Instruction set of the English Electric DEUCE.

Contents.

1. Instruction format and central registers.

The word length is 32 bits, using the two's complement fixed-point representation of variables. Double-length facilities are provided. When used as an instruction, the format is similar to that of the NPL Pilot ACE (see section N1X2), with a slight re-ordering of the *unassigned* bit-positions as follows:

1	*3*	*5 bits*	*5 bits*	*2*	*5 bits*	*4*	*5 bits*	*1*	*1*
U	***NIS***	***Source***	***Destin***	***Ch***	***Wait***	***U***	***Timing***	***U***	***Go***
0	1 3	4 8	9 13		16 20	21 24	25 29	30	31

Least sig *most sig*

U: unassigned
N NIS: next Instruction Source: indicates long delay line D1 -> D8
S Source: number of selected source (short or long delay line)
D Destination: number of selected destination (short or long delay line)
C Characteristic: gives length of transfer (0, 1 or 2 -> single, long or double) & drum read/write mode
W Wait number: gives first minor cycle* of transfer
T Timing number: gives minor cycle* of next instruction, & sometimes also last minor cycle* of transfer
G Go digit: If G = 0, wait for handkey to be pressed; if G = 1, full speed ahead.

* a minor cycle is equivalent to a word, ie to 32 digit-periods. W and T are specified in terms of minor cycles. A digit-period is one microsecond.

The written short form of an instruction is: < N S D C W T G >. If the current instruction is at position m in a long delay line, then the next instruction is taken from position (m + 2 + T) in the delay line specified by N. If the current instruction is at position m in a long delay line, then the transfer of operands commences at time (m + 2 + W).

An operand-address number for S or D can be in the range 0 to 31. Briefly, values 1 to 12 signify long (32-word) delay lines; values 13 to 21 refer to somewhat shorter lines (see below) and values 22 to 32 are used for special purposes. Specifically:

13, 14, 15, 16:	one-word *temporary stores* referred to as TS13, TS14, TS15, TS16.
17, 18:	quad-word stores referred to as QS17 and QS18
19, 20, 21:	double-word stores referred to as DS19, DS20 and DS21.

Certain values of S and D are used to specify operations, or functions. For example, destinations D = 22 and D = 23 signify respectively the additive and subtractive inputs to DS21. Destinations D = 25 and D = 26 signify respectively the additive and subtractive inputs to TS13. Logical operations on TS14 and TS15 are given by S25 and S26. If D = 24, then various values of S in the range 0 – 11 set or reset various flags and triggers. One such trigger starts the multiplication process, which could then be overlapped with other operations. Another trigger is used to pulse the console's audio amplifier (hooter). If D = 27 or D = 28, the *source* operand is tested according to one of two conditions and, according to the result of the test, then the instruction from one minor cycle later than the normal instruction is obeyed. This therefore gives conditional branching – (called *discrimination* in the original documentation). Input/output is available via S = 0 and D = 29. The control of transfers to/from the magnetic drum is controlled by D = 30 and D = 31.

The DEUCE system diagram in section N1X2 indicates the way in which the various *source* and *destination* numbers control the flow of information on DEUCE's internal data highways.

In conclusion, the meaning of different combinations of S and D values was quite complex. By choosing appropriate combinations, the equivalent of about thirty effectively distinct operations could be obtained. By carefully choosing values of the N, W and T fields of each instruction, the work done by each instruction could be maximized and the overall execution time of a program could be minimized. Minimising the execution time, a process that demanded some skill on the part of the user, gave rise to the term *optimum programming* or *minimum latency* programming.

The DEUCE terminology derives from the rather idiosyncratic scheme used at NPL for translating Alan Turing's original ideas into the Pilot ACE computer. This is not easy for a modern reader to understand. The next section therefore provides an informal translation between the original terminology and modern equivalents.

2. Transforming the DEUCE instruction set to modern terminology.

In order to explain the DEUCE instruction set, we will adopt the ideas presented in section N1X2 for the Pilot ACE architecture.

Table 3.1 (see below) uses the following modern concepts and abbreviations:

Mn = main memory line n, ie the n^{th} of twelve 32-word delay lines
Rn = central register n, ie the n^{th} of four one-word delay lines
DRn = double-length register n, ie the n^{th} of three two-word delay lines
QRn = the n^{th} of two quad-length delay lines
Fn = the n^{th} arithmetic or logical function – see below for more details
Cn = the n^{th} wired-in constant. (The values, 0, -1, +1, 2^{16} and 2^{31} are provided).

