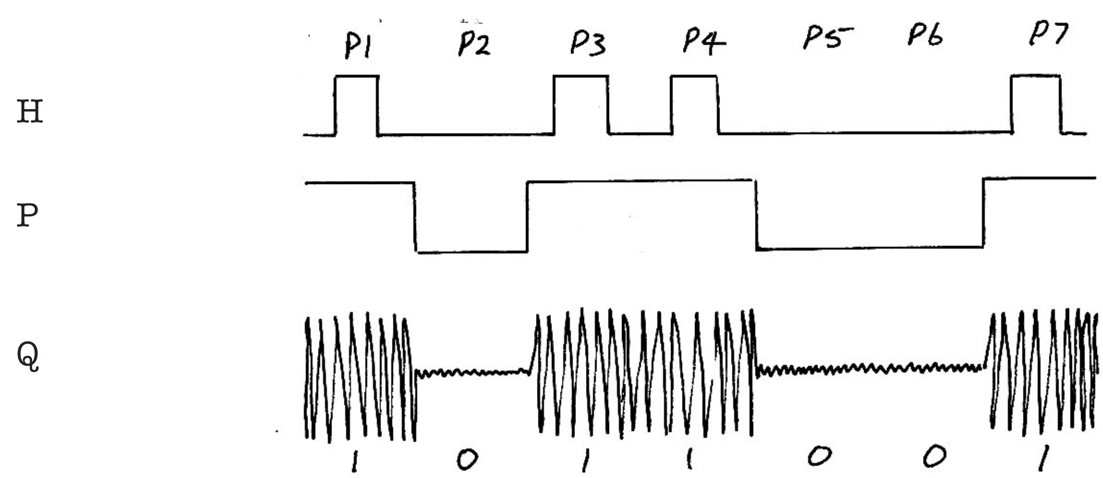
4. Circulation units

One of the main features which distinguishes Deuce from a desk calculating machine is its ability to store words. One way of storing words in Deuce is in a circulation unit the fundamental part of which is a mercury delay line.

A mercury delay line is simply a column of mercury with a piezo-electric crystal at each end. Mechanical vibrations produced in the mercury at one end, by exciting one crystal, will travel down the tube and produce an electrical signal from the other crystal. If the length of the column is such that a pulse takes 32 µsecs. to travel from one end to the other then one Deuce word could be stored in the column by recirculating it. If the tube were 32 times as long (ecuivalent to a 1024 µsec delay), 32 words could be stored. In practice the column is slightly shorter that it would otherwise be to compensate for the small delay in the recirculating electronics. We may now draw a diagram of one of the basic working units of Deuce:

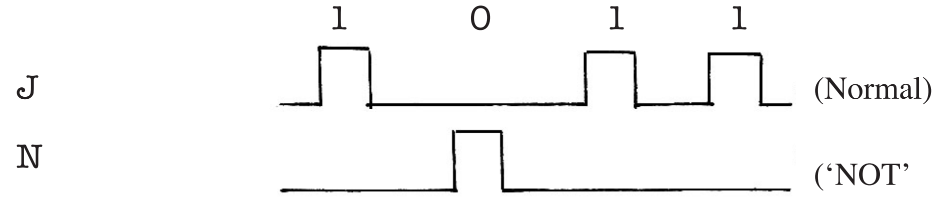


At H the pulses arriving are narrow. They are widened and used to modulate a 16 Mc/s carrier wave, this being generated by a frequency doubler from the 8 Mc/s supplied to each store circuit.



The modulated wave is transmited mechanically along the column of mercury regenerating an electrical signal at the other end. The signal is detected and thus wide pulses regenerated. Correctly timed narrow pulses are regenerated by the clock pulse gate which gives both a normal output and a 'NOT' output.   E.g.   P1   P2   P3   P4

1   0   1   1



J is normally connected to K but in some cases, for logical operation, signals from J and/or N are taken to a special unit before being returned to K.

At the source gate, which comes next, a copy of the circulating word may be extracted onto the "highway", the normal transfer path in the machine. The "highway" has an amplifier between the source and destination sides which introduces a 0.15 μsec. delay in transmitting words. The source side is therefore termed "highway early" and the destination side "highway late". A copy of, the word(s) circulating will only appear on the highway early (HWE) when the source control signal (Sc) and source selection signal (Sn) are applied to the source gate simultaneously.

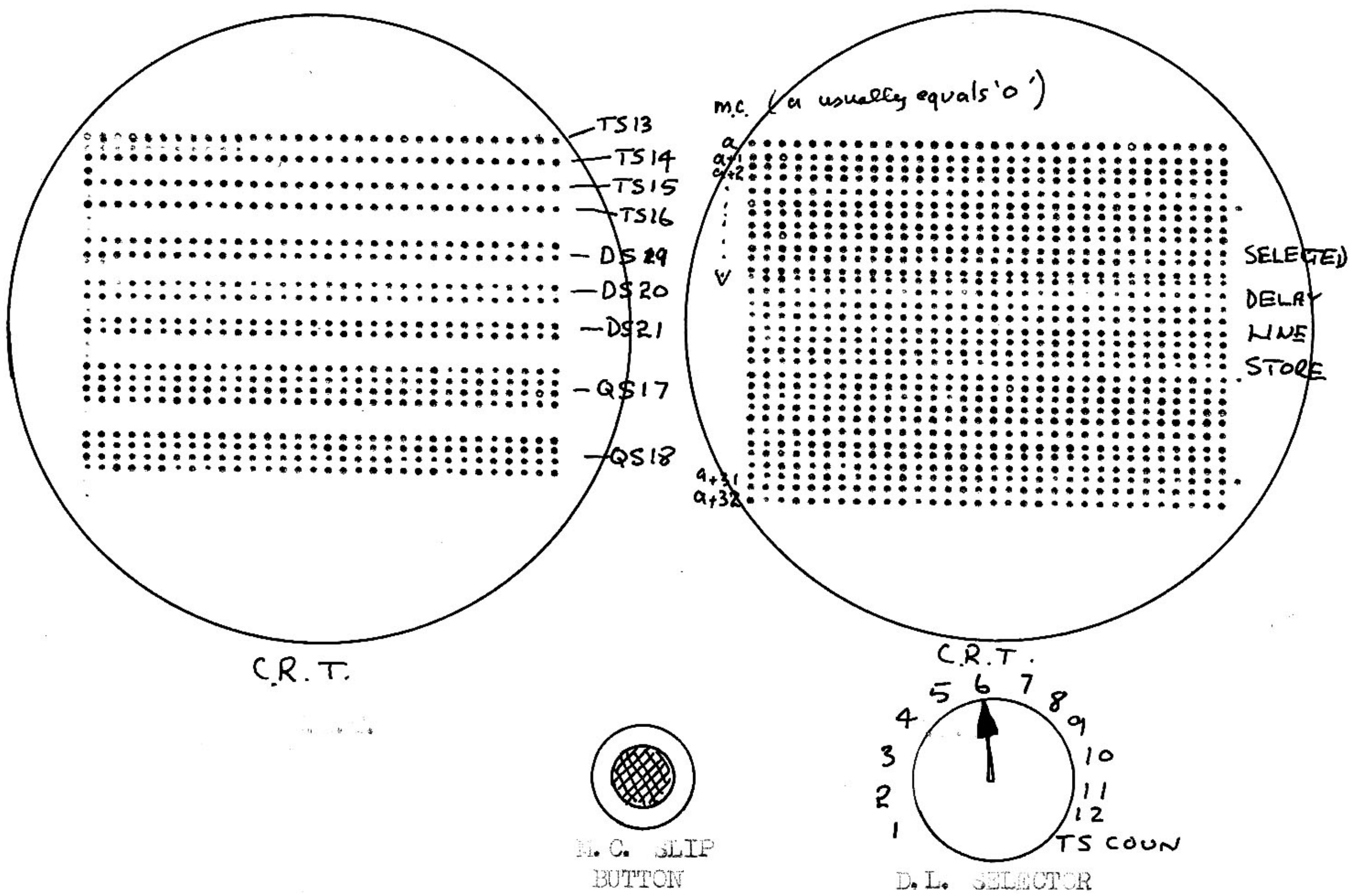
There then follows a 0.15 μsec. delay—to compensate for the delay in the highway amplifier—followed by the destination gate which operates similarly to the source gate except that it breaks the circulation path when operated to allow for the insertion of a new word into the store from the highway late (HWL). Signals TT and Dn (control and selection respectively) must be applied.

The next instruction source gate (NIS gate) is next. This operates in the same way as the source gate except that copies of the stored word pass onto the instruction highway (IHW) not the highway early. Only the NIS gates of delay lines 1 to 8 are in fact connected to the instruction highway. Signals NISc and NISn (control and selection respectively) must be applied.

In TS COUNT, the control store, none of the 3 gates is connected normally. They are used for special purposes.

There are several types of store employing mercury delay lines. One is the control store TS COUNT mentioned above. There are four other TS's or temporary stores which are associated with the logical and arithmetic operations of the machine, TS 13, TS 14, TS 15, and TS 16. They each will hold one word which is, therefore, available in every minor cycle of the machine's operation. There are then three double stores—or DL's—which hold two words. Clearly each word is available only every other cycle of the machine's operation. Connected with automatic instruction modification are two quadruple stores (or QL's) in which four words are available every fourth m.c. And finally there are the 12 so called delay line stores, or DL's, each of which holds 32 words and in which a given word is only available for 1 m.c. in every 32 m.c.—i.e. once every major cycle (M.C.). A useful diagram showing this and other information appears as Appendix 1.

The contents of these stores—the high speed stores—are displayed on the machine console by two cathode ray tubes layed out thus:



Each row is one word, each dot is one digit, glowing brightly for a '1' and hardly visible for a '0'. The m.c's are displayed in order and the 'scope is said to be 'lined up' when m. c. '0' is at the top of the display, though, of course, the position on the display does not affect the machine's operation. 'Lining up' may be achieved by pressing the m. c. slip button a number of times until the correct m.c. is at the top.

A given word in a delay line store is referred to by the number of the delay line and the particular minor cycle of storage (say $DL4_{15}$ for the 16th word in Delay Line 4).

5. Control

The instruction word consists of seven numbers expressed by the 32 digits in the following manner.

DIGITS	SYMBOL NAME	POSSIBLE VALUES	DESCRIPTION
2 - 4	NIS (Next Instruction Source)	0 - 7	Number of Next Instruction Source gate
5 - 9	S (Source)	0 - 31	Number of Source gate
10 - 14	D (Destination)	0 - 31	Number of destination gate

DIGITS	SYMBOL NAME	POSSIBLE VALUES	DESCRIPTION
15 - 16	C (Characteristic	0 - 2	Length of transfer required
17 - 21	W (Wait number)	0 - 31	Number of idle minor cycles from set-up to the first minor cycle of transfer
26 - 30	T (Timing number	0 - 31	Specifies the minor cycle of the next instruction and, for a long (C = 1) Characteristic, the last minor cycle of transfer
32	G (Go digit	0 or 1	Tells the machine whether it can obey the instruction without waiting for a "single shot"

Digits 1, 22 - 25 and 31 are spare and their value has no effect on the instruction, when obeyed, though they are "used" during modification of instructions, which function will be explained later.

Minor cycles of operation are referred to the m.c. in which the current instrtction is stored and, therefore, in which it enters TS COUNT. Let this be m.c. 'm'. Then:

(i) "Set up" is in m.c. (m + 1)

(ii) Transfer commences in m.c. (m + W + 2)

(iii) The next instruction is taken in m.c. (m + T + 2).

