[illegible]

National

ELLIOTT

803 FACTS

FACTS FOR ENGINEERS

The 803 is a small, medium-speed digital computer, flexible in operation and economical to run. The central processer, power unit, paper tape station and keyboard forming the minimal installation require only 400 sq. ft altogether, and power consumption is about 3½ kilowatts.

The design follows normal practice for single-address machines. One operand is contained in the accumulator and the other in the store location specified by the address given in the instruction. Either operand or the result of an operation is stored in this location after the function has been performed. The current instruction is held in the instruction register to which is connected a decoding system for selecting the function to be performed and the store location required. Facilities are provided for double-length working in multiplication and division.

The element which forms the basis of the complete logic design depends for its operation upon the rectangular hysteresis characteristic of ferrites, small toroidal cores of the material being used, carrying a number of windings. Similar cores, threaded on wires and arranged in 64×64 matrices, are used in the store. Core selection is by a simple coincident current technique. Reading is inherently destructive, so that data to be retained in the store must be rewritten. This is automatic.

Dimensions and Weights

Length, height and depth in inches, weight in pounds

Central processer	66 by 56 by 16	680
Extra working store	33 by 56 by 16	250
Power unit	33 by 56 by 26	380
Paper Tape Station	50 by 30 by 29	450
Film Controller	33 by 56 by 16	250
Film Handlers	27 by 56 by 32	650
Card Input	41 by 36 by 21	215
Keyboard table	60 by 30 by 30	215
Keyboard	28 by 14 by 22	66
Battery charger	30 by 22 by 19	200
Line printer	64 by 64 by 29	1000
Digital plotter	18 by 10 by 15	33
Plotter control box	22 by 8 by 16	40
High speed character printer & console	32 by 26 by 46	600

FACTS FOR PROGRAMMERS

The basic 803 uses five-track paper tape input and output and has a main storage capacity of 4096 words, extendable to 8192 words maximum. Punched card input and output, line printer, high speed character printer and digital plotter are available as optional extras and 35 mm magnetic film backing stores having at most over seven million characters' capacity can also be added. The automatic floating-point arithmetic unit is an optional extra.

Speeds

Tape input	500 char/sec.
Tape output	100 char/sec.
Direct output	10 char/sec.
Card input	300 cards/min.
Card output	100 cards/min.
High speed character printer	100 char/sec.
Line printer	300 lines/min.
Digital plotter	300 steps/sec.
Film transfer	4350 char/sec. in block just under 5 blocks/sec.

803 is a serial binary computer, fixed-point number representation being such that numbers are held in the range $-1 \leq x < +1$, with two's complement notation for negative numbers, and standard floating-point form is $x = a.2^b$ such that $-1 \leq a < -\frac{1}{2}$ or $a = 0$ or $\frac{1}{2} \leq a < 1$ and $-256 \leq b < 256$. The word length is 39 bits.

The keyboard carries eight control keys and a 39-bit word generator, as well as power switching controls. The word generator may be used as an input channel, as a source of instructions external to the store, as a set of program switches, or as a method of selectively stopping the 803. A loudspeaker, driven from one of the control signals, is also provided, to indicate to the practised ear the progress of a computation.

Two single-address instructions occupy one word. By means of a single B-line placed between the two instructions, the second instruction may be modified without loss of speed by adding to it the content of the location specified in the address portion of the first, wherever this location may be. As each instruction is obeyed, the count in the sequence contol register is increased by a half, unless the instruction causes a transfer of control. Selection of the succeeding instruction follows the completion of an operation immediately.

INSTRUCTION CODE

The contents of the accumulator and specified location are denoted by a and n. a' and n' indicate these contents after the function has been performed. AR is the auxiliary register. Operation times quoted with each instruction or group of instructions are given in microseconds and include all requisite store access times.

