National

ELLIOTT

# 803 B

# 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 16 | 380 |
| Paper Tape Station | 80 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 | 55 |
| Battery charger | 30 by 22 by 19 | 200 |

# 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 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 | 340 cards/min. |
| Card output | 100 cards/min. |
| 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 $-1 \leqslant x < +1$, with two's complement notation for negative numbers, and standard floating-point form is $x = a.2^b$ such that $-1 \leqslant a < -\frac{1}{2}$ or $a = 0$ or $\frac{1}{2} \leqslant a < 1$ and $-256 \leqslant b < 256$. 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 programme 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-bit 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.

## 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

| | | | |
|---|---|---|---|
| $\pi$ | = 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

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

## 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

| a′ | n′ | Code | Time |
|---|---|---|---|
| a | n | 00 | |
| −a | n | 01 | |
| n+1 | n | 02 | |
| a & n | n | 03 | 576 |
| a+n | n | 04 | |
| a−n | n | 05 | |
| 0 | n | 06 | |
| n−a | n | 07 | |

### GROUP 1

| a′ | n′ | Code | Time |
|---|---|---|---|
| n | a | 10 | |
| −n | a | 11 | |
| n+1 | a | 12 | |
| a & n | a | 13 | 576 |
| a+n | a | 14 | |
| a−n | a | 15 | |
| 0 | a | 16 | |
| n−a | a | 17 | |

### GROUP 2

| a′ | n′ | Code | Time |
|---|---|---|---|
| a | a | 20 | |
| a | −a | 21 | |
| a | n+1 | 22 | |
| a | a & n | 23 | 576 |
| a | a+n | 24 | |
| a | a−n | 25 | |
| a | 0 | 26 | |
| a | n−a | 27 | |

### GROUP 3

| a′ | n′ | Code | Time |
|---|---|---|---|
| n | n | 30 | |
| n | −n | 31 | |
| n | n+1 | 32 | |
| n | a & n | 33 | 576 |
| n | a+n | 34 | |
| n | a−n | 35 | |
| n | 0 | 36 | |
| n | n−a | 37 | |

### GROUP 4

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

(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 specfy the function.

## GROUP 5

| | | | |
|---|---|---|---|
| Halve, double-length, N times | 50 | N | 576+288N |
| Right shift **a** N times. Clear AR | 51 | N | 576+288N |
| Multiply (double-length product) | 52 | N | 12096−288**y** |
| Multiply (single-length product). Clear AR | 53 | N | 12384−288**y** |
| (Multiplication time depends on the number of consecutive ones or zeros, **y**, at the most significant end of the multiplier.) | | | |
| Double, double-length, N times | 54 | N | 576+288N |
| Double **a** N times. Clear AR | 55 | N | 576+288N |
| Divide (double-length dividend, single-length quotient). Clear AR | 56 | N | 12096 |
| Read AR to accumulator | 57 | | 576 |

## GROUP 6

| | | | |
|---|---|---|---|
| **a**+**n** in floating-point mode | 60 | N | 864 |
| **a**−**n** in floating-point mode | 61 | N | 864 |
| **n**−**a** in floating-point mode | 62 | N | 864 |
| **a**×**n** in floating-point mode | 63 | N | 4896 |
| **a**÷**n** in floating-point mode | 64 | N | 9792 (max.) |
| Convert 39 bit integer **a** to standard floating-point form | 65 | 4096 | 576 |
| (Functions 66 and 67 are not used) | | | |