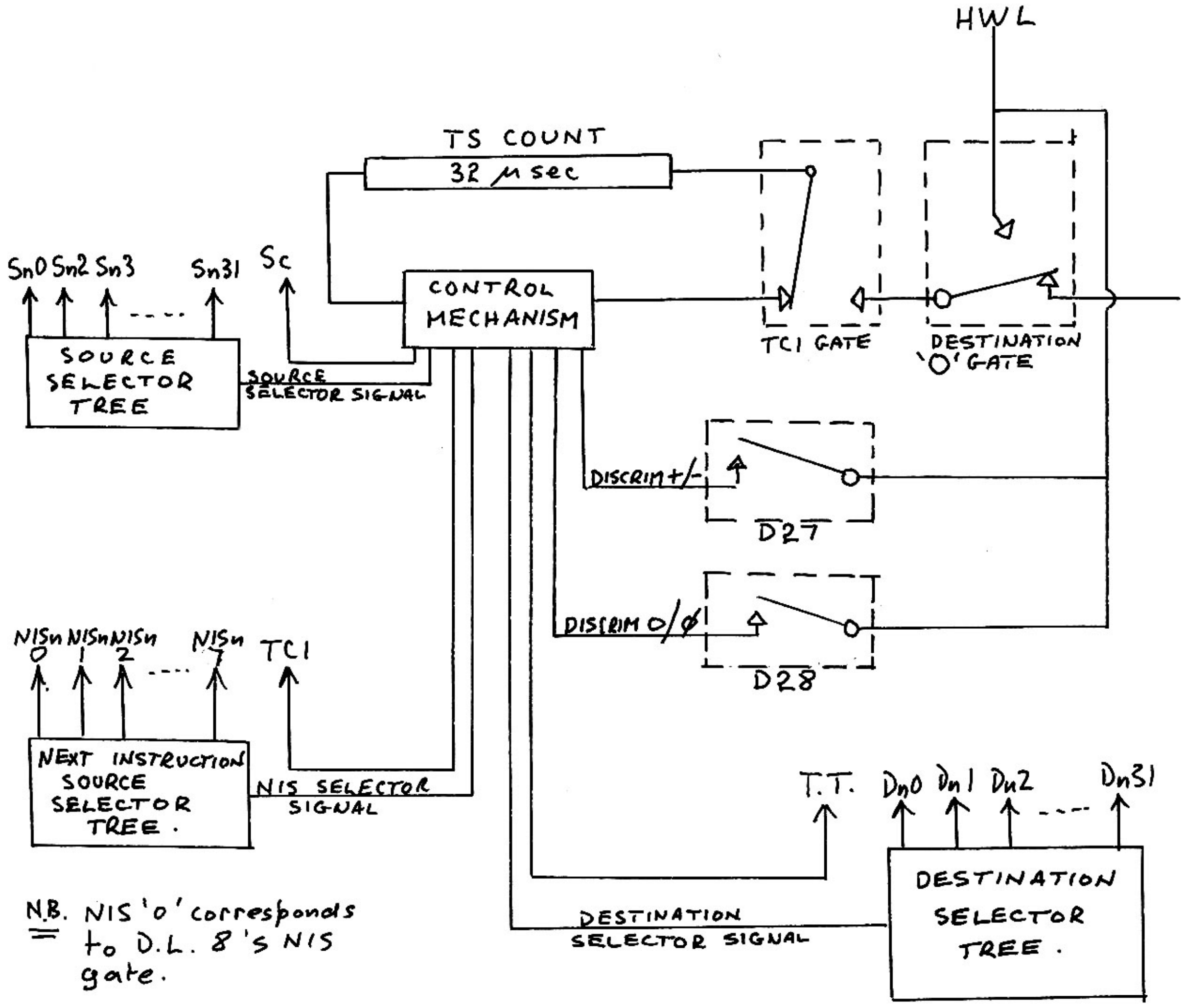
The above apply for any value of the characteristic C.

There are then 3 alternatives:

	Short Transfer C = 0	Long Transfer C = 1	Double Transfer C = 2
TRANSFER OCCURS	m.c. (m + W + 2)	from m.c. (m + W + 2) to m.c. (m + T + 2)	in m.c. (m + W + 2) and m.c. (m + W + 3)

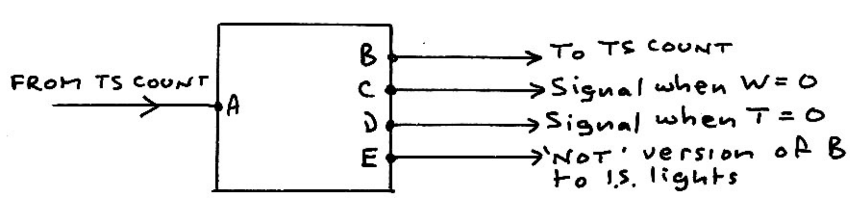
If W > T (or W = T and C = 2) then T has effectively the value (T + 32) since last m.c. of transfer cannot precede first, nor can the next instruction enter control while the current instructions in still being obeyed. The fourth possible value C, i.e. 3, has the same effect as C = 1 since the effect of the P16 digit is inhibited by the P15 digit.

Operations



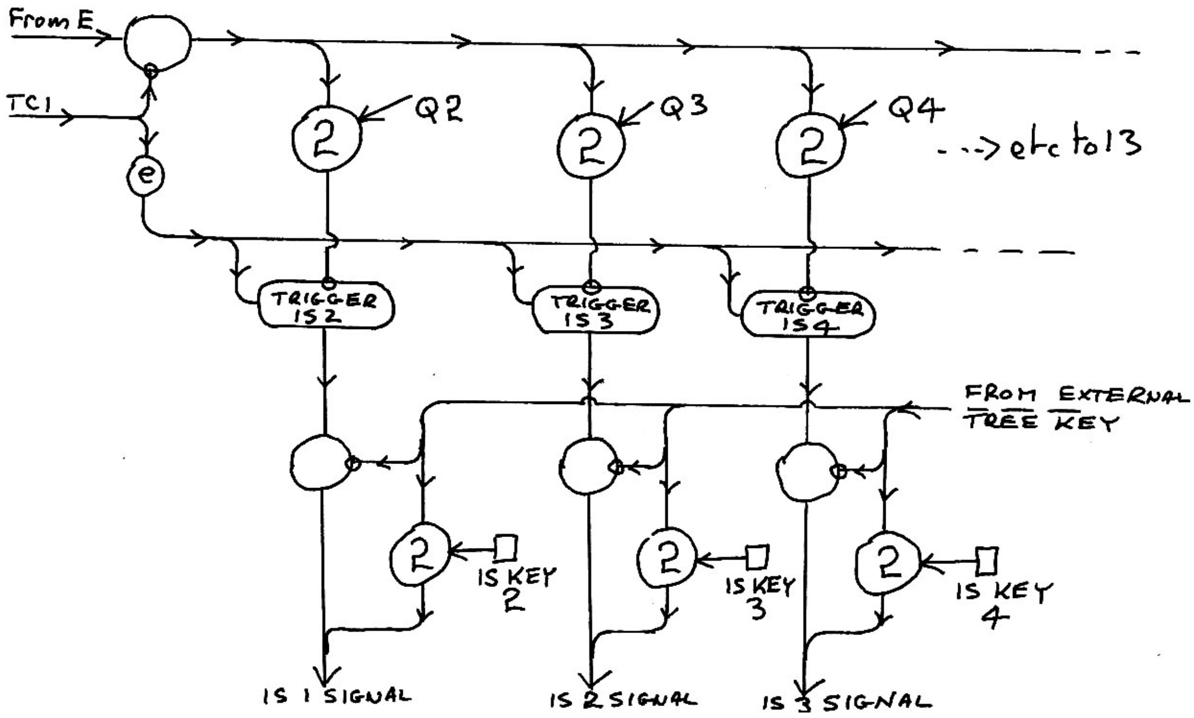
Normally the word for TS COUNT enters from the instruction highway which is connected to the NIS gates of delay lines 1 to 8. However by using destination 0 (see below & Appendix I), and ensuring that the TCI and TT signals are applied in the same m.c., words from any store may be sent to control. This can be useful for obeying instructions from a store where they have been constructed or modified.

The main part of the control mechanism is the unit subtracted.



The instruction word enters A and leaves B with 1 subtracted from the wait and timing numbers. A signal is produced at C and B when wait and timing respectively reach zero. Thus C emits a signal W m.c. after the word enters TS COUNT and starts TT, while D emits a signal T m c. after the word enters TS COUNT initiating the end of TT and marking the m.c. of TCI.

Finally E emits a 'NOT' version of the instruction word which goes to the Instruction Staticisor (IS).



If the logical operation of this diagram is followed it can be seen that in the event of a particular digit—say P2—being present in the instruction word there will be no signal from E in Q2 time and therefore trigger IS2 will remain on, having been switched on by the last clear TS signal): the clear IS signal is a short pulse at the end of the TCI signal, thus it occurs just after the instruction word enters TS COUNT but before the first digit reaches J/N. The signal from the IS trigger is used to set up source , destination and NIS selector trees.

The outputs of the IS trigger may be replaced by signals from the thirteen IS keys by applying a signal to the external tree line by depressing the external tree key, and operating the IS keys (see page 4).

The instruction in control will not be obeyed until the trigger GO has been stimulated. This may be done by either the P32 digit of the instruction word or by a "single shot signal". The instruction word will then be obeyed in the m.c. determined by the wait number and the next instruction will enter control in the m.c. determined by the timing number. The unit subtracted counts down continuously. Thus if the wait number reaches zero, and the 'GO' trigger has not been stimulated the count down starts again from 31. Thus when the 'GO' trigger is finally stimulated the instruction will be obeyed in the correct m.c. a whole number of major cycles (M. C.) later. The entry of the next instruction in the correct minor cycle is assured in a similar manner and explains why, if T < W—or T = W for a double—transfer then the effective value of T is (T + 32). “Single shots” may be supplied by the reader, punch or “single shot” key.

The distinction between single and double transfers is made by another trigger which is stimulated by the P16 digit of the instruction word. If there is a P15 digit the action of the P16 digit is inhibited. Thus from the two digits of the characteristic, P15 and P16, only 3 effects may be produced. ‘00’ gives a single transfer, ‘10’ gives a long transfer, ‘01’ gives a double transfer and ‘11’ gives a long transfer the action of the second ‘1’ or P16 digit being inhibited so that it is effectively ‘10’.

Control magnetic interlock (CMI)

Any operation involving transfer to or from the magnetic drum store taken 13 millisecs. To save time, such an operation, once started, proceeds to completion without further action from control, leaving control free to continue implementing other instructions. Clearly two magnetic transfers cannot take place at once, therefore if a magnetic instruction is being obeyed, a control magnetic interlock (CMI) signal is emitted which prevents another magnetic transfer instruction from being implemented until the flrst has finished. Some complication is in fact involved to prevent a magnetic transfer instruction from inhibiting its own execution.

Trigger TCI

The “timci” signal (TCI), which allows a new instruction into TS COUNT is supplied by a trigger which is put on during m.c (m + T + 2). A manual key may be used to stimulate the trigger continuously.

Trigger TT

The “transtim” - signal (TT), which allows transfer between the selected source and destination, is supplied by a trigger which is put on during m.c. (m + W + 2) and may then go off at the end of m.c. (m + W + 2) or (m + W + 3) for single or double transfer respectively; or, for a long transfer, at the end of m.c. (m + T + 2). A manual key may be used to stimulate the trigger continuously.

Stop release and single shot keys (and trigger STOP)

A stop key is provided which has three positions: “STOP”, “NORMAL”, and “AUG. STOP” (augmented stop). With the stop key at

“NORMAL” the machine will carry our instructions at high speed without stopping,until an instruction enters TS COUNT which has no P32 digit set to ‘1’. A “single shot” must then be given (by reader, punch, or single shot key) to make the machine carry on. With the stop key at “STOP” the P32 digit is prevented from stimulating trigger GO and a single shot must be given to cause each instruction to be obeyed, regardless of the presence or absence of a P32 (‘GO’) digit. With the stop key in “AUG. STOP”, trigger STOP is stimulated. The P32 digit of a “goer” instruction will clear trigger STOP but a single shot is necessary to cause the instruction to be obeyed. If the P32 digit is ‘0’ in an instruction then trigger STOP makes single shot ineffective and the programme can only continue when a special release is operated and even then only if a single shot is supplied. The “AUG. STOP” position is used, in conjunction with the programme display facility (see page 10). Single shots may be supplied from a single shot key by depressing it. Raising it causes single shot signals to be supplied at a rate ten per second. Single shot signals are also emitted by the reader or the punch when a card row is ready for reading or punching. Thus, for example, all instructions transferring to the punch are made “stoppers”, and each remains circulating in TS COUNT until a card row comes under the punch knives. A single shot then causes the instruction to he obeyed. This slows Deuce to the speed of the Hollerith punch.

Discriminations

If a word is sent to destinations 27 or 28 Deuce can discriminate in the former case between positive and negative, and in the latter case between zero and non-zero. This is achieved in the following manner. If the word sent to D27 is positive or that to D8 is zero then the machine behaves normally and takes the next instruction according to the timing number of the discrimination instruction (this is called the early path). If, however, the word is negative or non-zero, as appropriate, the machine selects, instead, the instruction stored in the minor cycle immediately succeeding the the instruction it should have taken in normal circumstances though, of course, from the same store (this is called the late path). The modifying trigger is stimulated by the presence of a P32 digit in the first case (indicating a negative numer) or, by the presence of any digit in the second case and causes TCI to be applied during m.c. (m + T + 3) as well as (m + T + 2) thus effecting the instruction replacement and inhibiting the interpretation of the first instruction selected.

Programme display

Actuation of the programme display key on the console causes special connection, to control, to trigger PUNCH, to the output staticisor lights, and to a special trigger,(H), to be made. Trigger H inhibits normal destination 29 instructions and also causes each instruction, as it is obeyed-to be sent to destination 29, which is the punch when trigger PUNCH is stimulated. The stop key must be in “AUG. STOP”. Thus as each instruction of a programme is obeyed the machine stops and the instruction is punched out and displayed on the lights. If the instruction was a “goer” then the single shot from the punch, as it punches the instruction, causes the next instruction to be obeyed and sent to D29. If the instruction was a “stopper” (P 32 = 0) the machine stops altogether and the programme can only be continued by operating the release key and giving a single shot manually. This facility is only used for programme testing.

Destination triggers

If an instruction calls for destination 24 the source control signal is supressed and the source selector signal is used to select one of the so-called “destination triggers”. These triggers stimulate the operation of certain facilities and do not refer to normal transfers at all. Thus 0—24 stimulates the automatic multiplier, 10—24 stimulates the punch,. and 9—24 clears the punch (or reader). For a full list see Appendix I.