GROUP 0

Code	Function a'	Function n'	Time
00	a	n	576
01	−a	n	
02	n+1	n	
03	a & n	n	
04	a+n	n	
05	a−n	n	
06	0	n	
07	n−a	n	

GROUP 2

Code	Function a'	Function n'	Time
20	a	a	576
21	a	−a	
22	a	n+1	
23	a	a & n	
24	a	a+n	
25	a	a−n	
26	a	0	
27	a	n−a	

GROUP 1

Code	Function a'	Function n'	Time
10	n	a	576
11	−n	a	
12	n+1	a	
13	a & n	a	
14	a+n	a	
15	a−n	a	
16	0	a	
17	n−a	a	

GROUP 3

Code	Function a'	Function n'	Time
30	n	n	576
31	n	−n	
32	n	n+1	
33	n	a & n	
34	n	a+n	
35	n	a−n	
36	n	0	
37	n	n−a	

GROUP 4

Code		Function	Time
40	44	Transfer control unconditionally	288
41	45	Transfer control if a negative	
42	46	Transfer control if a zero	
43	47	Transfer control if overflow indicator is set and clear indicator	

(40 to 43 transfer to the first instruction of a pair and 44 to 47 transfer to the second instruction.)

Note in groups 0 to 4 the address is specified in the normal way. In Groups 5 to 7, the address part of an instruction not requiring store access is used to further specify the function, in which case the number is indicated by N.

GROUP 5

Code		*Function*	*Time*
50	N	Halve, double-length, N times	576+288N
51	N	Right shift a N times. Clear AR	576+288N
52	N	Multiply (double-length product)	12096−288y
53	N	Multiply (single-length product). Clear AR	12384−288y

(Multiplication time depends on the number of consecutive ones or zeros, y, at the most significant end of the multiplier.)

54	N	Double, double-length, N times	576+288N
55	N	Double a N times. Clear AR	576+288N
56	N	Divide (double-length dividend, single-length quotient). Clear AR	12096
57		Read AR to accumulator	576

GROUP 6

Code		*Function*	*Time*
60	N	a+n in floating-point mode	864
61	N	a−n in floating-point mode	864
62	N	n−a in floating-point mode	864
63	N	a×n in floating-point mode	4896
64	N	a÷n in floating-point mode	9792 (max.)
65	4096	Convert 39 bit integer a to standard floating-point form	576

(Functions 66 and 67 are not used)

Times printed in green are quoted in milliseconds and are approximate only

GROUP 7

Code		*Function*	*Time*
70	0	Read from word generator to accumulator	576
73	N	Write the address of this instruction	576
Tape In			
71	0	Read first tape reader to accumulator	576
71	2048	Read second tape reader to accumulator	
Tape/Print Out			
74	N	Punch N on first punch	576
74	2048+N	Punch N on second punch	
74	4096+N	Print N on teleprinter (PTS 2A only)	
(Reader, punch and teleprinter channels include busy line facilities)			
Card In			
76	512	Read card input control word; prepare to read card	864
77	N	Read card to store locations N to N+79	175
Card Out			
76	2561	Read card output control word; prepare to punch card	
77	N	Transfer to card punch from store locations N to N+79	12
(Actual punching takes 600 milliseconds per card)			
Film Store			
76	1027	Read address of last block read or written	576
76	1024 1032 1040 1048	Read handler control word to accumulator; prepare to read on handler 1, 2, 3 or 4	864
76	1025 1033 1041 1049	Read handler control word to accumulator; prepare to write on handler 1, 2, 3 or 4	864
76	1026 1034 1042 1050	Read handler control word to accumulator; prepare to search on handler 1, 2, 3 or 4	864
77	N	Read, write or search as prescribed by 76 instruction	204

Times printed in green are quoted in milliseconds and are approximate only

GROUP 7 (Continued)

Digital Plotter

This device uses channel 2 and does not have a control word associated with it

Code		*Function* (pen moves in direction)	*Time*
72	7169	E	3.3
	7170	W	
	7172	N	
	7176	S	
72	7173	NE	3.3
	7174	NW	
	7177	SE	
	7178	SW	
72	7184	Pen raise	100
	7200	Pen lower	

Line Printer

76	3073	Read ANelex control word to accumulator: prepare to print one line	604
77	N	Transfer to line printer from store locations N to N+120	18

The contents of location N control paper-feed as follows:

M	(0≤M≤30)	feed M+1 lines and print
31	or 32	print on same line
32+M(1≤M≤30)		find channel M and print
63		find top of form and print

Serial Line Printer

72 0896+N output character with telecode value N to the S.L.P. (0≤N≤31)

SUMMARY OF INSTRUCTIONS FOR AUTOCODE

In these examples

A, B, C and D	represent	Floating-point variables
I, J, K and L	represent	Integer variables
l, m and n	represent	Positive integer constants
p, q and r	represent	Any integer constants
x, y and z	represent	Floating-point constants

Any variable except the one before the = sign may be replaced by a constant.