Original Source number	***Modern signal name***	***Original Destination number***	***Modern signal name***	***Original Next-instruction number***	***Modern signal name***
0	**SF0**	0	Instruction register	0	NM11
1	SM1	1	DM1	1	NM1
2	SM2	2	DM2	2	NM2
3	SM3	3	DM3	3	NM3
4	SM4	4	DM4	4	NM4
5	SM5	5	DM5	5	NM5
6	SM6	6	DM6	6	NM6
7	SM7	7	DM7	7	NM7
8	SM8	8	DM8		
9	SM9	9	DM9		
10	SM10	10	DM10		
11	SM11	11	DM11		
12	SM12	12	SM12		
13	SR1	13	DR1		
14	SR2	14	DR2		
15	SR3	15	DR3		
16	SR4	16	DR4		
17	SQR1	17	DQR1		
18	SQR2	18	DQR2		
19	SDR1	19	DDR1		
20	SDR2	20	DDR2		
21	SDR3	21	DDR3		
22	**SF1**	22	**DF6**		
23	**SF2**	23	**DF7**		
24	**SF3**	24	**Function triggers**		
25	**SF4**	25	**DF8**		
26	**SF5**	26	**DF9**		
27	SC1	27	**DF10**		
28	SC2	28	**DF11**		
29	SC3	29	**DF12**		
30	SC4	30	**DF13**		
31	SC5	31	**DF14**		

Table 3.1. An interpretation of DEUCE's *Source* and *Destination* numbers as equivalent signals using modern terminology.

The implied notation of Table 3.1 is that SF0 – SF7 and DF 8 – DF19 are similar to *op codes* or *function specifiers.* The actions of these signals is further explained in Table 3.2 – (see below).

Control signal name	*Approx. function*	*Description of action*
SF0	INPUT	Take a 32-bit row from the card reader as S. If the CDR is currently off, take the 32 bits set up on the console switches.
SF1	ARITH SHIFT RIGHT	Take DR3 / 2 (ie DR3 divided by 2) as S
SF2	ARITH SHIFT RIGHT	Take R2 / 2 (ie DR3 divided by 2) as S
SF3	ARITH SHIFT LEFT	Take (R2 x 2) as S
SF4	AND	Take (R2 & R3) as S
SF5	NEQ	Take (R2 NEQ R3) as S
DF6	ADD D	DR3 := DR3 + S
DF7	SUB D	DR3 := DR3 – S
DF8	ADD	R1 := R1 + S
DF9	SUB	R1 := R1 + S
DF10	JLT	Branch on sign of S (*see note 3*)
DF11	JNZ	Branch on whether S is zero (*see note 3*)
DF12	OUTPUT	Transfer a 32-digit row to a card (ie the card punch) If the card punch is currently off, the 32 digits will be displayed on the console lights.
DF13	DRUM TRANSFER	drum data transfer: read/write
DF14	DRUM TRACK	drum head move (ie select track)

Table 3.2. Description of the DEUCE arithmetic and logical functions controlled by SF0 to SF5 and DF6 – DF14 of Table 3.1.

3. Explanation of the Function Triggers.

With *Destination* = 24 various DEUCE actions and states may be activated, depending upon the number used as *Source* and the value taken by the *source-operand.* Table 3.3 explains the actions.

Source	*Bit-value sent*	*Action*
0	1	start the multiplication process (auto turn-off at end). *See Note 1*
1	1	start the division process (auto turn-off at end). *See Note 1.*
2	1	TIL (Twelfth Impulse Line) to discriminator. *See Note 2.*
3	1	turn on (stimulate) TCA. *See Note 3.*
4	0	turn off (clear) TCB *(set double-length arithmetic mode for DR3)*
5	1	turn on (stimulate) TCB *(set single-length arithmetic mode for DR3)*
6	0	turn off (clear) the alarum
7	1	turn on (stimulate) the alarum – (ie ‘hoot’ and switch on a warning light)
8	0	turn off (clear) OPS (output staticisor on console)
9	0	turn off (clear) both the card reader and the card punch
10	1	turn on (stimulate) card punch
11	1	turn on (stimulate) card reader

Table 3.3. Triggers available when *Destination* = 24.

Notes for table 3.3.

Note 1. To perform a multiplication, the multiplier is transferred to one half of DR3(DS21) and the multiplicand to R4 (TS16). Then start the process by issueing the instruction '0 – 24', which sends a trigger. Multiplication takes 65 minor cycles. During this time, other operations may be carried out in other parts of the computer.

Note 2. Signal TIL is valid for the time that the last row of a punched card is passing the reading or punching positions. The instruction '2 – 24' has the effect of sending the TIL signal to destination 28,which is function DF11, 'branch if zero'. This is called 'sending TIL to the Discriminator' and allows a program to check whether all rows on a card have been read or punched.

Note 3. R4 (TS16) has two modes of working, as controlled by the trigger-signal called TCA. When TCA is on, a special mode applies whereby the digits in M10 (DL10) delayed by one minor cycle become the source-operand. When R4 is next specified as a source, TCA is automatically switched off and the normal R4 mode applies.