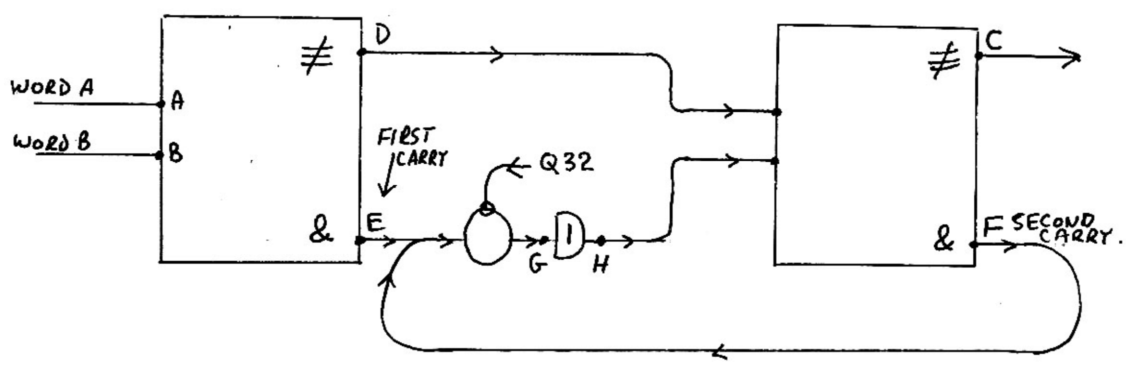
6. Local operations, sources, destinations and some machine facilities

Sources and destinations

For an exhaustive table of the significance of the various sources and destinations (32 of each, nuubered 0 to 31) see Appendix I. In general sources and destinations 1 to 21 refer to information stores while sources and destinations 0, and 22 to 31, have specia] significance. For example, source and destination 7 refer to delay line 7, while source 24 gives a copy TS 14 multiplied by two; and destination 24, of course, refers to the destination triggers. The source and destination, being the primary operating bits of the instruction word, are used by themselves in initial programming with side notes to indicate important things such as the characteristic. The storage is then organised in conjunction with the wait and timing numbers and thus the "housekeeping" parts of the instruction word may be added. Instructions are thus often considered in their basic form of S—D which explains the form of the instruction mentioned at the end of the last section.

Adder/subtractor

A word transferred to destination 25 is added to the contents of TS 13; and one transferred to destination 26 is subtracted. The process is fairly simple since, for a particular digit position, if both numbers have '0' the sum is '0'; if either has a '1' and the other '0' then a '1' appears in the sum. These two results can be obtained by a NOT gate. If both numbers have a '1' then a '1' must be carried to the next highest digit. This is effected thus:



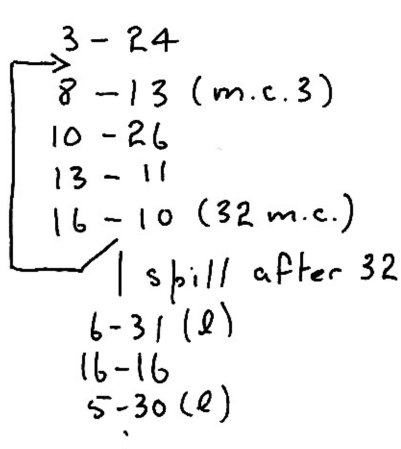
	1	2	4	8	16	32	
A	1	1	0	1	0	0	= 11
B	0	1	1	0	0	0	= 6
C	1	0	0	0	1	0	= 17
D	1	0	1	1	0	0	
E	0	1	0	0	0	0	
F	0	0	1	1	0	0	
G	0	1	1	1	0	0	
H	0	0	1	1	1	0	

Words A and B are fed into a gate having a NOT (≠) output and an AND (&) output. The NOT output is the sum of the two words except that there has been no carry (i.e. 0 with 0 gives 0; 0 with 1, or 1 with 0, gives 1, a 1 with 1 gives 0).

The "AND" output gives the carry which must be added to the next most significant digit position. Since the digits occur one after another at 1 μsec. intervals, with the least significant first, the application of the partial sum together with the "AND" output delayed by 1 μsec. to a second "AND/NOT" gate will give a better sum at the second "NOT" output with a second carry at the "AND" output. It would appear that a third gate would be required and so ad infinitum but the first and second carries can never be '1' simultaneously for if E is '1' then A and B are both '1' ,so D is '0' and therefore F is '0' for either value of H. The output of F can therefore be brought back to the input of the original delay without risk of confusion. Any carry from the P32 position must he suppressed (except for double length addition and subtraction in the adder associated with DS21) and a Q32 signal inhibits such carry to prevent it going to the first digit of the next word. Subtraction is accomplished by adding the negative of the word required.

Multiplying and dividing. Trigger TCA, TC, Bound ODD

Of the five machine facilities—shift, add, subtract, multiply and divide—all but the first use adder/subtractor units of equal or greater complication than that described above. There is a sixth machine facility, which is not arithmetical, called TCA. This is operated by trigger TCA. When TCA on the normal circulation path of TS16 is broken and a connection is made with DL10, copies of the word stored in that delay line pass into TS16 and are subject to a 32 μsec. delay (1 m.c.). Thus with T.CA 'on' the instruction "16—10 (m.c. 5)" would replace the word stored in m.c. 5 of DL10 with a copy of that stored in m.c. 4, since S16 emits the same as S10 except that everything is delayed by 1 m.c. (see Appendix I). The instruction 16—10 (32 m.c.), i.e. a long transfer, would shift all the words stored in DL10. up 1 m.c.. In this way numbers could be stored sequentially DL10 and operated on it turn by the same instruction.

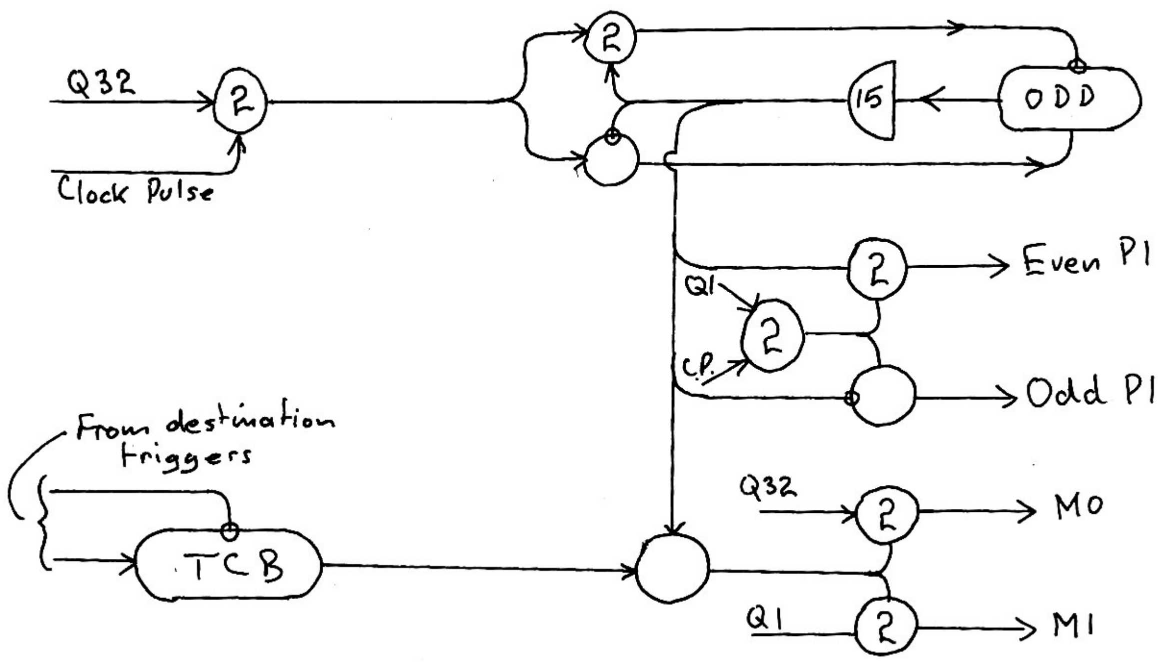


This sequence of instructions would take the number stored in m.c. 3 of DL8, and subtract from it each of the numbers stored in DL10, in turn, putting the 32 answers in DL11 and then storing them on track 6/5 of the magnetic drum store (see page 25). TCA. is brought on by a destination trigger instruction, 3—24, and cleared automatically when any transfer to destination 16 is made.

For multiplication and division TS16 and DS21 are used. The multiplicand or divisor is placed in TS16 and the multiplier or dividend placed in the odd m.c. of DS21 i.e. 21_3. Multiplication or division can then be carried out by selecting the appropriate destination trigger. A curious point should be noted here. For the double stores, instead of the two m.c.'s of storage being ‘0’ and ‘1’ they are ‘2’ and ‘3’. This is probably to avoid confusing the even ‘0’ with ‘0’ standing for odd when programming or coding. Oddness and even-ness are important with respect to a double length word since the P1 of the even m.c. is the least significant digit and the P.32 of the odd m.c. is the most significant digit of a 64 digit word when multiplying or dividing.

Thus, if “rounding off” to a 32 digit number, the even minor cycle must be discarded and any carry made to the P1 position of the odd m.c. A trigger (trigger ODD) distinguishes between even and odd minor cycles, being arranged to be on in odd m.c.'s and off in even ones. Thus the machine has odd and even m.c. built into it, though it has no sense of m.c. numbering. A programme must, therefore, be written so that, as it is run into the machine, it will automatically select an even m.c. to label m.c. ‘0’.

Another trigger must be considered in connection with DS21—trigger TCB. With TCB on DS21 is treated as an ordinary double store. With TCB off, however, DS21 assumes it's role of a 64 digit word store and any carry from the P32 position of the even m.c. will go to the P1. position of the odd m.c., instead of getting “lost”, and the sign of the number is indicated by the P32 digit of the odd m.c. Trigger TCB is put on by the destination trigger instruction 5—24 and off by 4—24.



With TCB off MO gives odd Q32's and M1 gives even Q1's. (It should be noted that the signal from trigger ODD is subject to a 15 μec delay and therefore covers the end of odd m.c.'s and the beginning of even ones). With TCB 'on' MO and M1 give Q32 and. Q1 respectively.

Shifting down double length

Source 22 gives a copy of DS21 divided by 2, i.e. shifted down one place.

DS21 A1, A2, … A31, A32 B1, B2, B3, … B31, B32

S22 A2, A3, … A32, X B2, B3, … B31, B32, B32,

If TCB is 'off' 'X' should be B1. If TCB is 'on' 'X' should be A32, as the even and odd minor cycles would be separate words each having a sign digit to be duplicated in the P32 position. The gating to ensure this makes use of the MO signal which gives an odd P32 or both odd and even P32 depending on the state of TCB.

Shifts down of up to 16 places may be performed with one instruction on a 64 digit number(TCB 'off'), or on two 32 digit numbers (TCB 'on'). The shift must be in order even m.c., odd m.c., (e, o).

The Long Accumulator

Words sent to D22 are added to the contents of DS21, those sent to D23 are subtracted. Some elaboration of the circuitry enables not only full double or single length working, but also mixed single/double length working. It is important to specify (e,o)

The Automatic Multiplier

To perform multiplications the multiplicand is sent to TS16 and the multiplier to $DS21_3$. $DS21_2$ is cleared and trigger MULT (multiply) stimulated by the instruction 0—24 (odd). After 65 m.c. the multiplication is complete and trigger MULT is automatically cleared. The 64 digit product is in $DS21_{2 \& 3}$ and the multiplier is lost. Programming rather than machine design deals with the sign convention.

Multiplication is achieved easily since multiplier digits are either '0' or '1'. At each stage the top digit of the 64-digit word in DS21 is examined, the word is shifted up one place and a copy of the multiplicand either added or not added into $D.S.21_2$ according to whether the digit examined was '1' or '0' respectively. The top digit is examined during the addition using one minor cycle and the shift uses another minor cycle. As the multiplier has 32 digits a multiplication will take (64 + 1) minor cycles, the extra minor cycle being taken up in the initial set-up, before adding in has started. After 65 m.c. trigger MULT is automatically switched off. To make the process clear a 4-digit model will be used to multiply 14 by 13.

Number in TS16 = 14 = 0111 (multiplicand)

Number in $DS21_3$ = 13 = 1011 (multiplier)

$DS21_2$ cleared.

	$DS21_2$				$DS21_3$				Action carried out by m.c. stated
	P1	P2	P3	P4	P1	P2	P3	P4	
Initial	0	0	0	0	:1	0	1	1	Trigger MULT 'on'
After 1 m.c.	0	0	0	0	:1	0	1	1	Top digit $DS21_3$ (td) is (1)
After 2 m.c.	0	0	0	0	0	:1	0	1	$D.S.21_{2 \& 3}$ shifted one place
After 3 m.c.	0	1	1	1	0	:1	0	1	TS16 added to $DS21_2$ td is 1
After 4 m.c.	0	0	1	1	1	0	:1	0	$D.S.21_{2 \& 3}$ shifted one place
After 5 m.c.	0	1	0	1	0	1	:1	0	TS16 added, td is 0
After 6 m.c.	0	0	1	0	1	0	1	:1	$D.S.21_{2 \& 3}$ shifted one place
After 7 m.c.	0	0	1	0	1	0	1	:1	No addition, td is 1
After 8 m.c.	0	0	0	1	0	1	0	1	$D.S.21_{2 \& 3}$ shifted one place
After 9 m.c.	0	1	1	0	1	1	0	1	TS16 added, trigger MULT 'off'

01101101 is the binary form of 182—the correct answer (remember, the binary number has the least significant binary digit on the left). It can be seen that multiplication is simply a matter of additions and shifts. TCB must be "off" during multiplication, a beginning element from trigger MULT ensures this automatically.