Arithmetic

A=B	A=−B	I=J	I=−J
A=B+C	A=−B+C	I=J+K	I=−J+K
A=B-C	A=−B-C	I=J-K	I=−J-K
A=B*C	A=−B*C	I=J*K	I=−J*K
A=B/C	A=−B/C		

Function

A=SIN B	A=LOG B	A=FRAC B	—
A=COS B	A=EXP B	A=INT B	I=INT A
A=TAN B	A=SQRT B	A=STAND I	—
A=ARCTAN B		A=MOD B	I=MOD J

Jump

JUMP @K

JUMP IF A=B@K JUMP IF I=J@K

JUMP UNLESS A=B@K JUMP UNLESS I=J@K

(K may not have any form of suffix).

Any permitted arithmetical instruction or function instruction may replace A=B or I=J, and > (%) or < ($) may replace =

Other Controls

SUBR n	EXIT	STOP	WAIT

Vary and Cycle

VARY A=B: C: L	VARY I=J: K: L
CYCLE A=B: C: D	CYCLE I=J: K: L
CYCLE A=x, y, z, . . .	CYCLE I=p, q, r, . .
REPEAT A	REPEAT I

(B, C, D, J, K, and L may have simple suffices only).

Input

```
READ A        READ I            INPUT I
```

Output

```
PRINT A, n: m PRINT A, n        PRINT A, n/     PRINT A
PRINT I, n    PRINT I           OUTPUT I
(In OUTPUT I, I may have a numerical suffix only).
LINE          LINES I           SPACES I        TITLE
CHECK A                         CHECK I
```

Setting and Start

```
              SETS    (Integer variables).
              SETV    (Floating-point variables).
              SETF    (Functions)
              SETR n    (Maximum reference number).
              START m    (Starting reference number)
In SETF    (i)  TRIG covers SIN, COS and TAN.
          (ii)  MOD and STAND need not be mentioned.
         (iii)  FILM allows use of film instructions.
          (iv)  CARD and PAR allow use of card reader
                instructions
```

Film

```
FILM(I) SEARCHJ(K)
FILM(I) TO J(K) or FILM(I) TO A(K)
FILM(I) FROM J(K) or FILM(I) FROM (A)K
JUMP IF FILM(I) SEARCHING @ L
JUMP UNLESS FILM(I) SEARCHING @ L
FILM(I) BLOCK NUMBER TO J(K)
FILM(I) ALLOW WRITE
FILM(I) PREVENT WRITE
```

K is a simple suffix

L cannot have any form of suffix

Card

```
J=PAR l, m, n
```

l, m, n may be replaced by integer variables having numerical suffices only.

```
A=CARD 1, A, J, K or I  =CARD 1, I, J, K
A=CARD 2, A, J, K or I  =CARD 2, I, J, K
```

K may be replaced by an integer constant.

SUMMARY OF 803 ALGOL STANDARD PROCEDURES

In the following parameters

X	represents	a real expression
Z	represents	a real variable
I & J	represents	*integer* expressions
M	represents	an *integer* variable
A	represents	a *real* array
B	represents	an *integer* array
Q	represents	a *Boolean* expression
S	represents	a *string*

Real

abs (X)
exp (X)
ln (X)
sqrt (X)
sin (X)
cos (X)
arctan (X)
tan (X)
arcsin (X)
arccos (X)
checkr (X)

Integer

entier (X)
sign (X)
address (A)
size (A)
range (A, I)
lowbound (A, I)
storemax
checki (I)

Boolean

buffer (I, S)
check B (Q)

Input

read Z, M, ...
reader (I)
instring (S, M)
advance (I)

Output

print X, I, S, ...
digits (I)
scaled (I)
freepoint (I)
aligned (I, J)
sameline
prefix (S)
leadzero (S)
grouping (I)
special (I)
punch (I)
outstring (S, M)
checks (S)

Control

wait
restart
stop
dump
precompile

Machine code

elliott (F1, F2, A1,
b, F3, F4, A2)

803 ALGOL HARDWARE REPRESENTATION

803		ALGOL	803		ALGOL
less	for	<	or	for	∨
gr	for	>	not	for	¬
lesseq	for	≤	£	for	‘
greq	for	≥	?	for	’
noteq	for	≠	*	for	×
equiv	for	≡	**	for	↑
and	for	∧	div	for	÷
			@	for	$_{10}$