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

## GROUP 7

| Function | Code | | Time |
|---|---|---|---|
| Read from word generator to accumulator | 70 | 0 | 576 |
| Write the address of this instruction | 73 | N | 576 |
| **TAPE IN** | | | |
| Read first tape reader to accumulator | 71 | 0 | 576 |
| Read second tape reader to accumulator | 71 | 2048 | |
| **TAPE/PRINT OUT** | | | |
| Punch N on first punch | 74 | N | 576 |
| Punch N on second punch | 74 | 2048+N | |
| Print N on teleprinter (PTS 2A only) | 74 | 4096+N | |
| (Reader, punch and teleprinter channels include busy line facilities) | | | |
| **CARD IN** | | | |
| Read card input control word; prepare to read card | 76 | 512 | 864 |
| Read card to store locations N to N+79 | 77 | N 175 | |
| **CARD OUT** | | | |
| Read card output control word; prepare to punch card | 76 | 2561 | |
| Transfer to card punch from store locations N to N+79 | 77 | N 12 | |
| (Actual punching takes 600 milliseconds per card) | | | |
| **FILM STORE** | | | |
| Read address of last block read or written | 75 | 1027 | 576 |
| Read handler control word to accumulator; prepare to read on handler 1, 2, 3 or 4 | 76 | 1024<br>1032<br>1040<br>1048 | 864 |
| Read handler control word to accumulator; prepare to write on handler 1, 2, 3 or 4 | 76 | 1025<br>1033<br>1041<br>1049 | 864 |
| Read handler control word to accumulator; prepare to search on handler 1, 2, 3 or 4 | 76 | 1026<br>1034<br>1042<br>1050 | 864 |
| Read, write or search as prescribed by 76 instruction | 77 | N 204 | |

# 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.

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 |

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.

## TABLE B

| *Multiple of 64* | *Binary equivalent* | *Difference* | *Binary equivalent* | *Difference* | *Binary equivalent* |
|---|---|---|---|---|---|
| 6144 | 1100000 | 0 | 000000 | 32 | 100000 |
| 6208 | 1100001 | 1 | 000001 | 33 | 100001 |
| 6272 | 1100010 | 2 | 000010 | 34 | 100010 |
| 6336 | 1100011 | 3 | 000011 | 35 | 100011 |
| 6400 | 1100100 | 4 | 000100 | 36 | 100100 |
| 6464 | 1100101 | 5 | 000101 | 37 | 100101 |
| 6528 | 1100110 | 6 | 000110 | 38 | 100110 |
| 6592 | 1100111 | 7 | 000111 | 39 | 100111 |
| 6656 | 1101000 | 8 | 001000 | 40 | 101000 |
| 6720 | 1101001 | 9 | 001001 | 41 | 101001 |
| 6784 | 1101010 | 10 | 001010 | 42 | 101010 |
| 6848 | 1101011 | 11 | 001011 | 43 | 101011 |
| 6912 | 1101100 | 12 | 001100 | 44 | 101100 |
| 6976 | 1101101 | 13 | 001101 | 45 | 101101 |
| 7040 | 1101110 | 14 | 001110 | 46 | 101110 |
| 7104 | 1101111 | 15 | 001111 | 47 | 101111 |
| 7168 | 1110000 | 16 | 010000 | 48 | 110000 |
| 7232 | 1110001 | 17 | 010001 | 49 | 110001 |
| 7296 | 1110010 | 18 | 010010 | 50 | 110010 |
| 7360 | 1110011 | 19 | 010011 | 51 | 110011 |
| 7424 | 1110100 | 20 | 010100 | 52 | 110100 |
| 7488 | 1110101 | 21 | 010101 | 53 | 110101 |
| 7552 | 1110110 | 22 | 010110 | 54 | 110110 |
| 7616 | 1110111 | 23 | 010111 | 55 | 110111 |
| 7680 | 1111000 | 24 | 011000 | 56 | 111000 |
| 7744 | 1111001 | 25 | 011001 | 57 | 111001 |
| 7808 | 1111010 | 26 | 011010 | 58 | 111010 |
| 7872 | 1111011 | 27 | 011011 | 59 | 111011 |
| 7936 | 1111100 | 28 | 011100 | 60 | 111100 |
| 8000 | 1111101 | 29 | 011101 | 61 | 111101 |
| 8064 | 1111110 | 30 | 011110 | 62 | 111110 |
| 8128 | 1111111 | 31 | 011111 | 63 | 111111 |

**FOR SCIENTIFIC APPLICATIONS**

ELLIOTT COMPUTING DIVISION
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