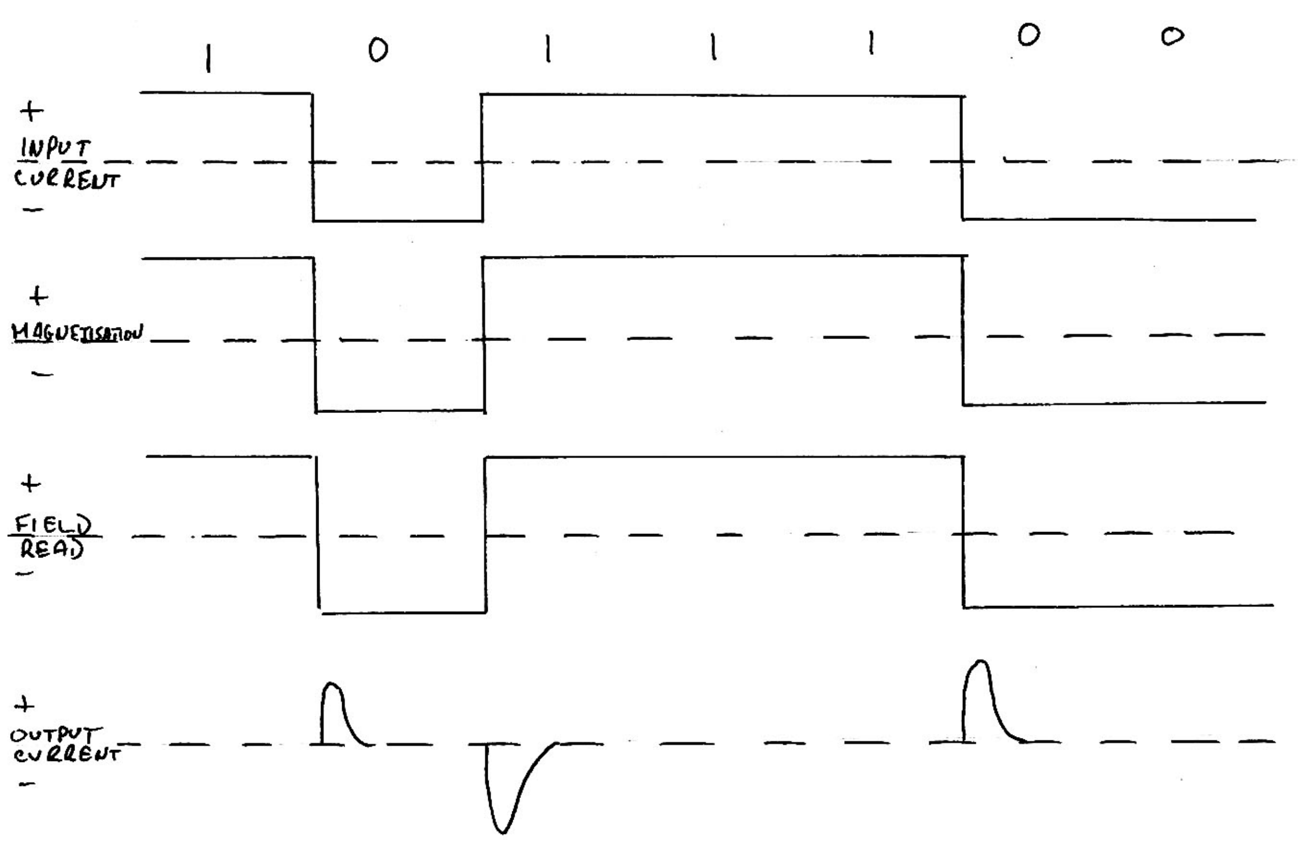
Automatic Divider

To perform division the divisor is put in TS16, the dividend is put in $DS21_3$, and then trigger DIV (divide) is stimulated by the instruction 1—24 obeyed in an odd minor cycle. In 65 m.c. the division is complete and trigger DIV is automatically cleared. The quotient and modified remainder are then in $DS21_2$ and $DS21_3$ respectively, and the divisor remains in TS16.

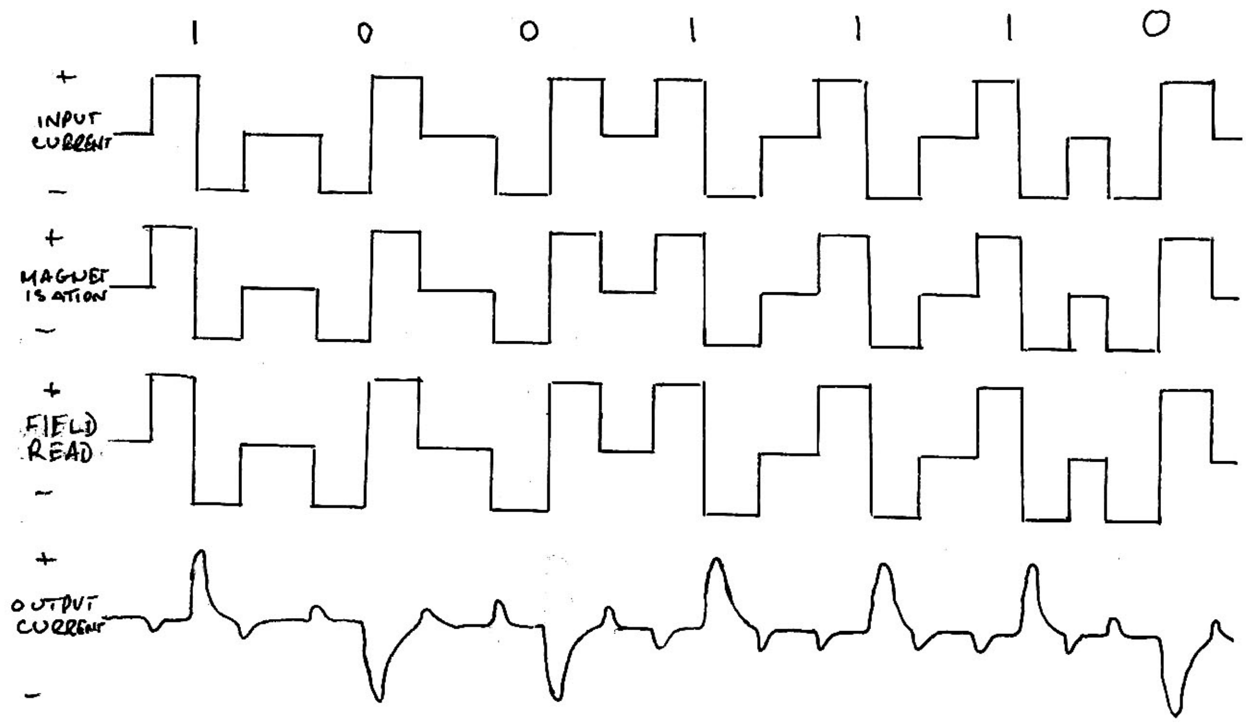
The Magnetic Drum Store

Construction and Operation

The magnetic drum store consists of a cylinder about 6 inches long and 4 inches in diameter whose curved surface is coated with material capable of being magnetised. A block of reading heads and a block of writing heads enable electrical pulses to be read off the drum or stored on it as the drum rotates. Now it is a feature of magnetisation that, when converting an electric current into magnetisation of some material, the magnetisation is directly proportional to the current. But when converting magnetisation back to an electric current the voltage and hence the current produced is proportional to the rate of change of magnetic field produced by the magnetisation as the drum surface passes under the reading head. Thus, if the input were a current one way for ‘1s’ and the other way for ‘0s’ the input and output would be related in this sort of manner:



Interpretation of this output signal would be rather difficult and therefore the system is adopted in which, for a '1', the current changes from positive to negative in the middle of the digit and, for a '0', the current changes from negative to positive in the middle of the digit. The relation between the input and the output then becomes:

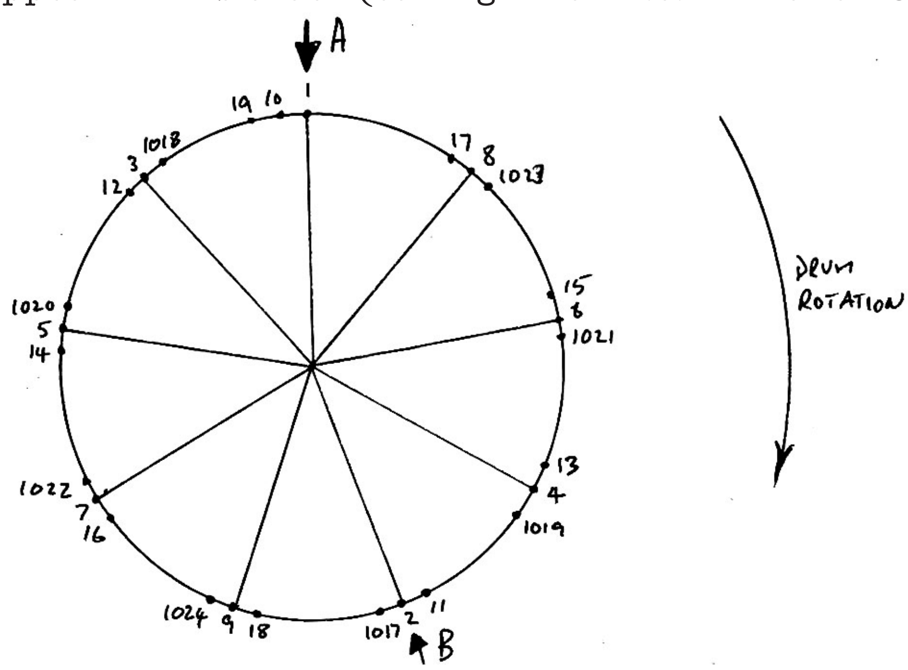


This output signal is readily interpretable.

It is possible to put 1024 dipoles round the drum circumference on one track which can therefore hold the equivalent of one DL There are 16 head-block positions, and 16 read or write heads in each block, giving 16 x 16 = 256 tracks or 32 x 256 = 8192 words. For the complication of eight DLs an extra storage capacity equivalent to 256 DLs is available. However, access time suffers for the DLs have a maximum access time of something of the order of 1 millisecond whereas the access time for a track on the drum is of the order of 13 milliseconds whilst the head-block shift can take 35 milliseconds in the worst case. The latter time is the inevitable result of using a mechanical device having inertia in its moving parts; the former time is a result of the mechanism of transfer and also associated with the limitations of mechanical systems.

Straight transfer of consecutive digits from the drum to a DL would require a drum rotational speed of 60,000 R.P.M. Instead the drum rotates at about 6,575 R.P.M. and every 9th digit of the words in DL11 is written on the drum for 9 major cycles (M.C.'s). The timing is carried out by a magnetic clock pulse (MCP) signal which gives a 1 µsec.

pulse every 9 µsecs. There is of course no particular virtue in starting reading or writing at a particular spot on the drum provided the timing and synchronisation are correct with respect to the magnetic clock pulses. To effect this a disc with 1024 teeth is attached to the drum and from these a reading head generates 1024- pulses each revolution. These are called drum pulses or DPs. The MCPs and DPs are fed into a discriminating circuit which compares their phase the output being a function of the phase difference between them. This is an error signal which can be used in control of the drum speed and also to stimulate an alarm if the error exceeds certain limits. Thus the digits recorded on a drum would appear in this order (calling P1 of m.c. 1 '1' and P32 of m.c. 32 '1024-'):



The reading head is at A and the writing head at B thus B's magnetic clock pulse must be delayed by 1/4 sec. and then both reading and writing heads will synchronise with the digits in the DL. It can be seen that there must be a fractional displacement from the position shown so that when the 10th major cycle starts then the '1' position will again be under A (rather than the position just before it). Note that each digit signal to the drum is effectively a 9 µsec signal rather than a 1 µsec signal. The polarity reverse occurs in the middle of this 9 µsec period.

Drum instructions and use.

S—D Ch (s = short characteristic, l = long characteristic)

S—30 s will read from head S

S—30 l will write to head S

S—31 s will shift the reading head block to position S

S—31 l will shift the writing head block to position S

The track numbering corresponds with head positions and heads in the following manner.