NOTES

TABLES OF BINARY

The purpose of these tables is to assist in the

1. Select the highest multiple of 64 less than (or equal
2. Set the first (left-hand) 7 buttons to the binary
3. Set the last (right-hand) 6 buttons to the binary

TABLE A

Multiple of 64	*Binary equivalent*	*Multiple of 64*	*Binary equivalent*	*Multiple of 64*	*Binary equivalent*
0	0000000	2048	0100000	4096	1000000
64	0000001	2112	0100001	4160	1000001
128	0000010	2176	0100010	4224	1000010
192	0000011	2240	0100011	4288	1000011
256	0000100	2304	0100100	4352	1000100
320	0000101	2368	0100101	4416	1000101
384	0000110	2432	0100110	4480	1000110
448	0000111	2496	0100111	4544	1000111
512	0001000	2560	0101000	4608	1001000
576	0001001	2624	0101001	4672	1001001
640	0001010	2688	0101010	4736	1001010
704	0001011	2752	0101011	4800	1001011
768	0001100	2816	0101100	4864	1001100
832	0001101	2880	0101101	4928	1001101
896	0001110	2944	0101110	4992	1001110
960	0001111	3008	0101111	5056	1001111
1024	0010000	3072	0110000	5120	1010000
1088	0010001	3136	0110001	5184	1010001
1152	0010010	3200	0110010	5248	1010010
1216	0010011	3264	0110011	5312	1010011
1280	0010100	3328	0110100	5376	1010100
1344	0010101	3392	0110101	5440	1010101
1408	0010110	3456	0110110	5504	1010110
1472	0010111	3520	0110111	5568	1010111
1536	0011000	3584	0111000	5632	1011000
1600	0011001	3648	0111001	5696	1011001
1664	0011010	3712	0111010	5760	1011010
1728	0011011	3776	0111011	5824	1011011
1792	0011100	3840	0111100	5888	1011100
1856	0011101	3904	0111101	5952	1011101
1920	0011110	3968	0111110	6016	1011110
1984	0011111	4032	0111111	6080	1011111

EQUIVALENTS

setting of binary addresses on the word generator.

to) the required address, and work out the difference (if any).

equivalent of the multiple, working from Table A.

equivalent of the difference, working from Table B.

Multiple of 64	*Binary equivalent*
6144	1100000
6208	1100001
6272	1100010
6336	1100011
6400	1100100
6464	1100101
6528	1100110
6592	1100111
6656	1101000
6720	1101001
6784	1101010
6848	1101011
6912	1101100
6976	1101101
7040	1101110
7104	1101111
7168	1110000
7232	1110001
7296	1110010
7360	1110011
7424	1110100
7488	1110101
7552	1110110
7616	1110111
7680	1111000
7744	1111001
7808	1111010
7872	1111011
7936	1111100
8000	1111101
8064	1111110
8128	1111111

TABLE B

Difference	*Binary equivalent*	*Difference*	*Binary equivalent*
0	000000	32	100000
1	000001	33	100001
2	000010	34	100010
3	000011	35	100011
4	000100	36	100100
5	000101	37	100101
6	000110	38	100110
7	000111	39	100111
8	001000	40	101000
9	001001	41	101001
10	001010	42	101010
11	001011	43	101011
12	001100	44	101100
13	001101	45	101101
14	001110	46	101110
15	001111	47	101111
16	010000	48	110000
17	010001	49	110001
18	010010	50	110010
19	010011	51	110011
20	010100	52	110100
21	010101	53	110101
22	010110	54	110110
23	010111	55	110111
24	011000	56	111000
25	011001	57	111001
26	011010	58	111010
27	011011	59	111011
28	011100	60	111100
29	011101	61	111101
30	011110	62	111110
31	011111	63	111111