Track number	Headblock position (P)	Head (H)	P/H Track number
0	0	0	0/0
1	0	1	0/1
2	0	2	0/2
:	:	:	:
:	:	:	:
14	0	14	0/14
15	0	15	0/15
16	1	0	1/15
17	1	1	1/1
:	:	:	:
:	:	:	:
239	14	15	14/15
240	15	0	15/0
241	15	1	15/1
:	:	:	:
:	:	:	:
255	15	15	15/15

Thus the track number = (P x 16 + H) and to write on track 239 the instructions:

14—31 1

15—30 1

must be given. To read from track 121 the instructions:

7—31 s

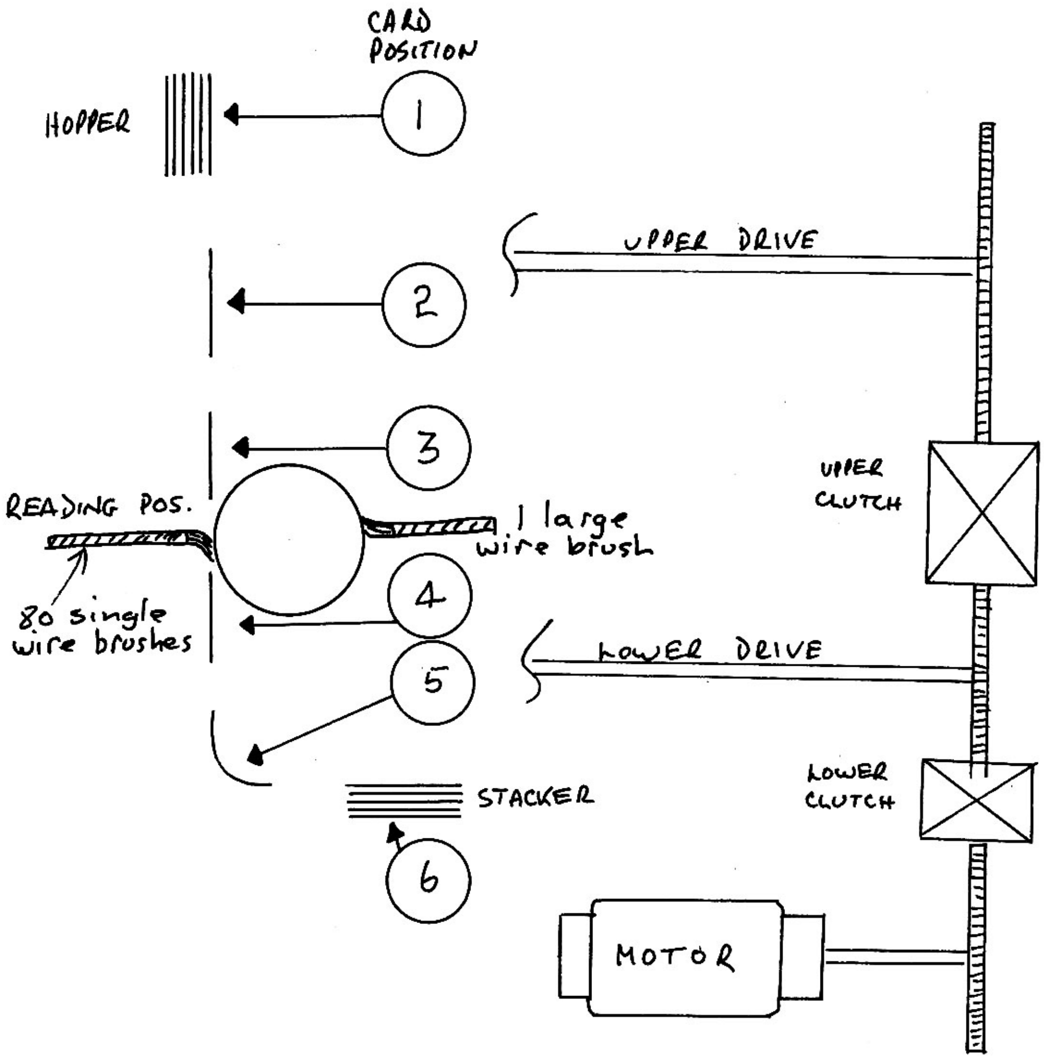
9—30 s

must be given. All reading and writing is done to or from DL11.

The whole set of 256 tracks can be filled with information from punched cards in about four minutes and therefore the additional complication of maintaining absolute rather than relative synchronisation was not considered worth while.

8. The Reader and the punch

The reader and punch are very similar in principle so the former will be described and then differences noted.



Cards to be read are placed in the hopper (1) and feed through to the stacker (6). In between are positions 2, 3, 4 and 5. The upper drive moves the cards from 1 to 2, 2 to 3 and 3 to 4, while the lower drive moves them from 4 to 5 and 5 to 6. The upper cannot operate without the lower but the lower can operate without the upper. There are three types of cycle: feeding, reading, and run out. Cards are placed in the hopper and the “run-in” key depressed. Two feeding cycles will then occur, bringing a card to position 3 and stimulating a “ready” relay. In this state a 12—24- instruction, stimulating the trigger READ of Deuce, will cause the reading cycles to start and source 0 to refer to the reader brushes rather than the input dynamicisor lights. In each reading cycle a card passes from position 3 to position 4, and is read. As each card row passes the reading brushes the reader emits a single shot signal (see page 14). As the twelfth row passes the brushes a “twelfth impulse line” or TIL signal is emitted. Discrimination on the presence or absence of the TIL signal is possible by the instruction 2—24, which stimulates destination trigger TIL DISCRIM. The early path is followed with TIL ‘off’ and the late path with TIL ‘on’. A key on the console allows this to be forced, one way or another, regardless of the state of the TIL signal.

When the required information has been assimilated the instruction 9—24 is used to clear the trigger READ. No further single shot or TIL signals are emitted, and source 0 reverts to the input dynamicisor lights, even if it is the middle of a card cycle. At the end of the current card cycle the upper drive is disengaged and two cycles later the lower drive stops. Thus the reader is left in the state it would be at the end of 'running in'—cards in positions 1, 2 and 3 only. Instructions to read (or punch) card rows are always made ‘stoppers’. Thus if the trigger READ (or PUNCH) were stimulated, and no reading cycles had been executed, the programme would remain stuck at that point, the next instruction would not be obeyed until the first row of a card came under the reading brushes (or punching knives) and a single shot signal was emitted. Single shots would still be required for subsequent rows. This slows Dence to the speed of the Hollerith machines and ensures correct timing. Raising the run-in key clears the ready relay and stops the reader whatever it is doing. To restart it the run-in key must be depressed again. In addition to the run-in key there is a run-out key, which actuates the lower drive only, and clears the machine of cards; and an initial input key (reader only). The initial input key has three effects, it clears all DL's, stimulates trigger READ, and initates the run-in key. It is used for the input of a new programme.

A maximum of 64 card columns (of the 80 columns in a12-row, 80 column standard Hollerith card) may he read by Deuce.

The Punch

The punch is very similar to the reader, though it has solenoid operated knives rather than reading brushes. When trigger PUNCH is stimulated by the instruction 10—24, a signal is emitted to clear the output staticisors, a connection is made from D29 (output staticisor lights) to the punch and, if the punch is ready, punching commences.

There are only 5 card positions in the punch, one of the positions before the working portion being omitted. Also the machine operates in a horizontal rather than a vertical plane. The first sixteen punch knives are operated by preset switches and a counter, leaving 64 available for Deuce instruction or data words.

32 or 64 column working

In the 64 columns available to Deuce it is clearly possible to have two 32 digit Deuce words end to end. Columns 17 to 48 are referred to as the α-field and 49-80 as the β-field. For 32-column working the α-field only is used, for 64-column working both are used. Now for α-field working the method of reading is as described, the α-field being read by means of “stopped” instructions. As soon as this instruction has been obeyed the machine will start to switch over to the β-field, the change over taking the following form:

m.c. after obeying ‘stopped’ instruction	field available at source 0 (S0)
0-4	α-field still available
5-21	mixture—gives rubbish
22 ... up to 2 Major Cycles	β-field available

Thus steps in reading from both α and β-fields on a card are:

(a) Read α-field with a stopped instruction.

(b) Take further copies (up to two) in next 4. m.c., if required, with an unstopped instruction.

(c) Wait at least 22 m.c. after the stopped instruction was obeyed (the time can be used for some other part of the programme in which case the wait number of the β-field instruction may be reduced by [the timing number + 2 (set up m.c.'s)] for each intervening instruction).

(d) Read the β-field with an unstopped instruction.

(e) Obtain further copies (if required) of the β-field for up to two Major Cycles after the stopped instruction.

Punching both α- and β-fields is similar, except that the time factor is less important, also an "8—24 1" instruction must be inserted between the instructions to punch α- and β-fields which switches the fields and clears the output staticisor triggers. The unstopped instruction to punch the β-field must be obeyed within 4 Major Cycles of the stopped instruction which punched the α-field. If β-field punching only is required the 8—24 1 instruction should be made a stopper, i.e. 8—24 1 x.

Other equipment associated with Deuce

Paper tape reader and punch

Information may be fed into or extracted from Dence by means of holes punched in paper tape rather than cards. The tape is continuous and has rows of holes of either five, seven or eight characters each.

Reading is accomplished by passing the tape between a light source and a set of photo-electric cells and feeding the resultant signals into Deuce. Punching is much the same as card punching though very many less punch solenoids are required. The paper tape punch is quite a small unit and rather slow.

Magnetic tape reader

The Short's Deuce has had extra circuitry added to enable magnetic tape produced by the instrumentation system on the "Belfast" flight test aeroplane to be fed directly into it.

The tape is played back on an Ampex PR 400 reader and the resulting signals are used by Deuce in the same way as signals from card or paper tape reader.

Information is on the tapes as 16-bit binary words written across the tape. Twelve of these are used to code information and the remaining four are a sign bit, a parity bit, a clock bit and a marker bit. It is relevant here to explain the function of the parity and clock tracks. Magnetic tape suffers from "drop-out", a phenomenon due to irregularities in the tape which increase the reading-head-to-tape-surface distance and can cause bits to be missed. To guard against undetected drop-out, the device which generates the word-on the tape always generates an even (even parity) or odd (odd parity) number of bits, the parity bit being inserted or not inserted depending on the number of bits in the rest of the word when the tape is written. Then, as the word is fed into Deuce, the number of bits is counted. If a bit has been missed, in reading, the parity will not be odd (or even), as it should be, and a failure warning will be triggered. The clock track is used to tell the machine when to read a word from the reading head of the Ampex. It is used to generate a TIL signal When TIL is off the machine will not read and when TIL comes on the TIL discrimination enables a segment of read programme to be entered. If nothing is read before TIL goes off again then a missed word signal is triggered. Both these warnings illuminate special lights on the Deuce control panel.

Comparison of reading and punching speeds

The paper tape is read at 850 x 8 bit characters per second and punched at only 25 x 8 bit characters per second. Magnetic tape on the other hand is read at 900 x 16 bit characters per second. These compare with card reading speed 200 cards per minute which is 40 x 64 bit characters per second and a card punching speed of 100 cards per minute or 20 x 64 bit characters per second. Thus using cards 2,560 bits are read per second, paper tape 6,800 bits per second, and magnetic tape 14,400 bits per second. Punching gives cards 1,280 bits per second and paper tape 200 bits per second.

Off-line equipment

So called “off-line equipment” is equipment used in association with, but independent of, Deuce. Most of consists of Hollerith card handling machines, although there is a paper-tape-controlled typewriter—a “Flex-o-writer.”

The Hollerith machines comprise: a tabulator, a reproducer, a sorter, and a collater. The tabulator will read cards at about 100 cards per minute and print the information on them up to 80 characters per line, one line per card. The reproducer will produce copies of cards at about 100 cards per minute. The sorter will sort on any one of the 80 card columns and feed the cards into one of 13 positions, one for each of the twelve rows, and one for rejects. The sorter operates at 600 cards per minute. The collater is probably the most versatile machine and to mention a few uses: it will merge together two packs of cards of like sequence, select unmatched cards from either of two groups, simultaneously merging the two packs if required; select into one pocket cards of a particular denomination, rejecting the remainder; and finally sequence check a pack of cards. The speed is about 200 cards per minute from each of the two feeds.