POWERS OF 2 IN DECIMAL

2^n	n	2^{-n}
2	1	·5
4	2	·25
8	3	·125
16	4	·062 5
32	5	·031 25
64	6	·015 625
128	7	·007 812 5
256	8	·003 906 25
512	9	·001 953 125
1 024	10	·000 976 562 5
2 048	11	·000 488 281 25
4 096	12	·000 244 140 625
8 192	13	·000 122 070 312 5
16 384	14	·000 061 035 156 25
32 768	15	·000 030 517 578 125
65 536	16	·000 015 258 789 062 5
131 072	17	·000 007 629 394 531 25
262 144	18	·000 003 814 697 265 625
524 288	19	·000 001 907 348 632 812 5
1 048 576	20	·000 000 953 674 316 406 25
2 097 152	21	·000 000 476 837 158 203 125
4 194 304	22	·000 000 238 418 579 101 562 5
8 388 608	23	·000 000 119 209 289 550 781 25
16 777 216	24	·000 000 059 604 644 775 390 625
33 554 432	25	·000 000 029 802 322 387 695 313
67 108 864	26	·000 000 014 901 161 193 847 656
134 217 728	27	·000 000 007 450 580 596 923 828
268 435 456	28	·000 000 003 725 290 298 461 914
536 870 912	29	·000 000 001 862 645 149 230 957
1 073 741 824	30	·000 000 000 931 322 574 615 479
2 147 483 648	31	·000 000 000 465 661 287 307 739
4 294 967 296	32	·000 000 000 232 830 643 653 870
8 589 934 592	33	·000 000 000 116 415 321 826 935
17 179 869 184	34	·000 000 000 058 207 660 913 467
34 359 738 368	35	·000 000 000 029 103 830 456 734
68 719 476 736	36	·000 000 000 014 551 915 228 367
137 438 953 472	37	·000 000 000 007 275 957 614 183
274 877 906 944	38	·000 000 000 003 637 978 807 092
549 755 813 888	39	·000 000 000 001 818 989 403 546
1 099 511 627 776	40	·000 000 000 000 909 494 701 773

SOME USEFUL CONSTANTS

π	= 3·141 592 653 590	$1/\pi$	= 0·318 309 886 184
$\log_{10} e$	= 0·434 294 481 903	$\log_e 10$	= 2·302 585 092 994
$\log_{10} 2$	= 0·301 029 995 664	e	= 2·718 281 828 459
$\sqrt{2}$	= 1·414 213 562 373	$\sqrt{3}$	= 1·732 050 807 569
1 radian	= 57·295 779 513 082°	1°	= 0·017 453 292 520 radian

803 TELECODE

			Character	
Binary	Decimal	Tape Punching	Figure Shift	Letter Shift
00000	0	·	bl ●	bl ●
00001	1	· o	1	1 A
00010	2	· o	2	2 B
00011	3	· oo	*	3 C
00100	4	·o	4	4 D
00101	5	·o o	$ or &	5 E
00110	6	·oo	=	6 F
00111	7	·ooo	7	7 G
01000	8	o·	8	8 H
01001	9	o· o	' ●	9 I
01010	10	o· o	,	10 J
01011	11	o· oo	+	11 K
01100	12	o·o	: ●	12 L
01101	13	o·o o	-	13 M
01110	14	o·oo	.	14 N
01111	15	o·ooo	%	15 O
10000	16	o ·	0	16 P
10001	17	o · o	(	17 Q
10010	18	o · o	)	18 R
10011	19	o · oo	3	19 S
10100	20	o ·o	? ●	20 T
10101	21	o ·o o	5	21 U
10110	22	o ·oo	6	22 V
10111	23	o ·ooo	/	23 W
11000	24	oo·	@ ●	24 X
11001	25	oo· o	9	25 Y
11010	26	oo· o	£	26 Z
11011	27	oo· oo	fs ●	fs ●
11100	28	oo·o	sp	sp
11101	29	oo·o o	cr ●	cr ●
11110	30	oo·oo	lf ●	lf ●
11111	31	oo·ooo	ls ●	ls ●

Notes This code also applies to the line printer where a sixth bit (value 32) is 0 for figure shift and 1 for letter shift.

● *On the line printer this character acts as space*

FOR SCIENTIFIC APPLICATIONS

ELLIOTT-AUTOMATION COMPUTERS LIMITED
ELSTREE WAY . BOREHAMWOOD . HERTS
Telephone: ELSTREE 2040

FOR COMMERCIAL APPLICATIONS

NCR ELECTRONICS

NATIONAL CASH REGISTER COMPANY LTD.
206-216 MARYLEBONE RD., LONDON, N.W.1
Telephone: PADdington 7070

CATALOGUE No. 801

JULY 1960