(Signed) David R. Hill

Flight Systems Engineer

APPENDIX I

SOURCE			DESTINATION		
AVAILABLE	NO.	DESCRIPTION	DESCRIPTION	NO.	AVAILABLE
EVERY	0	ID OR READ	TS COUNT	0	EVERY
1 IN 32	1			1	1 IN 32
1 IN 32	2	DELAY LINES 1, 2, 3, 4, 5, 6 & 7 WITH CORRESPONDING NEXT INSTRUCTION SOURCES		2	1 IN 32
1 IN 32	3			3	1 IN 32
1 IN 32	4			4	1 IN 32
1 IN 32	5			5	1 IN 32
1 IN 32	6			6	1 IN 32
1 IN 32	7			7	1 IN 32
1 IN 32	8	DL8 / NIS0	DL8	8	1 IN 32
1 IN 32	9	DELAY LINES 9, 10, 11 & 12 WITHOUT NEXT INSTRUCTION SOURCES		9	1 IN 32
1 IN 32	10			10	1 IN 32
1 IN 32	11			11	1 IN 32
1 IN 32	12			12	1 IN 32
EVERY	13	TEMPORARY STORES (TS) (SEE NOTES)		13	EVERY
EVERY	14			14	EVERY
EVERY	15			15	EVERY
EVERY	16	(a)	(b)	16	EVERY
1 IN 4	17	QUADRUPLE STORES		17	1 IN 4
1 IN 4	18			18	1 IN 4
1 IN 2	19	DOUBLE STORES (DS)		19	1 IN 2
1 IN 2	20			20	1 IN 2
1 IN 2	21			21	1 IN 2
1 IN 2	22	DS21 ÷ 2 (c)	DS21 + (c)	22	1 IN 2
EVERY	23	TS14 ÷ 2	DS21 - (c)	23	EVERY
EVERY	24	TS14 x 2	Triggers	24	EVERY
EVERY	25	TS14 & TS15	TS13 +	25	EVERY
EVERY	26	TS14 ≢ TS15	TS13 -	26	EVERY
EVERY	27	P1	Sign discrim. off (+), on (-)	27	EVERY
EVERY	28	P17	≢ Discrim. on (non-≢) off (≢)	28	EVERY
EVERY	29	P32	Output staticisors /PUNCH	29	EVERY
EVERY	30	Zeroes	Drum transfer (d) (e)	30	EVERY
EVERY	31	Ones	Head block shift (f) (g)	31	EVERY

Note: & gives a 1 for both '1', '0' otherwise; ≢ gives '1' if different, '0' otherwise

TRIGGERS

0—24	Puts TCB 'off' STIM MULT odd m.c.	21_3 x 16 --> $21_{2,3}$
1—24	PUTS TCB 'on' STIM DIV. odd m.c.	21_3 ÷ 12 --> 21_2

2—24	TIL DISCRIM
3—24	STIM TCA
4—24	CLEAR TCB
5—24	STIM TCB
6—24	CLEAR ALARM
7—24	STIM ALARM
8—24	CLEAR O-P STATICISORS
9—24	CLEAR READ/PUNCH
10—24	STIM PUNCH
11—24	NOT USED
12—24	STIM READ

(a) With TCA 'on' TS16 gives contents of DL10 delayed by 1 m.c.
(b) Clears TCA
(c) TCB 'on' gives DS21 treated as two tanks
(d) Write DL11 on track t (use t—30 l)
(e) Read from track t (use t—30 s)
(f) Shift write heads to p (use p—31 l)
(g) Shift read heads to p (use p—31 s)

C = 0 is short (s)
C = 1 is long (l)
C = 2 is double (d)

APPENDIX II

Some programming notes for Deuce

1. Automatic instruction modification

Deuce is really a "first generation" computor, yet it is still used and found efficient for many purposes. The main reason for its extended life is the addition of automatic instruction modification—AIM, as it is called, which allows more flexible and more efficient programmes to be written. Its basis is in the action of 17—0 or 18—0 instructions which have equal wait and timing numbers. Such instructions cause the relevant word to be transferred from QS17 or QS18 to control, as for any S—0 type instruction, but with certain digits added to it according to the value of the first four digits, regarded as a four bit binary number, of the 17—0 instruction. The effect of various values of these digits on the digits added to the word transferred is as follows

QS17 (For QS18 the signs are reversed.)

ADDS	P.5	P.10	P.17	P.18	Effect of making the transfer long (17—0 1)
Value of 4 digits (X.P1)	0	1	2	3	None
	4	5	6	7	Adds P.22 also *
	8	9	10	11	Subtracts instead of adding modifying digit.
	12	13	14	15	Subtracts instead of * adding modifying digit and also subtracts from P.22

* In these cases there will be no spill into the P.22 position. from the P.21 position.

Clearly in using this facility the P1 digit and the three NIS digits of the instruction are used to specify the modification to the word to be transferred. As any S—0 instruction with equal wait and timing numbers overrides the NIS selection anyway this does not matter.

Suppose we have in 17_0:

```
NIS  S—D   Ch.  W    J    T   G
 2   9—14        31  (14)  T
```

and the instruction 6.P1 17_0—0 W T is obeyed. Then the next instruction obeyed is the word in 17_0 modified by the addition of a P.17 and a P.22 (remembering there is no carry from the P.21 position) to give: 2 9—14. 0 (15) T

Briefly then;

(a) Execution of:

```
NIS  17—0   Ch.  W   (J)  T
```

causes the m.c. of QS17 defined by W to be modified.

(b) (i) If W = T or C = 1 (long), then the m.c. of QS17 defined by T will enter control and be obeyed.

(ii) If W ≠ T and C = O the NIS and T define the next instruction source and set up the m. c. in the normal way.

The rules are similar for QS18 though all signs are reversd.

2. Rules- of- coding

(a) Determinationof T (Timing number)

An instruction stored in A_{m1} leading to an instruction stored in B_{m2} requires a NIS of B and a T given by:

$$T = m2 - m1 - 2$$

if T comes out negative add 32.

(b) Determination of ‘W’ (Wait number)

First decide on the m.c. of transfer then, if m.c. of transfer is ‘M’ and the instruction causing the transfer enters control in m.c. ‘m’ then W is given by:

$$W = M - m - 2$$

If there is latitude in the choice of ‘M’, then clearly it should be chosen to give the smallest value of ‘W’.

(c) Long transfer

If it is required to make a transfer for ‘n’ m.c.'s then the characteristic should be long and:

$$W = T - n + 1$$

If the transfer must be for specific minor cycles then T is fixed to give the next instruction storage location in the same m.c. as the last m.c. of the desired transfer, and thus J is also fixed to start the transfer in the desired m.c. For example:

$$10_5—25 \text{ (6 m.c.)}$$

transfers m.c.'s 5, 6, 7, 8, 9 and 10, and the next instruction must be stored in m.c. 10. If the instruction entered control in m.c. ‘2’ from DL4 the NIS was ‘4’ then the full instruction would be

(Storage)

	NIS	S—P	Ch.	W	T
(4)	4	10-25	1	1	6

(d) Discrimination coding,

The pair of instructions following a discrimination must be in adjacent m.c.'s of the same delay line.

The zero and positive legs lead to the first of the pair and the negative or non-zero legs go to the second of the pair. These are termed the early and late paths. The discriminating instruction is coded with respect to the first of the pair, control effects the delay to the second, if appropriate.

3. General

All transfers to D22 & D23, and from S22, for more than 1 m.c., must occur in the order even m.c. first odd m.c. second. Any single length words sent to D22 or D23 will be converted to correctly signed double length words if TCB is ‘off’. Any transfer to D22 or D23 with TCB ‘off’ will be extended for a further m.c. The word transferred will be all zeros if the last digit of the first word was zero, or all ones if it was one.

If (m + W + 2) or (M + T + 2) exceeds 32, subtract 32 to get the effective m.c. of operation (though one whole M.C. later)

If T < W then T operates a whole M.C. later.

4. TIP (Tabular Interpretive Programme)—Purpose

The purpose of the Tabular Interpretive Programme is to provide a quick and easy method of programming computers without a specialised knowledge of the computer itself. This is done by presenting instructions to the computer in a notation which is simply a formalised version of that already in every day use with desk calculating machines. The TIP transcribes each instruction into a corresponding piece of computer programme (i.e. a segment of specialised instructions in the computer's own order code) and carries out the operations specified.

TIP is an efficient technique to employ on calculations of tabular form but does not make efficient use of the computer for non-tabular work. Before using TIP it is necessary to ensure that the problem is either one of tabular arithmetic or that it may he conveniently expressed in tabular form (for example, by solving several cases of a single line problem in parallel).

APPENDIX III

TIP Mk. II Notes

Introduction

This scheme will be found most useful for any calculation which normally would be carried out using a sheet of paper ruled into rows and columns where the numbers in any column are mathematical functions of the corresponding rows of one or more columns. Any engineer capable of presenting a calculation to desk machine operations will find this scheme very similar.

It lies computor-wise, somewhere between alpha code (used for calculations on a single variable) and GIP (used for bulk calculation in parallel, as typified by matrix operations).

Use

Data storage: 128 columns (0-127) each having 30 rows (0-29). There are also 128 spaces for constants (N.0, N.1, ... N.127). N.0. to N.31 are 32 preset constants, as follows:

N.0	0	N.16	1.23
N.1	1	N.17	65.7633
N.2	2	N.18	3.5
N.3	Dash	N.19	96.014
N.4	$\pi = 3.141593$	N.20	14.656
N.5	$g = 32.174$ (ft/sec^2)	N.21	360
N.6	-1	N.22	-26,200
N.7	M = 0.33	N.23	0.1
N.8	$180 \div \pi = 57.29578$	N.24	550
N.9	2240	N.25	0.491
N.10	10	N.26	0.6849
N.11	g = 386.088 (in./sec2)	N.27	7.8125×10^{-7}
N.12	12	N.28	-3.8281×10^{-5}
N.13	288.16	N.29	2.0711×10^{-3}
N.14	14.696	N.30	-2.9890×10^{-2}
N.15	273.16	N.31	100

These may be overwritten by other constants if required.

Code words are in four parts

(a/Na) . (b/Nb) . (c/Nc) . r

Where ‘a’/‘b’/‘c’ represents column numbers, ‘Na’/‘Nb’/‘Nc’ represent constant numbers and ‘r'' is the code number of the operation to be performed.

If a, b, or c has an asterisk by it, it will be increased by one after being obeyed. For example, if a, b*, c, r were obeyed three times in a loop

1st time:	a	b*	c	r
2nd time:	a	(b+1)*	c	r
3rd time:	a	(b+2)*	c	r
being left as:	a	(b+3)*	c	r

Up to 512 codewords can be used (0-511).

Calculations are generally carried out with a constant number of rows in all columns, this number being provided for the machine at the start of a calculation.

A “dash” (one of the preset constants) is a number which is unchanged by any operation on it. If punched out it will appear as a negative number, very small in relation to the other numbers in the column, and usually indicates a failure. For example, if any attempt is made to divide a number by zero, or to find the square root of a negative number, the machine will stop and, on being told to proceed, will insert a dash in the place where the answer should have gone.

Punching: In binary form a, b, and c are 9-digit numbers in columns 1, 9, 10-18, and19-27; and ‘r’ is a 5-digit number in columns 28-32. When a, b and c are columns or constants, the first seven digits give the number (0-127), the eighth digit is an asterisk, and the ninth digit indicates reference to a constant. Cards should be presented in triads, blank cards being used if necessary.

The TIP pack is followed by a parameter card giving: (the number of triads of codewords to be read × P.1) on the Y row, then the codeword triads, then a parameter card giving (the standard number of rows for the calculation × P.1) on the Y row.

There is a conversion programme which will take data for 3-D graphs which is punched as five digit numbers having 4 decimal places—4 numbers to a card—with signs in columns 1, 9, 17, and 25 and digits in columns 2-6, 10-14, 18-22, 26-30. The first card contains x_0, y_0, Δx and Δy followed by $f(x_0, y_0)$, $f(x_0, y_1)$ ...$f(x_0, y_n)$, $f(x1, y_0)$, $f(x_1, y_1)$, $f(x_1, y_2)$ ... etc. and the data is preceded by a parameter card giving ((the number of values of x) × P.17) on the Y row and ((the number of values of y) × P.17) on the X row

The programme will calculate maximum x and maximum y and punch these with the rest of the data in binary (29 d.p.). Note that x, y, and f(x,y) must all be less than 4 than in absolute value. All other reading and punching is in standard floating decimal. All other read routines have a decimal punching check which stops on 12—24 x. Given a Single Shot (S.S.) the card can be re-read.

Testing

1. P32 on ID will stop operation on each codeword displaying the codeword on the Output Staticisor (OS).
2. n P17 on the Input Dynamicisor (ID) will stop on codeword ‘n’
3. p P1 + P31 on ID will punch ‘p’ rows of column ‘c’ after obeying each codeword.
4. p P.1 + n P.17 + P.31 on I.D. will punch 'p' rows of column 'c' after obeying codeword ‘n’
5 When the programme is stopped on a codeword a new codeword may be inserted from the ID by putting TIL on & giving a SS, turning TIL ‘off’ and providing another SS. After obeying the inserted instruction control will return to the one this replaced.

Failure indications

7	31—29 x		Division: divisor zero
5	31—16 x		Square root: negative
7	31—13 x		Log a: a negative or zero
1	31—14 x		e^a : $\|a\| \geq 2^7$
6	31—16 x		Interpolation/extrapolation
5	31—14 x		$\sin^{-1}a$, $\cos^{-1}a$: a > 1.0
1	31—15 x		3-D interpolation/extrapolation
0	31—15 x		3-D interpolation/extrapolation
	13—28		Sum check failure

In the event of complete failure using 12—1 (32 m.c.) and entering 1_0 (8—31) will normally lead the programme to the codeword following the one in which the failure occurred. If however the failure was in “read constants”, or 3-D interpolation, it may be necessary to replace DLs 2—4 by tracks 8/13 - 15.

TIP Mk 2 Order Code.

In all instructions a, b & c may be replaced by Na, Nb, or Nc, if meaningful. Also, a = b = c is allowed, if meaningful.

CODEWORD				NOTE	MEANING
a	b	c	0		Multiplication a x b --> c) c takes length
a	b	c	1		Division a/b --> c) of shorter
a	b	c	2		Addition a + b --> c) column
0	0	c	4		Read standard length column to c
a	0	c	4	(x)	Read 'a' elements to column c
a	0	Nc	4		Read 'a' constants to Nc to (Nc + a + 1)
a	0	0	5		Punch column a
a	b	0	5		Punch first 'b' rows of column a
Na	0	0	5		Punch Na
a	0	c	6		Square root of a --> c
a	Nb	c	7		Sum series of arguments in column a, order of series in Nb, coefficients in N(b + 1) to N(b + Nb + 1)
a	0	c	8		Log (column a) --> column c
a	0	c	9		Exponential (column a) -- column c
a	0	c	10	(x)	Shift column 'a' up one row & put in 'c' (reduces length)
a	Nb	c	10	(x)	Shift column 'a' down one row putting Nb in row 0
a	0	c	11		Sin 'a' --> 'c'
a	0	c	12		Cos 'a' --> 'c'
a	0	c	23		Sin^{-1} 'a' --> 'c'
a	0	c	24		Cos^{-1} 'a' --> 'c'
a	0	c	13		Move column 'a' to 'c' (copies)
Na	b	c	13	(i)	Move Na to row 'b' of column 'c'
a	b	Nc	13		Move row 'b' of column 'a' to Nc
a	0	c	14		|column 'a'| --> column 'c'
a	0	c	15		Put a_0, $(a_0 + a_1)$, $(a_0 + a_1 + a_2)$, $(a_0 + a_1 \ldots + a_{n-1})$ --> 'c'
0	b	c	16	(ii)	Jump to 'c' till instructions from 'c' have been obeyed 'b' times, then proceed normally
0	Nb	c	16	(ii)	As above, but for Nb repetitions
a	Nb	c	17		Jump to 'c' and repeat loop until columns 'a' and (a + 1) are within % tolerance, then proceed normally
0	0	0	18		Reset asterisked codes in loop
0	0	c	19	(a)	Jump to codeword 'c'
0	0	c	20	(iv)	Jump to subroutines starting at code 'c'
0	0	0	21	(iv)	Jump to codeword following last entered subroutine
a	b	c	22	(v)	Linear interpolation
a	b	c	25	(b) (vi)	Branch
a	b	c	26	(vii)	Mix
a	b	c	27	(viii)	3-D linear interpolation
a	0	0	28		Punch column 'a' in binary
0	0	c	29		Read into column 'c' in binary
a	0	c	29	(x)	Read triads into columns 'c' to (c + a - 1)
0	0	0	30		NOT USED
0	0	0	31	(ix)	End of program
a	0	0	23	(c)	Will punch out columns 'a', (a + 1), (a + 2), and (a + 3) (corresponding rows on one card)

(a) Don's brick: a, b, c, 25 - move appropriate row of column 'a' to 'c' if 'b' is positive; move appropriate row of column (a + 1) to 'c' if 'b' is negative.

(b) Don's mod: a, b, c, 19 - replace codeword 'a' by codeword 'b' and jump to 'c' if (b + 0) meaning as before -

(c) Don's brick. [more notes on the next page]

Notes

(i) If (b + 1) > (the number of rows in column a) then Na is transferred to rows to b + 1 inclusive and the length of the column changed to (b + 1)

(ii) Up to 2nd order loops are permissible.

(iii) If this order is placed immediately after an order where r = 16 it will work backwards from the r = 16 order, examine each codeword in turn, resetting all asterisked codes to their original values. When it reaches number 'c'—specified by the r = 16 order—it will consider the operation finished. N.B. It goes through in reverse order to the way the codes are written, not that in which they are obeyed. Thus if a second order loop is being reset the inner loop will not be reset.

(iv) Up to second order subroutines of codewords are allowed.

(v) Column 'b' contains x_0 to x_{n-1}, (not necessarily equal intervals). Column (b + 1) contains f (x_0) to f (x_{n-1}). Column 'a' contains given arguments x. Linearly interpolated values of f(x) are produced in 'c'. Note that columns 'b' and (b + 1) will not, in general, be of standard length.

(vi) This will examine each row of column 'b'. If the row is '0' the corresponding element of 'a' will be put in column 'c' and a dash in (c + 1). Otherwise the element of 'a' will be put in column (c + 1) and a dash in column 'c'.

(vii) If a row of 'a' contains a number it will be transferred to 'c'; if it contains a dash the corresponding element of column 'b' will be transferred to 'c'. Length of 'c' is maximum of 'a' and 'b' if these are unequal.

(viii) Given x-coordinates in column 'a', y-coordinates in column (a + 1) this will store interpolated values of f(x, y) in column 'c' where f(x, y) is given in column 'b' onwards (see .this Appendix, page 3, for the storage of a 3-D graph).

(ix) This clears columns 0 - 127 and constants 32 - 127 resets constants 0 - 31 and calls for more programme.

(x) These operations may produce a column of non-standard length.

APPENDIX IV

S.C.5. FLIGHT TEST DATA EXTRACTION AND DATA HANDLING

0. Introduction

Data relevant to flight testing the “Belfast” will be displayed, as recorded during flight, in five ways:

(i) On the pilot’s engineer’s and other instrument panels.

(ii) On an auto-observer panel.

(iii) On a C.E.C. trace recorder

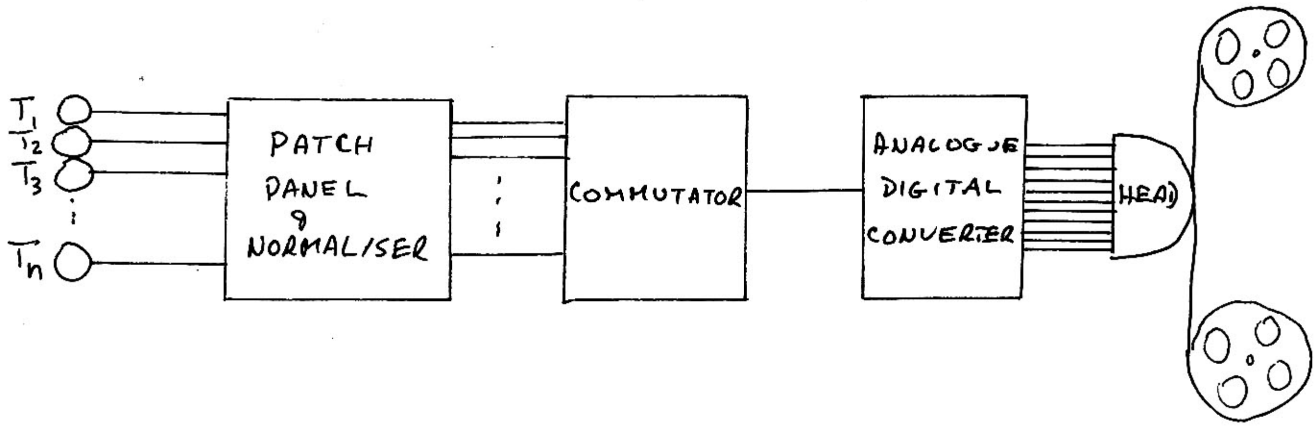
(iv) As a frequency modulated frequency multiplexed signal on magnetic tape.

(v) As a series of 16-digit binary numbers on magnetic tape.

This appendix is intended to go into the details of the last method from the point of view of the recording and the subsequent extraction and use of the data points. Consideration will also be given to the practical difficulties likely to be encountered during the operation of such a system, as well as to coordination between the departments involved.

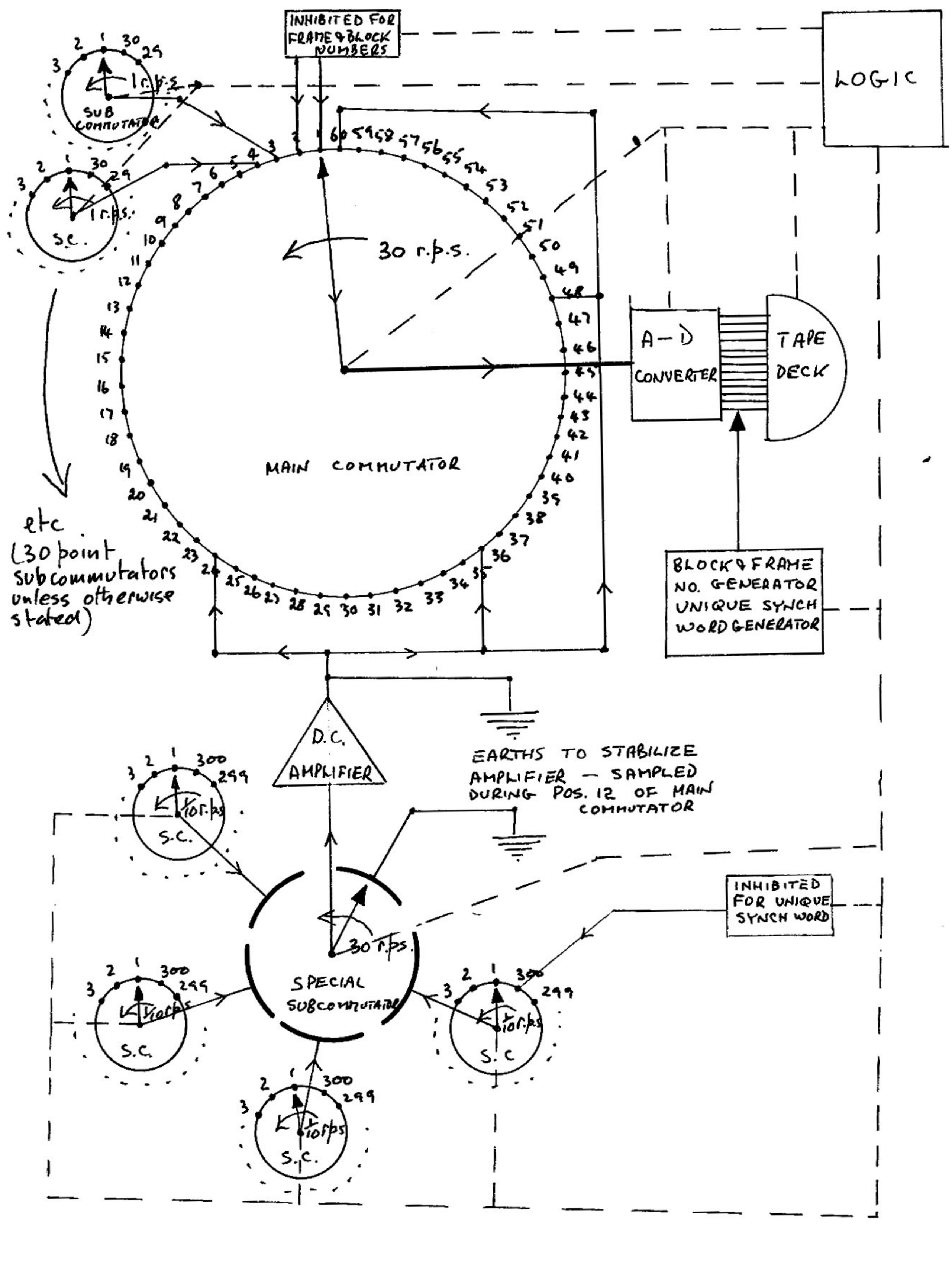
1. Recording

Variations in each quantity to be measured will produce variations in the output of the transducer associated with that quantity. The transducer outputs will be voltages. Thus, for each quantity to be measured, a varying voltage representing the quantity will be produced—an electrical analogue. All the outputs are taken to a patch panel and commutator system where they can be sampled at the rate of 1,800 samples a second. As each quantity is sampled it is passed through an analogue-to-digital converter, which has 12 outputs, to 12 writing heads in the head stack of an airborne 16-channel digital tape recorder.



The patch panel allows each transducer to be connected to one or more commutator inputs allowing, for a given quantity, sampling rates of up to 1,800 samples per second corresponding to an information frequency of 300 CPS, sampled 6 times per cycle. The most economical maximum sampling rate for the digital system is in fact 5 cps, quantities having a higher information frequency being recorded on the F.M./F.M. system. Clearly there would be room for only one quantity on the digital system at 1,800 samples/second.

In practice the design of the commutating system differs from the simple system suggested by theory and may be represented in the following manner.



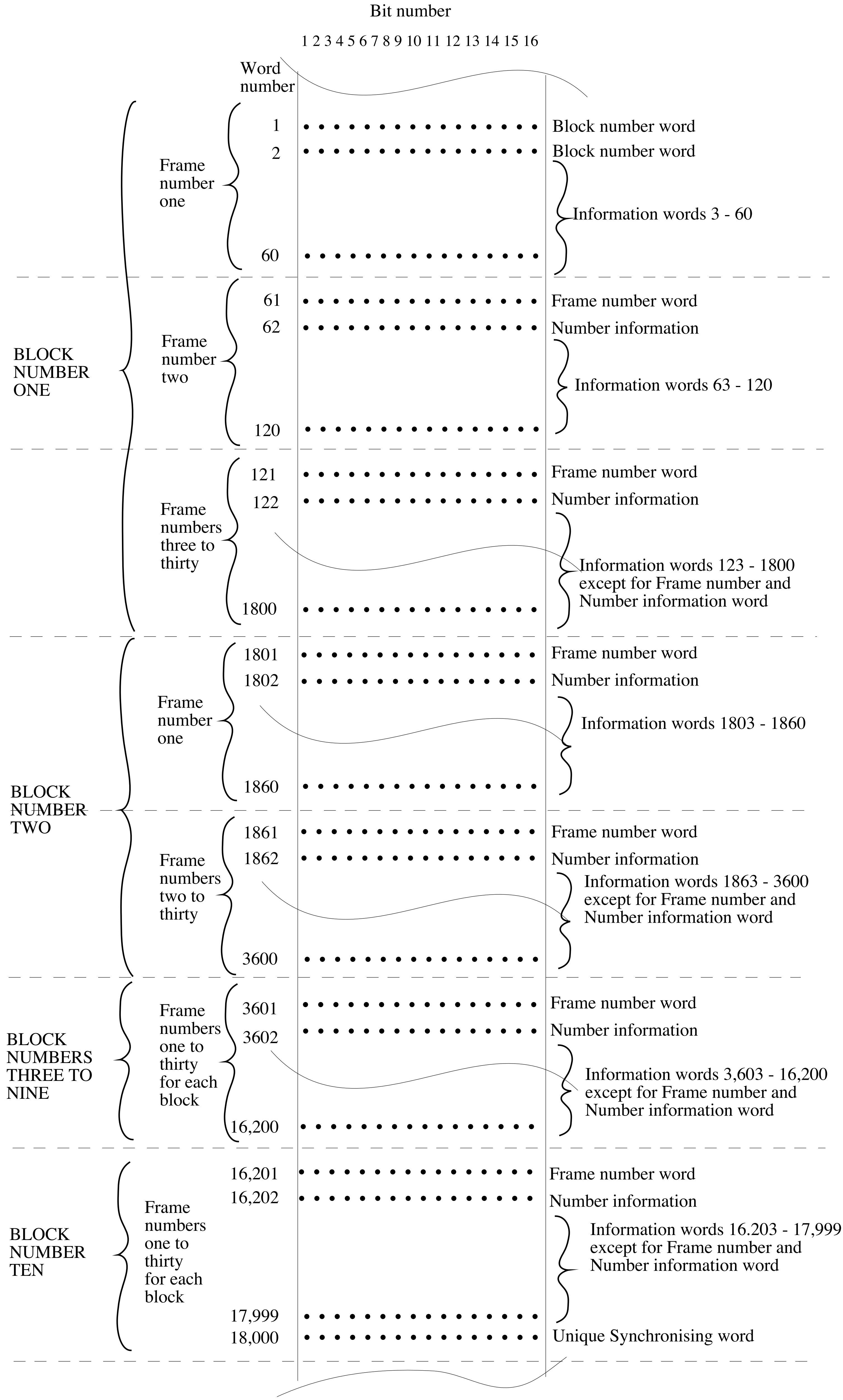
If the operation of this system is followed it can be seen that in each 1/30 th of a second there are two ('1' and '2') inputs reserved for frame or block number information (1 frame ≡ 1 revolution of the main commutator) four inputs (24, 36, 48 and 60) reserved for slowly, less frequently, sampled data, and the remaining 54 inputs are obtained from 30-position sub-commutators. The system is shown at the start of a cycle which takes exactly one second to complete.

The analogue-to-digital converter changes each electrical signal into digital form. The analogue signal may vary between 0% and 100% of full scale; the digital equivalent is three decimal digits giving a range from 00.0% to 99.9% of full scale. Each of the decimal digits is expressed in binary form—least significant bit on the left.

0 ≡ 0000	5 ≡ 1010
1 ≡ 1000	6 ≡ 0110
2 ≡ 0000	7 ≡ 1110
3 ≡ 1100	8 ≡ 0001
4 ≡ 0010	9 ≡ 1001

Thus 12 binary digits will express the required decimal digits, in what is termed "binary coded decimal", and are also suitable for putting on magnetic tape, strong magnetisation representing "ones" and no magnetisation "zeroes". The twelve digits are written across the tape together with four more digits (or "bits") which are: a sign bit, a parity bit, a clock bit and a marker bit. Thus, there are 16 bits across the tape and these constitute a word. (See page 29 of the report "Construction and Use of the Deuce Computer")

In each second 1,800 words appear on the tape. Most are of the above form but there are a few special words. Thus 18,000 words are put on tape during one commutator cycle or scan and the layout is as follows (compare with the commutator system layout shown above).



Bit number
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Word number
BLOCK NUMBER ONE
Frame number one
1 Block number word
2 Block number word
Information words 3 - 60
60
Frame number two
61 Frame number word
62 Number information
Information words 63 - 120
120
Frame numbers three to thirty
121 Frame number word
122 Number information
Information words 123 - 1800 except for Frame number and Number information word
1800
BLOCK NUMBER TWO
Frame number one
1801 Frame number word
1802 Number information
Information words 1803 - 1860
1860
Frame numbers two to thirty
1861 Frame number word
1862 Number information
Information words 1863 - 3600 except for Frame number and Number information word
3600
BLOCK NUMBERS THREE TO NINE
Frame numbers one to thirty for each block
3601 Frame number word
3602 Number information
Information words 3,603 - 16,200 except for Frame number and Number information word
16,200
BLOCK NUMBER TEN
Frame numbers one to thirty for each block
16,201 Frame number word
16,202 Number information
Information words 16.203 - 17,999 except for Frame number and Number information word
17,999
18,000 Unique Synchronising word

2. Extraction of Data

There are two methods of extracting data from the magnetic tape after a flight, a “quick look facility” which can print out the inputs to twelve commutator positions in percentage of full scale in one hour (for a five hour flight) and the normal facility which involves feeding information directly from the tape into the Deuce computer.

The quick look facility needs little elaboration. It can be used to obtain advance information on certain quantities to allow flight testing to continue, without waiting for more sophisticated results from the computer, and—in conjunction with the master observer’s log—for specifying areas of interest on the tape for full extraction.

The normal method of data extraction, via the Deuce, requires a little explanation to enable the reader to get some idea of the limitations, advantages and techniques of the process. For clarity the process of extracting and using information from a hypothetical tape will be considered and the other aspects pegged to this.

The magnetic tape can be played back straight into Deuce by the magnetic tape input of the machine. It presents, effectively, a continuous input of sixteen digit binary numbers up to approximately thirty million in a five hour tape. Only a very small percentage of these will be required for computation and output, approximately 0.08% as an average for aircraft handling and flight engineering trials, giving—for a 5 hour flight—a mere 26,000 data points to be extracted. The problem of selecting the right 26,000 points from 30,000,000 is entirely one of computer programming and using a fixed tape layout. However, with an 8,192 word store the job cannot be tackled in one bite. The method adopted is as follows.

Using the time/test history of the tape produced from the master observer’s log and—if necessary—information from the quick look facility, the tape is broken into overlapping sections. Each section contains as many data points as can conveniently be stored when allowance has been made for the storage of the extraction programme. The extraction programme has its own card input of constants of tape locations applicable to the test involved (based on the tape layout) so that when a section has been run through, it has as a store of required data points in binary unscaled form. These are punched out as 11-bit pure binary numbers (as opposed to binary coded decimal), expressing values as a percentage of full scale, 54 to a card with an element reference and a series row sums to allow subsequent programmes to do their own validity check on read-in. There will still be some redundancy at this stage, to make programming eaier, and it is referred to as Stage I.

Stage II of the data extraction process is a little more complex. The Deuce is fed with all the calibration curves of the transducers, each being identifiable by a reference. Next an "extract" programme is fed in and then the pack of data cards from Stage I, appropriäte to a particular test, with three special types of cards on top. First calibration reference cards which,in effect, tell the programme where to find the calibrations. Next are the "extract" cards which tell the programme which data to extract from the mass of data in the data cards. Then, on top of the pack of data cards, is a decimal identification card, which is simply a label for the pack of data which enables the computer to check that it has the correct pack of data. In the computer the required quantities are taken, the appropriate calibrations looked up and applied, and some simple calculations performed. Output is again punched cards, but consists of decimally punched actual values (in pounds, feet-per-second, etc.), and some computed results (mass flows, etc.).

For the determination of actual layout of the cards applicable to a given test, a number of factors must be borne in mind such as operational convenience, future use, and the data to be presented, and so on. To consider this further the future uses of cards must be explained.

2.1. Tabulation

Probably the commonest treatment of punched cards from Stage II will be to put them through the tabulator. The tabulator is a standard Hollerith machine which, as a card passes through, examines every hole in the card and then—according to connections on a plugboard (equivalent to a simple programme)—sets print wheels and causes a line of up to 80 characters to be printed out to correspond with the meanings of the holes. The speed of working is independent of the number of characters to be printed and is about 100 cards per minute. Clearly a lot of time can be spent setting up the plugboard so that it is sensible to lay out cards for tabulating in such a manner that a standard plugboard can be used to ensure that the digits and signs are correctly printed and suitably spaced (a space counts as a character). Letters as well as numbers can be printed if the card is suitably punched and the plugboard suitably plugged. Signs may be over punched on digits—again if the plugboard is suitably plugged.

The finished product from the tabulator is a series of sheets of paper joined by perforations on which there are as many lines as there were cards fed in.

2.2. Autoplotting

If data on the cards are to be plotted automatically, on the Benson-Lehner autoplotter, the primary consideration on layout is what is to be plotted against what. Two items to be plotted against one another must appear on the same card. Thus, for example, if there are a greater number of quantities to be plotted against time than can be punched in one card, then time must be duplicated on the second card.

Cards are read into the autoplotter by means of a plugboard-controlled Hollerith reader. However, the plugboard is much simpler than that required for the tabulator and there is less need to conform to a standard layout of cards. There is a different limitation. The reader's output is split into four-digit-with-sign fields and there is, therefore, no point in giving more than four card columns for significant digits for each quantity. The sign may be overpunched on the digits if required. If—say—five digits were required for tabulation, and the cards were subsequently to be autoplotted, the least (or most) significant digit could be ignored, using the plugboard. In the case where the most significant digit was ignored, due account of the missing digit could be taken by suitable scaling of the graph axes and by the use of a false zero.

The speed of autoplotting is of the order of 60 points per minute and the accuracy is ±0.025 of an inch.

3. The Use of the Data Processing Facilities

It is proposed that a given tape shall not be played into Deuce more than once, unless absolutely necessary. As far as the flight departments are concerned the basic store of data is not the tape, but the binary unscaled data punched out after stage I. It is relatively easy to select a batch of these cards for further processing, whereas to extract more information from the tape is much more time consuming, and the computer will be very busy while flight-testing is being, carried out.

Clearly, then, the first point in the use of the data facility is to ensure that requests for data to be extracted from the tape should be the maximum required, not the minimum. It would seem that there should be separate requests for Stage II results to avoid unnecessary Stage II processing. As an alternative the data required for a given test could be contained in certain specified areas of data so that a request for a certain test type for Stage II would result in considerably more data being extracted at Stage I than was strictly necessary for the desired Stage II results. Several other tests would require the same areas of data and, if those results were subsequently required, the data would already be available on cards.

At any event it is essential that all possible requirements are foreseen and allowed for in the “off-the-shelf” Stage I or Stage II programmes ,since programming can take some time, and there will not the time to spare for writing, let alone testing, programmes, once the flight trials are under way.

3. Form of Test Specification

For each test schedule (De-icing, Fuel, etc.) a technical programme must be issued by the relevant flight department. This will summarize the test types on the schedule, give a brief write up on the technical aims of the tests, specify the nomenclature to be used, and then give a specification of requirements for data and computation for each test type. Each of the items of the latter type is a test specification and will take the following form:

Section I — 1. Data or areas of data to be extracted from the tape.

Section II — 1. Calibrated results required;

2. Presentation required for calibrated results or intended treatment of calibrated results.

Section III — (Computation required)

1. Data to input: (a) Tables; (b) Constants; (c) Variable data.

2. Computation—lay down the computation precisely and step by step (steps (a), (b), (c), etc.).

3. Form of output required (similar to II 2.)

The sampling rate for each data point required should be stated, and can be less than or equal to the aircraft sampling rate. If it is necessary to know very precisely the relative phase between two quantities, this must he stated, since all quantities within a block are labelled with the same block time. For time histories this does not matter as successive values of the same quantity should have the same time error, but if, for instance, a control angle is following an error signal then the phase between the two would be vitally important.

Almost any type of computation may be specified, and it is relatively easy for the computor to take the mean of several values of the variables involved and do a single computation on the means, or to decide what are steady state values and use these. In the latter case, the allowable. tolerance for steady state must be given (e.g. “steady state to be assumed when the. pressure remains within ± 0.2 psi of a value over five successive samples”). Any limits on the meaningful values of results should be stated, and the action to be taken on encountering these limits specified.

For all initial data, intermediate results, and final results, anticipated maximum and minimum values should he given.

After a flight, data is obtained through the date processing group which will coordinate requests, operate the quick look facility, and liase with the Analytical Department—among its other activities—passing data back to the relevant flight department. With any request for data relevant to a particular test type, any constants or other data required during computation, which is not available on the tape, should be given with the request.

The following are the times expected to be taken for the various stages (Stage III is computation which cannot be included in Stage II; Stage IV is brochure comparison and only affects the Performance Section). For a five-hour Airborne tape:

Stage I	6 Hrs.
Stage II	5 Hrs
Stage III	7 Hrs.
Stage IV	12 Hrs. (but depends on number of points)

There should he some formalization and standardisation of data requests to facilitate liaison and record keeping. As the Data Processing Group are mainly those concerned, it should be their responsibility to organise this.

David R. Hill

<u>Flight Systems Engineer</u>